MEF UNIVERSITY

YELP REVIEW DATASET SENTIMENT ANALYSIS USING MACHINE LEARNING TECHNIQUES

Capstone Project

Anılcan Atik

İSTANBUL, 2021

MEF UNIVERSITY

YELP REVIEW DATASET SENTIMENT ANALYSIS USING MACHINE LEARNING TECHNIQUES

Capstone Project

Anılcan Atik

Advisor: Asst. Prof. Dr. Evren Güney

İSTANBUL, 2021

MEF UNIVERSITY

Name of the project: Yelp Review Dataset Analysis Using ML Techniques
Name/Last Name of the Student: Anılcan Atik
Date of Thesis Defense: 25/01/2021

I hereby state that the graduation project prepared by Anılcan Atik has been completed under my supervision. I accept this work as a “Graduation Project”.

25/01/2021
Asst. Prof Dr. Evren Güney

I hereby state that I have examined this graduation project by Anılcan Atik which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

25/01/2021

Prof Dr. Özgür Özlük

Director
of
Big Data Analytics Program

We hereby state that we have held the graduation examination of __________ and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Asst. Prof. Evren Güney	………………………..
2. Prof Dr. Özgür Özlük	……………………….

Academic Honesty Pledge

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University's ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Name	Date	Signature
Anılcan Atik	25/01/2021	

EXECUTIVE SUMMARY

YELP REVIEW DATASET SENTIMENT ANALYSIS USING MACHINE LEARNING TECHNIQUES

Anılcan Atik

Advisor: Asst. Prof Dr. Evren Güney

JANUARY, 2021, 21 pages

Today, internet review sites are becoming a significant criterion for users' consumption habits on products and services, while being vital source of feedback for businesses. This project aims to present quick feedback on whether consumers are satisfied with businesses' product and services, lessen the allocation of resources on information extraction towards these reviews, and provide a more agile environment for businesses, by automatizing the extraction of the information "whether the sentiment towards the business service or product is positive or negative" from textual data. The problem, binary classification out of textual data, is addressed through Yelp Company reviews dataset. Yelp is an internet review website, it enables users to review products, services, and businesses. Alongside with the text formatted restaurant reviews, star-rating is converted to 1 (positive) and 0 (negative). These values are obtained to provide the target column to predict the sentiment of the review text. 100,000 restaurant review records are used in 4 different machine learning algorithms to predict the binary classification problem of predicting whether the review sentiment is positive or negative. 2 neural networks one with pre-trained GloVe, SVM, and Logistic Regression models are used, and the success of these models is compared using F1-Score as a performance metric. These results are presented in the paper.

Key Words: Yelp reviews, supervised learning, sentiment analysis

ÖZET

YELP YORUMLAR VERİSETİNİN MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK DUYGU ANALİZİ

Anılcan Atik

Proje Danışmanı: Dr. Öğr. Üyesi Evren Güney

OCAK, 2021, 21 sayfa

Günümüzde, yorum ve değerlendirme web siteleri tüketicilerin kullanım alışkanlıklarını gittikçe daha çok şekillendirirken aynı zamanda, çeşitli işletmeler için de önemli bir geri bildirim kaynağı olmakta. Bu proje, tüketici değerlendirmelerinden, “verilen hizmete yada sunulan ürüne yaklaşımın pozitif mi yoksa negatif mi” olduğu bilgisinin elde edilmesinin otomatizasyonunu sağlayarak, bu bilginin analizi için ayrılan kaynakları azaltmayı, işletmelere daha hızlı şekilde kullanıcıların ürün ve servislerden memnun olup olmadığı geri bildirimini sağlamayı ve aynı zamanda işletmelere daha esnek bir iş ortamı sunmayı hedeflemektedir. Metin verisinden ikili sınıflandırma yapılmasını gerektiren bu problem için Yelp şirketinin yorumlar veri kümesi üzerinde çalışıldı. Yelp kullanıcıların içerisine girip çeşitli alanlarda hizmet veren işletmelerin, çeşitli ürün ve hizmetlerin değerlendirilip, yorum yapılabileceği bir internet sitesidir. Bu sitenin restoran yorumları ele alınmış ve metin formatındaki yorumlarla birlikte bu işletmeye verilen yıldız verisi de 1 (pozitif), 0 (negatif) değerlere çevrilmek üzere hedef kolonu olarak kullanılmıştır. 100,000 restorant yorumu kullanılarak 4 farklı eğitimli (supervised) makine öğrenmesi modeli kullanılmıştır. 2 tanesi yapay sinir ağı (biri önceden eğitilmiş GloVe matrisine sahip olmak üzere), SVM ve Lojistik Regresyon modelleri kullanılmıştır. Bu modellerin performansı F1- skoru kullanılarak ölçülmüştür. Ortaya çıkan sonuçlar, bu çalışmada sunulmuştur.

Anahtar Kelimeler: Yelp yorumları, eğitimli öğrenme, duygu analizi

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Amid the rise of the internet and smart devices, exposure to other people’s opinions on businesses and services -while choosing where to visit or what to eat- has become an integral part of our daily lives. Today, internet review sites and platforms like TripAdvisor and Yelp enable users to review products, services, and businesses while becoming a significant criterion for other users’ consumption habits and choices. These review sites are vital sources of information for businesses as well. First-hand experiences of customers can help businesses to address problems while encouraging them to remain in a certain quality range.

1.1. Business Application

As these platforms have become the ultimate collection of criticisms and praises of businesses and services, the need to analyze and deal with the overwhelming number of reviews has also become nearly impossible for both users and businesses to go through all these reviews. Fan & Khademi (2014) highlights that the inconsistent nature of these user-generated reviews (differences in length writing style and usefulness) pose another problem on imposing certain judgment metric for these reviews. One solution these platforms are using is the introduction of a 1-5-star(s) rating system, where along-side the review, each reviewer rates the business one to 5 stars, 5-star being excellent, 1-star being poor. These online ratings have become a crucial part of local commerce. In a study, Luca (2011) highlights that while consumer review sites enhance the information available on the quality of restaurants, a 1-star increase in rating results in an average 5-9% increase in revenue. However, the 5-star rating metric is also prone to reviewers’ criteria for what is ‘excellent’ and what is ‘poor’. Two different users can be very pleased to have the same meal in the same restaurant, while one reviewer could rate this pleasant experience with 3 stars, another reviewer could rate this experience with 5 stars. An alternative method for this bias problem can be sentiment analysis.

Sentiment analysis aims to extract the subjective information from given text by classifying the words in the text into associated categories. It is a widely used method, to get sensible automated feedback from text data.

