**Virtual Internship**



**Group members:**

Lingyi Zhang - 31113141

Wen Wen - 31514081

Shanrui Huang - 32445741

**Contents**

[**Executive summary**](#_7j2wxh2713d3) **2**

[**Introduction**](#_uu7rqwgdgrqz) **3**

[**Data Quality**](#_h9dc8n519g9t) **4**

[A preliminary tour of the dataset](#_fqhjphw2m476) 4

[Data clean](#_c2aat48p5070) 5

[Groupby](#_eym6ga33dplz) 5

[**Classifier using TF-IDF**](#_g45o3qoigahd) **6**

[TF-IDF](#_o4ipmsr514bs) 7

[**Weight table**](#_ejgicbnrcqub) **11**

[**Prediction Model using Logistic regression**](#_eesqcr3drdt2) **12**

[Improve Accuracy](#_u9k8l189m3ub) 15

[**Conclusion**](#_fgck4tfy9i8v) **18**

[**Appendix**](#_4s48bpiwo3ba) **19**

# **Executive summary**

The primary goal of this report is to forecast the final score based on the performance of the participants throughout the virtual internship. A virtual internship is a learning simulation in which students work as interns in a fictitious firm. The virtual internship students in this study are working in a biomedical engineering firm to create a prototype device for renal failure patients. Data such as member interactions, experimental experiments, and team choices will be captured during the teamwork process. This is the dataset that will be utilised in our group project.

We employed two methodologies in the study to determine the association between student performance and final score. On the one hand, tf-idf is used to investigate the connection between conversation content and final score. tf-idf is a method for determining the significance of words or phrases in a paragraph. We can obtain the relationship between the character data of the conversation content and the outcome score using tf-idf. On the other hand, logistic regression was employed to evaluate the impact of various factors other than conversation content on the outcome score. The goal of logistic regression is to create a prediction model using partial data in order to predict the final result using various factors.

We have had some success with the two ways described here. While our prediction model achieved 75% accuracy, we also obtained a chat content weight table, which illustrates the weight of keywords or phrases in different final scores. We can plainly observe the keywords in the conversation records of high-scoring pupils. This implies that we use the model to anticipate students' scores, examine their communication based on the weight table, and give good recommendations to develop their thinking and action for a certain job.

# **Introduction**

* **Background**

Virtual internships are educational simulations in which students act as interns in a fictitious firm. This Virtual internship comes from a biomedical engineering firm. Participants in the internship worked in groups to develop a prototype gadget to assist patients with renal failure. Conducting background research, understanding stakeholder needs, designing prototypes, testing and assessing prototypes, and justifying design decisions are among their roles.

* **Dataset**

Conversation records from five groups in the virtual internship programme are included in our dataset, which includes chat content as well as the speaker's "userID," "group id," "role name," and other information. Also, whether there are any essential notions in engineering discourse that are related to specific design actions and design justifications in the content. The final grades of students during their internships are also included in our dataset.

* **Goals, Problems and Solutions**

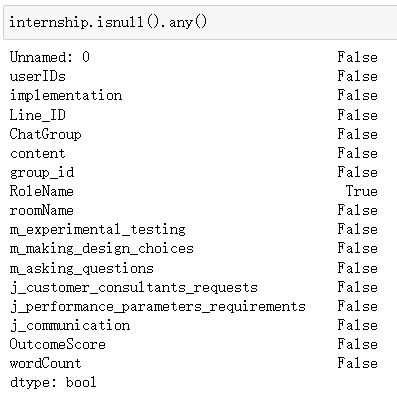
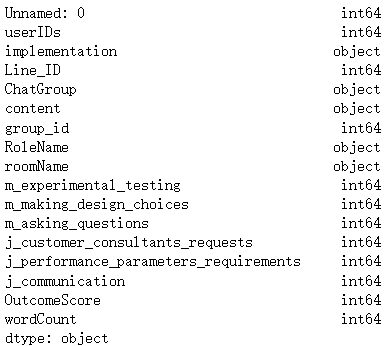
The purpose of this project is to use data from what the students talked throughout the internship to model performance on the report. The issue we have after modelling is that the prediction model's accuracy is quite poor. To address this problem, we first combine the data using "userIDs," then balance the "Pass" and "Fail" data, and last utilise Ada Boosting to identify the optimum estimator and learning rate. However, it can only enhance accuracy to 75% in the end.

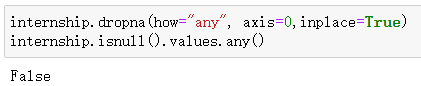
* **Contributions**
  + **Shanrui Huang**
    - Shanrui is primarily in charge of the data clean and groupby sections, as well as preparing for the subsequent TF-IDF and prediction model formation by constructing new binarized target columns and eliminating redundant invalid fields.
  + **Wen Wen**
    - Wen is mainly responsible for applying TF-IDF to calculate the link between player "content" and "OutcomeScore". Determine whether each content's score falls into the “Pass” or “Fail” category.
  + **Lingyi Zhang**
    - Lingyi is mainly responsible for establishing the weight table of content the players said and outcome score and the prediction model of different features and outcome score in virtual internship by using logistic regression, and improving the accuracy of the prediction model through balance and Ada Boosting, so as to obtain a highly accurate and efficient model.

# **Data Quality**

## **A preliminary tour of the dataset**

The data set comprises 17 columns and 19180 rows, according to preliminary browsing. There are numerous columns in the data set that pertain to user information. These columns will aid in the categorisation of the data points. The rest is required to construct our predictive model. At the same time, the field "RoleName'' has empty values. These issues must be addressed in the data clean and groupby steps that follow.

