

Electricity price prediction

A comparison of machine learning algorithms

Master thesis

Øystein Wormstrand

May 16, 2011

Halden, Norway

 **Østfold University College**

Abstract

In this master thesis we have worked with seven different machine learning methods to discover which algorithm is best suited for predicting the next-day electricity price for the Norwegian price area NO1 on Nord Pool Spot. Based on historical price, consumption, weather and reservoir data, we have created our own data sets. Data from 2001 through 2009 was gathered, where the last one third of the period was used for testing. We have tested our selected machine learning methods on seven different subsets. We have used the following machine learning algorithms: model trees, linear regression, neural nets, RBF networks, Gaussian process, support vector machines and evolutionary computation. Through our experiments we have found that a support vector machine using an RBF kernel has the best prediction ability for predicting the NO1 electricity price. We have made several interesting observations that can serve as a basis for further work in the topic of electricity price prediction for Nord Pool Spot.

Keywords: Electricity price prediction, Cubist, model trees, WEKA, linear regression, multilayer perceptrons, RBF network, Gaussian process, support vector machines, ADATE, Nord Pool Spot

Table of Contents

Abstract	i
1 Introduction	1
2 Background	3
2.1 The energy market	3
2.1.1 Nord Pool Spot	5
2.1.2 Different markets	7
2.1.3 Transparency issues	8
2.1.4 Evolving market	9
2.1.5 Market analysis	10
2.2 Machine learning tools and their methods	11
2.2.1 Predictive data mining and supervised learning	13
2.2.2 Classification And Regression Tree (CART)	13
2.2.3 Cubist	14
2.2.4 Weka	15
2.2.5 ADATE	22
2.3 Data	23
2.3.1 Price	24
2.3.2 Consumption	24
2.3.3 Weather	24
2.3.4 Reservoir	25
2.3.5 Ordered vs randomized data	26
2.3.6 Relative vs absolute data	26
2.3.7 Data pre-processing	26

2.4	Related work	28
3	Experiments	36
3.1	Experiment 1	37
3.1.1	Data set	38
3.1.2	Results	38
3.2	Experiment 2	41
3.2.1	Data set	42
3.2.2	Cubist	43
3.2.3	WEKA	45
3.3	Experiment 3	54
3.3.1	Cubist	54
3.3.2	WEKA	55
3.3.3	ADATE	61
4	Conclusion	63
5	Further work	67
	References	71
	List of figures	76
	List of tables	78
A	DVD	79
B	Results	80
B.1	Experiment 2	80
B.2	Experiment 3	97

Chapter 1

Introduction

The Nordic electricity market is an auction based market where prices are determined by supply and demand. Electricity sold on the spot market is traded on a day-ahead-basis, meaning that traders report to the power exchange how much energy they will sell or buy the next 12-36 hours. This implies that a company that has good analysis of the electricity market has an advantage in making successful trades. In this thesis we have set forth to predict the Norwegian electricity spot prices using different machine learning methods. We will primarily focus on the Norwegian price area NO1 which is the south-east region of Norway. This is a typical regression problem where we will be using several machine learning methods to research which algorithm is best at predicting the next-day price based on historical information. We have created our own data set based on data supplied by Nord Pool Spot¹, Montel², Meteorologisk Institutt³ and Statnett⁴. We will give an introduction to the Nordic power exchange and our data set later on in this paper. We have experimented with several machine learning methods, from linear regression to evolutionary computation, to learn which method is best suited for our data set. We have used several sub sets of our data set for our experiments. The first experiment was carried out by using Cubist and model trees. We then moved on to WEKA using linear regression, RBF network, Gaussian processes, neural nets and support vector machines. Finally, we have used an evolutionary computation program called ADATE⁵ and compared its results to our other selected methods. We will describe the tools and methods in the next chapter. The results will then be presented in its own chapter. Finally we will conclude on our

¹The Nordic power exchange

²Power news and data provider

³The national Norwegian weather institute

⁴<http://www.statnett.no/>

⁵Automatic Design of Algorithms Through Evolution

results and findings and propose new directions to further optimize electricity price prediction for Nord Pool Spot.

The inspiration for this project came from an earlier course at Oestfold University College, “Machine learning”. In this course we used decision trees, neural nets and evolutionary computation on a similar data set. This master thesis is an extension to the work previously done in that course and we will throughout the thesis refer to results gathered from earlier experiments in that course. This thesis will use a larger and more extensive data set and more machine learning methods. The previous data set only used price-, consumption- and weather data from four years on a daily resolution. The new data set uses data from nine years on an hourly and daily resolution with added information about the water reservoirs in Norway, Sweden and Finland.

There are several reasons why predicting the next-day prices are interesting. Obviously a trader who knows what the price will be the next day will benefit from this information. Having at hand the best prognosis for the next-day electricity prices will give an advantage in the trading situation, knowledge about when to trade, and how much to trade at a given hour. The producers will also benefit from having a good prognoses. If they know when the price will be high, they will know at which time they should produce and sell their energy or simply use the energy themselves. Another interesting aspect is the evolving smart grid technology. In the future, users will be connected to smart grids and a good electricity price prediction will help users of the smart grid to know when they e.g. should send electricity back to the grid and if they should minimize their own electricity consumption for any given hour. Consumers will in the future have greater control on how they use their energy and when they use it. So, there are many participants that can benefit from good electricity price predictions.

Even though we are looking at the spot market in this project, it might be possible to use the research from this report as an basis to create predictions for a longer period of time, e.g. one week ahead. You will then have a predictions that can be marketed towards participants in the future and forward markets⁶

In the next chapter we will give an introduction to the Nordic energy market and the tools and methods we have used in our project.

⁶Long term energy contracts ranging from one week to five years .

Chapter 2

Background

We have divided this chapter into four sections. First, we will have a look at the Nordic and Norwegian energy market. We will then present the different machine learning tools we have decided to use in our project, before we give a description of the input data we have gathered and how this data has been pre-processed. We will finish off this chapter by presenting some related work that have been done in the field of electricity price forecasting.

2.1 The energy market

The energy market is a volatile and non-stationary market. Electricity prices and consumption will change over time and there are no methods discovered which are able to exactly predict these changes. This section will describe some mechanisms in the Nordic energy market. We will describe the Nord Pool spot market which is the day-ahead market we are predicting with our regression models. We will also give an overview over general subjects that can have an impact on the price. Variables that will affect the electricity prices might be different for the short term and the long term market. The Nord Pool Spot market is based on bid and demand, implying that the main price driver for any given time is the amount of available energy and usage. ¹

Most of Norway's energy production is generated by hydro power. This means that most of the electricity in Norway comes from renewable sources. There are both positive and negative aspects with hydro power. Renewable energy sources are environmentally friendly and the CO₂ emittance from Norwegian electricity production is very low when compared to electricity produced by for instance coal plants. The downside is that we are very reliant upon weather and the fact that our

¹For a further description of the Nordic energy market you can visit <http://nordpoolspot.com>

water reservoirs constantly needs water inflow. If there is little precipitation, the reservoirs will be drained and the electricity prices will rise. In our project we will take into account not only the price, but also information about weather and the Nordic water reservoirs. Norwegian electricity is a popular product for the European continental market as well. European countries are bound by the Kyoto protocol, which states that the participating countries shall reduce their CO₂ emittance, making electricity produced in Norway a sought of product amongst countries that needs to lower their CO₂ emittance. Norway also exports energy in the form of gas and oil. Although there might be a connection between the electricity prices and the gas and oil prices, we have not taken these energy resources into consideration when doing this project.

In Norway there are big differences in the electricity price of the summer and the winter. In the summer months we use far less energy than in the winter since there is little need for extra heating. This means that we are able to store surplus energy as water in our mountain reservoirs. In the winter, we consume much more energy as the temperature drops. We will then use the stored capacities in our reservoirs. If there has been little precipitation in the areas of the reservoirs there will be a lot of tension in the electricity market since there is too little energy stored in the basins. This affects the electricity prices in the winter months. In the winter of 2010/2011 the Nordic electricity prices has been higher than previous years. A combination of a cold winter and little water flow to the water reservoirs must be partly blamed for this.

In Sweden there are several nuclear power plants producing energy for the Nordic market. The operational status of these power plants will also largely affect the electricity price. The 17. December 2009 we see an example of how an unplanned outage of a nuclear plant can affect the market. At this day the system price on Nord Pool Spot was 61.22EUR/MWh, but for the price areas affected by this outage the price reached a massive 251.04EUR/MWh². The highest hourly price that day was 1400EUR/MWh as opposed to around 40EUR/MWh for a regular hour in the same time period. This shows how fragile the market is, and how it can be affected by an unplanned nuclear power plant outage. Price peaks like the one just described are impossible to predict since there is no way to know when for instance a nuclear reactor must be shut down. In our work we have not taken into consideration the operational status of nuclear power plants.

There are many factors which will affect the day to day electricity price on Nord Pool Spot. We will now give a description of the Nordic energy exchange and describe how electricity is traded.

²<http://www.nordpoolspot.com/reports/areaprice/Post.aspx?i=1&p=5&u=0&g=0&a=0>

Date	SYS	NO1	NO2	NO3	NO4	NO5	DK1	DK2	FI	SE	EE	KT
23.02.11	68,76	69,72	66,27	70,61	70,61	69,72	60,05	62,75	70,61	70,61	41,69	
22.02.11	67,94	68,61	66,50	68,66	68,66	68,61	60,76	61,93	68,66	68,66	42,42	
21.02.11	68,36	68,93	65,72	70,17	70,17	68,89	61,34	64,37	71,74	70,17	44,37	
20.02.11	66,27	67,85	64,18	67,83	67,83	67,85	49,72	49,72	67,83	67,83	35,09	
19.02.11	65,91	67,31	64,32	67,10	67,10	67,31	50,72	50,72	67,10	67,10	35,78	
18.02.11	66,55	67,40	65,34	67,40	67,40	67,40	57,00	58,30	67,40	67,40	40,82	

Figure 2.1: The different price areas traded at Nord Pool Spot

2.1.1 Nord Pool Spot

In 1991 the Norwegian power market was deregulated which led to the start of the Nordic power exchange, Nord Pool[3]. Following the de-regularization of the Norwegian electricity market, the other Nordic countries also joined the Nordic power exchange. Today, Norway, Sweden, Denmark and Finland are connected to the Nordic power exchange which is one of Europe's largest power trading exchanges. The countries are divided into several price areas based on their transmission lines and grid quality. Norway has five price areas³, Denmark has two price areas, Finland has one price area and Sweden was divided from one into four price areas in 2011. Figure 2.1.1 shows the price areas that currently are traded at Nord Pool Spot. Only Nord Pool can carry out cross border electricity trading. Each country has a Transmission System Operator (TSO) which is responsible for the electricity grid⁴. The TSO is also responsible for the maintenance of the grid and that there at any given time is enough electricity in the grid to fill the demand.

In 2009, the el-spot turnover at Nord Pool Spot was 288 TWh⁵ which equals approximately a value of EUR 10.8 billion. About 72 percent of the consumption of electricity in Denmark, Sweden, Finland and Norway was traded via Nord Pool Spot⁶. This means that 28% was traded outside of Nord Pool Spot through bi-literal agreements. The Nordic energy exchange is not a monopoly and energy producers can freely trade their energy as long as there are no constraints in the transmission grid. The spot market is steadily evolving which makes it an interesting market to look into. Segmentation of price areas will affect future price prediction and as the TSOs expand their electricity grids, current congestions will be removed. We will now give an introduction to how electricity trading is conducted at Nord Pool Spot.

³Currently there are five price areas in Norway, but this number can differ from two to five depending on the transmission grid.

⁴Norway - Statnett, Sweden - Svenska Kraftnät, Denmark - Energinet.dk, Finland - Fingrid

⁵Terawatt hour = one billion watts

⁶Data is from Nord Pool Spot's annual report for 2009

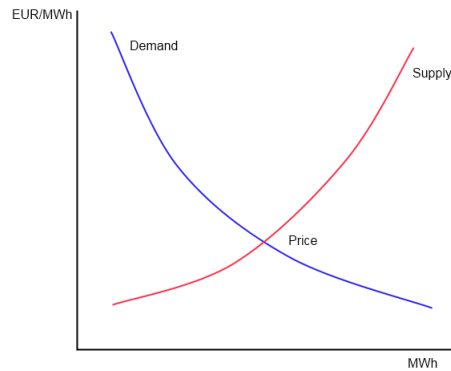


Figure 2.2: Illustration of supply and demand curve

Trading at Nord Pool Spot

Nord Pool Spot is a day-ahead auction market. People wanting to buy/sell energy must report this to Nord Pool at latest 12:00 the day before the energy is delivered to the grid. Bids are sent digitally to Nord Pool Spot which then creates a bid/demand curve based on purchase bids and sale offers 2.1.1. There are three bid methods used at Nord Pool Spot: hourly bids, block bids and flexible hourly bids. Participants must place their bids for the price area that the energy is produced or consumed. Meaning that if you have a hydro plant connected to the NO1 price area, you must also sell your electricity in this area. The price is set for the next 12 to 36 hours and is published at 14:00 every day. A system price and a price for each price area in the respective countries are calculated. The system price is a theoretical common price for the whole Nordic area. If there are no transmission constraints, all price areas will have the same price, the system price. Because of bottlenecks in the grid the price will often vary in the different price areas. Each 24 hour period spans from 00:00 to 23:00. When the prices have been calculated participants are notified on how much electricity they have sold/bought. This information is then sent to the respective TSO which calculates the balancing energy for the participants. In figure 2.1.1 we see an illustration of a typical supply and demand curve.

The following is an example of an actual trading situation at Nord Pool Spot: If for instance a retailer has his own energy production facility and knows that he will be needing 50 MWh at one hour, he can send a bid to Nord Pool stating that he will buy 50 MWh if the price for instance is

below 20 EUR/MWh. In the same bid he states that he will buy 30 MWh if the price is below 40EUR/MWh and that if the price is above 50 EUR/MWh he will not buy any energy at all, but rather produce all the energy himself. Using this example we can see how a producer can use a good price prediction to plan their production.

We do not have any information about bid and demands entered to Nord Pool Spot and it would be interesting to append this information to the data set. In a regular stock exchange, like Oslo Boers, information about current bids are constantly updated and traders can act upon this information. Since this information is not available on the energy exchange, the players will have to act on generated predictions for the market. In our transparency section we will write more about issues concerning available information and how this can affect the market and bidding situation.

2.1.2 Different markets

The energy bidding market is divided into three markets, the short term spot market, and the long term future and forward markets. Industry is a typical participant in the future and forward markets. The future market sells energy contracts for a period from one to six weeks while the forward market trades contracts up to five years ahead in time. By buying energy for a longer period of time, where the price is set for the whole period, the consumer will not be affected by sudden changes in the energy market. The spot market is more sensitive for day-to-day changes in the market, while the futures and forwards prices are more likely to be affected by political situations, infrastructure and international agreements such as the Kyoto Protocol⁷.

The spot market, which we have chosen to look into, is often referred to as a day-ahead market. This means that electricity is sold/bought up to 36 hours before it will be consumed. This is a more volatile market which is affected by sudden changes, e.g. outages in power plants, faults on the transmission grids and weather changes. A good example of how vulnerable the spot market is, is an incident at a Swedish nuclear plant. In December 2009 there was an unplanned outage at a Swedish nuclear plant which resulted in an enormous electricity price jump. In 2010 and 2011 the Nordic electricity prices have also been greatly influenced by planned maintenance at Swedish nuclear plants and Vattenfall not being able to get their reactors running at the scheduled time. The prices in the winter of 2010/2011 have also been quite high because of the cold winter and the lack of rain in the summer, resulting in low water inflow to the water reservoirs. These are all factors that will affect the next-day electricity prices and we have added some of these variables into our data set.

⁷An international environmental treaty with the goal of reducing the worlds carbon emittance

2.1.3 Transparency issues

As with any other market it is important with transparency. It is important that all the participants in the market have access to the same information. Privileged information available to some of the participants will give them an advantage in the trading situation. Available information is also important for making good predictions. If we are to create the best predictions for a market we need to know all the variables that will affect the price. Nord Pool Spot regards themselves for being a transparent exchange. Both Nord Pool Spot and Statnett reports urgent market messages, also called UMM. Through these messages participants involved in electricity trade are able to see if there are any planned/unplanned outages and failures on any of the power plants. The UMMs also reports hourly power system data such as production, consumption, cross border flows and regulating power market data. By analyzing UMMs, participants are able to see if there are any patterns that might influence the price. We have not taken regard to UMM information in our project. Information such as the UMMs are important in creating a transparent market. Although the market messages are available there might be information that yet is withdrawn from the market. Montel conducted a survey[24] where they asked several large gas producers if they were willing to give up information about planned outages to the other participants in the energy trading market. When asked this question the companies answered that they would for different reasons not give up this information. They regarded this information to be so closely tied up to their business that they would not share this information with other actors. Based on this survey one can assume that Norwegian energy producers/traders are reluctant to give up information that would make the energy market more transparent and thus equal for all its participants.

As mentioned earlier there is no information concerning the bid and demands entered into Nord Pool Spot. If we were to create a simulator of the stock exchange such information would be very useful for knowing when to buy/sell energy. In terms of next day price prediction, knowledge about previous bids/demands might affect the results from the prediction model.

Related to our project we see that a completely transparent market might benefit a model created by a machine learning method. Being able to have at hand all market related data will make it possible to optimize the data set and search for patterns within it. If we were able to add all information about planned outages and the effect these have on electricity prices we would most likely be able to generate even better predictions than we are doing in this project.

2.1.4 Evolving market

The available data will vary from year to year. Because of limitations in the transmission grid, the Norwegian TSO will divide Norway into price areas dependent on the grids capacity and expected usage. In 2011, Norway was divided from four to five price areas. Sweden will also be divided into four price areas in 2011 after a long time only having one price area. For Sweden there will not be any available historical data for all the coming price areas. This will have an impact on the Swedish price predictions since a system will need some time to adapt to the new structure of the new Swedish market. The coupling between the northern price areas in both Sweden and Norway will also be stronger since it will be easier to transmit energy between the northern areas. Traditionally the price in the Norwegian NO3 area has been higher than the other Norwegian price areas. The Swedish segmentation might have an affect on the electricity prices in NO3 since the coupling to Sweden becomes stronger and we will possibly see a normalization of the NO3 prices. The flow of energy in Sweden might be impacted by this as well, since the transfer from one price area to another will be restricted by the TSOs capacity. It will be interesting to see how this segmentation of the Swedish market will affect the Swedish and Nordic electricity prices.

It is also interesting to read a governmental report from Norway[6] where the price system in Norway is proposed changed. In this report the work group suggests that the Norwegian electricity market would benefit from even further segmentation. They propose to divide Norway into several price nodes where each node would be free of congestions and thus creating a more equal market for the consumers in Norway. These suggested changes in the price area structures might have an impact on systems trying to predict electricity prices. The data for each area will undergo structural changes from time to time and it will be interesting to see how for instance machine learning methods will cope with these changes.

The electricity grid is constantly changing, In Norway, the government has decided to build new power lines from eastern Norway to western Norway. This new power line will increase the electrical transfer capacity to the west and a major congestion in the Norwegian power grid will be removed. When these congestions are non existing the price areas will normalize themselves and we might see that we will more often have a common price for a larger part of Norway.

Another major change in the Norwegian electricity market that will happen in the near future is the implementation of a smarter grid and smart meters. This will further change how the market acts and there will possibly be even more actors in the electricity market that can affect the price setting. As the concerns are shifted towards the environment and the demand for energy efficient homes increases we will see a change in the electricity grid functionality[18]. A lot of research is

being done on how we can benefit from renewable energy sources and optimize the electricity grid. A result of this research is smart grids. In Norway we have so far only had a one way electricity flow. In the future, consumers will have the opportunity to become their own energy producers and send electricity back to the grid. Households can produce energy themselves by using e.g. wind mills, solar energy, energy stored in your electrical car and more. If a household produces more energy than it consumes, a smart grid offers the opportunity to transfers the excess energy back to the grid. The household will then be economically compensated for the electricity they send to the grid. In this example a system that operates the smart grid will benefit from having a good price prediction. Since all the energy that is sold/bought at Nord Pool Spot have to be reported to Nord Pool Spot 12-36 hours before the energy is consumed, a good prediction will help this system to sell the electricity when the price is the highest to maximize profit. At the same time the smart grid can buy electricity for its households when the price is low. All this depends on having a good algorithm for electricity price prediction. In the future we will see that appliances can be remotely controlled for maximum energy efficiency. We are likely to see services targeted at the consumers and we believe a good price prediction will be a part of these services.

2.1.5 Market analysis

In this section we will do a short market analysis with focus on participants of the spot market whom would gain by having a good electricity price prediction. If the results from our project proves to be equal or better than predictions already existing, we will have a good foundation for bringing our research to the next step which would be to either create an analysis tool for the traders at Nord Pool Spot or merge our model with an already existing analysis provider. As mentioned earlier, Nord Pool Spot traded a total of 288TWh in 2009 equaling about EUR 10.8 billion. Looking at these numbers we realize the spot market is very large and participants of the market should have a great interest in a good price prediction tool.

Currently there are 324 participants trading on the Nord Pool Spot markets⁸. Mainly the participant list consists of electricity producers like Oestfold Energi AS⁹ which is a local electricity provider in our area with a total capacity of 2.216TWh or 2216000MWh. If they were to sell all their energy on the spot market with at a mean price of EUR 40MWh/year their electricity assets would be worth NOK 88.64 million. In our case if we were to create a prediction which is merely 0.5% better than the estimates they are already using, they would profit NOK 443200/year. This

⁸<http://www.nordpoolspot.com/about/Participant-list---Elspot2/>

⁹<http://www.ostfoldenergi.no/>

tells us that even the smaller electricity providers would substantially increase their profit and benefit from using a better prediction method than they are already using. When we consider that Oestfold Energi only provides 0.0076% of the energy traded at Nord Pool Spot we are safe when assuming that a good prediction method would be of interest to the market.

Since the de-regularization of the electricity market, the need for a good market analysis has been present and there are several analysis providers that are playing the market. Some of the major actors in the Nord Pool Spot market are Markedskraft¹⁰, Montel¹¹ and Point Carbon¹².

The question is how good should the predictions be to be competitive in the market? This is a difficult question to answer since we do not know the prediction margins that other companies are offering. Through the work of Aggarwal et al.[1] we have seen that our predictions are not poorer than similar research done in earlier work. We have written about this report in our related work section. And if this is the scenario, then it is possible that the market will be positive to a new analysis tool that can provide them with a good price forecast.

In our related work section we have presented some previous projects that have worked on price prediction for electricity markets. There are few reports that have analyzed the Nord Pool Spot market, but we have gathered some of the results and will be comparing them to ours.

We will now go on with a presentation of the different machine learning tools we have used and the algorithms we have used from each tool kit.

2.2 Machine learning tools and their methods

In this section we present the machine learning tools we have used and the methods and algorithms we have used from each tool kit. There are a vast range of methods and tools to choose from and we have selected a few of them in this project. We will present each of the tool kits we have used and also describe which machine learning methods we have utilized from the different tool kits. The tool kits will be presented in the order that we tested them. In our experiment chapter we will compare the results from the different models and see how good they perform on our data sets. Our first choice was to use model trees with a program called Cubist. We then we moved on to a more comprehensive tool kit called WEKA. Finally, we used ADATE. We have also given a short introduction to two models popularly used in statistics for electricity price prediction. These two models are often referred to in related literature and we find it important that we also give an

¹⁰<http://markedskraft.no/>

¹¹<http://www.montelpowernews.com/>

¹²<http://www.pointcarbon.com/>

introduction to these two models, even though we have not used them ourselves in this project.

ARIMA and GARCH

Electricity price prediction is done by several research communities. We have looked into price prediction from a computer science point of view, but price prediction is more often used in economical studies. They use different models than machine learning models and in this section we have outlined the two most used models we have found in related work. We will have a look at the often referred to ARIMA model and we will finish this section by having a look at the GARCH model. Usually models used for price forecasting are univariate, meaning that the model only have one input variable. This means that only historical prices are used for input. In our models we have several input attributes meaning that we are working with multivariate models. In this section we will look at univariate ARIMA and GARCH.

Autoregressive integrated moving average (ARIMA) is a popular method used in statistics to analyze time series and is often used in electricity price forecasting. ARIMA is used for analyzing trends within the data and it is the trend that is important, the correlation between the values. ARIMA is often used for forecasting future values from historical time series. The methodology used is often referred to as the Box and Jenkins methodology[4], named after the statisticians George Box and Gwilym Jenkins. ARIMA models are summarized as $ARIMA(p, d, q)$ where p is the autoregressive parameters, d is the number of differencing passes and q is the moving average parameter. The input data for the ARIMA model needs to be stationary meaning it should have a constant mean, variance and autocorrelation through time. If the data are non-stationary the predictions will become unreliable. Log transforming is often used to transform the data to stabilize the variance. After the data are optimally differenced, the autoregressive and moving average parameters are estimated. These parameters are estimated by finding the minimum of the sum of the squared residuals by using a quasi-Newton method¹³. The estimated parameters are used in the last stage of the ARIMA process which is the forecasting of future values. ARIMA has the ability to add seasonal changes into its models which makes it good for processing longer time series.

Generalized autoregressive conditional heteroskedacity(GARCH) is another model popularly used by statisticians in electricity price forecasting. GARCH is an ARMA (Autoregressive moving average) model where the error variance has been assumed, meaning that the real value minus the forecasted value does not have a zero mean and a constant variance as with the ARIMA model[12]. This makes a GARCH model more adaptive to volatile changes in the time series such

¹³An algorithm for finding the local maxima and minima of functions

as price spikes. A homoskedastic model such as the ARIMA model has a constant variance while the GARCH model has different variances, hence heteroskedastic¹⁴.

2.2.1 Predictive data mining and supervised learning

Predictive data mining combines statistics, gathered data and machine learning methods. By using the increasingly growing computational power of computers and utilizing it on data stored in databases we are able to analyze and find patterns in the data. A machine learning method can help us create predictions based on input data and thus increase our knowledge in several fields. In medicine, for instance, predictive data mining can be used to aid researchers discover which variables are important in developing a certain disease. Banks can use stored information about their customers to discover an abnormal account usage to prevent fraudulent activity. In our scenario we have used predictive data mining to see whether there are any patterns in historical information related to the electricity trade that can be used to create good predictions about the next-day electricity price.

We use supervised learning, meaning that we supply the machine learning method with a training data set consisting of both input and output variables as opposed to unsupervised learning which only has input variables. Based on the input data, the algorithms create a regression function that can be used to create the prediction. We hope that in our work on this project we will be able to find an algorithm that is well suited to make predictions for the electricity prices on Nord Pool Spot.

2.2.2 Classification And Regression Tree (CART)

The CART methodology was developed in the 1980s by Breiman et al and presented in their “Classification and regression trees” publication[5]. They presented how data could be processed and divided into classification and regression trees. Their work is closely related to the work done by Morgan and Sonquist in 1963 and their automatic interaction detection(AID)[30]. Their CART methodology was supported by good analysis and probability theory as to why CART would work so well.

CART decision trees are created from the root node where data is split into branches and nodes. This is processed until all the data has been classified. Then, the tree is pruned back to the root via a cost-complexity pruning method. The pruning of the tree will then remove variables which are not found necessarily for the results. It is based on this methodology which Quinlan has created his M5

¹⁴From the the greek words hetero(different) and skedasis(dispersion)

model tree algorithm which we will present later in this chapter.

2.2.3 Cubist

Cubist is a program developed by Rulequest and Ross Quinlan¹⁵. Cubist uses a modified regression tree algorithm, the M5 algorithm, to build a rule based numerical model. Regression trees were introduced by Breiman et al.[5]. The model tree induction used by Cubist was first introduced by Quinlan in 1992[34]. The process used to create the model tree is called “binary recursive partitioning”. This is the process where the data is split into branches.

Cubist outputs a set of rule-based predictive models which makes it easy to interpret the results. Speed and ease of interpretation are two major benefits of the Cubist tool kit. It is also possible to prune the number of rules to see whether this will affect the results. Cubist will also give an overview of the variables that are most frequently used when making a prediction. We hope that this information can help us discover which variables we should use in our final data set. In our next section we will give an introduction to the model tree algorithm used by Cubist.

Model tree

A regression tree differs from the decision tree in that it has numeric values rather than classes at its leaf nodes[25]. Cubist uses the M5 algorithm which, as opposed to a regular regression tree, has the ability to implement multivariate linear models in its leaf nodes. Benefits of the model tree is that it can extrapolate values in its prediction, meaning that the predicted value can be outside the values of the training set. In a regular regression tree the output will always be in the range of the values in the training set since a mean value is calculated for the instances in each node. The model tree is smaller than the regression tree and easier to comprehend, but even so, the results using a model tree will be better than that of a regression tree[25] because of its ability to use local linear regression on its leaf nodes. In figure 2.2.3 we see a model tree showing CPU performance data[34]. All the values are divided into its respective leaf node and for each leaf node we have a linear expression for the values in that node. By combining regression trees and local linear regression the model adapts better to the data and the predictive performance improves.

¹⁵<http://www.rulequest.com>

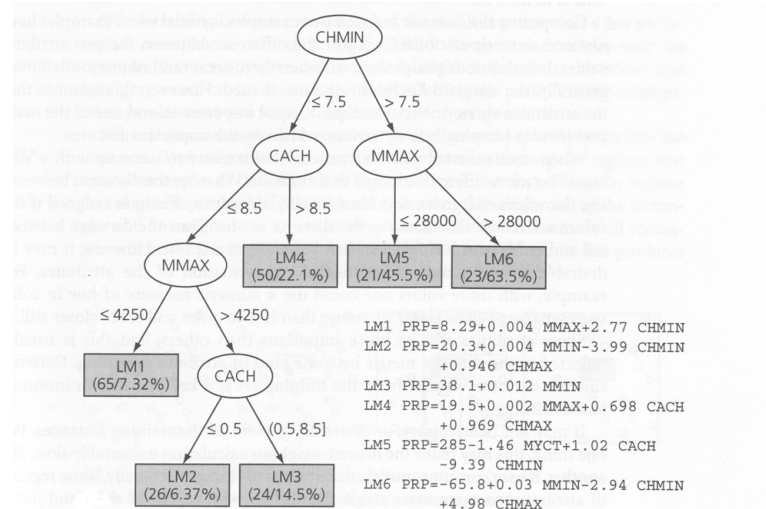


Figure 2.3: Example of a model tree taken from [34]

2.2.4 Weka

WEKA¹⁶ is a machine learning toolkit developed at the University of Waikato in New Zealand¹⁷. WEKA implements a wide range of machine learning methods that can be used for regression problems such as ours. We have selected several methods that we have used on our data sets. In the next sections we will describe our methods of choice that we used from the WEKA toolkit.

Linear regression

Linear regression analyzes the relationship between several input variables. In linear regression one tries to fit a straight line so that it matches the input variables as good as possible. Figure 2.2.4 shows an example of a linear regression model. If you have a good fitted linear line you can use this information to predict future values of the output variable. By applying weights to the variables in the input data one will fit a line that best matches the data. The weights are used to minimize the sum of the squared differences between the actual and predicted values[34].

WEKA performs standard least-squares linear regression and implements ridge regression[34], also known as Tikhonov regularization. Ridge regression is used to solve problems that are not well-posed, meaning that they will have poor chance of being solved by a stable algorithm.

¹⁶<http://www.cs.waikato.ac.nz/ml/weka/>

¹⁷<http://www.waikato.ac.nz/>

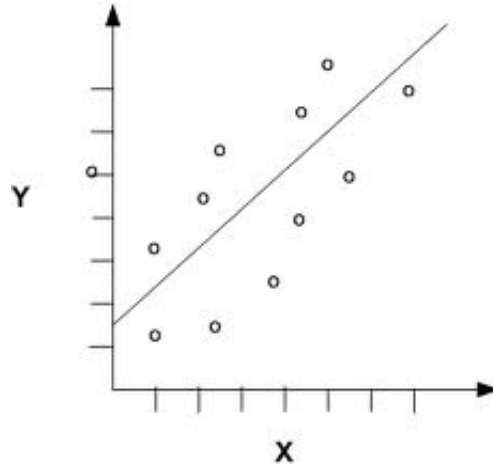


Figure 2.4: Illustration of a linear regression model

In linear regression the predicted value for the first instances's class can be written as

$$\sum_{j=0}^k w_j a_j^{(1)}$$

where $a^{(1)}$ is the attribute and w is the weight. This only gives the predicted value for one attribute, but the process must be repeated on all the attributes. Linear regression tries to minimize the sum of the squares for the difference between the actual value and the predicted value as shown above. The sum of the squares of difference is written as

$$\sum_{i=1}^n \left(x^i - \sum_{j=0}^k w_j a_j^{(i)} \right)^2$$

where x is the actual value. Based on this formula, we will minimize the squared difference between the actual and predicted value and create a linear line that separates the data optimally.

Linear regression models are a fast and basic approach for classifying numeric data, but if the data spread is too wide, the model will have problems calculating the proper weights and hence not create a good prediction. If we compare a pure linear regression model to the model tree described earlier, which also utilizes linear regression in its leaf nodes, the model tree will most likely perform better. This is due to the fact that in the model tree the data has been split into smaller sections in the nodes and thus it will be easier for a linear model to adapt itself to the data in that node.

