Sentiment Analysis of Healthcare Reviews

Rithesh Shetty
Student ID: 24003988
ritheshs@uci.edu

Utkarsh Chhapekar
Student ID: 21681026
uchhapek@uci.edu

***Abstract*—Our approach, as outlined in this report, involves utilizing sentiment analysis to gain insights into how healthcare services are perceived within a particular geographical region. The dataset was scraped from Yelp and preprocessed using Python's JSON library and the Natural Language Toolkit (nltk) library. Three models were trained on the preprocessed dataset: Naive Bayes, Support Vector Machine (SVM), and Transformer-based DistilBERT. The transformer-based DistilBERT model outperformed the other models, achieving an accuracy of 0.977 on the test set. The intended application of the project is to provide valuable insights into the general public's opinion on healthcare in a region, ultimately aiding the identification of areas for improvement and shaping future healthcare policy decisions. The project can be further improved by exploring hyperparameter tuning for the models and collecting a more balanced dataset.**

I. Introduction

The quality, accessibility and affordability of healthcare services can play a significant role in where people decide to settle and establish their lives. One way to measure the effectiveness of healthcare services is through the public opinion of healthcare delivery in a region. Yelp [1], one of the world's most popular review websites, hosts millions of reviews for various businesses, including doctors and healthcare facilities. This project aims to leverage the power of machine learning to scrape raw textual data in the form of doctor and clinic reviews from Yelp and perform sentiment analysis to learn about the perception of care in a region. By aggregating sentiment analysis results across all counties in the US, this project seeks to provide valuable insights to inform decision-making with regards to healthcare policy and provide a metric for the general public to evaluate the healthcare services.

II. Methodology

A. Data Collection

We looked into multiple sources for data collection. One of them was the Healthgrades Reviews API [2]. It has a large corpus of reliable reviews, but the free version of their API could only give us 30 reviews per call and we were limited by the number of available credits. After further deliberation, we decided to use Yelp reviews as our data source. Yelp's free API allowed for 5000 calls per day. We were able to collect around 50000 reviews in the JSON format.

B. Data Preprocessing

After collecting reviews from Yelp using their API, the JSON data needed to be parsed to gather relevant information and create a usable CSV file for easier processing. This was done using the JSON library in Python, resulting in a CSV file with 2 columns: reviews and number of stars. To perform sentiment analysis, the stars column was converted into categories: ratings of 4 and above considered positive, 3 considered neutral, and less than 3 considered negative.

We found that our dataset was biased towards mostly positive and negative reviews, with only a few neutral ones. This is likely due to the tendency of customers to leave reviews only when they have an extremely positive or negative experience. It was an imbalanced dataset. To address this bias and ensure a more balanced dataset, we subsampled the dataset to remove some of the positive and negative reviews.

TABLE I
Effect of balancing data in the dataset

Sentiment	Count	Post-processed Count
1 (Positive)	33394	5000
0 (Negative)	23053	5000
2 (Neutral)	2059	2059

We could also notice that across all the reviews, the frequency of sentiment of positive words were greater than the negative words.

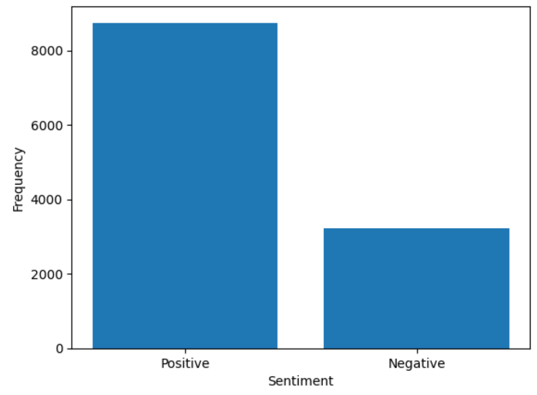
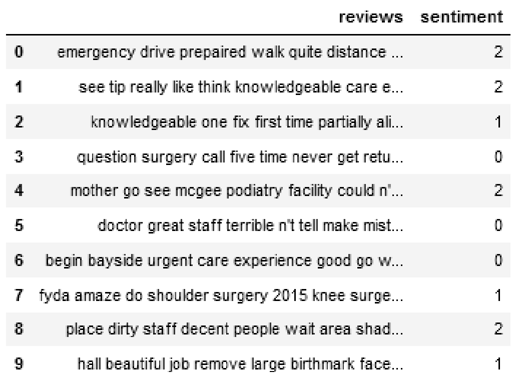


Fig. 1. Frequency of Positive vs Negative words

After the initial data collection, we utilized the Natural Language Toolkit (nltk) library [3] to preprocess the reviews by making everything lowercase, eliminating stopwords, punctuation marks, and special characters. To further refine the data, we utilized the WordNet lexical database to tag each word with its corresponding part of speech and then performed lemmatization on the reviews. This helped us to reduce the inflected forms of the words to their base form, enabling us to obtain a more meaningful representation of the text.



