MONASH University

VIRTUAL INTERNSHIP

Unit Title:
ADS2001 - Data Challenges 3

Chief Examiner:
Dr. Simon Clarke

Completed by:

Gue Zhen Xue	33521352
Andres Xue	34987274
Zohaib Javed	34290826
Denisha Fam Wen Hsiu	34091637

Date of Submission:
5th June 2025

Table of Contents

Table of Contents ii

Executive Summary iii

Chapter 1 Introduction 1

1.1 Background 1

1.2 Available Data 2

1.3 Problem Statement 3

1.4 Objectives 3

Chapter 2 Methodology 4

2.1 Nature of the Data, Data Quality and Potential Issues 4

2.2 Data Pre-processing 4

2.3 Exploratory Data Analysis 5

2.4 Data Processing 5

2.5 Baseline Modelling 7

2.6 Model Assessment 8

2.7 Model Development 11

Chapter 3 Result and Discussion 13

3.1 Assessing Effect of Model Development 13

3.2 Discoursing Relation between Features and Target Variables 15

Chapter 4 Conclusion and Future Work 17

4.1 Summary 17

4.2 Limitations and Future Work 17

References 19

Appendix 1: Text Normalization Challenges 21

Appendix 2: Calculation for TF-IDF & Sentiment Analysis 22

Appendix 3: Text Pre-processing Workflow 23

Appendix 4: Aggregation Approaches for Features and Target in Dataset 24

Executive Summary

Introduction

Kidney failure, or end-stage renal disease (ESRD) is a rising global epidemic that is irreversible and significantly reduces the quality of an individual's life. Despite the advancements in technology and medications, the existing treatments as of today such as dialysis and kidney transplant prove to be only methods to prolong the life of an individual, rather than cure the disease. These treatments also have significant limitations on an individual as the treatments require time and may have negative side effects.

Thus, *Nephrotex*, an educational simulation of an internship was created. This internship will mimic a biomedical engineering workspace where students act as interns to create and design a device that will assist patients with ESRD. The students will collaborate in teams via an online platform and may receive assistance on their creative decisions from mentors. Our project on this virtual internship aimed to create a predictive model that can successfully, and as accurately as possible, predict the final report score of the team based on their communication behaviours in the *csv* file containing the students' chatlogs and other important metadata.

Objectives

Therefore, 3 objectives were drawn to be accomplished in the project report, which are to aggregate and transform the chat data into team-level statistics, to build predictive models to predict final report scores based on team communication behaviours and to interpret the results of the models to understand how communication features relate to the team report performance.

Methodology

The *csv* file provided contains a dataset of approximately 19,000 rows of data and 16 columns of metadata such as the roles of the individual's chat content, their user ID, and the outcome score. As the data is unstructured, we needed to process the data, which involves content-based feature engineering, team frequency-inverse term frequency (TF-IDF), sentiment analysis, structure-oriented and progress-oriented feature engineering, as well as data aggregation.

Most of the raw data was inconsistent in terms of their format. Methodology of data preprocessing, feature engineering, and modelling was being used to preprocess our data before we were able to successfully create a model that can predict the team performance outcome.

Starting off with content-based feature engineering, we were able to normalize the many variations of text styles. We did lemmatization as well, which reduced the noise in the dataset, alongside team frequency-inverse team frequency (TF-IDF) to find the most relevant terms in the discussions. With that, we gained around 4400 new columns of data, thus we proceeded to find the sentiment analysis of the chats so we could quantify the emotional tones of the chats.

To obtain the player to mentor ratio, we had to do structure-oriented and progress-oriented feature engineering, which allowed us to distinguish the relevance of the mentors in the group discussions. Additionally, data aggregation of individual-level data was done to convert it into team-level statistics. Out of all the aggregation methods we did, Batch-Gradient Descent proved to be the best, thus we relied on that aggregation method to carry out further analysis on the 'Outcome Score' column.

Modelling Approach

As for our baseline model, we determined that using K-Nearest Neighbours was the best for this project due to its simplicity, fast training nature, high interpretability and its classification-based nature. However, we analysed that there was too large of a difference between the training accuracy score (0.581), testing accuracy score (0.269) and F1-score (0.241), which led us to conclude that the model was unable to predict unseen data. With that, we went back and further analysed our data and realized that the baseline model was performing poorly due to imbalanced nature of the target variables, high amounts of available features, difference in scale throughout the features, and lack of training samples for minority classes. Considering this, we decided to prioritise F1-score and testing accuracy score when evaluating model performance.

We then decided to resample our data by SMOTE oversampling, which provided better accuracy for our models. Besides that, we did normalization, multicollinearity removal by setting a threshold of $R^2 = 0.99$, setting variance threshold of $\sigma^2 = 0.01$, and statistical selection using ANOVA f-test and chi-squares test with p-value of 0.05. We redeveloped our models, and we decided to focus on Support Vector Machine (SVM), Decision Tree, Random Forest, Gradient Boosting, LightGBM, and Logistic Models.

Result

With the aforementioned methodology, we were able to build a couple of models to carry out our objective of creating a model that can successfully predict the final report score of the teams. The poorest model appeared to be our k-NN baseline model with testing accuracy score of 0.269 and F1-score of 0.241. Besides, out best model appeared to be tuned Random Forest Classifier which achieved 0.759 in accuracy score and 0.758 in F1-score. The balance between accuracy and F1-score in our best model showed that the strategies in tackling modelling issues are well-performed.

Evaluation of feature importances were conducted on the best performing model, which is the fine-tuned Random Forest Classifier. All features are showed with low importance scores, that suggested that no single feature had strongly influenced the outcome score. Mentor-related terms were found to be absent, which indicated that mentor-related factors had minimal impact on outcomes. Thus, relationship between features and outcome score are unable to be simply described, that indicated non-linear and complex relationship appearing between features and outcome scores.

Conclusions, Limitations and Future Work

Overall, our project demonstrated how communication and teamwork can contribute to a higher performance report. Contrary to our initial hypothesis, mentor-related features were found to be having minimal impact. All project objectives were met, including data transformation into team-level features, model development, and interpretation of communication-based predictors. Our findings showed balanced feature contributions within team dynamics.

Limitations were appearing in text normalization strategy mainly due to the limitations of NLTK pos tagging ability. Besides, biasness was still being observed in the models indicating limitation of current resampling strategy. Looking forward, future works should include advanced NLP tools like spaCy, and reconsideration of resampling techniques. Besides, future modelling should explore advanced algorithms like XGBoost and neural networks to integrate diverse feature representations.

Chapter 1 Introduction

1.1 Background

Kidney failure

Kidney failure, also known as end-stage renal disease (ESRD), is a medical condition where the kidneys of an individual are no longer capable of filtering waste and excess fluid from the blood without external assistance. Kidney failure may be caused by a multitude of reasons; the main two being diabetes and high blood pressure. Social and environmental factors such as income, stress, and diet, alongside other existing health issues may also contribute to the development of kidney failure.

Kidney failure, at this current time, is irreversible. Thus, patients with ERSD can only resort to treatments such as kidney dialysis and/or kidney transplant to increase the possibility of a longer life. Kidney dialysis can be categorized into two; haemodialysis and peritoneal dialysis. Both have the same goal of filtering waste and excess fluid from the blood using an external machine, but haemodialysis uses solely an external machine, whereas peritoneal dialysis uses the lining of the individual's abdomen as a filter alongside an external machine.

Dialysis

Dialysis requires the patient to receive the treatment at least 3 times a week, each session taking between 3 to 8 hours each, depending on the severity of the patients' kidney failure. Some patients may require a higher number of sessions, and may take longer time per session.

