Demand Prediction for Public Transport

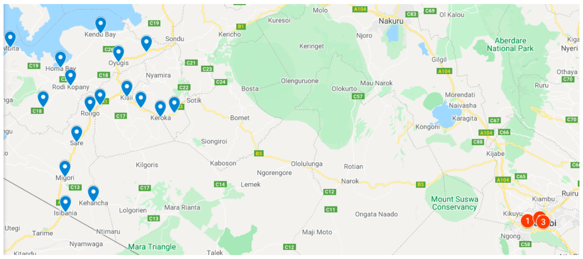
Aniket Raj, Kushal Kishor, A Balaji,
Akhilesh Bhardwaj, Aswathaman,
Data science trainees,
AlmaBetter, Bangalore

Overview:

Mobiticket is a travel technology company that enables travelers with a built-in transport ticket reservation and payment system via mobile.

Business Problem:

We build a model that predicts the number of seats that Mobiticket can expect to sell for each ride, i.e., for a specific route on a specific date and time. There are 14 routes in this dataset. All of the routes end in Nairobi and originate in towns to the North-West of Nairobi towards Lake Victoria.



The routes from these 14 origins to the first stop in the outskirts of Nairobi takes approximately 8 to 9 hours from time of departure. From the first stop in the outskirts of Nairobi into the main bus terminal, where most passengers get off, in Central Business District, takes another 2 to 3 hours depending on traffic.

The three stops that all these routes make in Nairobi (in order) are:

1. Kawangware: the first stop in the outskirts of Nairobi
2. Westlands
3. Afya Centre: the main bus terminal where most passengers disembark

Passengers of these bus (or shuttle) rides are affected by Nairobi traffic not only during their ride into the city, but from there they must continue their journey to their final destination in Nairobi wherever that may be. Traffic can act as a deterrent for those who have the option to avoid buses that arrive in Nairobi during peak traffic hours. On the other hand, traffic may be an indication for people’s movement patterns, reflecting business hours, cultural events, political events, and holidays.

Data Description:

Nairobi Transport Data.csv (zipped) is the dataset of tickets purchased from Mobiticket for the 14 routes from “up country” into Nairobi between 17 October 2017 and 20 April 2018. This dataset includes the variables: ride_id, seat_number, payment_method, payment_receipt, travel_date, travel_time, travel_from, travel_to, car_type, max_capacity.

Uber Movement traffic data can be accessed here. Data is available for Nairobi through June 2018. Uber Movement provided historic hourly travel time between any two points in Nairobi. Any tables that are extracted from the Uber Movement platform can be used in your model.

This dataset has around 51,645 observations in it with 10 columns and it is a mix between categorical and numeric values.

Variables description:

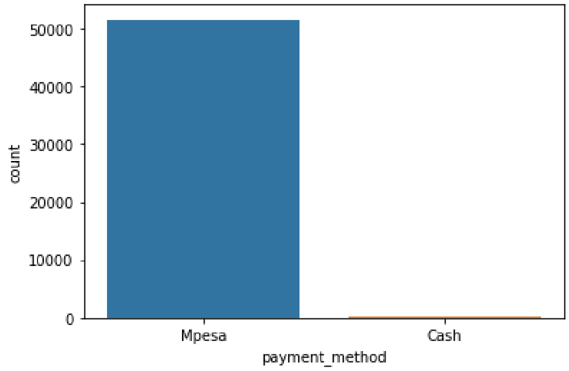
- **ride_id:** unique ID of a vehicle on a specific route on a specific day and time.
- **seat_number:** seat assigned to ticket.
- **payment_method:** method used by customer to purchase ticket from Mobiticket. (cash or Mpesa)
- **payment_receipt:** unique id number for ticket purchased from Mobiticket.
- **travel_date:** date of ride departure. (MM/DD/YYYY)
- **travel_time:** scheduled departure time of ride. Rides generally depart on time. (hh:mm)
- **travel_from:** town from which ride originated.
- **travel_to:** destination of ride. All rides are to Nairobi.
- **car_type:** vehicle type (shuttle or bus)
- **max_capacity:** number of seats on the vehicle.

