Automated Essay Grading System

Harpal Kaur Dhindsa[1], *Dr. Saravanagaru RA K*[2]

[1] M.Tech CSE with Big Data specialization ,SCOPE, VIT University

[2]Associate Professor ,SCOPE, VIT University

Abstract

At present essays are a significant piece of student assessment. These days , essays are in effect generally used to measure understudies in different undergrad/graduate level assessments. Assessment and reviewing of essays comes as a significant issue with human assessment being moderate , dreary , time taking in this manner utilizing automated framework is essential. Computerized assessment is progressively being utilized in classrooms and online tests. In this project, I attempt to comprehend the essays and their assessment models utilized for manual assessment and endeavours to foster comparative algorithmic frameworks. I investigate highlights and methods to distinguish and separate articles and use of models dependent on AI calculations to review these expositions. The objective of automated essay grading is to allot evaluations to papers and give feedback utilizing computer. The point of this undertaking is to foster AI models for performing automated essay scoring and assess their performance. I utilize a freely accessible essay data-set to prepare and test the viability of the embraced algorithms. Natural language processing methods are utilized to extract features from essays in the data-set. ML models are calculations were utilized on the picked data-set.

Keywords: TF-IDF, LDA, Automated grading, Hybrid method

Introduction

Reviewing student essays is an exorbitant task, both as far as time and cash for schools and instruction loads up. For this task, I try to construct models that consequently grade essay text dependent on different attributes. I worked with 12,978 essays across eight diverse essay sets and find that models dependent on straightforward factors, for example, paper length and scope of vocabulary perform shockingly well. I further investigated extra highlights, for example, tf-idf and LSA to accomplish better model execution.

Essays permit college students to specific thoughts and evaluations consequently are better suitable over direct true/false questions. They help check the understanding and the higher-order-questioning talents of college students. Hence they're broadly used way of assessing students , utilized by universities for enrollments. As manually grading the essay is costly , tedious and time consuming , it becomes vital to apply computerized structures to make the assessment efficient and permit quicker and less complicated grading of the essays. Automated

Essay Grading is a technique of comparing and grading essays the usage of computer programs. These structures are used in lots of excessive stake tests like GRE (Graduate Record Examination) , TOEFL (Test of English as a Foreign Language) , GMAT (Graduate Management Admission Test).These structures observe a supervised method and construct a version yielding first-rate of holistic grading similar to that of people. These structures examine the grading version primarily based totally on a data-set of graded essays through people. The dataset may also have resolved scores (combination of essay scoring accomplished through a couple of people) or scores from man or woman human graders. These structures employ positive properties of essays (known as features) a good way to estimate essay quality. In the grading technique , those features are extracted and scores assigned in step with manifestation of those features. Many forms of features were proposed for essay grading. They rely on degree of dependency to task , weakly based have a tendency to be topic independent like length , connection , style even as strongly dependent have a tendency to observe the particular essay just like the vital phrases associated with a subject matter. Association needs to be set up among features and scoring values to estimate accurate manifestation of these. Here , our goal to create a machine that grades the essays consistent with techniques and orientations of people and obtain an efficient framework of grading the essay replacing the people as a result leading to greater efficient work.

Literature Survey

Sr no	Year & Publication	Title	Authors	Keywords	Advantages	Disadvantages
1	2020 IEEE	“Experimenting with Latent Semantic Analysis and Latent Dirichlet Allocation on Automated Essay Grading”	Jalaa Hoblos	Latent Semantic Analysis, Latent Dirichlet Allocation , Semantic Similarity	The results shows relatively high correlation between the grades it returns, and the professor assigned grades. Preliminary results yield better accuracy when using LSI based technique.	The dataset tested on 118 essays, it is premature to adopt the tool as a substitution for human grader.

2	2020 IEEE	“Automated English Digital Essay Grader Using Machine Learning”	Yafet Salim, Valdi Stevanus, Edwardo Barlian, Azani Cempaka Sari, Derwin Suhartono	XGBoost, NLTK, spaCy	Feature selection is very important, since it allows one to analyze if that particular feature makes the accuracy better or not.	Large number of features do not guarantee a high level of accuracy because not all features present an essay or the pattern of the feature can be made with little training data
3	2020 CVIDL	“Deep Automated Text Scoring Model Based on Memory Network”	Shiyan Yang	LSTM, Neural Network, Memory Network, Deep Learning	It is an effective method in reducing the amount of human work and graders only need to set the weight matrix before embedding the standard answer.	The TF-IDF weight itself is not enough for concentration decision. Although the keywords exists, the relation of the keywords may vary and it is hard for the system to understand under difficult writing styles
4	2020 IEEE	“An Analysis of Automated Answer Evaluation Systems based on Machine Learning”	Birpal Singh J. Kapoor, Shubham M. Nagpure, Sushil S. Kolhatkar, Prajwal G. Chanore, Mohan M. Vishwakarma, Prof.	String Similarity, Content-Based Similarity	The assessment out comes proves cosine and LSA have the outstandingly forceful outcomes.	

			Rohan B. Kokate			
5	2019 ICECOS	"Term Frequency -Inverse Document Frequency Answer Categorization with Support Vector Machine on Automatic Short Essay Grading System with Latent Semantic Analysis for Japanese Language"	Anak Agung Putri Ratna, Aaliyah Kaltsum, Lea Santiar, Hanifah Khairunissa, Ihsan Ibrahim, Prima Dewi Purnamasari	TF-IDF, LSA, SVM	SVM with linear kernel generates the highest accuracy. The relation between penalty value and SVM accuracy score is that the greater amount of penalty score, the higher the accuracy scores given.	With a tiny value of penalty, the chance of getting misclassified examples are greater that makes the lesser amount of penalty will affect the accuracy score.

6	2018 CCIS	"Intelligent Auto-grading system"	Zining Wang, Jianli Liu, Ruihai Dong	Deep Learning, Nueral Network	The attention visualization successfully proves the interpretability of the scoring standard of the system. The scoring system can learn a pretty reasonable scoring mechanism which maps the essay vectors to the correct levels.	The test data needs more variety. It is uncertain that the models can also work well for the other types of subjective questions.

System Architecture

The system architecture consists of the modules, as shown in the figure below.

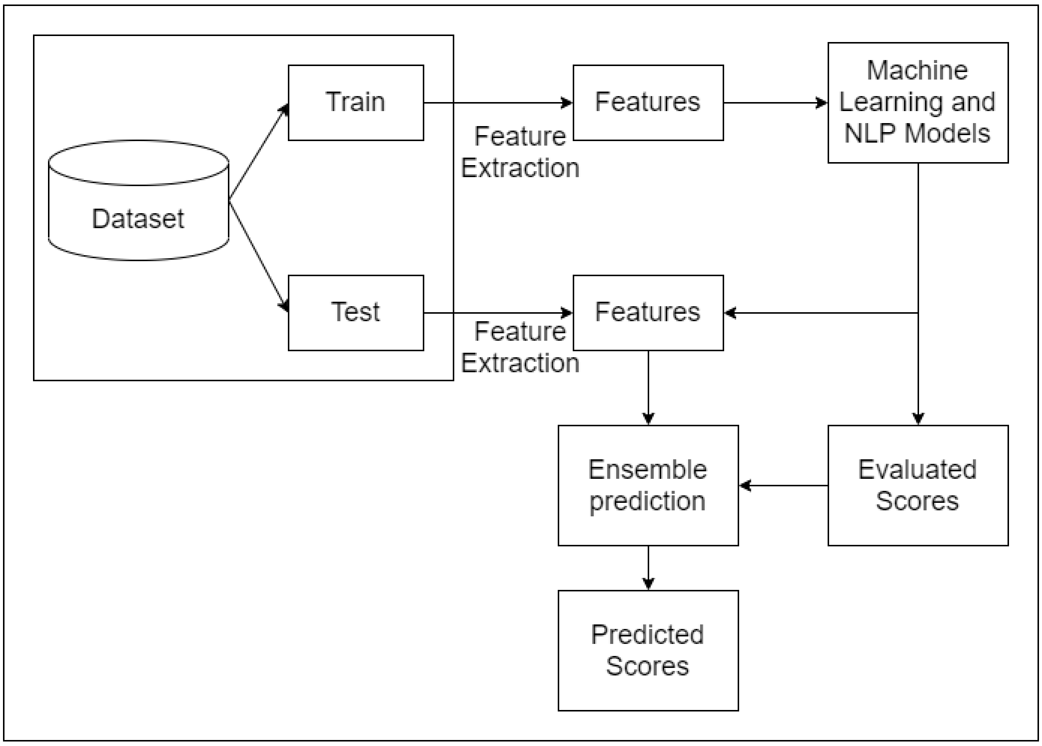


Figure 1. System Architecture

The system consists of three main modules:

1. Cleaning and preparing data for modelling
2. Building the following models (regularization by using Cross validation):
   a. Linear Regression
      i. Lasso
      ii. Ridge
   b. Random Forest Classifier
   c. Random Forest Regressor
   d. Latent semantic analysis (LSA) and KNN
   e. TF-IDF and Naïve Bayes Multinomial NB
   f. Ensemble Prediction consisting of all the models listed above
3. LDA Visualizations, and WordClouds

Implementation

Data:

The Hewlett Foundation Kaggle dataset. Here I have worked with the essay information, consisting of 12,978 papers that were composed by students from Grade 7 to Grade 10 across eight diverse exposition prompts.