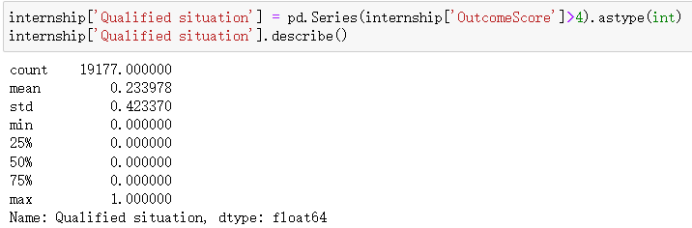


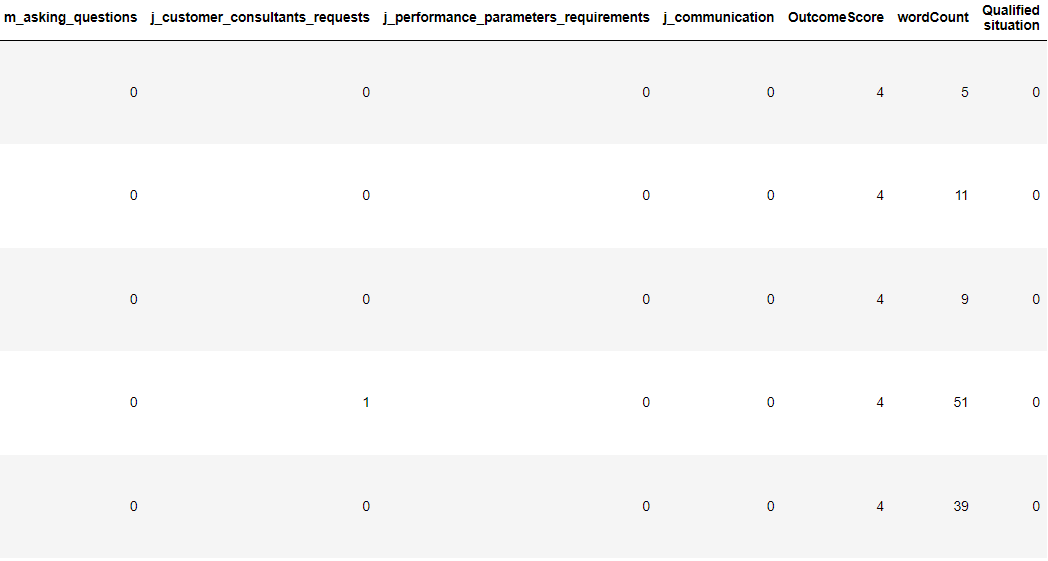
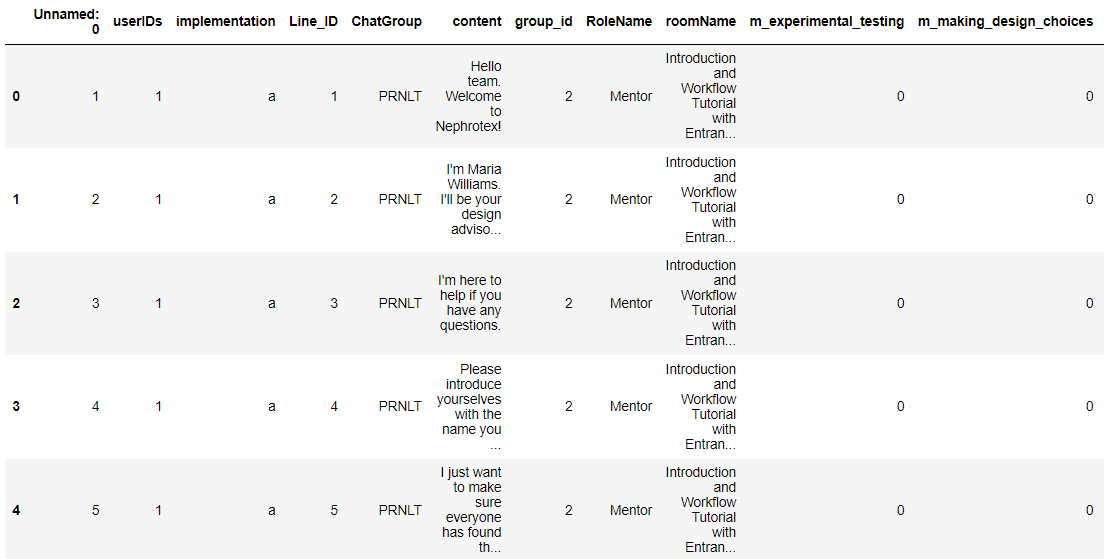


## **Data clean**

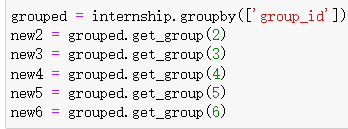
To remove the missing values, we just need to let df.dropna() traverse all of the rows, then identify the missing values again, and now it returns "False".

The following step is to combine the data. Our DataFrame's crucial column "OutcomeScore" is separated into nine outcomes ranging from zero to eight, which is quite unfavourable for our data analysis and subsequent logistic regression, thus we must address this issue during data clean. Our approach is to introduce a new column called 'Qualified Situation,' which classifies data values in the column "OutcomeScore" as 0 if they are less than or equal to 4, and 1 if they are more than 4. As a result, we receive a new column with a score of 0 for the high score and a score of 1 for the low score.

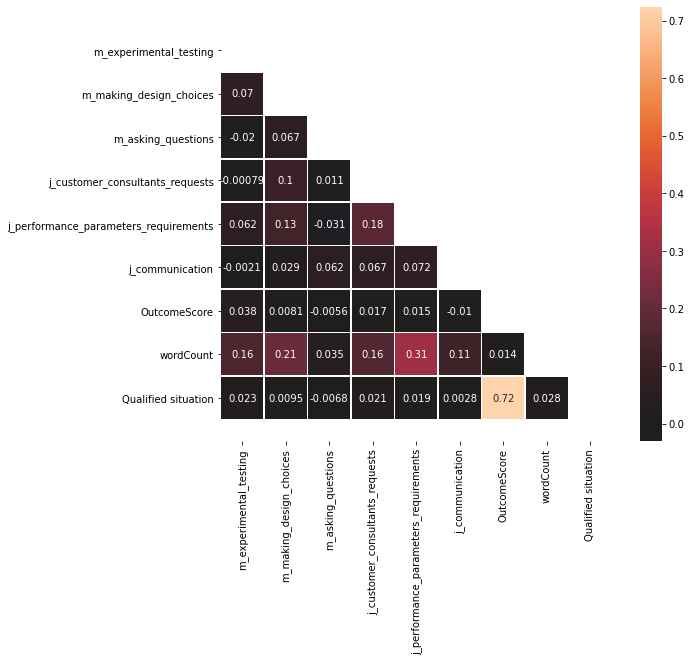


Now our dataset looks like this:

## **Groupby**

To make it easier to apply logistic regression to develop a prediction model later, we group the data by "group id" and then partition each group of data into a distinct DataFrame named "new2", "new3", "new4", "new5", and "new6".

