**Intro:**

Good Morning, I am Andrew and together with me in our team are Daniel and Jonathan. Today we will be presenting our mini-project on delivery services. Here is our table of contents.

The impact of COVID-19 caused many families stranded in their homes, facing many difficulties such as buying groceries and et cetera. This led to the increase in users opting for delivery services for their meals instead, using apps such as Food Panda, GrabFood and Uber-Eats to deliver their food straight from restaurants to their doorsteps. From this, we came up with the motivation of finding out what attracts customers to choose a restaurant in a delivery service.

The dataset we used is titled “Online Food Delivery Preferences - Bangalore region”. The dataset is a collection of data, collected by means of a survey from users of a particular delivery service in Bangalore, India. Data such as Demographics of Users, Delivery Preferences, Time and Food Factors and Reviews are recorded in the dataset. By analysing data from the survey results, we aim to identify factors that increase customer purchases.

Hence, we end up with our research question: “What are the optimal factors to attract consumers via food delivery service?”

We planned to investigate how various categories of user opinions on how important certain factors are when making delivery purchases and how it affects whether or not the user will purchase the food from the delivery service and restaurant again, leading to the outcome of predicting and analysing which factors are the most important to users.

**EDA and Data Cleaning:**

Next, we will proceed with Exploratory Data Analysis and Data Cleaning of our dataset.

**Basic Information:**

Firstly, with regards to the Basic Information, we can learn that the median age of most users is 24 and that most users are single, students and have no income. This shows how the majority of delivery services users are University Students to Young Adults who have just entered the workforce.

The Geolocation of the users such as their latitude and longitude are also recorded on the survey, and we can use the Google Maps API to plot their locations in a heatmap. It can be seen that although clients are spread out throughout Bangalore, they are more concentrated closer to the centre of the city. Hence, it may be beneficial for future restaurant owners to position their restaurants near darker spots on the heatmap to minimise delivery times.

**Factors:**

Next, the survey asked for a multitude of different factors relating to the preferences of the users and factors on food and time, asking users to rank how important these factors are on a scale of strongly agree to strongly disagree.

Some of the Delivery Preferences of users also include Preference for Non-Veg Foods, Lunch and Dinner, and also a delivery time of 30-40 minutes.

**Data Cleaning:**

Next, we performed data cleaning on our data set. First, we checked for any NaN values in our dataset by using isnull(). Next, as many of the data within our data set is of the categorical class, we used data encoding to convert categorical data such as ‘male’ ‘female’ or ‘agree’ ‘disagree’ into integers.

**Analysis:**

Next, we will be moving on to the analysis part of our presentation.

**Correlation Matrix:**

With this, we can now create the correlation matrix, that evaluates a linear relationship between variables. As our data points are ranked ordinal values, we use Spearman’s coefficient instead of Pearson’s correlation to determine the relationship between the variables, generating the following correlation matrix. This matrix is significant to our investigation, as it gives a preliminary view of seeing which variables could potentially be a more useful predictor of whether or not users will purchase again.

In the bottom row, we can see the relationship between output, the variable of whether or not users with purchase again with all the other different variables in the dataset. Those that are light in colour have a strong positive correlation, and those dark in colour would have a strong negative relationship.

**Bivariate Statistics:**

Hence, we can try and make a naive prediction on some of the factors that may be more useful in increasing the number of customers to a restaurant delivery service and plot them in a histogram together with Output. These include variables with a strong positive correlation such as Ease and convenience, Time-saving and Good Food quality. By separating output to ‘yes’ and ‘no’, we can plot these variables again in a histogram and compare them. From this, we can visualize the positive relationships between these variables, and whether or not a user will purchase again.

**Natural Language Processing:**

As a part of the dataset is in the form of reviews, we cannot analyse this without converting the text into numbers. This would prove to be one of the challenges to our investigation, and in our research, we came across techniques that could judge the sentiment of human text input, such as reviews and produce a numerical output, allowing for analysis and modelling.

The first method we used includes NLTK (Natural Langauge Toolkit) and SKLearn’s TFIDFVectorizer (Term frequency-inverse document frequency). This converts the reviews into a sparse matrix, that is large in size and hard to deal with. Hence, by using PCA (Principal Component Analysis), a technique for dimensionality reduction in machine learning, the matrix can be reduced to a single column of decimal numbers ranging from -1 to +1 to represent positive or negative sentiment.

Another method we considered is Pattern in Python’s open-source library for NLP. It is a parse function that cleans input data and can find the polarity of a text for modelling. Depending on common positive or negative adjectives, a sentiment score of +1 or - is assigned, together with a subjectivity score from 0 to 1 quantifying the amount of personal opinion and factual information found in the text.

Both methods will be considered in our modelling and evaluation, and we will compare the two to see which will provide greater and/or which would provide greater insight into answering our question.

