Problem:

Today, restaurant customers have many options to choose from. However, it takes lot of time and effort to select the restaurant that matches to their preferences. A typical restaurant has both positive and negative reviews. As a result, the customer must check many reviews and read between the lines to finalize his/her choice. On the other hand, a customer usually visits the restaurant with at least one person – family, coworker or a friend. If the customer's own perception and choice about the restaurant was correct, then all the members in his/her group would be happy. And if the prediction was not correct, it would result in a less than satisfactory experience. Therefore, I am motivated to solve this problem of restaurant selection. A restaurant recommendation system is our solution. This system would create a list of top 10 restaurants that closely matches to the customer preferences and his/her reviews on the prior restaurants visited.

I am interested in developing such restaurant recommendation system. The outcome of the project is to recommend top 10 restaurants to the customer based on his/her preferences and past ratings.

Background:

To learn the background, I have read relevant papers:

1) 'Effects of relevant contextual features in the performance of a restaurant recommender system' the prototype of an existing recommender
2) Context-Dependent Items Generation in Collaborative Filtering

In addition to these, I reviewed the papers that Ire presented in 'Workshop on Context-Aware Recommender Systems (CARS-2009)'

These provided us with essential background for our project.

Data: The data source for the project is UCI Machine Learning Repository. The dataset was obtained from the recommender system prototype. The dataset contains 9 files. Overall there are 7200 rows with 100 attributes

Restaurants

1. chefmozaccepts.csv
2. chefmozcuisine.csv
3. chefmozhours4.csv
4. chefmozparking.csv
5. geoplaces2.csv

Users

1. usercuisine.csv
2. userpayment.csv
3. userprofile.csv

User-Item-Rating

1. rating_final.csv

File Name	Instances	Attributes	Attribute	Attribute Type
chefmozaccepts.csv	1314	2	Place ID	Nominal
			Rpayment	Nominal
chefmozcuisine.csv	916	2	Place ID	Nominal
			Rcuisine	Nominal
chefmozhours4.csv	2339	3	Place ID	Nominal
			Hours	Nominal
			Days	Nominal
chefmozparking.csv	702	2	Place ID	Nominal
			Parking Lot	Nominal
geoplaces2.csv	130	21	placeID	Nominal
			latitude	Numeric
			longitude: Numeric	Numeric
			the_geom_meter	Nominal (Geospatial)
			name	Nominal
			address	Nominal
			city	Nominal
			state	Nominal
			country	Nominal
			fax	Numeric
			zip	Nominal
			alcohol	Nominal
			smoking_area	Nominal
			dress_code	Nominal
			accessibility	Nominal
			price	Nominal
			url	Nominal
			Rambience	Nominal
			franchise	Nominal
			area	Nominal
			other_services	Nominal
rating_final.csv	1161	5	userId	Nominal
			placeId	Nominal

			rating	Numeric
			food_rating	Numeric
			service_rating	Numeric
usercuisine.csv	330	2	userid	Nominal
			Rcuisine	Nominal
userpayment.csv	177	2	userid	Nominal
			upayment	Nominal
userprofile	138	19	userID	Nominal
			latitude	Numeric
			longitude	Numeric
			the_geom_meter	Nominal (Geospatial)
			smoker	Nominal
			drink_level	Nominal
			dress_preference	Nominal
			ambience	Nominal
			transport	Nominal
			marital_status	Nominal
			hijos	Nominal
			birth_year	Nominal
			interest	Nominal
			personality	Nominal
			religion	Nominal
			activity	Nominal
			color	Nominal
			Iight	Numeric
			budget	Nominal
			height	Numeric

Model:

Here, on a high level, I would like to mainly perform two tasks. First, I will assign ratings to the restaurants that the customer has not rated based on the ratings the customer has given to other restaurants. Second, I will recommend top 10 restaurants to the customer based on the overall ratings and customer preferences.

Initially, I will start with data preprocessing.

Exploratory data analysis: In this stage I computed statistics of each report and identified the target variable, features.
Each of the csv files are loaded into different data frames and total number of unique records in each csv file is identified.

Target variable:'Rating'

***Data Cleaning*:**
a) NaN Handling: The are many columns with missing values or '?'. I am replacing some of the missing features by average of the column or median. For categorical values, I am making few assumptions. For example, if parking column says '?' I am assuming it as 'No Parking' as more than 80% of all the parking column values are 'No Parking'.
b) Univariate Analysis: Here I am plotting histograms to look for outliers and unusual spikes in the variance

***Data Integration**:*
There are five data frames that has restaurant information for each place id where place id is the unique value given to each restaurant.These data is combined into one dataframe based on the place id. The resulting data has 83 features Index(['placeID', 'Rpayment', 'Rcuisine_American', 'parking_lot', 'latitude' ……...'accessibility', 'price', 'url', 'Rambience', 'franchise', 'area', 'other_services']. By merging the data, I found that I have a total of 938 restaurants.