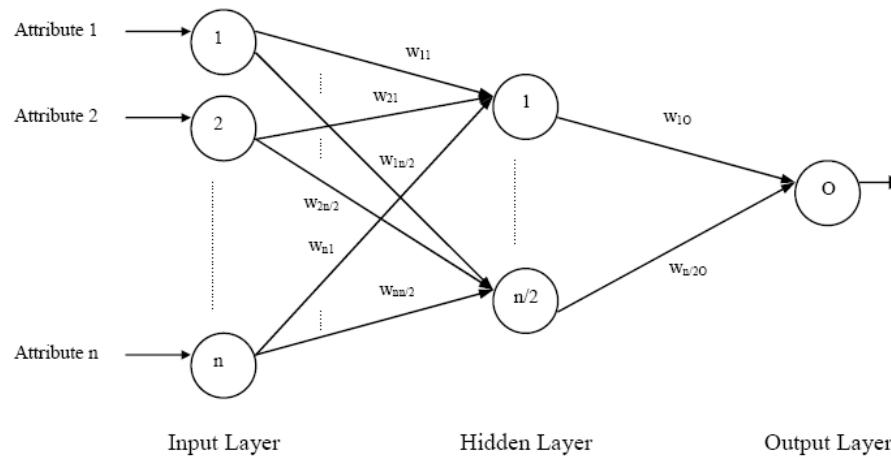


Figure 2.5: Illustration of a neural network with n input, one hidden and one output layer.

Multilayer perceptron

Opposed to the linear regression approach, we have a non-linear method known as multilayer perceptron (MLP). MLP is a neural network using backpropagation¹⁸ which is a learning method for the neural network. Through the use of the mathematical optimization algorithm, *gradient descent*, the weights passed with the input variables are optimized. When the problem at hand can not be solved with a linear method a good solution can be to use a neural net such as the multilayer perceptron. The neural net uses a hidden layer(s) with perceptrons, often compared to human brain neurons. The input data is connected to all the perceptrons in the hidden layer, and the perceptrons combined, yields the output. A constant bias is used to adjust the input data and the output from the hidden layer. Figure 2.2.4 shows a simple neural network with one input layer with n input attributes, one hidden layer with two perceptrons (neurons) and one output layer with one output value. This architecture is similar to the architecture we have used in our MLP experiments.

During the training of the neural net the weights to each neuron is adjusted after the data is passed and the result from the net is compared to the actual value it is trying to predict. The error in the entire output is given by

$$\epsilon(n) = \frac{1}{2} \sum_j e_1^2(n)$$

¹⁸Error values are propagated back through the net and new weights are calculated

and by using gradient descent the change in each weight will be

$$\Delta w_j i - \eta \frac{\partial \epsilon(n)}{\partial v_j(n)} y_i(n)$$

where y_i is the output of the previous neuron and η is the learning rate. This process will repeat itself until the weights are optimally adjusted and the prediction is as close as possible to the actual value being predicted. A problem with using gradient descent is that it only finds a local minima. A neural net might have several minimas and gradient descent might not be able to find the best suited local minima. The learning rate, which is a small constant, tells how large steps gradient descent will take in its process of finding the local minima. If the learning rate is too high, gradient descent might miss a local minima and if it is too low it will take too long to find the local minima. Therefore it is important to test with several learning rate constants to see which will give the best results. Another drawback of a neural net trained with backpropagation is overfitting. This might happen if the network is larger than it needs to be to solve the underlying problem.

Compared to the linear regression model, a neural net demands more computation resources to achieve its results. While the speed performance of the neural net is slower, the neural net will most likely yield better results than a linear model because of its ability to create nonlinear models based on the input data.

Gaussian Process

A Gaussian process (GP) is closely related to support vector machines which both are part of the “kernel machine” area in machine learning. In WEKA we have the opportunity to select between four kernels or covariance functions: the poly kernel, normalized poly kernel, RBF kernel and PUK kernel. We have given a closer description of these kernels in our support vector machines section. In our experiments we have only used the RBF kernel. A Gaussian process uses Bayesian inference, meaning that it calculates both a prior and posterior distribution over functions. The more observations we give our GP, the better it will perform. A GP is likely to perform poorer in areas with abnormal values since there are fewer observations to give an accurate prediction.

While a Gaussian *distribution* is specified by a mean vector and a covariance matrix, a Gaussian *process* is specified by a mean function, $m(x)$, and a covariance function, $k(x, x')$ [26]. A definition of a Gaussian process is given by *C.E. Rasmussen and C.K.I. Williams*[27]:

“A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.”

Gaussian processes are non-parametric and is fully specified by the mean and covariance function meaning that the calculations are done in the function space view, rather than the parametric weight space view[27]. A Gaussian process is better than for instance linear regression in handling volatile input data and has a better chance of adjusting itself to the occurrence of non-linearity.

A Gaussian distribution, or normal distribution, implements the fact that in data sets random variables are distributed approximately normally, or the central limit theorem[28] if you like. The Gaussian distribution is mathematically presented as

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x - \mu)^2}{2\sigma^2}}$$

where μ is the mean and σ^2 is the variance.

A Gaussian process is defined by its mean and covariance functions. These functions can be expressed as

$$m(x) = \mathbb{E}[f(x)]$$

for the mean function $m(x)$ and

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x')))]$$

for the covariance function $k(x, x')$. The Gaussian process $f(x)$ will then be given by

$$f(x) \sim GP(m(x), k(x, x'))$$

One problem when learning with Gaussian processes is to find suitable parameters to the covariance function. Another problem is limitations in the data set size because of the inversion of an $N \times N$ matrix.

Radial Basis Function Network

The Radial Basis Function Network (RBF) is in its simplest form a three layered feed forward neural network with one input layer, one hidden layer and one output layer[7]. It differs from an MLP in the way the hidden layer performs its computation. The connection between the input layer and the output layer is nonlinear, while the connection between the hidden layer and the output layer is linear. RBF networks are instance based, meaning that it will compare and evaluate each training

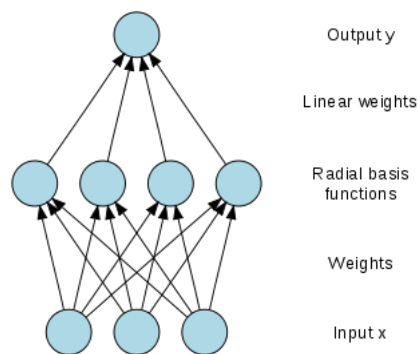


Figure 2.6: Illustration a Radial Basis Function network

case to the previous examined training cases. In an MLP all instances are evaluated once while in an RBF network the instances are evaluated locally[19]. Instance based methods use nearest neighbor and locally weighted regression methods. An RBF network can be trained more efficiently than a neural net using backpropagation since the input and output layer are trained separately.

The output of an RBF network is presented as

$$\rho(x) = \sum_{i=1}^N a_i p(\|x - c_i\|)$$

where N is the number of neurons in the hidden layer and c_i is the center vector for the neuron i , and a_i are the weights of the linear output neuron.

In figure 2.2.4 we see an illustration of a typical RBF network with three layers and one output node.

A disadvantage with an RBF network is that all the attributes are weighted the same[34], meaning that irrelevant attributes might lower the performance of the network. In regards to our experiments, the RBF network might favor our data set using the fewest attributes.

Support Vector Machines (SVM)

“In SVM the basic idea is to map the data x into a high-dimensional feature space F via a nonlinear mapping ϕ , and to do linear regression in this space (cf. Boser et al. (1992); Vapnik (1995)).”

Support Vector Machines is a blend of linear modeling and instance-based learning[34]. SVM

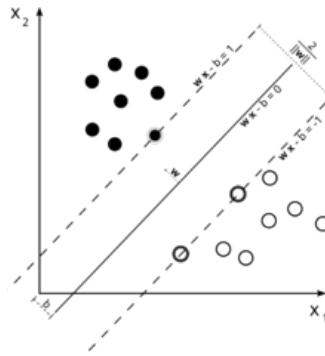


Figure 2.7: Illustration of data divided with a high margin

in its present form was mainly developed by AT&T Bell Laboratories in the 1990s[29]. Although SVM is based on the Generalized Portrait algorithm developed in the sixties, in the field of machine learning one can call this a fairly “new” approach to machine learning. A Support Vector Machine consists of two parts, firstly we have the kernel algorithm, often referred to as the “Kernel trick” and secondly, we have the optimizer algorithm. In WEKA SVM is utilized through the Sequential minimal Optimization for regression class (SMOreg). In this section we will give a short introduction to the SVM kernels and the optimizer we have used.

The kernel is used to divide non-linear data to high-dimensional space, thus making the data linearly separable[14]. By finding the optimal support vectors the kernel is able to divide the data. Optimally the data should be divided with a high margin. Figure 2.2.4 shows how data have been divided with a high margin. If we were to change the angle of the center line, the division would not be optimal and the margin would decrease. The values that are on the margin line are the support vectors.

The optimization algorithm is an improved version John Platt’s sequential minimal optimization (SMO)[23]. The SMO algorithm breaks down the results from the kernel trick into a two-dimensional sub-problem that can be solved analytically. We have used the improved smo algorithm offered by WEKA in all our support vector machine experiments.

The difference in the kernels lie in the way they separate the data for the optimizing algorithm. We used the following kernel algorithms on our data set:

Firstly we used the poly kernel which is represented by

$$K(x, y) = \langle x, y \rangle^p \text{ or } K(x, y) = (\langle x, y \rangle + 1)^p$$

We then moved on to using the normalized poly kernel

$$K(x, y) = \langle x, y \rangle / \sqrt{(\langle x, x \rangle \langle y, y \rangle)} \text{ where } \langle x, y \rangle = \text{PolyKernel}(x, y)$$

Then we tested the Gaussian Radial Basis Function kernel

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

And finally we tested the Pearson VII Universal Kernel

$$f(x) = \frac{H}{1 + \left[\left(\frac{2(x-x_0)\sqrt{2^{1/\omega}-1}}{\sigma}\right)^2\right]^\omega}$$

The differences between these kernels are how they split their data before the optimization algorithm calculates the prediction based on this split. A kernel that performs well on one data set does not necessarily perform well on another data set. This is the reason for testing several kernels.

2.2.5 ADATE

ADATE (Automated Design of Algorithms Through Evolution) is a system for automated programming developed by Roland Olsson at Oestfold University College. ADATE is the only evolutionary system we will use in our project and we will compare the results from ADATE with the results from our other methods. ADATE uses the same principles we find in biological evolution and its classification is based on the Linnean Taxonomy which divides evolved programs into kingdom, classes, order, family, genus and species[33]. Initially ADATE starts with an empty program and as it adds new individuals to the kingdom, the complexity of the programs increases and the kingdom expands. The process describing how ADATE works is shown in figure 2.2.5

First, ADATE selects an individual to be placed in the kingdom, then an individual is selected from the kingdom for expansion and a new individual is created using a compound transformation. The new individual is then inserted into the kingdom. This process is repeated indefinitely or until the user stops the process. Evolution is a slow process and it will go on forever, and that is also the case with ADATE. The longer you let a process run, the better it can become. Compared to our other methods that uses predefined algorithms to find a pattern, ADATE will create its own algorithms based on the problem to be solved. This makes ADATE perform slower than our other algorithms, but through evolution it might adapt itself better for the problem at hand and thus improve our

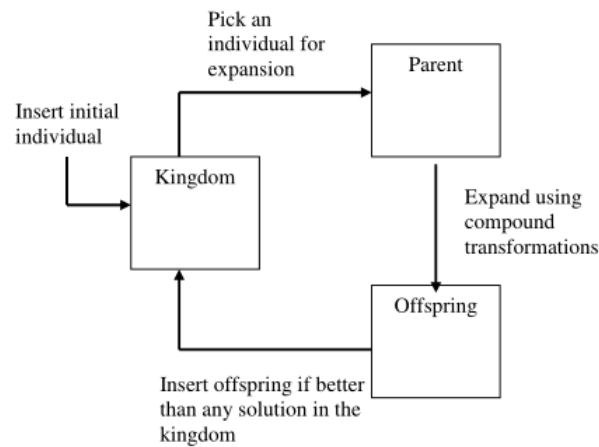


Figure 2.8: Figure of the ADATE search process as seen in the ADATE user manual

results.

ADATE uses six basic forms of transformations which are: replacement, replacement without making the individuals evaluation value worse, abstraction, case distribution, embedding and crossover. These transformations are the core of how a program will evolve using ADATE.

2.3 Data

In this section we will have a look at the basis of our data set and which attributes we have chosen to incorporate. The composition of a data set is very important when creating predictive models. It is important to use the “correct” data for the problem one wishes to solve. The data sources must be reliable and the data must be relevant. We have gathered data that we have found important to predict electricity prices. Knowing that we will use neural net and evolutionary computation on our data set we have also had to bear in mind that we must try to keep the number of input variables to a minimum in those experiments.

We have gathered data from several data sources, but common for all is that we have gathered data from 2001 through 2009. This will give us a descent amount of data to work with and should be sufficient for creating a good data set. We use 2/3 for training and 1/3 for testing, which gives 6 years for training and 3 years for testing.

The different subsets of our data set is described before each experiment. This section will only cover the data variables we have selected to use and how they have been pre-processed.

2.3.1 Price

We believe that the most important attribute in price prediction is the historical prices. We were given access to price data by Montel¹⁹ and we downloaded historical price data for all the areas through their Excel feeder. We have historical price data both in an hourly and daily resolution. We will then have the opportunity to see which resolution correlates best with the price we are trying to predict. We have downloaded price data for all the price areas in the Nord Pool Spot area. This means that we will use historical prices from four Norwegian, one Swedish, one Finnish and two Danish price areas. The prices are presented as Euro/MWh.

2.3.2 Consumption

We were granted access to Nord Pool Spots historical data and downloaded the consumption data from their servers. Consumption is closely related to the electricity prices and we should be able to find a correlation between price and consumption. We have consumption data from all the price areas on Nord Pool Spot on both an hourly and daily resolution.

2.3.3 Weather

Our weather data has been downloaded from The Norwegian Meteorological Institute through their web service at http://eklima.met.no/wsKlima/start/start_no.html. The temperature is very important when one looks at peoples electricity usage. When it is cold outside, consumers will use more energy to heat their homes. In Norway we primarily use electricity for heating, next to oil and firewood. We therefore expect to find a significant correlation between weather, electricity consumption and electricity prices. Our challenge has been to find consistent weather data from the whole time period from the same measuring station. We have selected two weather stations within the four price areas, except for the NO3 area where we only have one weather station. This means that we have historical data from seven weather stations. It has been an important factor that the at least one station in each area is situated in a densely populated area. Table 2.1 shows the weather stations that we have used data from. All of the weather stations have been operational and gathered hourly weather data for the whole time period in question.

¹⁹<http://www.montelpowernes.com>

Table 2.1: Weather stations used in this project.

Area	Station n ^o	Station name
NO1	18700	Oslo, Blindern
NO1	18950	Oslo, Tryvasshøgda
NO2	39040	Kjevik Lufthavn
NO2	50540	Bergen, Florida
NO3	71000	Steinkjær, Søndre Egge
NO4	82260	Bodø, Vågønes
NO4	90400	Tromsø, Holt

Weather values and resolution

All weather stations record a number of weather data, but not all stations record data on an hourly basis. The weather is a local phenomenon which means that two weather stations close to each other might record quite different values. In our data set we wanted to have weather readings from the same weather station, ensuring that the data gathered would be relevant throughout our whole time period. Each station records the maximum and minimum temperature for all hours, also known as TAX and TAN values. It is also possible to download the medium value, TAM, but this is an arithmetic value which has not been calculated for the whole period for all the weather stations. Only the TAX and TAN values were available. We have chosen to use the TAX value for each hour. The temperature changes within each hour will hopefully not fluctuate that much as to have large impact on our classifications.

Observations vs Prognosis

The downloaded data are observations from each station. Thus, we have not used weather forecasts which is the correct value to be used in a prediction situation. If we look at on-line weather information from the Norwegian Meteorological Institute, yr.no, we have found that the day-ahead forecast is very close to the observed value.

2.3.4 Reservoir

Since Norway in particular produces most of its energy from hydro power, we have chosen to also add information about reservoirs and their water inflow. This information has been downloaded from Statnett. Unfortunately information about the reservoirs are only given on a weekly resolution. This means that we have a very poor data resolution for the reservoirs and it will be interesting to

see whether the reservoirs have an impact on the price prediction.

2.3.5 Ordered vs randomized data

Our data set consists of historical time series data and because of this, it is important that we do not randomize our data in our experiment. If we let our machine learning methods randomize our data we would end up with a prediction where knowledge of the future were given in the input data. This would let the method use connections that simply have not occurred yet.

2.3.6 Relative vs absolute data

We have converted all our price and consumption data into relative values. By using relative values we achieve two benefits. First of all our results will be the actual percentage difference between input and output data, the MAE²⁰. If we used absolute values for our price data our results would be the absolute deviation in the prediction from the actual data. By using relative values we will have the percentage change from one day to another which means we will have a constant measure independent of the actual price. In 2001 electricity was sold for ca EUR20MWh while today the price is EUR60MWh. This means that the absolute price which electricity is sold for has increased three folds. We assume that the day-to-day change is more constant, and thus we are using relative values. In our earlier project in the machine learning course we also found that the results got improved by using the relative values as compared to the absolute values.

2.3.7 Data pre-processing

We downloaded our data from several sources which meant that our data came in different formats depending on the source. To adjust for these differences and to make sure the data matched in regards to the days they concerned, we had to pre-process the data in order to extract the wanted data and append it to our data set. For this purpose we created several programs in C# which made it possible to work with such large data amounts from a long time period. We got our consumption data from Nord Pool Spot, price data from Montel, weather data from wsKlima and reservoir data from Statnett. It was a formidable task working with multiple sources and different formats and adapting the data to the correct hour of the correct day.

The data we got from Nord Pool Spot was in flat comma separated files saved on a weekly basis and sorted by the participating countries. We had to extract the consumption values from

²⁰Will be described in the next chapter

these files and merge the information from each week into one large file with the values sorted in correct order by date. We did not have data for all the days in the period which meant that we created a function for creating empty values if there was no information present at that day. Ideally there should be information about all the days for the whole period, but some of this information was missing. When we had all the consumption data we converted the data into relative values. The predicted values is relative to the previous day, $R_1 = A_t/A_{t-1}$ (A =actual, t =time), while the previous daily values were relative according to their previous day. The first previous relative value was then $R_1 = A_{t-1}/A_{t-2}$ (R =Relative), the second $R_2 = A_{t-1}/A_{t-3}$ and the third previous value was $R_3 = A_{t-1}/A_{t-4}$. Thus the actual relative value which is being predicted is relative to its previous day while the previous relative values are all relative to the day before the actual day. By using this method we are able to see if there is a noticeable trend in the previous data. We have used the method to calculate the relative value on our price data as well.

We were given access to historical price data by Montel which were in Excel data sheets. This price information was easier to work with than the consumption data and we converted the absolute values to relative values according to the procedure described for the consumption data. Again there are a few days missing, but this is to be expected over such a large time span. We used the “datetime” object in C# to add the correct values for the correct day in our data set and to add an empty values if information about a day was missing.

The weather values has undergone few changes. We have used all the values as is and we have not calculated any relative change in temperature. Again we have used the “datetime” function in C# to match all the values to the corresponding date in the data set.

Reservoir prognoses are given on a weekly basis and have been adopted to match the rest of the data set. Reservoir prognoses has been entered into the database by expanding the the weekly prognoses to daily and hourly values. The prognosis value is the same throughout the week.

Through earlier work in our machine learning course we found that weather played an important role in determining the electricity prices. It is reasonable to assume that the temperature in dense populated areas will have a larger effect on electricity prices than the temperature in more rural areas. We have downloaded temperature information from several automatic weather measuring stations in all the Norwegian price areas.

It is always a challenge when working with large data sets like ours. We have spent much time assuring the credibility of our data and we have manually checked that the values are correct.

2.4 Related work

In this section we will describe several methods that have been used on similar problems dealing with predicting next-day prices. Through the years many methods have been used to foresee electricity prices. We have found and described a few that can be related to our problem. The authors writing these articles are often very content with their results and it will be interesting to compare their models to our models of choice. Although many articles concerning prediction of energy prices have been published, there might be research in this field which has not been publicly documented. A good prediction method will always be commercially interesting and thus not published for market reasons.

Although there has been done much work in the field of price prediction it seems that most studies only use price and consumption data for their data set. Several reports state the importance of importing weather into their dataset, but we have not seen any results based on weather input. Another important factor might be the reservoirs which is a main price driver for the Nordic electricity prices.

The background for writing this master thesis was the work we did in an earlier course at Oestfold University College. In that course we also predicted the electricity prices on Nord Pool Spot, but we used fewer machine learning algorithms and had less data at hand. It will be interesting to see whether our new choice of machine learning algorithms and the larger data set will improve the results from the previous course. We will now give a short description of the data set and algorithms we previously tested.

In that project we used historical data from four years. We gathered data about price, consumption and weather on a daily resolution. Weather data was gathered from all weather station within each price area and we then used the mean temperature value as measurement for each area. When we added historical price information and consumption we had the complete data set we used with our algorithms. The algorithms we used at that time was decision trees (c5.0), the Levenberg-Marquardt algorithm in MatLab and Roland Olssons ADATE. In that project we predicted both consumption and prices while in this thesis we are only looking at price prediction. Based on the results we got from that work, we were curious to see if it was possible to improve the results and thus making even better predictions for the electricity price. In table 2.2 we can see the best results we got from our previous work. We have left out the results from c5.0 since those results will not be comparable to our results because it uses decision trees which is not suited to tackle regression problems. In that project our best results was 3.35% MAPE with neural nets and 3.34% using an

evolutionary algorithm.

Table 2.2: Results for price prediction conducted in previous work

Area	Algorithm	MAPE
NO1	Levenberg-Marquardt	3.35%
NO1	ADATE	3.34%

The results from these experiments were quite good and it will be interesting to see whether a larger data set and other algorithms will perform even better.

In 2010 a master student, Sigbjørn Nome[21], with the Norwegian School of Economics wrote a thesis where he used ARIMA time series models to predict the electricity prices for the NO4 price area in Northern Norway. There are not many papers that explicitly are focused on the Nord Pool Spot price areas and therefore it is interesting to see the results gathered from this report. In his report he used historical price information for the NO4 price area and he removed the outliers since he wanted to base his prognosis on a regular functioning market without price peaks. He did his prognosis on one selected week in May and used ARIMA models for his prediction. By using his model he got an average hourly MAPE of 4.32% for that time period. As he stated the electricity prices for that week were very stable and there where no unforeseen factors that affected the price. He has not documented results for a larger time span and thus it will be difficult to directly compare his results with ours since we are making predictions for a longer period of time and the daily electricity price. Anyway, it is interesting to see results from other projects concerning Nord Pool Spot.

There are several popular techniques that are used for price forecasting. Auto Regressive Integrated Moving Average (ARIMA) is one of these models. A study performed by Javier Contreras et. al.[9] used ARIMA models to predict next-day prices on the Spanish and the Californian market. For the Spanish market they used data from one year, January 1. 2000 - December 31. 2000, and three weeks from this year was used to forecast and validate the performance of the ARIMA model. For the Californian market they only used data from January 1. 2000 to April 9. 2000 and one week for forecast and validation. It can be questioned whether this is enough data to create a good model. For their results they calculated a daily mean error for the 24 hours. Average errors in the Spanish market was around 10% and about 5% in the Californian market.

Another statistical model that have been used is the GARCH model (Generalized Autoregressive Conditional Heteroskedastic). A GARCH model consider the moments of a time series as variant, which means it will function better with volatile data. In another Spanish study by Reinaldo Garcia

et. al.[12] they used the GARCH model to predict prices in high volatility periods for the Spanish and the Californian market. For the Spanish market they used hourly data from 15 months while they used 12 months of data for the Californian market. The results when using the GARCH method was about 9% forecast error on both the Spanish and the Californian market.²¹

Using artificial neural networks is another popular approach for price prediction. Neural networks are adaptive systems that can recognize patterns and change its structure based on provided information. A neural network is well suited for a regression problem like next-day price prediction. This section will describe some cases where neural networks have been used for electricity price prediction.

S.Fan et al. proposes a method that can be used to forecast next-day electricity prices[11]. They have used a two stage hybrid network of self-organized map(SOM) and support-vector machine(SVM). The SOM network clusters the input data into several subsets and the SVM is used to fit the training data in each subset. They tested their model on the New England electricity market. Their data set consisted of data from 2003 till 2006 for the New England market. When using the hybrid network they averaged a MAPE of 7.13%. They found that the summer months were harder to predict than the winter months and that their method was not good at predicting price peaks.

Paras Mandal et. al. have used publicly available electricity market information to predict the hourly prices in the PJM²² day-ahead electricity market using a recursive neural network(RNN)[17]. By using RNN based on the similar days (SD) method they propose a method for price prediction. They have also identified several variables that are important in electricity price prediction. They have found load to be a natural choice for price prediction, but they have also identified factors like transmission line congestion, generator availability, generator bidding strategy on the generation side, outages, spinning reserves and more. In their research they have not used weather as an variable, but they mention that this is a variable that easily can be implemented in the presented model. Their model is a three layer feed-forward RNN adopted for forecasting electricity prices for the next 24 and 72 hours. In their model they used data from three years, 2004 - 2006. They found it easier to predict the price in the summer months than in the winter months. Their best MAPE with their RNN was 4.49% for May 13. 2006. On the same day the similar days method achieved a MAPE of 8.53%. They found that a recursive neural net outperforms the similar days method.

E. Livanis and A. Zapranis from the University of Macedonia have written a paper on “Forecasting the Day-Ahead Electricity Price in Nord Pool with Neural Networks”[16]. They try to

²¹Both of these studies was supported by the Ministry of Science and Technology in Spain and the European Union through grant FEDER-CICYT 1FD97-1598

²²A regional transmission organization coordinating the wholesale electricity in 13 states and the District of Colombia

investigate whether a non-linear neural network estimator can provide some incremental value versus a linear model. they used data from Nord Pool ASA from 2003 to 2006 where the the last year was used for testing. For their research they only used historical price data. With their neural net they achieved a MAPE of 8.17%.

P. Subbaraj and V. Rajasekaran have written an article “Evolutionary Techniques Based Combined Artificial Neural Networks for Peak Load Forecasting”[31]. They have combined the two best neural nets they trained for this data set to a CANN²³ module. They have used five different computational techniques, constrained, unconstrained, evolutionary programming, particle swarm optimization and genetic algorithm. The CANN module was compared to a conventional artificial neural net. The input data consists of load, temperature, humidity and wind speed. The net uses January through April for training and May, June, and July 2005 for testing. All the data is for Chennai city, India. Their conclusion is that the evolutionary programming algorithm is the best to predict the load for this area. While a conventional neural network has a MAPE of 2.92% for the month of May, EP based CANN has for the same month a MAPE of 2.14%.

S.K.Aggarwal et al.[1] has in an article compared results from several research projects from 1997 until 2006 concerning electricity price prediction. They have focused on price prediction of the electricity market rather than the load. As they mention in their report, load prediction is more mature than price prediction and thus they would like to focus on results regarding the price. They have analyzed the results based on model used, time horizon for prediction, input and output variables, analysis of result and the data points used for the analysis. They have divided the models used in three categories: univariate time series analysis like ARIMA, multivariate time series models, like dynamic regression and transfer functions, and non-linear models like artificial neural nets. Based on their research they have found that there is no one model that outperforms another. Although in many cases the ANN is better than the two other models, they have not found any conclusive results proving that ANN models always are better than the other models presented. They suspect that this might be because the deregulated electricity market has not existed so many years and thus there might be sparse data available for many regions. The projects referred to in the report have different time spans, ranging from one hour ahead to 12 months ahead for many electricity markets except for Nord Pool Spot. We have gathered the best results mentioned in this report in table 2.3. We have noted what method has been used, which market has been predicted, time span used and the percentage error. Although the time span is different, it is interesting to see the results from similar research projects.

²³Combined Artificial Neural Network

Table 2.3: Results for price prediction conducted in previous work, ARMA=Autoregressive Moving Average, DR=Dynamic Regression, TF=Transfer Function, WT=Wavelet Transform

Method	Market	Predicted period	Time horizon	Error %	Reference
ARMA	Leipzig Power Exchange	45 days	Hourly	3-7%	[10]
DR and TF	Spain and California	2 weeks, 1 week	Hourly	3-5%	[20]
		1 week	1 day ahead	3.5-5.16%	
WT	UK power pool	1 week	1 day ahead	4-7.5%	[15]
		2 days	1, 25, 49 h ahead	2.22-8%	[35]

These results gives just an estimate of how good predictions have been made for other markets than Nord Pool Spot, but it will be an indicator as of how good our predictions are.

Raúl Pino et al. have published a report in Engineering Applications of Artificial Intelligence in 2008[22] where they have looked at price forecasting of next day prices in the Spanish energy market using artificial neural nets. As with many other researchers they have compared their results with Box Jenkins ARIMA method. They have used input data consisting of hourly price and electricity demand from 1998 until 2004. They have made predictions for one day (all 24 hours) in each month of the year, meaning they only have made predictions for a total of 12 days, two days from each year from 1999 to 2004. This means that the earliest predictions made in 1999 had less input data than the predictions made for 2004. Since they have not made predictions of a whole consecutive year it is hard to actually see how their method would perform through the whole year. A prediction method should have predictions from a larger time span to see its actual performance as compared to the market. They use two methods to predict the hourly values, one where they predicted only one step ahead using only real data for their predictions and one iterative approach where they calculated the price based on their own predictions (24 steps ahead). For the Nord Pool Spot market we would have to use the iterative approach since the bids must be sent to Nord Pool Spot the day before the electricity is produced/sold. If we look at their results we will see that in most of their predictions the neural net performs better than the ARIMA method and not surprisingly the one step ahead method is better than the 24 step ahead iterative method. Their MAPE for their ANN predictions for the one step ahead method was 5.26% and 6.33% for the ARIMA method. For their 24 step ahead approach their results were respectively 6.89% and 12.35%. Their results shows that an ANN is more efficient than the ARIMA method. In their report they have also found that the results for predicting working days are more accurate than their results for weekend/holidays. This suggests that we could benefit from differentiating between work days and weekends and thus improve our results. Although their findings are not directly applicable to Nord Pool Spot we can see the contours of how good results

other researchers have on similar prediction problems.

Another report published in *Electrical Power and Energy Systems* in 2008 deals with day ahead forecasting using a mixed data model[2]. Nima Amjady and Farshid Keynia have proposed a hybrid forecast model consisting of neural nets and evolutionary algorithms. They have examined the Pennsylvania-New Jersey-Maryland (PJM) market in USA using a public domain data set. For their examination of their proposed method they set forth to predict one week for each season of the year at the PJM market in 2006. They have compared their results to results from similar models popularly used in price prediction, ARIMA and several multilayer perceptron algorithms. Their model is as follows: firstly, their original price signal is decomposed via a discrete wavelet transform (Daubechies wavelet) and a new data model is created based on features from the wavelet and time domains. The new model can then interpret information from both the wavelet and time domains for its price forecast. Second, they use feature selection considering both correlation and linear independency to find the best inputs. Third, they create 24 cascaded forecasters of each wavelet sub series where each forecaster is a combination between neural nets and an evolutionary algorithm. The parameters to their feature selection techniques and cascaded forecasters are fine tuned using cross-validation. Using this method they have created a seemingly good model for price prediction of the PJM market. If we look at their results for their examination weeks, which were stable seasonal mid weeks, they got a weekly mean average of 4.57% with their method. For the same time area they got a weekly mean error average of 13.25% using ARIMA and 9.67% using Levenberg-Marquardt. After their examination period they tested their method on the whole year of 2006. In this period they got an average weekly error of 5.029% which is a bit poorer than the result they got on their examination period. This slight error increase can be due to the fact that they now also predicted the not so stable parts of the year.

Antonio J. Conejo et al. have in their paper published in *IEEE transactions and on power systems* in 2005 used a wavelet transform combined with an ARIMA model to predict the day-ahead electricity prices for the Spanish electricity market[8]. Wavelets convert the input data into sets of constitutive series. These series have a stabler variance and no outliers. Using the output from the wavelet transforms in their ARIMA model they are able to get more accurate input data for their ARIMA model and thus increasing their prediction performance. They have compared their results from their hybrid method with the results from a pure ARIMA model using the actual price inputs and a naïve procedure. The naïve procedure uses the actual values from the previous week of the week being predicted. They have used their model to forecast the 24 hours of the next day and their results are presented as weekly averages. Through their work they have found

that preprocessing the input data with a wavelet transformation before using ARIMA gives better results than just using the original price data. They have used data from the Spanish market from 2002 on their proposed model. They predict four weeks of 2002, one week for each season and use these results to represent the whole year. It can be debated whether this will give a statistically correct representation of the whole year taking into consideration that some weeks are more volatile than others and that the market will change from week to week. For their model they have used historical information about the previous 48 days on an hourly resolution. MAPE has been used as the error measure in their report. The error percentage is lower for the weeks in winter and spring than in summer and fall in all their reviewed models indicating that it is a great variance in regards to which season is predicted. Their average error for the four weeks predicted were 15.56% for the naïve method, 9.96% for their ARIMA model and 8.11% for their hybrid wavelet/ARIMA model. Overall their new proposed model had the best results and achieved best results for the winter week with 4.78% and the poorest result for the fall week with 11.27%. Again it is difficult to interpret these results since they were done for only four weeks of a year.

