Østfold University College

Master's Thesis

Integrating NLP Techniques to Enhance Automated Essay Evaluation

Syeda Miral Kazmi

October 15, 2022

Masters in Applied Computer cience

Faculty of Computer Science, and Communication

Abstract

The ability of a student to communicate their ideas in writing is an important factor to consider when evaluating their creativity, knowledge, and intelligence. As a direct consequence of this, academic writing is frequently required to be submitted as part of the application process for universities and colleges as well as on standardized examinations and in classroom evaluations. The question of how to successfully evaluate the essays submitted by students using a set of criteria for writing that is somewhat objective has always been one that has been up for debate. When it comes to the grading of essays, however, the job of a teacher can be rather challenging. If this is the case, then the instructor may find that the automated essay scoring system is a useful aid in the process of decision-making and marking students' score.

Marking student's essay scores manually is a challenging task for teachers so in order to solve this problem, I held this thesis and built an automated essay marking system. This thesis work does a comparative analysis of several machine learning models using supervised machine learning algorithms and various strategies for vectorization. Models such as Ridge, Linear Regression, Random Forest, K Nearest Neighbor, and Decision Tree were explored as part of this thesis. These models were trained using the essays that were part of a provided labeled data set.

Evaluation of models were done using multiple evaluation matrices. These metrices were evaluated using training & testing variance score, Root mean squared error, Mean absolute error and R-squared score. Random forest was performing best than other models with training variance score equal to 99 percent and test variance score equal to 95 percent that was highest than any other models.

Keywords: Natural Language Processing, AES, Automated Essay Grading, Vectorization, Machine Learning, Bag of words, Sentiment Analysis, Syntactic Analysis, POS tagging.

Acknowledgment

My heartiest gratitude to my advisor Professor Hasan Ogul for guiding and the support throughout the master's course and thesis with sustained enthusiasm and motivation.

I would also like to thank Østfold University for providing opportunities in pursuing my master's degree without financial obstacles. The support of my family and friends was immense in maintaining the inspiration for completing this thesis.

1 Contents

Acknowledgment

List of Tables

List of Figures

List of Figures

1. Introduction & Problem Statement

1.1. Background

Writing well for academic purposes is an essential ability for language students to develop, and it is also a significant component of many standardized language examinations, such as the International English Language Testing System (IELTS) and the Test of English as a Foreign Language (TOEFL). Standardized testing is used to evaluate writing because it demonstrates an author's capacity for a highly thorough and innovative use of the language, which enables the author to communicate their views. An author is evaluated based on their ability to execute a writing job by utilizing the language skills and writing methods they have acquired. An essay of excellent quality will include well-developed subjects, a clear and focused stance, ideas that have been effectively generated and organized, and a clear and focused position.

Figure 1 (Hearst, M., 2000) illustrates the progression of writing assessment systems over the course of several decades. This timeline does not include all that happened. Research and development conducted by Educational Testing Services provided the foundation for this estimate.

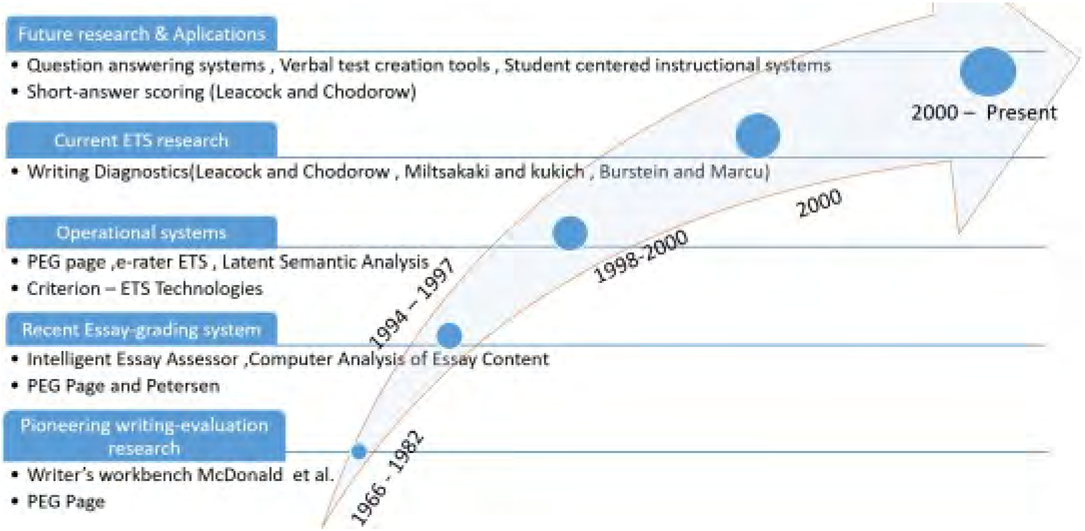

Future research & Aplications
• Question answering systems , Verbal test creation tools , Student centered instructional systems
• Short-answer scoring (Leacock and Chodorow)
2000 – Present
Current ETS research
• Writing Diagnostics(Leacock and Chodorow , Miltsakaki and kukich , Burstein and Marcu)
2000
Operational systems
• PEG page ,e-rater ETS , Latent Semantic Analysis
• Criterion – ETS Technologies
1998-2000
1994 – 1997
Recent Essay-grading system
• Intelligent Essay Assessor ,Computer Analysis of Essay Content
• PEG Page and Petersen
Pioneering writing-evaluation research
• Writer's workbench McDonald et al.
• PEG Page
1966 - 1982

Figure 1 Evaluation of Essay grading

It should come as no surprise that grading essays demands a significant amount of effort from the examiners because they are required to not only comprehend the material but also make an informed determination on the essay's overall quality (Zhang, Mo, 2013). The process of composing evaluations has gotten much more laborious, time-consuming, and pricey in recent years due to the rise in the number of tests as well as the total number of pupils. In addition, there is a possibility of bias occurring if the evaluation is only dependent on a human rater who is not

1.2. Problem Definition

Essay grading is one of the major works that the professors do while examining the papers. Grading is an important factor which access the potential student have in terms of their ability of writing. Unjustified grading system could have negative consequences on the student's confidence. It can demotivate them and could destroy their whole passion for this skill. Previously all of the essay grading is being done manually by teachers which is complicated task in itself. One has to spend a lot of time to grade an essay because essay grading is based on a lot of things like the structure, length, coherence, content quality etc. Secondly, maybe because of biasness, some of students have high marks and some have low even if they deserve better. This might result in discouraging students who have high quality writing skills.

Except that manual evaluation is very time consuming and a lot of human power is required to evaluate an essay manually. If the essay is being done by pen and paper, then it is more costly and not even environment friendly due to paper wastage and we are already facing a lot of serious issues. To tackle this Automated Essay Grading System could be the solution to overcome these problems. In this thesis, I set out to explore and test out a machine learning based Automated Essay Grading System and present the findings in relations to problems mentioned above.

1.3. Motivation

Various automated grading systems has already been built by other researchers. vast majority of them rely on very fundamental characteristics like the number of words and paragraphs in an essay as well as the average length of a sentence is the primary cause of their unreliability. Automated essay grading systems place a greater emphasis on the length and organization of the essay, rather than the essay's actual substance and overall quality. So, they focused only on the fundamental characteristics of essays, rather than content itself that being written in an essay. On the other hand, some of the researchers just focus on the content of essay and they do not consider the structure and fundamental parameters of the essay for evaluation. Therefore, the motivation in this thesis is to build a system that will consider both the content and structure of the essay to mark the grades, and that could be possible by increasing quantity of data that is accessible to work with and applying machine learning's algorithms as an increasingly appealing choice for a potential solution to this issue. This is one beneficial development that we can make in the field of automated essay grading. As also pointed out in (Zupanc, K., & Bosnić, Z., 2018), a grading model has to learn from data that indicates a chaotic connection between the characteristics of an essay and the mark it receives.

This project's objective is to build an automated grading system which will be based upon both the structure and content of essay and also, an attempt to increase the accuracy of the automatically generated essay scores by testing multiple algorithms. The flow of the project is illustrated in broad form in Figure 2, which may be seen here.

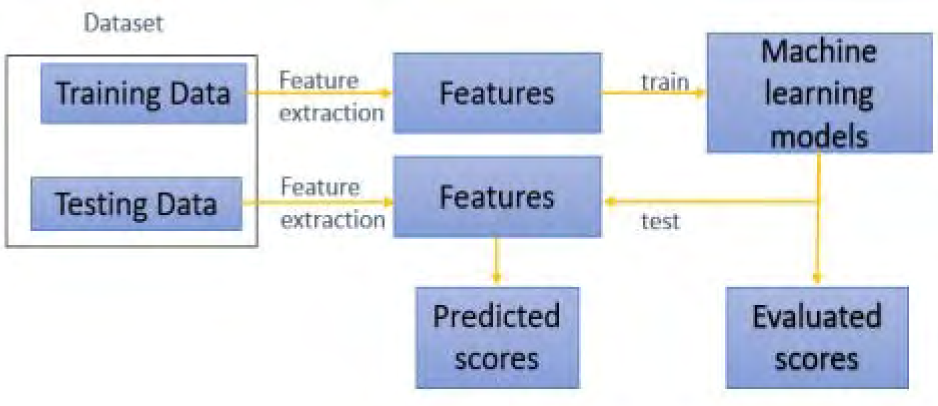


Figure 2 Flow of AES

1.4. Research Question

This paper aims to study different deep learning algorithms used for evaluating essays. We will use different techniques to automate the evaluation of essay. This can be beneficial in numerous diverse ways. By using an accurate algorithm, we can avoid ton of manual human work. This can also help to overcome problems like unfairness or human bias in the evaluation process.

We have tried to address various parameters like the progress of ideas, significance of the content timely, Consistency, and Rationality which needs to be keep in mind while evaluating an essay.

Following research questions (RQ) are framed to collect and answer through my research. These questions are:

- **RQ1:** What kinds of datasets are accessible for study on the process of automatically evaluating essays?

  We can use the open-source essays from various dataset for evaluating essays which provides a diverse data and assists in training an accurate model. We have used data called ASAP provided by The Hewlett Foundation was a part of competition on Kaggle.

- **RQ2:** What will be the effect of sentiment analysis on the overall accuracy?
  We will perform the sentiment analysis on dataset and further will study the results of all the models with and without considering sentiment analysis to see how it effects the overall performance of the models.

- **RQ3:** Which characteristics are taken into consideration during the grading of essays?
  For essay type of dataset, common characteristics considered are semantic, sentiment and syntax of the text. For this paper we will be using sentiment and syntactic characteristics of the essays.
- **RQ4:** What kinds of assessment measures are there to choose from in order to determine how accurate algorithms are?
  It is necessary to determine it for more accurate results. In this paper, we will be using methods like standard deviation, Training and testing sets, cross validation and repeated random test-train Splits to assess the accuracy of results.

- **RQ5:** How are the various Machine Learning approaches that are utilized in the process of automatically evaluating essays put into practice?

  Some fundamental machine learning approaches are supervised, reinforcement, semi-supervised and unsupervised learning. Which approach one should use entirely depends on dataset they want to predict. In this paper, we are using supervised learning because we are using training dataset, that contains samples of inputs and respective outputs.

1.5. AES

The Automated Essay Scoring (AES) project is a collaborative initiative including researchers from the fields of Education, Linguistics, and Natural Language Processing (NLP). The capacity of an NLP model to analyse long-term dependencies and infer meaning even when text is badly written is one of the measures that determines whether or not it is effective in AES. NLP enables to convert the text into a form that is acceptable by machine learning algorithms. Machine learning models cannot work directly with text, so natural language processing provides techniques through which we can clean our text and then can convert into useable format. Then after that machine learning models takes the processed data by NLP and make predictions.

We will also be using natural language processing techniques with machine and deep learning algorithms to build a successful automated essay scoring system. Vectorization and POS Tagging are among the methods being implemented in this thesis.

1.6. Report Outline

- Chapter 2: Literature Review

  This chapter provides background information of research work which already exist related to automated essay assessment and NLP algorithms. We also discuss the approaches we implemented.

- Chapter 3: Methodology

  This chapter describes various methods required for NPL models. We discuss the challenges involved in each method. We also explain the requisite of this methods to yields the required outcomes. We also explained the approaches that are involved.

- Chapter 4: Data

  We discuss about the available open-source dataset for this study, and we also explained statical description of the dataset. we will perform different procedures for data processing, it helps not to have negative effects on the performance of end result.

- Chapter 5: Model Implementation

  Model Implementation gives a detailed explanation of the complete execution of the various models. We present the figures and code that we use to implement different algorithms. We also intend to describe the algorithms and technologies used to detect an abnormality in the remote lab.

- Chapter 6: Results

  This section shows evaluation of all the models while comparing with the actual values. We will also compare the performance of different models and evaluate tradeoff of each model based on different parameters and scenarios. We address the advantages and limitations of our implementation.

- Chapter 7: Discussion

  In this chapter we will discuss about the techniques and machine learning models for automated grading of essays. We also explain assessment metrics of different methods and chosen dataset.

- Chapter 8: Conclusion & Future works

  Future Work and Conclusion reviews the contributions of the thesis and aims improvement and guidance of future work to make the essay assessment more efficient.

2. Literature Review

Research on writing assessment and implementation first began several decades ago and has since continued for increasingly sophisticated automated evaluation systems. The study work that was done on the essay grading or writing assessment is presented in the discussion piece that was written by (Hearst, M., 2000). It describes the progression of automated assessment tools, beginning with the PEG Writer's workbench and ending with the short answer scoring systems, which were created between the years 1960 and 2000.

2.1. Evolution of AES

2.1.1. Writing Quality and Computational Linguistics

The acceleration in the development of AES systems in recent years can be attributed to the use of text mining, online learning, and word processing software. Natural language processing is the source from which computational linguistic characteristics are most frequently obtained. The Coh-Metrix software is widely used for analyzing text. Memphis University developed a web-based text analysis tool called the Coh-Metrix (Graesser, A., McNamara, D., Louwerse, M., 2004) with the intention of determining the degree to which the textbooks used by students in the United States adhere to a logical framework. It is possible for Coh-Metrix to extract a broad variety of characteristics from the text. These variables can include things like the text's coherence and cohesiveness, syntactic difficulty, vocabulary information, and conceptual clarity. Text mining has developed to the point that it can make a reliable assessment of the quality of a piece of writing, and it is currently widely utilized in a wide range of academic fields and specializations (McNamara, D. S., Crossley, S. A., Roscoe, R. D., Allen, L. K., 2015). There are now just a few linguistic characteristics that are utilized for automated essay grading, and these characteristics focus on different parts of language. Grammar and syntax are the two components of language that are the easiest to learn and become proficient at. Among its eleven macro-features, e-rater, for instance, contains nine language characteristics and two content characteristics. The nine linguistic components that comprise the framework of an essay are, in order from most important to least important, writing, grammar, style, word frequency, and convention. There is a plethora of tiny traits that can be readily tallied and computed, in addition to the larger ones that were discussed before and mentioned above. The amount of time spent writing the essay has a considerable bearing on the characteristics of the structure. The section under "Measuring the Ideas and Content of Essays" discusses why it is more difficult to digitize content-related components of essays.

In contrast to the AES research conducted in English, the AES study conducted in Chinese got off to a sluggish beginning. When natural language processing methods such as word segmentation reached a more advanced stage of development in China at the beginning of the twenty-first century, a small group of Chinese AES researchers emerged (Zhang, J., and Ren, J., 2004), (Zhang J., and Ren, J. , 2014). Professor Liu and his colleagues at the Harbin Institute of Technology developed an automatic grading system for Chinese writing that may be used for the National College Entrance Examination (NCEE), the Chinese Proficiency Test (MHK), as well as for writing assignments in the classroom (Hao, S. D., Xu, Y. Y., Peng, H. L., Su, K. L., 2014), (Gong, J., 2016) (Fu, R., Wang, D., Wang, S., 2018). In the beginning, Chinese-oriented AES systems would extract information from Chinese text by using surface language characteristics like as the frequency of certain words (Zhang, J., and Ren, J. , 2014).

In recent years, there has been an increased emphasis placed on the recognition and study of rhetoric in Chinese essays, such as the detection of parallel and dual sentences as well as the identification of quotations (Gong, J., 2016). The quality of Chinese essays has been connected to linguistic characteristics discovered by academics working in other fields, like as computational linguistics. It was found that writers writing in Chinese had the greatest reliance distance out of all 20 languages that Zhang and Liu examined.

Linguists have shown that the complexity of a person's writing is connected to their writing talent, and a study conducted by (Wang, Y., and Liu, H., 2019) demonstrated that a person's ability to write in Chinese increases their syntactic dependency distance. The level of syntactic complexity may have an impact on the grade of an essay; however, further study is required to verify this hypothesis. We make use of a variable that is dependent on the dependency distance as a signal of syntactic difficulty in order to make projections on the quality of a composition.

This reveals that the current version of the AES does not place a great deal of importance on the characteristics of deep literary traits such as ideas and content when determining the overall quality of a piece. On the other hand, AES has a concentration that might be contested about this matter. If the elements of the grading system do not cover the most significant components of an essay's quality, then additional discussion of an automatic grading system should take place.

2.1.2. Examining Essays with Regard to Their Concepts and Material

AES is currently able to measure and evaluate two distinct categories of essay components, including the following: language style features, such as vocabulary and syntax, language norms and processes (such as English spelling and capitalization), and content or semantic elements. In the current AES techniques, the qualities of the first category of essays are given a high level of importance throughout the evaluation process. On the other hand, their capacity to evaluate the second type of content, which consists of elements with semantic features, is restricted.

On the other hand, AES content evaluation systems often base their content ratings on a comparison of an essay's language and substance with that of other essays that received high content scores (Landauer, T., Foltz, P., and Laham, D., 1998) (Kakkonen, T., Myller, N., Sutinen, E., and Timonen, J., 2008). Despite the fact that it is challenging to evaluate the quality of the ideas presented in an essay, several studies have attempted to do so. Despite this, (Somasundaran, S. Riordan, B., Gyawali, B, 2016) did not rate the major theme while evaluating the development of the essay.

The primary idea is often a crucial factor that is taken into consideration in the majority of the rubric rating systems. It is often described as having a distinct focus and demonstrates the author's viewpoint and perspective, as is the case below. It is usual practice to include the central idea as an essential factor to be considered when developing grading criteria or using a grading rubric. The author's individual feelings and points of view are referred to as the primary focus of the key notion (Cui, X., 2001). Students are required to first analyze the prompt and then consider how to express all of the aspects of the essay since the AES from this research is used for the topical writing job, which is common in Chinese writing examinations. The usage of Open Information Extraction, Fuzzy Logic, and Description Logic are among the most recent innovations in AES system design. Semantic Networks, Ontology, and Ontology are also among the more recent innovations (Zupanc, K., and Bosnic, Z., 2017) It is tough to evaluate the quality of the material and the central concept since semantic granularity analysis is such a demanding endeavor.

2.1.3. Use of Graphs in Writing's Evaluation

Communication relies on having a mental model that is built on perception and comprehension, and language is that model (Garnham, A., 1981), (Johnson-Laird, 2005). An effective method for evaluating the conceptualization of an essay is to analyze the essay using a concept map graph structure. This may be done by analyzing the essay (Kim, 2013). A successful expression of an author's thoughts in the form of a structured knowledge representation made up of concepts and relationships may be found in the graph structure (Seel, N., 1999) (Villalon, J., and Calvo, 2011). Linguists believe that essay sentences have surface and deep structures, aspects of sentence shape, and deep structure semantic substance (including the connections between concepts).

