Predicting cab booking

Demonstrate Predicting Taxi booking using Python for Github

The Problem Statement

- To improve the customer service for imaginary cab company called 'Lovelycabs'
- A certain percentage of booking gets canceled by the company due to the unavailability of a car.
- So, the challenge is to **build a predictive model**, which would classify the upcoming bookings as, if they would eventually get cancelled due to car unavailability, or not.
- So this is a **classification** problem.

The Dataset

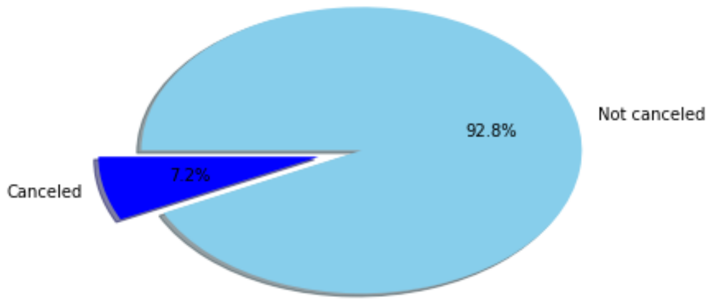
- Originally listed as a ‘Kaggle’ challenge.
- Downloaded as ‘csv file’ from the following : https://www.kaggle.com/c/predicting-cab-booking-cancellations2/data
- 43431 rows and 20 columns
- All the variables/data columns are categorical. The target variable/column is 'Car_cancellation', which takes the value "1", if the ride gets canceled, otherwise "0".

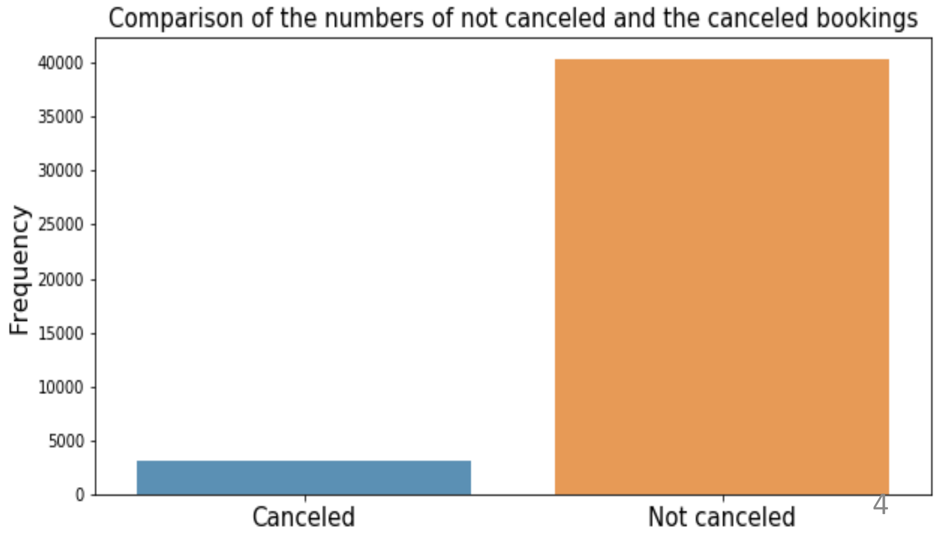
The first 5 lines of the data

	id	user_id	vehicle_model_id	package_id	travel_type_id	from_area_id	to_area_id	from_city_id	to_city_id	from_date
0	132512	22177	28	NaN	2	83.0	448.0	NaN	NaN	1/1/2013 2:00
1	132513	21413	12	NaN	2	1010.0	540.0	NaN	NaN	1/1/2013 9:00
2	132514	22178	12	NaN	2	1301.0	1034.0	NaN	NaN	1/1/2013 3:30
3	132515	13034	12	NaN	2	768.0	398.0	NaN	NaN	1/1/2013 5:45
4	132517	22180	12	NaN	2	1365.0	849.0	NaN	NaN	1/1/2013 9:00

Data Wrangling:

- Python packages used: **NumPy**, **Pandas**, **Scikit-learn**, **Matplotlib**, **Seaborn**
- **Data/column engineering: 'Booking_created'**: timestamp of the ride booking information . **'from_date'**: timestamp of the actual trip start information. We have split those 'DateTime' objects into separate day of the week, date, month and hour columns.
- **Class imbalance:** Only ~7% (only 3132, in total 43,431) of the total booking has been canceled.