Similarly, I combined all the user information which is scattered between 3 data frames based on user id where 'user id' is an unique value given to every user. After merging all the three data frames I found that there are 130 unique users to work with.

There are 1161 ratings given by 130 users to 938 restaurants.

Data Reduction: a) For this project, restaurant rating is most important variable. If the restaurants do not have rating data, I am not considering those restaurants. b) Some user features like (height, Iight, color) are irrelevant to our analysis So I am performing data reduction to reduce these data points.

Data Transformation: The data must be numeric to perform analysis on it. So, I mapped the data to numeric values.
a) One-hot encoding: Restaurant cuisine values are categorical. So I are performing one-hot encoding to convert them to numbers   The parking data has non numerical values like none, yes , public etc., So I gave the parking value as 0 is the restaurant has no parking ,1 if there is a fee and 2 if it is yes or public.
Similarly smoking_area, dress-code, price, ambience, area, accessibility are mapped to numerical values which previously had non numerical values.
b) Aggregation: I am performing row-wise summations for features that are more useful when they are aggregated. c) I am normalizing the data to address any issue of model being highly sensitive to scaling

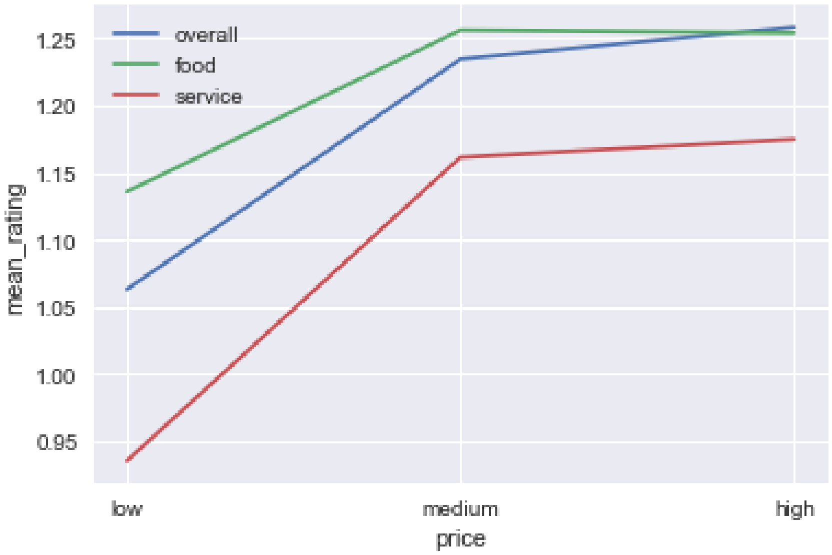
The rating data frame has values three different type of ratings for each restaurant user pair. I took average of food_rating, service_rating, overall_rating and added this to the restaurant dataframe based on the placeID. The user information is merged with restaurant information for feature engineering.

***Feature Engineering*:**

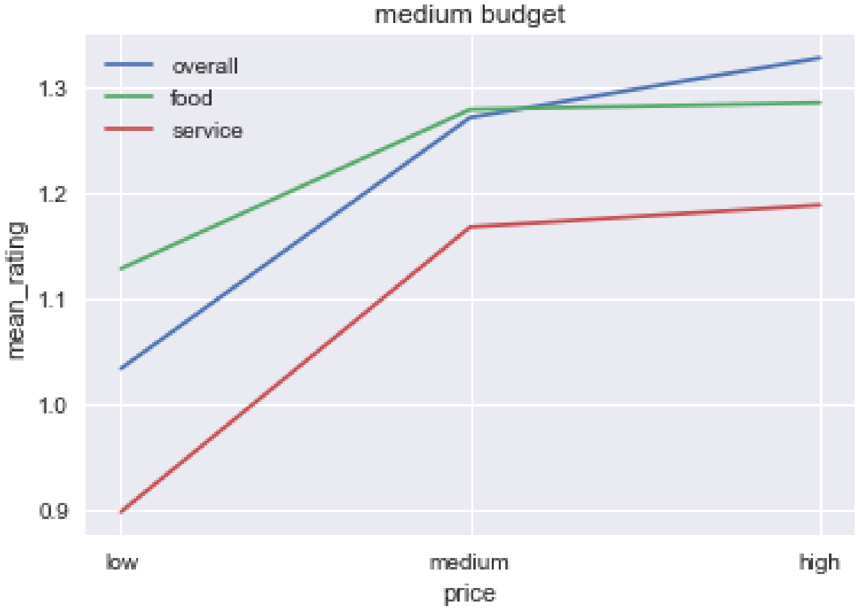
I applied correlation matrix to the combined data to understand the correlation between the features and the target variable.

