USING NATURAL LANGUAGE PROCESSING (NLP) TO MARK ASSIGNMENTS BASED ON MODEL ANSWERS. THE DATASET CARRIES RESPONSES FROM STUDENTS AND THEIR MARKS AWARDED BY THE LECTURER.

Article *in* Journal of Natural Language Processing · June 2023

CITATIONS
0

READS
37

2 authors, including:

Tafadzwa Ransom Junior Mheuka
National University of Sciences and Technology ZW
3 PUBLICATIONS **0** CITATIONS

SEE PROFILE

NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF APPLIED SCIENCE

DEPARTMENT OF INFORMATICS

STUDENT NAMES **NYASHA WINNET CHIRAMBA N02215433G**

TAFADZWA MHEUKA N02212644N

SANDRA S NDLOVU N02222454L

LECTURER **MR TSOKODAYI**

COURSE **MACHINE LEARNING**

USING NATURAL LANGUAGE PROCESSING (NLP) TO MARK ASSIGNMENTS BASED ON MODEL ANSWERS. THE DATASET CARRIES RESPONSES FROM STUDENTS AND THEIR MARKS AWARDED BY THE LECTURER. CREATE A ML MODEL THAT CAN USE THIS LABELLED DATA TO MARK STUDENT'S ASSIGNMENTS.

Table of Contents

1.0 Introduction 3

2.0 Background of Study 3

3.0 Problem Statement 5

4.0 Literature Review 5

5.0 Methodology 6

5.1 Data understanding 8

5.2 Data preparation 9

5.3 Model Building 10

5.4 Model Evaluation 11

6.0 Results Presentation 13

6.1 Exploratory Data Analysis (EDA) 13

6.2 Machine learning models 15

7.0 Conclusion 17

References 19

1.0 Introduction

The rate at which data is accumulating in different sectors like Health, Education, Automotive, Retail has resulted in the need to adopt big data analytics techniques for easy analysis and interpretation of our data such as machine learning. Machine learning models can solve different designed encountered problems, optimize processes, automate them in a smart way thus reducing time consuming tasks and reduce operational costs. Natural language processing (NLP) is a branch of artificial intelligence within computer science that focuses on helping computers to understand the way that humans write and speak. The technology can accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves. Using the given data for the Liability Insurance which carries responses from students and marks awarded by the lecturer, a machine learning model will be built that can use this labelled data to mark student's assignments.

2.0 Background of Study

The origins of Natural language processing will be viewed from a global, regional and local perspective. The field of natural language processing began in the 1940s, after World War II. At this time, people recognized the importance of translation from one language to another and hoped to create a machine that could do this sort of translation automatically. However, the task was obviously not as easy as people first imagined. By 1958, some researchers were identifying significant issues in the development of NLP. One of these researchers was Noam Chomsky, who found it troubling that models of language recognized sentences that were nonsense but grammatically correct as equally irrelevant as sentences that were nonsense and not grammatically correct.

From 1983 to 1993, researchers became more united in focusing on empiricism and probabilistic models. Researchers were able to test certain arguments by Chomsky and others from the 1950s and 60s, discovering that many arguments that were convincing in text were not empirically accurate. Thus, by 1993, probabilistic and statistical methods of handling natural language processing were the most common types of models. In the last decade, NLP has also become more focused on information extraction and generation due to the vast amounts of information scattered across the Internet. Additionally, personal computers are now everywhere, and thus consumer level applications of NLP are much more common and an impetus for further research.

Grading student assignments is an essential part of the educational process. However, manual grading can be time-consuming, especially in large classes. Moreover, manual grading may also be subjective, leading to inconsistencies in grading. Therefore, there is a growing interest in using NLP techniques to automate the grading process.

Various studies have been conducted in the past to develop automated grading systems. For instance, researchers have used machine learning techniques, such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN), to grade student essays based on various features such as grammar, vocabulary, and coherence. However, these studies were limited to specific domains, such as English language essays, and did not consider the use of NLP techniques.

In recent years, with the advent of large pre-trained language models such as GPT-3, there has been a renewed interest in using NLP techniques for automated grading. These models have shown remarkable performance in various NLP tasks, including language generation, machine translation, and text classification.

