LUDDY SCHOOL OF INFORMATICS

# YELP REVIEWS CLASSIFICATION

By:
Shyam kumar kanuru
Sumanth muvva
Sri Venkata Sai Anoop Bulusu

INDIANA UNIVERSITY BLOOMINGTON

# OVERVIEW

- Abstract
- Motivation
- About the dataset
- Approach
- Modelling and Results
- Conclusion
- Future work

SECTION 1

# ABSTRACT

# ABSTRACT

- Yelp is a local business directory and forum to review products, services, or places.
- Used Yelp's review data to determine user's sentiment or opinion about products, services, or places.
- Sentiment or opinion are classified into positive reviews, or negative reviews.

SECTION 2

# MOTIVATION

# MOTIVATION

- Considering 92% of consumers now read online reviews before purchasing, it might be time for all small businesses to start caring about what is said online; and more specifically, about their Yelp reviews.

- Our study is one such attempt to filter out fake reviews on social media  making it easy for the user to assess the product, services and places

SECTION 3

# ABOUT THE DATASET

# DATASET DESCRIPTION

- The Yelp reviews polarity dataset is built using the various customer ratings.
- The dataset contains 560,000 training samples and 38,000 testing samples.
- The dataset has 2 classes:

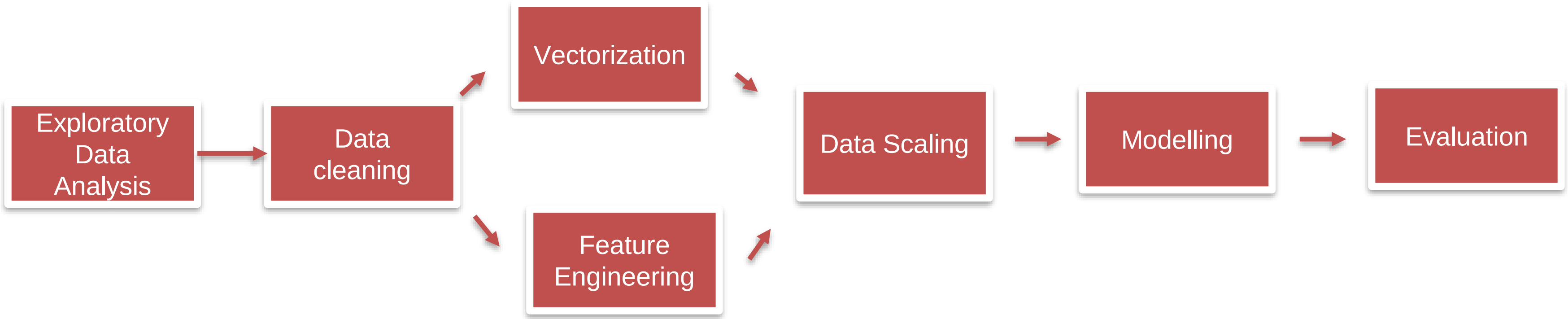
    1. Class 1: Negative polarity
    2. Class 2: positive polarity

- The files train.csv and test.csv contain all the training samples as comma-separated values.
- There are 2 columns in them, corresponding to class index (1 and 2) and review text.

SECTION 4

# APPROACH

# APPROACH

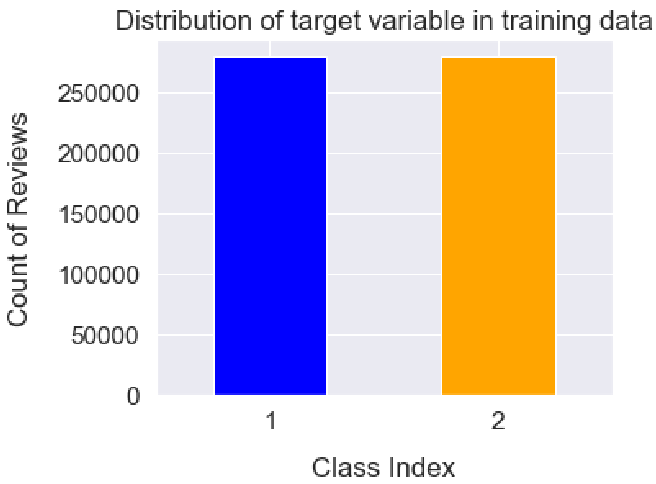

Exploratory Data Analysis
Data cleaning
Vectorization
Feature Engineering
Data Scaling
Modelling
Evaluation