The hidden importance of the deep structure can be uncovered by careful analysis of the surface structure (Bransford, J. D., and Johnson, 1972). (Jin, H., and Liu, H., 2016) conducted research on the structural characteristics of human language at several levels making use of modern Chinese as an example. This research was accomplished by employing a complex network methodology. According to Jin and Liu's research, the four Chinese network models (character co-occurrence, word co-occurrence, syntactic relationship, and semantic relationship) each display their own unique statistical features and reflect the similarities and linkages at all levels of the system. These models are character co-occurrence, word co-occurrence, syntactic relationship, and semantic relationship, respectively. Both the similarities and differences between these two systems provide convincing evidence of the existence of a robust relationship between language- related properties and human cognition. Recent innovations in AES employ graph theory rather than vectors to investigate the complexities of text as a network structure rather than a single vector. This allows for a more comprehensive analysis of the text (Somasundaran, S., Riordan, B., Gyawali, B, 2016).

The progression of the essay as well as the semantic connections between the various words may be noticed in the graph structure. To keep track of how the essay has developed, Somasundaran utilized the characteristics of the graph structure that were determined from the information included in the essay. The outcomes of the research show that the overall quality as well as the grading accuracy of concept creation may be improved by using a technique that is based on graph structure. To evaluate Chinese literature, use complex networks, taking into account things like in- and out-degree, clustering coefficient, and dynamic network features. (Ke et al. 2016) As a consequence of this, the connection between the qualities of the graph and the criteria for evaluating essays was not looked into very thoroughly in this research.

2.2. Relevant work

Work on AES was started very earlier, and a lot of other researchers built automated essay scoring systems using different techniques. Some of the important ones are explained below with the techniques they used.

2.2.1. Project essay Grade (PEG)

Ellis Page was the first individual to construct an AES system, which he accomplished at the request of the American College Board. The earliest iteration of this system, known as Project Essay Grade (PEG) (Page, 1994), was presented in 1966. The major characteristic of PEG is that it emphasizes the investigation of the surface structure of language over the analysis of its content. In addition, it relies heavily on the statistical principle of regression, wherein the number of words and the number of word-forms in essay serve as independent variable and the essay score serves as the dependent variable. The essay is ultimately judged based on a variety of quantitative elements. The PEG predicts the student's ability to express oneself based on the length of the essay, the number of various word forms predicts their command of word usage, and the variance in word length predicts their vocabulary. Training is conducted using regression analysis to obtain correlation coefficients between the parameters of interest and the grading of the text, resulting in the automatic scoring of essays.

2.2.2. Intelligent Essay Assessor (IEA)

In the late 1990s, Knowledge Analysis Technology, a division of the Pearson Group, created IEA (Intelligent Essay Assessor) (K. K. Y. Chan, T. Bond, and Z. Yan, 2022). IEA was the first automated essay scoring system based on latent semantic analysis, a statistical analytic approach that incorporates essay content analysis as a significant scoring reference indication. The fundamental premise of IEA is taken from Latent Semantic Analysis(LSA) (A. Kaur and M. Sasi Kumar, 2019), a statistical approach created by the psychologist Thomas Lindauer that is a statistical calculation to extract the precise meaning of words and phrases in a given context. It begins by describing the various semantic units of a composition in a high-dimensional semantic space, with each semantic unit being a point in this space. The semantic similarity between two semantic units is then evaluated based on their relative distance in the semantic space. LSA is a complicated statistical approach for acquiring and representing knowledge. It is a statistical model for statistical analysis of the semantics of words, based on the word bag theory, in which all the words in a text are grouped together, and if one of the words changes, the text's semantic information will also change. The latent semantic analysis of text assumes that the semantic quality of a text is dictated by the words inside the text and creates a text word matrix. It is necessary to remove dummy words from the text, i.e., words that have no real meaning but occur frequently, because increasing or decreasing the number of these words has little effect on the semantics of the text but increases the dimensionality of the word text vector and makes the calculation more challenging. In most situations, the document frequency and inverse document frequency (TF-IDF) values are employed as members of the text-word matrix (T. K. Landauer, D. Laham, and P. W. Foltz, 2000). The text vector representing the composition to be evaluated is compared with the text vector in the semantic space, the similarity is utilized as a weighted vector to add up the ratings of the training composition, and then the composition's semantic rating is obtained. The score for the essay's semantics is determined.

2.2.3. E-Rater

E-rater, which stands for "Electronic Essay Rater," is a system which is used by the Educational Testing Service in the United States for grading essays. It was made by Burstein et al. in the late 1990s as the first AES system to be used on a large-scale standardized exam. The system is based on techniques for processing natural language that are based on artificial intelligence and a regression algorithm that works like PEG. Natural language comprehension is the use of statistics, machine learning, and other research methods to give computers a natural understanding of human language. This makes it possible for computers and people to communicate freely with each other. E-rater uses syntactic analysis, expository analysis, and theme analysis as its main ways of processing natural language. The goal of syntactic analysis is to break up the text and look closely at the structure of the sentences, including the use of virtual voice and other complex phrases, in order to figure out what kinds of sentences are likely to be in the text. The expository analysis looks at how the text's phrases connect to each other and how they are put together in terms of standard writing rules. The purpose of a thematic analysis is to figure out how good an essay is by looking at the words it uses. So, E-Rater was made by using these three techniques.

2.3. Research Gap

The summary of above evolutions and systems that has been built illustrates that Page (PEG) used only the quantitative variables for estimating the essay scores. The content of Essay wasn't evaluated. Similarly in IEA, Technology Analysis used a statistical approach named Latent Symantic Analysis to extract the meaning of words and then using that extracted information, they built similar algorithms to mark student essay grade. While Burstein also used statistical model on processed data with the mix of three techniques to make the electronic essay rater. Therefore is still roam for work to be done in this field. They did not use any famous vectorization techniques. Some of them were focused on quantity of the essay, while others focused on content. In those studies, they did not consider to use them both at the same time.

So, in this research I will be using both structure and content to grade the essay. For the content I am using the actual text of essays and for structure I am calculating different quantitative measures like length of essay, number of nouns, pronouns, adjectives and so on many other structural quantities. And then on the merging of both content and structure I aim to build a machine or deep learning algorithm to predict the essay score.

Additionally, instead of applying only a single algorithm, to yield better and redundant results multiple algorithms are applied in the research.

3. Methodology

This chapter provides an overview of tools and methods that have been opted in this thesis. Our methodology relies on machine learning, so tools are discussed first followed by the methodology.

3.1. POS Tags

POS is used to identify the part of speech, and it frequently also indicates additional grammatical categories, such as tense, number (plural/singular), case, and so on. Searches conducted on corpora, as well as tools and algorithms for analyzing text, make use of POS tags.

A tagset is a collection of all of the POS tags that have been applied to a corpus. Typically, tagsets for several languages are distinct from one another. It is not always the case that they are entirely distinct for languages that are not linked to one another and highly similar for languages that are related to one another. Tagsets can also be broken down into several degrees of specificity. Tags for only the most common components of speech may be included in more fundamental tag sets (N for noun, V for verb, A for adjective etc.). On the other hand, it is more typical to go into further depth and differentiate between single and plural forms of nouns, as well as verbal conjugations, tenses, aspects, voices, and a great deal more. It is also possible that individual researchers would build their own highly specific tagsets in order to meet the requirements of their research.

POS tags are used in languages like English, where a same word can have distinct meanings depending on how it is used, such as the case with the word "work." These tags are used to differentiate between the uses of the word when it is a noun, a verb, or both.

One may also use POS tags to look for instances of grammatical or lexical patterns without having to identify a particular word. For instance, you might use them to look for instances of any plural noun that is not preceded by an article.

In this research I identified POS tags for each word and then counted them per essay. And the counted calculation behaves as the structural vectors of essay. For instance, how many verbs, nouns, adjectives, pronoun etc. have been used. Furthermore, I concatenated grammatical patterns with their respective words.

3.2. Vectorization Method

Word Embeddings, also known as Word Vectorization, is a technique used in natural language processing (NLP) to map words or phrases from a lexicon to a matching vector of real numbers. These vectors may then be used to determine word predictions, word similarities, and word semantics. There are many vectorization techniques available in Natural Language Processing. In this thesis following techniques are being used:

- Bag of Words (BOW)
- Term Frequency Inverse Document Frequency (TFIDF)

3.2.1. Bag Of Words (BOW)

The method of text modelling known as bag of words is utilized in Natural Language Processing (NLP). It is also referred as a method of feature extraction using text data if we speak in words that are more technical. Extracting features from documents may be done in a manner that is both straightforward and adaptable using this method.

A representation of text that represents the recurrence of words inside a document is known as a bag of words. Usually, the purpose is to count the number of words and pay no attention to the specifics of the grammar or the sequence of the words. Because none of the information on the order or structure of the words in the text is kept, the collection of words is referred to as a "bag".

A lexicon of known words and a measurement of the existing known words are both components of a bag-of-words. It gives a detailed account of the placement of words inside a given manuscript.

The model is concerned simply with determining whether or not known terms are included in the document. It makes the assumption that papers are comparable to one another if they include material that is similar to other documents and attempts to deduce the meaning of a document based solely on the content of the document.

We are unable to feed text into the algorithms used in NLP in a straightforward manner. They operate with numerical data. The text is transformed into a "bag-of-words" format by the model. The bag-of-words feature in that text keeps a tally of how many times the most common terms appear across the whole document.

The model takes the text and converts it into vectors of a specified length after counting the number of times each word appears. In this thesis, I applied BOW on the combination of the word and it’s part of speech.

3.2.2. Term Frequency Inverse Document Frequency (TFIDF)

For the purposes of document search and information retrieval, TF-IDF was developed. It does this by growing inversely proportionately to the number of times a word appears in a document, while simultaneously decreasing inversely proportionally to the total number of documents that include the term. Therefore, terms that are prevalent in every text, such as this, what, and if, score low despite the fact that they may occur a lot since they don't signify too much to that particular paper.

On the other hand, the fact that the term "Bug" appears several times in one document but it does not appear numerous times in others suggests that the information being discussed is likely extremely pertinent. For instance, if what we're doing is attempting to figure out which categories specific NPS replies fall under, the term "Bug" would very certainly end up being associated with the category "Reliability," given that the majority of responses including that phrase would be concerning the category in question.

The TF-IDF score of a word in a document is determined by multiplying the following two distinct metrics:

- The number of times a word appears in a given manuscript. There are a few different approaches to computing this frequency, with the most straightforward one being a simple count of the number of times a word appears in a given document. The length of a text or the raw frequency of the word that appears the most often in a document are two approaches to change the frequency.
- The number of papers in which the term appears less frequently than the total number of documents. This indicates how frequent or uncommon a term is throughout the full set of documents. The number's proximity to zero indicates the frequency with which a term is used. To determine this metric, take the total number of documents, divide that number by the number of documents that include a word, and then compute the logarithm of the result.

Therefore, if the term is used extremely frequently and appears in a large number of papers, this value will become closer and closer to 0. If not, it will become closer and closer to 1.

The TF-IDF score of a word in a given text may be calculated by multiplying these two integers together. When the score is greater, it indicates that the term in question is more pertinent to the content of the given text. Along with BOW, TFIDF is also applied on the string combination generated from the out of POS Tagging.

3.3. Sentiment Analysis Method

In NLP Sentiment analysis is a technique which is used to determine if the given text is positive, negative, or neutral. This can be used by brands to monitor their business and customer feedback. This can improve a brand by understanding customer feedback, urgency and/or sometimes intentions. Thus, this is important to understand human feeling regarding a sentence. In sentiment analysis, each sentence is assigned sentiment score known as Polarity. Based on polarity and threshold we can determine sentiment of the sentence. It is very important to determine threshold for accurate results. Some other use cases were sentiment analysis can be used are analysis users data on social media, reviews, news, political commentary, etc.

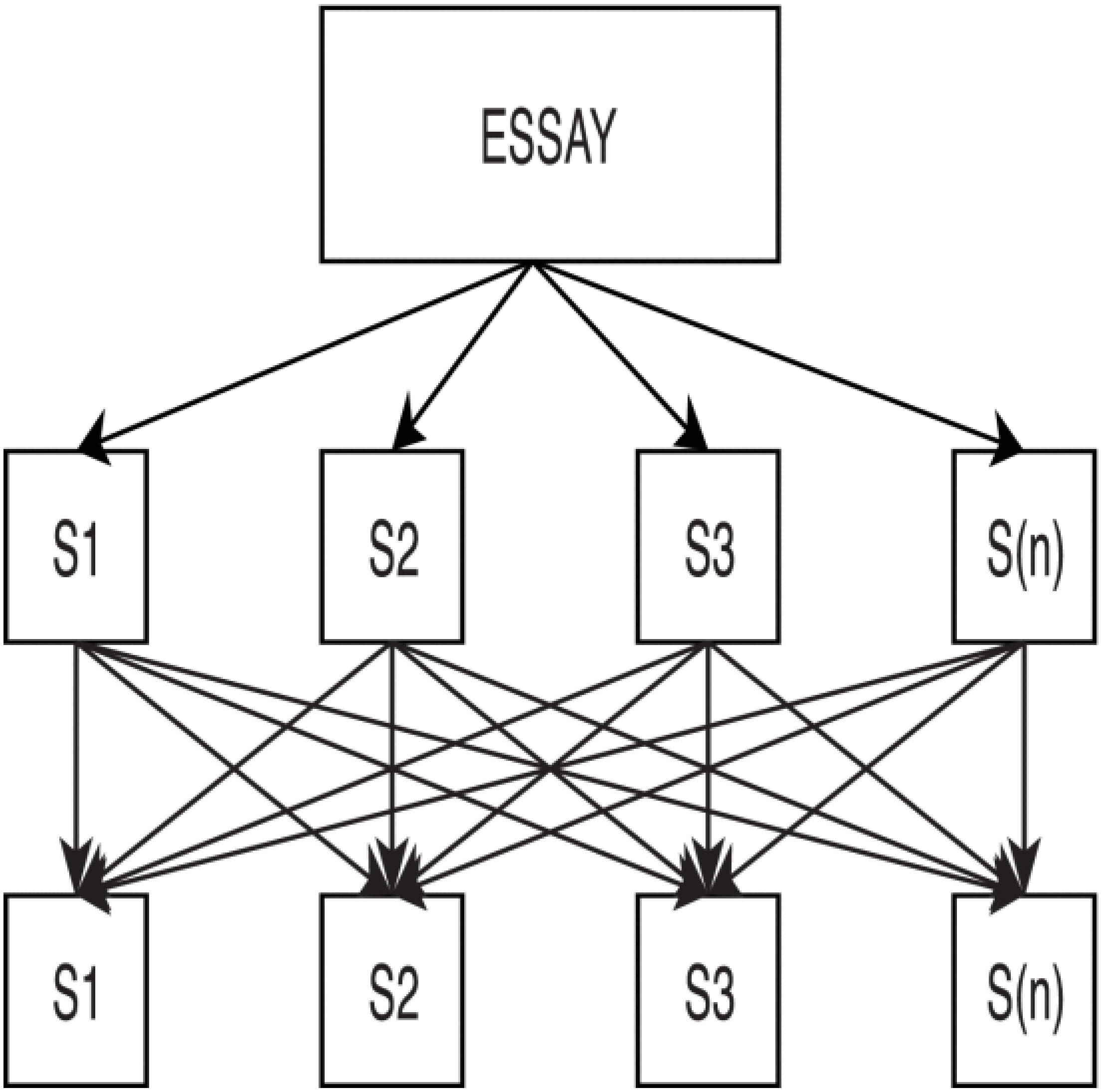


Figure 3 Sentiment Analysis Method

3.4. Unsupervised ML Techniques

Unsupervised learning, also known as unsupervised machine learning, is a method of learning that makes use of machine learning algorithms in order to evaluate and cluster unlabeled information. These algorithms uncover previously unknown patterns or data groupings without requiring any participation from a human researcher. Due to its capacity to detect similarities and contrasts in information, it is the best answer for exploratory data analysis, cross-selling techniques, customer segmentation, and picture identification. Moreover, it can also recognize images.

From unsupervised learning I used Topic Modeling in this thesis. The process of topic modelling is a form of machine learning that involves doing an automated analysis of text data in order to identify cluster terms for a collection of texts. Because it does not require a preexisting list of tags or training data that has been previously categorized by humans, this kind of machine learning is known as "unsupervised" machine learning.

Because topic modelling does not require training, it is a fast and simple approach to begin the process of studying your data. However, you cannot ensure that the results you obtain will be correct; for this reason, many companies choose to spend time training a topic categorization model instead. In order to infer themes from unstructured data, topic modelling entails counting words and categorizing word patterns that are similar to one another. For instance if one owns a software firm and is interested in learning what their clients think about certain aspects of the product they sell to them. An analysis of the texts using a topic modelling algorithm rather than spending a lot of time manually reading through piles of comments in an effort to determine which texts are discussing the subjects that are of interest.

A topic model groups input that is similar together, as well as phrases and expressions that appear most frequently, by identifying patterns such as the frequency of individual words and the distance between individual words. With this knowledge, it is quite easy to figure out what each group of sentences is talking about. Keep in mind that this method is "unsupervised," which means that there is no prerequisite training.

3.5. Supervised Machine Learning

The sort of machine learning known as supervised learning is one in which machines are trained using training data that has been appropriately "labelled," and then, using that data as a basis, machines predict the output. Data that has been labelled indicates that some of the input data has already been tagged with the appropriate output.

During the process of supervised learning, the training data that is given to the machines serves in the role of the teacher, instructing the machines on how to accurately predict the output. It utilizes the identical intellectual framework that is imparted to a pupil under the watchful eye of the educator.

The process of giving the machine learning model with both the appropriate input data and the desired output data is known as supervised learning. Finding a mapping function that will map the input variable (x) onto the desired output variable (y) is the goal of every supervised learning algorithm (y).

Supervised learning has a variety of applications in the real world, including risk assessment, image classification, fraud detection, and spam filtering, among others. The development of machine learning models is accomplished through the application of classification and regression strategies in supervised learning (Fišer, Darja and Jakob Lenardič, 2019).

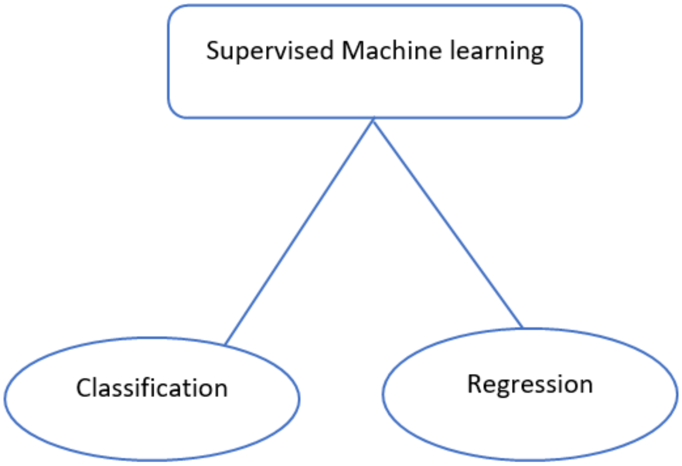


Figure 4 Types of supervised machine learning

3.5.1. Classification

The supplied data are categorized thanks to classification models. Classification approaches predict discrete responses. For instance, the email is authentic, or it is spam, and the tumor may be malignant, or it may be benign. Applications such as medical imaging, speech recognition, and credit scoring are examples of typical uses.

Taxonomy should be used if your data can be labelled, sorted into specific groups or classes, or categorized in any other way. Applications that can identify handwriting, for instance, make use of categorization in order to decipher written letters and numbers. Techniques of unsupervised pattern recognition are utilized in the fields of image processing and computer vision for the purposes of object detection and image segmentation.