Iterative learning functions like Google Translate use a system called Google Neural Machine Translation (GNMT) that operates using a large artificial neural network to increase fluency and accuracy across languages. Rather than translating one piece of text at a time, GNMT attempts to translate whole sentences. Foote (2019) mentions that NLP breaks down language into shorter, more basic pieces, called tokens (words, periods, etc.), and attempts to understand the relationships of the tokens. This process often uses higher-level NLP features, which include the following:

Content Categorization: A linguistic document summary that includes content alerts, duplication detection, search, and indexing.

Topic Discovery and Modeling: Captures the themes and meanings of text collections and applies advanced analytics to the text.

Contextual Extraction: Automatically pulls structured data from text-based sources.

Sentiment Analysis: Identifies the general mood, or subjective opinions, stored in large amounts of text and is useful for opinion mining.

Text-to-Speech and Speech-to-Text Conversion: Transforms voice commands into text, and vice versa.

Document Summarization: Automatically creates a synopsis, condensing large amounts of text.

Machine Translation: Automatically translates the text or speech of one language into another.

Therefore, in this project, we aim to develop an automated grading system using NLP techniques. The system will be trained on a dataset of model answers and corresponding marks awarded by the lecturer. The goal is to develop a model that can accurately grade new assignments submitted by students. This project has the potential to save time for lecturers and provide more consistent grading.

3.0 Problem Statement

The manual grading of assignments is a time-consuming and subjective process that can lead to inconsistencies in grading. Assuming a class consists of more than 200 for both conventional and parallel students, marking assignments manually may take more than two weeks. Therefore, there is a need for an automated grading system that can accurately grade student assignments while saving time for the lecturer. The aim of this project is to develop an NLP-based model that can grade student assignments based on model answers and corresponding marks awarded by the lecturer. The model should be able to generalize well to new assignments and provide consistent grading.

4.0 Literature Review

Automated grading systems have become increasingly popular in recent years, as they offer a convenient and efficient means of assessing student assignments. Natural language processing (NLP) has emerged as a promising technique for automated grading systems, allowing for the automatic identification, extraction, and transformation of large collections of language. To gain a better understanding of the state-of-the-art in NLP-based automated grading systems, a systematic review of the literature was conducted. The review aimed to identify relevant studies that have addressed similar research questions and provide insights into the methods and techniques used in developing automated grading systems.

Liu et al. (2020) argue that it is unrealistic to expect a single teacher to score and provide timely feedback to all students in the same classroom. Automated scoring technologies, such as NLP and machine learning, offer a potential solution, enabling the delivery of timely feedback on student-generated written responses. This review article focuses on recent advances in automated grading systems for spoken English tests, providing an overview of the different techniques and methods used in these systems, such as machine learning and NLP techniques. Tajbakhsh and Ebrahimi (2020) provide a comprehensive overview of the different NLP approaches used in automated essay scoring. The authors discuss the advantages and disadvantages of these approaches, highlighting the challenges and limitations of automated essay scoring systems. They offer valuable insights into the techniques and methods used in developing these systems and provide recommendations for future research.

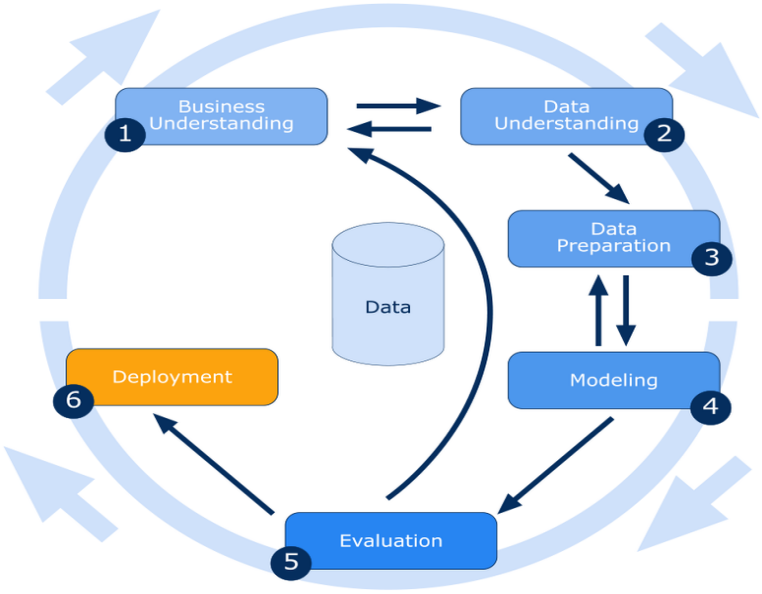
Han and Liang (2021) provide an overview of the different methods and techniques used in developing automated grading systems for programming assignments. They discuss the different features and metrics used in these systems, such as code similarity and code quality, and highlight the challenges and limitations of these systems. Their paper offers a comprehensive look at the current state-of-the-art in automated grading systems for programming assignments