1.2. Model Definition

In this project, I intend to create a sentiment analysis for the Yelp restaurant reviews dataset and create a sensible model predicting whether the analyzed review is negative or positive. For creating such a model, a labeled data, representing/associating with the sentiment of the review is necessary. The 5-star rating, submitted with review text is a useful reference for this purpose and will be used in this supervised machine learning model, alongside the review text.

1.3. Data

Yelp website is one of the leading online searching/reviewing platforms comprised of various service-industry sectors such as restaurants, home services, and shopping. These reviews are invaluable sources of information in the age of consumerism for users to determine where to visit or what to eat among a variety of available options. The dataset containing real business reviews are released for academic purposes and used in this paper. ("Yelp dataset," n.d.) The JSON formatted data set sizes 4.5 gigabytes compressed, 9.8 gigabytes uncompressed; includes 6,685,900 reviews, 192,609 distinct businesses and 1,223,094 tips by distinct 1,637,138 users. Due to time and computational power constraints, a relatively small portion of the dataset, containing 100,000 rows is randomly selected out of the restaurant reviews dataset.

The sampled dataset contains 46,706 unique businesses (out of 100,000), with the most reviewed business containing 128 reviews. In addition to that, the dataset contains 82,135 unique users, and the user with the most reviews has 55 posts. The dataset comprised of 43,780 5-stars ratings, 21,980 4-stars ratings, 11220 3 stars ratings, 8028 2-stars ratings, and 14992 1-star ratings meaning as shown in Figure-1.

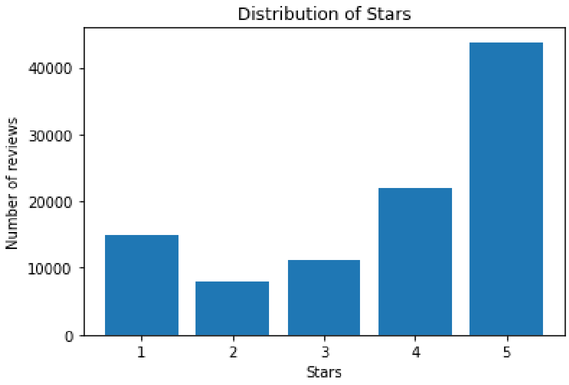


Figure 1: Distribution of Stars in Sampled Dataset

The sampled dataset contains 177,151 distinct words while positive reviews contain 123,905 distinct words and negative reviews contain 94319 distinct words. Figure 2 shows the most frequent words used in the sampled dataset.

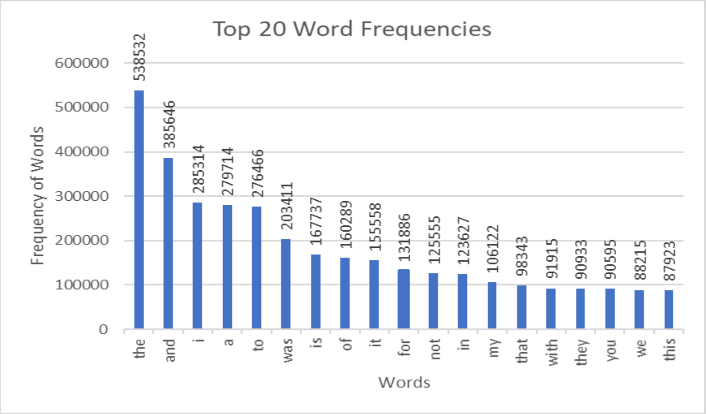


Figure 2: Distribution of Top 20 Words in Sampled Dataset

1.4. Success Metrics

Evaluation of the models and determining the evaluation criteria are fundamental to both business applications and models to become successful and sustainable. The main motivation of this paper is to present quick feedback on whether consumers are satisfied with businesses' product and services, while providing a more agile environment for businesses. For this, minimization of false positives, where although review indicates a negative sentiment, the model predicts the review as positive, poses high importance for businesses. Businesses might not become aware of these negative feedbacks and addressing these problems might be failed. Brand value is a fragile feature and any problem about that should be detected and addressed effectively. On the other hand, false negatives also might be very important to businesses. Positive feedback also poses great importance for businesses to both on the decision to sustain certain features that is getting positive reactions and motivation of the businesses. F1 score, trade of between precision, which is a good metric for limiting false positives, and recall which is preferred metric to limit false negatives presents a good performance metric for this paper.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

Figure 3: F1-Score Formula

As can be seen in the figure 3 above, F1 score is harmonic mean of precision and recall.

2. LITERATURE REVIEW

There are various papers written on a similar problem of determining sentiment out of textual content. In the paper, Mining the peanut gallery, Dave et al. (2003) were able to develop an opinion mining tool and produce fairly good results by distinguishing the poor, mixed, and good results via aggregated opinion pool. The authors introduced a system to find reviews' tags and sentiment scores of both tag and the grouped product. In the research dataset "there were five times as many positive reviews as negative ones, and certain products have many more reviews" they randomly selected an equal number of positive and negative reviews from the four largest categories, and in addition to reviews, they also used the length of the reviews. However, the length of the reviews did not provide any better differentiation. Dave et al. (2003) In the paper, authors used unigrams, bigrams, trigrams, and substitutions to generalize distracting words in different scopes. "Rare words were replaced globally, domain-specific words were replaced for categories and product names were replaced for products."(Dave et al., 2003) These metadata and statistical substitutions were aimed to address the elimination of the overly specific features, whereas "I called Esra" and "I called Ege" would ideally be converted to "I called X" as a product of the substitution process. However, these methodologies did not present significant performance improvements. In addition to substitutions, language-based modifications also attempted to overcome variations and dependencies. Stemming to remove suffixes and identifying negative phrases by turning "not good" into "Notgood" implementations were also failed to improve the performance. The authors noticed that when working on individual sentences, the performance was unsatisfying due to the noise and ambiguity. However, working on grouped sentences under products and specific tags, results were better. The best performance was presented in the Naive Bayes algorithm (%87.0) with Laplace smoothing where the unseen events are assigned to a non-zero probability and known probabilities are converted into less "sharp" values. (Dave et al., 2003)

The work of, Mingming and Maryam also attempted to predict business' star in the Yelp dataset from its reviews' text using regression models. (Fan & Khademi, 2014) Authors first formed a bag of words, picked the top K frequent words, and calculated the frequency of each top K words in all reviews of each business. The authors used 3 different input data formed by 3 different feature engineering methodology. These were; top frequent words directly from the raw text reviews, top frequent words after doing Part-of-Speech analysis, and top frequent adjectives after doing Part-of-Speech analysis on all reviews. They used linear regression, support vector regression, support vector regression with normalized features, and decision tree regression learning methods in each dataset. Instead of using accuracy, they used Root Mean Square Error as the models were limited to regression models. The authors concluded that regardless of the feature engineering methods the best performing model was Linear Regression. Performance appears to be increasing as the number of features (top frequent words used as features) increased. The most performing model was Linear Regression with top frequent words from raw data, with an RMSE score of 0.6. (Fan & Khademi, 2014)