We can now eliminate the factors that were not beneficial in predicting “OutcomeScore”after grouping them. Then, using the remaining data, create a correlation table.



We can see from the correlation chart that the correlation between all features and target values is quite low. This implies that the connection between these features and the target value cannot be simply linear. As a result, we will investigate the relationship between the two further using logistic regression.

# [**Classifier using TF-IDF**](#slide=id.g12e4c86d824_0_9)

We use tf-idf to investigate the association between data content and OutcomeScore in this section.

The inverse document frequency (IDF) is a measure of a word's overall importance, rather than its value in content. A highly weighted TF-IDF can be produced by a high word frequency within a single document and a low document frequency of that word over the whole document collection. As a result, TF-IDF has a tendency to filter out common terms while keeping crucial ones.

Furthermore, Term Frequency and Inverse Document Frequency can turn words into numbers so that computers can understand their relationship.

## **TF-IDF**

* **Shape of content**

We use Countvectorizer's fit\_transform method to convert the words in the text into a word frequency matrix. Because we used 19177 sentences and 5528 unique words, the shape after conversion is 19177 / 5528.





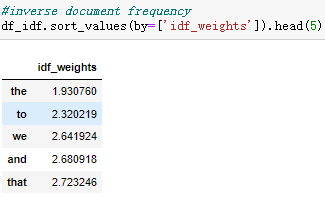
In this part, we did not group the data in order to examine the shape because the goal of this phase is to determine how many unique words exist in the content. If we observe the shapes after grouping and perform subsequent calculations, there is a high chance that the words will repeat.

* **Calculate IDF\_weight**

We use tfidf\_transformer.fit() to calculate the IDF\_weight



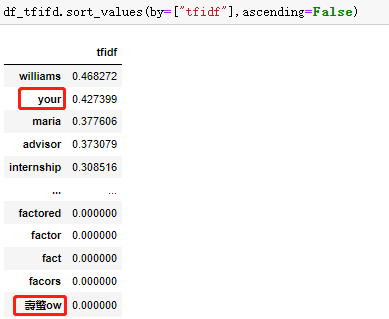
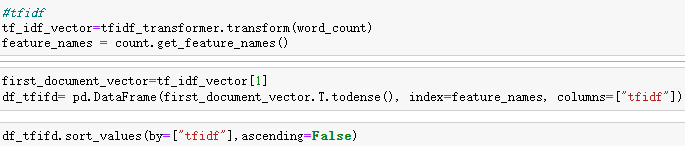
Our data is about content between player and mentor so the terms "the" and "to" appear frequently in content. Word frequency tends to exaggerate how frequently "the" appears, regardless of whether it's a good keyword. As a result, an Inverse Document Frequency factor is applied, which reduces the weight of frequently occurring words while raising the weight of infrequently occurring phrases in the document collection.



* **Continue TF-IDF conversion after IDF is completed**

Based on the number of words in the text and the document frequency in the content data, TF-IDF calculates the significance of a word in the full content data. It offers the advantage of filtering away common but insignificant words like "the" and "to," while maintaining keywords that affect the entire text.

We use to tfidf\_transformer.transform calculate the TF-IDF



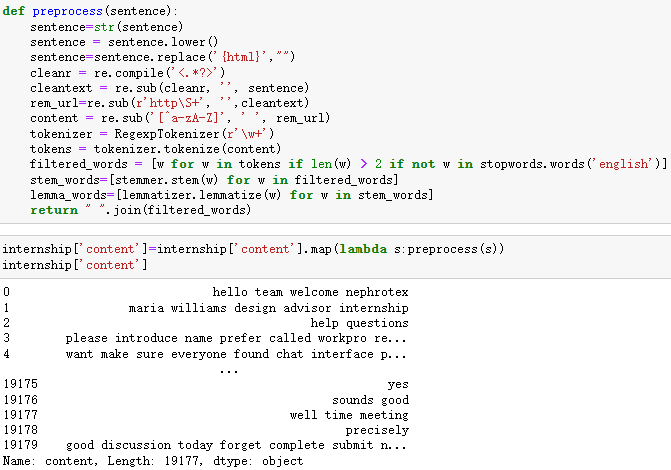
However, some unnecessary words, such as "your" ,which is known as "stop words" and special characters can still be seen in the results.

These words don't supply much in the way of useful information for analysis.

As a result, these terms must also be removed.

* **Create player content classifier using TF-IDF**

We discovered in the last section that the text contains not only stop words but also garbled bits, so we must eliminate not only stop words but also unreadable characters.



“mentor” in “RoleName” should also be eliminated.



Because the “OutcomeScore” of every mentor is 4. The “OutcomeScore” varies from 0 to 8, with 4 being a poor score, and if the 2275 rows of 4 are not deleted, the final result is likely to suffer.

For training and test sets according to a proportion. The “content” column contains X and the “Qualified situation” column contains Y.

We checked the dimension of the data before the data in the test set. 

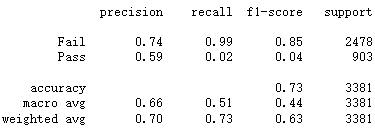
After that, I convert the test data into TF-IDF matrix format.



Since both numbers of features are 4618 a Multinomial Naive Bayes model will be created.

* **naïve Bayesian classifier**

We will use “naive\_bayes\_Classifier()” to predict the test set “X\_ test\_ TF” and store the predicted value in the variable y\_pred.

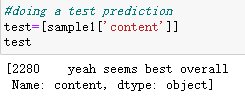
Then we use metrics.classification\_Repor() to output the evaluation index.



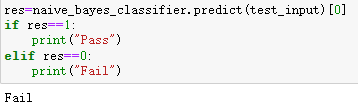
The accuracy is 0.73. Then check the confusion matrix. We know that TP = 2464, FP = 14, FN = 883, TN = 20. Accuracy = 2484 / 3381 = 0.734694... So the classifier performs very well.

