Predicting Restaurant Success and Rating with Yelp

Meenakshi Paryani - 011819392, Foram Mehta – 011548446, Noopur Joshi - 011540334

SJSU
meenakshi.paryani@sjsu.edu, foram.mehta@sjsu.edu, noopur.joshi@sjsu.edu

December 2, 2017

ABSTRACT— This paper aims to predict restaurant success and rating, and also provides top ten most important and not important features that impacted success. Using the Yelp Challenge Dataset, we ran a Chi-squared test to select the restaurant features that hold the most weight. We used binary and multiclass classification and linear regression model to predict a restaurant's success and rating.

INTRODUCTION

According to a frequently cited study by Ohio State University on failed restaurants, 60% do not make it past the first year, and 80% go under in five years. There are several factors that can make or break a restaurant's success. We want to know the cause behind and how we can predict restaurant success given its information before it opens. The findings from this project can help predict the success of a new restaurant given its current attributes, and give insight about what attributes can be improved to increase its success.

PROBLEM DEFINITION

The goal of our project is to:

- Predict restaurant success with binary classification (If it has more than 30 reviews and 3.5 star rating).
- Predict restaurant rating with multiclass classification (Rates the restaurant from 1 – 5 stars).
- Predict restaurant rating with linear regression (Predicts the rating with regression model).
- Provide top 10 features that determine restaurant's success.
- Provide top 10 features that determine restaurant's failure.

DATA PROCESSING

We used Yelp as our dataset. Yelp is an extremely popular, easy-to-use platform that people use to publish reviews and rate restaurants. The dataset contains the number of reviews, rating, and business attributes, such as Wifi, Noise Level, Cater, Price Range, Takes Reservations, Parking, Ambience, Attire, Good For, etc. There are 51624 records in the Yelp dataset after filtering by restaurants. The first step was to convert json values into pandas dataframes. As part of data preprocessing for binary classification we classified the restaurants into classes '0' and '1' based on the criteria as follows –
Class 1 – restaurants having star rating >= 3.0 and number of reviews >= 30
Class 0 – restaurants having star rating <= 3.0 and number of reviews <= 30

For multiclass classification, we rounded off the star rating and created five labels namely 1 to 5. In case of multiclass classification and regression we tried two different approaches. First one on the whole dataset and the other by removing outliers by eliminating all the restaurants having a review count less than 30.

We performed data preprocessing in the following two ways:

1. Data preprocessing using Label Encoding:

   For converting categorical attributes to numerical values, we used Label encoding technique. For instance, we had a column for Attire attribute. It had values like Classy, Casual, Dressy, Upscale, etc. So, for each unique value we assigned a numerical value like Classy – 0, Casual – 1, Dressy – 2, etc. For missing values, we assigned 'dummy' value and the label encoder function converted that to an integer value relevant to the column data.

2. Data preprocessing using Pandas (Get Dummies):

   In this case for converting categorical attributes to numerical values, we used the get_dummy function provided by pandas library. It converted each categorical attribute value into a new column and then assigned it a value of '0' or '1' based on its presence.

The final step of data preprocessing was to divide the dataset into training and test. We used k fold cross validation technique to divide the dataset into 80:20 ratio for training:test respectively.

DIMENSIONALITY REDUCTION

The data obtained from data preprocessing was converted to a CSR Matrix and then normalized using SKLearn library. The number of features obtained from that step were 881. We then, applied TruncatedSVD and Chi Square test as the dimensionality reduction techniques to trim the features down to around 50-100. When we tried to apply classification models on this data we got very low training accuracies less that 30%. So, we went back to the data preprocessing step and removed the irrelevant columns such as Corkage, BYOB, Zipcode, Address, etc. which did not contribute to the classification algorithm. In addition to that those columns also did not have sufficient data.
After applying data preprocessing again to the reduced number of attributes we obtained 143 columns for further processing. To this dataset we applied SelectKBest technique with Chi Square test for feature selection. For binary classification, we selected top 30 features from class 0 and top 30 features from class 1. We tried different values of k ranging from 5-60 and the best value of k for all algorithms was obtained in the range of 45-52.

