Virtual Internships

ADS2001 Final Report

(Top 25 Team Communication Apps for Businesses in 2023, 2021)

Arya Krishna
Ben Gale
Grita Gupta
Huy Hoang Ho
Mia Yabut
Saksham Sharma

Table of Contents

Executive Summary

Problem Statement

This report details the analysis of the virtual internship simulation at Nephrotex. This project is intended to investigate any notable patterns within these internships that can contribute to a student's performance. Studying these trends will be used to implement improvements in future runs of the internship to enhance the performance of students. As the data provided predominantly includes the chats and how each conversation relates to the engineering design task, the analysis involves exploring the communication between players and with mentors and how that further impacts the outcome. To find what aspects of the virtual internship can be changed for the better, modelling the performance based on the conversations was vital as the dataset was able to categorise the type of communication taking place and therefore can make conclusions from this in the behaviours and actions that should be prioritised to ensure a better outcome.

Methodology

The approach taken to this dataset included addressing and inconsistencies, exploratory data analysis and further modelling to survey the relationships on an individual level, team level, mentor level and overall. By splitting our investigation based on these levels and surveying the different variables individually at first (e.g. ‘content’, ‘topic of discussion’, ‘mentor effect’), we would be able to create a conclusion that had explored all aspects of the internship. Hence, the most important features impacting performance can be deduced.

To code with the data, Python was used alongside ‘pandas’ as a tool to handle the dataframe and ease of manipulation of that dataset; ‘Scikit-learn’ was also utilised to deal with more technical processes of the project in exploratory analysis and modelling.
A variety of strategies were applied for data analysis before modelling and in combating the inconsistencies dealt with at first glance. This included applying SMOTE (an oversampling technique) to handle bias due to the overwhelming proportion of average performing players to distinctly high and low performing players. Being able to reduce that bias allowed for our analysis to cover those ‘minority’ groups and produce further insight on what contributes to their outcomes too. Cleaning the data by removing null values and noting that some placeholder values such as the outcome score of 4 for all mentors can be negated also helped in reducing the large dimensionality of this dataset. To reduce the dimensionality in modelling certain parts of the dataset, PCA was applied to reduce but keep as much variability as possible such that less interpretation is lost.
To efficiently decode the worded content, TF-IDF was a tool used to discover any notable linguistic patterns between players themselves and mentors that had a correlation to the outcome score of these players. Clustering was also used to deduce how clusters of poor, average and high performing groups had performed and what variables in the internships would allude to their result.

Modelling was applied once we had discovered which aspects in the running of an internship would affect the player’s outcome score. Before applying these models a benchmark was created as a metric to measure how well the models can predict the correlations found in our previous analysis. The F1 score was used as this would highlight the false positives so we can see what truly makes up a good performance. Stop words were also removed to get more accurate analysis of the words used. In modelling the player’s words to predict the

outcome score, an SVM model was used alongside both TF-IDF and a ‘bag-of-words’ model. By taking the contents of what was discussed within the chat, this technique had shown the importance of certain words and even behaviours to predict what outcome score they would receive (Alemerien et al., 2024). As the data was unbalanced due to the bias mentioned beforehand, aiming to prioritise the non-average performing groups too helped in a better understanding and output of the model.

To investigate factors on a group level, a logistic regression model was used to predict how outcome scores can be affected by groups focusing on specific stages in the internship. To achieve interpretation on a team level, the data was aggregated based on the implementation and the group_id. The data was also grouped by the roomName (tasks) and its wordcount as its value to show how groups focused on each task. More was to be modelled in what contributes to a good performing group through investigating their worded discussion in that team level. This modelling was achieved through combining ‘TF-IDF’ and ‘NMF’ to train a ‘Random Forest Classifier.’ TF-IDF was used alongside NMF such that the size was reduced to five ‘topics’ that represent that chat content and the Random Forest model used to predict high, medium and low scorers. XGBoost Classifier alongside feature importance was used to model the mentor's influence on player performance. Player word count and design justification and moves with the mentor was a prime focus instead of looking at it from a group level. More complex models were used such as ANN (Artificial Neural Network) to further model these non-linear relationships. Unlike the previous models that would classify performance on poor, average and high, more classes can be predicted alongside including a wider range of variables like the sentiment patterns and group themes. This model was chosen as it had the capacity to deal with high dimensional datasets with complex relations. Hyperparameter tuning was used to further improve each model’s metric compared to our benchmark. Out of all the models, SVM was found to perform the best, followed by a few of the simpler models.

Key Findings

Through our analysis, we found that there is a strong correlation between strategic mentor involvement as opposed to constant mentor involvement. We also found a relationship between high outcome scores and semantic fields used by the students, with a focus on planning and positive reinforcement being the best indicators of success for the students. There was a relationship with specific words being used being most important for success as well, with most words at the highest level of importance being words surrounding productivity and staying on task. We also found a relationship between student initiative and high outcome scores, with students scoring well when there's high involvement in the beginning stages of the internship, as well as student-led reflection having the same effect.

Conclusions

These findings indicate Nephrotex should continue to cultivate an encouraging and positive environment such that students can take the initiative in these internships as our research shows it leads to a higher performance (Golnaz Arastoopour et al., 2012). Mentors can also be graded upon their performance in future runs of the internship such that their impact can be investigated further for more optimisation. Having more initiative/involvement as a player shows understanding and Nephrotex can prioritise this skill and a players understanding through mentor interaction and tasks to enhance their final result (University of Wisconsin, n.d.).

Introduction

Background
Virtual internships is an online educational simulation where students are interns at nephrotex, a fictional biomedical engineering company. Virtual internships includes 15 implementations of *Nephrotex*, with 5 groups in each implementation designing their own device prototype to assist patients with kidney failure. The aim of virtual internships is to serve as a learning experience for the interns in communication and justifying their actions as well as building valuable experience in their profession. The interns conduct background research, design the prototypes, test and evaluate their prototypes, consult with their customers and justify their design choices, amongst other things related to designing biomedical devices. The interns work in teams through the chat tool, from which the data was collected.

Data
Data was collected over the entire period of the internship program from the chat tool used by the students. This data makes up our given data set, and includes 19,180 rows and 17 columns. The rows depict every chat utterance from each user in every group for every implementation of the program.

