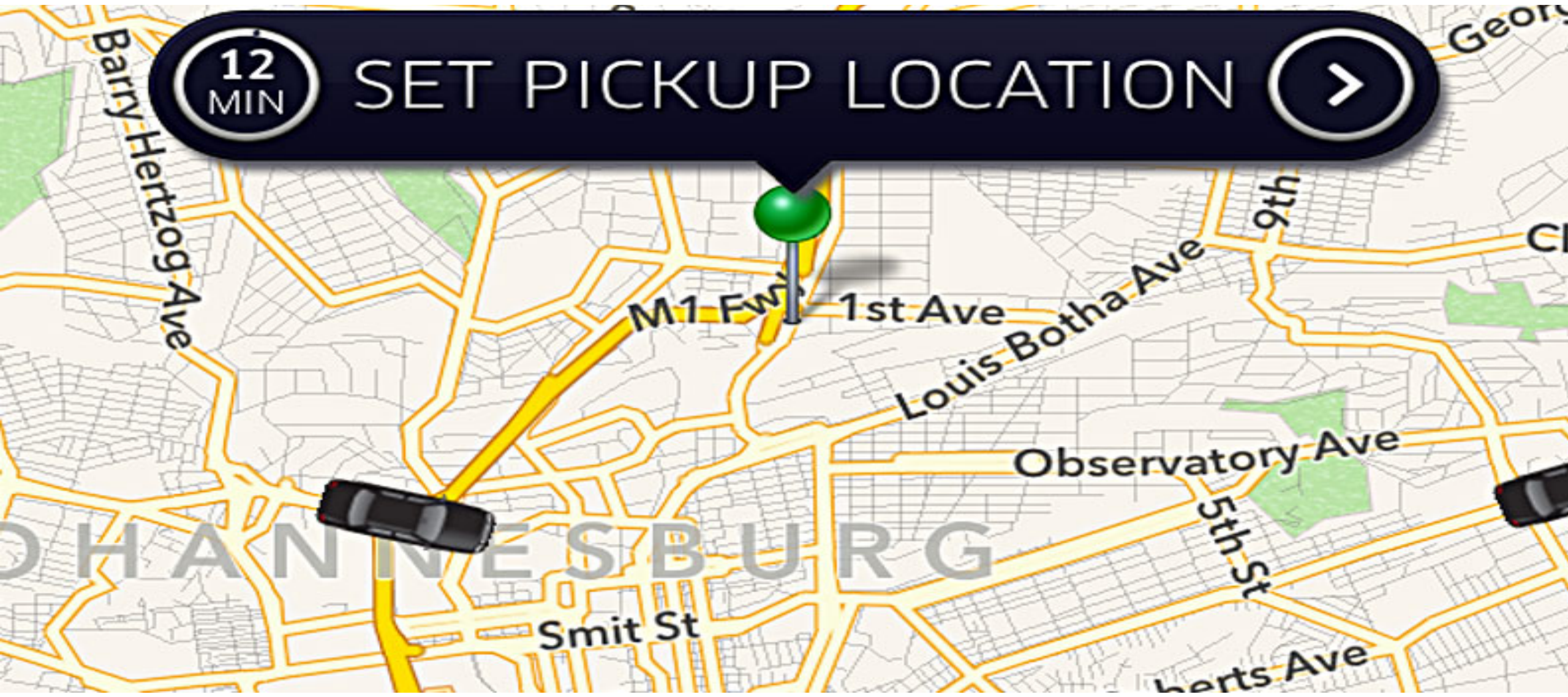


Predicting Cab Booking Cancellations

by Devesh Khandelwal

Agenda

- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

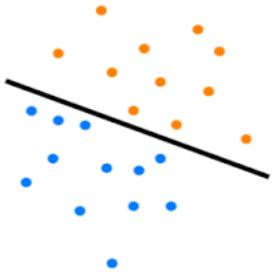
Problem Statement

Customers can cancel the booking up to the **last minute** of pick up at **no cost** to them

Cancelled booking dents the revenue of the company and adds operational overheads

Use the Data collected over time to predict the probability of booking cancellation

Problem Analysis



Classification Task – Classify the Cancellation feature into :

- ✓ ‘0’ (Not Cancelled)

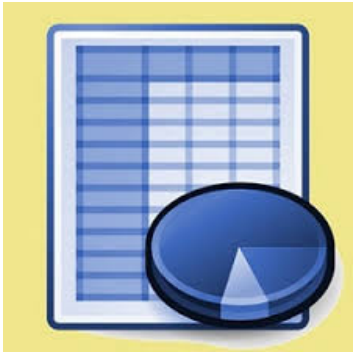
  or

- ✓ ‘1’ (Cancelled)