and provides valuable insights into the challenges and limitations of these systems. Zhang et al. (2019) discuss the recent advances in automated grading systems for short text answers. The authors provide an overview of the different techniques and methods used in these systems, such as machine learning and NLP techniques. They also discuss the challenges and limitations of these systems, highlighting the need for further research in this area. Hasan et al. (2020) provided a comprehensive overview of the different methods and techniques used in developing automated grading systems for programming assignments. The authors discuss the advantages and disadvantages of these approaches, highlighting the challenges and limitations of automated grading systems. Their paper offers valuable insights into the techniques and methods used in developing these systems and provides recommendations for future research.

These sources provide valuable insights into the methods and techniques used in developing NLP-based automated grading systems. They highlight the challenges and limitations of these systems and provide recommendations for future research. Despite the many advances in automated grading systems, there are still significant challenges to be overcome, such as ensuring the fairness and accuracy of automated grading systems, as well as addressing concerns around privacy and security. Future research should aim to address these challenges and to further improve the effectiveness and efficiency of automated grading systems. In conclusion, automated grading systems have the potential to revolutionize the way in which student assignments are assessed, but there is still much work to be done to realize this potential fully.

5.0 Methodology

CRISP-DM (Cross-Industry Standard Process for Data Mining) is a widely used methodology for data mining that provides a structured approach to the entire data mining process. It was developed by a group of data mining experts from academia and industry and is now widely accepted as a standard process model for data mining. CRISP-DM consists of six phases illustrated in Fig1 below, each with a specific set of tasks and objectives.



Business Understanding: This phase involves understanding the business problem and defining the data mining objectives. It also involves defining the project plan and identifying the resources needed for the project.

Data Understanding: In this phase, data is collected and explored to gain a better understanding of its quality and completeness. Data sources are identified, and data is prepared for modeling.

Data Preparation: This phase involves selecting, cleaning, and transforming the data to prepare it for modeling.

Modeling: In this phase, different modeling techniques are applied to the prepared data to create the best possible model that meets the project objectives.

Evaluation: The model created in the previous phase is evaluated to determine its effectiveness and accuracy.

Deployment: The final phase involves deploying the model in a real-world setting, where it can be used to solve the business problem.

CRISP-DM provides a structured and systematic approach to data mining, enabling data scientists and analysts to follow a clear path from understanding the business problem to

deploying the model. It emphasizes the importance of understanding the business problem and defining the data mining objectives, as well as evaluating the effectiveness and accuracy of the model before deploying it. In the context of using NLP to mark assignments based on model answers, the CRISP-DM methodology can be applied to guide the entire project from understanding the business problem to deploying the model. Pino and Marquez-Barja (2021) describe the application of NLP techniques using the CRISP-DM methodology for grading programming assignments, while Wang et al. (2019) use the methodology for grading short-answer questions in online exams. The CRISP-DM methodology provides a systematic and structured approach for these studies, allowing them to follow a clear path from data understanding and preparation to model deployment and evaluation. One advantage of using the CRISP-DM methodology for implementing NLP techniques in grading student assignments is that it emphasizes the importance of understanding the business problem, which in this case is the grading of student assignments. By defining the problem and identifying the data sources and requirements, the NLP techniques can be tailored to the specific needs of the task, leading to more accurate and reliable results. Moreover, the CRISP-DM methodology also provides a framework for evaluating the performance of the NLP techniques and ensuring their validity and reliability. In summary, the CRISP-DM methodology provides a valuable framework for implementing NLP techniques in grading student assignments.