Pre-processing and feature extraction

Essay grading requires to assess the error done by the students and to get the desired results the essays cannot be pre-processed on a lot of factors to maintain the originality of the essay , so only few steps are being taken into account for processing in the following order:

1. Lowercase: All of the words are changed to lowercase so that during part of speech tagging the model doesn't change the meaning on different instances.
2. Special Characters are removed : Every special character except the '@' symbol is removed from the essay as it is of no use in the essay assessment and featurization.
3. Removal of Stopwords: This processing step is used while correcting the essay as you can read in the next section after correcting the essay grammar then this step is used to identify and remove stopwords.
4. Lemmatization: The essay after correcting the errors and removal of stop words is lemmatization. The lemmatizer converts words to the base or dictionary form of the word, which is also known as the lemma. For instance, the word "brought" is converted to its lemma "bring". The lemmatizer by default considers each word as a noun, so I applied the lemmatizer twice, first the lemmatizer considers each word as a verb and then considering the default as the noun option.
5. Tokenize using Count Vectorizer and TF-IDF Vectorizer: The corrected essays after the extraction of the prior stated features was then after lemmatizing. It considers the importance of the word in the corpus and increases as the number of times that word occurs in the document.
6. I now create some new variables to include in our model:

   - number of sentences in each essay

- number of words in each essay
- number of clean words in each essay
- number of unique words in each essay
- misspelling rate (number of misspelled words / total wordcount)
- average sentence length
- number of characters in the longest word used in each essay
    - Here, the rationale is that longer words will be correlated with better overall essay quality, so I take the length of the longest word that is used in each essay. In further models, I can expand this variable by looking at the average number of characters of words or the proportion of words that are longer than some number of characters.
- number of nouns, adjectives, and verbs in each essay
- proportion of nouns, adjectives, and verbs in each essay
- component weight 1,2,3 for each essay set using LSA

7. Similarity Score using LSA

Building Models:

The below figure shows the correlation between the features.

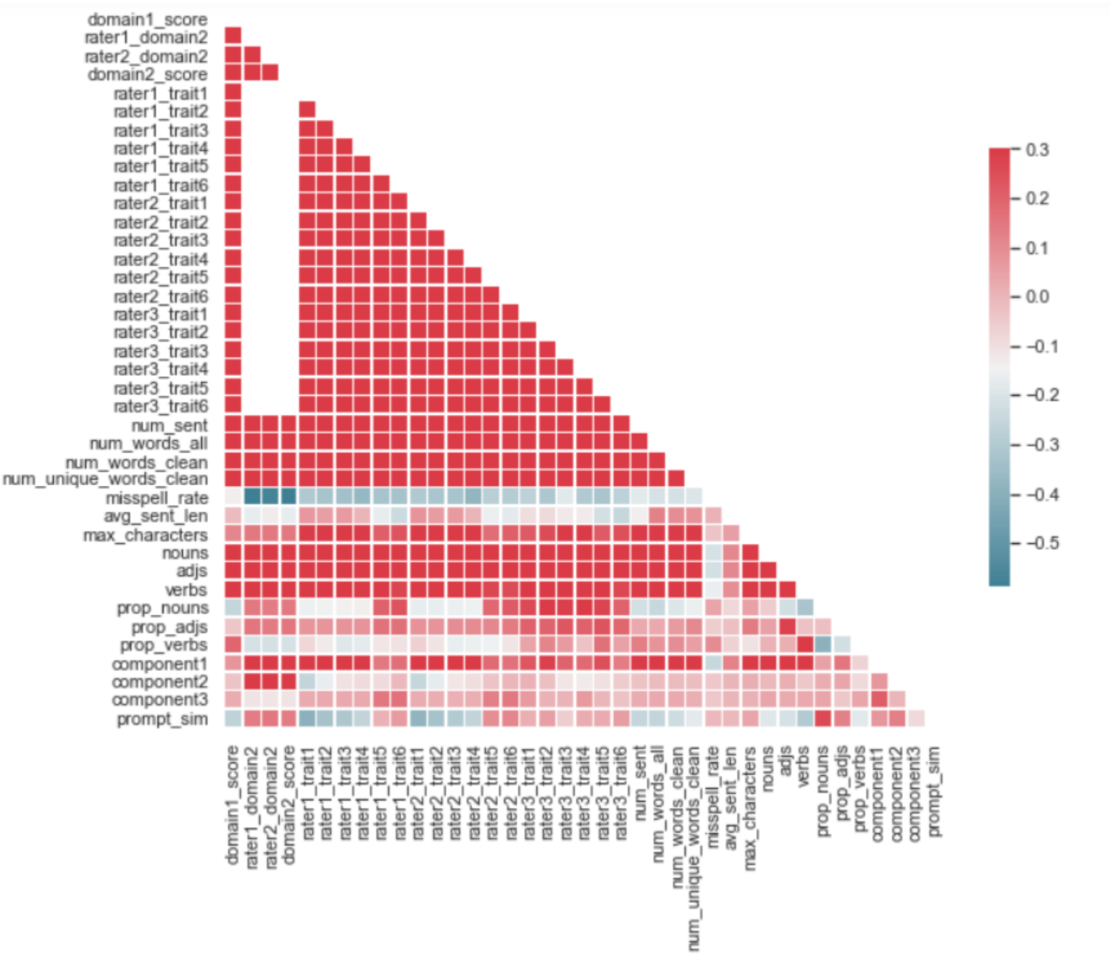


Figure 2. Feature Correlation plot

The below figure shows the count plot for word count, sentence count, and misspelling rate count.

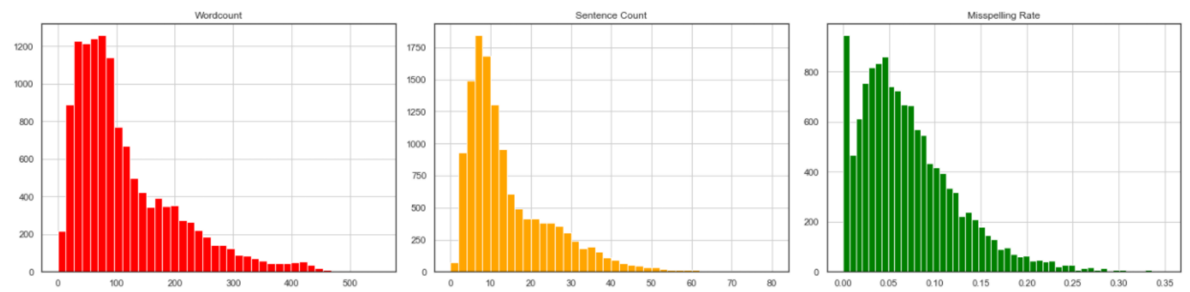


Figure 3. Count plots

The below figure shows the box plot for domain score 1 and domain score 2.

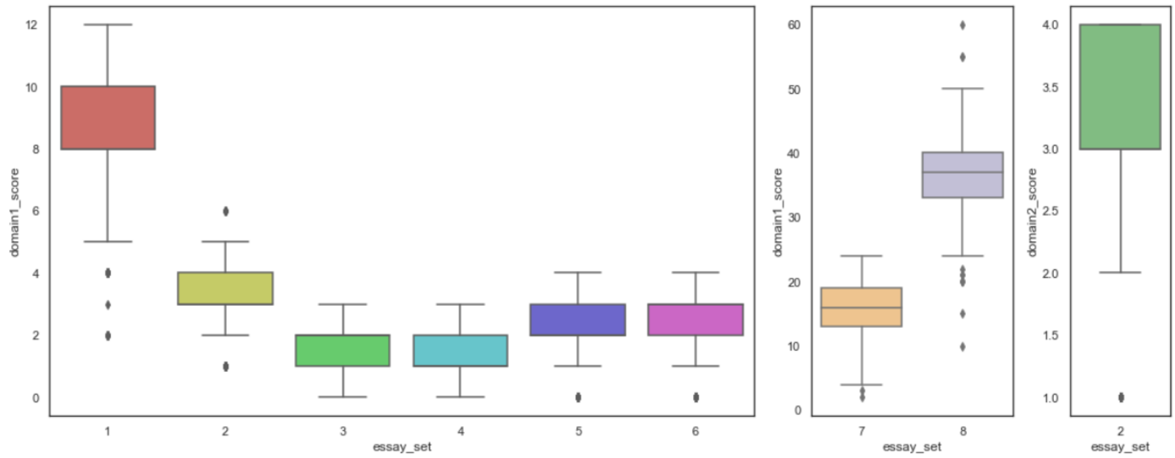


Figure 4. Box plot

A. Linear Regression is applied for essay sets 1, 7 and 8.

Lasso and Ridge are important for the Linear Regression family where the x and y are expected to have a linear relationship. The fundamental contrast among them is whether the model is penalized for its weights.