Agenda

- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

Dataset



Training Data-

- ✓ 43 K records
- ✓ 18 Features

Uneven Classes

- ✓ Approx 7% of the total bookings are actually Cancelled(Training Data)

Source:- https://inclass.kaggle.com/c/predicting-cab-booking-cancellations/data

Features at a Glance

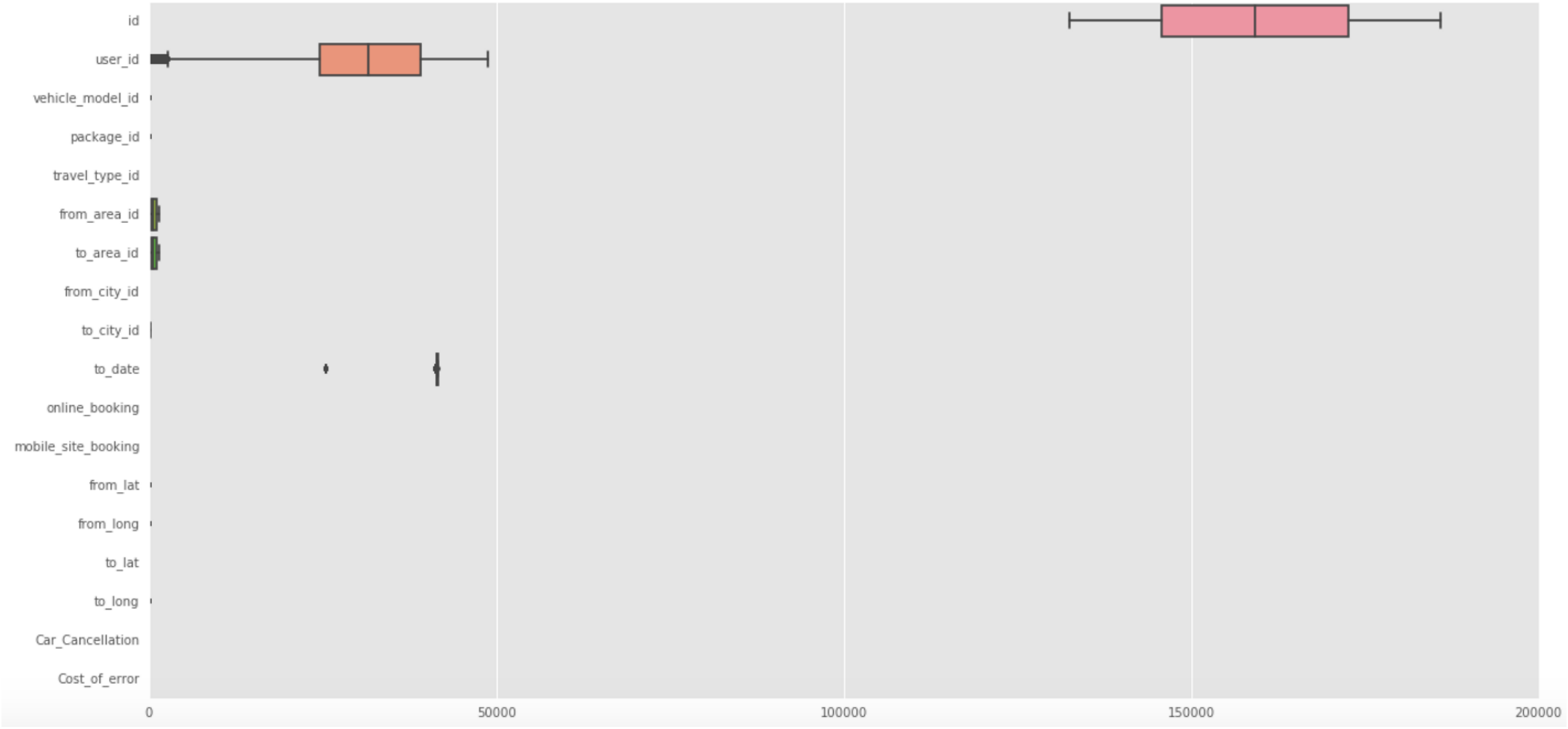
Features set includes:

✓ Vehicle attributes

✓Booking attributes including-

- Online
- GPS data
- Mobile
- Travel Type
- Source
- Destination

Features at a Glance(Contd..)