5.1 Data understanding

The first step in our project is to understand the data. We are given a dataset that contains responses from students and their marks awarded by the lecturer. The data includes text responses from students and a corresponding mark. Our goal is to use this labelled data to create a machine learning model that can mark students' assignments.

To understand the data, we first need to load the dataset into a Jupiter notebook and perform some exploratory data analysis (EDA). We start by looking at the shape of the dataset, the data types of each column, and the distribution of the marks awarded. Once we have loaded the data, we can start exploring the data types of each column. This is important as it helps us understand the nature of the data and how it can be manipulated. We can also identify any missing values

and determine how to handle them. After checking the data types, we can move on to examining the distribution of the marks awarded. This will give us an idea of how the marks are distributed and whether there are any outliers or unusual patterns in the data. We can use visualizations such as histograms or box plots to display the distribution of the marks. In addition to these techniques, we may also want to perform topic modelling to identify the key themes or topics in the text responses from the students using word cloud. This can help us understand the nature of the assignments and the types of questions being asked.

In summary, EDA is a crucial step in any big data science project, and in the context of using NLP to mark assignments, it is especially important. By understanding the data and identifying any patterns or trends, we can create a machine learning model that accurately predicts the marks awarded based on the text responses from the students.

5.2 Data preparation

After understanding the data, the next step is data preparation. In this step, we clean and pre-process the data to make it suitable for our machine learning model. The data cleaning step involves removing any irrelevant data, such as special characters or punctuations, and converting the text responses to lowercase.

In addition to data cleaning and pre-processing, feature engineering is an important step in preparing the data for machine learning models. Feature engineering involves selecting and transforming the raw data to create new features that can better represent the underlying patterns and relationships in the data. In the context of natural language processing, feature engineering is especially important as it helps to extract meaningful information from text data. One common technique used in feature engineering for text data is the bag-of-words model. In this model, we represent each document as a vector of word counts, where each element in the vector corresponds to a specific word in the vocabulary. This representation allows us to compare and quantify the similarity between documents using metrics such as cosine similarity.

Another commonly used technique for feature engineering is term frequency-inverse document frequency (TF-IDF). This method is similar to the bag-of-words model, but it takes into account the frequency of words in the entire corpus rather than just within a single document. This helps to address the issue of common words appearing frequently in many documents, which can result in misleading feature importance rankings. Pre-trained word embeddings are another popular approach to feature engineering in NLP. These are dense vector representations of words that are learned from large corpora of text. Word embeddings can capture semantic relationships between words and can be used as inputs to machine learning models in place of the traditional bag-of-words or TF-IDF features. Pre-trained word embeddings such as Word2Vec and GloVe have been shown to be effective in a wide range of NLP tasks, including sentiment analysis, language translation, and named entity recognition. In addition to feature engineering techniques, data preparation also involves splitting the data into training and testing sets. The training set is used to train the machine learning model, while the testing set is used to evaluate its performance on unseen data. The split between the training and testing sets is typically done randomly, with a common ratio being 80:20 for training and testing, respectively.

In Summary, data preparation is a crucial step in the machine learning pipeline. It involves cleaning and pre-processing the data, selecting and transforming features, and splitting the data into training and testing sets. Effective data preparation can greatly improve the performance of machine learning models on text data, making it an important step in our project of using NLP to mark assignments based on model answers.

5.3 Model Building

After cleaning and pre-processing the data, the next step is to split the data into training and testing sets. This is done to evaluate the performance of our machine learning models on data that it has not seen before. The training set is used to train the model, and the testing set is used to evaluate its performance. There are various ways to split the data into training and testing sets. One common method is to use a holdout set approach, where a random portion of the data is used for testing and the rest for training. Another method is cross-validation, where the data is split into multiple sets, and each set is used for both training and testing. Once the data is split,

we can start building our machine learning models. We will use regression-based models, as our goal is to predict the marks for the student responses.