3.5.2. Regression

Methods of regression can accurately forecast continuous reactions, such as shifts in temperature or variations in the amount of demand for power. The forecasting of power loads and algorithmic trading are two examples of typical uses (Fu, R., Wang, D., Wang, S., 2018). Use regression approaches whenever you are dealing with a data range or whether the nature of your answer is a real number, such as the temperature or the amount of time left until a piece of equipment breaks down. Linear and nonlinear models, regularization, stepwise regression, boosted and bagged decision trees, neural networks, and adaptive neuro-fuzzy learning are examples of common regression algorithms.

Our target is to predict the Essay score that is continuous feature so our problem is a regression problem not a classification. So, for that I used supervised machine learning and deep learning models for the prediction of Essay Score.

- Ridge
- Decision Tree
- Linear Regression
- K Nearest Neighbour
- Random Forest
- LSTM

3.5.2.1. Ridge Model

In situations in which the variables being studied are linearly independent but strongly correlated, ridge regression may be used as a technique for estimating the coefficients of multiple regression models. It has been implemented in a variety of disciplines, such as econometrics, chemistry, and engineering, among others.

In the works "RIDGE regressions: biased estimation of nonorthogonal issues" and "RIDGE regressions: applications in nonorthogonal problems" that were published in the journal Technimetrics in 1970 by Hoerl and Kennard, the knowledge was initially introduced for the first time (Hoerl, A. E., & Kennard, R. W, 1970).

Research into the topic of ridge analysis has been going on for 10 years prior to this discovery being made.

By creating a ridge regression estimator, ridge regression was developed as a possible solution to the imprecision of least square estimators when linear regression models have some multicollinear (highly correlated) independent variables. Ridge regression was developed as a possible solution to the imprecision of least square estimators when linear regression models have some multicollinear (highly correlated) (RR). This results in an estimate of the ridge parameters that is more accurate than prior estimates, since the variance and mean square estimator of this method are frequently found to be fewer than those generated using the least square method (Ghanta, Harshanthi, 2019).

Multicollinear data may be analyzed using Ridge regression, a model tuning approach. L2 regularization is accomplished using this approach. Due to the fact that least squares are unbiased, and variances are high, the projected values are far distant from the actual predicted values when there is multicollinearity.

The cost function for ridge regression:

$$Min\ (\left||Y - X(\boldsymbol{\theta})\right||^{2} + \lambda\left||\boldsymbol{\theta}\right||^{2})$$

The word for assessing penalties is lambda. The alpha parameter in the ridge function is indicated by here. We may regulate the penalty term by varying the alpha values. The penalty is greater and the size of the coefficients is less as alpha increases (Valenti, Salvatore, Francesca Neri, 2003).

- It reduces the scope. As a result, it serves as a safeguard against multicollinearity.
- The model's complexity is reduced as a result of the coefficient shrinking.

3.5.2.2. Decision Tree

A decision tree (see reference number is a representation of a classifier that is expressed as a recursive split of the instance space. The decision tree is made up of nodes that come together to form what is known as a root tree (Zupanc, K., and Bosnic, Z., 2017). This indicates that the decision tree is a distributed tree with a fundamental node known as root and no incoming edges.

Every single one of the other nodes only has a single incoming edge. A node is referred to be an internal node or a test node if it contains any edges that lead away from it. Leaves are the name given to the remaining nodes on the tree. Each test node in a decision tree is responsible for partitioning the instance space into two or more sub-spaces by applying a specific discrete function to the values that are input. In the most straightforward scenario, each test takes into account a single attribute, and the instance space is partitioned in accordance with the value of the attribute. In the event that the property in question is a number, the condition will relate to a range.

Each leaf is designated to a particular class, which ultimately denotes the value that is considered optimal.

There is a possibility that the leaf will store a probability vector that provides an indication of the likelihood of the target property having a particular value. The cases are categorized by navigating them from the root of the tree down to the leaf, with the classification being determined by the results of the tests that are performed along the journey (Shermis, Mark D and Jill C, 2003).

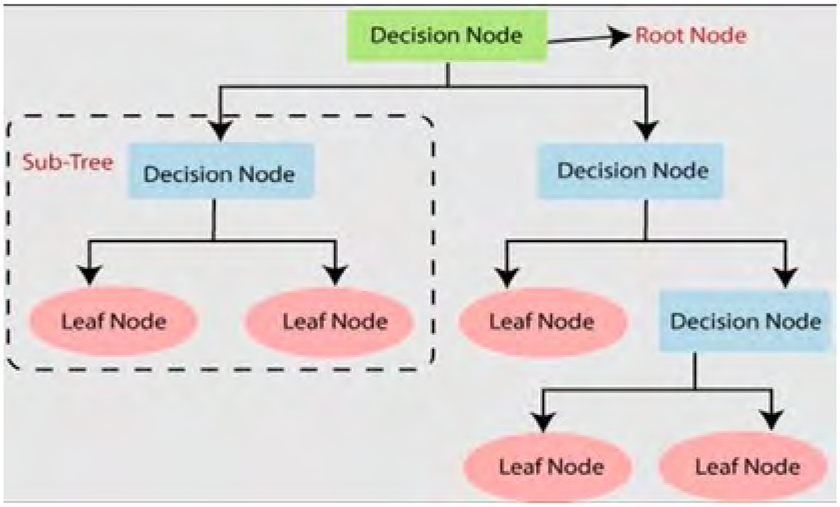


Figure 5 Illustration of Decision Tree

A straightforward application of the decision tree is shown in Figure IV. Each node in the tree is labelled with the attribute that it checks, and each branch that it has is labelled with the value that corresponds to it.

The analyst is able to forecast the response of some potential consumers and comprehend the behavioral features of the entire population of potential customers given this classifier.

3.5.2.3. Random Forrest

One of the more well-known machine learning algorithms, Random Forest, is categorized under the more general category of supervised learning. As its name suggests, it’s a classifier, "that has a number of decision trees on several subsets of the dataset and calculate the average to improve the predictive accuracy of that dataset." Random Forest is a technique that "takes the average to the predictive accuracy of that dataset." The random forest model does not rely on a single decision tree; rather, it considers the prediction from each tree in the forest and determines the final output based on which tree's prediction received the majority of votes (Madnani, Nitin and Aoife Cahill, 2018).

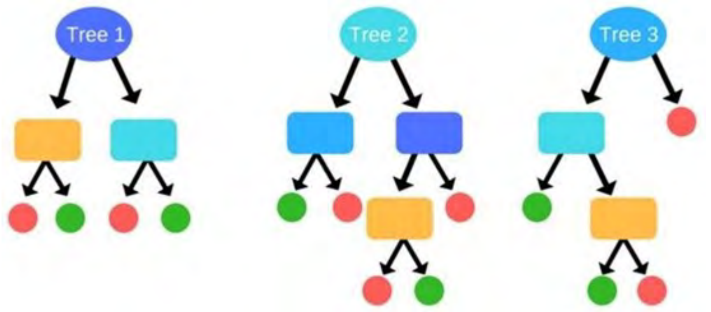


Figure 6 Working of Random Forrest

The greater the number of trees in the forest, the higher the level of accuracy achieved, as well as the prevention of the issue of overfitting.

- Linear Regression

The purpose of the linear regression, which belongs to the family of algorithms known as regressions, is to locate correlations and dependencies that exist between the variables being studied. It is a representation of a modelling relationship between a continuous scalar dependent variable denoted as y (also referred to as a label or target in the terminology of machine learning) and one or more (a D-dimensional vector) explanatory variables (also referred to as independent variables, input variables, features, observed data, observations, attributes, dimensions, data point, etc.) denoted as X. This relationship is modelled using a linear function. In regression analysis, the objective is to make a prediction about a target variable that is continuous, but in the field of classification, the goal is to make a prediction about a label chosen from a set that is finite. The model for a multiple regression that uses linear combinations of the input variables has the following form: (Mark, J. and Goldberg, M.A., 1988).

$$Y = b_0 + b_1X_1 + b_2X_2 + \cdots + b_nX_n$$

In this technique, we learn the model using a collection of labelled data, which is referred to as training data, and then we use the model to predict labels on data that has not been labelled (testing data).

3.5.3. Recurrent Method

An artificial neural network that processes sequential or time series data is referred to as a recurrent neural network, or RNN for short. These deep learning techniques are frequently used for ordinal or temporal issues, such as language translation, natural language processing (nlp), speech recognition, and picture captioning; they are integrated into famous applications like Siri, audio search, and alexa. For example, nlp stands for natural language processing, which refers to the processing of natural language. Learning in recurrent neural networks is accomplished by the use of training data, much as it is in feedforward and convolutional neural networks (CNNs). They are distinguished from one another by their "memory," which refers to the fact that information from earlier inputs is used to modify the input and output in the present. The output of recurrent neural networks is dependent on the components that came before it in the sequence, in contrast to the belief held by traditional deep neural networks that inputs and outputs does not depends on eahother. Even if future occurrences could also be helpful in defining the output of a given sequence, unidirectional recurrent neural networks are unable to include them into their predictions. This is despite the fact that such future occurrences might be useful.

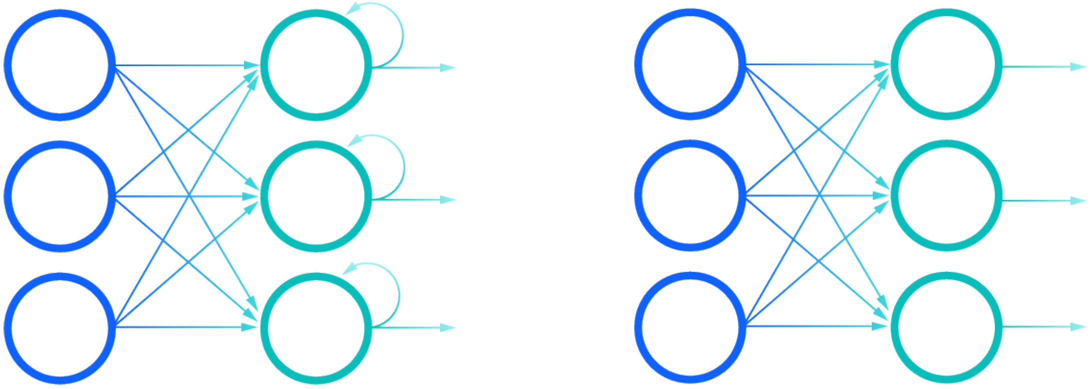


Figure 7 Recurrent VS Feed Forward Neural Network

One further thing that sets recurrent networks apart from other types of networks is the fact that their parameters are shared throughout all of the network levels. Recurrent neural networks, on the other hand, use the same weight parameter throughout each layer. This is in contrast to feedforward neural networks, which have unique weights assigned to each node in the network. However, in order to facilitate reinforcement learning, these weights are still updated through backpropagation and gradient descent(Ouyang, X., Zhou, P., Li, C.H. and Liu, L., 2015).

The backpropagation through time (BPTT) technique is used to compute gradients in recurrent neural networks. This approach is significantly different from ordinary backpropagation due to the fact that it is specific to sequence data. Traditional backpropagation is the conceptual foundation for BPTT, in which the model "self-trains" by computing errors from its output layer to its input layer. This is how conventional backpropagation works. Because of these computations, we are able to precisely adjust and fit the parameters of the model. The traditional approach does not total errors at each time step, but the BPTT does. This is in contrast to feedforward networks, which do not need to total errors since they do not exchange parameters between layers.

I used one recurrent technique that are LSTM and also tested it with attention. I created complex structure and trained them for the predictions of essay score.

3.5.4. Long Short-Term Memory

In the field of deep learning, LSTM is a sort of artificial recurrent neural network (RNN) architecture that is used to store and retrieve data for the goal of creating predictions about time series data. These predictions may then be utilized to make decisions. LSTM neural networks, on the other hand, make use of feedback connections, in contrast to more traditional feedforward neural networks. Handling individual data points (such photographs, for example) is possible, but the system is also able to analyse whole data streams (such as speech or video). The tasks of unsegmented, linked handwriting recognition and voice recognition as well as network traffic anomaly detection (also known as IDSS) are such examples (intrusion detection systems).

LSTM network models are a subtype of recurrent neural networks that are capable of learning and remembering through extensive sequences of input data. LSTM network models are also known as long short-term memory networks. They are designed for use with information that comprises of extended sequences of data, often ranging between 200- and 400-time steps in duration. They could be an appropriate fit for addressing this matter.

The long-term reliance issue that recurrent neural networks (RNNs) are plagued with is the primary motivation for the development of long short-term memory (LSTM) networks (due to the vanishing gradient problem). LSTMs are distinguished from more conventional feedforward neural networks by the presence of feedback connections in their architecture. This property enables LSTMs to process entire sequences of data (for example, time series) without treating each point in the sequence independently. Instead, they retain useful information about previous data in the sequence to help with the processing of new data points. This enables LSTMs to process entire sequences of data (for example, time series). As a consequence of this, LSTMs are exceptionally effective when it comes to the processing of sequences of data such as text, audio, and general time series(Ma, Y., Peng, H., Khan, T., Cambria, E. and Hussain, A., 2018).

Consider the following scenario: we are attempting to forecast the monthly sales of ice cream. These range from their lowest point in December all the way up to their greatest point in June, as one might assume, depending on the month of the year.

This repeating pattern that occurs once every 12 iterations of time is able to be learned by an LSTM network. It does not only rely on the prior forecast but rather remembers the context across a longer period of time, which allows it to circumvent the challenge of long-term reliance that is encountered by other models. It is important to note that this is a fairly basic example; nonetheless, LSTMs become increasingly beneficial when the pattern is separated by considerably longer periods of time (for example, in lengthy stretches of text).

3.6. Evaluation Metrices

Evaluation metrics are utilized in order to determine how well a statistical or machine learning model is doing. In every project, it is essential to do an analysis of the machine learning algorithms and models. When it comes to evaluating a model, there are a few different kinds of assessment metrics that may be employed. There are a variety of them, some of which include classification accuracy, logarithmic loss, the confusion matrix, and others. This is a regression problem and I used following evaluation metrices to evaluate my models.

- R2 Score
- Mean Absolute Error
- Root Mean Squared Error

3.6.1. R2 Score

In statistics, the coefficient of determination, also known as R2 or r2 and pronounced "R squared," is the proportion of the variation in the dependent variable that can be predicted based on the independent variable. This amount is suggested by the letter R2 or r2 and penned as "R squared" (s).

On the basis of other relevant data, it is a statistic that is employed in the context of statistical models, the primary objective of which is either the prediction of future outcomes or the testing of hypotheses. It gives a measure of how effectively observed results are replicated by the model, and it does so based on the fraction of total variance in outcomes that can be attributed to the model's explanations(T. K. Landauer, D. Laham, and P. W. Foltz, 2000).

There are various different definitions of R2, and only few of them are equal to one another. The case of simple linear regression, in which r2 is utilized rather than R2, is an example of one of these classes of situations. When there is no other predictor variable than an intercept, r2 is simply the square of the sample correlation coefficient (also known as r) between the outcomes thatwere seen and the predictor values that were observed. R2 is the square of the coefficient of multiple correlation, which is calculated when extra regressors are added in the analysis. In each of these examples, the usual range for the coefficient of determination is somewhere between 0 and 1.

Depending on the particular definition that is applied, there are situations in which the computational definition of R2 can provide values that are negative. This can happen if the predictions that are being compared to the relevant outcomes have not been produced via a model-fitting technique utilizing those data. In other words, the predictions are being made without using those data. It is possible for R2 to be negative despite the fact that a model-fitting technique has been carried out. One example of this is when linear regression is carried out without the inclusion of an intercept.

3.6.2. Mean Absolute Error

In the field of statistics, the term "mean absolute error" (MAE) refers to a method for determining the degree of error that exists between paired data that describe the same phenomena. Comparisons of initial time with future time, one technique of measurement versus another technique of measurement, and one technique of measurement versus an alternate technique of measurement are all examples of Y versus X. The mean absolute error (MAE) is determined by dividing the total number of absolute errors by the total number of samples(Chai, T. and Draxler, R.R., 2014).

$$MEA = \frac{\sum_{i=1}^{n} | \, y_i - x_i |}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

It is important to keep in mind that different formulations could use relative frequencies as weight factors. The scale that is being used to measure the data is also used for the mean absolute error. Because this is what's known as a scale-dependent accuracy measure, it can't be used to compare series that utilize different scales because it depends on the scale itself. In time series analysis, the mean absolute error is a frequent way to quantify the accuracy of forecasts, occasionally leading to misunderstanding with the more traditional definition of mean absolute deviation]. The similar level of misunderstanding can be found more broadly(Wang, Y., and Liu, H., 2019).

3.6.3. Root Mean Squared Error

The root-mean-square deviation (RMSD), also known as the root-mean-square error (RMSE), is a measurement that is widely used to determine the discrepancies between values (sample or population values) that are predicted by a model or an estimator and the values that are actually observed. The root means square deviation, often known as the RMSD, is calculated by taking the square root of the second sample moment of the differences in values that were anticipated and those that were observed, or calculating the quadratic mean of these differences (Willmott, C.J. and Matsuura, K., 2005). When the computations are carried out across the data sample that was used for estimate, these deviations are referred to as residuals. On the other hand, when they are computed outside of the sample, they are referred to as errors (or prediction errors).

4. Data

This chapter contains information of available dataset, and the dataset that we will be utilizing for this project has also been discussed. After that, discussion on the data analysis that is performed on our dataset, in addition to the data preparation phases, all of which are described in greater detail in this chapter. Let's begin with the information of available datasets(Liu, J., Xu, Y. and Zhu, Y., 2019)

4.1. Available Dataset

In the fields of data analytics, machine learning, and artificial intelligence, the data itself is the most crucial component. Considering its importance, all of our contemporary research and automation efforts are for naught. The acquisition of as much granular information as possible can cost large corporations a significant amount of money. We decided not to collect any kind of data in this research instead we choose to use an open-source dataset for this research. There are many datasets available for AES, some information about them is given below (zesch, 2021).

Table 1 Open-Source Dataset

Name	Language	Source/Participant	Link	Task	Prompts
ASAP-DE	German	Crowd workers (Unclear)	Click	Answers of short question (Biology)	3
ASAP-SAS	English	High School Students (Grade 10)	Click	Short Answers (Biology)	10
ASAP-AES	English	High School student (Grade 7 to 10)	Click	Essays	10
SRA Beetle	English	Students (Native)	Click	Short Answers (Science)	56
AR-ASAG	Arabic	University Students	Click	Short Answers (Cybercrime)	48
SWELL	Swedish	Language Learners	Click	Essay	2

Data

COPLE-2	Portuguese	Language Learners	Click	Essay	Multiple

Since our research problems are related to essays so, we are using an open-source essays dataset name ASAP-AES in this thesis that was having essays content with manual score assigned by teachers. The essays were written by 7 to 10th grade students and it was having 10 prompts.

4.2. Thesis Data

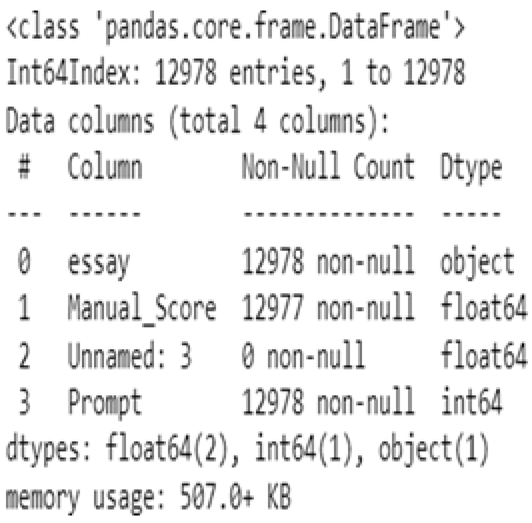
In this thesis I used the essay score prediction data to predict the scores of students in different essays. This dataset contains essays in English language that were written by students as answers to some queries. The length of the responses to the selected essays ranges from around 150 to 550 words on average. While some of the essays do not rely on any other sources for their material, others do. Students in grades ranging from seventh to tenth wrote each and every one of the replies. All of the essays were read, marked by hand, and then given a second score. Each of the eight data subsets possesses its own individual set of distinguishing qualities. The purpose of the variability is to push your scoring engine to its absolute boundaries and see how it performs. The names of different entities that are used in essays has been replaced with CAP.