Haemodialysis removes toxic waste and excess fluids in a patient with the use of a machine. This treatment is usually done at a dialysis clinic, 3 times a week, each taking anywhere between 3 to 4 hours to complete. The treatment can be done at home, but will require more frequent sessions at 5 to 6 days a week. The downside to this treatment, although effective, will disrupt the patients' ability to do other activities or go about their daily routine as they are unable to detach themselves from the machine. Thus, peritoneal dialysis is an alternative dialysis treatment that patients can consider (Chesler et al., 2015).

Peritoneal dialysis can be divided into two types; continuous ambulatory peritoneal dialysis (CAPD) and continuous cycling peritoneal dialysis (CCPD). During CAPD, the individual's abdomen will be filled with dialysate- a fluid that provides the individual with a "container" into which toxic waste and excess water will pass for removal from their body. The dialysate will then be left to remain for the dwell time, then the liquid containing toxic waste and excess fluid will be drained when they wake. Patients may need to exchange the dialysate 3 to 5 times a day, the longest dwell time being during the nighttime as they sleep. With this process, the patients can go about their daily routine and activities as opposed to spending hours at a dialysis clinic. As for CCPD, the patients are required to stay attached to an automatic cycler machine. The machine will do the removal of toxic waste and excess water by filling the patients' abdomen with dialysate and letting it dwell for a certain period of time. The patients are then to remove the sterile bag containing the waste in the morning when they wake. The period of time and when the patient undergo CCPD depends on their doctor's advice; most do it during the nighttime as they sleep while others may need to do it again during the morning which may last the whole day (Chesler et al., 2015).

Kidney transplant

Though kidney transplant allows the patient to have a higher rate of survival, it is not perfect. The patient's body may reject the new kidney, causing them to require lifelong immunosuppressant medications. This medication prevents the organ rejection and treats autoimmune diseases where the patient's system attacks

the body's own tissues. Though this medication is crucial for the patient's survival, it does not come without any possible negative side effects. To name a few, patients may experience an increase in their susceptibility to infections and nausea which can disrupt their daily routine (Chesler et al., 2015).

New biomedical treatments

With the rising cases of ESRD, finding new biomedical treatments for patients are crucial as to ensure that treatments are to be as accessible as possible. Thus, a virtual internship '*Nephrotex'* was set up. It is an educational simulation of a biomedical engineering workspace where students work in teams to design devices that will assist patients with kidney failure (Chesler et al., 2015).

Nephrotex

This internship encourages students to think, act, and justify like real professionals, as they have to do background research on stakeholder needs, which may vary in terms of performance metrics and requests. With their designed device, they need to be able to justify their design decisions whilst ensuring their ethical reasoning remains a top priority (Chesler et al., 2015).

The students need to find a perfect balance between engineering and the biomedical aspects of the project, ensuring that their device is up to standards for stakeholders such as patients, doctors, hospitals and manufacturers. The students might face several challenges in their design process; whether it be the priority of their device, the performance of their device, or even the feasibility of their device. Thus, this internship will push the students to improve their communication skills, their ethical reasoning, and their design and technical decision making. At the end of the internship, the students should have tested and evaluated their prototypes, and be able to justify their creative and technical decisions (Chesler et al., 2015).

Our report

This report will investigate the chatlogs between the students, alongside other behaviours in their converzations. This will allow us to create a model that will predict the outcome score of the group, which is our aim with this project. We plan to analyse the relationship between the communication features and how it relates to the outcome score of the group.

1.2 Available Data

The dataset provided for this internship project was from *Nephrotex* Virtual Internship program, where students collaborated to conduct series of simulation works for experimental learning (Chesler et al., 2015). Data was collected throughout meetings from students in which the students' chatlogs were being recorded. The *virtualInternshipData.csv* file contained all the data collected throughout the program. It consisted of approximately 19,000 rows (*n*) and 16 columns (*m*), each representing different features of the data.

Table 1.2.1: Description of Columns in Available Data

Column Name	Explanation
`user_id`	Unique ID for each student
`implementation_id`	Unique ID for each implementation session
`line_id`	Unique ID for each individual chat message
`team_id`	Team identifier; text for first half, number for second half of internship
`message`	The actual chat message content

group_id	Numeric group identifier (may repeat across students)
role	Role of the speaker: either `mentor` or `player` (student)
activity_name	Name of the internship activity (chat room)
mentions_testing	Indicates if message regards to using testing/experiments for design
mentions_design_choices	Indicates if message discusses making a design choice
mentions_questions	Indicates if the message includes a question
mentions_consultant_requests	Talks about meeting consultant (stakeholder) requests in design
mentions_performance_criteria	Justifies design using performance metrics or experiments
mentions_communication	Justifies design for better communication among engineers
OutcomeScore	Score from 0 to 8 for the quality of the student's final design report
word_count	Total number of words in the chat message

Observing Table 1.2.1, available columns included both numeric and non-numeric variables, ranging from string identifiers to textual content and encoded values. Briefly inspecting the table, the available data provided mainly data related to chatlogs (content) related data, indicating the importance of the 'content' column and potential of deriving information from 'content' column as qualitative data. The target column was determined to be 'OutcomeScore' which is the performance score from 0 to 8 for the quality of student's report.

1.3 Problem Statement

Hypothesis

We hypothesized that teams that asked more questions and get feedback are likely to have a higher performance report, especially with the help of the mentors. The team engagement should also prove to be a strong aspect of the team getting a high-performance report.

Problem Statement

Understanding the nature of the dataset in which based on the Virtual Internship program, which is a key factor in contributing to the performance of the model. The project aimed to explore the causal relationship between available or potential factors in the dataset and the outcome score of the project. In other words, the report mainly targeted to aim for exploring how factors (e.g. the converzations each group had, assists from mentors etc.) in a group influence the outcome, which is the performance of the project in the Virtual Internship program.

1.4 Objectives

Understanding the importance of exploring the causal influence between factors of the Virtual Internship program and their outcomes, we drew 3 objectives to be accomplished in the project report, which are as follows:

- Aggregate and transform the chat data into team-level statistics.
- Build predictive models to predict final report scores based on team communication behaviours.
- Interpret the results of the models to understand how communication features relate to the team report performance.

Chapter 2 Methodology

2.1 Nature of the Data, Data Quality and Potential Issues

Nature of the Data

Table 2.1.1: Datatype of Columns in the Available Data

Column Name	Datatype	Column Name	Datatype
userIDs	int64	**m_experimental_testing**	int64
Implementation	object	**m_making_design_choices**	int64
Line_ID	int64	**m_asking_questions**	int64
ChatGroup	object	**j_customer_consultants_requests**	int64
Content	object	**j_performance_parameters_requirements**	int64
group_id	int64	**j_communication**	int64
RoleName	object	**OutcomeScore**	int64
roomName	object	**wordCount**	int64

The dataset provided for this internship project consisted of approximately 19,000 rows (*n*) and 16 columns (*m*), each representing different features of the data. Based on Table 2.1.1, the dataset contained a combination of numeric and non-numeric features, which are mainly ID-related columns, content related columns, and target related column, which is OutcomeScore.

Data Quality and Potential Issues

In terms of quality, the dataset was generally usable, except for the content column, which required additional effort to maintain its context and semantic integrity. We faced numerous challenges during the initial exploration phase. Firstly, we found that establishing a link between the column definitions and the accompanying project explanation (PDF) was difficult, as several column names were unclear or ambiguous. Besides, certain binary columns lacked clear definitions, making it hard to interpret the meaning behind 0 and 1. Additionally, the "content" column was highly unstructured, including a mix of formal and informal English, Greek words, slang, numbers in odd formats, and inconsistent syntax. Furthermore, the same group IDs appeared multiple times, raising concerns about potential duplication or corruption. Moreover, repetition of user IDs led us to believe that rows might have been swapped or reused by students, which added to the confusion during analysis.

