ANALYTICS FOR MOBILITY ON-DEMAND

JATIN ACHENKUNJU

Student Number - 10539553

Applied Research Project submitted in partial fulfillment of the requirements for the degree of

Master of Science – Business Analytics

Dublin Business School

Supervisor: Shahram Azizi Sazi

January 2021

Declaration

I hereby declare that the research project entitled 'Analytics for Mobility On-Demand' submitted towards partial fulfilment of requirements for the award of Master of Science in Business Analytics is a record of an original work done by me under the guidance of Mr. Shahram Azizi Sazi and the dissertation has not formed the basis for any award of any degree, associate ship, fellowship or any similar title to the best of my knowledge.

Signed: JATIN ACHENKUNJU

Student Number: 10539553

Date: 11th January 2021

Acknowledgement

I would like to express my sincere gratitude to Mr. Shahram Azizi Sazi for his continuous guidance and kind supervision throughout the course of the project. Secondly, my deep sense of gratitude to Poppy Mobility for their support. Appreciation to all the helpful colleagues at Poppy Mobility for their guidance.

I would like to express my sincere gratitude to Mr. Shahram Aziz Sazi. Last but not the least I would like to thank the students and faculty of Dublin Business School for giving me this opportunity to do a research project on 'Analytics for Mobility On-Demand' and for also proving all the necessary resources and support for my thesis.

Dublin Business School January 2021

Contents

Abstract

The primary objective of this research is to use machine learning techniques to gain useful insights from the usage data of Poppy Mobility. Using these techniques, we need to find the factors impacting the usage as well as predict the usage which will help Poppy to maintain the number of vehicles on roads and reduce the booking wait time of the customers.

Keywords— Machine Learning, Binary Classification, Boosting, Bagging, Neural Network, Keras Classifier, Prediction, Data Visualization

1 Introduction

1.1 Context

Mobility On-Demand refers to the booking of different types of mobility by the customers based on their needs as when required. With the introduction of online booking and cashless transaction, customers can book the transport and pay it online. The ease and reduction of manual work and the availability of transport is what makes it more impressive. People are shifting from the traditional modes of transport like owning a vehicle to Mobility On-Demand as they don't have to worry about the maintenance of the vehicle, insurance or the fuel. This business is growing exponentially and is expected to hit the peak of $250 billion by 2026[22].

Poppy mobility is a Belgium based Mobility On-Demand based company build using the station less service that is after usage customers can park the ride wherever they want within the given radius and there is no particular garage. In Belgium, they mainly operate in the areas of Antwerp and Brussels. Poppy has three types of vehicles namely, cars, scooters and kick scooters. In combustion cars, there are Ibiza and Corsa. Whereas in electric cars there are Citigo, MII, G-Tron and E-Golf. Pay Per Minute policy is utilized by Poppy, where the customer must pay for the actual usage of the mobility and not the stoppage time or the park time[1].

Now, the main issue faced by Poppy is that, the customer wait time for booking is getting increased when they are putting the vehicles under maintenance and they are not able to predict the exact time of this spike in usage. This increase in booking time is mainly faced by the customers booking short trips. Also, they need to understand the factors impacting the usage so that they can focus their marketing towards increasing the customers and the business.

1.2 Objective

The primary objective of this research is to utilize data analytics and data mining techniques to predict the usage of Poppy mobility where in, using this insight Poppy can maintain the availability of rides to reduce the booking wait time. Using the data mining algorithms, we can also find the important factors impacting the usage. These important factors can be used to develop key strategies which will in turn increase the customers or the business of Poppy Mobility.

This can be successfully achieved by gaining the domain and technical knowledge in the Mobility domain. Discussing with the stakeholders and understanding the hidden requirements, gathering and preprocessing the data, finding the best suitable algorithms and deciding the best evaluation metrics are the

activities which needs to performed so that Poppy Mobility can gain best insights to make sustainable business decisions.

1.3 Research Question

How are the factors impacting the usage of shared mobility? Are we able to predict the mobility usage, based on the other factors? How does Neural network like Keras perform in comparison with the other machine techniques like Boosting, Bagging and Logistic Regression?

There are a lot of short trip bookings compared to long trip bookings. In this case, it increases the average wait time for customers. At the same time, if more rides are in maintenance, this will increase the overhead of wait time. The research question is to study and predict the usage of mobility at the same time find the factors impacting the same. Our aim is also to compare the boosting and bagging techniques. Also, compare their results with neural networks like Keras Classifier and conventional Logistic regression.

1.4 Scope

This research will help Poppy Mobility to develop a business model based on the important factors impacting the usage based on binary classification of the trip.

This will reduce the customer wait time for the ride. This will enhance the quality of service and in turn gain more customers.

The approach implemented in this research can be adopted in similar business model.

1.5 Hypothesis

- Null Hypothesis: The dataset is class imbalanced.
- Alternate Hypothesis: The dataset is class balanced.

1.6 Limitation

The research focuses on binary classification of short trips and long trips using the data provided by the Poppy Mobility. The data some missing vales and dummy data which must be removed when cleaning.

Further, prediction and other analytics can be performed on the live database, once everything is moved to the actual database. As of now, few data is in the back office, which is the excel format.

2 State of the Art

This includes literature review on the research done in Mobility On-Demand domain and implementation of data mining techniques on the same.

A survey was done to understand the effect of consequences of prices and reasons of the customers for choosing Mobility On-Demand. Easily accessible using applications, insignificant maintenance, easy on pocket and can accommodate more people together are few important reasons listed by the customers for choosing OnDemand cars. People are attracted towards discount coupons and similar offers while booking a car using any application, especially the one's who are price-conscious and hence applications with more coupons is observed to gain more customers. Along with discount coupons, brand value also plays an important role and therefore companies with high brand value and more discount coupons are observed to gain a lot more customers[15]. A survey was conducted to determine prime factors that are impacting the decision of a customer[10]. In that survey the most important factor that was impacting a customer's decision was the model of the car, customers have a choice and wants to travel in a car with a model of their choice. Other important factors were price consciousness and car's condition. Customers preferred to book a ride that has a discount and also provides a car with better condition. Inventive and different services such as initial deposit and late fee were also factors that customers considered before booking a rental car[10]. Ehsan Rahimi has suggested the impact on frequent travel modes by traveller's habits, routines and predispositions[24]. It also makes it easy for us to understand the different modes of transportation and its rates. Few features that is found to have a great impact on travel modes are that customers prefer to use travel modes like walking, biking, transit and other trips. These understandings help city planners and policy makers to develop better plans based on such detailed analysis. As per final results, factors impacting travel style are age, work status, education and the build environments[24].

A study was conducted to understand the effect of factors like weather, rush hours, customers age on travel choices with the help of a dataset that has travel choices of passengers using Mobility On-Demand [34]. Predictive and behavioural analysis were performed in which it was observed that weather, peak hours and age of the customer were important factors in travel choices. It was also found that middle-aged men ordered and used more Mobility On-Demand during peak hours of a rainy day. As Random forest's accuracy was better than logistic regression and with the help of Random forest, independent factors were found based on the usage Mobility On-Demand [34]. Random forest was also used in another study which was conducted using the resourcing trip level data available in Chicago to find sources of a ride where riders will get more rides based on demand[33]. Random forest was observed to be more efficient in terms of performance compared to models like adaboost and gradient boost. It could also be used to find the best destination zone – to – zone based on demand. It was also observed that variables that gained more significance were related to travel impedance, weather and travel supply[33]. A different research was conducted to analyse users profile based on their expenditure, usage, travel and social interaction[29]. In this research key factors were deducted using Self Organizing Map algorithm. As per the research most significant factors impacting the usage are time, space, money and social interaction[29]. Strategies on personal and household variables were designed using Decision Tree algorithm. It was also observed that occupation status can be used to design strategies[29].

It has become strenuous to manage, store and perform various operations on data as its increasing day

by day. As various machine learning algorithms are applied on distributed system like Amazon S3, Spark, Mongo DB, EC2 and EMR, accuracy and the performance of these distributed systems are used as a measuring factor. Algorithms such as Logistic Regression, PCA, Random Forest and Gradient Boost were used on New York taxi and Limousine Commission's datasets. It was then observed that Logistic Regression provided the best accuracy among other models with such complex datasets and Spark performed much better than the other distributed systems[8].

Surveys were conducted using online as well as offline platforms to accumulate data which can be used to gauge the satisfaction level of the customers using machine learning models like Neural network and logistic regression[16]. It was observed price was one of the major factors that had a negative impact on the satisfaction level of the customers. More the price less is the satisfaction whereas Ownership of car had a positive impact on satisfaction level of the customers which means even though the customer owned a car, he/she was immensely satisfied with the service. Other significant factors were usability of the application and how fast can the customer book a car[16]. Deep Learning techniques such as RNN (Recurrent Neural Network), LSTM (Long Term-Short Term Memory) and different variants of LSTM were used to predict user mobility[19]. These algorithms were used as a global model or individual models and the performance was evaluated using accuracy and was observed that individual models were better than a global model.

