Cities and Restaurants

Ahmet Oruc
oruc.ahm@gmail.com

Sergen Topcu
sergentopcu08@gmail.com

Denizkaan Araci
denizkaanaraci@gmail.com

Machine Learning Project, Department of Computer Engineering
Hacettepe University, Ankara, TURKEY

Abstract

In this paper, we offer restaurants for customers. When doing these things, we have taken into consideration the points that the customer gave to the restaurants s/he had visited before. Since our project predicted restaurants in the city that the user wanted, we decided to name our project "Cities and" Restaurants". In our project "Cities and" Restaurants" we used 3 different machine learning algorithms: decision tree, naive Bayes, and neural network algorithms.

1. Introduction

People want to eat outside for a good day, for socializing, for being touched. Choosing the right restaurant is an important part of the day's beautiful past. Today, the number of people who go to a restaurant and leave satisfied is quite high. We also want to make people happy with this project and find a nice restaurant without spending much time.

Nowadays, when the time is limited and valuable, this method will make it possible to make faster and more accurate choices. There are many types of research and projects related to this problem. Some of these projects are making restaurant estimates by looking at the proximity of the restaurant to some of the restaurant presentations and some of the restaurant's social areas. This work on the subject continues. At the same time, machine learning projects and new methods are being developed to help in this development.

Restaurants have a lot of features that make it a predict for the customers. For example, looking at the food, proximity, speed of service, alcohol or smoking, parking. These are some of the features customers consider when choosing a restaurant. In our project "Cities and Restaurants" we

want to develop a method that calculates the most appropriate restaurants that our customers need to consider.

Our first goal when developing this system is to suggest where the customers will appreciate. In addition, restaurants are expected to improve their services and add new features to earn high scores.

2. Related Work

Yelp dataset is a convenient dataset for many machine learning implementation. Since dataset is published by Yelp Company, a lot of research has been done on it.

First of all, in 2013 Kaggle has organized a competition called RecSys Challenge 2013: Yelp Business Rating Prediction[5] associated with ACM RecSys 2013. The theme of this years competition was personalized business recommendations for Yelp users. In the contest, Contestants were asked to predict the users future ratings of businesses. Participants created a model to predict the rating a user would assign to a business. Models were graded on accuracy using the root mean squared error metric. At the end, A total of $500 in prize money were presented to the winners.

In August 17, 2013, Naomi Carrillo, Idan Elmaleh, Rheanna Gallego, Zack Kloock, Irene Ng, Jocelyne Perez, Michael Schwinger, Ryan Shiroma from University of California has released a research name called Recommender Systems Designed for Yelp. com[9] for Kaggle that mentioned previously. They used business features and user ratings for predictions. They used Collaborative Filtering with Nearest Neighbor Method, Weighted Similarity-Jaccard Index, Matrix Factorization, User/Business means, Weighted averages, and clustering. End of the research they took bad results because of the low-quality reviews and untrusted users. Their Kaggle rankings are shown below the table.

Table 2: Results

Method	RMSE	Kaggle Ranking
Mean Predictor	1.25112	130th
Weighted Mean Predictor	1.25217	135th
SVD	1.25622	153rd
Boosted Nearest Neighbor	1.24977	122nd
Clustering	1.30861	288th
Vote Weighting	1.28893	255th
Jaccard Index Similarity	1.32948	366th
Blended Model	1.24039	51st

In 2016, Sumedh Sawant and Ginai Pai from Stanford University has made research name called Yelp Food Recommendation System.[7] They used business attributes and categories as a feature. For prediction, they used singular value decomposition, hybrid cascade of K-nearest neighbor clustering, clustered weighted bi-partile graph projection algorithms. In the end of the research, they got 1.14625 rmse score with clustered weighted bi-partile graph projection.

In 2016, Data Scientist Nick Becker from Enigma Technologies released a research called Predicting Ratings from Yelp Reviews.[4] He used users review texts and stars as a feature. He made text analyzing using stochastic gradient descent logistic regression. In the end of the research with changing ratings (change 3-star reviews to 5-star reviews), he got %80 accuracy.

In 2017, Drew Zhong from Stanford University made project called Yelp Recommendation System.[8] He used Item-based collaborative filtering. In the end, They got average %70 accuracy with Naive Bayes, Logistic Regression and Decision Tree.

