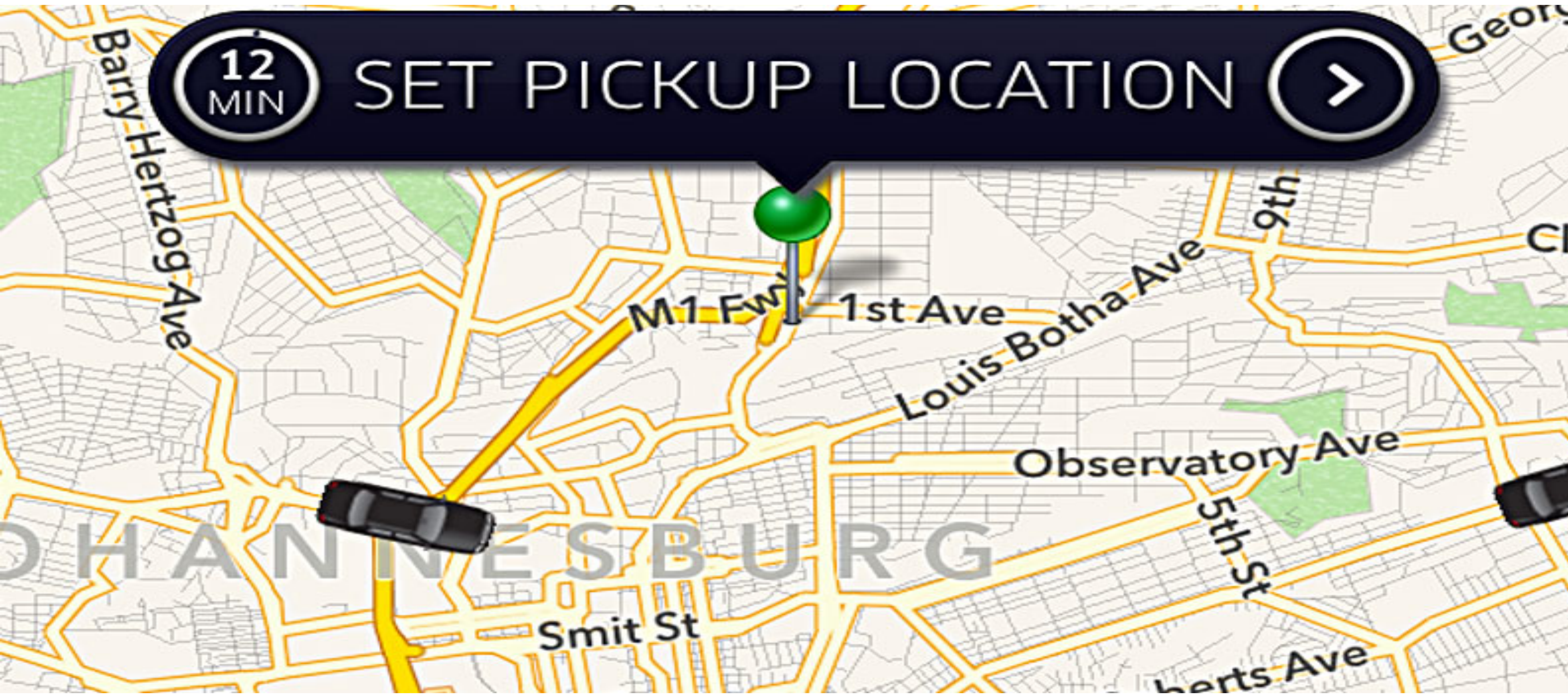


Predicting Cab Booking Cancellations

by Devesh Khandelwal

Agenda

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

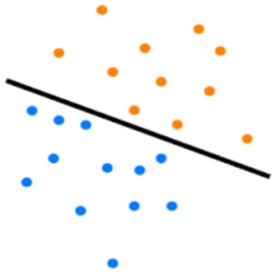
Problem Statement

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

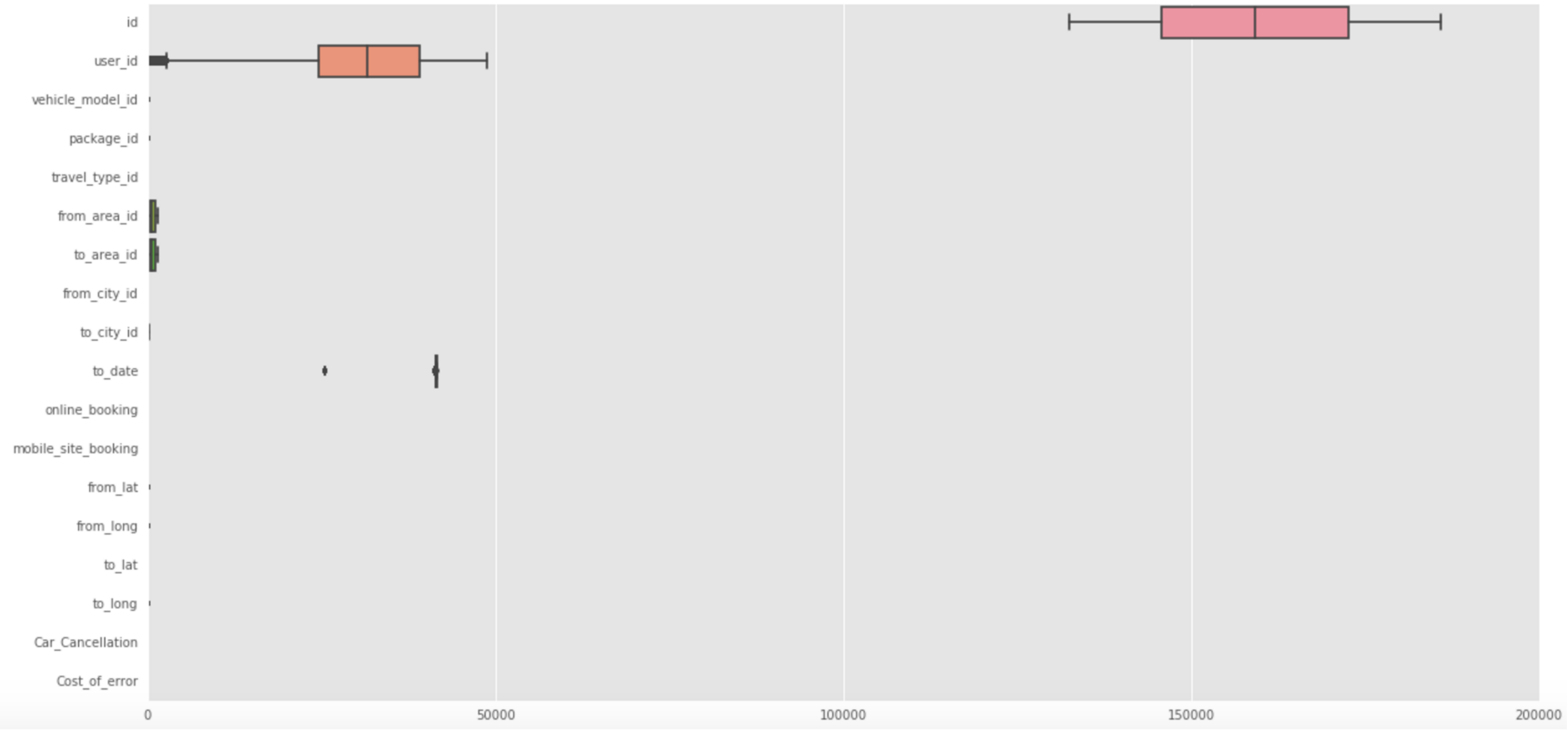
Features set includes:

✓ Vehicle attributes

✓Booking attributes including-

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

Features at a Glance(Contd..)

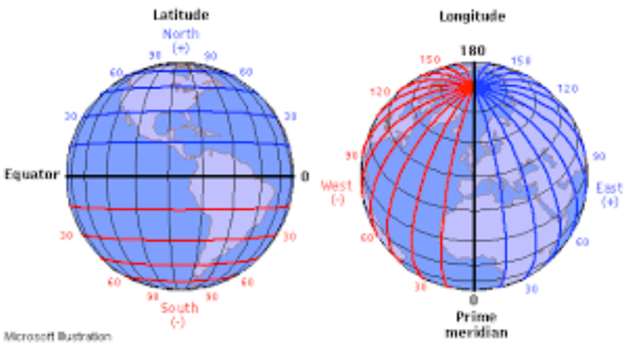


Agenda

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

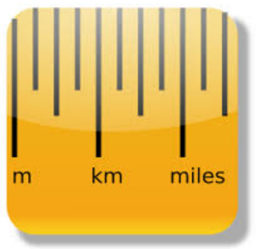
Feature Engineering

(GPS Data)



