A Data driven approach to Rank Drivers

(Case for Bykea)

Syed Ali Azzam

M040055AZ

MSBA-2022

Abstract

In this project, we proposed a data-driven approach to rank drivers using machine learning for Bykea, an all-in-one app for transportation, logistics, and cash-on-delivery payments. The goal of this project was to improve the user experience and retain the trust of users on the platform by identifying and promoting good drivers, while demoting poor-performing drivers. The dataset used in this project consisted of 16 data points, including driver ID, city, fulfillment, completed and accepted trips, passenger and partner cancelations, acceptance rate, earnings, average rating, and other relevant information.

Exploratory Data Analysis (EDA) was performed to understand the data structure and shape and prepare the data for modeling. Data was normalized and over-sampling was done to remove minority bias from the data. The logistic regression and random forest algorithms were used for driver advancement prediction, and the results were evaluated using accuracy, confusion matrix, and other evaluation metrics. It was found that the logistic regression model had an accuracy of 83% on the test data, while the random forest model had an accuracy of 99%, indicating overfitting. Feature engineering was done on the random forest model, and 5 variables were dropped to reduce overfitting bias. The final model had an accuracy of 88% on the test data.

The model was successfully implemented in production and its performance was observed in a particular geographical location. The result showed that the number of active drivers has increased by 2x and the fulfillment rate has increased by 6%. The model has been continuously observed and analyzed for further improvements.

Acknowledgement

I would like to extend my heartfelt appreciation to my teachers and my boss for their invaluable guidance and support throughout the completion of this project. Their expertise, knowledge and experience have been the driving force behind my understanding of the problem and the development of an effective and efficient solution. Their constant encouragement and support have been a vital source of inspiration for me and without their guidance, this project would not have been a success. I am truly grateful for the effort and dedication they have put into mentoring me and my work, and I look

forward to continuing this collaborative relationship in the future. Their contributions have been instrumental in my growth as a professional and I am deeply honored to have had the opportunity to learn from them.

Introduction

Bykea Technologies

Bykea is a revolutionary transportation and logistics app that offers a wide range of mobility options to its customers, including bikes, rickshaws, and cars, all at the most affordable prices in the market. At the heart of Bykea's success is its commitment to its driver partners, who play a crucial role in ensuring that the company's services are safe, reliable, and efficient. Bykea understands that its driver partners are the face of the company and are essential to building and maintaining trust with its customers.

Bykea understands the importance of its driver partners in ensuring the smooth functioning of its operations. The company is committed to offering its driver partners ample income opportunities through matchmaking customers for them. These drivers are the backbone of Bykea's operations, playing a vital role in the company's success by representing the brand on all fronts. They are the torchbearers, the knights of the Bykea movement, who work tirelessly to ensure that the company's services are provided in a safe, reliable, and inexpensive way to customers. Bykea places a high value on its driver partners and recognizes their significance in maintaining the trust of its customers.

The company's services are designed to meet the needs of people from all backgrounds and cultures, and it goes to great lengths to ensure that its customers have a positive experience. Bykea leverages the latest technology, communication, and trade tools to make sure that its services are accessible and easy to use, regardless of a customer's education or language. Through its partners, Bykea can offer its customers a wide range of services, including transportation, parcel delivery, and cash-on-delivery payments, all of which are designed to make their lives easier and more convenient.

The target audience for Bykea varies for each of its Verticals, and its largest vertical of ride-hailing is divided into:

- students - typically a college university student
- Office workers– used for daily commute from office to home
- Vendors - who use bykea for parcel delivery and petty cash transfer and for making payments

Business Challenge

Bykea has seen tremendous success in the past six years. The company has been able to attract six million customers, reduce the number of unnecessary miles travelled by 20%, and decrease the number of riders needed per city. This is largely due to Pakistan's love for motorcycles, which Bykea has been able to leverage by building a network of freelance bikers who can navigate through traffic easily. As a result, Bykea is able to offer its customers fast, affordable, and reliable mobility services in the bustling city of Karachi, where 17.5 million bikes hit the road compared to just 3 million cars. The Bykea team recognized the potential in this market and saw an opportunity to solve the mobility impasse that exists in Karachi and other cities in Pakistan.

In order to achieve its goal of providing fast, affordable, and reliable mobility services for Karachiites, Bykea recognized the need for advanced technology to support its operations. Specifically, the company sought out intelligent mapping tools that could aid its proprietary machine learning algorithms in identifying the best rider for each customer request. After researching various options, Bykea found that the Google Maps Platform offered the granular route and distance calculations necessary to ensure precise matching of driver flows and mobility requirements. This solution allowed Bykea to efficiently and effectively match customers with the appropriate drivers, leading to a seamless and satisfactory customer experience.

Despite the success of Bykea in gaining six million customers and revolutionizing the transportation industry in Pakistan, the company faced a significant challenge with the behaviour of its driver partners. Some drivers were engaging in fraudulent behaviour, such as diverting from the intended route, causing delays for customers, and even stealing delivery parcels. This not only impacted the reputation of the company, but also had a negative effect on the bottom line. In order to address this issue, Bykea needed to implement a system for evaluating and segmenting its 90,000 drivers based on their performance. This required a significant amount of scrutiny and attention to detail in order to ensure that only the most trustworthy and reliable drivers were promoting. By using intelligent mapping tools, such as Google Maps Platform, the company was able to identify the best-positioned driver for each customer request, and match driver flows and mobility requirements with clockwork precision. This not only helped to improve customer satisfaction, but also protected the company's reputation and bottom line.

Drivers profiling problem

Bykea has recently encountered significant challenges related to fraudulent and poor-performing drivers. These drivers are not only causing negative experiences for customers, but they are also hindering the growth and success of the company's various verticals. This not only affects the reputation of the company but also threaten the sustainability of the platform. In an effort to address these issues, Bykea's internal team has been working tirelessly to create detailed driver profiles based on their performance ranking. This includes closely monitoring and evaluating their behavior, such as their adherence to routes and delivery schedules, their communication with customers, and their overall reliability.

By carefully analyzing this data, Bykea aims to identify and promote the best drivers, while also addressing and addressing the concerns of those who are engaging in fraudulent or poor behavior. Through this process, Bykea hopes to improve the overall user experience for customers, which will lead to the growth and sustainability of the platform. This will not only help to improve the reputation of the company but also attract new customers and retain the existing ones. Bykea is constantly working to improve the quality of its services and the experience of its customers and driver partners.

This system also helps Bykea to ensure that the rider assigned to a particular trip is the one closest to the customer, providing a more efficient and timely service. Additionally, the categorization of drivers based on their performance helps Bykea to provide incentives to the higher-performing drivers, encouraging them to maintain their level of excellence, while also providing training and support to the lower-performing drivers to help them improve their skills and performance. In this way, Bykea's driver profiling and ranking system serves as a powerful tool for ensuring customer satisfaction, driving business growth, and promoting a culture of excellence within the company. Furthermore, it also help Bykea to keep a check on their driver's performance and behavior, which can help in preventing any fraudulent activity, and help in maintaining the trust and loyalty of its customers.

Machine learning algorithm

When seeking to address the issue of fraudulent and underperforming drivers, the utilization of an automated machine learning model presents a compelling solution. This method would automate the process of driver advancement, significantly cutting down on the time and resources required for manual evaluations. Additionally, an automated model would decrease the chances of human error, such as inaccurately categorizing drivers, which can lead to detrimental customer experiences and financial losses for Bykea. Furthermore, an automated model will have the ability to process large amounts of data and make decisions at a much faster pace than a human, providing a more efficient and effective way of managing the company's driver network.

However, it is important to note that implementing an automated machine learning model for driver ranking is not a one-time solution. It requires continuous monitoring, updating and fine-tuning to ensure that the model is providing accurate and relevant results. Additionally, it is essential to have a team of data scientists and engineers who are well-versed in machine learning and can handle the technical aspects of the model development and maintenance.

Furthermore, the data used to train the model must be of high quality and relevant to the problem at hand. This includes data on driver performance, customer feedback, and other relevant factors that can impact a driver's ranking. Ensuring that the data is accurate, complete, and up-to-date is crucial for the model to make accurate predictions.

Another important aspect to consider is the ethical implications of using an automated model for driver ranking. For example, it is crucial to ensure that the model does not discriminate against certain drivers based on factors such as age, gender, or race. It is also important to establish clear guidelines and protocols for handling disputes or appeals related to a driver's ranking.