In recent years, convolutional neural networks (CNNs) models also presented promising results for automatic text categorization and sentiment classification. Andrea Salinca (2017) has conducted an empirical study of a word-based CNNs for sentiment classification using the Yelp dataset. In her paper, she used 2 pre-trained CNN models Glove and word2vec's fastText extension for comparison to the previously done traditional machine learning algorithms such as Naive Bayes, SVM. (Salinca, 2017) Through hyperparameter tuning and different architectural settings, Salinca (2017) obtained an accuracy of 95.6% with 3-fold cross-validation on the Yelp dataset. Models proposed in this paper reinforced the notion of CNNs for text classification problems.

In light of the prior work on the sentiment analysis problem, while text cleaning procedures will be held during the pre-processing phase, processes like substitutions, length of the text as a feature, or language-based processing that would alter the text content will not be used. Following that, the performance metric will be "F1-score" because the reviews dataset is imbalanced, consisting of an uneven number of positive and negative reviews while favoring positive, a balance between recall and precision as a metric of success seems to be reasonable.

3. METHODOLOGY

3.1. Pre-processing

The unstructured nature of text data requires certain steps on the finalization of the sentiment analysis model. To reduce noise and improve the performance of machine learning algorithms, certain pre-processing methodologies are followed. These steps are vital in obtaining sensible results by addressing certain problems on the unstructured data. There are a variety of problems in the text format that can create various noises. Firstly, data content noise is tried to be minimized by using only restaurant reviews in our sampled dataset. As the noise can be caused by different keyword choices in reviews from different sectors. For example, hotel reviews might have words/phrases such as ‘television’, ‘fridge’ which would not occur in restaurant reviews. Secondly, various text cleaning procedures are applied. These cleaning functions include removing Html tags/URL links, punctuations, and extra whitespace. Although this dataset was obtained through Yelp, the dataset could be contaminated with HyperText Markup Language (HTML) tags or URL links that do not indicate any meaningful information for our intentions. (Aggarwal, 2018) To address this problem, beautifulSoup Html parser is used. The normalization process is held by altering words’ capitalization, contraction, and ascending characters that are addressed through various functions. For example, “You’re in a café aren’t you” is converted into “you are in a cafe are not you”. Thirdly, criteria for labeling positive and negative reviews are concluded. 1,2,3-star(s) ratings are labeled as negative and 4,5 stars are labeled as positive indicators of the texts’ content. As a result, 65,760 positive content and 34,240 negative content is produced. To model text data, a tokenization process is performed to create vocabulary -a set of unique tokens in corpus-. ("What is tokenization in NLP? Here's all you need to know," 2020) The tokenization and sequence creation phase is done through the TensorFlow tokenizer. During this process vocabulary size, indicating the maximum number of words to keep in vocabulary, including out of vocabulary tokens that were not in the training dataset, is set to 5000. (Chakravarthy, 2020) We will be using 15000 most frequent words to predict our model regarding computational and time constraints. In addition to that, through sequence encoding, text formatted reviews are converted into indexes of the corresponding words. As an example, text like “I liked the pancakes” would be converted into “2 7 1 10”. The max length of these encoded sequences is set to 50. This process is followed by vectorization, unigrams and bigrams are used where the minimum occurrence of bigram is set to be 10. In order to prepare our data for neural network models, pad sequencing is used. Through pad sequencing list of integers (indices of words), obtained by tokenization and vectorization, are converted into 2D integer tensor shapes. While pad sequencing and tokenization process will be used for neural network models, tfid vectorizer will be used for linear classification models. The term frequency-inverse document frequency method tries to rescale the information instead of dropping down the features. It adjusts weights of the features according to their frequency in that particular class, if a word is often found in that particular class but not in the other class then the higher weight is attained to that word (feature). In tfid vectorizer, unigrams and bigrams are adapted, and words that are recurred more than 80 times are included in this vectorization. These limitations adapted in the tfid vectorizer prevents overfitting of the model by removing miswritten or meaningless words that occur in particular classes but do not occur in other documents. As a result of tfid vectorization, 20,130 features will be used for linear models of linear support vector classification and logistic regression.

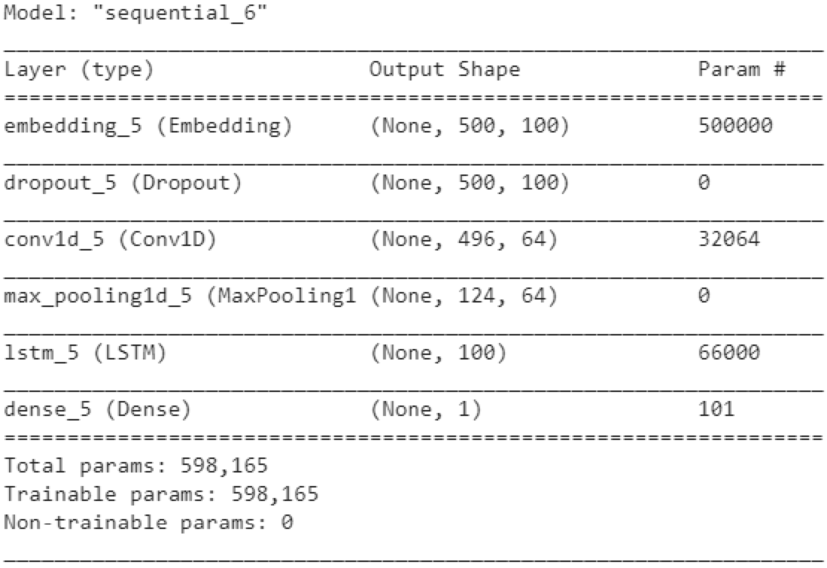
3.2. Models

F1-Score is used as the main performance criteria. To evaluate our results %80 of the data is used as a training-dataset and %20 of the dataset is used for testing. Various machine learning algorithms are used to provide the best model for sentiment analysis of the review's dataset.