Booking Coordinates
(Latitude ,longitude of
source & Destination)

Transformed to



New feature 'Distance'

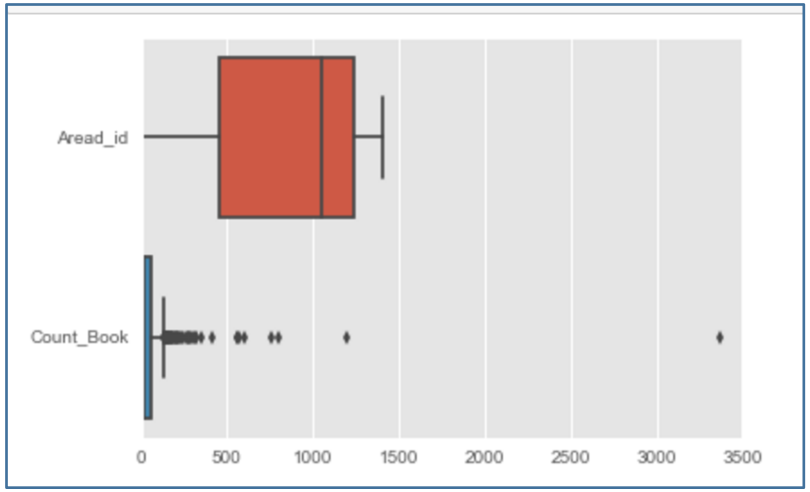
Implementation

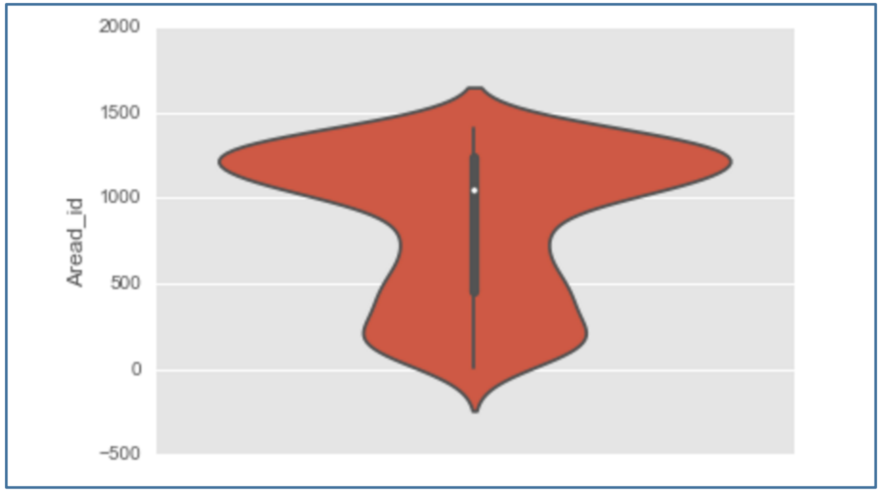
- df['distance'] = 6367 * 2 * np.arcsin(np.sqrt(np.sin(np.radians(df['to_lat']) - math.radians(37.2175900)/2)**2 + math.cos(math.radians(37.2175900)) * np.cos(np.radians(df['to_lat']) * np.sin(np.radians(df['from_long']) - math.radians(-56.7213600)/2)**2)))
- df['distance']=df.distance/1000
- df.distance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