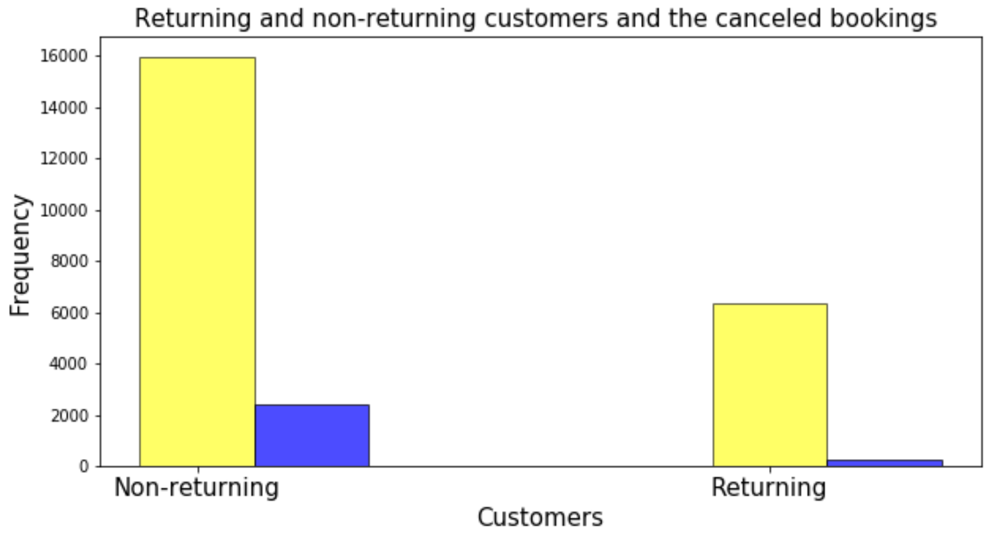


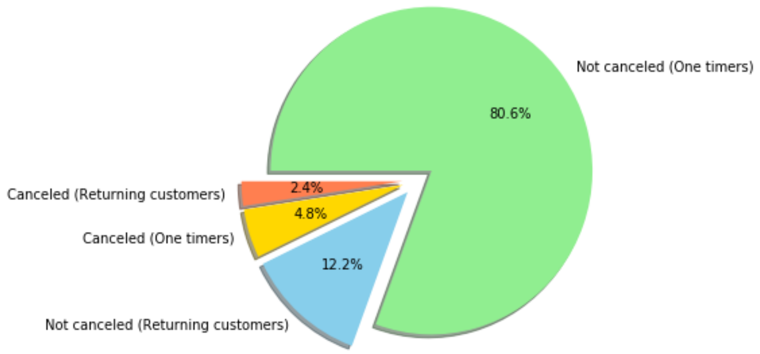


Exploratory Data Analysis (EDA)

User ID:

- Each user has been assigned a unique ‘User ID’
- Total 22267 user IDs.
- ‘user_id '29648' is the most frequent user (frequency 471).
- The no. of one-time users (non returning) are: 15935 and that of the returning customers are: 6332.
- ~16.6% of the total returning customers got their trips canceled.

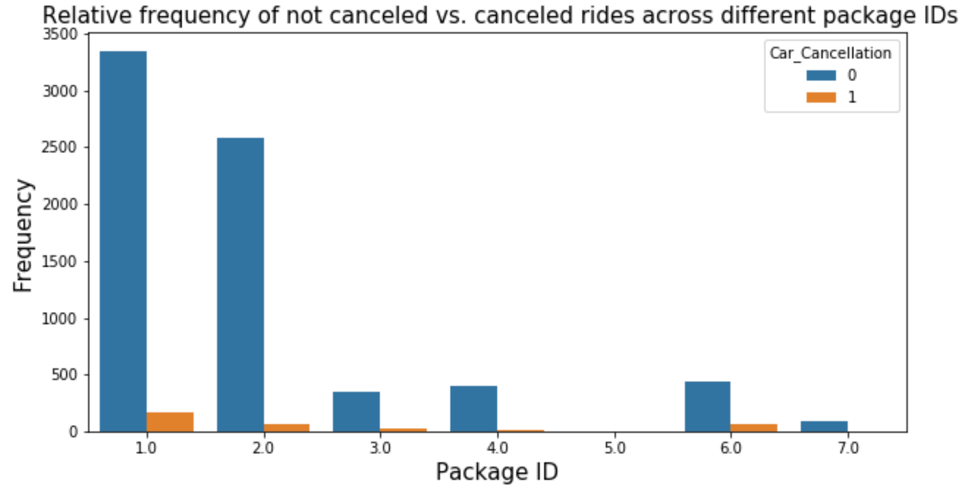




EDA – continued:

Package ID:

- Different package IDs are the various travel (booking) plans, from which customers can choose theirs.



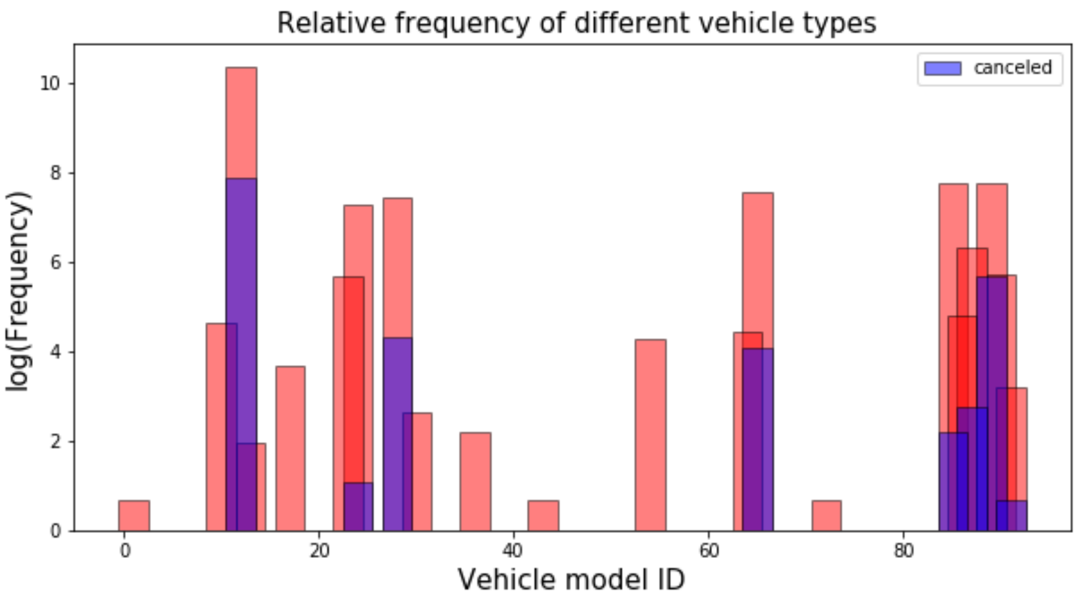
Travel type ID:

- Three different travel types are available to choose.
- Travel type '2' (i.e. for point to point travel) is the most popular.

EDA – continued:

Vehicle model ID:

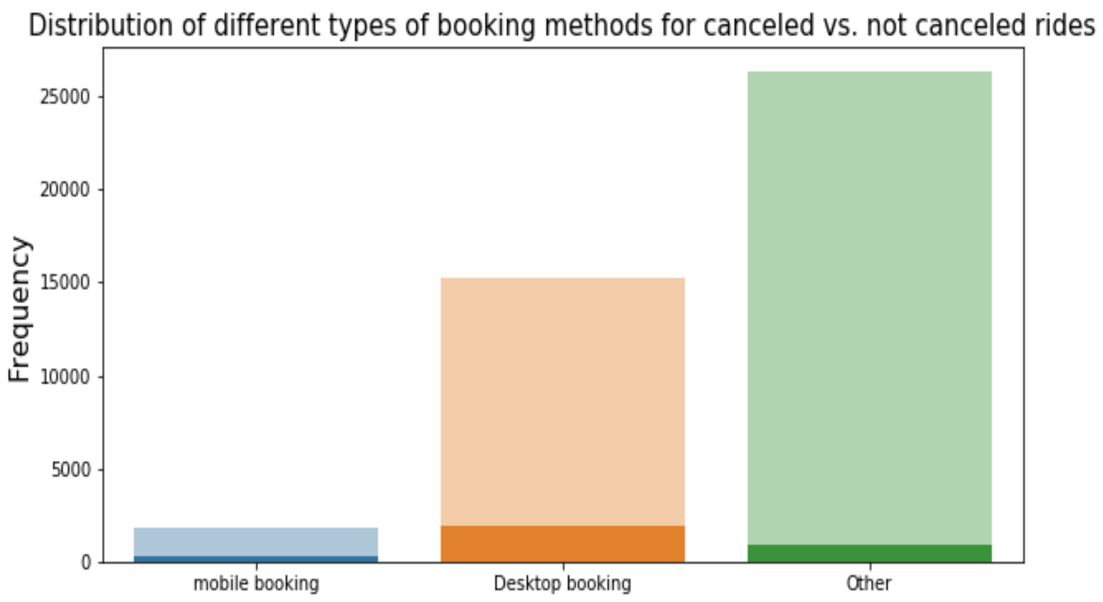
- 27 different types of vehicles have been listed.
- The most popular one is the vehicle with the vehicle ID no '12 (used 31859 times.)
- Got the maximum number of cancellations (2668 times) too.
- Y-axis has been resized by using logarithmic operation, to get a clear picture of the entire data.



EDA – continued:

Booking methods:

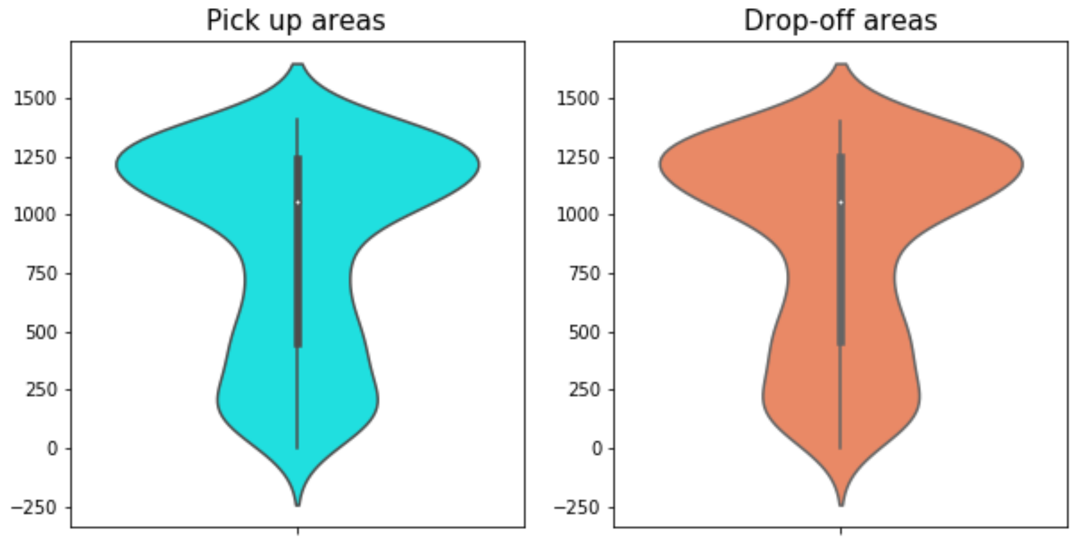
- Three different types of 'Booking methods.' : 'mobile booking' ,'desktop/website booking.' So, the remaining portion as 'other method' of booking.
- 1878 bookings have been made from mobile websites, 15270 bookings from desktop websites, so, 26283 bookings have been made differently! So, other methods of booking are mostly favored
- maximum cancellations correspond to the bookings made from the desktop websites.



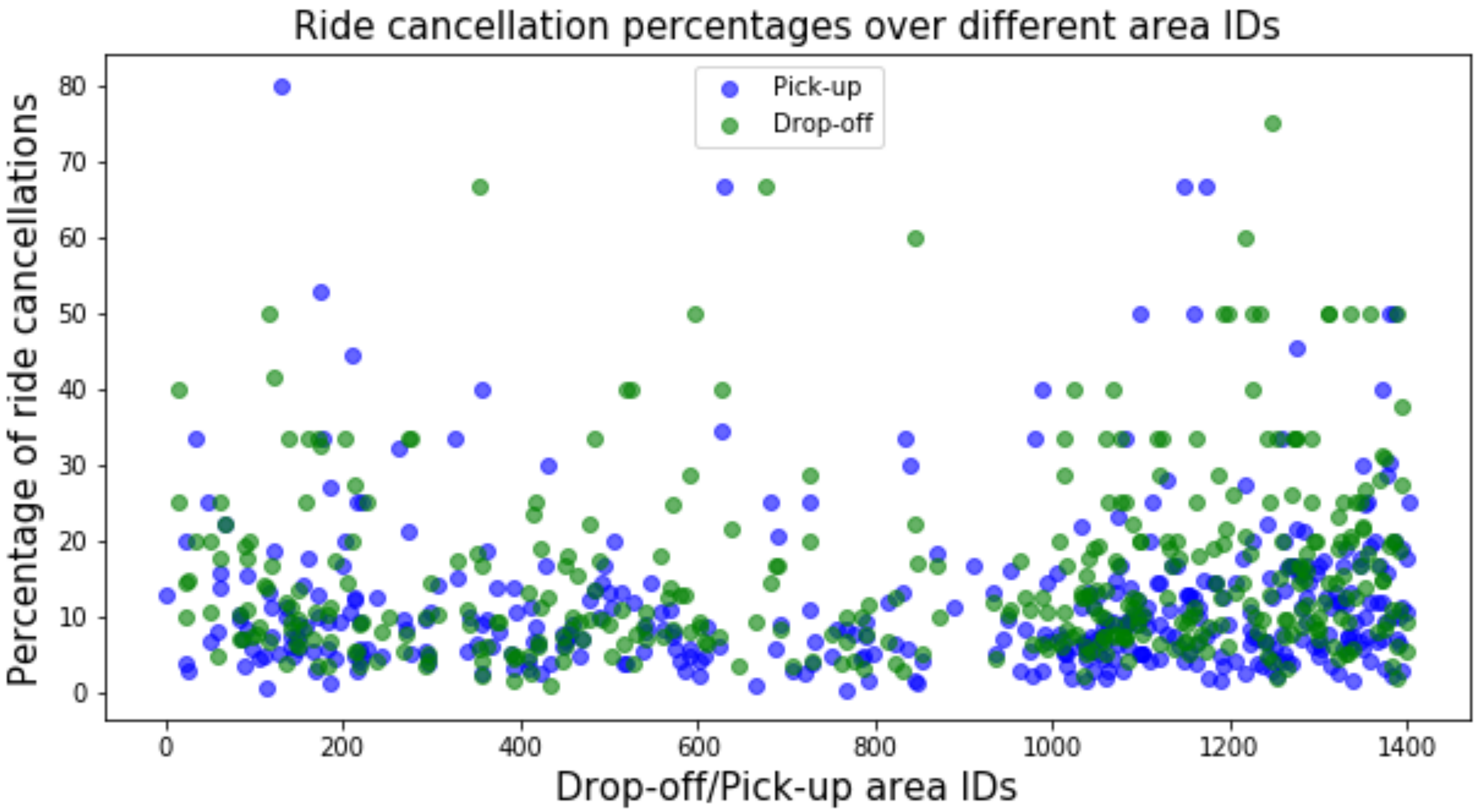
EDA – continued:

Pick-up/Destination Area ID:

- 598 unique origin and 568 destination area information have been listed.
- The most popular origin area is the area with ‘area_id’ no. '393', which is eventually the most popular destination area as well.
- 559 area IDs are common to both as the pick-up and drop-off locations.



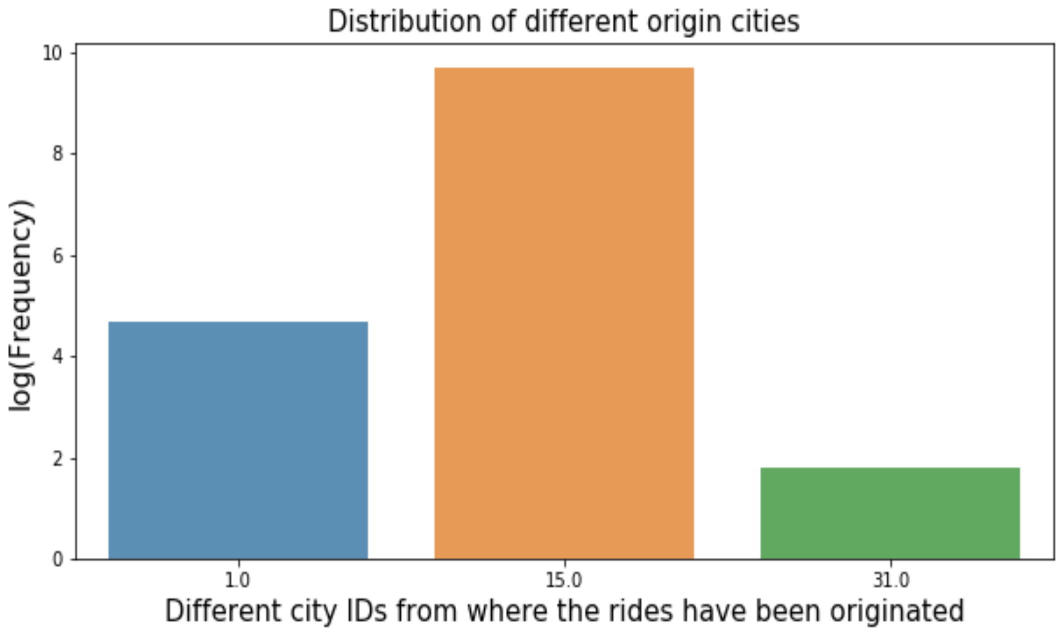
EDA – Area ID- continued:



EDA – continued:

Origin/Destination city ID :

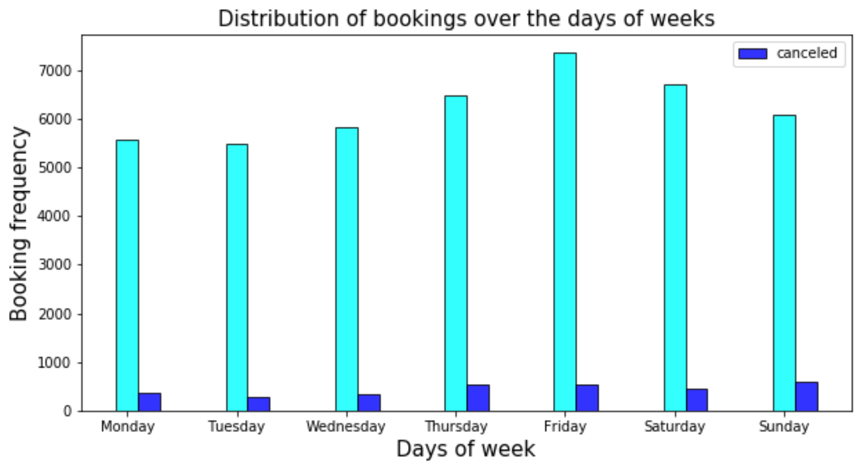
- Only 3 origin cities have been recorded.
- The most popular origin city is the city with the ID no: '15'. Whereas, 116 unique destination cities are there.
- The most popular destination city is the city with the ID no: '32' (475 rides have their destinations to this city.)
- However, most of the information is missing.

