Akshay Ramesh

Vignesh Chandrasekharan

Professor Nagar

CS 4301.002

27 April 2018

Yelp Dataset Challenge - Final Report

Introduction

Every day, hundreds of restaurants open around the world and add to the thousands of businesses to choose from. Yelp.com is a website that provides reviews obtained from the public about restaurants and small businesses. The website garners thousands of visitors every day because of its credible restaurant ratings and reviews. For the past few years, Yelp has hosted the Yelp Dataset Challenge annually, which encourages students and other enthusiastic data scientists to use Yelp's huge data resources to further the restaurant-review field. We decided to take part in this challenge and wanted to analyze how ratings of a specific restaurant is affected by some variables such as review sentiments, length of the text and number of useful, funny, and cool votes received by the review.

Problem Description

Our main goal was to run a regression model on a collection of reviews and predict the number of stars (ratings) given a sentiment score and a few other predictor variables such as the length of the text, and the number of useful, funny, and cool votes received by the text in Spark R.

Related Work

Yelp Dataset Challenge has completed 10 rounds to date and currently is in round 11, which started on January 18, 2018. The round closes on June 30, 2018. The following two links contain information on the Yelp Dataset. The first one shows all previous winners of the Yelp Dataset Challenge including a description of their submissions. The second one is a collection of hundreds of research papers indexed by Google Scholar.

Links are as follows:-

- https://www.yelp.com/dataset/challenge/winners
- https://scholar.google.com/scholar?q=citation%3A+Yelp+Dataset&btnG=&hl=en&as_sdt=0%2C5.

Dataset Description

Yelp has generously provided multiple datasets to analyze for this challenge. All the datasets are of either JSON or SQL type. For our analysis, we chose the datasets that are of JSON type and specifically the review.json dataset. The datasets and their descriptions are as follows:-

- Business.json - Contains business data about location, categories and their ratings.
- Review.json - Contains reviews written by users for a particular business, ratings and their votes.
- User.json - Contains users attributes such as their name, number of reviews given, date since they are yelping and number of votes received.
- Checkin.json - Contains business timings and number of people come in that timing.
- Tip.json - Contains shorter reviews and its attributes to convey information quickly about a particular business.

- Photos.json - Contains photo, business id, caption of the photo and category it belongs to.

Overall, the dataset has 5,200,000 reviews, 174,000 businesses, 200,000 pictures, 11 metropolitan areas, and 1,100,000 tips given by 1,300,000 users. In business.csv, there are over 1.2 million business attributes such as parking, availability, ambience and hours. Moreover, there are aggregated check ins for each 174,000 businesses.

Working with JSON was comparatively much harder to work with than CSV in databricks. Fortunately, Yelp had posted the same datasets to kaggle in CSV format, so we used review.csv specifically.