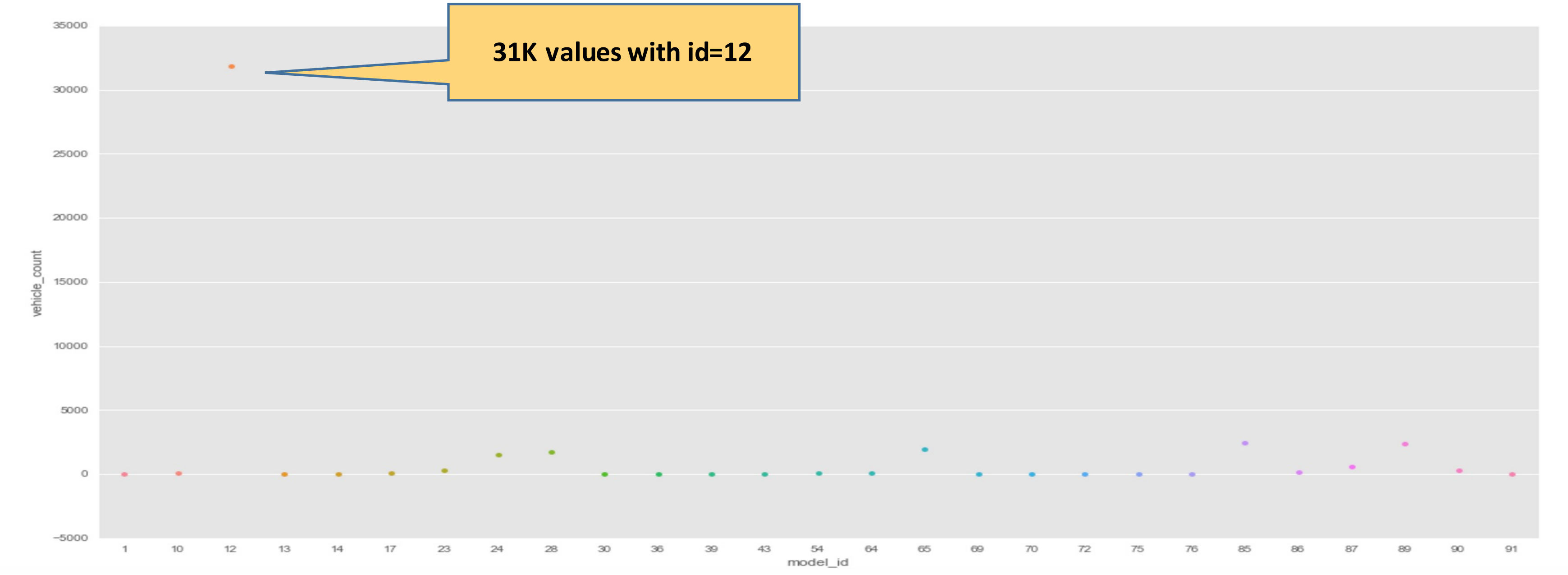


- Majority of the bookings cater to a few of the areas as is evident from the density function
- New feature 'Popular_Pickup'=0 if area_id of the booking is not from the popular_area and 1 otherwise
- New feature 'Popular_Drop'=0 if area_id of the booking is not from the popular_area and 1 otherwise

Feature Engineering

(Fleet Analysis)

Vehicle_Model_id- 16 unique values



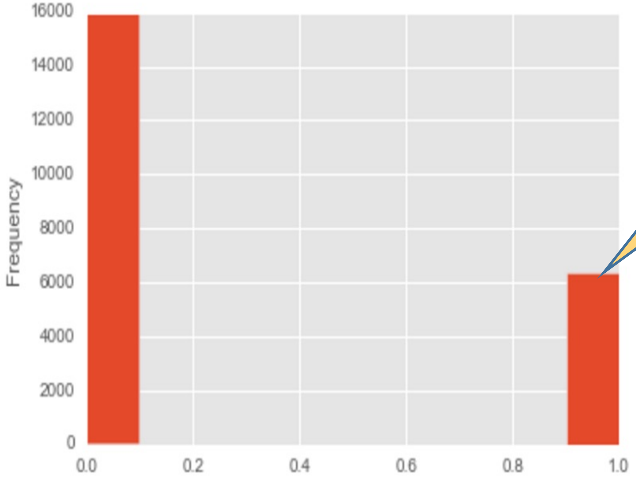
- Creating new_feature- vehicle_category

- 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



- 22K unique value
- 6K returning users

Distribution of User_id

Transformed to

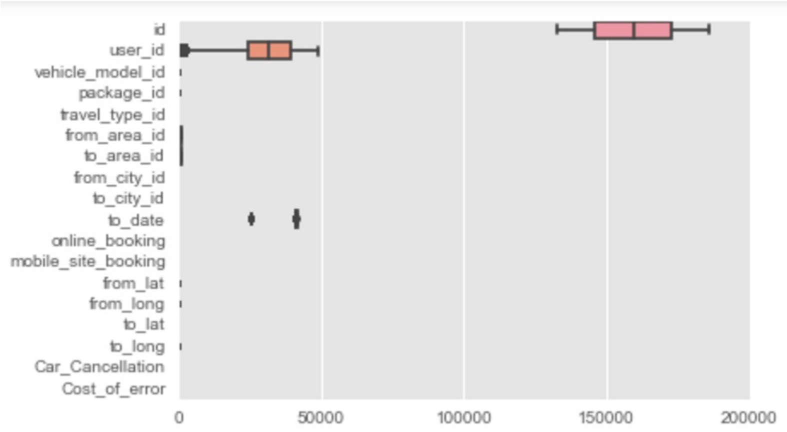
New Feature – is_frequent

✓ Is_frequent = 1 (returning user)