Basic issues including missing values and existence of unnecessary IDs could be resolved by conducting data pre-processing. Further explorations of data quality mainly related to features and target variables could be done by conducting exploratory data analysis. All the potential issues that could be resolved could be done by conducting a comprehensive data processing method.

2.2 Data Pre-processing

Imputation and Removal of Unnecessary ID's

While the dataset was largely intact, we did encounter a few missing rows. Initially, we considered imputation to fill in these missing entries. However, given that only four rows were missing out of around 19,000, we concluded that imputation would not significantly affect the model's performance. Moreover, the effort to impute such a small fraction of data was not justified in terms of outcome improvement. Instead, we

chose to remove these rows — particularly those with unnecessary or corrupted IDs to maintain data consistency and focus on refining the remaining dataset for better accuracy, reliability, and interpretability.

2.3 Exploratory Data Analysis

Revising Existing Data

Looking at the nature of the dataset, it would be sure that there exist mainly 3 types of data in the initial dataset, which are IDs, chat-related content and outcome score (target), as being observed in Table 2.1.1. There existed potential to conduct data processing (feature engineering) in aspects of content (that contains text-based data), structure (where exploration could be done on how the structural-based factor like team size or number of mentor could affect the result), and progress (where all information related to the ongoing virtual internship program could be documented, like the interaction between players, type of mentor etc). Additionally, as per the requirements of our project, data aggregation to team-level was required in order to acquire team-level statistics.

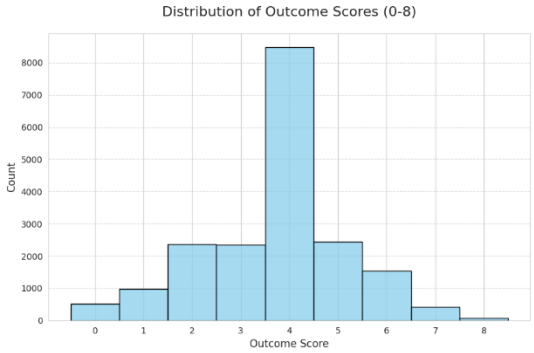

Figure 2.3.1: Distribution of Outcome Scores

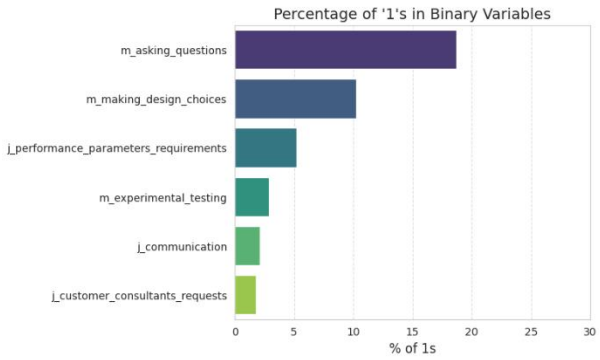

Figure 2.3.2: Observations of Binary Variables via Percentage of '1's

We could have observation of the available target variable by observing Figure 2.3.1, which the distribution of outcome scores being presented in histogram. From the figure, we could observe a very high peak appearing at the Class 4; while a moderate decline appeared at Class 0 and Class 8. This might cause potential imbalance of the data. In other words, modellings in our project might favour to Class 4 which appeared to be majority and dominant class in our target variables. This will result in severe performance in predicting other class, especially the minority classes.

While observing Figure 2.3.2, which is the observations of binary variables that are mainly linked to the "content" columns, we could observe an overall low occurrence of all variables with occurrence of < 20% throughout the dataset. m_asking_questions showed with highest occurrence with 18% and j_customer_consultants_requests appeared with lowest occurrence with < 3%. Low occurrence of these binary variables might cause severe underfitting on models, as these variables, would be unable to contribute well due to its low occurrence to predict the outcome score.

2.4 Data Processing

Content-Based Feature Engineering

The 'content' column that contains chatlogs represented the richest source of qualitative data. However, it still required substantial preprocessing before meaningful quantitative features could be extracted. There

existed numerous challenges that were prevalent in the informal chat-based 'content' column. Upon further analysing, we realized that the chat would be unlike formal written text that normally utilized standard English. The chat logs contained numerous criteria that would only appear in informal manner, such as abbreviations, slang, local terms, contractions in English, and punctuations that may affect the data noise-wise (Wang, 2022). Thus, we identified the challenges that exist in informal text and conducted handling on each challenge, in which we conducted initial clean-up, text normalization, slang normalization, spelling correction, stop word removal, name entity recognition, negation handling, and lemmatization (Amiot et al., 2025). Later on, we investigated the difference between pre-processed and raw text to ensure the consistency of the text-preprocessing workflow.

Term Frequency-Inverse Term Frequency (TF-IDF)

For term importance analysis, we implemented TF-IDF, which combines the weight of term frequency within individual chat lines with the inverse frequency across the entire corpus. The approach effectively identified terms that were particularly important to specific discussions while down-weighting common words (Rani et al., 2022). As the result of our TF-IDF feature engineering approaches, we gained 4400+ new columns containing all terms related to discussion. Upon discovering an increase in the number of columns added to our dataset, we decided that further investigation focused on filtering unnecessary columns would be done later on.

Sentiment Analysis

Sentiment Analysis is a technique that identifies and extracts subjective information from text, such as opinions, emotions, and attitudes. Based on the pre-processed texts, we performed sentiment analysis to quantify the emotional tone of communications. For each message, the sentiment score would be calculated as a weighted average of terms polarities that exist in range of [-1, 1]. This allowed us to track not only terms appeared in discussions, but also emotional context appeared in discussions (Rani et al., 2022).

Structure-Oriented and Progress-Oriented Feature Engineering

Beyond content-column related data processing, feature engineering was also conducted on structural aspects of team composition (Structure-Oriented Features) and progressive engagement patterns (Progress-Oriented Features) throughout the internship.

For Structure-Oriented Features, we calculated the group size, which consist of the number of players in a group. Then we calculated the number of mentors that exist in each group, as well as the player-mentor ratio of each group by using the formula: $\frac{number\ of\ player\ in\ the\ group}{number\ of\ mentor\ in\ the\ group}$. As the result, we gained 3 new columns, which are 'group_size", 'mentor_count', and 'player_to_mentor_ratio'.

As for Progress-Oriented Features, we calculated the activeness metrics by tracking message frequencies for both players and mentors. Engagement metrics were computed by taking the sum of topic-specific interaction flags. Particular focus would be on mentor-influenced metrics, in which questioning behaviour (frequency of asking questions) and directiveness (frequency of making design choices) were separately tracked on mentors. These metrics allow us to characterise different mentoring styles and their potential impact on team outcomes.

Data Aggregation

We then turned to the critical task of aggregating individual-level data to team-level data for our analysis. Different feature types demanded different aggregation strategies. For TF-IDF features, summation was being conducted to preserve cumulative term importance across team members. Sentiment scores were averaged to capture overall team tone. Structure-oriented features that were constant amongst teams could be aggregated by simply taking the first value, whereas progress-oriented features were summed to reflect total team engagement.

Nevertheless, the most challenging decision on aggregation approach would be on the 'OutcomeScore' column. This section must be completed accurately to ensure that we can uncover the underlying patterns in the original data. Thus, we evaluated each aggregation approaches by using Mean Squared Error (MSE) against original data. The lower the mean-squared error, the higher the usability of the aggregation approach (Feng et al., 2024).

There existed a number of potential aggregation approaches on the 'OutcomeScore' column, which were: (1) Mode, taking the most frequent score; (2) Mean, averaging of all scores; (3) Median, using the middle value; (4) Weighted Average (WAM), calculating $\frac{1}{\sum_{i=1}^{n} w_i}\sum_{i=1}^{n} w_i y_i$, weighted by player activeness; and (5) Batch Gradient Descent, which evaluate optimal weights of $\frac{1}{\sum_{i=1}^{n} w_i}\sum_{i=1}^{n} w_i y_i$ on each team member in every group by minimising MSE.

