

# AUTOMATED ARTICLE SCORING

LIM YOU QIAN

SESSION 2018/2019

FACULTY OF COMPUTING AND INFORMATICS  
MULTIMEDIA UNIVERSITY  
FEBRUARY 2019



# AUTOMATED ARTICLE SCORING

BY

LIM YOU QIAN

SESSION  
2018/2019

THIS PROJECT REPORT IS  
PREPARED FOR

FACULTY OF COMPUTING AND INFORMATICS  
MULTIMEDIA UNIVERISTY  
IN PARTIAL FULFILLMENT  
FOR

BACHELOR OF COMPUTER SCIENCE  
B.CS (HONS) DATA SCIENCE

FACULTY OF COMPUTING AND INFORMATICS

MULTIMEDIA UNIVERSITY

FEBRUARY 2019

**COPYRIGHT** of this report belongs to University Telekom Sdn. Bhd. As qualified Regulation 7.2 (c) of the Multimedia University Intellectual Property and Commercialisation Policy. No part of this publication may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical photocopying recording, or otherwise), or for any purpose, without the express written permission of University Telekom Sdn. Bhd. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2019 University Telekom Sdn. Bhd. ALL RIGHTS RESERVED.

## DECLARATION

I hereby declare that the work has been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

---

*Lim You Qian*

Faculty of Computing & Informatics  
Multimedia University

Date: 14: 02: 2019

## **ACKNOWLEDGEMENT**

I wish to express my sincere gratitude to everyone who has helped throughout this first part of the final year project. Firstly, this work would not have been possible without the guidance, monitoring, and advice from my supervisor, Dr. Ian Tan Kim Teck. I appreciate the encouragement and guidance from him very much every time I feeling doubtful during this project. Besides, another appreciation to Dr. Ian Tan for giving me the chance to explore Automated Article Scoring related field.

Moreover, I would also like to acknowledge with much appreciation the crucial role of my family members. The mental and financial supports are very important factors that lead this project to a success. Besides, suggestions that came from a different perspective are helpful as well.

Last but not least, I am also very grateful to have a group of supportive friends especially those who are also supervised by Dr. Ian Tan which are Kew Wai Chun, Goh Kun Shun and Chan Hoh Yue. They never hesitate in sharing their knowledge and ideas that help a lot in my project.

## **ABSTRACT**

Most of the organization still rely on manual evaluation by humans, it actually brings up some inconsistency in article scoring such as personal biased towards the writer or boredom when evaluating tons of articles. There are some commercial products such as E-rater and Intelligent Essay Assessor™ which produce a result that is acceptable by human but there is always a space for enhancement. We proposed to build an Automated Article Scoring (AAS) system with implementations of machine learning, non-machine learning model and combine it to provide a more reliable result. We have determined 14 features to build the machine learning and non-machine learning model. Support Vector Machine algorithm is used to build the machine learning model while a total of 6 methods has been proposed to build the non-machine learning model. Next, Weighted Mean Approach and including result from non-machine learning model are applied to perform the combination of models to achieve a better result. The overall accuracy of 83% is achieved by the second model combination approach which is the highest. The best model from machine learning, non-machine learning model and weighted mean approach each will be implemented in the AAS System.

# TABLE OF CONTENTS

<b>COPYRIGHT .....</b>	<b>I</b>
<b>DECLARATION.....</b>	<b>II</b>
<b>ACKNOWLEDGEMENT .....</b>	<b>III</b>
<b>ABSTRACT .....</b>	<b>IV</b>
<b>TABLE OF CONTENTS.....</b>	<b>V</b>
<b>LIST OF FIGURES .....</b>	<b>IX</b>
<b>LIST OF TABLE .....</b>	<b>XI</b>
<b>CHAPTER 1 INTRODUCTION .....</b>	<b>1</b>
1.1 Project Overview.....	1
1.2 Problem Statement .....	3
1.3 Objective .....	4
1.4 Project Scope.....	5
<b>CHAPTER 2 LITERATURE REVIEW .....</b>	<b>6</b>
2.1 Related Work .....	6
2.1.1 Project Essay Grader (PEG).....	6
2.1.2 Intelligent Essay Assessor™ (IEA) .....	7
2.1.3 E-Rater® .....	8
2.2 Features .....	9
2.3 Classification Method and Algorithm.....	11
2.4 Support Vector Machine's Kernel .....	13
<b>CHAPTER 3 RESEARCH METHODOLOGY .....</b>	<b>16</b>
3.1 Proposed Solution Overview.....	16



3.1.1 Building the essay evaluation models .....	16
3.1.2 Design the AAS System.....	18
Programming Language, Packages and Framework .....	21
3.2 Dataset.....	21
3.3 Features .....	22
3.4 Imbalance Class Distribution Dataset .....	24
3.5 Machine Learning Model 1: SVM Classification .....	25
3.5.1 Multiclass Classification Approach .....	25
3.5.2 Kernels and Parameters.....	28
3.6 Quantitative Analysis Model.....	29
3.6.1 Determine the separation point for each feature that separate the class of the essay .....	29
3.6.2 Sum up all the features and determine the best separation point among all possible set of separation points.....	30
3.6.3 Sum up all the features and assign weights to determine the best separation point among all possible set of separation points .....	30
3.7 Combination of Model Approach .....	31
3.7.1 Method 1: Weighted Mean Approach.....	31
3.7.2 Method 2: Machine Learning Model 2 .....	31
3.8 Model Evaluation .....	32
3.9 AAS System Design.....	33
<b>CHAPTER 4 IMPLEMENTATION .....</b>	<b>36</b>

4.1 Data Pre-processing .....	36
4.1.1 Replace Anonymization .....	36
4.1.2 Tokenization.....	37
4.1.3 Remove Punctuation .....	38
4.2 Features Extraction.....	38
4.3 Building Machine Learning model .....	39
4.3.1 Oversampling .....	40
4.3.2 Parameter Tuning .....	41
4.3.3 Overfitting .....	41
4.4 Building Non-machine Learning Model .....	42
4.4.1 Determine the separation point for each feature that separate the class of the essay .....	42
4.4.2 Sum up all the features and determine the best separation point among all possible set of separation points.....	43
4.4.3 Sum up all the features and assign weights to determine the best separation point among all possible set of separation points .....	43
4.5 Building the Model Combination 1 and 2.....	45
4.5.1 Model Combination 1: Weighted Mean Approach .....	45
4.5.2 Model Combination 2: Machine Learning Model 2 .....	45
4.6 Implementation Summary .....	46
<b>CHAPTER 5 IMPLEMENTATION RESULT EVALUATION.....</b>	<b>48</b>
5.1 Machine Learning Model .....	48

5.2 Non-machine Learning Model .....	50
5.2.1 Model Combination 1: Weighted Mean Approach.....	54
5.2.2 Model Combination 2: Machine Learning Model 2 .....	55
5.3 Implementation Result Summary.....	57
<b>CHAPTER 6 CONCLUSION .....</b>	<b>59</b>
6.1 Future Works.....	59
<b>REFERENCES.....</b>	<b>60</b>
<b>APPENDICES .....</b>	<b>63</b>
Appendix A: Draft Research Paper.....	63
Appendix B: Turnitin Report .....	68

## LIST OF FIGURES

Figure 2.1: Features' Kappa Score .....	11
Figure 2.2: Models' accuracies for each set of essay (Madala, Gangal, Krishna, Goyal, & Sureka, 2018) .....	12
Figure 2.3: Transformation of feature space by applying kernel function (Cambridgespark, 2016).....	14
Figure 3.1: An overview flowchart of AAS System Design from user's perspective	19
Figure 3.2 An overview flowchart of AAS System backend.....	20
Figure 3.3 Example of SVM separating data points with maximum margin (Mathieu, 2018) .....	25
Figure 3.4 Graphical demonstration of OVR method.....	26
Figure 3.5 Graphical demonstration of OVO method.....	27
Figure 3.6 UI of user upload essay.....	34
Figure 3.7 UI of user customize the scoring scheme for non-machine learning model .....	34
Figure 3.8 UI of essay result (.txt) .....	35
Figure 3.9 UI of essay result (.csv) .....	35
Figure 4.1: Example of each sentence tokenizer in NLTK (Word Tokenization with Python NLTK, n.d.) .....	37
Figure 4.2 Flowchart of building the machine learning model.....	40
Figure 4.3 Flowchart of building the non-machine learning model.....	46
Figure 5.1 Result of the model without oversampling.....	48
Figure 5.2 Result of model with oversampling the train set .....	49

Figure 5.3 Result of model with oversampling the whole dataset .....	49
Figure 5.4 Result of the model by mean method .....	52
Figure 5.5 Result of the model by median method .....	52
Figure 5.6 Result of the model by ratio method.....	52
Figure 5.7 Result of the model by summing up features without weight .....	53
Figure 5.8 Result of the model by summing up features with weight .....	53
Figure 5.9 Result of the model by summing up features with sparse weight .....	53
Figure 5.10 Result of the machine learning model 2 .....	55
Figure 5.11 Result of machine learning model 2 .....	56

## LIST OF TABLE

Table 2.1: Feature ranking by RELIEF (Madala, Gangal, Krishna, Goyal, & Sureka, 2018) .....	10
Table 2.2: Summary of time taken to train, predict of kernels and their accuracy. (Pahwa & Sinwar, 2015) .....	14
Table 3.1 Description of each required attribute in dataset .....	22
Table 3.2: Summary of number of essay and the score range for each essay set (Kaggle, n.d.).....	22
Table 3.3: Summary of training and test set. ....	22
Table 3.4: Selected features and its description .....	24
Table 3.5 Example of classifiers constructed in OVR method .....	26
Table 3.6 Example of classifiers constructed in OVO method.....	27
Table 3.7: Summary of model evaluation methods .....	33
Table 4.1: Example of Anonymization done by the dataset provider along the result after anonymization replacement (Kaggle, n.d.) .....	36
Table 4.2: Extracted features and the procedure of extracting it from the essay .....	39
Table 4.3 Parameter applied for tuning and its range .....	41
Table 4.4 Weights implemented for the weighted mean approach.....	45
Table 5.1 Result of parameter tuning .....	48
Table 5.2 Result of cross validation for the model without oversampling .....	50
Table 5.3 Separation points for each feature for the mean, median and ratio method .....	51

Table 5.4 Separation points for the sum up feature with and without weight method .....	52
Table 5.5 Result of parameter tuning for machine learning model 2 .....	55
Table 5.7 Result of cross-validation for machine learning model 2 .....	57

# CHAPTER 1 INTRODUCTION

## *1.1 Project Overview*

Article and essay writing is a good way to express personal thoughts and improve writing skills. But in the sense of writing a good essay or article, there are various criteria or requirements that the evaluator will look into to evaluate an essay or article. Practice and critical evaluation can improve a student's writing skills efficiently especially in essay organization. Usually, human evaluate an essay by obtaining an "overall" impression then only examine on the technical part such as grammar, vocabulary usage, and organization (Burstein, Andreyev, & Lu, 2006). In the other hand, the human evaluation may be inaccurate sometimes due to boredom by evaluating a large amount of essay with the same title or being biased to some of the writers.

In this case, Automated Essay Scoring (AES) is introduced to solve the problems above. AES can evaluate essays without being affected by boredom or personal bias to writers. Research shows that the AES has high accuracy and reliability which reach a high agreement level between AES systems and human raters (Semire D. , 2006). However, AES systems are not perfect. AES system performance may not be ideal when the number of criteria involves in the evaluation of essay increases beyond a certain level (Burstein, Andreyev, & Lu, 2006). Although the machine rater shows its downside, the attention that brings up by the community does not stop. Public schools,



universities, researchers and educators, testing companies are showing interested in AES system (Semire D. , 2006).

AES systems that use qualitative analysis to evaluate essays are more frequent than the system that uses quantitative analysis. Quantitative analysis uses statistical methods while qualitative analysis uses a non-quantifiable method such as machine learning to compute a score for an essay.

This project aims to build an Automated Article Scoring (AAS) System that contains multiple types of article evaluation models. Firstly, this system will generate shallow linguistic features from the input essay such as the word count and average length of sentences before evaluating the input essay. The first evaluation model implemented in the AAS System is a machine learning technique where it uses a multiclass classification approach to score the article. A large number of essays with labels will be collected for the training purpose of the machine learning model. Next, the second article evaluation model implemented in the AAS System is non-machine learning. The non-machine learning method will examine the generated features quantitatively based on the proposed scoring scheme in this project or personal preference. After that, the AAS System will perform a combination of model approach. The first combination of model approach is the AAS System retrieves the score from each model and perform a weighted mean scoring approach. Then, the second approach is to include the score computed from non-machine learning model as a feature to build another machine learning model. There will be a total of 4 article evaluation model in the AAS System. The main purpose of this research project is to determine a method to provide a better

and more reliable score of the essay based on the machine learning and non-machine learning method.

## ***1.2 Problem Statement***

Scoring an essay manually by the human is the generally acknowledged and accepted method but very inefficient. Scoring a single essay manually by human may takes probably a few minutes only. However, given an example that when it comes to an examination or a competition where the organizing department wanted to mark all the essay and choose the best one among all the essays, time-consuming turned out to be a serious issue for the organizing department. Moreover, the cost to score tons of essays is very expensive. It costs a lot when you need to hire many human scorers to score all the essays in a limited time. In this situation, the organizing department can either spent more to reduce the time needed or provide a longer period of time to complete the task only.

On top of that, human's generalizability may also be an issue. When a human manually scores a large number of essays, the scoring quality from the human scorer might be affected due to boredom. It may cause a disadvantage for the essays at the back. Next, the scoring quality of the human scorer will also be affected when the human scorer is biased towards some students.

Apart from that, the industry has found a solution to overcome all the problem stated above which is to build an automated machine where most of them use machine learning techniques to score the essays. For example, Project Essay Grader™ (PEG), Intelligent Essay Assessor™ (IEA), E-rater and BETSY™ are the AES systems which

already ready in the market. All these solutions may improve the situation in the industry. However, there is always a space for improvement and enhancement in terms of providing a better result of essay scoring. In the machine learning approach, the AES system fully depends on the result computed from pre-built machine learning model. We may apply some techniques or approaches other than machine learning in order to allow the user to tune the result. This lead to the main purpose and objective of this research project.