In the previous step we added a new column "Qualified situation", in this column 0 [OutcomeScore: 0-4] corresponds to “Fail” and 1 [OutcomeScore: 5-8] corresponds to “Pass”,

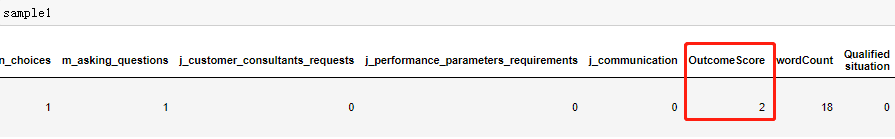
* **Doing a Test Prediction on content Classifier Using TF-IDF**

We randomly select a sample of content to test whether this content belongs to “Pass” or ”Fail”.

we can see this example is for “Fail”.

****

Go to the data source to check, “OutcomeScore” is 2, this content is indeed a “Fail” content.

****

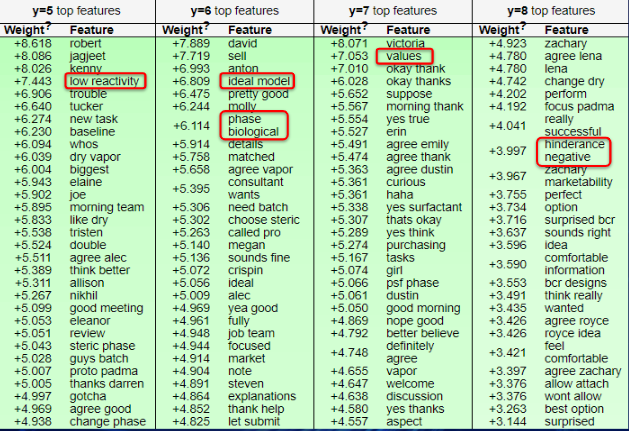
# **Weight table**

We used the combination of TF-IDF and logistic regression to produce the speech content of players with different scores in the virtual internship to see the speaking habits of players in different score segments.

* **Weight table of low outcome scores**



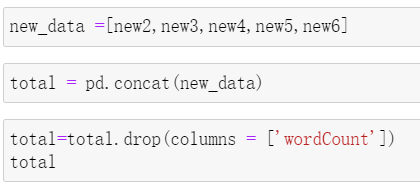
* **Weight table of high outcome scores**



The weight table was divided into two portions, with a score of less than or equal to 4 signifying "Fail," and a score of greater than 4 indicating "Pass," indicating the weight of words used by the player with relatively high result scores. We can see that these players are more likely to use mood words and greetings, as well as "mistake" for making mistakes and "agree" and "think yes" phrases of agreement, in the table for "Fail." Players in the upper segment, on the other hand, frequently use phrases with practical significance, such as "low reactivity" and "perfect model." This means that high-scoring players are usually better at thinking and coming up with important ideas, whereas low-scoring players are more likely to seek advice from others and make mistakes. If players wish to improve their outcome scores, they are suggested to study how players in the high score segment think, act, and speak.

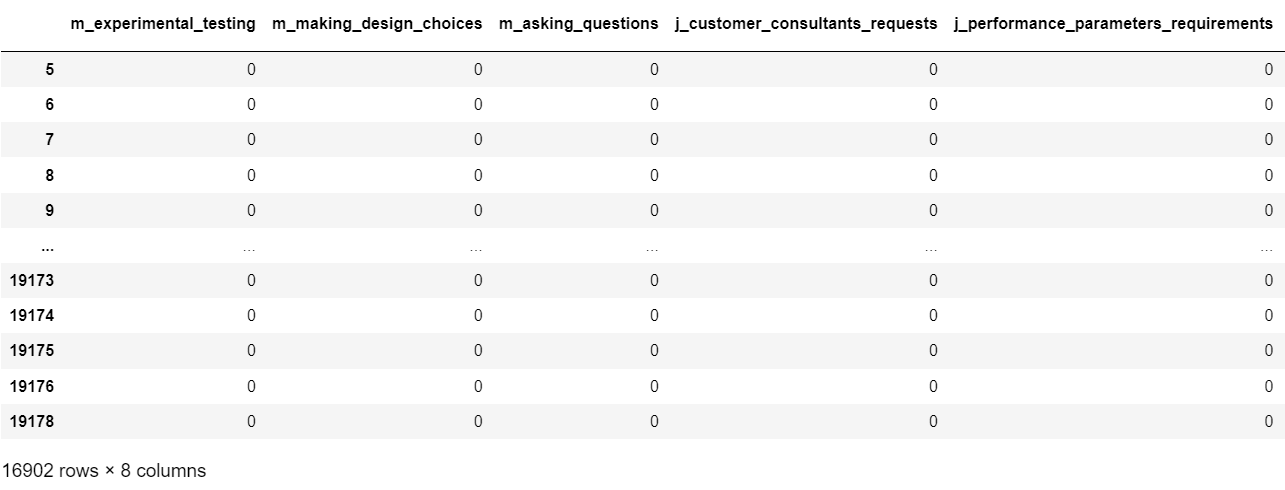
# **Prediction Model using Logistic regression**

* **Take a random sample of 1000 row in total data**



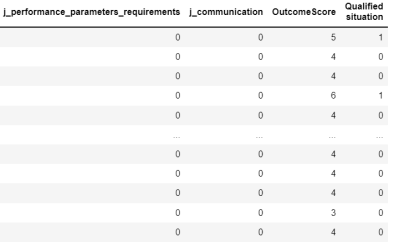
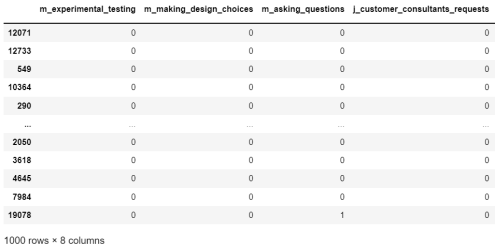
Since the data of virtual internship is not continuous data, but discrete data, and there is not a simple linear relationship between dependent variables and independent variables, we use logistic regression to create a prediction model.