Approach:

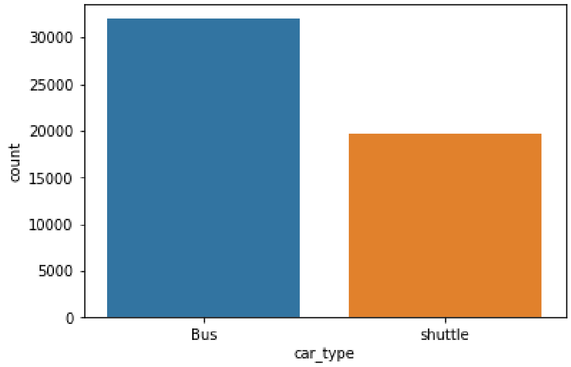
Our business wants to predicts the number of seats that Mobiticket can expect to sell for each ride, i.e., for a specific route on a specific date and time. Problem is a regression one and we can follow the steps as given below to develop a model —

- **Perform exploratory data analysis** — Observe various features which are impacting the number of ticket (number of seats).
- **Prepare Data** — Clean data — missing values, unknown values, encoding to ensure that data is ready for algorithm to consume.
- **Split data** — Split our data into training and test data. I went for 67–33 split.
- **Choose an algorithm** — Identifying a right algorithm for the problem is a major task and mostly it just doesn't happen in one go. I went for Regression algorithm as all features.
- **Predict and evaluate model** by using various metrics used for Regression, i.e., RMSE, MSE, MAE, RMSPE, MAPE.

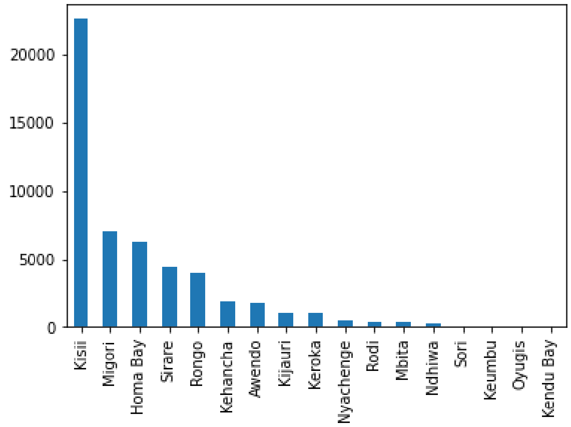
Results of Exploratory Data Analysis:



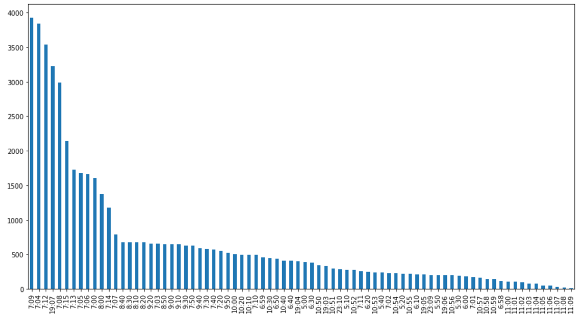
There are two type of payment methods people have used to buy the tickets.



There are two type of cars Bus and shuttle.



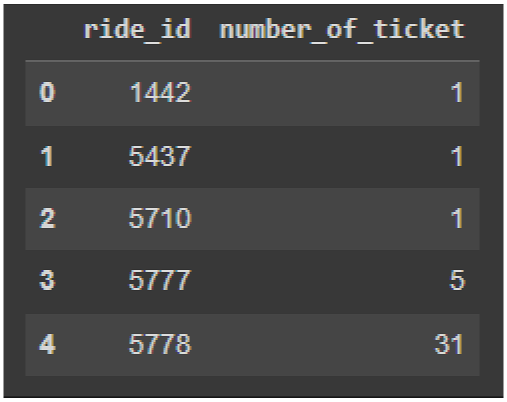
From the above bar graph, we can say most of the travelers traveling from Kisii to Nairobi.