Steps to retrieve data from AWS S3

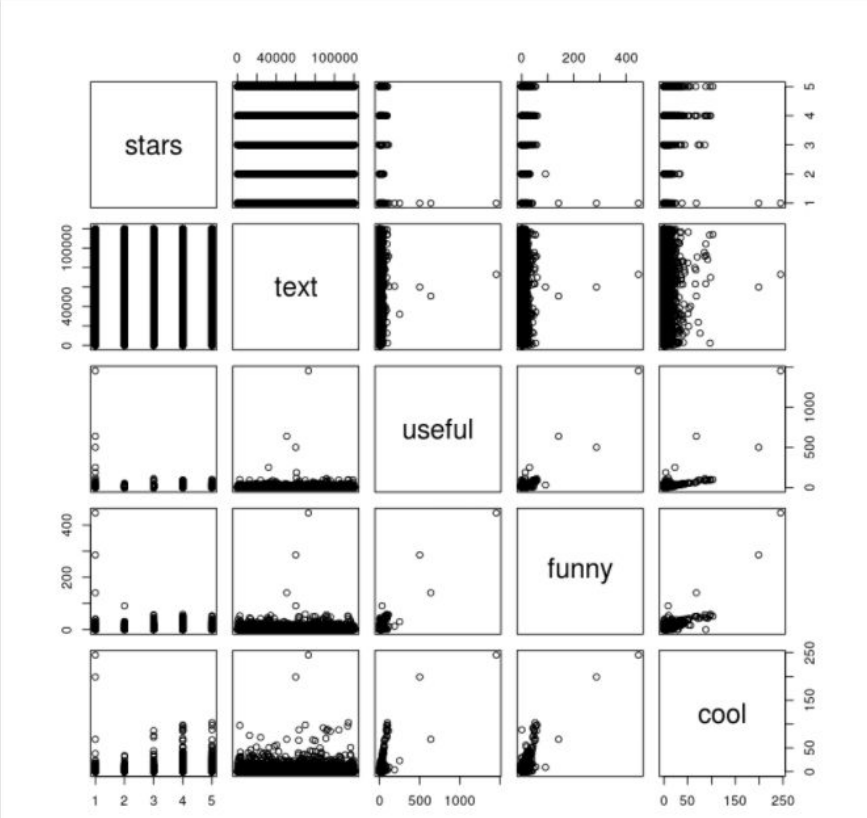
Since Databricks only allows a maximum of 1GB of local storage, we decided to store review.csv in an AWS S3 bucket. This was then accessed inside Databricks using an R package called aws.s3. To connect to the bucket and get the object, we had to pass an AWS Access Key, Secret Access Key, and Default Region as parameters. After retrieving the object which had the CSV file, we converted this object into a dataframe and stored the same in the variable called yelpData.

Preprocessing Techniques

The review.csv dataset originally had 5.2 million rows. To save time and computational requirements, we reduced it down to 120,216 rows and uploaded it to AWS S3 under the name "yelp_review". This was done using a software called csv splitter. The steps that we followed to link our Databricks Notebook to the S3 bucket are explained in the above section. After retrieving yelp_review.csv from S3, we previewed the dataset using R commands such as "display" to make sure that the file was correctly loaded. We also ensured that the dataset was

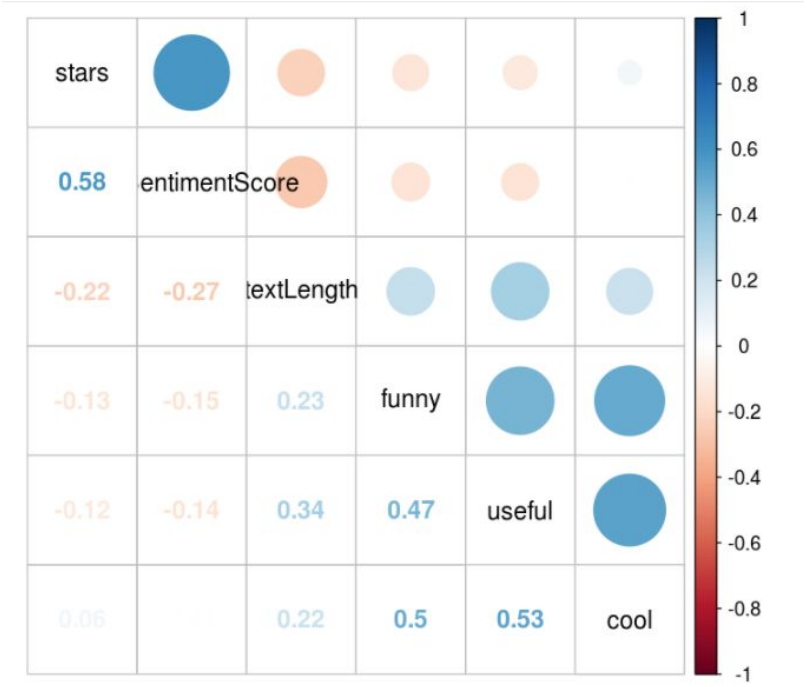
clean by checking for NA's or missing values and removing them. Surprisingly, the yelp dataset didn't have any NA's in them and it was clean. We started by examining the dataset itself by executing some commands such as "str" (gave structural summary of the dataset which included variable and their data types), "names" (gave all the column names) and "head" (which displayed the first 6 rows of the dataset). Since we didn't want to use all the columns, we just extracted the stars, text, useful, funny and cool column.