Table 2.4.1: Mean Squared Errors of Aggregation Approaches

Aggregation Approach	Number of rows	Mean-Squared Error
Raw data (for comparison)	19180	0.00
Mode	594	2.94
Mean	594	1.86
Median	594	1.87
Weighted Average Mark (WAM)	594	2.07
Batch-Gradient Descent	594	1.75

Observing Table 2.4.1, mode appeared to be the worst aggregation approach with the highest mean-squared error of 2.94; while Batch-Gradient Descent appeared to be the best, with lowest mean-squared error of 1.75. Therefore, we decided to adopt a batch-gradient descent aggregation approach for our aggregation method on the 'OutcomeScore' column.

2.5 Baseline Modelling

Training Baseline Model

For our baseline model, K-Nearest Neighbours (k-NN) was selected for several key reasons. As a lazy learner, k-NN offered fast training whilst maintaining simplicity and high interpretability in comparison to other complex models like Decision Trees or Random Forest (Suyal & Goyal, 2022). Its classification-based approach aligned well with the nature of our dataset, unlike other regression models that seemed unfeasible and unsuitable. Importantly, k-NN does not assume feature independence, making it much more appropriate in comparison to Naïve Bayes for our data structure (Suyal & Goyal, 2022). We used a 80/20 ratio training-testing set, and conducted initial evaluation through confusion matrix analysis and standard metrics.

Performance of the Baseline Model

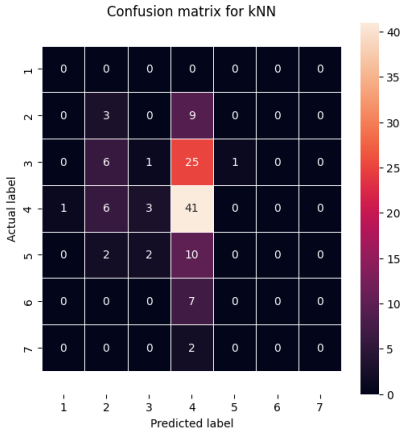

Figure 2.5.1: Confusion Matrix of Baseline Model

From the confusion matrix in Figure 2.5.1, we observed a domination of Class 3 and Class 4, which mainly show the existence of imbalance in data that causes the model to bias towards the major classes. Additionally, the high-dimensional nature of the data could be due to irrelevant features that were built up, causing the baseline model to be misled. Furthermore, as we investigate the potential of different scale showing in the dataset, improper scales might be a contributing factor that causes the data to perform. Looking at the confusion matrix, we could calculate the accuracy score based on $\frac{TP+TN}{TP+TN+FP+FN}$, the precision score based on $\frac{TP}{TP+FP}$, the recall score based on $\frac{TP}{TP+FN}$, and the F1-Score based on $\frac{2\ (Precision)(Recall)}{Precision+Recall}$.

Table 2.5.1: Performance of Baseline Model

Metrices	Accuracy (Train)	Accuracy (Test)	Precision	Recall	F1-Score
Performance	0.581	0.269	0.220	0.269	0.241

While looking at Table 2.5.1, we could see that there existed a large gap between the training accuracy and the testing accuracy. We hypothesized that the model seemed to memorize only the training data and failed to predict unseen data. Our hypothesis was supported when we looked at the low precision (high false positives), low recall (missing many true positives), and low f1-score. Those all confirmed the overall weak performance of our baseline model. Additionally, while evaluating the model performance, we decided to prioritise the F1-score, then prioritise testing accuracy over other metrics considering our nature of dataset that appeared to be imbalanced, and the F1-score appeared to be more robust metrics that combined precision and recall, that allow more evaluation on minority classes (Naidu et al., 2023).

2.6 Model Assessment

Normalization

Considering the difference in scale amongst the features, we assessed the effects of conducting normalization by using the baseline model.

Table 2.6.1: Performance Comparison

Version	Accuracy (Train)	Accuracy (Test)	Precision	Recall	F1-Score
Baseline Model	0.581	0.269	0.220	0.269	0.241
Baseline Model with Normalization	0.596	0.378	0.251	0.378	0.277

Observing Table 2.6.1, the experimental results demonstrate that normalising the features led to measurable improvements in the model's performance. Specifically, all metrics had shown increases that suggest the positive effect of normalization, allowing the distance-based k-NN algorithm to make more reliable predictions. However, despite the above observations, we decided that the overall performance of normalization is still limited, therefore decided to conduct further investigation of methods and approaches.

Feature Selection: Solving Multicollinearity & Setting Variance Threshold

The feature selection process began by addressing 2 critical issues in the 4456-features dataset, which were multicollinearity and low variance. Initial analysis was conducted on a range of feature correlations and standard deviations. As a result, extreme cases were revealed with absolute feature correlation that range from 0.00 to 1.00, with 827 pairs of features having absolute correlation of more than 0.95. Additionally, extreme cases of standard deviations ranging from 0.01 to 2999.40 in which 2148 features were revealed to be having standard deviations of less than 0.01. Therefore, further experimental approaches of features filtration were conducted by setting 0.95 as the correlation threshold, and 0.10 as the variance threshold (Kaur et al., 2021).

Table 2.6.2: Effect of Threshold Setting

Version	**Number of Features**	**Accuracy (Train)**	**Accuracy (Test)**	**Precision**	**Recall**	**F1-Score**
Baseline Model	4456	0.581	0.269	0.220	0.269	0.241
Baseline After Solving Multicollinearity	2829	0.587	0.269	0.222	0.269	0.242
Baseline After Solving Multicollinearity and Setting Variance Threshold	594	0.587	0.269	0.222	0.269	0.242

According to Table 2.6.2, the experimental results revealed key insights about the impact of feature selection on model performance. Starting with the baseline model with 4,456 features, we observed a significant gap between the training accuracy score (0.581) and the test accuracy score (0.269), along with poor precision (0.220), recall (0.269), and F1-scores (0.241), indicating severe overfitting and limited generalization capability. After addressing multicollinearity which reduced the features to 2,829 and setting variance threshold which reduced features to 594, the performance metrics of the model remained nearly identical. However, the complexity (i.e the dimension of feature space is greatly reduced) of the model was reduced, deeming that both solving multicollinearity and setting a variance threshold would be useful in improving our model.

Feature Selection: Conducting Statistical Testing

We decided to conduct statistical testing to our target column, ‘OutcomeScore’ by using Chi-squared test and ANOVA F-test. We set up hypothesis testing for Chi-squared test, with an alternative hypothesis (H_A) that there is an association between the feature and the target variable. As for the ANOVA F-test, we set up an alternative hypothesis (H_A) such that the mean of the features would be different for at least one class (Kaur et al., 2021). For each test, once the p-value is less than the certain threshold (0.05), it would mean that the alternative hypothesis would be adopted due to significance (Kaur et al., 2021). Additionally, the Bonferroni procedure was not adopted as the number of features would shrink from 4456 features to 3 features, proving itself as too strict in our case.

Table 2.6.3: Effect of Statistical Testing

Version	Number of Features	Accuracy (Train)	Accuracy (Test)	Precision	Recall	F1-Score
Baseline Model	4456	0.581	0.269	0.220	0.269	0.241
Baseline with both Chi-squared Test and ANOVA F-test	136	0.486	0.437	0.244	0.437	0.283

According to Table 2.6.3, we observed an improvement of accuracy, precision, recall and F1-score generally. This means that the feature selection process successfully eliminated uninformative features and appeared to be balanced between the bias and variance.

Resampling

Regardless of our significant process and new findings, we have yet to conquer the imbalanced nature of the data. Thus, upon deeper analysis, we decided to resample our data using the effect of undersampling, and SMOTE oversampling techniques that would be tested separately respectively (Wongvorachan & Bulut, 2023).