The columns present in the data set are below, along with a brief summary of what they represent:

Features	Description
userIDs	• Non-unique number for each player and mentor
Implementation	• Represents each enactment of the internship program, given from ‘a’ to ‘o’
Line_ID	• Unique number for the line number
ChatGroup	• Assigned group, of which there are 5 - given as: ['PRNLT', 'PMMA', 'PSF', 'PAM', 'PESPVP']. • Each of the 5 groups are present in each implementation, with different students
Content	• Messages sent in the chat
group_id	• Numerical value of ChatGroup, given from 2 to 6 (ie. 2 -> ‘PRNLT’, 3 -> ‘PMMA’, etc)
RoleName	• Given as Mentor or Player. Each group is given a mentor to help guide the students with their participation in the internship program
roomName	• Given as Mentor or Player. Each group is given a mentor to help guide the students with their participation in the internship program
m_experimental_testing	• Acts as a boolean operator

	● 1 means chat content mentions different experimental techniques to understand technical features of a design ● 0 means chat content does not include this.
m_making_design_choices	● Acts as a boolean operator ● 1 means chat content mentions making a design choice ● 0 means chat content does not include this.
m_asking_questions	● Acts as a boolean operator ● 1 means the chat asks a question ● 0 means the chat does not.
j_customer_consultants_requests	● Acts as a boolean operator ● 1 means chat content justifies a design choice by referencing customer consultant requests ● 0 means chat content does not include this.
j_performance_parameters_requirements	● Acts as a boolean operator ● 1 means chat content justifies a design choice by referencing performance parameter requirements ● 0 means chat content does not include this
j_communication	● Acts as a boolean operator ● 1 means chat content justifies a design choice by referencing communication amongst the engineers ● 0 means chat content does not include this.
OutcomeScore	● Score given to each user from 0-8 based on their final design report
wordCount	● Number of words in the message sent

Objectives and Expectations

The internship program culminates in a final design report that the students individually submit after close collaboration with their group. The students are evaluated with an outcome score for the quality of their final design report, ranging from 0-8. The question that this dataset poses essentially asks whether or not there is a correlation that links a student's outcome score with their communication amongst their peers through the chat tool. The objective is to find how discourse features relate to the students' outcome scores, using the linguistic data and numerical data to model these trends. This also includes any potential effect, for better or worse, that the mentor may have on the students.

There is an initial expectation that there will be a positive relationship between staying on task and outcome score. We expect that students who use the chat tool inefficiently by misusing the chat by sending messages that potentially misdirect the students' focus will have lower scores themselves, and potentially may detract from their teammates' scores too, as losing focus can inhibit efficiency and productivity. Hence, there is an expectation that there'd be a relationship between high concentration of chats that relate to design choices and design justifications.

Given this data set implores the analysis of chat messages sent, although categorised into different sections, doesn't paint the entire story of what is being sent in the chats. As much as analysis of these relationships between design choices/justifications and outcome score can be fruitful, it may have some inaccuracies given there isn't data on semantics used by the students in this data set. It also doesn't tell us the level of success the students reach through their messaging, or their competency in any form that would aid us in predicting outcome scores. So, there may be some potential issues modelling this without a deep dive into the content of the chats linked to each user.

Summary
In this report, we use logistic regression to map the relationship between the specific chat rooms used and outcome scores. We use a multiclass approach to find effect of mentors on the students' outcome scores, random forests to link semantic fields with student performance, a neural network system to model student interaction levels and message features in relation to outcome scores, and we employ a bag of words model onto a machine learning system that relates specific linguistic features, and their weights, to student outcome scores.

Data Quality

Missing and Placeholder Values

An initial review of the dataframe revealed three entries without the roleName assigned to the individual's chat logs. Further inspection showed that these three entries belong to the same user and with rough inspection it could be estimated they were most likely a player. However, largely due to the fact that this individual was given an outcome score of 0, along with 0 in all design moves and design justifications, it was decided that the 3 rows including this user should be removed. As it was only one user compared to the large number of entries in the dataset, it was determined that this removal would not impact the analysis made later on.

Figure 1. Missing RoleName content all belongs to the same user.

	content	userIDs
7765	Checking in	158
7771	Hey, I'm Rylee	158
7801	I completed the interview but I can't find whe...	158

Another issue with the dataset was that for each mentor, they were given an outcome score of 4, regardless of factors that may influence the quality of their mentorship within the internship. This may be because their performance was not measured and hence, to avoid missing values, they were given this outcome score as a placeholder. This however meant that in order to better understand what influences outcome scores of players, rows with mentors would be required to be removed. On top of this, to gauge the impact of mentors

later on, it was determined to be more beneficial to use the data given to create new variables for mentor analysis.

SMOTE

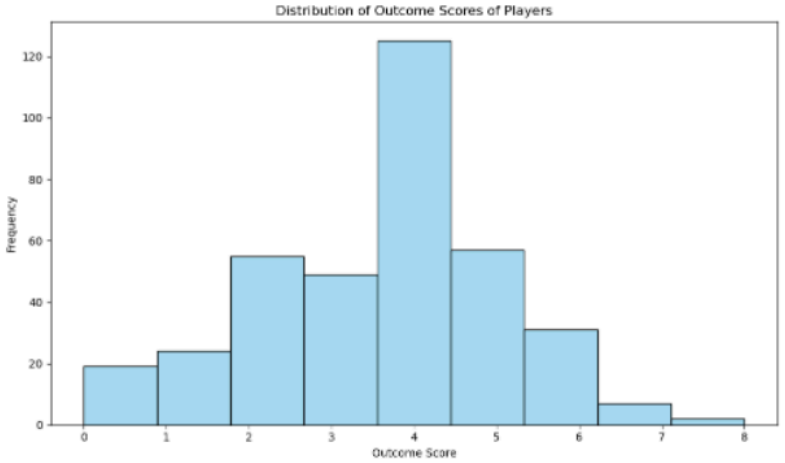

Figure 2. Distribution of Player Outcome Scores.

Figure 2 shows the distribution of Outcome Scores among players (students) after removing missing values, revealing a right-skewed pattern with a strong concentration around scores of 4, and relatively few students achieving very low (0–1) or very high (7–8) marks. This imbalance in class representation introduces a bias during model training, as the model may become disproportionately optimised for the majority class (score is 4), while underperforming on minority classes. Such bias reduces the model's ability to generalise and detect meaningful patterns in underrepresented outcomes whilst also inflating metrics such as accuracy.

To mitigate the potential bias caused by the distribution of Outcome Scores as seen in Figure 2, SMOTE (Synthetic Minority Oversampling Technique) was applied during training to synthetically balance the classes, enabling the model to learn more robust patterns across all performance levels.

From a practical standpoint, this distribution suggests that most students tend to perform around the average, with relatively few excelling or struggling significantly. We recommend that the company investigate why higher scores are so uncommon, enhancing mentor feedback, improving resource accessibility, or redesigning certain activities to better differentiate high performers. Additionally, SMOTE-adjusted models can serve as a tool to proactively identify students who may need support, based on chat behaviour patterns observed in lower-performing groups.

Data Analysis

A part of analysing the data was to investigate how specific tasks within a run of an internship can influence a group's overall performance. This would include getting the data to team level statistics by creating a new dataframe with only the player data; aggregated based on the implementation, group_id and the roomName variables with the values being the average wordCount per group in those respective tasks (roomName). This would be used to see if the word count in these tasks by the groups had contributed to their average outcome score.

To further investigate this relationship between the outcome score and their focus on specific tasks, KMeans Clustering and PCA was used. An inspection of this trend had shown major outliers (around 1000+ words in a single entry) which were removed as some were copying the instructions of their tasks into the chat to which it can be negated. When doing exploratory data analysis, one would assume that a higher word count in general would improve the overall outcome score however a clear trend was not apparent at first which prompted further investigation into why this may be; that perhaps focusing on specific tasks was a factor to consider.

