FINAL REPORT:

Yelp Dataset Challenge

Jay Nagle
Rohit Patil
Sameeksha Vaity
Sruthi Vani
Vignesh

Index

1. Introduction

This project introduced and challenged us to work with the yelp dataset. The yelp dataset consists of 5 JSON files and had a total size of 5.24 GB. Each of the 5 JSON files were composed of a single object type and had one respective json-object per line. The 5 JSON files were, yelp_academic_dataset_business.json, yelp_academic_dataset_checkin.json, yelp_academic_dataset_review.json,yelp_academic_dataset_tip.json, yelp_academic_dataset_user.json. We majorly focused on using the business, review and tip object type json files for this project.

Data Statistics:

Given dataset comprises of

Total number of businesses - 85901

Total number of reviews - 2685066

Total number of tips - 648902

The yelp dataset provided to us contains the information pertaining to businesses such as information on its location, city, state, latitude, longitude, stars, review_count, categories which can be very well exploited for our usage.

2. Task 1: Predict Categories of Business

In the first task we consider the user review and predict the business category it belongs to. Since there are many categories for each business, we used a multi label classification approach to predict the multi-labels of the business that a review belongs to.The Machine Learning approach is used in the multi-label classification.The performance of the SVM approach is much better than the MultinomialNB approach we implemented.

For example:

Business: “Awesome and Spicy Noodles” provides the different business services,

the categories of a business it belongs to as given by yelp are: ["Food", "Chinese", "Asian"].

Based on the any given review we can say if that review belongs to ["Food", "Chinese", "Asian"]category.

2.1. Algorithm

Following steps were executed to achieve the objective of improving the category metadata of the businesses:

1. Uploaded the business document, reviews and tips to MongoDB and created index.

2. Then we identified unique categories that are possible for those businesses by reading all categories from business information file.

3. Initially the text is filtered for more realistic training.

- Tokenization
- Removed Stop Words and Performed Stemming.

4. We used POS tagger to find nouns, verbs and adverbs from review and tips .

5. Convert the text into vectors of numerical values suitable for statistical analysis.

6. Achieved this using TfidfVectorizer.

7. Used Scikit Learn library and applied 2 classifiers on the obtained vectored data

- Naive Bayes classifier for multinomial models.
- Multi label SVM.

The Scikit Learn libraries are largely written in Python, with some core algorithms in scython to achieve higher performance.The library SVM we used for the Task 1 is implemented by Cython wrapper around LIBSVM.

2.2. Evaluation

To evaluate the baseline Naive Bayes classifier, we used sklearn SVM and compared both the results. After several executions we found that SVM performs slightly better than Naive Bayes.

2.3. Results

Algorithm	Accuracy	F-score
Multinomial NB	0.59166	0.57241
SVM	0.62047	0.65315

2.4. Future Work

The future work for this task consist of improving the data pre-processing techniques.We can use ARFF files to perform feature selection based on POS, Stemming ,Stop words and noun.We can use MEKA multilabel classifier to categorize the reviews into classes by various algorithmic techniques to each review existence in class. For each business category a class label can be introduced.We can use BR , MULAN and PCC classifier with various sub classifier.Further we can calculate the probability of each class occurring in the training set.

3. Task 2 : Predict 'N' Thriving Businesses for a location

The main aim of this task is to predict the top 'N' businesses around a given user location. From a business perspective, the top N business information predicted will be useful for knowing a popular business/category trend around the region and accordingly help set up a new business which can be prove to be successful.

From a user's perspective, the user can use the predicted results to know about the popular/successful businesses nearby his location and discover new exploration options around the region.

This task is performed by calculating a popularity score for each of the business. This popularity score is being calculated considering factors like rating and review count of the business, the positive and negative intuition of the review and tip text and the useful votes information from the provided yelp dataset.