To address the binary classification problem, 2 neural network models are used. 2 of these models are mostly differing from the first layer of the embedding layer. Embedding layers, taking word indices while returning corresponding word vectors, try to map words into geometric space in the form of dense vectors. Semantic relationships between words are expected to be more when the geometric distance of (such as L2 distance) two words are less. (Chollet, 2017)

In the first model, embedding layer, dropout layer, 1d convolutional, max-pooling layer, LSTM layer followed by output layers are added. This structure is mostly concluded through literature reviews, regarding the binary classification problems for textual datasets. In figure 3, the summary of model 1 can be seen. Model is getting 500 inputs and produces

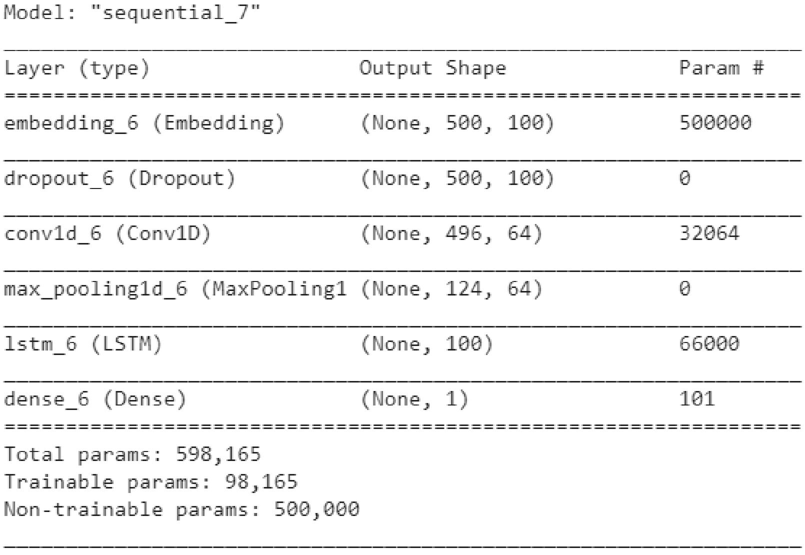
2 outputs, probabilities of 2 classes. Within the total nodes of 598.165, there are no non-trainable parameters as we do not use any pre-trained layers as in model 2.



```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_5 (Embedding)      (None, 500, 100)          500000    
_________________________________________________________________
dropout_5 (Dropout)          (None, 500, 100)          0         
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 496, 64)           32064     
_________________________________________________________________
max_pooling1d_5 (MaxPooling1 (None, 124, 64)           0         
_________________________________________________________________
lstm_5 (LSTM)                (None, 100)               66000     
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 101       
=================================================================
Total params: 598,165
Trainable params: 598,165
Non-trainable params: 0
_________________________________________________________________
```

Figure 4: Architecture of First Neural Network Model

In the second neural network model, embedding layer, dropout layer, 1d convolutional, max-pooling layer, LSTM layer followed by output layers are added. In comparison to model 1, due to pre-trained nodes, figure 4 illustrates that within the total parameters of 598,165, 500000 of them are non-trainable and 98,165 of these parameters are trainable parameters.



```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_6 (Embedding)      (None, 500, 100)          500000    
_________________________________________________________________
dropout_6 (Dropout)          (None, 500, 100)          0         
_________________________________________________________________
conv1d_6 (Conv1D)            (None, 496, 64)           32064     
_________________________________________________________________
max_pooling1d_6 (MaxPooling1 (None, 124, 64)           0         
_________________________________________________________________
lstm_6 (LSTM)                (None, 100)               66000     
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 101       
=================================================================
Total params: 598,165
Trainable params: 98,165
Non-trainable params: 500,000
_________________________________________________________________
```

Figure 5: Architecture of Second Neural Network Model

In the first model, the embedding layer, learns embedding structure by learning through the training data that is provided. The layer will start out with random word vectors and gradually adjust the weights of words in accordance with the fed training dataset. The word embeddings are tried not to take accuracy as a criterion during the formation of weights, rather the layer algorithm uses geometric proximity in accordance with the weighting words and training labels. On the course of training, weights are adjusted with the help of backpropagation.

In the second model, the embedding layer is using a pre-trained embedding matrix created through GloVe pre-trained model. GloVe is an unsupervised machine learning pre-trained algorithm, it tries to generate meaning out of vector representations of words. (Pennington et al., 2014) This unsupervised learning uses Euclidian distance (cosine similarity) between two-word vectors which helps to obtain useful information such as linguistic or semantic similarity of the corresponding words. (Pennington et al., 2014) Training of the model we used in this paper is performed out of aggregated global word-word co-occurrence statistics obtained from 2 Billion tweets, and 27 Billion tokens. (Pennington et al., 2014) Out of 25d, 50d, 100d, and 200d vector parameters for GloVe model. Through GloVe pre-trained weights, the embedding matrix is created and used as weights parameter of the embedding layer, while the trainable parameter is set to false in order to solely rely on GloVe pre-trained weights.

Aside from the first layer changes, the rest of the layers are structurally similar. In order to prevent the model to overfit the training dataset, the dropout layer is adapted. In the training phase, the dropout layer randomly disrupts or “drops out” data fed and stresses nodes to become less susceptible to noise. Due to the existence of the drop-out layer, however, epoch number must be increased as training will take more time to converge into the performing model. (Chollet, 2017) This layer is not activated during the prediction phase, it is only active during the training of the model.

Because we are using sequential, fixed-length segments of data, the introduction of convolutional networks as a layer could be beneficial. One dimensional convolutional neural network model is used. The main motive behind this was to model to consider the sequences of the words as one-dimensional convolutional neural networks consider each sequence as a whole. (Chollet, 2017) The sequence information would be lost in a dense layer while in convolutional networks characteristics of local sequences information are not lost. This means that a local pattern in a certain sequence can be recognized in other sequences. However, the limitation of this structure is that small positional changes in a sequence can result in different pattern recognition. One way to thinning this precise feature map is to reduce the precision of the output.

Max pooling is applied to reduce the size of feature maps, which would be used as an input for the LSTM layer. The maximum pooling method calculates the mean for each feature map, which creates a summarized or ‘downsampled’ version of feature maps. As we can see from both figure 3 and 4 (469, 64) is downsampled to (124,64) matrices.

As the fifth layer, LSTM is used. LSTM layer will add recurrent connections to our model. Long short-term memory (LSTM) is a Neural Network algorithm and it differs from conventional feedforwards neural networks by additional feedback connections. (Hochreiter & Schmidhuber, 1997) Due to the existence of gated functions, LSTM can perform efficiently while avoiding the problem of gradient vanishing. (Ma et al., 2018) In essence, the LSTM layer is aimed to re-introduce the past information later, preventing vanishing-gradient.