This led to removing stop words, PCA and KMeans clustering had allowed to reduce the dimensionality of such a large dataset. PCA would also ensure that the variability of the data is kept alongside reducing the size as much as possible which was beneficial as the data was diverse. This also further prepared the model to demonstrate how clusters of groups would perform based on their word count within each ‘roomName’ task.

(a) (b)

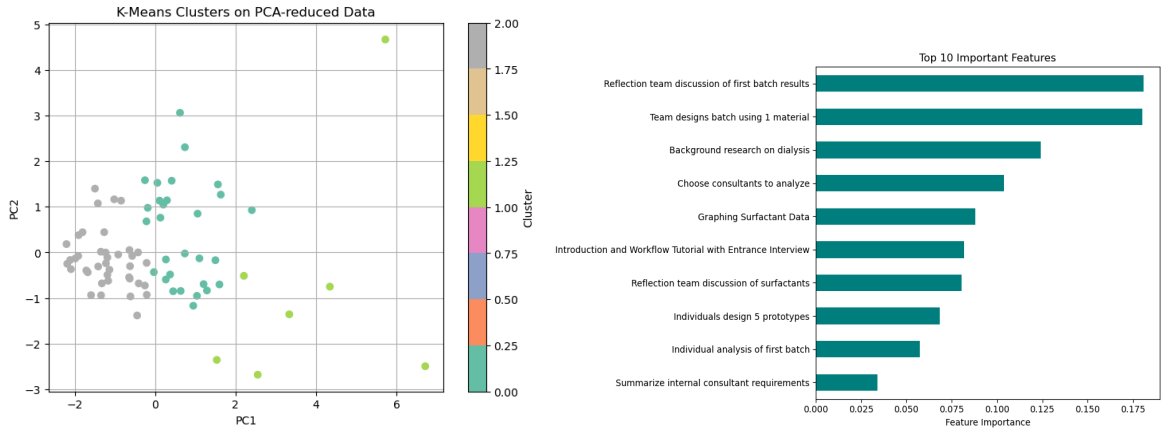

Figure 3. (a) K-Means clusters on PCA-Reduced Data. (b) Top 10 Features using PCA.

Seen above, the cluster of groups that had performed well had a higher PC1 mainly and when plotting the importance of these tasks. The main three tasks high achieving groups had focused on were “Reflection team discussion of first batch results,” “Team designs batch

using 1 material” and “Background research on dialysis.” Teams that had collaborated in a way such that they were able to reflect upon their decision making instead of solely creating a prototype or completing their tasks had exceeded in their performance. The top few tasks that had a positive effect on a group's average outcome score were centered around gaining additional information about their project which would be assumed to assist them in creating a better prototype and to know what can be said about it. This had provided insight into how prioritising different tasks could improve or decrease a group's outcome.

To explore the overall language patterns in student discussions during the visual internship, we generated a word cloud using the cleaned chat data. All messages were preprocessed to remove stopwords which allowed for more meaningful terms to stand out. The resulting word cloud showed us the most frequently used words in proportion to its size, with larger font sizes suggesting higher word frequency.

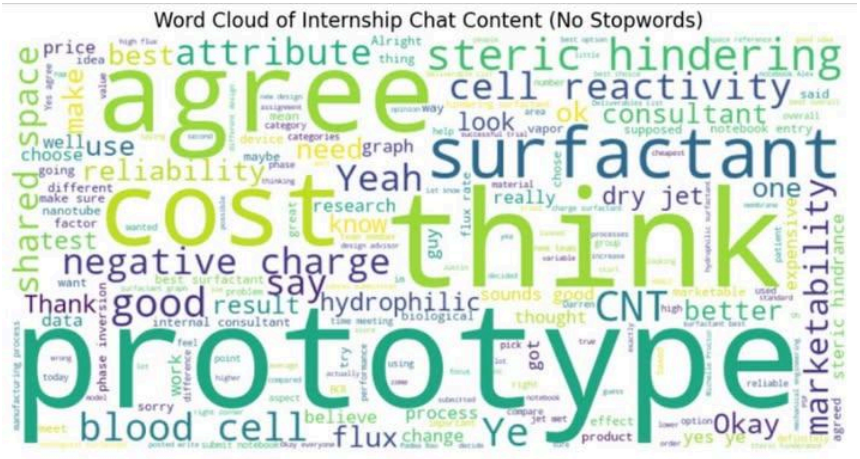

Figure 4. Word Cloud of Chat Content (no stopwords)

The word cloud revealed that terms such as “design”, “testing”, “membrane”, and “data” appeared much more frequently throughout the dataset , suggesting that students were frequently discussing key aspects of the engineering design process. This aligns with the internship's core focus on prototyping and performance evaluation. Words related to experimentation and performance criteria also appeared.

On the other hand, less visible or missing words suggested words that weren't used as often and these were often words related to stakeholder operations. These findings offered great insight into what students prioritized or discussed most frequently, and could help instructors identify gaps in their knowledge.

While this visualisation does not capture the full complexity of the internship, it served as a useful exploratory data analysis tool which guided us in the right direction for further modelling and predicting.

In order to further understand the influence of mentors in the internship, a similar method of unsupervised learning using KMeans clustering was applied, this time to engineered

features focusing on mentor statistics such as player and mentor word count, message count, and others. This would enable the exploration of underlying behavioural patterns among players in regards to their mentors. Since K-Means is a distance based algorithm the data required standardisation via StandardScalar to ensure features contributed equally to clustering. While it assumes equal variance across clusters, this is likely not reflected in the scenario of player and mentor dynamics. Regardless, it remained a suitable exploratory tool due to its simplicity and efficiency.

(a) (b)

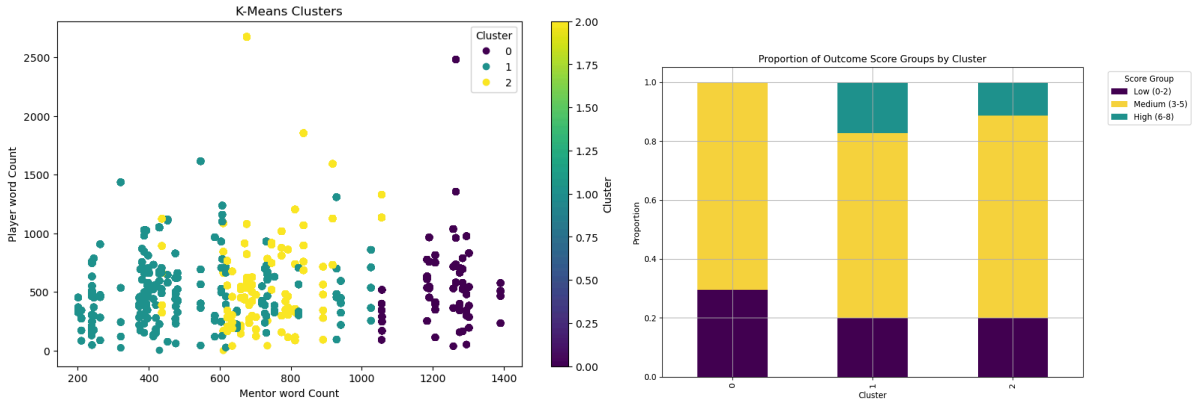

Figure 5. (a) K-Means Clustering based on word count. (b) Segmented bar plot depicting the average proportion of low, medium, and high performing players within each cluster.

