



Politecnico
di Bari

Big Data Course - Prof.ssa Simona Colucci

Second Project Report

28/01/2021

Group 6: Dario Di Palma, Marco Servadio

Overview

We will do various data analysis tasks on the “Restaurants and Consumers” dataset by UCI ML. Tasks range from simple statistical analysis to business-economical analysis. The final task implies the use of ML to create a recommendation engine with the Collaborative Filtering algorithm.

Targets

1. Preliminary analysis on restaurants and consumers.
2. Relation between certain features of a restaurant and the received rating.
3. Supply and demand analysis for each kind of cuisine.
4. Analysis on the ratings.
5. Recommendation system based on the most liked restaurant and type of cuisine.
6. Creation of a recommendation system based on Collaborative Filtering technique.

Project Notebook: [Big Data - Second Project - Group 6 - Deepnote](#)

Summary

Overview	1
Targets	1
Project Notebook: Big Data - Second Project - Group 6 - Deepnote	1
Summary	1
Dataset Description	3
1. Preliminary analysis on restaurants and consumers	4
1.1 General information about the dataset	4
1.2 General information about the users	4
1.3 General information about the restaurants	8
2. Relation between certain features of a restaurant and the received rating	10
2.1. Research of a possible relation between the best rated restaurants and the kind of cuisine they serve	10
2.2. Research of a possible relation between availability of parking and service rating	12
3. Supply and demand analysis for each kind of cuisine	13
4. Analysis on the ratings	14
4.1. Number of ratings and other statistical parameters	14
4.2. Rate of usage for each score	14
4.3. Number of reviews for each customer	15
4.4 Distribution of average ratings for each restaurant and number of ratings	15
5. Recommendation system based on the most liked restaurant and type of cuisine	16
5.1. Most liked restaurants	16
5.2. Search for the most popular restaurants by type of cuisine	16
6. Recommendation system using Collaborative Filtering with MLlib	17
6.1 Extraction of characteristics from the dataset and training of the model	19
6.2 Top 10 recommended	20
6.3 Evaluation of the model	20
6.4 Execution Time	22

Dataset Description

The Dataset we found is composed by 9 files divided in 3 groups:

1. Restaurants

- a. *chefmozaccepts.csv* → Instances: 1314 | Attributes: {placeID, Rpayment} | Type of Payments accepted by restaurants;
- b. *chefmozcuisine.csv* → Instances: 916 | Attributes: {placeID, Rcuisine} | Type of Cuisine of the restaurants;
- c. *chefmozhours4.csv* → Instances: 2339 | Attributes{placeID, hours,days} | Describe the working days/hours;
- d. *chefmozparking.csv* → Instances: 702 | Attributes{placeID, parking_lot} | Describe for each restaurant the type of parking if it has it;
- e. *geoplaces2.csv* → Instances: 130 | Attributes:{...“21 attributes”...} | Describe the location of the restaurant and all the others info about it.

2. Consumers

- a. *usercuisine.csv* → Instances:330 | Attributes:{userID, Rcuisine} | Describe the preferred cuisine of the user;
- b. *userpayment.csv* → Instances: 177 | Attributes:{userID, Upayment} | Describe the preferred payment methods used by user;
- c. *userprofile.csv* → Instances: 138 | Attributes: {...“19 attributes”...} | Describe all other elements that represent the user, like if he/she smokes or if he/she loves to drink or not.

3. Consumer-Restaurant-Rating

- a. *rating_final.csv* → Instances: 1161 | Attributes: {userID, placeID, rating, food_rating, service_rating} | Describes the rating released by the user and divides it into three ratings , the global one, the rating about food, and the rating about the service, from 0 and 2.

1. Preliminary analysis on restaurants and consumers

The first step of our research was to preliminary datas about the dataset recorded informations.

Next we proceeded to do some basic statistical analysis over the user and restaurant datas.

We are starting to understand which type of peoples and restaurants there are in our dataset. Such analysis includes for users the weight, height, budget, smoking habits, drinking habits, jobs, age, payment method, residence location and favorite cuisine. For the restaurants we analyzed accepted payments method, drinks served, price level, smoking rules, cuisine and geographical position.

For the above we used PySpark (via the PySparkSQL API and MapReduce API), Pandas, GeoPandas Matplotlib, Seaborn and Scipy.

1.1 General information about the dataset

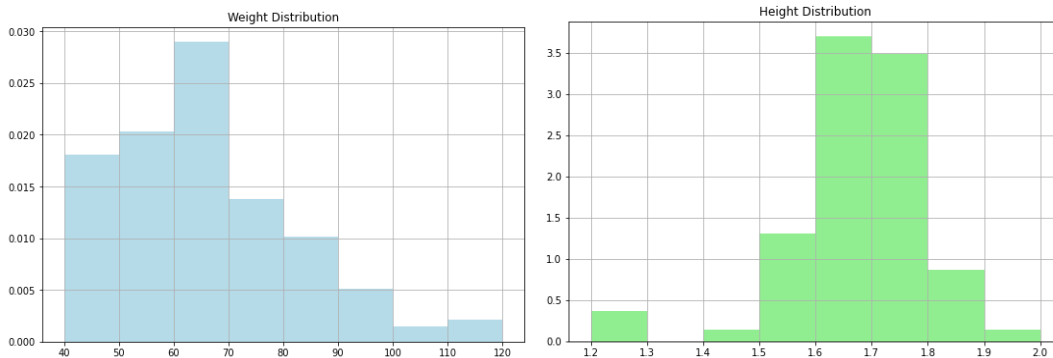
Initial approach to dataset

```
Total Restaurants: 615  
  
Payment Methods: [Bank_debit_cards..VISA...American_Express...]  
Types of cuisine: [International...Mexican...Turkish...]  
  
Total unique users that released a review: 138  
Total unique restaurants that were reviewed: 130
```

1.2 General information about the users

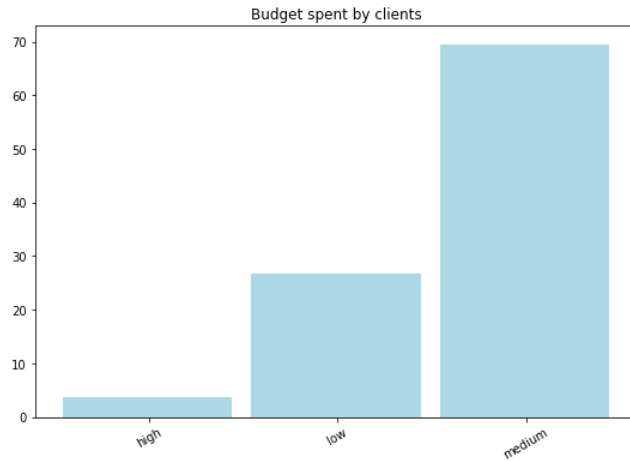
Our main purpose in this phase was to understand what kind of users we were dealing with, and to determine as much information as possible about them.

