Text Mining
On Yelp Reviews

BAN 675

By

Kishore Kumar | Ria Khanna | Kedar Patil | Ankur Bhagwat

13th December 2019

ABSTRACT

This project addresses the problem of sentiment analysis on yelp reviews; that is classifying reviews according to the sentiment expressed in them: cool, useful or funny. Analyzing the public sentiment is important for many applications such as firms trying to find out the response of their products in the market. The aim of this project is to develop a functional classifier for accurate and automatic sentiment classification of an unknown reviews. Exploratory Data Analysis was performed on the dataset, the data was cleaned, and sentiment analysis was performed on the reviews. Then a model was developed to relate the unique attributes of “cool”, “useful”, and “funny” to the text quality of the review. In this, TF-IDF was used as the basis of text quality. Finally, the model coefficients of different models were interpreted.

I. INTRODUCTION

Yelp is a business directory service and crowd-sourced review forum. Yelp relies on self submitted user reviews for the bulk of their content, having over 192 million reviews on its site. Because the reviews are so important for their business, getting important information out the text reviews can provide business value. Large volumes of texts are available via the reveiws, making text mining methods an appropriate form of analysis.

In this report, we work with a dataset of 10,000 text reviews. In addition to the text, each review has values for attributes such as Stars, Useful, Funny, and Cool. The data is understood through summary statistics and visualizations. Sentiment analysis is applied to each review to quantify the amount of positive, negative, and neutral sentiment in each review. The text quality of each review is quantified using two measures, TF-IDF and Lexical Diversity. The text mining measures are part of a linear model that relates Yelp unique attributes (Useful, Funny, and Cool) to text quality.

We have 10,000 Yelp reviews with 10 columns taken from Kaggle.com.
Attributes are as follows: business_id, Date, review_id, text, type, user_id, stars, cool, useful and funny.

- business_id, review_id, user_id are unique which is the ID for the business, review and user.
- date says what is the date of the review given.
- type says what is the type of the text. Since we have only reviews, it will always be “review”.
- Stars is a categorical variable with the number of stars the reviewer gave, from 1 to 5.

- Cool, Useful, and Funny are numerical variables with the number of votes the review got from other users in the 3 categories.

II. RESEARCH QUESTION AND MOTIVATION

Yelp reviews have the unique attributes ‘Cool’ ‘Funny’ and ‘Useful’, each a measure of how many times a user found this interview to match the respective adjective. Many websites have a ‘Useful’ vote feature, but ‘Cool’ and ‘Funny’ are novel to Yelp. It is assumed that a Useful review is preferable for users, but what about a Cool or Funny review? Does a review being Cool and/or Funny add to its informativeness? In this project, we research how these features are related to text quality. The results can be used by Yelp to inform what reviews they prioritize showing and as a starting point to develop their own quality measured.

III. DATA EXPLORATION AND STATISTICS

3.1. Frequent Word Visualization with a Word Cloud

Using the word cloud we can see the frequent words. Here in “Fig. 1” we can see that result.

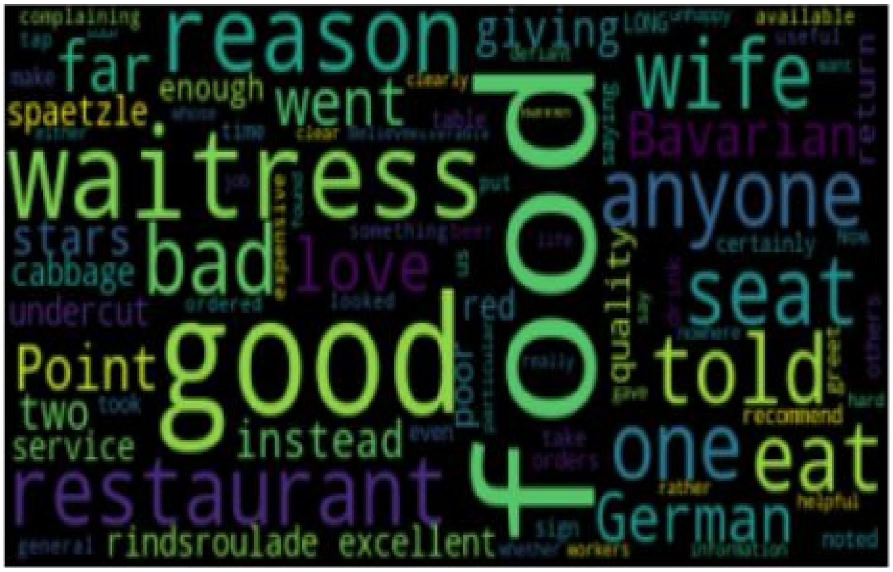


FIGURE 1. WORD CLOUD OF FREQUENT WORDS

3.2. Stars Distribution

3.2.1 Dummy Variables of Stars

Using one-hot encoding we created dummy variables for stars.

Here is the table.1 showing the first three rows of the stars column.

	1	2	3	4	5
0	0	0	0	0	1
1	0	0	0	0	1
2	0	0	0	1	0

TABLE 1. DUMMY VARIABLES OF STARS

3.2.2 Distribution of Stars

These are the bar plots showing the frequency of a particular star rating. Here in "Fig. 2" we can see that result.