In linear regression the model isn't penalized for its selection of weights. That implies, during the training stage, if the model feels like one specific element is especially significant, the model may put a huge weight to the element. This occasionally prompts over-fitting in small datasets. Consequently, following techniques were presented.

Lasso is an adjustment of linear regression, where the model is penalized for the amount of total upsides of the weight. Hence, the total upsides of weight will be diminished, and many will in general be zeros. During preparing, the target function becomes:

$$\frac{1}{2m}\sum_{i=1}^{m}(y - Xw)^2 + alpha\sum_{j=1}^{p}|w_j|$$

Lasso presents another hyper-parameter, alpha, the coefficient to penalize weights.

Ridge makes a stride further and penalizes the model for the amount of squared worth of the weights. Accordingly, the weights will in general have more modest outright qualities, yet additionally truly will in general penalize the limits of the weights, bringing about a gathering of weights that are all the more uniformly disseminated. The target function becomes:

$$\frac{1}{2m}\sum_{i=1}^{m}(y - Xw)^2 + alpha\sum_{j=1}^{p} {w_j}^2$$

B. Random Forest Classifier

Random forest, similar to its name infers, comprises of countless individual choice trees that work as an ensemble. Every individual tree in the irregular forest lets out a class forecast and the class with the most votes turns into the model's expectation.

The major idea driving random forest is a straightforward however amazing, the wisdom of groups. The explanation that the random forest model functions admirably is, an enormous number of generally uncorrelated models (trees) working as an advisory group will beat any of the individual constituent models.

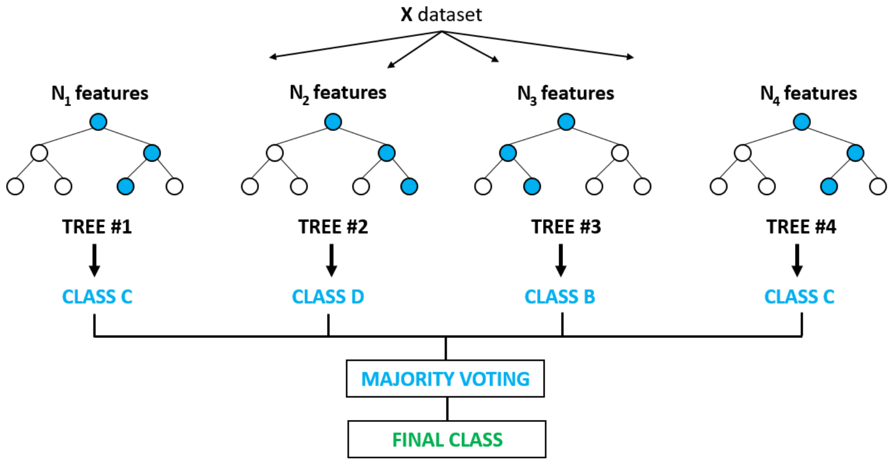


Figure 5. Random Forest Classifier

C. Random Forest Regressor

The Decision Tree is an effectively perceived and deciphered calculation and subsequently a solitary tree may not be sufficient for the model to take in the highlights from it. Then again, Random Forest is likewise a "Tree"- based calculation that uses the characteristics highlights of various Decision Trees for deciding.

Therefore, it very well may be alluded to as a 'Forest' of trees and consequently the name "Random Forest". The term 'Random' is because of the way that this calculation is a forest of 'Randomly made Decision Trees'.

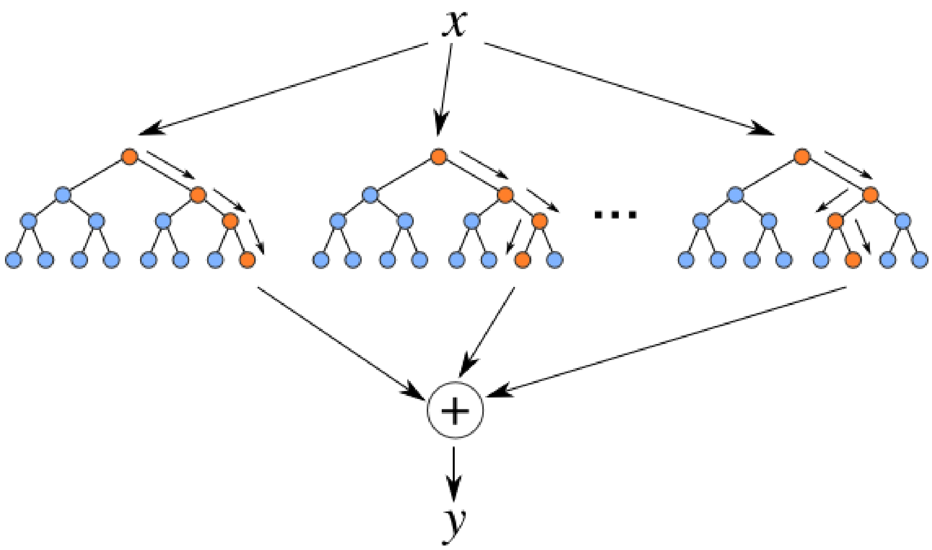


Figure 6. Random Forest regressor

The Decision Tree calculation has a significant weakness in that it causes over-fitting. This issue can be restricted by executing the Random Forest Regression instead of the Decision Tree Regression. Furthermore, the Random Forest calculation is likewise extremely quick and powerful than other regression models.

D. LSA and KNN

LSA is an information retrieval technique which analyses and identifies the pattern in unstructured collection of text and the relationship between them. LSA itself is an unsupervised way of uncovering synonyms in a collection of documents.

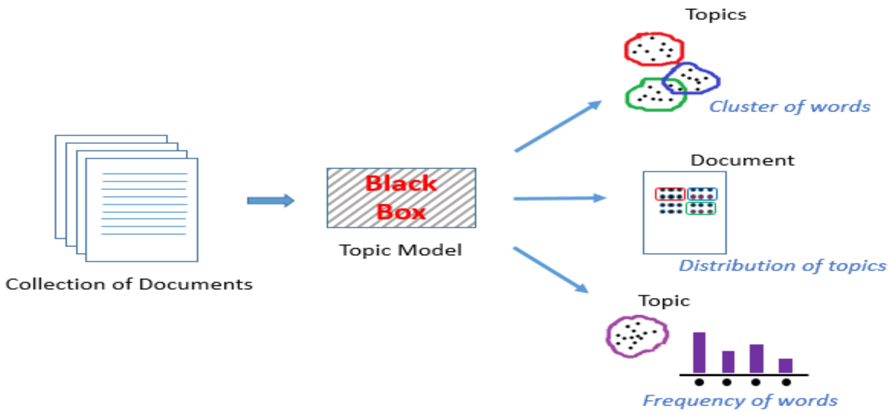


Figure 7. LSA working

K nearest neighbour (KNN):

1. Take unclassified data
2. Measure the distance (for example Euclidian or Manhattan) from the new data to all others data that is already classified
3. Gets the K smaller distances

4. Check the list of classes had the shortest distance and count the amount of each class that appears
5. The class with the maximum count is selected
6. Classify the new data with the class calculated in step 5

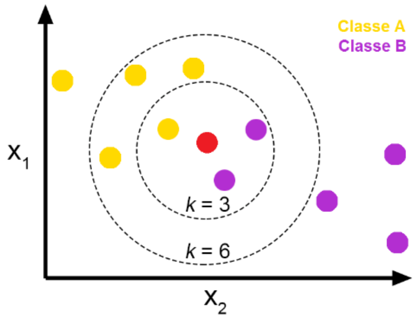


Figure 8. K nearest neighbour

There is a combination of SVM and KNN. Coordination of another calculation improves the order execution of the KNN classifier.

E. TFIDF + Naïve Bayes

Text vectorization, means to transform that text into numbers. Once the words have been transformed into numbers, in a way that's machine learning algorithms can understand, the TF-IDF score can be fed to algorithms such as Naive Bayes and Support Vector Machines, greatly improving the results of more basic methods like word counts.

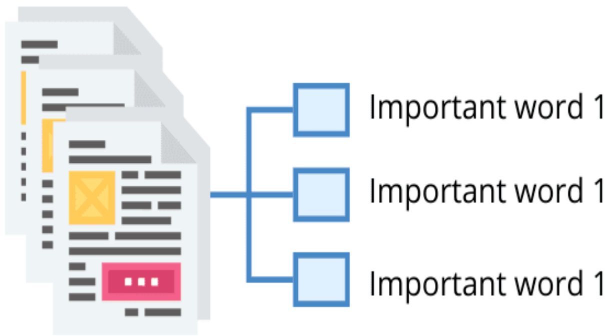


Figure 9. TF-IDF

F. Ensemble prediction

The ensemble method consists of finding the best model combinations through the smallest mean absolute error criterion, the ensemble models are applied for all eight sets.

For the final predicted values, a combination of models are selected from the models explained above, that produces the best prediction for each set, and ultimately build an ensemble model that outperforms any single regression or classification model.

Results

A. Pre-processing and feature extraction

The figures below describe the essay dataset.

