Scientext: Streamlining Essay Analysis Using Natural Language Processing

github: https://github.com/ayesha-mohsin/Scientext
Shaik Ayesha Mohsin || Areeba Hassan

Abstract:
In educational environments, the grading of essays is both pivotal and resource-intensive, often burdening educators with substantial time commitments and the challenge of maintaining objectivity. This project develops Scientext, an Automated Essay Scoring system aimed at alleviating these challenges by automating the evaluation of student essays. Utilizing advanced natural language processing techniques, Scientext analyzes essays from a comprehensive dataset, extracting linguistic features such as syntax complexity, lemma diversity, and orthographic accuracy. Employing machine learning algorithms including Linear Regression and AdaBoost, Scientext predicts essay scores that demonstrate high correlation with human assessments, as validated by measures like mean squared error and Cohen's kappa score. The implementation of this system promises substantial benefits: it significantly reduces the time educators spend on grading, enhances the consistency of essay evaluations, and provides immediate feedback to students, thereby supporting a more dynamic learning process. Scientext not only streamlines educational assessments but also offers insights into the key components of effective writing, paving the way for targeted educational interventions.

1. Introduction

Writing serves as a fundamental skill for assessing verbal, quantitative reasoning, and qualitative analysis capabilities from an early age in educational systems. Essays, in particular, are critical for evaluating student proficiency and comprehension. Traditionally, the grading of these essays has been labor-intensive, requiring meticulous attention to ensure fairness and objectivity. This process not only consumes significant time but also delays the feedback crucial for student development. Over the past few decades, advancements in computer technology, especially in natural language processing, have paved the way for innovative solutions like automated essay scoring systems, which promise to streamline this traditional method.

Scientext leverages these technological advancements to automate the grading process effectively. By analyzing essays through the extraction of statistical features and applying machine learning models, Scientext aims to replicate and potentially enhance the grading accuracy of human evaluators. The system evaluates various textual attributes such as vocabulary, style, and syntax, and further refines its assessments with structural metrics like sentence complexity and grammatical accuracy. By simulating human scoring mechanisms and validating them against established grading standards, Scientext not only enhances the efficiency of academic evaluations but also introduces a scalable, unbiased grading tool into the educational landscape.

2. Problem statement

Scientext around the challenge of grading essays, which is traditionally a time-consuming and subjective task, susceptible to biases. The hypothesis posits that by employing Natural Language

Processing (NLP) and machine learning techniques, it is feasible to develop a system that can reliably automate the grading process, mirroring the accuracy and objectivity of human graders.

Scientext is conceptualized as a machine learning project integrating several NLP tasks. The primary task is regression, where the system predicts numerical scores for essays based on their content quality. This is formalized mathematically as $y = f(X)$, where y is the predicted score and X represents the extracted features from the essays, such as word count, sentence complexity, and grammatical correctness. Additionally, classification aspects are also explored, categorizing essays into qualitative performance bands (e.g., fail, pass, and excellent), based on predefined scoring thresholds. These tasks utilize feature extraction techniques from text data, including tokenization, lemmatization, and syntactic analysis, to feed into algorithms like Linear Regression and AdaBoost for prediction, thus combining both traditional statistical methods and more complex ensemble methods in machine learning.

In essence, Scientext aims to systematize and expedite the essay evaluation process, enhancing both the scalability and fairness of educational assessments.