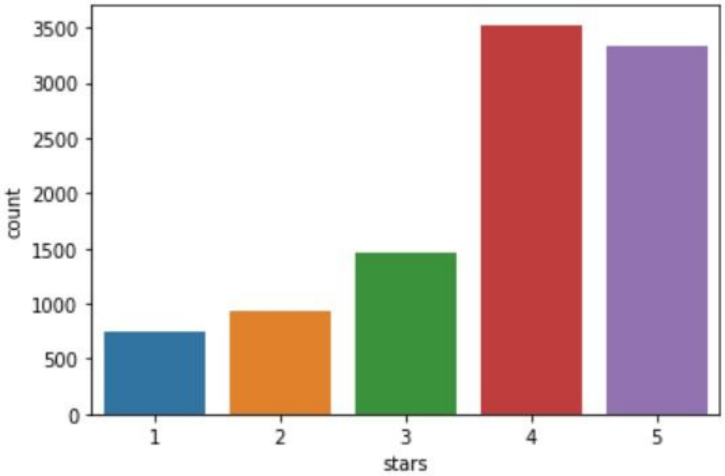


FIGURE 2. DISTRIBUTION OF STARS

3.3. Correlation and Heat Map

Value ranges between -1 and 1. If the value is closer to positive 1 then we can infer that the two variables are highly correlated. If the value is closer to 0 then they are not linearly dependent to each other. Using the Heatmap we can see the correlation between stars and the unique attributes. In "Fig. 3" we can see that result. Stars does not have high correlation with any other variable, while Cool Useful and Funny are all correlated to each other to varying degrees.

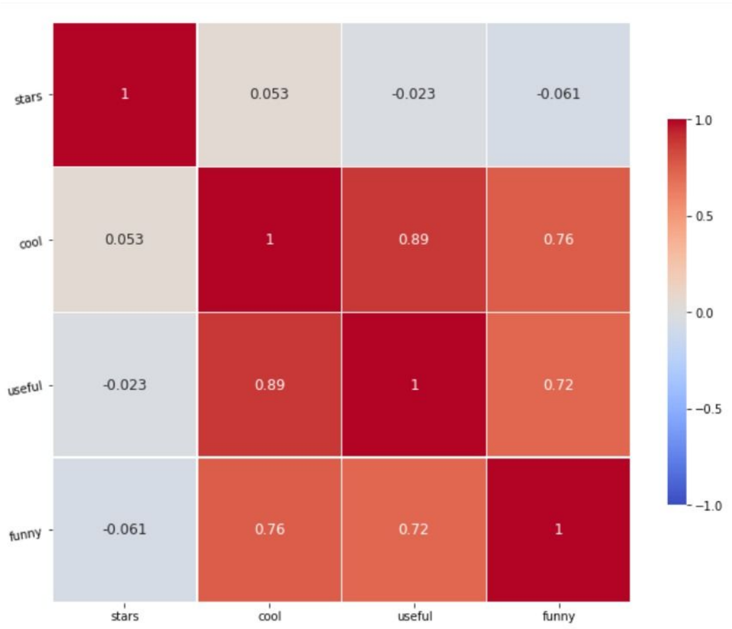


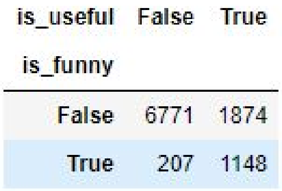
FIGURE 3. HEATMAP

3.4. Cross Tabulation

Cross Tabulation is primarily used to analyze categorical data. We can check the relationship among the variables of cool, funny, and useful by using cross tabulation. Is_cool, is_funny, and is_useful are binary variables with the value of True if a review has 2 or more votes for said adjective.

3.4.1 Cross Tabulation between Funny and Useful

Cross tabulation between Funny and Useful is seen in "Fig 4". The majority of funny reviews are also useful.



is_useful	False	True
is_funny		
False	6771	1874
True	207	1148

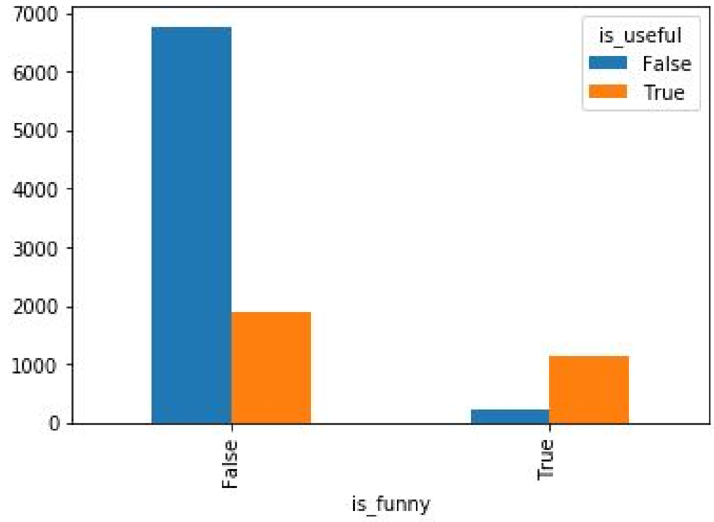
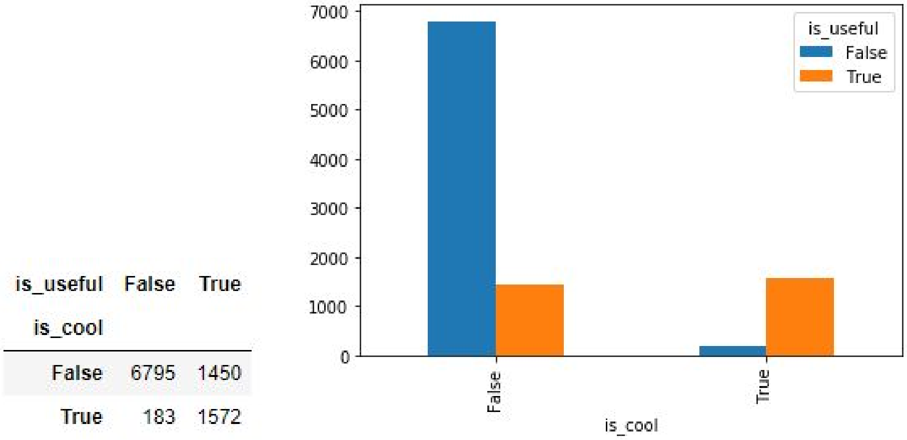


