Milestone II Final Report
Predicting Text Difficulty

Team 12: Ryan Eskuri, Zhentao Wang, Hank (Hyun Su) Ye

Introduction

In the rapidly evolving digital era, one size does not fit all. This holds especially true in the field of education, where learners exhibit a wide array of reading abilities. To enhance and personalize the learning experience, it becomes crucial to match the reading material's complexity to the learner's reading proficiency. Consequently, we proposed the problem of predicting text difficulty on our project. The objective is to develop an effective model that can accurately determine the difficulty level, easy or difficult, of a given text dataset.

A solution to this issue will not only help educators and curriculum developers to match texts with the appropriate grade levels but also aid in creating personalized learning pathways. The resulting model could be used in adaptive learning systems that adjust text complexity according to user ability, improving educational outcomes. Moreover, the model's usefulness could extend beyond educational applications to include content creation and editing software, where content complexity can be adjusted to suit a target audience.

Our project introduces a novel contribution to this field, particularly in the choice and computation of features used in our models. We incorporate domain-specific features derived from linguistic and readability studies to make our models sensitive to language subtleties that affect text comprehension. This project is a comprehensive study that not only predicts text difficulty but also provides insights into what makes a text difficult for learners.

Related Work

https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests

The Flesh-Kincaid readability tests were developed under contract with the U.S. Navy to determine the difficulty of text. The scores from the tests attempt to show what grade level of education is needed to understand a text. Our project is different in that we are not attempting to specifically state the level of difficulty a text has or what grade level of education can easily read the text.

Data Source

For our project, we utilized a large corpus of Wikipedia articles provided by the University of Michigan SIADS-696-SS23 Kaggle. The dataset is located at UMich Kaggle Text Difficulty..

The dataset comes in a CSV format, which is easy to import and manipulate using Python's data analysis libraries. The training data includes 416,768 sentences labeled into two categories: 'easy' and 'difficult', represented by integers 0 and 1 respectively. The test data contains 119,092 unlabeled sentences. The objective is to predict the difficulty category for each sentence in the test set.

The primary variables in the dataset are the sentence ID, the sentence itself, and the difficulty category for the training set. The sentence ID provides a unique identifier for each sentence, the sentence variable holds the text content, and the difficulty category defines the class label.

This dataset does not have a time element since it comprises static Wikipedia articles, and the analysis is based on the text content rather than its temporal characteristics.

In addition to the primary dataset, we utilized several helper resources provided alongside the Kaggle dataset. These resources include the Dale-Chall list of around 3000 elementary English words familiar to most American 4th-grade students, concreteness ratings for about 40k English words, and a list detailing the approximate age when each of 50k English words was learned. These resources were instrumental in augmenting our feature set, thereby enriching our models. The meaningful features we are seeing are such as:

In Dale Chall 3000 Word List:

- Percentage of word comprised in our dataset

In Read Concreteness_ratings_Brysbaert_et_al_BRM.txt:

- Mean of concreteness rating (Conc.M)
- Standard deviation of the concreteness ratings (Conc.SD)
- Percentage of participants who knew the word (Percent_know)
- SUBTLEX_US frequency count on total 51 million; Brysbaert & New, 2009 (SUBLTLEX)

In AoA_51715_words.csv:

- Age of acquaintance of lemmatized word
- Percentage of participants who know the lemmatized word

Initial preprocessing involved several steps: we performed text cleaning operations, such as removing special characters, converting all text to lowercase, and stemming or lemmatizing words to their base form. Secondly, we created meaningful features using additional resources and built-in function, Spacy. However, we found some potential noisy or missing data. Firstly, we checked for any missing values of features for each sentence and dropped any records that contained missing values.

Feature Engineering Process

In creating our model features, we went through the following stages of development:

Text Preprocessing Phase: The text data was first loaded in UTF-8 format. We then initiated an iterative process of refining regular expressions to cleanse the text data. In every iteration, we examined a sample of 25 sentences, identified any anomalies, and modified the regex and pandas string methods accordingly to ensure the elimination of textual errors. Our preprocessing involved steps such as lowercase conversion, substitution of HTML tags with relevant symbols, removal of surplus special characters, and correction of whitespace irregularities. The cleansed sentences were subsequently tokenized and lemmatized to decrease the overall vocabulary for facilitating feature extraction.

Generation of Basic Text Features: Employing the cleansed and tokenized text, along with the lemmatized versions, we produced a number of rudimentary features. These included metrics such as sentence length, word count, syllable count, and others.

Computation of Readability Scores: Our literature survey led us to certain studies that quantify the readability of text pieces. We utilized the textstat library to compute Flesch Reading Ease, Gunning Fog Index, and Smog Index scores for the processed text. To aid our classifiers, we binned these scores into three categories: 0 for 'easy', 1 for 'average', and 2 for 'difficult'.

Integration of Supplemental Data Sources: We utilized additional features from the Dale Chall list of basic English words, Age of Acquisition (AoA) words, and Brysbaert Concreteness datasets. We lemmatized these words and averaged the scores of similar words. Features created from these were Dale Chall word counts, AoA words separated into 'high' and 'low' categories separated at twelve years old, and the average concreteness of the words present in a sentence.

Development of Advanced Features: We deployed advanced NLTK tools to identify the counts of parts of speech and counts of words with multiple meanings within a sentence. An evaluation of the complete training set identified over 30 parts of speech. However, given that many of these were infrequent, we decided to concentrate on four prevalent ones: Nouns, Verbs, Adjectives, and Adverbs.

Employment of SpaCy Linguistic Features: In order to better discern text complexity, we harnessed the power of SpaCy. This helped us to expand our final features, incorporating metrics such as the count of clauses, number of noun subjects, unique verb tenses, and average dependency distances.

Unutilized Features: We attempted to utilize tf-idf weights and BERT transformations, but these approaches did not improve the F1 score when used independently. Attempts to incorporate these features with our existing features substantially increased the model training time without yielding immediate improvements. Future analyses may consider employing PCA or truncated SVD to integrate these additional features effectively.

Part A: Supervised Learning

In this project, our supervised learning workflow commenced with the use of a Dummy Classifier (with a 'mean' strategy) as a baseline model. This enabled us to gauge the performance of subsequent models relative to a simple, uninformed prediction model.

We then employed three diverse models to solve our binary classification problem. The models selected were Logistic Regression, Support Vector Machines (SVM), and Gradient Boosted Forest, each chosen to cover a spectrum of model complexity - from the simplicity of Logistic Regression, the non-linearity of SVMs, to the intricate mechanisms of Gradient Boosted Forests. Our objective was to probe potential trade-offs between computational efficiency and model performance.

Our feature engineering process was designed to capture the essence of the problem at hand. We extracted a wide array of features from the text, falling broadly into categories such as text characteristics, linguistic complexity, parts-of-speech counts, named entities, readability metrics, and word complexity. By including a diverse set of features, we aimed to encapsulate both simple and complex sentence structures, thereby providing our models with a rich representation of the text data.