In a ridesharing platform, matching riders and drivers in a very short time is a defining factor of the success of an application. In this research, they use Markov's chain to match the riders[5]. The matching time of Markov's chain was found to be considerably less time and gave optimal results on convergence rate of various algorithms. The model suggested that the convergence rate is quite fast, and this was proved using simulated as well as real datasets. This was compared to reinforcement algorithm, and the Markov's chain perform much better[5].

There may be instances where the driver is set to a destination and he is finding it difficult to find the new rider. To solve this problem, this research focuses on minimizing the average wait time with help of machine learning models by predicting the best destination where the driver can find more riders[7]. The dataset used here is publicly available data from RideAustin. Many models like Bayes Network, Naive Bayes, OneR, SVM, Linear Regression, Decision Tree, Stacking Meta like decision table. Best performing model was found to be Naïve Bayes with the best AUC of 0.827. Thus, it can be considered as a good predictive performance. This technique can also be used further in automated cars to predict the destination so that it can be more profitable. The best zip code will be fed to the car and it will travel to that destination[7]. Mobility On-Demand's key is to have the rental mobility available when there is demand from customers so that there is less waiting time. This study was performed on rental bikes data[23]. Bikes rented per hour and date related data was available. Furthermore, it had the data related weather like temperature, humidity, rainfall, snowfall and dewpoint to check impact of weather on bike availability. Models like Linear Regression, Gradient Boosting, SVM, Boosted trees and extreme gradient boosting. Best model was gradient boosting with an accuracy of 0.96. Thus, prediction was made based on availability of mobility based on weather, date and bikes rented per hour[23].

With the advancement of Social media, people send their current location and status with the help of GPS. A study was performed where the dataset obtained for GPS can be used to analyse the movement patterns of the customers[2]. Along with the patterns, we also identify the time spent by the people at a given location and mark them as a class in clusters. The different clustering algorithms identified are

OPTICS, K-Means, DBSCAN and Hierarchical algorithms. Once these places are identified, mark them with a with their importance. These places can be focused by the Mobility On-Demand companies to get more customers. DBSAN and K-Means performed the best [2]. A separate study was performed using Stationless car's data to understand the mobility patterns in Paris[27]. A set of features was generated to understand the mobility related patterns based the vehicle's activity. Then, clustering algorithm was performed to interpret the patterns and cluster them into vehicle type. A total of 68,613 unique vehicles were identified based on their trip patterns and were clustered into 4 main groups namely Morning, evening, long distance and frequent activity-based vehicles. In this cluster, a total of 81.9% of the vehicles were considered from the dataset. This data can be used by the company to allocate correct vehicles based on the trip ordered by the customer. This will increase the customer satisfaction[27].

In another study, Jason Van Hulse and team have performed a comparative study on boosting and bagging techniques on imbalanced and noisy data. This study was done as, in real world these two aspects namely, imbalance and noisy data are commonly found in the data. In this paper many bagging and boosting techniques were used like SMOTEBoost, RusBoost, Exacly balanced and roughly balanced bagging. For this experiment, many classifiers models were trained. Nine different evaluation metrics were used for this purpose[9]. After the comparative study it was found that bagging techniques performed way better than the boosting techniques. In our data, we can perform a comparative study between boosting and bagging techniques for binary classification. Boosting and bagging was used in another study by Shreya Batra and team for binary classification of the unbalanced data[26]. In unbalanced data, negative class dominated the positive class and the model is not able to learn efficiently which leads to misclassification of the target variable. Boosting and bagging, both are robust towards the unbalanced data which help to classify the labels which are lower compared to other classes. We can use this to our advantage to classify our target variable distance which is class imbalanced[26]

3 Methodology

3.1 CRISP DM

The main aim of this project is to find the important factors impacting the usage and predicting the usage. Distance of a trip is considered as usage. The Distance column is converted to short trip and long trip, based on the requirement. This makes it a binary classification problem. We will be using Boosting and Bagging techniques. Thereafter, comparing these with the Keras neural network and traditional Logistic regression. Logistic regression is known to provide good results for binary classification. Independent of the machine learning algorithms or the domain of the problem, we can use CRISP DM process model for the implementation of the data mining projects[32].

There are five important steps in the CRISP DM:

1. Business Understanding: Having the domain understanding is very important. In this case, its Mobility on demand. So, we need to understand the business and with help of the stakeholders, we need to convert them into a data mining problem. Here, Poppy's main problem is the customer wait time for the ride is increasing. Hence, they want the factors impacting the usage and, predict the usage. This will allow Poppy to plan the maintenance of the vehicle based on the customers usage.
2. Data Understanding: Here the data is in the Database and the back office. Understanding the data and the importance of the features with the stakeholders is vital. Visualization of the data and find the discrepancies.
3. Data Preparation: Merging the data from the different data sources and data wrangling is done in this step. This includes data cleaning and removing all the columns that are not required for this research. Here, we will the have final data that can be fed for modelling[32].
4. Data Modelling: In this step we decide the models which will be the best for binary classification of the data. We will compare the results of neural network with the conventional boosting and bagging techniques.
5. Model Evaluation: In this step we decide on an evaluation metric and find the best model. Further, with the help of the predicted model, we can visualize the data based on the business needs.
6. Model Deployment: Once, we have evaluated the best model, we need to discuss with the stakeholders and discuss the findings and the benefits of the model. If the model is approved by the management, then it can be deployed in the real world. After deployment, we need to continuously monitor the results to check if the deployed models are working as per requirement. If it does not provide the required results, we need to tweak the model to get the desired results[32].

3.2 Data Preparation and EDA

3.2.1 Data Acquisition

Poppy Mobility wants to gain insights on the usage related data. They need to know about factors impacting the usage, also predict the usage based on short or long trips. The data was available from two data sources namely, database and back office. Below data was retrieved from by joining the tables from database:

- Availability of Mobility
- Trip Duration

Below details were retrieved from back office:

- Trip data
- Customer Information

3.2.2 Data Pre-processing and feature engineering

We start with individual cleaning of the csv files retrieved from the database and back office. Now, we merge the files. To merge the file, we have trip_id column common in both the files. We, merge both the files using this column. After merging we remove all the columns which are not needed for analytics. These columns might have a unique id like trip id, customer id, vehicle plate number or customer information like name, email or address. Next, we remove the missing values from data. Mainly the missing vales are present in the age and distance columns.

We remove the rows if the status column has the value 'SUSPENDED'. This means that the trip got suspended due to some reason and this will not add any value to the analytics. Further, we remove the rows if the Cancelled_ride column has the true value. This means that ride was cancelled by the customer after booking the same. Also, we have removed the rows, if the archived column has a true value, this signifies that it is dummy data added by Poppy mobility for testing purpose. We remove rows which were added due to technical issue. If the distance is distance is less than 200, it is a technical issue and the row must be removed. In the gender column we have a few data where gender is undefined. We need to remove that as well.

Now, in feature engineering we add some columns. We add the age column and calculate the same based on date of birth. Next, we need to a column to find if the trip date was a holiday or not. We use the holidays library for this purpose. These are specific to Belgium. Using this library, we have the information if the trip date was a holiday or not. The duration of the trip is divided into the actual trip duration, booking duration and pause duration. To get the actual duration the booking duration, pause duration must be removed from the actual trip duration. This will be the new column named as distance_new.

We add few columns which will be helps for deeper analysis after the modelling with visualization and these columns will not be a part of modelling. Vehicles types are deducted from the name column. It is divided into Scooter, Kick scooters and Cars. Further, the cars are divided into Electric cars and Combustion cars. We create one more asset column where scooters and kick scooters are considered as soft assets.

Finally, we convert the target variable distance, as per requirement to short trips and long trips. The trips above 4000 m are considered as long trips.

Further, we plot the correlation heatmap using the plotly library to check if high correlation exists between the variables.If high correlation exists, we check for causation between the variables. If causation exists between these variables, we can remove one of the variables. Now, after plotting the correlation, we find the high correlation exists between zone_start and city, zone_start and zone_end. City is where the customers reside, which mostly same as the zone_start. Riders will be booking the ride from their location of residence. Hence, even causality is high been the zone_start and city. There is also high causality between the zone_start and zone_end. hence, we will remove zone_end and city.

3.3 Related Theory

3.3.1 Predictive classification modelling

In classification algorithm, we train the dataset and predict the target class. The important task here is to get the boundary conditions to predict the target class accurately[6]. The entire process of determining the boundary conditions and then predicting the target class is known as predictive classification modelling[28]. Below are the terminologies in classification modelling:

- Classifier: It is an algorithm that Maps input data set to a target class.
- Classification model: The classification model involves the entire process of training the data set that is getting the boundary conditions as well as predicting the target class. Firstly, it trains itself from the input data provided and then using the new input data it predicts the target class[28].
- Feature: It is a measurable label in a data set.
- Binary classification: In this classification, input data is mapped to two target classes.
- Multi class classification: In this classification, there are more than two target classes[6]. The input data is mapped to one of the target classes.
- Multi label classification: In Multi label classification, there the input data can be mapped to more than one more than one target classes[28].