LSTM output will be input for a linear layer, the output layer. The output layer is the layer that the goal of the model is stated. In both of the models, the output layer is directly encouraged to match the y, while other inner layers are not directly specified by y. For each x value in training data, f(x) is encouraged to y, while inner layers are configured as training data comes in. As an activation function, a sigmoid is selected. As a loss function, binary cross-entropy is selected as this is a binary classification problem, as an optimization scheme ‘adam’ is selected.

Support Vector Machine (SVM) is a widely used machine learning algorithm in the area of textual polarity. SVMs tries to learn the importance of each of the training data, to create the definition of the decision boundary. In order to predict the class of the introduced variable, the distance between training variables is measured through the Gaussian kernel. In the Gaussian kernel, the Euclidian distance of points are measured, and input gamma is the parameter that controls the width of this distance. On the other hand, another parameter C is used to adjust the regularization in the algorithm. Regularization adjusts the importance of each data point when determining the decision boundary. The higher the C is, the less regularization is implemented. (Müller & Guido, 2016) In the Linear Support Vector Classification model, parameters C is adjusted through intuitive grid searching. In addition to that, as input data, 20,130 features provided by TF-IDF vectorization will be used in the training phase.

Logistic Regression used in this paper is tried to be optimized, through the hyperparameters tuning process. Hyperparameters C indicating inverse of regularization strength, which tunes the strength of regularization, and Penalty indicating the norm used in penalization during regularization is tuned via Grid Search. The l2 regularization with C=10 is found to be the best parameters for our data. In figure 5, the top 25 features that indicate the positive sentiment (painted in blue color) and the top 25 features that indicate the negative sentiment (painted in red color) can be seen. When examined, these

features' coefficient magnitudes are in parallel with their semantic meanings. While features: worst terrible, rude, horrible are expected to present negative meanings, the features: delicious, great, amazing, and perfect indicating positive sentimental meaning is seen as the strongest indicators of the positive sentiment class. In addition to that, aside from unigram features, the bigram features are also presenting as strong features that weights more when predicting the sentiment of the texts. While 'very disappointed' is a strong indicator of negative sentiment, the word 'not disappointed' can be seen as a strong indicator of positive sentiment classification in the figure. On the other hand, features: 'to love', 'bad at', and 'be disappointed' are not in parallel with their semantic meanings compared to their weights.

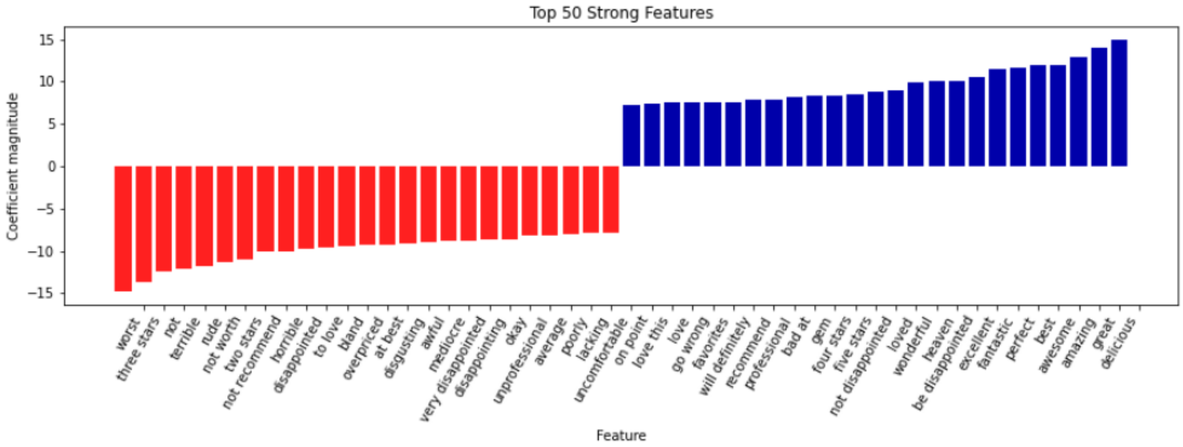


Figure 6: Logistic Regression Top 50 Strong Coefficients of Features

4. EVALUATION AND RESULTS

4.1. Fine Tuning and Optimization

Fine-tuning is the process of adjusting the parameters of the model in order to improve the performance of the model. In all of the models, fine-tuning process is done through grid search or randomized grid search. The main constraint of fine-tuning on neural network models was time. On average each epoch took 5-10 minutes, thus creating a constraint on a number of parameters that can be put into the randomized search. Thus, for the neural network model's epoch and batch size was the main concern during the fine-tuning process. In addition to that, through literature reviews, the range of parameters for fine-tuning is determined.

For model 1 and model 2, the neural network models, 2 parameters: Batch Size = [200, 1000 4000] and epochs = [5, 50] are tested while the randomized search is used. Training data consists of 80000 rows and batch size range is determined in accordance with that, while max epochs were limited to time constraint, as an increase in the epoch was directly multiplying the training time. In the randomized search number of iterations that presented a trade-off between runtime and the optimization of the model were set to 3 while the scoring criteria are set to 'f1'. For model 1 and model 2 batch sizes presented similar results where batch size = 1000, and epochs = 50.

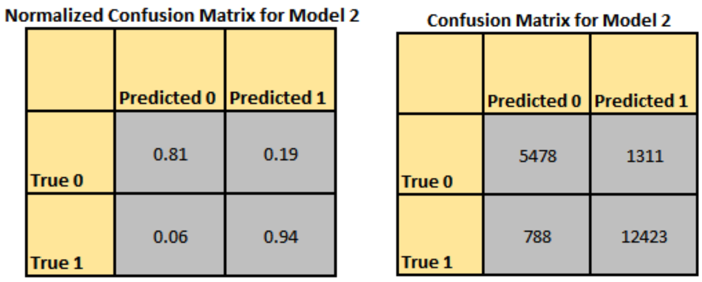
For SVC and Logistic regression, grid search is adapted as the training time was drastically lower compared to neural networks. The best performing parameter relative to 'f1' scoring was C: 50 values within the values [0.1, 20, 50, 100].

For Logistic Regression through the hyperparameters tuning process, it is tried to be optimized. Hyperparameters C indicating inverse of regularization strength, which tunes the strength of regularization, and Penalty indicating the norm used in penalization during regularization is tuned via Grid Search. Values [0.001, 0.01,0.1,1,10,100] is tried for C parameter, while for regularization types (penalty parameter) values ["l1", "l2", "elasticnet", none"] are tried. The l2 regularization which is the default regularization parameter with C=10 is found to be the best parameters for our data in the logistic regression model.