For model optimization, we utilized Grid Search to tune hyperparameters. Although computationally intensive, Grid Search systematically works through all possible combinations of the provided hyperparameters, enabling us to find the optimal values and evaluate our models under different conditions and configurations.

The rationale behind our choices revolved around the nature of our binary classification task. We aimed to explore the performance of models with varying levels of complexity and handling of linearity in the data. Furthermore, our comprehensive and diverse feature set was designed to accommodate the structural and syntactic variance in text data. In sum, our approach aimed to maximize model performance by leveraging a comprehensive feature set and a diverse set of models, while fine-tuning them through rigorous hyperparameter optimization.

Supervised Evaluation

We decided to use F1 scoring to be more robust than a pure accuracy score. While our EDA showed that there is a great class balance in our training set, we do not know if this is true for the test set. If the test set were to have a major class imbalance, accuracy alone could be misleading. We want to make sure recall is also being taken into account and F1 provides us that balance.

	Dummy Classifier	**Logistic Regression**	**SVM**	**Gradient Boosted Forest**
Mean F1 Score	**0.50**	**0.653**	**0.6667**	**0.736**
Standard Deviation	**0**	**0.0069**	**0.0056**	**0.0062**

(Figure 1)

Comparing the mean F1 scores across 5 fold cross validation, we can see that Gradient Boosted achieved the highest F1 score (0.736), followed by SVM (0.6667), Logistic Regression (0.653). However, prior to hyper-parameter tuning the differences in F1 scores between the classifiers are relatively small.

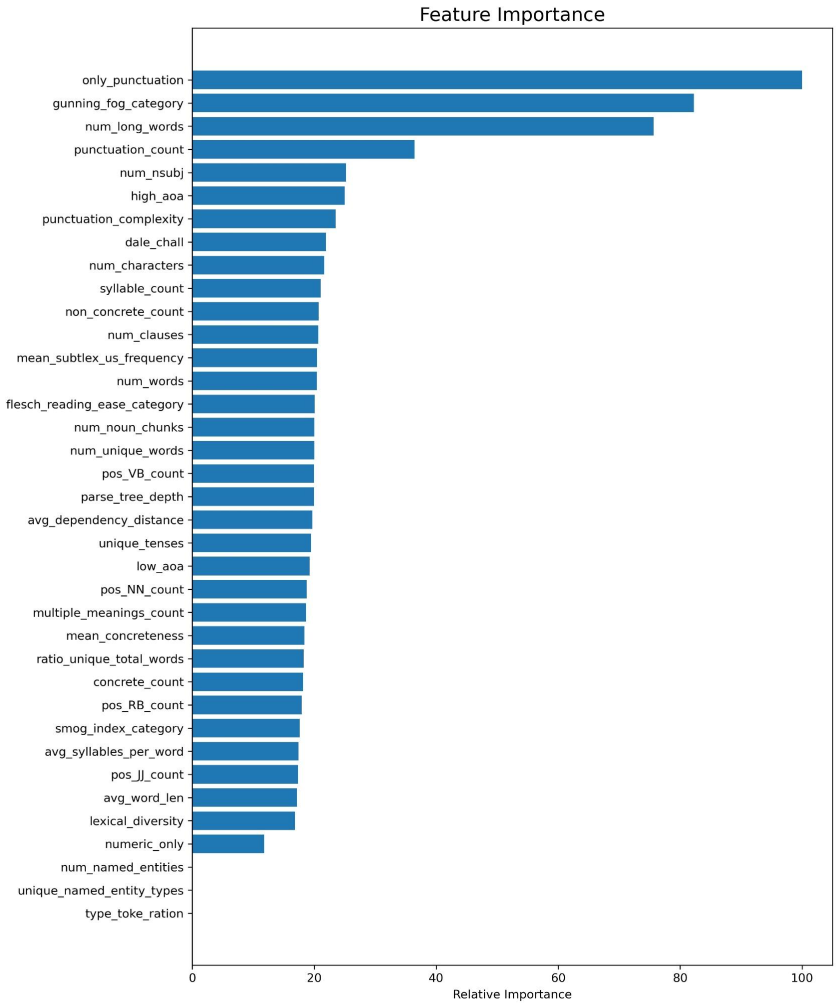
Considering the standard deviations, which represent the variability in performance across different folds, all classifiers have relatively low standard deviations. This suggests that the models are consistent in their performance across different cross-validation folds.

In summary, the low standard deviations of the models' performances underscore their consistency, attesting to the robustness of our selected features. While there is potential for enhancement through the use of more complex models or additional features, it's imperative to balance these considerations with other factors. Aspects like computational efficiency, model interpretability, and the specific demands of the problem at hand must all be weighed when selecting the most fitting classifier for the task.

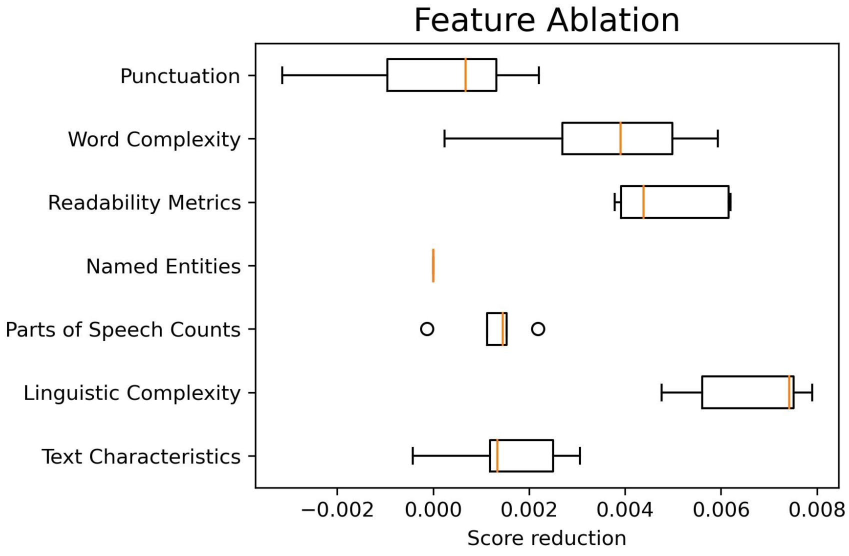
Exploration of Gradient Boosted Trees

Upon initial analysis of the feature importance (refer to Figure 2), several notable findings emerge. Firstly, the significance of the 'only_punctuation' feature is particularly surprising. This binary feature was developed based on the identification of numerous instances during exploratory data analysis (EDA) where sentences consisted solely of punctuation marks or special symbols, indicating the presence of junk sentences. Secondly, while the strong performance of the Gunning Fog Index, a real-world readability test, is expected, it is unexpected that the other two metrics, namely Flesch Reading Ease and SMOG Index, exhibit considerably less influence. Thirdly, it is noteworthy that three features appear to have negligible contribution when other similar features are present.

