Taxi Word Document Report

Dear Mr./Ms. President,

*Feature Extraction*

We extracted temporal information. We converted 'from\_date' with the Dtype of object to datetime. We also converted 'booking\_created' from object to datetime. Next, we created a new attribute ('time\_ahead') to represent the ahead time which is the time between creating the booking and the from date. Next, we created new attributes representing the day of the week and the hour of the day. Next, we extracted spatial information by creating new attributes representing the trip distance. Then, we created new attributes representing if the trip is in the same area. We removed the attributes that were processed. We also removed the city information and to\_date because they have most of the instances missing. We converted variables to binary indicators.

Then we calculated the percentage of missing values for each.

Column Missing Percentage

package\_id package\_id 82.48

to\_lat to\_lat 20.91

to\_long to\_long 20.91

distance distance 20.91

from\_lat from\_lat 0.15

from\_long from\_long 0.15

vehicle\_model\_id vehicle\_model\_id 0.00

travel\_type\_id travel\_type\_id 0.00

online\_booking online\_booking 0.00

mobile\_site\_booking mobile\_site\_booking 0.00

Car\_Cancellation Car\_Cancellation 0.00

time\_ahead time\_ahead 0.00

day\_of\_week day\_of\_week 0.00

hour\_of\_day hour\_of\_day 0.00

within\_area within\_area 0.00

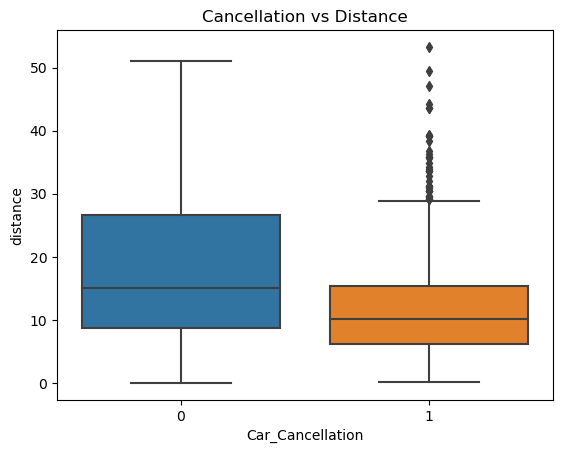
As we can see there is around 82% of the dataset without package\_id so it is safer to drop it.

Now the only attribute with missing values is the distance which is 21% which can be handled when building and running the machine building models.

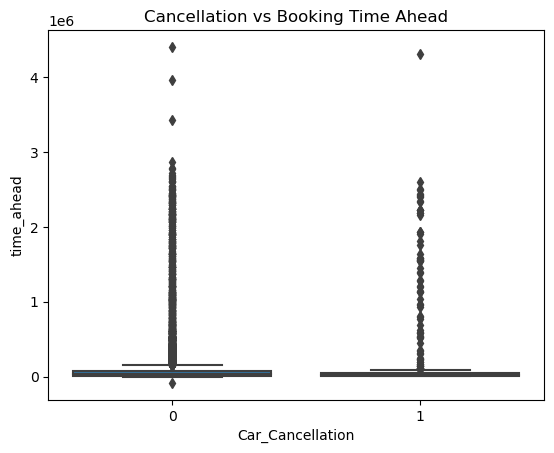
*Data Analysis and Visualization*

As we can see vehicle\_model\_id is dominated by a single value, hence we can drop it.

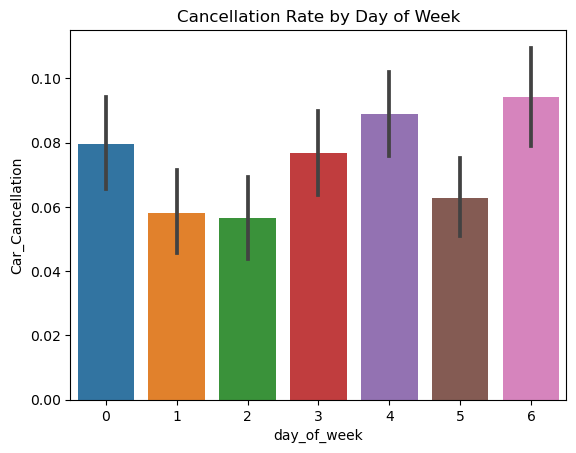
As we can see within area is always 0 so we can drop it.