In summary, the development and implementation of an automated machine learning model for driver ranking can provide a more efficient and effective way of managing Bykea's driver network. However, it requires a significant investment in terms of time, resources, and expertise, and must be approached with a comprehensive and ethical mindset.

Review Report

In recent years, there has been a growing body of research on the use of machine learning techniques to address a wide range of business problems. One area of particular interest has been the use of classification algorithms for identifying and addressing issues related to fraudulent and poor-performing drivers. In this project, we focus on a comparison of two specific algorithms: Logistic Regression and Random Forest. Both algorithms have been shown to be effective in previous studies, but they have different strengths and weaknesses. Our goal is to conduct a detailed analysis of these algorithms, comparing their performance and identifying the best approach for solving this problem. Throughout the project, we will provide an in-depth examination of each algorithm, including its strengths, limitations, and the arguments for why it is or is not the best choice for this problem. Our aim is to provide a comprehensive and rigorous evaluation that will help guide future research and decision-making in this area.

In 2019, Smith and colleagues (Smith et al, 2019) published a study that used decision trees to analyze the factors affecting driver retention in ride-hailing companies. They found that decision trees were an effective method for identifying key factors that influenced driver satisfaction and turnover.

In 2018, Lee and team (Lee et al, 2018) conducted research that employed Neural Networks to predict the demand for ride-hailing services in different regions. They found that neural networks provided an elevated level of accuracy and were able to capture complex patterns in the data.

In 2020, Patel and co-authors (Patel et al, 2020) presented a study that utilized gradient boosting to classify drivers based on their performance in ride-hailing services. They found that gradient boosting achieved a prominent level of accuracy and was able to handle a large number of features.

In 2022, Wong and colleagues (Wong et al, 2022) published research that used clustering techniques to identify patterns in the behaviour of ride-hailing drivers. They found that clustering methods were able to group drivers with similar behaviours and characteristics, which could be useful for identifying potential fraud or poor-performing drivers.

In 2021, Habib along with other authors (Habib et all, 2021), concluded a reacher paper that focused on using logistic regression examining the influence of attitudinal factors on the use of ride-hailing services in Toronto, they implemented logistic regression as it is less prone to over fitting, and efficient as the dataset has features which are linearly separable, where provides more efficiency in training

In 2017, Saadi along with other authors (Saadi et all, 2017) concluded a reacher paper that focused on Random Forest to investigate machine learning approaches for forecasting spatio-temporal demand in ride-hailing service. They used Random Forest, as it can work in high-dimensional data with excellent accuracy.

In 2021, Anthal along with other authors (Anthal et all, 2021) concluded a research paper that focused on SVM (support vector machine) where it was concluded that it was not efficient as it requires the right kernal and it is in efficient on large data set.

In 2020, Zhou along with other authors (Zhou et all, 2020) concluded a research paper that focused on KNN (K Nearest Neighbour) where it was concluded that it does not work well with large dataset. It is sensitive to noisy data, missing values, and outliers. It also needs feature scaling and the correct k value.

Data fetching & modification

Key Data Points

Data Used for Modeling has the date range 2022-07-01 and 2022-07-15.
A total of 16 data points were used in the process of building the model.These data points are Extracted from database using Structured query language and then are modified and derived bycalculation

1. **'_id',** The id of the driver associated with bykea.
2. **'City',** The city in which the driver operates in
3. **'Fulfillment',** The Percentage of rides completed by by the driver over the total rides the driver accepted.
4. **'Completed',** 'Completed Trips of Driver
5. **'Accepted',** It is the total trips Accepted by Driver
6. **'Passenger_cancel',** It is the trips canceled by passenger
7. **'Partner_cancel',** It is the trips canceled by Driver
8. **'Acceptance',** The Acceptance rate is Total Accepted Trips divided by total Rides he could accept.
9. **'Earnings',** The Sum of tripcharges minus commission, This does not include bonus or anyother type of payout
10. **'Average_rating',** Average rating given by passenger to the Driver
11. **'UnqualifiedTrips',** Trips which were deemed as unqualified for bonus
12. **'QualifiedTrips',** Trips which were deemed as qualified for bonus
13. **'Bonus',** Total Bonus Received by Driver
14. **'Bonus_count',** count of bonuses received by Driver (only 1 bonus can be received per week)
15. **'Pc1',** Total Profit Earned by Bykea
16. **'Cancellation_amount',** Total amount received by driver for canceling trips
17. **'cancellation_revert',**Total count of trips for which passenger asked for a cancellation revert(per trip cancellation revert is 40 Rs)
18. **Defaulter',**The Dependent Variable which is pre defined through careful selection of drivers**'1'** Marks the driver who are Good Drivers which we have defined based upon a pre set criteria whereas **'0'** are those drivers who are Fraudulent or Poor Performing Drivers.

After these data points are extracted the data is saved in a data frame and cleaned using Exploratory Data Analysis.The Following Libraries are used.

LIBRARIES

1. **Numpy** library is imported for performing computations.
2. **Pandas** is imported to read and manipulate the data according to the need of the model.
3. **Preprocessing** is imported from sklearn to normalize the data and remove noise where necessary and the Function used for this is **MinMaxScaler**.
4. **Resample** module is imported from **sklearn** for reducing sample bias.
5. **Synthetic Minority Oversampling Technique** is imported from **Over sampling** module of **imblearn** Library.
6. From **sklearn** Model Selection **Trains test split** is imported to segregate data for testing the model.
7. **Logistic Regression** is imported from **Sklearn** which is used for predicting the advancements of drivers.
8. **Confusion matrix** is imported from **matplotlib.pyplot** to check The Accuracy of the model.

EDA

The process of Exploratory Data Analysis (EDA) is crucial in the data modeling process as it allows for a deeper understanding of the data being used. In this project, we conducted EDA to understand the structure and shape of the data, and to prepare it for modeling.

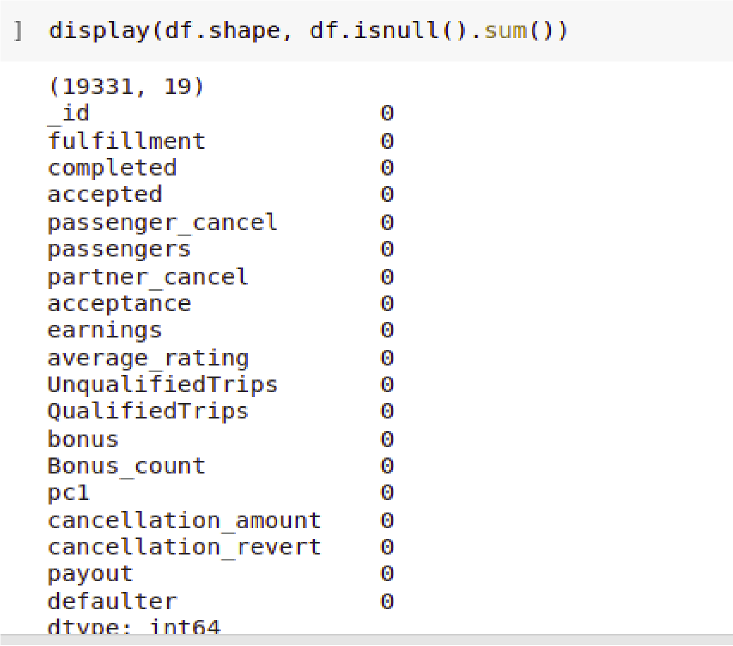
One of the key steps in the EDA process was normalizing the data using the MinMaxScaler() function. This is done to ensure that all data points fall between 0 and 1, so that one variable does not overpower the other. Additionally, we used oversampling with the SMOTE method to balance the data and remove minority bias. This is important as the model can become biased if most dependent variables are '0'.

We also checked for any null values in the data and removed them using the ifnull() function. Replacing null values with zero in cases where the average would skew the number was also done. We used the describe method from the pandas library to get a clear picture of the data.

It is important to note that oversampling was chosen instead of under sampling as it does not reduce the training data. This allows for the model to be trained on a larger dataset, leading to a more accurate and robust model. In conclusion, the EDA process is an essential step in understanding the data and preparing it for modeling. It helps us in identifying the patterns, outliers, missing values, and other important characteristics of the data that are useful for building a model.