In our project, we use Item-based collaborative filtering. If we look at previous projects that using Item-based collaborative filtering;

1. They all worked in restaurants in Yelp Dataset.
2. In some projects, They used reduced Dataset(Kaggle Competition). Some of them used the whole dataset.
3. They used naive Bayes, logistic regression, decision tree, clustered weighted bipartite graph projection, singular value decomposition, Weighted Similarity-Jaccard Index, Matrix Factorization, User/Business means, Weighted averages, and clustering.
4. If we look at the average, they got 50% 60% accuracy in their projects.
5. Generally, they got bad results because of the low-quality reviews and untrusted users.

If we were to look at the things we did differently from our other works;

1. Firstly we tried to clean dataset for to be as consistent as possible. We picked 3 cities with the most review. Then we deleted restaurants with low review count and inconsistent information about attributes and categories. Then we delete users that have low review count.
2. For restaurants feature selection, we analyze all features one by one. For example, if one feature has average more than 90% true or false for every restaurant, we decided that this feature has no influence to prediction Or feature that is not in 60% of restaurants has not good impact to prediction and reduces the quality of dataset.We did not add features like these in restaurants feature.
3. After preparing dataset we implemented Support Vector Machine, Multilayer Neural Network, and Naive Bayes algorithms. Users with more than 200 comments were made us more than 65% accuracy with all algorithms.

3. The Approach

In our "Cities and Restaurants" project, we aim to identify a new restaurant where people will go ahead with a character based on their characteristics. We obtained this prediction using 3 different machine learning algorithms. These 3 algorithms are classification algorithm. These 3 methods are Decision Tree, Neural Network and Naive-Bayes algorithm. First of all, we decided on 25 different features in these 3 algorithms. These features are what we get from the Yelp Dataset[6]. These are some of the things people want to have after going to a restaurant.

These features are as follows:

1. Has TV (True or False)
2. Alcohol (1: none, 2: beer and wine, 3: full bar)
3. Noise Level (1: quite, 2: average, 3: very loud)

4. Caters (True or False)
5. WiFi (0: no, 1: free, 2: paid)
6. Restaurants Reservations (True or False)
7. Bike Parking (True or False)
8. Restaurants TakeOut (True or False)
9. Good For Kids (True or False)
10. RestaurantsTable Service (True or False)
11. Outdoor Seating (True or False)
12. Restaurants Price Range (1,2,3)
13. Restaurants Delivery (True or False)
14. Wheelchair Accessible (True or False)
15. Good For Meal
    (a) dessert (True or False)
    (b) latenight (True or False)
    (c) breakfast (True or False)
    (d) dinner (True or False)
    (e) lunch (True or False)
    (f) brunch (True or False)
16. Business Parking
    (a) garage (True or False)
    (b) street (True or False)
    (c) validated (True or False)
    (d) lot (True or False)
    (e) valet (True or False)

The rest of the users were getting scores 1-5 on the restaurant. We have transformed them into 3 different categories. They are 1-2: do not like 3-4: like 5: very like

3.1. Multinomial Naive Bayes

Naive Bayes classifiers are linear classifiers known to be simple and effective to implement. The naive Bayes algorithm is derived from Bayes theory. It works better in small data sets than other classification algorithms.

Formula:

The formula[2] that the likelihoods of classes arrive is as follows:

$$P(w_j) = N_{wj}/N_c$$

N_{wj}: Count of samples from class w_j.
N_c: Count of all samples.

calculating the probabilities of all classes to arrive, the Bayes theorem[2] below is used to calculate the likelihood that a class will arrive based on a feature:

$$P(w|c) = count(w, c) + 1/count(c) + |V|$$

w = the number of the feature value in the data set for which the likelihood is calculated

c = total number of all features in class 0

$+1$ = If we assume that the feature is never found in a class, we can observe that the probability is 0. If we want a very small value instead of 0 to prevent it,

$|V|$= Total number of features (When formate + 1 is added, the generic is added to do not distort.)

We calculate the results, which are multiplied by the probability of all the probabilities and the likelihood of the class, and we prefer which class is more appropriate.[2]

$$P(X_1...X_d|Y) = \prod_{i=1}^{d} P(X_i|Y)$$

We got the results using the Multinational Naive Bayes algorithm in the scikit-learn library. We wanted to make three different class estimates for this algorithm. This algorithm used 25 different features mentioned above.