3. Methodology

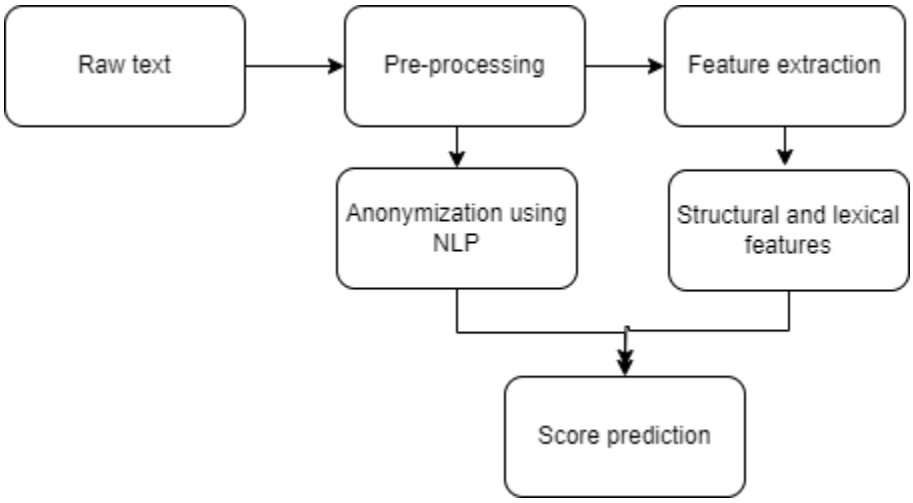


Fig 1. Workflow of the Scientext Essay Scoring System

3.1. Overall Workflow:

- Data Loading and Pre-processing:
  - The essay dataset is loaded from a TSV file, where each essay is associated with a numerical score.
  - Pre-processing steps involve cleaning the text, removing special characters, and tokenizing the essays into words and sentences.
- Feature Extraction:
  - Structural and lexical features are extracted from the pre-processed essays, including average word length, word count, character count, sentence count, and counts of nouns, adjectives, verbs, and adverbs.

- Anonymization Using NLP:
    - Named Entity Recognition (NER) is employed to anonymize personally identifying information such as names, locations, and dates, replacing them with generic placeholders.
- Model Development:
    - Various regression models including Linear Regression, Ridge Regression, Lasso Regression, Support Vector Regression (SVR), Random Forest Regression, and AdaBoost Regression are trained on the extracted features to predict the essay scores.
- Evaluation:
    - The performance of each model is evaluated using metrics such as Mean Squared Error (MSE), R-squared (R2) score, and Cohen's Kappa Score to assess the agreement between predicted and actual scores.

3.2. Architecture Diagram of Model:

- Raw Text:
  The raw essays are input into the system for processing.
- Pre-processing:
  Text undergoes pre-processing steps including cleaning, tokenization, and removal of special characters to prepare it for feature extraction.
- Feature Extraction:
  Structural and lexical features are extracted from the pre-processed text, providing numerical representations of various linguistic aspects of the essays.
- Anonymization Using NLP:
  Named Entity Recognition is applied to anonymize personally identifying information, ensuring privacy and data security.
- Structural and Lexical Features:
  Extracted features are utilized as input for the machine learning models.
- Score Prediction:
  Trained regression models utilize the extracted features to predict the essay scores.
- Evaluation:
  The predicted scores are evaluated against the ground truth using performance metrics to assess the model's accuracy and reliability.

3.3. Latent Semantic Analysis (LSA)

Latent Semantic Analysis is an advanced technique that helps in uncovering the latent semantic structure within the essay data by reducing the dimensionality of text data. By transforming text data into a lower-dimensional space, LSA helps in capturing the underlying meanings of words within the essays, which are critical for predicting the essay scores more effectively.

In our implementation, we employed the Truncated Singular Value Decomposition (SVD) to perform LSA on the TF-IDF (Term Frequency-Inverse Document Frequency) matrix derived from the essay texts. This approach not only simplifies the complexity of the text data but also enhances the

performance of our predictive models by focusing on the most significant elements that influence essay quality.

Steps involved:

- Text Preprocessing: Essays are cleaned to remove special characters and numbers, and they are then tokenized into words.
- TF-IDF Transformation: Convert the preprocessed essays into a TF-IDF matrix that quantifies the importance of words within the essays relative to the document set.
- SVD Application: Apply Truncated SVD to reduce the dimensionality of the TF-IDF matrix, extracting the most significant topics or concepts across the essays.
- Model Integration: The resulting LSA topic matrix serves as input for our machine learning models, improving their ability to discern patterns and predict essay scores accurately.