3.1. Algorithm

The top 'N' business prediction algorithm proposed by us utilizes MongoDB and Stanford CoreNLP library. We have used Java JSON parser package to parse the yelp dataset JSON files. Following are the steps of execution that gives a walk through of the algorithm:

1. Import business, reviews and tips JSON files to MongoDB to create business, review and tips collections respectively
2. Created index on each of the above collections on "*business_id*" field.
3. Parse "*business.json*" to create a new business information file "*geospatial_business.json*" which is compatible with Geospatial index formatting of MongoDB.
4. Create a MongoDB collection named "*businessLocation*" with Geospatial index in MongoDB using latitude and longitude coordinates for each business using the json file created in above step.
5. Find the neighborhood businesses, within a specific radius in miles given a location using the businessLocation collection
    a. Devised a "$*centreSphere*" and "$*geoWithin*" query in MongoDB to retrieve the search results.
    b. Having tried over a range of radius ranging from 0.2 miles to 10 miles, we confirmed on 2 miles radius based on consistent number of businesses retrieved.
6. The retrieved result are further filtered based on the star rating of businesses whose value is 3 and above.
    a. The primary motivation to do this is to reduce the data value points in consideration.
    b. Moreover, it was observed that the predicted results didn't show any major fluctuations when this filter was applied, indicating that the predicted businesses are generally amongst the businesses with star ratings of 3.5 and above.
7. Assign a popularity score to each business.
    a. A detailed description of score value calculation is provided in the subsequent sections.
8. Recommend the top 'N' scored businesses with their corresponding category set.

3.1.1. Data Preprocessing

The review and tip text included in the algorithm contains only the textual data with removal of all the special characters. The data is also preprocessed using Lucene libraries to remove stop-words (Stop Analyzer and WhitespaceTokenizer).

Moving ahead, stemming (Port Stemmer) is performed to normalize the words by indexing on the root word of the similar stem terms. Data preprocessing greatly reduces the resultant data size.

3.1.2. Popularity Score Calculation

According to our algorithm, each of the businesses has a calculated popularity score associated with it. Sorting the businesses in a decreasing order of the popularity score will give us the top 'N' businesses of that region.

The calculated weighted average popularity score is based on:

- Star Rating & Review Count Score
- Review & Tip Score
- Useful Vote count

The resultant popularity score of each business =

(70%)* (Weighted Rating score) +(30%)*(Weighted Review + Tip Score)

So as per the above mentioned formula, the rating score contributes 70% to the popularity score and the Review + Tip score contributes to 30% of the popularity score.

3.1.2.1. Rating Score Calculation

The Rating score is calculated using the Star Rating and the Review count attributes available in the business object of the yelp dataset.

For the correct calculation of the weighted rating score we have to consider the scenario wherein a business A has just two reviews each with a star rating of 5 on a scale of 1-5 and we have second business B which has 196 reviews and the star rating is 4.2. In this case, the business B should be assigned a higher rating score than the first one.

To handle this normalization, we formulate a weighted 2-dimensional matrix of star rating and review count range as shown below:

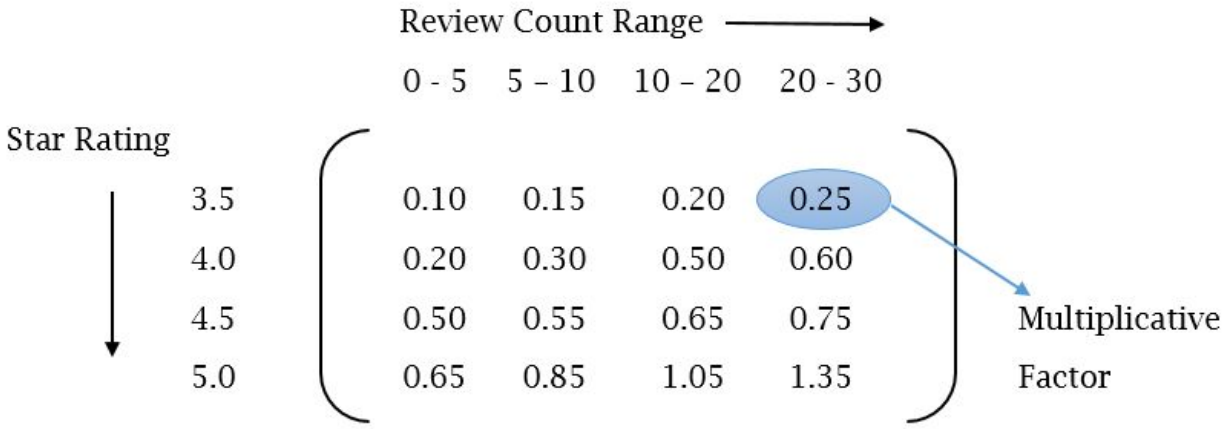

Figure 1 : Rating - Review Count Range Matrix

Following the above matrix we will have a multiplicative factor/weighted factor which when multiplied with the review count of that business will give us the weighted rating score for that business.