FIGURE 4. CROSS TABULATION BETWEEN FUNNY AND USEFUL

3.4.2 Cross Tabulation between Cool and Useful

Cross tabulation between Cool and Useful is seen in “Fig 5”. Most reviews that are cool are also useful. The most common category of review is neither cool or useful.

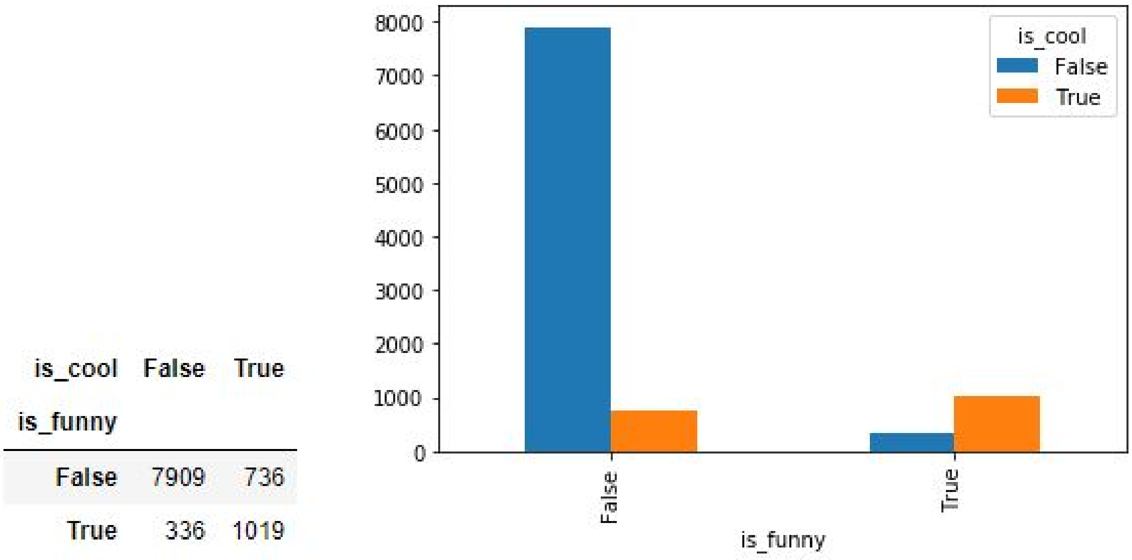


is_useful	False	True
is_cool		
False	6795	1450
True	183	1572

FIGURE 5. CROSS TABULATION BETWEEN COOL AND USEFUL

3.4.3 Cross Tabulation between Funny and Cool

Cross tabulation between Funny and Cool is in “Fig 6” For 89.2% of the reviews, cool and funny align.



is_cool	False	True
is_funny		
False	7909	736
True	336	1019

FIGURE 6. CROSS TABULATION BETWEEN FUNNY AND COOL

IV. METHODOLOGIES

4.1. Sentiment Analysis

Sentiment analysis was conducted using VADER Sentiment Analysis. For each review, a positive, native, and neutral sentiment score was obtained. “Fig 7”, “Fig 8”, and “Fig 9”show the distributions of the different scores.

“Fig 7” shows that most of the negative polarity is in between 0 and 0.1.