### ***1.3 Objective***

The main purpose of this research-based project is to build an AAS System that contains multiple evaluation models on the input essay to achieve a more precise, accurate and reliable result. This purpose leads to several objectives to be achieved in this project: -

- 1. To build a classification model with a large essay dataset to classify the essay.**

In the machine learning approach, a multiclass classification model uses the Support Vector Machine (SVM) algorithm is built to classify the input essay. A large number of essays labeled with a human-rated score is acquired from kaggle.com.

- 2. To determine the feature to build machine learning and non-machine learning model**

Statistical representations such as words count, the average length of each word and etc. will be determined and extracted for evaluation.

**3. To determine the method to combine the machine learning and non-machine learning model.**

In order to provide a better result of the input essay, we will determine the best approach to combine the two models.

**4. To evaluate the result of the models**

The result and the logicalness of the models will be evaluated.

### ***1.4 Project Scope***

In this project, several scopes are applied in order to achieve and fulfill the main purpose and objective.

1. Only essay written in English is covered.
2. The area of evaluation on the essay is focus on surface/shallow linguistic features of the essay only.
3. The project focus on the method to combine the machine learning and non-machine learning model instead of the classification algorithm. Therefore, only one classification algorithm, SVM will be used in this project.
4. An AAS system will be built to evaluate the essay. The function of the AAS System includes the upload of the input essay and download of the evaluation result.
5. Only “.txt” and “.csv” file format is accepted by the AAS system.

## **CHAPTER 2 LITERATURE REVIEW**

### ***2.1 Related Work***

Several organizations or company have started to put in effort in this area. Some of the systems are Project Essay Grader (PEG), Intelligent Essay Assessor™ (IEA), E-Rater®, IntelliMetric™, and Bayesian Essay Test Scoring System™ (BETSY). The main reason that people started to develop in the AES system is that AES system has one of the major reason is that it is able to score and provide feedback instantly which significantly improves the efficiency of writing the assessment.

#### ***2.1.1 Project Essay Grader (PEG)***

PEG focus on style analysis of surface linguistic features of a block of text but abandoned content analysis (Semire D. , 2006). PEG uses statistical and correlation technique to evaluate the intrinsic qualities of the essays. There are two important terms that used by the developer: trins and proxes. Trins are the intrinsic variables such as diction, grammar, punctuation, and fluency, proxes means the correlation of the intrinsic variables. Average word length, number of semicolons, counts of prepositions and word rarity are some instances of proxes. In the training stage, around 100 to 400 of human graded essays is used. Up to 30 proxes are determined by human and used in the regression equation along with the human grades to calculate the regression coefficients. In the scoring stage, the proxes from the unmarked essays is obtained and put into the prediction equation to compute a grade for the unmarked essay.

Performance, the latest experiments run by Page have produced a high multiple regression correlation scores of 0.87 with human graders (Valenti & Cucchiarelli, 2003).

### ***2.1.2 Intelligent Essay Assessor™ (IEA)***

IEA is different with PEG, it focuses mainly on the content of the essay. IEA uses Latent Semantic Analysis (LSA) technique to evaluate essays. LSA is a statistical model that used to represent the document and their words in a two-dimensional matrix. The row represents each unique words in the document while the column represents the context of the words. Then, Singular Value Decomposition (SVD) is applied to the matrix to discover the relationships between words and documents. For IEA, 300 model essays are required for evaluation. The matrix of the input essay will be transformed by SVD technique and reproduce using the reduced dimensional matrices that are already built for the essay model. Then, IEA uses cosine correlation to measure the similarity of the reduced dimensional space constructed from a model essay.

Performance: Research claim that the order of the words is not being utilized because it is not important to describe the sense of an essay (Valenti & Cucchiarelli, 2003). In performance wise, a test conducted on GMAT essays using the IEA system resulted in percentages for adjacent agreement with human graders between 85%-91% (Valenti & Cucchiarelli, 2003).

### **2.1.3 E-Rater®**

E-Rater® focus on both content and style of the essay. The features that are included in E-Rater® are the syntactic module, discourse module, and topical-analysis module. These features will be the outputs of the training model and scoring. E-Rater® requires a set of sample essays for model building also. It collects 465 essays rated by humans with a 6-point holistic score for model building. For the syntactic module, a parser identifies syntactic structures, such as subjunctive auxiliary verbs and a variety of clausal structures, such as complement, infinitive, and subordinate clauses (Semire D. , 2006). Next, the discourse module uses a conceptual framework of conjunctive relations including cue words, terms, and syntactic structures to identify discourse-based relationship and organization in essays. Lastly, the topical analysis module identifies the usage of vocabulary and topical content (Semire D. , 2006). A good essay is similar to other good essays and the same goes for poor essays. E-Rater® uses the vector-spec model to capture topic and vocabulary usage. All the training essays will be converted into word frequencies vectors and then transformed into word weights. Word frequencies refer to the number of words in a paragraph divide by the number of the occurrence of a specific word. Besides, word weight is calculated by dividing the frequency by the total number of words in the essay. In order to score a test essay, the training essay will be converted to a weight vector. Then it will search for the most similar training essay by measuring the cosine between the training and test essay's vectors. Lastly, the score of the closest training essay will be assigned to the test essay.

Performance: The agreement rate between human expert and E-Rater® over 750000 GMAT essays have been scored is have a consistent score of 97% (Valenti & Cucchiarelli, 2003). By comparing human and E-Rater® grades across 15 test

questions, the empirical results range from 87% to 94% (Valenti & Cucchiarelli, 2003).

## ***2.2 Features***

The dataset that I am using for this project will be students written essays with score rated by humans. From each essay, we can extract many features that “define” the essay itself in order to perform machine learning. In our scope, we have stated that this project will be focusing on surface linguistic features instead of the content of the essays.

The first and most obvious feature that can be extracted from the essay is word count and sentence count. For instances, essay's length in terms of words and sentences, unique word count, average sentence length, average word length, and each punctuation count are the most common statistical value that we can calculate in an essay. Besides, some part-of-speech (POS) tag count might also be a very good feature to define the essay. PEG is one of the products that based on this analysis. They included a total of 28 proxy variables such as the number of paragraphs and subject-verb openings in the initial PEG system. PEG applied this analysis with some additional software like grammar checker, part-of-speech tagger, electronic dictionary, and parser (Chung & O'Neil Jr, 1997). Apart from that, the number of spelling mistakes is a possible important feature that should be included when grading an essay (Ramalingam, Pandian, Chetry, & Nigam, 2018).



A feature selection techniques call RELIEF to identify the ranking for each feature. The idea of RELIEF is assigning a relevance score by estimate the attribute and the selection is based on a threshold value (Madala, Gangal, Krishna, Goyal, & Sureka, 2018). The result of the experiment as Table 2.1 below. We can see that the usage of unique vocabulary is at the top of the list while the usage of long sentences comes in last.

Item	Quantity	Attribute
1	0.0065584	Vocabulary
2	0.0047464	Word Count Limit Ratio
3	0.0026625	Semantic Similarity Topic Essay
4	0.002551	Voice
5	0.0007761	Semantic Similarity
6	0.0007395	Spell Errors
7	0.0004207	Tense
8	0.0000431	Grammatical Errors
9	-0.0018502	Long Sentences

*Table 2.1: Feature ranking by RELIEF (Madala, Gangal, Krishna, Goyal, & Sureka, 2018)*

From another research, they calculated the contribution of each feature they have chosen towards the overall quadratic weighted kappa (QWK) (Shenoy). The results are shown in Figure 2.1 below proof that the common feature such as word count and sentence count has the highest kappa score among the other features.

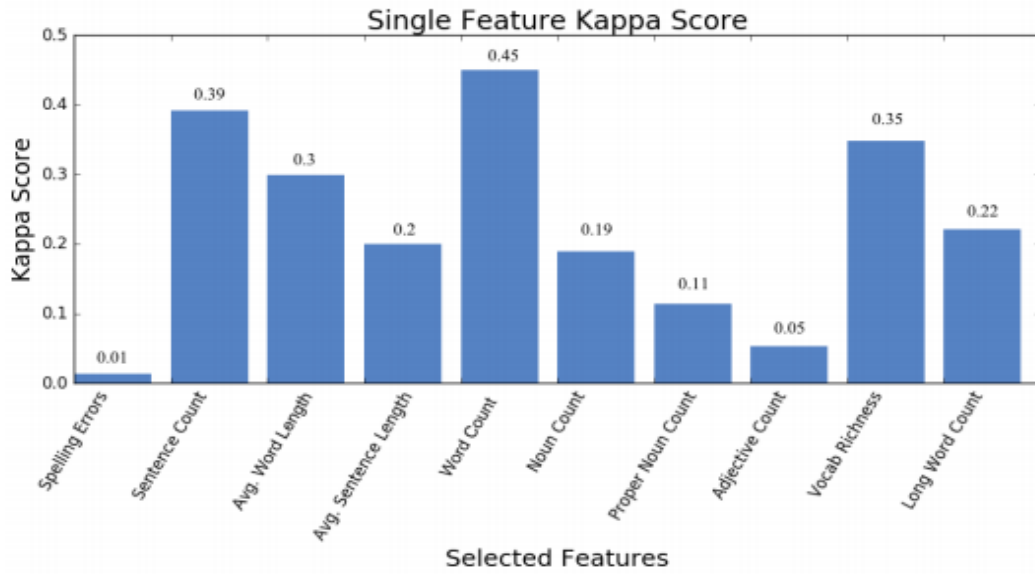


Figure 2.1: Features' Kappa Score

### 2.3 Classification Method and Algorithm

In an experiment, they have conducted an empirical analysis of machine learning models for automated essay grading (Madala, Gangal, Krishna, Goyal, & Sureka, 2018). For the analysis, they use same the datasets from “The Hewlett Foundation: Automated Essay Scoring”, a Kaggle’s competition which I will be using it as well. Only 3 sets of essays were chosen from the total of 8 sets of essays from the entire datasets for this experiment. Data normalization and scaling is very important before applying classification algorithms. During their data pre-processing stage, they implemented a logarithmic transformation to perform data normalization on the chosen 9 features in the range of 0 to 1.

The essay dataset will be classified into 4 classes which are A, B, C and D. A total of 3 classification algorithm were used in the experiments which are k-Nearest Neighbour

(kNN), Logistics Regression (LR) and Support Vector Machines (SVM). Figure 2.2 below summarized the accuracy of each algorithm for each essay set. The first observation here is that the accuracies for each algorithm in each dataset have only a maximum of 5% in difference. In Set 1, we can see that the accuracy of kNN is 82% (highest), 79.62% for SVM and 80.3% for LR. Apparently, in Set 7, kNN has the lower accuracy of 73.69% while SVM has the highest accuracy of 78.69% and LR comes in the middle of them with accuracy 77.32%. In the last essay set which is Set 8, kNN achieves the highest accuracy of 93.13%, followed by LR with the accuracy of 91.41% and SVM comes in last with an accuracy of 90.98%. Even though KNN have the most number of highest accuracy in the experiment which is Set 1 and Set 8, but in average, kNN has 82.94% accuracy, SVM has 83.09% (highest) and LR has 83.01% which only have a very small gap of not more than 1%.

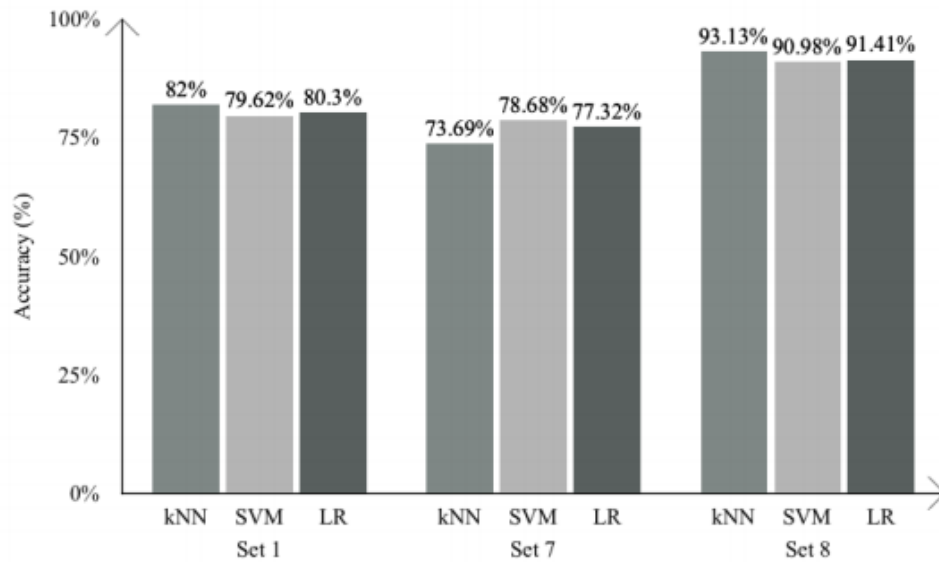


Figure 2.2: Models' accuracies for each set of essay (Madala, Gangal, Krishna, Goyal, & Sureka, 2018)

In the other experiment, SVM outperformed Multiple Linear Regression, Random Forest and kNN in most of the assessments in both default and after tuning the hyperparameter with the range of 41% to 63% of exact agreement between human rater and automated scoring machine (Chen, Fife, Bejar, & Rupp, 2016). Besides, SVM also has the highest score for quadratic-weighted kappa (QWK) in the range of 0.554 to 0.768 and 0.627 to 0.823 for Pearson correlation ( $r$ ).

## ***2.4 Support Vector Machine's Kernel***

SVM was introduced by Vapnik is a technique with the implementation of the kernel function. SVM has appeared as a crucial learning technique for solving classification and regression problems in several fields such as finance and text categorization (Apostolidis-Afentoulis, 2015). There are many kernels in SVM which are linear, radial basis function (RBF), polynomial and sigmoid kernel where different kernel may work well on different cases. For instance, the linear kernel works well when the parameter is linearly separable. Figure 2.3 below shows the difference between linearly separable attributes and nonlinearly separable attributes. In Figure 2.3, the points are not linearly separable, so we can apply “kernel trick” to transform all the data points into a new feature space to separate the data points. Figure 2.3 shows the “kernel tricks” that transform the feature space from 2-Dimensional to 3-Dimensional to classify the data.