3.1.2.2. Review and Tip Score Calculation

Each of the business has a review/ tip information associated with it, which is available from the yelp dataset. Further, each of the reviews and tips are associated a review/ tip score on the basis of their sentiment analysis class prediction polarity and sentiment analysis score.

Steps for calculating the Review/Tip score:

1. Filter the reviews and tips for the specified business_id.
2. Perform sentiment analysis on the preprocessed text data of Review and Tip.
3. Use sentiment analysis predicted class and sentiment score to understand the positivity and negativity of the each of the statement for each review.
4. Normalize the scores of each statement for the review.
5. Use this intermediate review score and average each of these reviews to get an normalized average review score for each of the business's.
6. Similarly, calculate the Tip score by repeating the above mentioned steps for Tip text.

7. Calculate the final Review and Tip score using the following formula:

   = (40%) * (Review Score) + (60%) * (Tip Score)

   The above formula assigns 60% weightage to Tip Score and 40% to Review Score. This is because after reviewing some actual samples of yelp dataset we observed that the Tip's text content length was usually observed to be relatively less than review text and contained few keywords that could be used as great targets to find useful information than in a huge volume of contextual data.

   Consider the following example,

   Review Text :

   "Quick and affordable! Whenever I host a party for my teenagers, this conveniently located pizza joint takes care of our needs. They always have pepperoni pizza in stock for $5. Amazing. They also offer cheese pizza. My favorite is the 3-Meat Pizza. They also give free cookies. They give out hot pepper flakes, but no Parmesan cheese. Our 2 pizzas rang up to $14 and change."

   Corresponding business's Tip Text:

   "Great service and the pizza was DELICIOUS!!!"

   So as we can see from the above example the Tip's text is concise but contains relevant and useful information and hence must be given a higher weightage as compared to Review's textual information.

3.2. Experimentation/Enhancements

After having performed the above mentioned baseline implementations for the predicting the top 'N' business's in a location we proposed, implemented few further experimentations to measure the deviation of the performance of the algorithm.

3.2.1. Review-Useful Votes attribute

To enhance the performance of our algorithm we thought of further validating the features used in the algorithm. After a careful observation of data it seemed reasonable to consider the useful votes count information present in the review object of the available yelp data set. So, to include

only only useful data in our algorithm we considered only those reviews for whom the useful votes count is greater than 5.

We additionally also tried experimenting by addition and combinations of the different features like Review & Tip + Rating score, or just using the rating score that were fed to our algorithm and observed the varied results which are further explained in the results subsection.

3.2.2. Sentiment analysis

As we have mentioned above we use sentiment analysis on the textual data of Review's and Tip's text of each business. We perform sentiment analysis to understand the opinion expressed by the text, to understand if the text is expressing a positive or a negative comment. The sentiment analysis nlp library has predefined set of class labels like *"very positive", "positive", "negative", "very negative"* and also calculates a sentiment score for each of the sentences. We use this sentiment score in our algorithm to get polarity degree associated with each of the sentences. The predicted classes assign values on a 5-point scale of 0 = very negative, 1 = negative, 2 = neutral, 3 = positive, and 4 = very positive.

Consider the following example,

Review Text :

"Quick and affordable! Whenever I host a party for my teenagers, this conveniently located pizza joint takes care of our needs. They always have pepperoni pizza in stock for $5. Amazing. They also offer cheese pizza. My favorite is the 3-Meat Pizza. They also give free cookies. They give out hot pepper flakes, but no Parmesan cheese. Our 2 pizzas rang up to $14 and change"

We get the predicted class as = **"very positive"**

And the normalized Review score = **3.8**

This kind of information helps us to efficiently distinguish between the different classes of sentiments.

3.3. Evaluation

3.3.1. Accuracy

Accuracy is a measure defined as the ratio of correctly classified points to the total number of data points. It is generally used when the class is balanced. In our case accuracy is going to be correctly predicted top 'N'' business divided by the sum of total 'N' business given a particular location.

Accuracy = Correctly predicted Top N business/total number 'N' business at the location

3.3.2. F1-score

It is defined as the harmonic mean of precision and recall. It can be represented as follows:

F = 2*(Precision.Recall)/(Precision+Recall)

Precision: Precision is generally defined as the ratio of True Positive to the sum of True Positive and False Positive.

