Tribhuvan University
Institute of Science and Technology

Seminar Report
On
"Sentiment Analysis of Twitter Data Using Logistic Regression"

Submitted to

Central Department of Computer Science and Information Technology
Tribhuvan University, Kirtipur
Kathmandu, Nepal

In the partial fulfilment of the requirement for Master's Degree in Computer Science and Information Technology (M. Sc. CSIT) First Semester

Submitted by

Raju Shrestha
Roll No. 48/2079
June, 2023

Tribhuvan University

Institute of Science and Technology

Supervisor's Recommendation

This is to certify that Mr. Raju Shrestha has submitted the seminar report on the topic **"Sentiment Analysis of Twitter Data Using Logistic Regression"** for the partial fulfilment of Master's of Science in Computer Science and Information Technology, first semester. I hereby, declare that this seminar report has been approved.

____

Supervisor

Asst. Prof. Mr. Bikash Balami

Central Department of Computer Science and Information Technology

Letter of Approval

This is to certify that the seminar report prepared by Mr. Raju Shrestha entitled **"Sentiment Analysis of Twitter Data Using Logistic Regression"** in partial fulfilment of the requirements for the degree of Master's of Science in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Evaluation Committee

…………….…………………………..

Asst. Prof. Sarbin Sayami

(H.O.D)

Central Department of Computer Science

and Information Technology

………..………………………………

Asst. Prof. Bikash Balami

(Supervisor)

Central Department of Computer Science

and Information Technology

……………………………

(Internal)

Acknowledgement

The success and final outcome of this report required a lot of guidance and assistance from many people and I am very fortunate to have got this all along the completion. I am very glad to express my deepest sense of gratitude and sincere thanks to my highly respected and esteemed supervisor **Asst. Prof. Bikash Balami,** Central Department of Computer Science and Information Technology for his valuable supervision, guidance, encouragement, and support for completing this paper.

I am also thankful to **Asst. Prof. Sarbin Sayami**, HOD of Central Department of Computer Science and Information Technology for his constant support throughout the period. Furthermore, with immense pleasure, I submit by deepest gratitude to the Central Department of Computer Science and Information Technology, Tribhuvan University, and all the faculty members of CDCSIT for providing the platform to explore the knowledge of interest. At the end I would like to express my sincere thanks to all my friends and others who helped me directly or indirectly.

Raju Shrestha(48/2079)

Abstract

Sentiment Analysis (also known as opinion mining or emotion AI) is a method for judging somebody's sentiment or feeling with respect to a specific thing written in a piece of text. It is used to recognize and arrange the sentiments communicated in writings. The web-based social networking sites like twitter draws in a huge number of clients that are online for imparting their insights in the form of tweets or comments. The tweets can be then classified into positive, negative, or neutral. In the proposed work, logistic regression classification is used as a classifier and unigram as a feature vector.

Keywords: Sentiment analysis, Opinion mining, Text classification, Unigrams, Polarity, Machine learning, Logistic regression, Natural Language Processing.

Table of Contents

Acknowledgement iii

Abstract iv

List of Figures vii

List of Abbreviations viii

Chapter 1: Introduction 1

1.1 Overview 1

1.1.1 Defining Sentiment 1

1.1.2. Characteristic of Tweets 1

1.2 Problem Statement 2

1.3 Objective 2

Chapter 2: Background Study and Literature Review 3

2.1 Background Study 3

2.1.1 Sentiment Analysis 3

2.2 Literature Review 3

Chapter 3: Methodology 5

3.1 Data Collection 5

3.2 Data Preprocessing 6

3.3 Feature Extraction 7

3.4 Logistic Regression 7

Chapter 4: Implementation and Testing 9

4.1 Description of Major Function 9

4.1.1 Preprocessing 9

4.1.2 Feature Extraction 10

4.1.3 Model Training 11

4.2 Testing 11

4.3 Model Evaluation 11

Chapter 5: Result and Findings 13

5.1 Evaluation Metrics 13

5.2 Model Performance 14

Chapter 6: Conclusion 15

References 16

Appendix 17

List of Figures

Figure 3.1: System Architecture..3

Figure 3.2: Twitter Sentiment Analysis Dataset..6

Figure 4.1: Data Preprocessing………..10

Figure 4.2: Feature Extraction………..10

Figure 4.3: Model Training…………..10

Figure 4.4: Train-Test Split…………..11

Figure 4.5: Model Evaluation………..12

Figure 5.1: Confusion Matrix ………..13

Figure (a): Sentiment Analysis Result...17

List of Abbreviations

API: Application Programming Interface

CSS: Cascading Style sheet

CSV: Comma Separated Values

HTML: Hypertext Markup Language

NLP: Natural Language Processing

NLTK: Natural Language Toolkit

POS: Part of Speech

URL: Uniform Resource Locator

Chapter 1: Introduction

1.1 Overview

In the present days, Micro blogging has become a very popular communication tool among Internet users. Many users share tweets or messages everyday on prevalent sites, for example,