To further assess the impact of various feature categories, a feature ablation analysis was conducted (refer to Figure 3). All features (refer to Feature Appendix B) were grouped into seven categories, and one category was systematically removed to evaluate the resulting performance changes. These measurements were obtained through 5-fold cross-validation and averaged across the data splits. It is unsurprising that the removal of Named Entities had no effect, as both features associated with this category exhibited zero importance. However, the feature importance and ablation charts provide valuable insights into the model's behavior. It suggests that certain features may be redundant, as removing several of them simultaneously has minimal impact on the overall score. Alternatively, it could indicate that the distribution of features contributes relatively equally to the model's performance. Additionally, the absence of a highly dominant feature is evident, as the performance shows only marginal fluctuations of no more than a hundredth in either direction.



(Figure 2)



(Figure 3)

Based on the overall feature importance and the outcomes of the feature ablation analysis, it can be concluded that the majority of features contribute incrementally to the success of the model. As our focus was on optimizing the F1 score, the best-performing model prioritized maintaining high levels of recall and precision. However, it encountered challenges when dealing with borderline sentences, as discussed in the subsequent failure analysis.

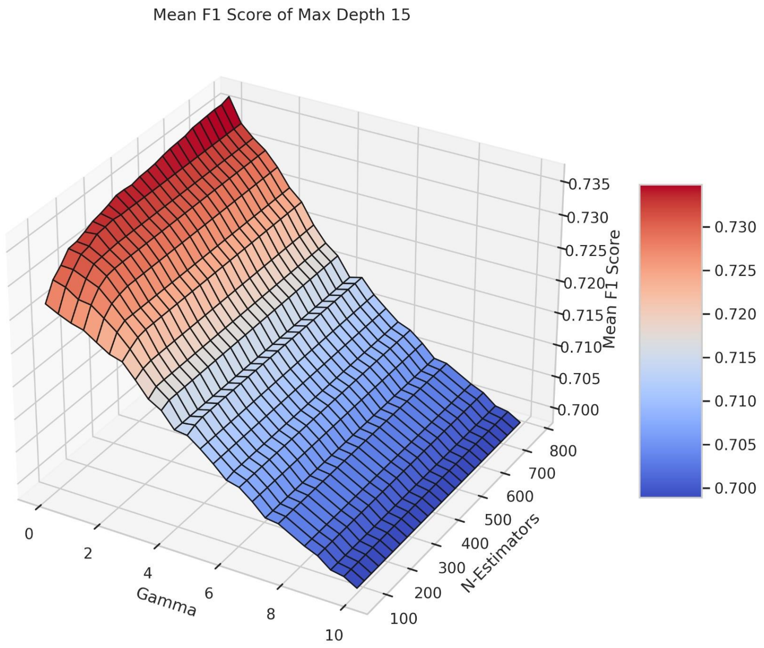
Given the substantial size of the training set, the model exhibited improved performance with a higher number of estimators and deeper tree depth. However, this preference for increased complexity resulted in longer training times, and the model's performance became sensitive to specific parameter settings.

Sensitivity Analysis

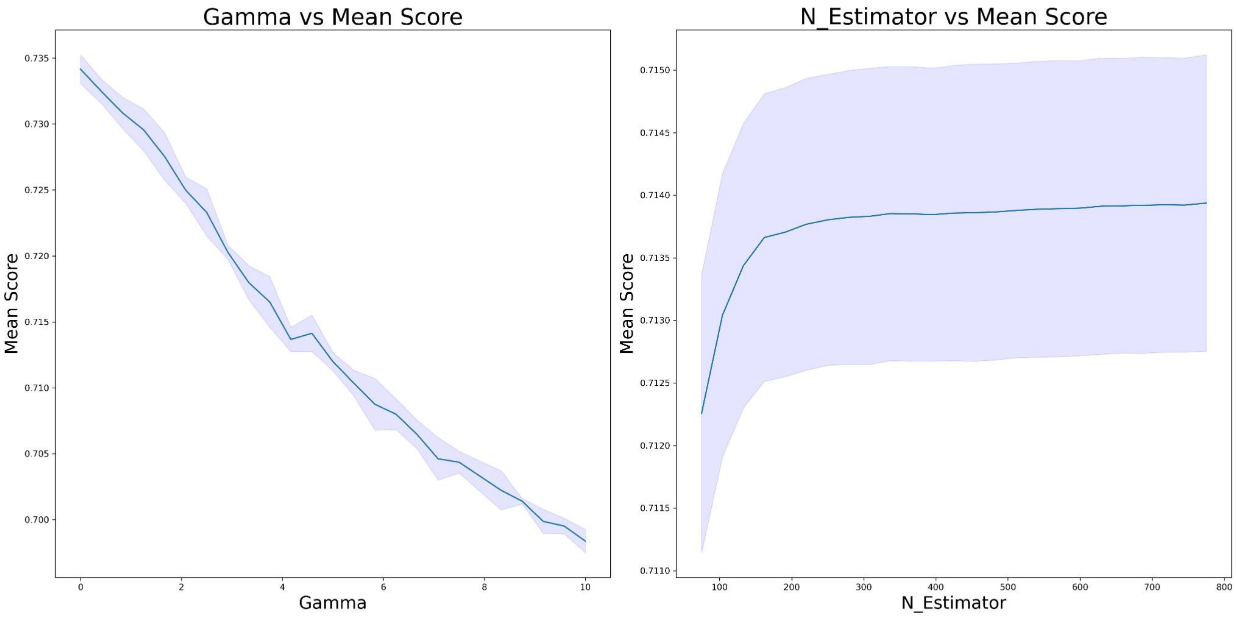
Sensitivity analysis was conducted by taking the mean F1 of 5-fold cross validation for each parameter. The 95% confidence interval is shown shaded around the line graphs.

In conducting the sensitivity analysis, several key observations can be made regarding the impact of various parameters on the F1 score. Specifically, increasing the number of estimators generally leads to an improvement in the score; however, it becomes evident that the returns diminish as the number of estimators continues to increase. This indicates that the initial gains in performance are more substantial compared to the incremental improvements achieved with additional estimators.

On the other hand, when exploring the effect of Gamma regularization, it is found that increasing the regularization parameter results in a sharp decrease in the F1 score. This suggests that a higher level of regularization imposes constraints that are too stringent for the model, negatively affecting its ability to accurately classify instances.