To create a model for predicting results, we first combined the data from each group that had been divided into different sections, containing six variables such as “m\_experimental\_testing”, “m\_making\_design\_choices”, “m\_asking\_questions”, “j\_customer\_consultants\_requests”, “j\_performance\_parameters\_requirements”, “j\_communication” respectively.These six variables are directly related to the outcome scores.



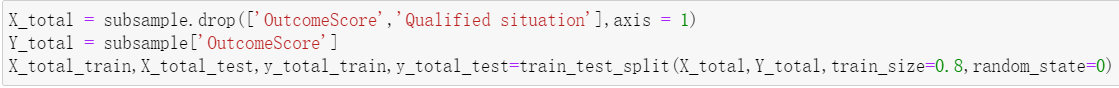
It can be seen that the whole data comprises about 16,000 lines, with many similar lines, which will impair the prediction model's accuracy. As a result, we conducted the modelling using a random sample of 1000 observations.

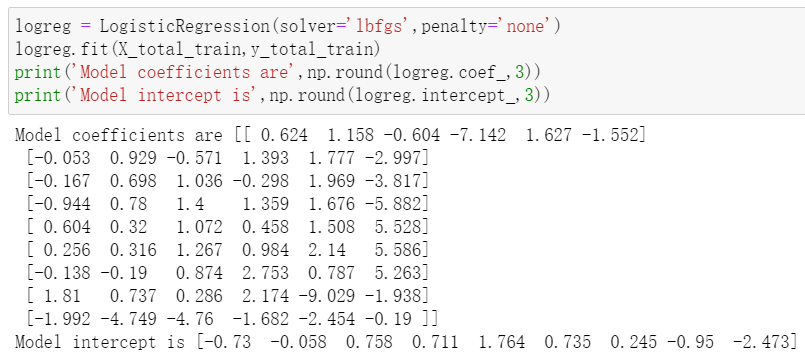




* **Obtain logistic regression model**

Then we took the six variables as X and outcome scores as Y and split the data into training and testing sets to create a prediction model.

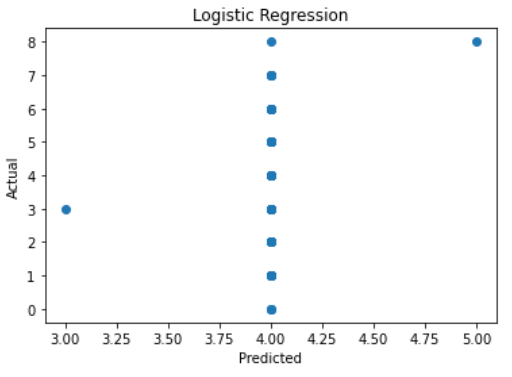




These represent the parameters of the prediction model with a total of 9 outcome scores from 0 to 8, which fit the expression:

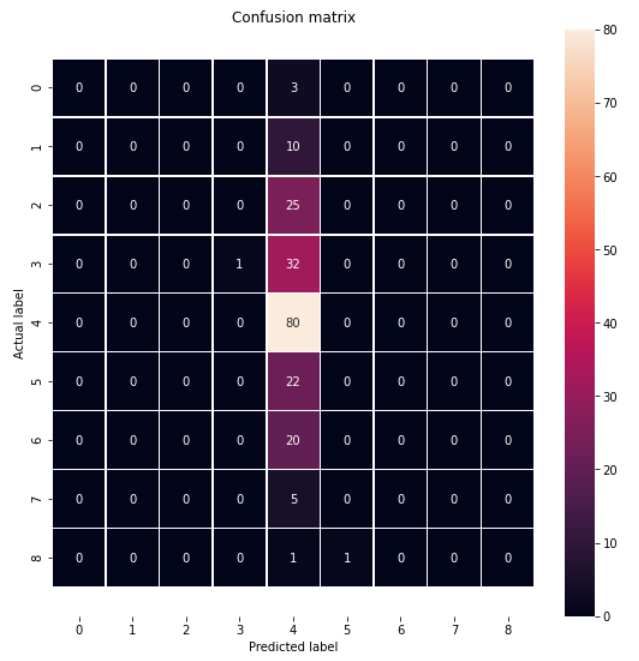


* **Predicted values V.S actual test values**



Then we plotted the predicted values against the actual test values.

However, we can observe that this model's accuracy in predicting result scores is low. Only the score at (4,4) is correct, all other predictions are incorrect. We created a confusion matrix to determine the number of correct and incorrect predictions.

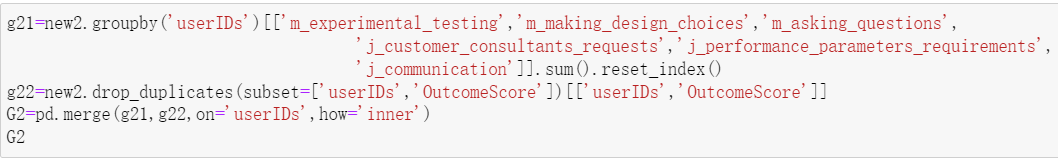


It can be seen that the number of correct predictions is only 80, which is a very small percentage of the total. Then we calculated the accuracy of the model which was only 0.405. The reason for this is that the current data set is made up of 1000 data points that were randomly selected from the original data set, and the number of players for each outcome score is uneven, resulting in low model accuracy. To improve the accuracy, we chose to balance the number of players in each score segment.