3.3.2 Boosting Techniques

This technique involves using of weak learners and converting them into strong learners. This problem can be well understood with the help of an example[30]. Let's look at the spam identification message problem:

1. If the message has a sentence like "Win a prize money worth", it is a SPAM.
2. If the message has more than 4 colours used, it is a SPAM.
3. If the message has an advertisement image, it is a SPAM.
4. If the message is from an official domain, it is not a SPAM.
5. If the message is from a credible source, it is not a SPAM.

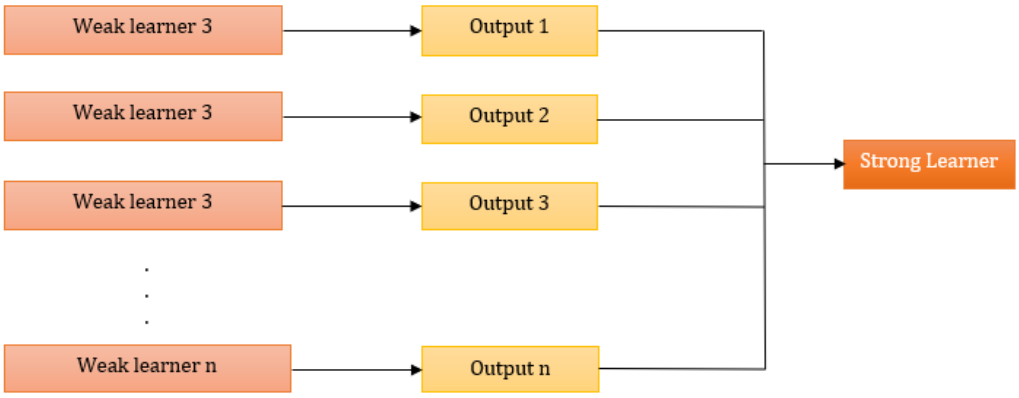


Figure 1: Boosting Algorithms

Thus, these are rules defined to categorize if the message is SPAM or NOT. But just these rules are not enough to classify the given message is a SPAM or not. This is the reason why these rules are known as weak learners. Boosting technique converts weak learner to strong learner. For this purpose, it uses Voting method. So, for example, if it gets 3 votes that it is a SPAM and two votes that it is not a SPAM, it will be classifies as a SPAM based on the maximum SPAM votes[30].

Thus, it combines the prediction of each learner and combines them to get a strong leaner and prediction. It is based on the below methods like:

1. Using Average or weighted average
2. Using maximum votes in a prediction.

The basic principle of boosting is combining weak learners prediction to acquire a strong rule. These weak rules are generated by applying machine learning algorithms on different dataset portions. One weak rule is generated in each iteration. Each of their rule is combined to get a strong rule[31]. Below are steps of boosting algorithm:

Step 1: Initially, the algorithm assigns each observation with equal weights.

Step 2: After classifying the it into target variables, it identifies the misclassified observation. Now it assigns higher weights to these misclassified observations and these weights along with the new data is fed as the next iteration[31].

Step 3: Second step is repeated until the algorithm classifies all the observations correctly.

3.3.3 Adaboost

Adaboost is one of the ensemble boosting classifiers where it combines the low performing classifiers to get a high performing classifier to gain high results. Any algorithm can be used as a base classifier[13]. The idea behind Adaboost is, it assigns equal weights to the observations and after each classification it focuses on misclassified observation. Below are the algorithm steps:

1. It selects training subsets from the training data randomly.
2. It trains the Adaboost model in each iteration based on the predictions made by the last classifier by selecting portions of the training subset.

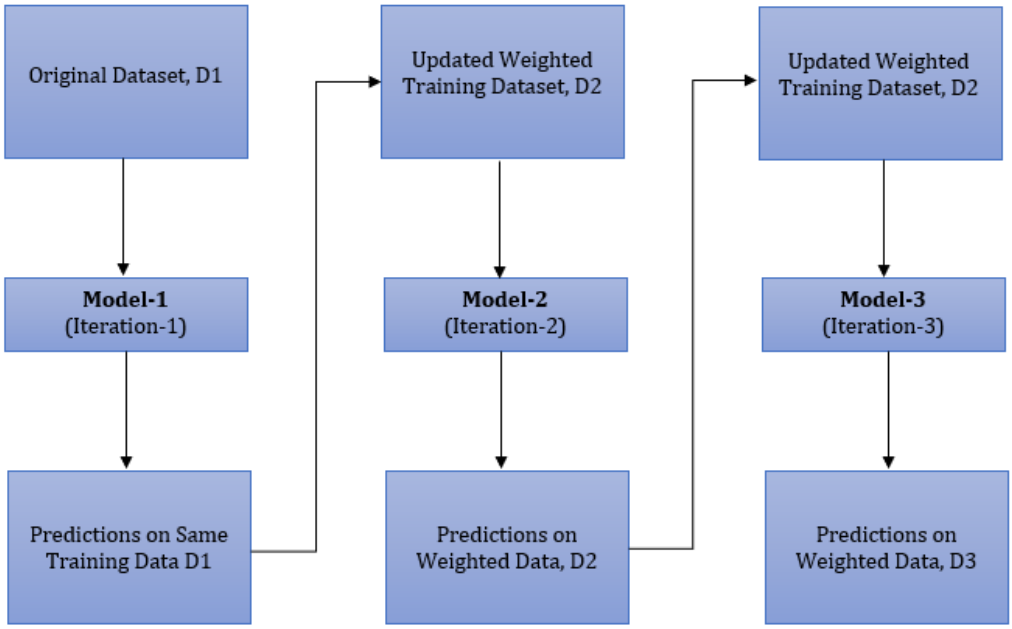


Figure 2: AdaBoost Algorithm

3. It assigns higher weight to misclassified observation and so that it will have higher probability of getting classified accurately in the next observation[13].
4. It also assigns weights to the classifiers based on the accuracy. Higher the accuracy, higher is the weight.
5. The iterations will continue till all the observations in the training data are correctly predicted.
6. Now, in classification, it performs vote for each observation across all the algorithms built.

Below is the mathematical equation behind Adaboost[18]:

$$x_i \epsilon R^n, y_i \epsilon \{-1, -1\}$$

Here,

n is the total number of features in the dataset

x is the set of data points

y is the target variable, which can be either -1 or 1 as it is a binary problem.

Higher the weight, more is the say the training data would have on the final dataset. If the weights are low, they have least influence in the training dataset[18]. Initially, all data points have same weights w:

$$w = 1/N \epsilon \{0, 1\}$$

Here,

N is total data points.

Now, we calculate the actual influence of classifier in classifying the observations correctly. It is given by:

$$\alpha_t = \frac{1}{2} ln \frac{1 - TotalError}{TotalError}$$

Alpha would be the actual influence of classifier on the final prediction. Total error would be the total misclassified observations divided by total size of training dataset. Lesser the error rate, high will be the actual influence[18].

3.3.4 Gradient Boost

Gradient boost is another boosting method. It mainly depends on weight minimization. In this boosting method, wherever a new learner is added the weight of the previous learners are frozen and kept unchanged. This is main step which helps gradient boost to have an upper hand over Adaboost. The classifier depends on a loss function and custom loss function can be used[21]. It depends on three main elements:

1. Loss function: The type of the problem defines the loss function to be used. Importantly, it must be differential but even custom loss functions can be used. As our problem is classification problem, we will be using logarithmic loss. The main benefit of gradient boost that sets it apart from Adaboost is that new loss function need not be derived on the introduction of new algorithm. As a matter of fact, any differential loss function can be used[17].
2. Weak learner: Trees are built using the best split based on the purity scores. Decision trees are used as the weak learners in gradient boosting.
3. An additive model: In each iteration, a tree is added and existing trees in the model are not changed[17].

The mathematical function is given by[21]:

$$g_t(x) = E_y[\frac{d\psi(y, f(x))}{df(x)}|x]$$

Where,
psi (y,f) is the loss function
t is each iteration
E is the expected loss function.

3.3.5 XGBoost

XGBoost uses the gradient boost framework and uses the decision tree-based ensemble technique. It is a combination and software and hardware optimisation to gain the best results in a short amount of time. XGBoost is said to have one of the best model performances. Below is the mathematical equation of XGBoost[14]:

$$L^{-t} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

Where,

t is the iteration.

f(x) is the loss function

Below are the main features of XGBoost:

Regularized learning: It avoids overfitting by smoothing the final learnt weights, this helps XGBoost to gain more accurate results[14].