From the above bar graph, we can say most of the travelers travel at 7:09 from the origin.

Target Variable:

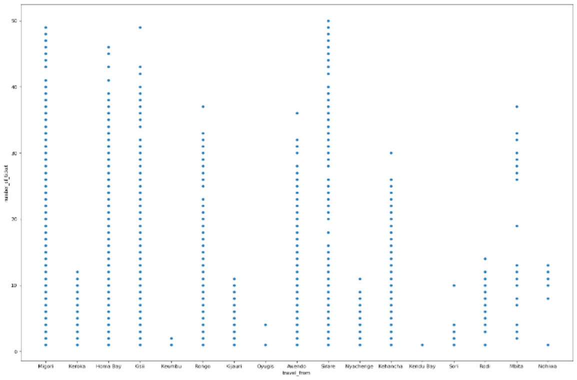
I will find the count of each ride_id and that will be the number_of_ticket as our target variable.

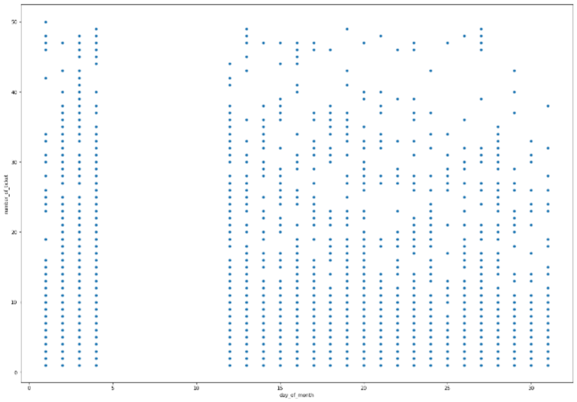


	ride_id	number_of_ticket
0	1442	1
1	5437	1
2	5710	1
3	5777	5
4	5778	31

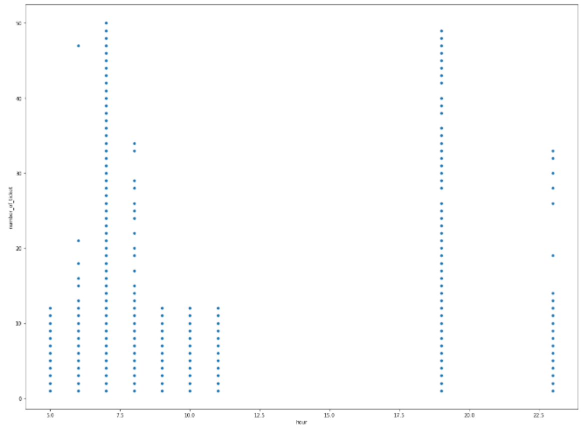
Preparing Data for training:

1.Separate dependent and independent variables (features and labels)





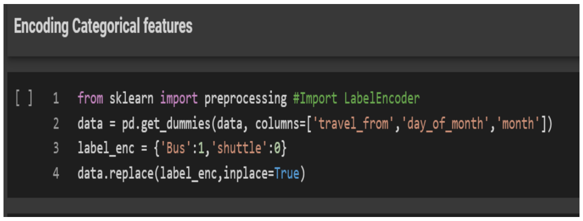
We can see that there is the gap between 5 to 11 in the day of the month. We can assume that there is official holiday of public transport between these days. we can also say that the number of tickets in all the days of month are same.



Summary

- We can see that most of the ticktes were sold at 7 AM and 8 PM. And that seems true because in the morning most of the people go to the work and office.
- From the above we can say that there is not ride between 12pm to 5.30Pm.