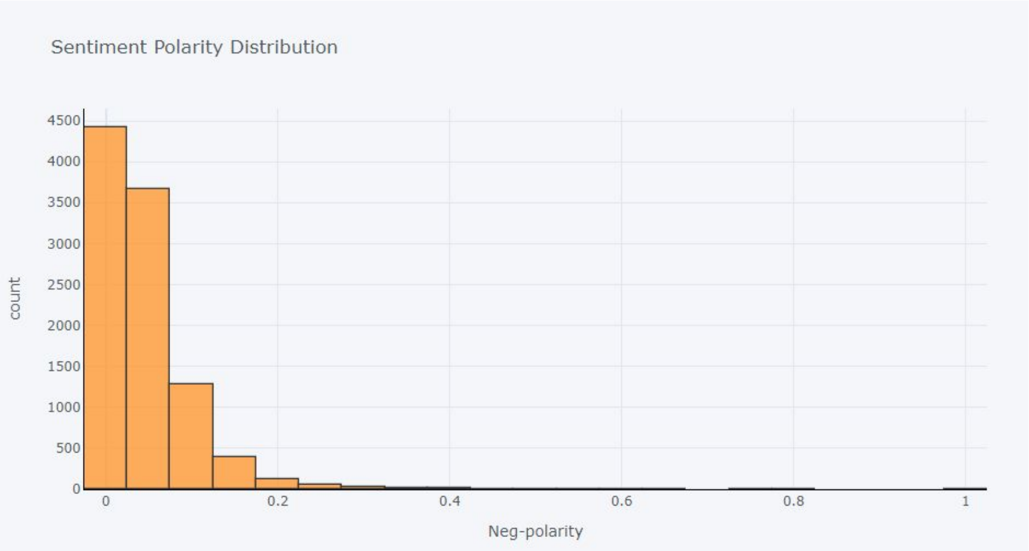


FIGURE 7. DISTRIBUTION OF NEGATIVE POLARITY SCORES.

“Fig 8” shows that the positive polarity is mostly distributed in the uniform range between 0 to 0.4

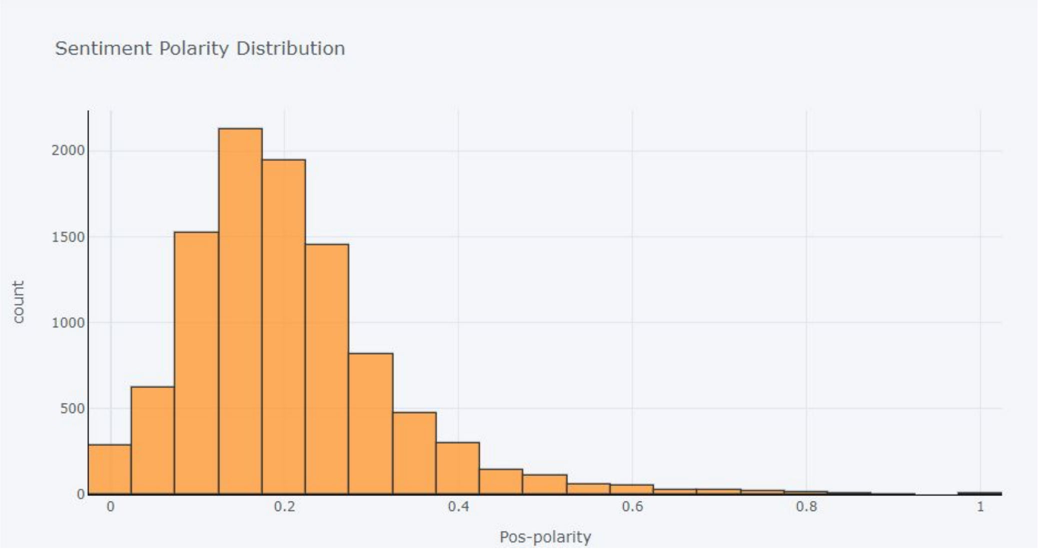


FIGURE 8. DISTRIBUTION OF POSITIVE POLARITY SCORES.

“Fig 9” shows that the neutral polarity is mostly distributed in the uniform range between 0.6 to 1.

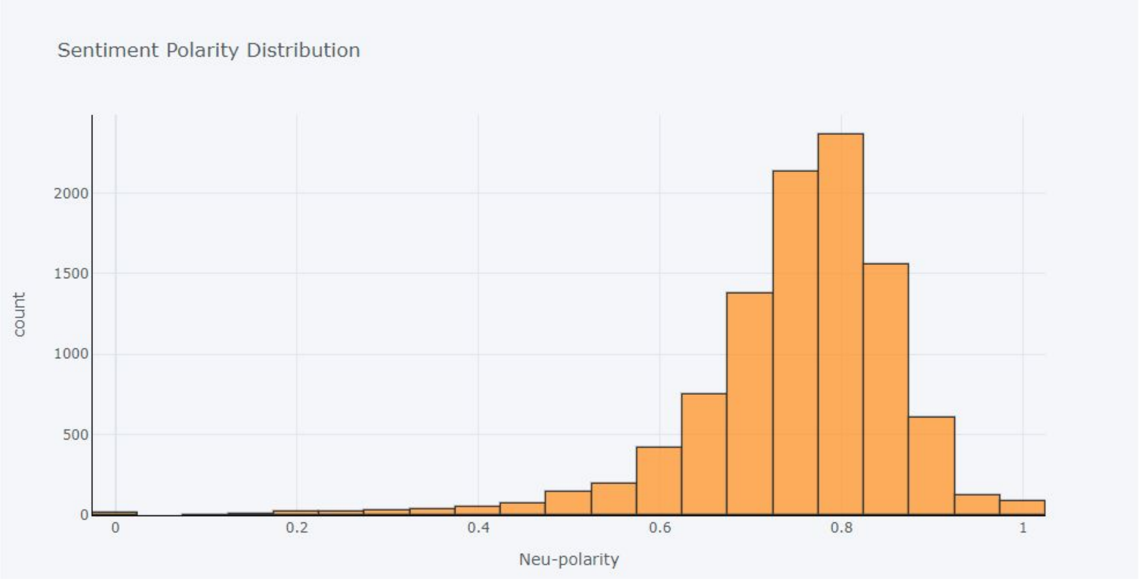


FIGURE 9. DISTRIBUTION OF NEUTRAL POLARITY SCORES.