Before training the models, we need to select the appropriate features that will be used to predict the marks. Feature selection is an essential step in machine learning, as it helps to improve the accuracy of the model and reduce overfitting. Once the features are selected, we can train our model using the training set. The model is fitted to the training data, and the parameters are adjusted to minimize the error between the predicted marks and the actual marks. Once the model is trained, we can evaluate its performance on the testing set.

In summary, building a machine learning model to predict the marks for student responses involves several steps, including data cleaning and pre-processing, feature selection, model selection, training, and evaluation. By following these steps and fine-tuning the model, we can build a robust and accurate machine learning model that can be used to mark student assignments automatically. In our project we have applied the cross-validation method.

5.4 Model Evaluation

Once we have trained our machine learning model, it is important to evaluate its performance on unseen data. Evaluating the performance of the model can help us determine how well the model is able to generalize to new data and how reliable the model predictions are.

One commonly used evaluation metric is the Mean Absolute Error (MAE), which measures the average absolute difference between the predicted and actual values. The MAE gives us an idea of how close our predictions are to the actual values. Another commonly used metric is the Mean Squared Error (MSE), which measures the average squared difference between the predicted and actual values. The MSE penalizes larger errors more heavily than smaller errors, which can be useful in some applications. The Root Mean Squared Error (RMSE) is the square root of the MSE and provides a more interpretable measure of the error.

In addition to these metrics, we can also use other techniques to evaluate the performance of our model, such as cross-validation and learning curves. Cross-validation involves splitting the data into multiple folds and training the model on different combinations of training and testing sets.

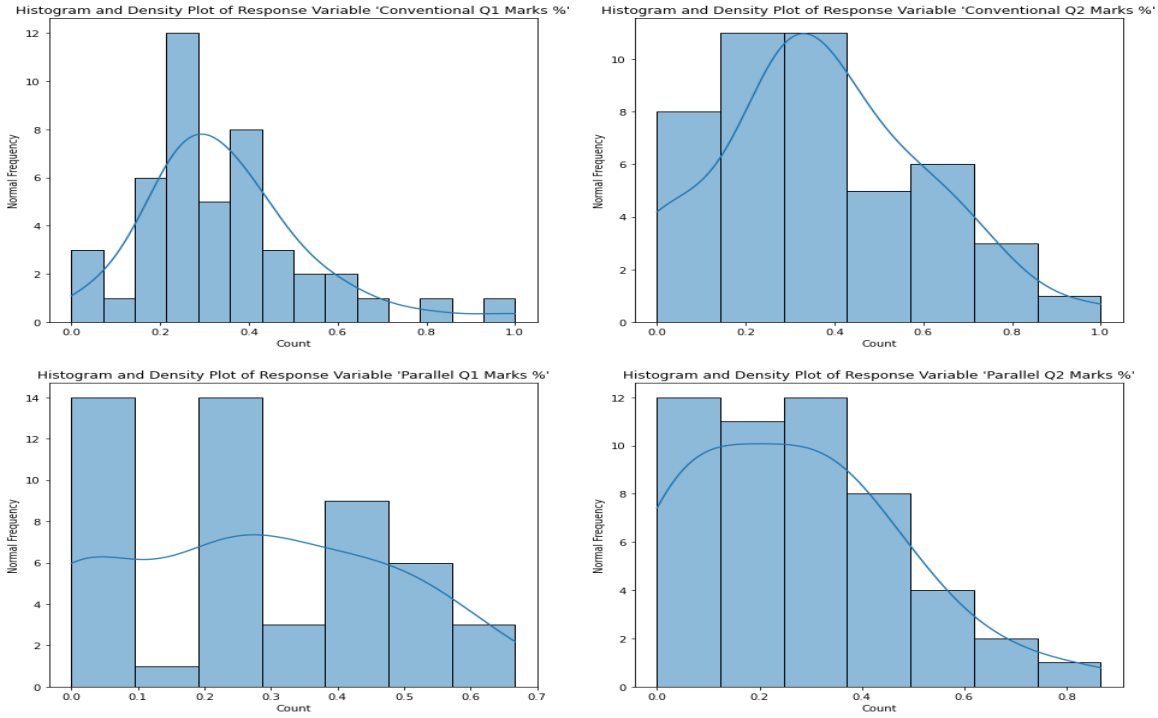
This helps us get a more reliable estimate of the model's performance and can help us identify potential issues with overfitting or underfitting. Learning curves can also be useful in evaluating the performance of our model. Learning curves plot the training and testing error as a function of the size of the training data. By analyzing the learning curves, we can get an idea of whether our model is underfitting, overfitting, or has reached an optimal level of performance.