This method has proven beneficial, particularly in enhancing the interpretability of our models and in the accurate scoring of essays, as evidenced by improved MSE and R^2 scores in our regression analyses.

3.4. Text Embeddings

Text Embeddings provide a more nuanced approach to understanding textual data by converting words into vectors where semantically similar words are mapped close together in the vector space. For our project, we leveraged Word2Vec embeddings to convert the essays into numerical form that captures not just the frequency but also the context of words used within the essays.

Steps involved:

- Tokenization: Essays are tokenized into sentences and then into words.
- Word2Vec Training: Train a Word2Vec model on the tokenized text to create word embeddings that capture the semantic meanings of words based on their context within the essays.
- Embedding Averaging: For each essay, average the Word2Vec embeddings of all words to create a single vector that represents the entire essay.
- Model Application: These essay embeddings are then used as features for training our machine learning models, providing a depth of understanding that typical lexical features cannot offer.

Implementing Word2Vec has allowed our system to capture subtle nuances in language use that are often indicative of higher or lower quality essays, thereby refining our scoring capabilities and enhancing model reliability.

4. Dataset

For the Scientext project, we utilize a dataset initially curated for the 2012 Hewlett Foundation Automated Essay Scoring Kaggle competition. This dataset comprises eight distinct sets of essays, each derived from different educational levels and tasks, ranging from persuasive and narrative to expository styles. These essays are penned by students from the 7th to 10th grades, varying in length from 100 to 600 words, with each set containing around 1,500 samples. A key aspect of the dataset is its division by student level and task type, with some sets focused on specific topics and others on general writing abilities.

Each essay set varies significantly, not only in the style and educational level of the essays but also in the scoring range, which can vary from 0 to 60, depending on the set. To standardize these scores for analytical purposes, they are normalized to a uniform scoring range of 0-3. The essays are anonymized to remove personal identifiers like names and locations, using sophisticated natural language processing tools such as the Named Entity Recognizer from the Stanford NLP group. This ensures the privacy of the students and the neutrality of the data.

Essay set no	Type of the Essay	Grade Level	Number of Samples	Min domain1 score	Max domain1 score
1	persuasive / narrative / expository	8	1783	2	12
2	persuasive / narrative / expository	10	1800	1	6
3	source dependent responses	10	1726	0	3
4	source dependent responses	10	1772	0	3
5	source dependent responses	8	1805	0	4
6	source dependent responses	10	1800	0	4
7	persuasive / narrative / expository	7	1569	0	30
8	persuasive / narrative / expository	10	723	0	60

Fig 2. Dataset Description

Design of Features/Labels:

Scientext extracts a variety of features from these essays to predict their quality and corresponding scores. These features include:

- Structural Features: Word count, sentence count, character count, and average word length. These features provide insights into the essay's complexity and verbosity.
- Lexical Features: Noun, verb, adjective, and adverb counts, which help assess the richness of the language used.
- Other Features: Lemma count, which measures the diversity of vocabulary, and misspell count, which indicates the accuracy of the language.

These features are crucial for the regression and classification models used to predict the essay scores. The models aim to emulate the grading process of human raters and are trained to recognize patterns and linguistic traits that correspond to higher-quality essays.

5. Exploratory data analysis

- Data Visualization:
    - The dataset was visualized to gain insights into the distribution and characteristics of the essays and their corresponding scores.
    - Histograms were plotted to visualize the distribution of essay scores across different bins, providing an overview of the score distribution.
    - Score distributions were analyzed with respect to various textual features such as character count, lemma count, sentence count, word count, and average word length. This allowed for an understanding of how these features vary across different score categories.
- Data Pre-processing:
    - Raw text data was pre-processed using various techniques to extract meaningful features for analysis and modeling.
    - Essays were cleaned to remove non-alphanumeric characters and tokenized into words and sentences using natural language processing tools from the NLTK library.
    - Features such as average word length, sentence count, word count, character count, lemma count, and misspell count were extracted from the pre-processed text data.
    - Missing values in the scores were identified and dropped from the dataset to ensure data integrity for further analysis and modeling.
    - Numerical scores were converted into categorical labels based on percentile thresholds to facilitate classification tasks.
    - Data was split into training and testing sets for regression and classification analyses, ensuring a robust evaluation of the models' performance.

