Yelp Dataset Analysis - Restaurant Cuisine Classification and Rating Prediction

Dhivya Swaminathan, Shilpa Singh, Yuhan Zeng

I. TEAM MEMBER CONTRIBUTION

There are various parts to this project of Yelp dataset analysis and they are listed below:

1) Data Cleaning and Final dataset generation for the two different approaches for Task1 .
2) Ground truth generation for Task1-Approach1
3) Task1 - Approach1 - Multi class classification
4) Ground truth generation for Task1-Approach2
5) Task1 - Approach2 - Multi label classification
6) Data Cleaning and Final dataset generation for Task2
7) Task2 - Restaurant Rating Prediction

Data cleaning and final dataset generation that was used for both approaches for Task1 was carried out by Dhivya. This includes collating data for all businesses, filtering out restaurants and generating ground truth file for multi class classification. The machine learning approach to Multi Class Classification was also carried out by Dhivya Swaminathan. This also includes building the final feature set by using a TF-IDF vectorizer and selecting top k words using chi-squared values, building a Naive Bayes model and carrying out analysis for various k values to determine the best k.

Task1- Data cleaning and preparation for Approach 2 Multi Label Classification and the classification was carried out by Yuhan Zeng. This includes filtering restaurants by a given set of categories and assign multiple labels that belong to the categories set to each restaurant. An One-Vs-Rest-Classifier was utilized for multi-label classification, and the text was converted into a count vector, which the labels were transformed into binary form by a multi-label binarizer so that binary classifiers can be trained on them, including an SVC classifier, a Naive Bayes classifier and a random forest classifier.

Task2 - Shilpa Singh : Loading the yelp data into HDFS. Feature Extraction and Transformation of Review text and creating different models for Linear Regression. Integrating Stanford-core nlp libraries with Spark.Tuning Spark configurations to run LDA faster.Experiments related to LDA model convergence and hyper-parameter tuning.Running Linear Regression and getting the final results.

II. TASK 1: PREDICTING CATEGORY LABEL FROM REVIEW AND TIP TEXT

A. INTRODUCTION

The Yelp dataset includes business, reviews and tips files that have information about various businesses, multiple reviews and tips for the same along with other demographic information. Analysis of the patterns in data is set to solve various interesting use cases. In this project, we have taken up Restaurant Cuisine Classification and Restaurant Rating prediction. For Task 1, two approaches have been carried out - Multi Class and Multi Label Classification. Our first classification approach was to determine a single best category for a business using multi-class classification.

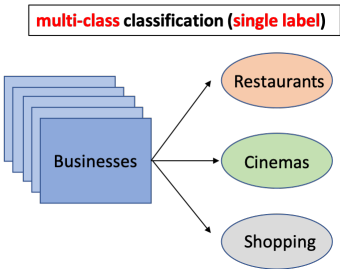

Fig. 1. Multi-Class Classification

However, a business on yelp usually belongs to multiple categories. Therefore, each business can be assigned multiple categories, which is our second approach, multi-label classification. Multi-label classification is a variant of the conventional classificationproblem where multiple labels may be assigned to each instance. For instance, a business might belong to Restaurants and Bars at the same time, and we want to assign as many correct categories to them as possible. Here we utilized a OneVsRest strategy to have one classifier for each label. After converting the multiple labels into a binarized matrix, conventional binary classifiers can be applied to classify each label. Here we used an SVC model, a multinomial Naive Bayes model, and a random forest model.