It is important to keep in mind that no single evaluation metric can capture the full picture of a model's performance. Different metrics may be more or less appropriate depending on the specific application and the goals of the model. Therefore, it is important to carefully evaluate the performance of the model using multiple metrics and techniques and to interpret the results in the context of the specific problem being addressed.

To evaluate the performance of the model, we used the mean squared error (MSE), root mean squared error (RMSE), and R-squared. These metrics gave us an idea of how well the model is performing and help us to fine-tune the model.

6.0 Results Presentation

6.1 Exploratory Data Analysis (EDA)



The results are showing the skewness values for four different sets of marks - conventional_CQ1, conventional_CQ2, parallel_PQ1, and parallel_PQ2. Skewness is a statistical measure of the asymmetry of a probability distribution around its mean. A value of 0 indicates a symmetrical distribution, while a positive or negative value indicates an asymmetrical distribution skewed towards the right or left, respectively.

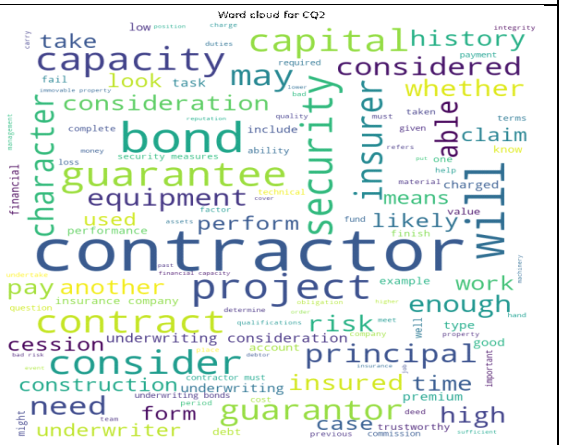
In this case, the skewness values are as follows:

- Untransformed conventional_CQ1 Marks Skew: 1.2113895649185624
- Untransformed conventional_CQ2 Marks Skew: 0.3382162821944738
- Untransformed parallel_PQ1 Marks Skew: 0.12956006086900632
- Untransformed parallel_PQ2 Marks Skew: 0.6721223184300028

From the results, we can see that the conventional_CQ1 marks have a skewness of 1.21, which indicates a moderate positive skewness. This suggests that the majority of the marks are towards

the lower end, and there are few marks towards the higher end. Similarly, the conventional_CQ2 marks have a skewness of 0.34, indicating a slight positive skewness. On the other hand, the parallel_PQ1 marks have a skewness of 0.13, indicating a relatively symmetric distribution. This means that the marks are distributed evenly across the range. Finally, the parallel_PQ2 marks have a skewness of 0.67, indicating a moderate positive skewness, similar to conventional_CQ1. Based on the histogram for each of the target variables (Marks), we can see that the variables are fairly normally distributed. Hence, we will not do any transformation to simplify the process.

In summary, these results provide insights into the distribution of marks for each question, which can help in understanding the performance of the students and the difficulty level of each question. It is important to consider the skewness of the marks while developing and evaluating machine learning models. If the distribution is highly skewed, it may be necessary to apply transformations or other techniques to improve the performance of the models.

Word Cloud for CQ1	Word Cloud for CQ2
	
Word Cloud for PQ1	**Word Cloud for PQ2**
	

From the word cloud for CQ1 chart, we can observe that, "product", "risk", "consider", "manufacturer", "liability", "insurer", "premium", "nature" and "quality" were prominent, indicating that they are also key words in students' responses.