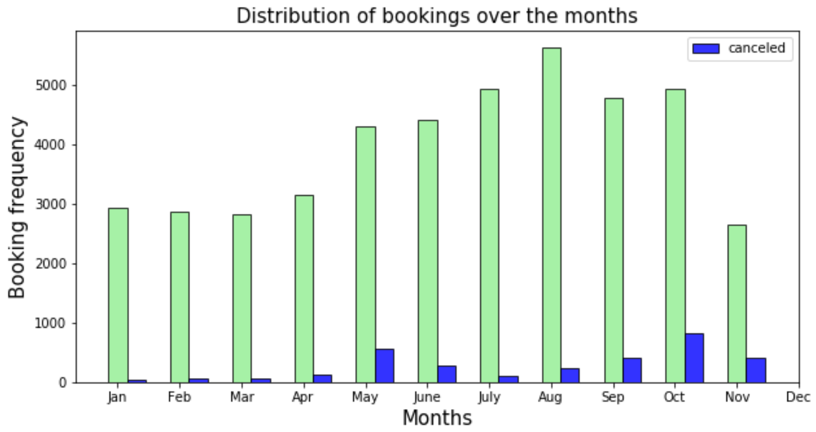


EDA – continued:

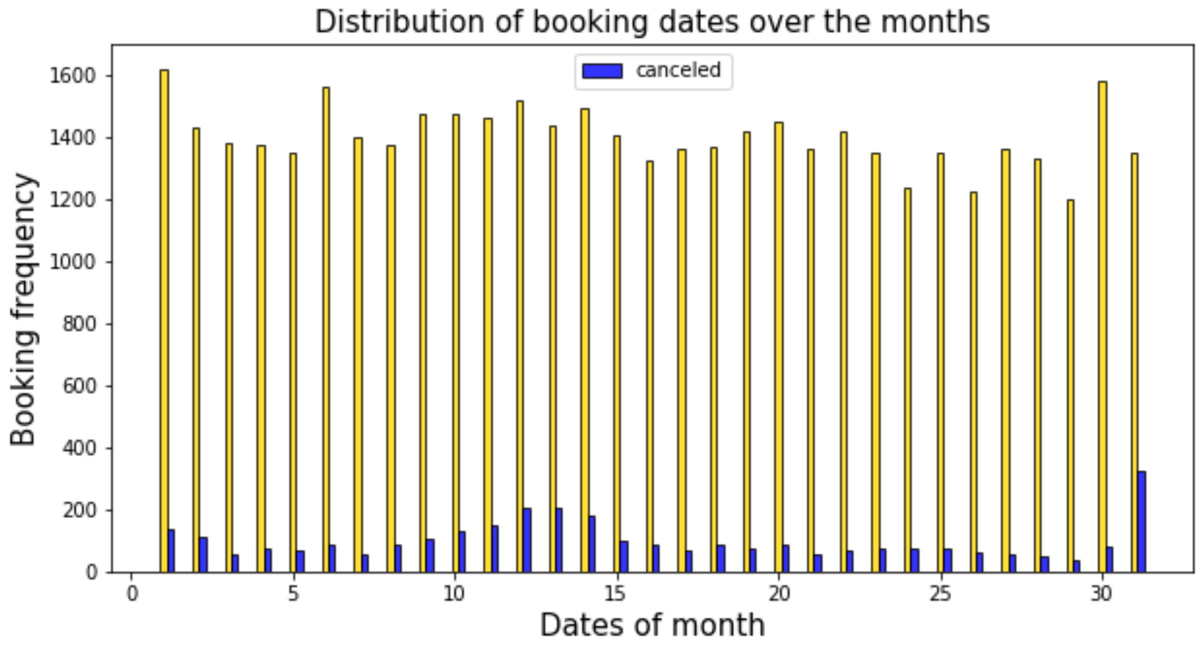
Booking time:

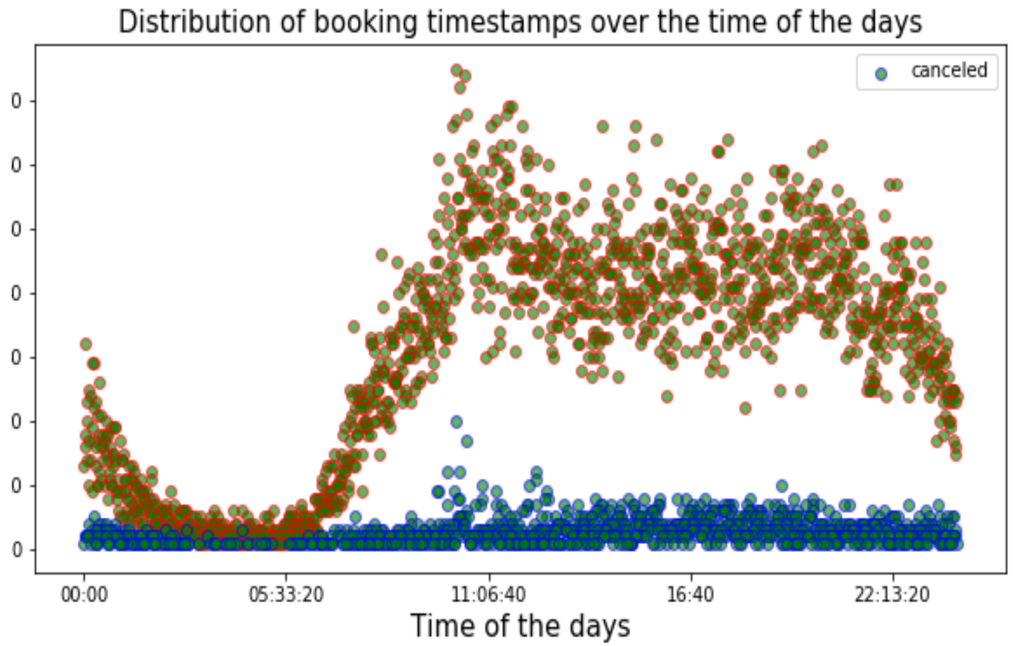
- Timestamp of the booking (when somebody booked the cab).
- Maximum no. of bookings made at a given timestamp is, 18. Corresponding date-time is 2013-10-31 10:30:00.
- Maximum bookings were made on Fridays.
- Bookings were made almost equally throughout the month.
- Maximum bookings were made in August.