The barplot in Figure 5(b) revealed how players with high outcome scores were typically associated with mentors who used fewer words, suggesting a more targeted and efficient mentoring style. These players may have been more self-directed, requiring less clarification, while still benefiting from precise, timely feedback.

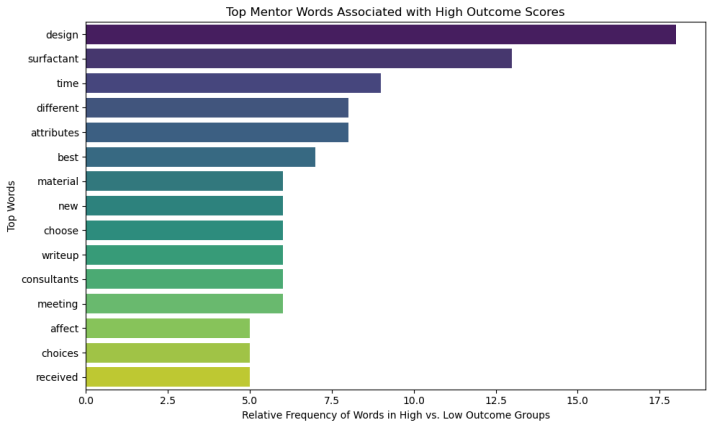

Figure 6. Top mentor words associated with high outcome scores compared to low scores.

To go beyond the quantity of mentor communication, the content of the mentor messages would depict the quality of their contribution. This prompted the use of sentiment frequency analysis techniques, where following standard preprocessing, word frequency differences were calculated. This was done by splitting player outcomes into low, medium, and high

scoring groups and then subtracting word counts in low performing interactions from those in high performing ones. The approach here helped isolate the language more strongly associated with successful players, offering another perspective on the communicative aspects of mentoring dynamics.

Model Development

Individual Student Modelling- SVM Model (base)

In the earlier stages of our modelling process, we trained a Support Vector Machine (SVM) to predict student performance (outcome score) using a standard Bag of Words representation of their chat messages. After the initial cleaning of the text by removing stopwords, we transformed each student's chat contents into a Bag Of Words matrix, where each feature represented the frequency of a specific word.

The goal remained to classify students as either high achievers (high outcome score) or low achievers (low outcome score). However, this version of the model was trained on the original, unbalanced data set where low scorers significantly outnumbered high scorers. We had not applied any balancing techniques such as SMOTE, etc. at this point in time.

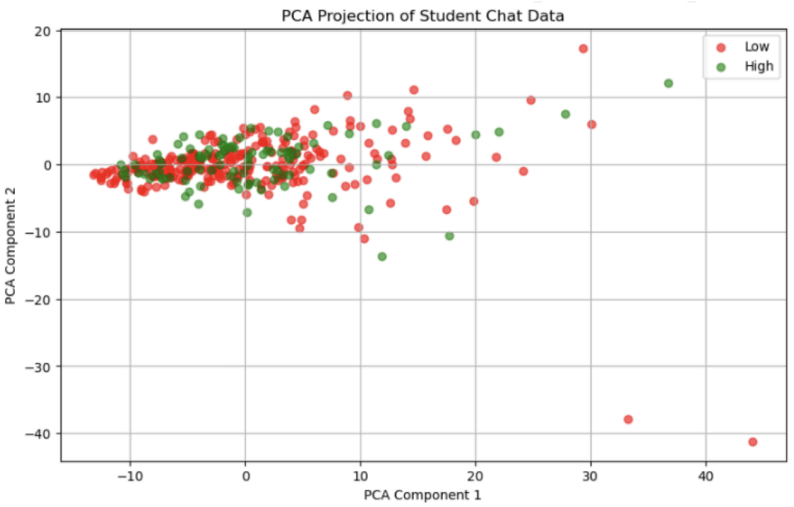

Figure 7. PCA Projection Of Base SVM Model

As a result, the model learned to heavily favor the majority class, leading to limited predictive performance. The accuracy hovered near the majority baseline, but the F1 score for the minority class (high outcome score students) was quite limited and poor. This indicated that

the model generally favoured toward predicting low performance and struggled to identify students with stronger outcomes.

Despite its simplicity, the Bag of Words model provided a useful benchmark. It highlighted the need for more intricate and complex models such as TDF-IDF and class balancing strategies such as SMOTE to enable the model to capture meaningful signals in the chat data and avoid skewed predictions.

Individual Student Modelling- SVM Model

To model individual student performances during the virtual internship, we developed a classification model based on theTF-IDF model as opposed to the Bag Of Words model we used previously. Each student's messages were processed to remove stopwords and vectorised using TF-IDF, allowing us to capture the importance of words across all participants.

The goal was to once again predict whether a student's outcome score was high or low. Since the dataset was initially unbalanced, with significantly more low scorers, we decided to unsample the minority class to ensure a balanced process.

We trained a support vector machine model once again, on the balanced dataset. While several models were explored during development, the support vector machine was chosen for its ability to handle high dimensional data such as TF-IDF. The model achieved much better and improved metrics which will be discussed more in the results and metrics section.

To better understand how the model understands the data, we applied PCA reduction to reduce the high-dimensional TF-IDF data into two dimensions. The visualisation showed that high and low scoring students were mixed together in the graph. This means their chat messages weren't different enough for the model to separate them easily, which helps explain why the model didn't perform very well.

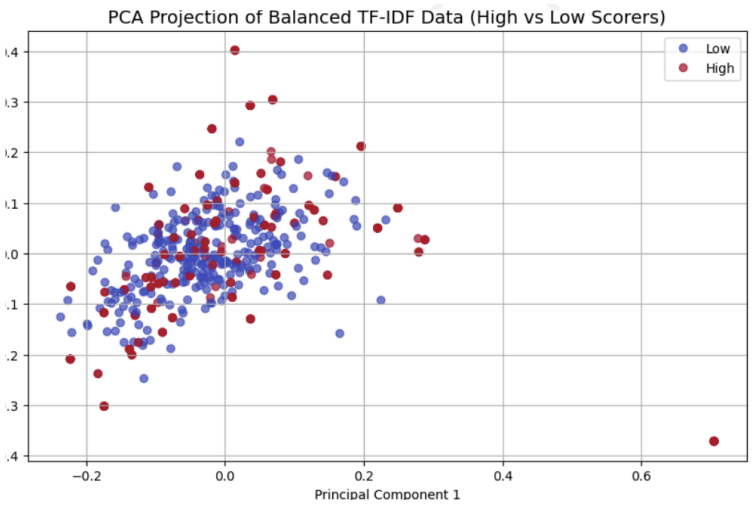

Figure 8. PCA Projection Of Improved SVM Model

This modelling approach offered insight into how student chat conversations during the internship related to their individual performance (outcome score), and highlighted the importance of NLP (Natural Language Processing) tools.