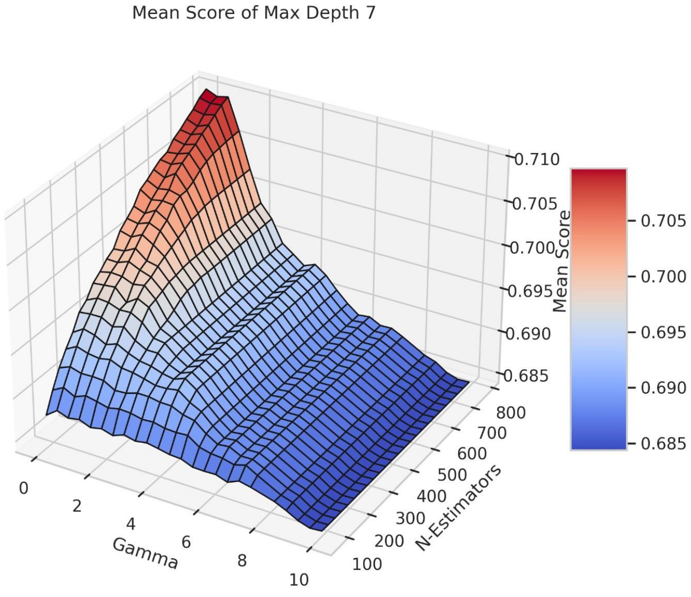


(Figure 4)

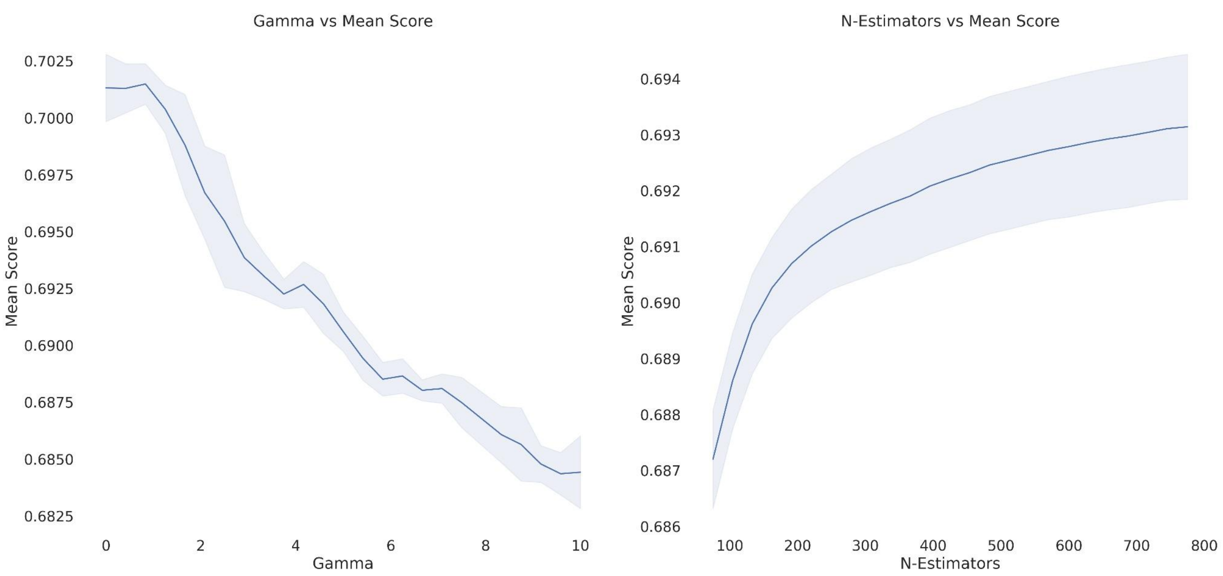


(Figure 5)

Furthermore, the maximum tree depth parameter was found to have a discernible impact on the model's performance. Specifically, the worst-performing models with a maximum depth of 15 exhibited performance similar to the best-performing models with a maximum depth of 7. This finding highlights the importance of carefully selecting the appropriate tree depth, as excessively deep trees may lead to overfitting or unnecessarily complex models.



(Figure 6)



(Figure 7)

Failure analysis

Here are three sentences, in their original form, that were incorrectly classified by our best gradient boosted model:

1. 'Some of the 1930s trams are still in regular service in virtually unchanged condition .'
2. 'In other countries , potassium iodate is used as source for iodine .'
3. 'If processors can read the same old value after the write made by P2 , the memory would not be coherent .'

Sentences 1 and 2 actual label is to be simplified, sentence 3 does not need to be simplified. Here are some potential reasons why our classifier got this wrong. As seen in figure 8, these sentences scored near the average training sentence in several features, but also had a mix of complex and simple sentence qualities. Both 1 and 2 have a low amount of noun subjects, punctuation marks and syllable counts, but have few concrete words and many unique words. This suggests short, diverse sentences with a mix of easy and difficult words are difficult to classify. Sentence 3 has high counts of short, easy, concrete words, but seems to be misclassified due to being longer than average with a high syllable count and a diverse vocabulary.

True Label		1	1	0
Predicted Label		0	0	1

	Average Sentence	**Sentence 1**	**Sentence 2**	**Sentence 3**
Important Features				
Gunning Fog	0.78	1	0	1
Number of Long	4.86	5	2	2

Words				
Punctuation Count	3.44	1	2	2
Noun Subjects	5.22	2	4	4
High AoA Count	1.42	2	2	2
Syllable Count	28.32	23	17	31

	Average Sentence	**Sentence 1**	**Sentence 2**	**Sentence 3**
Less Important Features				
Flesch Readability	1.11	2	1	2
Ratio Unique words	0.88	0.93	1	0.86
Dale Chall Count	10.9	8	7	15
Number of Words	21.84	15	13	23
Lexical diversity	4.01	4.19	3.8	4.06
concrete count	4.85	1	3	4

(Figure 8)

The mismatch between Flesch Readability and Gunning Fog categories also suggests that these sentences are probably right near the border of being classified correctly.

Future improvements might include binning for more features and increasing the number of bins to accommodate more nuance. For example Flesch Readability scores ranged from 0 to 120 and were binned into three categories to match with Gunning Fog and SMOG indexes which are based on how old someone should be to understand a sentence. It would also be useful to normalize the features that are just pure counts. Sentence length becomes highly correlated with these count only features. Leading to short sentences being seen as simple and long sentences being seen as complex in general.

Part B: Unsupervised Learning

Methodology

In this task, I employed two distinct unsupervised learning methods: Gaussian Mixture Model (GMM) and K-means clustering. These methods were chosen due to their different underlying mechanisms, allowing for the exploration of a diversity of results.

GMM, a probabilistic model, assumes that data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. This model is particularly useful when data clusters are not necessarily spherical, as it allows for different shapes of clusters through covariance matrices. I chose this approach due to its inherent flexibility and suitability for complex datasets. On the other hand, K-means is a non-probabilistic, centroid-based clustering algorithm that partitions the dataset into K clusters. Each observation belongs to the cluster with the nearest mean. This technique is simple, easy to implement, and computationally efficient compared to GMM, making it a pragmatic first step.

For feature representations, a comprehensive feature engineering process was applied. Feature selection was based on domain knowledge, exploratory data analysis, and previous research, offering a better understanding of text complexity.