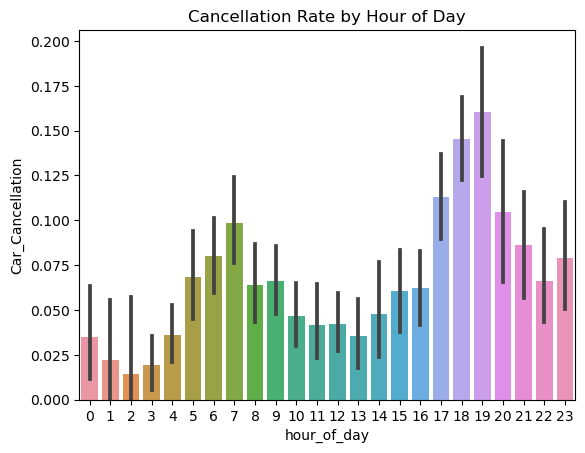


Cancelled bookings have less average and median distance than non-cancelled booking, however, they have a lot of outliers.



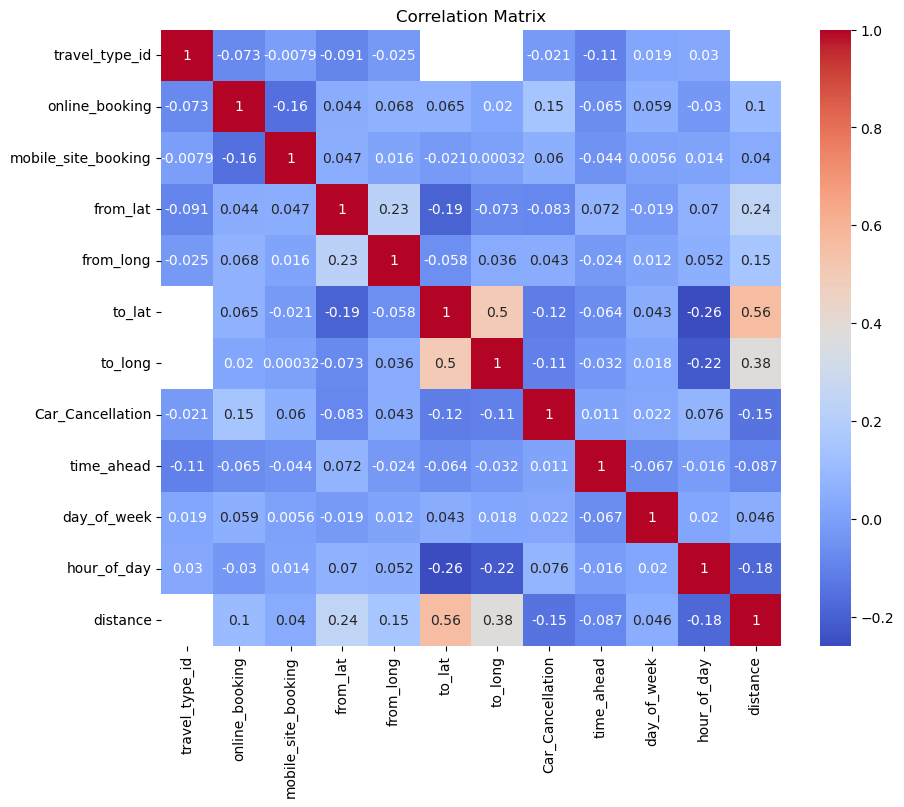
Most of the bookings whether cancelled or not happened directly near the actual trip time.





