**Customer Churn Model**

Built a customer churn model for predicting the returning customers in the next 6 months.

Performed below steps.

1. Data Quality checks
2. Data Transformation
3. Exploratory Data Analysis
4. Build Machine Learning model

I have given detailed comments and description in both the python notebooks.

I will try to provide the required answers.

1. all the code and files you used (both for exploration and the final ones) as a GIT repository (see below for further details)

Yes, I have uploaded all the necessary files in GIT repository.

1. a summary of the data transformations that you tried, and the ones you kept (we should be able to see this in the commit history)

Below are the Data transformation logic

1. Excluded the customers having only failed orders. (518 customers are present)
2. Created new features:

1. How old the customer is associated with business according to order\_date

2. From how many days Customer has not ordered

3. Customer ordered for a period of n days (continuously)

4. Most Common order hour

5. Number of successful orders

6. Number of failed orders

7. Average voucher amount used

8. Average delivery fee paid

9. Average amount paid

10. Customer has ordered from how many unique number of restaurants

11. Customer has ordered from how many unique number of cities

12. Customer has used how many unique number of payment\_id

13. Customer has ordered from how many unique number of platforms

14. Customer has used how many unique number of transmission\_id\*\*

15. Has customer used any voucher

16. Has customer paid any delivery fee

17. Has customer paid any amount ever

18. What is the order % distribution according to day-time

19. Distribution of percentage of order according to unique payment\_id

20. Distribution of percentage of order according to unique platform\_id

21. Distribution of percentage of order according to unique transmission\_id

1. a description of the ML models you tried, and the one you kept (we should be able to see this in the commit history)

I tried 5 different models with the default parameters (couldn’t do hyper parameter tuning due to time-constraint)

Tried a) Logistic Regression, b) K Nearest Neighbor, c) Decision Tree, d) Random Forest, e) XGBoost

Final model: XGBoost with 0.55 F1-score and 0.82 ROC

1. a measurement of how good your models are (at least for the best one)

I have given a model’s accuracy comparison based on F1-score, ROC, Precision and Recall metrics in the notebook.

XGBoost is having the best accuracy among 5 models.

1. any other findings on the data that you would like to share

Below are couple of things I observed.

* + - 1. Some customers are having only failed orders and returning. Some are returning even after 1 year.
      2. “last\_order\_before” feature’s histogram distribution is significantly different for returning and non-returning customers. This is also the feature which is used in tree splitting the most number of times.
      3. Most important features out of XGBoost model
* last\_order\_before how many days
* Number of successful orders
* Average amount\_paid
* first\_order\_before how many days
* Number of days between first and last order
* Average voucher amount used
* Most common hour of order

1. Ideas about any further models or data transformations that you would try if you had more time
2. **Data Transformation**

* Want to create 2 customer segments for New and Existing customers.  
  e.g. Customers who are less than 3 months old can be New and older than that can be Existing customers
* Segment-wise data can also be prepared based on amount\_paid

1. **Build 2 Different segment models for New and Existing customers**.  
   Need to check the accuracy of these 2 meta models. If better need to replace the model which is built on overall dataset.
2. **Try other Other ML techniques**

* Multi-variate Analysis
* Proper Outlier Analysis
* Imbalanced Data Handling.  
  Undersampling and Oversampling (as Returning customers are 20% of overall population)
* Hyperparameter tuning for XGBoost algorithm.
* Try LightGBM and tune its parameters.

Order of Script Execution:

There are 2 flows. 1 is Training and 2nd one is Prediction.

* + - 1. Training: Prepare Data and train model

Run

a) [script\_data\_transform.py](https://github.com/ashutosh3060/friday-burger-mojito/blob/master/script_data_transform.py)

b) [script\_model\_training.py](https://github.com/ashutosh3060/friday-burger-mojito/blob/master/script_model_training.py)

* + - 1. Prediction:

Run

a) [script\_data\_transform.py](https://github.com/ashutosh3060/friday-burger-mojito/blob/master/script_data_transform.py)

b) [script\_prediction.py](https://github.com/ashutosh3060/friday-burger-mojito/blob/master/script_model_training.py)

Dockerfile and requirement.txt can be used for running the script.

(In all the scripts below things need to be modified according to user specific environment.

Input file path, name

API, Application to be used)

Submitted By:

Ashutosh Behera