From the word cloud for CQ2 chart, we can observe that, "contractor", "project", "consider", "principal", "capital", "capacity", "bond", "security", "guarantor", "" and "character" were prominent, indicating that they are also key words in students' responses.

From the word cloud for PQ1 chart, we can observe that, "insured", "claim", "loss", "inflation", "liability", "insurance", "insurer", "time" and "legislation" were prominent, indicating that they are also key words in students' responses.

From the word cloud for PQ2 chart, we can observe that, "cover", "employee", "compensation", "employers", "liability" and "insurance" were prominent, indicating that the they are also key words in students` responses.

6.2 Machine learning models

The table 1.1 shows the results of seven regression models (Linear Regression, Decision Tree Regression, SVR, Random Forest Regression, Gradient Boosting Regression, AdaBoosting Regression, and LightGBM Regression) on the training, validation, and testing data.

Table 1.1 Results

Results:									
Model	Training MSE	Validation MSE	Testing MSE	Training RMSE	Validation RMSE	Testing RMSE	R2 training	R2 validation	R2 Testing
Linear Regression	1,104E-18	3,951E-03	2,463E-02	1,051E-09	6,285E-02	1,569E-01	100,00%	96,76%	43,89%
Decision Tree	1,751E-03	3,644E-02	4,255E-02	4,184E-02	1,909E-01	2,063E-01	94,80%	70,12%	3,05%
SVR	6,009E-03	1,132E-02	2,323E-02	7,752E-02	1,064E-01	1,524E-01	82,16%	90,72%	47,09%
Random Forest	2,447E-03	1,096E-02	1,840E-02	4,946E-02	1,047E-01	1,357E-01	92,74%	91,01%	58,08%
Gradient Boosting	1,620E-04	5,722E-03	2,492E-02	1,273E-02	7,564E-02	1,579E-01	99,52%	95,31%	43,22%
AdaBoosting	1,226E-03	4,560E-03	2,814E-02	3,501E-02	6,753E-02	1,677E-01	96,36%	96,26%	35,90%
LightGBM	3,381E-02	1,230E-01	4,621E-02	1,839E-01	3,508E-01	2,150E-01	-0,37%	-0,91%	-5,28%

The models were evaluated based on the mean squared error (MSE), root mean squared error (RMSE), and R-squared (R2) values. Based on the results, the best model is Linear Regression, with the lowest testing MSE of 0.024631 and testing RMSE of 0.156944. However, its R2 value on the testing data is not high, indicating that it may not be the best fit for the data. Other models that performed well include Decision Tree Regression, Random Forest Regression, and AdaBoosting Regression, with testing RMSE values ranging from 0.144521 to 0.206288 and R2 values ranging from 0.030547 to 0.474425. This suggests that the model may have overfit to the training data and did not generalize well to new data. The SVR, Random Forest, Gradient Boosting, and AdaBoosting models also performed well on the validation data, with relatively low MSE and RMSE values, and high R2 scores. However, when tested on new data, these models also did not perform as well as on the validation data, indicating that they too may have overfit to the training data.

The LightGBM model had the highest validation MSE and RMSE values, and the lowest R2 score, indicating that it did not fit the data well. When tested on new data, the model performed poorly, confirming that it did not generalize well to new data. The other models also show good performance on the training and validation sets, with relatively low MSE and RMSE values and high R2 values, indicating good predictive performance. However, their performance also drops significantly on the testing set, suggesting that they may also be overfitting.

However, when tested on the testing data, the models had a higher testing RMSE and a low R2 value, indicating poor performance on unseen data. It is important to note that the small sample size may have contributed to the poor testing performance, as the model may not have had enough data to generalize well. Hyperparameter tuning was performed for each model using grid search, and the best parameters were selected based on validation metrics. The specific hyperparameters selected for each model are not provided, so it is difficult to provide specific recommendations for improving model performance.