Gradient Tree Boosting: The models are added in an additive model and model can be optimised using the normal optimization methods.

Shrinkage and Column Sampling: It uses further two methods to prevent overfitting. Once the weights are added, shrinkage scales these weights by a factor. Shrinkage is important as it allows the trees which will be added in future to improve the model. Column sub sampling speeds up the computation than the traditional row sub sampling. This reduces the overfitting even further[14].

3.3.6 Light Gradient Boosting Method [LGBM]

LGBM and XGBoost both use the leaf wise growth strategy, while accessing the next leaf in the decision tree. While splitting the data or training the DTree, there are two methods of accessing the nodes namely, level wise growth strategy or leaf wise growth strategy[25].

Leaf wise growth accesses the leaf with least loss whereas level wise growth accesses the node in a length wise balanced manner. Leaf wise strategy is more efficient than level wise because we can construct a level wise node access using a leaf wise strategy, but we cannot do vice versa. Leaf wise training is more prone to overfitting, but it is still more efficient and flexible.

Not all the data points or features contribute to the training of the model equally. Only the features with smaller gradient contribute more. This means that, to make the model more efficient we need to concentrate on the features with larger gradient. But we cannot directly ignore the datapoints with least gradient which will lead to biased sampling. In order to mitigate this, LGBM randomly samples the data with smaller gradients. This will however lead to a bias towards data with higher gradients, but it will give higher weights to the data points with small gradients[25].

Tree takes a lot of time to split, which is the reason it takes time to build a tree. Histogram based splitting splits the features into bins instead of keeping them in features. Now, even before the building of the tree, the features are binned which highly speeds up the training process[25].

The main advantages of LGBM are[25]:

1. High training speed
2. Better accuracy and high efficiency
3. It uses low memory
4. It can handle bulk of data or heavy data.

3.3.7 Bagging Algorithm:

Bagging is a very efficient ensemble method. Ensemble method is where it is combining multiple machine learning algorithms to increase efficiency rather than using one model to get inaccurate results. Decision trees which have high variance, bagging technique can be used to reduce the variance considerably. Decision trees are sensitive to specific part of the training dataset. Suppose we gain the prediction to our target variables from a part of the dataset and we use different chunk of the training dataset, it will give very different results[20].

When using bagging with decision trees, we are least concerned about overfitting of the individual trees. Hence, we allow the trees to grow and not prune them. These trees will have low variance and high bias. The important thing here is to get the number of trees or number of samples right. This can be done by hypertuning the parameters and deciding on the best one based on the results[20].

3.3.8 Random Forest Algorithm:

Ensemble combines the algorithm to get better results. It works more efficiently when the models have less correlation or are loosely correlated. Random forest produces the results in such a way that the predicted results of the models are loosely correlated[12]. Gini index is used to determine the number of nodes on a decision branch and it is given by:

$$Gini = 1 - \sum_{i=1}^{c} (p_i)^2$$

Pi is the relative frequency if the class being observed

c is the total number of class

Other technique to use is Entropy to decide the number of nodes on a decision branch.

$$Entropy = \sum_{i=1}^{c} -p_i * log_2(p_i)$$

Entropy uses probability of certain outcome in order to come up with the decision of number of nodes. The log function in the Entropy helps it to decide not more mathematically accurate.

Below are the steps of Random forest[12]:

1. Firstly, select random samples from the training data set
2. Now the algorithm will create decision tree for sample and provide a result or a prediction for each decision tree.
3. Next step is voting for each predicted result
4. Now the algorithm counts the votes and class is decided based on maximum votes.

Complexity and time consumption are the main disadvantages of random forest algorithm. Reducing the variance and overfitting compared to a single decision tree are one of the greatest advantages of random forest algorithm.

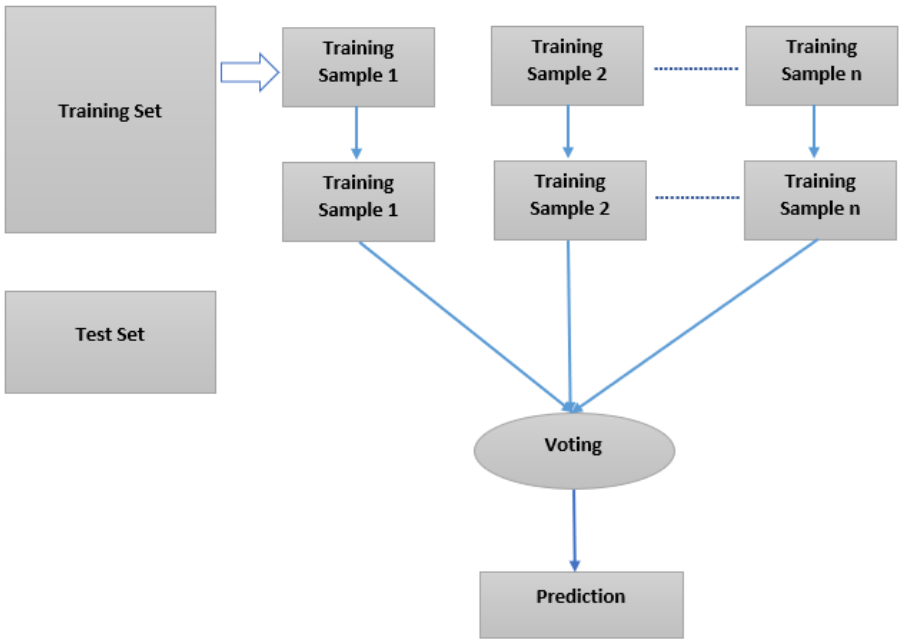


Figure 3: Random Forest

3.3.9 Logistic Regression:

The name logistic regression comes from the logistic function which happens to be the core of this algorithm. It is also called as the sigmoid function, which is an S shaped curve. It can classify a real number value to 0 or 1. The function is given by[4]:

$$\frac{1}{1 + e^{-value}}$$

Where,

e is Euler's number or base of natural logarithms

value is the actual number which we want to classify.

Below is an example of Sigmoid function and plotting the same into the range of 0 and 1.

In logistic regression, the input values (x) are combined with weights and fed to the function to predict the value of y. The main difference between the logistic regression and linear regression is that in logistic regression, the model classifies the input variables to values 0 or 1 whereas in linear regression, model classifies the input variables to a numerical value[4]. The mathematical function of the logistic regression classifier is given by:

$$y = \frac{e^{b0+b1*x}}{1 + e^{b0+b1*x}}$$

Where,

y is the predicted output

b0 is the intercept

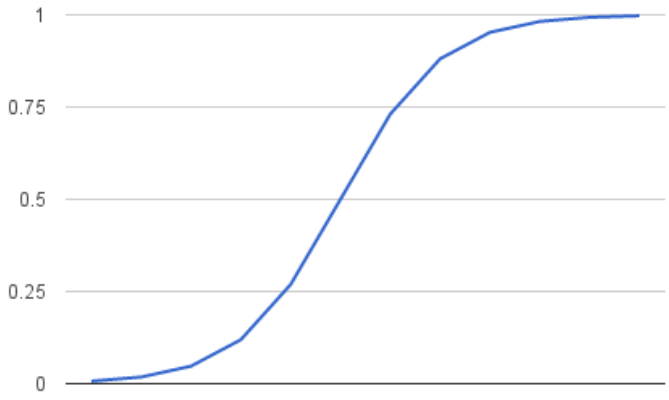


Figure 4: Sigmoid Function

b1 is the coefficient of single input value x

The coefficient of the input value is estimated with the help of the training data. This estimation is done with the help of Maximum Likelihood Estimation. These coefficients that are determined by the maximum likelihood estimation, would result in a model that would predict the target variable close to 1 or 0[4].

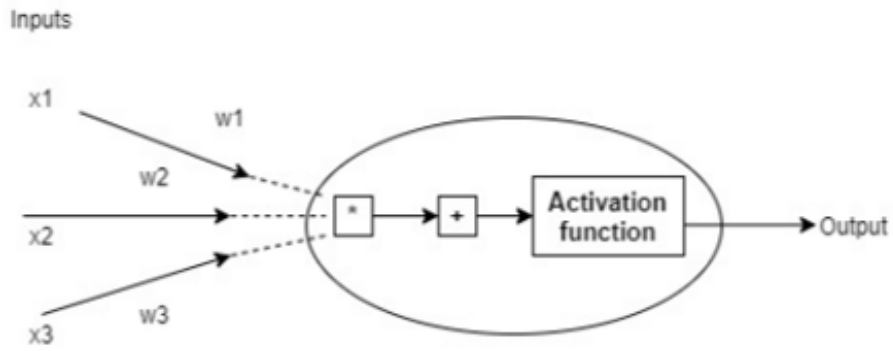


Figure 5: Neural Network

3.3.10 Neural Network:

The name neural network has come from the neurons in the brain of human beings. As the neurons in the brains receive the message, in the same way neurons or nodes in the neural network are fed with the input variables. The multiple input variables act like dendrites and output act as axon and as soon as it receives an activation trigger, it reaches a threshold and passes the signal to the next neuron and it creates a complete network[11]. A neuron with three input variables will be as given below:

Below are steps performed:

1. Firstly, inputs are multiplied by their weights.

$$x_1 = x_1 * w_1$$

$$x_2 = x_2 * w_2$$

$$x_3 = x_3 * w_3$$

2. The weighted input received from the first step are added with a bias.

$$(x_1 * w_1) + (x_2 * w_2) + (x_3 * w_3) + Bias$$

3. The sum received from the second step is passed through an activation function.

$$y = f((x_1 * w_1) + (x_2 * w_2) + (x_3 * w_3) + Bias)$$

In the traditional machine learning algorithms, we train the system to process and learn from the data where in deep learning, it trains itself to process and it learns from the data. This is possible with the help of Artificial Neural network. Traditional neural network has just three layers. When there are more than three layers, an input and output and multiple hidden layers, it is called as deep neural network[11].

