Swiggy Techstein - Predicting and identifying segments of delays in delivery time.

-by Vijay Rajan and Battula Manohar **(Team Top Down)**

Overview

We at team Top-Down have taken up the challenge of setting customer expectations with the delivery time for orders by that could highly exceed 50 minutes.
Our approach is two fold

- Make a predictive model that will help identify at runtime orders that could exceed 50 minutes in delivery time
- Provide insights & a tool for operations and analytics team to identify common problem area that can help Swiggy take preventive or corrective action to solve this problem in the long term

We end this discussion by providing prescriptive actions that could help Swiggy set customer expectations, help the LMD-boy communicate with customer on pickup location and alse better utilization of fleet.

Approach Data Science

Prediction

Given the raw data and using past knowledge of variables which tend to affect delivery times, we identified the following variables

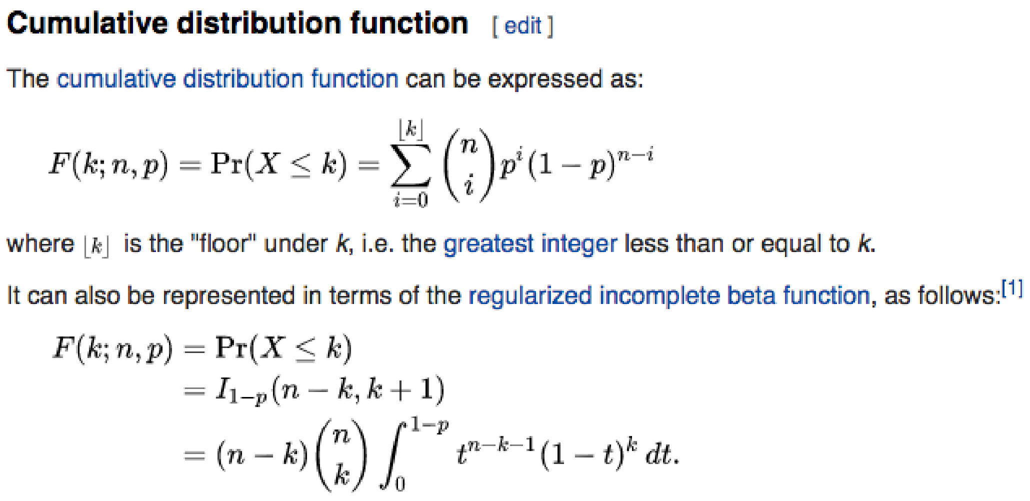
1. Restaurant_popularity
2. City
3. Zone_popularity
4. DOW
5. Day_type
6. Part_of_day
7. Month
8. Order_size
9. Discount
10. Delayed

11. Total
12. Classification

Raw orders were taken and classified as delayed_beyond_50_minutes or not. Once this was done, we aggregated across all the features namely

1. Restaurant_popularity
2. City
3. Zone_popularity
4. DOW
5. Day_type
6. Part_of_day
7. Month
8. Order_size
9. Discount

and counted how many of such cases had a bad count of delayed_more_than_50_minutes using binomial distribution with expected probability to be (0.15)



Cumulative distribution function [edit]

The cumulative distribution function can be expressed as:

$$F(k;n,p) = \Pr(X \le k) = \sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} p^i (1-p)^{n-i}$$

where $\lfloor k \rfloor$ is the "floor" under k, i.e. the greatest integer less than or equal to k.

It can also be represented in terms of the regularized incomplete beta function, as follows:[1]

$$\begin{aligned} F(k;n,p) &= \Pr(X \le k) \\ &= I_{1-p}(n-k, k+1) \\ &= (n-k)\binom{n}{k}\int_0^{1-p} t^{n-k-1}(1-t)^k \, dt. \end{aligned}$$

After doing this we checked how many percent of the cases were indeed bad in terms of percentage for what we called out as TRUE(as in delayed data points) and those that were not. While this was not very flattering we observed the following

Cohorts / Segments marked as ...	Delay more than 50 mutes	Total Bookings	Percentage of TRUE
TRUE	44972	144326	32%
FALSE	76523	676696	11%

We clearly needed more variables like

- Availability of supply at order time
- Weather conditions
- Traffic conditions
- Distance from restaurant to customer
- Time for preparation of the dish / item

These variables would have helped us build a much better and stronger/accurate classifier. That said, based on what we called out as “True” and “False” we did see a good classifier despite the imbalanced class of 17%.
Our code, Confusion Matrix and ROC Curve results are as shown below. The data for the classifier is available here .

Code

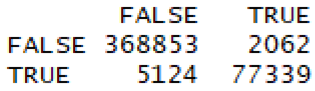
```
library(randomForest)
library(ROCR)
d <- read.csv("SwiggyClassifierFinal.csv")
d[,"Restaurant_popularity"] = as.factor(d[,"Restaurant_popularity"])
d[,"City"] = as.factor(d[,"City"])
d[,"Zone_popularity"] = as.factor(d[,"Zone_popularity"])
d[,"DOW"] = as.factor(d[,"DOW"])
d[,"Day_type"] = as.factor(d[,"Day_type"])
d[,"Part_of_day"] = as.factor(d[,"Part_of_day"])
d[,"Month"] = as.factor(d[,"Month"])
d[,"Order_size"] = as.factor(d[,"Order_size"])
d[,"Discount"] = as.factor(d[,"Discount"])
d[,"Classification"] = as.factor(d[,"Classification"])
smp_size <- floor(0.45 * nrow(d))
set.seed(123)
train_ind <- sample(seq_len(nrow(d)), size = smp_size)

train <- d[train_ind, ]
test <- d[-train_ind, ]
set.seed(123)
model <- randomForest( Classification ~ Restaurant_popularity + City + Zone_popularity + DOW + Day_type +
                    Part_of_day + Month + Order_size + Discount , data = train, ntree=100, mtry = 4,

                   keep.forest=TRUE, importance = TRUE, cutoff=c(.5,.5))
predictions <- predict(model, test, type = "prob")
```