Table 2 Original Form of Data

Essay	Manual_Score	Unnamed: 3	Prompt
Dear local newspaper, I think effects...	8.0	nan	1
Dear @CAPS1, I believe that...	9.0	nan	1
Dear, @CAPS1 @CAPS2 @CAPS3 More and...	7.0	nan	1
Dear Local Newspaper, @CAPS1 I have...	10.0	nan	1
Dear @LOCATION1, I know having computers...	8.0	nan	1

4.2.1. Dataset Description

This dataset has 12978 rows and 4 attributes named essay, manual_score, unnamed and prompt. Attribute ‘essay’ is object type and prompt are of integer type. Other two are float data type.

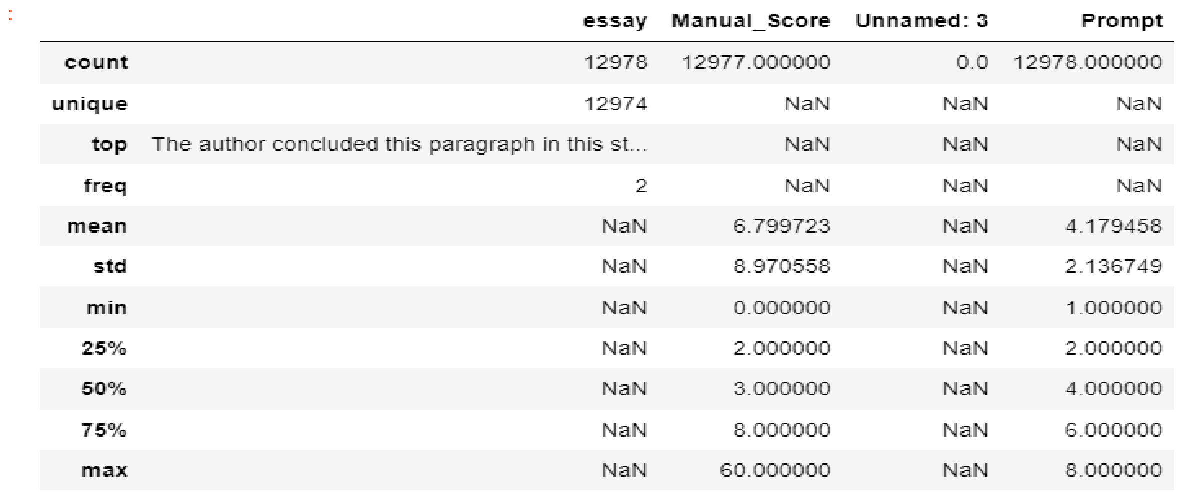


```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12978 entries, 1 to 12978
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   essay         12978 non-null  object
 1   Manual_Score  12977 non-null  float64
 2   Unnamed: 3    0 non-null      float64
 3   Prompt        12978 non-null  int64
dtypes: float64(2), int64(1), object(1)
memory usage: 507.0+ KB
```

Figure 8 Data Description

4.2.2. Statistical Description

All of the numerical aspects of the data are displayed by the statistical distribution of the data. The figure that follows reveals to us that the essay column contains 12974 different values in its entirety. It can be seen from the manual score that fifty percent of the pupils have taken three marks out of the possible six. 75 percent of the pupils have received an 8, which indicates that the class has a strong overall average.



:

	essay	Manual_Score	Unnamed: 3	Prompt
count	12978	12977.000000	0.0	12978.000000
unique	12974	NaN	NaN	NaN
top	The author concluded this paragraph in this st...	NaN	NaN	NaN
freq	2	NaN	NaN	NaN
mean	NaN	6.799723	NaN	4.179458
std	NaN	8.970558	NaN	2.136749
min	NaN	0.000000	NaN	1.000000
25%	NaN	2.000000	NaN	2.000000
50%	NaN	3.000000	NaN	4.000000
75%	NaN	8.000000	NaN	6.000000
max	NaN	60.000000	NaN	8.000000

Figure 9 Statistical description of data

4.2.3. Graphical Representation of Scores & Prompt

A better and more precise understanding of the facts can be achieved through the use of graphical representation. We have developed a function that first determines which values from the characteristics are unique and then counts each of those values. For the purpose of displaying values along with percentages, we have decided to use a bar plot. The bar plots that may be found below depict the many values that can be assigned to the manual score.

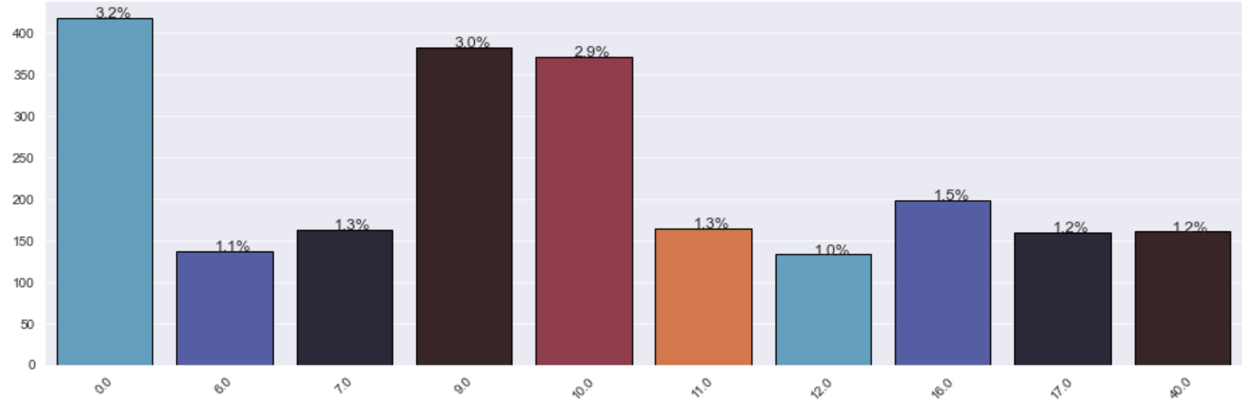


Figure 10 manual score representation

This is another figure for the target feature representation of prompt. According to figure, 8 has the least occurrence in the attribute.

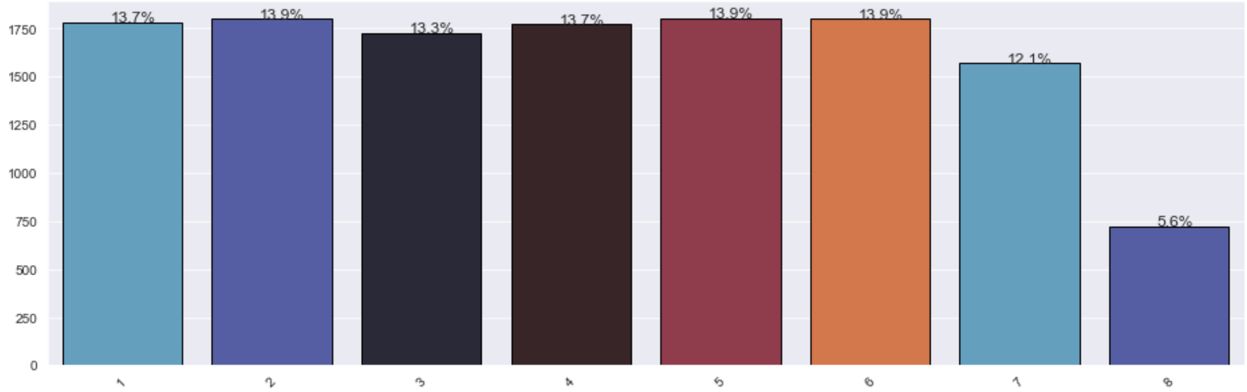


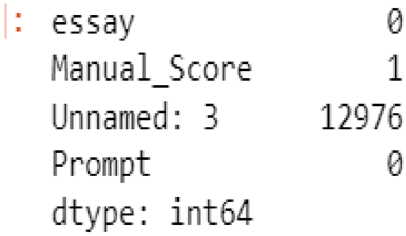
Figure 11 Prompt Representation

4.3. Data Processing

In this section, we will perform different procedures for data processing, It’s significant preliminary phase in NLP. The outcome of particular problem depends on proper data because the algorithms learn from the data and it is also important not to have negative effects on the performance of end result.

4.3.1. Handling Null Values

The empty cells in a column that have been left blank as a result of improper data handling or entering are referred to as having null values. We made use of the isnan() function, which iterates through all of the cells and returns the cells that are empty. The sum() function will then take the count of all of those cells and provide a single count value that represents the cells that are missing. It is less useful in the dataset because there is one missing value in the manual score column, and there are 12976 missing values in the prompt column, as seen in the following figure.

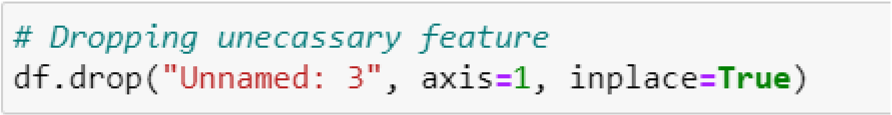


```
: essay               0
  Manual_Score        1
  Unnamed: 3      12976
  Prompt              0
  dtype: int64
```

Figure 12 Column wise null values

4.3.2. Handling Missing Values

Missing or incorrect data influences the model results a lot. Therefore, we have to handle these values before passing data to model. In above figure we have seen the unnamed column has almost all the cells empty that makes it less significant in essay score prediction method. That’s why we have deleted the whole column by using the drop method.



```
# Dropping unecassary feature
df.drop("Unnamed: 3", axis=1, inplace=True)
```

Figure 13 Dropping the irrelevant data

4.3.3. Cleaning Textual Data

As the main data in our dataset is the content of the essay that is in English language. We cannot feed this textual data directly to machine learning models for score prediction. First, we have to clean it so that we can have only useful words in our essays. For that I wrote a function to clean the data it performed following things to text.

- Removed Punctuations
- Lowered the text
- Lemmatized the words

Lemmatization is basically the process of extracting base words. For example, if we have multiple forms of a word like "teaches", "teaching" then it will convert to its base words that is "teach".

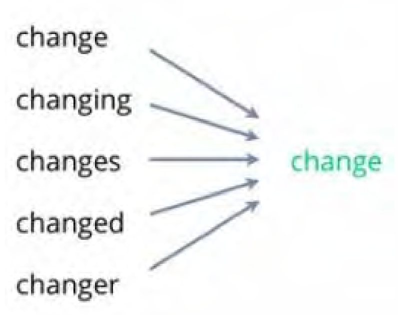


Figure 14 Lemmatization

So, I used lemmatizer from nltk library and applied it on each of the word in the each of the essay.

4.3.4. Stop Words

Stop words are a set of often used terms in any language. For example, in English, “the”, “and”, and “is” would easily qualify as stop words. In NLP and text mining applications, stop words are used to delete unnecessary words, allowing programs to focus on the important words instead. In below figure, we have represented the data before removing and after removing the stop words. Column clean text is the data after handling stop words.

Table 3 Clean Text

Essay	Manual_Score	Prompt	clean_Text
Dear local newspaper, I think effects computers have on people...	8.0	1	dear local newspaper think effect computer people great learning skill...
Dear @CAPS1 @CAPS2, I believe that using computers will benefit...	9.0	1	dear cap cap believe using computer benefit u many way...
Dear, @CAPS1 @CAPS2 @CAPS3 More and more people use computers,...	7.0	1	dear cap cap cap people use computer everyone agrees benefit...
Dear Local Newspaper, @CAPS1 I have found that many experts...	10.0	1	dear local newspaper cap found many expert say computer benifit...
Dear @LOCATION1, I know having computers has a positive effect...	8.0	1	dear location know computer positive effect people computer connect

Data

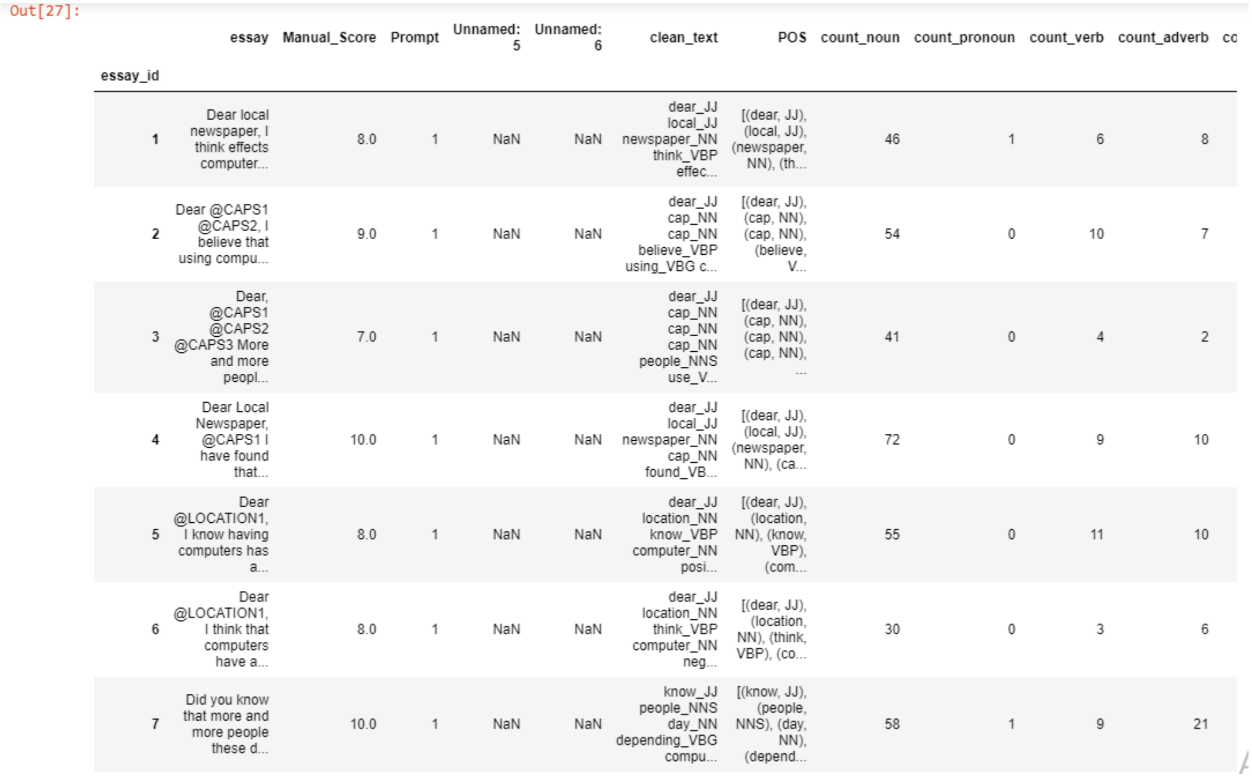
			family...

4.3.5. Identifying POS

Part of speech tagging, often known as POS tagging or POST, is the process of categorizing words according to their respective parts of speech and assigning labels to them in accordance with those parts of speech. As a result, the collection of labels or tags is referred to as a tag set. Each tagger includes a tag() method that can accept a list of tokens (often a list of words produced by a word tokenizer), where each token represents a single word. Using the tag() function will yield a list of tagged tokens, which is a tuple of (word, tag). In addition to being an essential component of learning the grammar of any language, becoming familiar with the various parts of speech is also an important step in the process of text preprocessing for natural language processing (NLP). As is well known, the goal of NLP is to teach a computer to communicate effectively with either a human or another computer. Because of this, it is essential for a machine to comprehend the different parts of speech.

After the tagging, I concatenated the grammatical patterns with their respective words. For instance, in Figure 14 below, in the first row "JJ" (noun) is concatenated with "dear". The purpose is this to apply vectorization method on concatenated output.

Data



Out[27]:

essay_id	essay	Manual_Score	Prompt	Unnamed: 5	Unnamed: 6	clean_text	POS	count_noun	count_pronoun	count_verb	count_adverb	co
1	Dear local newspaper, I think effects computer...	8.0	1	NaN	NaN	dear_JJ local_JJ newspaper_NN think_VBP effec...	[(dear, JJ), (local, JJ), (newspaper, NN), (th...	46	1	6	8	
2	Dear @CAPS1 @CAPS2, I believe that using compu...	9.0	1	NaN	NaN	dear_JJ cap_NN cap_NN believe_VBP using_VBG c...	[(dear, JJ), (cap, NN), (cap, NN), (believe, V...	54	0	10	7	
3	Dear, @CAPS1 @CAPS2 @CAPS3 More and more peopl...	7.0	1	NaN	NaN	dear_JJ cap_NN cap_NN cap_NN people_NNS use_V...	[(dear, JJ), (cap, NN), (cap, NN), (cap, NN), ...	41	0	4	2	
4	Dear Local Newspaper, @CAPS1 I have found that...	10.0	1	NaN	NaN	dear_JJ local_JJ newspaper_NN cap_NN found_VB...	[(dear, JJ), (local, JJ), (newspaper, NN), (ca...	72	0	9	10	
5	Dear @LOCATION1, I know having computers has a...	8.0	1	NaN	NaN	dear_JJ location_NN know_VBP computer_NN posi...	[(dear, JJ), (location, NN), (know, VBP), (com...	55	0	11	10	
6	Dear @LOCATION1, I think that computers have a...	8.0	1	NaN	NaN	dear_JJ location_NN think_VBP computer_NN neg...	[(dear, JJ), (location, NN), (think, VBP), (co...	30	0	3	6	
7	Did you know that more and more people these d...	10.0	1	NaN	NaN	know_JJ people_NNS day_NN depending_VBG compu...	[(know, JJ), (people, NNS), (day, NN), (depend...	58	1	9	21	

Figure 15 Part of Speech Count

The method takes each row one by one and split it according to parts of speech. After identifying the parts of speech, we have passed each pos to a function count_pos() that will return the number of occurrence of each part in each essay. This will tell the worth of an essay if they have made the use of specific parts of speech important for grading.

4.3.6. Sentiment Analysis

As shown is figure below, we can determine sentiment of each sentence. For this we calculate polarity of the sentence. Polarity values can vary from -1 to 1. If value is from -1 to 0 (not including 0) then sentence is called positive. Similarly, if the value is from 0 to 1 (not including) then it’s considered as negative. If value is 0 then sentence is neutral without conveying anything positive or negative.

4.3.7. Correlation of POS

A correlation matrix is a table that lists the correlation coefficients for each of the variables under consideration. The matrix illustrates the correlation between all of the many possible pair-wise combinations of values in the table. It is an effective tool for summing up enormous datasets, finding patterns in the data that has been provided, and visualizing those patterns. The illustration of such visuals is provided below. This demonstrates correlations between the parts of speech that are indicated to be important. The primary diagonal demonstrates that there is always a perfect correlation between each variable and itself. This diagonal can be identified by the line of 1.00s that runs from the top left to the bottom right. This matrix is symmetrical, with the same correlation presented above and below the main diagonal, with the difference being that those given above are a mirror image of those shown below. We can see that noun and adjective has the highest correlation of 0.88 followed by adverb and adjective.