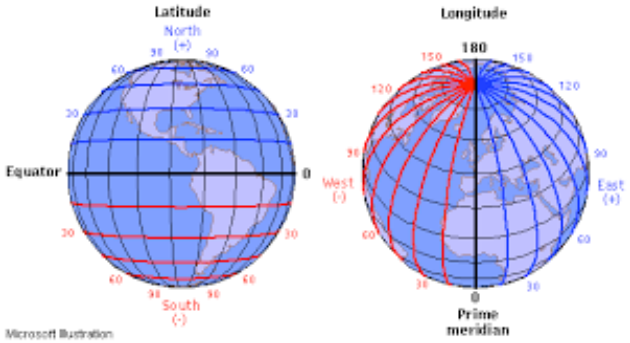


Agenda

- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

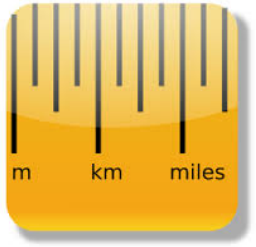
Feature Engineering

(GPS Data)



Booking Coordinates (Latitude ,longitude of source & Destination)

Transformed to



New feature 'Distance'

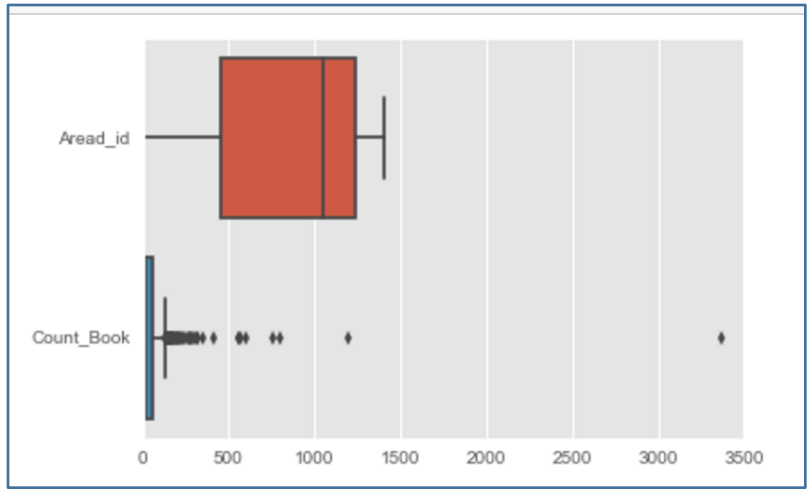
Implementation

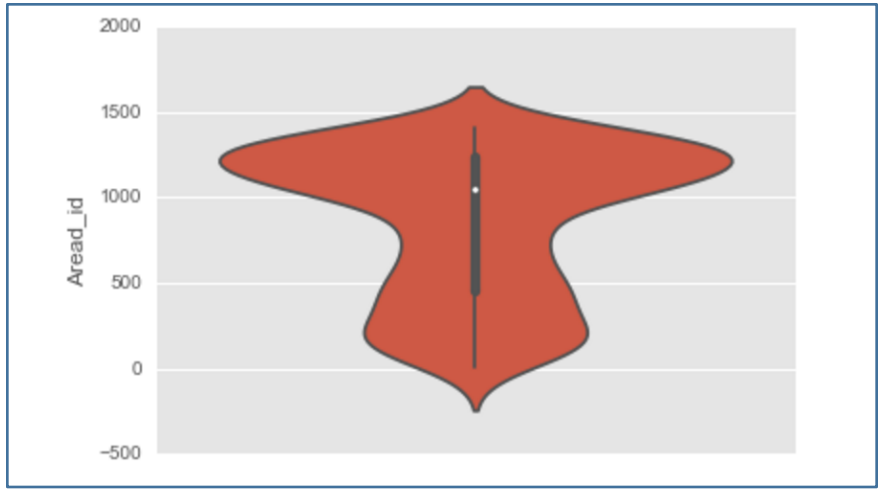
- df['distance'] = 6367 * 2 * np.arcsin(np.sqrt(np.sin(np.radians(df['to_lat']) - math.radians(37.2175900)/2)**2 + math.cos(math.radians(37.2175900)) * np.cos(np.radians(df['to_lat']) * np.sin(np.radians(df['from_long']) - math.radians(-56.7213600)/2)**2)))
- df['distance']=df.distance/1000
- df.distance = df.distance.apply(replace_null)

Feature Engineering

(Area information)

- Data set has features **from_area_id** and **to_area_id** that depicts the location of the origin and destination
- 599 unique values for feature- '**Area_id**'