I plotted graphs to understand 1) how ratings vary with restaurant features 2) 1) how ratings vary with restaurant features for various user profiles. Some of the plots are shown below:

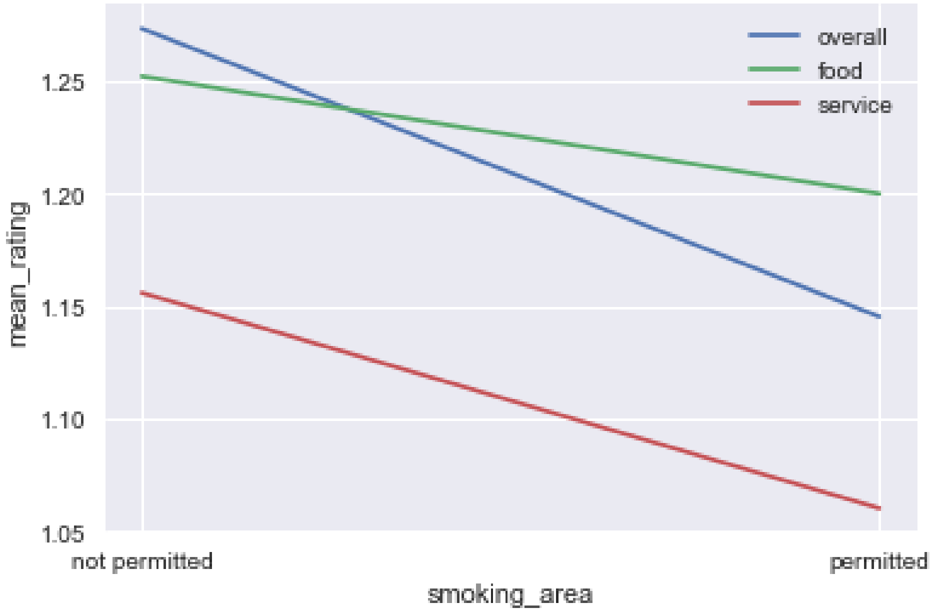
Average Restaurant ratings vs. price



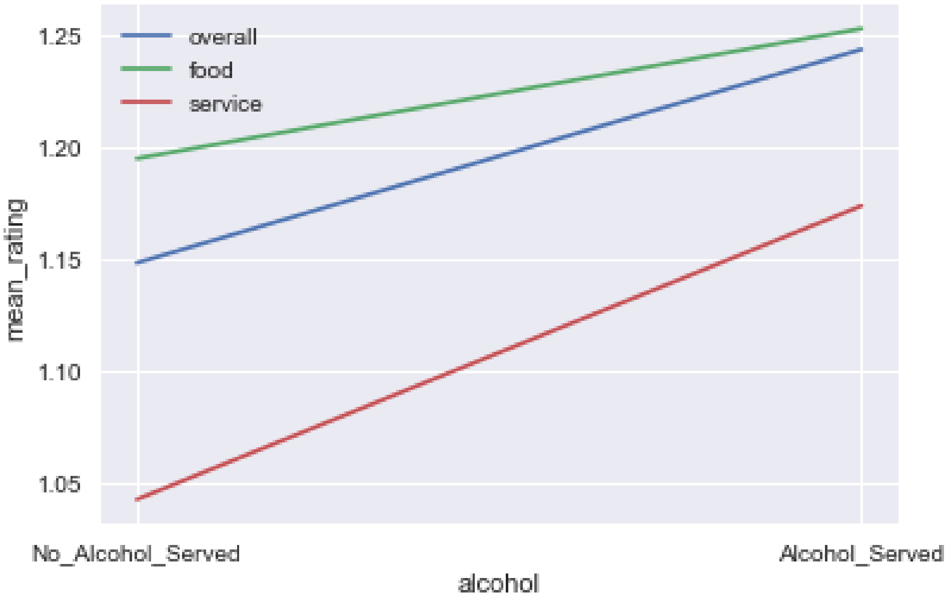
Average Restaurant ratings vs. price for medium budget category users



