**Name:** ABHISHEK KUMAR SHARMA

**Email address:** KEHSIHBA.SHARMA12@GMAIL.COM

**Contact number:** 8290959527-7703947530

**Anydesk address:** **909 008 714**

**Years of Work Experience:** **01 MONTH**

**Date:** **29th Apr 2021**

## Self Case Study -1: Restaurant Recommendation Challenge.

---

"After you have completed the document, please submit it in the classroom in the pdf format."

Please check this video before you get started:
https://www.youtube.com/watch?time_continue=1&v=LBGU1_JO3kg

---

### Overview:

This is a **recommendation based problem**, data contains 10,000 customers in the test set. These are the customers you will need to recommend a vendor to. Each customer can order from multiple locations (LOC_NUM). Data is divided into 3 Train and test sets (Customers, Locations and Vendors).

There are approx.35,000 customers in the train set. Some of these customers have made orders to at least one of 100 vendors. Using all the above sets, **we have to apply the theory of recommendation system to recommend them to a vendor.**

The objective of this competition is to **build a recommendation engine to predict what restaurants customers are most likely to order from** given the customer location, restaurant information, and the customer order history.

**PERFORMANCE METRICS: (AS DESCRIBED IN THIS EVALUATION SCHEME)**

The error metric for this competition is the F1 score, which ranges from 0 (total failure) to 1 (perfect score). Hence, the closer your score is to 1, the better your model.

**Precision :** This is an indicator of the number of items correctly identified as positive out of total items identified as positive. The formula is given as: TP/(TP+FP)

**Recall / Sensitivity / True Positive Rate (TPR):** This is an indicator of the number of items correctly identified as positive out of total actual positives. The formula is given as: TP/(TP+FN)

**F1 Score:** A performance score that combines both precision and recall. It is a harmonic mean of these two variables. The formula is given as: 2*Precision*Recall/(Precision + Recall)

**Where:**

1. **TP**=True Positive
2. **FP**=False Positive
3. **TN**=True Negative
4. **FN**=False Negative

---

**The submission should like in this format:**

Where 1 indicates that a customer will order from that restaurant and 0 that they will not order from that restaurant.

**CID**=CUSTOMER UNIQUE ID
**LOC_NUM**=LOCATION NUMBER
**VENDOR**=VENDOR RECOMMENDED

| **CID** X **LOC_NUM** X **VENDOR** | Target |
|---|---|
| A7B8IGM X 0 X 105 | ---- 0 |
| NS70FA9 X 0 X 105 | ---- 1 |
| WTWOE69 X 0 X 105 | ---- 0 |

---

The data overview is described as below:

1. **test_customers.csv** - customer id's in the test set.
2. **test_locations.csv** - latitude and longitude for the different locations of each customer.
3. **train_locations.csv** - customer id's in the test set.
4. **train_customers.csv** - latitude and longitude for the different locations of each customer.
5. **orders.csv** - orders that the customers **train_customers.csv** from made.
6. **vendors.csv -** vendors that customers can order from.
7. **VariableDefinitions.txt** - Variable definitions for the datasets
8. **SampleSubmission.csv** - is an example of what your submission file should look like. Shown above.

## Research-Papers/Solutions/Architectures/Kernels

*** Mention the urls of existing research-papers/solutions/kernels on your problem statement and in your own words write a detailed summary for each one of them. If needed you can include images or explain with your own diagrams. it is mandatory to write a brief description about that paper. Without understanding of the resource please don't mention it***

---

1. https://www.kaggle.com/stuartday274/reorder-from-most-frequently-previous-vendor

   **Summary:** This solution predicts the most favourite restaurant/vendor recommendation for the users who have bought items from the vendors.This aheads with using various tables and after preprocessing returns a final table that can be used for final prediction. Using the probabilities of a customer ordering from the same vendor solution is described. Helps us to get an idea how data will be used and mapped in the final structure to get our results.

2. https://zindi.africa/competitions/akeed-restaurant-recommendation-challenge/discussions/2667