B. DATA PRE-PROCESSING

As part of the Data pre-processing, the steps carried out to clean and obtain the final workable dataset is as shown in Figure 3.[1]

1) APPROACH1 - MULTI CLASS CLASSIFICATION: For Approach 1, the task taken up was to classify all the restaurants into one of the following cuisines: Indian, Mexican, Chinese, Italian, Asian Fusion, American, Japanese,

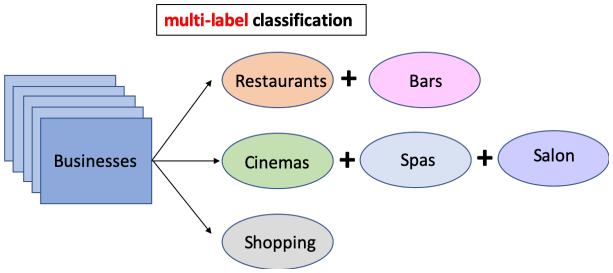

Fig. 2. Multi-Label Classification

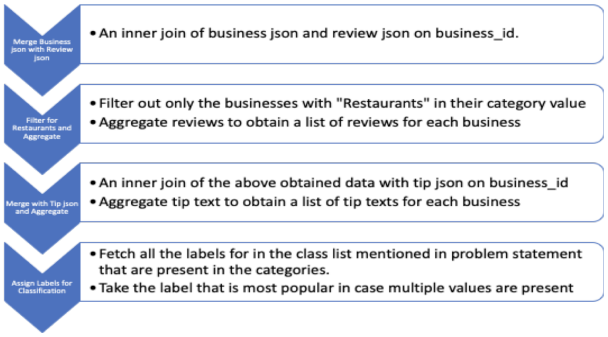

Fig. 3. Data Cleaning and Final dataset Generation

Thai, Seafood, Mediterranean, French. These categories (cuisines) were finalized after manually looking at the frequency chart of multiple cuisines available as part of the dataset.

The reviews and tips of all restaurants were collated and brought to the format of business - list of reviews - list of tips, along with other demographic information. The nulls were removed from the dataset and this was taken forward as a final dataset.

2) APPROACH2 - MULTI LABEL CLASSIFICATION: For Approach 2, the dataset was first filtered by only restaurants in Phoenix, Arizona. Then, we further filtered the dataset by whether their categories contain one or more of these labels: Chinese, Fast Food, Italian, Pizza and Sandwiches, and manually assigned one or multiple labels to each restaurant if their original categories contains one or more labels that belong to the above ten labels. Reviews and tips were collated and joined with the restaurants by the business id. Nulls were removed from the dataset.

C. METHODOLOGY

1) MULTI CLASS CLASSIFICATION - MACHINE LEARNING APPROACH: For Multi class classification, a machine learning approach was taken. The ground truth values were generated by picking up the most frequent cuisine present in categories field. Now, post this, a TF-IDF vectorizer was used to give a score for each word in the review and tip data separately. Using Chi-Square values, top k words were taken as part of the final dataset. An 80-20 split was done and a train-test data was obtained. A Naive Bayes model was built and the same analysis was carried out for top k words for k=100 to 700.

2) MULTI LABEL CLASSIFICATION - MACHINE LEARNING APPROACH: For multi-label classification, a OneVsRest strategy was utilized. The OneVsRest approach converts multiple labels into a binarized matrix, and thus a multi-label classification problem can be decomposed into several single-label classification problems. After the labels were binarized by a MultiLabel Binarizer, we used three classifying models: an SVC model, a multinomial Naive Bayes model, and a random forest model, to classify each restaurant based on the review or tip text. A 70:30 train-test split was done on the dataset.

D. RESULTS

1) MULTI CLASS CLASSIFICATION: Post analysis, it was found that at k=700, i.e., the top 700 words, when taken as part of the dataset, gave the vest results. It was also found that beyond a value of 700 for k, the accuracy was increasing, along with it, the model complexity was also increasing. Therefore, the analysis was stopped at k=700 and a k=500 words was chosen as optimal, balancing both complexity and accuracy.