4.2. Evaluation of the Results

Various performance metrics can be used as the fundamental evaluation criteria to assess the success of machine learning algorithms that predict the sentiment orientation of Yelp reviews. As our data was relatively imbalanced with 66% positive and 34% negative reviews, the weighted mean of recall "(how many of the Actual Positives our model capture through labeling it as True Positive) and precision (how many of the predicted positive values are actual positives)", the F1-Score is used as the main performance metric. (Shung, 2020) It accounts for both false positives and negatives whereas the cost of false positives and negatives is higher compared to accuracy. (Shung, 2020)

For model 1, neural network model with no pre-trained embedding layer, accuracy is tested as 0.9 while our main performance metric f1-score is calculated as 0.84. Figure 7 illustrates that the true positive ratio is higher than the true negative ratio. The total number of positive labels in test data is 13211, while the total number of negative labels is 6789.



Normalized Confusion Matrix for Model 2

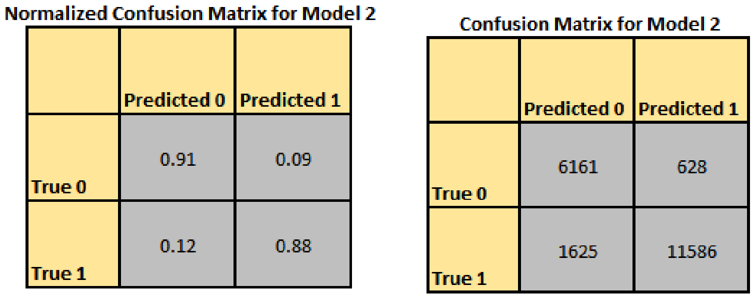
	Predicted 0	Predicted 1
True 0	0.81	0.19
True 1	0.06	0.94

Confusion Matrix for Model 2

	Predicted 0	Predicted 1
True 0	5478	1311
True 1	788	12423

Figure 7: Confusion Matrixes for Model 1

For model 2, neural network model with pre-trained GloVe layer, accuracy is tested as 0.91 while our main performance metric is calculated as 0.86. Table 2 shows the confusion matrixes for model 2, here we can see that although imbalanced dataset, the model's prediction ratios are relatively better for the 'negative' class with 0.91 success whereas the true positive rate is 0.88.



Normalized Confusion Matrix for Model 2

	Predicted 0	Predicted 1
True 0	0.91	0.09
True 1	0.12	0.88

Confusion Matrix for Model 2

	Predicted 0	Predicted 1
True 0	6161	628
True 1	1625	11586

Figure 8: Confusion Matrixes for Model 2

For the linear support vector machine model, accuracy is tested as 0.86 while our main performance metric f1-score is calculated as 0.79. Figure 9 illustrates that the true positive rate is tested as 0.89 while the true negative is tested as 0.8. SVC model is performing relatively better for the prediction of positive sentiments compared to negative sentiments. The count of true negative is 5401, while false negative is 1388, the false positive on the other hand is tested as 1472 and true positive is tested as 11739.

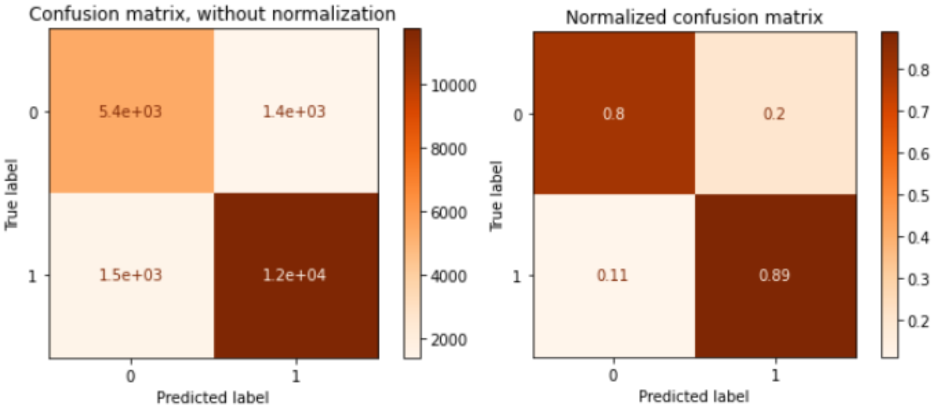


Figure 9: Confusion Matrixes for SVC Model

For the logistic regression model, accuracy is tested as 0.9, while the f1-score is calculated as 0.85. Figure 10 shows, the that true positive rate is 0.94 while the true negative is tested as 0.84. The positive sentiment prediction is performing relatively better compared to the true negative ratio. In addition to that, the count of true positive is 12372 and false positive is 1073, while the count of true negative is 5716, the count of false negative is tested as 839.

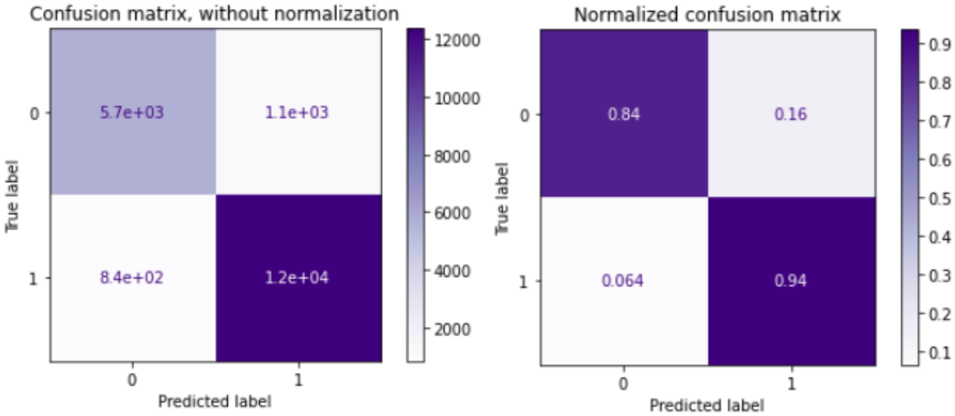


Figure 10: Confusion Matrixes for Logistic Regression Model

During the evaluation, the most successful model was GloVe with 0.86 F1-Score, followed by Logistic Regression with 0.85, followed by SVM with 0.79, and LSTM with 0.84. The pre-trained model GloVe was especially performing well on the prediction of negative sentiments, this might be caused by the pre-trained layer's less exposure to the imbalanced training dataset. In addition to that, table 1 illustrates that while precision values are nearly the same for 3 of the models, the value of precision for the linear SVC was significantly lower compared to the rest of the models. In addition to that, Logistic regression was also performing relatively better on detecting negative sentiments, as its recall value is the second-highest among models.