Hyperparameters for both models were tuned using Grid Search, an exhaustive search over specified parameter values. For the GMM, the 'covariance_type' parameter was tuned, while parameters like 'init' and 'n_init' were explored for the K-means model. This systematic approach of running the models with different parameters enabled the selection of the set that delivered the best performance.

The evaluation of the clustering methods was carried out using the labels already provided in the dataset, indicating whether the text was 'easy' or 'difficult'. This semi-supervised evaluation served as a measure of how well each model performed in identifying text difficulty.

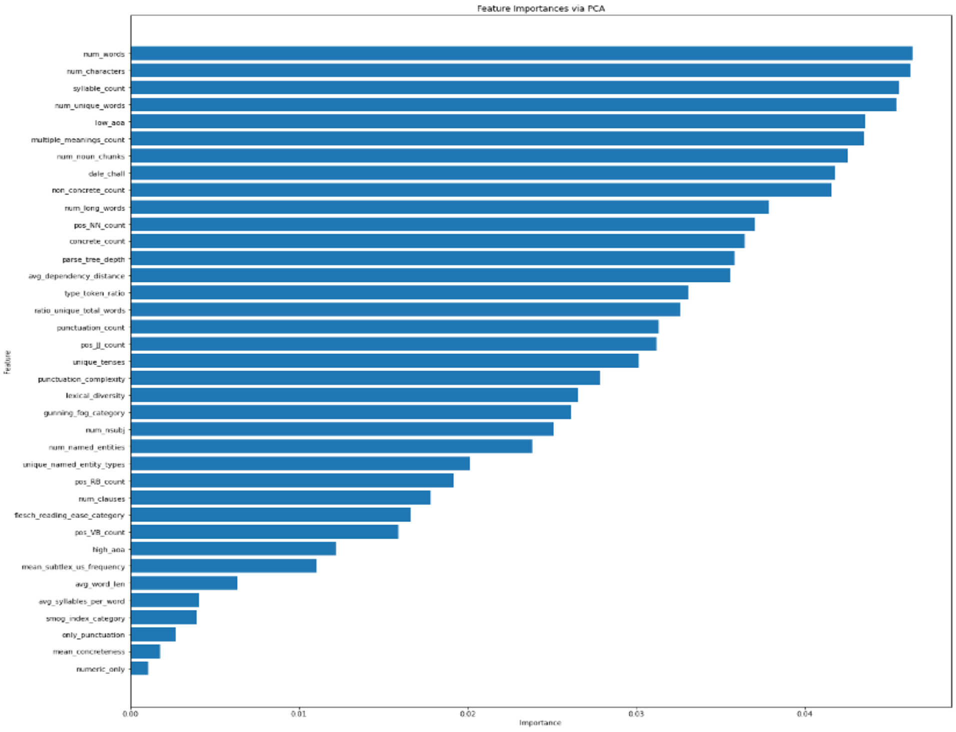
Finally, an important aspect of this task was determining the optimal number of clusters ('n_clusters') for each model. To do this, I used the Elbow Method, a technique where the total intra-cluster variance (or total within-cluster sum of square (WSS)) is explained as a function of the number of clusters. The 'elbow' point - the point of inflection on the curve - provides a good indication of the optimal number of clusters. This approach allowed for an informed decision on the 'n_clusters' parameter, contributing to the overall effectiveness of the unsupervised learning task.

Unsupervised Evaluation

Result and Sensitivity Analysis

For the text difficulty detection task, two unsupervised learning methods were employed: K-means clustering and the Gaussian Mixture Model (GMM). The evaluation of these methods' effectiveness was based on how closely the clusters they created matched with the existing labels of 'easy' or 'difficult' in the dataset. This semi-supervised evaluation approach allowed us to use classification accuracy as a metric, despite using unsupervised methods.

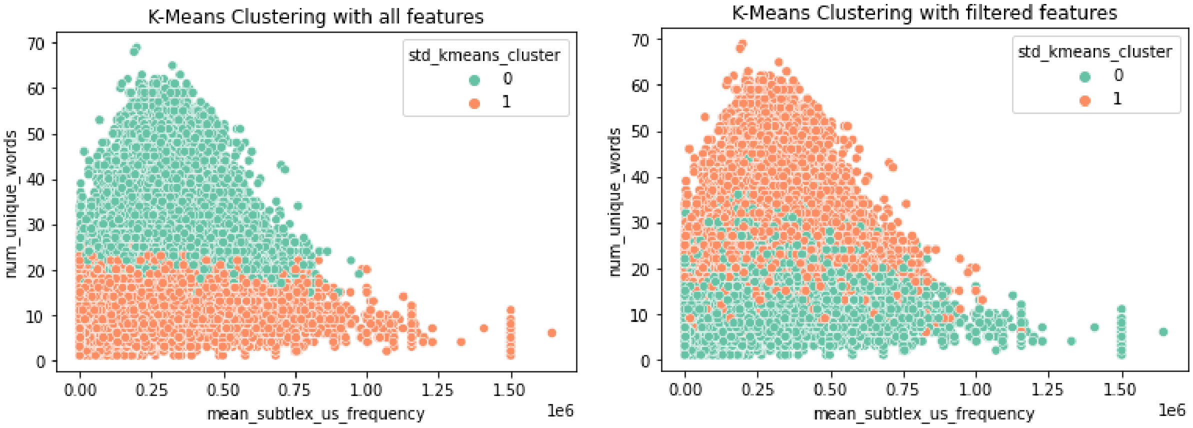
The initial application of the K-means clustering algorithm with all the pre-processed features resulted in a classification accuracy of 0.3944. The relatively low accuracy prompted an in-depth feature analysis using Principal Component Analysis (PCA) for dimensionality reduction and visualizing feature importance. Notably, the 'num_words' and 'num_characters' features demonstrated significant influence on accuracy.



(Figure 9)

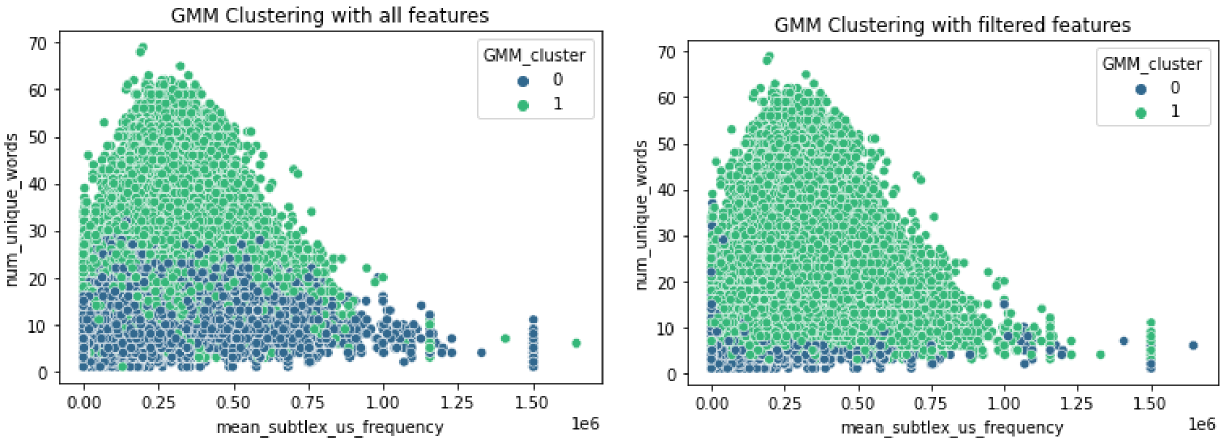
To understand the impact of each feature, individual features were omitted and the model accuracy was measured. The results were in the appendix C, table1.