Group Tasks Modelling - Logistic Regression

As mentioned in the exploratory analysis section the data containing the aggregated player data had PCA applied to for analysis in reducing dimensionality and keeping variability. This PCA data had been created as a separate dataframe to feed as input for a classification model. This model would be used to predict a group's outcome score based on how group's prioritised different tasks in their respective internship. Having the raw information being reduced to another dataframe of PCA data had significantly benefited the runtime of the code and the high dimensionality. A model suitable for classification was recommended for this relation which included experimenting with models such as SVM, logistic regression and random forests. The random forests model was chosen instead of decision trees as the data was still susceptible to bias due to the overwhelming ratio of average performing groups to distinctly higher performing groups. The use of random forests was made to prevent the overfitting that could have been caused if the decision tree model was used; the random forests model takes the average of '*multiple trees*' which improves prediction *(Pandya et al., 2023)*. Additionally, logistic regression is a model that can handle high dimensionality with a larger number of features, especially categorical variables, combining this model with the PCA was presumed to increase the performance of the model since the data would reduce in dimensionality.

Things to consider when modelling was the bias towards average performing groups and the variability due to having a large number of tasks to consider that was reduced. Even though PCA was applied before modelling, some interpretability will still be lost.

Mentor Modelling - XGBoost

To build on prior EDA and investigate how mentor and player communication metrics predict performance, an XGBoost classifier was implemented. This method was chosen as a natural progression from decision tree models due to its ability to build on a sequence of trees at each iteration to optimise prediction through boosting while maintaining interpretability through feature importance metrics (Ramraj et al., 2016). Framed as a multiclass classification problem, XGBoost was used to predict player outcome groups (Low, Medium, High) as it is suitable for handling non-linear relationships, complex interactions between features, and imbalanced class distributions (Lai et al., 2021), all of which are relevant characteristics of the dataset.

The classifier was trained on the transformed dataframe including mentor statistics where each row represented an individual player, and categorical variables such as ChatGroup and Implementation were deliberately excluded to avoid introducing noise or patterns that are not generalisable. This structure allowed for aggregation of both original numerical features and engineered metrics, including mentor and player word counts, design related moves, and design justifications, all providing as a summary for each individual player. This enabled the model to focus on player success based on behavioural and communicative indicators instead of group level.

Hyperparameter tuning was performed to optimise model performance, including adjustment of parameters such as learning rate, maximum tree depth, and number of estimators, while preprocessing steps ensured that class imbalance was addressed.

To gain a better understanding of each feature's contribution to the model, focus was shifted towards feature importance metrics, where ‘weight’ represents the features used to split the data most often and ‘gain’ refers to the features that improve the model most when used. Applying both provided a better understanding of the influence of various features on the model’s decisions.

ChatGroup modelling - TF-IDF & NMF

First, we aggregated individual messages to the team level by grouping on ChatGroup × implementation. For each group, we concatenated all “Player” messages’ content and computed aggregate variables: six binary behavior-flag proportions (e.g., fraction of posts that ask questions), total wordCount, and raw counts for each implementation. We then performed textual feature extraction via a two-step pipeline: (1) TF-IDF vectorization of the aggregated content (top 1,000 tokens after removing English stop-words), and (2) Non-Negative Matrix Factorization (NMF) to reduce TF-IDF dimensions into five latent topics. Each topic component captures coherent semantic themes such as alignment language (Topic 4) or positive feedback loops (Topic 5). We merged these five topic-intensity scores with the six binary flags, wordCount, one-hot encodings for ChatGroup, and one-hot encodings for implementation into a final feature matrix.

We trained a Random Forest classifier on this combined feature set, using 200 trees and setting class_weight='balanced' to address class-imbalance across low, median, and high score categories. Random Forest was chosen because it handles nonlinear interactions between features without extensive preprocessing and is robust to noise in high-dimensional data. We tuned hyperparameters such as the number of NMF topics (five), TF-IDF vocabulary size (1,000), number of trees (200), and maximum tree depth via manual cross-validation on a 70/30 train/test split. Key assumptions include that five topics sufficiently capture the dominant semantic variance in chat and that binning scores into three categories retains essential performance distinctions. Limitations include potential oversimplification of discourse nuances by compressing text into only five topics and sensitivity of tree-based models to hyperparameter choices.

Results

Individual Student Modelling - SVM (Improved)

The improved support vector machine model was trained using a more refined principle. Instead of relying on raw word counts, this version used TF-IDF vectorisation, which gives a higher weightage to specific words and reduces the influence of common ones. Additionally, the original dataset was unbalanced, with significantly more low-scoring students than

high-scoring ones. To address this, we applied random upsampling to the minority class to ensure both classes were equally represented during training.

After retraining the SVM model on the balanced dataset, we observed a considerable improvement in performance across all of the key metrics. The model achieved an overall accuracy of 82%, a notable increase compared to the baseline model's 57%. More importantly, the precision, recall, and F1 scores for both classes were well balanced.

These results show that the model is now able to identify both high and low performers with almost equal effectiveness. The F1 scores are above 80% for both classes, indicating strong overall reliability. This balance suggests that the combination of the TF-IDF model and the class rebalancing successfully corrected the bias observed in the base SVM model, in which the high performers were often misclassified.

Balanced SVM Accuracy: 0.82

	precision	recall	f1-score	support
High	0.81	0.81	0.81	52
Low	0.82	0.82	0.82	57
accuracy			0.82	109
macro avg	0.82	0.82	0.82	109
weighted avg	0.82	0.82	0.82	109

Figure 9. Accuracy Metrics Of Improved SVM Model

Individual Student Modelling - SVM (Base)

The baseline support vector machine model was trained on the bag of words models without any class rebalancing. This means the model was trained on the original dataset, where low scoring students were far more common than high scoring students. As a result, the model learned to favour the majority class.

The model achieved an accuracy of approximately 59%, which may initially seem acceptable. However, because the dataset was unbalanced, this number is quite misleading. A model that simply predicts ‘low’ for every student would still score well as the baseline did. Therefore, we looked at more informative metrics such as precision, recall, and F1 score for both classes.

These results show that the model performed well at identifying low scorers, but performed poorly at identifying high scorers, the class we often care about the most in education settings. The F1 score of just 25% for high performers indicates that the model rarely predicted ‘high’ correctly, and when it did, it often got it wrong.

This unbalance in prediction suggests that the model learned to prioritise recall for the dominant class at the expense of minority class performance. These results highlighted the need for TF-IDF and class balancing methods such as SMOTE to improve fairness and effectiveness in prediction.

SVM Accuracy: 0.59

	precision	recall	f1-score	support
High	0.26	0.24	0.25	21
Low	0.71	0.74	0.72	53
accuracy			0.59	74
macro avg	0.49	0.49	0.49	74
weighted avg	0.58	0.59	0.59	74

Figure 10. Accuracy Metrics Of Base SVM Model

Group Task Modelling - Logistic Regression

When predicting a group's average outcome score through their word count in each task of the internship, different models were compared before tuning it more. The logistic regression model had performed considerably well compared to the random forests and SVM model which had an accuracy score of less than random, however this was done with and without PCA. As logistic regression was the only model to have shown metrics better than random prediction, we had used it to predict for multiclass classification; subpar performing groups, average performing groups and high performing groups based on their mean outcome score.