Table 2.6.4: Effect of Undersampling and Oversampling

Version	Training Samples	Accuracy (Train)	Accuracy (Test)	Precision	Recall	F1-Score
Baseline Model	475	0.581	0.269	0.220	0.269	0.241
Undersampling	16	0.126	0.126	0.016	0.126	0.028
Oversampling	1752	0.534	0.432	0.172	0.134	0.135

From Table 2.6.4, the baseline model showed the poorest overall performance, indicating fundamental issues with class imbalance or feature quality. Undersampling the data failed due to excessive data reduction, while oversampling improved test accuracy. Therefore, we concluded that the oversampling technique would be a better approach for our project.

Model Assessment Insights

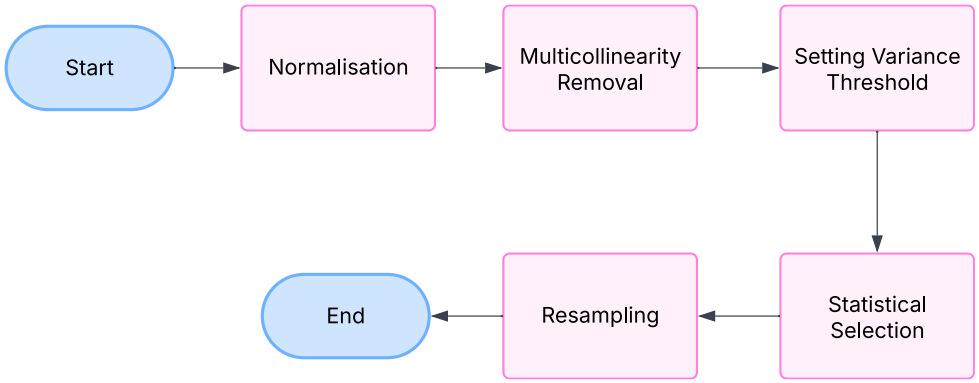

Figure 2.6.2: Data Processing Workflow Prior to Model Development

After conducting experimental approaches on normalization, feature selection, and resampling, we concluded one data processing flow prior to model development, in which each processing step would be based on justification of usability for experimental approaches conducted before, according to Figure 2.6.2.

Table 2.6.5: Comparison between Baseline Model and Improved Model

Version	Number of Features	Training Samples	Accuracy (Train)	Accuracy (Test)	Precision	Recall	F1-Score
Baseline Model	4456	475	0.581	0.269	0.220	0.269	0.241
Improved Model	136	1752	0.763	0.720	0.767	0.719	0.704

Based on Table 2.6.5, we could see a clear reduction in number of features with increase of all metrices. This has proven the usability of the architecture shown in Figure 2.6.2 prior to model development, which would be adopted for the following modelling options proven by success example of k-NN baseline model.

2.7 Model Development

Model Selection Strategy

The modelling approach in this project would be to focus on classification-based algorithms rather than regression-based algorithms or clustering methods, as our problem involved predicting categorical outcomes rather than continuous values that required supervised learning. Therefore, regression methods that are designed for continuous targets and clustering methods that are designed for unsupervised learning situations would be less suitable in our case. Therefore, the only considerable models for us to consider in this project would be Support Vector Machine, Decision Tree, Random Forest, Gradient Boosting, LightGBM and Logistic Model.

Support Vector Machine (SVM)

SVM is a classifier that finds optimal decision boundaries by maximising margin between classes. SVM can handle non-linear decision boundaries using different kernel functions to map the input data into higher dimensional spaces, which deemed appearing to be a suitable model in our case (Balaji et al., 2021; Gupta et al., 2022). For SVM, we expected a limitation of sensitivity to noise and outliers, considering the large number of features appearing in our dataset. Additionally, for this model, we would expect a high training time as it would be quite computationally intensive (Singla & Shukla, 2019).

For SVM, first we ran the baseline SVM model based on default settings. Then, we conducted grid-search based hyperparameter tuning: on kernel with choices of ‘poly’, ‘linear’, ‘rbf’ and ‘sigmoid’; regularization parameter with choices of 0.1, 1, 5, 10, 100; gamma with choices of ‘auto’, ‘0.01’, ‘0.001’, ‘0.0001’; and decision function shape of ‘ovr’ and ‘ovo’. For each hyperparameter tuning loop, 5-fold cross validation was conducted to ensure the consistency of result.

Decision Tree

Decision Tree is a type of supervised learning method that allow recursive splits of data into subsets based on given feature values. It appeared to be suitable here as its ability to handle classification tasks (Gupta et al., 2022). For decision tree, we would expect a risk of overfitting to training data, in which when the number of depth is too high, it might be too deep to capture noise in the training data, causing a bad testing result. This might cause it to bias towards classes that dominate dataset, in this case, Class 4 (Gupta et al., 2022).

A similar approach was conducted on Decision Tree, in which a baseline Decision Tree was first computed. Then, hyperparameter tuning with 5-fold cross validation was conducted with the setting options of max_depth with choices of 5, 10, 15, 25, 30, 50; min samples split of 2, 5, and 10; min samples leaf of 1, 2, 4; and splitting criteria of either based on gini or entropy.

Random Forest

Random Forest, is an ensemble learning method that constructs multiple trees for training. Due to its bagging and random feature selection natures, it appeared to be a suitable model here to enhance model robustness (Balaji et al., 2021; Gupta et al., 2022). However, despite its availability of solutions of feature selection, it would be less interpretable to understand how trees "votes" and "make decisions", in which it would be harder to interpret all trees to understand how pruning works in trees (Gupta et al., 2022).

Similarly, a baseline random forest model was first computed. Then, hyperparameter tuning with 5-fold cross validation was conducted with the setting options of n_estimators with choices of 50, 100 and 200; max_depth of None, 5, and 10; min_samples_split of 2 or 5; min_samples_leaf of 1 or 2 and max_features of sqrt, log2 or auto to ensure the max number of features would be limited to reduce variance and bias.

Gradient Boosting

Gradient Boosting is an ensemble method that builds decision trees sequentially, with each new tree correcting errors made by previous ones. It is suitable here as it could capture nonlinear relationships and its ability to handle interactions between features (Balaji et al., 2021; Gupta et al., 2022).

We trained a model using default settings (n_estimators = 100, learning_rate = 0.1, max_depth = 3). Due to computational limitations, further hyperparameter tuning and cross validation was not conducted. Nevertheless, gradient boosting is a model that sensitive to hyperparameters, in which the performance of the gradient boosting is highly related to the choice of hyperparameters. Therefore, in our case, we would expect a moderate result of modelling due to our computational constraints (Bentéjac et al., 2020).

LightGBM

LightGBM is an advanced gradient boosting framework that employs leaf-wise tree growth and histogram-based algorithms to achieve exceptional computational efficiency and scalability (Gupta et al., 2022). We chose it for its ability to handle imbalance dataset and scalability with large datasets.

However, due to computational constraints, we used default settings (num_leaves = 31, learning_rate = 0.1), which provided good performance without extensive tuning. Its built-in regularization helped prevent overfitting while maintaining reasonable accuracy. Our limitations could further contribute to biasness of model, as similarly to gradient boosting, LightGBM is also a model that sensitive to hyperparameters. Thus, in our case, we would expect a similar, moderate result of modelling in LightGBM (Bentéjac et al., 2020).

Logistic Model

Logistic Model, that its nature that could be extended to multilabel tasks by training multiple classifiers made it to be suitable in our case. It is not only fast and efficient, but also contains built-in regularizations, in which allowing more focus to be put on errors (Balaji et al., 2021; Gupta et al., 2022). Nonetheless, considering the complex nature of our dataset, Logistic Model might struggle to handle our data that are mostly formed by binary variables, in which its nature of assuming linear relationships between features might struggle in predictions (Gupta et al., 2022).