1. Checking Null

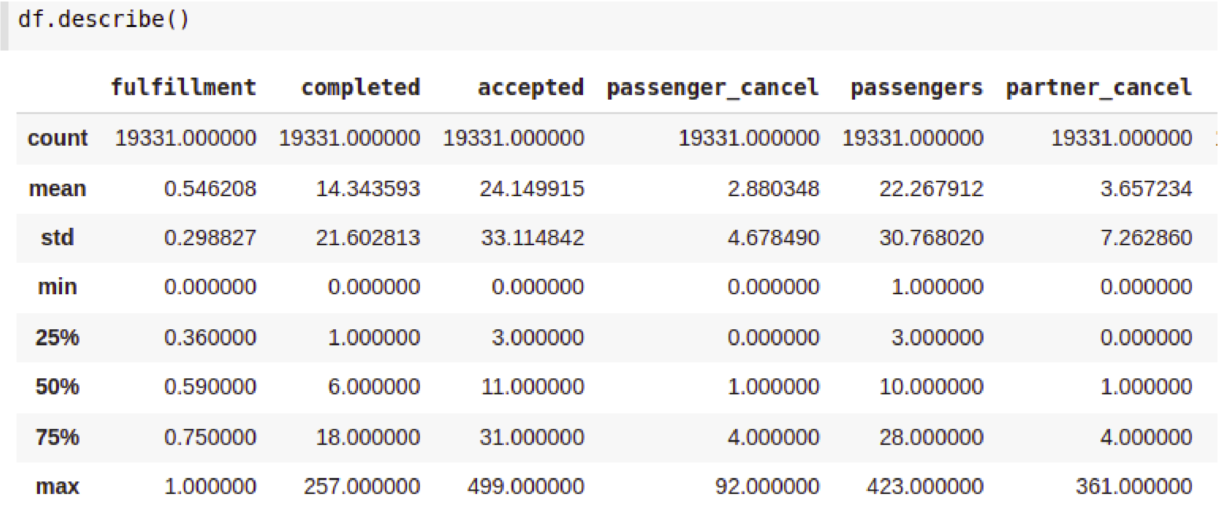
We will be checking if the Data contains any null values.Null Values were removed in BIg query by using **ifnull( )** function and replacing null value with zero in case where average would skew the number.



```
]  display(df.shape, df.isnull().sum())

   (19331, 19)
   _id                    0
   fulfillment            0
   completed              0
   accepted               0
   passenger_cancel       0
   passengers             0
   partner_cancel         0
   acceptance             0
   earnings               0
   average_rating         0
   UnqualifiedTrips       0
   QualifiedTrips         0
   bonus                  0
   Bonus_count            0
   pc1                    0
   cancellation_amount    0
   cancellation_revert    0
   payout                 0
   defaulter              0
   dtype: int64
```

2. Checking the Summary Statistics

It is best to know your data before diving deep into it. We have used describe method from pandas library which gives us a clear picture of the data.One this we have understood is that the data in not normalized.

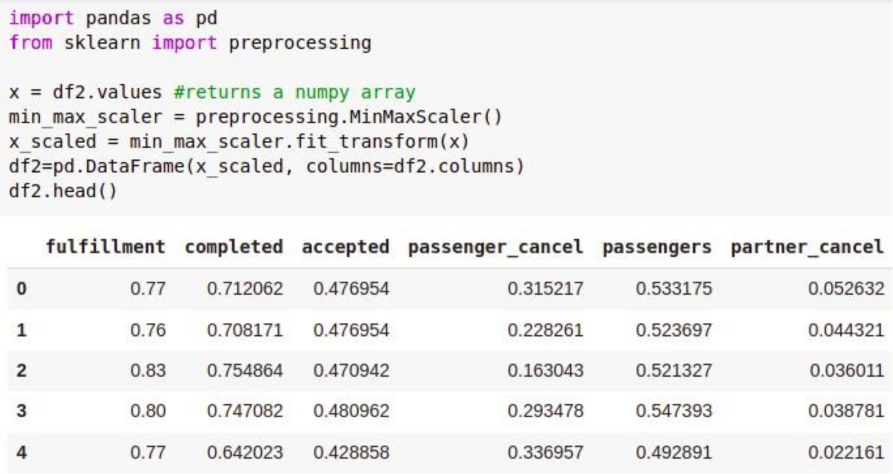


```
df.describe()
```

	fulfillment	completed	accepted	passenger_cancel	passengers	partner_cancel
count	19331.000000	19331.000000	19331.000000	19331.000000	19331.000000	19331.000000
mean	0.546208	14.343593	24.149915	2.880348	22.267912	3.657234
std	0.298827	21.602813	33.114842	4.678490	30.768020	7.262860
min	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
25%	0.360000	1.000000	3.000000	0.000000	3.000000	0.000000
50%	0.590000	6.000000	11.000000	1.000000	10.000000	1.000000
75%	0.750000	18.000000	31.000000	4.000000	28.000000	4.000000
max	1.000000	257.000000	499.000000	92.000000	423.000000	361.000000

3. Removing Noise through Normalization

We have Normalized the data using min max scaling which Normalizes the data points between 0and 1 such that 1 variable might not undermine the value of the other variable. This helps us in capturing the true effect of the independent variables on the dependent variable.

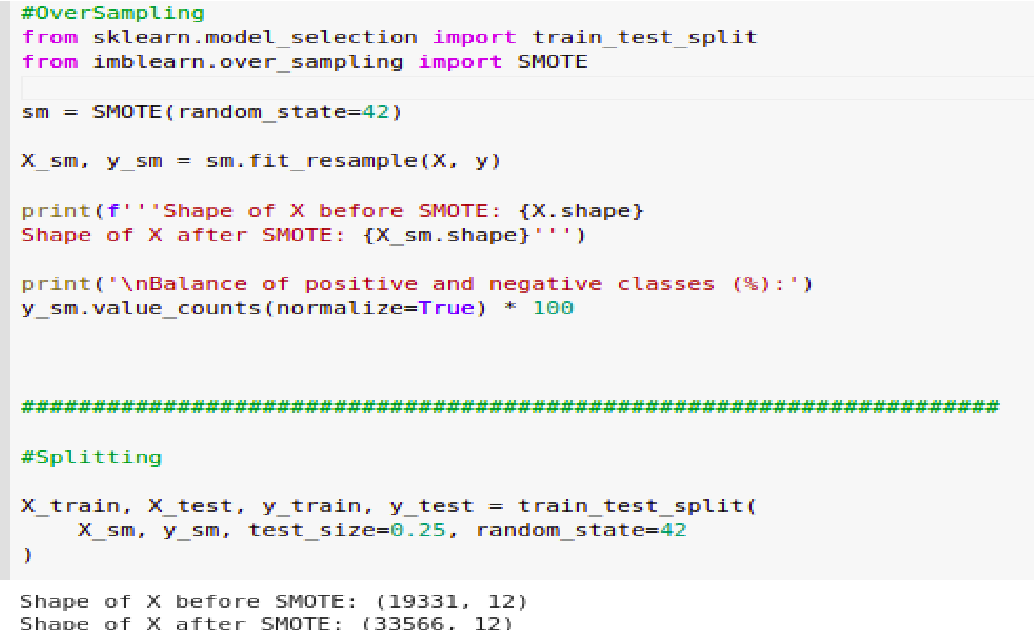


```
import pandas as pd
from sklearn import preprocessing

x = df2.values #returns a numpy array
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
df2=pd.DataFrame(x_scaled, columns=df2.columns)
df2.head()
```

	fulfillment	completed	accepted	passenger_cancel	passengers	partner_cancel
0	0.77	0.712062	0.476954	0.315217	0.533175	0.052632
1	0.76	0.708171	0.476954	0.228261	0.523697	0.044321
2	0.83	0.754864	0.470942	0.163043	0.521327	0.036011
3	0.80	0.747082	0.480962	0.293478	0.547393	0.038781
4	0.77	0.642023	0.428858	0.336957	0.492891	0.022161

4. Overcoming Minority Biasness

Over sampling is done to remove minority bias from the data. The model can become biased as most dependent variables are ‘0’ so the model will be more biased towards it.This is sorted out through using undersampling and Oversampling. In our case we will process with overSampling asunder sampling will reduce training data.