Another article published by Zhongfu Tan et al. in 2010 have used a similar approach using wavelets combined with both ARIMA and GARCH models[32]. The article was published in Applied Energy and shows some great results for price prediction of the Spanish and PJM electricity markets. They have made the same experience as Amjady et al. and Conjeo et al. that using a wavelet transform to decompose and reconstruct ill-behaved price series into better-behaved constitutive series performs better. They have used data for the Spanish market from 2002 and data for the PJM market from 2006. They have also used their model on the same four weeks as Conjeo et al. for comparison of their results. Their new model has the best predictions we have seen so far when it comes to next-day price prediction of any market. Their proposed model has an average of the four weeks of just 1.16% for the Spanish market and 0.74% for the PJM market. These are impressive results and far better than any results we have seen so far. Their research is fairly new and it will be interesting to see if there will be conducted more research in this area to back up their results. Another interesting finding with their model is that they are able to predict price peaks very well. They have selected one day from the PJM market where they have predicted 24 hours of August 4 2006 where the minimum price was \$44.17 and the maximum price was \$157.43 during that day. This day has large intra-day changes and should be hard to predict. Their poorest prediction for this day was an error of 1.16% with an average intra-day error of 1%. When looking at these results it seems that using wavelet transformations on the input data has a great advantage over using the input data as is. Based on these results and knowing that machine learning methods

often performs better than ARIMA models, one can utilize the strengths of wavelet transforms and for instance support vector machines in creating very good electricity price predictions. We have not used wavelets in our project for the Nord Pool Spot market, but that would be an interesting future approach.

Chapter 3

Experiments

In this chapter we will present our experiments and the results from each experiment. We will use the machine learning methods described in the previous chapter and analyze the results from each method. At the beginning of the experiment sections we will describe the data set we have used during the following experiments. We will be using the same input attributes on all our methods which will make it easy to compare the results and decide which algorithm is best suited for solving the problem at hand. During our experiments we have had great focus on having robust data. When doing predictions it is important that the integrity of the data is intact. In large data sets, such as this, there is a possibility that some values are missing or even wrong. We have taken great effort in creating a robust data set which has a good integrity. Our best results from our experiments will be documented in the appendix. Due to the complexity of our data set we are not able to supply this in this report, but we will give a close description of the data sets used.

Regardless of the variables used in the data set there are some common denominators. In all our experiments we have made predictions for the NO1 price area in Norway. Our data set only consists of relative values for consumption and price. Both price and consumption have increased dramatically the last ten years. When using relative values it does not matter what the nominal value was for any given time since it is the relative change from day to day that has been used. In chapter 1 we have written about the attributes we have used in our experiments. All our experiments were conducted with 2/3 used for training and 1/3 used for testing. We have calculated MAPE (Mean Absolute Percentage Error) for all our experiments. By using MAPE as an error measurement we can easily compare our results to that of other projects which also have used MAPE in their error

estimates. MAPE for one instance is as follows

$$MAPE = \left| \frac{A - P}{A} \right|$$

where A is the actual value and P is the predicted value.

For all our instances MAPE is calculated as

$$MAPE = \frac{1}{n} \sum_t^n \left| \frac{A_t - P_t}{A_t} \right|$$

where t is the current value calculated.

In addition to MAPE we will present the mean absolute error (MAE) which is the default output by our selected machine learning tool kits. MAE is the absolute difference between the predicted and the actual value and is for all instances presented as

$$MAE = \frac{1}{n} \sum_t^n |P_t - A_t|$$

Unknown output has been removed from the data sets. This ensures that we only get predictions for days where we have valid output values. This has decreased the data set with several days each year, but not significantly in terms of reliability of the data. The input data is evenly distributed throughout the whole year.

We will now go on with the presentation of our experiments.

3.1 Experiment 1

Our first experiments were conducted with the Cubist toolkit from Rulequest. Cubist uses model trees for classifying the data and predict the desired output. Our initial data set was very extensive and consisted of about 3000 input variables. We hoped that Cubist would help us identify the preferred variables used by the model tree, and based on this information help us generate a new data set that we could conduct further experiments on. Our initial approach was to start wide and then narrow down the variables used in the data set. This data set was only run on Cubist since the data set in its current state would be too large to be run with most of our other methods. Using this data set with WEKAs multilayer perceptron would lead to overfitting and use an enormous amount on time. We must also bear in mind that we want to find an optimal data set that can be run on

ADATE. ADATE uses evolutionary computation for solving problems and thus should only be run with a limited number of input attributes.

This was our initial experiment with Cubist where we used a very extensive and large data set. We have also conducted several other experiments using Cubist and these results will be presented later in this chapter. We will now give an overview of the data set we used in our initial experiment.

3.1.1 Data set

For our initial experiment we ran two tests with two different data sets. In the first data set we used all input values on an hourly resolution while on the second data set we used an hourly resolution for all our data except for the reservoir data which we had on a daily resolution. The reservoir data we have at hand are only given as a weekly prognosis. This means that we have converted the data into hourly and daily resolution, but all the input variables for each hour/day of the week are the same. This means that by converting a weekly prognosis into hourly values we will only create more input data which might not be too useful for the machine learning methods. By testing with both daily and hourly resolutions we will determine if our conversions will make any differences in the results. In table 3.1 we can see an overview of all the variables that have been used in these two first experiments. The “H” and the “D” denotes whether the values are on an hourly or daily resolution. The “Previous” statement tells that we have used the values from the previous days in the prediction. “Previous 1-5 H” in the “Price” column tells us that we have used the five previous days price values on an hourly resolution. The “Actual” statement tells us that we have used the actual daily value for that attribute, not historical. If we look at the reservoir column we have both used actual values on an hourly and a daily resolution(the difference between the two runs). We will continue to use this notation on our tables when explaining the data sets used further on in this project. Both our data sets have input data from 07.01.2001 to 30.12.2009. This gives us a total of 3370 days which are divided into training and test sets. We have used 2/3 of the input data for training and 1/3 of the input data for testing.

In our next section we will outline the results from this initial experiments using Cubist and our two data sets.

3.1.2 Results

In table 3.2 we can see the results for our initial experiments with Cubist. Our first run was with reservoir data on an hourly resolution and the second run was with reservoir on a daily resolution.

Table 3.1: Variables used in experiment 1, H=hourly, D=daily

Area	Price	Consumption	Weather	Reservoir	Output
NO all	-	Previous 1-5 H	-	Actual D/Actual H	
NO1	Previous 1-5 H	Previous 1-5 H	Actual H	-	Actual D
NO2	Previous 1-5 H	Previous 1-5 H	Actual H	-	-
NO3	Previous 1-5 H	Previous 1-5 H	Actual H	-	-
NO4	Previous 1-5 H	Previous 1-5 H	Actual H	-	-
SE	Previous 1-5 H	Previous 1-5 H	-	Actual D/Actual H	-
FI	Previous 1-5 H	Previous 1-5 H	-	Actual D/Actual H	-
DK all	-	Previous 1-5 H	-	-	-
DK1	Previous 1-5 H	Previous 1-5 H	-	-	-
DK2	Previous 1-5 H	Previous 1-5 H	-	-	-

Except for for this difference, the data sets are identical.

Table 3.2: Our first runs using Cubist with hourly resolution showing MAE for the training data and MAE and MAPE for test data

	1. run	2. run
MAE train	2.6%	2.6%
MAE test	4.91%	4.91%
MAPE test	5.34%	5.34%
Nr of input variables	3149	2942

When we examine our results we can see that we get fairly good predictions, but unfortunately there is much overfitting in our results. By using this data set we see that the results on the training set is much better than on the test set, implying that Cubist gets to specialized on the training data. The reason for this is likely that we have too many input variables, making it difficult for Cubist to find a definitive pattern that is equally good on both training and testing data. We also see that the results are exactly the same for the run with reservoir data on an hourly and daily resolution with a MAPE of 5.34%. This tells us that since the resolution for the reservoir data is too coarse to begin with, we will not get better results by creating a more dense resolution and thus increasing our input variables. Both of our output models from Cubist uses 13 rules to determine the prediction.

We will now use our data set with the daily reservoir values and conduct experiments with the rules option in Cubist. This function will force Cubist into reducing the number of rules used for its prediction. This is an efficient way of pruning the model tree. When using rules Cubist will place more instances in the leaf nodes of the model tree. The tree structure will be smaller, but there is a chance that the linear regression utilized in each leaf node will be harder to adapt to the increased

number of instances in each node and thus decrease the results. If on the other hand the data in each node keeps their linearity we can achieve similar or even better results by using rules to prune the model tree.

We have done experiments with 1 - 10 rules and the results can be seen in table 3.3. When we compare the results from using rules to our first two runs without rule pruning, we can see that there is less occurrence of overfitting. There is better correlation between the results on the training and test data. This is as expected since the usage of rules will force Cubist into using fewer input variables and thus reduce the gap between training specialization and testing. We can also read from the results that using rules gets better results than the runs without rules. Using 7-10 rules gives us a better result than not using rules, but as we increase the number of rules we also increase the occurrence of overfitting and the results become poorer. We achieve our best results when using 7 rules which gives a MAPE of 5.03% on test data which is better than the 5.34% MAPE we got on our experiment without rules.

Table 3.3: Results using 1-10 rules in Cubist showing MAPE

	1 rule	2 rules	3 rules	4 rules	5 rules	6 rules	7rules	8 rules	9rules	10 rules
Train	3.44%	3.17%	3.02%	2.98%	2.91%	2.88%	2.83%	2.77%	2.74%	2.73%
MAE	4.96%	4.94%	5.25%	5.25%	5.18%	4.98%	4.68%	4.73%	4.75%	4.79%
MAPE	5.18%	5.2%	5.67%	5.67%	5.61%	5.4%	5.03%	5.08%	5.11%	5.14%

In table 3.1.2 we have illustrated the results for our best run with Cubist in our first experiment using 7 rules. In the figure we can see the predicted values compared to the actual values in the time period from January 1. 2009 to December 5. 2009. The y-axis represents the price in EUR/MWh while the x-axis represents the time. If we look at this graph we can see that there are some price peaks that Cubist is not able to predict. Except for these price peaks, Cubist seems to make good predictions for the stable period for the time period presented. In our project we will present the results from the different machine learning methods in the same fashion. We will present results from the test data from the same period, but with the results from each algorithm.

Cubist outputs the variables which it finds most important in creating its prediction. We were hoping that by analyzing the output that we could recognize the attributes that were most important in creating the predictions. Because of the large amount of input attributes used, we are not able to conclude on which attributes are most important for determining the next day price.

Our first experiments with our data sets and Cubist were our introduction to electricity price prediction on Nord Pool Spot. Our results will work as a good indicator of how good results we



Figure 3.1: Cubist results using 7 rules predicting the NO1 price for 2009

can expect to achieve when we move on to our other experiments. In our related work we have written about many articles covering price prediction for other power markets than the Nord Pool Spot market and we have seen predictions that are worse than our initial results with Cubist. We are likely to improve our results when we reduce the number of input attributes and we will then compare our best results with the results of other projects regarding electricity price prediction.

In our next experiments we will try to improve our results from these first runs. We will have to improve our data set and test other the machine learning methods.

3.2 Experiment 2

There are several methods which can be used for determining which input variables to be used in the data set, also known as feature selection. Some feature selection methods are principal component analysis, correlation analysis and spectrum analysis. Through feature selection the input data is analyzed and as a result we get the attributes that correlates best to the desired output. Selecting the correct input attributes will have an affect on both time performance and results. We will write more about feature selection in chapter 5. In our project we have not used any feature selection algorithms for finding the optimum data set. We have manually decreased the number of input attributes to see how this will affect our results. We had a strong focus on price variables and using hourly resolution

for the NO1 area we are trying to predict. The data set derived from this manual decrease is the data set that we have initially used for our first experiments with WEKA. The data set consists of both hourly and daily resolution. We ran experiments with this data set on all our selected tool kits and algorithms. In the following section we will describe our data set and then we will present our results from the experiments.

3.2.1 Data set

In this data set we have reduced the number of input variables. We have used a combination of an hourly and daily resolution. Our previous data set had too many attributes which lead to overfitting. Our evolutionary computation algorithm, ADATE, would also have problems with handling this many input attributes and get a result in a descent amount of time. The data set has been shrunk from just below 3000 variables to only 111 input variables. In table 3.4 we can see which variables we used in our new and improved data set. The main difference is that we have reduced the number of hourly inputs and replaced them with daily inputs. The categories are the same: price, consumption, weather and reservoir data. For an explanation of the table look at the description of table 3.1. Alongside using only daily inputs for most of the attributes we have also decreased the number of included previous days. In experiment 1 we used historical information about the five previous days for all our attributes, while in these experiments we will only use information about the prior one to three days of the day being predicted. For the NO1 price area we have kept the hourly resolution for the two previous days. That is now the only hourly resolutions in the data set.

Since we know that the main driver for predicting the price is the price it self, we have also created a second data set which we are testing in these experiments. In this subset we have removed all but the historical price data. By comparing a price only model with our all attribute model¹ we will observe if we get an effect by adding information about consumption, weather and reservoirs. Does this extra information add something to the performance of the prediction or will it only be considered as noise. If we look at the data set in figure 3.4 our price data set consists only of the attributes in the price column. The time period used for these data sets is the same as in the data set used in experiment 1. We have also used the same 2/3 and 1/3 split for training and test data. Further on in this project we will refer to the two data sets as “prices only” data set and “all attributes” data set.

The benefits of these data sets are clearly that we can use it more extensively on other machine learning methods. The chance of overfitting has been reduced and we are hoping to get some good

¹From now on we will refer to the data sets as: “prices only” and “all attributes”

Table 3.4: Variables used in experiment 2, D=daily, H=hourly, prev=previous, *=input from two weather stations, **=both content and inflow

Area	Price	Consumption	Weather	Reservoir	Output
NO all	-	Previous 1-3 D	-	Actual D**	
NO1	Prev 1-3 D, prev 1-2 H	Previous 1-3 D	Actual D*, prev 1-2 D*	-	Actual D
NO2	Previous 1-3 D	Previous 1-2 D	Actual D	-	-
NO3	Previous 1-3 D	Previous 1-2 D	Actual D*	-	-
NO4	Previous 1-3 D	Previous 1-2 D	Actual D*	-	-
SE	Previous 1-3 D	Previous 1-2 D	-	Actual D**	-
FI	Previous 1-3 D	Previous 1-2 D	-	Actual D**	-
DK all	-	Previous 1-2 D	-	-	-
DK1	Previous 1-3 D	Previous 1-2 D	-	-	-
DK2	Previous 1-3 D	Previous 1-2 D	-	-	-

results with this data set. Ideally we would have used feature selection when reducing our data set, but we have concentrated on the values we seem fit to work with our different methods. This new data set has been run on all our selected methods and all the results will be compared to each other. The results will be presented over the next sections.

3.2.2 Cubist

In our first experiment we used Cubist and model trees with a large data set. Now, we have created data sets with fewer input attributes which ought to perform better and reduce the occurrence of overfitting. We are likely to see a decrease in rules used by Cubist and the model tree created will be smaller. With less values to be classified in each leaf node we are likely to see that Cubist will be able to create a better linear regression for the values in the nodes and thus increase its prediction performance. In our initial experiment with Cubist our best result was a MAPE of 5.03% for our test data using rules and 5.34% for the runs not using rules. There is a large gap between the training and test results and we hope that this gap, the overfitting, will be smaller on our new data set.

When we look at the initial results with the new data set in table 3.5, we see that there is a slight increase in performance. This is as expected since there are fewer input attributes. What is interesting is to see that the result when using only prices is marginally better than our data set using the all attributes data set. There is also less overfitting in the run with only the prices. This again has to do with the fact that there are fewer attributes in this data set and a smaller chance for the algorithm to be too specialized on the training data. Using Cubist on our new data set will give a MAPE of 5.02% for the price only data set and 5.1% for the all attribute data set. In our previous experiment Cubist used 13 rules to create its model. In these experiments Cubist used 14 rules for

classifying the data set with all the attributes, while it only used 7 rules on the prices only data set. This tells us that the model used with prices only data is simpler and yet better than the data set with the added attributes.

Table 3.5: Predicting NO1 using Cubist and model trees

	Prices only	All attributes
MAE train	3.21%	2.72%
MAE Test	4.78%	4.73%
MAPE	5.02%	5.1%

It is interesting to see that the prices only data set performs better than the all attributes data set. Cubist suggests that we can get better results by only using price for input for this regression problem. We will see if this trend is apparent in our other experiments with the other methods. As with our first experiment we have also used the rules function of Cubist to see if we can further improve our results with this method.

With rules

The rules experiment has been done on the all attributes data set. In table 3.6 we can see the results from each of the runs using the rules function. Again, we have used 1-10 rules and compared the results to the model not using the rule pruning function. When we compare these results to the experiment in the previous section where we did not constrain Cubist with the rules option, we can see that by utilizing rules we can achieve better results. While we on the same data set got an error of 5.1% without rules, we can see that in our run with 5 rules we got an error of 4.86%. This is an improvement of 0.24% points better than the experiment not using rules. This is also what we experienced on our first data set when we used rules. On our data set it is therefore proven that by reducing the complexity of the model tree we can improve our results. in figure 3.2 we can see the performance for our best Cubist run using five rules for our selected period of 2009. In this graph we can see that the model tree algorithm predictions follows the actual price very well. There is one period at the end of July where the predictions are not so good. The overall performance will decrease because of these mis-predictions. If we examine our data set for that period, we find that there are several hours between the 26. of July and 1. of August 2009 where the hourly prices vary greatly. The prices are abnormally low ranging from 0 EUR/MWh to over 30 EUR/MWh. When we then calculate the relative price in regards to that price, we get an abnormally large price change. When the linear regression method then uses the calculated constants to calculate the predicted



Figure 3.2: Cubist results using 5 rules predicting the NO1 price for 2009

price, the estimate will be very wrong. We are uncertain to why we have these price peaks for that period, but it has a great affect on the predictions. Linear regression, and thus model trees, is very vulnerable to price peaks and in such volatile periods the performance will decrease.

Table 3.6: Using rules on the reduced data set gives the following results

	1 rule	2 rules	3 rules	4 rules	5 rules	6 rules	7rules	8 rules	9rules	10 rules
Train	3.63%	3.37%	3.27%	3.17%	3.16%	3.06%	3.06%	3.05%	3.00%	2.94%
MAE	4.80%	4.9%	4.73%	4.62%	4.56%	4.59%	4.71%	4.71%	4.69%	4.69%
MAPE	5.03%	5.18%	5.0%	4.92%	4.86%	4.89%	5.09%	5.09%	5.03%	5.01

We will now go on to WEKA and other machine learning methods to see how good predictions they make for the prices only and all attribute data set.

3.2.3 WEKA

In this section we will describe our experiments conducted with the WEKA toolkit. In these experiments we have used the same data set as with the Cubist runs described and found in 3.2.1. We have chosen several algorithms from the WEKA tool kit. From the basic linear regression to the more advanced support vector machine algorithm. All our algorithms have been described in chapter 1.

Linear regression

Our first experiment in WEKA is with linear regression. Linear regression is one of the most rudimentary algorithms in the tool kit. Linear regression will try to fit a linear line through the data and adjust the values using a least squares equation. In comparison to Cubist and model trees we expect to see that the results from linear regression will be slightly poorer. This has to do with the nature of the model tree that places values with the same characteristics in leaf nodes before it uses linear regression. When we use only linear regression it is implementing the same methodology, but on a larger set of input attributes meaning it will be harder to linearly separate the data. In WEKA we have chosen to use the M5 method for attribute selection in linear regression.

Presented in table 3.7 are the results for the two preliminary runs using linear regression. The first run only used prices for input data, while run 2 uses prices, consumption, weather and reservoir. This is the same data set and variables used in the Cubist experiment, presented in table 3.5.

Table 3.7: Predicting NO1 using WEKA and linear regression, showing MAE and MAPE on test data

	Prices only	All attributes
MAE	8.71%	8.89%
MAPE	8.87%	9.04%

If we look at our results from the linear regression experiments and compare them to the results from Cubist we can see that linear regression is far from as good as model trees. With linear regression our best MAPE is 8.87% while it was 4.86% with model trees. This indicates that model trees is a better choice for solving our regression problem and that WEKAs linear regression method has problems finding a linear line that perfectly separates the data. Again we see that we get better results on the data set containing only the price information. In figure 3.2.3 we can see a graph for the period in 2009 that we have looked at earlier. Generally the predictions are very close to the actual values, but there are several price peaks where linear regression performs poorly. This is the same case as with our Cubist experiment where we had abnormal prices at the end of July resulting in poor performance. Model trees behaves better than linear regression in this period and gives an overall better performance. If it was not for these price peaks, linear regression would perform much better.

In our next experiment we will use a neural network approach for predicting the electricity price. We are likely to see a better adaption to the input data when using neural nets and the results will improve, but will it be able to perform as well as model trees?

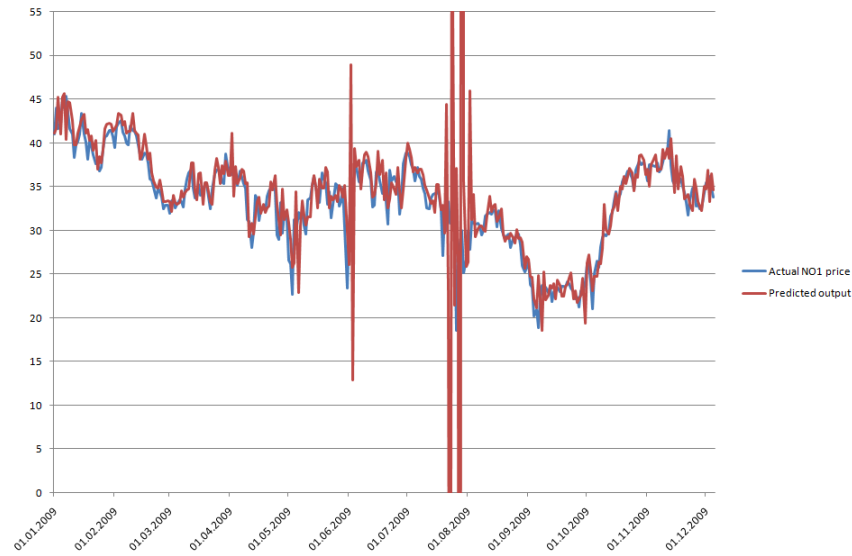


Figure 3.3: Linear regression for the NO1 price for 2009

Multilayer perceptron

In WEKA artificial neural net is implemented through their multilayer perceptron algorithm (MLP). MLP uses backpropagation to adjust the weights to each connection between the hidden layer and the output layer. Backpropagation uses gradient descent to minimize the error output of the network and adjust the weights accordingly. We have used the same data set as in our previous run. In these experiments we expect that the difference between the two data sets will become more apparent.

For these experiments we created a neural net with five nodes in the hidden layer for both of the data sets. We have trained the neural net for 500 epochs with a learning rate of 0.25 and a momentum of 0.2. The number of epochs gives how long the neural net will run while the learning rate and momentum tells how the weights are adjusted.

The results can be seen in table 3.8. For the first time we can see that the all attributes data set performs better than the prices only data set. For prices only we have a MAPE of 6.15% while we have a MAPE of 5.45% on the all attribute data set. The reason for this is the gradient descent method used for minimizing the error from the network. Linear regression (and model trees) uses a least squares method when adapting the data. A neural net is better suited for non-linear data, thus it will adapt itself better to the multivariate data set. MLP is better than linear regression, but on this data set it is still Cubist and model trees that have the best results. Depending on the settings we

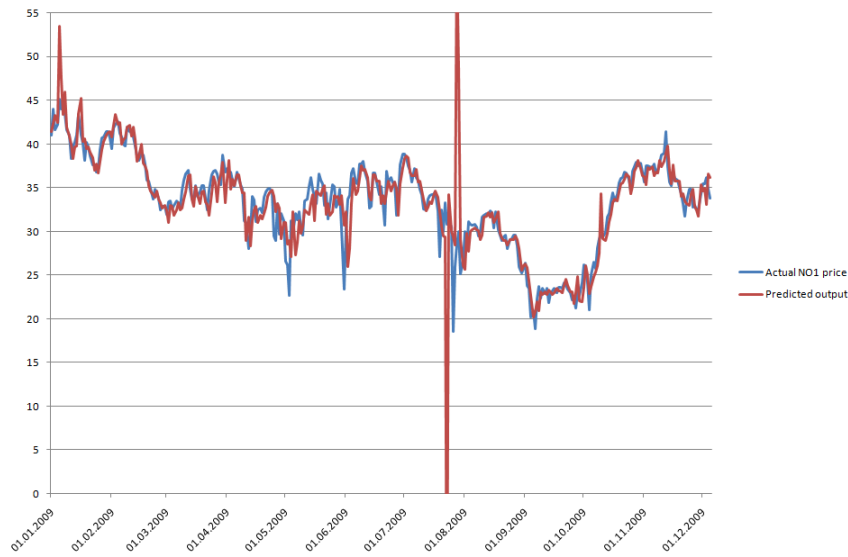


Figure 3.4: MLP for the NO1 price for 2009

apply to our neural net the results will differ. Our neural net is not optimally tuned, meaning that further experiments with the neural net settings could improve the performance by finding a better suited local minima.

Table 3.8: Predicting NO1 using WEKA and multilayer perceptron

	Prices only	All attributes
MAE	6.15%	5.45%
MAPE	6.15%	5.59%

In figure 3.2.3 we see a graphic representation of the test results for 2009. Again we find that our machine learning method has problems with the period at the end of July. If we were to remove the outliers from the data set in this period, MLP would perform much better. We have chosen not to remove any outliers at this time and keep the data set as intact as possible to see if our other machine learning methods will behave in the same way in regards to this time period.

Our next method in the WEKA tool kit is Radial Basis Function (RBF) network. RBF is in its form also a neural net with the exception that it does not use backpropagation for adjusting the weights.

Radial Basis Function network

An RBF network is an instance based learning method which in its simplest form is a three layered network. If we compare an RBF network to the MLP algorithm, we find that the RBF network does not use backpropagation. Since the input and output network are trained separately we will see an increase in speed performance when using the RBF network. In table 3.9 we see the results from the radial basis function network runs. The MAPE is 5.41% for the prices only data set and 5.44% for the all attributes data set. In figure 3.2.3 we can see the predicted output from the RBF network compared to the actual price. The predicted values from the RBF network follows the actual values very well. If we examine the prediction values from the RBF network we find something interesting. All the predicted relative values are 1.004. So for all our test cases the RBF network predicts that the relative price change from one day to another is 1.004. Our RBF network has found a constant for all our test cases that will predict the next day price with a MAPE of 5.44% using the all attributes data set. When we examine our data set we find that 1.004 is the average daily relative change from all the output values. This means that in our experiment with the all attribute data set, our RBF network will only use the average daily change as a constant. By using the average relative change we get better performance at the end of July period where model trees, linear regression and MLP fails. In our later experiment with ADATE we will see that the constant chosen by our RBF network will be very similar to a constant chosen by ADATE.

With MLP we found that the all attributes data set performed better than the prices only data set. With the RBF network we see that the prices only data set again is better than the all attribute data set, but both runs are much better than the neural net implementation. A low variation in the input data might benefit the RBF network since it is an instance based learning method.

Table 3.9: Predicting NO1 using a Radial Basis Function network

	Prices only	All attributes
MAE test	5.16%	5.21%
MAPE test	5.41%	5.44%

As we progress with our methods we see that the results are getting better for each method we have selected for testing. We will now move on to the results gathered from our Gaussian process experiment.

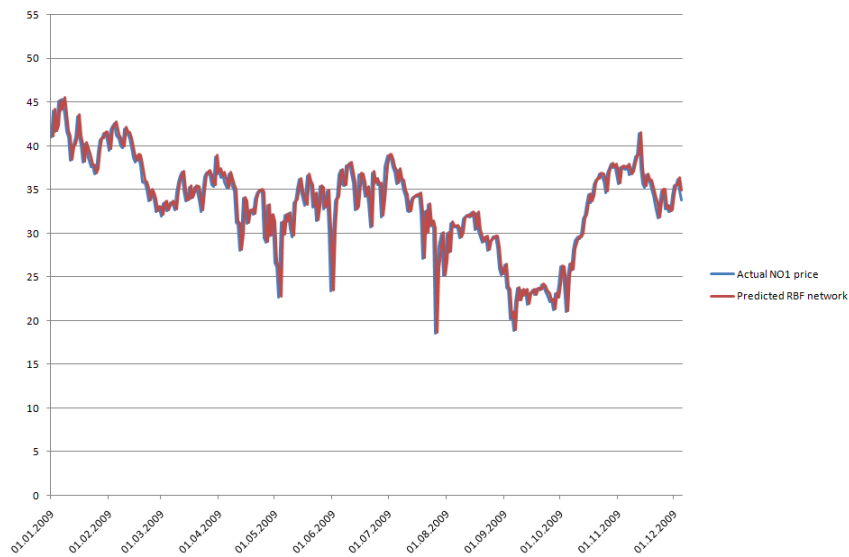


Figure 3.5: RBF network for the NO1 price for 2009

Gaussian Processes

The Gaussian process (GP) experiment were run with the normalized poly kernel. A GP is dependent upon its mean function and covariance function. In figure 3.2.3 we can see the results for the GP experiment. The results using GP are the best so far with a MAPE of 4.15% on the prices only data set and 4.68% for the all attributes data set. In figure 3.2.3 we see the graph for our GP experiment. For the earlier mentioned period in July, GP performs well, but it is unable to follow the actual price peaks. A GP will perform poorer when there are regions with few training values. The peaks illustrates such areas where there are fewer training data points leading to the poorer predictions. GP uses Gaussian distribution for its classification of the data and when values are outside the Gaussian distribution it will perform poorer.

Table 3.10: Predicting NO1 using WEKA and GaussianProcesses with Normalized Poly kernel

	Prices only	All attributes
MAE test	4.12%	4.43%
MAPE test	4.15%	4.68%

In our next experiment we have used another kernel machine, support vector machine.

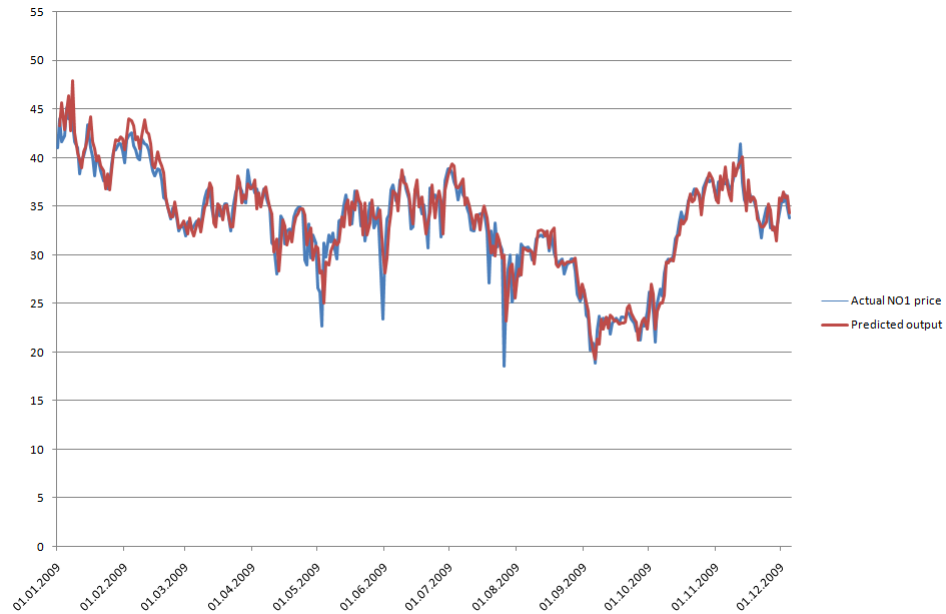


Figure 3.6: Gaussian processes for the NO1 price for 2009

Support vector machines

When we use the Support Vector Machine (SVM) implementation in WEKA we get some interesting results. The support vector machine divides non-linear data in several dimensions using the “kernel trick”. Based on the support vectors selected by the kernel method the optimizer calculates the prediction. In our first experiment we have used the poly kernel and WEKA’s SMOregOptimizer algorithm. In table 3.11 we see the results from our initial run with SVM. We get the best results for the data set using the all attributes data set. With 5.85% MAPE the support vector machine is better than the MLP algorithm we investigated earlier. Our initial results using the SVM is slightly poorer than the results of the RBF network, 5.44% and the Gaussian process method, 4.68%, we tested earlier.

Table 3.11: Predicting NO1 using WEKA’s SMOreg using the Poly kernel

	Prices only	All attributes
MAE	5.72%	5.64%
MAPE	5.91%	5.85

Although the results are not the best so far we have decided to do more experiments with SVMs.

We have also conducted experiments with other kernels that are available in WEKAs SVM. The Gaussian process uses the normalized poly kernel for its predictions. We can also use this kernel in our SVM and it will be interesting to see if we can improve our results using this kernel. We will also test with the RBF kernel and the PUK kernel. The different kernels are described in chapter 1. In table 3.12 we can see the results for the SVM using the different kernels. If we look at the results we see that by varying the kernel for our SVM, the results will differ. Our best result for the prices only data set is with the normalized poly kernel with a MAPE of 4.23%. Using the poly kernel on this data set gives the poorest results with a MAPE of 5.91%. Our best results with the all attributes data set is 4.19% using the RBF kernel. With this data set the poly kernel gives the poorest results with 5.85%. It is again interesting to see that we get the best performance with the prices only data set. The SVM experiments also suggests that we have nothing to gain by implementing consumption and weather attributes. This is an interesting observation and we will conduct further experiments to see if this pattern continues.