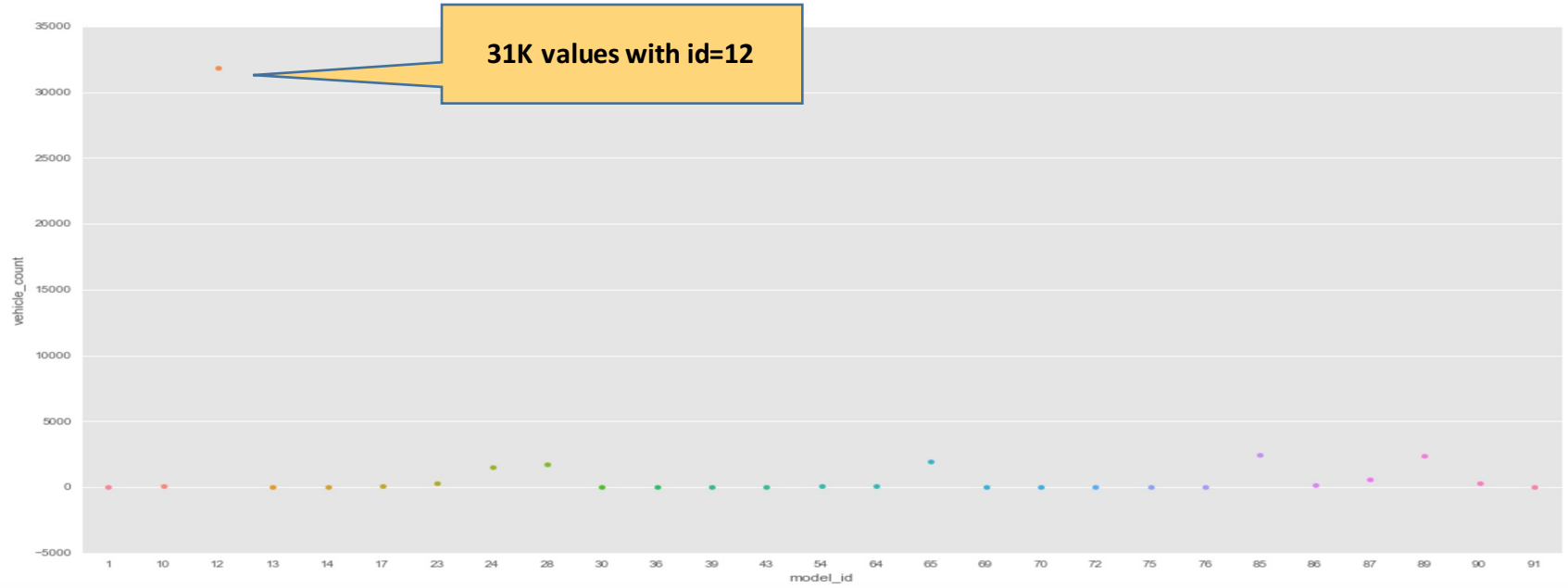


- Majority of the bookings cater to a few of the areas as is evident from the density function
- New feature 'Popular_Pickup'=0 if area_id of the booking is not from the popular_area and 1 otherwise
- New feature 'Popular_Drop'=0 if area_id of the booking is not from the popular_area and 1 otherwise

Feature Engineering

(Fleet Analysis)

Vehicle_Model_id- 16 unique values



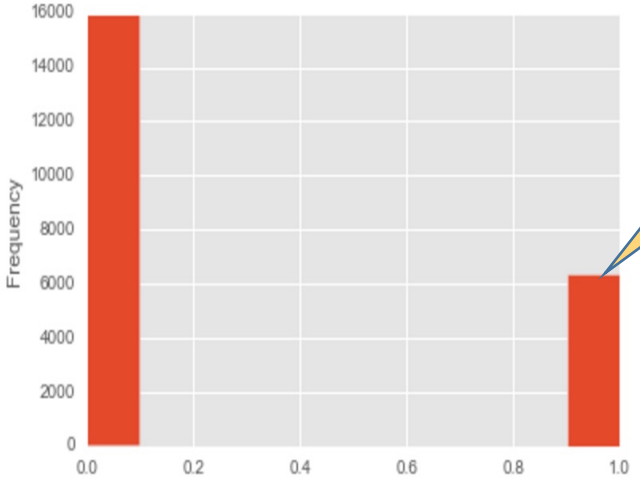
- Creating new_feature- vehicle_category

- cat_1 = vehicle_cat_df.vehicle_count.max()
- cat_2 = round(vehicle_cat_df.vehicle_count.quantile(.75))
- cat_3 = round(vehicle_cat_df.vehicle_count.quantile(.5))
- cat_4 = round(vehicle_cat_df.vehicle_count.quantile(.25))

Feature Engineering

(User segmentation)