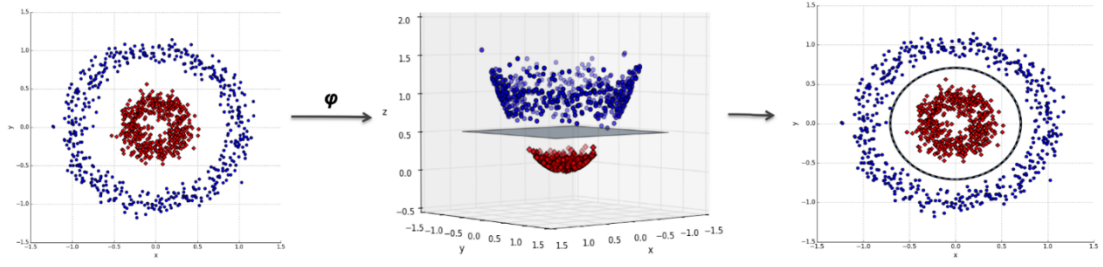


Figure 2.3: Transformation of feature space by applying kernel function (Cambridgespark, 2016)

An experiment tested various datasets on a different kernel with the same parameter found out that linear kernel has the best accuracy and lowest prediction time while RBF kernel takes the shortest training time (Pahwa & Sinwar, 2015). The summary of the result in average can be found in Table 2.2 below.

Kernels	Training Time (s)	Prediction Time (s)	Accuracy (%)
Linear	10.353	<b>4.078</b>	<b>88.206</b>
RBF	<b>4.927</b>	5.689	80.173
Polynomial	25.641	5.874	76.177
Sigmoid	5.664	7.146	84.532

Table 2.2: Summary of time taken to train, predict of kernels and their accuracy. (Pahwa & Sinwar, 2015)

There are some other reasons that RBF is a reasonable first choice kernel in most cases is that polynomial kernel has more hyperparameter than the RBF kernel which affects the complexity of model selection. Lastly, the RBF kernel has fewer numerical difficulties (Apostolidis-Afentoulis, 2015). In contrast, the RBF kernel may not

perform as good as the linear kernel in some cases such as when the number of features is very large (Apostolidis-Afentoulis, 2015).

## **CHAPTER 3 RESEARCH METHODOLOGY**

### ***3.1 Proposed Solution Overview***

AAS System is proposed and built to achieve the main purpose and objective of this project. Before we design the whole AAS System, we need to determine and build the four main essay evaluation model. The dataset used to build all the models is acquired from kaggle.com.

#### ***3.1.1 Building the essay evaluation models***

Firstly, the dataset that we acquired from kaggle.com contains redundant attributes where is it not needed for this project. The necessary data needed for this project are essay itself and the final score rated by human will be selected from the raw dataset. After all the redundant information have been removed, all unwanted characters and texts in the essay will be removed and eliminated to guarantee the cleanliness of data in order to achieve an accurate result. After that, word and sentence tokenization will then apply to the essay itself for feature extraction purposes later on as well as perform more data cleaning step like removing punctuations.

Next, all statistical representations of the essay that we have studied in the literature review will be extracted from the essay dataset. A full list of features and its description will be discussed in feature extraction later on. The statistical representation of the essay dataset will then use to build all the essay evaluation models. The essay will be

evaluated into 3 classes which are class 1 for bad, class 2 for medium and class 3 for a good essay.

#### ***3.1.1.1 Machine Learning Model***

Before building the machine learning model, the essay dataset split into train and test set. Then standardization is applied to the train and test set as the feature scaling method because we want to make all features fit into the same scale, while also maintaining their internal variance and not affecting their relative importance. After that, the training data is used to fit the SVM model while the test set is used to evaluate the fitted SVM model. Next, parameter tuning is performed in order to allow the SVM model's behavior to be adapted to the given data. The best performing set of parameters will be selected and used to build the machine learning model.

#### ***3.1.1.2 Non-Machine Learning Model***

This model evaluates the essay without any machine learning techniques. The method proposed for this model are: -

- Determine the separation point for each feature that separate the class of the essay
- Sum up all the features and determine the best separation point among all possible set of separation points
- Sum up all the features and assign weights to determine the best separation point among all possible set of separation points



### ***3.1.1.3 Models Combination 1: Weighted Mean Approach***

The first method to combine the machine learning model and non-machine learning models is weighted mean score approach.

### ***3.1.1.4 Models Combination 2: Machine Learning Model 2***

The second method to combine the machine learning model and non-machine learning models is to include the result computed by the non-machine learning model into the machine learning model as a feature.

## ***3.1.2 Design the AAS System***

After we have built all the essay evaluation models, we can design the AAS system that allows the user to evaluate their essay. Firstly, the user is allowed to upload a single essay in (.txt) format or multiple essays in (.csv) format. Then, the user is allowed either to use the suggested value for the separation points for the non-machine learning model and weights for the model combination 1 or customize it. After that, the user may submit the essay and all the inputs to the system to evaluate the uploaded essays. The system will retrieve the essay and perform data pre-processing and feature extraction on the essay. Then, the essay will be evaluated by all 4 models. Next, all the extracted features and computed results will be displayed on the user interface. The user is allowed to download the output as well. Last but not least, Figure 3.1 refers to the flowchart of the AAS System from user's perspective while Figure 3.2 shows the AAS System backend's flowchart.

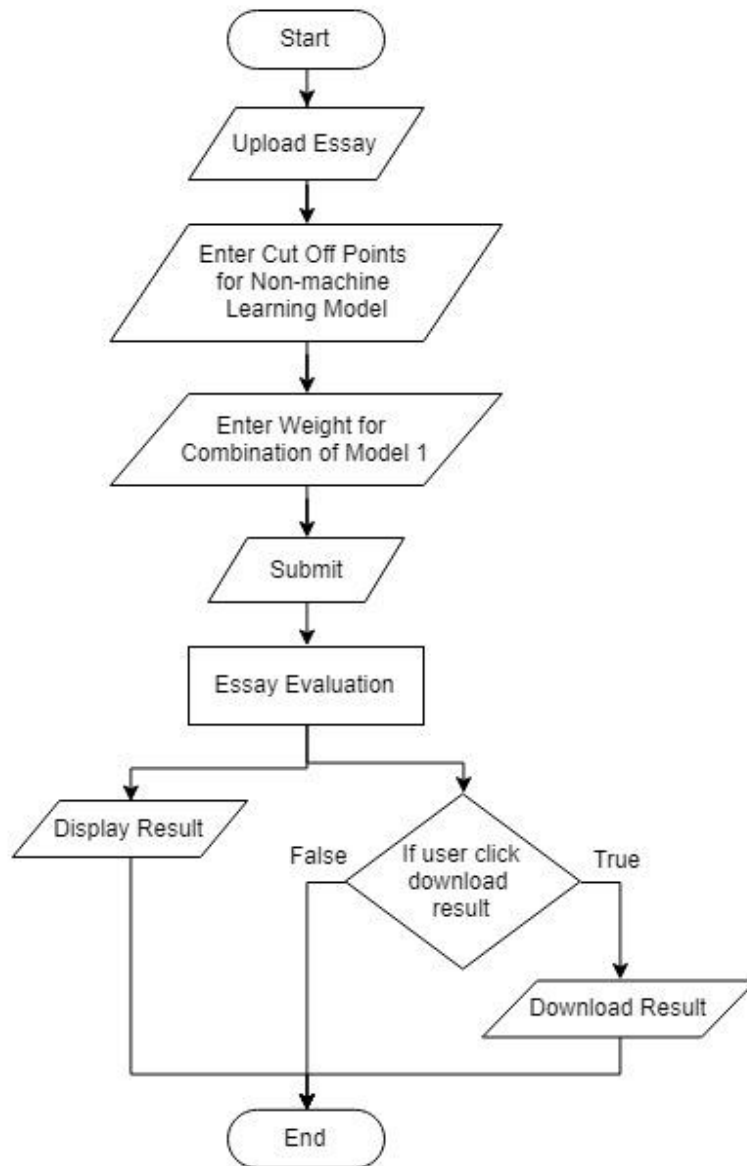


Figure 3.1: An overview flowchart of AAS System Design from user's perspective

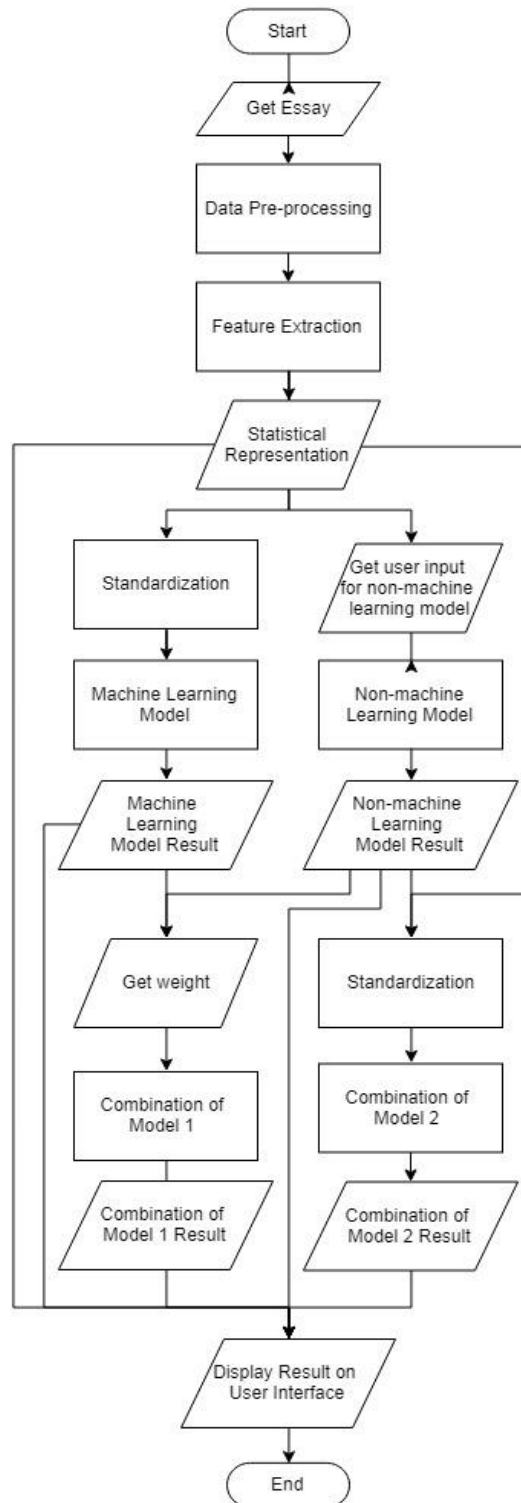


Figure 3.2 An overview flowchart of AAS System backend

## ***Programming Language, Packages and Framework***

The selected programming language to build this system is Python. Python has large data mining and machine learning libraries to perform statistical natural language processing (SNLP). The main library that we will be using to perform the knowledge discovery process for this system is Natural Language Toolkit (NLTK) and Scikit-learn. Besides, Django will be selected as the framework to build this system's user interface (UI).

### ***3.2 Dataset***

In this research-based project, I have acquired a huge essay dataset from a past competition call The Hewlett Foundation: Automated Essay Scoring provided by Hewlett Foundation on kaggle.com. The attributes that we are going to use in this project are `essay_set`, `essay`, and the class: `domain1_score`. The description or definition of the data can be found in Table 3.1 below. We will utilize all sets of essays (8 essay set) for this project. Each essay set have different range of scores. We will rescale all the scores from 1 to 3 in order to build the 3 classes SVM classifier. The range of average essay length is 250 - 650 words per essay. The dataset contains total 12,987 essays and we decided to split it into 75% for training and 25% for testing. S summary of the data shown in Table 3.3 and Table 3.4.

<b>Attributes</b>	<b>Definition/Description</b>
<code>essay_set</code>	An id for each set of essays. Each set represent a unique prompt.
<code>essay</code>	The ascii text of a student's response.

score	Resolved score between the raters (all essays have this).
-------	---

*Table 3.1 Description of each required attribute in dataset*

Set	Number of essay	Range of score
1.	1,783	2 - 12
2.	1,800	1 - 6
7.	1,569	0 - 30
8.	723	0 - 60
Total:	5,875	

*Table 3.2: Summary of number of essay and the score range for each essay set (Kaggle, n.d.)*

Dataset	Number of essay
Training set	4,406
Test set	1,469
Total:	5,875

*Table 3.3: Summary of training and test set.*

### 3.3 Features

A total of 14 features that we are going to extract from the essays are described as below: -

No.	Features	Description
1.	Word count	Total number of words in the essay. This feature indicates the volume of the essay. Besides, this feature had achieved the highest kappa score among all the features in the research. (Shenoy)

2.	Long word count	Total number of words that contain 7 or more characters in the essay. The more the long word used by the writer, the greater the language depth of the writer. (Shenoy)
3.	Average word length	The average length of each words in the essay. The longer average length of each words may indicate the complication of the essays.
4.	Unique word count	Vocabulary richness; Total number of unique words in the essay. This feature may reveal the vocabulary richness of the writer.
5.	Sentence count	Total number of sentences in the essay. This feature may represent the volume of the essay in a different way other than word count. It also achieved a second highest kappa score in the research. (Shenoy)
6.	Long sentence count	Total number of sentences that contain more than 15 words. Long sentences are hard to understand due to its complexity which might be less coherent and effective.
7.	Average sentence length	Average number of words count in each sentences. This feature may represent the writing habit of the writer on sentence construction.
8.	Noun usage count	Total number of noun used in the essay. Number of part-of-speech usage may indicate the writing style of the writer.
9.	Proper Noun usage count	Total number of proper noun used in the essay. Number of part-of-speech usage may indicate the writing style of the writer.
10.	Adjective usage count	Total number of adjective used in the essay. Number of part-of-speech usage may indicate the writing style of the writer.
11.	Verb usage count	Total number of verb used in the essay. Number of part-of-speech usage may indicate the writing style of the writer.
12.	Adverb usage count	Total number of adverb used in the essay. Number of part-of-speech usage may indicate the writing style of the writer.
13.	Tense ratio	The number of dominant tense divided by the total number of verbs. The consistency of tense usage indicates the writing skills where the higher the consistency of tense usage, the better the writing skills. (Cite)
14.	Error count	Total number of writing error such as spelling errors and grammar errors occur in the essay. This feature may reveal the

		english skill level in term of constructing a sentence and carefulness of the writer.
--	--	---

*Table 3.4: Selected features and its description*

### ***3.4 Imbalance Class Distribution Dataset***

Machine learning model requires splitting the dataset to train and test set. In the training set, the class is imbalance where class 1 has 330 instances, class 2 has 3342 instances, and class 3 has 734 instances. When we have a highly skewed class distribution in our dataset, resample technique is applied to ensure that the classifier put the same amount of attention to each class (Gary M and McCarthy, Kate and Zabar, Bibi, 2007).