Table 1: Model Results

Model	Accuracy	Recall	Precision	F1-Score
Model 1	0.90	0.81	0.87	0.84
Model 2 (pre-trained)	0.91	0.85	0.87	0.86
SVC	0.86	0.80	0.79	0.79
Logistic Regression	0.90	0.84	0.87	0.85

4.3. Discussion

Various problems might be posing problems to our models' performances. These problems can be categorized under the quality of data and quality of models. The incorrect words written in reviews can cause additional noise in our models. For example, words 'cozynnrecommendationngo', 'sweeeeeeeeeeeeeet', 'hesitationsnthank' were distinct to positive labeled sentiments whereas words 'minsnnaround', 'capacitynnalthough', 'menusnunfortunately were distinct to negative labeled sentiments. If not eliminated these features can pose overfitting for training data, increase computational requirement and decrease performance. In tfid vectorization process by setting df parameter to 80, the words that recurred less than 80 times are eliminated. While for the pad sequencing vectorization process, the top 5000 words are addressed to eliminate these. Regarding improvement, both parameters could be tuned. In addition to that due to the extensive dataset and number of features, and limited time and computational power, the fine-tuning process is limited to a limited number of parameters. On the other hand, the fine-tuning process could be applied to layer parameters in neural networks, such as in the embedding layer, in addition to the pre-trained GloVe matrix, the parameter ‘training’ could be set as True, which would enable pre-trained parameters to be reshaped. This new model’s performance could be compared to the existing pre-trained neural network model.

5. CONCLUSION

In this project, sentiment prediction of Yelp Company reviews dataset is studied. Yelp platform has become one of the major collection of criticisms and praises of businesses and services, the need to analyze and deal with the overwhelming number of reviews has become nearly impossible for both users and businesses to go through all these reviews. Considering this problem, the project aimed to create a more agile environment for businesses while improving the analysis of feedback from reviews by creating sentiment analyzing models using machine learning techniques.

Text formatted restaurant reviews are used as a feature column while star-rating is converted to 1 (positive) and 0 (negative) to provide the target. The unstructured nature of text data required certain steps on the finalization of the sentiment analysis model. Thus, for algorithms to work properly and improve the performance of the model, certain pre-processing methodologies are followed into randomly sampled 100,000 restaurant reviews.

These records are used in 4 different machine learning algorithms to predict the binary classification problem of predicting whether the review sentiment is positive or negative. 2 neural networks models (one with pre-trained GloVe matrix), SVM, and Logistic Regression algorithms are used, and the success of these models is compared using F1-Score as a performance metric.

During testing, the pre-trained neural network algorithm proved to be the best performing method within 4 algorithms. This model presented a 0.86 F1-Score, followed by Logistic Regression with 0.85. These are followed by a neural network model with no pre-trained embedding matrix presenting 0.84 F1-Score, and linear SVC with 0.79.

REFERENCES

Aggarwal, C. C. (2018). Machine learning for text: An introduction. *Machine Learning for Text*, 22-40. https://doi.org/10.1007/978-3-319-73531-3_1

Chakravarthy, S. (2020, July 10). *Tokenization for natural language processing*. Medium. https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4

Chollet, F. (2017). *Deep learning with Python*. Manning Publications.

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. *Proceedings of the twelfth international conference on World Wide Web - WWW '03*. https://doi.org/10.1145/775152.775226

Fan, M., & Khademi, M. (2014). Predicting a business star in Yelp from its reviews text alone. *arXiv*. https://arxiv.org/abs/1401.0864

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Luca, M. (2011). Reviews, reputation, and revenue: The case of Yelp.com. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1928601

Ma, Y., Peng, H., Khan, T., Cambria, E., & Hussain, A. (2018). Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis. *Cognitive Computation*, *10*(4), 639-650. https://doi.org/10.1007/s12559-018-9549-x

Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: A guide for data scientists*. O'Reilly Media.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://doi.org/10.3115/v1/d14-1162

Salinca, A. (2017). Convolutional neural networks for sentiment classification on business reviews. https://arxiv.org/abs/1710.05978

Shung, K. P. (2020, April 10). *Accuracy, precision, recall or F1?* Medium. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

What is tokenization in NLP? Here's all you need to know. (2020, July 12). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/

Yelp dataset. (n.d.). Restaurants, Dentists, Bars, Beauty Salons, Doctors - Yelp. https://www.yelp.com/dataset/documentation/main

Yelp Sentiment Analysis Google Colaboratory Document. (n.d.). https://colab.research.google.com/drive/1oU4DaAwSJjWJAxFnvLB13eQNmG6evE6s?usp=sharing

APPENDIX A

Diagram 1 illustrates the followed architecture during Yelp reviews dataset sentiment analysis. ("Yelp Sentiment Analysis Google Collaboratory Document," 2020)