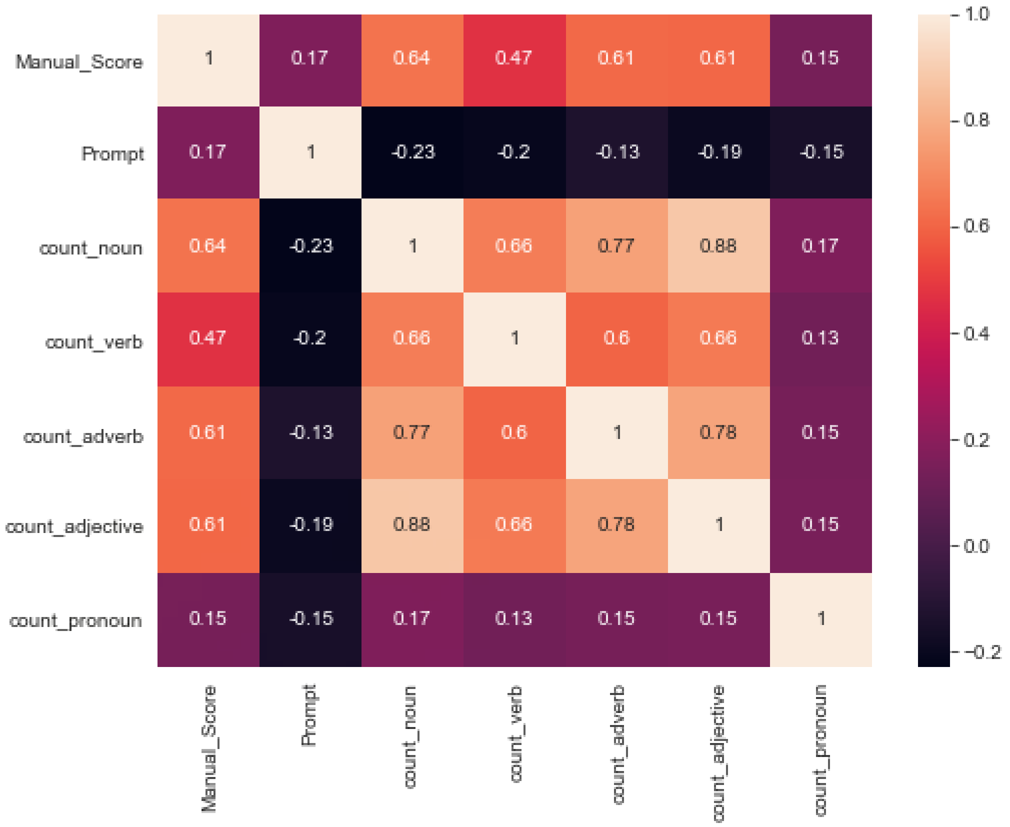


Figure 16 Correlation Matrix

From this graph we can see that part of speech has an effect on the essay grading. Table is giving below for the relationship between the score and different part of speech in terms of percentage.

Table 4 Correlation of POS with Score

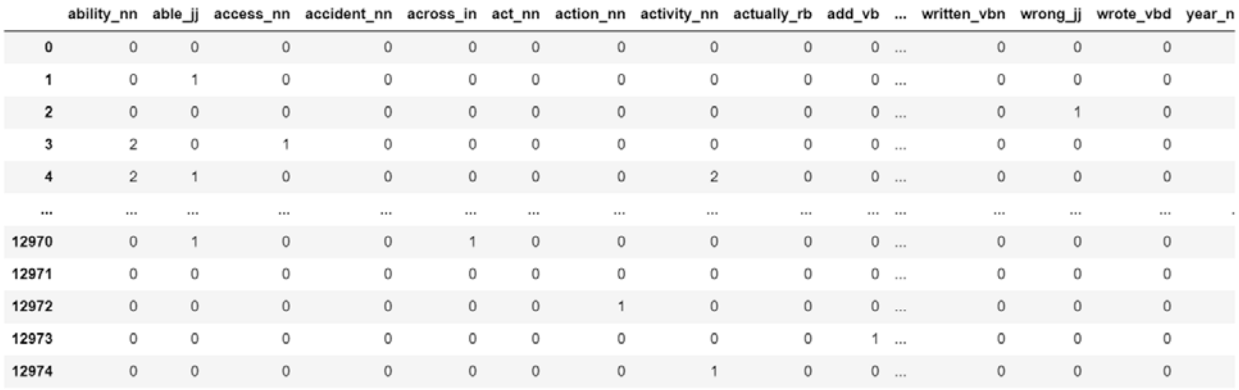
PART OF SPEECH	CORRELATION (%)
NOUN	64
VERB	47
ADVERB	61
ADJECTIVE	61
PRONOUN	15

As can be noticed from the Table 4 that the number of different part of speeches has a great impact on the essay score even after the content. So, these were the quantitative or structural measures that we want to merge with the content of essays. Then these two merged vectors will be used for making the prediction system.

5. Model Implementation

Modeling is the process of creating, training and evaluating a machine learning or deep learning model. In my thesis project, I implemented models, trained them on training data's output generated as illustrated in Section 4 and then evaluated their performance to see how they are performing.

5.1. Vectorization



	ability_nn	able_jj	access_nn	accident_nn	across_in	act_nn	action_nn	activity_nn	actually_rb	add_vb	...	written_vbn	wrong_jj	wrote_vbd	year_n
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
1	0	1	0	0	0	0	0	0	0	0	...	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	1	0	
3	2	0	1	0	0	0	0	0	0	0	...	0	0	0	
4	2	1	0	0	0	0	0	2	0	0	...	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
12970	0	1	0	0	1	0	0	0	0	0	...	0	0	0	
12971	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
12972	0	0	0	0	0	0	1	0	0	0	...	0	0	0	
12973	0	0	0	0	0	0	0	0	0	1	...	0	0	0	
12974	0	0	0	0	0	0	0	1	0	0	...	0	0	0	

Figure 17 Vectorization

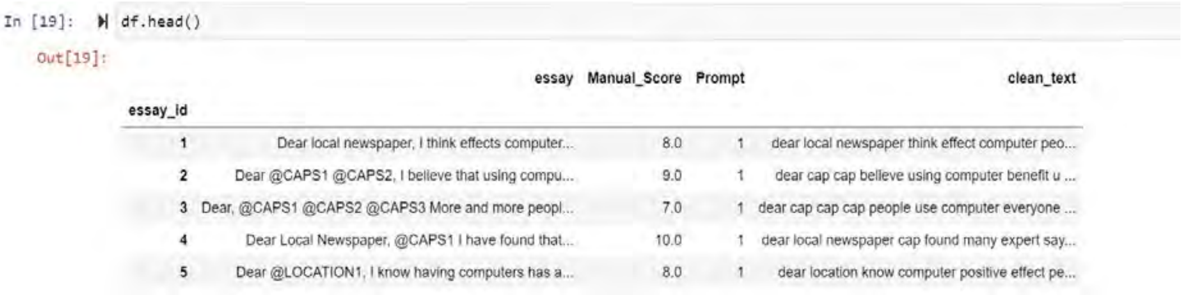
These are the vectors for the content of the essays. I merged these content vectors with the quantitative measures that we have calculated above

5.2. Building Models

In the subsequent stage, we proceeded to train the models by first separating the data into a train set and a test set. I used five machine learning models and one deep learning model with two variations.

We use Automated Student Assessment Prize (ASAP) dataset to train our models specified above. The dataset contains 12978 essays in total, each essay containing 150 to 550 words with manual scoring. We initialize the training by preprocessing the data with methods below,

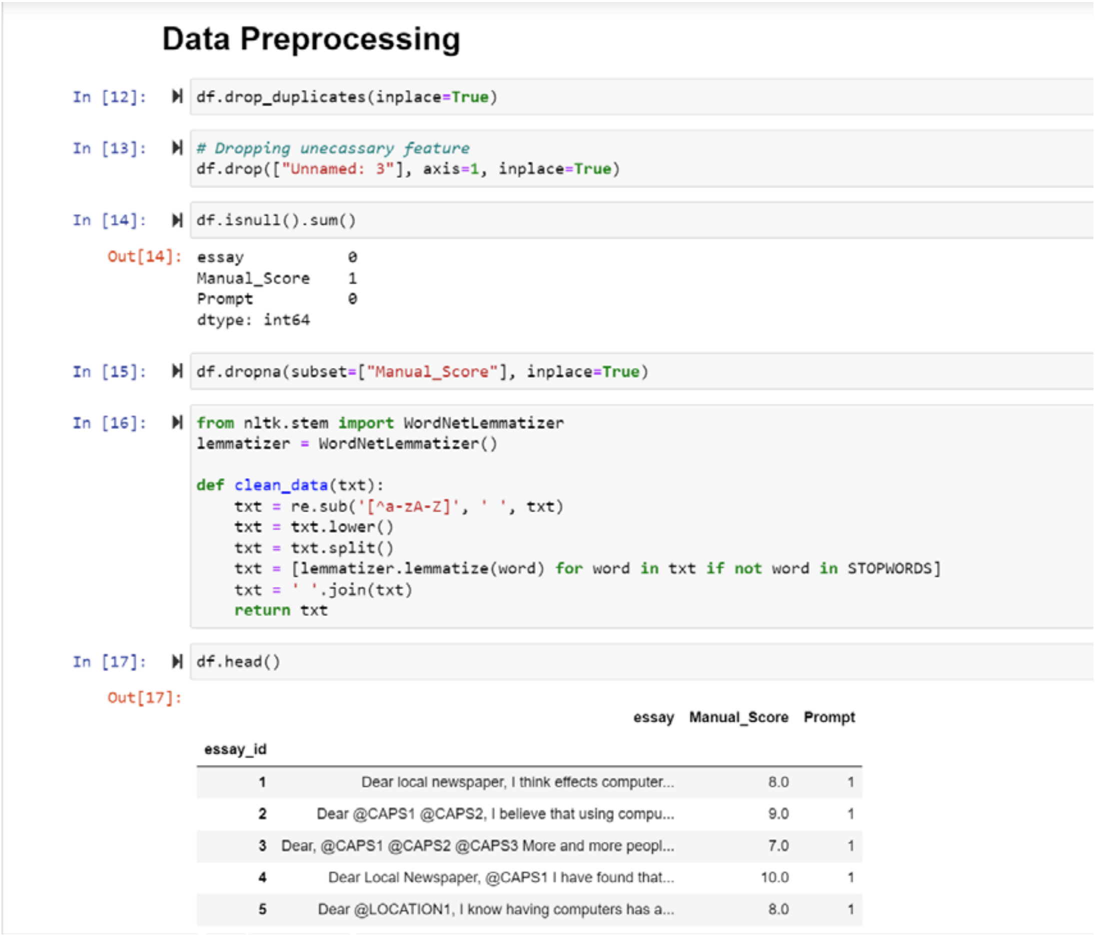
- Removing duplicate essays
- Dropping unnecessary columns
- Lemmatize the words to group similar words
- Clean essay texts by removing special characters



```
In [19]:  df.head()
Out[19]:
```

essay_id	essay	Manual_Score	Prompt	clean_text
1	Dear local newspaper, I think effects computer...	8.0	1	dear local newspaper think effect computer peo...
2	Dear @CAPS1 @CAPS2, I believe that using compu...	9.0	1	dear cap cap believe using computer benefit u ...
3	Dear, @CAPS1 @CAPS2 @CAPS3 More and more peopl...	7.0	1	dear cap cap cap people use computer everyone ...
4	Dear Local Newspaper, @CAPS1 I have found that...	10.0	1	dear local newspaper cap found many expert say...
5	Dear @LOCATION1, I know having computers has a...	8.0	1	dear location know computer positive effect pe...

Figure 18 Result of clean essay text



Data Preprocessing

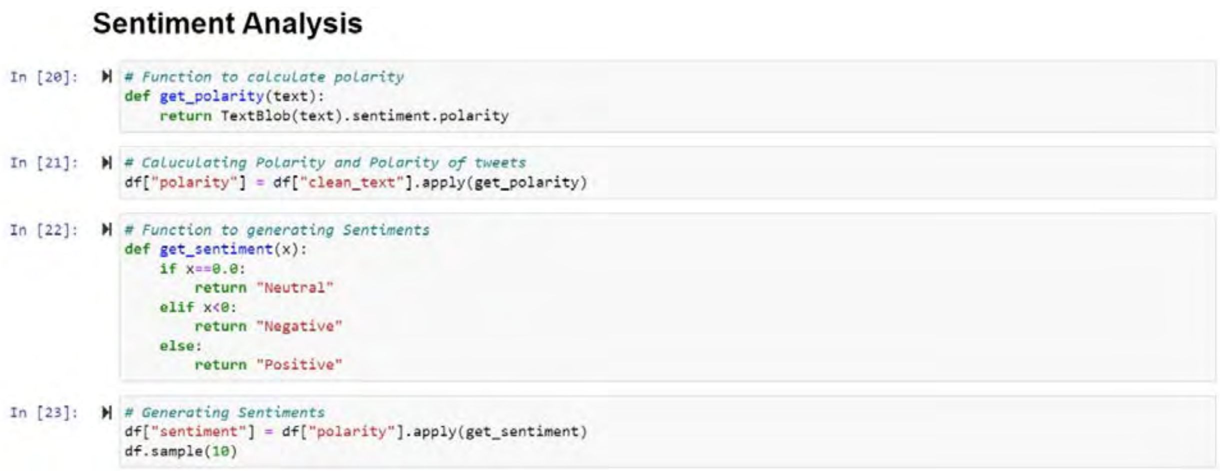
```
In [12]:  df.drop_duplicates(inplace=True)

In [13]:  # Dropping unecassary feature
          df.drop(["Unnamed: 3"], axis=1, inplace=True)

In [14]:  df.isnull().sum()

Out[14]:  essay           0
          Manual_Score    1
          Prompt          0
          dtype: int64

In [15]:  df.dropna(subset=["Manual_Score"], inplace=True)

In [16]:  from nltk.stem import WordNetLemmatizer
          lemmatizer = WordNetLemmatizer()

          def clean_data(txt):
              txt = re.sub('[^a-zA-Z]', ' ', txt)
              txt = txt.lower()
              txt = txt.split()
              txt = [lemmatizer.lemmatize(word) for word in txt if not word in STOPWORDS]
              txt = ' '.join(txt)
              return txt

In [17]:  df.head()
Out[17]:
```

essay_id	essay	Manual_Score	Prompt
1	Dear local newspaper, I think effects computer...	8.0	1
2	Dear @CAPS1 @CAPS2, I believe that using compu...	9.0	1
3	Dear, @CAPS1 @CAPS2 @CAPS3 More and more peopl...	7.0	1
4	Dear Local Newspaper, @CAPS1 I have found that...	10.0	1
5	Dear @LOCATION1, I know having computers has a...	8.0	1

Figure 19 Data preprocessing script

5.3. Sentiment Analysis

For sentiment analysis we determine the polarity of the texts by parsing the string and extracting the tone of the essay to check whether it is positive, negative, or neutral.

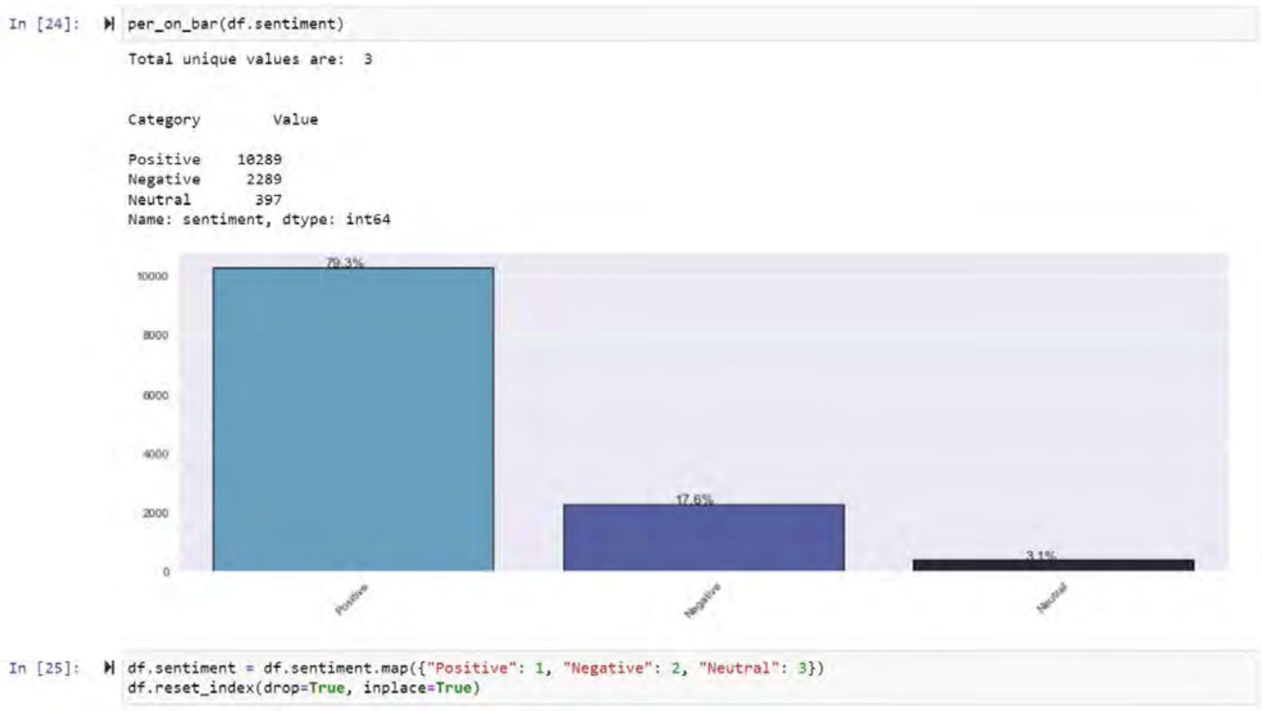


Sentiment Analysis

```
In [20]:  # Function to calculate polarity
          def get_polarity(text):
              return TextBlob(text).sentiment.polarity

In [21]:  # Caluculating Polarity and Polarity of tweets
          df["polarity"] = df["clean_text"].apply(get_polarity)

In [22]:  # Function to generating Sentiments
          def get_sentiment(x):
              if x==0.0:
                  return "Neutral"
              elif x<0:
                  return "Negative"
              else:
                  return "Positive"

In [23]:  # Generating Sentiments
          df["sentiment"] = df["polarity"].apply(get_sentiment)
          df.sample(10)
```

Figure 20 Sentiment Analysis polarity detection

Visualization of sentiments applied in the dataset:



```
In [24]:  per_on_bar(df.sentiment)

          Total unique values are:  3


          Category        Value

          Positive    10289
          Negative     2289
          Neutral       397
          Name: sentiment, dtype: int64
```

```
In [25]:  df.sentiment = df.sentiment.map({"Positive": 1, "Negative": 2, "Neutral": 3})
          df.reset_index(drop=True, inplace=True)
```

Figure 21 Visualization of sentiment analysis

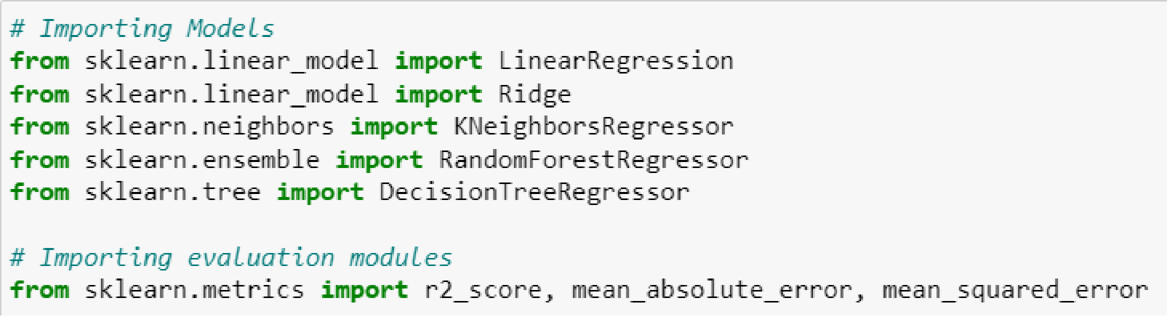
5.4. Machine Learning Models:

We begin to train and test our models below with the preprocessed data with Manual_Score column as the output row to fit the model. Below figure elucidates the code script that shows the regressors that are appended to the main model list,

These models are…

- Linear Regression
    - Ordinary least squares Linear Regression
- Ridge Regression
    - This model resolves a regression model where the regularization is provided by the l2-norm and the loss function is the linear least squares function.
- K Neighbour Regression
    - K-nearest neighbor-based regression. The target is predicted using local interpolation of the targets linked to the training set's closest neighbors.
- Random Forest Regression
    - A random forest is a meta estimator that employs averaging to increase predicted accuracy and reduce overfitting after fitting numerous classification decision trees to different dataset subsamples.
- Decision Tree Regression
    - Decision tree regressor

Here I imported the machine learning models.