1. Physiological aspects



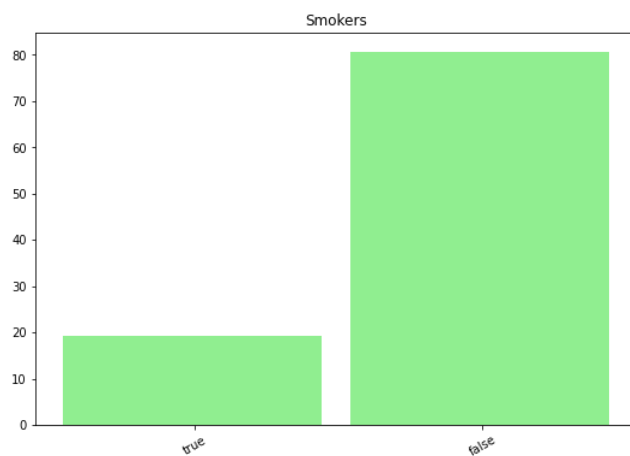
We have observed that most of our users have a weight between 60-70kg and a height between 1.6-1.8m

2. Budget spent by our users



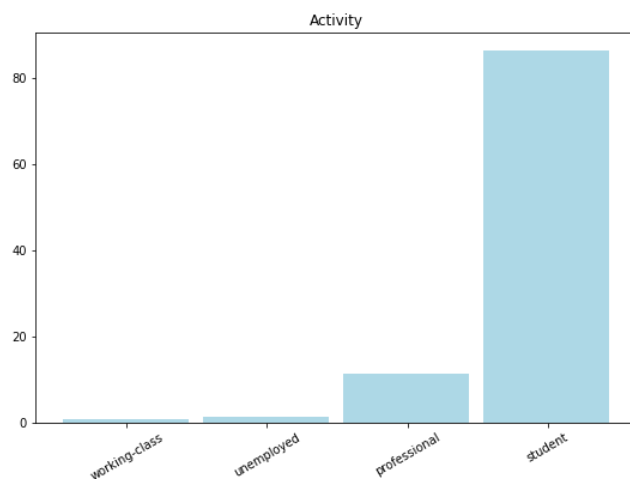
As it is possible to observe users spend a lot on an average budget, this leads us to assume that most restaurants have an average cost.

3. Percentage of smoking users



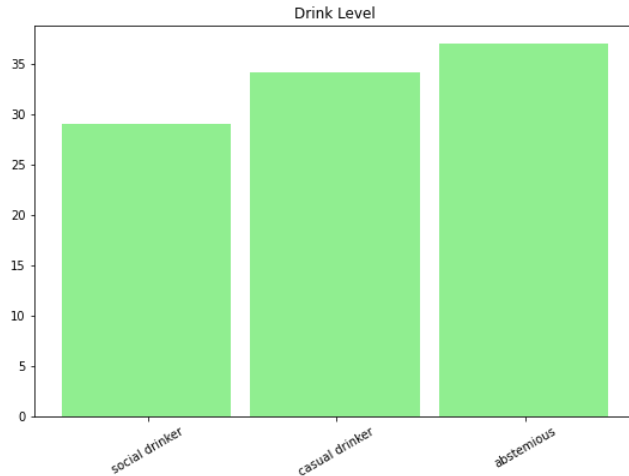
In this case, 80% of our users are non-smokers, this makes us assume that restaurants where smoking is prohibited may be preferred.

4. Distribution of user jobs



80% of our users are students this could greatly influence the rest of the characteristics, such as the budget spent in restaurants.

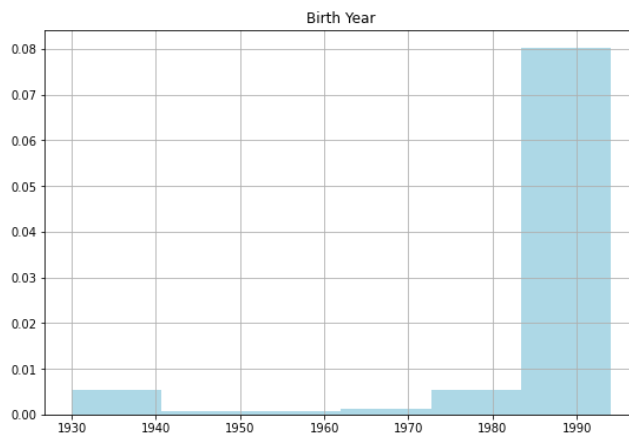
5. Willingness of users to drink alcohol



We have 35% of users abstemious and the rest 65% drinking alcohol in different ways.

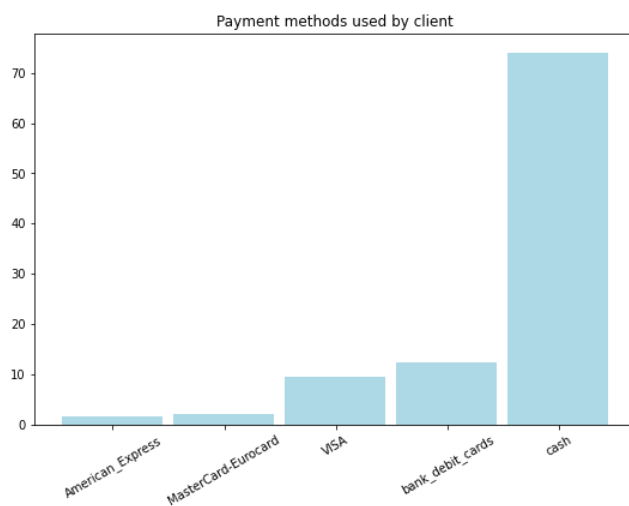
With this graph we can assume that restaurants where alcohol is served are preferred.

6. Determine the most influential age range in our dataset



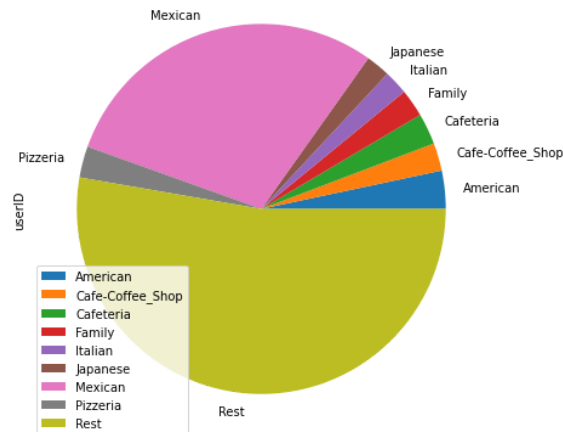
Having previously obtained that the dataset is mainly made up of students, we have verified if there is a prevalence of young people, and that's it.