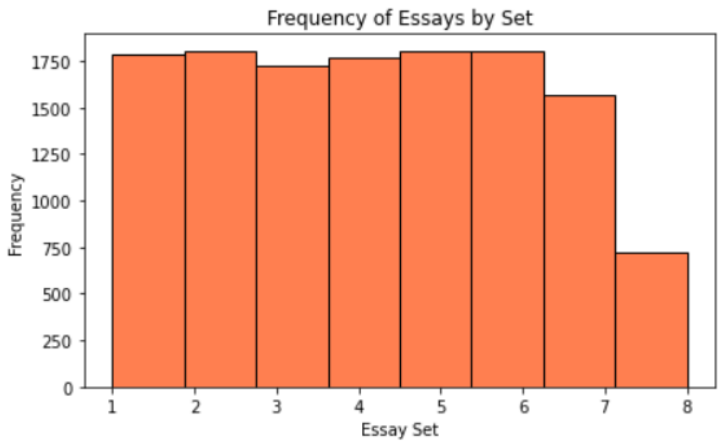


Figure 10. Frequency of Essays by set

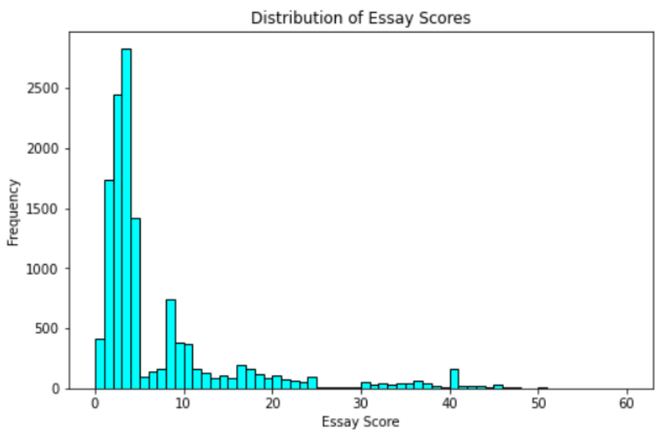


Figure 11. Distribution of Essay Scores

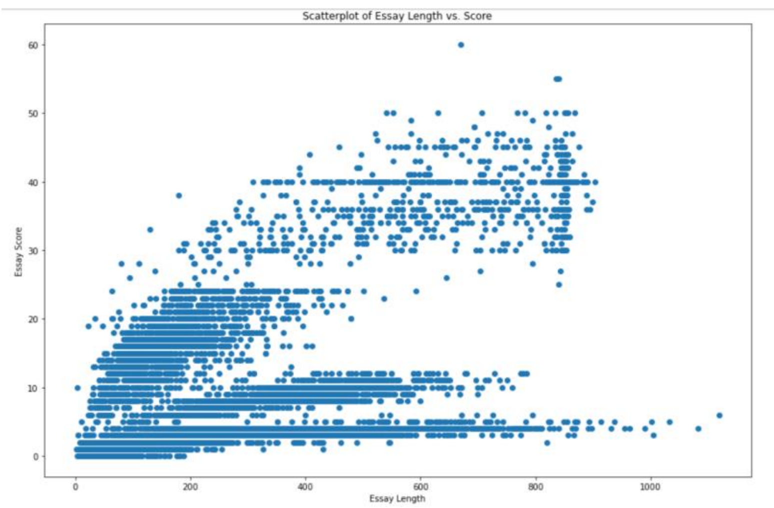


Figure 12. Scatter plot of essay length vs score

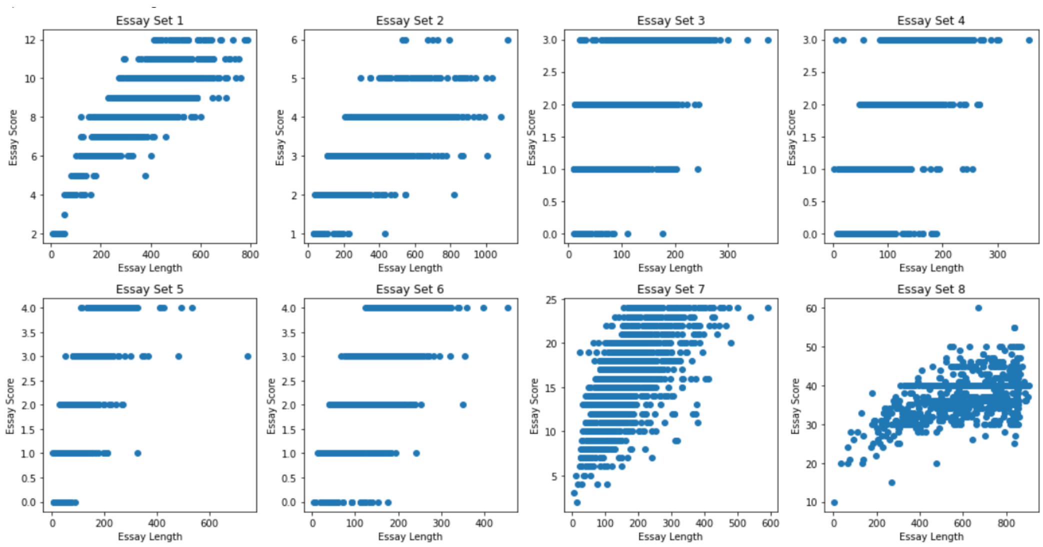


Figure 13. Scatter plot for each essay set vs respective essay score

The figures below demonstrate the 3D scatter plots of components using LSA:

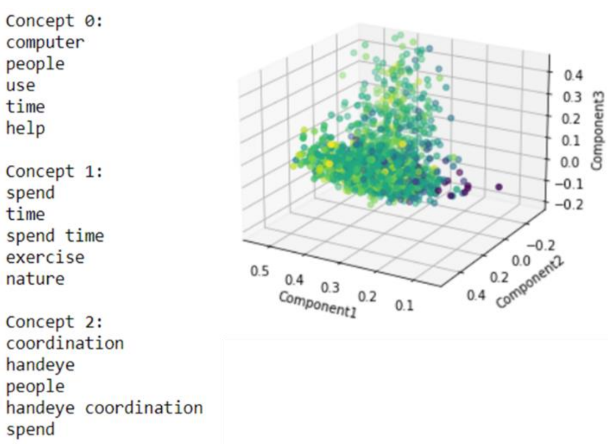


Figure 14. LSA for Essay set 1

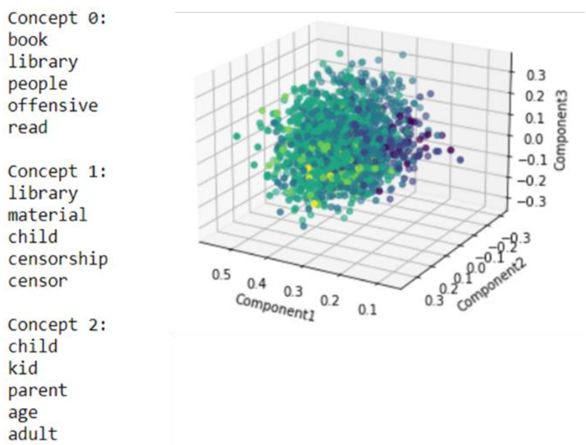


Figure 15. LSA for Essay set 2

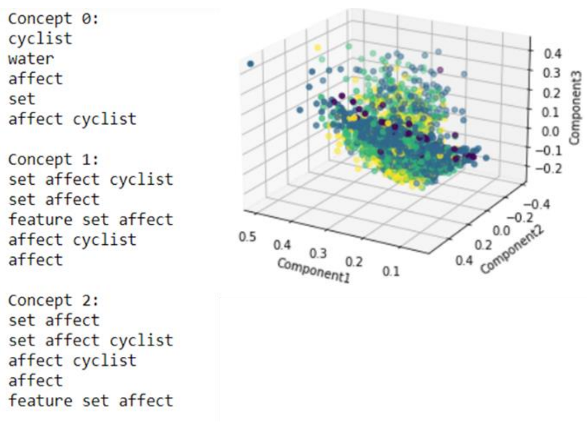


Figure 16. LSA for Essay set 3

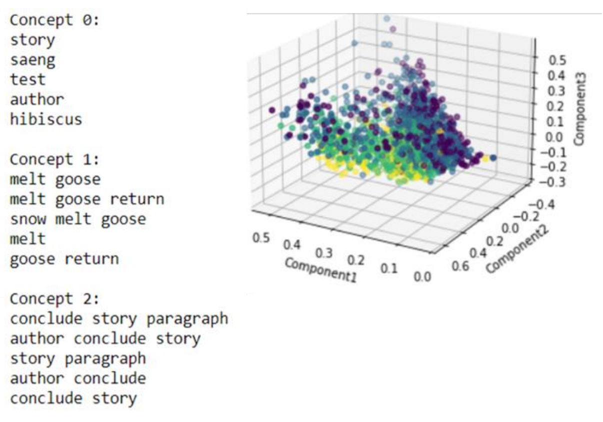


Figure 17. LSA for Essay set 4

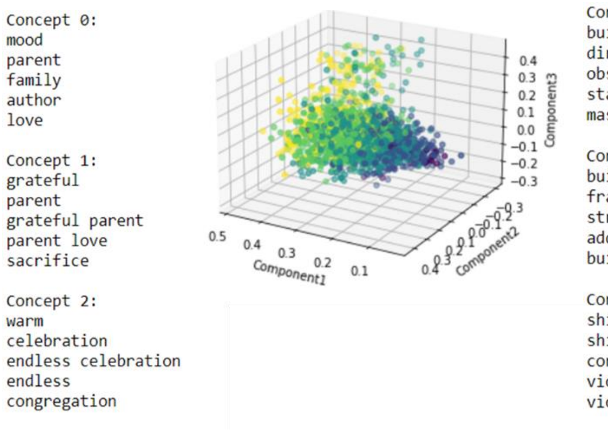


Figure 18. LSA for Essay set 5

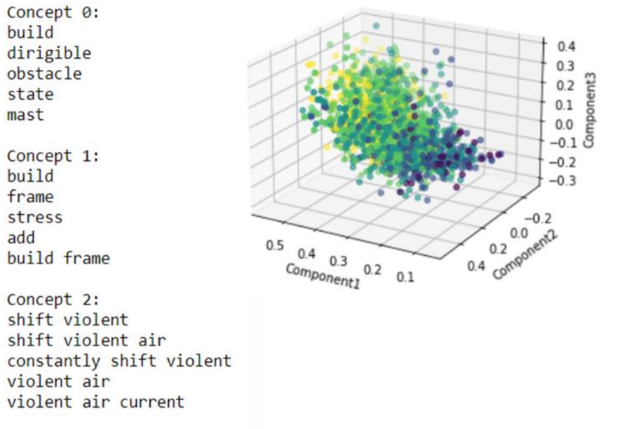


Figure 19. LSA for Essay set 6

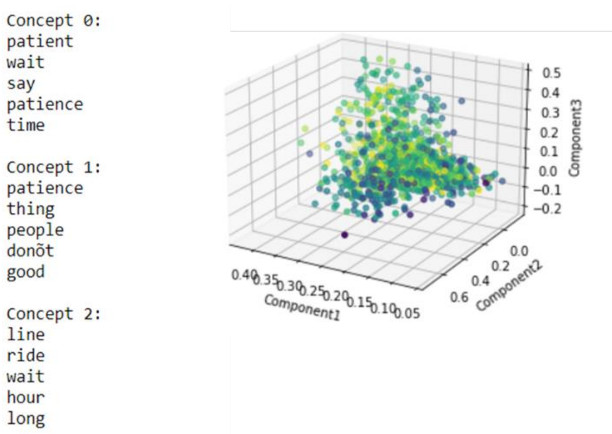


Figure 20. LSA for Essay set 7

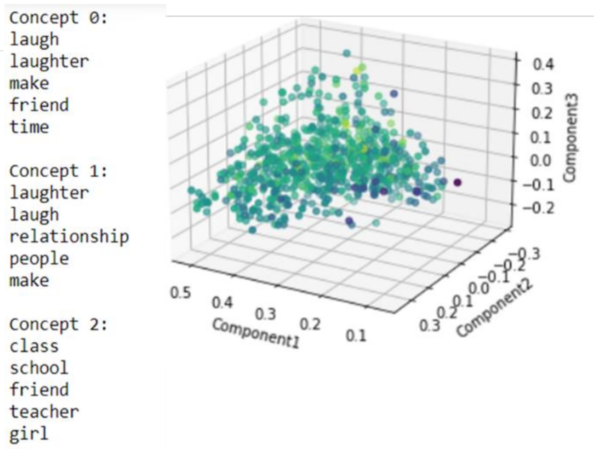


Figure 21. LSA for Essay set 8

B. Building Models

i. Linear Regression

The lasso results with the following important features number of sentences, number of words, number of words cleaned, and number of unique words cleaned.

ii. Random Forest Classifier

The cross validation results of the Random forest classifier are shown in the figure below. We can see that depths 2-6 perform reasonably well across these essay sets.

```
--------------------------------
1
best max_depth: [4]
mean validation score at this depth: 0.529
--------------------------------
2
best max_depth: [24]
mean validation score at this depth: 0.693
--------------------------------
3
best max_depth: [11]
mean validation score at this depth: 0.685
--------------------------------
4
best max_depth: [14]
mean validation score at this depth: 0.64
--------------------------------
5
best max_depth: [14]
mean validation score at this depth: 0.682
--------------------------------
6
best max_depth: [21]
mean validation score at this depth: 0.627
--------------------------------
7
best max_depth: [5]
mean validation score at this depth: 0.198
--------------------------------
8
best max_depth: [5]
mean validation score at this depth: 0.248
```

Figure 22. Cross validation result Random forest classifier

iii. Random Forest Regressor

The cross validation results of the Random forest regressor are shown in the figure below. We can see that depths 1, 4, 5, 6 and 7 perform reasonably well across these essay sets.

```
--------------------------------
1
best max_depth: [5]
mean validation score at this depth: 0.721
--------------------------------
2
best max_depth: [6]
mean validation score at this depth: 0.562
--------------------------------
3
best max_depth: [4]
mean validation score at this depth: 0.528
--------------------------------
4
best max_depth: [8]
mean validation score at this depth: 0.607
--------------------------------
5
best max_depth: [6]
mean validation score at this depth: 0.705
--------------------------------
6
best max_depth: [7]
mean validation score at this depth: 0.629
--------------------------------
7
best max_depth: [12]
mean validation score at this depth: 0.632
--------------------------------
8
best max_depth: [12]
mean validation score at this depth: 0.518
```

Figure 23. Cross validation result Random forest regressor

iv. LSA and KNN

The cross validation results of the LSA and KNN are shown in the figure below.

```
--------------------------------
2 Components
best n_neighbors: [40]
mean validation score at this n_neighbors: 0.41
--------------------------------
3 Components
best n_neighbors: [40]
mean validation score at this n_neighbors: 0.409
--------------------------------
4 Components
best n_neighbors: [30]
mean validation score at this n_neighbors: 0.403
--------------------------------
5 Components
best n_neighbors: [30]
mean validation score at this n_neighbors: 0.399
--------------------------------
6 Components
best n_neighbors: [30]
mean validation score at this n_neighbors: 0.418
--------------------------------
7 Components
best n_neighbors: [40]
mean validation score at this n_neighbors: 0.436
--------------------------------
8 Components
best n_neighbors: [50]
mean validation score at this n_neighbors: 0.432
--------------------------------
9 Components
best n_neighbors: [40]
mean validation score at this n_neighbors: 0.437
--------------------------------
--------------------------------
10 Components
best n_neighbors: [20]
mean validation score at this n_neighbors: 0.433
--------------------------------
11 Components
best n_neighbors: [30]
mean validation score at this n_neighbors: 0.442
--------------------------------
12 Components
best n_neighbors: [40]
mean validation score at this n_neighbors: 0.445
--------------------------------
13 Components
best n_neighbors: [20]
mean validation score at this n_neighbors: 0.451
--------------------------------
14 Components
best n_neighbors: [20]
mean validation score at this n_neighbors: 0.447
--------------------------------
15 Components
best n_neighbors: [30]
mean validation score at this n_neighbors: 0.449
--------------------------------
16 Components
best n_neighbors: [10]
mean validation score at this n_neighbors: 0.446
--------------------------------
17 Components
best n_neighbors: [10]
mean validation score at this n_neighbors: 0.449
--------------------------------
--------------------------------
18 Components
best n_neighbors: [20]
mean validation score at this n_neighbors: 0.449
--------------------------------
19 Components
best n_neighbors: [30]
mean validation score at this n_neighbors: 0.446
--------------------------------
20 Components
best n_neighbors: [10]
mean validation score at this n_neighbors: 0.448
```

Figure 24. Cross validation result for LSA and KNN model

v. TF-IDF and Naïve Bayes

The cross validation results of the Random forest classifier are shown in the figure below.

```
---------------------------------------
Accuracy of train set is: 0.43028846153846156
Accuracy of test set is: 0.351401869158785
---------------------------------------
Accuracy of train set is: 0.6142857142857143
Accuracy of test set is: 0.5074074074074074
---------------------------------------
Accuracy of train set is: 0.6374172185430463
Accuracy of test set is: 0.5405405405405406
---------------------------------------
Accuracy of train set is: 0.5907990314769975
Accuracy of test set is: 0.5657894736842105
---------------------------------------
Accuracy of train set is: 0.5882818685669042
Accuracy of test set is: 0.46863468634686345
---------------------------------------
Accuracy of train set is: 0.5936507936507937
Accuracy of test set is: 0.5074074074074074
---------------------------------------
Accuracy of train set is: 0.23224043715846995
Accuracy of test set is: 0.10828025477707007
---------------------------------------
Accuracy of train set is: 0.29446640316205536
Accuracy of test set is: 0.17972350230414746
```

```
Accuracy of test set is: 0.4
---------------------------------------
Accuracy of test set is: 0.5574074074074075
---------------------------------------
Accuracy of test set is: 0.4671814671814672
---------------------------------------
Accuracy of test set is: 0.5206766917293233
---------------------------------------
Accuracy of test set is: 0.4225092250922509
---------------------------------------
Accuracy of test set is: 0.45
---------------------------------------
Accuracy of test set is: 0.11889596602972399
---------------------------------------
Accuracy of test set is: 0.1935483870967742
---------------------------------------
```

Figure 25. Accuracy for the TF-IDF and Naïve Bayes model

vi. Ensemble prediction

The ensemble prediction consists of all the models, and gives a combination of models as output which would give the best output for that particular set.