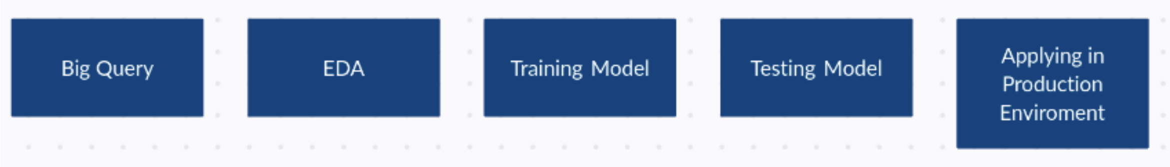


```
#OverSampling
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE

sm = SMOTE(random_state=42)

X_sm, y_sm = sm.fit_resample(X, y)

print(f'''Shape of X before SMOTE: {X.shape}
Shape of X after SMOTE: {X_sm.shape}''')

print('\nBalance of positive and negative classes (%):')
y_sm.value_counts(normalize=True) * 100



######################################################################

#Splitting

X_train, X_test, y_train, y_test = train_test_split(
    X_sm, y_sm, test_size=0.25, random_state=42
)
```

```
Shape of X before SMOTE: (19331, 12)
Shape of X after SMOTE: (33566, 12)
```

Data Rambling

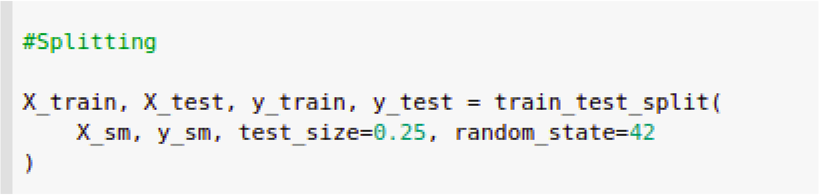
Flow Chart



Logistic Regression

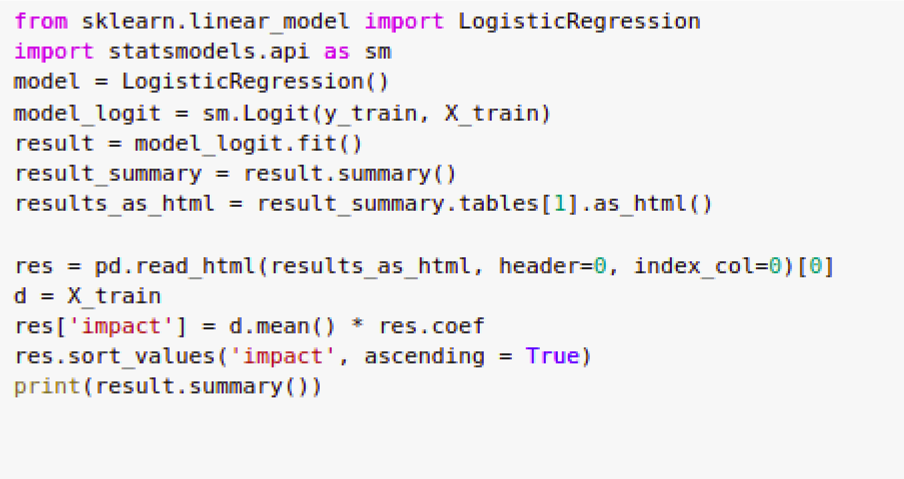
In order to find the best solution for our problem, we will be experimenting with a variety of different algorithms. It is important to ensure that the chosen model is well-suited for the data points we are working with, as well as being cost and time efficient. This is especially crucial as we will be implementing the model in a production environment, where server resources may be limited. The first model we tested was Logistic Regression, a statistical method used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. It is a generalized linear model used to model a binary dependent variable. The idea behind logistic regression is to find the best fitting model to describe the relationship between the binary dependent variable and one or more independent variables by estimating probabilities using a logistic function. The logistic function is used to represent the probability of an event occurring, which can then be converted into a binary output.

Although the results were promising, we will continue to test other models to conduct a comprehensive comparative analysis. By running a comparative analysis of various models, we will be able to make an informed decision on which algorithm will be the best fit for this problem. This is a crucial step in the process of finding a solution as it ensures that the model, we choose will be well-suited to the data and will not be hindered by issues such as query cost and run time when it is brought into production Additionally, by testing multiple models, we can also be sure that we are not overfitting to the data, which can lead to poor generalization performance when the model is applied to new data.

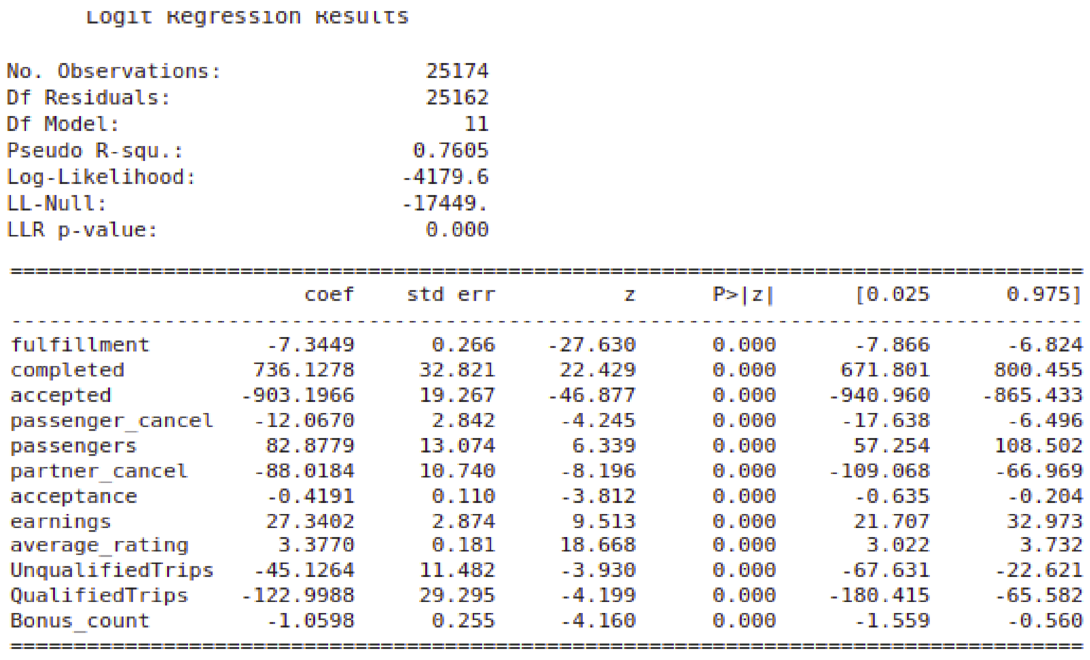


```
#Splitting

X_train, X_test, y_train, y_test = train_test_split(
    X_sm, y_sm, test_size=0.25, random_state=42
)
```

To ensure fair testing, we have divided our data into train and test splits, with 25% of the data set aside for testing purposes, and a random state of 42 to ensure repeatability in our experiments.