	reviews	sentiment
0	emergency drive prepaired walk quite distance ...	2
1	see tip really like think knowledgeable care e...	2
2	knowledgeable one fix first time partially ali...	1
3	question surgery call five time never get retu...	0
4	mother go see mcgee podiatry facility could n'...	2
5	doctor great staff terrible n't tell make mist...	0
6	begin bayside urgent care experience good go w...	0
7	fyda amaze do shoulder surgery 2015 knee surge...	1
8	place dirty staff decent people wait area shad...	2
9	hall beautiful job remove large birthmark face...	1

Fig. 2. Dataset after nltk preprocessing

III. Model Development

For the Sentiment Analysis task, we tried three models on the preprocessed dataset. Naive Bayes [4] is a probabilistic classification algorithm that works on the principles of Bayes' theorem and assumes the independence of features, making the classification process efficient and fast. It calculates the probability of a text belonging to a particular sentiment class based on the probabilities of the individual words in the text belonging to that class

SVMs [5] create a boundary between classes by finding a hyperplane that maximizes the margin, the distance between the hyperplane and the nearest points of each class. It takes in a new piece of text as input and assigns it to one of the sentiment categories based on the learned patterns

DistilBert [6] is a state-of-the-art pre-trained language model based on the transformer architecture. It can be fine-tuned for various natural language processing (NLP) tasks, including Sentiment Analysis. The model produces an embedding representation for each word in the sentence. These embeddings are then fed into a multi-layer neural network, which predicts the sentiment label based on the context of the text.

A. Naive Bayes

The training process in Naive Baye's is quite simple and fast as it just involves the computation of features' and class probabilities. To generate the features from the textual data (reviews), we used the Bag-of-Words model. Training is done on this embedding of the data.

```
from sklearn.naive_bayes import MultinomialNB

classifierNB = MultinomialNB()
classifierNB.fit(X_train, y_train)

y_nb = classifierNB.predict(X_test)
```

Listing 1. Naive Baye's initialization

Naive Bayes does not have many hyperparameters to tune and it is less critical to do so, as it produces good performance in its default implementation. Therefore, in our development of the model, we did not perform any hyperparameter tuning.

B. Support Vector Machine (SVM)

Training process of the SVM is again performed on the Bag-of-Words embedding of the data. Hyperparameter tuning is very critical for model performance, hence we performed a 3-fold randomized search cross validation to estimate the optimal hyperparameter values for kernel, C, and gamma. Table II shows the values determined, upon which the model was built.

TABLE II
Hyperparameter values for the SVM

kernel	rbf
C	0.806630522
gamma	0.02406242504

```
param_dist = {'C': uniform(0.1, 100), 'gamma':
    uniform(0.001, 1), 'kernel': ['rbf', 'sigmoid']}

svc = svm.SVC()

clf = RandomizedSearchCV(svc, param_distributions=
    param_dist, n_iter=50, cv=3, random_state=42)
clf.fit(X, y)

classifierSVM = svm.SVC(kernel='rbf', C =
    0.8066305219717406
, gamma = 0.024062425041415758
)
classifierSVM.fit(X_train, y_train)

y_hat = classifierSVM.predict(X_test)
```

Listing 2. Hyperparameter tuning and initialization for SVM

C. Transformer-based DistilBERT

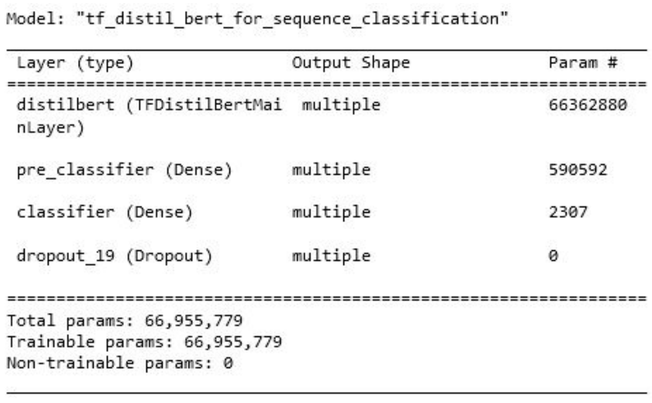
DistilBERT has its own input format/embedding that the processed reviews need to be converted into before the training process: Like other transformer models, training is more of a fine-tuning process and does not require many epochs or number of training samples. DistilBERT has several hyperparameters.

Due to lack of computing resources, the grid search cross validation method was taking too much time and hence, we decided to proceed with default values for the hyperparameters.

```
model.compile(optimizer=tf.keras.optimizers.Adam(
    learning_rate=3e-5, epsilon=1e-08, clipnorm=1.0)
    ,           loss=tf.keras.losses.
    SparseCategoricalCrossentropy(from_logits=True),
              metrics=[tf.keras.metrics.
    SparseCategoricalAccuracy('accuracy')])
```

Listing 3. DistilBERT parameter compilation

We plotted the accuracy and loss curves for this model over 3 training epochs that can be seen in Fig 4.



```
Model: "tf_distil_bert_for_sequence_classification"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 distilbert (TFDistilBertMai  multiple                 66362880
 nLayer)

 pre_classifier (Dense)      multiple                  590592

 classifier (Dense)          multiple                  2307

 dropout_19 (Dropout)        multiple                  0

=================================================================
Total params: 66,955,779
Trainable params: 66,955,779
Non-trainable params: 0
_________________________________________________________________
```