The figures below shows the ensemble result for all the set:

```
the best combination is ['RandomForestRegressor', 'Ridge']
this has mean absolute error of 0.5850467289719626
```

Figure 26. Ensemble result: Set 1

```
the best combination is ['RFRegressor', 'RFClassifier', 'Ridge', 'Lasso', 'LDA', 'Naive Bayes', 'Logistic']
this has total absolute error of 162.0
```

Figure 27. Ensemble result: Set 2

```
the best combination is ['LDA', 'RFClassifier', 'SVM']
this has total absolute error of 171.0
```

Figure 28. Ensemble result: Set 3

```
the best combination is ['Ridge', 'Lasso', 'RFRegressor', 'RFClassifier', 'LDA']
this has total absolute error of 193.0
```

Figure 29. Ensemble result: Set 4

```
the best combination is ['RFRegressor']
this has total absolute error of 173.0
```

Figure 30. Ensemble result: Set 5

```
the best combination is ['RFClassifier', 'RFRegressor', 'LDA']
this has total absolute error of 199.0
```

Figure 31. Ensemble result: Set 6

```
the best combination is ['RandomForestRegressor', 'Ridge']
this has mean absolute error of 2.1337579617834397
```

Figure 32. Ensemble result: Set 7

```
the best combination is ['Ridge', 'RandomForestRegressor']
this has mean absolute error of 2.857142857142857
```

Figure 33. Ensemble result: Set 8

The resultant ensemble scores are shown in the figure below. The highest score attained is 68.9%.

```
[0.6485927835212009,
 0.6685185185185185,
 0.6891891891891891,
 0.6109022556390977,
 0.6771217712177122,
 0.6518518518518519,
 0.6487394918691498,
 0.4967178282580065]
```

Figure 34. Ensemble scores

The accuracy score summary table for all the models is displayed in the figure below.

	Unnamed: 0	Lasso	Ridge	LogReg	KNN	SVM	LDA	RFC	RFR	NB	LSAKNN
0	Set1	0.439252	0.476636	0.484112	0.491589	0.504673	0.489720	0.514019	0.493458	0.386916	0.373832
1	Set2	0.624074	0.653704	0.627778	0.609259	0.655556	0.631481	0.672222	0.681481	0.527778	0.527778
2	Set3	0.646718	0.648649	0.654440	0.583012	0.681467	0.685328	0.667954	0.687259	0.476834	0.341699
3	Set4	0.588346	0.631579	0.588346	0.577068	0.593985	0.639098	0.629699	0.646617	0.515038	0.460526
4	Set5	0.675277	0.680812	0.638376	0.625461	0.662362	0.658672	0.677122	0.688192	0.431734	0.422509
5	Set6	0.561111	0.607407	0.594444	0.544444	0.616667	0.620370	0.637037	0.640741	0.498148	0.435185
6	Set7	0.127389	0.129512	0.220807	0.131635	0.184713	0.201699	0.188960	0.121019	0.144374	0.091295
7	Set8	0.115207	0.064516	0.230415	0.133641	0.253456	0.184332	0.267281	0.078341	0.248848	0.179724

Figure 35. Accuracy score summary table

The ensemble method out performs other models in terms of the prediction accuracy. The advantage of the ensemble method is not as outstanding for the classification approaches, but the ensemble model's accuracy scores are better than most of the scores from the other models.

