OR 568 – Applied Predictive Analytics   
Prof. Vadim Sokolov – Spring 2017   
Final project – Data Analysis Project   
Predicting Price tier from Yelp listings in the New York area based on user reviews and business information

Team members:

Saurabh Rao Donthineni   
Gade Venkata Sai Akshay   
Kaushik Kandlakunta  
Shashank Reddy Karnati  
Megha Sainath Reddy

**Problem description**

For the final project, we have worked on data from Yelp. Yelp is a business that publishes crowd-sourced reviews about local businesses. Apart from this, they also work on online reservations and food delivery. We have considered three aspects of a business listing on Yelp for this project – price tier , business information and user reviews. The price tier of a restaurant on Yelp indicates the amount of money a user is likely to spend at said location. The amount of money likely to be spent by a user at a location increases as the price tier increases. User reviews are crowd-sourced comments about the business. The business information about a restaurant is a listing of all the features that it has , like parking, washrooms, delivery etc. For this project, we have tried to predict the price tier of restaurants in the New York area based on the user reviews and business information.

**Data description**

The final dataset on which we experimented with different machine learning algorithms consisted of the following columns:   
**a.** All possible business information stored as Boolean values (1 if the restaurant has a feature and 0 if it does not)

**b.** The basic restaurant information like name, total ratings, price range, category etc.

**c.** We classify reviews into good and bad reviews and get the columns total good review count, total bad review count.

**d.** After topic modelling has been done, we have probability for top 30 topics for good reviews and bad reviews, with each topic having a column of its own.

**e.** List of positive and negative reviews for each restaurant.

**Process**

**Step 1: Web scraping for basic business information**

The aim of this process is to extract the basic restaurant information, business information and user reviews and store them on disk. Firstly, we focus on getting the restaurant information. We use xpath to extract the business name, review count, categories, rating , address , price range and url for each restaurant. We do this by specifying the span class for each attribute and extracting the data from that span class. During this stage, we also clean the data by using the strip function. We save the output to a csv file. During this stage of the data extraction, we have the URL for each restaurant (until now, we were gathering the URLs from a paginated list that lists all restaurants.

**Step 2: User review scraping , labelling and local storage**

We focus on gathering the business information and user reviews for each restaurant. Before we do this, it is important to understand the fact that we should store the reviews for each restaurant separately. To do this, we first create a folder for each restaurant, with the name of the restaurant as the name of the folder (which is obtained from the csv output file mentioned above). Then, using the URL for each restaurant we gathered in the previous step, we locally save the webpages of each restaurant in its folder. Each webpage is saved as a text file.   
After this is done, we have to extract the business information and the user reviews. We use beautifulsoup to generate a HTML tree from which we can extract the data we need. First, we create a parser class function to parse the business information. After this is done, we extract the business information and run the parse function on it. After this is done, we use the find function to find the reviews (from the review class file in the HTML file saved as a text file in the local drive for each restaurant). We then normalize the reviews using NFKD, ignore the ASCII encoding and change the encoding to UTF-8. The problem with ASCII encoding is that is has a lot of characters which cannot be tokenized and it causes problems later. Now, we have to classify each review as good or bad. If the review value is 4 or 5 , it is said to be a good review. If the review is 3 or lesser, it is said to be a bad review. We also get the good review count and the bad review count during this stage. At this stage, we have good review count, bad review count, review count, good reviews, bad reviews.

**Step 3: Topic modelling on good and bad reviews using Latent Dirichlet Allocation algorithm**

Now that we have the good reviews and the bad reviews with us, we work on topic modelling for them. We do this using LDA modelling. It allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. The process that is undertaken is applied to the good reviews and the bad reviews as well. We begin with the assumption that each document that we are considering consists of a mixture of topics. Firstly, we tokenize all the reviews and remove all stop words in English. After this is done, we transform the documents into a count matrix and then get the vocabulary. We now have the feature names with us. After this is done, we initialize the LDA model and fit it to the dataset. After this is done, we get the top 20 terms for every topic and arrange them to get the best terms at the top. Then, we write the top terms as a topic mixture for each document and arrange them to get the best terms at the top. The final output file that we get consists, for each restaurant, 30 good topics and 30 bad topics and the overall probability of each of them. After this is done, we add these topics as features to our dataset.

**Step 4: Data munging**

Before we can run the machine learning algorithms, we have to append the business information for each restaurant as features to our dataset. Since the data for business information is in the form of yes / no / null values , we convert this into Boolean with value for astype(Bool) being true if the string “Yes” is encountered. After this is done, we append those values to the dataset, which was stored as a pandas dataframe until now. Now that we have a dataset that can be used to run machine learning algorithms on, we have to save the dataset to disk so that it can be used for further processing and analysis by machine learning algorithms in the next step. We convert the dataset into a pickle file which is saved to disk. The major reason for doing this is that pickle files allows dataframes to be saved a serializable character frame. This means that this character stream contains all the information to use the object (in our case, a dataframe) in another python script, which we are doing.

**Step 5: Machine Learning**

We load the file into another python script. We are working with 4 models – KNN, Multi class logistic regression, Decision Tree and Multinomial Naïve Bayes models. For each model, we build a parameter grid using GridSearchCV. We then build a gridsearch to find the best parameters. We run the grid and get the mean score for each model. Then, we use the VotingClassifier to predict the best possible model. We get the best model as the Multi class Logistic Regression. We get an overall accuracy of 0.7.

**Step 6: Confusion Matrix**

The overall confusion matrix for the 4 price tiers is as follows:   


**Results**

We understood the best model that we have, which is Multi class Logistic Regression. We have an accuracy of 70%. We also have the confusion matrix for the 4 price tier categories.

**Future Work / Improvements**

* Instead of considering all the business information variables, we can restrict ourselves to variables that are most likely to be available for most of the restaurants. For instance, a restaurant in a very remote area might not have apple pay , but is likely to have parking.
* We can use more models.
* We can run the same process for different cities, especially other metro cities that are likely to have a lot of restaurants with Yelp profiles.

**Team member contribution**

Megha Sainath Reddy : Web scraping , user review scraping and labelling.

Saurabh Rao Donthineni: User review scraping and labelling, LDA.

Shashank Reddy Karnati : Machine Learning and Results

Gade Venkata Sai Akshay: Machine Learning and Results

Kaushik Kandlakunta : Data aggregation and user review scraping