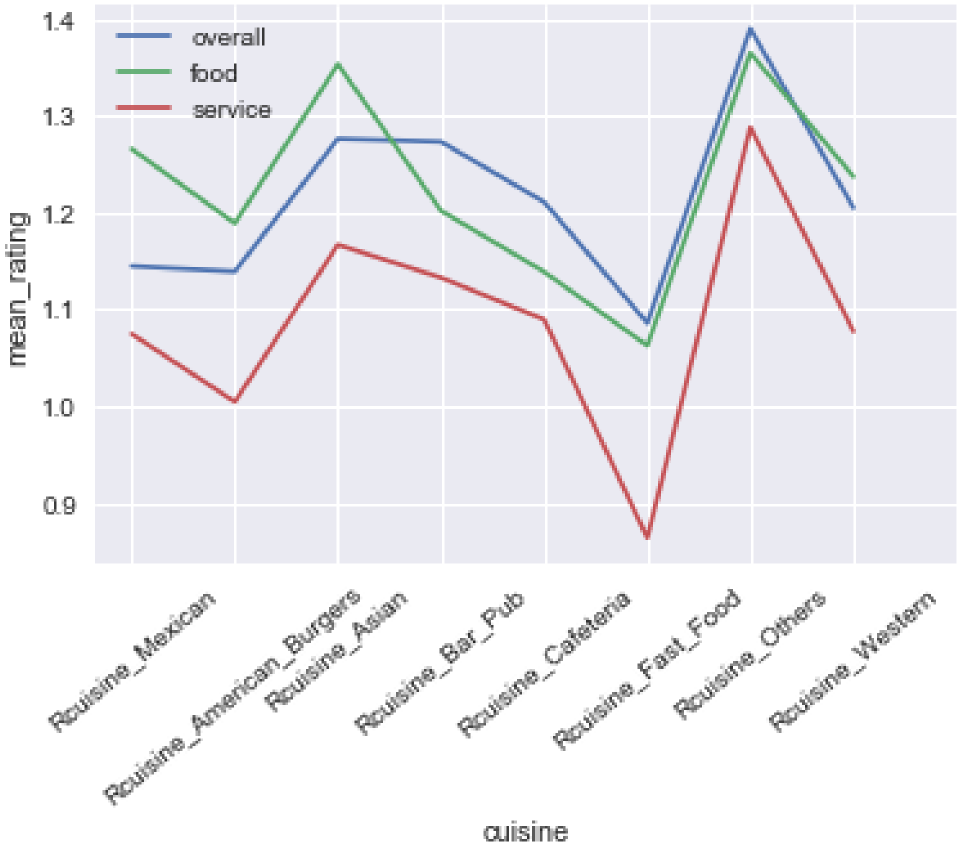
Average Restaurant ratings vs. smoking area



Average Restaurant ratings vs. alcohol



Average Restaurant ratings vs. cuisine categories



Some key observations:

Price	Medium priced restaurants are the most preferred restaurants among all the user groups
Smoking	Restaurants that permit smoking has lesser rating than the restaurants that do not permit smoking
Alcohol	Alcohol affects rating but it depends on the type of the drinkers.
Other services	Restaurants with other service perform better than those with no services.
Area	Users prefer closed space over open space
Cuisine	Other food category and Asian food category has highest ratings. Fast-food categories had the lowest ratings

Model Selection:

As the data is labeled data, I am using a supervised learning model. As the target variable is continuous, I am using regressors.

Linear Regression: I would like to do Linear Regression as a benchmark and quickly calculate coefficients and understand significance.

K-Nearest Neighbors: I would like to use kNN regressor as in kNN nearer neighbors contribute more to the average than the more distant ones. Given that I are trying to find the restaurants that closely match to the user preferred restaurants with similarity in features, I believe kNN is a good model. I will evaluate the model by calculating RMSE scores by applying k-fold cross-validation.

***Evaluation*:**

Since it is a regression problem, I decided to RMSE (Root Mean Squared Error) to evaluate the model. RMSE measures the differences between ratings predicted by a model and the ratings observed. I didn’t go with MAP (Mean Average Precision) as I proposed earlier because MAP just gives if the ranking of the predictions is accurate, but it doesn’t give any information about the error in predicted rating. So, I used RMSE to evaluate our model.

	LINEAR REGRESSION	KNN REGRESSION
RMSE	0.74	0.77

Assumptions:

I assume that the food rating, service rating and overall service rating given by any user is genuine. I predict the recommendations based on the user's ratings. I assume the user ratings are genuine because they are directly collected from them.

I assume that the user's future choice would be like their previous choices because it's mostly likely that the user would prefer similar restaurant again.

Predictions:

Predicted ratings to all the restaurants that the user hasn't rated.

What I changed:

I did not go with MAP (Mean Average Precision) evaluation metric as I proposed earlier. MAP focuses on whether the ranking of the predictions is accurate or not. It does not give any information about the error in predicted rating. So, I instead used RMSE to evaluate our models.