A pattern emerged when omitting certain features improved model accuracy. Features containing “num”, “count”, “len”, “ratio”, “unique”, “dale_chall”, and “low” were then excluded. The resultant accuracy improved to 0.6455, suggesting that the complexity of the text was not determined by factors like the length of words or the number of characters.



(Figure 10)
As you can see the figures, compared to the accuracy between filtered features and all the features, the model with filtered features was yielded. Also, the cluster shows the completely opposite result. Therefore, some features led to different clustering results. It appears that the complexity of the text is not determined by either the length of words or the number of characters.

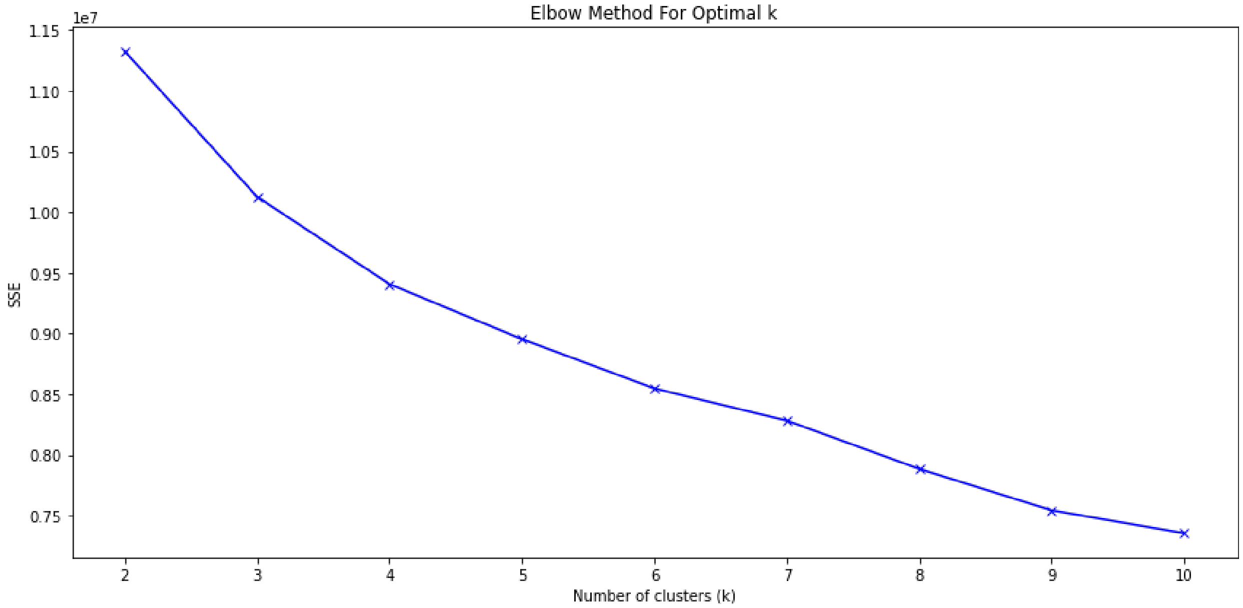
The GMM was subsequently implemented, producing an initial accuracy of 0.5809 with all features. A similar feature exclusion process was conducted, yielding an accuracy of 0.5853, further validating our feature selection approach.



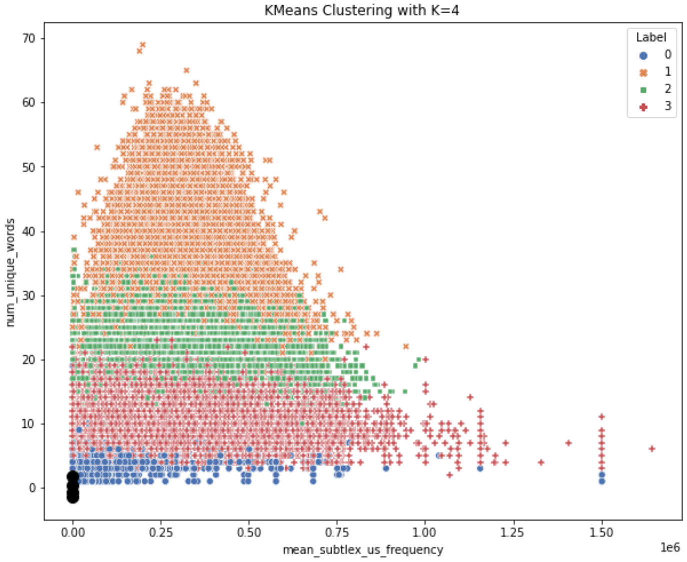
(Figure 11)

We took KMeans for hyper-parameter sensitive analysis and the result was in appendix C, table 2. There was multiple best models and it was pretty stable with hyper-parameters.

To further optimize these models, the number of clusters was varied, and the within-cluster sum of squares (WCSS) was calculated for each number. The Elbow method was used to identify the optimal number of clusters. For K-means, the decline in the WCSS value appeared to slow after k=4, suggesting that this might be the optimal number of clusters for this dataset.



(Figure 12)
The visualizations presented in this analysis include feature importance plots and elbow method plots, offering diverse visual insights into the unsupervised learning process. This analysis offers a comprehensive evaluation of the unsupervised learning methods used for the text difficulty detection task. Through strategic feature selection, model choice, and parameter tuning, substantial improvements in model performance were achieved. From the values, it seems like the decline in the WCSS value is slowing down after k=4 (8952938.737396594). The differences between subsequent WCSS values from k=5 onwards are getting smaller.



To calculate the silhouette score, the computation is too heavy, so we set the sample size as 50000. The average silhouette score was 0.208848 and 0.2259 in 2 clusters and 4 clusters, respectively. So, the 4 clusters worked better in this dataset. The text difficulty criteria may make more sense when it has more levels.

(Figure 13)

Discussion

Part A: Supervised Learning

We gleaned several key insights from working on the supervised learning side. First and foremost, working with natural language data, even on a large scale, presents a unique set of challenges. Despite the use of advanced features derived from NLTK and SpaCy libraries, the enhancement of the f1 score was marginal and in certain instances, even regressive. This experience shed light on the complex dynamics involved in feature selection, and how sometimes, the seemingly useful features might not contribute to performance as expected.