(a)

```
[[1 2 1]
 [5 2 0]
 [2 1 1]]
```

	precision	recall	f1-score	support
0	0.12	0.25	0.17	4
1	0.40	0.29	0.33	7
2	0.50	0.25	0.33	4
accuracy			0.27	15
macro avg	0.34	0.26	0.28	15
weighted avg	0.35	0.27	0.29	15

(b)

```
[[6 3]
 [3 3]]
```

	precision	recall	f1-score	support
0	0.67	0.67	0.67	9
1	0.50	0.50	0.50	6
accuracy			0.60	15
macro avg	0.58	0.58	0.58	15
weighted avg	0.60	0.60	0.60	15

F1 Score (minority class): 0.5

Figure 11. (a) Output with multiclass and before optimisation. (b) output of model with binary classification and optimisations (both with PCA data).

When running binary classification for this model (average performing groups and high performing groups), applying SMOTE, stratified sampling and balanced weight had improved the F1 score and accuracy significantly in reducing bias of the average performing groups as the minority group can also be predicted more.

Mentor Modelling - XGBoost

The XGBoost classifier was trained to predict player performance categories (Low, Medium, High) based on a range of design and behavioural features aggregated at the player level. After hyperparameter tuning, the model achieved an overall accuracy score of 0.58.

Classification Report:

	precision	recall	f1-score	support
High	0.44	0.40	0.42	10
Low	0.28	0.22	0.24	23
Medium	0.69	0.76	0.72	58
accuracy			0.58	91
macro avg	0.47	0.46	0.46	91
weighted avg	0.56	0.58	0.57	91

Figure 12. XGBoost model classification report of player outcomes across Low, Medium, and High categories.

The medium group showed the strongest performance, while the low and high groups had lower precision and recall values.

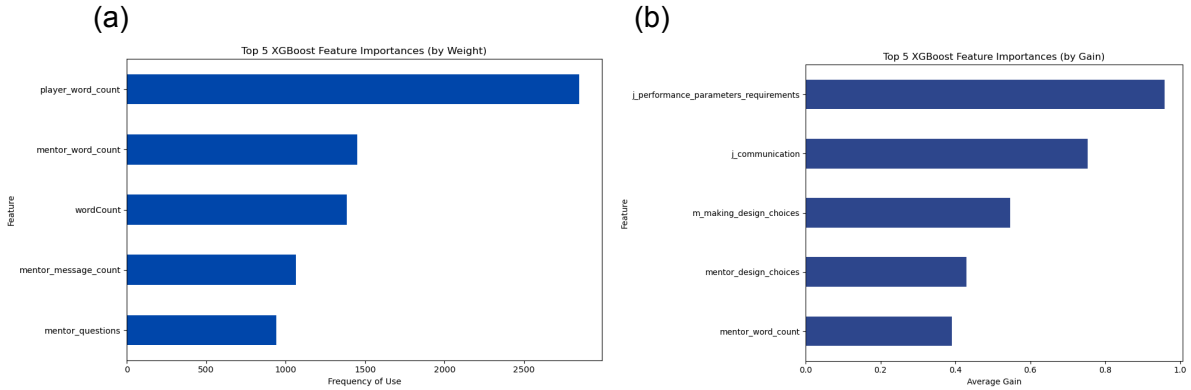

Figure 13. (a) Top features most frequently used to split data across decision trees in XGBoost model. (b) Features that contributed most to improving classification performance in XGBoost model

To assess feature influence on the model's decisions, feature importance was visualised above via weight and gain. Here, weight highlights mentor and player word counts as frequently used features. In terms of gain, it is variables related to complex design justifications and mentor involvement that contribute most to performance improvements.

The XGBoost model revealed several key patterns linking mentor and player communication to performance outcomes. The model performed best in predicting the medium outcome group, and even though class imbalance was addressed, it may still be affecting model performance, however it also suggests that players in this category exhibited the most consistent and predictable behavioural patterns. In contrast, high and especially low outcome groups may have involved more variable behaviours that are context dependent, and so this could be harder for the model to classify accurately. This is something that can therefore be investigated in more detail in the future to better understand the intricacy of the model.

Importantly, the results support the idea that strategic mentor input, rather than excessive amounts of communication, is more strongly associated with successful player outcomes.

Player success is suggested to be linked with focused discussions around core design and scientific decisions, often when paired with concise and deliberate mentor involvement.

ChatGroup modelling - TF-IDF & NMF

On the PRNLT subset alone, our topic-augmented Random Forest achieved a macro F_1 of 0.68 and 72 % accuracy on a 70/30 train/test split. The confusion matrix (low, median, high) revealed that 70 % of high-score teams were correctly identified, up from roughly 45 % using behavior flags only. Median-score detection also improved markedly ($F_1 \approx 0.73$), while low-score recall increased to 0.40. These gains demonstrate that adding semantic topic features (via NMF) and wordCount captured the nuanced language patterns that distinguish strong teams.

```
=== Random Forest with Topic Features of PRNLT ===
Macro F1 score: 0.681

Confusion Matrix:
             pred_low  pred_median  pred_high
true_low          159          154          8
true_median        93          572         19
true_high          17           11         64

Classification Report:
              precision    recall  f1-score   support

         low       0.59      0.50      0.54       321
      median       0.78      0.84      0.81       684
        high       0.70      0.70      0.70        92

    accuracy                           0.72      1097
   macro avg       0.69      0.68      0.68      1097
weighted avg       0.72      0.72      0.72      1097
```

```
=== Random Forest (Unified Model) ===
Macro F1 score: 0.593

Confusion Matrix:
             pred_low  pred_median  pred_high
true_low          529          564         64
true_median       306         2827        171
true_high          38          326        247

Classification Report:
              precision    recall  f1-score   support

         low       0.61      0.46      0.52      1157
      median       0.76      0.86      0.81      3304
        high       0.51      0.40      0.45       611

    accuracy                           0.71      5072
   macro avg       0.63      0.57      0.59      5072
weighted avg       0.70      0.71      0.70      5072
```

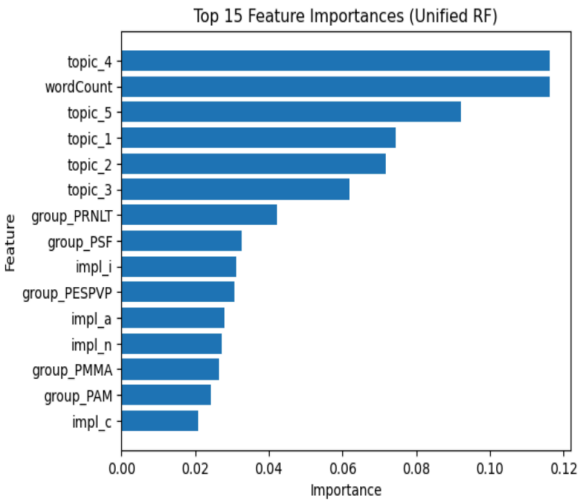

Figure 14. (a) Random Forest via PRNTL. (b) Random Forest Unified Model. (c) Top feature Importances.