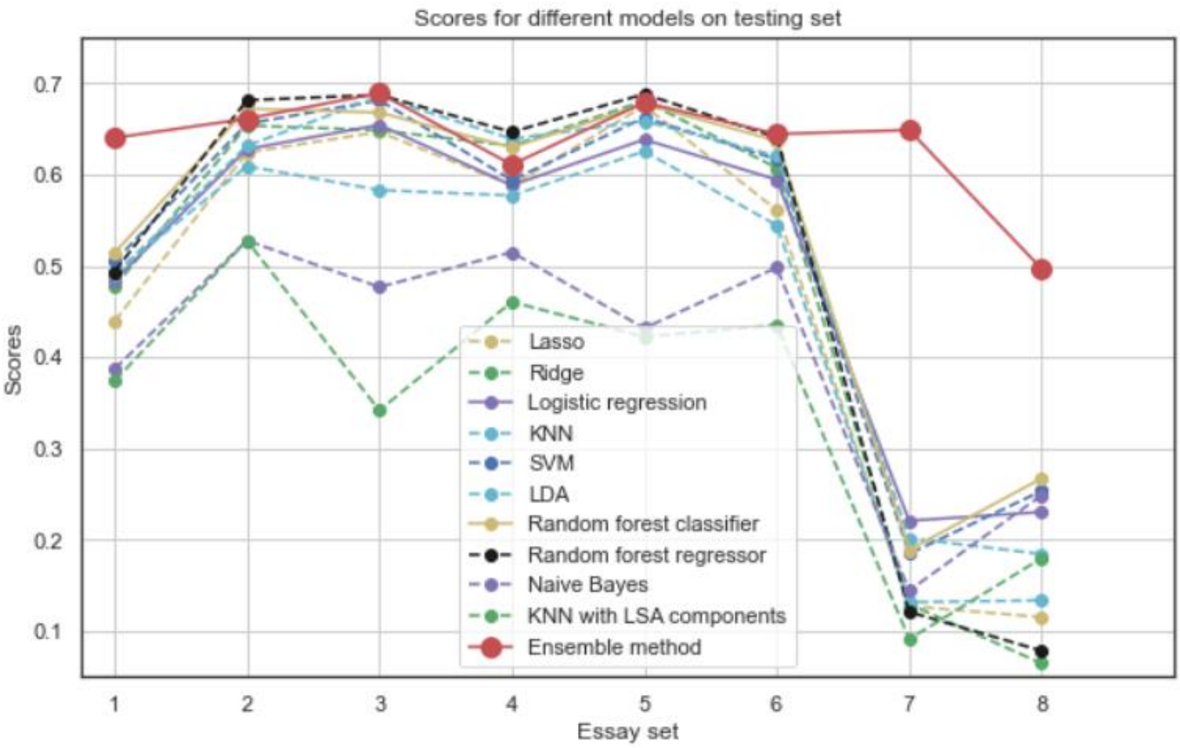


Figure 36. Plot for scores for different models

C. LSA Visualizations and WordCloud

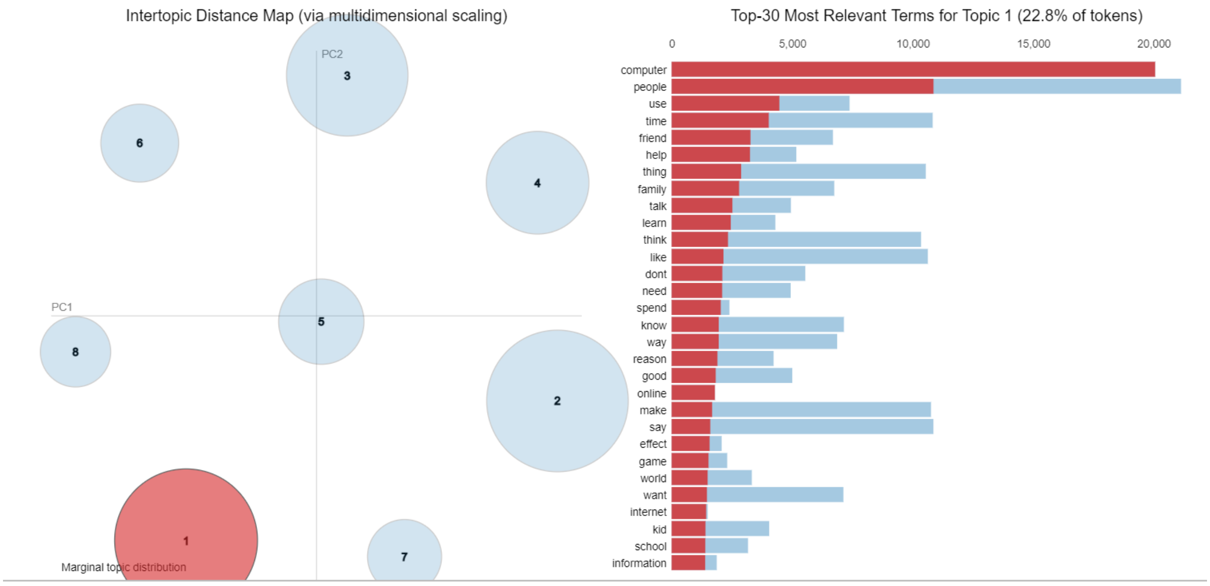


Figure 37. All sets: 8 topics and relevant terms using CountVectorizer

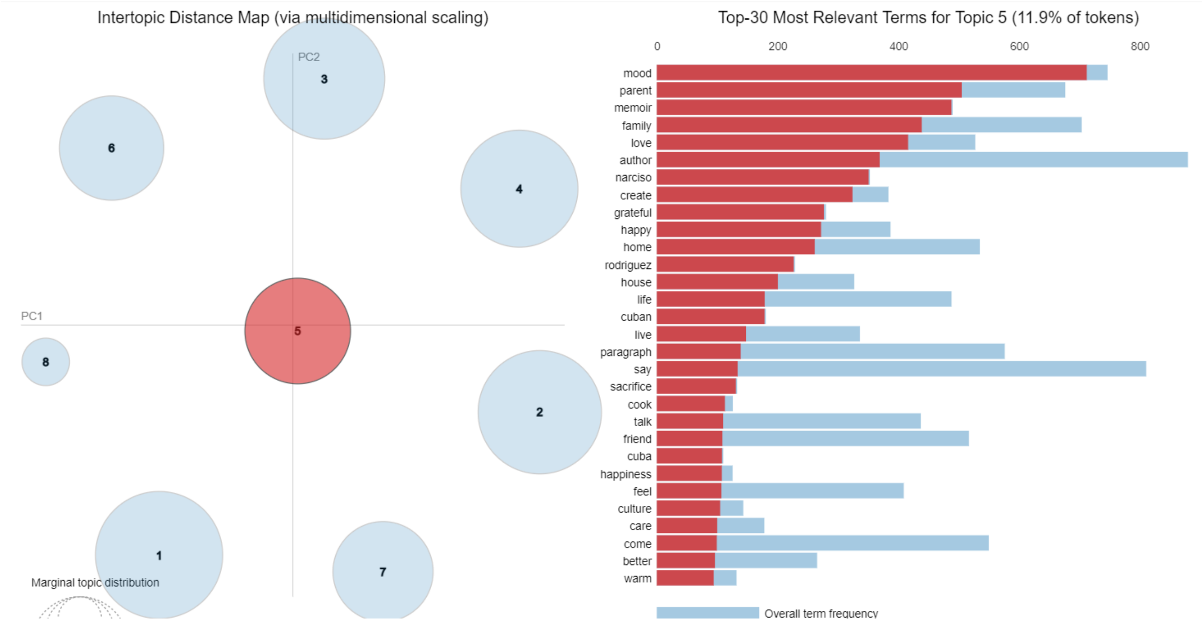


Figure 38. All sets: 8 topics and relevant terms using TF-IDF vectorizer

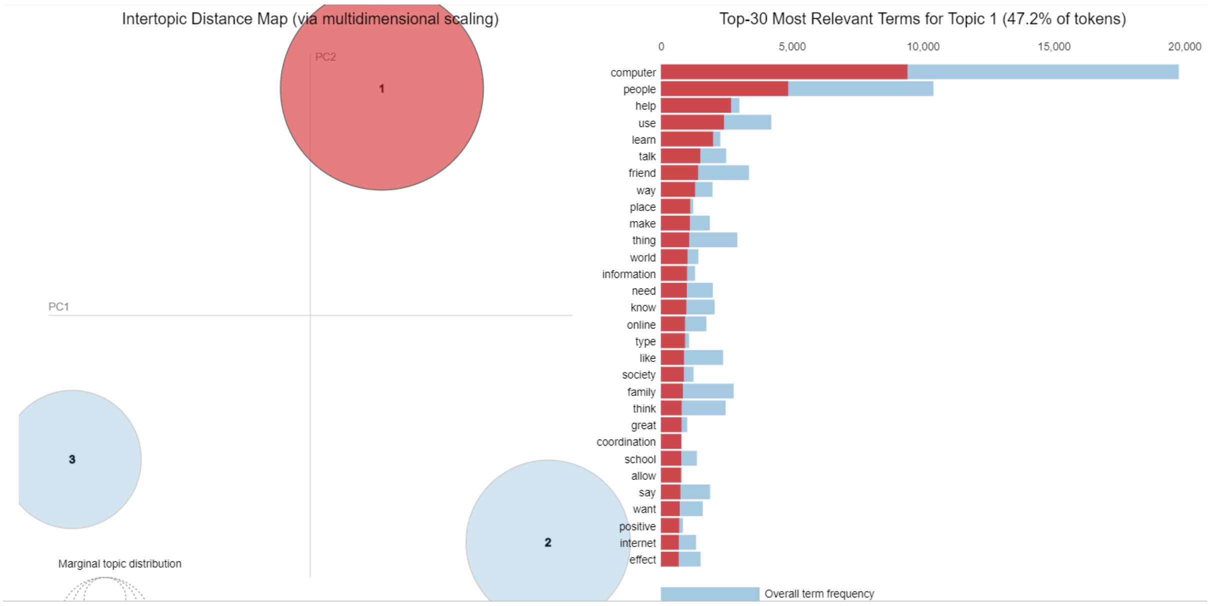


Figure 39. Topics in Set 1

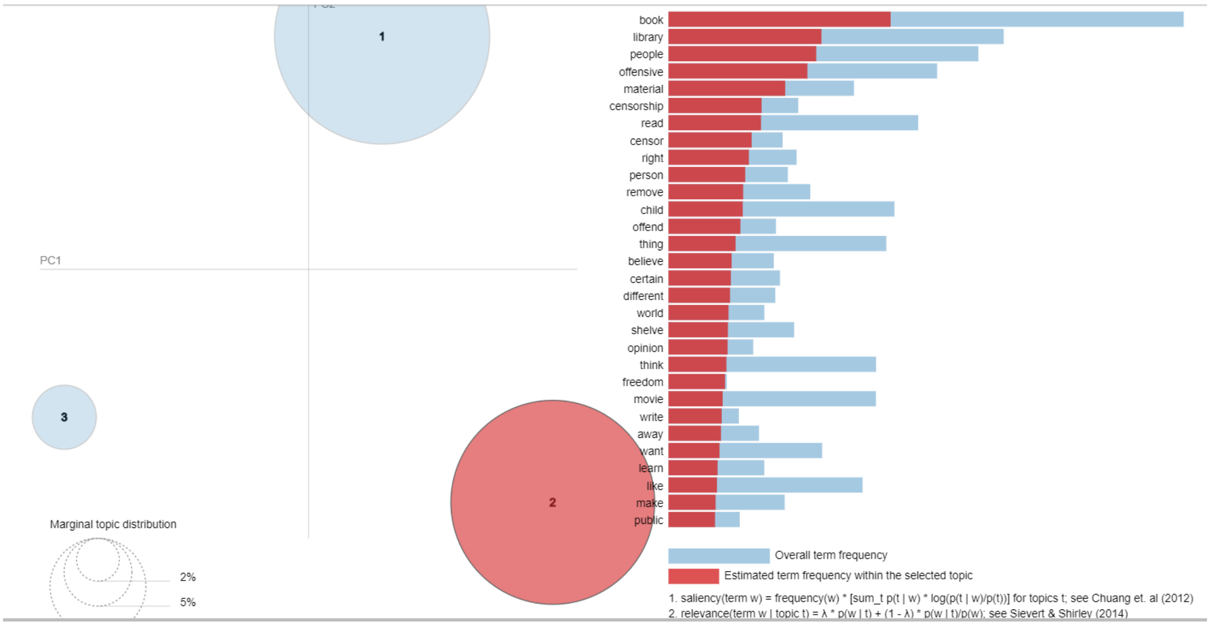


Figure 40. Topics in Set 2

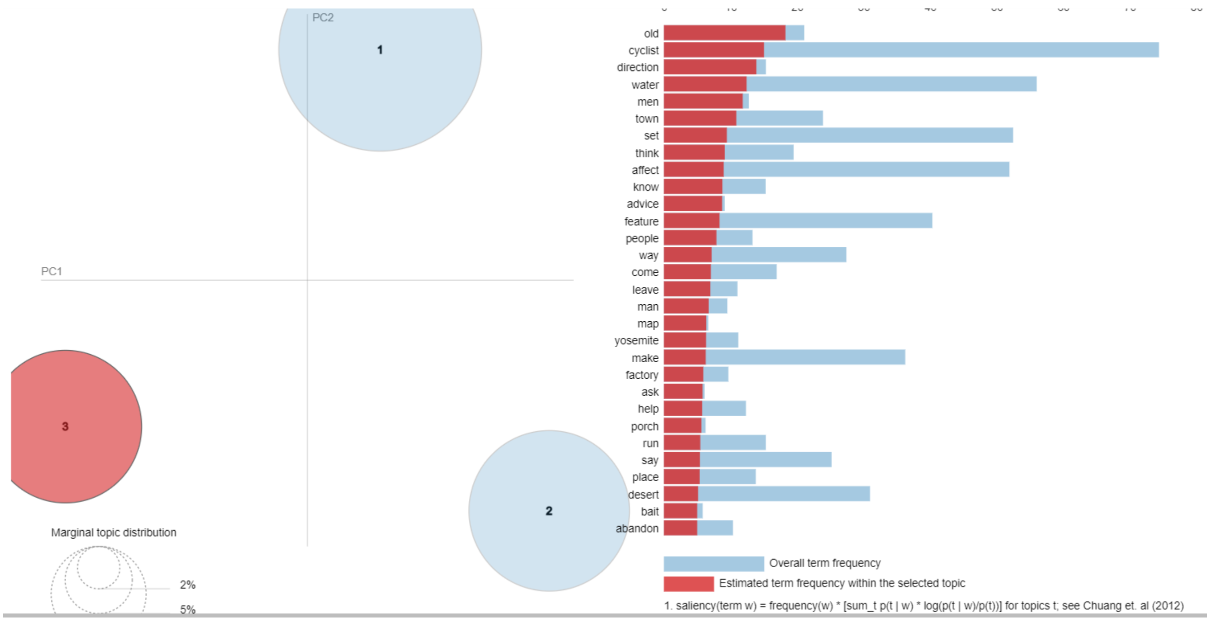
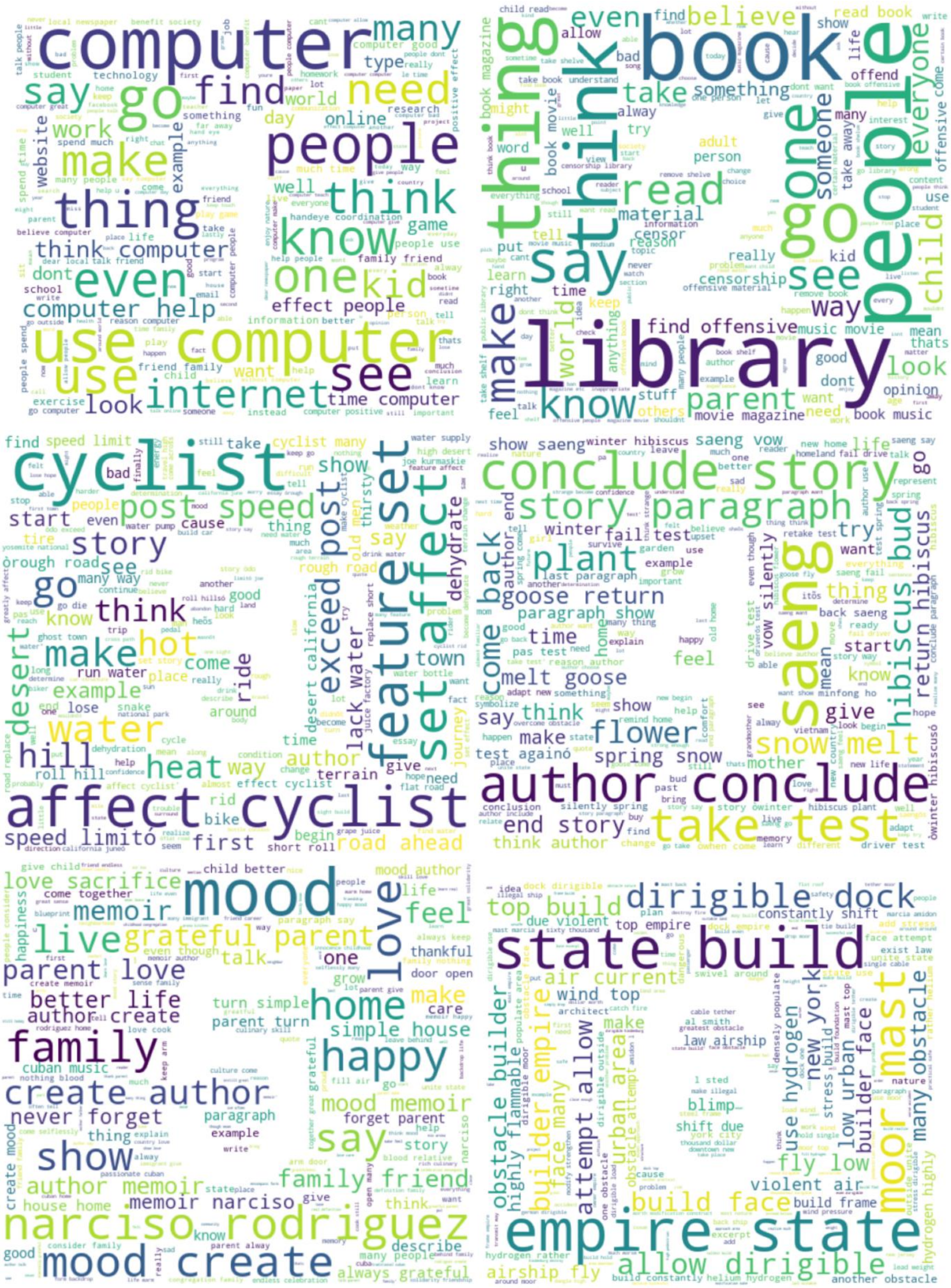
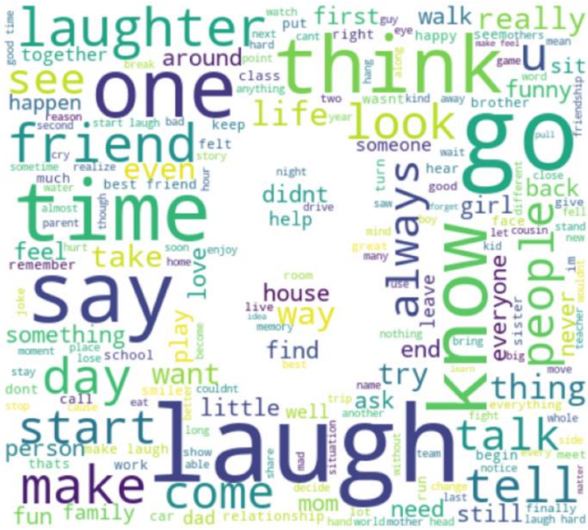


Figure 41. Topics in Set 3

WordClouds:

The figures below show the word clouds for the respective essay set number displayed in the centre of the image.



Conclusion

Automated Essay Grading applies natural language processing to a type of instructive assessment. By grouping an enormous arrangement of text information into few discrete classes, this methodology generally decreases the expense related with estimating instructive accomplishment and is getting progressively mainstream. The application of TF-IDF and LSA analysis and put together an ensemble final prediction model that outperforms all the other single models for each essay set.

Since the response variable is the evaluated essay score, the modelling fundamentally reflects and predicts the dynamic cycle of the human exposition graders. Notwithstanding, paper evaluating is an emotional cycle that requires a great deal of setting, and it is hard to measure the nature of an article into only a couple measurements. For instance, our model shows that article length is decidedly corresponded with scores. On the off chance that student knows how the robotized scoring calculation functions, the individual in question would be boosted to compose however many words as could be allowed and increment the paper length superfluously.

Future Scope