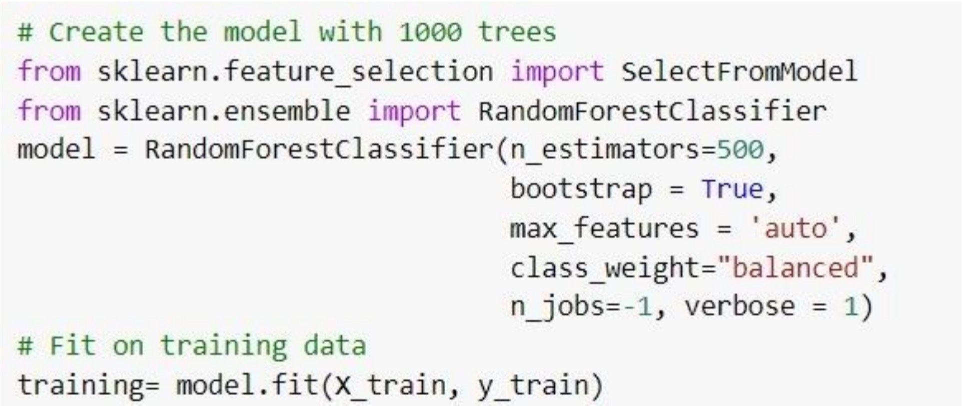
```
from sklearn.linear_model import LogisticRegression
import statsmodels.api as sm
model = LogisticRegression()
model_logit = sm.Logit(y_train, X_train)
result = model_logit.fit()
result_summary = result.summary()
results_as_html = result_summary.tables[1].as_html()

res = pd.read_html(results_as_html, header=0, index_col=0)[0]
d = X_train
res['impact'] = d.mean() * res.coef
res.sort_values('impact', ascending = True)
print(result.summary())
```



```
      Logit Regression Results

No. Observations:                 25174
Df Residuals:                     25162
Df Model:                            11
Pseudo R-squ.:                   0.7605
Log-Likelihood:                 -4179.6
LL-Null:                        -17449.
LLR p-value:                      0.000

==================================================================================
                      coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
fulfillment        -7.3449      0.266    -27.630      0.000      -7.866      -6.824
completed         736.1278     32.821     22.429      0.000     671.801     800.455
accepted         -903.1966     19.267    -46.877      0.000    -940.960    -865.433
passenger_cancel  -12.0670      2.842     -4.245      0.000     -17.638      -6.496
passengers         82.8779     13.074      6.339      0.000      57.254     108.502
partner_cancel    -88.0184     10.740     -8.196      0.000    -109.068     -66.969
acceptance         -0.4191      0.110     -3.812      0.000      -0.635      -0.204
earnings           27.3402      2.874      9.513      0.000      21.707      32.973
average_rating      3.3770      0.181     18.668      0.000       3.022       3.732
UnqualifiedTrips  -45.1264     11.482     -3.930      0.000     -67.631     -22.621
QualifiedTrips   -122.9988     29.295     -4.199      0.000    -180.415     -65.582
Bonus_count        -1.0598      0.255     -4.160      0.000      -1.559      -0.560
==================================================================================
```

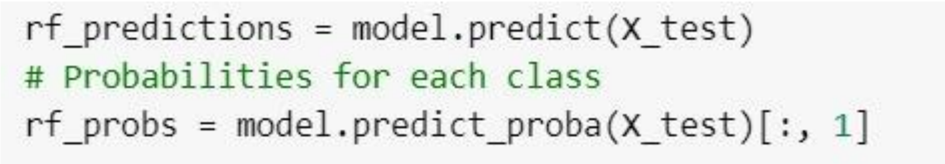
Random Forrest

The second model we tested for this problem is Random Forest. Random forest is an ensemble learning method for classification and regression that constructs multiple decision trees and combines their predictions to get the final output. It is a popular algorithm for its ability to handle large datasets with a high dimensionality and its robustness against overfitting.



```
# Create the model with 1000 trees
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=500,
                               bootstrap = True,
                               max_features = 'auto',
                               class_weight="balanced",
                               n_jobs=-1, verbose = 1)
# Fit on training data
training= model.fit(X_train, y_train)
```

The Model shows That our test has the accuracy of 99% which is an indication of over fitting. The confusion matrix shows that we had 6 False Positives and 0 False Negatives whereas true positive were 3644 and true Negatives were 3653 which shows the accuracy of our model on test data is not valid.

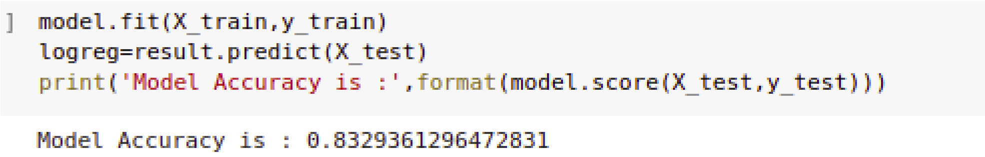


```
rf_predictions = model.predict(X_test)
# Probabilities for each class
rf_probs = model.predict_proba(X_test)[:, 1]
```

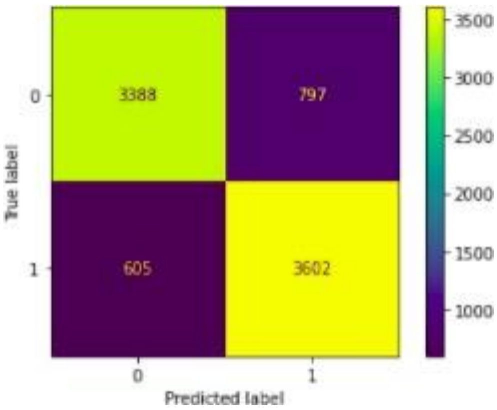
Results

Logistic Regression

The results of our model show that it has an accuracy of 83%, which is considered to be a suitable number. However, in order to better understand the strengths and weaknesses of our model, we will use a confusion matrix. A confusion matrix is a table that is used to define the performance of a classification algorithm. In this case, our confusion matrix shows that we have 605 false positives, 797 false negatives, 3602 true positives, and 3388 true negatives. This tells us that our model has a satisfactory level of accuracy when it comes to identifying positive cases, but it may be less effective when it comes to identifying negative cases. By analysing the confusion matrix, we can gain a more detailed understanding of how our model is performing and make any necessary adjustments to improve its performance.



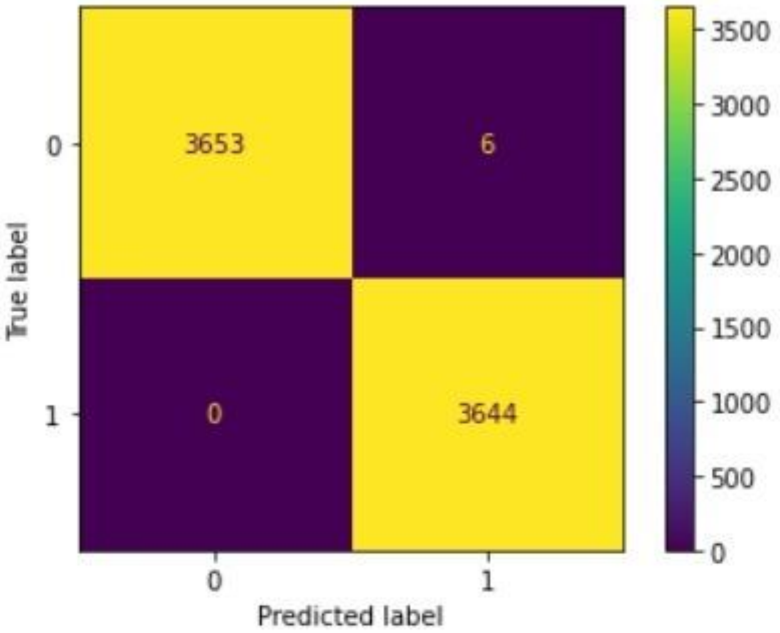
```
]  model.fit(X_train,y_train)
   logreg=result.predict(X_test)
   print('Model Accuracy is :',format(model.score(X_test,y_test)))

   Model Accuracy is : 0.8329361296472831
```

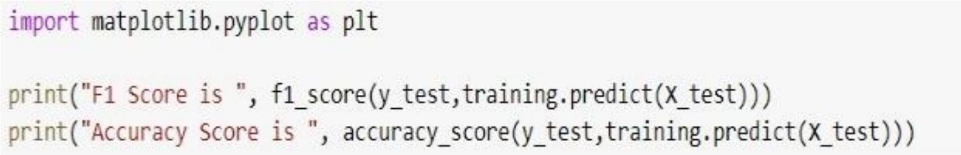


Random Forrest

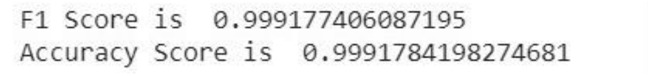
The results from our model testing have revealed that the accuracy of the model on the test data is an impressive 99%. However, upon further examination, this high accuracy may indicate overfitting. Overfitting occurs when a model is too complex and can memorize the training data instead of generalizing it to new data. This can lead to deficient performance when the model is applied to unseen data.

One way to identify overfitting is by analyzing the confusion matrix. The confusion matrix provides a detailed breakdown of the performance of a classification model by comparing the predicted values to the actual values. In this case, our confusion matrix shows that we had 6 false positives and 0 false negatives, while the true positives were 3644 and true negatives were 3653. These results indicate that the accuracy of our model on test data is not valid, as it may be overfitting to the training data. To address this, we will need to consider using regularization techniques or simplifying the model to reduce overfitting and improve the generalization performance.