Table 3.12: Predicting NO1 price using WEKAs SMOregs different kernels

	Poly Kernel	Normalized Poly Kernel	RBF kernel	PUK kernel
MAE prices	5.72%	4.18%	4.51%	5.01%
MAPE prices	5.91%	4.23%	4.71%	5.21%
MAE all	5.64%	4.3%	4.19%	4.44%
MAPE all	5.85%	4.55%	4.46%	4.71%

In figure 3.2.3 we can see a graph illustrating the price from 1. January 2009 until 5. December 2009. The y-axis shows the price in EUR/MWh while the x-axis shows the date of year. If we look at this figure we see that most of the selected kernels are following the price changes quite well. The poly kernel has some problems with some dates at the end of July. This is the same pattern we have seen with our model tree, linear regression and MLP algorithms.

We have now presented the results for all our chosen machine learning methods in our 2. experiment. The best results so far is 4.15% for the prices only data set using a Gaussian process and 4.46% for the all attributes data set using support vector machines with an RBF kernel. We will now perform new experiments with the same methods, but on another data set consisting of a different training and test period.

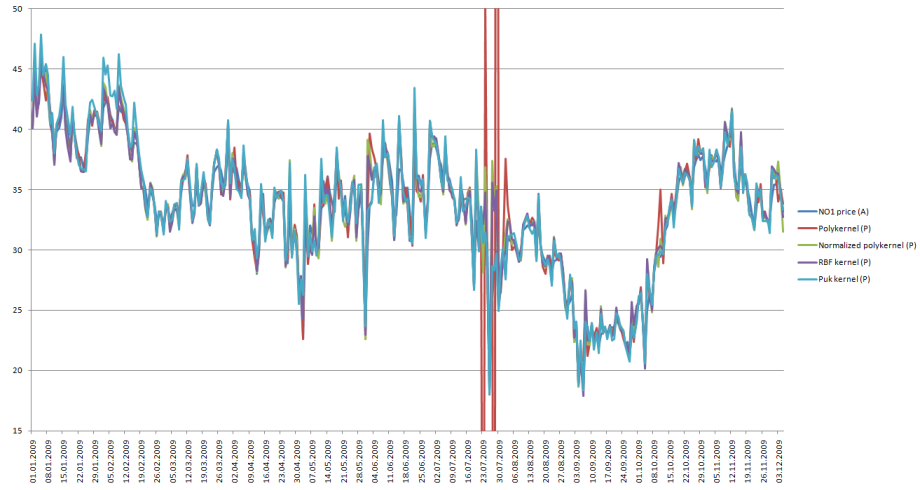


Figure 3.7: Comparison of the kernels predicting the NO1 price for 2009

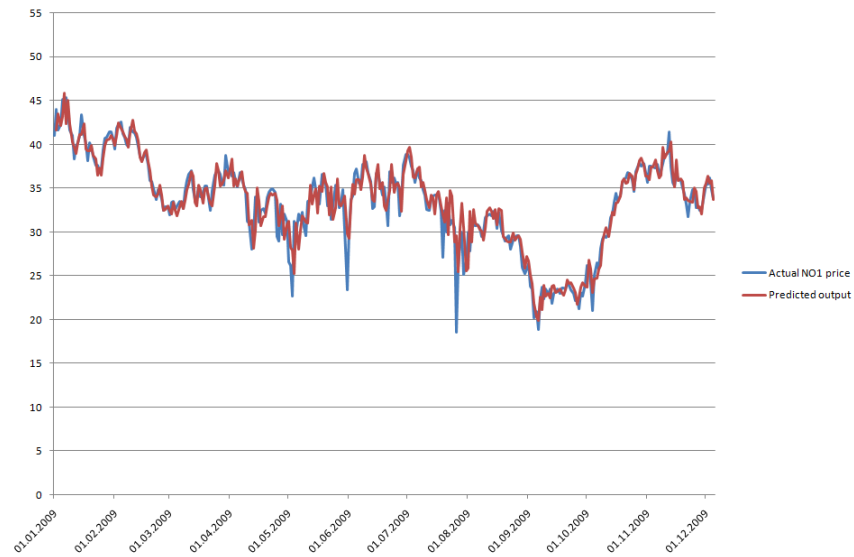


Figure 3.8: SVM with the RBF kernel for predicting the NO1 price for 2009

3.3 Experiment 3

In these experiments we have reduced the number of input rows in our data set. The input attributes are the same as experiment 2, but we have removed input data from the years 2001 through 2004. This means that we will now be using a data set with data from January 4, 2005 to December 5, 2009. By reducing the data set we are hoping to see an improved performance. We suspect that the usage and price pattern from our earlier years are different from the pattern in the later years. Since our data are analyzed sequentially we have previously used training data that is from another time period than the test data. By removing the earliest training data the training data will be more similar to the data in the test data set. We have conducted this experiment in the same fashion as experiment 2. We can then directly compare all the models and see if there are models that will benefit or not from this data set reduction. Our first experiment with the new data set is conducted with Cubist. A problem with reducing the data set in this manner can be that we delete volatile periods that are significant for the prediction of similar periods in the future. We will also be using a shorter time period for our test data when we reduce our data set, but we are still using 479 days for testing which should be enough for accurate test results.

3.3.1 Cubist

In these experiments with Cubist we have not utilized the rules function, although we got marginally better results using rules in experiment 1 and 2. In our previous experiment we found that Cubist had poorer performance than for instance support vector machines. The results from our new experiment with Cubist can be seen in table 3.13. With the previous data set we got approximately the same results for both the prices only data set and the all attributes data set, respectively 4.55% and 4.56%. In our new experiment we see that the MAPE for our two runs are 3.47% for prices only and 3.36% for the all attributes data set. Both our Cubist experiments have improved greatly by reducing the number of input days to the data set. In figure 3.3.1 we see the results from the prediction compared to the actual values for the same time period we have previously used for our illustrations. It is interesting to see that for the time period at the end of July, which in experiment 2 had very poor predictions, Cubist predicts very well when compared to experiment 2. This means the model tree for this experiment is different from the model tree in experiment 2. Cubist manages to split the current data set better and the local linear regression performed in each node approximates better to the actual values.

Table 3.13: Predicting NO1 using Cubist and model trees

	Prices only	All attributes
MAE	3.38%	3.5%
MAPE	3.47%	3.56%

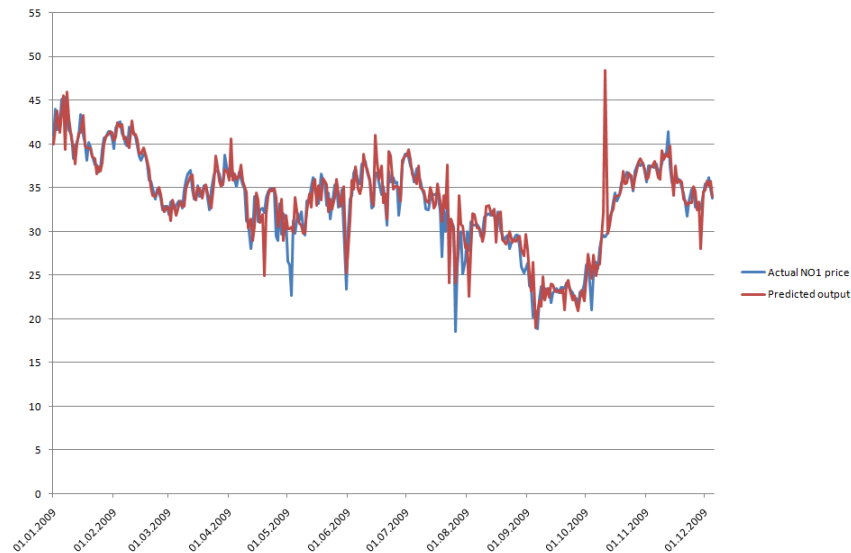


Figure 3.9: Cubist all attributes for predicting the NO1 price for 2009

3.3.2 WEKA

In experiment 2 we got our best results from WEKA using its support vector machine implementation. Our results from Cubist on the new data set is noticeable better and it will be interesting to see if we can see similar improvements with the other algorithms from the WEKA tool kit.

Linear regression

With our previous experiment with linear regression we got an error of 8.87% on the prices only data set and a MAPE of 9.04% on the all attributes data set. Linear regression had the poorest results in our previous experiment. In table 3.14 we see the results from our new experiment. The performance has decreased dramatically to a MAPE of 20.69% and 19.48%. In opposite to Cubist, linear regression has poorer performance on the new data set. If we look at the output from WEKA we see that there are two days that are predicted very wrong. July 29 has an MAE of 2147% and July 30 has an MAE of 5303%. This is the same time period we have mentioned earlier. The overall

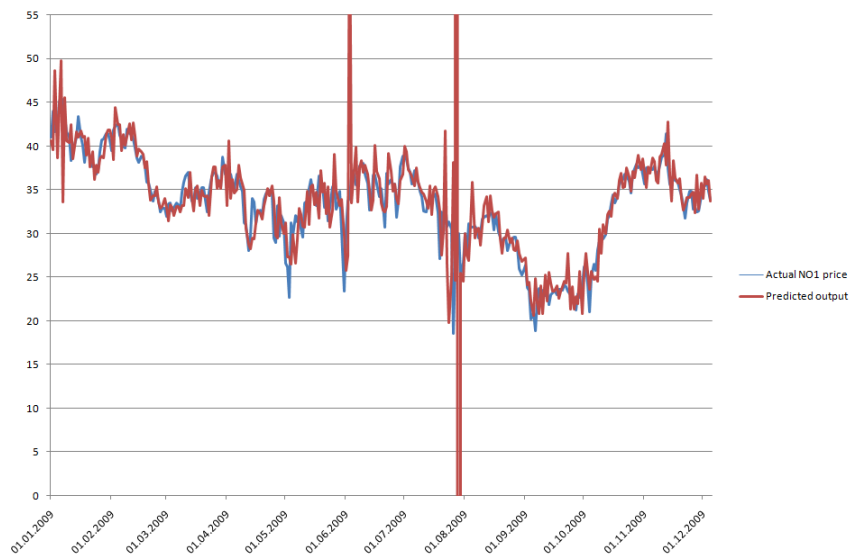


Figure 3.10: Price prediction with linear regression for 2009

performance of linear regression has become poorer. In this experiment we use less observations and have fewer input days for our methods to adapt to. By removing outliers from the data set we would probably increase the performance with linear regression, but that is another experiment. We have used the same input data set for all our experiments and we will therefore treat the mean MAE and MAPE output from the linear regression experiment as it is. This proves that linear regression is very sensitive for fluctuations in the input data and that two days can reduce the overall performance. We have presented the linear regression results for 2009 in figure 3.3.2. We will now go on to the experiment with multilayer perceptron.

Table 3.14: Predicting NO1 using linear regression

	Prices only	All attributes
MAE test	21.66%	20.31%
MAPE test	20.69%	19.48%

Multilayer perceptron

For our experiments with MLP we have used the same settings for the neural net as in experiment 2. In that experiment we had a MAPE of 6.15% and 5.59% for the prices only data set and the all attributes data set. In our new experiment we have improved our results to 4.28% and 4.99% as seen

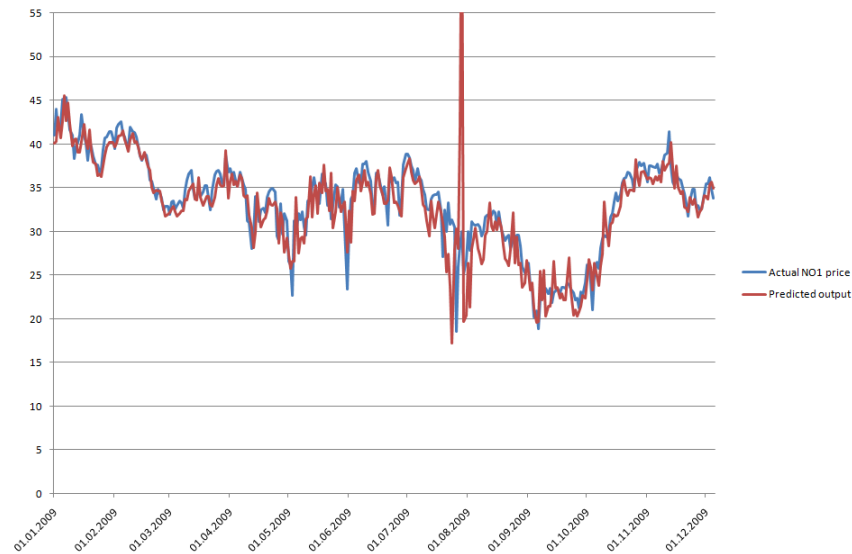


Figure 3.11: MLP for the NO1 price for 2009

in table 3.15. With MLP we see an increase in performance with the new data set. The best increase is with the prices only data set. In experiment 2 the all attributes data set performed better than the prices only data set which is not the case here. As with our Cubist experiment we see that the prices only data set performs best. Again we suspect that this has to do with the longer time period we used earlier which made it easier for the network to adapt to the all attributes data set. When we removed the earliest years from the data set it seems that the all attributes data set has more noise which makes it more difficult to calculate the proper weights for the network. In figure 3.3.2 we see the results for our MLP in experiment 3.

Table 3.15: Predicting NO1 using multilayer perceptrons

	Prices only	All attributes
MAE test	4.22%	4.98%
MAPE test	4.28%	4.99%

Radial Basis Function Network

A radial basis function network uses instance based learning for predicting an output. In our new data set we have reduced the number of inputs that the RBF method can learn from. In experiment 2 we had a MAPE of 5.41% and 5.44% from our RBF network. In table 3.16 we see the results

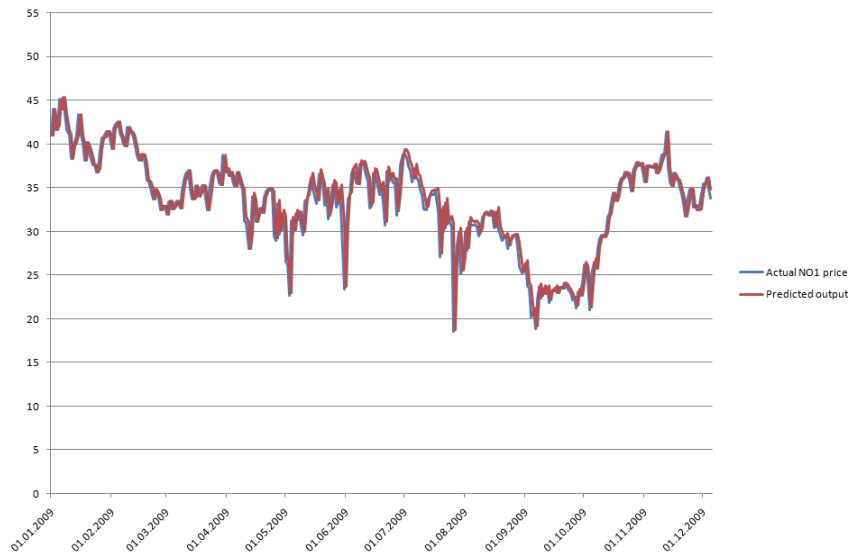


Figure 3.12: RBF network for the NO1 price for 2009

from our new experiment. The MAPE from our runs in experiment 3 are 3.93% and 3.94%. As in experiment 2 the results for the two data sets are almost identical, but the results have improved. It is interesting to see that the results are very similar. In experiment 2 our RBF network predicted the same output for all our test outputs equaling the average output. In this experiment our RBF network have used six constants for predicting the next day relative change². Again this is related to how the RBF network uses a nearest neighbor method when creating its prediction. With the reduced number of input cases, the RBF network has adapted to the new data set. In figure 3.3.2 we see the graphical presentation for our RBF network. Again we can see that the predicted output follows the actual output very well. The RBF method adapts better to the volatile period at the end of July.

Table 3.16: Predicting NO1 using WEKA and RBFNetwork

	Prices only	All attributes
MAE	3.90%	3.91%
MAPE	3.93%	3.94%

²Values are: 1, 1.003, 1.004, 1.012, 1.013 and 1.014

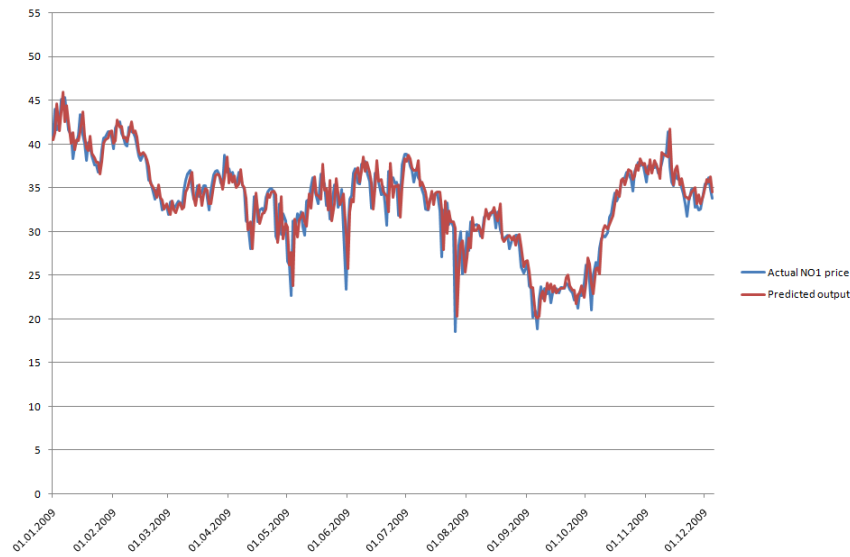


Figure 3.13: Gaussian processes for the NO1 price for 2009

Gaussian processes

In this experiment we have used the same settings for the GP method as in experiment 2. In experiment 2 we got the best results for the NO1 area prediction with Gaussian processes as compared to the other methods. We then had a MAPE of 4.15% and 4.68%. With Gaussian processes the results have improved with our new data set. In table 3.17 we see the results from this experiment. The results have improved to 3.72% and 3.57% for the prices only and the all attributes data set. A difference from the earlier experiment is that Gaussian processes now are best on the all attributes data set. The performance improvement we see with our GP can be related to the fact that we have a better data distribution in our new test data set. The data points are more evenly distributed which makes the adaption easier for our Gaussian process.

Table 3.17: Predicting NO1 using Gaussian Processes with Normalized Poly kernel

	Prices only	All attributes
MAE	3.68%	3.53%
MAPE	3.72%	3.57%



Figure 3.14: Price prediction with the RBF kernel for 2009

Support vector machines

We have conducted the same experiments for our support vector machine algorithms as we did in experiment 2. We have tested four kernels with the same settings as in our previous experiment. The results from our new experiments can be seen in table 3.18. If we compare our results from this experiment with the results from experiment 2 in table 3.11, we see that all the results have improved. Our best result for the prices only data set is with the normalized poly kernel with a result of 3.31%. This is also the best result we have got for the prices only data set so far. For the all attribute data set the best result is achieved with the RBF kernel with a MAPE of 3.14%. So far in our experiments this is the best overall result when predicting the next day price for the NO1 price area. In figure 3.3.2 we see a graph presenting the results using the RBF kernel. In this graph we can see that using SVM with the RBF kernel follows the contours of the actual price very well.

Table 3.18: Predicting NO1 price using WEKAs SMOregs different kernels

	PolyKernel	NormalizedPolyKernel	RBFkernel	PUK kernel
MAE prices	5.77%	3.25%	3.50%	3.45%
MAPE prices	5.72%	3.31%	3.51%	3.49%
MAE all	7.32%	3.14%	3.10%	3.27%
MAPE all	7.21%	3.21%	3.14%	3.30%

Figure 3.15: The optimal program suggested from ADATE for the NO1 price prediction

```

tanh(
  realSubtract(
    tanh( tanh( X2_no1_21 ) ),
    tor( rconst( 85, 0.3924974052932026E~6, ~0.14247801791 ) )
  )
)

```

In our 3. experiment we have also experimented with ADATE. We will now see how well an evolutionary algorithm will perform on our best data set.

3.3.3 ADATE

We have done one experiment with ADATE and to make our data set compatible with ADATE we had to do some changes to our data set. ADATE can not handle unknown input variables. In our time series we have many unknown input variables since there have been several changes in the Norwegian market concerning price areas. To make our data set run on ADATA we have removed price and consumption data for the NO3 and NO4 price areas. These price areas have not existed through the whole time period and thus they were removed. We could have calculated a mean price for the missing values, but this would do more harm to our data set than just removing them. In addition we have removed a total of 23 days from our data set that contained unknown values. We do not expect that these changes will affect the results significantly. When using this method with our best support vector machine method, we got an MAE of 3.16% as opposed to an MAE of 3.1% when we used NO2, NO3 and the missing days on our support vector machine algorithm. We have now a data set with 102 input attributes.

Through the ADATE experiment we experienced a lot of overfitting as the program developed by ADATE evolved. As ADATE introduced more and more attributes to the population, we found that the results got poorer and that the overfitting increased. We have gathered two results from ADATE that we want to present. Firstly, through ADATE we have found a constant relative day to day change of -0.997. This coincides with the suggested constants we got from our RBF network. Secondly, we will present the program which ADATE has created for predicting the next-day electricity price for the NO1 area. The program can be seen in figure 3.15

When we examine the suggested program we can see that it is quite simple. It only uses one input variable for predicting the NO1 price. This attribute is the hourly price for hour 21 two days

before the predicted day. An interesting observation is that this same attribute is found by Cubist as an attribute that plays an important part in predicting the price using model trees. With this program we get a MAPE of 3.34% on our test data. This is slightly poorer than our support vector machine model which on the same data set gives an MAE of 3.16%. Our ADATE results shows that an evolutionary algorithm can adapt to our data set and give nearly as good predictions as our best support vector machine result. The drawback is that ADATE needs far more computer resources and running time to achieve these results.

We have now presented all our experiments and their results. In our next chapter we will conclude upon our findings, before we suggest further work in the field of electricity price prediction.

Chapter 4

Conclusion

When we started this project, we wanted to research which machine learning methods would be best suited for predicting the next-day electricity price for the Norwegian market. We have compared seven different machine learning algorithms using the same data set for all our algorithms. We have made predictions for the NO1 price area in Norway and as such we do not know how well our selected algorithms would perform on the other price areas, but we assume that given the same input attributes we would get fairly similar predictions for the other price areas as well. We know that there are many participants who trade with electricity on the spot market and that there is an interest in developing a good price prediction model. We have seen that even small improvements to an existing price predicting model can benefit companies trading on Nord Pool Spot. With the introduction of smart grids, we will see the emerging of a new market segment which could benefit from having price predictions built into their systems. Both existing and new players of the electricity market can benefit from a good price prediction tool.

One of the key elements in creating a good price prediction is the data set. The data set must be historical correct and it must comprise of relevant attributes for the attribute being predicted. In our work we have chosen to use historical data about electricity prices, consumption, weather and reservoirs. We have not done any feature selection in this project, but in a future project it would be natural to implement a feature selection method to effectively remove attributes that do not contribute to making a good prediction. We have tested seven different subsets of our data set. We started out with a large data set with several thousand attributes and shrunk the data set to 102 attributes which we used for our ADATE runs. For our second and third experiment we used two data sets for our experiments. One data set containing only price data and one data set containing price, consumption, weather and reservoir data. The reason for doing this was to see

whether we improved our prediction when adding non-price information to the data set. We chose seven algorithms which we did our experiments with. These algorithms are model trees, linear regression, multilayer perceptron, Gaussian processes, RBF network, support vector machines and evolutionary computation. We used software which had an implementation of the different methods, Cubist, WEKA and ADATE. We expected that there would be a difference between the algorithms which we also have documented through our results.

From our previous work in machine learning, where we also did predictions on the next-day electricity prices, our best results for the NO1 area were 3.34% using ADATE and 3.35% using neural nets. In related work we have found the best results being 0.74% for next-day price prediction for the Spanish market. This prediction was only made for four weeks from 2002 and we are unsure how well their method will do over a whole year. In our experiments we received our best results using a support vector machine on our all attributes data set. For the NO1 next-day price prediction our best result was 3.14% using an RBF kernel on a support vector machine using test data from August 5. 2008 to December 5. 2009. Our best results can be seen in table 4.1. This is better than the results we achieved in our earlier project. All our best results was from our 3. experiment, except for the linear regression results which are from our 2. experiment. The kernel based machine learning methods, Gaussian process and support vector machine, and evolutionary computation, ADATE, performed best on our all attributes data set while the other methods performed best on the prices only data set. Our best result from the prices only data set was 3.31% using SVM with a normalized poly kernel. It is interesting to see that our best result with ADATE is exactly the same result we got with ADATE in our previous work. Although our input data was different, the result was the same.

Table 4.1: Our best results predicting next day NO1 prices

	Model trees	Linear regression	MLP	RBF network	GP	SVM	ADATE
MAPE	3.47%	8.87%	4.28%	3.93%	3.57%	3.14%	3.34%
Data set	Prices	Prices	Prices	Prices	All	All	All

Although 3.14% was our best price prediction result and a support vector machine was the best machine learning method in our experiments, we have also made several other important observations through our work. In our data set we found that in many experiments we got marginally better results when only using price information for input attributes versus using consumption, weather and reservoir input attributes as well. This indicates that we can create reliable and very good predictions based only on historical price information. We have talked about the importance of the

water reservoirs and how they play an important part in the Nordic electricity prices. Through our work we have found that the reservoir data supplied by Statnett is too coarse and thus not vital for making good price predictions using our models. Another feature from our data set that affected the methods predictability was the occurrence of outliers. An outlier represents either a wrong value or a natural price abnormality. These abnormalities affected the results from our model tree, linear regression and multilayer perceptron experiments negatively. This tells us that these mentioned models are unable to handle outliers in the data set. The methods are good at predicting the stable period of the year, but if there are any abnormalities in the data pattern we can not rely on these models. The RBF network, Gaussian process, support vector machines and ADATE handled outliers better and is thus better suited for electricity price prediction.

We also find the results from our experiments with the RBF network interesting. In experiment 2 the RBF network suggests that the change from one day to another is constant. This constant equals the average value of the output values. In this experiment we will have a MAPE of 5.41%. All other methods in experiment 2 that achieves a poorer MAPE than 5.41% is therefore not suited for price prediction. In experiment 3 our RBF network suggests six constants which gives a MAPE of 3.9%. The constant suggested by our RBF network in experiment 2 coincides with the constant that was suggested from our run with ADATE. ADATE also coincides with Cubist in regards to which input attribute that has the most significance in predicting the price. In ADATE we have found that the optimal program created for predicting the price only uses the NO1 price for hour 21 two days before the predicted day when predicting the next day price. This is the same input attribute Cubist and model trees find most valuable for predicting the price. It is interesting that both algorithms have found this price to play an important role when predicting the price.

The difference between experiment 2 and 3 was the historical time period we used for our data set. In experiment 2 we used data from 2001 till 2009 while in experiment 3 we used data from 2005 till 2009. This change resulted in an improvement in our best method¹ from 4.46% to 3.14%. This is a major improvement and there are several possible reasons why the improvement was so drastic. There might be a pattern change, meaning that the data pattern in our earliest data from 2001 - 2004 deviates from the pattern in the test data set. When we changed the time period for our data set, we also had to change the period for our test data. The new test data in experiment 3 was not the same test data set we used in experiment 2. By changing the test data, the results from the test will also change. When we shortened the test data set, the test period might represent a more stable period over time, meaning that the predictions will become better.

¹Support vector machine

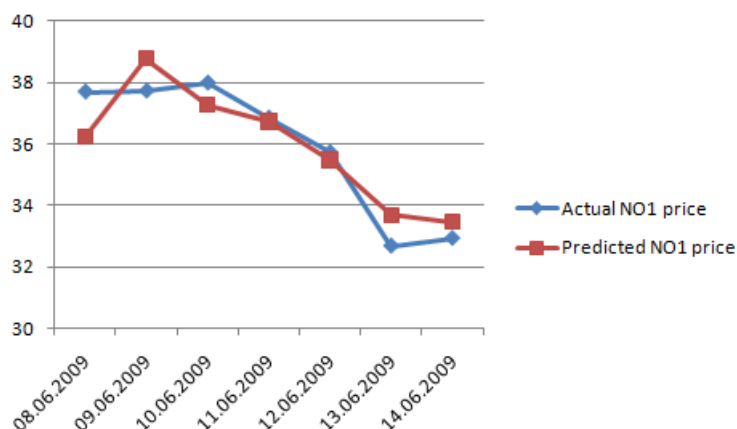


Figure 4.1: SVM predictions for June 8 to June 14 2009

From related work we have found that many models were only tested on a shorter time period. In for instance [32] they had very good predictions using wavelets combined with ARIMA and GARCH, but they only tested their model for four weeks of the year. We have created an example to illustrate this. From our best support vector machine run we have selected one random week from June 2009. A graph showing the predictions and actual values can be seen in figure 4. For our selected week we have a MAPE of 2.04%. This is far better than the 3.14% MAPE we have on the entire test data for this run. This shows that one need to look at a longer time span than just a short selected period. Many of the reports we have cited in our related work section only looks at some random weeks of the year. They get very good results for this selected period, but their results do not tell how good their method is over time. When building a model for price forecasting, the model must give good results over time, not on the odd selected week.

To summarize our findings, we have discovered that support vector machines is the best machine learning method for predicting the electricity price for the NO1 area with our data set. There are minor performance differences between the best methods and we get decent results by only applying a constant day-to-day relative change. Day-ahead electricity price prediction is a vast area to research and there are still many improvements to be made. In our next chapter we will outline some further work we find interesting in the field of electricity price prediction.

Chapter 5

Further work

Price prediction is a vast area which needs further exploration in order to create a very good price analysis. We have only scratched the surface with our project and we hope that this project will inspire other students to pursue the price prediction problematic on Nord Pool Spot. During our work we have identified several factors that are important in doing a good price prediction, but we have also stumbled across several areas which could improve the day ahead prediction, and if we could do this project again, we would most likely choose a different path on our price prediction journey. We will now present some possible future work in electricity price prediction that we have identified through our project.

Predicting several days ahead

Electricity is traded on Nord Pool Spot in the terms of next-day, futures and forwards. We have found that we can make good predictions for the day ahead spot market and we believe that we can use our methods to create predictions for the next days as well. Through our work we have found that price is the most important factor in next-day electricity price prediction. Applying information about the weather and reservoirs does not necessarily improve the day-ahead prediction. This means that we can utilize historical price information in making predictions and not take into account uncertainty about the next weeks weather forecast into our model. It would be interesting to research if support vector machines could create good predictions for the next week, in other words create a seven day price prediction. We would then create a next-day prediction which then again is used as input for the second day and so forth. Having a good model that can compete both in the next-day spot market and the futures market would be of interest to participants concerned with

electricity trading.

Excess or surplus capacity

The price in each price area is determined by the transmission capacity in each area. By calculating the excess or surplus of total available capacity to and from each price area we might see a pattern when capacity is close to the maximum threshold the price will increase[1]. If we were to implement information about consumption in each price area in regards to distance from the maximum threshold we might get an indication of what the next-day price in each area will be. This threshold is also likely to be connected to the weather since temperature will have an affect on how much electricity that will be consumed.

Seasonal changes

The Nordic market is greatly affected by the seasons. Electricity prices are more sensitive to market changes in the winter than in the summer. In the winter we use more energy and the weather will have a large impact on how much electricity we use to heat our homes. In the summer there is less variance in the electricity prices. A solution for this can be to segment the data set in regards to which season one wants to predict. Can we get a better prediction for the summer months by only using input from the same historical summer periods, and vice versa for the winter period? On a daily basis we could also add information about the current day. Is it weekday, weekend or holiday? The usage pattern will be different in those days and the question is if we could further improve the performance of our model by adding this information.

Removing outliers

Through our experiments we have identified some periods where the price differs greatly from one day to another. This can be caused by wrong values in the data set, or natural price peaks. If we remove all the outliers from the data set, we would with some methods increase our performance and probably get an overall better result. Since we are not able to identify the price peaks, we might as well remove them to maximize our prediction for the stable periods of the year. If we look at our experiments we see that linear regression, model trees and multilayer perceptron are vulnerable for price peaks. We have identified a period at the end of July 2009 to be a crucial period with high intra day price changes and poor prediction performance. If we would remove the outliers from our data set we are likely to see an performance increase with these algorithms as well.

Urgent Market Messages - UMM

Both Statnett and Nord Pool Spot provide participants with urgent market messages. In our project we have not done an analysis of information given in UMMs. Typically an UMM contains information about the current load in the grid and the capacities of the electricity providers connected to the grid. The UMMs will also contain information about incidental outages. We are not sure whether the UMMs will aid us in understanding the market better and create better predictions, but it should be looked into if UMM data can contribute with valuable information that can be used to increase the performance of the predictions.

Spike prediction

In our project we have seen that price spikes are very difficult to predict. In [36] they propose a model for predicting price spikes for the National Electricity Market in Australia. They suggest that it is possible to create a price prediction model that handles both regular prices and price spikes using feature selection, statistical analysis and an SVM. In a traders perspective it is the price spikes that are interesting to identify. Knowing when there will be a spike is a major trading advantage. Research should be done to identify all the price spikes in recent history and see if there are any commonalities between the price spikes. If there are, then we can possibly adapt this information to a new price spike prediction model.

Feature selection

In our project we have not used feature selection on our data set to select the optimal attributes to use in our data set. We used many input attributes and in regards of predicting the NO1 price, our data set is likely to have input attributes which are redundant. By doing an attribute selection we are likely to create a data set with fewer input attributes and still get the same results. The optimal program suggested by ADATE for predicting the NO1 price uses only one input variable which renders the other attributes redundant. Some techniques used for feature selection are principal component analysis, correlation analysis, sensitivity analysis and spectrum analysis.