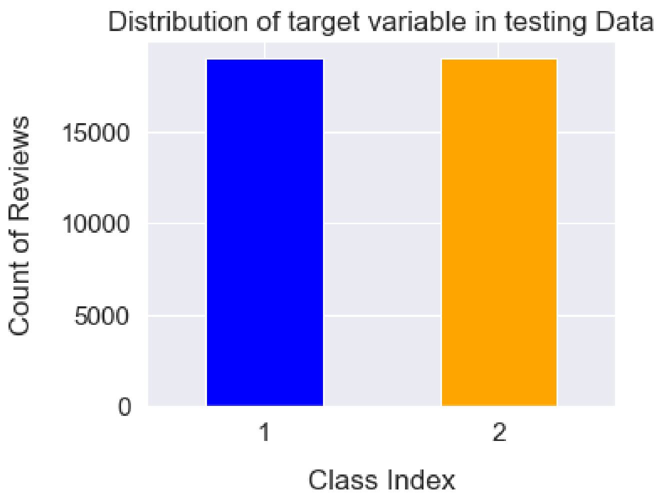
SECTION 4.1

# EXPLORATORY DATA ANALYSIS(EDA)

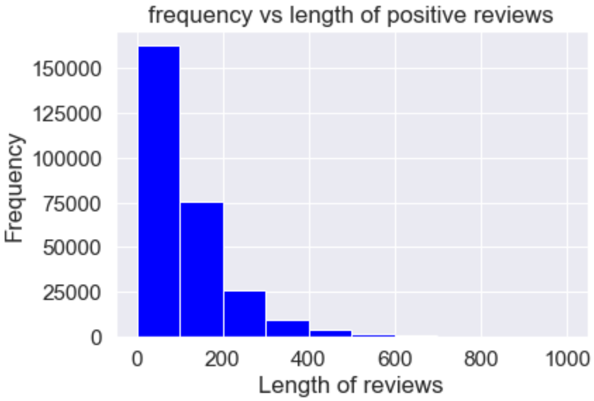
# EDA

- Target data is equally distributed

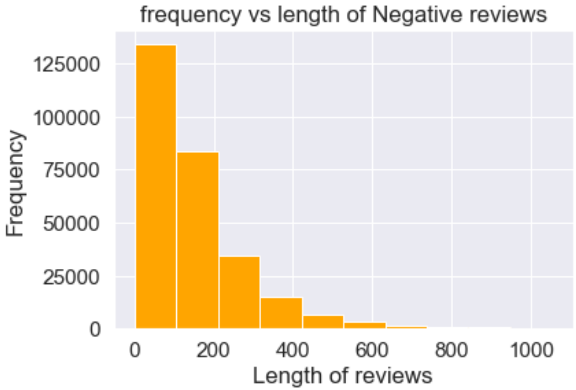


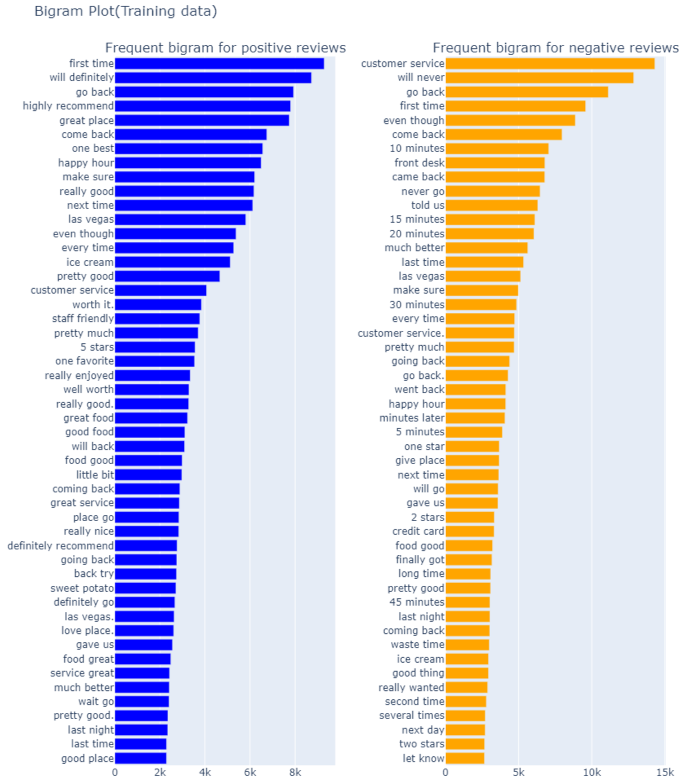
- Length of negative reviews in the training data seems to be more.

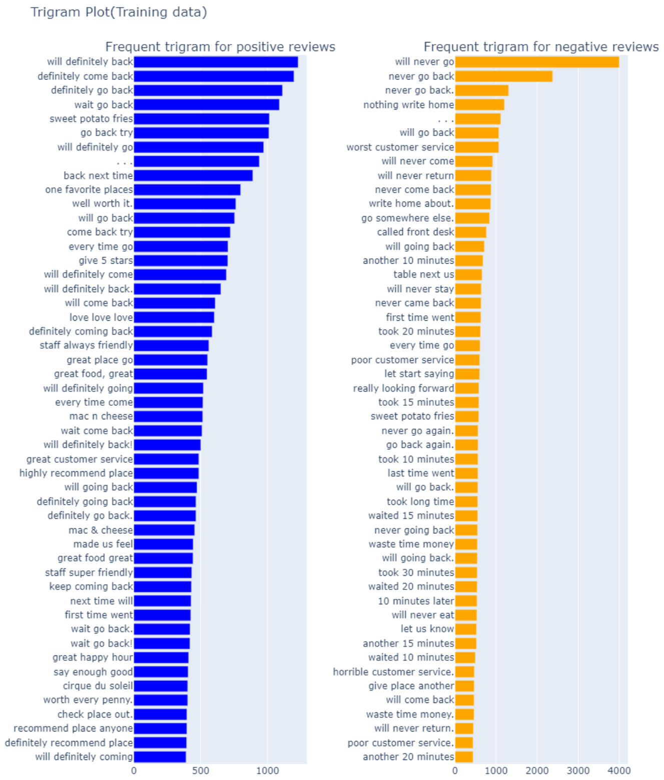






- Frequent Bigram and Trigram for positive and negative reviews



SECTION 4.2

# PRE-PROCESSING

# TEXT CLEANING

- Removed all the punctuations, URL’s, numbers, unwanted characters, etc.

```python
#cleaning the reviews
def cleaning_text(review):

    #removing the url's
    review = re.sub('http\S+\s*', ' ', review)
    #removing the  punctuations
    review = re.sub('[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>@[\]^_`{|}~"""), ' ', review)
    #removing non-ascii characters
    review = re.sub(r'[^\x00-\x7f]',r' ', review)
    #removing mentions (i.e, @)
    review = re.sub('@\S+', '  ', review)
    #removing hashtags
    review = re.sub('#\S+', ' ', review)
    #remove numbers
    review = re.sub("\d+", ' ', review)
    #removing extra whitespaces, wherever applicable
    review = re.sub('\s+', ' ',review)
    #converting the text into lowercase
    review = review.lower()

    return reviews
```