User_id – Id of the user requesting the service



Distribution of User_id

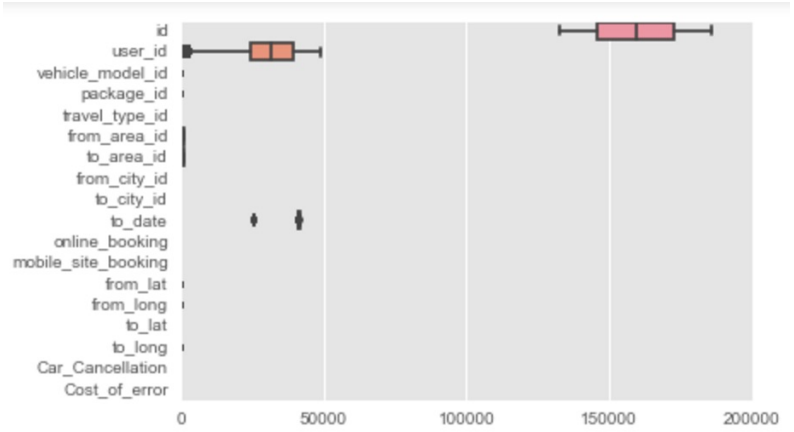
Transformed to

New Feature – is_frequent

- ✓ Is_frequent = 1 (returning user)
- ✓ Is_frequent = 0 (one time user)

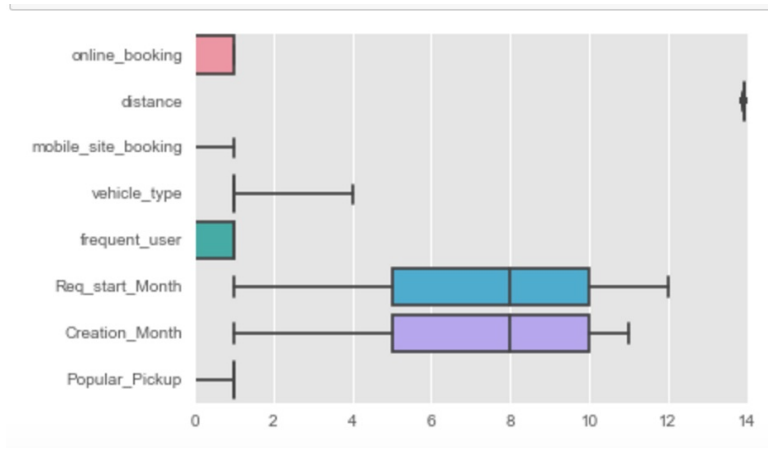
Feature Engineering

(Summary)



Raw Data set

Transformed to



Transformed Data
(final data set)

Stratified Sampling

- Uneven Data Set- less than 7% of the booking are cancelled
- Creating a balanced data set with equal distribution of dependent variable

```
y_0 = df[df.Car_Cancellation == 0]
y_1 = df[df.Car_Cancellation == 1]
n = min([len(y_0), len(y_1)])
y_0 = y_0.sample(n = n, random_state = 0)
y_1 = y_1.sample(n = n, random_state = 0)df_strat = pd.concat([y_0, y_1])
X_strat = df_strat[['online_booking','distance','mobile_site_booking','vehicle_type','frequent_user','Req_start_Month','Creation_Month','Popular_Pickup']]y_strat = df_strat.Car_Cancellation
```

Agenda

- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

Modelling-Stats Model

(Kitchen Sink Strategy)

Output of Stats Model

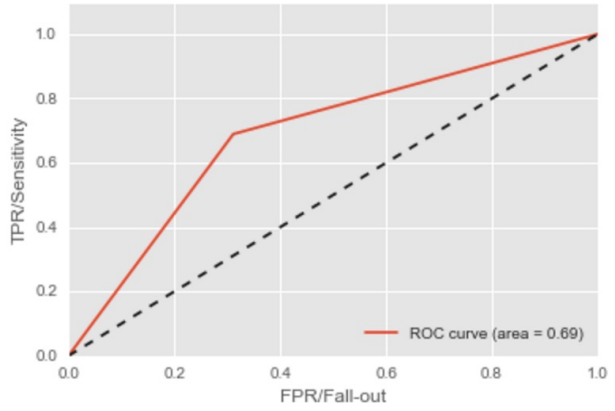
	coef	std err	z	P>|z|	[95.0% Conf. Int.]
const	-908.5756	4799.228	-0.189	0.850	-1.03e+04 8497.738
online_booking	1.2302	0.047	26.333	0.000	1.139 1.322
distance	63.2429	2.440	25.923	0.000	58.461 68.024
mobile_site_booking	1.3237	0.080	16.562	0.000	1.167 1.480
vehicle_type	-0.8444	0.056	-15.117	0.000	-0.954 -0.735
travel_type_id	12.8902	2399.554	0.005	0.996	-4690.149 4715.929
frequent_user	-0.7271	0.043	-16.901	0.000	-0.811 -0.643
Req_start_Month	0.7830	0.077	10.134	0.000	0.632 0.934
Creation_Month	-0.5925	0.078	-7.583	0.000	-0.746 -0.439
Popular_Pickup	-0.3916	0.049	-7.946	0.000	-0.488 -0.295
Popular_Drop	-0.1377	0.048	-2.867	0.004	-0.232 -0.044