There are two main methods to overcome this issue, which are undersampling and oversampling. Undersampling reduces the number of instance in the majority class in training set while oversampling increase the number of instance in the minority class (Liu, 2004). Undersampling will greatly reduce the size of the training set which will reduce the computation time of training a model. However, we might lose the information that is important or useful to our classifier from the training set when building the model (Liu, 2004). This may cause a very bad result from the model.

Oversampling technique increases the size of data rather than removing any information from the data. This will definitely increase the training time and memory required during the training period (Liu, 2004). There is no conclusive results state that whether undersampling or oversampling is better to improve a classification model (Liu, 2004). If we do not take time constraint into consideration, oversampling is better

than undersampling technique because it does not remove any information from the training set.

### ***3.5 Machine Learning Model 1: SVM Classification***

The main concept of SVM classifier is to separate 2 groups of data point with a hyperplane (solid line) that maximizing the distance from the hyperplane to the nearest data points, which known as support vectors. Figure 3.3 below shows the example of SVM classifier separating a binary class with maximum margin.

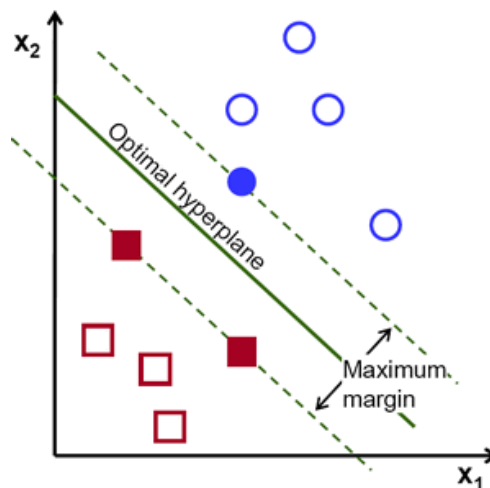


Figure 3.3 Example of SVM separating data points with maximum margin (Mathieu, 2018)

#### ***3.5.1 Multiclass Classification Approach***

There are two methods to perform SVM multiclass classification which are the one-vs-rest (OVR) and one-vs-one(OVO). The OVR method is probably the earliest implementation for SVM multiclass classification (Hsu & Lin, 2002). In the OVR approach, the SVM will fit the data into # number of the classifier depends on the



number of classes. For example, the machine learning problem in this project consists of 3 classes then the SVM will create 3 classifiers. The classifiers are shown in Table 3.5 below while a graphical example is shown in Figure 3.4. Then, an instance will be fitted into each classifier and determine the class of the instance based on the decision function. The decision function is the distance calculated from the hyperplane to the instance perpendicularly. The classifier with the highest decision function value will be selected as the class of the instance.

Classifier	Class
1	1 vs 2, 3
2	2 vs 1, 3
3	3 vs 2, 3

Table 3.5 Example of classifiers constructed in OVR method

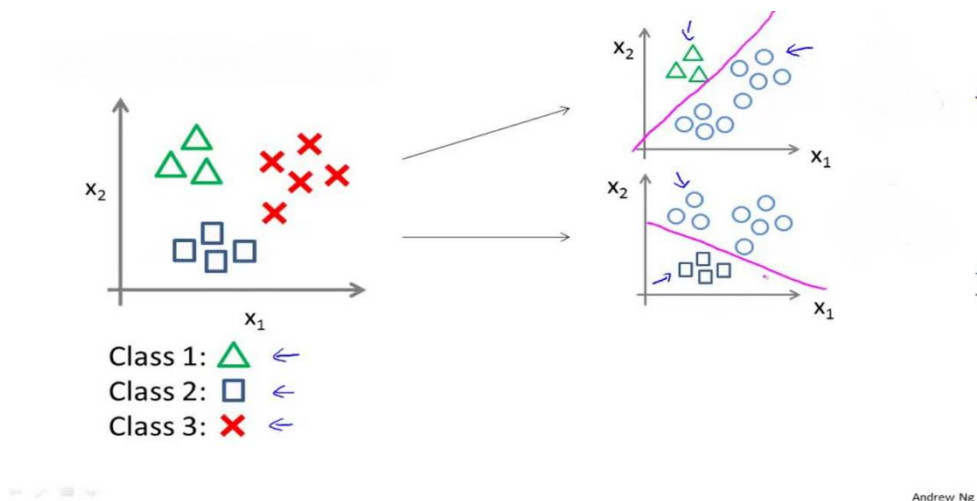


Figure 3.4 Graphical demonstration of OVR method

Next, the OVO approach will build  $k(k - 1)/2$  number of classifier where  $k$  refers to the number of classes. For instance, a machine learning problem with 3 classes will produce 3 classifiers in OVO approach. The classifiers are shown in Table 3.6 and the

graphical example is shown in Figure 3.5. After all of the classifiers are constructed, a voting strategy will be used to choose the class as the prediction (Hsu & Lin, 2002). The voting strategy says that if  $x$  is in the  $i$ th class, then the vote for the  $i$ th class is added by one. Otherwise,  $j$ th is increased by one. Then,  $x$  will be predicted to belong to the class with the highest vote. In the case that more than one classes have the same amount of vote, the class with the smallest index will be selected.

Classifier	Class
1	1 vs 2
2	1 vs 3
3	2 vs 3

Table 3.6 Example of classifiers constructed in OVO method

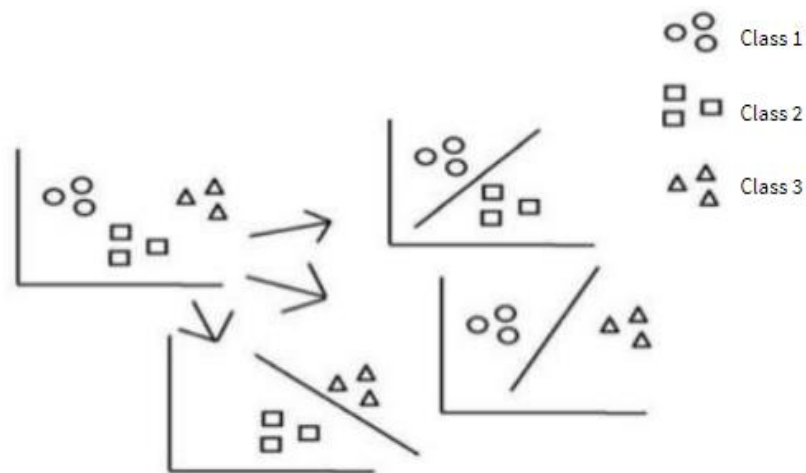


Figure 3.5 Graphical demonstration of OVO method

### ***3.5.2 Kernels and Parameters***

When you are given data that is linearly inseparable, the function kernel is to take the data as input and transform it into the required form. Kernels function is used to map the input space into a linear separable feature space where the linear classifier is applicable again (Hofmann, 2006). The kernels are Linear, Polynomial, Radial Basis and Sigmoid kernel function. A graphical example is shown in Figure 2.3.

The purpose of the C parameter is to set the SVM how many you want to avoid misclassifying training example (Hsu & Lin, 2002). The C parameter controls the tradeoff between the classification of training points accurately and a smooth decision boundary. A large C value of SVM will select a smaller margin hyperplane if it can classify all the training points correctly. In the other hand, a very small C value of SVM will prefer a larger margin hyperplane even if it misclassified more training points (Hsu & Lin, 2002). The C parameter is applicable to all kernel functions.

Next, Gamma ( $\gamma$ ) parameter defines how the influence of a single training example reaches. If the value of Gamma is high, then the decision boundary will depend on data points close to the decision boundary and nearer data points carry more weights than far data points. In the other hand, if the value of Gamma is low, then the far data points carry more weights than nearer points and thus our decision boundary becomes more like a straight line. This parameter is applicable to all kernel functions except for the linear kernel.

Lastly, the degree parameter controls the flexibility of the decision boundary. The higher the degree, the more flexible the decision boundary. This parameter only applies to the polynomial kernel function.

### ***3.6 Quantitative Analysis Model***

The machine learning model is not a perfect solution because it will not give us a 100% accuracy for our essay evaluation. That is why a non-machine learning model is introduced. The whole idea of the non-machine learning model is to determine the best separation point in order to classify the data. We proposed 3 methods to build the non-machine learning model. Note that the evaluation scheme does not justify by experts that possess English language proficiency but it is a proof of concept applied to improve the AAS result.

#### ***3.6.1 Determine the separation point for each feature that separate the class of the essay***

The first proposed method is to determine the separation or cutoff points for each feature. This project aims to classify the essay into one of the 3 classes (bad, medium & good). Due to time constraint and limited resources, we are not able to try all combinations of separation point to determine the best separation point for each feature. To overcome this issue, we have decided to determine the separation points for each feature based on: -

- The midpoint between the mean value of each classes
- The midpoint between the median value of each classes

- The ratio of number of instances for each class

This approach will evaluate a total of 14 features which are exactly the same as the features used in the machine learning model. Each feature evaluation will result in a point of 1, 2 or 3 where the higher point indicates the better essay. Then, we will sum up all the point achieved from each feature and divide by the number of features, which is 14. The formula to compute the final score or the class for the essay is: -

$$class = \frac{\text{number of points achieved from each feature}}{\text{number of feature}} \equiv \frac{n}{14}$$

### ***3.6.2 Sum up all the features and determine the best separation point among all possible set of separation points***

This method proposed to sum up all the features and determine the best separation point by trying all the possible set of separation points. Then, the set of separation points with the highest accuracy will be selected. This method proposed to simplify the work where the system only needs to determine 1 set of separation points instead of 14.

### ***3.6.3 Sum up all the features and assign weights to determine the best separation point among all possible set of separation points***

The difference between this method and the previous one is that we assign a weight for each feature before summing up all the features for each essay. We believe that

each feature has its own weight from an evaluator perspective. In this case, two sets of weight are defined based on the research in the literature review. As mentioned, the set of separation points with the highest accuracy will be selected.

### ***3.7 Combination of Model Approach***

#### ***3.7.1 Method 1: Weighted Mean Approach***

In this research project, we do not rely on machine learning or non-machine learning individually. Instead, we will make use of both of the model which can improve the reliability of the essay scoring result. Weight Mean is a different kind of average computing method where we can assign a weight to each data points. This method will utilize the results from machine learning and non-machine learning model by computing the weighted mean result from the machine learning and non-machine learning model. The benefit of this formula is that it allows the user to calibrate the weight for each data point and find the best weight pair that gives us a more reliable result. In order to determine the best weight for each result from machine learning and non-machine learning model, we will test out all possible set of weights. The Weighted Mean Formula is written as below: -

$$\bar{x} = \frac{\sum_{i=1}^n (x_i w_i)}{\sum_{i=1}^n w_i}$$

#### ***3.7.2 Method 2: Machine Learning Model 2***

In the consideration of the non-machine learning model's score may be a very good feature for the machine learning model, we propose this method to improve the

classifier. This approach is to utilize the result of non-machine learning by including the result into the classifier as one of the features. In this method, the result of the machine learning model may be improved and twisted by the non-machine learning model.

### ***3.8 Model Evaluation***

In this project, all three classes are important so during the model evaluation, we will aim to achieve the model that is not biased to one of the classes. Overall accuracy, precision, recall, and F1-score will be chosen to measure the result of all the model. The formula and the description of accuracy, precision, recall, and F1-score are presented in Table 3.6. In a classification model, there will be 4 possible scenarios: -

1. True Positive (TP): These refer to the positive tuples that were correctly labeled by the classifier. Let TP be the number of true positives.
2. True Negative (TN): These are the negative tuples that were correctly labeled by the classifier. Let TN be the number of true negatives.
3. False Positive (FP): These are the negative tuples that were incorrectly labeled as positive. Let FP be the number of false positives.
4. False Negative (FN): These are the positive tuples that were mislabeled as negative. Let FN be the number of false negatives.

Since our classification model is multiclass classification, each class will be represented by one confusion matrix where the positive value is the selected class and negative value is the rest of the class. This is for the case of calculating precision, recall, and F1-score. The average value of the precision, recall, and F1-score is

achieved by finding the mean, weighted by their supports where supports are the number of actual data lie on the actual class. Besides, the calculation of overall accuracy will be the number of tuples correctly classify divide by the total number of tuples. Table 3.7 below is the summary of the model evaluation methods: -

Measurement	Formula	Description
Overall Accuracy	$\frac{TP + TN}{ALL}$	Percentage of test set tuples that are correctly classified.
Precision (P)	$\frac{TP}{TP + FP}$	The percentage of tuples that the classifier labeled as positive are actually positive.
Recall (R)	$\frac{TP}{TP + FN}$	The percentage of positive tuples did the classifier label as positive.
F1-score	$\frac{2PR}{P + R}$	The weighted average of the precision and recall, where an F1-score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal.

Table 3.7: Summary of model evaluation methods

Last but not least, cross-validation will be used to determine whether the machine learning model 1 and 2 are overfitting or not.

### 3.9 AAS System Design

A web UI will be created for the AAS System using Python's web framework, Flask. This prototype allows the user to upload their essay for evaluation, edit the features scoring scheme for non-machine learning model and also download their result. User is allowed to observe all the extracted features and results of each model. For (.csv)



file type, the system will show the number of essay in each class and minimum, maximum and average of each feature.