✓ Is_frequent = 0 (one time user)

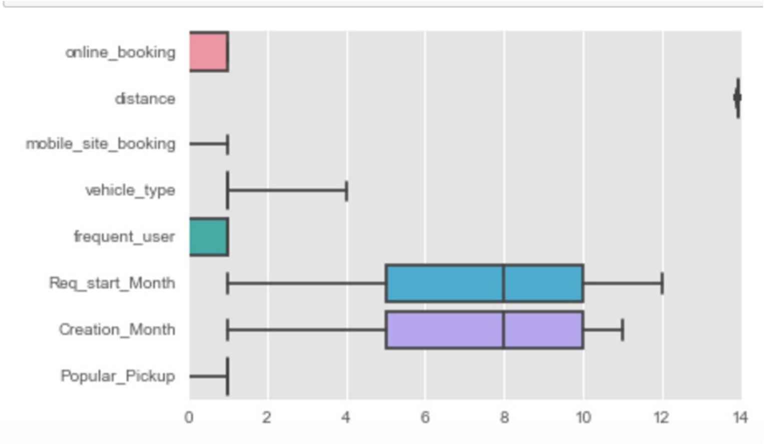
Feature Engineering

(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

Data Science Workflow – Parse, Mine, Refine the Data

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

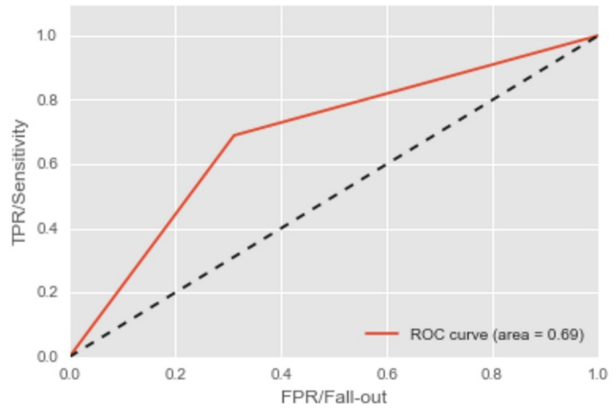
	coef	std err	z	P>|z|	[95.0% Conf. Int.]
const	-908.5756	4799.228	-0.189	0.850	-1.03e+04 8497.738
online_booking	1.2302	0.047	26.333	0.000	1.139 1.322
distance	63.2429	2.440	25.923	0.000	58.461 68.024
mobile_site_booking	1.3237	0.080	16.562	0.000	1.167 1.480
vehicle_type	-0.8444	0.056	-15.117	0.000	-0.954 -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

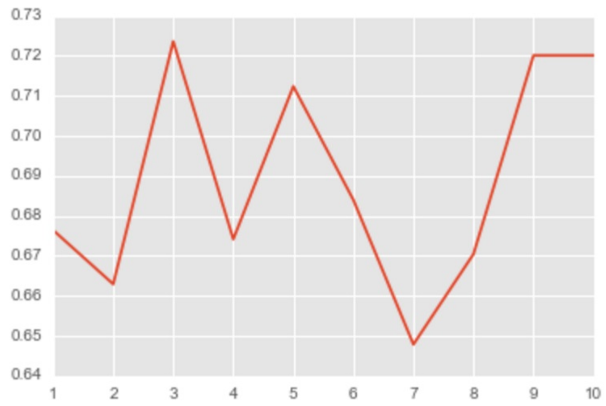
(Logistic Regression)

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)
```

0.69999999999999996

Modelling

(Decision Trees)

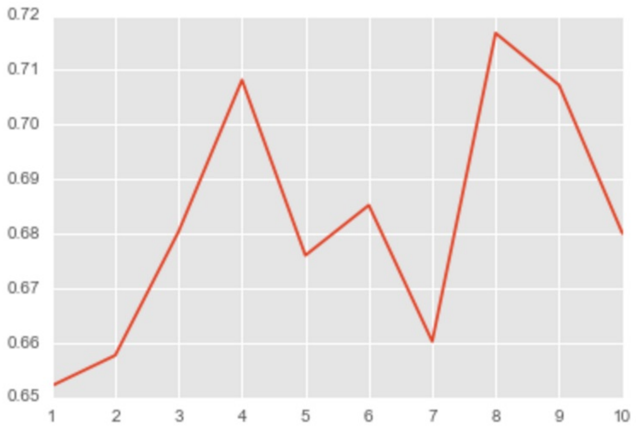
Training