One surprising outcome was the comparable performance of Logistic Regression to Gradient Boosting, specifically when hyperparameters were not tuned. This unexpected observation emphasized that simpler models can, at times, closely compete with their more complex counterparts, a factor to be considered when aiming for efficiency.

We grappled with several challenges, most notably handling the large training set. Our attempts to incorporate tf-idf and BERT weights were impeded due to the large vocabulary, leading to prohibitive computational requirements. Adding thousands of tf-idf columns seemed to overshadow the contribution of other features. However, efforts to reduce their dimensionality via PCA or SVD compromised their effectiveness - a classic case of the curse of dimensionality.

To overcome the challenge posed by large computational requirements, we employed the Great Lakes computing cluster and divided the workload to ensure efficiency. We also had to compromise on feature selection and reduce the scope of the project to accommodate the extensive model training time.

Interestingly, some unexpected findings highlighted the nuances inherent in text data. The influence of punctuation as a feature came as a surprise, particularly when compared to seemingly more sophisticated features like lexical diversity. These unexpected insights underscored the necessity to approach feature selection in NLP tasks with both thoroughness and flexibility, open to the potential importance of less intuitively significant features.

In terms of future developments, with additional time and resources, we would love to explore the capabilities of neural networks, given our belief that the large dataset could generate a significantly higher performing classifier.

Part B: Unsupervised Learning

In undertaking Part B of this project, I gained a deeper understanding of the nuances and complexities inherent in unsupervised learning. One key lesson was the critical role of feature selection, which can greatly impact model performance. Contrary to initial expectations, I found that certain features, while seemingly essential, were actually detrimental to the performance of the clustering algorithms. This was a surprising revelation that underscored the importance of thorough feature analysis and validation in machine learning tasks. Furthermore, this project served as a stark reminder that unsupervised learning is an iterative and exploratory process rather than a linear one. For example, identifying the optimal number of clusters in K-means proved to be a crucial yet challenging task. This process required a delicate balance between numerical and visual analysis, emphasizing the necessity of diverse approaches in model optimization.

Assessing the performance of unsupervised learning models presented another challenge, given the absence of a clear ground truth. Our approach involved using available labels in a semi-supervised evaluation, which yielded effective results. However, this also highlighted the need for more sophisticated evaluation methods, particularly in scenarios where such labels might not be available. Identifying the optimal number of clusters for K-means was another significant hurdle. The challenge lay in avoiding oversimplification on one hand and overfitting on the other. The Elbow method was a valuable tool in navigating this challenge, though it did bring attention to the inherently subjective nature of this decision-making process.

Further complicating the process were computational limitations. Despite the Silhouette score being a useful metric for assessing unsupervised learning performance, the computational demands associated with it made iteration and fine-tuning difficult. Looking to the future, this work could be extended in several ways with more time and resources. For instance, I'm particularly interested in exploring deep learning methods, such as autoencoders. These methods could offer more meaningful feature extraction, potentially capturing more complex patterns in the data. Thus, even as this project drew to a close, it opened up a wealth of possibilities for further exploration and learning in the field of unsupervised learning.

Ethical Considerations

Part A: Supervised Learning

1. **Bias**

Machine learning models learn from the data they are given, and if this data contains biases, the models may learn and perpetuate these biases. In the context of text difficulty classification, biases could be introduced if the training data doesn't adequately represent various levels of difficulty, styles, genres, or dialects. For instance, if the dataset includes more examples of certain difficulty levels or favors a certain dialect, the model could be biased towards those. Ensuring training data is from a diverse and truly random sample of text would mitigate this.

2. **Transparency and Explainability**

It's important to ensure the classification models are transparent and their decision-making process can be explained. Users should understand why a text is labeled a certain way, and there should be a process to challenge or question the system's decisions. A more complex model may have a better F1 score, but might appear as a black box to any users. A more general model that uses less features might sacrifice accuracy, but increase explainability.

Part B: Unsupervised Learning

1. **Data Bias**

Unsupervised learning models identify patterns and structures within data. If the input dataset is biased in terms of its content, style, or language, the model will inherit these biases, which may result in skewed or unfair classifications of text difficulty. For example, if the dataset predominantly contains academic or scientific articles, the model might categorize relatively straightforward non-academic texts as 'difficult' due to the bias in the training data. Ensuring diversity in the input dataset is critical to address this concern. The dataset should be representative of various domains, styles, and complexity levels of language to avoid bias in the model.

2. **Lack of Interpretability**

Unsupervised learning models often suffer from a lack of interpretability, which can be ethically concerning. It may be difficult to justify why a specific text was categorized as 'difficult' or 'easy', and there might be a risk of unfair or inconsistent categorization due to the black-box nature of many unsupervised algorithms. Using explainable models or leveraging methods to increase the interpretability of the model can help address this issue. Sensitivity analysis and feature importance can provide insights into the model's decision-making process.

3. **Accessibility**

As the model categorizes texts into difficulty levels, there could be a risk of exacerbating accessibility issues if it is used by a search engine. For instance, individuals with lower reading proficiency could be not exposed to certain texts deemed 'too difficult'. The model should be used to enhance accessibility rather than limit it. Its primary function should be to aid in tailoring content to suit individual proficiency levels or to provide additional support for complex texts rather than limiting access to information.

Statement of Work

Ryan Eskuri: Feature Engineering, Supervised Models, Sensitivity + Error Analysis, Visualization, Final Report
Zhentao Wang: Feature Engineering, Supervised Models, Grid Search, Cross-validation, Final Report
Hank Ye: Unsupervised Learning Modeling, Visualization, Code Review, and Final Report

References

Readable. (n.d.). Flesch reading ease & Flesch-Kincaid grade level. Retrieved May 23, 2023, from https://readable.com/readability/flesch-reading-ease-flesch-kincaid-grade-level/

Readable. (n.d.). SMOG Index. Retrieved May 23, 2023, from https://readable.com/readability/smog-index/#:~:text=What%20is%20a%20SMOG%20Index,of%2030%20sentences%20or%20more.