3.2. Decision tree

Decision tree creates classification or regression models in the form of a tree structure. We use the classification model because the problem in our project is classification problem. By dividing a data set into smaller and smaller subsets, an associated decision tree is gradually developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. The leaf node represents a classification or decision. Decision trees can handle both categorical and numerical data. This algorithm uses 25 different features. 3 different class estimates were made.

Some of the advantages[1] of Decision Trees are:

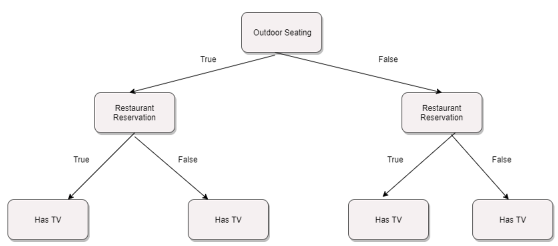
1. It's simple to understand and interpret. Trees can be visualized.
2. The cost of using a tree is the logarithm of the number of data points used to train the tree.
3. Be able to tackle problems that are too big.

The disadvantages[1] of decision trees include:

1. Decision tree learners can create extremely complicated trees that do not generalize the data.

2. Decision tree learners, if some classes dominate, form prejudiced trees. For this reason, it is necessary to balance the data set before it coincides with the decision set.

3. Decision trees are difficult concepts to learn because they do not express them easily, like XOR, parity or multiplexer problems.



3.3. Neural Networks

Neural Network is a structure established as layers. The first layer is called the input, the last layer is called the output. Layers in the middle are called 'Hidden Layers'. Each layer contains a certain number of 'Neurons'. These neurons are linked to each other by 'Synapse'. Synapses contains a coefficient. These coefficients tell us how important the information on the neuron is. This information is what we use in the project. It is estimated from 3 different classes using 25 different features.

Process steps[3]:

1. The system is given certain inputs and values to be issued

2. These inputs are multiplied by the coefficients.

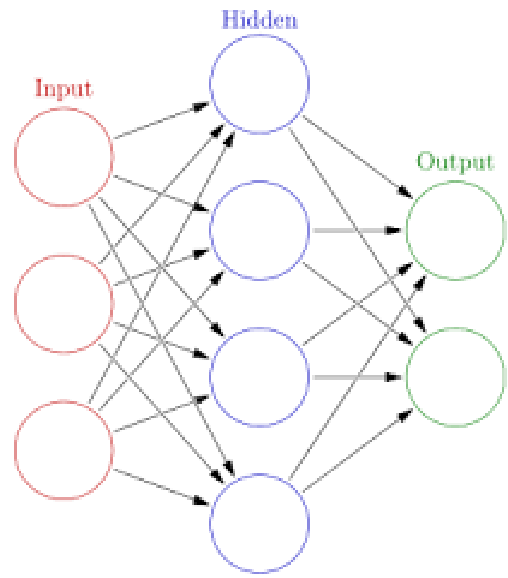
3. We get an output by inserting this sum into an activation function

4. We find the error rate by taking the difference of the output we obtained with the expected value.

5. And we update our coefficients again according to this error rate.

6. This process repeats in a certain cycle.

This is the neural network learning process in this section. Ultimately, our coefficients find the most accurate value. And our network is learning to solve a particular problem.



4. Experimental Results

We mentioned that we used 3 different machine learning algorithms in our project. These were Decision Tree, Naive Bayes and Neural Network algorithms. Here are the results we achieved using these 3 algorithms:

The number of reviews on the table refers to the total number of comments one person made to restaurants.

Reviews	Users Count
40-70	1301
70-100	442
100-200	292
200-300	35
300-∞	16

You can see the review numbers of the following table users. The range of the maximum number of users is 40-70. The total number of users in our data set is 2086. The results we get from these users are as follows:

4.1. Decision tree

The Decision Tree algorithm has three different properties that affect the success of the estimates. These are max depth, min sample leaf and test size.