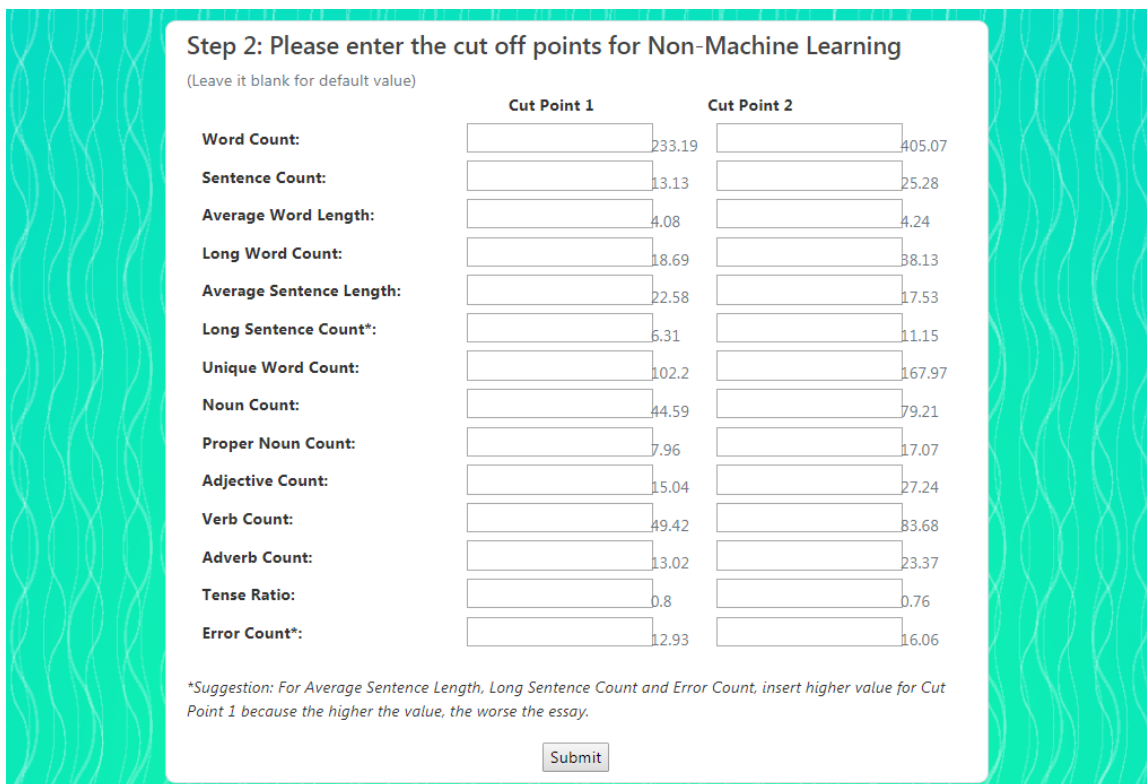


Automated Article Scoring System Home

Step 1: Please upload your article here (.txt or .csv)

Choose File No file chosen

Figure 3.6 UI of user upload essay



Step 2: Please enter the cut off points for Non-Machine Learning

(Leave it blank for default value)

	Cut Point 1	Cut Point 2
Word Count:	233.19	405.07
Sentence Count:	13.13	25.28
Average Word Length:	4.08	4.24
Long Word Count:	18.69	38.13
Average Sentence Length:	22.58	17.53
Long Sentence Count*:	6.31	11.15
Unique Word Count:	102.2	167.97
Noun Count:	44.59	79.21
Proper Noun Count:	7.96	17.07
Adjective Count:	15.04	27.24
Verb Count:	49.42	83.68
Adverb Count:	13.02	23.37
Tense Ratio:	0.8	0.76
Error Count*:	12.93	16.06

\*Suggestion: For Average Sentence Length, Long Sentence Count and Error Count, insert higher value for Cut Point 1 because the higher the value, the worse the essay.

Submit

Figure 3.7 UI of user customize the scoring scheme for non-machine learning model

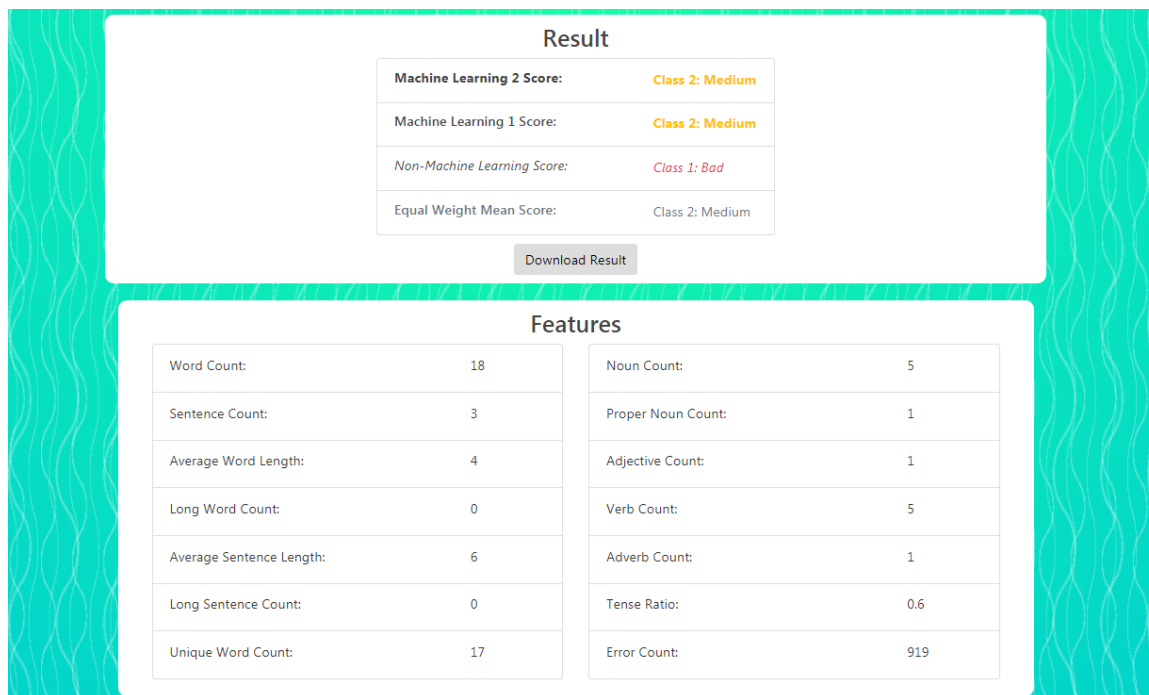


Figure 3.8 UI of essay result (.txt)

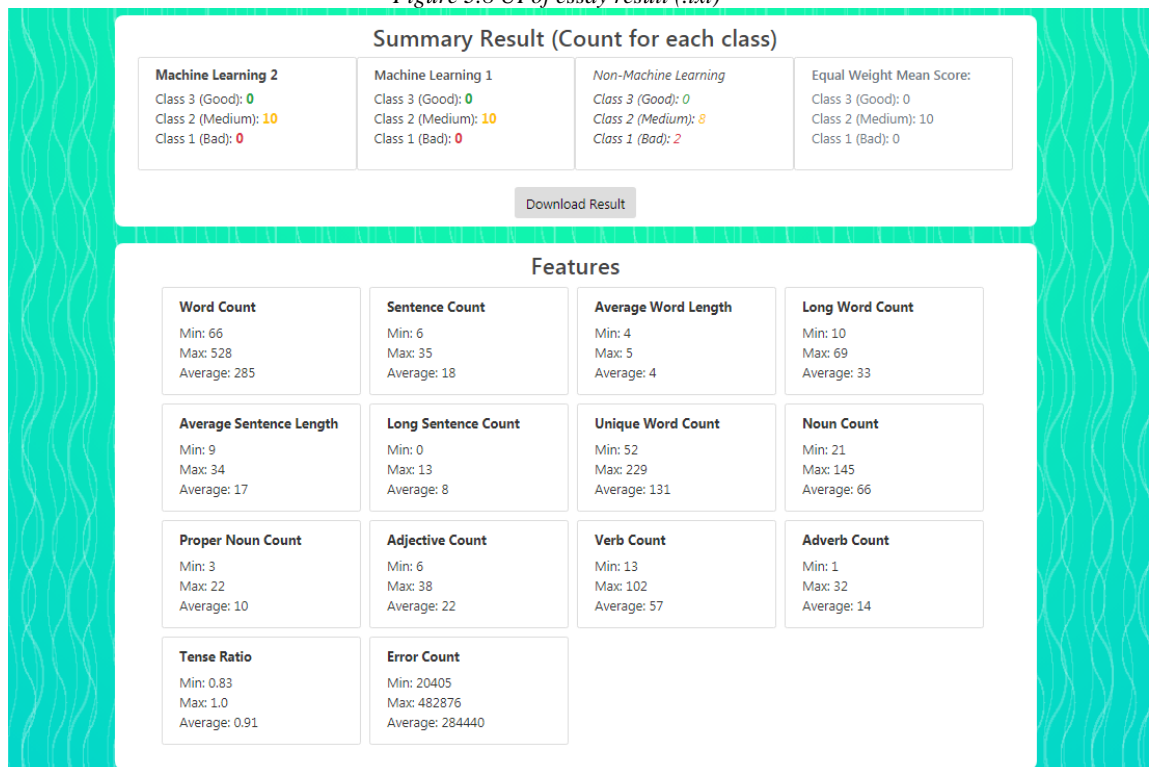


Figure 3.9 UI of essay result (.csv)

## CHAPTER 4 IMPLEMENTATION

### 4.1 Data Pre-processing

#### 4.1.1 Replace Anonymization

All the essay in the dataset will be loaded into a list in Python. The content of essays may carry some personally identifying information such as a person's name, date, location, and organization. The provider of the dataset had identified and replaced all the personally identifying information by using Named Entity Recognizer (NER) created by Stanford Natural Language Processing and other types of approaches. Some examples are provided in Table 4.1 below. In this case, we will use the regular expression to identify all the word that starts with "@" and replace it. All the anonymization will be replaced by the word "Sunday" since all the named entity will be recognized as a proper noun in NLTK's POS-tagger later on. Except for "@TIME" and "@NUM", this anonymization will be replaced by "123" because both of them are numbers. This may affect the content and meaning of the sentence but it does not matter in this project because this project focuses on shallow linguistic features only.

Before Anonymization	After Anonymization	After Replacement
I attend Springfield School...	I attend @ORGANIZATION1	I attend Sunday
my phone number is 555-2106	my phone number is @NUM1	my phone number is 123
once my family took my on a trip to Springfield.	once my family took me on a trip to @LOCATION1	once my family took me on a trip to Sunday

Table 4.1: Example of Anonymization done by the dataset provider along the result after anonymization replacement (Kaggle, n.d.)

### 4.1.2 Tokenization

Tokenization, a very important feature supplied by NLTK library for this project. It can split the essays by words or sentences which is a very crucial step before extracting the features. We have selected TweetTokenizer for this project's word tokenization. Figure 4.1 below shows the difference in the result of tokenization between TreebankWordTokenizer, WordPunctTokenizer, WhitespaceTokenizer, and TweetTokenizer.

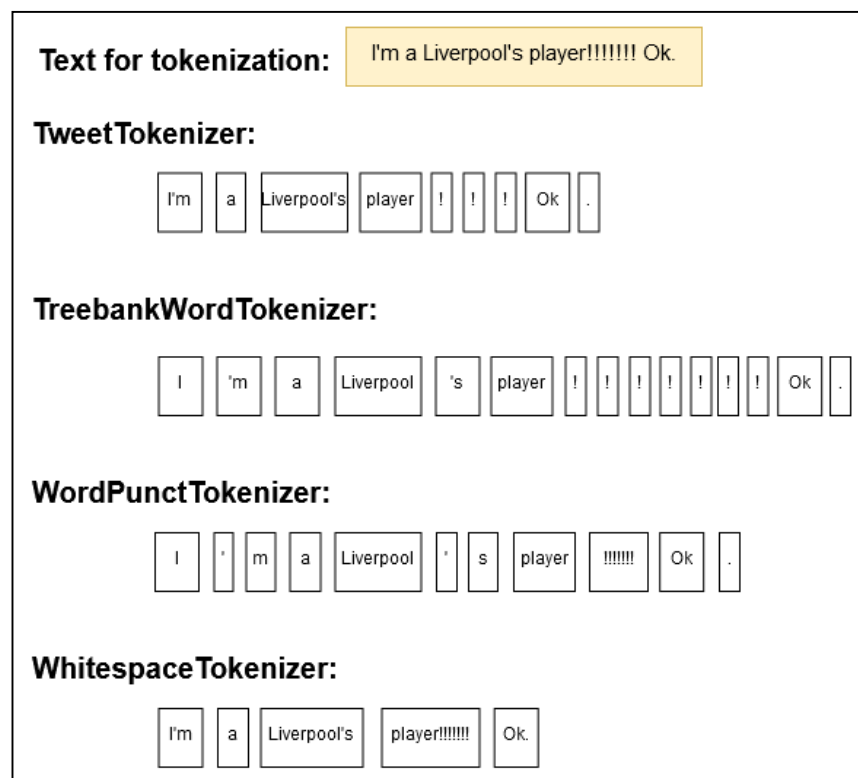


Figure 4.1: Example of each sentence tokenizer in NLTK (Word Tokenization with Python NLTK, n.d.)

TweetTokenizer will separate the punctuations from the words that they are not meant to be together such as a comma or a full stop at the end of a sentence but not the

apostrophe that appears in a contraction of words. This feature will benefit features extraction stage such as word count. Unlike TreebankWordTokenizer and WordPunctTokenizer, they do not satisfy the requirement above. In addition, WhitespaceTokenizer is basically split words by whitespace in between words which will cause all the punctuations will not be separated. In this case, this prevents problem occurs when counting the number of unique words during features extraction stage. Besides, we use the NLTK default sentence tokenizer to tokenize our essays in sentences.