Overall, these exploratory data analysis and pre-processing steps laid the foundation for subsequent modeling tasks by providing insights into the dataset's characteristics and preparing the data for effective analysis and interpretation.

6. Implementation

6.1. Algorithms

Regression

In our regression analysis, we employed both Linear Regression and AdaBoost Regression to forecast scores, with AdaBoost showing slight superiority in certain cases. While we anticipated AdaBoost to outperform consistently due to its enhanced learning capacity, the outcomes did not consistently support this expectation. We gauged model performance using Mean Squared Error (MSE) and R2 Score, where R2 scores averaged around 60, indicative of satisfactory model fit.

$$MSE = \frac{\sum_{i=1}^{n}(Ideal - Prediction)^2}{n} (n = \#\ of\ samples)$$

$$R2 = \frac{Explained\ variation}{Total\ variation}$$

Classification

$$K\left(x^{(i)}, x^{(j)}\right) = \phi(x^{(i)})^T \phi(x^{(j)})$$
$$= \exp\left(-\gamma \left\| x^{(i)} - x^{(j)} \right\|^2\right), \qquad \gamma > 0$$

For classification tasks, we opted for SVM with an RBF kernel, as linear separation proved inadequate for our data's complexity. The choice of gamma, crucial for the RBF kernel, was determined empirically, yielding optimal performance when set to 10^-5. We evaluated our classification models using precision, recall, F1-score, and kappa score. While our kappa scores, averaging around 0.55, indicate imperfect agreement between predictions and labels, the other metrics fall within the 0.70 range, suggesting reasonable model performance.

6.2. Implementation libraries

- For regression analysis: sklearn.linear_model.LinearRegression, sklearn.ensemble.AdaBoostRegressor.
- For classification analysis: sklearn.svm.SVC, sklearn.metrics.precision_recall_fscore_support, sklearn.metrics.cohen_kappa_score.
- For text preprocessing and feature extraction: nltk, re, string.
- For data manipulation: pandas, numpy.
- For visualization: matplotlib, seaborn.

6.3. Explanation of implementation

- The implementation begins by extracting features from the essays, such as word count, sentence count, average word length, and others, using functions like avg_word_len, word_count, sent_count, etc.
- The data is preprocessed to handle missing values and scale features if necessary.
- For regression analysis, Linear Regression and AdaBoost Regression models are trained and evaluated using mean squared error (MSE) and R2 score.
- For classification analysis, scores are relabeled based on their distribution, and an SVM with RBF kernel is trained and evaluated using precision, recall, f1-score, and Cohen's kappa score.
- The entire process is repeated for each essay set individually and then combined for a more general analysis.
- The results are presented in tables showing the performance metrics of each model for regression and classification tasks under different scenarios.

7. Results

The performance of the models was evaluated using various metrics and visualizations.

Performance Measures:

- For regression models, both mean squared error (MSE) and R2 score were utilized as evaluation metrics. MSE represents the average squared deviation from the ideal value, while R2 score indicates the goodness of fit.

```
Testing for Linear Regression

Mean squared error: 0.04501437384270856
Mean squared error in percentage: 1.5004791280902854
Variance score: 0.7134352182307053
Testing for Adaboost Regression

Mean squared error: 0.04450911613048784
Mean squared error in percentage: 1.4836372043495947
Variance score: 0.7166517256188947
```