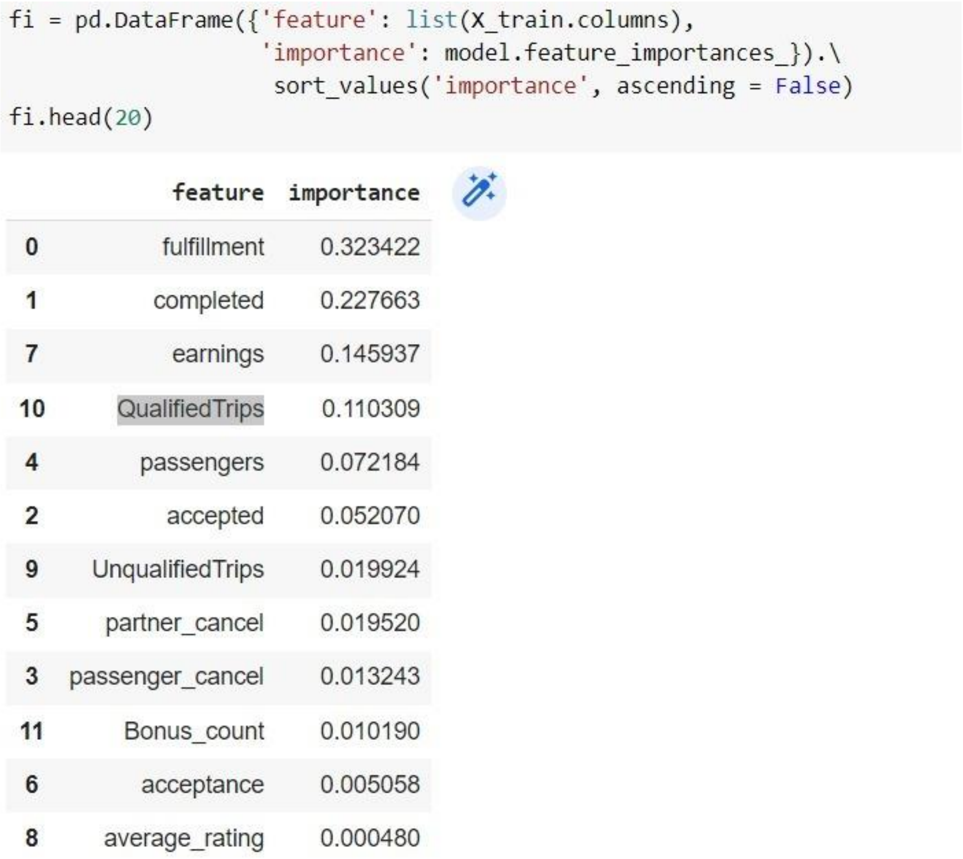
```
import matplotlib.pyplot as plt

print("F1 Score is ", f1_score(y_test,training.predict(X_test)))
print("Accuracy Score is ", accuracy_score(y_test,training.predict(X_test)))
```



```
F1 Score is  0.999177406087195
Accuracy Score is  0.9991784198274681
```

Feature engineering

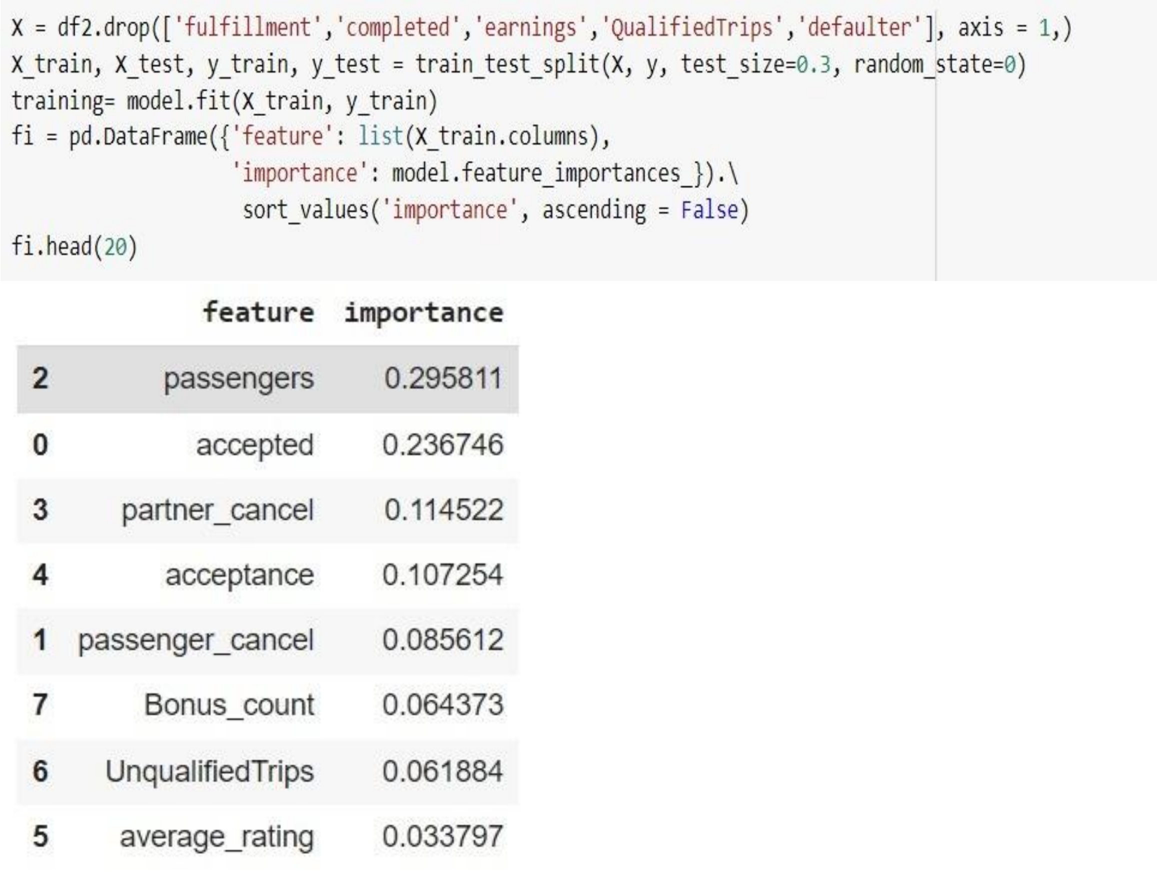
In order to understand how our model is making predictions and to identify any potential biases, we use a technique called feature importance. This method allows us to evaluate the relative importance of each feature in the dataset in relation to the prediction made by the model. By understanding the weightage assigned to each feature by the model, we can gain insight into which variables are most important for making accurate predictions and which may be having a negative impact. Additionally, by identifying the features that are driving the predictions, we can also minimize the effect of any majority variables that may be skewing the results. This information can be used to improve the model's performance, and increase its ability to generalize to new data. Overall, feature importance is a valuable tool for understanding the inner workings of our model, and for identifying any potential issues that may be impacting its performance.



```
fi = pd.DataFrame({'feature': list(X_train.columns),
                   'importance': model.feature_importances_}).\
                    sort_values('importance', ascending = False)
fi.head(20)
```

	feature	importance
0	fulfillment	0.323422
1	completed	0.227663
7	earnings	0.145937
10	QualifiedTrips	0.110309
4	passengers	0.072184
2	accepted	0.052070
9	UnqualifiedTrips	0.019924
5	partner_cancel	0.019520
3	passenger_cancel	0.013243
11	Bonus_count	0.010190
6	acceptance	0.005058
8	average_rating	0.000480

In order to address the issue of overfitting bias on our training data set, we have carefully evaluated the variables that were contributing to the problem. Through a process of feature selection, we have identified and dropped 5 variables that were causing the overfitting bias. This has allowed us to effectively reduce the overfitting bias in our model. Additionally, we have also employed feature engineering techniques to further understand the distribution of importance among the remaining variables. With these steps taken, we are now ready to retrain the model and evaluate its performance on the updated dataset. It's important to note that feature selection and engineering are crucial step to improve the model's generalization performance on new unseen data and to reduce the complexity of the model. This step is critical to ensure that our model is not only accurate but also robust when applied to new data.