4.2. Quality Measures: TF-IDF and Lexical Diversity

TF-IDF is a feature vectorization method used in text mining to find the importance of a term to a document in the corpus. Words with high TF-IDF are those that are very important in the current document but not frequent in other documents. these words are the key words of the current article. If a word is not present in a given document/review, the TF_IDF value for the word in that review will be 0. We use the average non-zero TF-IDF to measure the text quality of a review. By taking the average TF-IDF over non-zero values, we are considering the words that are present in document only. The more informative these words are, the higher the non-zero TF-IDF will be.

Lexical Diversity is another measure of text quality. It is calculated as the ratio of unique words in a review to the total number of words in the review. The more reducant words are used, the lower the Lexical Diversity value.

4.3. Model Development

Linear models were developed with each quality measure as the dependent variable. Many independent were variables included to control for aspects of the reviews. These were a binary of variable for each possible value of Stars (ie 5 dummy variables), the number of words, the number of sentences, a binary variable indicating if the review contained an exclamation mark. The results from sentiment analysis (positive, negative, and neutral scores) were also included as independent variables. Finally, the binary variables is_cool, is_funny, and is_useful (as previously defined) were included.

The dataset was split into training and test sets randomly. 75% of the reviews were used for training. For each model, many combinations of the input variables were tested. The best, significant model was chosen based on the basis of test set performance. The model with the lowest test set error (RMSE) and highest test set score (R2) were chosen as the final model. Then, the coefficients for is_cool, is_funny, and is_useful were interpreted to understand how each variable was related to text quality,

V. RESULTS

5.1. TF-IDF based models

AVERAGE TF-IDF over NON-ZERO Values

For the mode predicting average TF-IDF over non-zero, the best model 13 terms. These can be seen below in “Fig10”. The best model has a score of 0.58.

The coefficient for is_pos	0.1776610308952805
The coefficient for is_neg	0.29508945662741454
The coefficient for is_cool	-0.004534872285599778
The coefficient for is_funny	0.0037141008361668494
The coefficient for is_useful	-0.006841376212065588
The coefficient for star_1	0.005164718954332752
The coefficient for star_2	0.0013352766978667818
The coefficient for star_3	-0.0037653422133125847
The coefficient for star_4	-0.002491458025598286
The coefficient for star_5	-0.00024319541328987788
The coefficient for n_sent	-0.0007719838666253795

The coefficient for n_paras	0.000626502083430801
The coefficient for n_words	-0.0002613597650564792
Test RMSE::	0.0448489285742446
Test score::	0.5815078220074996

TABLE 2. COEFFICIENTS FOR TF-IDF MODEL.

From the results, we see a cooler review has a lower average non-zero TF-IDF. A useful review also has a lower value. Meanwhile, a funny value has a higher value. We also see a higher star rating on the review (3, 4, or 5) has a lower quality value than the lower star ratings (1 and 2).

5.2. Lexical Diversity model

For the mode predicting Lexical Diversity, the best model 13 terms. These can be seen below in “Fig 11”. The best model has a score of 0.664 on the test set. This is higher than the average non-zero TF-IDF : so we can say that lexical diversity is predicted better that TF-IDF

The coefficient for is_pos	0.18033372221490884
The coefficient for is_neg	0.1328886101988586
The coefficient for is_cool	-0.004119851707756732
The coefficient for is_funny	0.007530928161972233
The coefficient for is_useful	-0.009049614958928512
The coefficient for star_1	0.013371522737168854
The coefficient for star_2	0.0024804628244485937
The coefficient for star_3	-0.007474675004204516

The coefficient for star_4	-0.0053844890725696405
The coefficient for star_5	-0.0029928214848438757
The coefficient for n_sent	-0.004102250493901543
The coefficient for n_paras	0.0017785616670686337
The coefficient for n_words	-0.0005984872488307424
Test RMSE::	0.07167141428743737
Test score::	0.6641465025372459

TABLE 3. COEFFICIENTS FOR MODEL OF LEXICAL DIVERSITY.

From the results, we see a cool or useful review has a lower lexical diversity. A funny review will have a higher lexical diversity value. Meanwhile, a funny value has a higher value. Of these three variables, is_useful has the greatest absolute effect, lowering the lexical diversity by 0.009. We see that a higher star rating on the review (3, 4, or 5) has a lower quality value than the lower star ratings (1 and 2).

5.3. Interpretation of models

Funny has a positive relationship with lexical diversity and average non-zero of TF-IDF. Cool and Useful have negative relationships with lexical diversity and average non-zero of TF-IDF. These are non-intuitive results, as it was assumed a higher quality text would lead to a more useful review. For these measures, cool and useful work in the same direction and in the opposite direction of funny. However, since the results do not make logical sense, it is difficult to judge if these relationships hold true for general quality. Additionally, even though cool and funny usually align, as seen by a cross tabulation, they contribute in opposite directions in the linear models. Even when one of these variables was removed from the model, the signs of useful and the remaining variable did not change.

The non-intuitive results suggest that lexical diversity and average non-zero of TF-IDF are not measuring Yelp review quality, since usefulness has a negative relationship with these measures. The more useful a review is for Yelp users, the less rich the text of that review was found to be. Thus, for Yelp, some other quality measures should be developed to assess the informativeness of a review. General text quality does not appear to apply. More data with more attributes might also improve the results.