Essay set no	Linear Regression		AdaBoost Regression	
	MSE in percentage	R2 Score	MSE in percentage	R2 Score
1	1.63	0.71	1.47	0.74
2	2.34	0.52	2.43	0.50
3	10.9	0.50	12.8	0.42
4	11.6	0.60	14.0	0.52
5	5.5	0.66	5.4	0.67
6	6.85	0.54	8.42	0.44
7	5.35	0.54	5.27	0.55
8	1.16	0.58	1.34	0.51
9 (All sets combined)	10.9	0.28	8.55	0.44

Fig 3. Findings from the regression models across various conditions

- Classification models were assessed using precision, recall, f1-score, and kappa score. Kappa score measures the agreement between predictions and labels, with scores categorized based on their magnitude.

```
Cohen's kappa score: 0.6327261963244278

              precision    recall  f1-score   support

           0       0.79      0.72      0.75        79
           1       0.80      0.87      0.83       307
           2       0.80      0.69      0.74       149

    accuracy                           0.80       535
   macro avg       0.80      0.76      0.78       535
weighted avg       0.80      0.80      0.79       535
```

Essay set no	Precision	Recall	F1-Score	Kappa score
1	0.79	0.79	0.79	0.62
2	0.74	0.74	0.74	0.53
3	0.76	0.78	0.77	0.55
4	0.72	0.72	0.70	0.52
5	0.74	0.73	0.73	0.56
6	0.74	0.76	0.74	0.45
7	0.65	0.64	0.63	0.39
8	0.62	0.63	0.62	0.41
9 (all sets combined)	0.63	0.63	0.61	0.37

Fig 4. Findings from the classification model

Visual Diagrams for Results with Explanation:

- Exploratory data analysis revealed the importance of certain features, with lemma count emerging as the most significant variable across multiple sets. Higher lemma counts correlated positively with essay scores, reflecting richer content. Conversely, average word length exhibited minimal impact on scores, consistent with its random distribution in essays.
- Distribution fitting for features such as lemma count and average word length illustrated distinct patterns among different score categories. These visualizations provided insights into the relationship between features and essay scores, aiding model interpretation and refinement.