## **Improve Accuracy**

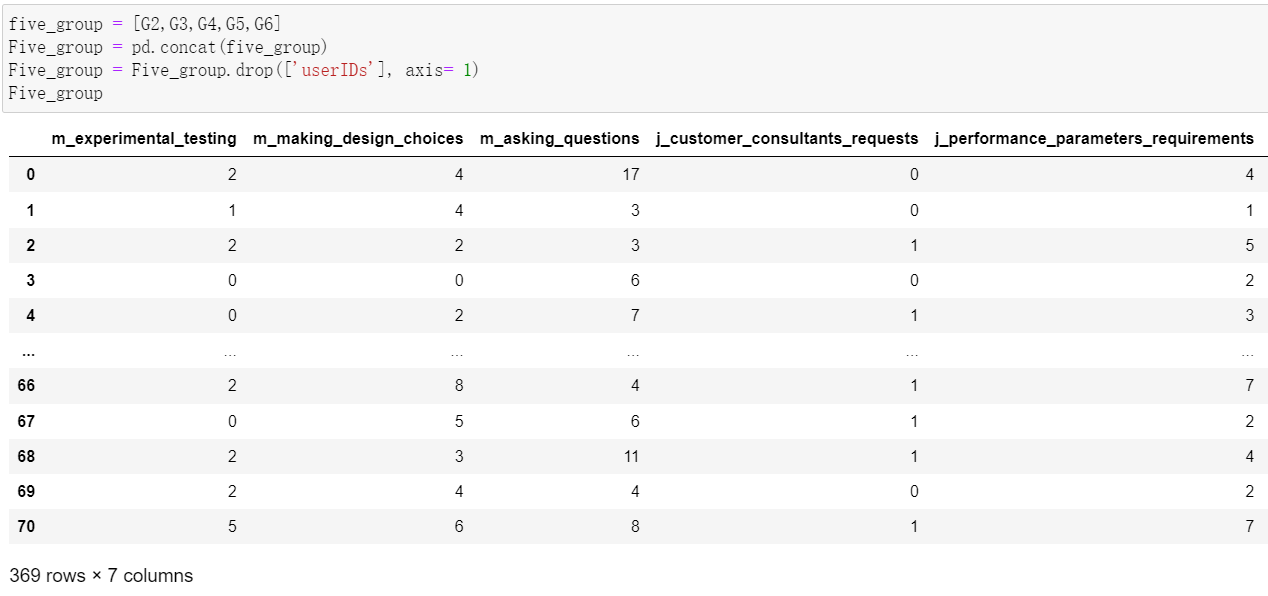
* **Merge data**

We merged the columns with the same userIDs in each group to delete duplicate rows occupied by the same player. In the first group, for example, there are only 78 lines left, compared to almost 3,000 previously, significantly reducing the model's accuracy in predicting result scores.



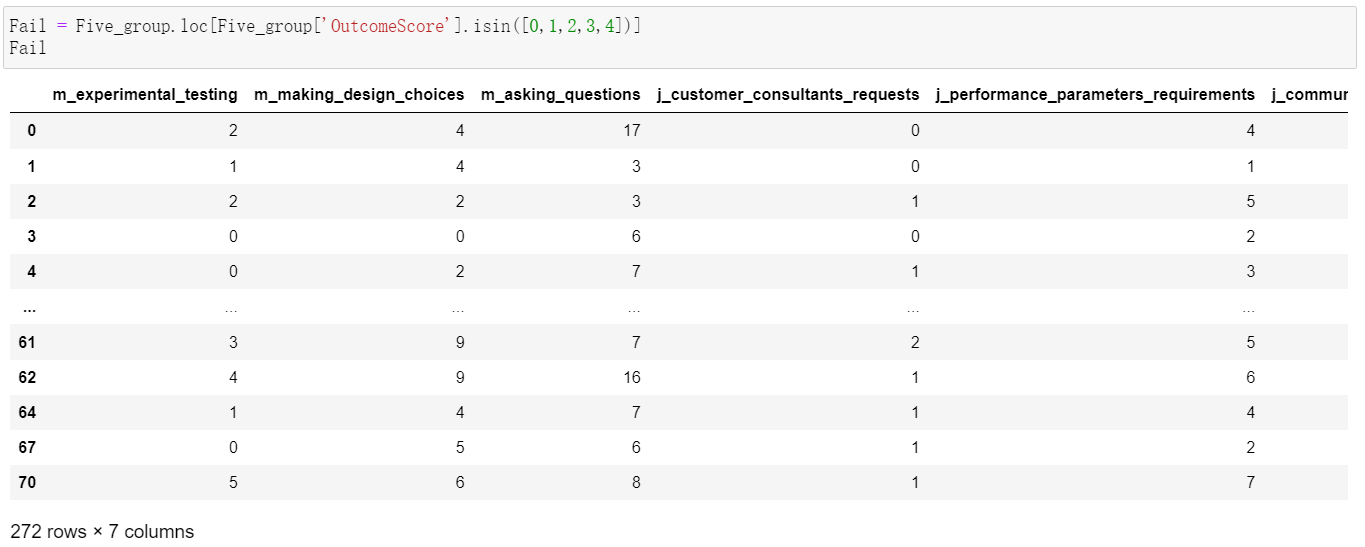


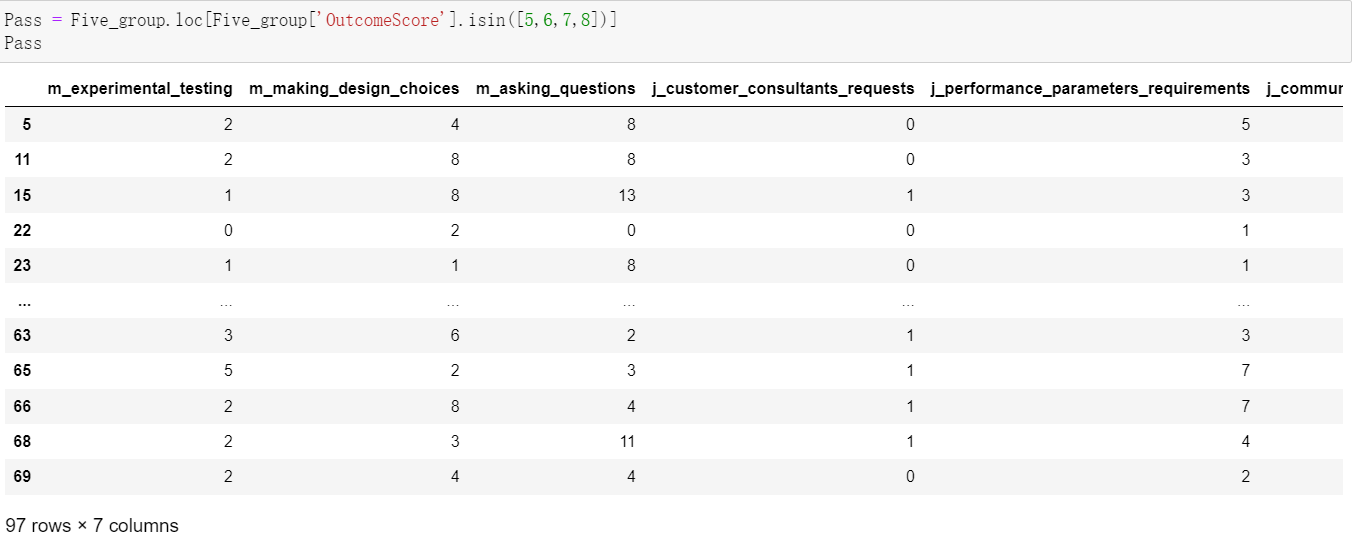
Then we merged five groups of data and obtained 369 rows with unique userIDs, which means that there are only 369 players participating in a virtual internship. This greatly reduces the interference of repeated data to model predictions.



