Big Data Final Project

Yelp Data Analysis

Group Members:
Soham Mahabaleshwarkar, William Bresee, Rutwik Chinchole, Ronak Fofaliya

Abstract:

The goal of this project was to develop a model that could accurately predict the sentiment of a review from the text and attributes associated with that review, while employing some of the technologies and tools that we've studied in class. We selected the Yelp dataset to analyze restaurant reviews and help existing business owners. We applied various Pyspark ML technologies to extract features from this dataset, and then tried to predict review sentiment from those features.

Introduction:

Yelp reviews comprise a massive number of customer reviews, in which customers have rated businesses. The dataset we worked with is a subset of the Yelp database which was made available by Kaggle. Each review comprises a few lines of text describing an experience with a business, and reviews often include some additional business attributes, as well as a star rating.

Data Introduction:

Data Source: Kaggle (link: https://www.kaggle.com/yelp-dataset/yelp-dataset) This dataset is a subset of Yelp's businesses, reviews, and user data. It was originally put together for the Yelp Dataset Challenge, a chance for students to conduct research or analysis on Yelp's data and share discoveries.

Inspiration:

We were inspired by the work of Nicha Ruchirawat, Tina Peng, and Maise Ly (Medium Article), who conducted a similar experiment. We handled the dataset differently by eliminating the use of MongoDB; instead, we converted data jsons to CSV format using R, experimented with storing them in S3, and then moved them from S3 into PySpark. We also performed experiments on larger subsets of the data, while the notebooks that inspired our work focused on smaller subsets of the dataset, limited by geographical location.

Reading the Data:

Our goal was to read JSON files and write out as CSV files. We decided to use R to convert JSON to a CSV file rather than use database capabilities through MongoDB or MySQL. R provides an efficient package j*sonlite* to transfer file format with only one line. Although the total size of downloaded data files are 4 GB, a little large to run in R, we transferred each data file separately and the R code takes no more than 20 minutes for the largest file in size of 2.9 GB. Once all necessary files had been converted to CSV format, we stored the files in S3, and then imported them into PySpark.

Data Analysis:

We had to format data, extract features, develop Logistic Regression, Support Vector Machine, and Decision Tree classification models, and use these models to analyze and predict the general sentiment of the reviews. For this task we decided to use PySpark, due to our familiarity with the tool, and due to its scalable nature and robust machine learning libraries.

Data Engineering: (Cleaning and Preprocessing)

Step 1: First we converted the JSON file to a CSV file using our jsonlite package.

Step 2: Next we will upload the CSV files in our AWS S3 bucket.

Step 3: We will load our data from the S3 bucket.

Step 4: We will drop our NULL values in the dataset.

Descriptive Analysis:

At the outset of the project, we were required to choose ways to represent non-numerical data as numerical or boolean data. Exploratory data analysis often helps us decide which questions to ask, and what to look for in the data. In deciding how to represent our data, and how to develop a set of useful features from the data, we tried exploring and representing our data in a number of different ways.

In the Business dataset, we looked for correlations between different attributes, as well as between certain attributes and the number of stars. In both seeking correlations and trying to predict sentiment, we had better success with converting star ratings to a binary negative/positive sentiment label. We included a number of graphs in the notebook which provide insight into the distribution of stars in this dataset, as well as correlations between attributes and stars.

In the Review dataset our goal was to analyze the compliments that each restaurant received after each user visit. The reviews were given attributes such as “useful”, “cool” and “funny” categorically and classified on that basis. We noticed that these compliments showed a one-to-one relationship, there are less reviews when the compliments increase. More than half of the reviews had neither of the three compliments. Another trend was that users mark a review more often as useful/funny.

Sentiment Analysis:

<u>Logistic Regression:</u>

Our primary goal was to predict ratings for businesses and restaurants on Yelp through analyzing reasonable attributes in data. We considered the distribution of ratings from 1 star to 5 stars, and set parameters. We define 3 stars or below as a low rating and set the value to 0, and for ratings above 3 stars as high ratings and set them as 1. Thus we get a boolean column “stars” for predictions with a value or 1 or 0.
We have to select particular attributes as independent variables to consider in our Logistic prediction. Some of the variables we picked are, Business ID, User ID, Rating Stars in review dataset, business location etc
Model:

1. Drop our NULL values in the dataset and the stars column.
2. Convert the prediction variable to a boolean column. Converting ratings below 3 to 0 and above 3 to 1.
3. Build a logistic regression model.
4. Create a feature vector by combining all our features together.
5. Split the data into a training and testing dataset.
6. Train and test the model with the split data.
7. Check model predictions, results, accuracy and iterate.