#### ***4.1.3 Remove Punctuation***

A package called “string” will be used in this step. We will loop through the word tokenization list to identify the all the punctuations and remove it except for the apostrophe (') in between a contracted word.

## ***4.2 Features Extraction***

The method used to extract all 14 features are as below: -

<b>Features</b>	<b>Method</b>
Word count	1. Count the length of the word tokenized list.
Long word count	1. Count the number of word that contains more than 7 characters.
Average word length	1. Sum up all the number of characters in the essay and divide by word count.

Unique word count	<ol style="list-style-type: none"> <li>1. Stem the word in word tokenized list using NLTK's PorterStemmer. This method can achieve the root word by cutting down redundant character such as "s", "ing", "ed" and etc.</li> <li>2. Count the length of the stemmed word tokenized list by using set() function. set() function will return unique element in a list.</li> </ol>
Sentence count	<ol style="list-style-type: none"> <li>1. Count the length of the sentence tokenized list.</li> </ol>
Long sentence count	<ol style="list-style-type: none"> <li>1. Count the number of word that contains more than 15 words.</li> </ol>
Average sentence length	<ol style="list-style-type: none"> <li>1. Sum up all the number of words in the essay and divide by sentence count.</li> </ol>
POS-tag -Noun usage count -Proper Noun usage count -Adjective usage count -Verb usage count Adverb usage count	<ol style="list-style-type: none"> <li>1. Determine the POS tag of each word by using NLTK's default POS tagger.</li> <li>2. Count the number of word that are:-               <ol style="list-style-type: none"> <li>a. Noun</li> <li>b. Proper Noun</li> <li>c. Adjective</li> <li>d. Verb</li> <li>e. Adverb</li> </ol> </li> </ol>
Tense ratio	<ol style="list-style-type: none"> <li>1. Compare and acquire the dominant tense by comparing the number of tense used in the essay</li> <li>2. Divide the number of dominant tense by the verb usage count</li> </ol>
Error count	<ol style="list-style-type: none"> <li>1. Use the Language Check Package to identify the error occur in the essay.</li> <li>2. Count the number of error.</li> </ol>

Table 4.2: Extracted features and the procedure of extracting it from the essay

### 4.3 Building Machine Learning model

Firstly, we rescale all the classes manually into a range from 1 to 3 which is a total of 3 classes. Then, we split the data into the training set and test set with a ratio of 0.75:0.25. Then, we perform standardization by using Scikit-learn's StandardScaler(). This scaler will transform all the values such that all values in each feature will have a

mean value of 0 and a standard deviation of 1. Data standardization can reduce the training time of our classifier as well as produce a better prediction accuracy. Next, we will fit the training set into the SVM to train the classification model. The test set will be used to evaluate the model. Figure 4.2 below shows the flowchart of building the machine learning model.

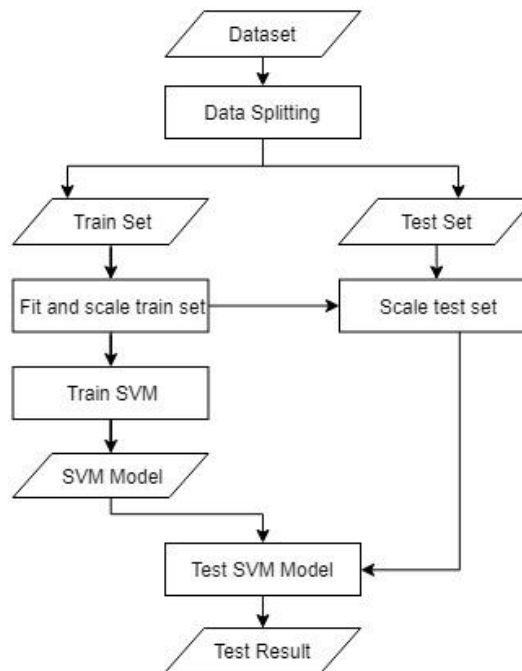


Figure 4.2 Flowchart of building the machine learning model

#### 4.3.1 Oversampling

The method we implement in this project to oversample the training data is random oversampling. Random oversampling is a simple yet effective approach to resampling the data in the training set. This approach will randomly duplicate the instance in minority class until all minority classes have the same ratio as the majority class.

### 4.3.2 Parameter Tuning

In order to get the best of the SVM classification model, we perform parameter tuning for both with and without oversampling data. Scikit-learn provided a package call GridSearchcv where it allows us to execute and evaluate all possible combination of parameters with cross-validation. 3-fold cross-validation is used in this parameter tuning process. We tune the C parameter and Gamma parameter in a range of [0.1,1,10] while we apply the range of [2,3,4] to tune the degree parameter for the polynomial kernel. Due to limited time and resource, we do not consider any further value for parameter tuning process. The accuracy of the test result will be used to determine the best parameter that suits our data most. The parameter and its range that is going to be tuned shown in Table 4.3.

Parameter	Tuning Range	Kernel applied
C	[0.1,1,10]	all
Gamma, $\gamma$	[0.1,1,10]	RBF, polynomial and Sigmoid
Degree	[2,3,4]	Polynomial

Table 4.3 Parameter applied for tuning and its range

### 4.3.3 Overfitting

In order to determine whether the machine learning model 1 is overfitting or not, we perform 10-fold cross-validation to examine the model.



## ***4.4 Building Non-machine Learning Model***

The dataset is not split into train and test set to build this non-machine learning model because we want to acquire the whole information from the dataset.

### ***4.4.1 Determine the separation point for each feature that separate the class of the essay***

The first two methods to determine the separation point for each feature that separates the class of the essay are separate by mean and median. The steps to calculate the midpoint between the mean/median value for each class are the same. The steps are: -

1. Split the data according to the class.
2. Determine the mean/median of each class.
3. Sum up the mean/median of class 1 & 2 and divide by 2 to get the midpoint.
4. Repeat step 3 for class 2 and 3.

Next, the third method we proposed is to determine the separation point for each feature is by the ratio of number of instances for each class. The steps to build this model are: -

1. Split the data according to the class.
2. Divide the number of instance for each class over the total number of instance to acquire the ratio for each class.
3. Get the range value by minus the minimum value from the maximum value in the feature.
4. Next, times the value from step 3 with the ratio for class 1.

5. Then, add up the value from step 4 and the minimum value in the feature to acquire the first separation point.
6. Repeat step 3 to 5 to calculate the second separation point but with the sum of the ratio of class 1 and 2.

#### ***4.4.2 Sum up all the features and determine the best separation point among all possible set of separation points***

Instead of looking for the separation points for each feature, this method proposed to sum up all the features first then only we determine the separation points. The value of long sentence count and error count feature will convert to negative value because these features are inversely proportional to the quality of the essay. The steps are: -

1. Standardize the feature using StandardScaler() from Scikit-learn.
2. Normalize the the result in step 1 with Min-max normalization to reduce the size of range.
3. Sum up all the features, let's call it X\_sum.
4. Define all possible set of separation point within the range in of X\_sum.
5. Apply each possible set of separation point to the data and compute the result.
6. The set of separation with highest accuracy will be selected.

#### ***4.4.3 Sum up all the features and assign weights to determine the best separation point among all possible set of separation points***

This method is similar with the previous method except we are assigning weight to each feature before we sum up all the features. The steps are: -

1. Standardize the feature using StandardScaler() from Scikit-learn
2. Normalize the the result in step 1 with Min-max normalization to reduce the size of range
3. Assign weights for each feature
4. Sum up all the features, let's call it X\_sum
5. Define all possible set of separation point within the range in of X\_sum
6. Apply each possible set of separation point to the data and compute the result
7. The set of separation with highest accuracy will be selected

In this approach, we will assign 2 set of weight and evaluate the result. The difference between set 1 and 2 is that the range of weight in set 2 is significantly large compare to set 1. The weights are shown in Table 4.4.

Feature	Weight Set 1	Weight Set 2
Word count	9.0	45.0
Long word count	8.0	40.0
Average word length	3.0	1.0
Unique word count	5.0	5.0
Sentence count	2.0	0.5
Long sentence count	1.0	0.2
Average sentence length	10.0	50.0
Noun usage count	4.0	4.0
Proper Noun usage count	4.0	4.0
Adjective usage count	4.0	4.0
Verb usage count	4.0	4.0

Adverb usage count	4.0	4.0
Tense ratio	7.0	35.0
Error count	6.0	8.0

*Table 4.4 Weights implemented for the weighted mean approach*

## ***4.5 Building the Model Combination 1 and 2***

The machine-learning and non-machine learning mode have to be done before we can build a combination of model 1 and 2. This is because this model requires the result from the machine learning model and non-machine learning model.

### ***4.5.1 Model Combination 1: Weighted Mean Approach***

After we have computed the result from machine learning and non-machine learning model. We can apply the weighted mean formula to compute a final combined result. To determine the best weight empirically, we will execute the formula with a set of weight. The weight with the highest accuracy score will be selected for this model. The set of weights that is going to be applied are (2, 1), (1.8, 1), (1.5, 1), (1.3, 1), (1, 1), (1, 1.3), (1, 1.5), (1, 1.8), (1, 2).

### ***4.5.2 Model Combination 2: Machine Learning Model 2***

In this approach, we include or append the result from the non-machine learning model as a feature of the essay to build another machine learning model. Similar to machine learning model 1, we perform parameter tuning to acquire the parameter that suits our

data best. The steps and procedures are the same as Machine Learning Model 1. We will only build 1 model for this approach which depends on the result of machine learning model 1. The flowchart of building this model is shown in Figure 4.3. Cross-validation will also be implemented to determine whether this model is overfitting or not.

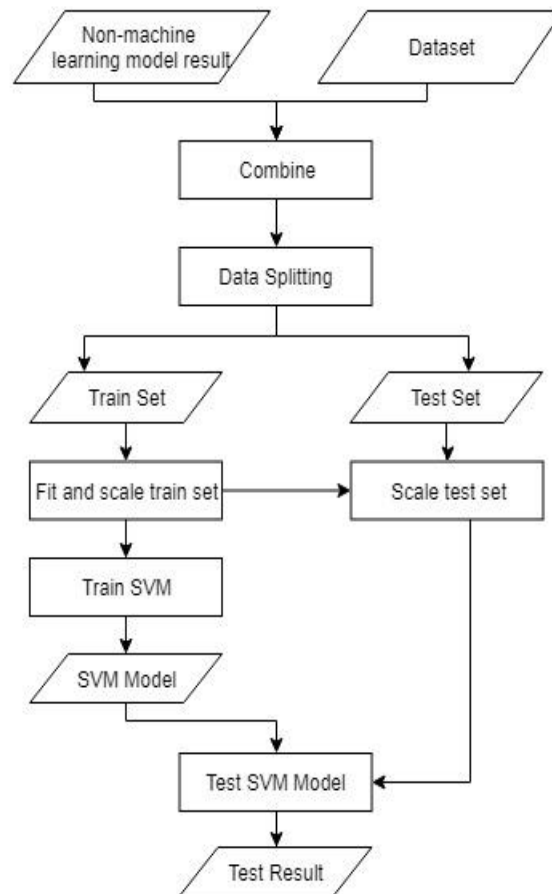


Figure 4.3 Flowchart of building the non-machine learning model

#### 4.6 Implementation Summary

First, we perform data pre-processing such as replace the anonymization with a proper word. Next, word and sentence tokenization is applied to the essay before removing

all the punctuation. Lastly, all 14 features are extracted from the essay. Next, we have applied a different approach to build the 4 essay evaluation models. The machine learning model has been built with and without oversampling technique as well as parameter tuning to find the parameters that give the best result. Next, for the non-machine learning model, we implemented 3 approaches which are to determine separation point for each feature with mean, median and ratio of each class and sum up all the features and find the best separation point with and without weight for each feature. On top of that, we proposed to implement 2 methods to combine the machine learning and non-machine learning model which are weighted mean approach and machine learning model 2. The weighted mean approach takes the result from both models and find the average result with different weighting while machine learning model 2 includes the result from non-machine learning as a feature of the essay to build the model.

## CHAPTER 5 IMPLEMENTATION RESULT

### EVALUATION

#### 5.1 Machine Learning Model

We are going to compare the result of the model with and without oversample the training dataset after parameter tuning. In other words, the result of the models that we are going to compare is using the parameters that give the best result. Table 5.1 below shows the best parameter values and the result of parameter tuning which is the average accuracy for each model. Figure 5.1, 5.2 and 5.3 shows the result of each model.

SVM Model	Kernel	C	Gamma, $\gamma$	Average Accuracy
Without oversampling	RBF	1	0.1	81%
Oversample train set only	RBF	10	1	77%
Oversample whole dataset	RBF	10	10	96%

Table 5.1 Result of parameter tuning

Train Accuracy: 0.8293236495687698				
Test Accuracy: 0.8230088495575221				
	precision	recall	f1-score	support
1	0.77	0.46	0.58	110
2	0.83	0.96	0.89	1114
3	0.74	0.37	0.49	245

Figure 5.1 Result of the model without oversampling

Train Accuracy: 0.9989028525832835				
Test Accuracy: 0.763784887678693				
	precision	recall	f1-score	support
1	0.55	0.45	0.50	110
2	0.82	0.89	0.85	1114
3	0.50	0.35	0.41	245

Figure 5.2 Result of model with oversampling the train set

Train Accuracy: 1.0				
Test Accuracy: 0.9874326750448833				
	precision	recall	f1-score	support
1	1.00	1.00	1.00	1114
2	0.96	1.00	0.98	1114
3	1.00	0.96	0.98	1114

Figure 5.3 Result of model with oversampling the whole dataset

We constructed the SVM Model with the given parameters for each approach. The second model's result which is the model applied to oversample the train set in Figure 5.2 shows the model is overfitting. The difference between the accuracy of train and test set is around 23% which is very large. This proof that the algorithm learned rules specifically for the train set but those rules does not generalize well beyond the train set. For the third model in Figure 5.3, the result turn out that oversampling the whole dataset has a almost perfect train and test accuracy. Besides, the precision, recall and f1-score is almost 100% as well. This model may consider as perfectly trained.