BINARY CLASSIFICATION

We tried five different classification models namely –

1. **Decision Tree Classifier**

   We used the decision tree classifier of SKLearn for binary classification. The advantage of using decision tree is that its performance is not affected by nonlinear relationships between features. However, decision trees tend to be sensitive to changes in data and overfit data. We observed a train accuracy of 0.9899 and a test accuracy of 0.7865.

2. **Random Forest**

   We used the random forest classifier in the ensemble class of SKLearn library. Since Random Forest uses averaging to improve accuracy and control over-fitting. We observed a train accuracy of 0.9001 and a test accuracy of 0.7852. In this case, the maximum depth of the tree was 15.

3. **Multi-layer Neural Network**

   We used the MLPClassifier (multi-layer perceptron) class in the neural_network package of SKLearn with a quasi-Newton solver 'lbfgs'. Neural networks use multiple hidden layers to learn a non-linear function. We observed a train accuracy of 0.8411 and a test accuracy of 0.8016. The regularization parameter was set to 1e-5 and 12 hidden layer network.

4. **Adaboost Classifier**

   We used the AdaBoostClassifier class in the ensemble package of SKLearn library. Adaboost applies classifiers to additional copies of the dataset in order to focus more on the difficult classifications. We observed a train accuracy of 0.8340 and a test accuracy of 0.8053. The number of estimators that is the maximum termination condition before which we can find the perfect fit was set to 100.

5. **Gaussian NB Classifier**

   We used the GaussianNB class in the naive_bayes package of SKLearn library. Naive Bayes looks at each feature independently and assigns probabilities to each feature value based on the y value. We observed a train accuracy of 0.6958 and a test accuracy of 0.5130.

MULTICLASS CLASSIFICATION

We tried three different classification models namely –

1. **Support Vector Machines(SVM)**

   We used the SVC class in the svm package of SKLearn library. SVM treats feature vectors as high dimensional points in space, which it tries to separate with a hyperplane in order to create the largest margin between the points and the decision boundary. We observed a train accuracy of 0.6152 and a test accuracy of 0.6135. Furthermore, it used significantly more computation time compared to the other algorithms.

2. **Random Forest**

We used the random forest classifier in the ensemble class of SKLearn library. Since Random Forest uses averaging to improve accuracy and control over-fitting. We observed a train accuracy of 0.65 and a test accuracy of 0.6280. In this case, the maximum depth of the tree was 15.

3. **Multi-layer Neural Network**

We used the MLPClassifier (multi-layer perceptron) class in the neural_network package of SKLearn with a quasi-Newton solver 'lbfgs'. Neural networks use multiple hidden layers to learn a non-linear function. We observed a train accuracy of 0.6270 and a test accuracy of 0.6227. The regularization parameter was set to 1e-5 and 12 hidden layer network.

REGRESSION

For regression, we only used one model that is Linear Regression model. We used the logistic regression method in the linear_model class of SKLearn library. While logistic regression is primarily binary, we used the multiclass variant of the method for predicting the star rating. Logistic Regression assumes that features are roughly linear and the problem is linearly separable. Additionally, it is robust to overfitting and noise. We observed a variance score of 0.14 and sum squared error of 0.31.

Technology & Tools

- Jupyter Notebook for python
- Pandas
- SKLearn
- Matplotlib
- Numpy
- Python Flask
- HTML/ Javascript
- Scipy
- Json
- Csv

COMPARITIVE ANALYSIS

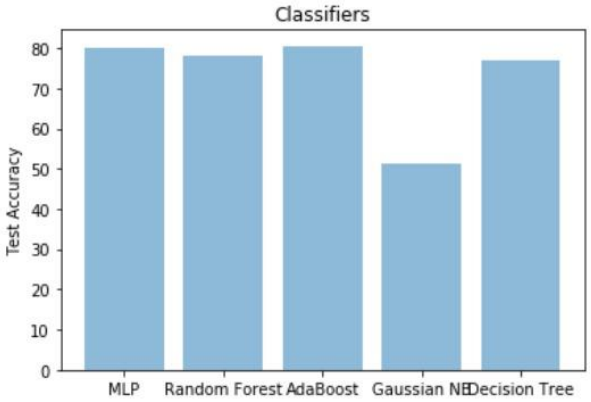
Binary Classification						
	LabelEncoding			Binarization		
	Train_Accuracy	Test_Accuracy	k = number of features	Train_Accuracy	Test_Accuracy	k = number of features
MLP	84.11	80.16	50	83.76	79.08	50
Random Forest	85.99	77.96	50	90.1	78.52	55
AdaBoost	83.4	80.53	55	83.27	79.77	45
Gaussian NB	69.58	51.3	50	71.3	62.45	50
Decision Tree	98.99	76.85	50	97.77	77.01	50