Then, we executed the "pairs" command, which displayed the graph of correlation between all variables :



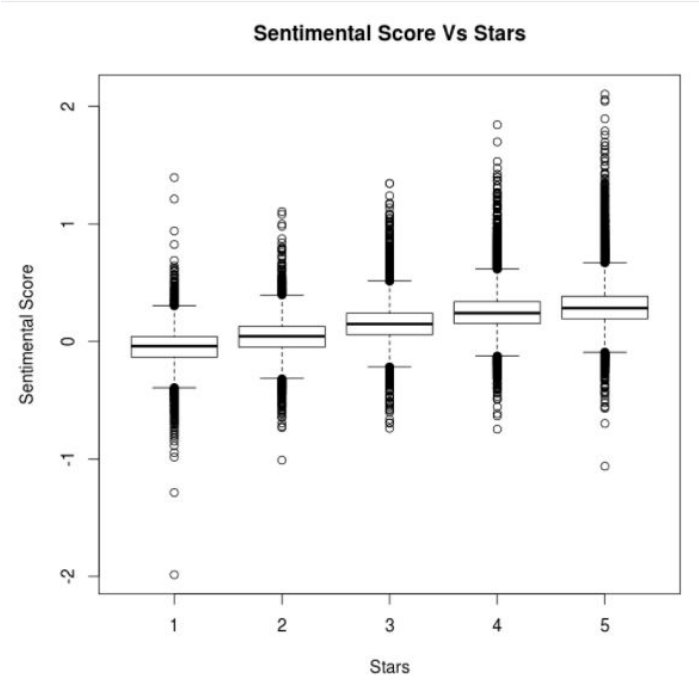
From the pairs graph above, we can see that useful and funny is correlated positively with cool variable because of the upward sloping line and grows fast. The correlation between useful and funny is the slowest and upward sloping.

Next, we calculated the sentimental scores of all reviews using a package called sentiment R and put those scores in a separate column. Afterwards, we calculated the length of the text and put it in another column. Now we used only stars, sentimental score, text length, funny, useful and cool to calculate and see the correlation between variables in the corrplot below:-



The scale on the right in the above correlation graph shows that the darker the blue color, the more positively correlated it is and on the other hand, the darker the red color, the more negatively correlated the variable it is. We can see that stars is very positively correlated with sentimental score, and funny is also positively correlated with useful and cool, whereas (text length and sentimental score), (text length and stars), (funny and stars), (useful and stars) are negatively correlated.

Next, we wanted to see how stars (on the scale of 1-5) related to sentimental score (positive and negative.) in the form of box plot shown below:-



The box plot above shows that as the ratings increased, the sentimental score increased, which means that the higher the number of stars, the more positive the review is. Therefore, we can conclude that number of stars and review sentimental score are directly related.

Lastly, the dataset was split into 75% training data and 25% testing data as input for our regression models.

Proposed Solution and Methods

Our plan was to try four regression machine learning algorithms (Linear Regression, Decision Tree, Random Forest and Boosting) to get a solid understanding of how each variable correlated with the number of stars. This would also provide enough results to see why one model performed better or worse than another. We will be using correlation accuracy of the particular

model as a metric and compare all correlation accuracies of all models to see which algorithm/model performed better on our dataset.