For logistic regression hyperparameter tuning, firstly we computed the baseline logistic model. Then, we employed GridSearchCV to systematically evaluate different regularization approaches and solver combinations, by testing each L1 penalty, L2 penalty, elasticnet, and no penalty configurations across, using 5-fold cross-validation (cv = 5) to identify the optimal model configuration that balances regularization strength and solver performance.

Chapter 3 Result and Discussion

3.1 Assessing Effect of Model Development

Table 3.1.1: Performance of The Models

Version	Accuracy (Train)	Accuracy (Test)	Precision	Recall	F1-Score
Baseline Model (kNN) Without Data Processing	0.581	0.269	0.220	0.269	0.241
Baseline Model (kNN) After Data Processing	0.763	0.720	0.767	0.719	0.704
SVM Baseline	0.731	0.695	0.739	0.695	0.689
SVM Hyperparameter Tuned (linear, C = 5.0)	0.774	0.713	0.780	0.713	0.710
Decision Tree Baseline	0.861	0.713	0.753	0.713	0.706
Decision Tree Hyperparameter Tuned (Max Depth = gini, min samples split = 1)	0.861	0.702	0.734	0.702	0.690
Random Forest Baseline	0.861	0.754	0.800	0.754	0.751
Random Forest Hyperparameter Tuned (Max Depth = 200, Min Sample Split = 5, n_estimators = 200, max_features = sqrt)	0.851	0.759	0.808	0.759	0.758
Logistic Model Baseline	0.727	0.699	0.740	0.699	0.696
Logistic Model Hyperparameter Tuned	0.732	0.699	0.739	0.699	0.696
LightGBM	0.835	0.702	0.766	0.702	0.702
Gradient Boosting	0.810	0.706	0.743	0.706	0.697

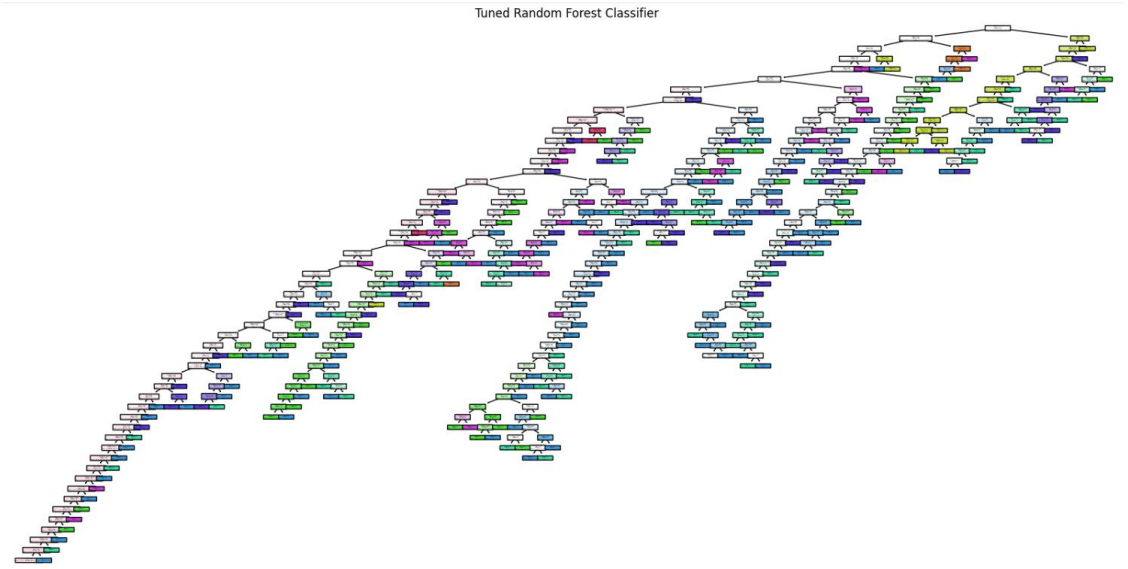

Figure 3.1.1: Decision Tree from Tuned Random Forest Classifier

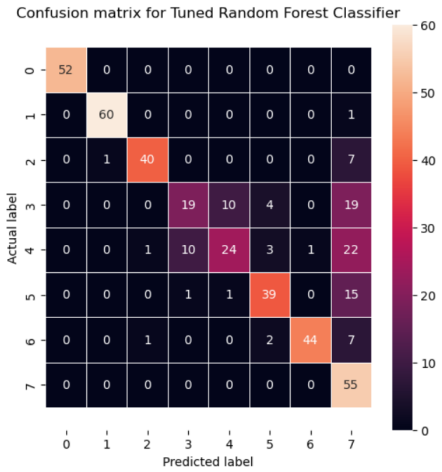

Figure 3.1.2: Confusion matrix of Tuned Random Forest Classifier

From Table 3.1.1, the Hyperparameter tuned Random Forest Classifier performed the best out of all the other models. The most optimal hyperparameters were max Depth = 200, Min Sample Split = 5, n_estimators = 200, max_features = sqrt. It has the testing accuracy score of 0.759 and F1-Score of 0.758, meaning that the model is performing reasonably well with generally 76% in both metrics. The max depth of this model is 200, as it is illustrated by Figure 3.1.1, the model grows heavily towards left, indicating a deeper and more complex decisions made on that path.

Additionally, in Figure 3.1.2, we observe that the model predicts class 7 more than any other classes. This suggests that although the dataset is balanced, the model is still biased towards class 7. Finally, the reason Random Forest performs so well is since it can handle both categorical and regression variables, which aligns well with the mixed-type features presented in our dataset (Parmar et al., 2018).

The model with the worst performance was the baseline model (kNN) without any data processing, with the testing accuracy score 0.269 and F1-Score of 0.241. One of the key issues present in our unmodified dataset is imbalanced classes. This means that the target variable's classes were not equally proportioned (i.e some classes have significantly higher sample than others). (Chawla et al., 2004)

In our case, the class with highest number of samples is class 4. This results in the model favouring towards class 4, it makes more predictions on class 4 rather than others. The tendency is evident through the confusion matrix in Figure 2.5.1. The model has made overall 94 predictions on class 4, with an accuracy score of 0.44 for that class alone.

Additionally, because the dataset was unprocessed, the texts in the "content" were not converted into machine interpretable numerical format. Since kNN relies on numeric feature input to calculate distances between samples, this lack of transformation contributed significantly to the poor performance of the model.

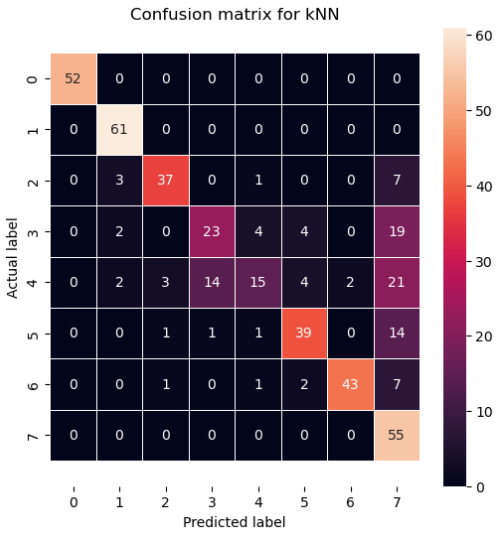

Figure 3.1.3: Confusion matrix of kNN model using Processed Data

Figure 3.1.3 presents the confusion matrix for kNN model using the processed data. By comparing Figure 2.5.1 with Figure 3.1.3, it is quite clear the model is making less biased predictions, however it seems still favour to class 7. The model has made a perfect accuracy score of 1 in predicting class 0, suggesting that it has learnt quite well distinguishing class 0 from all other classes. Furthermore, the testing accuracy score is 0.720 and F1-Score is 0.704, which increased dramatically. This substantial increase of accuracy score and F1-Score strongly suggest that the data preprocessing steps were effective and contributed positively to the model's learning (Saarela & Jauhiainen, 2021). Therefore, we conclude that the data preprocessing approaches were effective, and the modified dataset should be used to build the models.