Hybrid network

We only focused on using separate models for our price predictions. We have seen in related work that researchers have combined different models with great success. An interesting research topic in regards to predicting prices at Nord Pool Spot would be to create a hybrid network. In [11] Fan

et al. has created a hybrid network using a self organized map (SOM) and support vector machines in predicting the prices for the New England electricity market. The performance of their hybrid network was better than just using support vector machines. If this can be related to our work, we are likely to improve our results as well.

Committee machines

Since different machine learning methods varies in performance it is not given that any prediction method will out perform all other models on all test cases on the given data set. A committee machine will use the response from multiple networks and weigh their performance in regards to the actual value being predicted. Jau-Jia Guo et al.[13] have created a committee machine using an RBF network and an MLP network and tested it on the New England electricity market. Their committee machine performed better than only using either an RBF or MLP network. In our scenario we could for instance combine the output from support vector machines and Gaussian processes and see if there is an increase in performance.

Test period

In our experiments we found that the performance of our models were influenced by the test period. In experiment 3 our results became better when we used a shorter test period for our predictions. By conducting several experiments where we use a fixed test period prior to the day being predicted, we can determine what our optimal test period should be. For example we could use the 60-150 previous days prior to the day being predicted and see if we get an increase in performance.

Wavelet transform

Finally we believe an interesting research topic would be to use wavelet transforms on our data. In related work we found that researchers have gotten great results using wavelet transform on the input data. In [32] we can see a great performance boost when using wavelet transforms. A wavelet transform is a process where the input data is decomposed and reconstructed into better behaving time series without outliers and a more stable variance. The transformed input data is then used with for instance support vector machines and an inverse wavelet transform is used to reconstruct the estimated prices. It would be interesting to see if we successfully can use wavelet transforms and improve the prediction performance for Nord Pool Spot.

References

- [1] S.K. Aggarwal, L.M. Saini, and A. Kumar. Electricity price forecasting in deregulated markets: A review and evaluation. *International Journal of Electrical Power & Energy Systems*, 31(1):13–22, 2009.
- [2] N. Amjady and F. Keynia. Day ahead price forecasting of electricity markets by a mixed data model and hybrid forecast method. *International Journal of Electrical Power & Energy Systems*, 30(9):533–546, 2008.
- [3] Nord Pool Spot AS. The Nordic Electricity Exchange Nord Pool Spot and the Nordic Model for a Liberalised Electricity Market, 2009.
- [4] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, et al. *Time series analysis: forecasting and control*, volume 16. Holden-day San Francisco, CA, 1976.
- [5] L. Breiman. *Classification and regression trees*. Chapman & Hall/CRC, 1984.
- [6] T. Bye, M. Bj rnsdal, G. Doorman, G. Kj lle, and Riis. C. Flere og riktigere priser, et mer effektivt kraftsystem, 2010.
- [7] R. Callan. *Essence of neural networks*. Prentice Hall PTR Upper Saddle River, NJ, USA, 1998.
- [8] A.J. Conejo, M.A. Plazas, R. Espinola, and A.B. Molina. Day-ahead electricity price forecasting using the wavelet transform and ARIMA models. *Power Systems, IEEE Transactions on*, 20(2):1035–1042, 2005.
- [9] J. Contreras, R. Espinola, F.J. Nogales, and A.J. Conejo. ARIMA models to predict next-day electricity prices. *IEEE transactions on power systems*, 18(3):1014–1020, 2003.

- [10] J. Crespo Cuaresma, J. Hlouskova, S. Kossmeier, and M. Obersteiner. Forecasting electricity spot-prices using linear univariate time-series models. *Applied Energy*, 77(1):87–106, 2004.
- [11] S. Fan, C. Mao, and L. Chen. Next-day electricity-price forecasting using a hybrid network. *Generation, Transmission & Distribution, IET*, 1(1):176–182, 2007.
- [12] R.C. Garcia, J. Contreras, M. Van Akkeren, and J.B.C. Garcia. A GARCH forecasting model to predict day-ahead electricity prices. *IEEE Transactions on Power Systems*, 20(2):867–874, 2005.
- [13] J.J. Guo and P.B. Luh. Improving Market Clearing Price Prediction by Using a Committee Machine of Neural Networks. *IEEE TRANSACTIONS ON POWER SYSTEMS*, 19(4):1867, 2004.
- [14] V. Jakkula. Tutorial on Support Vector Machine (SVM). *School of EECS, Washington State University*, 2006.
- [15] C. Kim, I.K. Yu, and YH Song. Prediction of system marginal price of electricity using wavelet transform analysis. *Energy Conversion and Management*, 43(14):1839–1851, 2002.
- [16] E. Livanis and A. Zaprani. Forecasting the Day-Ahead Electricity Price in Nord Pool with Neural Networks: Some Preliminary Results.
- [17] P. Mandal, A.K. Srivastava, T. Senjyu, and M. Negnevitsky. A new recursive neural network algorithm to forecast electricity price for PJM day-ahead market. *International Journal of Energy Research*, 34(6):507–522, 2010.
- [18] Microsoft. Smart Energy Reference Architecture, 2009.
- [19] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.
- [20] F.J. Nogales, J. Contreras, A.J. Conejo, and R. Espínola. Forecasting next-day electricity prices by time series models. *Power Systems, IEEE Transactions on*, 17(2):342–348, 2002.
- [21] S. Nome. Prispognooser og produksjonsplanlegging for et vannkraftverk, 2010.
- [22] R. Pino, J. Parreno, A. Gomez, and P. Priore. Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks. *Engineering Applications of Artificial Intelligence*, 21(1):53–62, 2008.

- [23] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines, 1998.
- [24] Montel Powernews. Montel Powernews magazine December 2009, 2009.
- [25] J.R. Quinlan. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, pages 343–348. Citeseer, 1992.
- [26] C.E. Rasmussen. Gaussian processes in machine learning. *Advanced Lectures on Machine Learning*, pages 63–71, 2004.
- [27] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- [28] M. Rosenblatt. A central limit theorem and a strong mixing condition. *Proceedings of the National Academy of Sciences of the United States of America*, 42(1):43, 1956.
- [29] A.J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.
- [30] D. Steinberg and P. Colla. CARTclassification and regression trees. *San Diego, CA: Salford Systems*, 7, 1997.
- [31] P. Subbaraj and V. Rajasekaran. Evolutionary Techniques Based Combined Artificial Neural Networks for Peak Load Forecasting. *International Journal of Electrical Power and Energy Systems Engineering*, 2008.
- [32] Z. Tan, J. Zhang, J. Wang, and J. Xu. Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models. *Applied Energy*, 87(11):3606–3610, 2010.
- [33] G. Vattekar. Adate User Manual. Technical report, Technical report, Ostfold University College, 2006.
- [34] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann Pub, 2005.

- [35] H. Yang and X. Duan. Chaotic characteristics of electricity price and its forecasting model. In *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on*, volume 1, pages 659–662. IEEE, 2005.
- [36] J.H. Zhao, Z.Y. Dong, X. Li, and K.P. Wong. A general method for electricity market price spike analysis. In *Power Engineering Society General Meeting, 2005. IEEE*, pages 286–293. IEEE, 2005.

List of Figures

2.1	The different price areas traded at Nord Pool Spot	5
2.2	Illustration of supply and demand curve	6
2.3	Example of a model tree taken from [34]	15
2.4	Illustration of a linear regression model	16
2.5	Illustration of a neural network with n input, one hidden and one output layer.	17
2.6	Illustration a Radial Basis Function network	20
2.7	Illustration of data divided with a high margin	21
2.8	Figure of the ADATE search process as seen in the ADATE user manual	23
3.1	Cubist results using 7 rules predicting the NO1 price for 2009	41
3.2	Cubist results using 5 rules predicting the NO1 price for 2009	45
3.3	Linear regression for the NO1 price for 2009	47
3.4	MLP for the NO1 price for 2009	48
3.5	RBF network for the NO1 price for 2009	50
3.6	Gaussian processes for the NO1 price for 2009	51
3.7	Comparison of the kernels predicting the NO1 price for 2009	53
3.8	SVM with the RBF kernel for predicting the NO1 price for 2009	53
3.9	Cubist all attributes for predicting the NO1 price for 2009	55
3.10	Price prediction with linear regression for 2009	56
3.11	MLP for the NO1 price for 2009	57
3.12	RBF network for the NO1 price for 2009	58
3.13	Gaussian processes for the NO1 price for 2009	59
3.14	Price prediction with the RBF kernel for 2009	60
3.15	The optimal program suggested from ADATE for the NO1 price prediction	61

4.1 SVM predictions for June 8 to June 14 2009 66

List of Tables

2.1	Weather stations used in this project.	25
2.2	Results for price prediction conducted in previous work	29
2.3	Results for price prediction conducted in previous work, ARMA=Autoregressive Moving Average, DR=Dynamic Regression, TF=Transfer Function, WT=Wavelet Transform	32
3.1	Variables used in experiment 1, H=hourly, D=daily	39
3.2	Our first runs using Cubist with hourly resolution showing MAE for the training data and MAE and MAPE for test data	39
3.3	Results using 1-10 rules in Cubist showing MAPE	40
3.4	Variables used in experiment 2, D=daily, H=hourly, prev=previous, *=input from two weather stations, **=both content and inflow	43
3.5	Predicting NO1 using Cubist and model trees	44
3.6	Using rules on the reduced data set gives the following results	45
3.7	Predicting NO1 using WEKA and linear regression, showing MAE and MAPE on test data	46
3.8	Predicting NO1 using WEKA and multilayer perceptron	48
3.9	Predicting NO1 using a Radial Basis Function network	49
3.10	Predicting NO1 using WEKA and GaussianProcesses with Normalized Poly kernel	50
3.11	Predicting NO1 using WEKAs SMOreg using the Poly kernel	51
3.12	Predicting NO1 price using WEKAs SMOregs different kernels	52
3.13	Predicting NO1 using Cubist and model trees	55
3.14	Predicting NO1 using linear regression	56
3.15	Predicting NO1 using multilayer perceptrons	57
3.16	Predicting NO1 using WEKA and RBFNetwork	58

3.17 Predicting NO1 using Gaussian Processes with Normalized Poly kernel	59
3.18 Predicting NO1 price using WEKAs SMOregs different kernels	60
4.1 Our best results predicting next day NO1 prices	64

Appendix A

DVD

We have appended our additional data on a DVD. On the DVD we have added all our data sets, results from all the experiments and the most important code used for converting our data. Due to the size of the output from our experiments, we have added all our results to the DVD.

The following structure is on the appendix DVD:

- Code (Folder): Containing five C# projects used in hour data gathering and data pre-processing.
- Data set (Folder): For each experiment there is a folder containing the data sets used in each experiment.
- Results (Folder): For each experiment there is a folder containing the results for each experiment.

Appendix B

Results

In this paper we have only added the best results for experiments 1 and 2 . The output from Cubist and our 1. experiment is to comprehensive to be added to the appendix. The other results can be found on the DVD.

B.1 Experiment 2

Our best results from our second experiment was with support vector machines using the normalized poly kernel on the prices only data set. In the first column we see the day tested, the second column shows the actual value and the third column shows the predicted value. The fifth column shows the absolute difference between the actual and the predicted value. Since the values are relative the absolute difference is the a percentage deviation.

Listing B.1: Experiment 2 SVM with normalized polykernel prices only

inst#.	actual	predicted	error		
1	0,876	0,96	0,083	0,084	0,095890411
2	1,12	1,105	-0,014	0,015	0,013392857
3	0,965	0,897	-0,068	0,068	0,070466321
4	0,955	0,969	0,014	0,014	0,014659686
5	0,982	0,994	0,013	0,012	0,012219959
6	1,041	0,948	-0,093	0,093	0,089337176
7	1,025	1,085	0,06	0,06	0,058536585
8	1,034	0,957	-0,077	0,077	0,074468085
9	0,96	0,95	-0,01	0,01	0,010416667
10	1,035	0,97	-0,065	0,065	0,062801932
11	0,983	1,024	0,041	0,041	0,041709054
12	0,99	1,043	0,053	0,053	0,053535354
13	1,149	1,075	-0,074	0,074	0,064403829
14	0,968	1,005	0,038	0,037	0,03822314
15	0,975	0,976	0,001	0,001	0,001025641
16	0,978	0,984	0,006	0,006	0,006134969
17	0,958	0,996	0,039	0,038	0,039665971

18	0,967	0,929	-0,038	0,038	0,039296794
19	0,965	0,953	-0,012	0,012	0,012435233
20	1,002	1,049	0,047	0,047	0,046906188
21	1,036	1,025	-0,011	0,011	0,010617761
22	1,047	0,986	-0,061	0,061	0,0582617
23	0,997	1,009	0,012	0,012	0,012036108
24	0,965	1,006	0,041	0,041	0,042487047
25	0,95	0,933	-0,018	0,017	0,017894737
26	1,005	0,959	-0,047	0,046	0,045771144
27	1,001	1,025	0,024	0,024	0,023976024
28	1,05	1,01	-0,04	0,04	0,038095238
29	0,987	0,997	0,01	0,01	0,010131712
30	0,947	0,999	0,051	0,052	0,054910243
31	0,952	0,973	0,021	0,021	0,022058824
32	0,97	0,985	0,015	0,015	0,015463918
33	0,972	0,994	0,022	0,022	0,022633745
34	1,045	1,017	-0,028	0,028	0,026794258
35	0,926	0,982	0,056	0,056	0,060475162
36	0,976	0,953	-0,023	0,023	0,023565574
37	0,969	1,043	0,074	0,074	0,076367389
38	0,983	0,986	0,003	0,003	0,003051882
39	0,932	0,921	-0,011	0,011	0,011802575
40	0,999	0,985	-0,014	0,014	0,014014014
41	1,092	1,117	0,025	0,025	0,022893773
42	0,967	0,98	0,013	0,013	0,01344364
43	1,027	0,999	-0,027	0,028	0,027263875
44	0,969	0,959	-0,01	0,01	0,010319917
45	1,035	0,988	-0,047	0,047	0,045410628
46	0,955	0,984	0,029	0,029	0,030366492
47	0,985	0,976	-0,009	0,009	0,009137056
48	1,2	1,092	-0,108	0,108	0,09
49	0,996	1,025	0,029	0,029	0,029116466
50	1,098	0,994	-0,104	0,104	0,094717668
51	1,14	0,924	-0,216	0,216	0,189473684
52	0,768	0,896	0,128	0,128	0,166666667
53	0,954	1,002	0,049	0,048	0,050314465
54	0,955	1,088	0,132	0,133	0,139267016
55	1,137	1,029	-0,108	0,108	0,094986807
56	0,987	1,003	0,015	0,016	0,01621074
57	0,956	0,977	0,02	0,021	0,021966527
58	1,067	0,98	-0,087	0,087	0,08153702
59	0,916	0,981	0,065	0,065	0,070960699
60	0,92	0,944	0,024	0,024	0,026086957
61	1,014	1,024	0,01	0,01	0,009861933
62	1,09	1,087	-0,003	0,003	0,002752294
63	1,064	1,028	-0,036	0,036	0,033834586
64	1,03	1,011	-0,02	0,019	0,018446602
65	1,008	0,998	-0,01	0,01	0,009920635
66	0,963	0,91	-0,053	0,053	0,055036345
67	0,958	0,978	0,02	0,02	0,020876827
68	0,99	0,976	-0,014	0,014	0,014141414
69	1,145	1,028	-0,117	0,117	0,102183406
70	0,907	0,961	0,054	0,054	0,059536935
71	0,977	0,965	-0,013	0,012	0,012282497
72	0,989	1	0,011	0,011	0,011122346
73	0,969	1,017	0,049	0,048	0,049535604
74	0,96	0,95	-0,01	0,01	0,010416667
75	1,011	1,034	0,023	0,023	0,022749753
76	1,094	1,119	0,025	0,025	0,02285192
77	1,096	0,991	-0,105	0,105	0,09580292
78	0,939	0,971	0,032	0,032	0,034078807
79	1,081	0,991	-0,089	0,09	0,083256244
80	0,92	0,914	-0,006	0,006	0,006521739
81	0,936	0,968	0,032	0,032	0,034188034

82	0,976	1,004	0,027	0,028	0,028688525
83	1,083	1,081	-0,002	0,002	0,001846722
84	0,989	0,997	0,007	0,008	0,008088979
85	0,95	0,975	0,025	0,025	0,026315789
86	1,014	0,975	-0,038	0,039	0,038461538
87	0,962	0,977	0,015	0,015	0,015592516
88	0,968	0,986	0,018	0,018	0,018595041
89	0,979	1,023	0,044	0,044	0,04494382
90	1,076	1,033	-0,043	0,043	0,039962825
91	0,965	1	0,034	0,035	0,03626943
92	1,005	0,973	-0,031	0,032	0,031840796
93	0,978	0,979	0,001	0,001	0,001022495
94	0,969	1,021	0,052	0,052	0,053663571
95	0,945	0,964	0,019	0,019	0,02010582
96	0,964	0,971	0,006	0,007	0,007261411
97	1,086	1,079	-0,007	0,007	0,006445672
98	0,993	1,018	0,025	0,025	0,025176234
99	1,002	0,977	-0,025	0,025	0,0249501
100	0,994	0,995	0,001	0,001	0,001006036
101	0,968	0,981	0,013	0,013	0,013429752
102	0,952	0,94	-0,012	0,012	0,012605042
103	0,915	1,044	0,129	0,129	0,140983607
104	1,206	1,136	-0,07	0,07	0,058043118
105	1,037	1,019	-0,018	0,018	0,017357763
106	0,977	0,957	-0,02	0,02	0,020470829
107	0,994	0,966	-0,028	0,028	0,028169014
108	0,961	1,01	0,049	0,049	0,050988554
109	0,935	0,935	0	0	0
110	0,982	0,986	0,003	0,004	0,00407332
111	1,081	1,058	-0,023	0,023	0,021276596
112	1,017	0,998	-0,019	0,019	0,018682399
113	0,968	0,981	0,013	0,013	0,013429752
114	0,984	0,976	-0,007	0,008	0,008130081
115	0,989	0,995	0,005	0,006	0,006066734
116	1,011	0,97	-0,041	0,041	0,040553907
117	0,987	1,025	0,038	0,038	0,038500507
118	1,046	1,027	-0,019	0,019	0,018164436
119	1,026	1,003	-0,023	0,023	0,022417154
120	0,978	1,018	0,04	0,04	0,040899796
121	0,975	0,975	0	0	0
122	0,982	0,985	0,002	0,003	0,00305499
123	0,989	1,006	0,017	0,017	0,01718908
124	0,98	0,999	0,019	0,019	0,019387755
125	0,99	1,022	0,032	0,032	0,032323232
126	1,14	1,053	-0,087	0,087	0,076315789
127	0,988	0,981	-0,007	0,007	0,00708502
128	1,025	0,976	-0,048	0,049	0,047804878
129	0,952	0,998	0,046	0,046	0,048319328
130	0,951	0,977	0,027	0,026	0,027339642
131	0,935	0,976	0,041	0,041	0,043850267
132	1,083	1,024	-0,059	0,059	0,054478301
133	1,008	0,995	-0,013	0,013	0,012896825
134	1,021	1,006	-0,016	0,015	0,014691479
135	0,988	1,017	0,029	0,029	0,029352227
136	0,993	0,978	-0,016	0,015	0,01510574
137	1,001	0,987	-0,014	0,014	0,013986014
138	0,957	0,995	0,038	0,038	0,039707419
139	1,089	1,019	-0,069	0,07	0,064279155
140	0,992	0,998	0,006	0,006	0,006048387
141	0,965	1,002	0,037	0,037	0,038341969
142	0,956	0,975	0,019	0,019	0,019874477
143	0,945	0,992	0,047	0,047	0,04973545
144	0,931	0,982	0,051	0,051	0,054779807
145	0,919	0,922	0,003	0,003	0,003264418

146	1,149	1,077	-0,072	0,072	0,062663185
147	0,912	1,031	0,119	0,119	0,130482456
148	1,125	0,98	-0,145	0,145	0,128888889
149	1,026	0,991	-0,035	0,035	0,03411306
150	0,999	1,001	0,002	0,002	0,002002002
151	0,913	0,964	0,051	0,051	0,055859803
152	1,036	0,991	-0,045	0,045	0,043436293
153	1,117	1,087	-0,03	0,03	0,026857654
154	0,99	1,013	0,023	0,023	0,023232323
155	0,967	0,959	-0,008	0,008	0,008273009
156	1,06	1,046	-0,015	0,014	0,013207547
157	0,964	0,983	0,019	0,019	0,019709544
158	0,936	0,943	0,006	0,007	0,007478632
159	0,92	0,985	0,065	0,065	0,070652174
160	1,159	1,125	-0,034	0,034	0,029335634
161	1,016	0,991	-0,026	0,025	0,024606299
162	1,049	0,944	-0,105	0,105	0,100095329
163	0,848	1,004	0,155	0,156	0,183962264
164	1,042	1,025	-0,017	0,017	0,016314779
165	0,94	0,951	0,011	0,011	0,011702128
166	0,983	0,993	0,01	0,01	0,01017294
167	1,191	1,158	-0,034	0,033	0,027707809
168	0,958	0,964	0,006	0,006	0,006263048
169	0,995	0,951	-0,043	0,044	0,044221106
170	0,942	1,036	0,094	0,094	0,099787686
171	0,963	1,051	0,088	0,088	0,091381101
172	0,987	0,966	-0,021	0,021	0,021276596
173	0,905	0,92	0,015	0,015	0,016574586
174	1,197	1,079	-0,118	0,118	0,098579783
175	1,089	1,095	0,006	0,006	0,005509642
176	1,019	0,977	-0,042	0,042	0,041216879
177	0,843	0,933	0,09	0,09	0,106761566
178	1,081	0,997	-0,084	0,084	0,077705828
179	0,931	0,957	0,026	0,026	0,02792696
180	0,836	0,969	0,133	0,133	0,159090909
181	1,107	1,045	-0,062	0,062	0,056007227
182	0,969	1,004	0,035	0,035	0,036119711
183	1,108	1,047	-0,061	0,061	0,055054152
184	0,987	0,995	0,008	0,008	0,00810537
185	0,979	0,964	-0,015	0,015	0,015321757
186	1,02	1,005	-0,016	0,015	0,014705882
187	0,957	0,975	0,018	0,018	0,018808777
188	1,083	1,057	-0,025	0,026	0,024007387
189	1,051	1,032	-0,019	0,019	0,018078021
190	1,041	0,985	-0,056	0,056	0,053794428
191	1,024	0,953	-0,071	0,071	0,069335938
192	1,015	1	-0,015	0,015	0,014778325
193	0,983	1,006	0,023	0,023	0,023397762
194	0,975	0,968	-0,006	0,007	0,007179487
195	1,093	1,053	-0,04	0,04	0,036596523
196	1,018	1,038	0,02	0,02	0,019646365
197	0,984	0,975	-0,009	0,009	0,009146341
198	1,034	1,019	-0,015	0,015	0,01450677
199	0,988	1,028	0,04	0,04	0,04048583
200	0,97	0,934	-0,035	0,036	0,037113402
201	0,985	0,996	0,011	0,011	0,011167513
202	1,066	1,117	0,05	0,051	0,047842402
203	0,98	0,997	0,017	0,017	0,017346939
204	0,999	0,975	-0,024	0,024	0,024024024
205	1,013	0,966	-0,047	0,047	0,046396841
206	1,007	1,014	0,006	0,007	0,006951341
207	0,941	0,953	0,013	0,012	0,012752391
208	0,971	1,016	0,045	0,045	0,046343975
209	1,057	1,017	-0,04	0,04	0,037842952

210	0,979	1,016	0,036	0,037	0,037793667
211	0,943	1,01	0,066	0,067	0,071049841
212	0,951	0,942	-0,009	0,009	0,009463722
213	0,743	0,965	0,223	0,222	0,298788694
214	1,228	1,078	-0,15	0,15	0,122149837
215	0,64	0,833	0,193	0,193	0,3015625
216	1,58	1,382	-0,198	0,198	0,125316456
217	1,074	1,27	0,196	0,196	0,182495345
218	0,964	1,184	0,22	0,22	0,228215768
219	0,873	1,008	0,135	0,135	0,154639175
220	1,004	1,013	0,01	0,009	0,008964143
221	0,914	0,867	-0,047	0,047	0,051422319
222	0,891	0,928	0,037	0,037	0,041526375
223	1,071	1,047	-0,024	0,024	0,022408964
224	0,837	1,038	0,201	0,201	0,240143369
225	0,582	1,049	0,467	0,467	0,802405498
226	0,302	1,013	0,711	0,711	2,354304636
227	1,073	?		#VERDI!	#VERDI!
228	1,433	0,889	-0,544	0,544	0,379623168
229	1,08	1,169	0,089	0,089	0,082407407
230	1,605	1,153	-0,452	0,452	0,281619938
231	1,346	0,982	-0,364	0,364	0,270430906
232	1,153	1,002	-0,151	0,151	0,130962706
233	1,059	1,127	0,068	0,068	0,06421152
234	0,88	0,97	0,09	0,09	0,102272727
235	0,819	0,897	0,078	0,078	0,095238095
236	0,326	0,948	0,622	0,622	1,90797546
237	0,885	?		#VERDI!	#VERDI!
238	2,225	1,179	-1,046	1,046	0,47011236
239	1,568	1,033	-0,535	0,535	0,34119898
240	0,713	1,083	0,37	0,37	0,518934081
241	0,959	0,998	0,039	0,039	0,040667362
242	0,928	0,893	-0,035	0,035	0,037715517
243	1,062	0,936	-0,126	0,126	0,118644068
244	1,38	1,033	-0,348	0,347	0,251449275
245	0,764	1,131	0,367	0,367	0,480366492
246	0,627	1,003	0,376	0,376	0,599681021
247	1,086	1,07	-0,016	0,016	0,014732965
248	1,014	0,908	-0,106	0,106	0,104536489
249	1,062	1,015	-0,047	0,047	0,044256121
250	1,033	1,159	0,126	0,126	0,121974831
251	1,397	1,433	0,036	0,036	0,025769506
252	1,173	0,931	-0,242	0,242	0,20630861
253	1,077	0,924	-0,153	0,153	0,142061281
254	0,603	0,953	0,35	0,35	0,580431177
255	0,653	1,01	0,357	0,357	0,546707504
256	0,785	0,962	0,177	0,177	0,225477707
257	1,003	0,938	-0,065	0,065	0,064805583
258	1,309	1,088	-0,221	0,221	0,168831169
259	1,182	0,988	-0,194	0,194	0,164128596
260	0,982	1,023	0,04	0,041	0,041751527
261	1,077	1,005	-0,072	0,072	0,066852368
262	1,12	1,004	-0,116	0,116	0,103571429
263	1,155	1,002	-0,153	0,153	0,132467532
264	1,15	1,112	-0,038	0,038	0,033043478
265	1,421	1,217	-0,204	0,204	0,143560873
266	1,294	1,035	-0,259	0,259	0,20015456
267	1,155	1,023	-0,133	0,132	0,114285714
268	1,129	1,047	-0,082	0,082	0,072630647
269	1,055	1,023	-0,032	0,032	0,030331754
270	1,033	0,963	-0,07	0,07	0,067763795
271	1,026	1,043	0,017	0,017	0,016569201
272	1,034	1,067	0,033	0,033	0,031914894
273	1,112	1,067	-0,045	0,045	0,040467626

274	1,058	0,984	-0,074	0,074	0,069943289
275	1,065	0,984	-0,081	0,081	0,076056338
276	1,081	1,013	-0,068	0,068	0,062904718
277	1,038	1,011	-0,027	0,027	0,026011561
278	0,985	1,024	0,039	0,039	0,039593909
279	1,049	1,026	-0,023	0,023	0,021925643
280	0,949	1,006	0,057	0,057	0,060063224
281	1,003	0,985	-0,018	0,018	0,017946162
282	1,018	0,991	-0,028	0,027	0,026522593
283	0,993	0,999	0,006	0,006	0,006042296
284	0,957	0,967	0,01	0,01	0,010449321
285	0,992	0,997	0,005	0,005	0,005040323
286	1,007	0,999	-0,008	0,008	0,007944389
287	1,041	1,023	-0,018	0,018	0,017291066
288	0,996	1,007	0,011	0,011	0,011044177
289	0,951	0,972	0,021	0,021	0,022082019
290	1,035	0,971	-0,064	0,064	0,061835749
291	0,915	0,989	0,074	0,074	0,080874317
292	0,925	0,984	0,059	0,059	0,063783784
293	1,051	0,992	-0,059	0,059	0,056137012
294	0,945	1,015	0,071	0,07	0,074074074
295	0,986	1,018	0,032	0,032	0,032454361
296	1,014	0,963	-0,051	0,051	0,050295858
297	1,087	1,008	-0,08	0,079	0,072677093
298	1,036	0,993	-0,043	0,043	0,041505792
299	1,072	1,012	-0,061	0,06	0,055970149
300	1,049	1,034	-0,016	0,015	0,014299333
301	1,005	1,014	0,009	0,009	0,008955224
302	1,043	1,003	-0,04	0,04	0,038350911
303	1,085	0,987	-0,098	0,098	0,090322581
304	1,096	1,038	-0,058	0,058	0,052919708
305	1,036	1,029	-0,007	0,007	0,006756757
306	1,006	1,006	0	0	0
307	1,002	1,034	0,032	0,032	0,031936128
308	1,033	1,017	-0,016	0,016	0,015488867
309	1,047	0,999	-0,048	0,048	0,045845272
310	1,089	1,014	-0,075	0,075	0,068870523
311	0,997	1,04	0,043	0,043	0,043129388
312	1,062	1,013	-0,049	0,049	0,04613936
313	1,002	1	-0,003	0,002	0,001996008
314	1,054	1,055	0,001	0,001	0,000948767
315	1,04	0,997	-0,043	0,043	0,041346154
316	1,011	0,999	-0,012	0,012	0,011869436
317	1,04	0,994	-0,046	0,046	0,044230769
318	1,035	1,021	-0,014	0,014	0,01352657
319	1,037	1,012	-0,025	0,025	0,024108004
320	0,992	1,017	0,025	0,025	0,025201613
321	1,028	1,021	-0,006	0,007	0,006809339
322	1,008	0,998	-0,01	0,01	0,009920635
323	0,997	0,989	-0,007	0,008	0,008024072
324	1,035	0,991	-0,044	0,044	0,042512077
325	1,004	1,028	0,024	0,024	0,023904382
326	0,998	0,988	-0,009	0,01	0,01002004
327	0,959	0,993	0,034	0,034	0,035453597
328	1,037	0,974	-0,063	0,063	0,06075217
329	0,978	1,04	0,061	0,062	0,063394683
330	0,998	1,038	0,04	0,04	0,04008016
331	0,974	0,956	-0,019	0,018	0,018480493
332	0,988	0,98	-0,008	0,008	0,008097166
333	0,965	0,99	0,025	0,025	0,025906736
334	1,025	0,997	-0,028	0,028	0,027317073
335	1,036	1,047	0,011	0,011	0,010617761
336	1,01	0,991	-0,019	0,019	0,018811881
337	1,005	0,99	-0,015	0,015	0,014925373

338	1,006	0,976	-0,03	0,03	0,029821074
339	0,982	0,982	0	0	0
340	1,016	1,001	-0,015	0,015	0,01476378
341	1,009	1,017	0,007	0,008	0,007928642
342	1,032	1,015	-0,017	0,017	0,016472868
343	1,021	1,004	-0,017	0,017	0,016650343
344	1,005	1,027	0,023	0,022	0,021890547
345	1,032	1,01	-0,021	0,022	0,021317829
346	0,964	0,992	0,028	0,028	0,029045643
347	1,002	0,959	-0,043	0,043	0,042914172
348	0,973	0,984	0,011	0,011	0,011305242
349	1,05	1,01	-0,04	0,04	0,038095238
350	1,009	1,01	0,001	0,001	0,00099108
351	1,006	1,007	0,002	0,001	0,000994036
352	1,004	1,004	0,001	0	0
353	1,016	0,993	-0,023	0,023	0,022637795
354	0,985	0,999	0,014	0,014	0,014213198
355	0,985	0,976	-0,009	0,009	0,009137056
356	1,035	1,019	-0,016	0,016	0,015458937
357	1,059	1,052	-0,008	0,007	0,006610009
358	0,963	0,968	0,006	0,005	0,005192108
359	0,984	0,962	-0,022	0,022	0,022357724
360	0,98	0,974	-0,005	0,006	0,006122449
361	0,953	0,982	0,029	0,029	0,03043022
362	0,981	0,986	0,005	0,005	0,00509684
363	1,039	1,021	-0,018	0,018	0,01732435
364	0,998	1,004	0,006	0,006	0,006012024
365	0,987	1,01	0,023	0,023	0,023302938
366	0,983	0,959	-0,023	0,024	0,024415056
367	0,975	0,984	0,009	0,009	0,009230769
368	1,008	0,996	-0,011	0,012	0,011904762
369	1,004	1,002	-0,002	0,002	0,001992032
370	1,054	1,012	-0,042	0,042	0,039848197
371	1,026	1,005	-0,021	0,021	0,020467836
372	1,009	1,015	0,006	0,006	0,005946482
373	0,996	0,996	0	0	0
374	1,019	0,999	-0,02	0,02	0,019627085
375	0,984	0,987	0,003	0,003	0,00304878
376	0,987	1,01	0,022	0,023	0,023302938
377	1,019	1,021	0,001	0,002	0,001962709
378	1,017	1,027	0,011	0,01	0,009832842
379	0,979	0,974	-0,005	0,005	0,005107252
380	0,982	0,982	0	0	0
381	0,985	1,007	0,022	0,022	0,022335025
382	0,973	0,988	0,014	0,015	0,015416238
383	0,996	0,959	-0,037	0,037	0,037148594
384	0,993	0,995	0,001	0,002	0,002014099
385	0,999	0,972	-0,027	0,027	0,027027027
386	1,004	0,989	-0,015	0,015	0,014940239
387	0,996	1,012	0,016	0,016	0,016064257
388	1,004	0,997	-0,007	0,007	0,006972112
389	0,981	0,993	0,012	0,012	0,012232416
390	1	0,986	-0,013	0,014	0,014
391	1,041	1,005	-0,036	0,036	0,034582133
392	1,015	0,981	-0,034	0,034	0,033497537
393	0,989	0,98	-0,009	0,009	0,009100101
394	0,983	0,949	-0,034	0,034	0,034587996
395	1,012	1,006	-0,006	0,006	0,005928854
396	1,045	1,051	0,006	0,006	0,005741627
397	0,962	0,995	0,034	0,033	0,034303534
398	0,985	0,975	-0,009	0,01	0,010152284
399	1,005	0,987	-0,018	0,018	0,017910448
400	1,006	1,003	-0,003	0,003	0,002982107
401	0,983	1,002	0,019	0,019	0,019328586