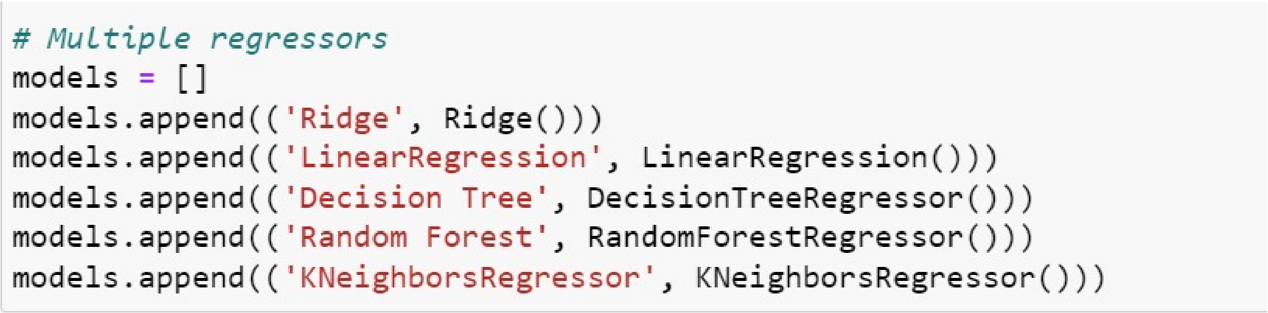


```
# Importing Models
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor

# Importing evaluation modules
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

Figure 22 Models and evaluation metrics

I have also imported different evaluation metrics from sklearn’s metric such as r2 score, mean squared error and mean absolute error to check the performance of each model on the train and test data.

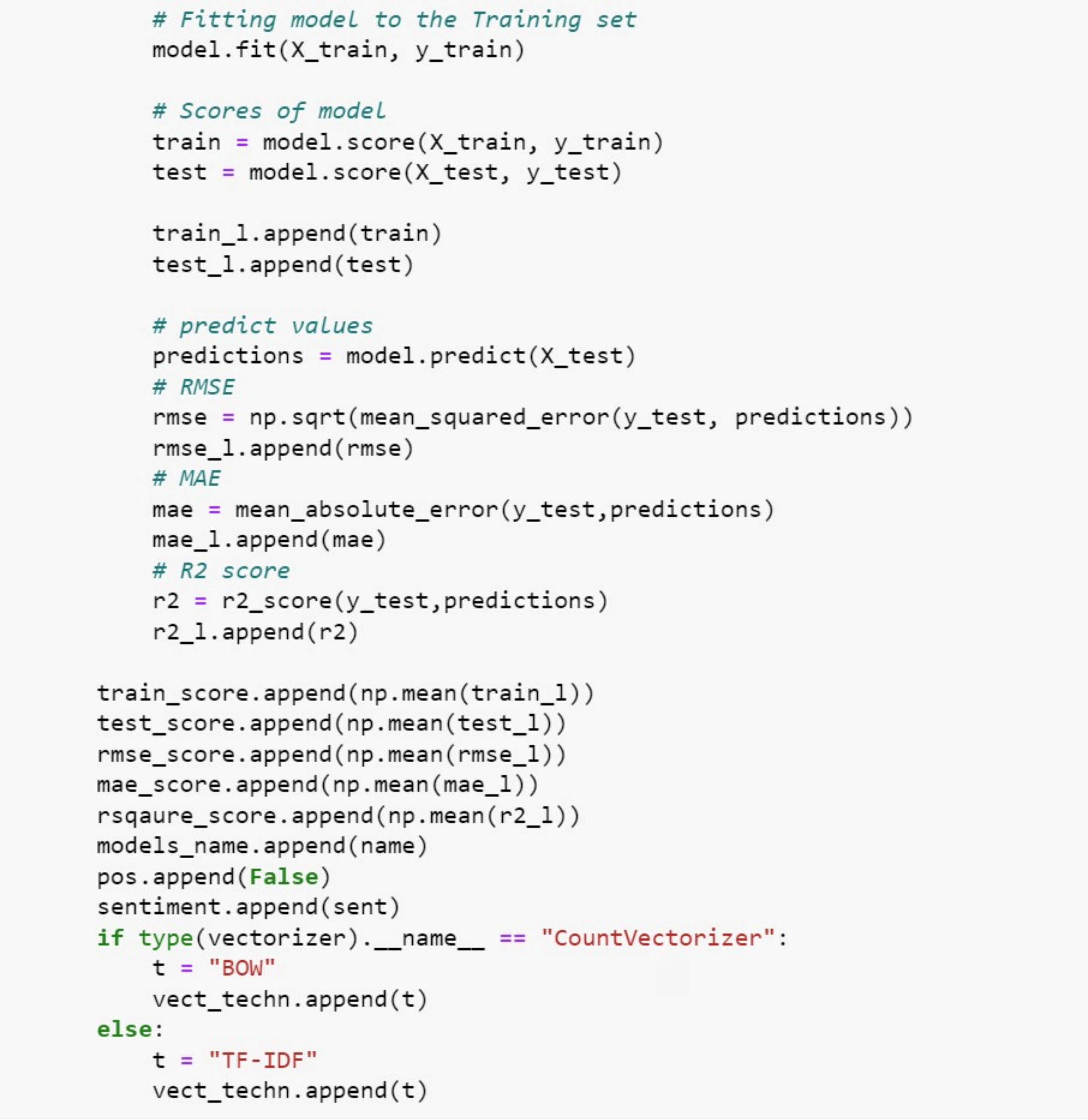
Model Implementation



```
# Multiple regressors
models = []
models.append(('Ridge', Ridge()))
models.append(('LinearRegression', LinearRegression()))
models.append(('Decision Tree', DecisionTreeRegressor()))
models.append(('Random Forest', RandomForestRegressor()))
models.append(('KNeighborsRegressor', KNeighborsRegressor()))
```

Figure 23 Regressor model

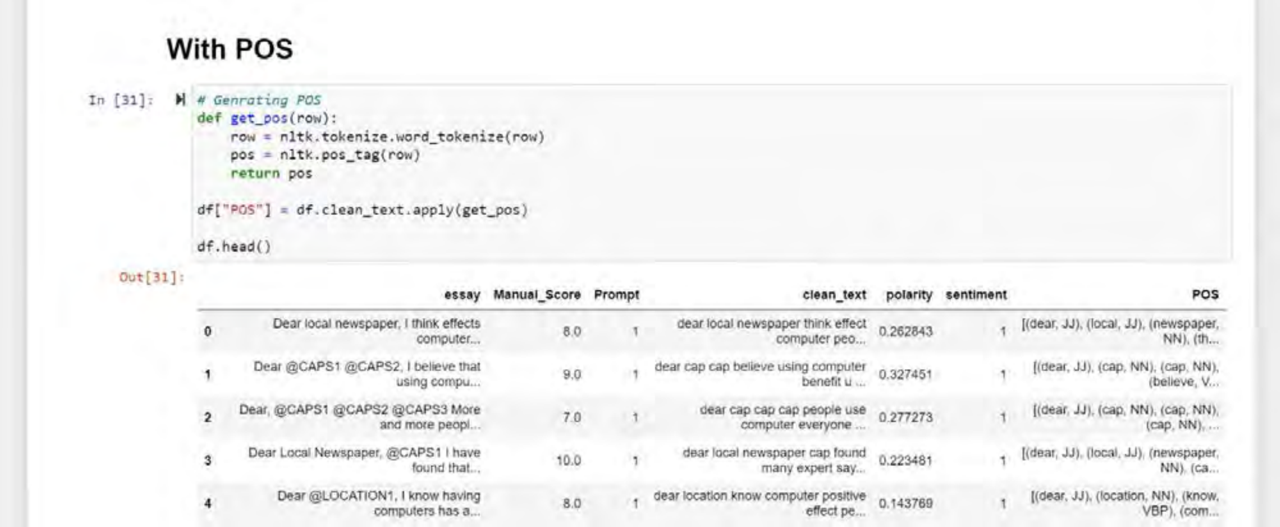
I fit and test the model using the test data as shown below script,



```
        # Fitting model to the Training set
        model.fit(X_train, y_train)

        # Scores of model
        train = model.score(X_train, y_train)
        test = model.score(X_test, y_test)

        train_l.append(train)
        test_l.append(test)

        # predict values
        predictions = model.predict(X_test)
        # RMSE
        rmse = np.sqrt(mean_squared_error(y_test, predictions))
        rmse_l.append(rmse)
        # MAE
        mae = mean_absolute_error(y_test,predictions)
        mae_l.append(mae)
        # R2 score
        r2 = r2_score(y_test,predictions)
        r2_l.append(r2)

    train_score.append(np.mean(train_l))
    test_score.append(np.mean(test_l))
    rmse_score.append(np.mean(rmse_l))
    mae_score.append(np.mean(mae_l))
    rsqaure_score.append(np.mean(r2_l))
    models_name.append(name)
    pos.append(False)
    sentiment.append(sent)
    if type(vectorizer).__name__ == "CountVectorizer":
        t = "BOW"
        vect_techn.append(t)
    else:
        t = "TF-IDF"
        vect_techn.append(t)
```

Figure 24 Building Model

In addition to that, we add two of the vectorization methods for syntactic analysis bag of words and other with TF-IDF. We train the models without using part of speech features for comparison. We then add the parts of speech to the word list that we used for training the model as below and run the training again inclusive of sentiments,



With POS

```
In [31]:  # Genrating POS
          def get_pos(row):
              row = nltk.tokenize.word_tokenize(row)
              pos = nltk.pos_tag(row)
              return pos

          df["POS"] = df.clean_text.apply(get_pos)

          df.head()
```

Out[31]:

	essay	Manual_Score	Prompt	clean_text	polarity	sentiment	POS
0	Dear local newspaper, I think effects computer...	8.0	1	dear local newspaper think effect computer peo...	0.262843	1	[(dear, JJ), (local, JJ), (newspaper, NN), (th...
1	Dear @CAPS1 @CAPS2, I believe that using compu...	9.0	1	dear cap cap believe using computer benefit u ...	0.327451	1	[(dear, JJ), (cap, NN), (cap, NN), (believe, V...
2	Dear, @CAPS1 @CAPS2 @CAPS3 More and more peopl...	7.0	1	dear cap cap cap people use computer everyone ...	0.277273	1	[(dear, JJ), (cap, NN), (cap, NN), (cap, NN), ...
3	Dear Local Newspaper, @CAPS1 I have found that...	10.0	1	dear local newspaper cap found many expert say...	0.223481	1	[(dear, JJ), (local, JJ), (newspaper, NN), (ca...
4	Dear @LOCATION1, I know having computers has a...	8.0	1	dear location know computer positive effect pe...	0.143769	1	[(dear, JJ), (location, NN), (know, VBP), (com...

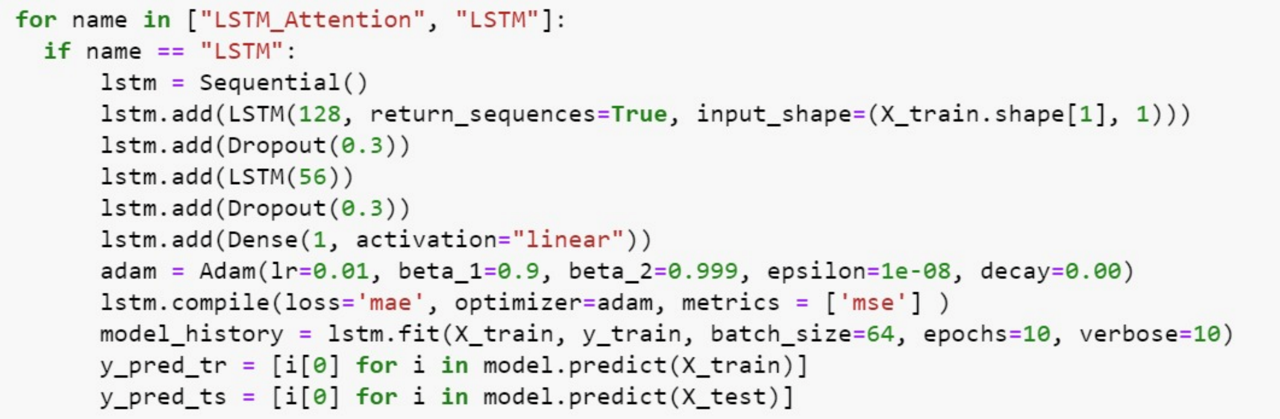
Figure 25 With Part of speech

To compare and contrast different approaches we exclude sentiment with parts of speech and train the model to measure the performance. We split the data into 80 percent training and 20 percent testing to verify the accuracy of the models and plot a graph accordingly.

We add the below Long Short-Term Memory layered deep learning models to improve the accuracy and reduce overfitting of the data,

- LSTM Regression
- LSTM Attention Regression

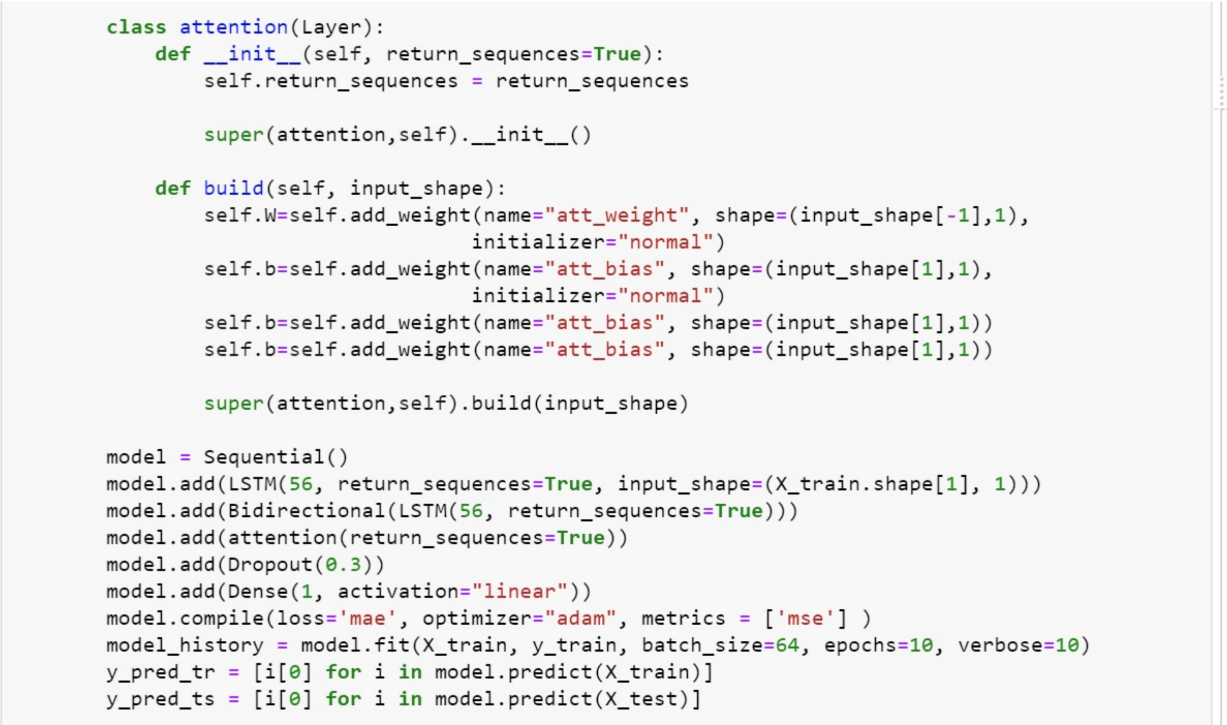
The below code implements the layers in the model,



```
for name in ["LSTM_Attention", "LSTM"]:
  if name == "LSTM":
      lstm = Sequential()
      lstm.add(LSTM(128, return_sequences=True, input_shape=(X_train.shape[1], 1)))
      lstm.add(Dropout(0.3))
      lstm.add(LSTM(56))
      lstm.add(Dropout(0.3))
      lstm.add(Dense(1, activation="linear"))
      adam = Adam(lr=0.01, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.00)
      lstm.compile(loss='mae', optimizer=adam, metrics = ['mse'] )
      model_history = lstm.fit(X_train, y_train, batch_size=64, epochs=10, verbose=10)
      y_pred_tr = [i[0] for i in model.predict(X_train)]
      y_pred_ts = [i[0] for i in model.predict(X_test)]
```

With learning rate 0.01 and epoch 10 as the initial training. Then we use the trained model to predict the test data.

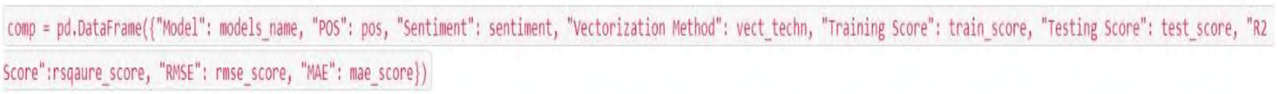
In the below code snippet, we build the LSTM model with attention,



```
class attention(Layer):
    def __init__(self, return_sequences=True):
        self.return_sequences = return_sequences

        super(attention,self).__init__()

    def build(self, input_shape):
        self.W=self.add_weight(name="att_weight", shape=(input_shape[-1],1),
                               initializer="normal")
        self.b=self.add_weight(name="att_bias", shape=(input_shape[1],1),
                               initializer="normal")
        self.b=self.add_weight(name="att_bias", shape=(input_shape[1],1))
        self.b=self.add_weight(name="att_bias", shape=(input_shape[1],1))

        super(attention,self).build(input_shape)

model = Sequential()
model.add(LSTM(56, return_sequences=True, input_shape=(X_train.shape[1], 1)))
model.add(Bidirectional(LSTM(56, return_sequences=True)))
model.add(attention(return_sequences=True))
model.add(Dropout(0.3))
model.add(Dense(1, activation="linear"))
model.compile(loss='mae', optimizer="adam", metrics = ['mse'] )
model_history = model.fit(X_train, y_train, batch_size=64, epochs=10, verbose=10)
y_pred_tr = [i[0] for i in model.predict(X_train)]
y_pred_ts = [i[0] for i in model.predict(X_test)]
```

Figure 26 Building LSTM model with attention

In the next section we present the comparison between the models we had implemented in this section and contrast the performance between them. We use the below code snippet for comparison between models,