7. Determine the most popular payment methods among customers



In this case we discovered how bank debit cards are more used than cards on the VISA circuit, and how the cash payment method is still widely used.

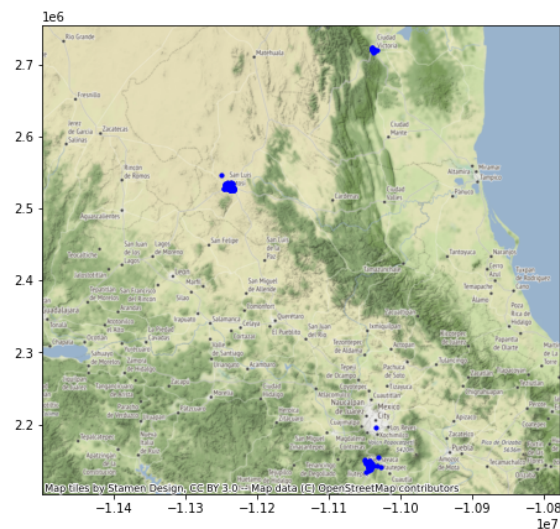
8. Check the division of customers by favourite cuisine



Users' preference falls on Mexican cuisine, it was very likely since the customers of the dataset are located in Mexico.

There're a lot of cuisine types, some with very low share of the public, so we grouped them in the "Rest" slice, that is fairly big. So we can deduce that the market is well distributed apart from the mexican (local) cuisine.

9. Verify the location of users



To deepen the previous analysis we looked for which cities our users were and found a high concentration in the cities of "Ciudad Victoria", "San Luis Potosi" and "Cuernavaca", so they are Mexican users.

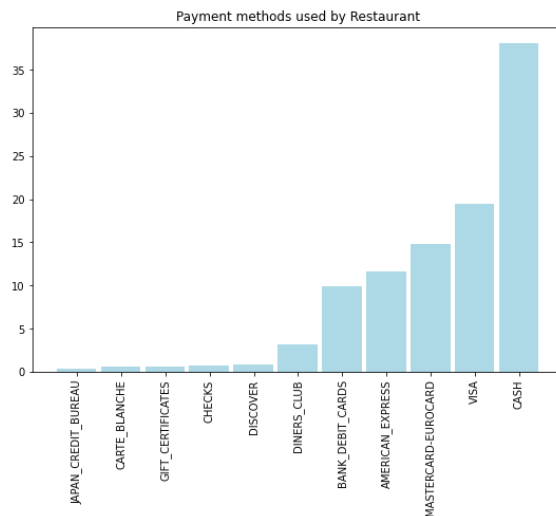
With this first analysis we can say that we have an high probability that our average user is a Mexican, tall between 1.60-1.70m, weights between 60-70kg, is a non-smoking student and with an high probability of drinking, that he was born in around 1990 and that he pay mostly in cash.

A very important result knowing that until a few minutes ago we had no idea who our users were.

And following the supply demand rule, we can already make assumptions on the characteristics of our restaurants.

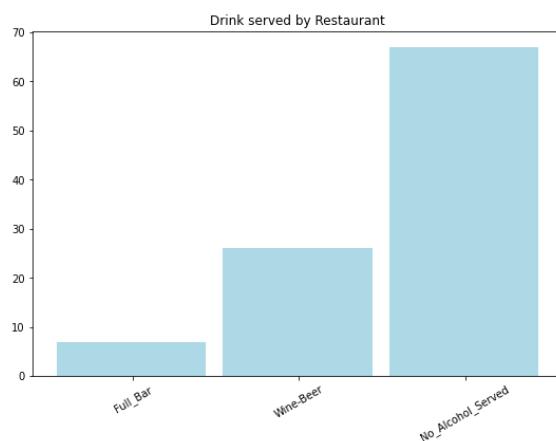
1.3 General information about the restaurants

1. Popular payment methods in restaurants



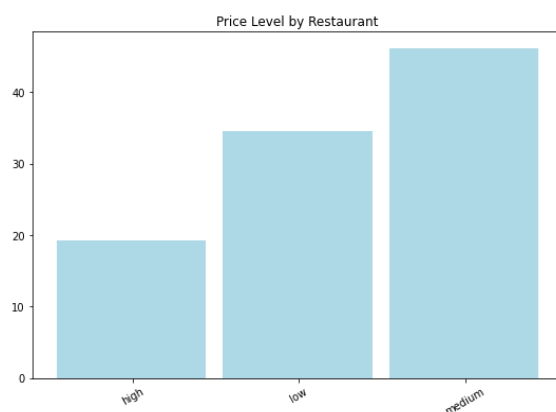
Obviously the restaurants provide many more ways of payment, mainly in cash and through VISA / MASTERCARD and other circuits.

2. Alcoholic drinks served by restaurants



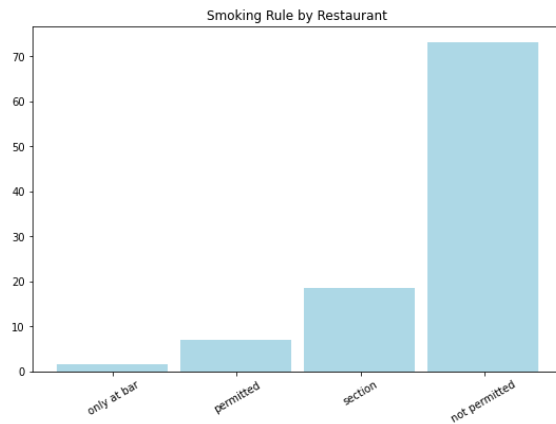
Another unexpected result was that 65% of restaurants do not serve alcohol, in complete opposition with users demands.

3. Price level of restaurants



As for prices, they are in line with the average expenditure of users.

4. Smoking allowed in the restaurant

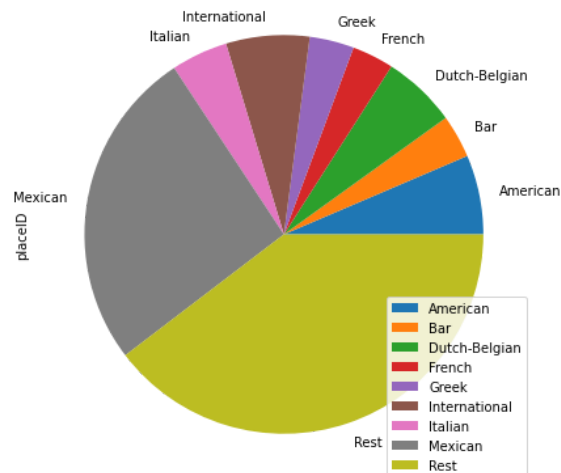


Around 70% of restaurants forbid smoking

Around 20% have dedicated areas

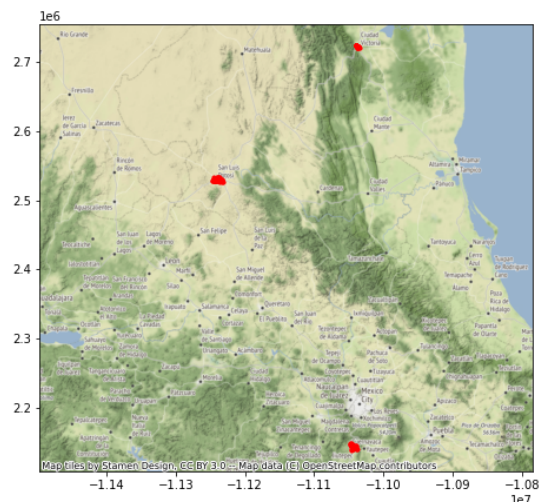
The remainings allows it at the bar or anywhere