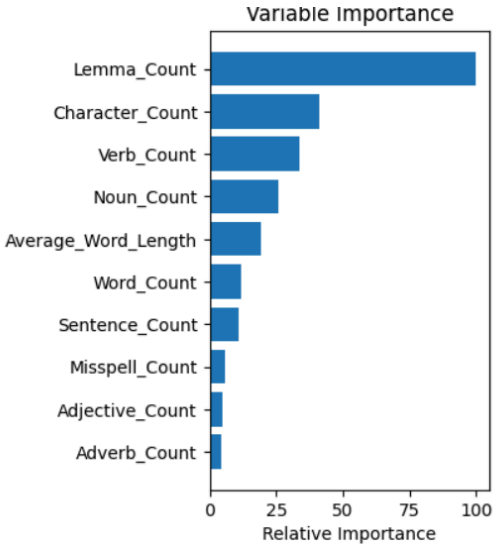


Fig 5. Importance of variables in the regression model across all sets

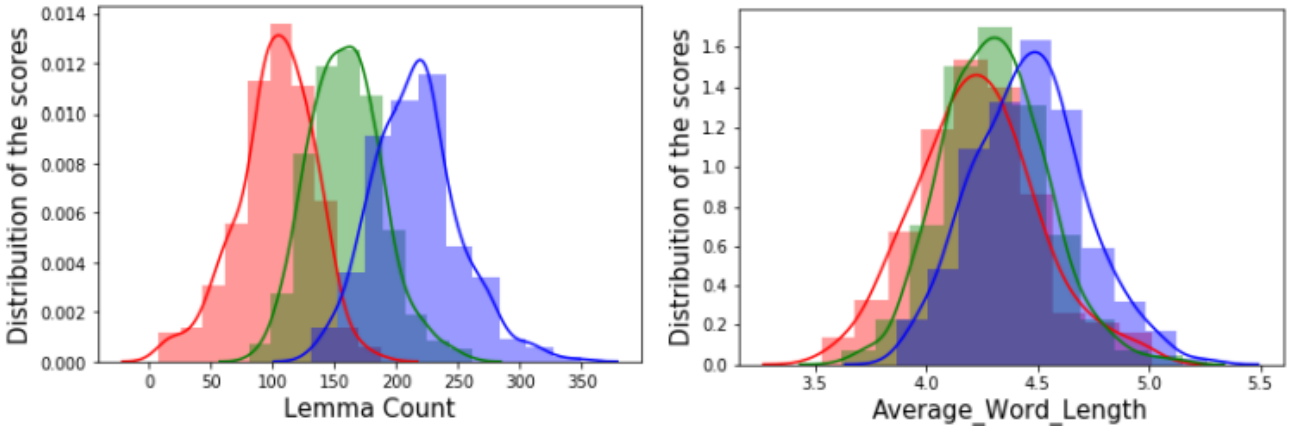
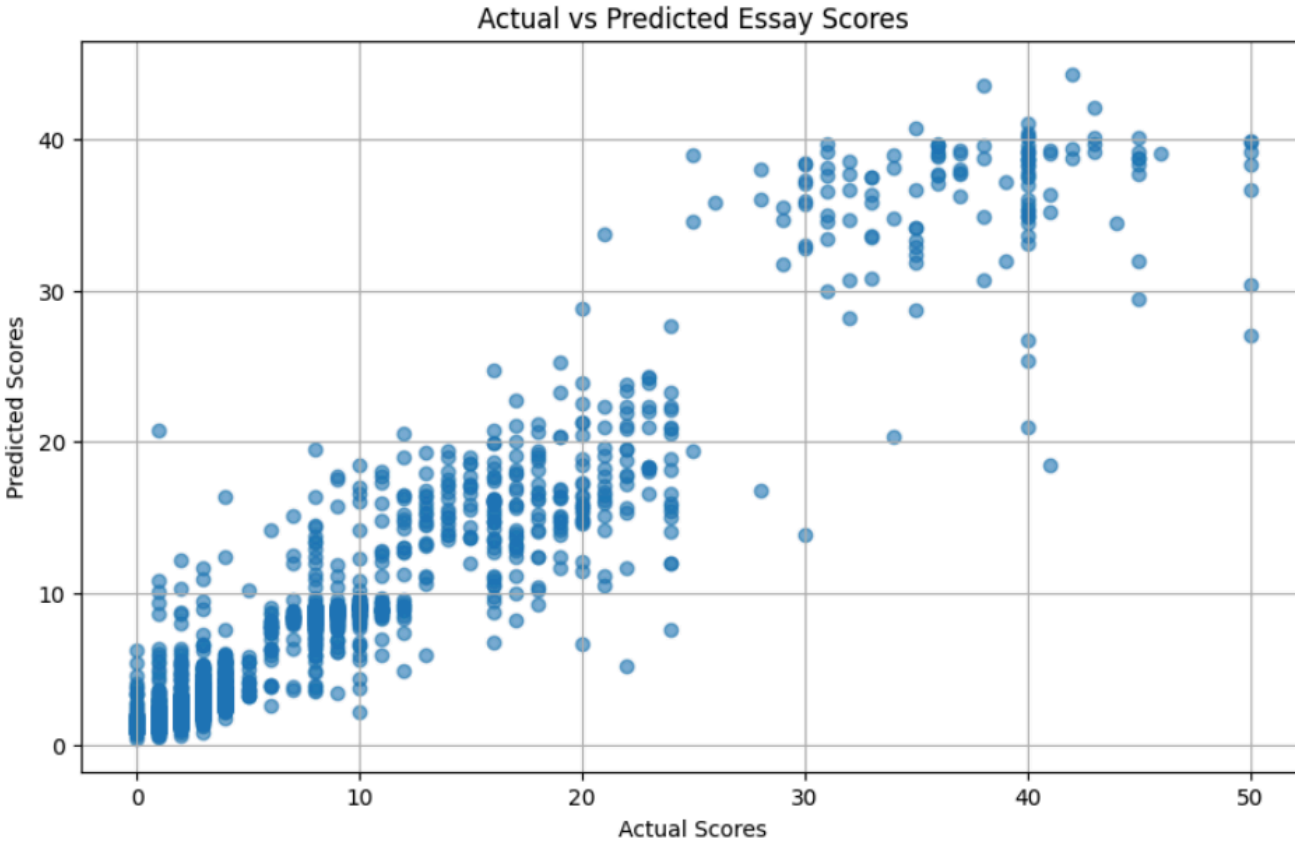