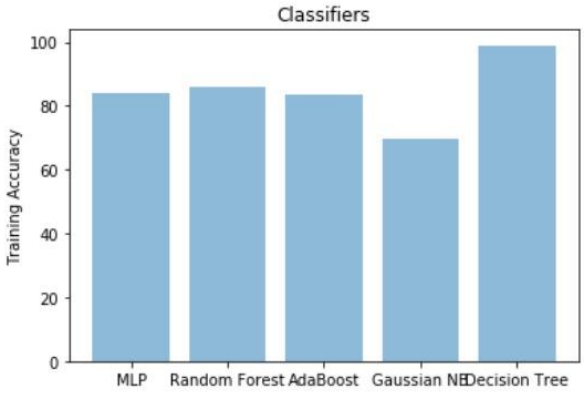
Multiclass Classification												
	Partial Dataset						Full Dataset					
	LabelEncoding			Binarization			LabelEncoding			Binarization		
	Train_Accuracy	Test_Accuracy	k = number of features	Train_Accuracy	Test_Accuracy	k = number of features	Train_Accuracy	Test_Accuracy	k = number of features	Train_Accuracy	Test_Accuracy	k = number of features
MLP	77.3	75	50	76.75	75.01	50	62.7	62.7	50	63.87	62.45	50
Random Forest	75.93	75	50	81.08	74.99	55	65	62.8	50	68.87	62.46	55
SVM	75.6	74.93	55	75.59	74.93	45	61.52	61.35	55	61.58	61.46	45

Linear Regression								
	Partial Dataset				Full Dataset			
	SSE_Train	Variance_Train	SSE_Test	Variance_Test	SSE_Train	Variance_Train	SSE_Test	Variance_Test
LabelEncoding	0.31	0.1	0.32	0.09	0.56	0.08	0.57	0.08
Binarizarion	0.29	0.17	0.3	0.14	0.53	0.14	0.53	0.15

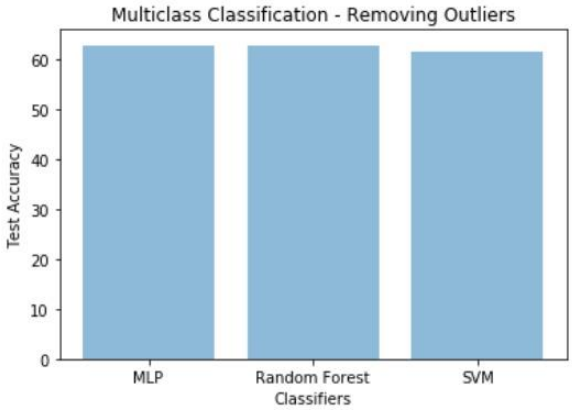
GRAPHS/ STATISTICS

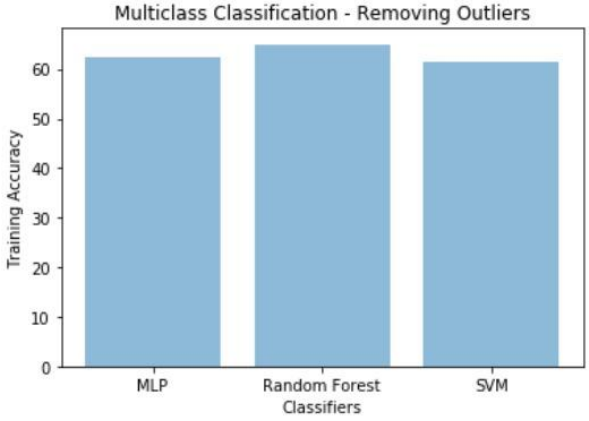
- **Label Encoding for Binary Classification**