```
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

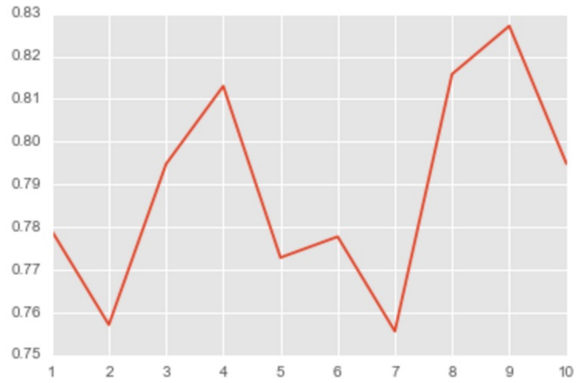
(Random Forests - no of trees=10000)

Training

```
model_forest.score(train_X_strat, train_y_strat)
0.9862612612612612
```

- 98 % Accuracy on the Training Data

Cross Validation



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

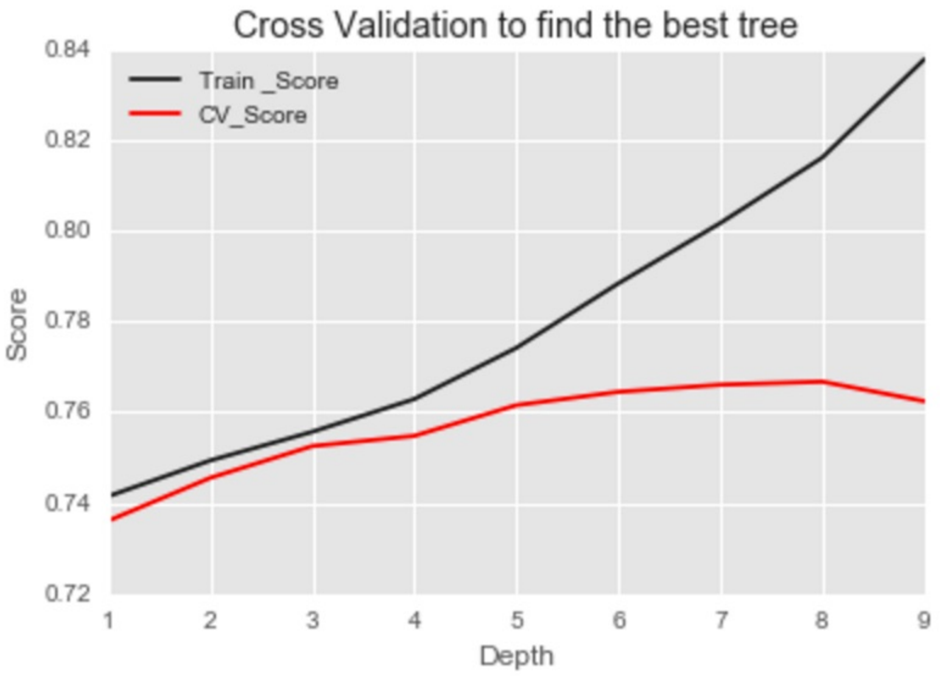
Conclusion

- High Accuracy on Training Data
- Huge gap exists between Training Score and Validation Score

Tuning the Model

(Random Forests - no of trees=10000)

Cross validation on max-depth



	CV_score	Depth	Train_Score
0	0.736044	1.0	0.741441
1	0.745497	2.0	0.749324
2	0.752485	3.0	0.755631
3	0.754732	4.0	0.762838
4	0.761492	5.0	0.774099
5	0.764422	6.0	0.788288
6	0.765992	7.0	0.801577
7	0.766674	8.0	0.815991
8	0.762393	9.0	0.837613

- Cross Validation score on forests with trees of depth 6 seems to provide the best score with minimum complexity

Selecting the best Model

(Random Forests - no of trees=10000)

Model Comparison

Un-tuned Model(Default Parameters)

Training Score	CV Score	Test Score
98.7 %	79 %	70 %

Tuned Model(on Max_depth = 6)