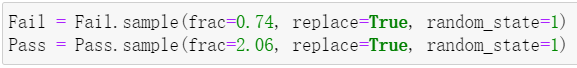
* **Balance data**

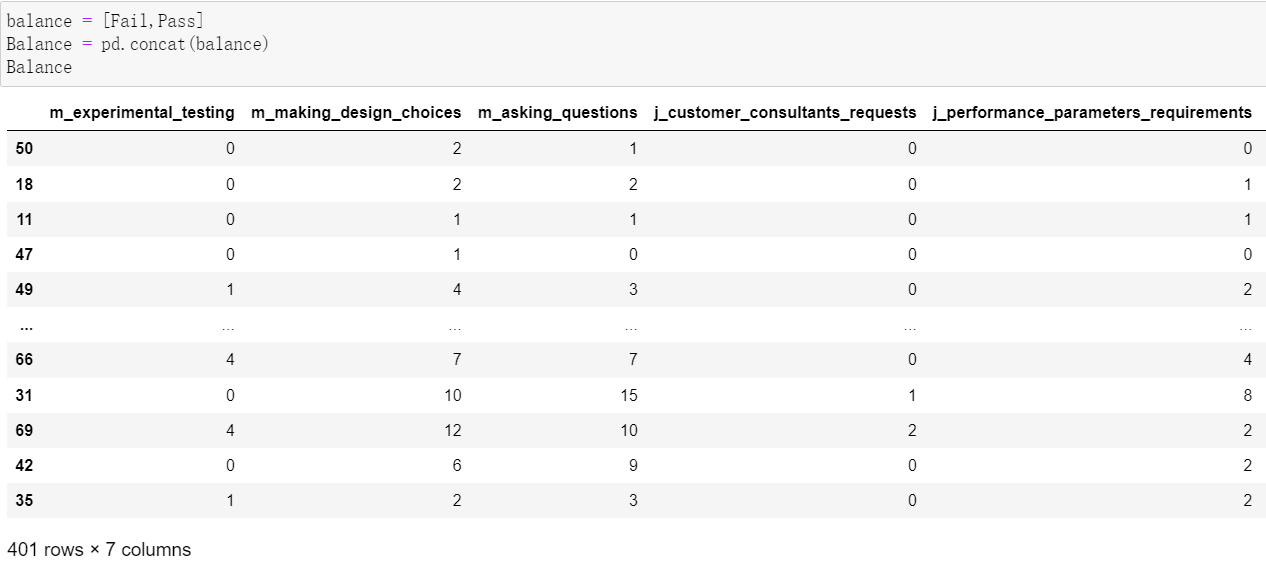
In the balance part, we first grouped the players with outcome scores 0,1,2,3,4 as “Fail” and 5,6,7,8 as “Pass”. At this point, there was a big difference in the number of lines between two parts of data. The “Fail” section has 272 rows while the “Pass” section has only 97 rows.





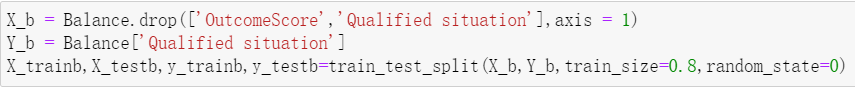
So we balanced the two parts to make the data in both parts was about 200 lines, and combined them to make the total data about 400 lines.





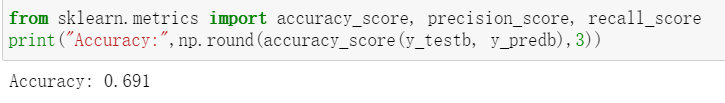
* **Redo logistic regression**

After balancing the data, the existing data set becomes a binary classified data set, and we perform a logistic regression on this data set again. We re-split training and testing sets and obtained a logistic regression model of binay classification.



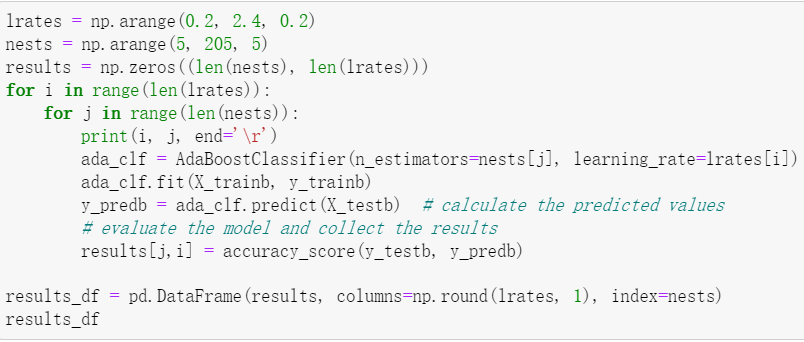


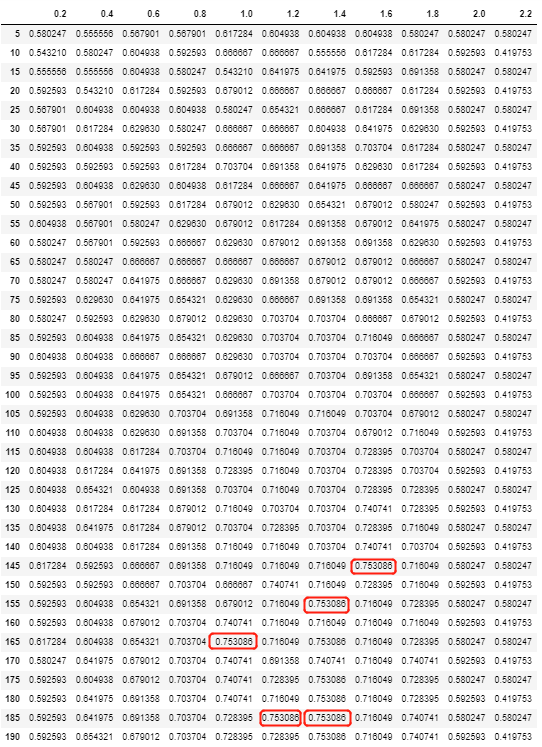
Fortunately, we got a higher accuracy which is 0.691.