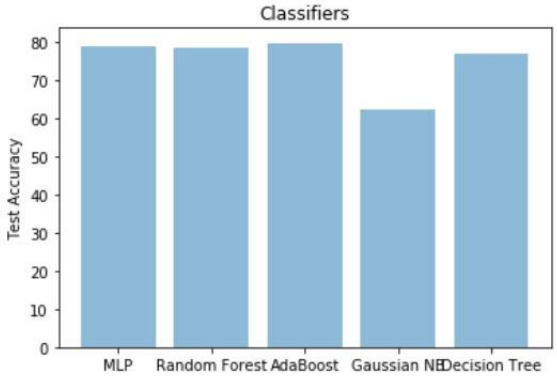


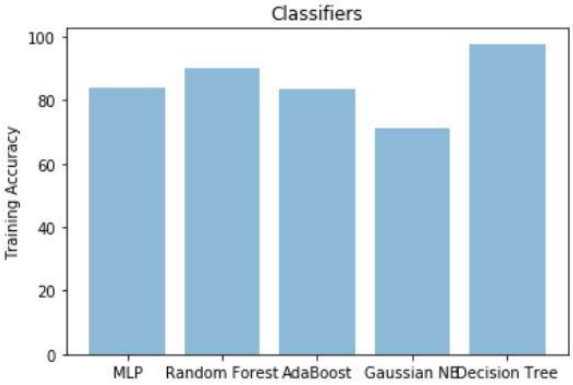
- **Label Encoding for Multiclass Classification**



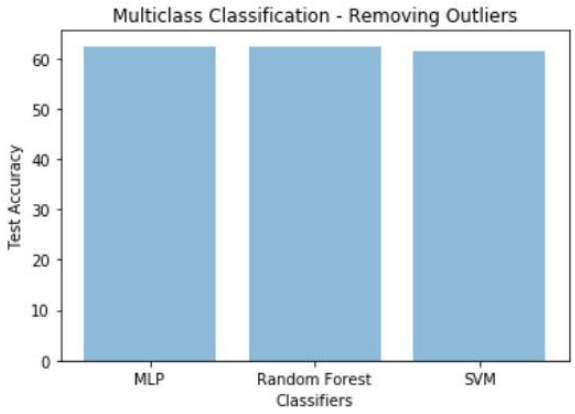


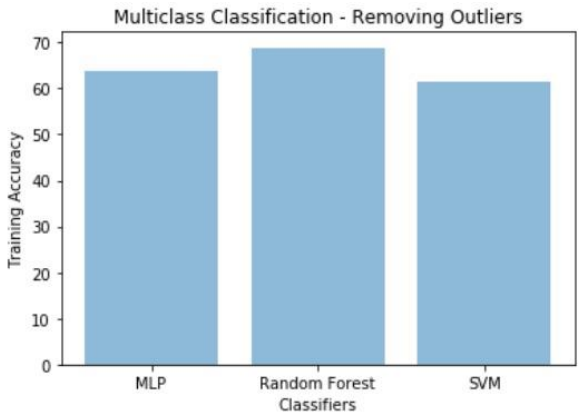
- **Binarization for Binary Classification**