3.3.11 Keras Neural Network:

Keras is python library in machine learning which uses efficient libraries like TensorFlow and Theano. Below are the steps of Keras[3]:

1. Firstly, we decide the shape of the input variables.
2. Define the model based.
3. Then we pass the input variables to first layer consisting of some cells. These layers built using sequential model.
4. Multiple layers are built based on the data.
5. Compile the model and choose the best optimiser suited for your data.
6. Use binary_crossentropy as loss function as our problem is binary classification.
7. Train and fit the model based on epoch and batch size received from hypertuning the parameters using GridSearchCV.

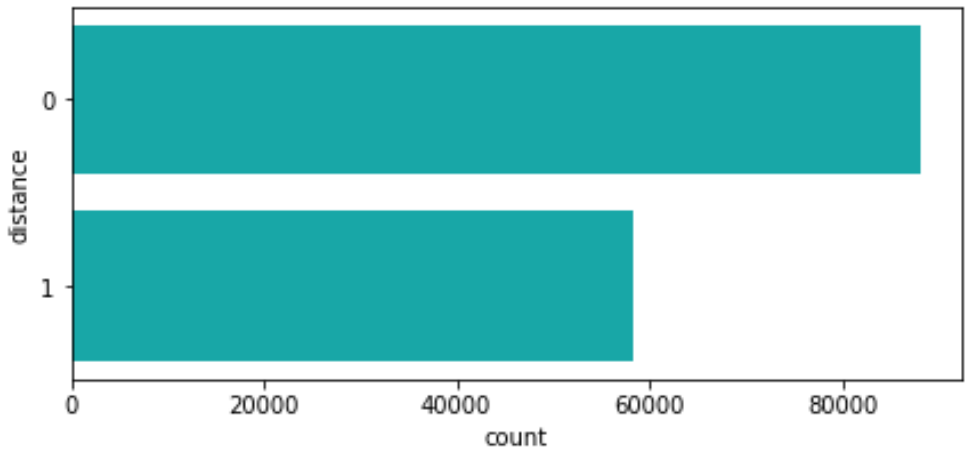


Figure 6: Class Imbalance

4 Implemetation

4.0.1 Train – Test Split:

Before fitting the machine leaning algorithms on the dataset, the dataset must be divided into test and trainset. We divide the dataset into 70%-30% split. After the split, the training dataset has 102457 rows and the test set has 43911 rows.

4.0.2 Handling Class Imbalance:

As we had discussed above, we had transformed the target variable distance to binary. As we had assumed in the hypothesis, if the dataset is class balanced we reject our Null hypothesis. But, if the dataset is class imbalanced, we cannot reject out Null hypothesis. After transformation, we find the target variable was class imbalanced. We visualize the same in the above graph using seaborn library for plotting. Thus, we accept our Null hypothesis.

There is a class imbalance, as in the target variable total number of 1s are 61,557 and 0s are 40,900. A class imbalance data can lead to overfitting and can severely impact the results.

Hence, it is vital to handle the imbalanced data by either over-sample the minority data or downscale the majority data. Simple way to deal with this issue is to create duplicates of the minority data. It does not add any new data, but it creates synthesized examples. This is also called as Synthetic Minority Oversampling Technique or SMOTE. After the application of SMOTE the observations of each class are as follows:

$$1 = 61557$$

$$0 = 61557$$

duration_new	0.797510
age	0.119573
vehicleModel_OKAI Kickscooter	0.023530
vehicleModel_SCOOTER	0.020224
vehicleModel_IBIZA - All	0.010342
vehicleModel_CORSA - Brussels	0.007781
vehicleModel_MII - Antwerp	0.004055
vehicleModel_CITIGO - Antwerp	0.002901
zone_start_Antwerp	0.002441
zone_start_Brussels	0.002112
is working_True	0.001557
gender_FEMALE	0.001527
is working_False	0.001510
gender_MALE	0.001508
vehicleStatus_maintenance	0.001150
vehicleStatus_available	0.001143
zone_start_Zaventem	0.001023
zone_start_Charleroi	0.000114

Figure 7: Random Forest - Important Features

4.0.3 Model Evaluation Criteria:

- At Poppy's, short distance trips occur more often than a long-distance trip. As people book rides more for short distance trips there is a shortage of rides and this increases the customers wait time. Also, it happens that vehicles are on routine maintenance which adds to the overhead of wait time.
- Hence, in order to reduce the wait time, short distance trips must be predicted correctly, or we need to reduce the incorrect classification of the short distance trips. For this purpose, we will be focusing on reducing the false negative. Firstly, it trains itself from the input data provided and then using the new input data it predicts the target class.
- As we are focusing on predicting the short distance correctly, in order to reduce the wait time, we can ignore the false positives.
- The model will least false negative, will be best model and can be deployed in real world.

4.0.4 Implementation of Random Forest

- Grid search was used to find the best parameters and hyper tune the parameters. We are using cross validation technique and here we are using 5-fold cross validation.
- After hypertuning we found the best value parameter to be 600.
- We are focusing on reducing the false negatives throughout the implementation; hence we have used scoring parameter as recall.
- Now, we train the model by fitting the training data and find the important features. Above are important features.
- Then, we finally use the predict function and predict the target variable. Below is the confusion matrix of this prediction.
- Here, we have 3133 false negatives. Also, we have used all the variables. We will further try to reduce the false positives by using the top variables from the feature importance.

```
Confusion matrix:
 [[22963  3516]
 [ 3133 14256]]
TP:  14256
TN:  22963
FP:  3516
FN:  3133
```

Figure 8: Random Forest - Confusion matrix

```
Confusion matrix:
 [[22984  3495]
 [ 3132 14257]]
TP:  14257
TN:  22984
FP:  3495
FN:  3132
```

Figure 9: Random Forest - Confusion matrix (Best Features)

- After using the top 16 variables from the dataset, we were able to further reduce the false negatives to 3132 and this is the optimum Random forest model. Above is the confusion matrix.

4.0.5 Boosting Algorithms

4.0.6 Adaboost Implementation

- Grid search was used to find the best parameters and hyper tune the parameters. We are using cross validation technique and here we are using 5-fold cross validation.
- After hypertuning we found the best value parameter to be 6.
- Now, we train the model by fitting the training data and find the important features. Below are important features.
- Then, we finally use the predict function and predict the target variable. Below is the confusion matrix of this prediction.

```
duration_new                    0.833333
vehicleModel_OKAI Kickscooter   0.166667
is working_False                0.000000
gender_FEMALE                   0.000000
gender_MALE                     0.000000
vehicleModel_CITIGO - Antwerp   0.000000
vehicleModel_CORSA - Brussels   0.000000
vehicleModel_IBIZA - All        0.000000
vehicleModel_MII - Antwerp      0.000000
is working_True                 0.000000
vehicleModel_SCOOTER            0.000000
vehicleStatus_available         0.000000
vehicleStatus_maintenance       0.000000
zone_start_Antwerp              0.000000
zone_start_Brussels             0.000000
zone_start_Charleroi            0.000000
zone_start_Zaventem             0.000000
age                             0.000000
```

Figure 10: AdaBoost - Important Features

```
Confusion matrix:
 [[23676  2803]
 [ 2739 14650]]
TP:  14650
TN:  23676
FP:  2803
FN:  2739
```

Figure 11: AdaBoost - Confusion martix

duration_new	0.970626
vehicleModel_OKAI Kickscooter	0.007974
age	0.006704
zone_start_Brussels	0.003178
vehicleModel_MII - Antwerp	0.002443
vehicleModel_SCOOTER	0.002247
vehicleModel_CITIGO - Antwerp	0.001867
gender_MALE	0.001746
gender_FEMALE	0.001440
vehicleModel_CORSA - Brussels	0.000575
zone_start_Zaventem	0.000316
vehicleModel_IBIZA - All	0.000314
is working_False	0.000194
is working_True	0.000185
zone_start_Antwerp	0.000144
zone_start_Charleroi	0.000033
vehicleStatus_available	0.000016
vehicleStatus_maintenance	0.000000

Figure 12: Gradient Boost - Important Features

- Here, we have a total of 2739 false negatives. AdaBoost is giving better results than Random Forest. Boosting techniques are giving a better result than bagging techniques for our data.