Results: Our model ended up with an accuracy of **82.6%**.

<u>Support Vector Machines:</u>

Target variable stars is relabelled so that any reviews with 4 stars or above are assigned to 1 ie. positive while anything else is deemed to be 0 which means a negative review.
We first processed the text and then tokenized it. We removed punctuation, stop words from our text to reduce the clutter and feed important data only.

We use PySpark's ML library for these functionalities.

We then created trigrams with frequency >10 using MapReduce functionality to identify them. The original text is replaced by trigrams and then tokenized.

Model:

1. Split the data into training and testing dataset.
2. Set the hyper-parameters in the SVM models, regParam = 0.4, numIterations = 125.
3. Evaluation of model: F1 score - a weighted average of precision and recall.

Result: Our model ends up with a result of **88.632%**.

Feature Extraction: Business

Model:

1. We first converted the data from JSON to CSV format to import into PySpark.
2. We then cleaned and filtered the dataset. This step largely consisted of filtering out all non-restaurant data for a more cohesive dataset, as well as filtering out all information associated with a missing stars rating.
3. For all columns which contained only boolean values, we converted False to 0.0, True to 1.0 and missing data to 0.5.
4. The most difficult step of this component of the project involved converting non-boolean columns to numeric values. To do so, we first parsed all non-boolean columns for keywords and converted each distinct keyword, with a few exceptions, into its own column. For all keywords associated with a positive boolean value, or present without an associated boolean value, we filled the corresponding column with a 1.0. For all keywords associated with a negative boolean value, we filled the corresponding column with a 0.0. We filled all remaining fields with 0.5 to represent missing or undefined data.
5. We filter all columns with only one distinct value, which implies all data points had the same value for this column.
6. We compiled a vector of features from these numerically defined and filtered columns.
7. We plotted the of distribution of stars, a correlation heat map between some attributes, and the relationship between numbers of stars and one attribute. These graphs are representative of the sort of exploratory analysis we conducted to better understand this dataset before we began the machine learning component of the project.

The machine learning component of this part of the project involved trying out almost all classification models and several regression models provided by the pyspark.ml library. For classification algorithms, we trained them against the boolean labels we had generated. In this case, all stars ratings greater than 3.5 were given a label of 1.0. All stars ratings less than or equal to 3.5 were given a label of 0.0. We had limited success, and found that models generally achieved roughly a 70% accuracy rate. Generally, when models predicted 1.0, they were often correct, but when models predicted 0.0, they were sometimes wrong. The regression algorithms were trained against the original stars ratings column. They usually predicted a value between 3.0 and 3.5, with a fair amount of variation from that but little correlation between the variation and the actual stars ratings.

Furthermore, we developed a deep neural network using pytorch and pyspark, with the sparktorch library as an interface between the two frameworks. We achieved comparably limited success with our neural network.

Conclusion:

Descriptive Analysis:

We concluded that people on average tend to give primarily neutral or positive reviews to restaurant owners. However this is not the case for users or people with very few reviews, as they tend to have extreme feelings towards that business due to a particular experience and are more likely to give either 1 or 5 stars based on their experience.

Logistic Model Analysis:
We concluded with confidence that the model could make a prediction on whether or not a business would receive a high or low rating/stars based on attributes in our dataset such as number of stars, useful reviews, geographic factors, and average rating stars.

SVM Analysis:

We achieved a F1 score of **88.632%** using SVM to predict the star(postive/negative sentiments)

Business Feature Extraction Analysis:

The results of this analysis were not the same as the previous models, despite significant feature, model experimentation and training time, a prediction engine for extracting correlation between features and business did not come out strongly.

Within the context of restaurant data, these business attributes are not particularly significantly correlated with the number of stars in a review and thus cannot by used alone to accurately predict review sentiment.

In conclusion, we predict that businesses with a large variety of attributes are capable of getting great Yelp ratings. Furthermore, we propose that parsing review text is a more successful method of sentiment analysis and prediction than parsing business attributes.

Barriers and fixes:

1. The dataset was massive, so we ended up having to convert the data to CSV using the R jsonlite library.
2. Not many examples and resources for the SparkTorch and Pyspark(ML) tools were available. We encountered significant difficulty running the SparkTorch library, in particular, on our “single-node cluster.”
3. Setting up the AWS S3 integration with Jupyter notebook. We considered the option of using Amazon EMR which had easier integration as well as higher computation power with Amazon S3, however it was too costly and not free to use for us.