**Classification Modelling:**

To answer our question, we have decided to choose several classification models such as Logistic Regression, KNN Classifier, Random Forest and XGBoost Classifier and used Cross-Validation GridSearch to improve each of the models. The goal of this is to eventually be able to tell which are the most important variables in getting users to purchase again, and also to evaluate the likelihood of users purchasing again from a hypothetical restaurant based on ratings of the variables in our dataset.

However, we realised that the imbalance in our already small dataset would pose a problem for the training of our models, and hence needed a solution to fix this problem. In our research, we came across the over-sampling technique SMOTE (Synthetic Minority Oversampling Technique), which could serve as a solution to overcome the imbalanced distribution between the categories ‘yes’ and ‘no’ in our output. SMOTE synthesizes elements for the minority class based on existing data points and adds them between chosen points and their neighbours using k-nearest neighbours. Hence, the problem of overfitting in models due to a small dataset is solved, but costing the true accuracy of certain models, and may cause noisy data points.

After resampling for the minority class, with an equal number of positive and negative classes, we can then use the ROC-AUC score instead of the F1 score for evaluation.

**Logistic Regression:**

Firstly, Logistic Regression is chosen instead of Linear Regression as the latter uses an algorithm that predicts continuous values, unlike the former which has an algorithm that helps with Binary Classification, corresponding with our analysis of Output consisting of ‘yes’ and ‘no’.

For all our models, we split our dataset into a training set and a test set using a 20 80 split, training the model using the training set and evaluating it using our test set. After this, we generate the statistics such as Explained Variance, MSE, RMSE, FPR, TPR, Accuracy, Precision, Recall and most importantly the AUC score to evaluate the model. The Logistic Regression Model ends up with an AUC score of 0.89, which is a significant score.

As our model uses a relatively small data size over a large number of variables to consider, we use the cross-validation method of Repeated Stratified K-Fold to ensure our results are not skewed, hence increasing the robustness of modelling and reducing overfitting. Furthermore, using GridSearchCV, we can find what are the best parameters that can improve our model. After searching, the hyperparameters that were found to produce the best score are applied to the Logistic Regression Model, and evaluated again, producing a score of also 0.89, showing little to no improvement for the Logistic Regression model. Hence, other models have also been used for the evaluation

**KNN Classification:**

Next, we use a common technique known as K Nearest Neighbours Classification to classify data based on how a data point’s neighbours are classified. Measures of distance such as Euclidean distance, Hamming distance and Manhattan distance are used to determine the K-closest neighbours.

**Random Forest:**

Another model we use, similar to Logistic Regression is the Random Forest model, which uses the average of decision trees to do binary classification.

**XGBoost:**

Lastly, we tried using a modern, open-source model XGBoost which uses gradient boosting to tune models, minimising the loss function of the model by adding weak learners using gradient descent. This iterative learning process reduces problems such as class imbalance, sample duplication, overfitting, inappropriate node splitting etc.

**Table:**

A table summarising the AUC scores can be seen here.

|  |  | Logistic Regression | KNN Classification | Random Forest | XGBoost |
| --- | --- | --- | --- | --- | --- |
| First Results | NLP 1 | 0.888 | 0.927 | 0.950 | 0.947 |
| NLP 2 | 0.922 | 0.955 | 0.967 | 0.968 |
| After Tuning | NLP 1 | 0.888 | 0.938 | 0.966 | 0.955 |
| NLP 2 | 0.923 | 0.956 | 0.986 | 0.972 |

**Evaluation:**

All 4 of our models have AUC score of above 85%, after taking into consideration of the issue of overfitting using SMOTE, CrossValidation techniques and tuning it to its best parameters using GridSearch. This is considered an excellent score and so they serve as good models to answer our question.

In general, Random Forest was the most accurate model of all, followed closely by XGBoost, then KNN then Logistic Regression. However, it should be noted that due to the use of the KNN Classifier with SMOTE, the accuracy might be influenced positively, causing bias in the score.

Also, although XGBoost did not perform as well as Random Forest, since XGBoost does shrinkage and column sub-sampling, it avoids over-fitting even further. Also, due to computation time, we could not include all the possible hyperparameters.

Comparing NLP methods 1 and 2, we can see that method 2 generally provides a higher AUC score than method 1. From this, we also realised how sentiment analysis is a deep concept and that our attempt at using the TFIDF vectorizer produced good results, but Pattern produced a better AUC ROC score comparatively.

**Conclusion:**

With our Random Forest Model having the highest accuracy out of all 3, we can now use the model to find out what are the most important variables that affect a user’s decision to or to not purchase again.

By showing the feature importance, we can see the top 3 most important variables in decreasing impact are Time Saving, Ease and Convenience, and More Offers and Discount, which is consistent for both NLP methods. Also, we can see how this result is similar to what we predicted based on the correlation matrix, with the exception of Good Food Quality. Using this, we can conclude that for a restaurant in Bangalore to attract more customers via delivery services, it should have a delivery service that is easy and convenient, its delivery time needs to be fast, and the restaurant gives offers and discounts regularly.