4.0.7 Gradient Boost Implementation

- Grid search was used to find the best parameters and hyper tune the parameters. We are using cross validation technique and here we are using 5-fold cross validation.
- The hypertuning parameters here are max_depth, max_leaf_nodes and n_estimators. We received the best parameters as 11, 24, 50 for max_depth, max_leaf_nodes and n_estimators respectively.
- Now, we train the model by fitting the training data and find the important features. Above are important features.
- Then, we finally use the predict function and predict the target variable. Below is the confusion matrix of this prediction with 2636 false negatives.
- Gradient Boost have even better results than Adaboost algorithm.

```
Confusion matrix:
 [[23775  2704]
 [ 2636 14753]]
TP:  14753
TN:  23775
FP:  2704
FN:  2636
```

Figure 13: Gradient Boost - Confusion Matrix

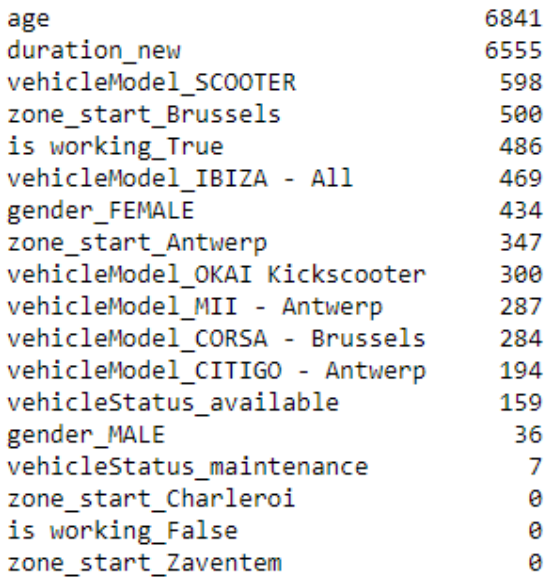


Figure 14: Light Gradient Boost - Important Features

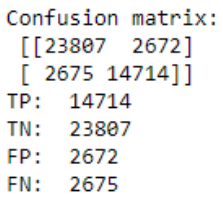


Figure 15: Light Gradient Boost - Confusion Matrix

4.0.8 Light Gradient Boost Implementation

- Grid search was used to find the best parameters and hyper tune the parameters. We are using cross validation technique and here we are using 5-fold cross validation.
- The hypertuning parameters here are lamda_l1, lamda_l2, min_data_in_leaf, num_leaves and reg_alpha. We received the best parameters as 0, 1, 400, 200 and 0 for lamda_l1, lamda_l2, min_data_in_leaf, num_leaves and reg_alpha respectively.
- Now, we train the model by fitting the training data and find the important features. Above are important features.
- Then, we finally use the predict function and predict the target variable. Above is the confusion matrix of this prediction with 2675 false negatives.
- Gradient boost performed even better than light gradient boost.

4.0.9 XGBoost Implementation

- Grid search was used to find the best parameters and hyper tune the parameters. We are using cross validation technique and here we are using 5-fold cross validation.

```
vehicleModel_OKAI Kickscooter    0.238890
duration_new                     0.196603
zone_start_Zaventem              0.104946
vehicleModel_CORSA - Brussels    0.092223
vehicleModel_CITIGO - Antwerp    0.066534
vehicleModel_SCOOTER             0.053066
vehicleModel_IBIZA - All         0.050434
vehicleModel_MII - Antwerp       0.049568
zone_start_Charleroi             0.032488
zone_start_Brussels              0.018366
gender_MALE                      0.015154
age                              0.014077
zone_start_Antwerp               0.013767
gender_FEMALE                    0.013740
is working_False                 0.010502
is working_True                  0.010086
vehicleStatus_available          0.009787
vehicleStatus_maintenance        0.009770
```

Figure 16: XGBoost - Important Features

```
Confusion matrix:
 [[23823  2656]
 [ 2681 14708]]
TP:  14708
TN:  23823
FP:  2656
FN:  2681
```

Figure 17: XGBoost - Confusion Matrix

- Here, the hypertuning parameters are learning_rate, max_depth, min_child_weight, subsample, colsample_bytree, n_estimators and objective, for which we have received the best parameters as 0.1, 10, 5, 0.5, 0.7, 100 and binary: logistic, respectively.
- Now, we train the model by fitting the training data and find the important features. Above are important features.
- Then, we finally use the predict function and predict the target variable. Above is the confusion matrix of this prediction with 2681 false negatives.

4.0.10 Logistic Regression Implementation

- Grid search was used to find the best parameters and hyper tune the parameters. We are using cross validation technique and here we are using 5-fold cross validation.
- Here, the hypertuning parameters are penalty and C, for which we have received the best parameters as l2 and 100, respectively.
- Then, we finally use the predict function and predict the target variable. Below is the confusion matrix of this prediction with 2353 false negatives. This performed better than the boosting and bagging algorithms.

4.0.11 Keras Neural Network Implementation

- We will be using Keras Classifier wrapper to utilize the Keras models. We fit the model within the classifier, where we try different epoch and batch size values and it provides the neural network

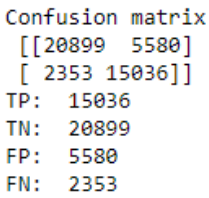


Figure 18: Logistic Regression - Confusion Matrix

model as the output.

- Initially, we create a model to start off with the layers. We decide a sample number of neurons which will be later hyper tuned.
- Input_dim is the total input variables. The initialization is performed using Rectifier activation. As our problem is binary classification, sigmoid is used to generate values between 0 and 1 and one neuron is used in the output layer.
- Finally, to compile the model, we use the binary_crossentropy as the logarithmic loss function as our problem is binary. Also, we initially use the optimizer as Adam for gradient descent and metrics.
- Now, we fit the model using the epoch and batch size as 300 and 40 respectively. This will be hyper tuned.
- We are using GridSearchCV to hypertune the parameters. We set the cross validation to 3. Initially, we hyper tune the batch size and number of epochs. Batch size shows the total number of patterns to be shown in the read. It also defines how many patterns are to be read and kept in memory. Number of Epochs shows that the total number of times total dataset must be shown network. This happens during training of the dataset. After hyper tuning, we find the best parameters of epoch and batch size as 150 and 10, respectively.
- Now, we tune the optimizer. We try the optimizers like SGD, Adam, Adamax, Nadam, Adagrad, Adadelta and RMSprop. We find the best optimizer as Adam.
- We then tune the kernel_initializers in the layers.

  We use different kernel_initializers in the GridSearchCV as uniform, normal, zero, he_normal, he_uniform. We find the best parameter as uniform.
- As our problem is binary, we, use linear in the second layer and sigmoid in the final output layer. Finally, we fit the model using KerasClassifier model.
- After prediction, we print the confusion matrix. Please find the below confusion matrix:
- After hyper tuning, we have False negative as 2342. This is the best model among all the previously used algorithms.

```
Confusion matrix:
 [[20986  5493]
 [ 2342 15047]]
TP:  15047
TN:  20986
FP:  5493
FN:  2342
```

Figure 19: Keras Classifier - Confusion Matrix

4.1 Results

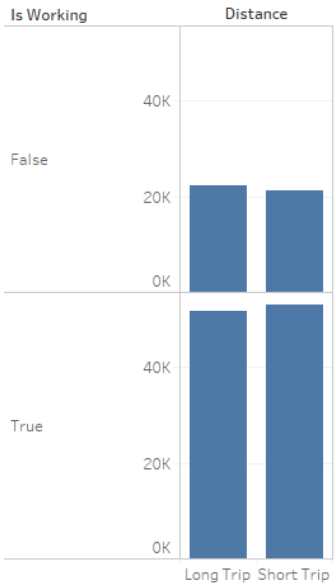
The seven models were compared based on False negative generated using the confusion matrix. The best random forest model was the least performing among all the models, as it had the highest number of false negatives. Boosting techniques performed better than the bagging random forest. Among the boosting techniques, gradient boost was the best performing model and more accurate. Keras neural network was the best performing model among all the models. The tradition logistic regression was in par with the Keras model as it is known to perform well with binary classification.