Additionally, more advanced models, with different features, can be developed to incorporate more of the variance in the data and thus present a more robust solution.

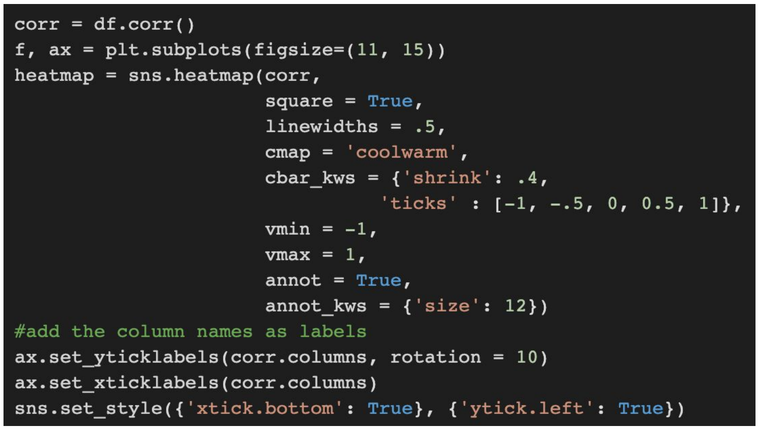
VI. CONCLUSION

We explored Yelp review data using summary statistics, text mining methods and measures, and finally a linear model to relate quality to the attributes of a review. We did this to explore the relationship between attributes ‘Cool’,‘Funny’, and ’Useful’ to text quality/informativeness. We were able to create linear models for text quality that controlled for different features of a review. However, it was found that ‘'Useful' has negative relationships with the text quality measures of lexical diversity and average non-zero of TF-IDF. This suggests that lexical diversity and average non-zero of TF-IDF are not measuring Yelp review quality since usefulness has a negative relationship with these measures. Further research is needed to relate attributes of the text to informativeness.

VII. APPENDIX

7.1 Codes for Data Exploration and Statistics

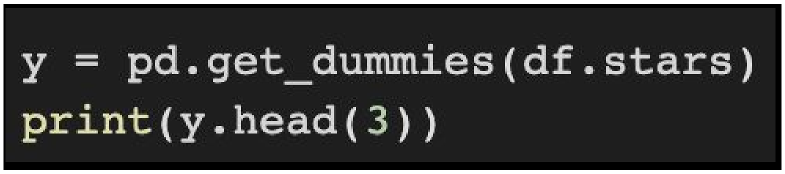
Code for 3.1 i.e., Fig1.Word Cloud of frequent words



```
corr = df.corr()
f, ax = plt.subplots(figsize=(11, 15))
heatmap = sns.heatmap(corr,
                      square = True,
                      linewidths = .5,
                      cmap = 'coolwarm',
                      cbar_kws = {'shrink': .4,
                                'ticks' : [-1, -.5, 0, 0.5, 1]},
                      vmin = -1,
                      vmax = 1,
                      annot = True,
                      annot_kws = {'size': 12})
#add the column names as labels
ax.set_yticklabels(corr.columns, rotation = 10)
ax.set_xticklabels(corr.columns)
sns.set_style({'xtick.bottom': True}, {'ytick.left': True})
```

CODE 1. CODE FOR WORD CLOUD OF FREQUENT WORDS

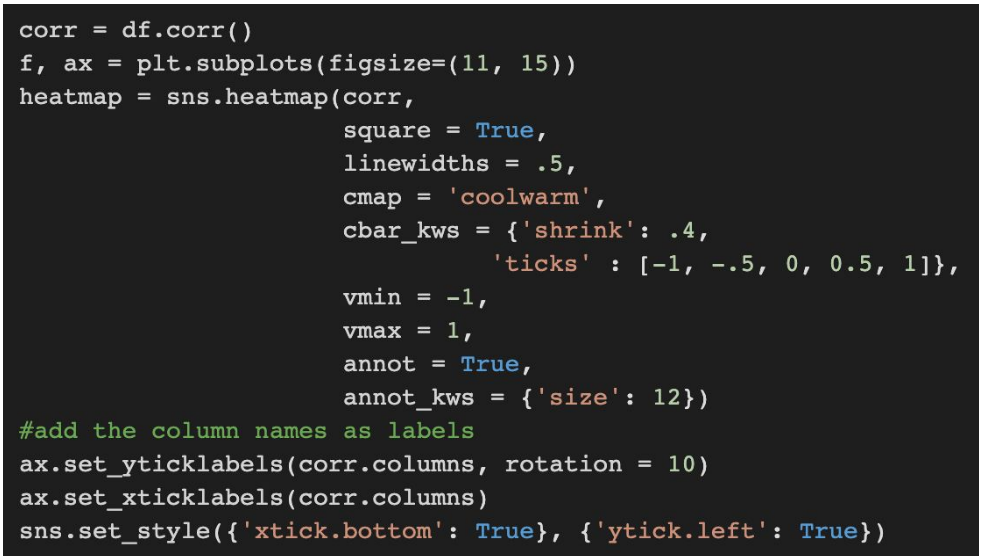
Code for 3.2 i.e., Table 1. Dummy Variables of Stars



```
y = pd.get_dummies(df.stars)
print(y.head(3))
```