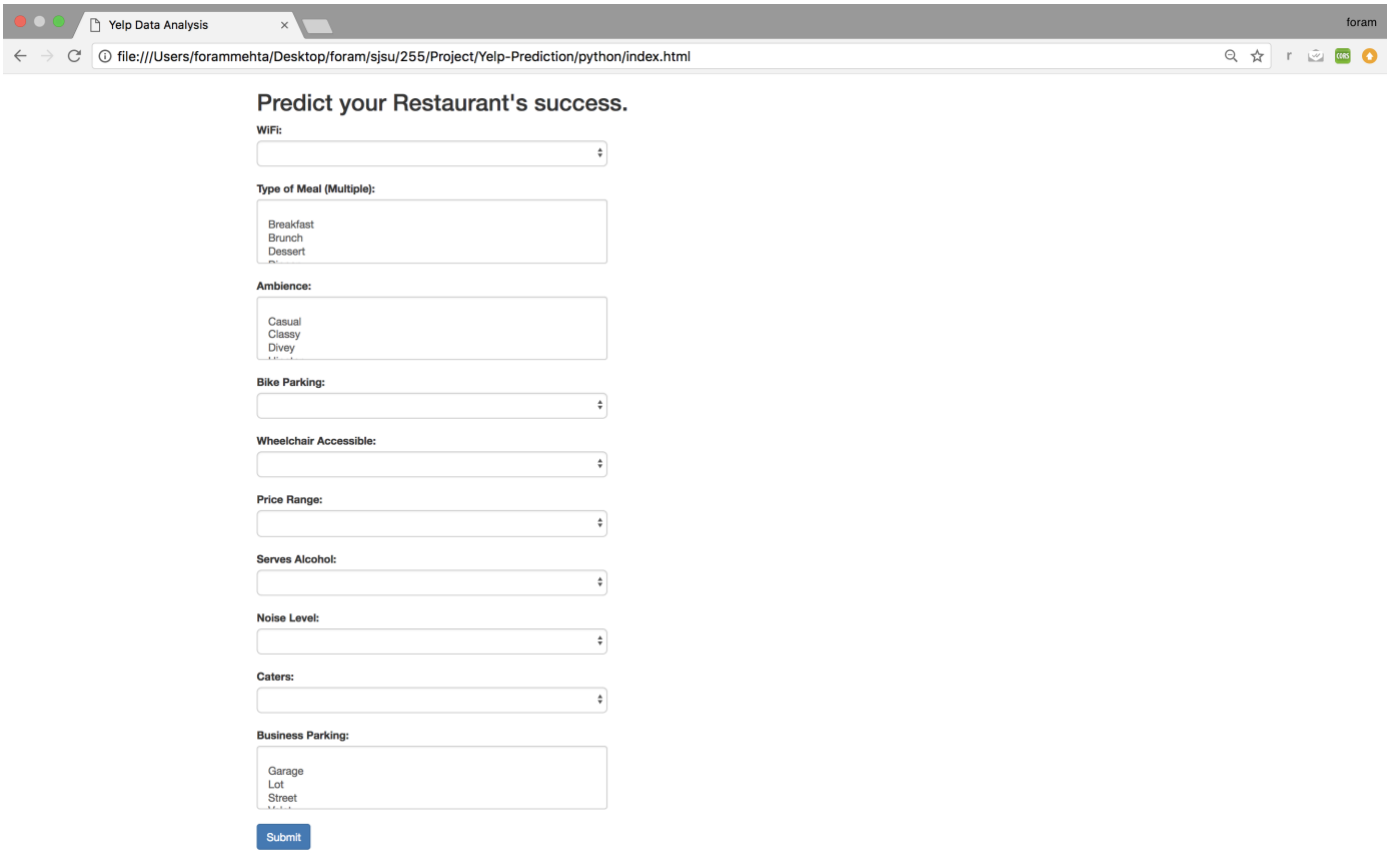


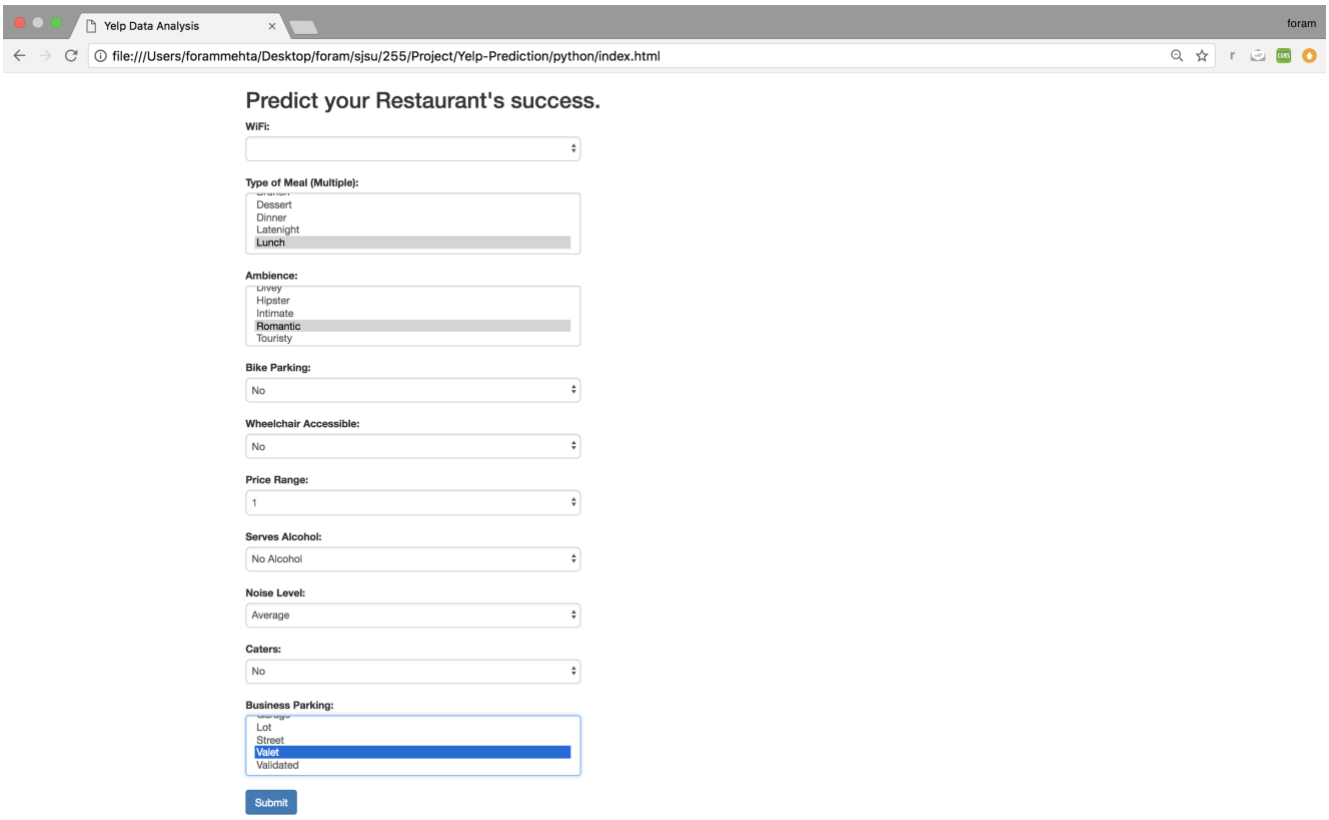
- **Binarization for Multiclass Classification**