5. Division of restaurants by type of cuisine



Being in Mexico it is quite obvious that the prevalence of restaurants is focused on Mexican cuisine, but the portion that covers the other types is very large which implies that we are in the presence of foreigners

6. Location of restaurants in the dataset



Also in this case we had confirmation that the restaurants were located in the three cities listed above

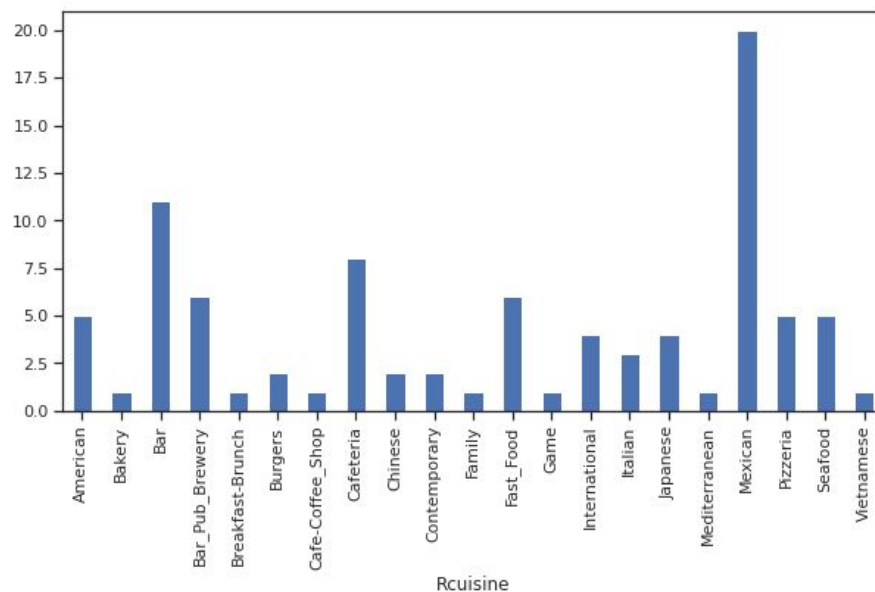
2. Relation between certain features of a restaurant and the received rating

2.1. Research of a possible relation between the best rated restaurants and the kind of cuisine they serve

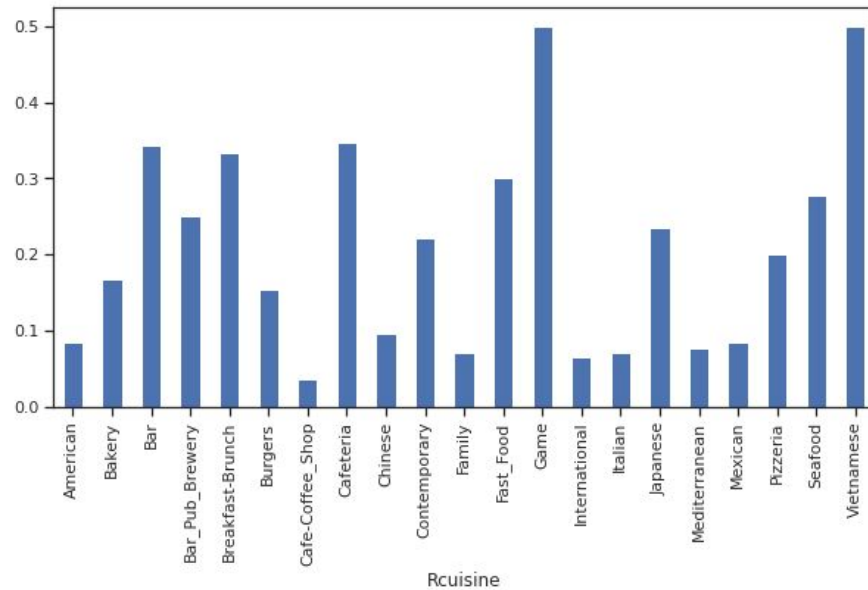
The first task is the research of a possible relation between the restaurants rated excellent (score 2 in all classes) and the typology of cuisine.

We used a combination of PySpark SQL queries, Pandas and Python scripting to achieve these results.

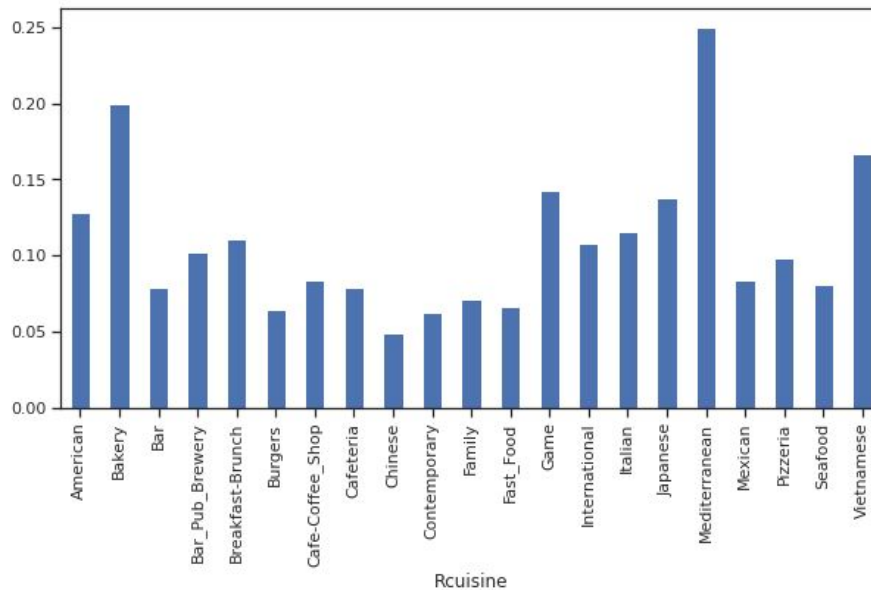
We did this analysis with three different methods, the first attempt was with absolute datas, but the result did not satisfy us because they depended heavily on the number of restaurants for a kind of cuisine, then the number of reviews.