Reviews	T.S:0,3	T.S:0.2	T.S:0,1
40-70	58,5533	57,1175	57,4151
70-100	61,3762	60,1254	60,4421
100-200	65,4381	64,1979	64,4084
200-300	70,7346	69,8604	69,8241
300-∞	73,3169	72,8863	72,8431

Max Depth = 2 - Min Sample Leaf =5 - T.S=Test Size

This graph expresses the results of the decision tree algorithm according to the number of reviews and the test size. As you can see in the table, we can observe that the results are better when you test. At the same time, you can see that the decision tree algorithm makes more successful predictions in people with more reviews. In this graph, max depth = 2 and min sample leaf = 5 are fixed and results are collected according to the test size.

Reviews	M.D:20	M.D:5	M.D:2
40-70	54,8517	54,8874	58,5533
70-100	56,4069	56,4554	61,3762
100-200	59,1907	59,8524	65,4381
200-300	64,0912	65,8972	70,7346
300-∞	67,5755	70,0249	73,3169

Test Size=0.3 - Min Sample Leaf =5 - M.D=Max Depth

This graph expresses the results of decision tree algorithm according to the number of reviews and max depth. As you can see in the table, when the max depth value increases, we can observe that the success in the results decreases. At the same time, you can see that the decision tree algorithm makes more successful predictions in people with more reviews. In this graph, the test is fixed at size = 0.3 and min sample leaf = 5, and the results are collected according to the max depth.

Reviews	M.S.L:20	M.S.L:10	M.S.L:1
40-70	61,6480	58,6015	56,2902
70-100	62,3403	60,4659	59,4425
100-200	64,8929	64,3478	63,7342
200-300	70,3332	69,9130	69,4801
300-∞	73,0441	72,7478	72,6450

Test Size=0.3 - Max Depth = 2 - M.S.L = Min Sample Leaf

This graph expresses the results of decision tree algorithm according to the number of reviews and max depth. As you can see in the table, when the max depth value increases, we can observe that the success in the results decreases. At the same time, you can see that the decision tree algorithm makes more successful predictions in people with more reviews. In this graph, the test is fixed at size = 0.3 and max depth = 2, and the results are collected according to the min sample leaf.

4.2. Multinomial Naive Bayes

Reviews	T.S:0,3	T.S:0.2	T.S:0,1
40-70	58,3808	58,4413	58,5207
70-100	60,5902	60,6718	60,6635
100-200	64,0505	64,1151	64,3862
200-300	69,0857	69,3153	69,7107
300-∞	72,4770	72,7283	72,8529

This graph represents the results of the Multinomial Naive Bayes algorithm according to the number of reviews and the size of the test. As you can see in the table, we can observe that the test results are more successful when we test you. This is the biggest reason why we are increasing the number of train data to increase the machine learning. At the same time, you can see that Multinomial Naive Bayes algorithm makes more successful estimates in people with more reviews.

4.3. Neural Networks

H.L.S=Hidden Layer Sizes

Reviews	50,100	100,100	100,50	100,25
40-70	60,1421	60,1124	60,0983	60,3539
70-100	62,6963	62,6911	62,4388	62,8979
100-200	66,7559	66,6051	66,6905	66,8344
200-300	70,8985	71,1009	71,1242	70,7813
300-∞	74,5065	74,3539	74,5107	74,4261

Activation = Reul - Learning rate = Constant

This graph expresses the results of Neural Network algorithm with Hidden Layer size change. As you can see in the table, 4 different Hidden Layer size values are used. These are (50,100), (100,50), (100,100), (100,25). We can observe that when the value of the hidden layer size changes, the success results also change. We can observe that these changes are less than the other graphs. At the same time, you can see that the Neural Network algorithm makes more successful predictions in people with more reviews.

Activation

Reviews	Logistic	Identity	Tanh	Reul
40-70	61,3487	60,0903	59,6042	60,3539
70-100	63,0243	62,7104	62,0334	62,8979
100-200	66,7104	66,5330	66,3242	66,8344
200-300	70,8420	70,9602	71,0176	70,7813
300-∞	74,3806	74,3806	74,3594	74,4261

Hidden Layer Sizes = (25,100)

This graph expresses the results of the Neural Network algorithm according to the activation functions. As you can see in the table, there are 4 different activation functions. These are Logistic, Identity , Tanh and Relu. We can observe that the successes of these activation functions are different from each other. We see that the Logistic and Relu activation functions are often better than the other 2 activation functions. At the same time, you can see that the Neural Network algorithm makes more successful predictions in people with more reviews. In this graph, the hidden layer (25,100) is fixed and the results are collected according to the activation functions.