```
comp = pd.DataFrame({"Model": models_name, "POS": pos, "Sentiment": sentiment, "Vectorization Method": vect_techn, "Training Score": train_score, "Testing Score": test_score, "R2
Score":rsqaure_score, "RMSE": rmse_score, "MAE": mae_score})
```

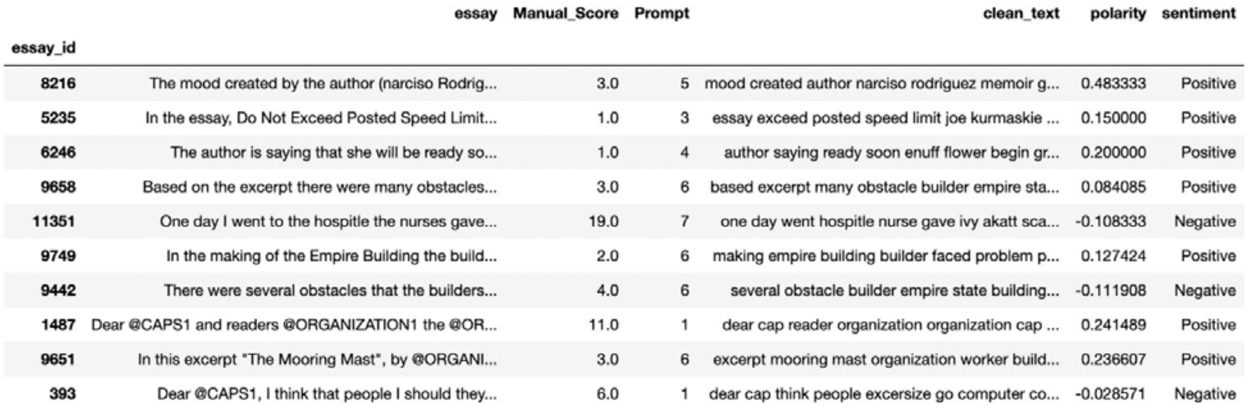
6. Results

Finally, we have achieved the results of all machine learning models with two vectorization methods for each model parallelly. Then the model has been evaluated using root mean squared error, mean absolute error and r2 score. The results of these models are explained below.

6.1. Evaluation Of Sentiment Analysis

From the below table we can see the polarity of each sentence and based on this value, we can see if the given sentence is positive negative or neutral. Here for this study, we have considered ideal threshold and thus sentence is considered negative if the polarity is between -1 and 0. Sentence is neutral for polarity 0 and it is positive if polarity is between 0 and 1.

Below we can see sentiment analysis of our data set. Polarity of sentence changes from 0.4833 to -0.111, this means some of the sentences are negative while other asserts positive sentiment.



	essay	Manual_Score	Prompt	clean_text	polarity	sentiment
essay_id						
8216	The mood created by the author (narciso Rodrig...	3.0	5	mood created author narciso rodriguez memoir g...	0.483333	Positive
5235	In the essay, Do Not Exceed Posted Speed Limit...	1.0	3	essay exceed posted speed limit joe kurmaskie ...	0.150000	Positive
6246	The author is saying that she will be ready so...	1.0	4	author saying ready soon enuff flower begin gr...	0.200000	Positive
9658	Based on the excerpt there were many obstacles...	3.0	6	based excerpt many obstacle builder empire sta...	0.084085	Positive
11351	One day I went to the hospitle the nurses gave...	19.0	7	one day went hospitle nurse gave ivy akatt sca...	-0.108333	Negative
9749	In the making of the Empire Building the build...	2.0	6	making empire building builder faced problem p...	0.127424	Positive
9442	There were several obstacles that the builders...	4.0	6	several obstacle builder empire state building...	-0.111908	Negative
1487	Dear @CAPS1 and readers @ORGANIZATION1 the @OR...	11.0	1	dear cap reader organization organization cap ...	0.241489	Positive
9651	In this excerpt "The Mooring Mast", by @ORGANI...	3.0	6	excerpt mooring mast organization worker build...	0.236607	Positive
393	Dear @CAPS1, I think that people I should they...	6.0	1	dear cap think people excersize go computer co...	-0.028571	Negative

Figure 27 Evaluation of Sentiment Analysis

6.2. Evaluation of ML models

For all these models I did cross validation and created 5 folds. Then for each fold model is trained and evaluated all the models one by one and their results are given below.

6.2.1. Evaluation of Ridge

For this model, parameters used are:

- alpha = 1.0 (Controls regulation strength by multiplying with L2 term.)
- solver = auto (scikit will attempt to used best of ‘svd, ‘cholesky, and ‘sparse_cg’, `lsqr`, `sag` and ` lbfgs`)

For ridge model the results are given below:

Table 5 Evaluation results of Ridge model

EVALUATION METRIC	SCORE
R2	0.891
MAE	1.881
RMSE	2.578

Comparison between actual and predicted values is given below in the form of graph.

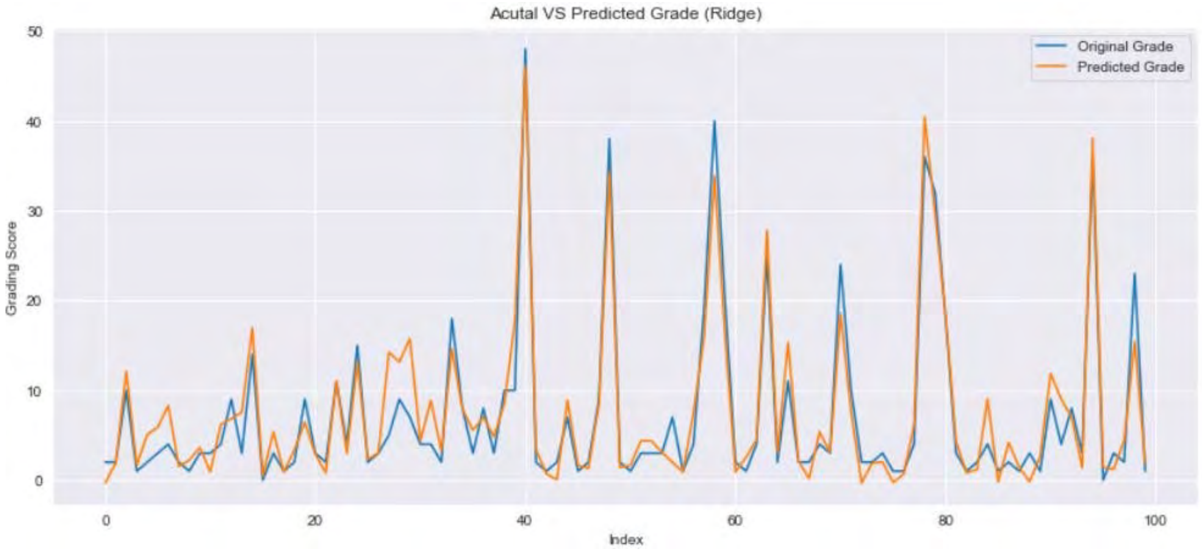


Figure 28 Actual VS Prediction (Ridge)

This model is performing as much well because this model is designed for linear dataset. But the nature of our data is not linear, and model is not able to correctly approach the actual grades.

6.2.2. Evaluation of Linear Regression

This model is exactly performing same to the previous model. Because both are of same structure. For this model the results are given below.

Table 6 : Evaluation results of Linear regression model

EVALUATION METRIC	SCORE
R2	0.891
MAE	2.92
RMSE	2.63

Comparison between actual and predicted values is given below in the form of graph.

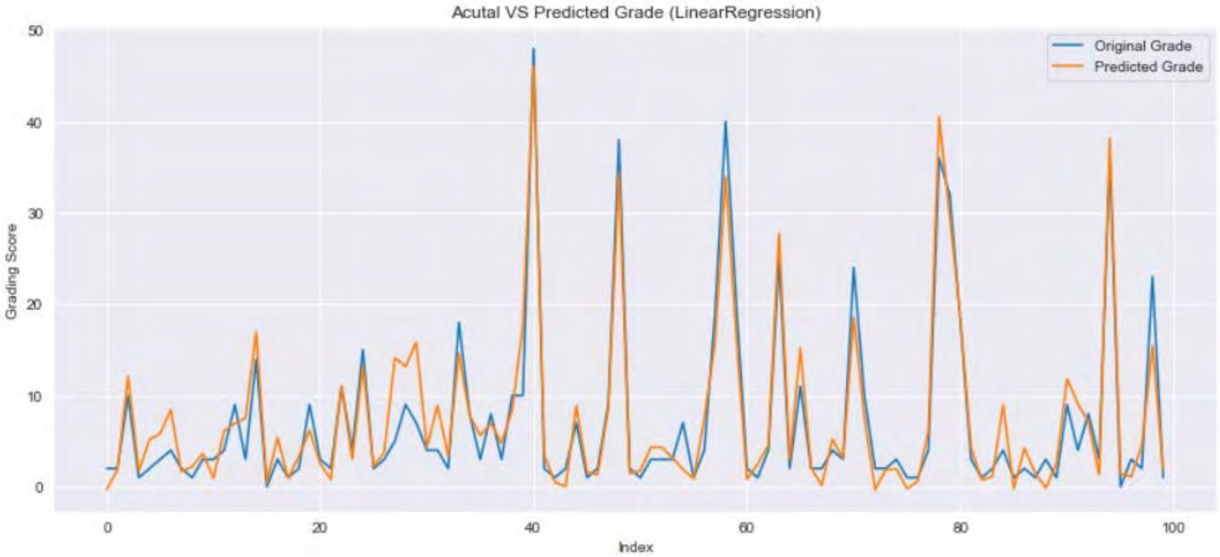


Figure 29 Actual VS Prediction (Linear Progression Tree)

6.2.3. Evaluation of Decision Tree

This model has different structure than the above two models that's the reason there is an improvement in the results of model.

For this model, parameters used are:

- splitter = "best" (can be either best or random)
- criterrion = squared_error (Used to measure quality of split)
- min_simple_split = 2 (minimum samples needed to split an internal node)

Table 7 Evaluation results of Decision Tree

EVALUATION METRIC	SCORE
R2	0.908
MAE	1.209
RMSE	2.198

Comparison between actual and predicted values is given below in the form of graph.

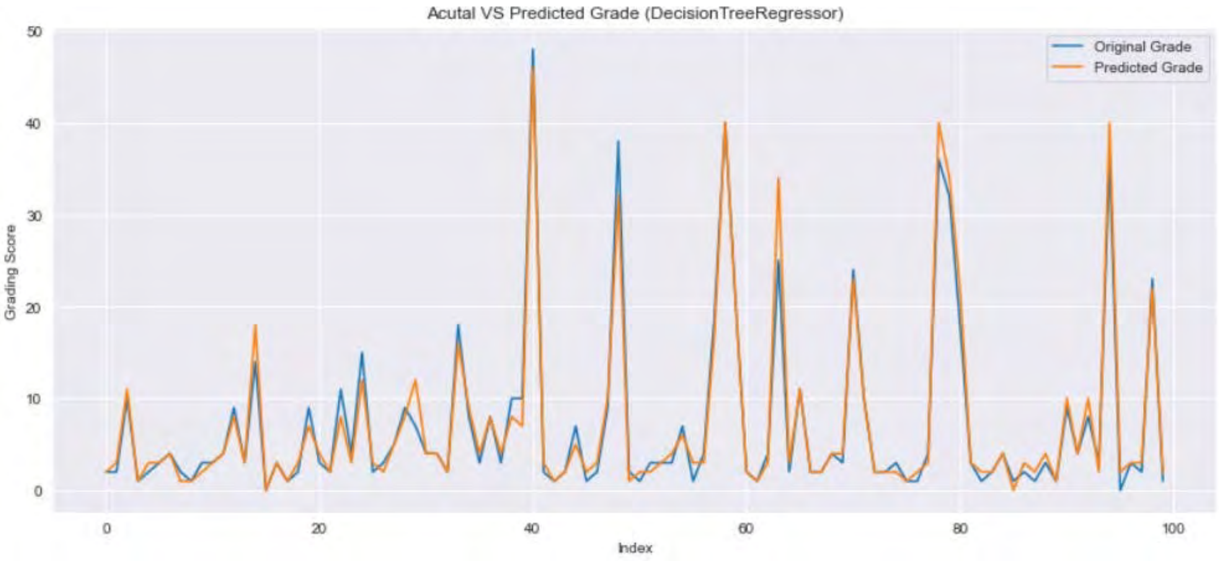


Figure 30 Actual VS Prediction (Decision Tree)

This model is performing much better than other two models because of its non-linear structure.

6.2.4. Evaluation of Random Forest

This model has a very complex but time-consuming structure and due to that it is very optimal when you want to get the higher results. I comprise many decision trees and make predictions as we have seen above in methodology part. And due to that the variance in the results is very low in this model and gives very close results.

For this model, parameters used are:

- n_estimators = 100 (This value specifies number of trees)

- criterrion = squared_error (Used to measure quality of split)
- min_simple_split = 2 (minimum samples needed to split an internal node)
- verbose = 0

For this model the results are given below.

Table 8 Evaluation results of Random Forest

EVALUATION METRIC	SCORE
R2	0.903
MAE	1.209
RMSE	2.198

Comparison between actual and predicted values is given below in the form of graph

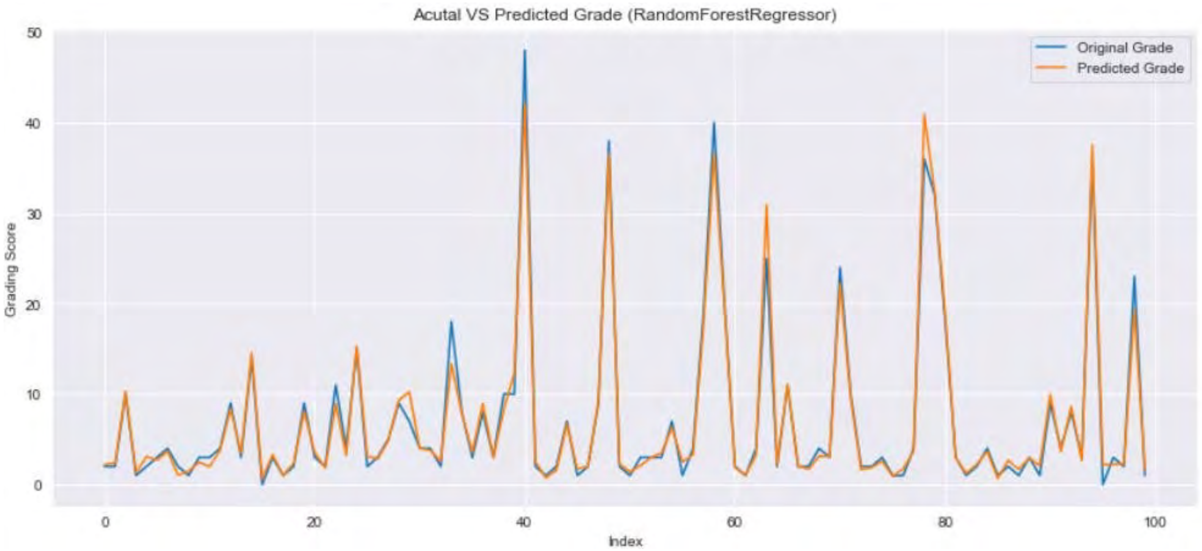


Figure 31 Actual VS Prediction (Random Forest)

6.2.5. Evaluation of KNN

This model is not designed for high amount of data. If the amount of the data and its dimension is low then this model will perform well otherwise it is not.

For this model, parameters used are:

- No. of neighbors = 5 (Weights are uniformly divided for each neighborhood)
- Algorithm = auto (scikit will attempt to used best of 'ball_tree', 'kd_tree', and

'kd_tree')

- Leaf Size = 30 (This parameter can affect the speed of query and construction as well as memory required for storing tree.
- P value = 2 (This is equivalent of using euclidean_distance (l2))

For this model the results are given below.

Table 9 Evaluation results of KNN

EVALUATION METRIC	SCORE
R2	0.765
MAE	2.53
RMSE	4.27

Comparison between actual and predicted values is given below in the form of graph.

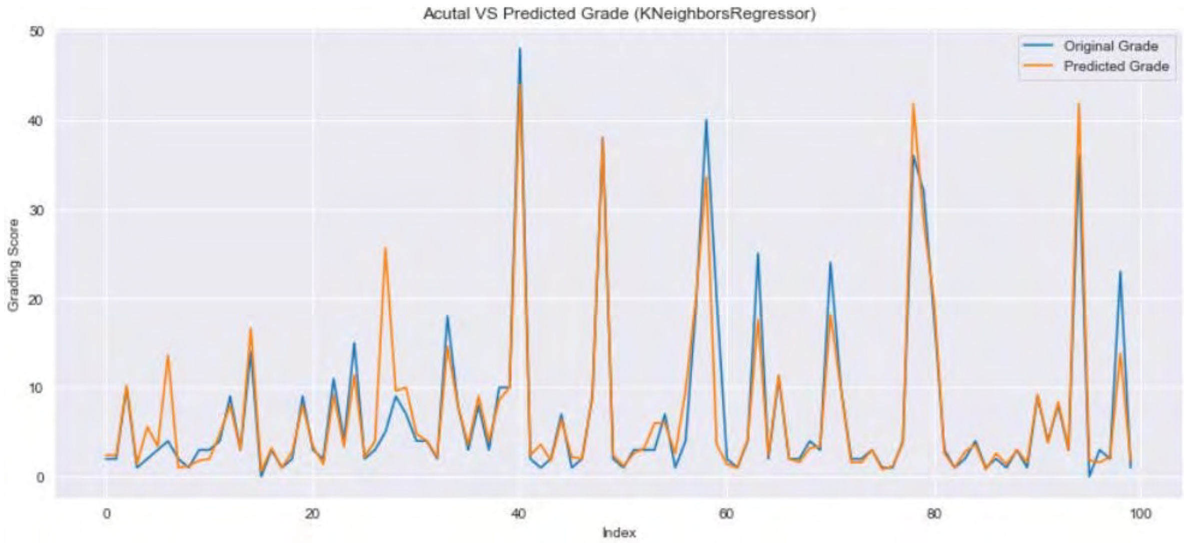


Figure 32 Actual VS Prediction (KNN)

6.3. Evaluation of Deep Learning Models

As we discussed earlier, I have used one deep learning model with two variations. One is simple LSTM model and the other is LSTM Attention model.

In this paper, bag of word and TFID vectorization techniques are used for building both the variation of LSTM model. Parameters used for bag of words are ngram_range = (1,1), max_features = 1000 and stop words = English. While for TF-IDF vectorization we have used max_features = 1000 and stop words = English.

6.3.1. Evaluation of LSTM Model

For this model results are:

Table 10 Evaluation of LSTM Model

EVALUATION METRIC	SCORE
R2	0.91
MAE	2.52
RMSE	3.36

6.3.2. Evaluation of LSTM Attention Model

For this model results are:

Table 11 Evaluation of LSTM Attention Model

EVALUATION METRIC	SCORE
R2	0.135
MAE	4.312
RMSE	8.35

6.4. Comparison of All Models

In this section, we have compared all the models with the following variations:

- Without POS, without Sentiment
- Without POS, with Sentiment
- With POS; without Sentiment
- With POS and Sentiment

6.4.1. Comparison Without POS, without Sentiment

The below Table 12 illustrates results of all the machine learning and deep learning models without POS, without Sentiment.

Table 12 Comparison without POS, without Sentiment

Model	Vectorization Method	Training Score	Testing Score	R2 Score	RMSE	MAE
Ridge	BOW	0.8825	0.8207	0.8207	3.79682	2.5090
LinearRegression	BOW	0.8832	0.8203	0.8203	3.80064	2.5128
Decision Tree	BOW	0.9952	0.8926	0.8926	3.54367	1.7082
Random Forest	BOW	0.9899	0.8903	0.8903	2.63280	1.3427
KNeighborsRegressor	BOW	0.8408	0.7400	0.7400	4.56874	2.2455
Ridge	TF-IDF	0.9023	0.8738	0.8738	3.18348	2.1700
LinearRegression	TF-IDF	0.9040	0.8665	0.8665	3.27224	2.2649
Decision Tree	TF-IDF	0.9927	0.8686	0.8686	3.81576	1.7797
Random Forest	TF-IDF	0.9888	0.8906	0.8906	2.74042	1.3825
KNeighborsRegressor	TF-IDF	0.8592	0.7010	0.7010	3.30779	1.6887
LSTM	TF-IDF	0.8964	0.8447	0.8447	3.34169	2.4399

We can see that, here Decision Tree have best accuracy with training score of 0.8926 with BOW method. Overall, accuracy of all the models ranges between 82% to 89%. The above can be further improved by considering sentiment and/or POS.

6.4.2. Without POS, with Sentiment

The below Table 13 illustrates results of all the machine learning and deep learning models without POS while considering Sentiment.

Table 13 Comparison without POS, with Sentiment

Model	Vectorization Method	Training Score	Testing Score	R2 Score	RMSE	MAE
Ridge	BOW	0.9612	0.8211	0.8211	3.7916	2.5066
LinearRegression	BOW	0.9632	0.8209	0.8209	3.7931	2.5128
Decision Tree	BOW	0.9969	0.8937	0.8937	3.5378	1.7048
Random Forest	BOW	0.9900	0.8914	0.8914	2.6622	1.3452
KNeighborsRegressor	BOW	0.8411	0.7431	0.7431	4.5496	2.2334
Ridge	TF-IDF	0.9225	0.8739	0.8739	3.18233	2.1719
LinearRegression	TF-IDF	0.9242	0.8672	0.8672	3.27618	2.3241
Decision Tree	TF-IDF	0.9932	0.8983	0.8983	3.81576	1.8118
Random Forest	TF-IDF	0.9889	0.8966	0.8966	2.74042	1.3746
KNeighborsRegressor	TF-IDF	0.8617	0.7164	0.7164	3.25084	1.6775
LSTM	TF-IDF	0.9052	0.8578	0.8578	3.34169	2.4399

In the above results, we can notice that there is very minor improvement in results, thus considering sentiment does not improve result significantly.

6.4.3. With POS, without Sentiment

The below Table 14 is showing the results for all machine learning and deep learning models with POS but without taking in account of Sentiment. This time, with the vectorization method BOW with LSTM attention model is showing good accuracy with score of 0.9056. It’s also important note that results from Decision tree and random forest are very close and with POS, overall results of models improved significantly.

Table 14 Comparison with POS, without Sentiment

Model	Vectorization Method	Training Score	Testing Score	R2 Score	RMSE	MAE
Ridge	BOW	0.9211	0.8687	0.8687	3.2200	2.3439
LinearRegression	BOW	0.9411	0.8684	0.8684	3.2237	2.3464
Decision Tree	BOW	0.9968	0.8973	0.8973	2.2061	1.9606
Random Forest	BOW	0.9959	0.8964	0.8964	1.6489	0.9769
KNeighborsRegressor	BOW	0.8937	0.7630	0.7630	2.9491	1.4573
Ridge	TF-IDF	0.9329	0.8658	0.8658	2.5793	1.8821
LinearRegression	TF-IDF	0.9348	0.8543	0.8543	2.6352	1.9292
Decision Tree	TF-IDF	0.9945	0.8942	0.8942	2.1684	1.2016
Random Forest	TF-IDF	0.9938	0.8937	0.8937	1.6569	0.9827
KNeighborsRegressor	TF-IDF	0.8779	0.7360	0.7360	4.2491	2.5244
LSTM	TF-IDF	0.9135	0.8729	0.8782	3.34169	2.4399
LSTM Attention	TF-IDF	0.9438	0.9056	0.9056	3.2164	2.3687

6.4.4. With POS and Sentiment

The below Table 15 illustrates results of all the machine models while taking in account both POS and sentiment.

Table 15 Comparison with POS and Sentiment

Model	Vectorization Method	Training Score	Testing Score	R2 Score	RMSE	MAE
Ridge	BOW	0.9281	0.8693	0.8693	3.2111	2.3332
LinearRegression	BOW	0.9218	0.8698	0.8698	3.2143	2.3358
Decision Tree	BOW	0.9998	0.9081	0.9081	2.1984	1,2097
Random Forest	BOW	0.9958	0.9040	0.9040	1.6520	0.9779

Results

KNeighborsRegressor	BOW	0.9369	0.8912	0.8912	2.9579	1.4600
Ridge	TF-IDF	0.9429	0.8914	0.8914	2.5786	1.8819
LinearRegression	TF-IDF	0.9439	0.8910	0.8910	2.6350	2.9296
Decision Tree	TF-IDF	0.9999	0.9085	0.9085	2.1986	1.2093
Random Forest	TF-IDF	0.9958	0.9037	0.9037	1.6551	0.9820
KNeighborsRegressor	TF-IDF	0.8850	0.7658	0.7658	4.2760	2.5365
LSTM	TF-IDF	0.9350	0.9102	0.9102	3.36620	2.5210

In the above results shows that LSTM with POS gives accuracy of 91% and LSTM with POS and sentiment yields the best result. Here, we can see that LSTM model can be improved drastically when POS and sentiment both are considered.

Above models was run with epoch value set to 60. Results can be further improved by increasing the value of epoch for the deep learning algorithm. But this requires very high computational power which can be very expensive and thus for this paper value of epoch was kept 60.

7. Discussion

This chapter starts with a discussion on an attempt to answer the research questions this thesis is intended for. Followed by a discussion on how data set is used and refined along with the models and methods applied on that.

7.1. Techniques for Grading of Essay

From the relevant work done in the section 2.2, it can be assumed that previously quantitative variables have been used up till this point for estimating the essay scores. This suggests that content of essays was not considered during the grading process. Furthermore, electronic essay rater used statistical model on processed data with the mix of three techniques.

Lack in use of vectorization techniques in any of the previous studies proves that there is still room for improvements when talking about Essay Grading System.

7.2. Assessment Metrics of Methods and Techniques

In terms of evaluation metrics in order to determine how well a statistical or machine learning model is doing, in this thesis three matrices have been utilized. R2 Square, Mean Absolute Error, Root Mean Squared Error.

One hand, RMSE tells the typical distance between the predicted value made by the regression model and the actual value. while R2 tells how well the predictor variables can explain the variation in the response variable. MAE on the other hand, in context of Machine Learning is absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation. They are discussed in section 3.5.

7.3. ML Models for Automatically Grading Essays

Modeling is the process of creating, training, and evaluating a machine learning or deep learning model. In my thesis project, two Vectorization methods are being utilized. BOW is one of them, which is NLP technique for feature extraction from the text data. Second Vectorization method utilized is TFIDF for the purpose of document search and information retrieval.