3.2 Discoursing Relation between Features and Target Variables

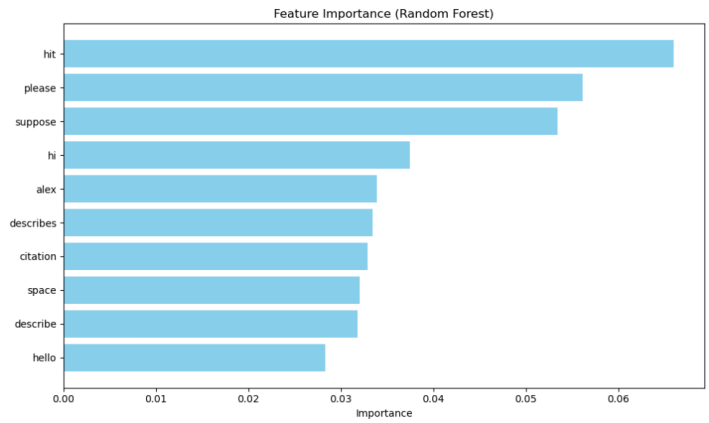

Figure 3.2.1: Feature Importance Scores Based on Fine Tuned Random Forest Classifier

Figure 3.2.1 shows the top 10 highest feature importance scores, based on our best performed model (Fine Tuned Random Forest Classifier). Based on this graph, we observe that the keywords "hit" and "please" were the most two important features. Despite this, the overall feature importance score for all the features within the dataset is quite low, which suggests that the model does not strongly depend on any feature, and each feature contributed similar amount in predicting the target (Saarela & Jauhiainen, 2021).

Additionally, we also observed that the there is no any mentor related features appeared on this graph. This is suggesting that mentor-related features did not significantly contribute to model's predictions and that it may have minimal impact on helping players.

Furthermore, we noticed that there is overlap and redundancy within the dataset. For instance, the words "describe" and "describes" both present in the graph indicates the limitation of NLTK pos tagging ability in lemmatising. Addressing this through proper text normalization could improve the performance of the model (Rani et al., 2022).

Overall, we are unable to simply describe the relationship between features and target variables. This indicates that a non-linear and complex relationship between input and target

Chapter 4 Conclusion and Future Work

4.1 Summary

Summary of Results

We were able to conclude that the best model for our project was the hyperparameter tuned random forest classifier, as it is able to handle both categorical and regression variables. From there, we revisited our objectives and ended on a positive note that we have met all our objective successfully.

We conclude that all the communication features play significant parts in contributing to the team report performance, and that the contribution of mentors do not significantly affect the team report performance, unlike what we hypothesized.

Objectives

Looking back to our project objectives: (1) Firstly, all data was successfully transformed into team-level statistics for modelling purpose using content-based feature engineering, progress-oriented feature engineering, structure-oriented feature engineering and data aggregation to group level using gradient descent approach. (2) We had built predictive models to predict final report scores based on team communication behaviors, which the highest performing model appeared to be tuned random forest classifier. (3) We interpreted the results of the models to understand how communication features relate to the team report performance, in which for this project all the filtered features contributed fairly well with the same amount in predicting the outcome score.

4.2 Limitations and Future Work

Limitations due to the nature of the data

Although the "content" column is rich in containing valuable information, its format (unstructured textual data) is difficult to interpret directly by the machine learning models. It must be processed/transformed into numeric format to allow models to learn and make predictions. Since most models require numeric input, these texts must be transformed into a numeric representation – such as through vectorization (e.g. TF-IDF). However, this step is often complicated and complex. If the data is not transformed properly, this may result in the underperformance of the model – the model will struggle to extract meaningful patterns. Additionally, improperly processed text data can cause the model may to miss important semantic relationships (e.g., synonyms). Some words may have similar definitions however different sentiment polarities. This will also result in the reduction of accuracy scores and poor generalization.

Limitations of methodology

In our data-processing section, we have noticed that there were some overlaps in our modified dataset. As it is shown in Figure 3.2.1, the words "describe" and "describes" both present in the feature importance graph. Although these words share the same root meaning but differ only in grammatical tense, they were treated as separate features in our feature space. This is the result due to not properly applying text normalization techniques, such as lemmatization or stemming during the preprocessing part. This inconsistency can negatively affect the model's performance by introducing noise and obscuring semantic patterns.

Additionally, based on the performance of the Fine-Tuned Random Forest Classifier (Figure 3.1.3), we also see that despite we have resampled our dataset, the model is still making disproportionately high number of predictions for class 7 compared to all other classes. This indicates that the model is biased towards predicting

class 7. Such behaviour may be attributed to potential issues such as improper application of the resampling technique, or possible data leakage between training and testing data.

Expectations in Future

To overcome the problem of inconsistencies in text normalization, first we can manually inspect the samples (20-50) of the processed data, checking whether the common variations of words are properly reduced (e.g. “run” vs “running”), stop words or punctuation correctly removed, or if the meaning words are preserved. Then we should consider a reliable Natural Language Processing library (e.g. NLTK, spaCy), for text normalization.

Additionally, the persistent bias towards class 7 after applying resampling is an issue that we certainly need to address. Instead of applying resampling to the entire dataset, we need to ensure that it is only applied to training set. If resampling is applied before the train-test split, it can lead to data leakage (which in our case, we have applied to the entire dataset and before the train-test split). This contaminates the evaluation process, causing the model to overfit and results in misleading performance metrics, greatly impact its accuracy score. Therefore, by strictly applying resampling to training set only after the train-test split, we can achieve a model that is less biased with overall better performance.

Finally, for future modelling efforts, greater emphasis should be placed on the tree-based algorithms, such as CatBoost, XGBoost, as they are well-suited for mixed data types effectively. Furthermore, deep learning approaches should also be considered, particularly neural networks, as it combines dense text features with categorical embeddings and numerical features, offering a more powerful modelling framework.

References

Amiot, C., Charoy, F., & Dinet, J. (2025, May 2). Chatbots in Collaborative Settings and their Impact on Virtual Teamwork. *Proceedings of the ACM on Human-Computer Interaction*, *9*(2), 1–18. https://doi.org/10.1145/3710945

Balaji, T.K., Annavarapu, C. S. R., & Bablani, A. (2021, March 20). Machine learning algorithms for social media analysis: A survey. *Computer Science Review*, *40*(1). https://doi.org/10.1016/j.cosrev.2021.100395

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2020, August 24). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*(3). https://doi.org/10.1007/s10462-020-09896-5

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004, June 1). Editorial. *ACM SIGKDD Explorations Newsletter*, *6*(1), 1. https://doi.org/10.1145/1007730.1007733

Chesler, N. C., Ruis, A. R., Collier, W., Swiecki, Z., Arastoopour, G., & Williamson Shaffer, D. (2015, February 1). A Novel Paradigm for Engineering Education: Virtual Internships With Individualized Mentoring and Assessment of Engineering Thinking. *Journal of Biomechanical Engineering*, *137*(2). https://doi.org/10.1115/1.4029235

Feng, X., Liu, H., Yang, H., Xie, Q., & Wang, L. (2024, February 9). Batch-Aggregate: Efficient Aggregation for Private Federated Learning in VANETs. *IEEE Transactions on Dependable and Secure Computing*, *21*(5), 4939–4952. https://doi.org/10.1109/tdsc.2024.3364371