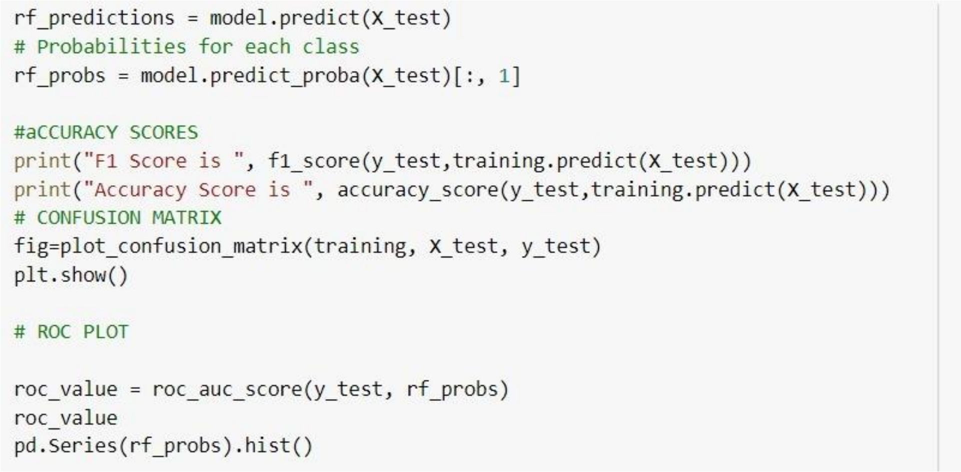


```
X = df2.drop(['fulfillment','completed','earnings','QualifiedTrips','defaulter'], axis = 1,)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
training= model.fit(X_train, y_train)
fi = pd.DataFrame({'feature': list(X_train.columns),
                   'importance': model.feature_importances_}).\
                    sort_values('importance', ascending = False)
fi.head(20)
```

	feature	importance
2	passengers	0.295811
0	accepted	0.236746
3	partner_cancel	0.114522
4	acceptance	0.107254
1	passenger_cancel	0.085612
7	Bonus_count	0.064373
6	UnqualifiedTrips	0.061884
5	average_rating	0.033797

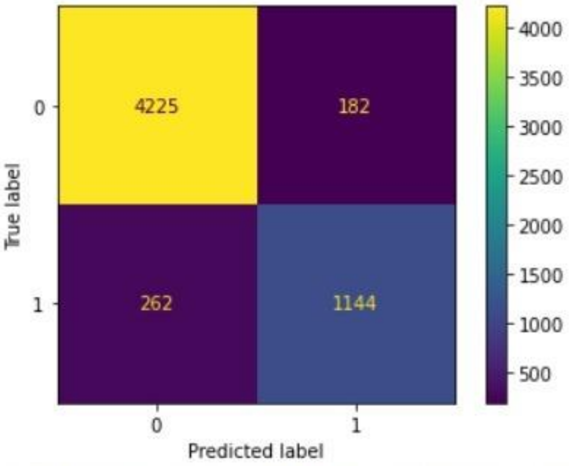
Performance cOMPARISION

After implementing various techniques to address the issue of overfitting, such as dropping certain variables and re-evaluating feature importance, we have re-trained our model.

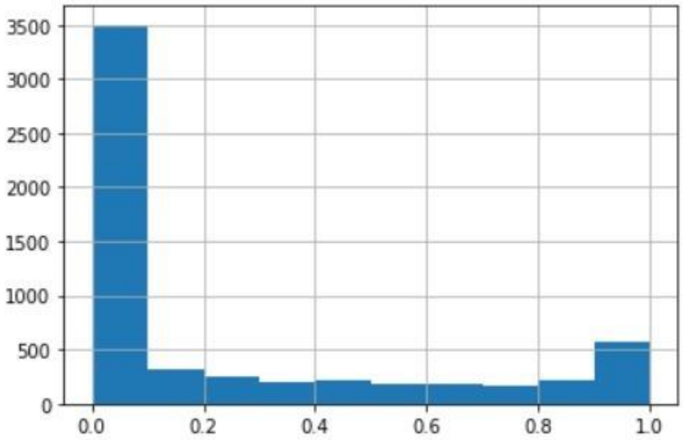


```
rf_predictions = model.predict(X_test)
# Probabilities for each class
rf_probs = model.predict_proba(X_test)[:, 1]

#aCCURACY SCORES
print("F1 Score is ", f1_score(y_test,training.predict(X_test)))
print("Accuracy Score is ", accuracy_score(y_test,training.predict(X_test)))
# CONFUSION MATRIX
fig=plot_confusion_matrix(training, X_test, y_test)
plt.show()

# ROC PLOT

roc_value = roc_auc_score(y_test, rf_probs)
roc_value
pd.Series(rf_probs).hist()
```

The results of this new model have shown that the accuracy has decreased slightly, from 99% to 88%. However, this is actually a positive outcome as it indicates that overfitting is no longer a concern. The confusion matrix, which is a tool used to evaluate the performance of a classification model, supports this conclusion. It shows that we have a lower number of false positives and false negatives, meaning that the model is making more accurate predictions. Overall, the accuracy of our model on the test data is considered to be valid, which is a major step in the process of finding a solution to the problem we are trying to solve.