- Kitchen Sink strategy on the Data set further reduces the features
- Travel_type_id gets eliminated from further analysis due to the higher p value

Modelling

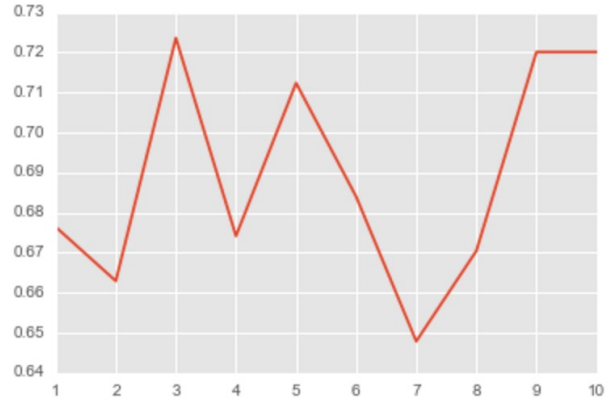
(Logistic Regression)

Training



- 69% Accuracy on the Training Data

Cross Validation



- 69% mean Accuracy on the CV Data(10 folds)

Test Data

```
model.score(test_X_strat,test_y_strat)
```

```
0.6999999999999996
```

Modelling

(Decision Trees)

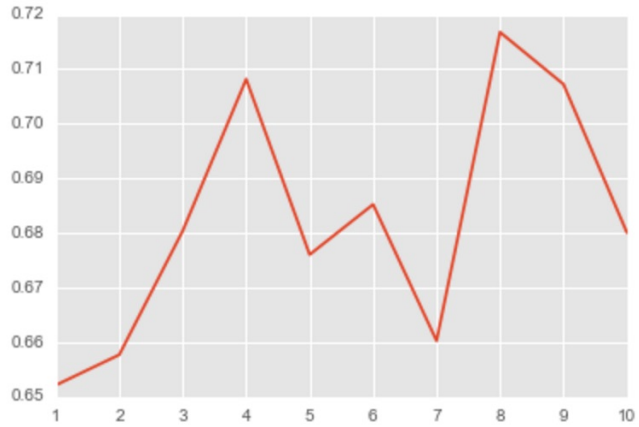
Training

```
model_tree.score(train_X_strat, train_y_strat)
```

```
0.96877189424135701
```

- 97 % Accuracy on the Training Data

Cross Validation



- 68.2% mean Accuracy on the CV Data(10 folds)

Test Data

```
model_tree.score(test_X_strat , test_y_strat)
```

```
-0.20076622358025387
```

```
(tree_y_hat == test_y_strat).mean()
```

```
0.67927927927927922
```

Modelling

(Random Forests - no of trees=10000)

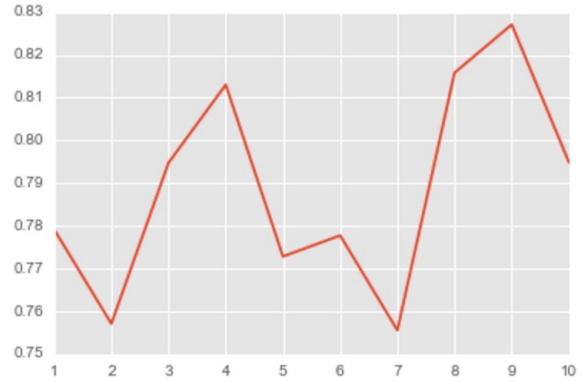
Training

```
model_forest.score(train_X_strat, train_y_strat)
```

```
0.98626126126126124
```

- 98 % Accuracy on the Training Data

Cross Validation



- 79% mean Accuracy on the CV Data(10 folds)

Test Data

```
model_forest.score(test_X_strat, test_y_strat)
```

```
0.71621621621621623
```

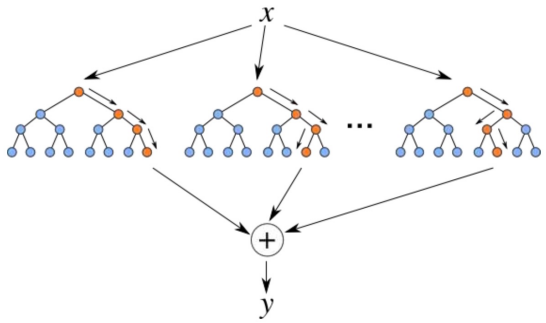
Modelling

(Random Forests-Feature Importance & Co-relation)

Feature	%age	Co-Relation with the dependent variable
distance	62.4	0.261690
Creation_Month	10.4	0.262376
Req_start_Month	9.1	0.262179
online_booking	6.2	0.255332
frequent_user	4.1	-0.158572
vehicle_type	3.3	-0.154804
mobile_site_booking	2.2	0.104083
Popular_Pickup	1.9	-0.056936
Total	96	

