# Recommendation Engine - Project Report

## Motivation/ Objectives:

The objective of this project is to build a recommendation engine, which offers relevant suggestions to the user based on his previous experiences or from his friend’s networks. The project tries to analyze users and their experiences and hence forth produces better and relevant suggestions for the future visits.

The project also demonstrates the accuracy of the model being used for producing recommendations, by considering the 80% of the dataset as a train dataset to build the data required for the proposed model and then recommend the results and then compare the results with the actual data. Thus getting an accurate measure of the accuracy of the model.

## Data Mining/ Analysis tasks tackled:

Data mining tasks that were implemented in the process of generating recommendations are,

### Classification Technique by classifying user {input}’s friends into 2 categories

* Foodie( these users are probably those one’s with enough data to analyze and produce recommendations)
* Occasional visitor(these users are probably those one’s with very less or no data)

The proposed model, considers the foodie’s experience to generate popular recommendations from time to time. Thus ensuring the user is provided with the best suggestions.

Classification Technique - Partitioning datasets into test, train data for measuring accuracy:

The project uses the property of dividing the whole data set into multiple datasets by splitting them into different proportions and then use the data to build the corpus and then test the corpus on the remaining data of the dataset. I have used similar idea to measure the accuracy of my model.

In measuring the accuracy, first I read the whole dataset to get the definitive values for the information about the user and then using the proposed model, I read only 80% of the available information and then produce the recommendations and then measure the accuracy of the recommendations generated for the 20% of the data.

### Data Wrangling:

I have implemented the concept of data wrangling on the data set to separate the useful information from the whole data set. The separation is key in the process of application’s performance. Data wrangling here separates the key information from each of the categories of JSON files.

I am implementing this at the moment of reading the files, thus ensuring to build a good data structure right from the beginning. Whenever a key information is read from the JSON file, the data is stored into Python Dictionary/Lists apparently to work around later for the other data mining tasks.

This task also enables the high performance of the whole application since, the data that is built at this moment is enough and the read operation over the files need not be done again and again. With single time read operation over huge data sets I have reduced the overall time required for building the data part of the application.

### Data Classification:

Data classification here focuses on classifying the users into different categories for the later use in the application. Classification here focuses on the users into categories like foodie, regular and rare visitor on the basis of the Check-in’s, reviews. The data classification uses some predefined parameters for these values to classify a user into a relevant category.

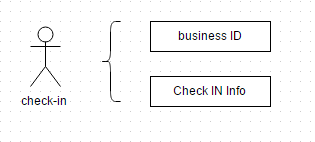
## Design of methods:

### Building the corpus:

#### Check-in JSON

The Check-in JSON has the information about the check-in information for each of the business and the number of check-ins for different hours of the day. The information is useful to calculate the popularity of the hotel.

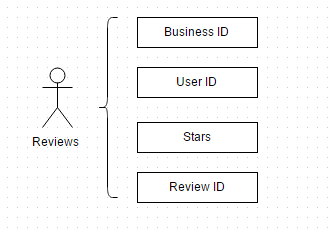
The data structure that I built for Check-in dataset for different businesses is as below,



**Note**: Figure representing the information of the parts of check-In json used for the data model.

#### Review JSON

The Review JSON has the information about the user opinions of the different users about different businesses. Capturing this information is key in processing good results for place recommendation. I have opted to capture information like business\_id, user\_id, starts, review\_id for a particular review and the resultant model for this json is as mentioned below:

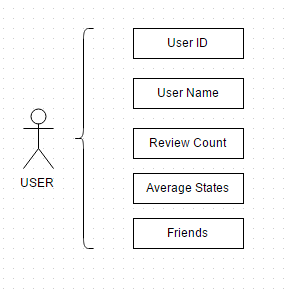


**Note**: Figure representing the information of the parts of Reviews json used for the data model.

#### User JSON

The yelp data set has a total of 366K users and capturing all of the users information in the model is not useful so I intend to capture only certain parts like, user id, user name, review count, average starts, friends.

Information about friends of a particular user is key to the finding of a particular restaurant as a recommendation. In some cases when you have many friends, it brings additional options to suggest recommendations from.

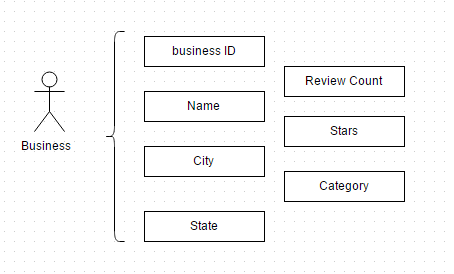


**Note**: Figure representing the information of the parts of USER json used for the data model.

#### Business JSON

Yelp data set has lots of useful information about each business that may be key aspect and may be useful in implementing different data mining tasks so as to provide good recommendations for the user for their future visits.

The data model of this project aims to capture key information like business id, business name, city of the business, business state, average stars for a particular business, and review count of each business and the wide range of categories that business comes under.

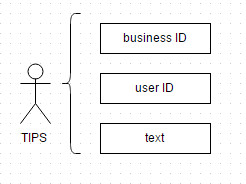


**Note**: Figure representing the information of the parts of Business json used for the data model.

#### Tips JSON

Tips for a particular business is another key aspect that is important in understanding about a business and also gives insights about experiences of different users and what they like, dislike in that restaurant.

The current data model for the project considers three parts of every tip from the tips.json. They are business ID, user ID, and tip text.



**Note**: Figure representing the information of the parts of Tips json used for the data model.

The Next section of the project code are different functions which communicate with each other to generate and return information

## Implementation of Methods:

The project to perform additional functionalities in the future and also to modularize the different concept implementations, I opted to implement different concepts in the corresponding functions. The function names are chosen to be familiar high level words to make a newbie understand the scene behind the code.

get\_recommendation\_for\_accuracy:

The function mentioned above executes other functions and generates the data required for the measure of accuracy

measure\_accuracy: Once the required information is generated, the accuracy is measured by comparing the main recommendations and the data extracted for the 80% of the data from the data set.

get\_top\_restaurants\_from\_rest\_i\_visited\_eff:

This function creates a dictionary of the restaurants and the number of times a user has visited a particular restaurant. This method has certain different blocks of code that perform the measurement of accuracy.

gettopfriends: retrieves the friends after classifying them into two categories for a particular user

get\_unique\_rest\_names\_from\_uservisits: The function creates a dictionary without any duplicates. The dictionary has a key value pairs that are related to the business and the number of visits a user has made.

get\_top\_restaurants\_from\_rest\_i\_visited: Gets the information of all the restaurants that a user has visited by reading the checkin and review jsons. And there after fetches the information of the different businesses

getuservisitedplaces: This function returns a list of places with duplicates but belong to the user he visited.

## Results and Evaluation:

### Calculating the accuracy of the recommendation engine

The function assumes a particular user to implement a test case and tries to measure the accuracy of the recommendation engine by comparing the results for the last 10% of the visits made the user and the recommendations made for that particular user.

Function measure\_accuracy() returns the percentage of accuracy that this model performs in generating the recommendations.

Initially, the same functions get\_recommendations() that are used to generate the recommendations and the data required to measure the accuracy. The function get\_recommendations() sets the dictionary uniq\_restaurants.

The observations mentioned in the website give more insight of the project

## Presentation/ Visualization of the Outcome:

Outcome of the results are displayed in the website and the observations are discussed below the output section in the website.

## URL of project website:

https://omega.uta.edu/~uxm0282/dmcproj/index.html

If the website is down because of the scheduled maintenance please check index.html file in the project files submitted.