Second attempt was with the data normalized on the total restaurant for each cuisine, the results were already far better but this kind of analysis penalized too much the most populars cuisine.



Third and last attempt was with the data normalized on the total number of reviews for each cuisine. This query was the most correct from a statistical standpoint because the number of excellent reviews was normalized on the total number of reviews.

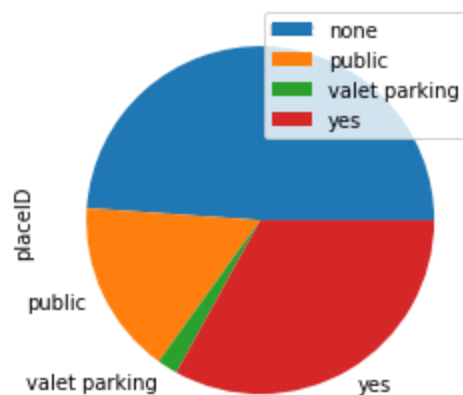


2.2. Research of a possible relation between availability of parking and service rating

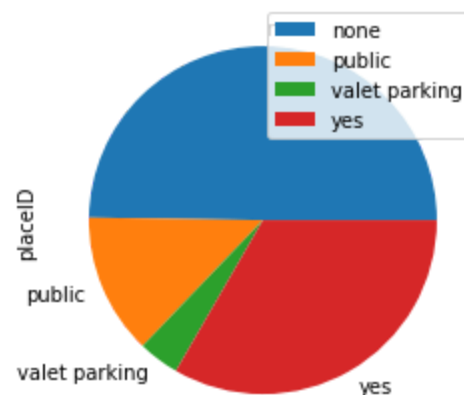
The next task was to find a possible relation between the presence of car parking and the rating given to service.

So via PySpark SQL and Pandas we plotted the data regarding the restaurant with the reviews with the minimum and maximum service ratings joined with the restaurant's parking availability.

Parking in restaurants with minimum score



Parking in restaurants with maximum score



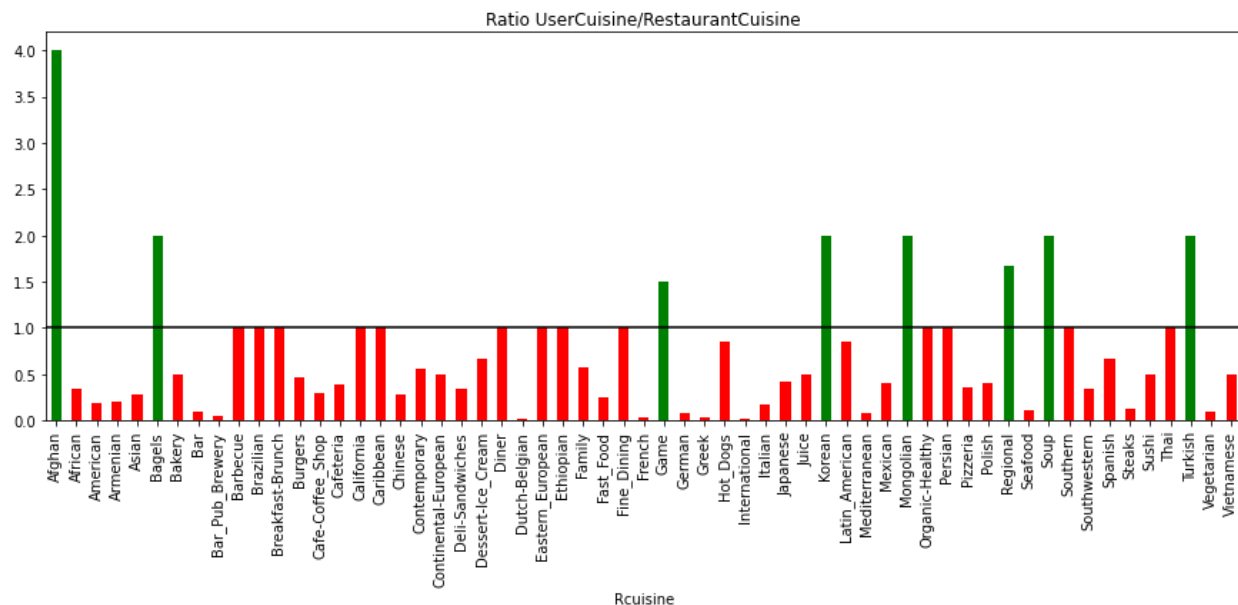
The conclusion was that parking and service ratings are mostly incorrelated. The only small change we were able to find was in the valet parking service percentage. However this service is often offered by high profile restaurants, so it's natural that the service score is generally higher in these kinds of places.

3. Supply and demand analysis for each kind of cuisine

Supply and demand is a micro-economic model for price determination in a free, competitive market. It states that the price of a good will settle at a point where his demand and supply will reach a transaction economic equilibrium.

We can apply similar rules to our model of market where users like a certain type of cuisine and some restaurants offer them.

We coded this analysis using just Pandas importing the data from a PySpark DataFrame.



The green bars in the graph represent the kinds of cuisine for which the demand is not metted quite enough. So in a purely business-economic analysis these are the types of restaurants that are more likely to be successful.

For distinction we threshold ratio of 1.0 that seemed quite right to exclude a large part of the already saturated market.

4. Analysis on the ratings

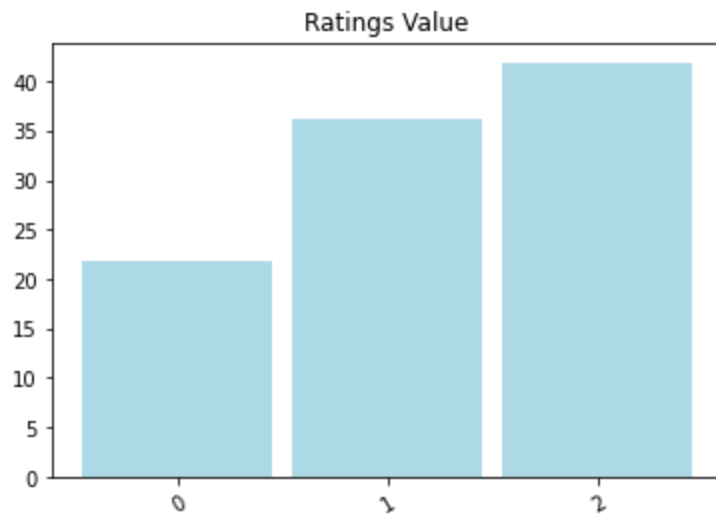
4.1. Number of ratings and other statistical parameters

Through PySpark DataFrame API and MapReduce we ran some basic analysis specifically on the *final_rating.csv* part of the dataset.