Readable Formulas. (n.d.). Gunning Fog Readability Formula. Retrieved May 25, 2023, from https://readabilityformulas.com/gunning-fog-readability-formula.php

ScienceDirect. (n.d.). Age of Acquisition. In Encyclopedia of Psychology. Retrieved May 18, 2023, from https://www.sciencedirect.com/topics/psychology/age-of-acquisition

Data Accessed From:

https://www.kaggle.com/competitions/umich-siads-696-ss23-predicting-text-difficulty

Feature Appendix A

1. Basic Text Features
    - num_words: Number of words in the text
    - num_characters: Number of characters in the text
    - num_unique_words: Number of unique words in the text
    - ratio_unique_total_words: Ratio of unique words to total words
    - num_long_words: Number of long words in the text
    - syllable_count: Total count of syllables in the text
    - avg_syllables_per_word: Average number of syllables per word
    - lexical_diversity: Measure of lexical diversity in the text
    - type_token_ratio: Ratio of unique words to total words (another measure)
    - multiple_meanings_count: Number of words with multiple meanings
    - numeric_only: Presence of numeric values only
    - only_punctuation: Presence of punctuation marks only
    - punctuation_count: Total count of punctuation marks in the text
2. Part-of-Speech (POS) Features
    - pos_NN_count: Number of nouns (NN) in the text
    - pos_VB_count: Number of verbs (VB) in the text
    - pos_JJ_count: Number of adjectives (JJ) in the text
    - pos_RB_count: Number of adverbs (RB) in the text
3. Readability Features
    - flesch_reading_ease_category: Binned Flesch Reading Ease score
    - gunning_fog_category: Binned Gunning Fog Index score
    - smog_index_category: Binned of SMOG Index score
4. Linguistic Features
    - Dale_chall: Number of words from Dale Chall list
    - high_aoa: Number of words with Age of Acquisition (AoA) above 12
    - low_aoa: Number of words with Acquisition (AoA) of 12 or below
    - concrete_count: Number of concrete words in the text
    - non_concrete_count: Number of non-concrete words in the text
    - mean_concreteness: Mean concreteness score of the words
    - mean_subtlex_us_frequency: Mean SUBTLEX-US frequency of the words
    - parse_tree_depth: Depth of the parse tree
    - num_clauses: Number of clauses in the text
    - num_noun_chunks: Number of noun chunks in the text
    - num_named_entities: Number of named entities in the text
    - avg_word_len: Average word length in the text
    - num_nsubj: Number of subject nouns in the text
    - punctuation_complexity: Complexity of punctuation usage
    - unique_tenses: Number of unique verb tenses used
    - unique_named_entity_types: Number of unique named entity types
    - avg_dependency_distance: Average dependency distance between words

Feature Appendix B

Text Characteristics:

- num_words
- num_characters
- num_unique_words
- ratio_unique_total_words
- num_long_words
- avg_word_len
- lexical_diversity
- type_toke_ration

Linguistic Complexity:

- syllable_count
- avg_syllables_per_word
- multiple_meanings_count
- parse_tree_depth
- num_clauses
- num_noun_chunks
- avg_dependency_distance
- punctuation_complexity
- unique_tenses

POS Counts:

- pos_NN_count
- pos_VB_count
- pos_JJ_count
- pos_RB_count
- num_nsubj

Named Entities:

- num_named_entities
- unique_named_entity_types

Readability Metrics:

- flesch_reading_ease_category
- gunning_fog_category
- smog_index_category
- dale_chall

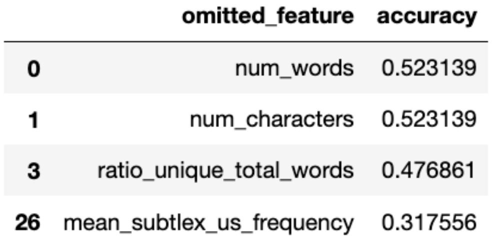
Word Categories Complexity:

- high_aoa
- low_aoa
- concrete_count
- non_concrete_count
- mean_concreteness
- mean_subtlex_us_frequency

Punctuation:

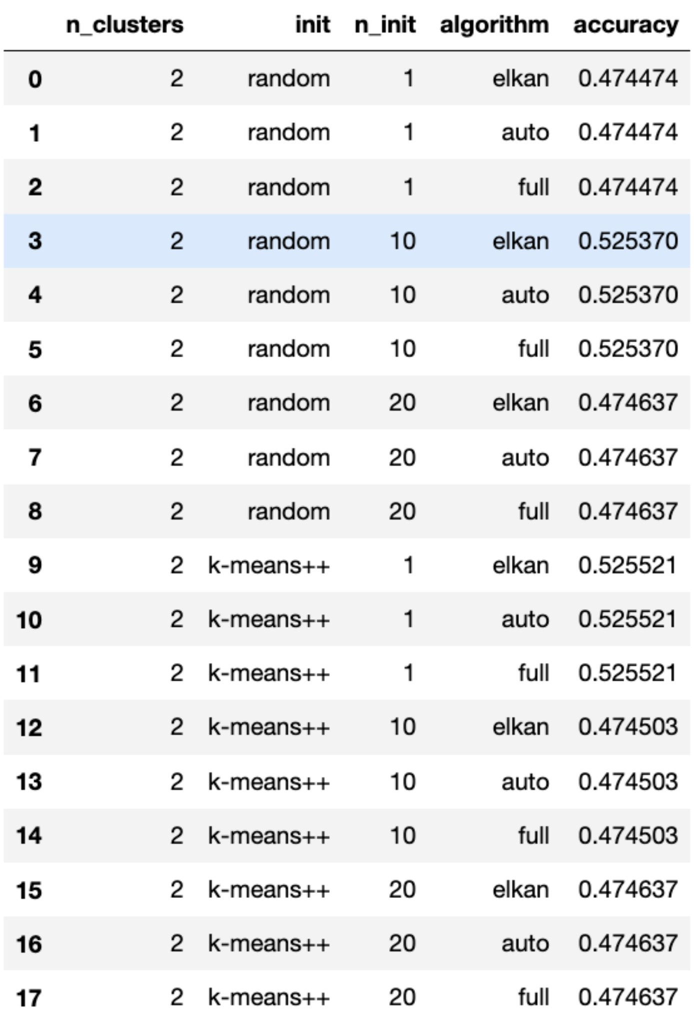
- only_punctuation
- punctuation_count

Unsupervised Learning Part Appendix C



	omitted_feature	accuracy
0	num_words	0.523139
1	num_characters	0.523139
3	ratio_unique_total_words	0.476861
26	mean_subtlex_us_frequency	0.317556

Tabel1: Accuracy and the feature is omitted when fitting the KMeans model



	n_clusters	init	n_init	algorithm	accuracy
0	2	random	1	elkan	0.474474
1	2	random	1	auto	0.474474
2	2	random	1	full	0.474474
3	2	random	10	elkan	0.525370
4	2	random	10	auto	0.525370
5	2	random	10	full	0.525370
6	2	random	20	elkan	0.474637
7	2	random	20	auto	0.474637
8	2	random	20	full	0.474637
9	2	k-means++	1	elkan	0.525521
10	2	k-means++	1	auto	0.525521
11	2	k-means++	1	full	0.525521
12	2	k-means++	10	elkan	0.474503
13	2	k-means++	10	auto	0.474503
14	2	k-means++	10	full	0.474503
15	2	k-means++	20	elkan	0.474637
16	2	k-means++	20	auto	0.474637
17	2	k-means++	20	full	0.474637

Table 2: Sensitive analysis for hyperparameter