2.Encoding categorical features



Encoding Categorical features

```
from sklearn import preprocessing #Import LabelEncoder
data = pd.get_dummies(data, columns=['travel_from','day_of_month','month'])
label_enc = {'Bus':1,'shuttle':0}
data.replace(label_enc,inplace=True)
```

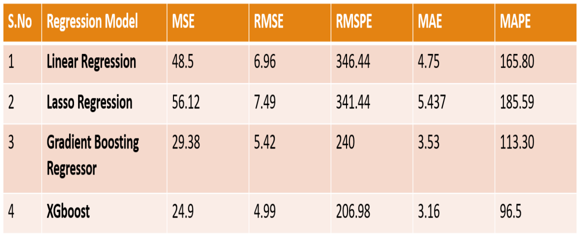
Encoding for categorical features are done by using get_dummies method.

The shape of the final data set is (6042,52)

3.Perform train, test split

4.Create Model object and train model

5.Perform Predictions

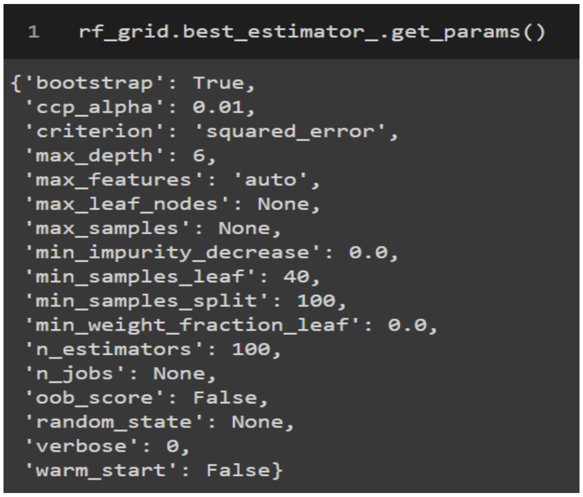


S.No	Regression Model	MSE	RMSE	RMSPE	MAE	MAPE
1	Linear Regression	48.5	6.96	346.44	4.75	165.80
2	Lasso Regression	56.12	7.49	341.44	5.437	185.59
3	Gradient Boosting Regressor	29.38	5.42	240	3.53	113.30
4	XGboost	24.9	4.99	206.98	3.16	96.5

Hyperparameter Tuning:

1.Grid Search CV on Random Forest:

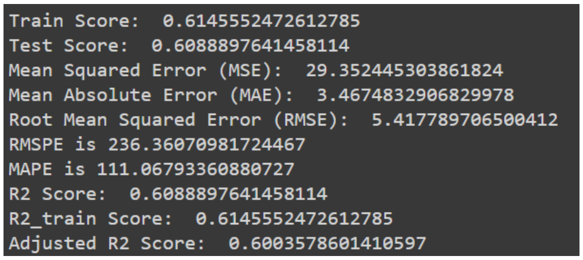
The best parameters for random forest are—



```
rf_grid.best_estimator_.get_params()
```

```
{'bootstrap': True,
 'ccp_alpha': 0.01,
 'criterion': 'squared_error',
 'max_depth': 6,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 40,
 'min_samples_split': 100,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': None,
 'verbose': 0,
 'warm_start': False}
```

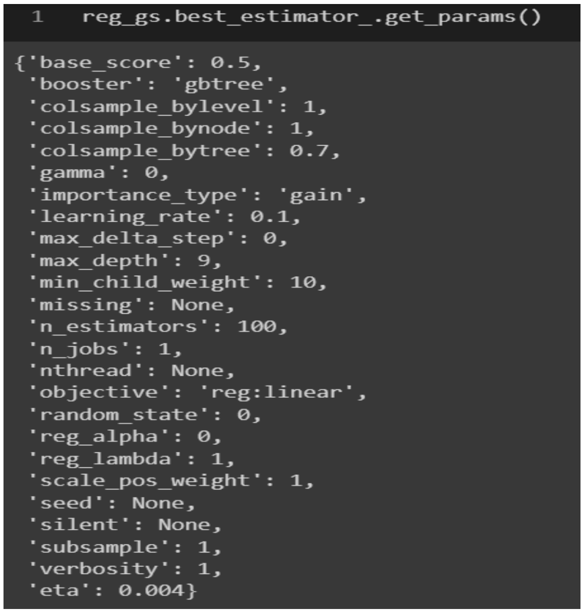
The Evaluation metrics for this model are—



```
Train Score:  0.6145552472612785
Test Score:  0.6088897641458114
Mean Squared Error (MSE):  29.352445303861824
Mean Absolute Error (MAE):  3.4674832906829978
Root Mean Squared Error (RMSE):  5.417789706500412
RMSPE is 236.36070981724467
MAPE is 111.06793360880727
R2 Score:  0.6088897641458114
R2_train Score:  0.6145552472612785
Adjusted R2 Score:  0.6003578601410597
```