The results are the following:

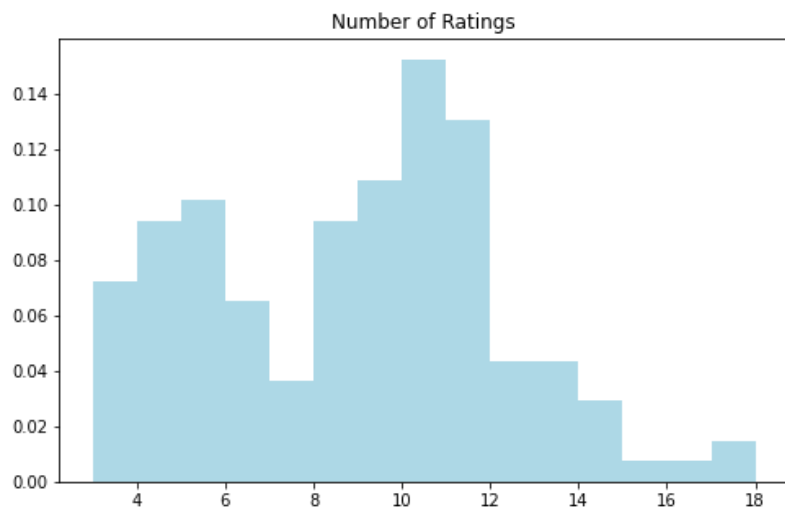
```
Total Ratings = 1161
Total Users = 138
Total Places = 130
Min rating: 0
Max rating: 2
Average rating: 1.20
Median rating: 1
Average of rating released by user: 8.41
Average of ratings receipts per places: 8.93
```

4.2. Rate of usage for each score



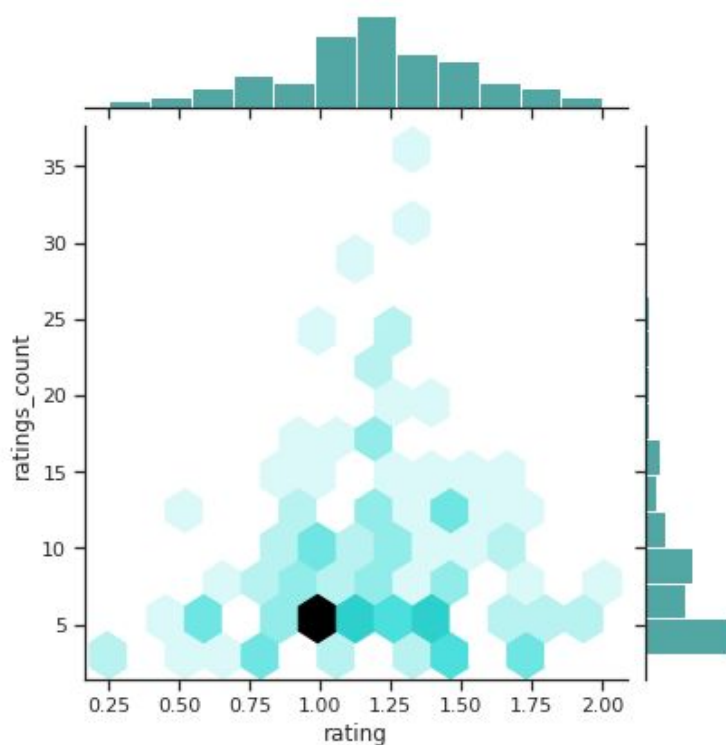
Here we can see that the ratings are well distributed, with a trend towards higher ratings.

4.3. Number of reviews for each customer



The number of ratings for customers seems to be clustered in two main classes, customers with around 5 reviews and customers with around 10 reviews.

4.4 Distribution of average ratings for each restaurant and number of ratings



We wanted to create a more comprehensive visualization for the ratings data, for this we decided to use a different library: Seaborn.

Seaborn is a Python data visualization library based on matplotlib, it provides a high-level interface for drawing attractive and informative statistical graphics.

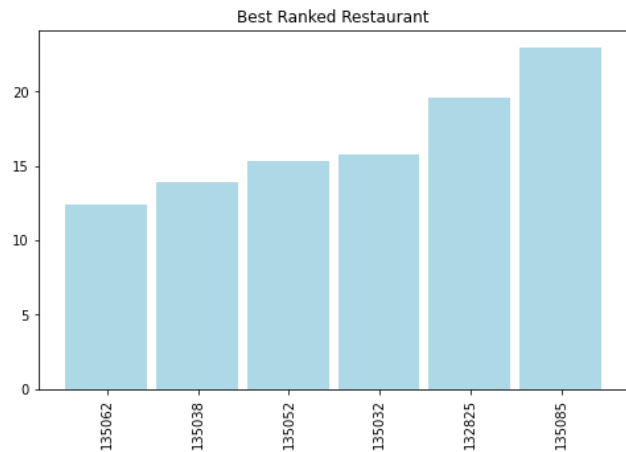
Here we used the *joint plot*, a plot of two variables with bivariate and univariate graphs.

This kind of plot is useful to observe the distribution of the average ratings for every restaurant together with the rating count. The hexagons at the center are colored to represent

the interval where the data are more dense. Here we can observe that there is a large number of restaurants with an average rating of around 1.0 and just over 5 reviews.

5. Recommendation system based on the most liked restaurant and type of cuisine

5.1. Most liked restaurants



1. Restaurante El Cielo Potosino
2. Restaurant la Chalita
3. La Cantina Restaurante
4. Cafeteria y Restaurant El Pacifico
5. Puesto de tacos
6. Tortas Locas Hipocampo

The idea is to suggest the best restaurant based on its absolute popularity among reviewers.

5.2. Search for the most popular restaurants by type of cuisine

	Name	Cuisine	Rating
0	tacos los volcanes	American	1.66667
5	little pizza Emilio Portes Gil	Armenian	1.25
6	Chaires	Bakery	1.4
7	Restaurant Bar Hacienda los Martinez	Bar	1.66667
20	emilianos	Bar_Pub_Brewery	2
26	la parroquia	Breakfast-Brunch	1
27	Log Yin	Burgers	1.75
32	Preambulo Wifi Zone Cafe	Cafe-Coffee_Shop	1.58333
33	cafe punta del cielo	Cafeteria	1.83333
42	Restaurant Wu Zhuo Yi	Chinese	1.25
45	Restaurante la Parroquia Potosina	Contemporary	1.75
47	Mariscos Tia Licha	Family	1.6
49	tortas hawai	Fast_Food	1.33333
57	KFC	Game	1.42857
58	Restaurant Las Mananitas	International	2
62	El Mundo de la Pasta	Italian	1.5
66	Michiko Restaurant Japones	Japanese	2
71	Log Yin	Mediterranean	1.75
72	La Estrella de Dimas	Mexican	1.8
99	Little Cesarz	Pizzeria	1.3
104	puesto de gorditas	Regional	0.5
105	Mariscos El Pescador	Seafood	1.69231
110	Restaurant Familiar El Chino	Vietnamese	1.16667






The idea behind this job is to list and recommend the best restaurant for every kind of cuisine, based on the average ratings that customers have given to it.