Booking Time- EDA – continued:

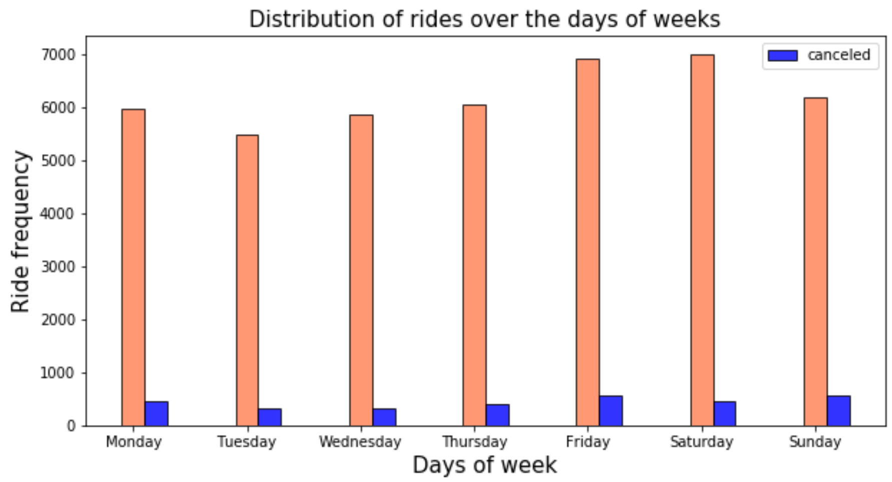


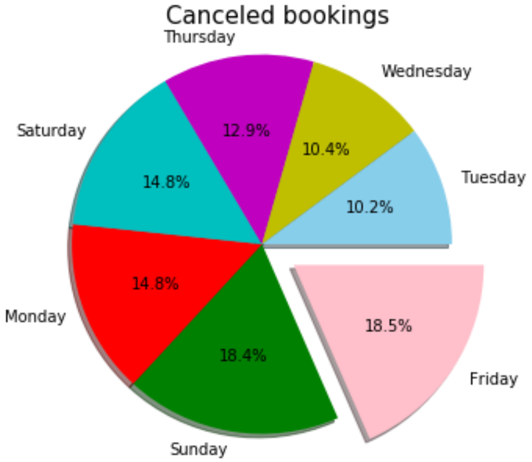


EDA – continued:

Timestamp of the actual ride:

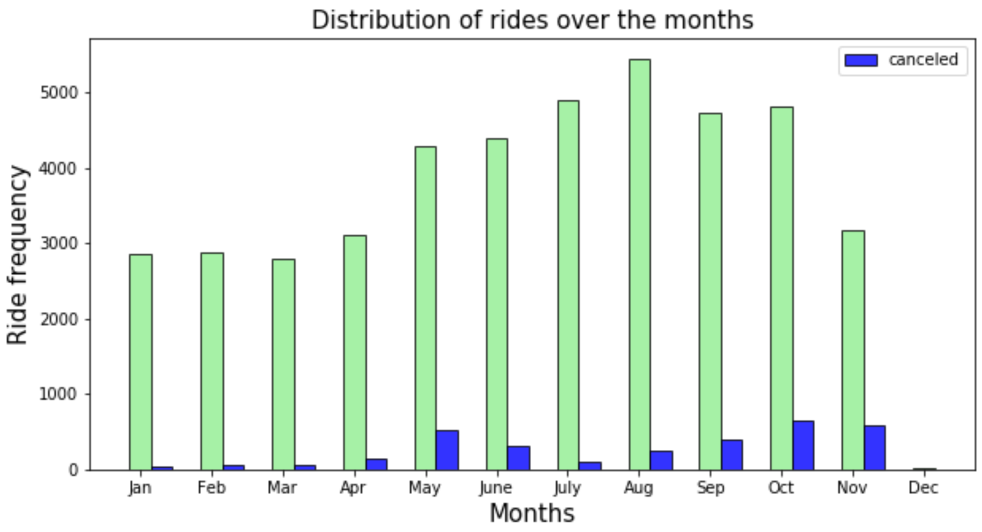
- Timestamp of the actual rides.
- Maximum no. of trips started at a given timestamp is, 20 and the corresponding date-time is: 2013-10-12 06:00:00 and 2013-07-04 22:15:00.
- Maximum frequency (6990) of rides correspond to Saturday,' followed by 'Friday.'
- The maximum cancellations (578) correspond 'Friday,' followed 'Sunday.'

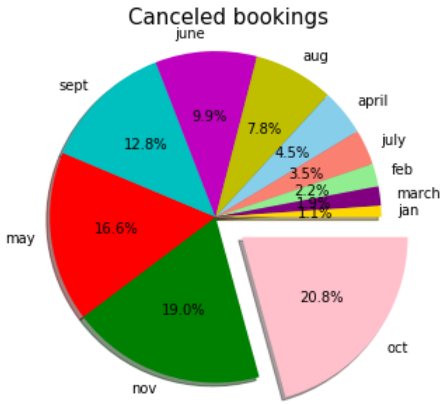




EDA – continued:

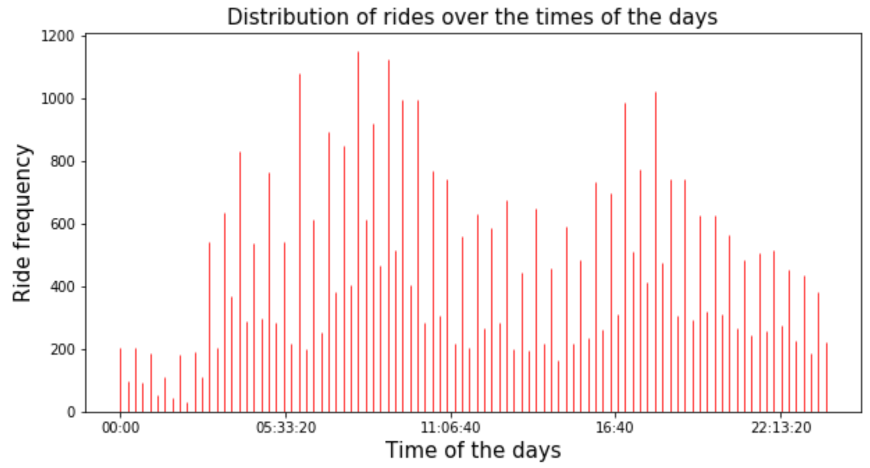
- Extracted the ride frequency over the months of the year.
- Maximum frequency (5445) corresponds to the month of 'August,' followed by 'July.'
- On the same figure, we have plotted the canceled ride frequencies. Maximum cancellation (650) correspond to the month of 'October,' followed by 'November.'