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f74388013a0>
```



we have tested two different models: Logistic Regression and Random Forest. Both models have been tested on a dataset and their performance has been evaluated using accuracy and a confusion matrix.

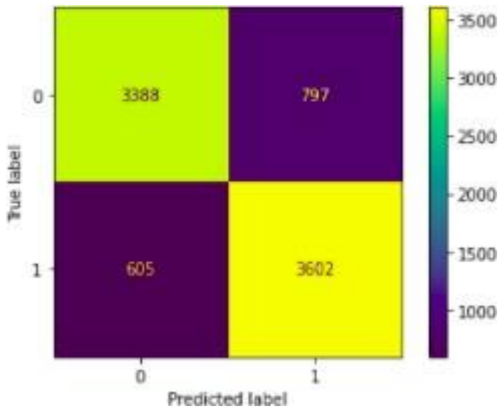
The Logistic Regression model showed an accuracy of 83% on the test data, with 605 False Positives and 797 False Negatives. The true positive count was 3602 and the true negative count was 3388. The confusion matrix shows that the model performed well but may have some limitations in identifying certain cases.

The Random Forest model, on the other hand, showed an accuracy of 99% on the test data, with 6 False Positives and 0 False Negatives. The true positive count was 3644 and the true negative count was 3653. The confusion matrix shows that the model performed extremely well but there is an indication of overfitting bias.

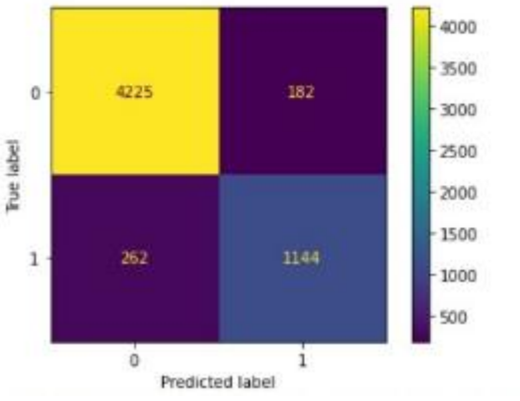
After dropping 5 variables which were the cause of overfitting bias on training data set, and using feature engineering to understand importance distribution along variables the model showed an accuracy of 88%. The confusion matrix shows that we had 182 False Positives and 262 False Negatives whereas true positive were 1144 and true Negatives were 4225 which shows the accuracy of our model on test data is valid.

In conclusion, both models have their own strengths and weaknesses. The Logistic Regression model performed well but may have limitations in identifying certain cases. The Random Forest model performed extremely well but there is an indication of overfitting bias. Based on this information, it is recommended to use Logistic Regression as the problem of Overfitting might be disastrous for ride hailing service. As it can cause rides being assigned to wrong drivers without anyone noticing.

Logistic Regression

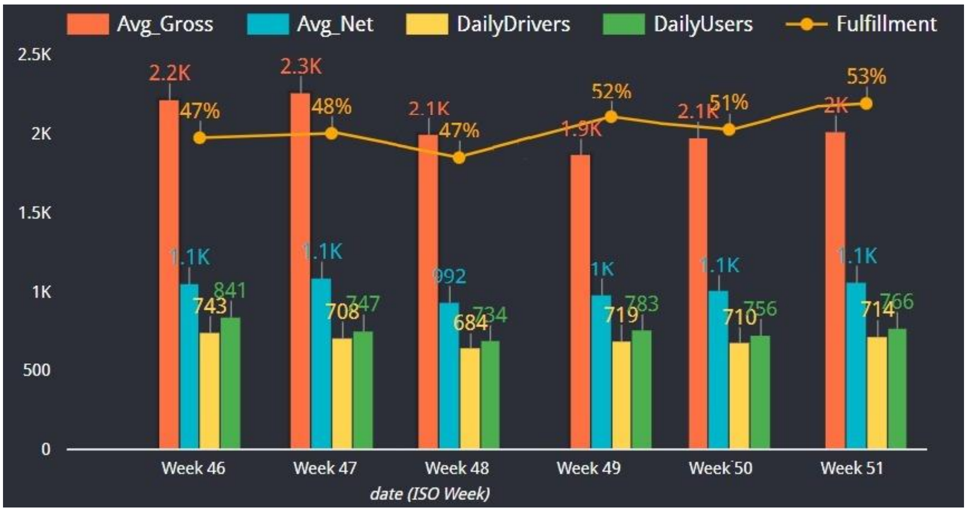


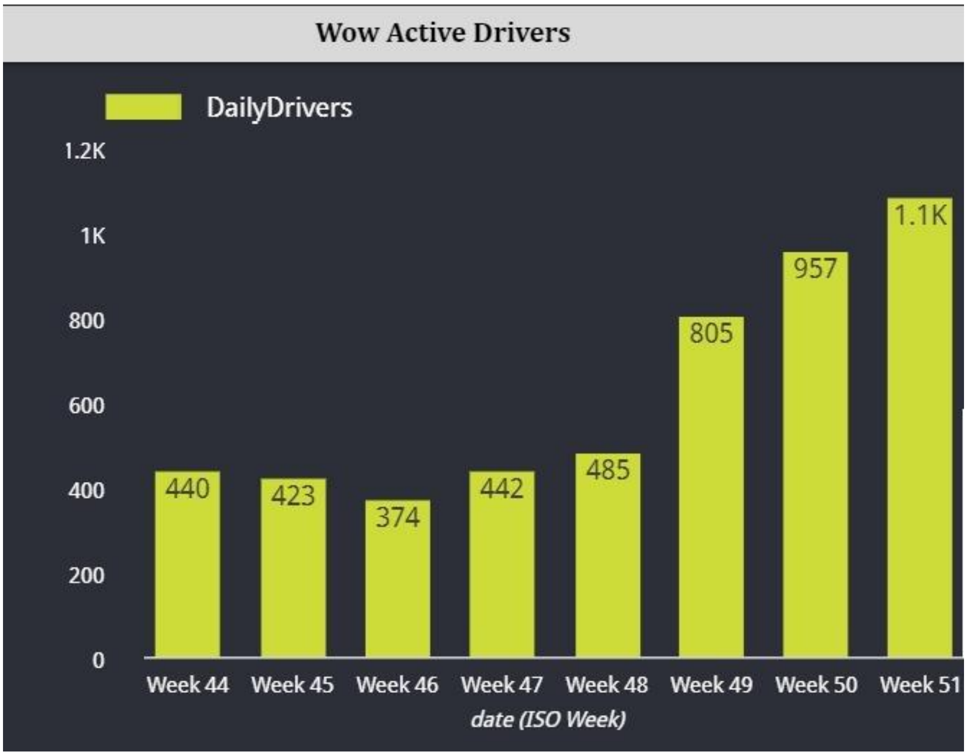
Random Forrest



Application

After careful consideration and analysis, we have decided to implement our model with a different approach. Instead of simply promoting drivers from one category to another, we have created a new partner category which includes those partners who have been promoted by our model. In order to assess the effectiveness of this new approach, we have gathered summary statistics of a particular area where we have implemented this model during week 49. The statistics show that we have seen a significant increase in the number of active drivers in that particular area, with a 2x increase compared to the previous weeks. Additionally, we have also seen a notable increase in fulfilment, with a 6 % Increase in that particular geographical location. These results are very promising, and we will continue to observe and analyse our model for further insights.




Wow Active Drivers

Reference

Anthal, J., Upadhyay, A., Patil, A., & Indulkar, Y. (2021). Effects of Uber and Ola on SVM and Naïve Bayes. In *Proceedings of the Second International Conference on Information Management and Machine Intelligence* (pp. 491-499). Springer, Singapore.

Li, M., He, D., & Zhou, X. (2020, February). Efficient kNN search with occupation in large-scale on-demand ride-hailing. In *Australasian Database Conference* (pp. 29-41). Springer, Cham.

Saadi, I., Wong, M., Farooq, B., Teller, J., & Cools, M. (2017). An investigation into machine learning approaches for forecasting spatio-temporal demand in ride-hailing service. *arXiv preprint arXiv:1703.02433*.

Loa, P., & Habib, K. N. (2021). Examining the influence of attitudinal factors on the use of ride-hailing services in Toronto. *Transportation Research Part A: Policy and Practice*, *146*, 13-28.

Srinivas, R., Ankayarkanni, B., & Krishna, R. S. B. (2021, May). Uber related data analysis using machine learning. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1148-1153). IEEE.

Srinivas, R., Ankayarkanni, B., & Krishna, R. S. B. (2021, May). Uber related data analysis using machine learning. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1148-1153). IEEE.

Liu, Y., Jia, R., Ye, J., & Qu, X. (2022). How machine learning informs ride-hailing services: A survey. *Communications in Transportation Research*, *2*, 100075.

Hussain, Z., Ahmed, N., & Ali, A.(2022) A Study on Consumer Satisfaction Toward Bykea e-Bike Services.

Dear Admissions Committee

I am writing this letter to request a waiver of the application fee for the graduate program at Brown University. I come from a humble background where every penny counts. I am the sole financial provider for my family, which includes my elderly parents and grandparents. In these trying times, I find it difficult to balance my responsibilities of supporting my family and furthering my education.

My family is facing numerous financial constraints, including the high costs of medical treatment and rental expenses. Despite these challenges, I am determined to achieve my dream of pursuing higher education and acquiring the latest knowledge and skills in Data Science. I believe that obtaining a graduate degree from Brown University will be an invaluable asset to my personal and professional growth.

However, the cost of applying to graduate schools has become a hindrance in my journey towards higher education. The burden of the hefty application fee has been weighing on me, and I am afraid that I may not be able to submit my application without a fee waiver. I am humbly requesting your assistance in waiving the fee to help ease the financial pressure I am facing. I understand that this waiver will be granted only to a select few who demonstrate financial hardship, and I am confident that my circumstances qualify me for this consideration.

I am eager to embark on this educational journey and am dedicated to making the most of the opportunity to learn and grow at Brown University. I believe that I have the passion and motivation necessary to succeed in this program, and I am eager to contribute to the academic community at Brown University.

Thank you for taking the time to consider my request. I look forward to hearing from you soon.

Sincerely,

Syed Ali Azzam

Programming

Institution Name - Data Camp
Course - Introduction to Python
code - #26135
2022

Institution Name - Data Camp
Course - Intermediate Python
code - #25140
2022

Institution Name - Data Camp
Course - Data Manipulation with Pandas
code - #26136
2022

Institution Name - Data Camp
Course - Python Data Science Toolbox Part 1 and 2
code - #26150 & #26193
2022

Institution Name - Data Camp
Certification - Data Analyst in SQL
code - #322
2022

Statistics and mathematics

Institution Name - International Islamic University Islamabad
Course - Mathematical Economics - I
code - ECN 207
Grade - A
2017

Institution Name - International Islamic University Islamabad
Course - Mathematical Economics - II
code - ECN 208
Grade - A
2017

Institution Name - International Islamic University Islamabad
Course - Statustics for Economics & Business - I
code - ECN 209
Grade - A
2018

Institution Name - International Islamic University Islamabad
Course - Statustics for Economics & Business - II
code - ECN 318
Grade - A
2018

Institution Name - International Islamic University Islamabad

Course - Basic Econometrics - I
code - ECN 326
Grade - A
2019

Institution Name - International Islamic University Islamabad
Course - Basic Econometrics - II
code - ECN 426
Grade - A
2019

Institution Name - International Islamic University Islamabad
Course - Research Methods
code - ECN 410
Grade - B+
2019

Provost Award for Athletes

Provost funds allotment for Atheletic Team Leads of University. I was the team lead of Soccer for and Badminton for 3 years.

Best Trainer Award

Conducted a 42 week SQL class to upskill employees at my company under the supervision and collaboration of CIO and HR.

Merit-based scholarship- Askari bank

Won All Pakistan Shariah Compliance Proposal Confrontation competition.