Experimental Results and Analysis

In all models, we used :

Target - stars

Predictors - useful, funny, cool, textLength , textSentimentScore

1.Linear Regression

- Summary:- All the chosen variables had 3 stars indicating low p values, which means that the variables are strongly correlated with the stars target variable.

```
summary(lm1) #summary of the linear regression model
```

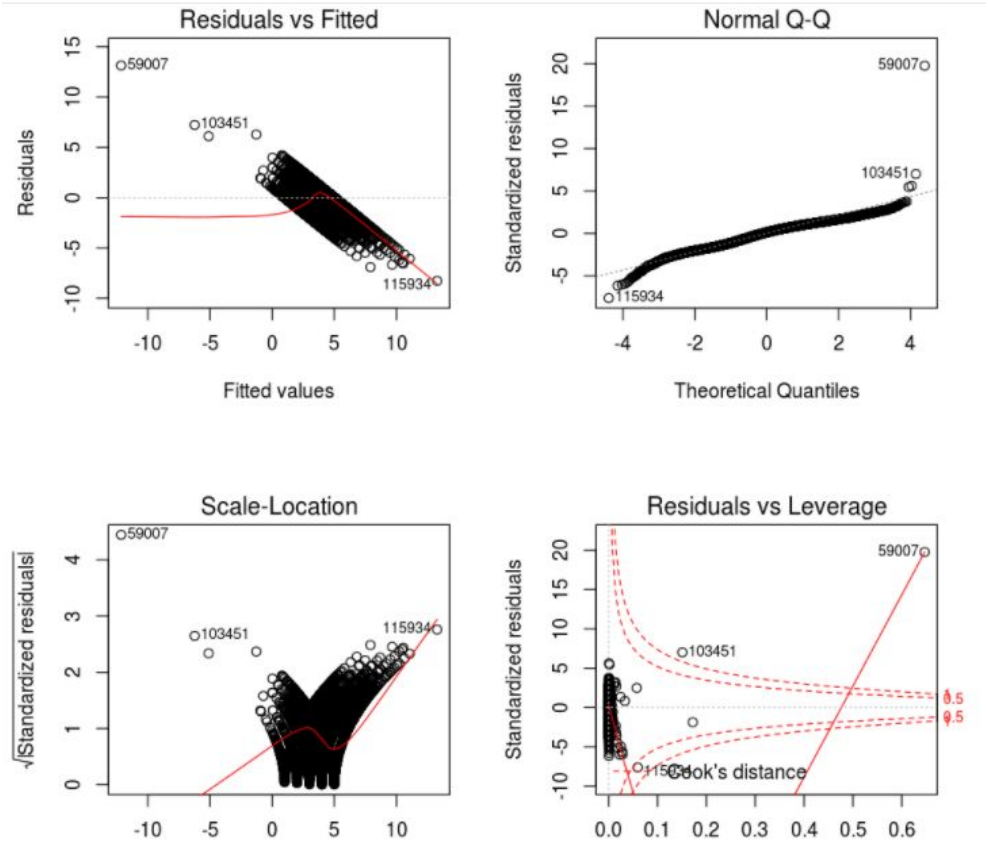
```
Call:
lm(formula = stars ~ useful + funny + cool + textLength + textSentimentScore,
    data = train_yelpData)

Residuals:
    Min      1Q  Median      3Q     Max
-8.2710 -0.7870  0.1441  0.8470 13.1437

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         3.045e+00  7.202e-03  422.75   <2e-16 ***
useful             -1.166e-02  1.076e-03  -10.83   <2e-16 ***
funny              -6.084e-02  3.333e-03  -18.25   <2e-16 ***
cool                1.193e-01  3.007e-03   39.68   <2e-16 ***
textLength         -2.204e-04  6.909e-06  -31.90   <2e-16 ***
textSentimentScore  4.037e+00  1.884e-02  214.32   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- Residuals graph:- The four graphs below does not show good linearity of the model (straighter the line, better the model). The bell shaped curves indicate that our model is

not a great linear fit.



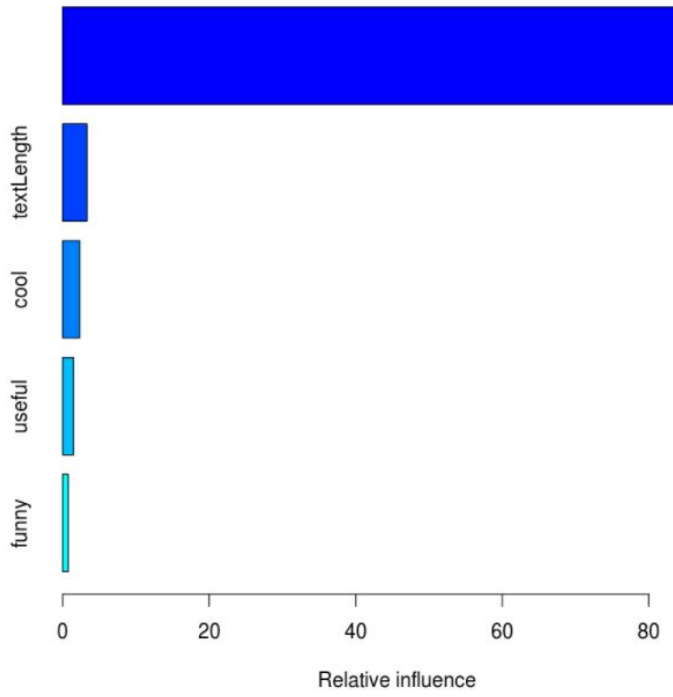
- Correlation Accuracy of linear regression:- 61.1402883620416 %

2. Decision Tree:-

- Correlation Accuracy of Decision Tree:- 66.0161485632484 %

3. Boosting:-

- Summary Graph of variable relative influence:- We can see that from the graph below, text length has better relative influence towards the boosting model.



- Correlation Accuracy of Boosting:- 70.1557471617871 %

4. Random Forest:-

- Correlation Accuracy of Random Forest:- 68.9701268056481 %

When comparing all the above correlation accuracies of 4 different algorithms, the ranking based on correlation accuracy (from highest accuracy to lowest accuracy) are as follows:-

Boosting > Random Forest > Decision Tree > Linear Regression

Boosting algorithm performed much better than the other three models with an accuracy of 70.15%. Boosting performed better because it is designed to create lot of trees consecutively,