402	1	0,993	-0,007	0,007	0,007
403	1,017	1,008	-0,009	0,009	0,008849558
404	0,972	0,997	0,025	0,025	0,025720165
405	1	0,977	-0,022	0,023	0,023
406	0,999	0,999	0	0	0
407	0,983	1,002	0,019	0,019	0,019328586
408	0,959	0,971	0,013	0,012	0,012513034
409	1,002	0,988	-0,015	0,014	0,013972056
410	1,021	1,013	-0,008	0,008	0,007835455
411	0,992	1,026	0,034	0,034	0,034274194
412	0,993	1,001	0,007	0,008	0,008056395
413	0,974	0,973	0	0,001	0,001026694
414	0,972	0,975	0,003	0,003	0,00308642
415	0,969	0,965	-0,004	0,004	0,004127967
416	1,005	0,98	-0,025	0,025	0,024875622
417	1,052	1,064	0,012	0,012	0,011406844
418	0,966	0,993	0,026	0,027	0,027950311
419	1,001	0,975	-0,026	0,026	0,025974026
420	0,984	0,994	0,01	0,01	0,010162602
421	0,969	0,971	0,002	0,002	0,002063983
422	1,007	0,966	-0,042	0,041	0,040714995
423	1,001	1,038	0,037	0,037	0,036963037
424	1,021	1,026	0,006	0,005	0,00489716
425	0,986	1,013	0,028	0,027	0,027383367
426	0,985	0,966	-0,019	0,019	0,01928934
427	0,975	0,974	-0,001	0,001	0,001025641
428	1,022	1,004	-0,018	0,018	0,017612524
429	0,977	0,969	-0,007	0,008	0,008188332
430	0,995	0,993	-0,001	0,002	0,00201005
431	1,038	1,034	-0,004	0,004	0,003853565
432	1,019	1,003	-0,016	0,016	0,015701668
433	0,999	0,998	-0,001	0,001	0,001001001
434	0,999	0,999	0	0	0
435	0,985	1,01	0,025	0,025	0,025380711
436	0,979	0,991	0,012	0,012	0,012257406
437	0,991	0,959	-0,032	0,032	0,032290616
438	1,007	0,981	-0,026	0,026	0,025819265
439	1,01	1,014	0,003	0,004	0,003960396
440	0,979	1,001	0,023	0,022	0,02247191
441	0,951	0,985	0,033	0,034	0,03575184
442	0,929	0,95	0,021	0,021	0,022604952
443	0,956	0,941	-0,015	0,015	0,015690377
444	0,97	0,978	0,008	0,008	0,008247423
445	1,051	1,016	-0,035	0,035	0,033301618
446	0,953	1,011	0,058	0,058	0,060860441
447	0,929	0,967	0,038	0,038	0,040904198
448	1,031	0,976	-0,055	0,055	0,053346266
449	1,005	0,993	-0,013	0,012	0,011940299
450	0,938	0,985	0,047	0,047	0,05010661
451	0,972	0,984	0,012	0,012	0,012345679
452	1,109	1,051	-0,058	0,058	0,052299369
453	0,981	0,996	0,014	0,015	0,01529052
454	0,979	0,993	0,014	0,014	0,014300306
455	0,98	0,969	-0,011	0,011	0,01122449
456	0,988	0,977	-0,011	0,011	0,011133603
457	0,941	0,991	0,05	0,05	0,053134963
458	0,971	0,962	-0,01	0,009	0,009268795
459	1,063	1,034	-0,029	0,029	0,027281279
460	0,995	1,009	0,014	0,014	0,014070352
461	0,971	0,969	-0,002	0,002	0,002059732
462	0,946	0,987	0,041	0,041	0,043340381
463	0,979	0,985	0,006	0,006	0,006128703
464	0,999	0,98	-0,019	0,019	0,019019019
465	0,983	0,988	0,005	0,005	0,00508647

466	1,016	1,031	0,016	0,015	0,01476378
467	1,011	1,006	-0,004	0,005	0,004945598
468	1,011	0,976	-0,035	0,035	0,034619189
469	1,041	0,994	-0,048	0,047	0,045148895
470	0,996	1,019	0,023	0,023	0,023092369
471	1	0,994	-0,005	0,006	0,006
472	1,002	1,003	0,001	0,001	0,000998004
473	1,005	1,007	0,002	0,002	0,00199005
474	0,99	0,997	0,007	0,007	0,007070707
475	0,993	1,005	0,011	0,012	0,012084592
476	0,986	0,949	-0,036	0,037	0,037525355
477	1,011	0,986	-0,026	0,025	0,024727992
478	0,971	0,98	0,009	0,009	0,009268795
479	0,979	0,963	-0,016	0,016	0,016343207
480	1,023	0,993	-0,03	0,03	0,029325513
481	1,001	1,021	0,019	0,02	0,01998002
482	1,007	0,993	-0,014	0,014	0,013902681
483	0,958	0,975	0,017	0,017	0,017745303
484	0,986	0,983	-0,003	0,003	0,003042596
485	0,995	0,984	-0,011	0,011	0,011055276
486	1,007	0,999	-0,009	0,008	0,007944389
487	1,016	0,999	-0,017	0,017	0,016732283
488	1,002	1,014	0,013	0,012	0,011976048
489	1,061	1,002	-0,059	0,059	0,055607917
490	1,083	0,968	-0,115	0,115	0,106186519
491	0,981	0,974	-0,007	0,007	0,007135576
492	1,006	1,029	0,023	0,023	0,022862823
493	1,01	1,027	0,017	0,017	0,016831683
494	1,096	1,021	-0,075	0,075	0,068430657
495	0,98	1,014	0,035	0,034	0,034693878
496	1,068	0,964	-0,104	0,104	0,097378277
497	1,136	0,957	-0,178	0,179	0,157570423
498	0,965	1,064	0,099	0,099	0,102590674
499	0,943	1,066	0,122	0,123	0,130434783
500	0,968	0,976	0,009	0,008	0,008264463
501	1,019	0,98	-0,039	0,039	0,038272816
502	0,937	1,005	0,068	0,068	0,072572038
503	0,97	0,974	0,004	0,004	0,004123711
504	0,914	0,955	0,042	0,041	0,044857768
505	0,916	0,96	0,045	0,044	0,048034934
506	0,94	0,961	0,021	0,021	0,022340426
507	0,999	1,046	0,047	0,047	0,047047047
508	0,944	0,971	0,027	0,027	0,028601695
509	0,982	0,925	-0,057	0,057	0,058044807
510	0,895	1,006	0,111	0,111	0,124022346
511	0,709	0,955	0,246	0,246	0,34696756
512	1,008	0,914	-0,094	0,094	0,093253968
513	0,766	1,045	0,278	0,279	0,364229765
514	0,803	0,972	0,169	0,169	0,210460772
515	1,377	1,25	-0,127	0,127	0,092229484
516	0,992	1,046	0,053	0,054	0,054435484
517	1,052	1,038	-0,014	0,014	0,013307985
518	0,762	0,935	0,173	0,173	0,227034121
519	1,177	0,925	-0,252	0,252	0,214103653
520	0,707	0,957	0,251	0,25	0,353606789
521	0,821	0,984	0,163	0,163	0,198538368
522	1,376	1,292	-0,084	0,084	0,061046512
523	1,141	1,118	-0,022	0,023	0,020157756
524	1,045	0,897	-0,148	0,148	0,141626794
525	1,029	0,95	-0,079	0,079	0,076773567
526	0,889	1,029	0,14	0,14	0,157480315
527	1,023	0,972	-0,051	0,051	0,049853372
528	1,16	1,03	-0,13	0,13	0,112068966
529	1,209	1,114	-0,094	0,095	0,078577337

530	1,127	0,998	-0,129	0,129	0,114463177
531	0,983	0,991	0,008	0,008	0,008138352
532	1,059	0,968	-0,091	0,091	0,085930123
533	1,075	0,959	-0,116	0,116	0,107906977
534	0,993	1,083	0,09	0,09	0,090634441
535	0,871	0,963	0,092	0,092	0,105625718
536	1,154	1,036	-0,118	0,118	0,102253033
537	1,016	1,033	0,017	0,017	0,016732283
538	0,938	1,011	0,072	0,073	0,07782516
539	0,863	0,933	0,07	0,07	0,081112399
540	1,073	0,89	-0,183	0,183	0,17054986
541	0,874	1,012	0,138	0,138	0,157894737
542	1,046	1,074	0,029	0,028	0,026768642
543	0,982	1,01	0,029	0,028	0,028513238
544	0,896	1,01	0,115	0,114	0,127232143
545	1,001	0,967	-0,035	0,034	0,033966034
546	0,965	0,935	-0,03	0,03	0,031088083
547	1,059	1,019	-0,04	0,04	0,037771483
548	1,028	0,992	-0,036	0,036	0,035019455
549	1,07	1,086	0,016	0,016	0,014953271
550	1,098	1,066	-0,032	0,032	0,029143898
551	1,089	1,005	-0,085	0,084	0,077134986
552	1,107	1,03	-0,077	0,077	0,069557362
553	1,181	1,044	-0,137	0,137	0,116003387
554	1,129	1,061	-0,067	0,068	0,060230292
555	1,092	0,989	-0,103	0,103	0,094322344
556	1,059	1,045	-0,013	0,014	0,013220019
557	1,133	1,085	-0,049	0,048	0,042365402
558	1,118	1,034	-0,084	0,084	0,075134168
559	1,009	1,02	0,011	0,011	0,010901883
560	0,993	1,009	0,016	0,016	0,01611279
561	0,947	0,993	0,046	0,046	0,048574446
562	0,994	0,952	-0,042	0,042	0,042253521
563	0,927	0,978	0,051	0,051	0,055016181
564	0,994	1,054	0,06	0,06	0,060362173
565	1,089	1,058	-0,031	0,031	0,028466483
566	0,996	0,979	-0,017	0,017	0,017068273
567	0,967	0,99	0,023	0,023	0,023784902
568	0,968	0,971	0,004	0,003	0,003099174
569	1,019	0,954	-0,066	0,065	0,063788027
570	0,925	1,008	0,083	0,083	0,08972973
571	1,094	1,074	-0,02	0,02	0,018281536
572	1,016	1,003	-0,012	0,013	0,012795276
573	1,024	1,011	-0,013	0,013	0,012695313
574	1,029	0,988	-0,04	0,041	0,039844509
575	0,999	1,008	0,009	0,009	0,009009009
576	0,997	0,995	-0,003	0,002	0,002006018
577	0,956	0,995	0,039	0,039	0,040794979
578	1,041	1,031	-0,011	0,01	0,009606148
579	0,992	1,011	0,019	0,019	0,019153226
580	1,003	0,994	-0,009	0,009	0,008973081
581	1,042	0,978	-0,064	0,064	0,061420345
582	1,007	1,028	0,021	0,021	0,020854022
583	1,01	0,987	-0,024	0,023	0,022772277
584	0,97	0,998	0,028	0,028	0,028865979
585	1,04	1,036	-0,004	0,004	0,003846154
586	1,028	1,006	-0,022	0,022	0,021400778
587	0,99	0,989	-0,001	0,001	0,001010101
588	1,011	0,994	-0,017	0,017	0,016815035
589	1,008	1,003	-0,005	0,005	0,004960317
590	1,001	0,997	-0,005	0,004	0,003996004
591	0,781	1	0,22	0,219	0,280409731
592	1,137	1,126	-0,01	0,011	0,009674582
593	1,195	1,239	0,044	0,044	0,036820084

594	1	1,116	0,117	0,116	0,116
595	1,01	0,971	-0,039	0,039	0,038613861
596	1,011	0,981	-0,03	0,03	0,029673591
597	1,035	1,006	-0,03	0,029	0,028019324
598	0,915	1,003	0,087	0,088	0,096174863
599	1,126	1,072	-0,054	0,054	0,047957371
600	0,975	1,02	0,045	0,045	0,046153846
601	0,987	0,987	0	0	0
602	1,022	0,99	-0,032	0,032	0,031311155
603	0,988	1,013	0,024	0,025	0,025303644
604	0,954	0,981	0,027	0,027	0,028301887
605	0,924	0,991	0,068	0,067	0,072510823
606	1,15	1,062	-0,088	0,088	0,076521739
607	0,915	0,999	0,084	0,084	0,091803279
608	1,093	1,044	-0,05	0,049	0,044830741
609	1,02	0,947	-0,073	0,073	0,071568627
610	0,978	0,982	0,004	0,004	0,00408998
611	1,005	1,019	0,014	0,014	0,013930348
612	0,932	0,987	0,055	0,055	0,059012876
613	1,086	1,037	-0,049	0,049	0,045119705
614	1,044	1,015	-0,029	0,029	0,027777778
615	0,953	0,988	0,035	0,035	0,036726128
616	0,992	1,018	0,026	0,026	0,026209677
617	1,039	0,995	-0,043	0,044	0,042348412
618	0,952	0,963	0,011	0,011	0,011554622
619	0,896	0,981	0,085	0,085	0,094866071
620	1,174	1,095	-0,078	0,079	0,067291312
621	1,053	0,968	-0,085	0,085	0,080721747
622	1,027	0,966	-0,061	0,061	0,0593963
623	1,049	1,01	-0,04	0,039	0,037178265
624	1,047	1,006	-0,04	0,041	0,039159503
625	1,06	1,026	-0,035	0,034	0,032075472
626	0,961	1,016	0,056	0,055	0,05723205
627	1,084	1,058	-0,026	0,026	0,02398524
628	1,012	1,011	-0,001	0,001	0,000988142
629	1,016	0,977	-0,039	0,039	0,038385827
630	1,013	0,984	-0,029	0,029	0,028627838
631	1,03	1,021	-0,01	0,009	0,008737864
632	0,984	0,985	0,001	0,001	0,00101626
633	0,977	0,985	0,008	0,008	0,008188332
634	1,035	1,031	-0,004	0,004	0,003864734
635	1,012	1,007	-0,005	0,005	0,004940711
636	0,989	0,981	-0,007	0,008	0,008088979
637	1,021	1,003	-0,018	0,018	0,017629775
638	0,984	0,992	0,009	0,008	0,008130081
639	0,985	0,987	0,002	0,002	0,002030457
640	0,964	0,991	0,027	0,027	0,028008299
641	1,048	1,039	-0,009	0,009	0,008587786
642	1,023	1,008	-0,015	0,015	0,014662757
643	0,98	0,985	0,005	0,005	0,005102041
644	0,997	0,982	-0,015	0,015	0,015045135
645	0,982	0,985	0,003	0,003	0,00305499
646	1	0,988	-0,011	0,012	0,012
647	0,998	1,009	0,011	0,011	0,011022044
648	1,029	1,011	-0,018	0,018	0,017492711
649	1,003	1,01	0,008	0,007	0,006979063
650	0,975	0,993	0,018	0,018	0,018461538
651	1,007	0,98	-0,027	0,027	0,026812314
652	1,01	1,003	-0,008	0,007	0,006930693
653	1,015	1	-0,015	0,015	0,014778325
654	0,998	0,995	-0,003	0,003	0,003006012
655	1,026	1,015	-0,011	0,011	0,010721248
656	1,012	1,012	0	0	0
657	1	1	0	0	0

658	0,993	0,99	-0,002	0,003	0,003021148
659	0,992	0,996	0,003	0,004	0,004032258
660	0,99	0,992	0,002	0,002	0,002020202
661	0,977	0,982	0,005	0,005	0,005117707
662	1,021	1,002	-0,019	0,019	0,018609207
663	0,997	1,021	0,024	0,024	0,024072217
664	0,971	0,994	0,023	0,023	0,023686921
665	1,019	0,975	-0,044	0,044	0,043179588
666	1,005	0,994	-0,011	0,011	0,010945274
667	0,988	1,004	0,016	0,016	0,016194332
668	0,977	0,976	-0,002	0,001	0,001023541
669	1,028	1,037	0,009	0,009	0,008754864
670	0,989	1,014	0,025	0,025	0,025278059
671	0,974	0,977	0,003	0,003	0,003080082
672	0,969	0,971	0,001	0,002	0,002063983
673	0,986	0,98	-0,006	0,006	0,006085193
674	0,973	0,976	0,003	0,003	0,003083248
675	0,967	0,981	0,013	0,014	0,014477766
676	1,009	0,997	-0,012	0,012	0,011892963
677	0,976	1	0,023	0,024	0,024590164
678	0,995	0,983	-0,012	0,012	0,012060302
679	0,983	0,984	0	0,001	0,001017294
680	0,97	1,008	0,038	0,038	0,039175258
681	0,977	0,969	-0,009	0,008	0,008188332
682	0,979	0,978	-0,001	0,001	0,00102145
683	0,958	1,012	0,053	0,054	0,056367432
684	0,969	1,001	0,032	0,032	0,033023736
685	1,014	0,964	-0,05	0,05	0,049309665
686	0,991	0,989	-0,003	0,002	0,002018163
687	0,989	0,991	0,002	0,002	0,002022245
688	1,039	0,979	-0,06	0,06	0,057747834
689	0,847	1,01	0,163	0,163	0,19244392
690	1,168	1,105	-0,063	0,063	0,053938356
691	1,006	0,977	-0,029	0,029	0,028827038
692	1,024	0,968	-0,056	0,056	0,0546875
693	1,034	0,982	-0,053	0,052	0,050290135
694	1,03	1,028	-0,002	0,002	0,001941748
695	0,999	1,003	0,004	0,004	0,004004004
696	0,981	0,994	0,013	0,013	0,013251784
697	1,02	1,006	-0,014	0,014	0,01372549
698	1,019	1,014	-0,005	0,005	0,004906771
699	0,987	1,007	0,02	0,02	0,020263425
700	1,001	0,973	-0,028	0,028	0,027972028
701	1,023	0,981	-0,042	0,042	0,041055718
702	0,983	1,009	0,026	0,026	0,026449644
703	0,942	0,992	0,05	0,05	0,053078556
704	1,035	1,004	-0,03	0,031	0,029951691
705	1,018	1,022	0,004	0,004	0,003929273
706	1,017	1,003	-0,015	0,014	0,013765978
707	1,009	1,005	-0,004	0,004	0,003964321
708	0,965	0,982	0,017	0,017	0,01761658
709	0,94	0,959	0,019	0,019	0,020212766
710	1,002	0,969	-0,034	0,033	0,032934132
711	1,067	1,047	-0,02	0,02	0,018744142
712	0,97	1,02	0,05	0,05	0,051546392
713	0,978	0,97	-0,008	0,008	0,008179959
714	0,979	0,962	-0,017	0,017	0,017364658
715	0,981	0,997	0,016	0,016	0,016309888
716	0,964	0,983	0,019	0,019	0,019709544
717	1,002	0,983	-0,02	0,019	0,018962076
718	1,055	1,03	-0,025	0,025	0,023696682
719	0,976	1,034	0,057	0,058	0,05942623
720	0,991	0,992	0,001	0,001	0,001009082
721	0,949	0,94	-0,009	0,009	0,009483667

722	1,016	0,991	-0,025	0,025	0,024606299
723	1,016	1,002	-0,014	0,014	0,013779528
724	1,006	1,002	-0,004	0,004	0,003976143
725	1,1	1,014	-0,086	0,086	0,078181818
726	0,938	0,997	0,058	0,059	0,062899787
727	0,992	0,987	-0,005	0,005	0,005040323
728	0,992	0,972	-0,02	0,02	0,02016129
729	0,944	0,985	0,041	0,041	0,043432203
730	0,981	1,006	0,025	0,025	0,0254842
731	0,99	0,985	-0,005	0,005	0,005050505
732	1,027	1,022	-0,005	0,005	0,004868549
733	1,023	1,01	-0,013	0,013	0,012707722
734	1,047	1,013	-0,034	0,034	0,032473734
735	0,971	0,972	0,001	0,001	0,001029866
736	1,006	0,978	-0,028	0,028	0,027833002
737	0,969	1,013	0,045	0,044	0,045407637
738	0,959	0,972	0,013	0,013	0,013555787
739	1,097	1,045	-0,052	0,052	0,047402005
740	0,941	1,013	0,072	0,072	0,076514346
741	0,97	0,98	0,01	0,01	0,010309278
742	0,957	0,998	0,041	0,041	0,042842215
743	0,949	0,972	0,023	0,023	0,024236038
744	0,96	0,958	-0,001	0,002	0,002083333
745	1,001	0,965	-0,036	0,036	0,035964036
746	1,044	1,011	-0,033	0,033	0,031609195
747	1,016	1,035	0,02	0,019	0,018700787
748	0,978	1,008	0,03	0,03	0,030674847
749	0,986	0,958	-0,028	0,028	0,028397566
750	1,026	0,987	-0,039	0,039	0,038011696
751	1,036	1,035	-0,001	0,001	0,000965251
752	1,002	1,038	0,036	0,036	0,035928144
753	1,047	1,015	-0,032	0,032	0,030563515
754	1,003	0,996	-0,007	0,007	0,006979063
755	0,975	0,998	0,023	0,023	0,023589744
756	0,988	0,982	-0,006	0,006	0,006072874
757	1,075	1,012	-0,063	0,063	0,058604651
758	0,944	1	0,056	0,056	0,059322034
759	1,015	0,984	-0,031	0,031	0,030541872
760	1,069	1,024	-0,045	0,045	0,042095416
761	0,976	1,034	0,058	0,058	0,05942623
762	1,03	0,91	-0,121	0,12	0,116504854
763	0,96	1,016	0,057	0,056	0,058333333
764	0,956	0,993	0,037	0,037	0,038702929
765	0,986	0,963	-0,023	0,023	0,023326572
766	0,934	0,994	0,06	0,06	0,064239829
767	1,038	0,988	-0,051	0,05	0,048169557
768	1,009	1,003	-0,006	0,006	0,005946482
769	1,021	1,018	-0,003	0,003	0,002938296
770	1,059	0,988	-0,071	0,071	0,067044381
771	0,944	0,97	0,026	0,026	0,027542373
772	0,977	0,966	-0,011	0,011	0,011258956
773	0,951	1,009	0,058	0,058	0,060988433
774	1,056	1,032	-0,024	0,024	0,022727273
775	0,985	0,997	0,011	0,012	0,012182741
776	0,974	0,979	0,005	0,005	0,00513347
777	0,975	0,995	0,02	0,02	0,020512821
778	1	0,953	-0,046	0,047	0,047
779	0,977	1,002	0,026	0,025	0,025588536
780	1,013	1,026	0,013	0,013	0,012833169
781	1,058	1,027	-0,031	0,031	0,029300567
782	1,034	1,003	-0,031	0,031	0,029980658
783	1	0,986	-0,014	0,014	0,014
784	1,016	1,002	-0,014	0,014	0,013779528
785	1,002	0,999	-0,002	0,003	0,002994012

786	0,981	0,989	0,008	0,008	0,008154944
787	0,971	1,005	0,034	0,034	0,035015448
788	1,06	1,013	-0,047	0,047	0,044339623
789	1,009	1,001	-0,008	0,008	0,007928642
790	1,009	0,993	-0,016	0,016	0,015857284
791	0,967	0,99	0,023	0,023	0,023784902
792	0,99	0,997	0,007	0,007	0,007070707
793	0,98	0,996	0,016	0,016	0,016326531
794	0,996	0,995	-0,001	0,001	0,001004016
795	1,054	1,02	-0,034	0,034	0,032258065
796	0,988	1,013	0,025	0,025	0,025303644
797	0,998	0,978	-0,02	0,02	0,02004008
798	0,986	0,995	0,009	0,009	0,009127789
799	0,976	0,991	0,015	0,015	0,015368852
800	0,972	0,975	0,004	0,003	0,00308642
801	0,987	0,99	0,003	0,003	0,003039514
802	1,02	1,028	0,008	0,008	0,007843137
803	0,996	1,016	0,02	0,02	0,020080321
804	0,974	0,977	0,003	0,003	0,003080082
805	0,951	0,972	0,022	0,021	0,022082019
806	0,996	0,986	-0,01	0,01	0,010040161
807	0,979	0,965	-0,013	0,014	0,014300306
808	0,963	0,989	0,026	0,026	0,026998962
809	1,035	1,015	-0,02	0,02	0,019323671
810	0,989	1,026	0,037	0,037	0,037411527
811	0,98	0,982	0,002	0,002	0,002040816
812	0,961	0,957	-0,004	0,004	0,004162331
813	1,014	0,995	-0,019	0,019	0,018737673
814	0,999	1,006	0,007	0,007	0,007007007
815	0,973	1,001	0,028	0,028	0,028776978
816	1,044	0,99	-0,054	0,054	0,051724138
817	1,005	1,02	0,015	0,015	0,014925373
818	0,97	0,981	0,011	0,011	0,011340206
819	1,014	0,981	-0,034	0,033	0,032544379
820	1,016	1,002	-0,014	0,014	0,013779528
821	0,993	1,009	0,016	0,016	0,01611279
822	0,982	1,005	0,023	0,023	0,023421589
823	1,062	1,022	-0,039	0,04	0,037664783
824	1,035	1,007	-0,028	0,028	0,02705314
825	1,018	0,989	-0,029	0,029	0,02848723
826	1,012	0,999	-0,013	0,013	0,01284585
827	0,941	0,997	0,056	0,056	0,059511158
828	0,967	0,965	-0,002	0,002	0,002068252
829	1,002	0,989	-0,013	0,013	0,012974052
830	1,045	1,052	0,007	0,007	0,006698565
831	0,964	0,994	0,029	0,03	0,031120332
832	1,016	0,978	-0,038	0,038	0,037401575
833	1,021	1,012	-0,009	0,009	0,008814887
834	0,998	0,998	0	0	0
835	0,959	0,971	0,012	0,012	0,012513034
836	0,962	1,001	0,039	0,039	0,040540541
837	1,068	0,993	-0,075	0,075	0,070224719
838	1,05	1,018	-0,032	0,032	0,03047619
839	1,012	1,022	0,01	0,01	0,009881423
840	1,004	0,998	-0,006	0,006	0,005976096
841	0,987	0,954	-0,033	0,033	0,03343465
842	0,973	0,977	0,004	0,004	0,004110997
843	0,994	1,036	0,042	0,042	0,042253521
844	1,096	1,05	-0,046	0,046	0,041970803
845	0,95	0,913	-0,037	0,037	0,038947368
846	1,011	0,995	-0,016	0,016	0,015825915
847	0,978	1,061	0,083	0,083	0,084867076
848	1,011	0,97	-0,041	0,041	0,040553907
849	0,974	0,992	0,019	0,018	0,018480493

850	0,979	1,001	0,022	0,022	0,02247191
851	1,049	1,023	-0,026	0,026	0,02478551
852	0,981	1,007	0,027	0,026	0,026503568
853	0,98	0,983	0,003	0,003	0,003061224
854	0,986	0,991	0,006	0,005	0,005070994
855	0,896	0,998	0,101	0,102	0,113839286
856	0,993	0,947	-0,046	0,046	0,04632427
857	0,904	1,011	0,107	0,107	0,118362832
858	1,062	1,015	-0,047	0,047	0,044256121
859	1,14	1,053	-0,087	0,087	0,076315789
860	0,989	0,983	-0,006	0,006	0,006066734
861	0,927	1,024	0,097	0,097	0,104638619
862	1,041	1	-0,041	0,041	0,039385207
863	1,006	0,983	-0,023	0,023	0,022862823
864	0,984	0,985	0,001	0,001	0,00101626
865	1,055	0,999	-0,057	0,056	0,053080569
866	1,021	1,012	-0,009	0,009	0,008814887
867	1,007	0,984	-0,023	0,023	0,022840119
868	1,003	0,97	-0,032	0,033	0,032901296
869	0,989	1,013	0,023	0,024	0,024266936
870	0,852	0,987	0,135	0,135	0,158450704
871	0,982	0,99	0,008	0,008	0,00814664
872	1,147	1,084	-0,062	0,063	0,054925894
873	0,895	1,012	0,117	0,117	0,130726257
874	1,08	1,042	-0,039	0,038	0,035185185
875	0,974	0,972	-0,002	0,002	0,002053388
876	0,85	0,997	0,147	0,147	0,172941176
877	0,985	1,07	0,085	0,085	0,086294416
878	0,868	1,047	0,179	0,179	0,206221198
879	1,375	1,193	-0,182	0,182	0,132363636
880	0,955	1,029	0,074	0,074	0,077486911
881	1,076	0,916	-0,159	0,16	0,148698885
882	0,979	0,978	-0,001	0,001	0,00102145
883	1,029	1,023	-0,006	0,006	0,005830904
884	0,945	0,985	0,041	0,04	0,042328042
885	0,972	0,988	0,016	0,016	0,016460905
886	1,129	1,033	-0,096	0,096	0,085031001
887	1,009	1,051	0,042	0,042	0,041625372
888	1,042	1,007	-0,035	0,035	0,033589251
889	1,028	1,008	-0,02	0,02	0,019455253
890	0,964	0,994	0,03	0,03	0,031120332
891	0,976	0,93	-0,046	0,046	0,047131148
892	0,975	1,065	0,09	0,09	0,092307692
893	1,105	1,022	-0,083	0,083	0,075113122
894	0,978	0,994	0,016	0,016	0,016359918
895	0,99	1,018	0,028	0,028	0,028282828
896	0,93	1,004	0,073	0,074	0,079569892
897	1,045	0,97	-0,075	0,075	0,071770335
898	0,911	1,002	0,091	0,091	0,099890231
899	1,05	1,033	-0,017	0,017	0,016190476
900	1,072	0,99	-0,082	0,082	0,076492537
901	0,994	1,027	0,033	0,033	0,033199195
902	0,934	0,966	0,032	0,032	0,034261242
903	1,015	0,982	-0,033	0,033	0,032512315
904	1,046	1,021	-0,025	0,025	0,023900574
905	0,872	1,012	0,14	0,14	0,160550459
906	0,771	0,944	0,174	0,173	0,224383917
907	1,293	1,185	-0,108	0,108	0,083526682
908	1,115	1,161	0,046	0,046	0,041255605
909	1,011	1,062	0,051	0,051	0,050445104
910	1,075	1,048	-0,027	0,027	0,025116279
911	1,014	0,976	-0,039	0,038	0,037475345
912	0,955	1,009	0,053	0,054	0,056544503
913	0,997	0,96	-0,036	0,037	0,037111334

914	1,064	1,01	-0,054	0,054	0,05075188
915	1,001	1,031	0,03	0,03	0,02997003
916	1,007	1,012	0,005	0,005	0,004965243
917	0,97	0,977	0,007	0,007	0,007216495
918	0,97	0,963	-0,007	0,007	0,007216495
919	0,915	0,935	0,02	0,02	0,021857923
920	1,008	1,026	0,019	0,018	0,017857143
921	1,115	1,086	-0,029	0,029	0,026008969
922	0,999	1,027	0,029	0,028	0,028028028
923	0,973	0,976	0,002	0,003	0,003083248
924	0,959	1,012	0,053	0,053	0,055265902
925	1,029	0,943	-0,086	0,086	0,083576288
926	0,948	1,011	0,064	0,063	0,066455696
927	0,921	0,986	0,065	0,065	0,070575461
928	1,203	1,098	-0,105	0,105	0,087281796
929	0,966	1,022	0,056	0,056	0,057971014
930	1,013	0,997	-0,015	0,016	0,015794669
931	0,983	0,964	-0,018	0,019	0,019328586
932	1,003	1,022	0,019	0,019	0,01894317
933	0,896	1,001	0,105	0,105	0,1171875
934	1,058	1,018	-0,04	0,04	0,037807183
935	1,114	1,043	-0,071	0,071	0,063734291
936	1,033	1,005	-0,027	0,028	0,027105518
937	1,002	1,01	0,008	0,008	0,007984032
938	0,989	1,011	0,023	0,022	0,022244692
939	0,975	0,996	0,021	0,021	0,021538462
940	0,988	0,966	-0,021	0,022	0,022267206
941	0,962	0,989	0,027	0,027	0,028066528
942	1,045	0,995	-0,05	0,05	0,04784689
943	0,972	1,012	0,04	0,04	0,041152263
944	0,993	0,985	-0,008	0,008	0,008056395
945	0,97	0,974	0,004	0,004	0,004123711
946	0,983	0,996	0,012	0,013	0,013224822
947	0,95	0,999	0,049	0,049	0,051578947
948	0,997	1,02	0,023	0,023	0,023069208
949	1,04	1,013	-0,027	0,027	0,025961538
950	1,012	1,015	0,003	0,003	0,002964427
951	1,003	0,96	-0,043	0,043	0,042871386
952	0,999	0,988	-0,011	0,011	0,011011011
953	1,008	1,029	0,021	0,021	0,020833333
954	0,938	0,958	0,02	0,02	0,021321962
955	0,839	0,991	0,152	0,152	0,181168057
956	1,195	1,166	-0,03	0,029	0,024267782
957	0,924	1,179	0,255	0,255	0,275974026
958	1,11	1,032	-0,078	0,078	0,07027027
959	0,925	0,821	-0,104	0,104	0,112432432
960	1,017	1,142	0,125	0,125	0,122910521
961	0,973	0,923	-0,05	0,05	0,051387461
962	0,608	0,995	0,387	0,387	0,636513158
963	1,4	1,353	-0,048	0,047	0,033571429
964	1,098	1,12	0,022	0,022	0,02003643
965	1,05	1,099	0,048	0,049	0,046666667
966	0,84	0,999	0,159	0,159	0,189285714
967	1,062	1,004	-0,058	0,058	0,054613936
968	1,124	0,92	-0,204	0,204	0,181494662
969	0,926	1,037	0,111	0,111	0,11987041
970	1,122	1,036	-0,086	0,086	0,076648841
971	0,987	1,066	0,079	0,079	0,080040527
972	0,997	0,977	-0,02	0,02	0,020060181
973	1,003	0,986	-0,017	0,017	0,016949153
974	0,991	0,993	0,002	0,002	0,002018163
975	0,966	0,996	0,031	0,03	0,031055901
976	1,018	1,005	-0,013	0,013	0,012770138
977	1,055	1,009	-0,046	0,046	0,043601896