2.Grid search CV on Xgboost algorithm

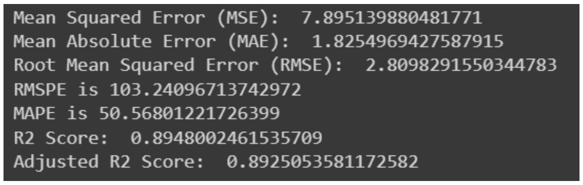
The best parameters for Xgboost algorithm are—



```
reg_gs.best_estimator_.get_params()
```

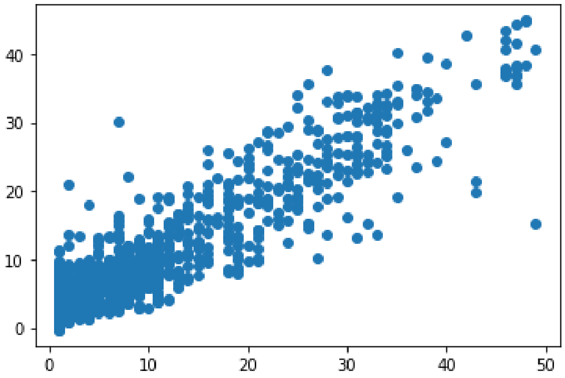
```
{'base_score': 0.5,
 'booster': 'gbtree',
 'colsample_bylevel': 1,
 'colsample_bynode': 1,
 'colsample_bytree': 0.7,
 'gamma': 0,
 'importance_type': 'gain',
 'learning_rate': 0.1,
 'max_delta_step': 0,
 'max_depth': 9,
 'min_child_weight': 10,
 'missing': None,
 'n_estimators': 100,
 'n_jobs': 1,
 'nthread': None,
 'objective': 'reg:linear',
 'random_state': 0,
 'reg_alpha': 0,
 'reg_lambda': 1,
 'scale_pos_weight': 1,
 'seed': None,
 'silent': None,
 'subsample': 1,
 'verbosity': 1,
 'eta': 0.004}
```

The Evaluation metrics for this model are—

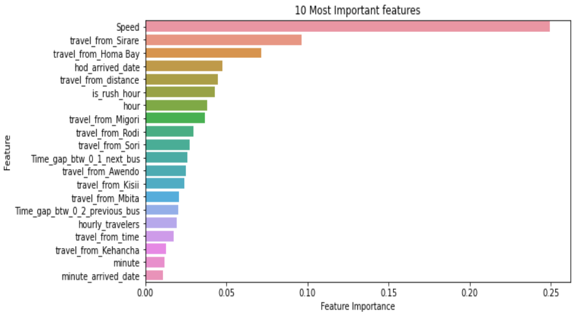


```
Mean Squared Error (MSE):  7.895139880481771
Mean Absolute Error (MAE):  1.8254969427587915
Root Mean Squared Error (RMSE):  2.8098291550344783
RMSPE is 103.24096713742972
MAPE is 50.56801221726399
R2 Score:  0.8948002461535709
Adjusted R2 Score:  0.8925053581172582
```

Residual Analysis:



Feature Importance:



Conclusion:

We used different type of regression algorithms to train our model like, Linear Regression, Regularized linear regression (Lasso), GBM, Random Forest Regressor, XGboost regresssor. and also, we tuned the parameters of Random Forest regressor and XGboost regressor and also found the important features for training the model. Out of them XGboost with tuned hyperparameters gave the best result.