# FEATURE ENGINEERING

- Added few custom features like number of words in each review, average length of each word in a review, etc.

```
def custom_features(data):
    #number of words in each review
    data['no_of_words'] = data['clean_review_text'].apply(lambda x: len(str(x).split()))
    #average length of each word in each review
    data['avg_length_word'] = data['clean_review_text'].apply(lambda x: np.average([len(each_word)
                                                                          for each_word in str(x).split()]))
    #number of characters in each review
    data['no_of_characters'] = data['clean_review_text'].apply(lambda x: len(str(x)))
    #number of unique words in each review
    data['no_of_unique_words'] = data['clean_review_text'].apply(lambda x: len(set(str(x).split())))

    return data
```

# FEATURE ENGINEERING

- The following is the data frame obtained after performing feature engineering

```
#train data after adding custom features
yelp_train_data.head()
```

| | class_index | review_text | clean_review_text | no_of_words | avg_length_word | no_of_characters | no_of_unique_words |
|---|---|---|---|---|---|---|---|
| **0** | 1 | Unfortunately, the frustration of being Dr. Go... | unfortunately the frustration of being dr gold... | 118 | 4.237288 | 618 | 80 |
| **1** | 2 | Been going to Dr. Goldberg for over 10 years. ... | been going to dr goldberg for over years i thi... | 98 | 3.908163 | 481 | 71 |
| **2** | 1 | I don't know what Dr. Goldberg was like before... | i don t know what dr goldberg was like before ... | 213 | 4.234742 | 1115 | 132 |
| **3** | 1 | I'm writing this review to give you a heads up... | i m writing this review to give you a heads up... | 203 | 4.029557 | 1021 | 108 |
| **4** | 2 | All the food is great here. But the best thing... | all the food is great here but the best thing ... | 76 | 4.105263 | 388 | 53 |

# VECTORIZATION

- Bag of words (Count Vectorizer)
- Term Frequency inverse document frequency (TF-IDF Vectorizer)

Experimented all the models using both the techniques.

SECTION 4.3

# MODELLING AND RESULTS

# RESULTS

| MODEL | TRAINING ACCURACY | TEST ACCURACY | OVERFITTING OR NOT |
|---|---|---|---|
| Logistic Regression- CV | 85.86 | 86.02 | NO |
| Logistic Regression- TFIDF | 91.56 | 91.7 | NO |
| Support Vector Machines -CV | 90.94 | 91.06 | NO |
| Support Vector Machines - TFIDF | 89.94 | 90.02 | NO |
| Naive-Bayes - CV | 88.72 | 88.18 | NO |
| Naive-Bayes - TFIDF | 76.82 | 76.41 | NO |
| XGBoost - CV | 85.08 | 85.11 | NO |
| XGBoost - TFIDF | 85.14 | 85.14 | NO |
| Random Forest - CV | 92.95 | 85.82 | YES |
| Random Forest-TFIDF | 93.94 | 86.21 | YES |
| DL MODEL- CV (LSTM/CNN) | Future work | Future work | |
| DL MODEL-TFIDF (LSTM/CNN) | Future work | Future work | |

SECTION 5

# CONCLUSION

# CONCLUSION

1. From the above table we incur that Logistic regression with TFIDF and SVM are giving better results with the test data
2. The Random forest model is overfitting with both count vectorizer and TFIDF data.

SECTION 6

# FUTURE WORK

# FUTURE WORK

1. Perform feature selection and then implement Random forest with better parameters in order avoid overfitting.

2. In future we will try to implement and explore more deep learning model such as CNN, LSTM.

SECTION 7

# REFERENCES

# REFERENCES:


1. https://towardsdatascience.com/building-a-sentiment-analysis-model-using-yelp-reviews-and-ml-ensemble-methods-80e45db6d0c7

2. https://www.yelp.com/dataset

3. https://www.ics.uci.edu/~vpsaini/

4. https://xgboost.readthedocs.io/en/stable/

5. https://scikit-learn.org/stable/modules/svm.html