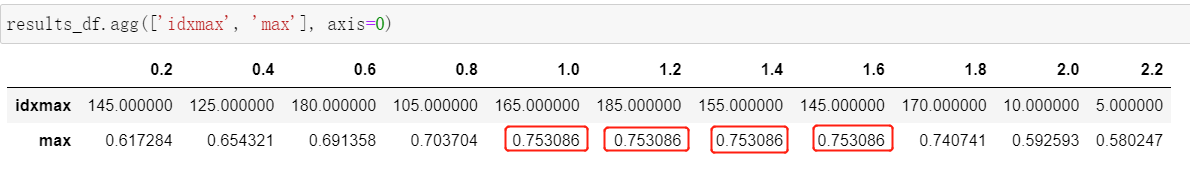


* **Ada Boosting**

To further improve accuracy, we used Ada boosting and found the best estimator and learning rate.



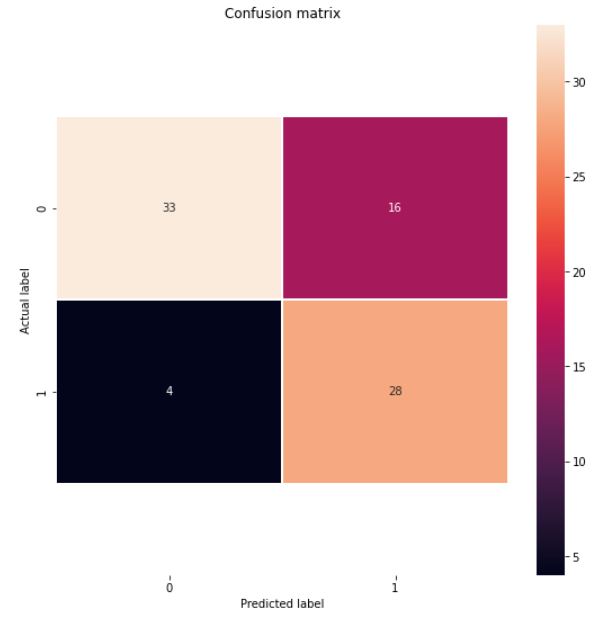




The figures above show the estimator and learning rate of the model with maximum accuracy. It can be seen that there are four groups of different estimators and learning rates that achieve the same accuracy. Therefore, we randomly selected a group for accuracy calculation, and the accuracy rate was effectively improved to 0.753.

* **Confusion Matrix**

We created a confusion matrix again and found that the number of correct predictions significantly increased, which proved that the prediction model we obtained is an effective model with higher accuracy.

****

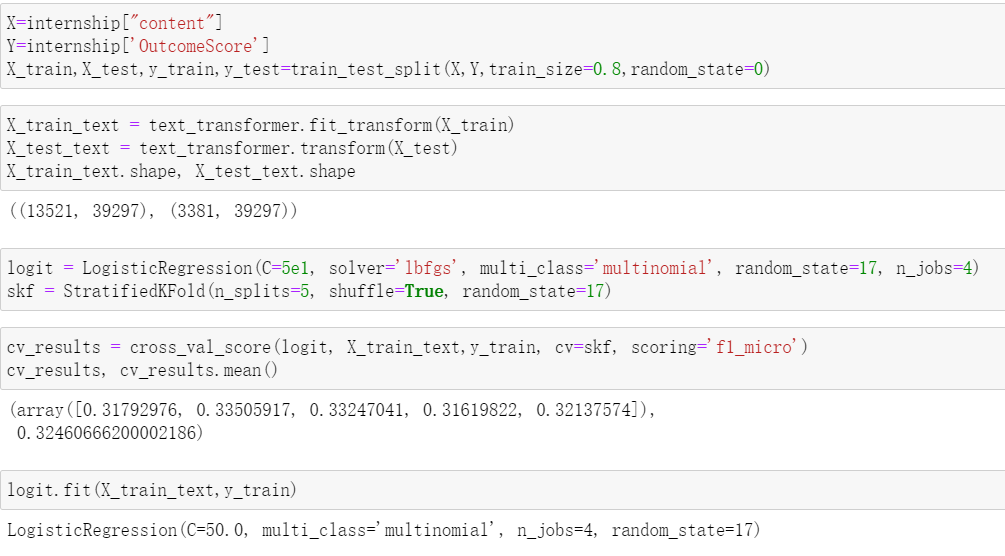
# **Conclusion**

In conclusion, we explored the relationship between different factors in virtual internship and player scores. We used TF-IDF and Naive Bayes to develop classifiers to analyse the player teamwork and communication in an online environment. Through the combination of TF-IDF and logistic regression, we analysed the speaking habits of players with different scores. We also built a model to predict player scores by logistic regression, and Ada boosting was used to improve the accuracy of the model. In the future, we can use this model to help students predict their scores and analyse different content they said and advice on how to get a high score to strengthen their thinking and actions related to specific careers. For further suggestions, since the accuracy of our final model is not more than 0.8, further finding a model with higher accuracy is still the target for improvement in the future.

# **Appendix**

* **The code we used to create weight table**

We divided the content of the players (X) and the outcome score (Y) into training and testing sets. The content and outcome scores are then modelled using logistic regression.



We created the weight table of content and outcome score to interpret model weights with ELI5.(ELI5 is primarily a machine learning library for processing text categorization.Currently, ELI5 allows the interpretation of weights and predictions for scikit-learning linear classifiers and regressors.