Since the second model is overfitting and third model is almost perfect, we may proceed with our analysis on the first model. We can observe that the recall for class



1 and 3 are only 0.46 and 0.37 which is very low while the recall for class 2 is 0.96. This explains that the model will bias towards class 2. In other words, the model will have a very high chance to predict all new essay as class 2. However, among these model, we still prefer the model without oversampling over the other two models. The result of 10-fold cross validation for this model is shown in Table 5.2 while the average accuracy is 81%.

Nth fold	Accuracy
1	81%
2	83%
3	82%
4	86%
5	87%
6	87%
7	75%
8	69%
9	76%
10	79%

*Table 5.2 Result of cross validation for the model without oversampling*

## ***5.2 Non-machine Learning Model***

The first method we proposed is to determine the best separation points for each feature by using the mean, median and ratio of the class. Table 5.3 below shows the best separation points for each feature in each method.

Features	Method					
	By Mean		By Median		By Ratio	
	Separation Points					
Word count	233.19	405.07	212.75	391.00	83.31	886.53
Sentence count	13.13	25.28	12.00	24.50	8.11	80.17
Average word length	4.08	4.24	4.11	4.27	3.24	4.96
Long word count	18.69	38.13	17.00	38.50	11.76	130.84
Average sentence length	22.58	17.53	17.71	16.2	247.15	24.81
Long sentence count	6.31	11.15	5.50	10.5	29.17	2.62
Unique word count	102.20	167.97	97.00	165.50	37.33	374.85
Noun usage count	44.59	79.21	41.50	81.00	20.3	225.84
Proper Noun usage count	7.96	17.07	5.00	13.50	24.42	271.68
Adjective usage count	15.04	27.24	13.50	26.50	6.07	67.50
Verb usage count	49.42	83.68	44.50	78.50	20.47	217.67
Adverb usage count	13.02	23.37	11.00	21.00	6.52	72.50
Tense ratio	0.80	0.76	0.85	0.81	0.07	0.83
Error count	12.93	16.06	10.00	14.00	127.50	11.46

*Table 5.3 Separation points for each feature for the mean, median and ratio method*

Next, the other methods that we proposed to build the non-machine learning model determine the best separation points after summing up the features with and without weights. The result is shown in Table 5.4.

Method	Separation Points	
Separation point after sum up feature without weights	2.40	7.30
Separation point after sum up feature with weights	9.10	40.90
Separation point after sum up feature with sparse weights	14.10	133.20

Table 5.4 Separation points for the sum up feature with and without weight method

In order to evaluate the model, the reports are shown in figures below.

Accuracy: 0.6823829787234043			
	precision	recall	f1-score
1	0.21	0.41	0.28
2	0.81	0.77	0.79
3	0.50	0.41	0.45

Figure 5.4 Result of the model by mean method

Accuracy: 0.705531914893617			
	precision	recall	f1-score
1	0.28	0.48	0.35
2	0.81	0.81	0.81
3	0.51	0.32	0.39

Figure 5.5 Result of the model by median method

Accuracy: 0.7588085106382979			
	precision	recall	f1-score
1	1.00	0.00	0.01
2	0.76	1.00	0.86
3	0.00	0.00	0.00

Figure 5.6 Result of the model by ratio method

Accuracy: 0.7623829787234042				
	precision	recall	f1-score	
1	1.00	0.00	0.01	
2	0.77	0.99	0.86	
3	0.59	0.07	0.12	

Figure 5.7 Result of the model by summing up features without weight

Accuracy: 0.7617021276595745				
	precision	recall	f1-score	
1	0.07	1.00	0.14	
2	0.00	0.00	0.00	
3	1.00	0.00	0.00	

Figure 5.8 Result of the model by summing up features with weight

Accuracy: 0.7617021276595745				
	precision	recall	f1-score	
1	0.07	1.00	0.14	
2	0.00	0.00	0.00	
3	1.00	0.00	0.00	

Figure 5.9 Result of the model by summing up features with sparse weight

From the result, we can observe that model 3, 4, 5, and 6 have a very bad and biased result where we should not take them into consideration as the non-machine learning model. The third and fourth models which are the models that determine the separation point for each feature by the ratio of class and by summing up all the feature before determining the separation point are biased towards class 2. Meanwhile, the last two models which are the models that determine the separation points by summing up all the features with weights are biased towards class 1. Apart from that, the first and

second model which is the model that determines the separation point for each feature by mean and median have a better result than the other models. The precision and recall for class 1 and 2 from the first model are lower than the second model while the result for class 3 from the first model is slightly better than the second model due to their recall (0.41 for the first model and 0.32 for the second model). In term of choosing the model that would not bias towards certain class, the first model which is the model that determines the separation point for each feature by mean of the class will be chosen as the non-machine learning model.

#### ***5.2.1 Model Combination 1: Weighted Mean Approach***

In this weighted mean approach, we applied 9 sets of weights to determine the best model. It turns out that when the weight for machine learning is higher, the weighted mean approach result is exactly the same as machine learning model 1. The same situation also applied to non-machine learning weight. This is due to the class that we try to classify are discrete value. During the process of finding the weighted mean score, all decimal values have been rounded into a discrete number. In this case, we can conclude that there is no reason to assign imbalanced weight to the models. Figure 5.10 shows the result of the balance weight (1,1). The overall accuracy is dropped compared to the machine learning model 1. This is because the result of non-machine learning is also lower. Besides, the precision for class 1 and 3 are 0.83 and 0.77 which are higher than machine learning and non-machine learning model. Moreover, the recall for class 1 and 3 are 0.14 and 0.28 which is lower than machine learning and non-machine learning model.

Test Accuracy: 0.7991831177671885			
	precision	recall	f1-score
1	0.83	0.14	0.23
2	0.80	0.98	0.88
3	0.77	0.28	0.41

Figure 5.10 Result of the machine learning model 2

In conclusion, this is not a good method to combine the machine learning and non-machine learning model since the result especially the recall for class 1 and 3 is too low.

### 5.2.2 Model Combination 2: Machine Learning Model 2

In this model, we include the result from non-machine learning model as a feature to build another machine learning model. Table 5 shows the result of parameter tuning while Figure 5.11 shows the result of the machine learning model 2 after parameter tuning.

SVM Model	Kernel	C	Gamma, $\gamma$	Average Accuracy
Without oversampling	RBF	1	0.1	81%

Table 5.5 Result of parameter tuning for machine learning model 2

Train Accuracy: 0.8361325465274626			
Test Accuracy: 0.8250510551395507			
	precision	recall	f1-score
1	0.77	0.49	0.60
2	0.84	0.96	0.89
3	0.73	0.38	0.50

Figure 5.11 Result of machine learning model 2

We use the step same as Machine Learning Model 1's first model because it still has room for improvement. The best parameter for the model without oversampling is  $C = 1$  and  $\gamma = 0.1$  with RBF kernel. This result is exactly the same as the machine learning model 1.

On top of that, the result of the machine learning model 2 shows that it has a slightly better result compare to the machine learning model 1. Firstly, the overall accuracy and recall for class 1 and 3 of the machine learning model 2 is higher than the machine learning model 1. In the other hand, the precision of class 3 has dropped 0.01 compare to machine learning model 1 which is not an important point. Apart from that, the other value in the result is actually similar to machine learning model 1. In summary, machine learning model 2 is slightly better than machine learning model 1. Lastly, the result of cross-validation for this model is shown in Table 5.7 while the average accuracy is 80% where the model does not overfit.

Nth fold	Accuracy
1	81%
2	84%

3	81%
4	86%
5	88%
6	87%
7	75%
8	69%
9	76%
10	78%

*Table 5.6 Result of cross-validation for machine learning model 2*

### ***5.3 Implementation Result Summary***

To conclude the result above, the best essay evaluation model is Machine Learning Model 1's third model with the method of oversampling the whole dataset before splitting into train and test set. Moreover, Model Combination 2: Machine Learning Model 2 did improve the result of Machine Learning Model 1's first model. On top of that, the best non-machine learning model from the comparison is the first model which is to determine the separation points for each feature by the mean of class. The overall accuracy achieved is 68% which is the lowest among other method but it is the model with most balance value in precision and recall for each class. The result may not be as good as the machine learning model 1 and 2 but it does improve the result of the machine learning model 1. Lastly, the weighted mean approach is not an appropriate method to combine the model due to the effect of rounding the weighted mean score into discrete value actually cause the result similar with the machine learning model 1 or non-machine learning model.



Last but not least, the AAS System will build based on the best model from each approach. A total of 4 essay evaluation model will be implemented in the AAS System. In the AAS System, we are not able to determine the mean of each class from a newly input essay. In this case, the user is allowed to customize the separation points for each feature based on the user's understanding or experience on the input essay. Last but not least, the weighted mean approach will be included as well but only with equal weight. It depends on the user's decision to follow which model's result.

## CHAPTER 6 CONCLUSION

This project aims to build an AAS System to overcome some issue that is occurring in the industry. Hence, the objective is set to build an AAS System that provides multiple essay evaluation models. We have achieved the first objective which is to build a classification model with a large essay data to classify essay. A SVM multiclass classification model (Machine Learning Model 1) is built parameter tuning and comparison between oversampling and non-oversampling technique. Next, we have determined a total of 14 features that are used to build the machine learning and non-machine learning model from literature reviews. Besides, an extra feature which is the result from non-machine learning model is discovered and proposed to consider as the 15th feature to build another machine learning model (Model Combination 2: Machine Learning Model 2). On top of that, we have achieved the third objective which is to combine the machine learning and non-machine learning with the proposed method: Model Combination 1: Weighted Mean Approach and Model Combination 2: Machine Learning Model 2. The result shows the Model Combination 2: Machine Learning 2 has improved compare to Machine Learning Model 1. The last objective is also achieved by evaluating and comparing the result from each model. Last but not least, the best model is selected from machine learning, non-machine learning, and model combination to build the AAS System.

### ***6.1 Future Works***

The non-machine learning model's feature is based on the current literature review. It can be improved by investigating research papers regarding essay scoring as well as

consulting qualified experts in the industry. On top of that, the ensemble learning classification approach may be applied to tackle the class unbalanced problem. Ensemble learning classification approach is well known to tackle this issue (Krawczyk, 2016). Moreover, we can perform feature selection to improve the machine learning model. Last but not least, the results from this research-based project is shown but yet to be able to explain theoretically such as the reason for the kernel's performance and etc. This project is able to proceed with deeper research in future postgraduate study.

## REFERENCES

Apostolidis-Afentoulis, V. (2015). SVM Classification with Linear and RBF kernels.

Burstein, J., Andreyev, S., & Lu, C. (2006, aug # "8"). Automated essay scoring.

Google Patents.

*Cambridgespark*. (2016, 11 25). From Support Vector Machines:

<http://beta.cambridgespark.com/courses/jpm/05-module.html>

Chen, J., Fife, J. H., Bejar, I. I., & Rupp, A. A. (2016). Building e-rater Scoring

Models Using Machine Learning Methods. *ETS Research Report Series*, 1--12.

Chung, G. K., & O'Neil Jr, H. F. (1997). Methodological Approaches to Online

Scoring of Essays.

Gary M and McCarthy, Kate and Zabar, Bibi. (2007). Cost-sensitive learning vs.

sampling: Which is best for handling unbalanced classes with unequal error costs? *DMIN*, 35--41.

- Hofmann, M. (2006). Support vector machines—Kernels and the kernel trick. *Notes*.
- Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 415--425.
- Kaggle. (n.d.). From The Hewlett Foundation: Automated Essay Scoring: <https://www.kaggle.com/c/asap-aes/data>
- Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 221--232.
- Liu, A. C. (2004). The effect of oversampling and undersampling on classifying imbalanced text datasets. *The University of Texas at Austin*.
- Madala, D. S., Gangal, A., Krishna, S., Goyal, A., & Sureka, A. (2018). *An empirical analysis of machine learning models for automated essay grading*.
- Mathieu, E. (2018, 8 8). *EMILE MATHIEU*. From An Efficient Soft-Margin Kernel SVM Implementation In Python: [http://emilemathieu.fr/blog\\_svm.html](http://emilemathieu.fr/blog_svm.html)
- Mock Flow. (n.d.). From Mock Flow: <https://mockflow.com/app>
- Pahwa, S., & Sinwar, D. (2015). Comparison Of Various Kernels Of Support Vector. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 532-536.
- Ramalingam, V., Pandian, A., Chetry, P., & Nigam, H. (2018). Automated Essay Grading using Machine Learning Algorithm. *Journal of Physics: Conference Series*, (p. 012030).
- Semire, D. (2006). An overview of automated scoring of essays. *The Journal of Technology, Learning and Assessment*.
- Semire, D. (2006). Automated Essay Scoring. *The Turkish Online Journal of Distance Education*.

Shenoy, V. N. (n.d.). A Machine Learning Model for Essay Grading via Random Forest Ensembles and Lexical Feature Extraction through Natural Language Processing.

Valenti, S., & Cucchiarelli, F. N. (2003). An Overview of Current Research on Automated Essay Grading. *Journal of Information Technology Education: Research*, 319--330.

*Word Tokenization with Python NLTK*. (n.d.). From Word Tokenization with Python NLTK: <https://text-processing.com/demo/tokenize/>

## **APPENDICES**

### ***Appendix A: Draft Research Paper***

# Automated Article Scoring

You Qian Lim  
Multimedia University  
line 3: name of organization  
Persiaran Multimedia, 63100  
Cyberjaya, Selangor  
Email: limyq0202@gmail.com

**Abstract**—Most of the organization still rely on manual evaluation by humans, it actually brings up some inconsistency in article scoring such as personal biased towards the writer or boredom when evaluating tons of articles. There are some commercial products such as E-rater and Intelligent Essay Assessor™ which produce a result that is acceptable by human but there is always a space for enhancement. We proposed to build a Article Scoring System (AAS) with implementation of machine learning, non-machine learning model and combine it to provide a more reliable result. We have determined 14 features to build the machine learning and non-machine learning model. Support Vector Machine algorithm is used to build the machine learning model while total of 6 method has been proposed to build the non-machine learning model. Next, Weighted Mean Approach and including result from non-machine learning model is applied to perform the combination of model to achieve better result. An overall accuracy of 83% is achieved by the second model combination approach which is the highest. The best model from machine learning, non-machine learning model and weighted mean approach each will be implemented in the AAS System.