L.R = Learning Rate

Reviews	Invscaling	Adaptive	Constant
40-70	59,1686	60,3514	60,3539
70-100	60,9964	62,8893	62,8979
100-200	66,1508	66,8419	66,8344
200-300	70,6606	70,7496	70,7813
300-∞	74,3806	74,4528	74,4261

Hidden Layer Sizes = (25,100) - Activation = Relu

This graph expresses the results of the Neural Network algorithm according to the activation function of the Relu activation function. As you can see in the table, there are 3 different activation types. These are Invscaling, Adaptive, Constant. We can observe that the successes of the types of these activation functions are different from each other. We can observe that the results of the Adaptive and Constant types are more successful. At the same time, you can see that the Neural Network algorithm makes more successful predictions in people with more reviews. In this graph, the activation function is fixed as Relu and Hidden Layer (25,100), and the results are obtained according to the activation functions.

L.R.I = Learning Rate Init

Reviews	0.01	0.005	0.001	0.0005
40-70	54,8260	56,3885	60,3539	60,7513
70-100	56,7809	58,9626	62,8979	62,9510
100-200	62,0337	64,2868	66,8344	66,8006
200-300	68,6926	70,1401	70,7813	71,0475
300-∞	72,2591	73,2951	74,4261	74,4539

Hidden Layer Sizes = (25,100) - Activation = Relu - Learning Rate = Constant

This graph expresses the results of Neural Network algorithm with the change of learning rate value. As you can see in the table, four different learning rate values are used. These are 0.1, 0.005, 0001, 0.0005. As learning rate changes, we can observe that success outcomes have changed. We can observe that the results are more successful when the learning rate value decreases. At the same time, you can see that the Neural Network algorithm makes more successful predictions in people with more reviews. In this graph, the activation function is fixed as Relu, Hidden Layer = (25,100), learning rate = constant, and the results are collected according to the learning rate value.

5. Conclusions

You can see the average results of the following three machine learning algorithms on the table:

Reviews	Decision Tree	Naive Bayes	Neural Network
40-70	58,5533	58,5207	60,7513
70-100	61,3762	60,6635	62,9510
100-200	65,4381	64,3862	66,8006
200-300	70,7346	69,7107	71,0475
300-∞	73,3169	72,8529	74,4539

As you see in the table above, the average results we get with 3 different algorithms are different from each other. We can say that the success rates we have received are very close to each other for the 3 algorithms. You can see that the results we obtained from the Neural Network algorithm are the most successful. We can observe that Decision Tree algorithm yields slightly better results than Naive Bayes algorithm. These observations are made according to the data we use. At the same time, as the number of people making reviews increases, we can say that the results are more successful.

The ways to increase the accuracy rates are as follows:

1. The dirty parts of the data we use can be cleaned.
2. Program can focus on people who have done more reviews.
3. Different results can be obtained by different machine learning algorithms.
4. The number of train data can be increased.

References

[1] Decision tree algorithm. http://scikit-learn.org/stable/modules/tree.html.

[2] Hacettepe University BBM406:Fundamentals of Machine Learning - Naive Bayes Classifier. https://web.cs.hacettepe.edu.tr/~aykut/classes/fall2017/bbm406/slides/l8-bayes-classifier.pdf. page:15 and 41.

[3] Neural network. https://medium.com/@sametaylak/neural-network-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-4363922b8f6a. Author: Samet Aylak.

[4] Predicting ratings from 1+ gbs of yelp reviews. https://beckernick.github.io/yelp-reviews/.

[5] Recsys2013: Yelp business rating prediction. https://www.kaggle.com/c/yelp-recsys-2013.

[6] Yelp Dataset. https://www.yelp.com/dataset.

[7] Yelp food recommendation system. https://github.com/shivambijoria/yelp_recommendation.

[8] Yelp recommendation system. https://github.com/drew109zhn/Yelp-Recommendation-System/.

[9] N. Carrillo, I. Elmaleh, R. Gallego, Z. Kloock, I. Ng, J. Perez, M. Schwinger, and R. Shiroma. Recommender systems designed for yelp. com, 2013.