978	1,006	1,002	-0,005	0,004	0,003976143
979	1,007	1,017	0,01	0,01	0,009930487
980	0,994	0,999	0,006	0,005	0,005030181
981	1,016	0,999	-0,016	0,017	0,016732283
982	0,994	1,011	0,017	0,017	0,017102616
983	0,946	0,969	0,023	0,023	0,024312896
984	1,062	1,043	-0,019	0,019	0,017890772
985	0,937	0,998	0,061	0,061	0,065101387
986	0,977	0,98	0,003	0,003	0,003070624
987	0,979	0,952	-0,028	0,027	0,027579162
988	1,017	0,991	-0,025	0,026	0,025565388
989	1,005	0,984	-0,021	0,021	0,020895522
990	0,95	1,013	0,063	0,063	0,066315789
991	1,036	1,049	0,013	0,013	0,012548263
992	1,004	0,995	-0,01	0,009	0,008964143
993	1,011	0,981	-0,03	0,03	0,029673591
994	1,002	0,982	-0,02	0,02	0,01996008
995	0,959	0,992	0,034	0,033	0,034410845
996	0,916	0,979	0,063	0,063	0,068777293
997	0,974	0,986	0,012	0,012	0,012320329
998	1,018	1,042	0,024	0,024	0,023575639
999	1,024	1,056	0,032	0,032	0,03125
1000	0,903	0,921	0,018	0,018	0,019933555
1001	0,987	1,01	0,023	0,023	0,023302938
1002	0,861	0,926	0,065	0,065	0,075493612
1003	1,036	1,046	0,01	0,01	0,00965251
1004	0,902	1,155	0,253	0,253	0,280487805
1005	1,178	1,214	0,036	0,036	0,030560272
1006	1,063	0,964	-0,1	0,099	0,093132643
1007	0,944	0,991	0,048	0,047	0,049788136
1008	1,053	0,987	-0,066	0,066	0,062678063
1009	0,972	0,979	0,007	0,007	0,007201646
1010	1,029	0,976	-0,053	0,053	0,051506317
1011	0,931	1,024	0,094	0,093	0,099892589
1012	1,05	1,09	0,04	0,04	0,038095238
1013	1,008	0,995	-0,014	0,013	0,012896825
1014	1,014	0,974	-0,041	0,04	0,039447732
1015	0,976	1,004	0,028	0,028	0,028688525
1016	1,028	1,008	-0,019	0,02	0,019455253
1017	1,001	0,972	-0,03	0,029	0,028971029
1018	0,994	1,012	0,018	0,018	0,018108652
1019	1,027	1,009	-0,017	0,018	0,017526777
1020	0,991	0,989	-0,002	0,002	0,002018163
1021	0,978	0,998	0,02	0,02	0,020449898
1022	0,985	0,993	0,008	0,008	0,008121827
1023	0,962	0,984	0,022	0,022	0,022869023
1024	1,011	0,981	-0,031	0,03	0,029673591
1025	0,949	1,007	0,058	0,058	0,061116965
1026	1,087	1,092	0,005	0,005	0,004599816
1027	0,981	0,857	-0,124	0,124	0,126401631
1028	1,063	1,078	0,015	0,015	0,014111007
1029	1,086	0,956	-0,13	0,13	0,119705341
1030	0,997	1,016	0,019	0,019	0,019057172
1031	0,926	0,996	0,07	0,07	0,075593952
1032	0,873	1,026	0,153	0,153	0,175257732
1033	1,187	1,243	0,056	0,056	0,047177759
1034	1,059	0,976	-0,083	0,083	0,078375826
1035	0,971	1,017	0,045	0,046	0,047373841
1036	1,093	1,01	-0,084	0,083	0,075937786
1037	1,037	1,04	0,003	0,003	0,00289296
1038	1,013	1,045	0,032	0,032	0,031589339
1039	0,995	1,002	0,007	0,007	0,007035176
1040	1,016	1,022	0,007	0,006	0,005905512
1041	1,059	1,045	-0,014	0,014	0,013220019

1042	1,012	1,035	0,022	0,023	0,022727273
1043	1,042	0,995	-0,047	0,047	0,045105566
1044	1,03	1,006	-0,024	0,024	0,023300971
1045	0,974	0,967	-0,007	0,007	0,007186858
1046	1,02	1,035	0,015	0,015	0,014705882
1047	1,041	1,029	-0,012	0,012	0,011527378
1048	1,014	1,027	0,013	0,013	0,012820513
1049	1,001	1	-0,001	0,001	0,000999001
1050	1,017	0,974	-0,043	0,043	0,042281219
1051	0,998	1,008	0,01	0,01	0,01002004
1052	0,982	1,002	0,021	0,02	0,020366599
1053	0,963	0,975	0,011	0,012	0,012461059
1054	1,064	1,03	-0,034	0,034	0,031954887
1055	1,015	1,016	0,001	0,001	0,000985222
1056	1,013	1,012	-0,002	0,001	0,000987167
1057	0,988	1,011	0,022	0,023	0,023279352
1058	1,01	1,008	-0,001	0,002	0,001980198
1059	0,973	0,98	0,007	0,007	0,007194245
1060	0,968	0,992	0,024	0,024	0,024793388
1061	1,052	0,968	-0,084	0,084	0,079847909
1062	1,003	1,021	0,019	0,018	0,017946162
1063	0,997	0,993	-0,004	0,004	0,004012036
1064	0,997	1,014	0,017	0,017	0,017051153
1065	1,012	0,995	-0,017	0,017	0,016798419
1066	0,971	0,978	0,007	0,007	0,007209063
1067	1,009	1,008	-0,001	0,001	0,00099108
1068	1,023	1,046	0,023	0,023	0,022482893
1069	1,023	0,999	-0,024	0,024	0,023460411
1070	1,008	0,997	-0,011	0,011	0,010912698
1071	1,061	0,995	-0,066	0,066	0,062205467
1072	0,904	1	0,096	0,096	0,10619469
1073	0,953	0,972	0,019	0,019	0,019937041
1074	0,989	0,962	-0,027	0,027	0,027300303
1075	1,04	1,068	0,028	0,028	0,026923077
1076	0,983	0,986	0,003	0,003	0,003051882
1077	0,997	1,002	0,005	0,005	0,005015045
1078	0,977	0,98	0,003	0,003	0,003070624
1079	0,978	0,972	-0,006	0,006	0,006134969
1080	0,968	0,969	0	0,001	0,001033058
1081	0,956	1,022	0,066	0,066	0,069037657
1082	1,056	1,016	-0,04	0,04	0,037878788
1083	1,041	0,984	-0,057	0,057	0,054755043
1084	1	1,014	0,014	0,014	0,014
1085	0,94	0,997	0,057	0,057	0,060638298
1086	1,007	0,991	-0,016	0,016	0,015888779
1087	0,984	0,982	-0,002	0,002	0,00203252
1088	1,003	0,974	-0,03	0,029	0,02891326
1089	1,053	1,056	0,004	0,003	0,002849003
1090	1,033	1,019	-0,014	0,014	0,013552759
1091	1	1,011	0,011	0,011	0,011
1092	1,022	1,032	0,01	0,01	0,009784736
1093	0,961	0,963	0,002	0,002	0,002081165
1094	0,973	0,957	-0,016	0,016	0,016443988
			4,18	4,23	

B.2 Experiment 3

In our third experiment, we got our best result using support vector machine with an RBF kernel on the all attribute data set. This was also our overall best result. Compared to our experiment 2

results, we can see that we have fewer test days in this experiment.

Listing B.2: Experiment 3 SVM with RBF kernel all attributes

```

=== Run information ===

Scheme:      weka.classifiers.functions.SMOreg -C 1.0 -N 0 -I
"weka.classifiers.functions.supportVector.RegSMOImproved -L 0.0010 -W 1 -P 1.0E-12 -T 0.0010 -V" -K
"weka.classifiers.functions.supportVector.RBFKernel -C 250007 -G 0.04"
Relation:    exp3_all
Instances:   1433
Attributes:  112
              [list of attributes omitted]
Test mode:   split 66.0% train , remainder test

=== Classifier model (full training set) ===

SMOreg

Support vectors:
-1.0 * k[0]
+1.0 * k[1]
+1.0 * k[2]
-1.0 * k[3]
-1.0 * k[4]
-1.0 * k[5]
-1.0 * k[6]
-1.0 * k[7]
+1.0 * k[8]
-1.0 * k[9]
+1.0 * k[10]
-0.23301393348593932 * k[11]
+1.0 * k[12]
+1.0 * k[13]
+1.0 * k[14]
+1.0 * k[15]
+1.0 * k[16]
+1.0 * k[17]
+1.0 * k[18]
-1.0 * k[19]
-0.8866279309229639 * k[20]
-1.0 * k[21]
+1.0 * k[22]
-1.0 * k[23]
-1.0 * k[24]
-1.0 * k[25]
-1.0 * k[26]
-1.0 * k[27]
+0.5745440738150808 * k[28]
+0.6157724296843361 * k[29]
+1.0 * k[31]
+1.0 * k[32]
+1.0 * k[33]
-0.7086759157338716 * k[34]
+1.0 * k[35]
-0.4036204651678564 * k[36]
+0.3294536039149465 * k[37]
+1.0 * k[38]
+1.0 * k[39]
+0.7954642646242399 * k[40]
-0.32329257958080193 * k[41]
+1.0 * k[43]
-1.0 * k[44]
+1.0 * k[46]

```



```
+1.0 * k[47]
+1.0 * k[48]
+1.0 * k[49]
+0.16597671369645062 * k[50]
+0.0019980718609157733 * k[51]
+0.26799544280317456 * k[52]
+1.0 * k[53]
+1.0 * k[54]
-1.0 * k[55]
-1.0 * k[56]
+1.0 * k[57]
-1.0 * k[58]
+1.0 * k[60]
+1.0 * k[61]
+0.12151348116235794 * k[62]
-1.0 * k[63]
-0.9380429482083702 * k[64]
+1.0 * k[65]
+1.0 * k[66]
-1.0 * k[67]
-1.0 * k[68]
+0.31619992494723903 * k[69]
+1.0 * k[70]
-1.0 * k[71]
+1.0 * k[72]
+1.0 * k[73]
+1.0 * k[74]
+0.43889194433394796 * k[75]
-1.0 * k[76]
+1.0 * k[77]
+1.0 * k[78]
-1.0 * k[80]
+0.5089280013866815 * k[81]
-1.0 * k[82]
-1.0 * k[83]
-1.0 * k[84]
-1.0 * k[85]
-1.0 * k[86]
-1.0 * k[87]
-1.0 * k[88]
+0.25603855601811754 * k[89]
-0.10777767816780176 * k[90]
+1.0 * k[91]
-0.7480961855945363 * k[92]
+1.0 * k[94]
-1.0 * k[96]
+0.7047269421150245 * k[97]
+1.0 * k[98]
-1.0 * k[99]
+0.6089092514262409 * k[100]
+1.0 * k[101]
-1.0 * k[102]
+1.0 * k[103]
-1.0 * k[104]
+1.0 * k[106]
+1.0 * k[107]
-0.22292572847298472 * k[108]
+1.0 * k[109]
-1.0 * k[110]
+1.0 * k[111]
+1.0 * k[112]
-1.0 * k[113]
+1.0 * k[114]
-0.17025554295052434 * k[115]
```

```
-1.0 * k[116]
-1.0 * k[117]
-1.0 * k[118]
-0.926735369460606 * k[119]
-1.0 * k[120]
+1.0 * k[121]
+1.0 * k[122]
-1.0 * k[123]
+1.0 * k[124]
-1.0 * k[125]
-1.0 * k[126]
+1.0 * k[127]
-0.4042504992539194 * k[128]
+0.6343283505115748 * k[129]
+1.0 * k[130]
-0.5851464554286491 * k[131]
+1.0 * k[132]
-1.0 * k[133]
-1.0 * k[134]
-1.0 * k[135]
-1.0 * k[136]
+1.0 * k[137]
-1.0 * k[138]
-1.0 * k[139]
+1.0 * k[140]
-1.0 * k[141]
+1.0 * k[142]
+1.0 * k[144]
+0.6456264912414611 * k[145]
-1.0 * k[146]
+1.0 * k[147]
+1.0 * k[148]
+1.0 * k[149]
-1.0 * k[150]
+1.0 * k[151]
-1.0 * k[152]
-1.0 * k[153]
+1.0 * k[154]
-1.0 * k[155]
+1.0 * k[156]
+1.0 * k[157]
-1.0 * k[158]
-1.0 * k[159]
-1.0 * k[160]
-1.0 * k[161]
+1.0 * k[162]
-1.0 * k[164]
-0.5412620768635121 * k[165]
+1.0 * k[166]
-1.0 * k[167]
-1.0 * k[168]
+1.0 * k[169]
+0.04413910497053986 * k[170]
-1.0 * k[171]
-1.0 * k[172]
-1.0 * k[173]
+1.0 * k[174]
+0.9722917345997554 * k[175]
-1.0 * k[176]
-1.0 * k[177]
+1.0 * k[178]
+1.0 * k[179]
+1.0 * k[180]
-1.0 * k[181]
```

```
+1.0 * k[183]
-1.0 * k[184]
+1.0 * k[186]
+0.9466473183230297 * k[187]
-1.0 * k[188]
+1.0 * k[190]
-1.0 * k[191]
+1.0 * k[192]
+1.0 * k[193]
-1.0 * k[194]
+1.0 * k[195]
+1.0 * k[196]
-1.0 * k[197]
-1.0 * k[198]
+1.0 * k[199]
+1.0 * k[200]
-0.2185851025182597 * k[201]
+1.0 * k[202]
-1.0 * k[203]
-1.0 * k[204]
-1.0 * k[205]
+0.4109907923583994 * k[206]
-1.0 * k[208]
-1.0 * k[209]
+1.0 * k[210]
+1.0 * k[211]
+1.0 * k[212]
-0.25554643827017337 * k[213]
+1.0 * k[214]
-1.0 * k[215]
-1.0 * k[216]
+1.0 * k[217]
+1.0 * k[219]
+1.0 * k[220]
-0.024075632492338673 * k[221]
-1.0 * k[222]
-1.0 * k[223]
+1.0 * k[225]
-0.9152626795266633 * k[226]
+1.0 * k[227]
+0.8226169024353998 * k[228]
+1.0 * k[229]
+1.0 * k[230]
+1.0 * k[231]
+1.0 * k[232]
+1.0 * k[233]
+1.0 * k[234]
+0.2423777818888194 * k[235]
+1.0 * k[236]
+1.0 * k[237]
+1.0 * k[238]
-1.0 * k[239]
-1.0 * k[240]
-1.0 * k[241]
-1.0 * k[242]
-1.0 * k[243]
-1.0 * k[244]
+1.0 * k[245]
-1.0 * k[246]
-1.0 * k[247]
+1.0 * k[248]
-1.0 * k[249]
-1.0 * k[250]
+1.0 * k[251]
```

```
+1.0 * k[252]
-1.0 * k[253]
+0.18031284295853253 * k[254]
+1.0 * k[255]
-0.01120432121491914 * k[256]
+0.4630482893958106 * k[257]
+1.0 * k[258]
+1.0 * k[259]
+1.0 * k[260]
-1.0 * k[261]
-0.7234522611972721 * k[262]
-1.0 * k[263]
-1.0 * k[264]
+1.0 * k[265]
+1.0 * k[266]
-1.0 * k[267]
-0.7141787346966356 * k[268]
+0.9641786658349776 * k[269]
-1.0 * k[270]
-1.0 * k[271]
-1.0 * k[272]
+1.0 * k[273]
-1.0 * k[274]
+0.13538347705813614 * k[275]
+1.0 * k[276]
-1.0 * k[278]
+1.0 * k[279]
+1.0 * k[280]
-1.0 * k[281]
+1.0 * k[282]
+1.0 * k[283]
+1.0 * k[285]
+1.0 * k[286]
+1.0 * k[287]
+1.0 * k[288]
-1.0 * k[289]
-1.0 * k[291]
-0.6360833940855648 * k[293]
+1.0 * k[294]
+1.0 * k[295]
-1.0 * k[296]
+0.21033013385221103 * k[297]
-1.0 * k[298]
-1.0 * k[299]
+0.7143678429757451 * k[300]
+1.0 * k[301]
+1.0 * k[302]
-1.0 * k[303]
-1.0 * k[304]
+1.0 * k[305]
-1.0 * k[307]
+1.0 * k[308]
-1.0 * k[309]
-1.0 * k[310]
+1.0 * k[311]
+1.0 * k[312]
+1.0 * k[313]
-1.0 * k[314]
+1.0 * k[315]
+1.0 * k[316]
-1.0 * k[317]
+1.0 * k[318]
-1.0 * k[319]
-1.0 * k[320]
```

```
-1.0 * k[321]
+1.0 * k[322]
-1.0 * k[323]
+1.0 * k[324]
-1.0 * k[325]
-1.0 * k[326]
-1.0 * k[327]
+0.18889029490807102 * k[328]
+1.0 * k[330]
-1.0 * k[331]
+1.0 * k[332]
-1.0 * k[334]
-1.0 * k[335]
-1.0 * k[336]
+1.0 * k[337]
+1.0 * k[338]
-1.0 * k[339]
+1.0 * k[340]
+1.0 * k[341]
-1.0 * k[342]
+1.0 * k[343]
+1.0 * k[344]
-1.0 * k[345]
+1.0 * k[346]
-1.0 * k[347]
+1.0 * k[348]
-1.0 * k[349]
-1.0 * k[350]
+1.0 * k[351]
-0.4698737733994485 * k[352]
+1.0 * k[353]
-1.0 * k[354]
-1.0 * k[355]
+1.0 * k[356]
-1.0 * k[357]
-1.0 * k[358]
-1.0 * k[359]
+1.0 * k[360]
-0.3607259401336072 * k[361]
-1.0 * k[362]
+1.0 * k[363]
+1.0 * k[364]
-1.0 * k[365]
+1.0 * k[366]
-0.530030554439346 * k[367]
-1.0 * k[368]
-1.0 * k[369]
-1.0 * k[370]
-1.0 * k[371]
+1.0 * k[372]
-1.0 * k[373]
+1.0 * k[374]
-1.0 * k[375]
+1.0 * k[376]
-0.22728372961267948 * k[377]
+1.0 * k[378]
+1.0 * k[379]
-1.0 * k[380]
+1.0 * k[381]
-0.9293680065495149 * k[382]
+1.0 * k[383]
-1.0 * k[384]
-1.0 * k[385]
+1.0 * k[386]
```

```
-1.0 * k[387]
+1.0 * k[388]
+1.0 * k[389]
-1.0 * k[390]
-1.0 * k[391]
-1.0 * k[392]
+1.0 * k[393]
+1.0 * k[394]
+0.9736961675450153 * k[395]
+1.0 * k[396]
-1.0 * k[397]
-1.0 * k[398]
-1.0 * k[399]
+1.0 * k[400]
+1.0 * k[401]
-1.0 * k[403]
-1.0 * k[405]
-1.0 * k[406]
+1.0 * k[407]
-1.0 * k[408]
-1.0 * k[409]
+0.474795788381077 * k[410]
-1.0 * k[411]
-0.39962103186412556 * k[412]
-1.0 * k[413]
+1.0 * k[414]
+1.0 * k[415]
-1.0 * k[416]
+1.0 * k[417]
-0.058421811421222325 * k[418]
-1.0 * k[419]
-1.0 * k[420]
+1.0 * k[421]
-1.0 * k[423]
+1.0 * k[424]
-1.0 * k[425]
-1.0 * k[426]
-1.0 * k[427]
+1.0 * k[428]
-1.0 * k[429]
+1.0 * k[430]
-1.0 * k[431]
-1.0 * k[432]
-1.0 * k[433]
+1.0 * k[434]
+1.0 * k[435]
+0.317376140450955 * k[436]
+1.0 * k[437]
+0.5460608243942277 * k[438]
-1.0 * k[439]
-0.7264231378883265 * k[440]
-1.0 * k[441]
+1.0 * k[442]
+1.0 * k[443]
-1.0 * k[444]
+1.0 * k[445]
-1.0 * k[446]
-1.0 * k[447]
-1.0 * k[448]
+1.0 * k[449]
+1.0 * k[450]
-1.0 * k[451]
+0.6178155670876123 * k[452]
+1.0 * k[453]
```

```
+1.0 * k[454]
+1.0 * k[456]
+1.0 * k[457]
-1.0 * k[458]
-1.0 * k[459]
-1.0 * k[460]
-1.0 * k[461]
-1.0 * k[462]
-1.0 * k[463]
+1.0 * k[464]
+1.0 * k[466]
-1.0 * k[467]
-1.0 * k[468]
-1.0 * k[469]
+1.0 * k[470]
+0.024029555173692076 * k[471]
+1.0 * k[472]
-1.0 * k[473]
-1.0 * k[474]
+1.0 * k[475]
-1.0 * k[476]
+1.0 * k[477]
+0.962036920409361 * k[478]
+1.0 * k[479]
-1.0 * k[480]
-1.0 * k[481]
-1.0 * k[482]
-1.0 * k[483]
+1.0 * k[484]
-1.0 * k[485]
+1.0 * k[486]
+1.0 * k[487]
+0.009393782840873957 * k[488]
-1.0 * k[489]
+1.0 * k[490]
+1.0 * k[491]
-1.0 * k[492]
-1.0 * k[493]
+1.0 * k[494]
-1.0 * k[495]
-1.0 * k[496]
-1.0 * k[497]
+1.0 * k[498]
+1.0 * k[499]
+1.0 * k[500]
-1.0 * k[501]
+1.0 * k[502]
-1.0 * k[503]
-1.0 * k[504]
+1.0 * k[505]
-1.0 * k[506]
+1.0 * k[507]
-1.0 * k[508]
-1.0 * k[509]
+1.0 * k[510]
-1.0 * k[511]
+1.0 * k[512]
+1.0 * k[513]
+0.5689884445585343 * k[514]
-1.0 * k[515]
+1.0 * k[516]
-1.0 * k[517]
-1.0 * k[518]
+1.0 * k[519]
```

```
+1.0 * k[521]
-0.9465413351489261 * k[522]
+0.8574274085291836 * k[523]
-1.0 * k[525]
-1.0 * k[526]
+0.22423644587538424 * k[527]
-1.0 * k[528]
-0.25347199385177177 * k[529]
+1.0 * k[530]
-1.0 * k[531]
-1.0 * k[532]
+1.0 * k[533]
+1.0 * k[534]
-1.0 * k[535]
+1.0 * k[536]
-1.0 * k[537]
+1.0 * k[538]
-1.0 * k[539]
-1.0 * k[540]
-1.0 * k[541]
-0.40836832458668787 * k[542]
+1.0 * k[543]
-0.7197116196464467 * k[544]
-1.0 * k[545]
-1.0 * k[546]
+1.0 * k[547]
-1.0 * k[548]
-1.0 * k[549]
-1.0 * k[550]
-1.0 * k[551]
+1.0 * k[552]
-1.0 * k[553]
+1.0 * k[554]
-0.027030860130143428 * k[555]
-0.5194815178810226 * k[556]
-1.0 * k[557]
+0.15643240227859737 * k[558]
-1.0 * k[559]
-1.0 * k[560]
+1.0 * k[561]
-1.0 * k[562]
-1.0 * k[563]
-1.0 * k[564]
+1.0 * k[565]
+1.0 * k[566]
-1.0 * k[567]
+1.0 * k[568]
+1.0 * k[569]
-0.43742966237973485 * k[570]
-1.0 * k[571]
-1.0 * k[572]
-1.0 * k[573]
-1.0 * k[574]
-1.0 * k[575]
+1.0 * k[576]
+0.6573776645029682 * k[577]
-1.0 * k[578]
-0.836446609343021 * k[579]
-1.0 * k[580]
+1.0 * k[581]
+1.0 * k[582]
-1.0 * k[583]
-1.0 * k[584]
+1.0 * k[585]
```



```
+1.0 * k[586]
+1.0 * k[587]
-1.0 * k[588]
+1.0 * k[589]
+1.0 * k[590]
+1.0 * k[591]
-1.0 * k[592]
-1.0 * k[593]
-1.0 * k[594]
-0.4949069836658351 * k[595]
+1.0 * k[596]
+1.0 * k[597]
-1.0 * k[598]
+1.0 * k[599]
+1.0 * k[600]
+1.0 * k[601]
+1.0 * k[602]
+1.0 * k[603]
+1.0 * k[604]
+1.0 * k[605]
+1.0 * k[606]
+1.0 * k[608]
+1.0 * k[609]
-1.0 * k[610]
+1.0 * k[611]
+1.0 * k[612]
+1.0 * k[613]
+1.0 * k[614]
+1.0 * k[615]
-1.0 * k[616]
-0.0305963545930719 * k[617]
-1.0 * k[618]
-0.43549106329151244 * k[619]
-0.44561750268702605 * k[620]
-1.0 * k[621]
-1.0 * k[622]
+0.3371878822249494 * k[623]
-1.0 * k[624]
+1.0 * k[625]
-1.0 * k[626]
-1.0 * k[627]
+1.0 * k[628]
-1.0 * k[629]
-1.0 * k[630]
+1.0 * k[631]
-1.0 * k[632]
-1.0 * k[633]
+1.0 * k[634]
+1.0 * k[635]
+1.0 * k[636]
+1.0 * k[637]
-0.8949344870784947 * k[638]
-1.0 * k[639]
+1.0 * k[640]
+1.0 * k[641]
+1.0 * k[642]
-0.35400141600578633 * k[643]
-0.7386725576173881 * k[644]
-1.0 * k[645]
+1.0 * k[646]
+1.0 * k[647]
+1.0 * k[648]
-1.0 * k[649]
+1.0 * k[650]
```

```
+1.0 * k[651]
+1.0 * k[652]
+1.0 * k[653]
-0.05562343042754429 * k[654]
+1.0 * k[655]
+1.0 * k[656]
+1.0 * k[657]
-1.0 * k[658]
-1.0 * k[659]
-1.0 * k[661]
+1.0 * k[662]
-1.0 * k[663]
+1.0 * k[664]
-1.0 * k[665]
+1.0 * k[666]
-1.0 * k[667]
+1.0 * k[668]
-1.0 * k[669]
-1.0 * k[670]
-1.0 * k[671]
+1.0 * k[672]
-1.0 * k[673]
+1.0 * k[674]
+0.8566038577600684 * k[675]
+1.0 * k[676]
-1.0 * k[677]
+1.0 * k[678]
+1.0 * k[679]
+1.0 * k[680]
+1.0 * k[681]
-1.0 * k[682]
+1.0 * k[683]
-1.0 * k[684]
+1.0 * k[685]
-1.0 * k[686]
+1.0 * k[687]
-1.0 * k[688]
-1.0 * k[689]
-0.4611361064627772 * k[690]
+1.0 * k[691]
-1.0 * k[692]
+0.5864775536226583 * k[693]
-0.0678924072831471 * k[694]
+1.0 * k[695]
-0.7589892904648089 * k[696]
+0.46473981510661005 * k[697]
-1.0 * k[698]
-1.0 * k[699]
-1.0 * k[700]
-1.0 * k[701]
-1.0 * k[702]
+0.03837591963734946 * k[703]
-1.0 * k[704]
-1.0 * k[705]
+1.0 * k[707]
+1.0 * k[708]
+1.0 * k[709]
-1.0 * k[710]
-1.0 * k[711]
+1.0 * k[712]
+1.0 * k[713]
-0.5725987609775978 * k[714]
-1.0 * k[715]
+1.0 * k[716]
```

```
-0.6310444265503652 * k[717]
-0.7299116496074143 * k[718]
-0.7657215952431434 * k[720]
+1.0 * k[721]
-1.0 * k[722]
+0.1581904851669775 * k[723]
-1.0 * k[724]
-1.0 * k[725]
+1.0 * k[726]
+1.0 * k[727]
+1.0 * k[728]
+1.0 * k[729]
+1.0 * k[730]
+0.2214731741128516 * k[731]
+1.0 * k[732]
+1.0 * k[733]
+1.0 * k[734]
-1.0 * k[735]
+1.0 * k[736]
+1.0 * k[737]
+1.0 * k[738]
-1.0 * k[739]
+1.0 * k[740]
-1.0 * k[742]
+1.0 * k[743]
-0.1965698820592391 * k[744]
-1.0 * k[745]
-0.023235892730786637 * k[746]
+1.0 * k[747]
-1.0 * k[748]
-1.0 * k[749]
+0.9222224075545065 * k[750]
-0.7570265748018958 * k[751]
+1.0 * k[752]
-1.0 * k[753]
+1.0 * k[754]
+0.3633483113964382 * k[755]
-1.0 * k[756]
+1.0 * k[757]
-1.0 * k[759]
+1.0 * k[760]
-0.7282644494135787 * k[761]
-1.0 * k[763]
+1.0 * k[764]
-0.8787220449781243 * k[765]
+1.0 * k[766]
+1.0 * k[767]
+1.0 * k[768]
-1.0 * k[769]
+1.0 * k[770]
+0.2036988545420522 * k[771]
+0.3945717419358555 * k[772]
-1.0 * k[773]
-0.11581165041441968 * k[774]
+1.0 * k[775]
-0.9718824731082687 * k[776]
+1.0 * k[777]
-1.0 * k[778]
-1.0 * k[779]
-1.0 * k[780]
-1.0 * k[781]
-1.0 * k[782]
+1.0 * k[783]
-1.0 * k[784]
```

```
-1.0 * k[785]
+1.0 * k[786]
+1.0 * k[787]
-1.0 * k[788]
-1.0 * k[789]
+1.0 * k[790]
-0.7291095681569735 * k[791]
-1.0 * k[792]
+1.0 * k[794]
-1.0 * k[795]
+1.0 * k[796]
+1.0 * k[797]
+0.46331564319209273 * k[798]
-1.0 * k[799]
-1.0 * k[800]
-1.0 * k[801]
+1.0 * k[802]
+0.14495331209640117 * k[803]
-1.0 * k[804]
+1.0 * k[805]
+1.0 * k[806]
+1.0 * k[807]
+0.7212822203211757 * k[808]
-1.0 * k[809]
+1.0 * k[811]
-0.45230160615862824 * k[812]
+0.09335342856567837 * k[813]
-1.0 * k[814]
+1.0 * k[815]
-1.0 * k[816]
+1.0 * k[817]
-0.20779248163174605 * k[818]
-1.0 * k[819]
+1.0 * k[820]
-1.0 * k[821]
+1.0 * k[822]
+1.0 * k[823]
+1.0 * k[824]
-0.40826474686724107 * k[825]
+1.0 * k[827]
+1.0 * k[828]
+1.0 * k[829]
+1.0 * k[831]
+1.0 * k[832]
-1.0 * k[833]
+1.0 * k[834]
+1.0 * k[835]
-1.0 * k[836]
-1.0 * k[837]
-1.0 * k[838]
+1.0 * k[839]
-1.0 * k[840]
-0.09673550861438368 * k[841]
-1.0 * k[842]
-1.0 * k[843]
-1.0 * k[844]
+0.43803417011889334 * k[845]
-1.0 * k[846]
+1.0 * k[847]
-1.0 * k[848]
-1.0 * k[849]
-1.0 * k[850]
-1.0 * k[851]
-1.0 * k[852]
```

```
+1.0 * k[853]
+0.25519660266403255 * k[854]
+1.0 * k[855]
-1.0 * k[856]
+1.0 * k[857]
-1.0 * k[858]
-1.0 * k[859]
+1.0 * k[860]
+0.11685884750010983 * k[861]
-0.021240323231413494 * k[862]
+0.5668129566475719 * k[863]
-1.0 * k[864]
-0.17102919912866535 * k[865]
+1.0 * k[866]
+1.0 * k[867]
+1.0 * k[868]
-1.0 * k[869]
+1.0 * k[870]
+1.0 * k[871]
-1.0 * k[872]
-1.0 * k[873]
+1.0 * k[874]
+1.0 * k[875]
-1.0 * k[876]
-1.0 * k[877]
+1.0 * k[878]
-1.0 * k[879]
+1.0 * k[880]
-1.0 * k[881]
-1.0 * k[882]
+1.0 * k[883]
-1.0 * k[884]
+1.0 * k[885]
+1.0 * k[886]
+1.0 * k[887]
+0.8062483255044816 * k[888]
+1.0 * k[889]
+1.0 * k[890]
+1.0 * k[891]
+1.0 * k[892]
+1.0 * k[893]
+1.0 * k[894]
+1.0 * k[895]
+1.0 * k[896]
-1.0 * k[897]
-1.0 * k[898]
-1.0 * k[899]
+1.0 * k[900]
-1.0 * k[901]
-1.0 * k[902]
+1.0 * k[903]
-0.33719575177181643 * k[904]
-1.0 * k[905]
-1.0 * k[906]
+1.0 * k[907]
-1.0 * k[908]
+1.0 * k[909]
+1.0 * k[910]
+1.0 * k[911]
+1.0 * k[912]
-1.0 * k[913]
-1.0 * k[914]
-1.0 * k[915]
+0.05006194303407599 * k[916]
```

```
-1.0 * k[917]
+0.9279091661209615 * k[918]
+1.0 * k[919]
-1.0 * k[920]
+1.0 * k[921]
-1.0 * k[922]
-1.0 * k[923]
+0.7877088375197555 * k[924]
-1.0 * k[925]
+1.0 * k[926]
-1.0 * k[927]
+1.0 * k[928]
-1.0 * k[929]
+1.0 * k[930]
+1.0 * k[931]
+1.0 * k[932]
+1.0 * k[933]
+0.10637290120162893 * k[934]
+1.0 * k[935]
-1.0 * k[936]
+1.0 * k[937]
-1.0 * k[938]
+0.28232932275349126 * k[939]
-1.0 * k[941]
-1.0 * k[942]
-1.0 * k[943]
+1.0 * k[944]
-1.0 * k[945]
+1.0 * k[946]
+1.0 * k[947]
-1.0 * k[948]
-1.0 * k[950]
+1.0 * k[951]
+1.0 * k[952]
-1.0 * k[953]
-1.0 * k[954]
+1.0 * k[955]
-1.0 * k[956]
-1.0 * k[957]
+1.0 * k[958]
+1.0 * k[959]
+1.0 * k[960]
+1.0 * k[961]
+1.0 * k[962]
+1.0 * k[963]
-1.0 * k[964]
+1.0 * k[965]
-1.0 * k[966]
+1.0 * k[967]
+1.0 * k[968]
+1.0 * k[969]
-1.0 * k[970]
-1.0 * k[971]
-1.0 * k[972]
-1.0 * k[974]
+1.0 * k[975]
-1.0 * k[976]
-1.0 * k[977]
-1.0 * k[978]
+1.0 * k[979]
+1.0 * k[980]
-1.0 * k[981]
+1.0 * k[982]
-1.0 * k[983]
```

```
+1.0 * k[984]
+1.0 * k[985]
-0.536219924209692 * k[986]
+0.047643309164470946 * k[987]
-1.0 * k[988]
+1.0 * k[989]
+1.0 * k[990]
+1.0 * k[991]
+1.0 * k[992]
-1.0 * k[993]
-0.21368210365767953 * k[994]
-0.3278801001389992 * k[995]
-1.0 * k[996]
-1.0 * k[997]
+0.7926596977828743 * k[999]
-1.0 * k[1000]
+0.09002726444275957 * k[1001]
-1.0 * k[1002]
+1.0 * k[1003]
+1.0 * k[1004]
-1.0 * k[1005]
+1.0 * k[1006]
-1.0 * k[1007]
-1.0 * k[1008]
-1.0 * k[1009]
-1.0 * k[1010]
+1.0 * k[1011]
-1.0 * k[1012]
-1.0 * k[1013]
-1.0 * k[1014]
-1.0 * k[1015]
-0.021865285290165074 * k[1016]
-1.0 * k[1017]
-1.0 * k[1018]
+1.0 * k[1020]
-1.0 * k[1021]
-1.0 * k[1022]
+1.0 * k[1023]
+1.0 * k[1024]
-1.0 * k[1025]
+1.0 * k[1026]
-1.0 * k[1027]
+1.0 * k[1028]
+0.051319885515555955 * k[1029]
+1.0 * k[1030]
-1.0 * k[1033]
+0.06546705114328723 * k[1034]
-0.855076404283006 * k[1035]
+1.0 * k[1036]
-1.0 * k[1037]
+1.0 * k[1038]
+1.0 * k[1039]
-1.0 * k[1040]
-1.0 * k[1041]
-0.10432895916303012 * k[1042]
+0.34736139204586075 * k[1043]
+0.087721046797636 * k[1044]
+1.0 * k[1045]
-1.0 * k[1046]
-1.0 * k[1047]
+1.0 * k[1048]
-1.0 * k[1049]
-0.9885694287906338 * k[1050]
-0.5911741890612024 * k[1051]
```

```
-1.0 * k[1052]
-1.0 * k[1053]
-1.0 * k[1054]
+0.4575323359953414 * k[1055]
+1.0 * k[1056]
-0.8975859631295727 * k[1057]
+1.0 * k[1058]
-1.0 * k[1059]
+0.5640788184692284 * k[1060]
+1.0 * k[1061]
+0.6691295704490059 * k[1062]
+1.0 * k[1063]
-1.0 * k[1064]
-1.0 * k[1065]
-1.0 * k[1066]
-1.0 * k[1067]
-1.0 * k[1068]
+0.451137074175246 * k[1069]
+1.0 * k[1071]
+1.0 * k[1072]
-0.9983659569048197 * k[1073]
+1.0 * k[1074]
-1.0 * k[1075]
-1.0 * k[1076]
+1.0 * k[1077]
-1.0 * k[1078]
+1.0 * k[1079]
-1.0 * k[1080]
-1.0 * k[1081]
-1.0 * k[1082]
+1.0 * k[1083]
+1.0 * k[1084]
-1.0 * k[1085]
-1.0 * k[1086]
+1.0 * k[1087]
-0.13855034903351282 * k[1088]
+1.0 * k[1089]
-0.4976334384531677 * k[1090]
+1.0 * k[1091]
+1.0 * k[1092]
-1.0 * k[1093]
+1.0 * k[1094]
+1.0 * k[1095]
-1.0 * k[1096]
+0.8403213300858916 * k[1097]
+1.0 * k[1098]
-0.07868133565152796 * k[1099]
+1.0 * k[1100]
-1.0 * k[1101]
-1.0 * k[1102]
+1.0 * k[1103]
-1.0 * k[1104]
+1.0 * k[1105]
-0.79695407723649 * k[1106]
+1.0 * k[1107]
+1.0 * k[1108]
-1.0 * k[1109]
+1.0 * k[1110]
-1.0 * k[1111]
+1.0 * k[1112]
+0.11103547094350595 * k[1113]
-1.0 * k[1115]
+1.0 * k[1116]
-1.0 * k[1117]
```



```
+1.0 * k[1118]
+1.0 * k[1119]
+1.0 * k[1120]
+1.0 * k[1121]
+1.0 * k[1122]
+1.0 * k[1123]
-1.0 * k[1124]
-1.0 * k[1125]
+1.0 * k[1126]
+1.0 * k[1128]
-0.11392029246186425 * k[1129]
+0.20323928371602115 * k[1130]
-1.0 * k[1131]
-1.0 * k[1132]
+1.0 * k[1133]
-1.0 * k[1134]
-0.10577930384885882 * k[1135]
-1.0 * k[1136]
-1.0 * k[1137]
-1.0 * k[1138]
-1.0 * k[1139]
-1.0 * k[1140]
+0.3490216189645393 * k[1141]
-0.7560330666394387 * k[1142]
-1.0 * k[1143]
+1.0 * k[1144]
+1.0 * k[1145]
-1.0 * k[1146]
-1.0 * k[1148]
-1.0 * k[1149]
-1.0 * k[1150]
+1.0 * k[1151]
-1.0 * k[1153]
+1.0 * k[1154]
+1.0 * k[1155]
+1.0 * k[1157]
+1.0 * k[1158]
-1.0 * k[1159]
+0.051468313340601105 * k[1160]
+1.0 * k[1161]
+1.0 * k[1162]
+1.0 * k[1163]
+1.0 * k[1164]
-1.0 * k[1165]
-0.10359186010167715 * k[1166]
+1.0 * k[1167]
-0.9871433157604592 * k[1168]
-1.0 * k[1169]
+1.0 * k[1170]
+1.0 * k[1171]
+1.0 * k[1172]
-1.0 * k[1173]
-1.0 * k[1174]
+1.0 * k[1175]
+1.0 * k[1176]
-1.0 * k[1177]
-0.8216478593285048 * k[1178]
+1.0 * k[1179]
+1.0 * k[1180]
-1.0 * k[1181]
+1.0 * k[1182]
+1.0 * k[1183]
-1.0 * k[1184]
-1.0 * k[1185]
```

```
+1.0 * k[1186]
-1.0 * k[1187]
-0.8339641760967659 * k[1189]
-1.0 * k[1190]
-0.1733084968902232 * k[1191]
+0.2701226930301737 * k[1192]
-1.0 * k[1193]
+1.0 * k[1194]
-1.0 * k[1195]
+1.0 * k[1196]
+1.0 * k[1197]
-1.0 * k[1198]
-1.0 * k[1199]
+1.0 * k[1200]
+1.0 * k[1201]
+1.0 * k[1202]
+1.0 * k[1203]
+1.0 * k[1204]
+0.003174018177162452 * k[1205]
+1.0 * k[1206]
+0.10138954697665818 * k[1207]
-1.0 * k[1208]
-1.0 * k[1209]
+1.0 * k[1210]
-1.0 * k[1211]
+1.0 * k[1212]
+1.0 * k[1213]
-1.0 * k[1214]
-1.0 * k[1215]
-1.0 * k[1216]
+1.0 * k[1217]
-1.0 * k[1218]
+1.0 * k[1219]
+1.0 * k[1220]
+1.0 * k[1221]
-1.0 * k[1222]
-1.0 * k[1223]
+1.0 * k[1224]
-0.06575147550019952 * k[1225]
+1.0 * k[1226]
+1.0 * k[1227]
-1.0 * k[1228]
+1.0 * k[1229]
-1.0 * k[1230]
+1.0 * k[1231]
-0.06983024704829556 * k[1232]
-1.0 * k[1233]
-1.0 * k[1234]
+1.0 * k[1235]
-1.0 * k[1236]
+1.0 * k[1237]
+1.0 * k[1238]
-1.0 * k[1240]
+0.8247915977263075 * k[1241]
+1.0 * k[1242]
-1.0 * k[1243]
-1.0 * k[1244]
+1.0 * k[1245]
+1.0 * k[1246]
+0.07227978724531965 * k[1247]
+1.0 * k[1248]
+1.0 * k[1249]
-1.0 * k[1250]
+1.0 * k[1251]
```

```
+1.0 * k[1252]
-1.0 * k[1253]
+1.0 * k[1254]
-1.0 * k[1255]
-1.0 * k[1256]
-1.0 * k[1257]
+1.0 * k[1258]
-1.0 * k[1260]
-1.0 * k[1262]
+1.0 * k[1263]
+1.0 * k[1264]
-1.0 * k[1265]
+1.0 * k[1266]
-1.0 * k[1267]
+1.0 * k[1268]
+1.0 * k[1269]
-1.0 * k[1271]
+1.0 * k[1272]
+1.0 * k[1273]
+1.0 * k[1274]
-1.0 * k[1275]
-1.0 * k[1276]
-1.0 * k[1277]
+1.0 * k[1278]
-1.0 * k[1279]
+1.0 * k[1280]
-1.0 * k[1281]
+1.0 * k[1282]
-1.0 * k[1283]
-1.0 * k[1284]
-1.0 * k[1285]
+1.0 * k[1286]
-1.0 * k[1287]
+1.0 * k[1288]
+1.0 * k[1289]
-0.3827717557462844 * k[1291]
-1.0 * k[1292]
-1.0 * k[1293]
+1.0 * k[1294]
-1.0 * k[1295]
+1.0 * k[1296]
+0.03975246714978066 * k[1297]
+0.2633396081453308 * k[1298]
+0.3794594629654527 * k[1299]
-1.0 * k[1300]
+1.0 * k[1301]
+0.31918961756310543 * k[1302]
-1.0 * k[1304]
+1.0 * k[1305]
+1.0 * k[1306]
+0.1840590841250348 * k[1307]
+1.0 * k[1308]
-1.0 * k[1309]
-0.2513005724169569 * k[1310]
+1.0 * k[1311]
-0.009420031538181987 * k[1312]
-1.0 * k[1313]
+1.0 * k[1314]
+1.0 * k[1315]
+0.3641197764793174 * k[1316]
-1.0 * k[1317]
-1.0 * k[1318]
+1.0 * k[1319]
-1.0 * k[1320]
```

```
-1.0 * k[1321]
-0.058626607170328485 * k[1322]
-1.0 * k[1323]
+1.0 * k[1324]
-0.057980371549176624 * k[1325]
+1.0 * k[1326]
+1.0 * k[1327]
-1.0 * k[1328]
-1.0 * k[1329]
+1.0 * k[1330]
+1.0 * k[1332]
-1.0 * k[1333]
-1.0 * k[1334]
-0.7765109339055645 * k[1335]
-1.0 * k[1336]
+1.0 * k[1337]
-1.0 * k[1338]
-1.0 * k[1339]
-1.0 * k[1340]
+1.0 * k[1341]
-0.0648906243160674 * k[1342]
+1.0 * k[1343]
+1.0 * k[1344]
-1.0 * k[1345]
+1.0 * k[1346]
-1.0 * k[1347]
+1.0 * k[1348]
-1.0 * k[1349]
-1.0 * k[1350]
-1.0 * k[1351]
-1.0 * k[1352]
+1.0 * k[1354]
+1.0 * k[1355]
+1.0 * k[1356]
-1.0 * k[1357]
-1.0 * k[1358]
-1.0 * k[1359]
-1.0 * k[1360]
-1.0 * k[1361]
+1.0 * k[1362]
-1.0 * k[1363]
-0.6140189843682908 * k[1364]
-0.9567445061867597 * k[1365]
+0.748399507758999 * k[1366]
+0.9985005570200769 * k[1367]
-1.0 * k[1368]
-1.0 * k[1369]
-1.0 * k[1370]
+1.0 * k[1371]
+1.0 * k[1372]
-0.3434754031824682 * k[1373]
+1.0 * k[1374]
-1.0 * k[1375]
-1.0 * k[1376]
-0.37950250415944015 * k[1377]
-0.02044564623242874 * k[1378]
+1.0 * k[1379]
-1.0 * k[1380]
+1.0 * k[1381]
+1.0 * k[1382]
-1.0 * k[1383]
-0.011102472443296318 * k[1384]
+0.7111524339101915 * k[1385]
+1.0 * k[1387]
```

```

+1.0 * k[1388]
-1.0 * k[1389]
-1.0 * k[1390]
-0.34941005607140274 * k[1391]
+1.0 * k[1392]
+1.0 * k[1394]
-1.0 * k[1395]
+1.0 * k[1396]
-1.0 * k[1397]
-1.0 * k[1398]
+1.0 * k[1399]
+1.0 * k[1400]
+1.0 * k[1401]
-1.0 * k[1402]
+1.0 * k[1403]
+0.8541476346141995 * k[1404]
+1.0 * k[1405]
-1.0 * k[1406]
+1.0 * k[1407]
+1.0 * k[1409]
-1.0 * k[1410]
-1.0 * k[1411]
+1.0 * k[1412]
-1.0 * k[1413]
+1.0 * k[1414]
+0.22875318215298487 * k[1415]
-1.0 * k[1416]
+1.0 * k[1417]
-1.0 * k[1418]
-1.0 * k[1419]
+1.0 * k[1420]
+1.0 * k[1421]
-0.2440754349022638 * k[1422]
-1.0 * k[1423]
+1.0 * k[1424]
-1.0 * k[1425]
+1.0 * k[1426]
-1.0 * k[1427]
+1.0 * k[1428]
-1.0 * k[1429]
+0.15774352396069205 * k[1430]
-1.0 * k[1431]
-1.0 * k[1432]
+      0.5407