Twitter and Facebook. Authors of those messages write about their life, share opinions on variety of topics and discuss current issues. Because of a free format of messages and an easy

accessibility of micro blogging platforms, Internet users tend to shift from traditional communication tools (such as traditional blogs or mailing lists) to micro blogging services. As more and more users post about products and services they use, or express their political and religious views, micro blogging websites become valuable sources of people's opinions and sentiments. Such data can be efficiently used in research, business or social science.

Sentiment is positive or negative reviews about product or on any topics. We people can identify tweets by reading whether it is positive or negative. But if there is huge data to be read then it would be tedious and time consuming. So, if all this process could be done with the help of automated program then it would be easier and above manual process could be eliminated.

Sentiment Analysis of Twitter Data Using Logistic Regression is a web-based application which takes tweets as an input and gives sentiment value as an output.

1.1.1 Defining Sentiment

A sentiment is defined as a view or opinion that is expressed. It is a feeling of someone that he/she expresses either in textual or verbal form. A sentiment can be defined as a personal positive or negative feeling. For example: This is the best budget smartphone. This is positive sentiment. And, this phone have bad resolution is consider as negative sentiment.

1.1.2. Characteristic of Tweets

Twitter message have many unique attributes [1] which are as follows: Tweets and Length: Tweets are the status posted by user which is of 280 length.

Username: Username in twitter is started with @followed by text and number. Eg @barackobama

1.2 Problem Statement

Every day millions of data is being collected on social media like twitter which contains people opinion about many things like the product and services they use, political and religious views etc. And, the data is unstructured and not organized in a pre-defined manner. These text are usually difficult, time-consuming and expensive to analyze, understand, and sort through.

Many company wants to know how positive or negative peoples are about their product and services. People want to know other people how much positive or negatives tweets he/she tweet. Tweets Sentiment Analysis Using Logistic Regression Algorithm will provide the positive or negative sentiment on tweets that people have tweeted.

1.3 Objective

The main objectives of this seminar report is to classify the tweets into positive or negative using logistic regression.

Chapter 2: Background Study and Literature Review

2.1 Background Study

Natural Language Processing (NLP) is a dynamic field of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language. It encompasses a wide range of tasks, from basic text processing like language translation and sentiment analysis to more complex applications such as chatbots, speech recognition, and language generation. NLP algorithms employ techniques from machine learning, linguistics, and computer science to bridge the gap between human communication and computational systems. With its rapid advancements and real-world applications across industries like healthcare, finance, and customer service, NLP plays a pivotal role in re-shaping how humans interact with machines and access information, making it a vital area of study and innovation in today's digital age.

2.1.1 Sentiment Analysis

Sentiment Analysis, also known as opinion mining, is a Natural Language Processing (NLP) technique that involves the automated analysis of textual data to determine the emotional tone or sentiment expressed within it. By utilizing algorithms and machine learning models, sentiment analysis classifies a piece of text into categories such as positive or negative based on the underlying sentiments conveyed by words, phrases, or a piece of text. This is incredibly useful for businesses and organizations as it helps them to make decisions based on public opinion, improve customer experiences, and respond to trends in the digital world. Sentiment analysis is widely used in areas like market research, customer feedback analysis, social media monitoring, and brand reputation management. It's a crucial tool for extracting valuable insights from the vast amount of text data available today.

2.2 Literature Review

Sentiment analysis have become the growing area in the field of natural language processing. Supervised machine learning algorithm like Logistic Regression algorithm plays vital role in the sentiment analysis. There are many researched carried out for sentiment analysis.

In paper [2] studied various technique for sentiment analysis for the movies review. They compare the different classification algorithm like Naive Bayes classification, Maximum

Entropy classification and Support Vector Machine. They also consider the different factor affecting the sentiment like unigrams, bigrams, Part of speech (POS) etc. They achieved accuracy of above 80% for all three algorithm using unigrams + bigrams.

In paper [3] have focused mainly on the Naive Bayes classifier. They take the baseline for their research as [2]. They display the result on pie chart for positive, negative and neutral for the specific keyword.