Modelling

Conclusion

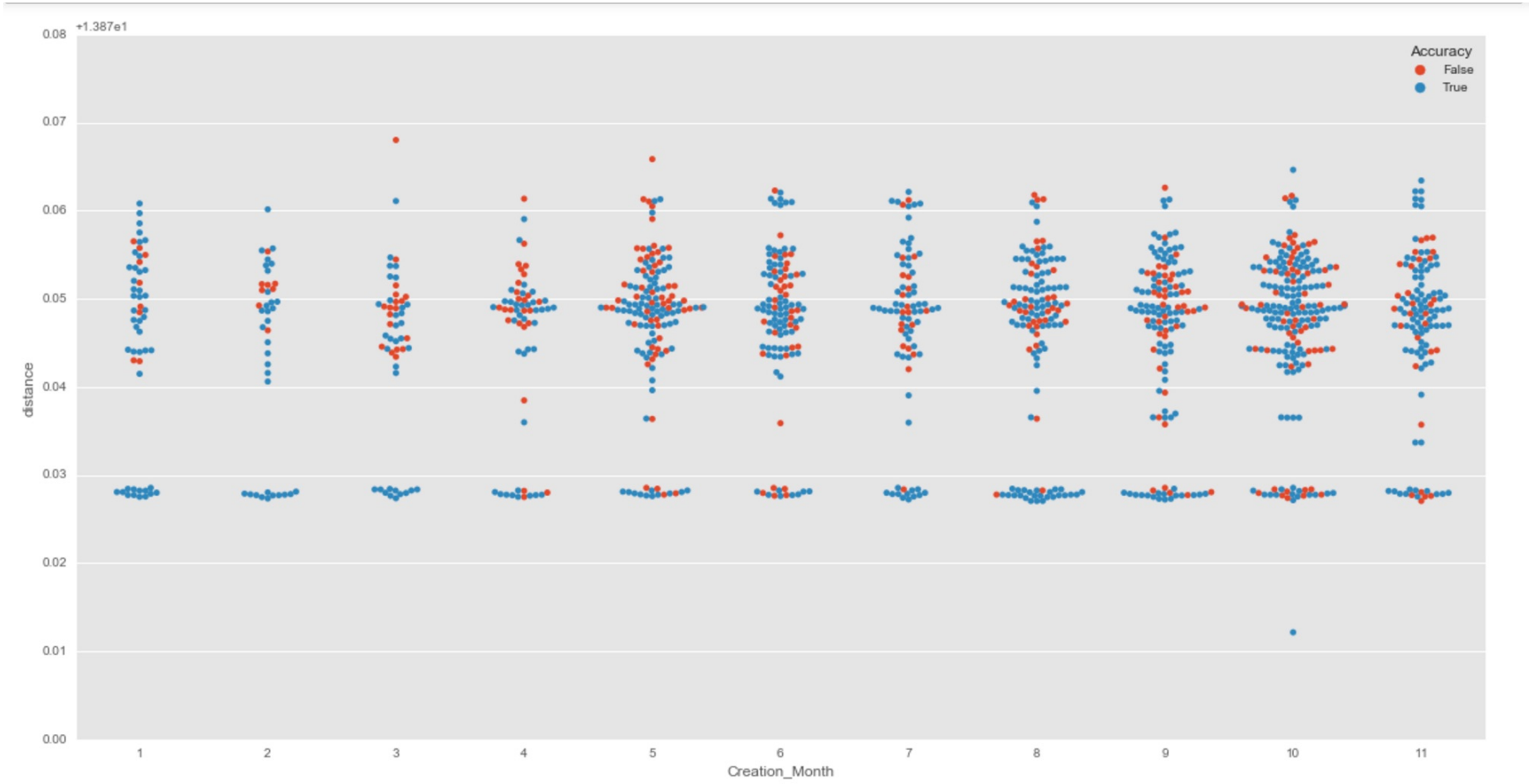


- Random forest seems to be the best amongst all the models
- Random forest also seem to cut off the nose and make the best decision on the important features
- Chance of over -fitting is less as compared to Decision trees(which is most likely to have overfit – Training score of 97%