We can see here that the highest cancellation rates happen between hours 18-19.

There is also another peak in cancellation rate in the morning hours between 6-7.



As we can see from the correlation map there are very negligible to absent linear correlation between features and between each feature and car cancellations. We can deduce from here that linear classifiers such as logistic regression will not perform well.

*Data Set Balance*

0 9257

1 743

Name: Car\_Cancellation, dtype: int64

We have 743 instances out of 10,000 with 'Car\_Cancellation'=1, around 7.4% of the data is positive and the rest is negative.

Given that we are dealing with an imbalanced data set we have to maintain the imbalance ratio when splitting the data into training and validation.

*Machine Learning and Model Development and Predictions*

We split the data set. We performed an 80/20 split with stratify enabled to ensure that the target class ratio was maintained in both splits. We created pipelines for 5 different algorithms (random forest, logistic regression, naive bayes, neural network, and a decision classification tree). Each pipeline contained an imputer which filled the missing values using the mean value.

We also did scaling (transformation) of the data using the standard scaler approach for only the neural network classification model. The other classification methods did not require transformed (scaled) data and we used the original data for them.

*Training and Evaluating the Model*

Metrics for Random Forest:

Accuracy: 0.9215

Precision: 0.40476190476190477

Recall: 0.11409395973154363

F1-Score: 0.17801047120418848

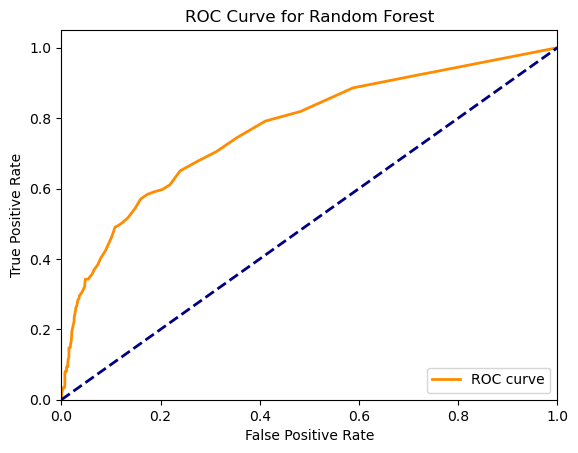
ROC AUC: 0.5502938734368145

Confusion Matrix:

[[1826 25]

[ 132 17]]

==============================



Metrics for Logistic Regression:

Accuracy: 0.9255

Precision: 0.0

Recall: 0.0

F1-Score: 0.0

ROC AUC: 0.5

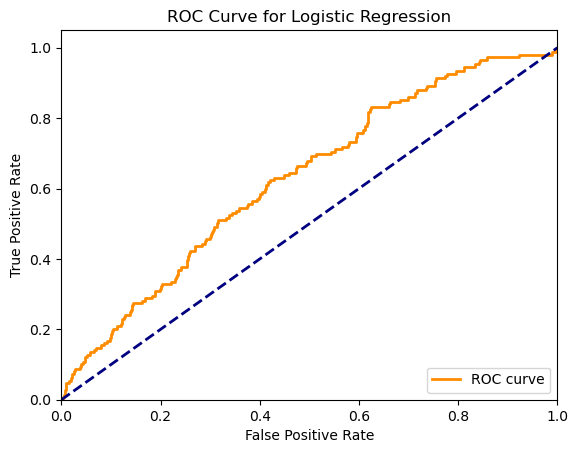
Confusion Matrix:

[[1851 0]

[ 149 0]]

==============================

C:\Users\nicol\anaconda3\lib\site-packages\sklearn\metrics\\_classification.py:1318: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 due to no predicted samples. Use `zero\_division` parameter to control this behavior.\_warn\_prf(average, modifier, msg\_start, len(result))



Metrics for Naive Bayes:

Accuracy: 0.909

Precision: 0.14893617021276595

Recall: 0.04697986577181208

F1-Score: 0.07142857142857144

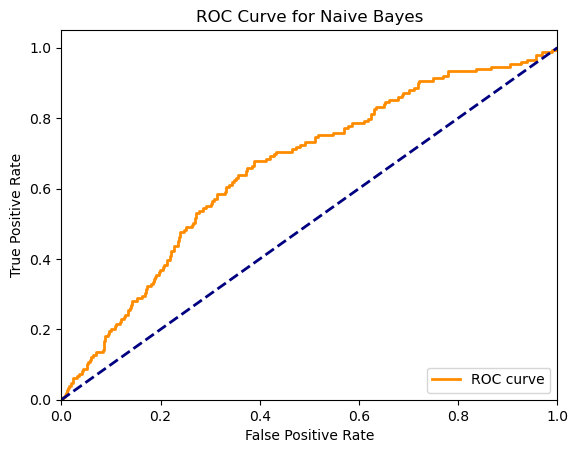
ROC AUC: 0.5126849625995744

Confusion Matrix:

[[1811 40]

[ 142 7]]

==============================



Metrics for Decision Tree:

Accuracy: 0.8785

Precision: 0.2235294117647059

Recall: 0.2550335570469799

F1-Score: 0.2382445141065831

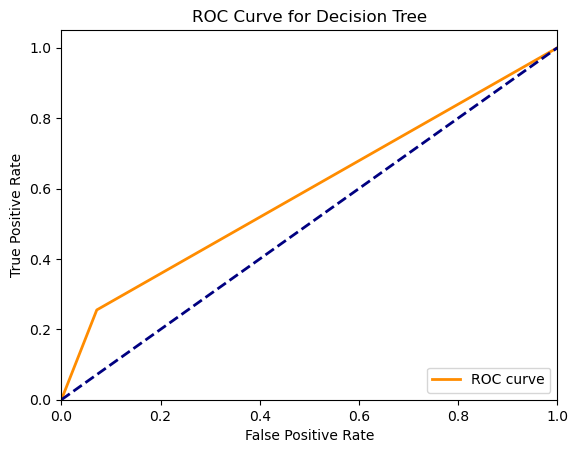
ROC AUC: 0.5918603765785954

Confusion Matrix:

[[1719 132]

[ 111 38]]

==============================



Metrics for Neural Network:

Accuracy: 0.925

Precision: 0.45454545454545453

Recall: 0.03355704697986577

F1-Score: 0.0625

ROC AUC: 0.5151577779469831

Confusion Matrix:

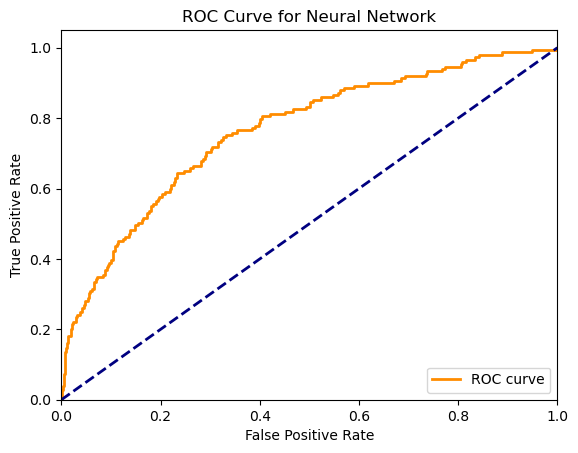
[[1845 6]

[ 144 5]]

==============================

C:\Users\nicol\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:692: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimization hasn't converged yet.

warnings.warn(



Overall, the decision tree achieved the best results relatively compared to all the other models. This is shown from evaluation metrics used as well as from the ROC curve.

Logistic regression acted as a majority classifier where it predicted all the instances but one as class 0.

Although random forest predicted a couple of instances as class 1, the recall was very low, which showed the model was very biased towards class 0.

Naive bayes achieved relatively the best overall performance when compared to logistic regression and random forest. However, it still had a low performance to detect positive classes.