Fig 6. Feature distributions for lemma count and average word length across different score categories (fail in red, pass in green, and good in blue) for set 1.



The MSE of approximately 6.67 and the R^2 Score of approximately 0.91 indicate that the model performs well in predicting essay scores based on the LSA topics.

The Automated Essay Scoring system can be made more accurate and robust by including new features and methods.This car be achieved by highlighting key features or parts of the essay that contributed significantly to its score.

Fig 7. Results from Latent Semantic Analysis

Analysis and Discussion:

- Classification model results displayed inconsistencies attributed to the biased scoring scheme employed during label conversion. The division of scores into three categories based on score

distribution may not accurately reflect the variability in essay quality, particularly in scenarios where all students achieve high scores.

- Regression models exhibited favorable performance, with MSE consistently below 10% across most training scenarios, indicating accurate score predictions with minor errors. Notably, models excelled in predicting scores for persuasive, narrative, and expository essays, while performance lagged for source-dependent essays.
- Despite the computational overhead associated with misspell count feature extraction, it remains indispensable for essay evaluation. While alternative approaches may offer efficiency gains, they often lack coverage of contemporary language, such as slang and social media terms, compromising their effectiveness.

Overall, the results underscore the effectiveness of the regression models in predicting essay scores, while highlighting the challenges inherent in classification tasks due to scoring biases and feature complexities. Continued refinement of models and feature engineering strategies is essential to enhance performance and address emerging linguistic trends.

8. Project Management

Work Completed:

- Development and validation of an automated essay scoring system using advanced NLP techniques.
- Integration of Latent Semantic Analysis and Text Embeddings to enhance model performance.

Description:
The project involved loading and preprocessing essay data, extracting features, training regression and classification models, and evaluating their performance. The addition of LSA and Text Embeddings aimed to improve the understanding of essay content at a deeper semantic level.

Responsibility:
- Latent Semantic Analysis: Implemented by Shaik Ayesha Mohsin.
- Text Embeddings Integration: Carried out by Areeba Hassan.
- General Project Workflow and Management: Jointly handled by both team members.

Contributions:
- Shaik Ayesha Mohsin: 50% - Focused on integrating and tuning Latent Semantic Analysis, preprocessing data, and feature engineering.
- Areeba Hassan: 50% - Responsible for incorporating Text Embeddings, model training, and performance evaluation.

Issues/Concerns:
- Initial challenges included optimizing the feature extraction process to balance between performance and computational efficiency.

- Some issues with model overfitting were addressed by adjusting model parameters and enhancing the training dataset.

9. References/Bibliography

1. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media Inc.
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (pp. 1247-1250).
3. Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1), 37-46.
4. Hewlett Foundation. (2012). Automated Essay Scoring. Kaggle. https://www.kaggle.com/c/asap-aes
5. Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. Discourse processes, 25(2-3), 259-284.
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.
7. Perelman, L. (2014). Critique of Automated Essay Scoring. MIT University.
8. Shermis, M. D., & Hamner, B. (2013). Contrasting state-of-the-art automated scoring of essays: Analysis. In Proceedings of the National Council on Measurement in Education.
9. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.
10. Zhang, H., & Litman, D. (2018). Co-training succeeds in computational linguistics. Journal of Artificial Intelligence Research, 61, 797-833.