In paper [4] have focused on topic based classification based on the Logistic Regression. They also have used the confusion matrix as a classifier model. They achieved the accuracy of 92% for the tweets classification into selected topics.

In paper [5] have proposed research based on Logistic Regression. They have used Logistic Regression as classifier and unigram as a features vectors. For increasing the accuracy K-fold cross validation and tweet subjectivity is used. To further speed up the classification process they also use the idea of effective word score heuristics that find out the polarity score of the words which are frequently used.

In paper [6] have proposed research based on logistic regression. They have used bag of words (BOW) as a feature vector and TF-IDF as a features extractor to emphasize the importance of specific word within the document. They achieved that TF-IDF based model shows better performance compared to BOW based model.

In paper [7] have studied various machine learning approaches such as Naïve Bayes, Maximum Entropy, Support Vector Machine and Semantic Analysis (WordNet). They have achieved better performance while using Semantic Analysis.

Supervised machine learning classifier required the trained data set to work. For this I have used publicly available labeled dataset.

Chapter 3: Methodology

Sentiment Analysis of twitter data uses supervise machine learning algorithm i.e. Logistic Regression. Logistic Regression requires labeled data for training the classifier. The detail architecture of the system is presented below.

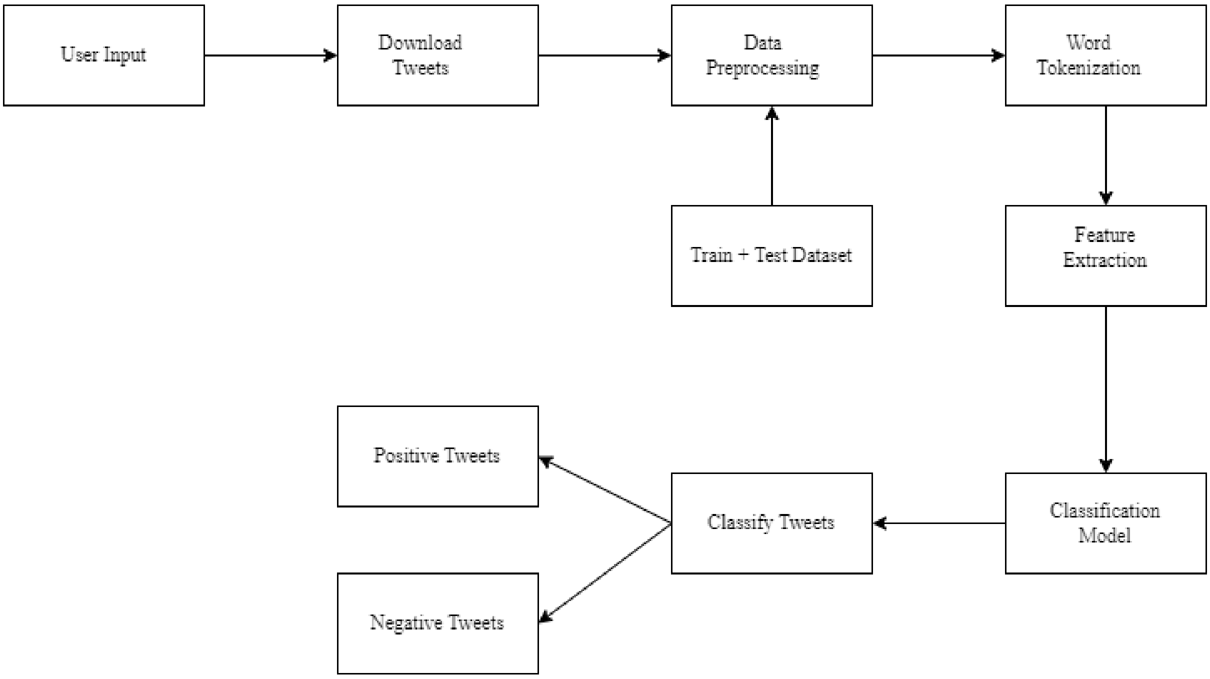
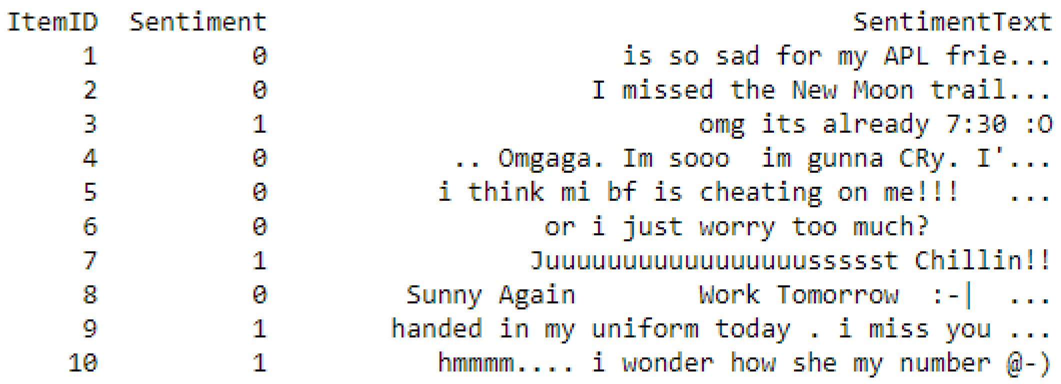