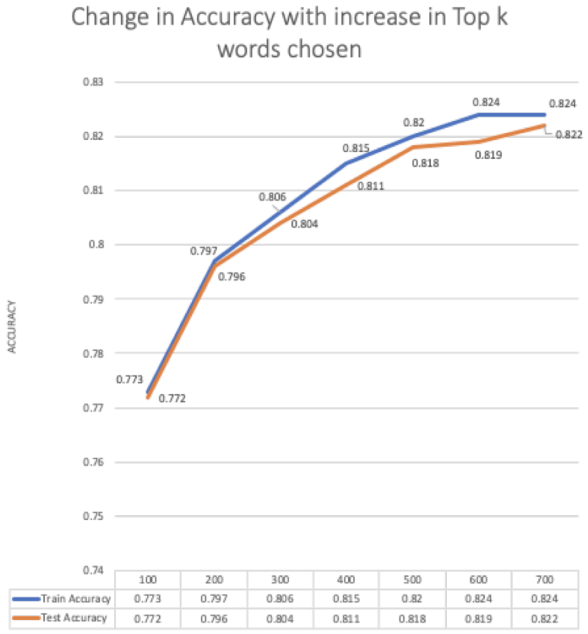

Fig. 4. Top K words Analysis Results

2) MULTI LABEL CLASSIFICATION: The metrics for evaluating the multi-label classification results were accuracy and precision as calculated below.

Where n is the total number of records in the test data, Y_i stands for the predicted label vector for each instance i, and Z_i stands for the ground truth label vector.

$$Accuracy, A = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$

$$Precision, P = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|}$$

Tips	SVC	Naive Bayes	Random Forest
Accuracy	8.6%	64.3%	48.1%
Precision	9.0%	70.2%	49.8%

Table 1: Accuracy and Precision based on tips

Reviews	SVC	Naive Bayes	Random Forest
Accuracy	58.0%	78.0%	67.4%
Precision	59.0%	89.9%	68.8%

Table 2: Accuracy and Precision based on reviews

Naive Bayes showed the best performance among the three classifiers. The problem is that the data sets we used in the two approaches for task 1 are not consistent. We attempted to use the same data set for the second approach - multi-label classification, but because of the large size of the data set, it needed a huge memory as well as a enormous amount of time, so that we have not been able to get a result by the time of writing the report.

E. CONCLUSION

As part of task 1 multi class classification, it was determined that as more and more top words were included, the accuracies were improving, given the increase was not considerable, the complexity of the model was increasing manifold. Therefore, considering a k of 500, i.e., top 500 words was determined to be the ideal k. For multi-label classification, as the classification task is to assign multiple labels to each business, the accuracy and precision is slightly lower than those of the multi-class classification. The difficulty increases in predicting more than one labels compared to predicting a single label.

III. TASK 2: PREDICTING RATING FROM REVIEW

A. Problem Statement

Almost everyone prefer to go to restaurants which have higher rating.However,we know that not all user ratings are objective all the time.It is quite feasible that a very positive review may have a five-star rating while a similar review ends up with a three-start rating.Rating standards of individual vary.This may affect the business adversely.The underlying goal of this task is to attenuate the effect of subjective reviews by learning rating from a large number of examples.

B. Proposed Solution

A simple approach of review rating prediction might only consider the sentiment of words in the review text.However, topics in reviews are also likely to play an important role.For example, a negative sentiment associated with the topic of hygiene may be more influential than a negative sentiment associated with the topic of price and may result in lower rating.In our approach, we investigate whether topic model benefits rating prediction task.We also investigate Tf-Idf and its interaction with sentiment score.In the proposed solution, we have used 5 techniques to generate the features for rating prediction task.

1) Sentiment Score of words.
2) Topic Distribution of the words.
3) Topic Distribution of the words in the combined with the sentiment score.
4) Tf-Idf of the words.
5) Tf-Idf of the words combined with their sentiment score.

We have also divided the above categories further into nouns, adjectives and nouns + adjectives and examine their effect separately w.r.t the type of feature.

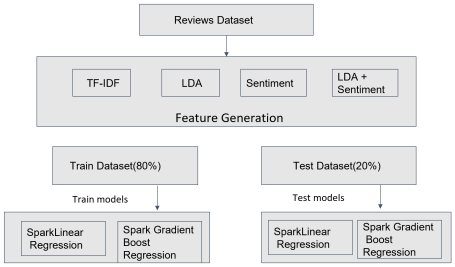

Fig. 5. Technical Approach

C. Dataset

We have used a subset of Reviews data for which the Business fall under Restaurant category which consists of 1347779 reviews belonging to 59371 Restaurants.