```

Number of kernel evaluations: 1768963 (88.709% cached)

Time taken to build model: 1.63 seconds

=== Predictions on test split ===

inst#,	actual,	predicted,	error
1	1.093	1.021	-0.073
2	1.02	0.998	-0.022
3	0.978	1	0.022
4	1.005	1.005	0
5	0.932	0.985	0.053
6	1.086	1.064	-0.023
7	1.044	0.999	-0.045
8	0.953	0.999	0.045
9	0.992	1.018	0.026
10	1.039	1.007	-0.032

11	0.952	0.994	0.042
12	0.896	0.989	0.093
13	1.174	1.077	-0.097
14	1.053	1.034	-0.02
15	1.027	0.994	-0.033
16	1.049	1.024	-0.025
17	1.047	1.008	-0.038
18	1.06	1.039	-0.022
19	0.961	1.011	0.05
20	1.084	1.066	-0.018
21	1.012	1.032	0.02
22	1.016	1.012	-0.003
23	1.013	1.016	0.003
24	1.03	1.022	-0.008
25	0.984	1.002	0.019
26	0.977	1.003	0.026
27	1.035	1.05	0.016
28	1.012	1.02	0.008
29	0.989	1.005	0.017
30	1.021	1.01	-0.011
31	0.984	1.008	0.025
32	0.985	0.998	0.013
33	0.964	0.99	0.026
34	1.048	1.04	-0.008
35	1.023	1.013	-0.01
36	0.98	0.996	0.016
37	0.997	0.988	-0.009
38	0.982	0.985	0.004
39	1	0.992	-0.008
40	0.998	0.988	-0.01
41	1.029	1.034	0.006
42	1.003	1.006	0.003
43	0.975	1.011	0.036
44	1.007	0.988	-0.019
45	1.01	0.99	-0.021
46	1.015	0.998	-0.017
47	0.998	0.986	-0.013
48	1.026	1.042	0.016
49	1.012	1.018	0.006
50	1	1.013	0.013
51	0.993	0.998	0.006
52	0.992	1.005	0.013
53	0.99	0.987	-0.003
54	0.977	0.973	-0.004
55	1.021	1.048	0.027
56	0.997	1.005	0.008
57	0.971	1.012	0.041
58	1.019	0.992	-0.027
59	1.005	0.994	-0.011
60	0.988	0.993	0.006
61	0.977	0.975	-0.002
62	1.028	1.05	0.022
63	0.989	1.001	0.012
64	0.974	0.992	0.018
65	0.969	0.992	0.023
66	0.986	0.97	-0.016
67	0.973	0.987	0.014
68	0.967	0.977	0.01
69	1.009	1.034	0.025
70	0.976	1.014	0.038
71	0.995	0.997	0.002
72	0.983	1.007	0.024
73	0.97	1.002	0.032
74	0.977	0.979	0.001

75	0.979	0.975	-0.004
76	0.958	1.017	0.059
77	0.969	1.014	0.045
78	1.014	0.978	-0.036
79	0.991	0.99	-0.001
80	0.989	0.997	0.009
81	1.039	0.988	-0.051
82	0.847	0.992	0.144
83	1.168	1.078	-0.09
84	1.006	1.019	0.012
85	1.024	1.002	-0.022
86	1.034	1.027	-0.007
87	1.03	1.028	-0.002
88	0.999	1.015	0.016
89	0.981	0.989	0.008
90	1.02	1.03	0.01
91	1.019	1.006	-0.013
92	0.987	0.998	0.011
93	1.001	0.989	-0.012
94	1.023	1	-0.023
95	0.983	0.999	0.017
96	0.942	0.971	0.028
97	1.035	1.036	0.001
98	1.018	1.013	-0.005
99	1.017	1.018	0.001
100	1.009	1.01	0.001
101	0.965	0.98	0.014
102	0.94	0.962	0.022
103	1.002	0.973	-0.029
104	1.067	1.072	0.006
105	0.97	1.093	0.124
106	0.978	1.026	0.048
107	0.979	0.996	0.017
108	0.981	1.008	0.027
109	0.964	1	0.035
110	1.002	0.998	-0.004
111	1.055	1.045	-0.009
112	0.976	0.995	0.019
113	0.991	0.973	-0.017
114	0.949	0.977	0.027
115	1.016	1.007	-0.008
116	1.016	1.007	-0.009
117	1.006	1.002	-0.004
118	1.1	1.033	-0.067
119	0.938	0.996	0.057
120	0.992	1.003	0.01
121	0.992	1.001	0.009
122	0.944	0.991	0.047
123	0.981	1.005	0.024
124	0.99	0.99	0
125	1.027	1.034	0.006
126	1.023	1.011	-0.012
127	1.047	0.994	-0.053
128	0.971	0.986	0.015
129	1.006	0.997	-0.009
130	0.969	0.994	0.025
131	0.959	0.986	0.027
132	1.097	1.055	-0.042
133	0.941	0.986	0.045
134	0.97	0.961	-0.009
135	0.957	0.988	0.031
136	0.949	0.986	0.037
137	0.96	0.974	0.015
138	1.001	0.984	-0.017

139	1.044	1.025	-0.019
140	1.016	1.021	0.005
141	0.978	0.998	0.019
142	0.986	0.976	-0.011
143	1.026	1.022	-0.004
144	1.036	1.021	-0.015
145	1.002	1.013	0.011
146	1.047	1.006	-0.041
147	1.003	0.998	-0.004
148	0.975	0.988	0.013
149	0.988	0.972	-0.016
150	1.075	1.019	-0.056
151	0.944	0.988	0.044
152	1.015	1.004	-0.011
153	1.069	1.042	-0.027
154	0.976	0.988	0.012
155	1.03	0.935	-0.096
156	0.96	1.008	0.049
157	0.956	0.975	0.019
158	0.986	0.969	-0.017
159	0.934	0.982	0.048
160	1.038	1.02	-0.018
161	1.009	1.009	-0.001
162	1.021	1.007	-0.014
163	1.059	0.989	-0.07
164	0.944	0.979	0.035
165	0.977	0.971	-0.006
166	0.951	0.988	0.036
167	1.056	1.028	-0.027
168	0.985	0.984	-0.002
169	0.974	0.972	-0.002
170	0.975	0.986	0.011
171	1	0.967	-0.033
172	0.977	0.991	0.015
173	1.013	0.992	-0.021
174	1.058	1.028	-0.03
175	1.034	1.005	-0.029
176	1	0.982	-0.018
177	1.016	0.991	-0.025
178	1.002	0.99	-0.011
179	0.981	0.99	0.009
180	0.971	0.986	0.015
181	1.06	1.033	-0.027
182	1.009	1.017	0.008
183	1.009	0.995	-0.014
184	0.967	0.968	0.001
185	0.99	0.99	0.001
186	0.98	0.992	0.012
187	0.996	1.001	0.005
188	1.054	1.034	-0.02
189	0.988	1.019	0.031
190	0.998	1.006	0.008
191	0.986	1.004	0.018
192	0.976	0.996	0.02
193	0.972	0.984	0.012
194	0.987	0.993	0.007
195	1.02	1.025	0.005
196	0.996	1.003	0.007
197	0.974	0.984	0.01
198	0.951	0.988	0.037
199	0.996	0.978	-0.018
200	0.979	0.963	-0.015
201	0.963	0.982	0.019
202	1.035	1.027	-0.008

203	0.989	1.011	0.022
204	0.98	0.988	0.008
205	0.961	0.963	0.002
206	1.014	0.997	-0.017
207	0.999	0.993	-0.006
208	0.973	0.979	0.006
209	1.044	1.003	-0.04
210	1.005	0.999	-0.006
211	0.97	0.969	-0.001
212	1.014	0.982	-0.032
213	1.016	0.997	-0.018
214	0.993	0.988	-0.005
215	0.982	0.979	-0.003
216	1.062	1.019	-0.043
217	1.035	1.001	-0.034
218	1.018	0.993	-0.026
219	1.012	0.997	-0.015
220	0.941	0.988	0.047
221	0.967	0.956	-0.011
222	1.002	0.99	-0.012
223	1.045	1.038	-0.007
224	0.964	0.984	0.02
225	1.016	0.985	-0.03
226	1.021	0.996	-0.026
227	0.998	0.989	-0.009
228	0.959	0.971	0.012
229	0.962	0.986	0.024
230	1.068	1.012	-0.056
231	1.05	1.021	-0.029
232	1.012	1.031	0.019
233	1.004	1.005	0.001
234	0.987	0.961	-0.027
235	0.973	0.958	-0.015
236	0.994	1.007	0.014
237	1.096	1.056	-0.04
238	0.95	0.937	-0.013
239	1.011	1.009	-0.002
240	0.978	1.016	0.038
241	1.011	0.975	-0.036
242	0.974	0.991	0.018
243	0.979	0.976	-0.003
244	1.049	1.037	-0.013
245	0.981	1.003	0.022
246	0.98	0.978	-0.002
247	0.986	0.979	-0.007
248	0.896	0.967	0.07
249	0.993	0.984	-0.009
250	0.904	1.004	0.101
251	1.062	0.99	-0.073
252	1.14	1.008	-0.132
253	0.989	1.034	0.044
254	0.927	0.984	0.057
255	1.041	0.986	-0.056
256	1.006	0.971	-0.035
257	0.984	0.969	-0.016
258	1.055	1.014	-0.041
259	1.021	1.002	-0.018
260	1.007	0.999	-0.008
261	1.003	0.982	-0.021
262	0.989	0.983	-0.006
263	0.852	0.965	0.113
264	0.982	0.974	-0.008
265	1.147	1.067	-0.08
266	0.895	1.006	0.11

267	1.08	0.986	-0.094
268	0.974	0.937	-0.037
269	0.85	0.948	0.098
270	0.985	1.006	0.021
271	0.868	1.052	0.184
272	1.375	1.069	-0.305
273	0.955	0.99	0.036
274	1.076	1.003	-0.072
275	0.979	0.938	-0.041
276	1.029	0.974	-0.055
277	0.945	0.962	0.017
278	0.972	1.007	0.035
279	1.129	1.044	-0.085
280	1.009	1.034	0.025
281	1.042	0.991	-0.051
282	1.028	1.005	-0.023
283	0.964	0.988	0.024
284	0.976	0.947	-0.029
285	0.975	1.027	0.051
286	1.105	1.033	-0.071
287	0.978	1.008	0.031
288	0.99	0.993	0.003
289	0.93	0.994	0.064
290	1.045	0.969	-0.075
291	0.911	1.035	0.124
292	1.05	0.979	-0.071
293	1.072	1.008	-0.064
294	0.994	1.005	0.011
295	0.934	0.967	0.034
296	1.015	0.999	-0.016
297	1.046	0.998	-0.048
298	0.872	0.976	0.104
299	0.771	0.945	0.175
300	1.293	1.146	-0.147
301	1.115	1.054	-0.06
302	1.011	1.038	0.027
303	1.075	1.012	-0.063
304	1.014	0.988	-0.026
305	0.955	1.006	0.051
306	0.997	0.986	-0.011
307	1.064	1.035	-0.029
308	1.001	1.01	0.009
309	1.007	0.979	-0.028
310	0.97	0.98	0.009
311	0.97	0.969	-0.001
312	0.915	0.96	0.045
313	1.008	0.993	-0.015
314	1.115	1.101	-0.014
315	0.999	1.023	0.025
316	0.973	0.97	-0.004
317	0.959	0.989	0.03
318	1.029	0.991	-0.037
319	0.948	0.929	-0.019
320	0.921	1.012	0.091
321	1.203	1.094	-0.11
322	0.966	1.025	0.058
323	1.013	0.986	-0.026
324	0.983	0.976	-0.006
325	1.003	0.991	-0.012
326	0.896	0.988	0.092
327	1.058	1.006	-0.052
328	1.114	1.082	-0.031
329	1.033	1.015	-0.018
330	1.002	1.008	0.006

331	0.989	1.022	0.033
332	0.975	0.997	0.021
333	0.988	0.979	-0.009
334	0.962	0.977	0.014
335	1.045	1.034	-0.011
336	0.972	0.992	0.02
337	0.993	0.967	-0.025
338	0.97	0.973	0.003
339	0.983	0.982	-0.001
340	0.95	0.983	0.033
341	0.997	0.986	-0.012
342	1.04	1.047	0.007
343	1.012	1.001	-0.012
344	1.003	0.948	-0.055
345	0.999	0.994	-0.005
346	1.008	1.01	0.001
347	0.938	0.978	0.04
348	0.839	0.988	0.15
349	1.195	1.074	-0.122
350	0.924	0.994	0.07
351	1.11	0.988	-0.122
352	0.925	0.949	0.024
353	1.017	1.01	-0.007
354	0.973	0.941	-0.032
355	0.608	0.983	0.375
356	1.4	1.19	-0.211
357	1.098	1.237	0.139
358	1.05	1.237	0.187
359	0.84	0.953	0.112
360	1.062	0.977	-0.085
361	1.124	1.007	-0.117
362	0.926	0.927	0.001
363	1.122	1.015	-0.107
364	0.987	1.01	0.022
365	0.997	0.996	-0.001
366	1.003	0.988	-0.015
367	0.991	0.993	0.002
368	0.966	0.997	0.032
369	1.018	0.979	-0.039
370	1.055	1.053	-0.002
371	1.006	1.017	0.011
372	1.007	1.051	0.044
373	0.994	1.008	0.014
374	1.016	1.003	-0.013
375	0.994	1.011	0.016
376	0.946	0.98	0.034
377	1.062	1.058	-0.004
378	0.937	1.002	0.065
379	0.977	0.969	-0.008
380	0.979	0.978	-0.002
381	1.017	0.998	-0.019
382	1.005	0.995	-0.01
383	0.95	0.996	0.046
384	1.036	1.067	0.031
385	1.004	0.984	-0.02
386	1.011	1.001	-0.01
387	1.002	0.987	-0.015
388	0.959	1.004	0.046
389	0.916	0.972	0.056
390	0.974	0.984	0.011
391	1.018	1.067	0.049
392	1.024	1.017	-0.007
393	0.903	0.958	0.054
394	0.987	1.014	0.027

395	0.861	0.943	0.082
396	1.036	0.997	-0.039
397	0.902	0.891	-0.011
398	1.178	1.101	-0.077
399	1.063	0.974	-0.089
400	0.944	1.046	0.102
401	1.053	0.997	-0.056
402	0.972	0.999	0.027
403	1.029	0.991	-0.038
404	0.931	1.011	0.081
405	1.05	1.093	0.043
406	1.008	1.02	0.012
407	1.014	1.022	0.007
408	0.976	0.982	0.006
409	1.028	1.007	-0.021
410	1.001	0.98	-0.021
411	0.994	0.994	0
412	1.027	1.065	0.038
413	0.991	1.031	0.039
414	0.978	1.009	0.031
415	0.985	1.004	0.02
416	0.962	1.004	0.041
417	1.011	0.978	-0.033
418	0.949	0.983	0.034
419	1.087	1.062	-0.025
420	0.981	1.022	0.042
421	1.063	1.034	-0.029
422	1.086	1.043	-0.043
423	0.997	1.043	0.046
424	0.926	1.008	0.082
425	0.873	0.971	0.097
426	1.187	1.133	-0.054
427	1.059	1.001	-0.058
428	0.971	0.986	0.014
429	1.093	1.025	-0.068
430	1.037	1.046	0.009
431	1.013	1.088	0.074
432	0.995	1.045	0.05
433	1.016	1.016	0.001
434	1.059	1.023	-0.036
435	1.012	1.021	0.009
436	1.042	1.015	-0.027
437	1.03	1.006	-0.024
438	0.974	0.987	0.013
439	1.02	1.018	-0.001
440	1.041	1.042	0.001
441	1.014	1.016	0.002
442	1.001	0.992	-0.009
443	1.017	0.992	-0.025
444	0.998	1.004	0.006
445	0.982	0.981	-0.001
446	0.963	0.978	0.014
447	1.064	1.047	-0.017
448	1.015	1.009	-0.006
449	1.013	0.999	-0.014
450	0.988	1.004	0.016
451	1.01	1.003	-0.006
452	0.973	0.982	0.009
453	0.968	0.981	0.014
454	1.052	1.017	-0.034
455	1.003	0.986	-0.017
456	0.997	0.976	-0.02
457	0.997	1.023	0.027
458	1.012	0.993	-0.019

459	0.971	0.968	-0.003
460	1.009	0.984	-0.025
461	1.023	1.07	0.048
462	1.023	1.008	-0.016
463	1.008	1.008	0.001
464	1.061	0.988	-0.074
465	0.904	0.975	0.071
466	0.953	0.972	0.019
467	0.989	0.981	-0.008
468	1.04	1.065	0.025
469	0.983	0.982	-0.001
470	0.997	0.991	-0.006
471	0.977	0.989	0.012
472	0.978	0.965	-0.013
473	0.968	0.973	0.004
474	0.956	1.005	0.05
475	1.056	1.042	-0.014
476	1.041	1.002	-0.039
477	1	1	0
478	0.94	0.994	0.054
479	1.007	0.992	-0.016
480	0.984	1	0.016
481	1.003	0.996	-0.007
482	1.053	1.065	0.012
483	1.033	1.01	-0.023
484	1	1.006	0.006
485	1.022	1.008	-0.014
486	0.961	0.994	0.033
487	0.973	0.976	0.003

=== Evaluation on test split ===

=== Summary ===

Correlation coefficient	0.6045
Mean absolute error	0.031
Root mean squared error	0.0484
Relative absolute error	79.1488 %
Root relative squared error	79.6929 %
Total Number of Instances	487