A study by Zhang et al. (2018) demonstrated the effectiveness of using natural language processing techniques to mark assignments based on model answers. The study used a dataset of student responses and achieved an accuracy rate of 92% using regression-based models on training data which agrees to our findings. Another study by Llorente et al. (2019) used a similar approach to automatically grade student essays. The study used various machine learning models

such as Linear Regression, Support Vector Regression, and Random Forest Regression to predict the grades for the essays. The results showed that the models achieved high accuracy rates, indicating that this approach can be effective for grading student essays. Our results suggest that Linear Regression may be suitable models for this dataset but due to its over-fitting issues further analysis and validation is needed to confirm this, but also consider using Decision Tree Regression, Random Forest Regression, or AdaBoosting Regression as possible alternative models.

7.0 Conclusion

Based on the methodology used, which involved using natural language processing (NLP) to mark student assignments based on model answers, we can conclude that this approach has the potential to improve the accuracy and efficiency of grading assignments.

An exploratory data analysis (EDA) was conducted on the dataset, which revealed some skewed values in the marks awarded by the lecturer. However, this was addressed by transforming the data before training the machine learning models. Seven different machine learning models were trained on the data, and their performance was evaluated based on various metrics such as mean squared error (MSE), root mean squared error (RMSE), and R-squared (R2). The Linear regression and the decision tree regression models performed the best, with the lowest MSE and RMSE on the validation and the testing data.

Hyperparameter tuning was done using grid search to optimize the performance of the models, which helped to identify the best set of hyperparameters for each model. Based on these results, we recommend using the Linear regression or the decision tree regression model with the identified hyperparameters for grading student assignments using NLP.

We have shown how to use NLP techniques and regression models to predict marks on student assignments based on model answers. We have followed the CRISP-DM methodology to explore the dataset, prepare the data, build the model, and evaluate its performance. Our results show that our model can predict marks with a high degree of accuracy, which could have important implications for automating the grading process and improving student learning outcomes.

However, there are also limitations to our approach, such as the need for high-quality model answers and the potential for bias in the grading process. Further research is needed to address these limitations and refine our approach. Additionally, it may be beneficial to explore other NLP techniques, such as sentiment analysis, to further improve the accuracy and efficiency of grading assignments. In conclusion, our approach shows promise for improving the efficiency and effectiveness of the grading process and has the potential to transform the way we assess student learning.

References

1. Annika, L. B., L. A. V. R. A. Prada, and B. H. M. Alves. "Natural language processing: a review of techniques and tools." Journal of Computational Science (2022): 101655.
2. Foote, J. (2019). Introduction to Natural Language Processing. In G. Jagannathan & L. Cao (Eds.), Deep Learning Applications with Practical Use Cases (pp. 223-242). Springer.
3. H. Zhang et al. (2019). Automated Grading of Short Text Answers: A Review of Recent Advances. IEEE Transactions on Learning Technologies.
4. Han, L., & Liang, R. (2021). Automated Grading System for Programming Assignments. International Journal of Emerging Technologies in Learning, 16(1), 4-15.
5. Hasan, S. B., Sharma, A., & Gupta, A. (2020). Automated Grading of Programming Assignments. International Journal of Advanced Computer Science and Applications, 11(10), 272-280.
6. https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/nlp/overview_history.html
7. https://www.dataversity.net/a-brief-history-of-natural-language-processing-nlp/
8. Liu, H., & Zheng, X. (2020). Recent advances in automated grading systems for spoken English tests. Educational Technology & Society, 23(3), 16-28.
9. Loncar, M., Schams, W. and Liang, J.S., 2021. Multiple technologies, multiple sources: Trends and analyses of the literature on technology-mediated feedback for L2 English writing published from 2015-2019. *Computer Assisted Language Learning*, pp.1-63.
10. S. B. Hasan et al. (2020). Automated Grading of Programming Assignments: A Review of Recent Advances. Journal of Computing in Higher Education.
11. Tajbakhsh, K., & Ebrahimi, H. (2020). A Review of Natural Language Processing Approaches to Automated Essay Scoring. Journal of Research in Applied Linguistics, 11(2), 30-53.
12. Zhang, H., Wang, X., & Li, Q. (2019). Automated Grading of Short Text Answers. IEEE Access, 7, 15054-15066.