Training Score	CV Score	Test Score
80.1 %	76.6 %	74.3%

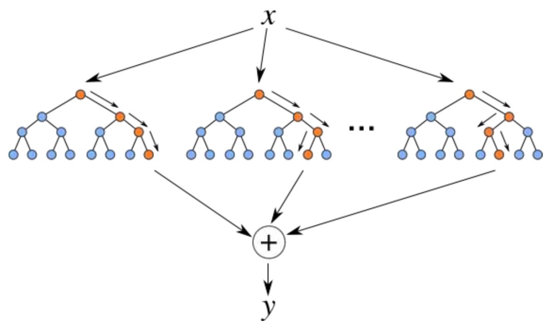
Modelling

(Random Forests-Feature Importance)

Feature	%age
distance	30.32
Creation_Month	19.60
online_booking	16.6
Req_start_Month	16.18
frequent_user	7.6
vehicle_type	4.5
mobile_site_booking	3.8
Popular_Pickup	1.1
Total	99.7

Modelling

Conclusion

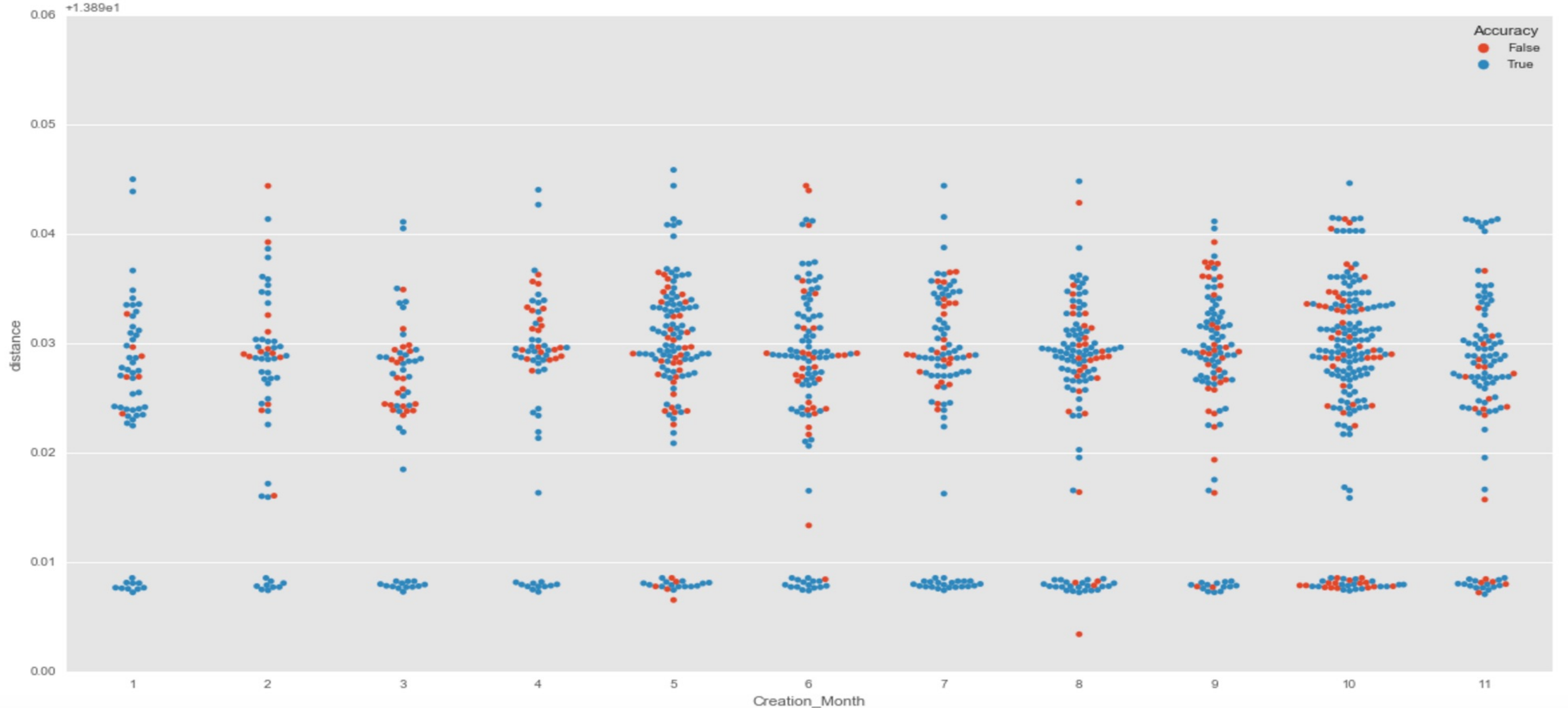


- Random forest seems to be the best amongst all the models
- Random forest also seem to cut off the nose and make the best decision on the important features
- Chance of over -fitting is less as compared to Decision trees(which is most likely to have overfit – Training score of 97%)