Model	**False Negative**
Random Forest	3132
AdaBoost	2739
Gradient Boost	2636
Light Gradient Boost	2681
XGBoost	2675
Logistic Regression	2353
Keras Neural Network	2342

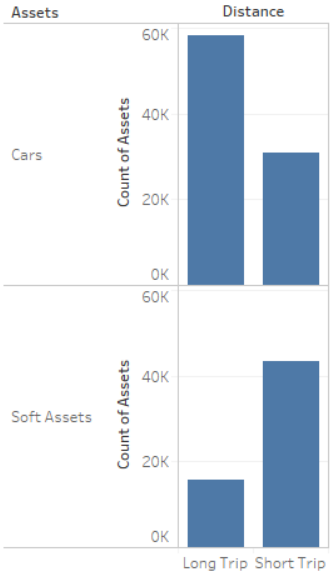
Table 1: Results

4.2 Insights

Looking at the important features generated by the models, we can find the important features impacting the usage. We found that duration of the ride is highly impacting the trip's usage. Other factors found to impact the trip are age and gender. Holidays were also found to be a key factor. Combustion cars are found to be more impacting than the Electric cars. As electric cars are more expensive than the combustion cars, people prefer combustion cars. Among the start zones of the vehicles, Antwerp and Brussels are more important than the rest. Based on the above factors we have visualized the data which will help Poppy to focus their marketing strategies.



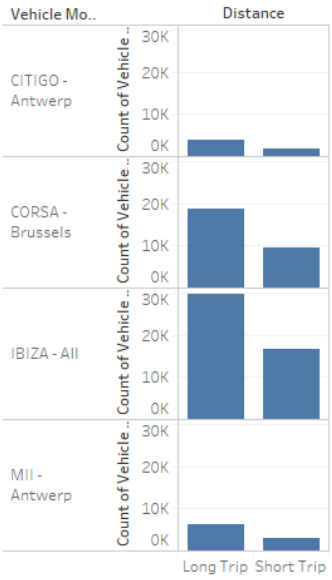
The above graph represents distance against holiday. It is found that on a holiday, long trips are more as compared to short trips. Whereas, on a working day, short trips are more compared to long trips. This may be because, on a holiday or weekends, customers may tend to go a vacation or long trips and, on a weekday, they tend to go on short trips which may be work related travels.



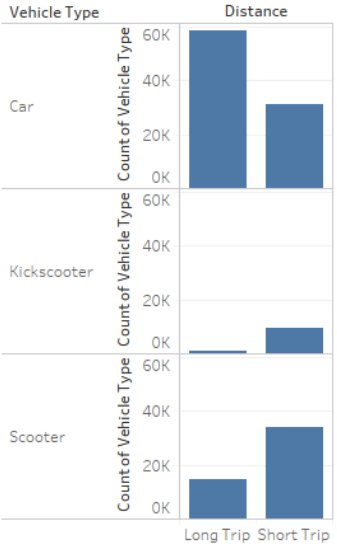
The above graph represents distance against assets. The soft assets are kick scooters and scooters. It is found that short trips are completed more using soft assets, as compared to long trips. Cars are more used for long trips.

The below graph represents distance against car models. We can see that combustion cars are used more than the electric cars. Corsa and Ibiza are the combustion cars used more than the Electric cars. This is because the electric cars are more expensive than the combustion cars, though it is more environment friendly. Small population do prefer electric cars, as it is environment friendly. But, in general customers

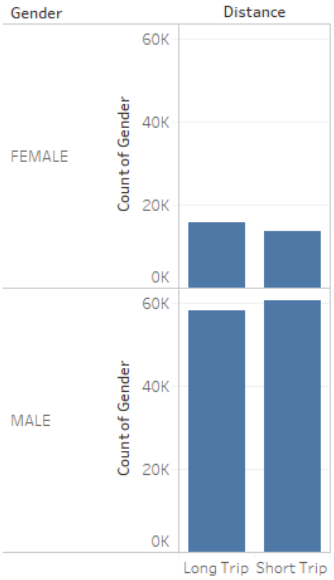
prefer to save money rather than the environment.



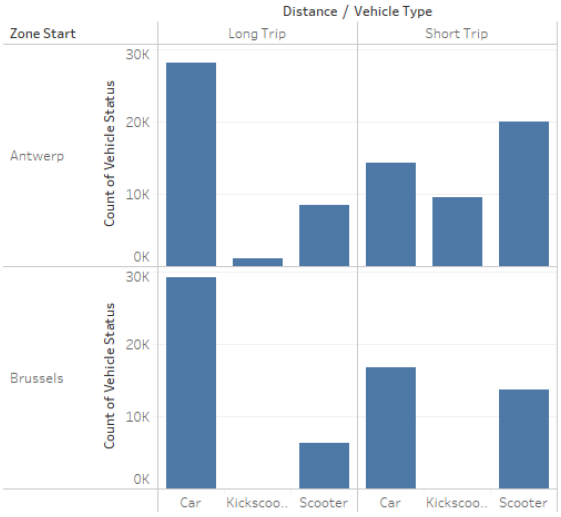
The below graph represents the distance against vehicle type. This is same as the asset graph we saw above. Here, we can see the breakdown in the soft assets that is the kick scooters and scooters. We already know that cars are used more for long trips and soft assets are used more for short trips. Even further, Scooters are used more than the kick scooters for short trips.



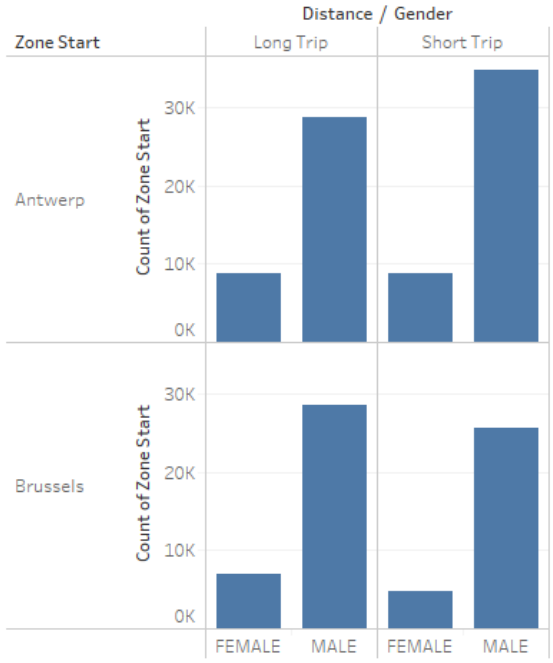
The below graph represents distance against gender. With the graph it's clear that male users are more than the female users. Among the female users, the users that book the ride for long trip is slightly more than the short trips. Whereas, among the male users.



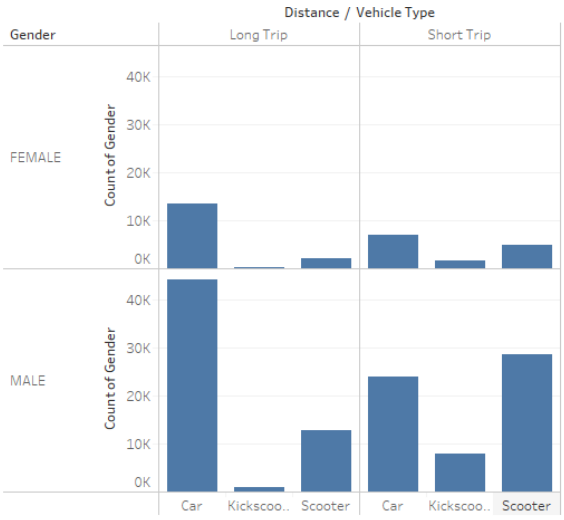
Start zone was also found as an important factor impacting the usage. In prior graph, we had found that for long trips cars are used more than short trips. But in Brussels, Cars are used more than the scooters for short trips whereas, in Antwerp, scooters are used more than the cars for short trips. Brussels have more users that book cars for long trips as compared to Antwerp. Antwerp have more users that book scooters and kick scooters for short trips as compared to Brussels. Antwerp has more users that kick scooters for short trip as compared to long trips.



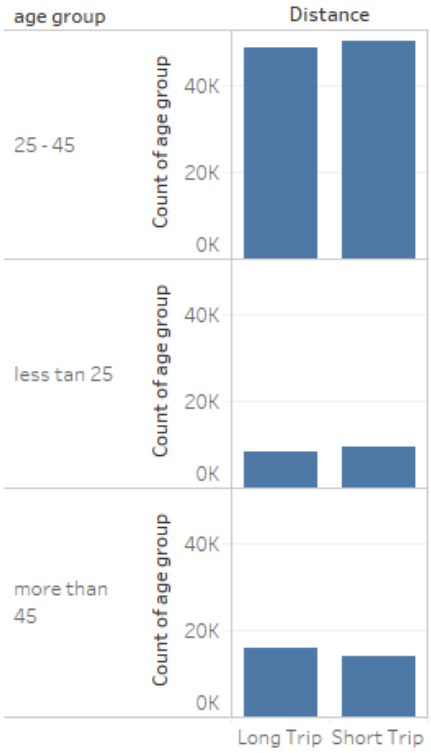
The below graph shows in depth gender breakdown of users as per zone. We already know there are more male users than female users. In Antwerp, there are more male users who book ride for short trips as compared to ride booking for long trips. Whereas, in Brussels, it's the opposite. The male users who book the ride for long trip is more than the male users who book the ride for short tips. In Brussels, the female users who book the ride for long trips are more as compared to ride bookings for short trips.