For instance, we can envision a situation where an student composes a great and one of a kind essay that got a high score from a human grader yet that got a low score from the group model. Then again, we can likewise envision a case wherein an essay performed ineffectively when evaluated by a human grader however performed well with the computerized calculation. Looking further into papers in which a huge error exists between the human grader's score and the calculation's score would give us knowledge into other likely highlights to consider and how to improve our model by and large.

Second, we could consider text characterization with Word2Vec, a gathering of two-layer neural organizations models that are utilized to create word embeddings. Semantic vectors, will safeguard the greater part of the important data about a content while having generally low dimensionality. This element permits preferable AI treatment over straight one-hot encoding of words.

References

[1] “Experimenting with Latent Semantic Analysis and Latent Dirichlet Allocation on Automated Essay Grading”, Jalaa Hoblos, IEEE-2020

[2] “Automated English Digital Essay Grader Using Machine Learning”, Yafet Salim, Valdi Stevanus, Edwardo Barlian, Azani Cempaka Sari, Derwin Suhartono , IEEE-2020

[3] “Deep Automated Text Scoring Model Based on Memory Network” , Shiyan Yang, CVIDL-2020

[4] “An Analysis of Automated Answer Evaluation Systems based on Machine Learning”, Birpal Singh J. Kapoor, Shubham M. Nagpure, Sushil S. Kolhatkar, Prajwal G. Chanore, Mohan M. Vishwakarma, Prof. Rohan B. Kokate, ICICT-2018

[5] “Term Frequency-Inverse Document Frequency Answer Categorization with Support Vector Machine on Automatic Short Essay Grading System with Latent Semantic Analysis for Japanese Language” , Anak Agung Putri Ratna, Aaliyah Kaltsum, Lea Santiar, Hanifah Khairunissa, Ihsan Ibrahim, Prima Dewi Purnamasari, ICECOS, 2019

[6] “Answer Categorization Method Using K-Means for Indonesian Language Automatic Short Answer Grading System Based on Latent Semantic Analysis”, Anak Agung Putri Ratna, Naiza Astri Wulandari, Aaliyah Kaltsum, Ihsan Ibrahim, Prima Dewi Purnamasari, IEEE-2018

[7] “Syntactic, Semantic and Sentiment Analysis: The Joint Effect on Automated Essay Evaluation”, Harneet Kaur Janda , Atish Pawar , Shan Du, AND Vijay Mago, IEEE-2019

[8] “Analysis of Algorithms in Automated Marking in Education: A Proposed Hybrid Algorithm”, Binita Prasain, Simi Bajaj, IEEE-2020