Agenda

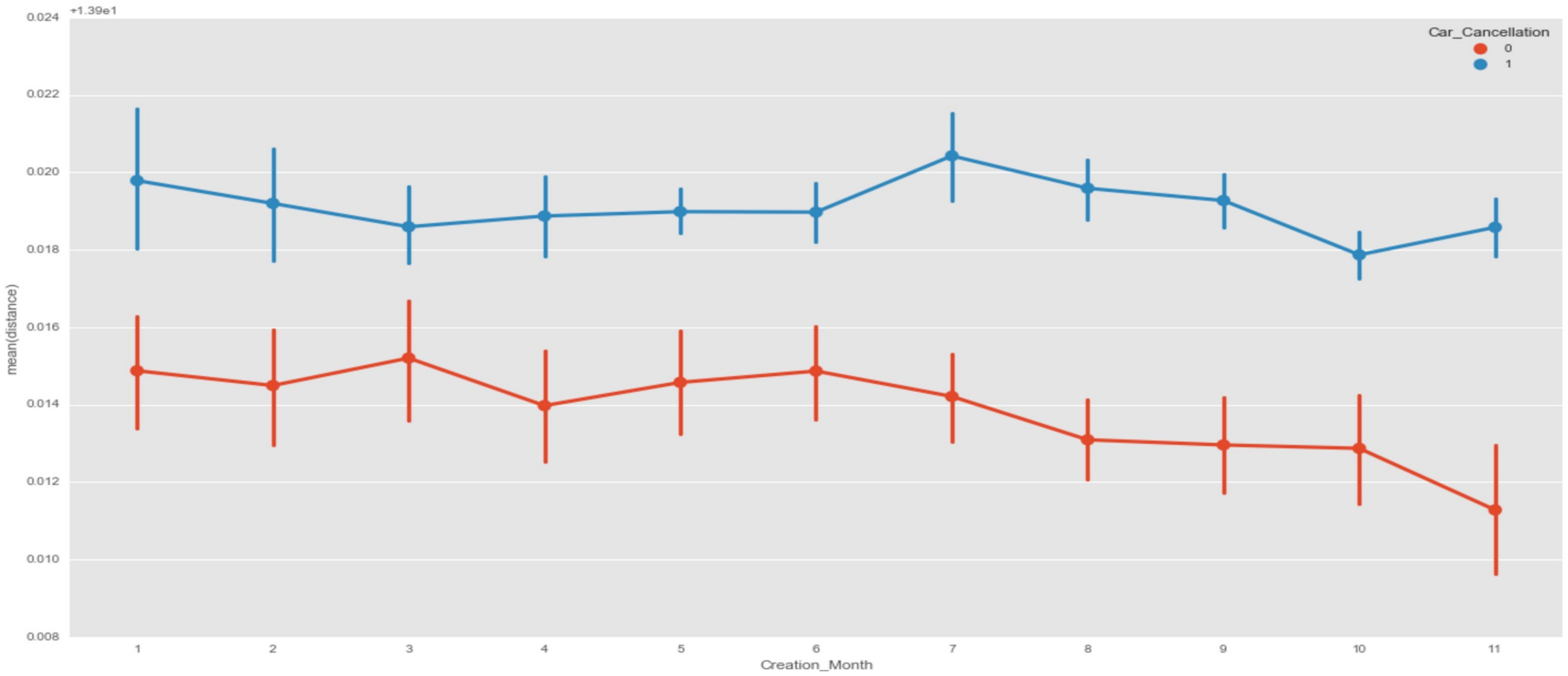
- ✓ Problem Statement
- ✓ Data Source and Features
- ✓ Feature Engineering and Exploratory Data Analysis
- ✓ Machine learning
- ✓ Inference

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
- Cancellations increases a lot between Jun-Aug and then follow the same pattern as rest of the year

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

(https://github.com/deveshkhandelwal/Modeling)

Questions/Feedback