Agenda

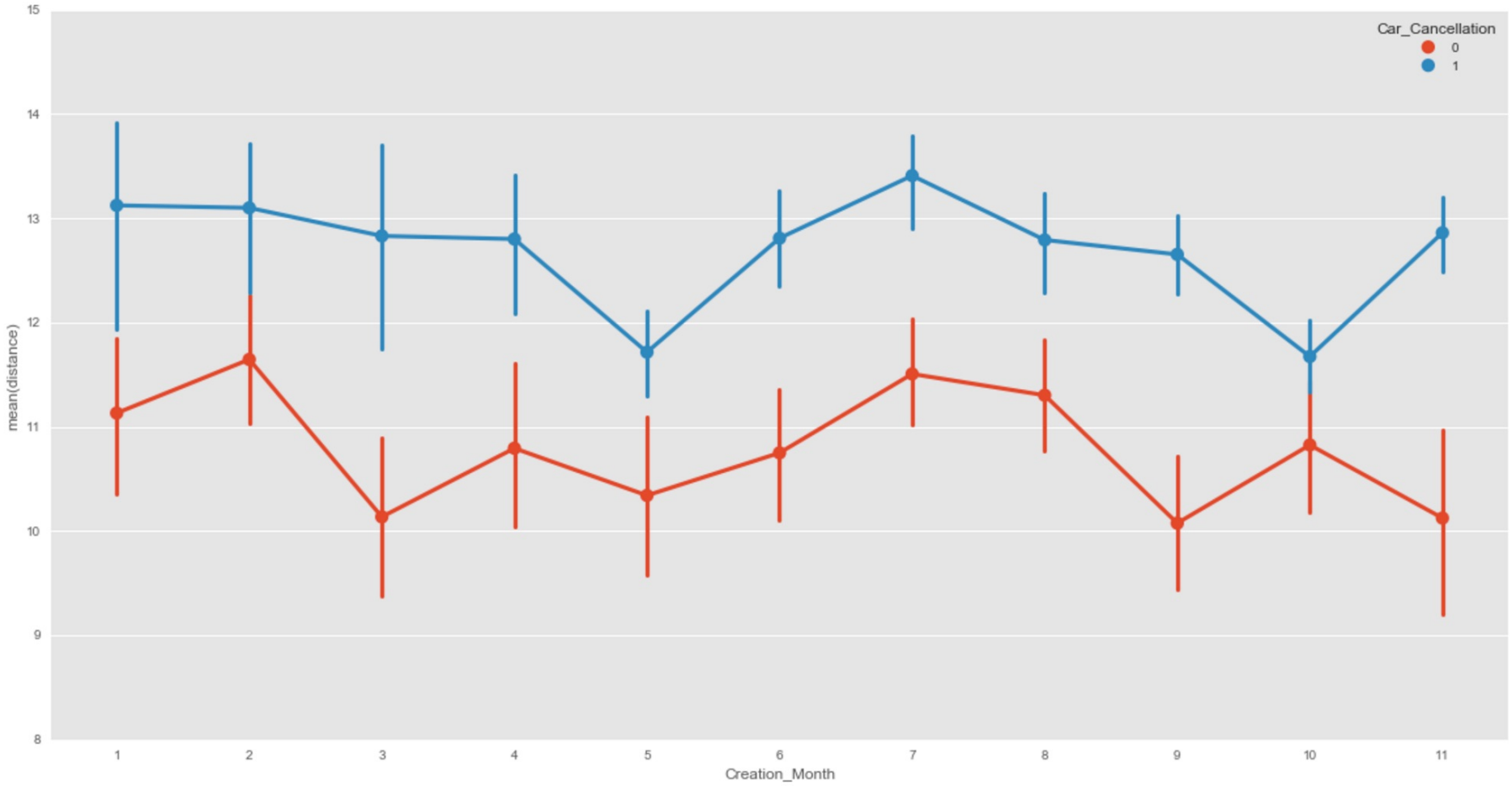
- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

Model Accuracy
(Random Forest on Test set)



- Appears that the Maximum number of misclassifications are occurring in Apr,May

Interpretation

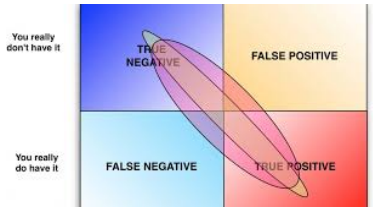


- Appears that the chances for the cancellation is maximum in Jul when the mean travel distance is between 13 -14 KMs

Next Steps

Cyclic- The booking cancellation seems to follow a cycle with period-5. Further Time series analysis could give interesting insight

Gap Analysis – Further feature engineering to reduce the gap between Training and CV score



Testing Accuracy– Further improve the test accuracy by analyzing the model

Next Steps

(integration ideas with Ride sharing Apps)

Push Notifications- For booking that have a high chance of cancellations send a push notification to customer ,seeking reconfirmation

Fleet reduction- For those months that have a high chance of cancellations consider reducing the fleet size

Decline the Booking- if the distance is less and booking has a high probability for cancellation- Don't Accecpt the booking

Note- This can hamper customer satisfaction and can turn away users

References

Technical Reference and Source code can be downloaded from:
Git Hub

Questions/Feedback