7.4. Thesis dataset

In this thesis an open-source dataset, ASAP-AES, is considered for the research. The chosen dataset has 12978 rows and 4 attributes named essay, manual score, unnamed and prompt. Attribute 'essay' is object type and prompt are of integer type. Other two are float data type. Essay score prediction data is used to predict the scores of students in different essays. Essays are in English language that were written by students as answers to some queries. The length of the essay's ranges from around 150 to 550 words on average.

The dataset is preprocessed first by handling the null values, followed by handling missing data and cleaning Textual data. Furthermore, stop words and handled before applying POS is applied on the data. The grammatical patterns with their respective words are concatenated before applying the ML models and methods on the preprocessed data.

8. Conclusion & Future Works

The proliferation of the internet led to the development of a method to teaching and grading writing that takes place online. An automatic marking system for English writing that is based on wireless networks has the potential to heighten students' impressions of improper language phenomena, assist them in avoiding making the same mistakes again, and minimize the amount of effort that English instructors put in. However, the current automatic scoring system for English essays suffers from slow scoring efficiency, low accuracy, and weak portability.

In this study, we successfully present an autonomous English essay grading system that uses machine learning techniques as its foundation. It includes not only checks grammar and but also semantics-based relationships within the essay content and polarity of opinion expressions. Therefore, our study will help to lowers the number of independent features required to be separated from the text while using the most essential features required in automated essay assessment for improved accuracy.

For the syntax of essay text, I used two vectorization methods, Bag of Words and TF-IDF, and also joined the counting of part of speeches with the vectors and these vectors were given to machine learning algorithms to get trained on the vectors so that it can make the prediction of the score. And then I used five machine learning algorithms and two deep learning algorithm that is LSTM and LSTM Attention.

With deep learning algorithm I have also studied how Part of Speech (POS) and Sentiment effects the accuracy of deep leaning model. Machine learning algorithms were ridge, linear regression, KNN, random forest and decision tree. From all of the models it was concluded with deep learning along with POS and sentiment performs the best with accuracy of about 91%. Although in this thesis we attempted to recognize the count and content of essay for grading. This work provides all details required to reproduce the results along with accuracy.

Our study not only offers accurate values but also delivers details to the users using it reproducible. If we compare other previous systems, the work done in this study is clearer and more repeatable (Janda, H.K., Pawar, A., Du, S. and Mago, V., 2019). So, this study can eradicate the manual work for academic professionals, providing them more time to focus on teaching and also will help students to be assure of fair and reliable evaluation during each submission.

For future, we believe this model lacks the study of semantic analysis in this study. Semantic analysis attempts to understand the context of sentence and read text structure that helps to make NLP applications more accurate. we could say that focus can be more specific on several types of essays, for instance bad-faith essays or grading essays written in different language other than English. Results can further be improved if models are trained with more epochs using better hardware with superior computational power.

Furthermore, we might need to test our model on more than one dataset, this would improve the generalization ability of our model.

9. Reference

A. Kaur and M. Sasi Kumar, 2019. Performance analysis of LSA for descriptive answer assessment. *Innovations in Computer Science and Engineering,* Volume 79, p. 57–63.

Bransford, J. D., and Johnson, 1972. Contextual prerequisites for understanding: some investigations of comprehension and recall. Volume 11, p. 717–726.

Cui, X., 2001. Comparison of Chinese and American composition evaluation standards. Volume 22, p. 28–29.

Janda, H.K., Pawar, A., Du, S. and Mago, V., 2019. Syntactic, semantic and sentiment analysis: The joint effect on automated essay evaluation. IEEE Access, 7, pp.108486-108503.
Dikli, Semire, 2006. Automated essay scoring. Turkish Online Journal of Distance.

Fišer, Darja and Jakob Lenardič, 2019. Overview of tools for text normalization. *Automated Essay Evaluation Using Natural Language Processing and Machine Learning.*

Fu, R., Wang, D., Wang, S., 2018. Elegart sentence recognition for automated eassay scoring. Volume 32, p. 88–97.

Garnham, A., 1981. Mental models as representations of text. Volume 9, p. 560–565.

Ghanta, Harshanthi, 2019. Automated Essay Evaluation Using Natural Language.

Gong, J., 2016. The Design and Implement of the Rhetoric Recognition System Oriented Chinese Essay Review. *Harbin Institute of Technology (HIT).*

Graesser, A., McNamara, D., Louwerse, M., 2004. Coh-metrix_analysis of text on cohesion and language.. *Methods Instrum,* Volume 36, p. 193–202.

Hao, S. D., Xu, Y. Y., Peng, H. L., Su, K. L., 2014. Automated Chinese essay scoring from topic perspective using regularized latent semantic indexing. *Proceedings of the 22nd International Conference on Pattern Recognition 2014.*

Hearst, M., 2000. The debate on automated essay grading. *IEEE Intelligent Systems and Their Applications,* Volume 5, pp. 22-37.

Jin, H., and Liu, H., 2016. Chinese writing of deaf or hard-of-hearing students and normal-hearing peers from complex network approach. Volume 7, p. 1777.

Johnson-Laird, 2005. Mental models and thoughts. *The Cambridge Handbook of Thinking and Reasoning,* p. 185–208.

K. K. Y. Chan, T. Bond, and Z. Yan, 2022. Application of an automated essay scoring engine to English writing assessment using many-facet rasch measurement. *Language Testing.*

Kakkonen, T., Myller, N., Sutinen, E., and Timonen, J., 2008. Comparison of dimension reduction methods for automated essay grading. Volume 11, p. 275–288.

Landauer, T., Foltz, P., and Laham, D., 1998. An introduction to latent semantic analysis. Volume 25, p. 259–284.

Landauer, Thomas, W Kintsch, 2003. Latent semantic analysis AutomatedEssay Scoring. *A Cross-disciplinary Perspective,* p. 87.

Linden, Anné and Kathrin Perutz, 2008. Mindworks: An introduction to NLP. *Crown House Publishing.*

Madnani, Nitin and Aoife Cahill, 2018. Automated scoring: Beyond natural language processing”. In: Proceedings of the 27th International Conference on Computational Linguistics. p. 1099–1109.

McNamara, D. S., Crossley, S. A., Roscoe, R. D., Allen, L. K., 2015. A hierarchical classification approach to automated essay scoring. Volume 23, p. 35–59.

Meurers, Detmar, 2012. Natural language processing and language learning. *Encyclopedia of applied linguistics,* p. 4193–4205.

Page, E. B., 1994. Computer grading of student prose, using modern concepts and software. *The Journal of Experimental Education,* Volume 2, p. 127–142.

Seel, N., 1999. Semiotics and structural learning theory. Volume 14, p. 1–10.

Shermis, Mark D and Jill C, 2003. Automated essay scoring. *A cross-disciplinary.*

Somasundaran, S., Riordan, B., Gyawali, B, 2016. Evaluating argumentative and narrative essays using graphs,” in Paper Presented at the the 26th International Conference on Computational Linguistics. *26th International Conference on Computational Linguistics.*

T. K. Landauer, D. Laham, and P. W. Foltz, 2000. The intelligent essay assessor. *IEEE Intelligent Systems,* Volume 15, p. 27–31.

Valenti, Salvatore, Francesca Neri, 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education,* p. 319–330.

Villalon, J., and Calvo, 2011. Concept maps as cognitive visualizations of writing assignment. Volume 14, p. 16.

Wang, Y., and Liu, H., 2019. The effects of source languages on syntactic structures of target languages in the simultaneous interpretation: a quantitative investigation based on dependency syntactic treebanks. Volume 45, p. 89–113.

A. Kaur and M. Sasi Kumar, 2019. Performance analysis of LSA for descriptive answer assessment. *Innovations in Computer Science and Engineering,* Volume 79, p. 57–63.

Bransford, J. D., and Johnson, 1972. Contextual prerequisites for understanding: some investigations of comprehension and recall. Volume 11, p. 717–726.

Cui, X., 2001. Comparison of Chinese and American composition evaluation standards. Volume 22, p. 28–29.

Dikli, Semire, 2006. Automated essay scoring. *Turkish Online Journal of Distance.*

Fišer, Darja and Jakob Lenardič, 2019. Overview of tools for text normalization. *Automated Essay Evaluation Using Natural Language Processing and Machine Learning.*

Fu, R., Wang, D., Wang, S., 2018. Elegart sentence recognition for automated eassay scoring. Volume 32, p. 88–97.

Garnham, A., 1981. Mental models as representations of text. Volume 9, p. 560–565.

Ghanta, Harshanthi, 2019. Automated Essay Evaluation Using Natural Language.

Gong, J., 2016. The Design and Implement of the Rhetoric Recognition System Oriented Chinese Essay Review. *Harbin Institute of Technology (HIT).*

Graesser, A., McNamara, D., Louwerse, M., 2004. Coh-metrix_analysis of text on cohesion and language.. *Methods Instrum,* Volume 36, p. 193–202.

Hao, S. D., Xu, Y. Y., Peng, H. L., Su, K. L., 2014. Automated Chinese essay scoring from topic perspective using regularized latent semantic indexing. *Proceedings of the 22nd International Conference on Pattern Recognition 2014.*

Hearst, M., 2000. The debate on automated essay grading. *IEEE Intelligent Systems and Their Applications,* Volume 5, pp. 22-37.

Jin, H., and Liu, H., 2016. Chinese writing of deaf or hard-of-hearing students and normal-hearing peers from complex network approach. Volume 7, p. 1777.

Johnson-Laird, 2005. Mental models and thoughts. *The Cambridge Handbook of Thinking and Reasoning,* p. 185–208.

K. K. Y. Chan, T. Bond, and Z. Yan, 2022. Application of an automated essay scoring engine to English writing assessment using many-facet rasch measurement. *Language Testing.*

Kakkonen, T., Myller, N., Sutinen, E., and Timonen, J., 2008. Comparison of dimension reduction methods for automated essay grading. Volume 11, p. 275–288.

Landauer, T., Foltz, P., and Laham, D., 1998. An introduction to latent semantic analysis. Volume 25, p. 259–284.

Landauer, Thomas, W Kintsch, 2003. Latent semantic analysis AutomatedEssay Scoring. *A Cross-disciplinary Perspective,* p. 87.

Linden, Anné and Kathrin Perutz, 2008. Mindworks: An introduction to NLP. *Crown House Publishing.*

Madnani, Nitin and Aoife Cahill, 2018. Automated scoring: Beyond natural language processing". In: Proceedings of the 27th International Conference on Computational Linguistics. p. 1099–1109.

McNamara, D. S., Crossley, S. A., Roscoe, R. D., Allen, L. K., 2015. A hierarchical classification approach to automated essay scoring. Volume 23, p. 35–59.

Meurers, Detmar, 2012. Natural language processing and language learning. *Encyclopedia of applied linguistics,* p. 4193–4205.

Page, E. B., 1994. Computer grading of student prose, using modern concepts and software. *The Journal of Experimental Education,* Volume 2, p. 127–142.

Seel, N., 1999. Semiotics and structural learning theory. Volume 14, p. 1–10.

Shermis, Mark D and Jill C, 2003. Automated essay scoring. *A cross-disciplinary.*

Somasundaran, S., Riordan, B., Gyawali, B, 2016. Evaluating argumentative and narrative essays using graphs," in Paper Presented at the the 26th International Conference on Computational Linguistics. *26th International Conference on Computational Linguistics.*

T. K. Landauer, D. Laham, and P. W. Foltz, 2000. The intelligent essay assessor. *IEEE Intelligent Systems,* Volume 15, p. 27–31.

Valenti, Salvatore, Francesca Neri, 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education,* p. 319–330.

Mark, J. and Goldberg, M.A., 1988. Multiple regression analysis and mass assessment: A review of the issues. Appraisal Journal, 56(1).

Villalon, J., and Calvo, 2011. Concept maps as cognitive visualizations of writing assignment. Volume 14, p. 16.

Wang, Y., and Liu, H., 2019. The effects of source languages on syntactic structures of target languages in the simultaneous interpretation: a quantitative investigation based on dependency syntactic treebanks. Volume 45, p. 89–113.

zesch, 2021. *ltl-ude, Github.* [Online]
Available at: https://github.com/ltl-ude/EduScoringDatasets
[Accessed 2022].

Zhang, J., and Ren, J. , 2014. Experimental research report of chinese language test electronic grader. Volume 10, p. 27–32.

Chai, T. and Draxler, R.R., 2014. Root mean square error (RMSE) or mean absolute error (MAE). Geoscientific Model Development Discussions, 7(1), pp.1525-1534.

Hoerl, A.E. and Kennard, R.W., 1970. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), pp.55-67.

Zhang, Mo, 2013. Contrasting automated and human scoring of essays. *R & D Connections,* pp. 1-11.

Ouyang, X., Zhou, P., Li, C.H. and Liu, L., 2015, October. Sentiment analysis using convolutional neural network. In 2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing (pp. 2359-2364). IEEE.

Ma, Y., Peng, H., Khan, T., Cambria, E. and Hussain, A., 2018. Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis. Cognitive Computation, 10(4), pp.639-650.

Zupanc, K., & Bosnić, Z., 2018. Increasing accuracy of automated essay grading by grouping similar graders. *Proceedings of the 8th International Conference on Web Intelligence.*

Zupanc, K., and Bosnic, Z., 2017. Automated essay evaluation with semantic analysis. Volume 120, p. 118–132.