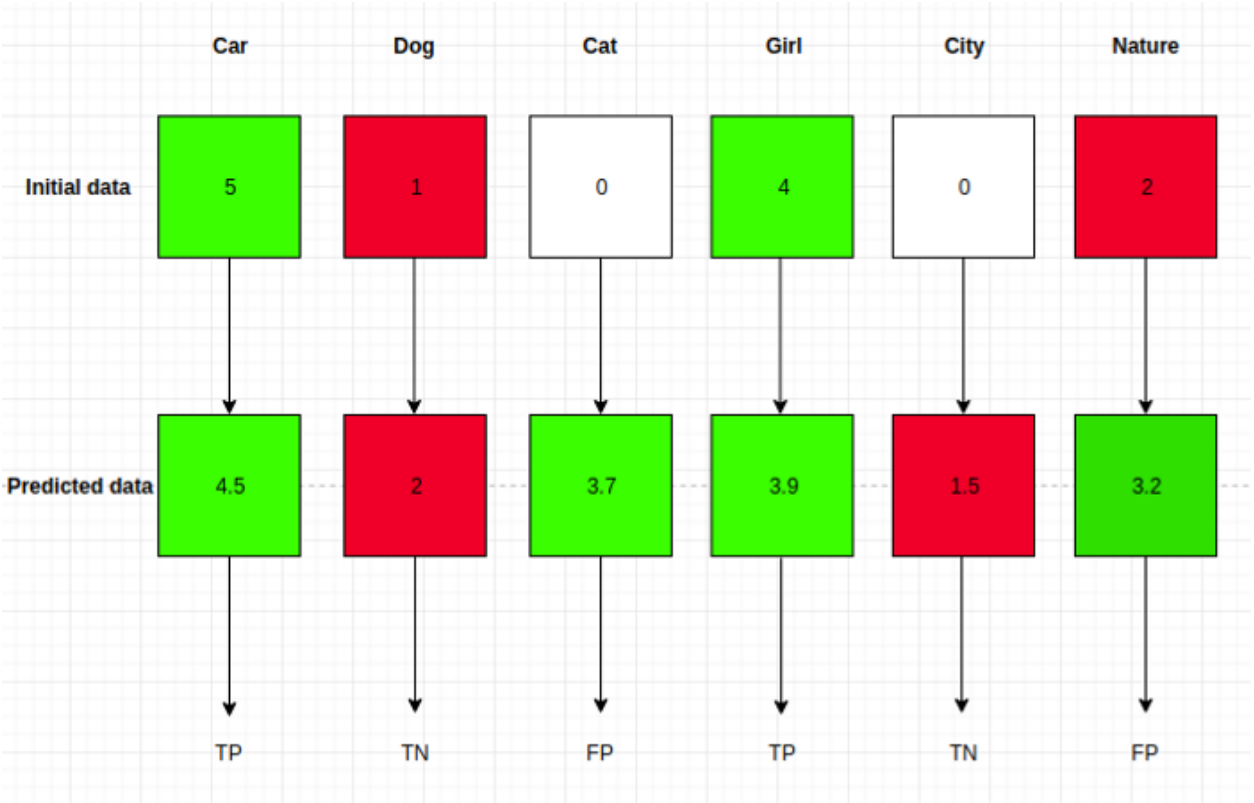
   **Summary:** The best solution submitted for this problem and methods tried in this approach are as follows.

   1. This solution used the clustering technique using vendors and their locations, Hence 6 clusters were made using the requirements and EDA.
   2. Main features lat/lon, min-max-avg-std for coordinates by customer, haversine distance and diff by coordinates to restaurant.
   3. The k-fold cross validation was used and on testing the customers were mapped to each clusters and then to their respective vendors.

4. The main advantage of this approach was clustering and separate models for main and bad coordinates clusters

3. https://bond-kirill-alexandrovich.medium.com/precision-and-recall-in-recommender-systems-and-some-metrics-stuff-ca2ad385c5f8

**Summary:** As for this problem, we needed an F1 score as our performance metric, so we need to understand if Precision,Recall can be used in recommendation systems. We got our answer from here.




As we can not directly use this metric but indirectly we can apply this.

The first line consists of an actual and predicted one. So this can be solved using the theoretical part. We can categorize that less than 2 rating=bad, 3= neutral and more than 3= great.

We have created 3 labels by categorizing our data and hence we can calculate our precision and accuracy.

TP = 2, TN = 2, FP = 2, FN = 0

And calculate three indicators: accuracy, precision and recall.

**Accuracy** =(2+2)/6 =0.67 => 67% accuracy

**Precision** = 2/(2+2) =0.5 =>50% precision

**Recall** = 2/(2+0) = 1 => 100% recall

So, using vendor ratings, we can use this if during feature extraction, we find the rating an important factor.

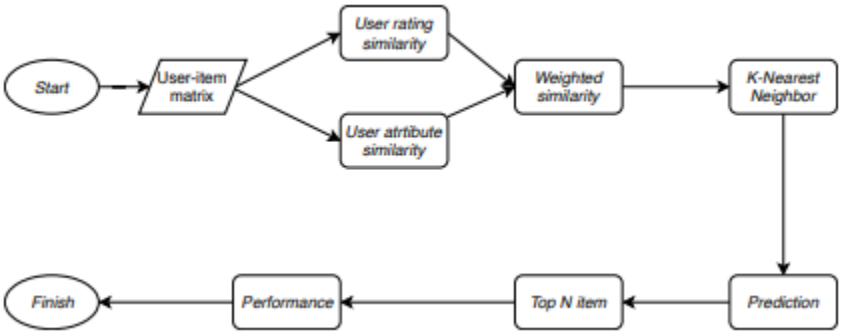
4. https://iopscience.iop.org/article/10.1088/1742-6596/1192/1/012023/meta

**Summary:** In this paper, they have built a recommender system that can recommend the restaurant in the Bandung area. They implemented a user-based collaborative filtering method for recommending a restaurant personally, based on ratings given by other users.