Fig. 3. The model summary for DistilBERT

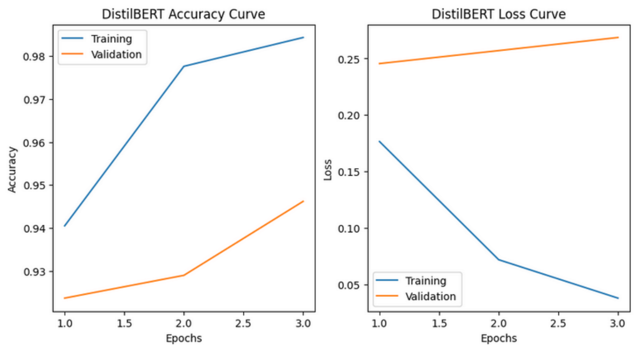


Fig. 4. Accuracy and Loss curves for DistilBERT

IV. Results

From the metrics shown in Table III, it is clear that the transformer-based DistilBERT model greatly outperforms the SVM and Naive Bayes models and should be our model of choice for the sentiment analysis task.

We also plotted confusion matrices for the above models to better show the visual representation of the different model's performance (Fig 5 - 7).

A. Intended application

By assimilating reviews on a region-by-region basis and analyzing them using a sentiment analysis model, we can generate a a perception score from 1 to 10 through a weighted sentiment computation. This valuable metric can help decision-makers better understand the strengths and weaknesses of healthcare services in different regions, allowing them to make informed decisions about resource allocation and service improvement.

TABLE III
Comparison of model performance on test data using classification metrics

Model/Metrics	Accuracy	Precision	Recall	F1-score
Naive Bayes	0.80292	0.80006	0.80292	0.80137
SVM	0.78938	0.77790	0.78938	0.76180
DistilBERT	0.97705	0.97711	0.97705	0.97707

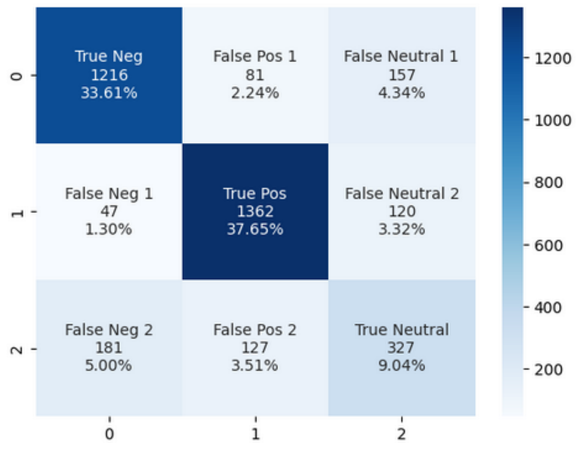


Fig. 5. Confusion Matrix for Naive Baye's on the test set

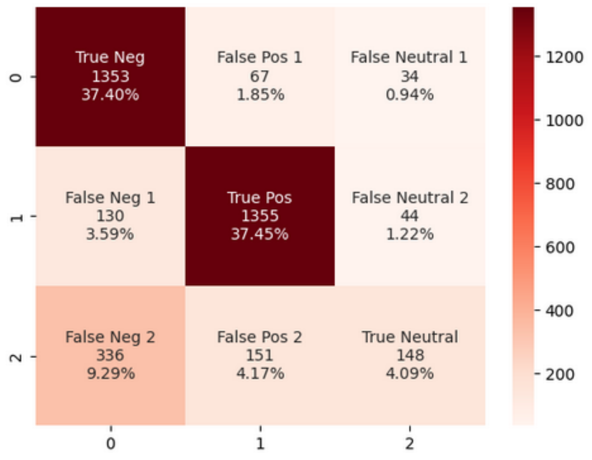


Fig. 6. Confusion Matrix for SVM on the test set

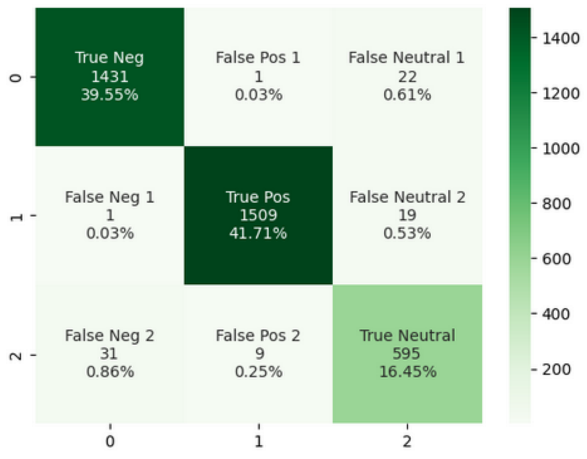


Fig. 7. Confusion Matrix for DistilBERT on the test set