D. Distributed approach through Spark ML lib implementation

We used Spark as our Distributed Computing platform and Spark MLlib for the algorithm implementation. This provides a simple interface that abstract the complicated aspects of MapReduce programming and all distributed computations are done on RDDs (Resilient Distributed Datasets), which are linear data structures distributed across the nodes in the cluster and utilizes RAM of these machines. When data is kept in memory, disk serialization/deserialization is greatly reduced. This allows us to distribute iterative algorithms like LDA much faster because we don't have to wait for the data to write to disk after each iteration.

E. Implementation Overview

We have implemented text pre-processing, model creation and evaluation through Spark MLlib and feature transformation Apis. The feature transformation Apis are used to perform basic NLP processing like tokenzation,removing Stopwords, countvectorization and IDFModel creation.We also used Stanford-core nlp for POS tagging and filtering only nouns,adjectives and adjectives + nouns as well as getting sentiment score of the words.

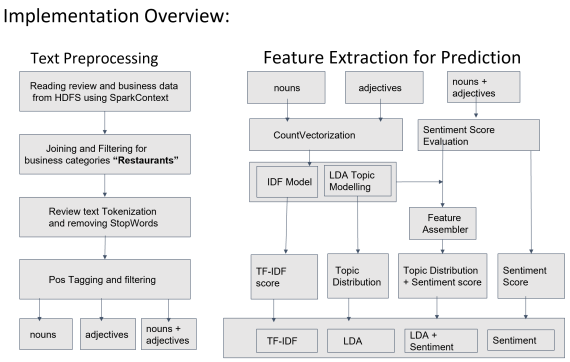

Fig. 6. Implementation Overview

F. LDA Convergence and Hyper-parameter Tuning

We used the RDD-based LDA algorithm developed in Spark. It takes as input a collection of documents as vectors of word counts and the following parameters:

- k: Number of topics
- optimizer: Online or EM(Expectation Maximization approach where the parameter learning uses constant time and memory).We have used EM for our approach because it converges faster.
- docConcentration: Dirichlet parameter for prior over documents' distributions over topics (α).
- topicConcentration: Dirichlet parameter for prior over topics' distributions over words (η).
- maxIterations: Limit on the number of iterations.

1) Convergence: First, we want to ensure that the LDA algorithm is converging. EM only supported 100 iterations, after that it gives an overflow error. We can observe (see figure 5) that the EM algorithm is converging after 40 iterations.

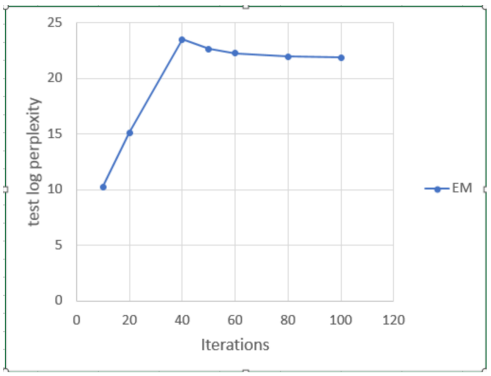

Fig. 7. convergence

2) Tuning hyperparameters: We want to get the hyper-parameters (α and η) that give the best test perplexity for EM algorithm. The experiments consist in holding fixed one parameter (either α or η), the number of topics k and the number of iterations. Then we varied the free parameter to observe when the test perplexity converges. This happens with EM for α=3.5 and η=1.5 (see fig. 6).We also varied the number of topics (k), in fig. 7 and we can see that the test perplexity increases as the number of topics increase, so we can say that the data is better represented with a low number of topics.We have taken k=20 for our experiments.