6. Recommendation system using Collaborative Filtering with MLlib

The last and most comprehensive goal of our project is to develop a recommendation algorithm for the restaurant and the users present in our dataset.

We went for an approach that is broadly used in industry, Collaborative Filtering.

The main advantage of this approach is that it requires less data features, only the Item, User and Ratings are used to generate predictions.

		Cuisines				
						
C u s t o m e r s	Marco	1	2	2	1	
	Dario		2		1	0
	Alex	1	0		2	
	Elettra	1		0	2	X
	Paola		2			2
	Oriana	1		2	1	

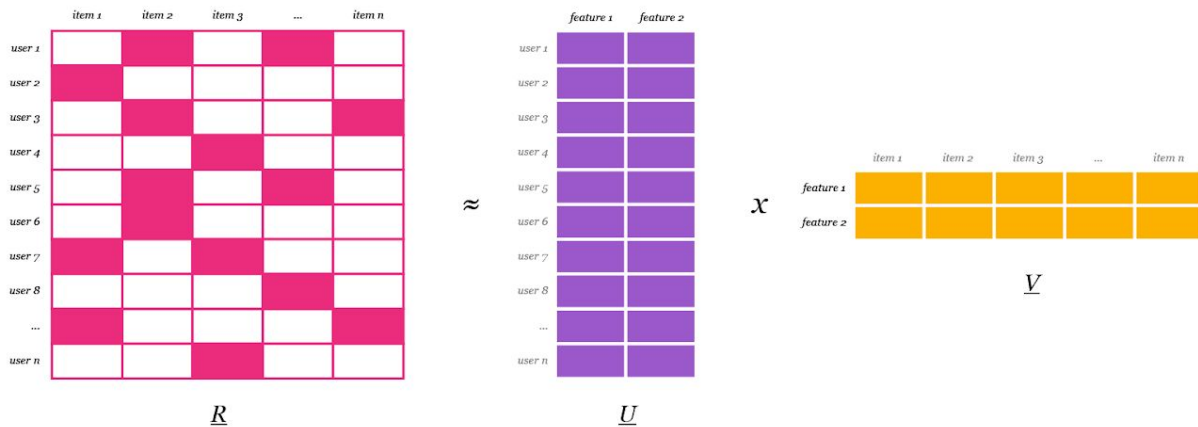
Prediction

The algorithm can perform fully without the need to track user habits and tastes (other than the data explicitly given to the system with reviews). This leads to a far better efficiency in data storage and mainly to less privacy concerns for users. So it is even easier to run these kinds of algorithms without complex data use policies or in restrictive legislation (EU with GDPR).

The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an item, A is more likely to have B's opinion on a different item than that of a randomly chosen person. In less words the algorithm finds users that share the same tastes.

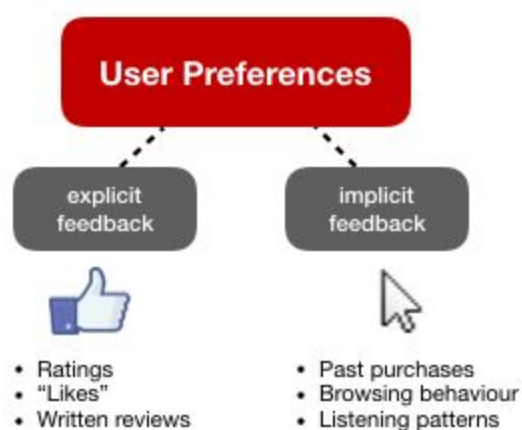
The main drawback for this method are: data sparsity (number of items \gg number of ratings), cold start (for both new users and new items), gray sheeps (users that do not consistently agree or disagree with any others), shilling attack (auto-promotion or competitors demotion) and diversity decrease (need to artificially promote items in the "long tail"). To overcome these problems, in real-world cases, hybrid approaches are implemented.

There're many mathematical models for Collaborative Filtering, here the choice was on the one implemented by MLlib (which is also one of the best): Alternating Least Squares (ALS).

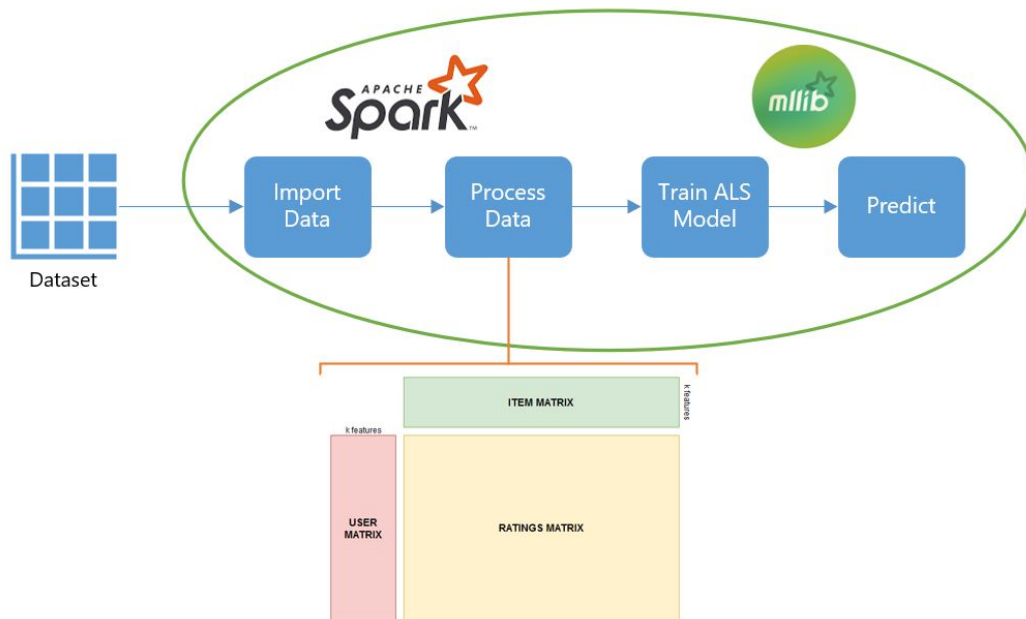


The ALS algorithm applies an approximate factorization to the sparse User-Item-Rating matrix, discovering a number (rank) of hidden features that can be used to make predictions.