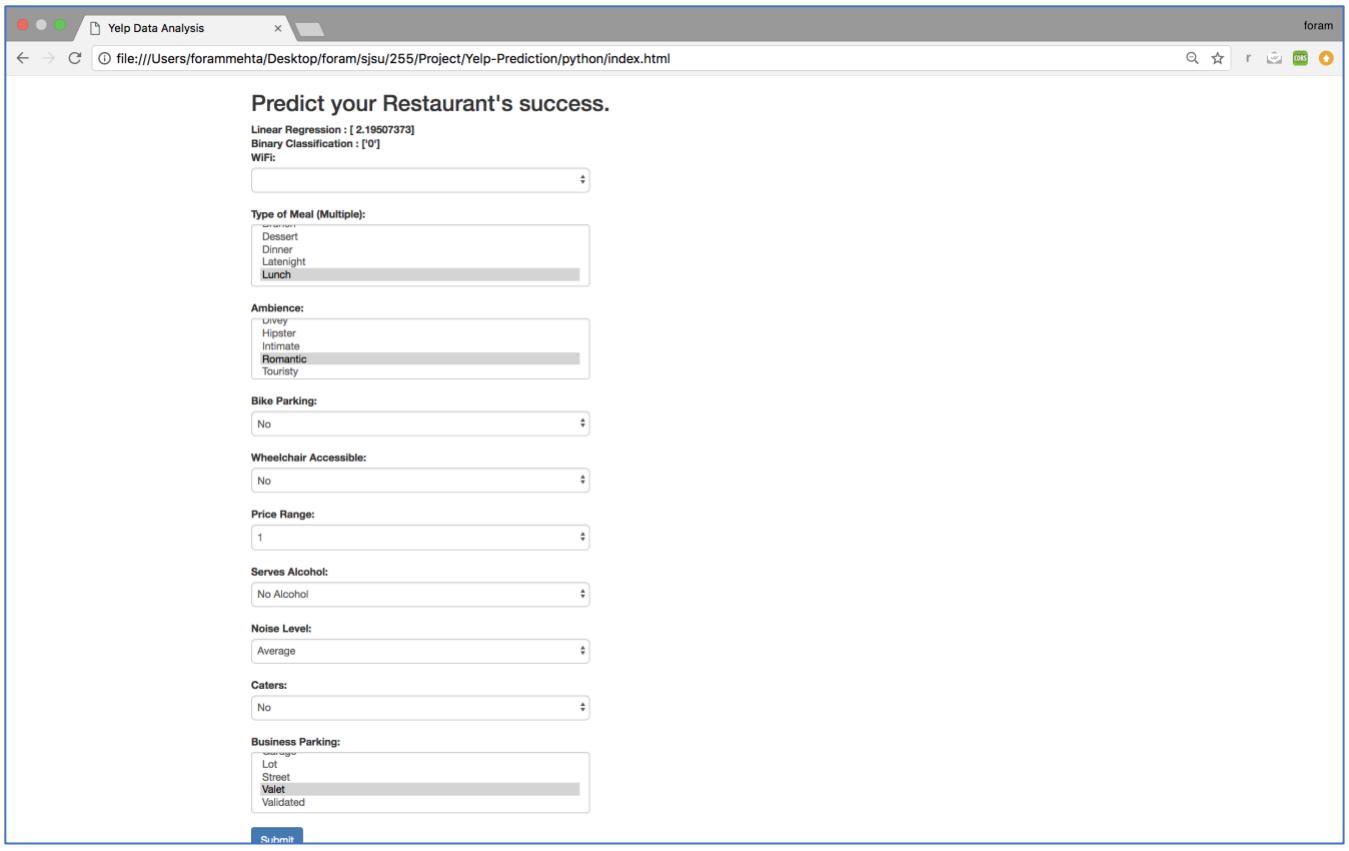




APPLICATION SCREENSHOTS




Yelp Data Analysis
foram
file:///Users/forammehta/Desktop/foram/sjsu/255/Project/Yelp-Prediction/python/index.html
Predict your Restaurant's success.
WiFi:
Type of Meal (Multiple):
Dessert
Dinner
Latenight
Lunch
Ambience:
Hipster
Intimate
Romantic
Touristy
Bike Parking:
No
Wheelchair Accessible:
No
Price Range:
1
Serves Alcohol:
No Alcohol
Noise Level:
Average
Caters:
No
Business Parking:
Lot
Street
Valet
Validated
Submit


Yelp Data Analysis
foram
file:///Users/forammehta/Desktop/foram/sjsu/255/Project/Yelp-Prediction/python/index.html
Predict your Restaurant's success.
Linear Regression : [2.19507373]
Binary Classification : ['0']
WiFi:
Type of Meal (Multiple):
Dessert
Dinner
Latenight
Lunch
Ambience:
Hipster
Intimate
Romantic
Touristy
Bike Parking:
No
Wheelchair Accessible:
No
Price Range:
1
Serves Alcohol:
No Alcohol
Noise Level:
Average
Caters:
No
Business Parking:
Lot
Street
Valet
Validated

DIFFICULTIES FACED :-

- Converting categorical data to numerical data
- Selecting the best features
- Handling missing values.
- Selecting relevant features for processing.
- Deciding on classification models

CONCLUSION :-

We got the best classification results on binary classification models as compared to multiclass classification. The best classification models were Decision Tree Classifier and Random Forests. Label encoding as a preprocessing step played an important role in this classification model.