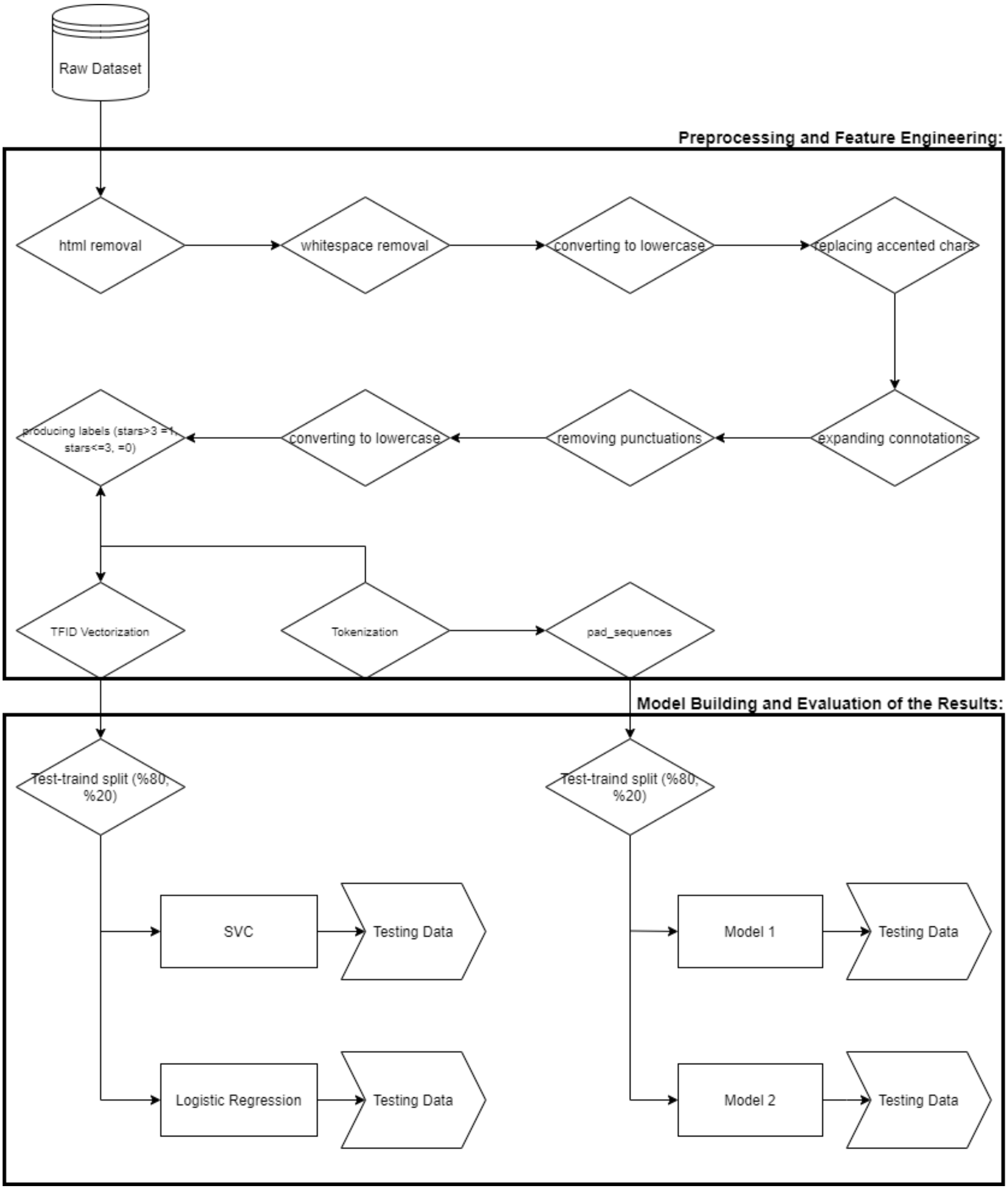
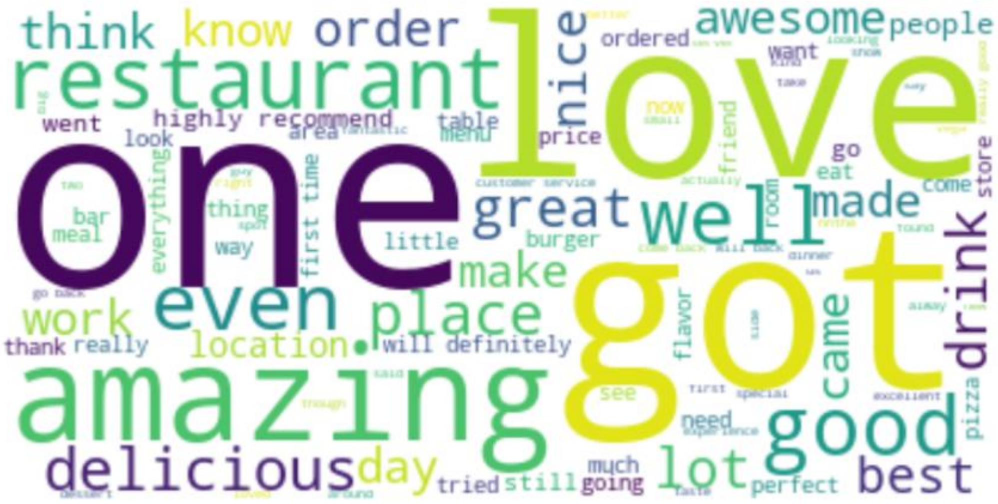


Diagram 1 Pre-processing and Model Building Architecture

APPENDIX B

Diagram 2 illustrates the Word Cloud for positive frequent words used in the Yelp reviews sampled dataset. ("Yelp Sentiment Analysis Google Collaboratory Document," 2020)

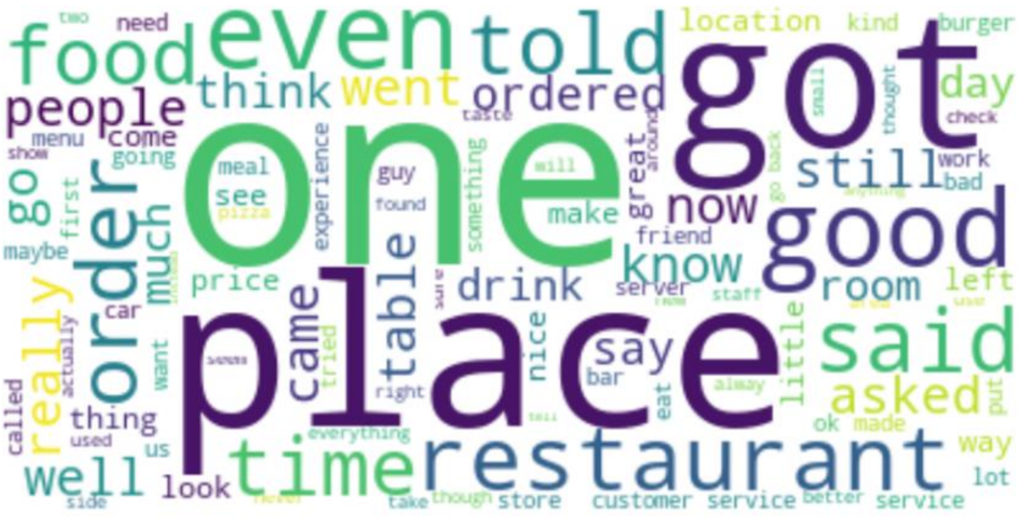
Diagram 2 Word Cloud for Positive Reviews



APPENDIX C

Diagram 3 illustrates the Word Cloud for negative frequent words used in the Yelp reviews sampled dataset. ("Yelp Sentiment Analysis Google Collaboratory Document," 2020)

Diagram 3 Word Cloud for Negative Reviews


one
place
got
restaurant
good
even
food
told
order
time
said
know
people
think
went
ordered
still
now
day
table
came
drink
room
asked
really
well
much
go
say
location
kind
burger
need
two
menu
come
going
meal
see
experience
guy
something
great
make
friend
first
maybe
price
server
staff
left
nice
little
car
bar
eat
put
thing
tried
called
actually
want
right
always
everything
ok
made
way
used
us
look
lot
side
take
though
store
customer service
better
service
show
pizza
found
will
taste
around
small
thought
check
work
bad
go back