CODE 2. CODE FOR DUMMY VARIABLES OF STARS

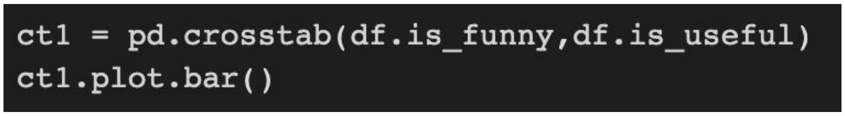
Code for 3.3 i.e., Fig3. Heatmap



```
corr = df.corr()
f, ax = plt.subplots(figsize=(11, 15))
heatmap = sns.heatmap(corr,
                      square = True,
                      linewidths = .5,
                      cmap = 'coolwarm',
                      cbar_kws = {'shrink': .4,
                                'ticks' : [-1, -.5, 0, 0.5, 1]},
                      vmin = -1,
                      vmax = 1,
                      annot = True,
                      annot_kws = {'size': 12})
#add the column names as labels
ax.set_yticklabels(corr.columns, rotation = 10)
ax.set_xticklabels(corr.columns)
sns.set_style({'xtick.bottom': True}, {'ytick.left': True})
```

CODE 3. CODE FOR HEATMAP

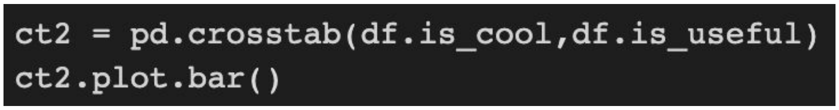
Code for 3.4.1 i.e., Fig4. Cross Tabulation between funny and useful



```
ct1 = pd.crosstab(df.is_funny,df.is_useful)
ct1.plot.bar()
```

CODE 4. CODE FOR CROSS TABULATION BETWEEN FUNNY AND USEFUL

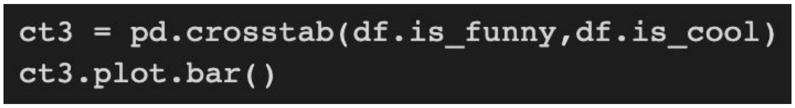
Code for 3.4.2 i.e., Fig5. Cross Tabulation between Cool and useful



```
ct2 = pd.crosstab(df.is_cool,df.is_useful)
ct2.plot.bar()
```

CODE 5. CODE FOR CROSS TABULATION BETWEEN COOL AND USEFUL

Code for 3.4.3 i.e., Fig6. Cross Tabulation between funny and cool

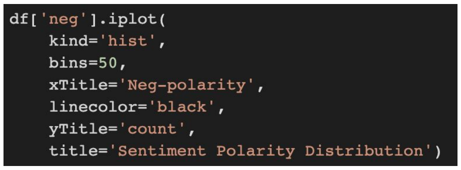


```
ct3 = pd.crosstab(df.is_funny,df.is_cool)
ct3.plot.bar()
```

CODE 6. CODE FOR CROSS TABULATION BETWEEN FUNNY AND COOL

7.2 Codes for Methodologies

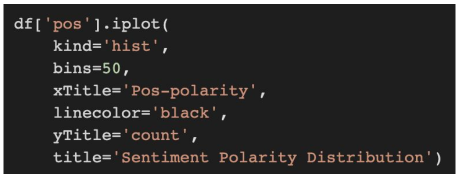
7.2.1 Code for distribution of negative polarity scores



```
df['neg'].iplot(
    kind='hist',
    bins=50,
    xTitle='Neg-polarity',
    linecolor='black',
    yTitle='count',
    title='Sentiment Polarity Distribution')
```

CODE 7. CODE FOR DISTRIBUTION OF NEGATIVE POLARITY SCORES

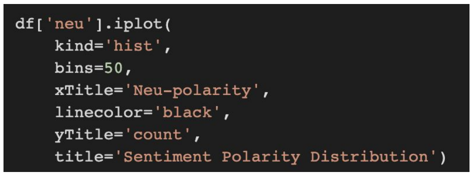
7.2.2 Code for distribution of positive polarity scores



```
df['pos'].iplot(
    kind='hist',
    bins=50,
    xTitle='Pos-polarity',
    linecolor='black',
    yTitle='count',
    title='Sentiment Polarity Distribution')
```