which resulted in improved performance. Moreover, each tree has a better fit on a modified version of the previous tree when Boosting is used.

Conclusion

The best performing algorithm was the Boosting algorithm with the accuracy of 70.15%. Overall, the models we ran averaged around 67% accuracy. This was what we were expecting since the dataset has more than 100,000 rows and contains raw information from thousands of reviews. We learned that the text sentiment score that we calculated as well as the other predictors we chose had a high correlation with the number of stars. Working on this project helped us gain experience in Spark R, evaluating machine learning models, various data preprocessing techniques, and other big data analytics technologies.

Contribution of Team Members

Vignesh Chandrasekharan:-

- Used aws.s3 (R package) to link AWS S3 to the Databricks notebook and get the object from the bucket.
- Examined the dataset using display, head and other preprocessing commands
- Used Spark R char API to calculate the text length of all the reviews and put it in a new column.
- Coded Linear Regression & decision Tree model and calculated their correlation accuracies and plotted graphs.
- Wrote introduction, problem description, related work, dataset description and steps to retrieve data from AWS S3 in the report.

Akshay Ramesh:-

- Obtained CSV data from kaggle and reduced the dataset to have 120,216 rows using csv splitter software.
- Uploaded the dataset into AWS S3 and made the bucket public to be accessed by databricks.
- Used sentiment R package to calculate the sentimental score of all the reviews and put them in a new column.
- Coded Bagging & Random Forest model and calculated their correlation accuracies and plotted graphs.
- Wrote preprocessing techniques, proposed solution & methods, experimental results and analysis and conclusion in the report.

References

"Yelp Dataset." *Yelp.com*. N. p., 2018. Web. 22 Apr. 2018.

"Sparkr (R On Spark) - Spark 2.3.0 Documentation." Spark.apache.org. N. p., 2018. Web. 22 Apr. 2018.

"Sparkr Overview — Databricks Documentation." Docs.databricks.com. N. p., 2018. Web. 22 Apr. 2018.