Precision = TP/(TP+FP)

Recall: Recall is generally defined as the ratio of True Positive to the sum of True Positive and False Negative.

Recall = TP/(TP+FN)

True Positive (TP): It is the number of actual top 'N' business which matches the predicted top 'N' business at a given location.

False Positive (FP): It is the number of business incorrectly predicted in the list of top 'N' business in a given location, which does not match the actual list of top 'N' business at the location.

False Negative(FN): It is the number of business categories which are actually in list of top at a particular location, but not predicted in the top 'N' categories.

With a good balanced data like the problem in hand, accuracy is a fair measure to have. We have the F-1 score as well to add further validation to the predictive model. This helps us understanding the model developed more which accuracy alone cannot reveal.

3.4. Results:

Features	Accuracy	F-Score
Baseline ratings score (using review count)	0.56023	0.64002
(Review + Tip)	0.58110	0.63315
Review + Tip + Star rating	0.61504	0.60877
Weighted(review + Tip) + Star rating - Based on useful votes.	0.59711	0.65531

The results were tuned using a wrapper method of feature selection, forward selection. Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model [5].

The subset of features was divided into four categories as explained in the algorithm:

1. Baseline Star Rating Score (Using Review Count):
2. Review + Tip
3. Review + Tip + Star Rating
4. Weighted(review + Tip) + Star rating

As anticipated, the accuracy of the model was the lowest in the first case, where only the Baseline rating scores contributed to the feature set. The low accuracy

can be to two reasons, low correlation of the Baseline Star Rating Score with the predicted class label, and underfitting in the model viz. The true dynamics in the data was not learned by the model completely. The best performance was obtained on using the Review + Tip + Star Rating as the feature set. The increase in accuracy of the model can be attributed to the increase in feature vector, which explains a positive correlation of the added features with the class label. A nominal decrease in accuracy was observed on using the weighted version of the high performing feature set. A comparative increase in F -1 score is because it ignores the True Negatives in the prediction of the model. Considering the business or the user's perspective, either of them would not be interested in the business categories which were correctly not predicted in the top 'N' thriving business. This explains the importance of analysing the performance of the model using F -1 score , for the problem in hand.

3.5. Future Work

The future work for this data set and our approach on this could be better utilized if we also include the user's information into consideration while returning the top 'N' business's near this region. The user's information can be used to return the user the top "n' business's near this region depending upon the user's past experiences and learning user's history and taste.

Another proposal for future work would be to identify the individual most influencing factors or the factors contributing the most to a particular type of business. This can be done by using Association Rule mining on the business information. This resultant information can be used by the new businesses to identify the attributes that can potentially contribute to the success of the business.

4. Conclusion

A machine learning approach was devised to predict business categories from reviews and tips data. Applied Scikit Learn library classifiers Multinomial Naive Bayes and SVM on the vectored data obtained by POS tagging and TfidfVectorizer. The results show a clear increase in the accuracy when the business categories were predicted by SVM classifier.

Predicting top 'N' thriving business will be helpful to those who want to establish a new business at a particular location and also to the users who wants to know popular businesses at a given location. Based on the above Task 2 results and our experimentation efforts it showed progressive increase in the accuracy of predicting the top 'N' business's

when we used the added information of useful votes for considerate selections of reviews and tips and sentiment analysis for analyzing the reviews and tips text.

5. References:

1. http://nlp.stanford.edu/IRbook/pdf/05comp.pdf.
2. https://docs.mongodb.com/v3.2/applications/geospatial-indexes/
3. http://stanfordnlp.github.io/CoreNLP/
4. https://www.tutorialspoint.com/lucene/lucene_indexing_process.html
5. goo.gl/aznGd3

6. Team Member's Work Distribution

Team Member	Responsible for
Jay Nagle	Task1-Algorithm Design, MongoDB, Multinomial Naive Bayes, Multi label SVM implementation.
Rohit Patil	Task2-Algorithm Design, Data preprocessing, JSON parser code implementation, MongoDB, geo-spatial index creation and Filtering logic for relevant businesses, Rating Score normalization.
Sameeksha Vaity	Task2-Algorithm, MongoDB, code for sentiment analysis, Evaluation.
Sruthi Vani	Task 1-Initial Data Analysis and Data preprocessing
Vignesh	Task 2- Evaluation, Prediction Analysis and Interpretation