They also implemented two similarities, i.e., user rating similarity and user attribute similarity to find the proximity between users.

For the performance metric, they use Mean Absolute Error (MAE) to evaluate accuration of rating prediction.

The short diagram depicts the process, since we have too ratings in our tables for each restaurant, we can use this technique for good results.




5. https://www.upgrad.com/blog/data-preprocessing-in-machine-learning/

**Summary :** As we have lots of data processing for better results, This article depicts the best practices used for data preprocessing.

The seven main points under this consists of:

A. Acquiring the relative dataset
B. Getting all the important libraries for fast conversion
C. Importing the datasets
D. Missing values filling based on many techniques described in the course.
E. Encoding the categorical data
F. Splitting the dataset based on size and other factors.
G. Feature scaling

We can use this technique to make our final table perform better.

6. https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/

Summary: This article clears basic steps behind recommendation technique and states that **A recommendation engine filters the data using different algorithms and recommends the most relevant items to users. It first captures the**

**past behavior of a customer and based on that, recommends products which the users might be likely to buy.**

Through this I understood that we will use User-User collaborative filtering  and in this using correlation values. We can use the geographical location of users to filter out correlation between the users and will recommend them to the vendors.

---

## First Cut Approach

*** Explain in steps about how you want to approach this problem and the initial experiments that you want to do. ***(MINIMUM 200 words)*** ***

*** When you are doing the basic EDA and building the First Cut Approach you should not refer any blogs or papers ***

1. I will do basic EDA on train_customers,train_locations,train_full and vendors to find out which features should be taken in and out during preprocessing of data. And 7 steps methods have been seen above for best filtering without loss of potential data.
2. I will try to find out the best method to relate various above tables so that I can easily map 3 tables together with preprocessed values.
3. Using above two methods and  **TRAIN_CUTSOMERS** , **ORDERS** AND **TRAIN LOCATIONS**, I will try to map the customers with orders and locations so that I can create a common dataframe to find patterns and do some more EDA.

   Let's name this table as **MAP_TABLE**

4. Using the users, their locations and their ratings for each vendor in the MAP_table we can use our models based on the article mentioned below and described above.

https://iopscience.iop.org/article/10.1088/1742-6596/1192/1/012023/meta

5. For F1 score- as mentioned above we can calculate using category based scoring. And can get the final dataset.
6. After finalizing the train and validation set using the above data, I will do hyper parameter tuning to get the best results and use it in the approach.
7. Finally will test using the test data and check our accuracies.
8. Using flask and Web deployment methods to make our final project output and procedures.
9. So in this approach we are using ratings as our tool for recommendation of vendors for new users and using the geographical status we will find the best correlation in between the users for performing user-user collaborative filtering.
10. This is the first thing that came to my mind for the start.

---

**Notes when you build your final notebook**:

1. You should not train any model either it can be a ML model or DL model or Countvectorizer or even simple StandardScalar
2. You should not read train data files
3. The function1 takes only one argument "X" (a single data points i.e 1*d feature) and the inside the function you will preprocess data point similar to the process you did while you featurize your train data
    a. Ex: consider you are doing taxi demand prediction case study (problem definition: given a time and location predict the number of pickups that can happen)
    b. so in your final notebook, you need to pass only those two values
    c. def final(X):
        preprocess data i.e data cleaning, filling missing values etc
        compute features based on this X
        use pre trained model
        return predicted outputs
    final([time, location])

    d. in the instructions, we have mentioned two functions one with original values and one without it
    e. final([time, location])  # in this function you need to return the predictions, no need to compute the metric
    f. final(set of [time, location] values, corresponding Y values)  # when you pass the Y values, we can compute the error metric(Y, y_predict)
4. After you have preprocessed the data point you will featurize it, with the help of trained vectorizers or methods you have followed for your train data
5. Assume this function is  like you are productionizing the best model you have built, you need to measure the time for predicting and report the time. Make sure you keep the time as low as possible
6. Check this live session: https://www.appliedaicourse.com/lecture/11/applied-machine-learning-online-course/4148/hands-on-live-session-deploy-an-ml-model-using-apis-on-aws/5/module-5-feature-engineering-productionization-and-deployment-of-ml-models