Applying the same pipeline across all five ChatGroups in a single Random Forest yielded a macro F_1 of 0.59 and 70 % accuracy. In this unified model, the top feature importances were

overwhelmingly semantic: topic_4 (alignment) and topic_5 (positive feedback loops), followed by “wordCount”.ChatGroup indicators, which are group_PSF, group_PRNLT, also ranked highly, reflecting group-specific grading nuances. Overall, these results confirm that semantic topic features and message volume robustly predict team report outcomes which are far more so than different features both within individual groups and across the entire dataset.

Artificial Neural Network

Taking a break from simple models, we took a deep dive into the world of deep learning, implementing an Artificial Neural Network using Tensorflow to address the classification task. For clarity, an artificial neural network or ANN is a type of machine learning model inspired by the way the human brain processes information. It consists of layers of interconnected nodes ("neurons") that learn patterns in data by adjusting weights through training. ANNs are particularly effective for handling large datasets with complex relationships, as they can model both linear and non-linear interactions between variables. The ANN was selected due to its ability to model complex, non-linear relationships within large datasets. Given the dataset’s size and variety of features including textual sentiment and categorical groupings, ANNs were considered suitable over traditional linear models which may struggle to capture high-dimensional, abstract interactions (Alwosheel, 2018).

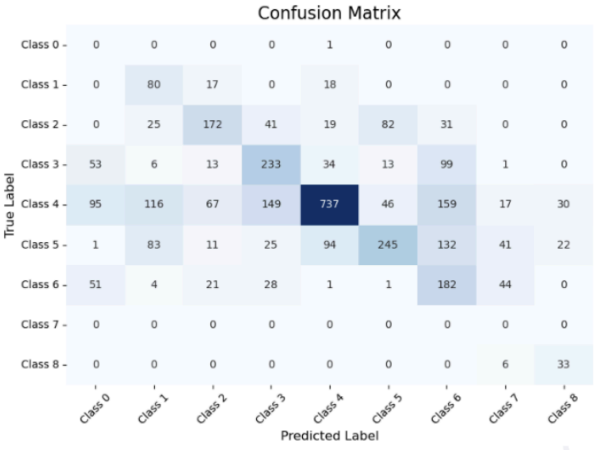

Figure 15. Confusion Matrix.

Our Artificial Neural Network achieved an accuracy of 49.8%, indicating that the model correctly predicted the class label in just under half of all test cases. While this may seem modest, it is important to note that the classification task involved nine possible outcome classes and an imbalanced class distribution, making the problem inherently more difficult (Yang, 2024) whilst also considering the benchmark for randomness was 0.11 for precision, accuracy, recall and f1-score. The precision of the model was 0.384, suggesting that when the model predicted a particular class, it was correct approximately 38.4% of the time. The recall score of 0.441 indicates that the model successfully identified 44.1% of all relevant instances from the true labels. This trade-off between precision and recall resulted in an F1 score of 0.379, reflecting the model’s overall balance between identifying relevant classes

and avoiding false positives. Moreover, figure 15, emphasises the balance in predictions because of SMOTE. These metrics highlight that while the model captures some meaningful patterns in the data, there is substantial room for improvement, particularly in boosting precision and overall predictive power, possibly through further hyperparameter tuning or improved feature selection.

Hidden Layers	2
Activation function (Input and Hidden Layers)	RELU
Activation function (Output layer)	Softmax
L1 regularization	0.001
Learning Rate	0.01
Neuron structure	64-32-32-9
Optimizer	Adam
Loss	categorical cross-entropy
n_splits (GroupKFold)	10

Figure 16. ANN Architecture.

To prepare the dataset for model training, several feature engineering steps were applied. Categorical variables, including chatgroup and roomname, were transformed using one-hot encoding to ensure compatibility with the neural network model. A sentiment score was also generated for each message using the VADER sentiment analysis tool, converting textual content into a numerical representation of emotional tone. To address class imbalance, we applied SMOTE (Synthetic Minority Oversampling Technique) on the training set, which synthetically increased the representation of minority classes. Finally, all features were standardised using StandardScaler to normalise input values, which improves training stability and convergence in neural networks. As shown in Figure 16, our Artificial Neural Network was structured with two hidden layers, using a neuron architecture of 64-32-32-9, where the final layer corresponds to the nine output classes. The model employed the ReLU activation function in the input and hidden layers, and a softmax activation function in the output layer for multi-class classification. To determine the most effective configuration, we experimented with all activation functions available in the TensorFlow documentation (Module: tf.keras.activations, n.d.), and found ReLU to offer the most stable performance with the fastest convergence.

Several assumptions underpinned our modelling approach. We assumed that the VADER-derived sentiment scores meaningfully reflected user tone and had predictive value. We also assumed that SMOTE-generated samples were sufficiently realistic to support

generalisable learning. Lastly, we relied on feature standardisation to ensure stable and efficient training across all input features.

Figure 17. An illustration of validation loss versus training loss for the ANN trained using groupKFold cross validation.

Despite our efforts to mitigate risks, the model has several limitations. ANNs are prone to overfitting, particularly when using synthetic data as can be seen in Figure 17 due to the validation loss being much less than the training loss. They are also less interpretable, which can make it difficult to explain individual predictions to non-technical stakeholders. Additionally, the model was computationally expensive to train, especially given the use of 10-fold cross-validation, which limited our ability to test alternative architectures or tuning strategies. Finally, there is a risk that the model may reproduce biases present in the data, such as those arising from sentiment misclassification.

This project applied an Artificial Neural Network to predict student performance in the Nephrotex virtual internship using features extracted from team chat data. The model achieved an accuracy of 49.8%, F1 score of 0.379, and moderate recall and precision, indicating limited but meaningful predictive power in a complex, imbalanced classification task. Despite challenges like class imbalance and potential overfitting, the results suggest that features such as sentiment, question-asking, and design justifications are crucial to students' final report outcomes. With further tuning and refinement, this approach could contribute to real-time feedback systems in virtual learning environments.

Conclusions

Pairing the XGBoost model with initial EDA, findings reinforce the value of student-led problem solving guided by strategic mentorship, rather than abundant intervention. Effective mentor communication appears to involve empowering players to engage deeply with design and project content while providing guidance that is relevant but minimal. Alongside the wide range of machine learning approaches explored, complex models such as Artificial Neural Networks (ANNs) proved effective at an interface level, capturing subtle utterance patterns and tonal cues within the chat data. The ANN was designed to predict nine different Outcome Score classes, unlike simpler models targeting only two or three classes, which

inherently lowers the expected benchmark for performance (around 0.11 for each of the four main metrics). Although it is challenging to directly compare such an extensive model with simpler alternatives due to differing classification tasks, the ANN showed solid performance relative to the dataset's condition and the challenges of modelling unstructured team communication. From a broader perspective, this reflects the complexity and richness of the Nephrotex virtual internship, where nuanced peer collaboration and engineering discourse play a key role in shaping student outcomes. However, simpler models like SVM exceeded the randomness benchmark by a larger margin than the ANN, and for this reason, we concluded that SVM was the most effective overall model.