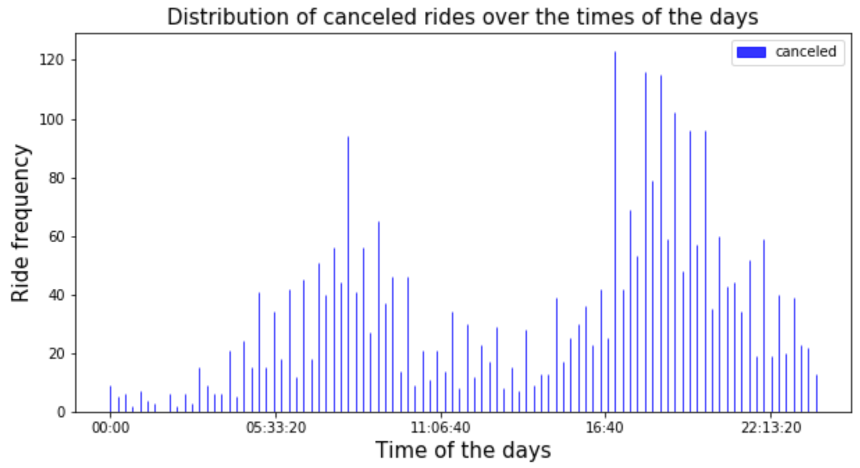




EDA – continued:

- These are the frequencies of the rides across different times of the day.
- The two humps/clusters in the distributions of the ride frequencies. One is around the morning and another for the evening time.
- The ride cancellation distribution also follows the same trend. Maximum numbers of rides got canceled in these two peak hours.





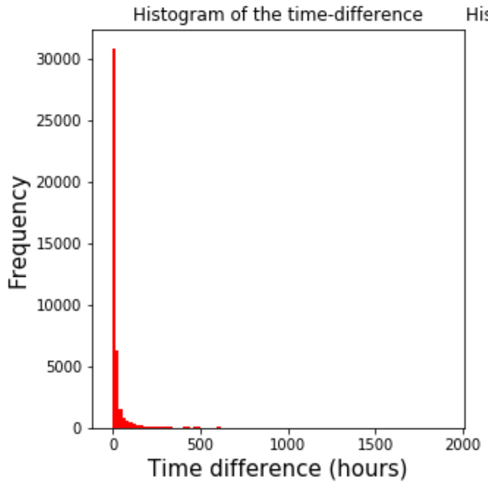
EDA – continued:

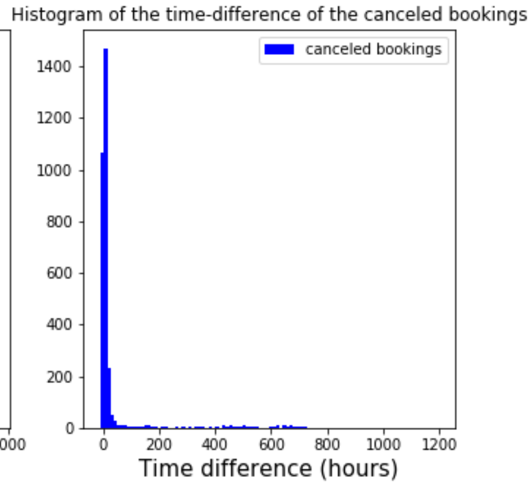
Time difference:

- This is the numerical feature created.
- This is the difference in the timestamps (in hours) between the ‘booking created’ and the ‘trip start time’.
- There are 42 entries of the dataset, for which the time difference is negative, which is unphysical. – Dropped.

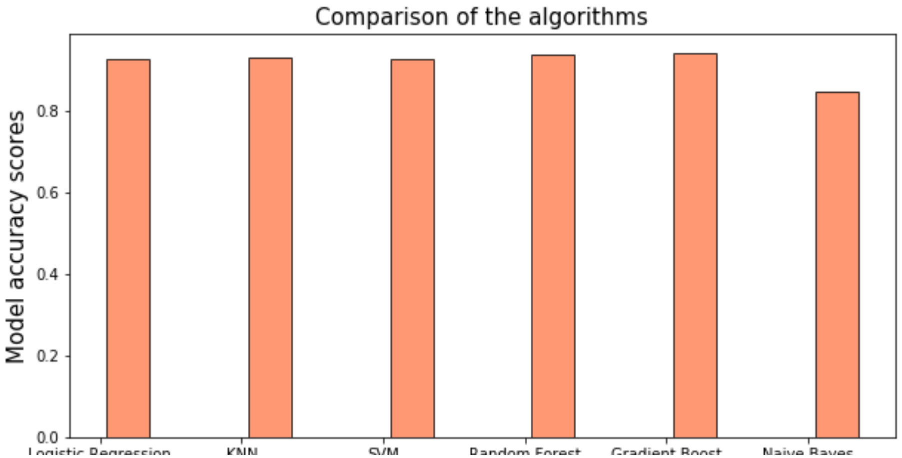
count	43389.000000
mean	33.976458
std	94.274862
min	0.000000
25%	2.900000
50%	8.750000
75%	18.333333
max	1906.900000

Descriptive Statistics

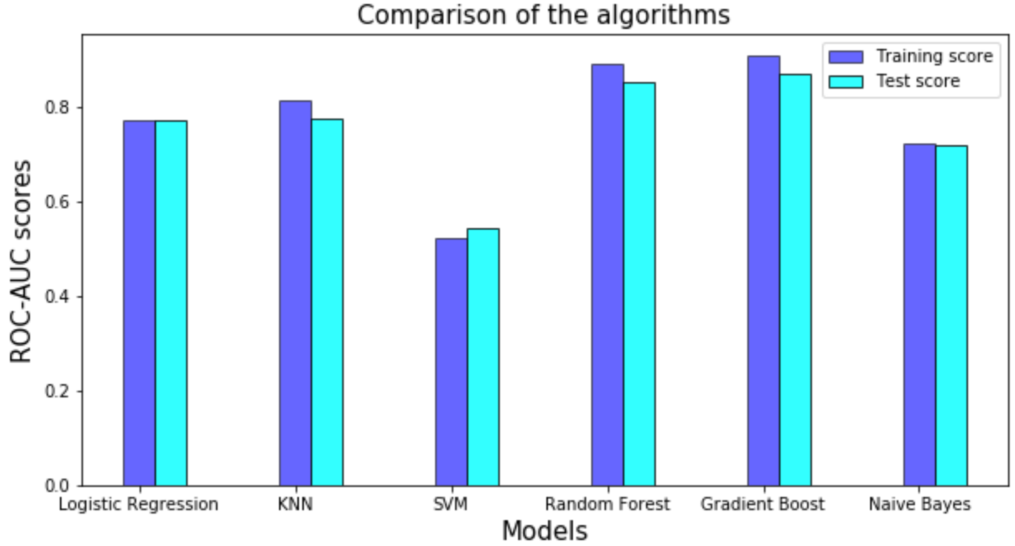




Applying Machine Learning models and comparing their performances:



	Algorithm	Model accuracy score
0	Logistic Regression	0.928248
1	KNN	0.930936
2	SVM	0.928478
3	Random Forest	0.939771
4	Gradient Boost	0.941077
5	Naive Bayes	0.845356



	Algorithm	ROC-AUC train score	ROC-AUC test score
0	Logistic Regression	0.772894	0.771349
1	KNN	0.812429	0.774589
2	SVM	0.521361	0.544122
3	Random Forest	0.890843	0.852856
4	Gradient Boost	0.908122	0.870345
5	Naive Bayes	0.722899	0.718472

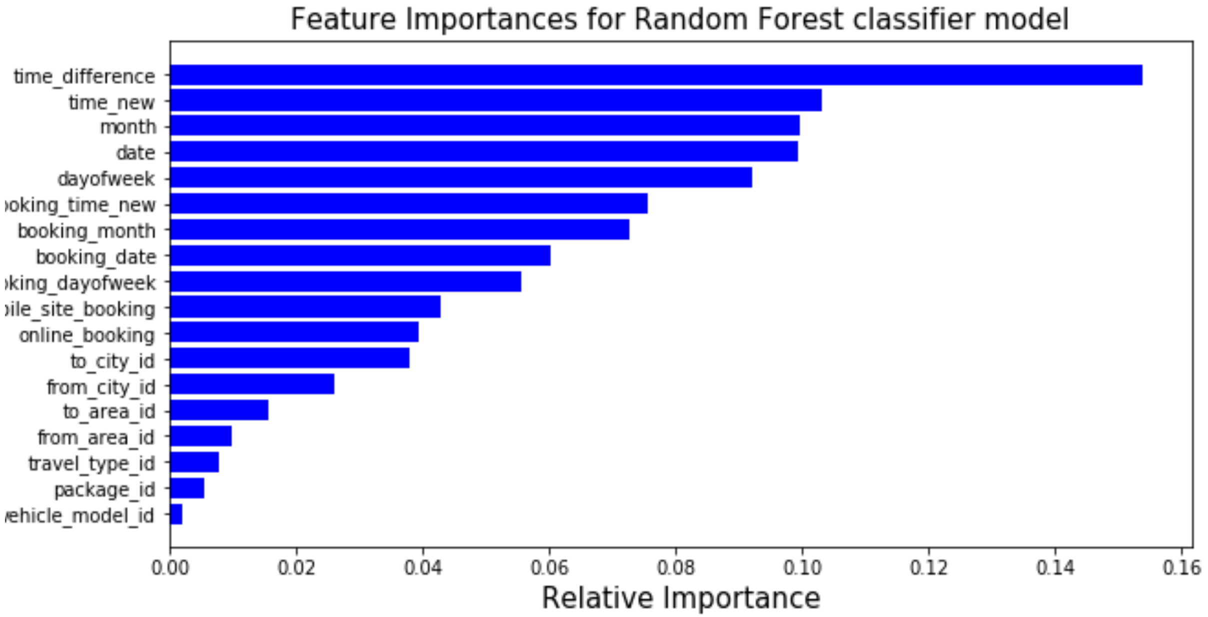
Hyperparameter Tunning:

- the **Gradient Boost**, and the **Random Forest** are the two best performing models.
- Performed the hyperparameter tuning, through the gridsearch, for the two ML models.
- Fitting these models with optimized hyperparameters (found through the grid search), we evaluated the model performance in terms of **ROC-AUC** score.

Model	ROC-AUC Score
Random Forest (RF)	0.8860217314758018
Gradient Boost (GB)	0.8987293089109146

Feature Importance (RF):

	Features	Importance scores
0	vehicle_model_id	0.002008
1	package_id	0.005594
2	travel_type_id	0.007688
3	from_area_id	0.009800
4	to_area_id	0.015770
5	from_city_id	0.026119
6	to_city_id	0.037965
7	online_booking	0.039360
8	mobile_site_booking	0.042919
9	booking_dayofweek	0.055539
10	booking_date	0.060290
11	booking_month	0.072862
12	booking_time_new	0.075725
13	dayofweek	0.092141
14	date	0.099270
15	month	0.099709
16	time_new	0.103225
17	time_difference	0.154015



*** Same For GB**

Conclusion:

- Two final prediction (result) files **'final_result_gb.csv'**, and **'final_result_rf.csv'**.
- Two columns, named **User ID** and **Car_cancellation**. It shows the prediction for cab booking cancellations (0 if there no cancellation. 1 for cancellation) corresponding to the user_IDs.
- What the company can do is:

1. Run the model at every one-hour interval
2. Call the customer who is flagged by the model
3. Confirm with the customer if the booking will be canceled or not
4. Send cab only after the confirmation from the customer

Future Direction:

- Here we have used only the data of one year. The model can be improved, if we can use the data from at least another year.
- Use ensembles of the machine learning models to average out bias and improve performance.
- Try to use more feature engineering. Especially, here we have neglected the Latitude/longitude (GPS data) info. We could have extracted the route information out of them, and use that as a feature.
- Try to fit and predict using the Extreme Gradient boost classifier model.

Acknowledgement:

- Mentor: Max Sop
- Kaggle
- Springboard Team
- Cover Image: Internet

For detailed analysis:

https://github.com/debisree/Springboard_Debisree/tree/master/Capstone_1_predicting_cab_booking_cancellation

Thank you!