## I. INTRODUCTION

Article and essay writing is a good way to express personal thoughts and improve writing skills. But human evaluation may be inaccurate sometimes due to boredom by evaluating a large amount of essay with the same title or being biased to some of the writers. In this case, Automated Essay Scoring (AES) can evaluate essays without being affected by boredom or personal bias to writers. Research show that the AES has a high accuracy and reliability which reach a high agreement level between AES systems and human raters [1]. However, AES systems are not perfect. AES system performance may not be ideal when the number of criteria involves in the evaluation of essay increases beyond a certain level [2].

AES systems that use qualitative analysis to evaluate essays are more frequent than the system that uses quantitative analysis. Quantitative analysis uses statistical method while qualitative analysis uses a non-quantifiable method such as machine learning to compute a score for an essay. This research is aim to build an Automated Article Scoring (AAS) System that contain multiple type of article evaluation models. This system will generate shallow linguistic features from the input essay such as the word count and average length of sentences before evaluating the input essay.

The first evaluation model implemented in the AAS System is machine learning technique where it uses a multiclass classification approach to score the article. A large number of essay with labels will be collected for the training

purpose of the machine learning model. Next, the second article evaluation model implemented in the AAS System is a non-machine learning. The non-machine learning method will examine the generated features quantitatively based on proposed scoring scheme in this project or personal preference. After that, the AAS System will perform a combination of model approach. The first combination of model approach is the AAS System retrieves the score from each model and perform a weighted mean scoring approach. Then, the second approach is to include the score computed from non-machine learning model as a feature to build another machine learning model.

There will be a total of 4 article evaluation model in the AAS System. The main purpose of this research project is to determine a method to provide a better and more reliable score of the essay based on the machine learning and non-machine learning method.

## II. LITERATURE REVIEW

### A. Related Work

Several organizations or company have started to put in effort in this area. Some of the systems are Project Essay Grader (PEG), Intelligent Essay Assessor™ (IEA), E-Rater®, IntelliMetric™, and Bayesian Essay Test Scoring System™ (BETSY). The main reason that people started to develop in the AES system is that AES system has one of the major reason is that it is able to score and provide feedback instantly which significantly improves the efficiency of writing the assessment.

### B. Features

The dataset that I am using for this project will be students written essays with score rated by humans. From each essay, we can extract many features that “define” the essay itself in order to perform machine learning. The project’s scope stated that this project will be focusing on surface linguistic features instead of the content of the essays.

The first and most obvious feature that can be extracted from the essay is word count and sentence count. For instances, essay’s length in terms of words and sentences, unique word count, average sentence length, average word length, and each punctuation count are the most common statistical value that we can calculate in an essay. Besides, some part-of-speech (POS) tag count might also be a very good feature to define the essay. PEG is one of the products that based on this analysis. They included a total of 28 proxy variables such as the number of paragraphs and subject-verb openings in the initial PEG system. PEG applied this analysis

with some additional software like grammar checker, part-of-speech tagger, electronic dictionary, and parser [3]. Apart from that, the number of spelling mistakes is a possible important feature that should be included when grading an essay [4].

### C. Classification Method and Algorithm

In an experiment, they have conducted an empirical analysis of machine learning models for automated essay grading [5]. Data normalization and scaling is very important before applying classification algorithms. During their data pre-processing stage, they implemented a logarithmic transformation to perform data normalization on the chosen 9 features in the range of 0 to 1.

The essay dataset will be classified into 4 classes which are A, B, C and D. A total of 3 classification algorithm were used in the experiments which are k-Nearest Neighbour (kNN), Logistics Regression (LR) and SVM. Figure 2 Even though KNN have the most number of highest accuracy in the experiment which is Set 1 and Set 8, but in average, kNN has 82.94% accuracy, SVM has 83.09% (highest) and LR has 83.01% which only have a very small gap of not more than 1%.

In the other experiment, SVM outperformed Multiple Linear Regression, Random Forest and kNN in most of the assessments in both default and after tuning the hyperparameter with the range of 41% to 63% of exact agreement between human rater and automated scoring machine [6]. Besides, SVM also has the highest score for quadratic-weighted kappa (QWK) in the range of 0.554 to 0.768 and 0.627 to 0.823 for Pearson correlation ( $r$ ).

### D. Support Vector Machine's Kernels

There are many kernels in SVM which are linear, radial basis function (rbf), polynomial and sigmoid kernel where different kernel may work well on different cases. For instance, linear kernel works well when the instance is linearly separable. When the instance linearly inseparable, the "kernel trick" can be applied to transform all the data points into a new feature space to separate the data points.

## III. METHODOLOGY

AAS System is proposed and built to achieve the main purpose and objective of this project. Before we design the whole AAS System, we need to determine and build the four main essay evaluation model.

### A. Dataset

The dataset used to build all the models is acquired from kaggle.com. A total of 5875 persuasive/narrative/expository essay will be selected from the raw dataset. Each essay set have different range of scores. We will rescale all the scores into 1, 2 and 3 in order to build the 3 classes SVM classifier. The range of average essay length is 250 - 650 words per essay. The dataset will be splitted into train:test with ratio of 75:25.

### B. Data Pre-processing

The works that need to be done in data pre-processing stage are: -

- Rescale the label
- Replace Anonymization
- Tokenization
- Remove Punctuation

### C. Features

The features that is going to be extracted are Word count, Long word count, Average word length, Unique word count, Sentence count, Long sentence count, Average sentence length, Noun usage count, Proper Noun usage count, Adjective usage count, Verb usage count, Adverb usage count, Tense ratio, Error count.

### D. Machine Learning Model 1

Before building the machine learning model, the essay dataset splitted into train and test set. Then standardization is applied to the train and test set as the feature scaling method because we want to make all features fit into the same scale, while also maintaining their internal variance and not affecting their relative importance. After that, the train data is used to fit the SVM model while the test set is used to evaluate the fitted SVM model. Next, parameter tuning is performed in order to allow the SVM model's behavior to be adapted to the given data. The best performing set of parameter will be selected and used to build the machine learning model. The step to build the machine learning model are: -

- Split the dataset into train and test set
- Fit and scale the train set
- Scale the test set based on train set
- Train the SVM using train set
- After the SVM Model is built, use the SVM Model to predict the test set
- Evaluate the result

### E. Non-machine Learning Model

This model evaluates the essay without any machine learning techniques. The method proposed for this model are: -

- Determine the separation point for each feature that separate the class of the essay. The steps of mean and median methods are: -
  - Split the data according to the class.
  - Determine the mean/median of each class.
  - Sum up the mean/median of class 1 & 2 and divide by 2 to get the midpoint.
  - Repeat step 3 for class 2 and 3.
- The steps of ratio method are: -
  - Split the data according to the class.
  - Divide the number of instance for each class over the total number of instance to acquire the ratio for each class .



- Get the range value by minus the minimum value from the maximum value in the feature.
- Next, times the value from step 3 with the ratio for class 1.
- Then, add up the value from step 4 and the minimum value in the feature to acquire the first separation point.
- Repeat step 3 to 5 to calculate the second separation point but with the sum of the ratio of class 1 and 2.
- Next is sum up all the features and determine the best separation point among all possible set of separation points. The steps are: -
  - Standardize the feature using StandardScaler() from Scikit-learn.
  - Normalize the the result in step 1 with Min-max normalization to reduce the size of range.
  - Sum up all the features, let's call it  $X\_sum$ .
  - Define all possible set of separation point within the range in of  $X\_sum$ .
  - Apply each possible set of separation point to the data and compute the result.
  - The set of separation with highest accuracy will be selected.
- Lastly, sum up all the features and assign weights to determine the best separation point among all possible set of separation points. The steps are: -
  - Standardize the feature using StandardScaler() from Scikit-learn
  - Normalize the the result in step 1 with Min-max normalization to reduce the size of range
  - Assign weights for each feature
  - Sum up all the features, let's call it  $X\_sum$
  - Define all possible set of separation point within the range in of  $X\_sum$
  - Apply each possible set of separation point to the data and compute the result
  - The set of separation with highest accuracy will be selected

#### F. Models Combination 1: Weighted Mean Approach

The first method to combine the machine learning model and non-machine learning models is weighted mean score

approach. After we have computed the result from machine learning and non-machine learning model. We can apply the weighted mean formula to compute a final combined result. To determine the best weightage empirically, we will execute the formula with a set of weightage. The weightage with highest accuracy score will be selected for this model. The set of weight that is going to be applied are (2, 1), (1.8, 1), (1.5, 1), (1.3, 1), (1, 1), (1, 1.3), (1, 1.5), (1, 1.8), (1, 2).

#### G. Models Combination 2: Machine Learning Model 2

The second method to combine the machine learning model and non-machine learning models is to include the result computed by the non-machine learning model into the machine learning model as a feature. The steps are similar to Machine Learning Model 1

### IV. EXPERIMENTS AND DISCUSSION

#### A. Machine Learning Model

Table X below shows the result of parameter tuning while Figure X, X and X shows the results of model without oversampling, oversample train set and oversample whole dataset. Based on the result, the second model is overfitted because the difference between train and test accuracy is too large. Next, the third model shows an almost perfect result but we have to remove this model because it constructed conceptually wrong. We cannot oversample the test set. Lastly, even though the first model has low recall for class 1 and 3, we will still choose this model because it has the best result and constructed appropriately.

SVM Model	Kernel	C	Gamma, $\gamma$	Average Accuracy
Without oversampling	RBF	1	0.1	81%
Oversample train set only	RBF	10	1	77%
Oversample whole dataset	RBF	10	10	96%

Train Accuracy: 0.8293236495687698 Test Accuracy: 0.8230088495575221				
	precision	recall	f1-score	support
1	0.77	0.46	0.58	110
2	0.83	0.96	0.89	1114
3	0.74	0.37	0.49	245

Train Accuracy: 0.9989028525832835 Test Accuracy: 0.763784887678693				
	precision	recall	f1-score	support
1	0.55	0.45	0.50	110
2	0.82	0.89	0.85	1114
3	0.50	0.35	0.41	245

Train Accuracy: 1.0 Test Accuracy: 0.9874326750448833				
	precision	recall	f1-score	support
1	1.00	1.00	1.00	1114
2	0.96	1.00	0.98	1114
3	1.00	0.96	0.98	1114

### B. Non-machine Learning Model

The model that gives the best result is to determine the separation points for each feature by using the mean of class. Figure X below shows the result of the model. The reason of this model is being selected is because it has the most balance result for each class while the median method model has higher overall accuracy but unbalanced result. Moreover, the other models are all towards 1 class which very bad and can be ignored.

Accuracy: 0.6823829787234043				
	precision	recall	f1-score	
1	0.21	0.41	0.28	
2	0.81	0.77	0.79	
3	0.50	0.41	0.45	

### C. Model Combination 1: Weighted Mean Approach

We found out that all the imbalanced weight will bias towards one result. For instance, when the result from machine learning model 1 is higher, then the final result will be same as machine learning model 1 result. This is due to our class is discrete value the weighted mean score is affected by rounding. So, we may only consider the result for equal weights. Figure X below shows the result. This model is not recommended due to the result is skewed to class 2 badly.

Test Accuracy: 0.7991831177671885				
	precision	recall	f1-score	
1	0.83	0.14	0.23	
2	0.80	0.98	0.88	
3	0.77	0.28	0.41	

### D. Model Combination 2: Machine Learning Model 2

The result of this model is slightly better than machine learning model 1. The overall accuracy and recall for class 1 and 3 has increased compare to machine learning model 1. This is important because we do not want the model that skewed towards 1 class. Figure X below shows the result of the model.

Test Accuracy: 0.8250510551395507				
	precision	recall	f1-score	
1	0.77	0.49	0.60	
2	0.84	0.96	0.89	
3	0.73	0.38	0.50	

### CONCLUSION

The machine learning model has very great recall and precision for class 2 but it also has very low value of recall for class 1 and 3. This skewed model will cause the prediction in future will be inaccurate for class 1 and 3 essays. Next, the non-machine learning model may not be high-achieving but it does improve the machine learning model 1 in mode combination 2. In the AAS system, user is allowed to use the suggested separation points or customize based on their personal experience and knowledge about the input essay. Moreover, the model combination 1: weighted mean approach has shown a very bad result which is unusable. This model will still be implemented in the system. It is depending on user which model that want to follow. Lastly, the last model which is the model combination 2 shows the best result where it improves the recall for class 1 and 3.

### REFERENCES

- [1] D. Semire, "An overview of automated scoring of essays," *The Journal of Technology, Learning and Assessment*, 2006.
- [2] J. Burstein, S. Andreyev and C. Lu, *Automated essay scoring*, Google Patents, 2006.
- [3] G. K. Chung and H. F. O'Neil Jr, "Methodological Approaches to Online Scoring of Essays," 1997.
- [4] V. Ramalingam, A. Pandian, P. Chetry and H. Nigam, "Automated Essay Grading using Machine Learning Algorithm," in *Journal of Physics: Conference Series*, 2018.
- [5] D. S. V. Madala, A. Gangal, S. Krishna, A. Goyal and A. Sureka, "An empirical analysis of machine learning models for automated essay grading," 2018.
- [6] J. Chen, J. H. Fife, I. I. Bejar and A. A. Rupp, "Building e-rater Scoring Models Using Machine Learning Methods," *ETS Research Report Series*, pp. 1-12, 2016.

## Appendix B: Turnitin Report

Fyp2

ORIGINALITY REPORT

9%

SIMILARITY INDEX

6%

INTERNET SOURCES

6%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Bahcesehir University

Student Paper

1%

2

espace.curtin.edu.au

Internet Source

<1%

3

escholarship.bc.edu

Internet Source

<1%

4

www.kaggle.com

Internet Source

<1%

5

documents.mx

Internet Source

<1%

6

Ayako Masuda, Tohru Matsuodani, Kazuhiko Tsuda. "A Comparative Study Using Discriminant Analysis on a Questionnaire Survey Regarding Project Managers' Cognition and Team Characteristics", 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), 2017

Publication

<1%

7

scitepress.org