In conclusion, we found that student planned and student led decisions which are guided by effective mentor support help drive higher student performance in the Nephrotex virtual internships program. It was found that when students afford themselves more time to plan decisions, reflect upon their decisions and designs, and take initiative in the early stages of the internship, these students earned higher outcome scores than students who did not. The TF-IDF model shows this through its feature importance illustrating that sentimental language revolving around planning was the top indicator of student performance. The logistic regression model also proved this through its feature importance, demonstrating that team reflection and background research were in the top 3 most important factors of that model in determining outcome score.

Mentor input was found to have an inverse effect on student performance, through the analysis of mentor word count. It was shown in the XGBoost model that mentors with higher word count often attributed to lower student outcome scores, and lower mentor input correlated to higher student outcome scores. This implies that an overreliance on mentors to support the students ended up not having the desired effect in boosting student performance. This also links back to the point made previously, that students who didn’t take initiative, to the point where mentors had to overstep, earned lower outcome scores, which further shows that increased mentor involvement doesn’t increase student performance

It was also found that linguistic features and sentiment analysis proved effective, and showed a relationship between particular word usage and semantic field produced an effect on outcome scores. It was found that positive feedback loops and alignment/planning language had a large importance in the TF-IDF model, showing that positivity in the groups and a higher level of focus were associated with higher outcome scores. Also, the findings of top words associated with higher student outcome scores were related to the tasks involved, suggesting that when students and mentors were focussed and on task, they were able to perform better than students and mentors who weren’t more effective and efficient with their time. This suggests that a high level of efficiency was required in the virtual internships program, and that students who were more effective with their time were able to produce better results.

Overall, we found that student initiative, effective use of the chat tool (by students and mentors), and other sentimental features all had great effects on the outcome score of students. Our recommendation to the mentors and managers running the virtual internships is that an importance on promoting student-led decision making and initiative should be reflected by all running the program, as this yielded great results for the students. Reflection,

planning and participation by all students should also be heralded and encouraged, as a substantial amount of evidence proves this greatly influences outcome score. We'd also suggest that mentors should only use the chat tool when absolutely necessary, as we found that an over use of the chat tool by mentors actually inhibited student performance. Ultimately, mentors and managers running the virtual internships should aim to:

- Promote student-led decision making, initiative, planning, reflection and participation from all students
- Create a positive environment for students to prosper
- Involve mentors in conversation when absolutely necessary, to allow students to work through issues themselves, instead of immediate mentor interjection

References

Alemerien, K., Al-Ghareeb, A., & Alksasbeh, M. Z. (2024). Sentiment Analysis of Online Reviews: A Machine Learning Based Approach with TF-IDF Vectorization. *Journal of Mobile Multimedia*, 1089–1116.
https://doi.org/10.13052/jmm1550-4646.2055

Alwosheel, A., van Cranenburgh, S. and Chorus, C.G. (2018). Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. *Journal of Choice Modelling*, 28, pp.167–182.
doi:https://doi.org/10.1016/j.jocm.2018.07.002.

Golnaz Arastoopour, Chesler, N. C., D'Angelo, C. M., Shaffer, D. W., Opgenorth, J. W., Reardan, C. B., Haggerty, N. P., & Lepak, C. G. (2012). Nephrotex: Measuring First-year Students' Ways of Professional Thinking in a Virtual Internship. *Asee.org*, 25.971.1–25.971.15.
https://peer.asee.org/nephrotex-measuring-first-year-students-ways-of-professional-thinking-in-a-virtual-internship

Lai, S. B. S., Shahri, N. H. N. B. M., Mohamad, M. B., Rahman, H. A. B. A., & Rambli, A. B. (2021). Comparing the performance of AdaBoost, XGBoost, and logistic regression for imbalanced data. *Mathematics and Statistics*, *9*(3), 379-385.
doi:10.13189/ms.2021.090320

Pandya, A., Nanavaty, R., Pipariya, K., & Shah, M. (2023). A Comparative and Systematic Study of Machine Learning (ML) Approaches for Particulate Matter (PM) Prediction. *Archives of Computational Methods in Engineering*, *31*(2), 595–614.
https://doi.org/10.1007/s11831-023-09994-x

Ramraj, S., Uzir, N., Sunil, R., & Banerjee, S. (2016). Experimenting XGBoost algorithm for prediction and classification of different datasets. *International Journal of Control Theory and Applications*, *9*(40), 651-662.

TensorFlow. (n.d.). *Module: tf.keras.activations | TensorFlow Core v2.4.1*. [online] Available at:
https://www.tensorflow.org/api_docs/python/tf/keras/activations.

University of Wisconsin. (n.d.). Collaborating in a Virtual Engineering Internship.
https://repository.isls.org/bitstream/1/2511/1/626-630.pdf

Yang, Y., Hadi Akbarzadeh Khorshidi and Uwe Aickelin (2024). A review on over-sampling techniques in classification of multi-class imbalanced datasets: insights for medical problems. *Frontiers in Digital Health*, 6.
doi:https://doi.org/10.3389/fdgth.2024.1430245.

Zhu, C., Idemudia, C. U., & Feng, W. (2019). Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques. *Informatics in Medicine Unlocked*, *17*, 100179.
https://doi.org/10.1016/j.imu.2019.100179

(Title image source) *Top 25 Team communication apps for businesses in 2023*. (2021, January 28).
https://clariti.app/article/best-team-communication-apps/

Appendix

```
from imblearn.over_sampling import SMOTE
from sklearn.preprocessing import StandardScaler
from tensorflow.keras import regularizers
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.layers import Dense, Dropout



gkf = GroupKFold(n_splits=10)
groups = player_df['userIDs'].values
histories = []
for fold, (train_idx, val_idx) in enumerate(gkf.split(X, y, groups=groups)):
    print(f"\n Training Fold {fold + 1}")
    X_train, X_val = X[train_idx], X[val_idx]
    y_train, y_val = y[train_idx], y[val_idx]


    smote = SMOTE(random_state=42)
    X_train, y_train = smote.fit_resample(X_train, y_train)
```

```
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_val = scaler.transform(X_val)


    y_train = to_categorical(y_train, num_classes=9)
    y_val = to_categorical(y_val, num_classes=9)

    ann = tf.keras.models.Sequential()
    ann.add(tf.keras.layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],),
                            kernel_regularizer=regularizers.l2(0.001)))
    #tf.keras.layers.Dropout(0.2)
    ann.add(tf.keras.layers.Dense(32, activation='relu',
kernel_regularizer=regularizers.l2(0.001)))
    #tf.keras.layers.Dropout(0.2)
    ann.add(tf.keras.layers.Dense(32, activation='relu',
kernel_regularizer=regularizers.l2(0.001)))
    #tf.keras.layers.Dropout(0.2)
    ann.add(tf.keras.layers.Dense(9, activation='softmax'))

    opt = tf.keras.optimizers.Adam(learning_rate=0.01)
    ann.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])

    early_stopping = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)

    history = ann.fit(X_train, y_train,
                      validation_data=(X_val, y_val),
                      batch_size=64, epochs=100,
                      callbacks=[early_stopping])
    histories.append(history)
```