In our model we used explicit preferences (given by the user), this fits well with our dataset where we have reviews. In real-world applications implicit feedback is more used (ratings gained from a data mining pipeline), but the results are not as good, because it adds another degree of uncertainty.

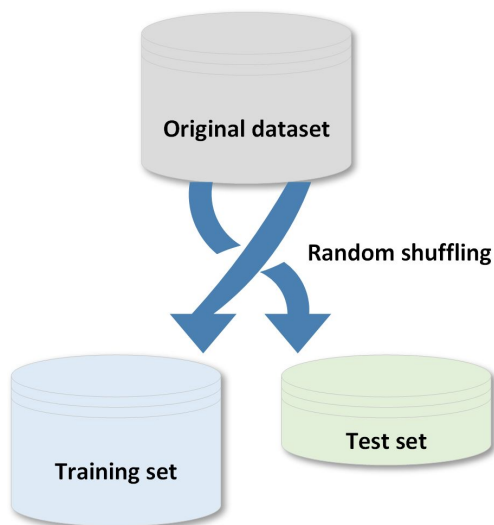


Following is our complete ML pipeline:



6.1 Extraction of characteristics from the dataset and training of the model

With the help of RDD and MapReduce we extracted useful data (UserID, PlaceID, Rating) from the *rating_final.csv* file.



We then divided it in two parts, training set and test set to later evaluate the accuracy of our recommendation model.

6.2 Top 10 recommended

We have two types of recommendations based on Collaborative-Filtering:

Top 10 restaurants suggested for a user

User	Restaurant	Rating
1077	135025	2.16364
1077	135042	2.03145
1077	132825	1.98648
1077	135075	1.93129
1077	135041	1.86017
1077	135057	1.81492
1077	132955	1.73593
1077	135032	1.73527
1077	132958	1.69776
1077	135072	1.69612

Top 10 users suggested for a restaurant

User	Restaurant	Rating
1109	135108	2.16664
1053	135108	2.04773
1054	135108	2.0391
1111	135108	1.98165
1088	135108	1.98087
1126	135108	1.97576
1078	135108	1.96684
1037	135108	1.94618
1108	135108	1.77642
1071	135108	1.76059

6.3 Evaluation of the model

For evaluating the Recommender System we divided the dataset in two portions (training set and test set), and following the workflow. For evaluating the accuracy is popularly used two main measures: Root Mean Squared Error (RMSE) and Mean Absolute Error(MAE).

They are used depending on the context of the dataset:

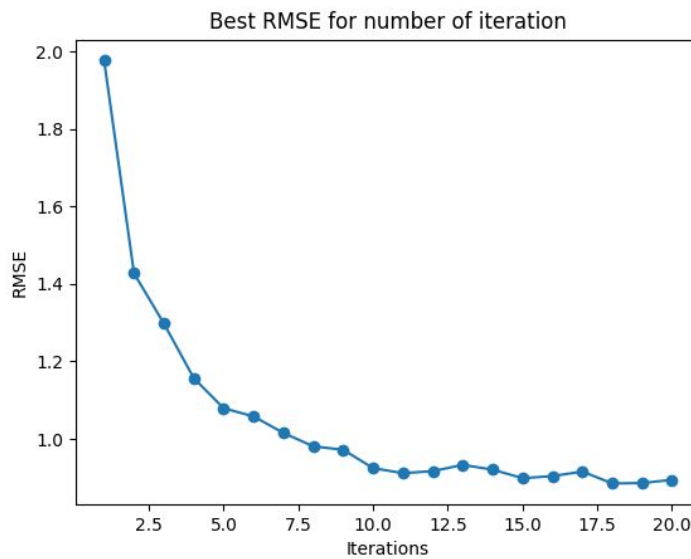
- Mean Average Error (MAE) does not give any bias to extrema in error terms. To get a holistic view of the Recommender System, it is better to use MAE. If there are outliers or large error terms, it will weigh those equally to the other predictions.
- Root Mean Squared Error (RMSE) is more prone to being affected by outliers or bad predictions.

```
Mean Squared Error = 0.7764162727197318
Root Mean Squared Error = 0.8811448647752149
```

For the training of the model we have tested two parameters:

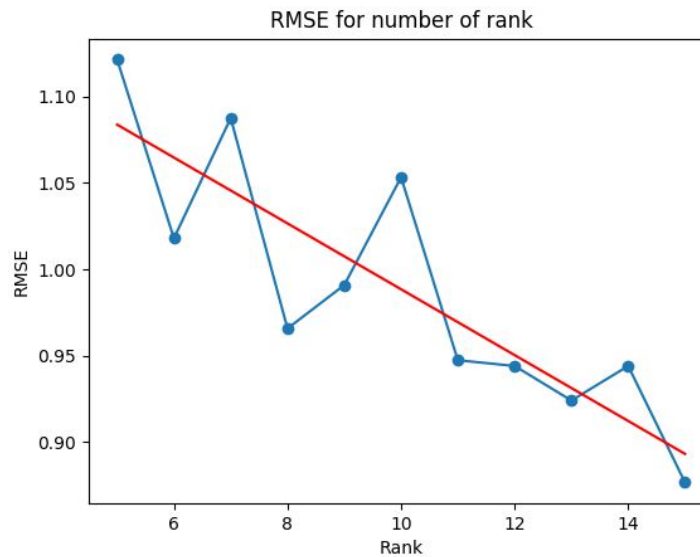
1. *rank*: This refers to the number of factors in our ALS model, that corresponds to the number of hidden features in our low-rank approximation matrices.
2. *iterations*: This refers to the number of iterations to run, each iteration in ALS is guaranteed to decrease the reconstruction error of the ratings matrix.

Increasing these two parameters increase significantly the computational cost, so it is a good practice to determinate the right value to have a good compromise with prediction results and computational cost.



We tested the ALS models and looked that it will converge to a reasonably good solution after relatively few iterations. We can deduce that starting from 10 iterations the resulting RMSE is pretty stabilized, so we choose this value for an optimal usage of computing resources.

We also tested what happened by increasing or decreasing the number of ranks compared to the RSME value.



Testing the RMSE on a set of ranks from 5 to 15 showed a decreasing trend with some noise (this will get lower as dataset size increases), based on the available computing resources, result obtained and recommended values from MLlib authors we choose a value of 11.

6.4 Execution Time

- Total Notebook execution time: ≈ 66 sec
- Model Training rank=11 and iterations=10:
 - CPU times: user 21.2 ms, sys: 87 μ s, total: 21.3 ms
 - Wall time: 3.29 s
- Top-K Execution time:
 - Top-10 restaurants for userID:
 - CPU times: user 5.62 ms, sys: 171 μ s, total: 5.79 ms
 - Wall time: 222 ms
 - Top-10 customers for placeID:
 - CPU times: user 708 μ s, sys: 4.16 ms, total: 4.86 ms
 - Wall time: 114 ms