CODE 8. CODE FOR DISTRIBUTION OF POSITIVE POLARITY SCORES

7.2.3 Code for distribution of neutral polarity scores

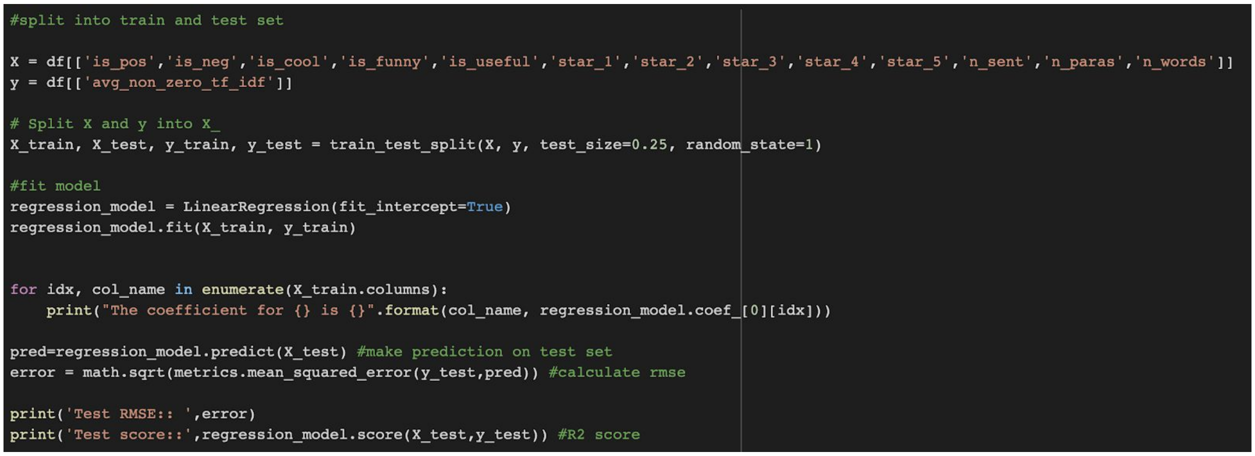


```
df['neu'].iplot(
    kind='hist',
    bins=50,
    xTitle='Neu-polarity',
    linecolor='black',
    yTitle='count',
    title='Sentiment Polarity Distribution')
```

CODE 9. CODE FOR DISTRIBUTION OF NEUTRAL POLARITY SCORES

7.3. Codes for Result

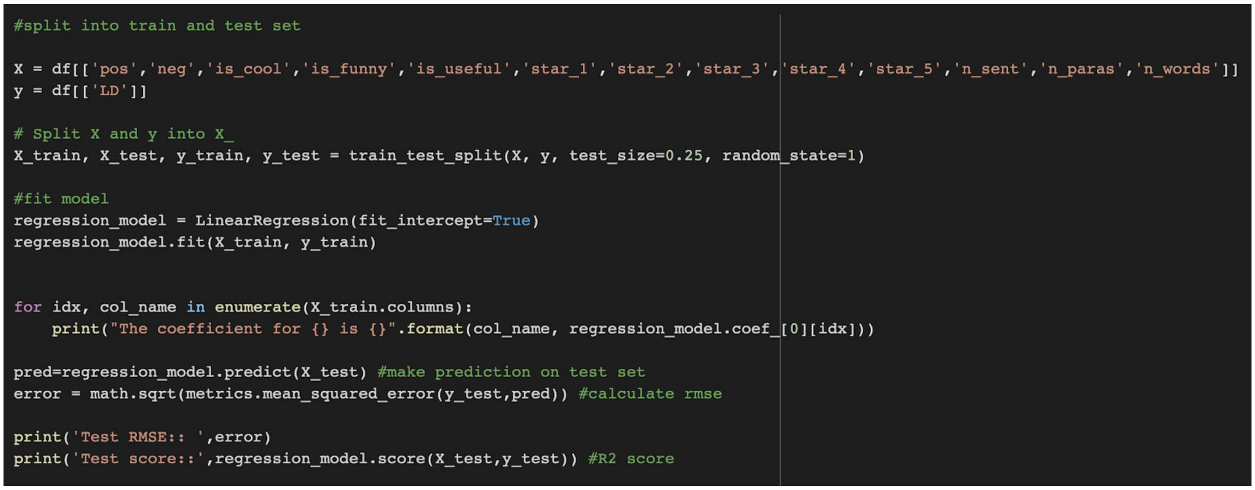
7.3.1 Code for Table 2. i.e., coefficients for TF-IDF model



```
#split into train and test set

X = df[['is_pos','is_neg','is_cool','is_funny','is_useful','star_1','star_2','star_3','star_4','star_5','n_sent','n_paras','n_words']]
y = df[['avg_non_zero_tf_idf']]

# Split X and y into X_
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=1)

#fit model
regression_model = LinearRegression(fit_intercept=True)
regression_model.fit(X_train, y_train)


for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))

pred=regression_model.predict(X_test) #make prediction on test set
error = math.sqrt(metrics.mean_squared_error(y_test,pred)) #calculate rmse

print('Test RMSE:: ',error)
print('Test score::',regression_model.score(X_test,y_test)) #R2 score
```

CODE 10. CODE FOR COEFFICIENTS FOR TF-IDF MODEL

7.3.2 Code for Table 3. i.e, Coefficients for model of Lexical Diversity



```
#split into train and test set

X = df[['pos','neg','is_cool','is_funny','is_useful','star_1','star_2','star_3','star_4','star_5','n_sent','n_paras','n_words']]
y = df[['LD']]

# Split X and y into X_
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=1)

#fit model
regression_model = LinearRegression(fit_intercept=True)
regression_model.fit(X_train, y_train)


for idx, col_name in enumerate(X_train.columns):
    print("The coefficient for {} is {}".format(col_name, regression_model.coef_[0][idx]))

pred=regression_model.predict(X_test) #make prediction on test set
error = math.sqrt(metrics.mean_squared_error(y_test,pred)) #calculate rmse

print('Test RMSE:: ',error)
print('Test score::',regression_model.score(X_test,y_test)) #R2 score
```

CODE 11. CODE FOR COEFFICIENTS FOR MODEL OF LEXICAL DIVERSITY

Link for Dataset: https://www.kaggle.com/omkarsabnis/yelp-reviews-dataset

About Yelp:

https://en.wikipedia.org/wiki/Yelp

https://www.wordstream.com/blog/ws/2013/07/22/yelp-reviews