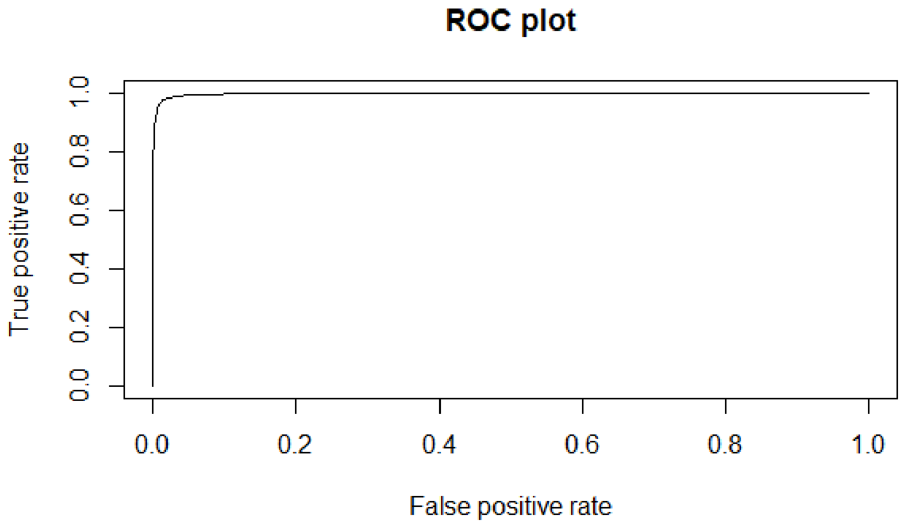
- table(predictions[,2] >0.37,test$Classification)
- predictions = as.vector(predictions$votes[,2])
- pred=prediction(predictions[,2],test$Classification)
- #table(pred=ifelse(predictions >= 0.37, TRUE, FALSE),test$Classification)
-
- perf_AUC=performance(pred,"auc")
- AUC=perf_AUC@y.values[[1]]
- AUC
- perf_ROC=performance(pred,"tpr","fpr")
- plot(perf_ROC, main="ROC plot")

Confusion Matrix



```
       FALSE   TRUE
FALSE 368853   2062
TRUE    5124  77339
```

ROC Curve



Exploratory Data Analysis

While predictive analysis and ML models make a great real time intervention for out-of-training data and make a good solution for setting customer expectation, EDA(exploratory data analysis) forms a more important long term strategy. The link here takes you to the EDA. Please find the glimpse of the tabled data below.
The technique used here is derived from “Frequent Itemset Mining” and is specifically derived for FACT Tables in Data Warehouses. The code for this was developed at Fratics Solutions founded by one of the authors(participants)

Level	Rule	Orders that took greater than 50 minutes to deliver	Orders that took less than 50 minutes to deliver	Total orders	Percentage
4	day_type=WEEK_DAY and discount_flag=1 and distinct_item_count=5_or_more_distinct_items_ordered and metropolitan_area=4nx	1778	3251	5029	35.35
3	**day_type=WEEK_DAY and discount_flag=1 and distinct_item_count=5_or_more_distinct_items_ordered**	1852	3388	5240	35.34
3	**discount_flag=1 and distinct_item_count=5_or_more_distinct_items_ordered and metropolitan_area=4nx**	1847	3419	5266	35.07
2	discount_flag=1 and distinct_item_count=5_or_more_distinct_items_ordered	1922	3563	5485	35.04
4	day_type=WEEK_DAY and distinct_item_count=5_or_more_distinct_items_ordered and metropolitan_area=4nx and zone_popularity=ZONE_POPULARITY_INDEX_3	2592	5271	7863	32.96
3	day_type=WEEK_DAY and distinct_item_count=5_or_more_distinct_items_ordered and zone_popularity=ZONE_POPULARITY_INDEX_3	2891	5977	8868	32.6
3	**distinct_item_count=5_or_more_distinct_items_ordered and metropolitan_area=4nx and zone_popularity=ZONE_POPULARITY_INDEX_3**	2699	5601	8300	32.52
2	distinct_item_count=5_or_more_distinct_items_or	3009	6344	9353	32.17

	dered and zone_popularity=ZONE_POPULARITY_INDEX_3				

Prescriptive Remedy

While delays are inevitable, predicting the delays help customer expectations.

- Can Swiggy incentivise the customer come to apartment or block entrance to pick up the order to cut time for delivery agent from “searching” and further delaying the order?
- Can Swiggy keep a delivery agent in large apartment complexes?
- Can Swiggy do forecasting of expected items at restaurants that are very popular and have them pre-package top items?
- Can Swiggy device new strategies for consistently ailing “segment slices” found through exploratory data analysis?
- Larger orders with 4 or more items should be packaged in a single carry on bag so that customers who pick up the order from the Delivery Agent from Apartment or Block entrances can pick up the order easily

Conclusion

Continuous monitoring of operational mishaps and mending ties with irate customers as well as making systemic changes along with setting customer expectations right go a long way in maintaining perceived Brand Value.
While ML and DL make great advances, there is a void that needs to be filled in the EDA space so that changing patterns of worries can be monitored and rectified.