The below graph represents the distance and vehicle trip against gender. We can see a new pattern in short trips. The male use users prefer scooters over cars for short trips. Whereas, female riders prefer car over scooters for short trips.



Below graph represents distance against age group of riders. It is found that most riders are between the age group of 25 to 45. The age group of users more than 45 prefer long trip rather than short trips. The working class of age group between 25 to 45 have more short trips compared to long trips. This may me work related travels.


age group
Distance
25 - 45
less tan 25
more than 45
Count of age group
40K
20K
0K
Long Trip
Short Trip

5 Conclusion

5.1 Strategic Decisions

Based on the above factors impacting, we can derive certain strategies for Poppy Mobility. They can release below strategic promotional offers to increase the customers, based on our analysis.

- Release offers on long trips on a holiday and for short trips on a working day.
- Offers for using soft assets on a working day and cars on a holiday as soft assets are used more for short trip compared to long trips
- As people choose combustion cars over electric cars, marketing strategies should be used to educate the customers the benefits for electric cars so that more customers will shift towards electric cars as well.
- As scooters are used more than kick scooters for short trips, provide offers on kick scooters to increase the customers use of kick scooters as well.
- Provide promotional offers for short trips to female users, so that they will be encouraged to go for short trips as well.
- Release region specific offers. Offers for customers who use scooters and kick scooters for long routes in Brussels and offers for customers who use cars for short routes in Antwerp.

Above decisions will help Poppy to gain customers in the aspects where there as comparatively less customers.

5.2 Conclusion

The main objective of the project was to find the factors impacting the trip at the same time also predict the usage of Poppy Mobility. We considered distance as usage and were successfully able to predict the trips as short trips and long trips. Shorts trips where given more priority as short trip booking occurs in more number. Shorts trips were predicted based on other factors and this will reduce the average wait time. This will help Poppy to reach to a decision as to when the vehicles are to be deployed on the roads and when they can be put in maintenance based on usage. The research also aimed at comparing the models. Where in, boosting techniques outperformed the bagging random forest. But, among all the models, Keras neural network with binary cross entropy as logarithmic loss proved to be best among all the models. Also, tradition logistic regression's performance was good on this data. Thus, this research highly recommends Keras neural network and logistic regression for the binary classification of the Poppy mobility's usage data.

5.3 Future Work

The focus of this research was to find the factors impacting the usage and predict the usage of Poppy Mobility. For prediction we used Neural network's Keras, Logistic regression and, other boosting and bagging techniques. As Neural network provided the best result, we must use other neural network techniques to get the best results. Based on the impacting factors and graphs provided in the findings section, we can generate this graph on the live data in the form of dashboard visualizations. This will help to make

decision on the real time data. Also, Analytics on the real time data to predict the usage, this will help Poppy Mobility to determine number of vehicles to be on live in order to reduce the waiting time and gain more customers.

References

[1] (2019), https://poppy.be/

[2] Bhadane, C., Shah, K.: Clustering algorithms for spatial data mining. In: Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis. pp. 5–9 (2020)

[3] Borgli, R.J., Halvorsen, P., Riegler, M., Stensland, H.K.: Automatic hyperparameter optimization in keras for the mediaeval 2018 medico multimedia task. In: MediaEval (2018)

[4] Brownlee, J.: Logistic regression for machine learning. Machine Learning Mastery. https://machinelearningmastery. com/logistic-regression-for-machine-learning. Accessed **1** (2016)

[5] Curry, M., Dickerson, J.P., Sankararaman, K.A., Srinivasan, A., Wan, Y., Xu, P.: Mix and match: Markov chains and mixing times for matching in rideshare. In: International Conference on Web and Internet Economics. pp. 129–141. Springer (2019)

[6] Gama, J., Brazdil, P.: Characterization of classification algorithms. In: Portuguese Conference on Artificial Intelligence. pp. 189–200. Springer (1995)

[7] Hillsgrove, T., Steele, R.: Machine learning-based wait-time prediction for autonomous mobility-on-demand systems. In: 2019 SoutheastCon. pp. 1–7. IEEE (2019)

[8] Howard, A., Lee, T., Mahar, S., Intrevado, P., Woodbridge, D.: Distributed data analytics framework for smart transportation. In: 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). pp. 1374–1380. IEEE (2018)

[9] Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: Comparing boosting and bagging techniques with noisy and imbalanced data. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **41**(3), 552–568 (2010)

[10] Koul, S., Datta, C.V., Verma, R.: Car rentals' knowledge and customer choice. In: 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). pp. 1–5. IEEE (2020)

[11] Kukreja, H., Bharath, N., Siddesh, C., Kuldeep, S.: An introduction to artificial neural network. Int J Adv Res Innov Ideas Educ **1**, 27–30 (2016)

[12] Kumari, R., Srivastava, S.K.: Machine learning: A review on binary classification. International Journal of Computer Applications **160**(7) (2017)

[13] Lateef, Z.: A beginners guide to boosting machine learning algorithms — edureka (Jun 2019), https://www.edureka.co/blog/boosting-machine-learning/

[14] Morde, V.: Xgboost algorithm: Long may she reign! - towards data science (Apr 2019), https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99b

[15] More, S.V., Rakshit, P.: A review of literature to understand car rental service market. Advance and Innovative Research p. 281 (2020)

[16] Narsaria, I., Verma, M., Verma, A.: Measuring satisfaction of rental car services in india for policy lessons. Case Studies on Transport Policy (2020)

[17] Natekin, A., Knoll, A.: Gradient boosting machines, a tutorial. Frontiers in neurorobotics **7**, 21 (2013)

[18] Navlani, A.: Adaboost classifier in python (2018), https://www.datacamp.com/community/tutorials/breakcites adaboost-classifier-python

[19] Pamuluri, H.R.: Predicting user mobility using deep learning methods (2020)

[20] Panov, P., Džeroski, S.: Combining bagging and random subspaces to create better ensembles. In: International Symposium on Intelligent Data Analysis. pp. 118–129. Springer (2007)

[21] Prettenhofer, P., Louppe, G.: Gradient boosted regression trees in scikit-learn (2014)

[22] Report, M.S.: Mobility on demand - market research — recent trends and growth forecast 2025. cuereport.com pp. 22–35 (2020)

[23] Sathishkumar, V., Park, J., Cho, Y.: Using data mining techniques for bike sharing demand prediction in metropolitan city. Computer Communications **153**, 353–366 (2020)

[24] Shamshiripour, A., Rahimi, E., Shabanpour, R., Mohammadian, A.K.: Dynamics of travelers' modality style in the presence of mobility-on-demand services. Transportation Research Part C: Emerging Technologies **117**, 102668 (2020)

[25] Sharma, A.: What makes lightgbm lightning fast? - towards data science (Oct 2018), https://towardsdatascience.com/what-makes-lightgbm-lightning-fast-a27cf0d9785e

[26] Singhal, Y., Jain, A., Batra, S., Varshney, Y., Rathi, M.: Review of bagging and boosting classification performance on unbalanced binary classification. In: 2018 IEEE 8th International Advance Computing Conference (IACC). pp. 338–343. IEEE (2018)

[27] Sun, D., Leurent, F., Xie, X.: Floating car data mining: Identifying vehicle types on the basis of daily usage patterns. Transportation Research Procedia **47**, 147–154 (2020)

[28] Upasana: Classification algorithms — types of classification algorithms — edureka (Jan 2019), https://www.edureka.co/blog/classification-algorithms/

[29] Victoriano, R., Paez, A., Carrasco, J.A.: Time, space, money, and social interaction: Using machine learning to classify people's mobility strategies through four key dimensions. Travel Behaviour and Society **20**, 1–11 (2020)

[30] Vidhya, A.: Boosting algorithm — boosting algorithms in machine learning (Nov 2015), https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-breakcites machine-learning/

[31] Webb, G.I.: Multiboosting: A technique for combining boosting and wagging. Machine learning **40**(2), 159–196 (2000)

[32] Wirth, R., Hipp, J.: Crisp-dm: Towards a standard process model for data mining. In: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining. pp. 29–39. Springer-Verlag London, UK (2000)

[33] Yan, X., Liu, X., Zhao, X.: Using machine learning for direct demand modeling of ridesourcing services in chicago. Journal of Transport Geography **83**, 102661 (2020)

[34] Zhao, X., Yan, X., Yu, A., Van Hentenryck, P.: Prediction and behavioral analysis of travel mode choice: A comparison of machine learning and logit models. Travel behaviour and society **20**, 22–35 (2020)