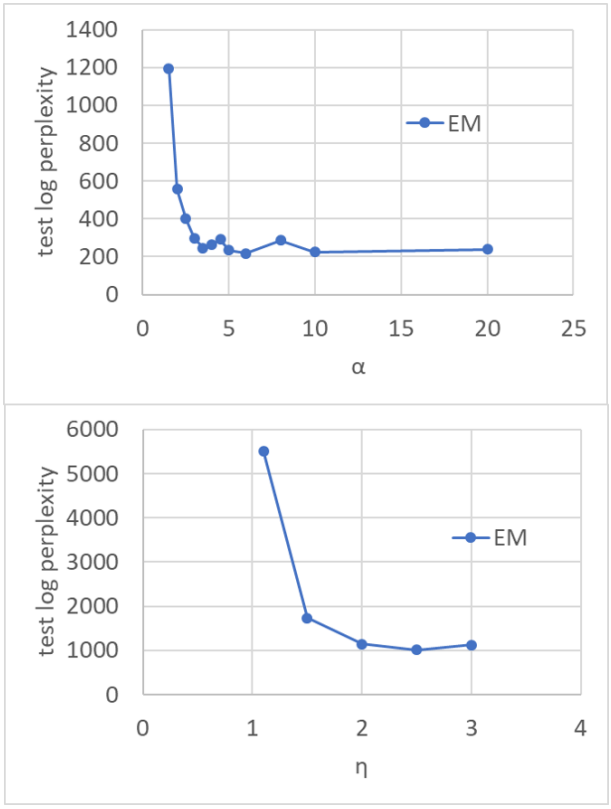

Fig. 8. parametersEM

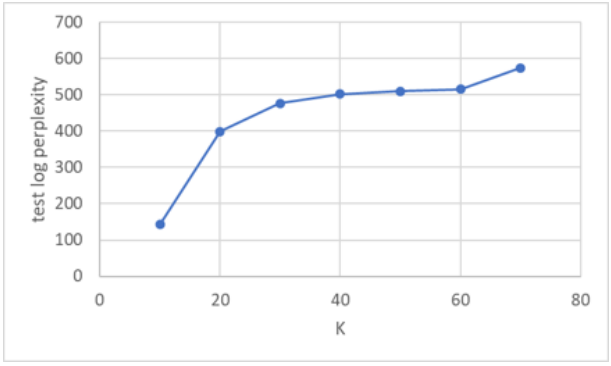

Fig. 9. kTuning

G. Results

Since our goal is to predict star rating from reviews of users,we have used RMSE and MAE as our evaluation metric and linear regression as the prediction model.We have given

the features for nouns, adjectives and nouns + adjectives to Spark Linear Regression and compared the values for them.

We got different values of RMSE and MAE for these models.

$$RMSE = \sqrt{\frac{\sum_{j=1}^{n}(y - y_j)^2}{n}}$$

Fig. 10. RMSE Formula

Model	MAE	RMSE
LDA	1.085	1.303
TF-IDF	1.145	1.363
LDA+Sentiment	0.862	1.021
Sentiment	1.152	1.371

Table1:nouns

Model	MAE	RMSE
LDA	1.139	1.357
TF-IDF	1.122	1.336
LDA+Sentiment	0.944	1.116
Sentiment	1.084	1.299

Table1:adjectives

Model	MAE	RMSE
LDA	0.980	1.187
TF-IDF	1.119	1.332
LDA+Sentiment	1.032	1.210
Sentiment	1.121	1.340

Table1:nouns+adjectives

H. Conclusion

The question we were trying to investigate was whether topic models,in addition to sentiments,benefit rating prediction or not and the 4 models are an answer to this.As we can see from these tables that all models that involve LDA contribute better to rate prediction for nouns and nouns+adjectives than sentiment.We can also see that for nouns and nouns+adjectives even TF-IDF as features do better in rating prediction than sentiment.LDA seems to have better results than TF-IDF for nouns and nouns+ adjectives.

REFERENCES

[1] *https://www.yelp.com/dataset/documentation/main*