Gupta, V., Mishra, V. K., Singhal, P., & Kumar, A. (2022, December 1). *An Overview of Supervised Machine Learning Algorithm*. IEEE Xplore. https://doi.org/10.1109/SMART55829.2022.10047618

Kaur, A., Guleria, K., & Kumar Trivedi, N. (2021, April 20). Feature Selection in Machine Learning: Methods and Comparison. *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. https://doi.org/10.1109/icacite51222.2021.9404623

Naidu, G., Zuva, T., & Sibanda, E. M. (2023, July 9). A Review of Evaluation Metrics in Machine Learning Algorithms. *Lecture Notes in Networks and Systems*, *724*, 15–25. https://doi.org/10.1007/978-3-031-35314-7_2

Parmar, A., Katariya, R., & Patel, V. (2018, December 21). A Review on Random Forest: An Ensemble Classifier. *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*, *26*, 758–763. https://doi.org/10.1007/978-3-030-03146-6_86

Rani, D., Kumar, R., & Chauhan, N. (2022, October 1). Study and Comparision of Vectorization Techniques Used in Text Classification. *IEEE Xplore.* https://doi.org/10.1109/ICCCNT54827.2022.9984608

Saarela, M., & Jauhiainen, S. (2021, February 3). Comparison of feature importance measures as explanations for classification models. *SN Applied Sciences*, *3*(2). https://doi.org/10.1007/s42452-021-04148-9

Singla, M., & Shukla, K. K. (2019, December 2). Robust statistics-based support vector machine and its variants: a survey. *Neural Computing and Applications*, *32*(15), 11173–11194. https://doi.org/10.1007/s00521-019-04627-6

Suyal, M., & Goyal, P. (2022, July 18). A Review on Analysis of K-Nearest Neighbor Classification Machine Learning Algorithms based on Supervised Learning. *International Journal of Engineering Trends and Technology*, *70*(7), 43–48. https://doi.org/10.14445/22315381/ijett-v70i7p205

Wang, Y. (2022, September 19). Using Machine Learning and Natural Language Processing to Analyze Library Chat Reference Transcripts. *Information Technology and Libraries*, *41*(3). https://doi.org/10.6017/ital.v41i3.14967

Wongvorachan, T., He, S., & Bulut, O. (2023, January 16). A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining. *Information*, *14*(1), 54. https://doi.org/10.3390/info14010054

Appendix 1: Text Normalization Challenges

Table 5.1: Text Normalization Challenges

Challenge Type	Example	Normalised Form	Impact if Unaddressed
Abbreviations	"lol"	"laugh out loud"	Misinterpretation of informal terms
Slang Expressions	"WHO ELSE WA"	"who else"	Noise in sentiment/TF-IDF analysis
Misspellings	"complx"	"complex"	Inconsistent term frequency counts
Local Terms	"Macca's"	"McDonald's"	Geographic bias in term relevance
Contractions	"can't"	"cannot"	Split terms may lose semantic meaning
Punctuation	"!!!" or ":D"	"[excitement]" / "[smile]"	Distorted sentence segmentation
Capitalization	"HELLO"	"hello"	Duplicate terms in case-sensitive models

Appendix 2: Calculation for TF-IDF & Sentiment Analysis

Table 6.1: Calculation for TF-IDF and Sentiment Analysis

Component	Formula	Description
Term Frequency (TF)	$TF(t,d) = \frac{Count\ of\ t\ in\ d}{Total\ terms\ in\ d}$	Measures local importance of terms
Inverse Term Frequency (IDF)	$IDF(t,D) = \log\left(\frac{\lvert D\rvert}{Documents\ containing\ D}\right).$	Rank global importance of terms
TF-IDF	$TF - IDF(t,d,D) = TF(t,d) \times IDF(t,D).$	Final weighted importance of terms
Sentiment Scores	$Sentiment = \frac{\sum_{i=1}^{n} polarity \cdot weight}{\sum_{i-1}^{n} weight}$	Sentiment Analysis classifies sentiment as positive, negative, or neutral.

Appendix 3: Text Pre-processing Workflow

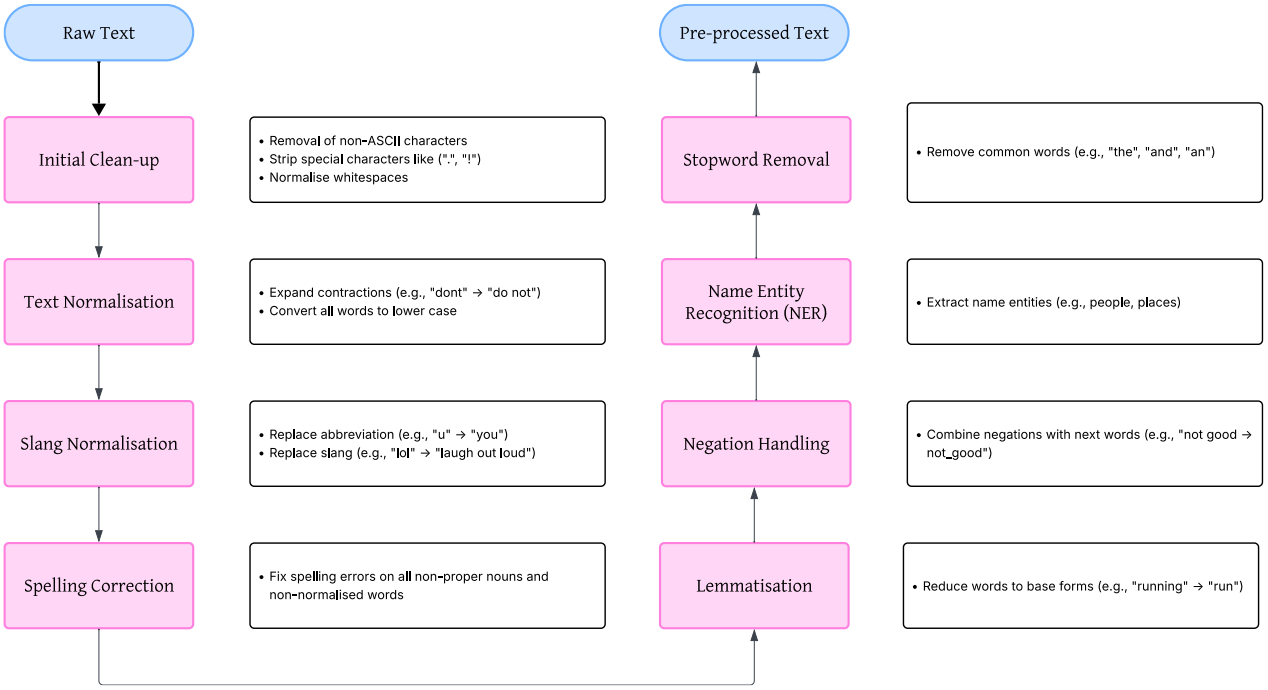

Figure 7.1: Text Pre-processing Workflow for Each Text Normalization Challenges

Table 7.2: Effect of the Text-Preprocessing

Aspect	Observations
Number of terms removed	9191
Number of terms added	1692
Average length reduction (text per line)	6.069
Jaccard similarity between raw text and pre-processed text	0.245

Appendix 4: Aggregation Approaches for Features and Target in Dataset

Table 8.1: Aggregation Approaches According to Rationalised Reasons

Columns	Aggregation Way
Columns constructed from TF-IDF	By taking the sum value of common rows
Columns constructed from Sentiment Analysis	By taking mean of the common rows
Columns constructed from Structural-Oriented Factor	By taking the first value of common rows
Columns constructed from Progress-Oriented Factor	By taking the sum value of common rows
ID's in raw dataset	By taking the first value of common rows
Metrics based Columns in raw dataset	By taking the sum value of common rows
Target Column	By taking the computed scores based on Batch Gradient Decent Algorithm