Figure 3.1: System Architecture

3.1 Data Collection

The dataset used for Sentiment Analysis of Twitter Data Using Logistic Regression is Twitter Sentiment Analysis dataset which is collected from the publicly available source i.e. from Kaggle, an online repository known for its diverse collection of data. The dataset consists of 1,00,000 training data in csv format and is labeled 0 for negative and 1 for positive. So, this project uses only the positive and negative datasets. The data sample dataset used in this project is shown below.



```
ItemID  Sentiment                                       SentimentText
     1          0                           is so sad for my APL frie...
     2          0                         I missed the New Moon trail...
     3          1                                omg its already 7:30 :O
     4          0                .. Omgaga. Im sooo  im gunna CRy. I'...
     5          0               i think mi bf is cheating on me!!!   ...
     6          0                      or i just worry too much?
     7          1                     Juuuuuuuuuuuuuuuuussssst Chillin!!
     8          0             Sunny Again        Work Tomorrow  :-|  ...
     9          1            handed in my uniform today . i miss you ...
    10          1               hmmmm.... i wonder how she my number @-)
```

Figure 3.2: Twitter Sentiment Analysis Dataset

3.2 Data Preprocessing

The twitter data consist of different properties in which most of it is not useful for sentiment analysis. Data preprocessing includes various step.

1. Usernames: Twitter consists of username which consist of symbol @ at the beginning.eg @sparkingroshan. It is replaced by the word AT_USER in data sets which is started by @ in the datasets.

2. Usages Link: User includes the link in the tweets for the more detail information which is not useful for sentiment analysis. The link is replaced by the word ‘URL’.

3. Stop Words: Stop word are those filler words which are not useful in for sentiment. These words includes most repeated word like a, an, the, for, etc. These words does not give any sentiment hence they are filtered out form the datasets.

4. Removing Hash-tags: Hash-tag in a tweet is used to emphasize a particular word or sentence, for example #thisisgood. Removal of these hash-tags is important because these hash-tags do not define any sentiment. Thus pre-processing is done and hash-tags before any word are removed.

5. Repeated Letters: Tweets contain the very causal language [3] so the word such as hurrayyy is replaced with actual word hurray. The letter repeated more time reduced to the one.

6. Stemming: Change a word in the text into its base term or root term. Example, happiness to happy.

3.3 Feature Extraction

After preprocessing the tweets, tweets is converted into feature vector. Feature vector are the most important concept in implementing classifier [8]. Feature vector is used for building the model and is used to train the model which is further used to classify the unseen data. Feature vector is the n-dimensional vector of numerical features that represent the some object. In tweets we can consider the presence or absence of words that appear in the tweets. The tweets in training data is split into words and each words into feature words. The feature words may consist of words unigram or bigrams. This project consider unigram as feature words. For eg. This is the ball is represented as this, is, the, ball as unigrams. The entire feature vector will be the combination of each of this feature words.

3.4 Logistic Regression

The algorithm used is Logistic Regression. Logistic Regression is predictive analysis model based on binary classification. It classify the tweets based on the probability given to tweets belong to that particular class. To predict the tweets into positive and negative. I have used label dataset with probability value 0 for negative and 1 for the positive tweets.

Logistic Regression is a discriminative model which means computing P(y|x) by discriminating among the different possible values of the class y based the given input x. The equation for this is as shown below:

$$P(C|x) = \sum_{i=1}^{N} w_i . f_i$$

To generate a value of P(y|x) of an output that is in between value 0 and 1, the following exp function is used:

$$P(C|x) = \frac{1}{z} exp \sum_{i} w_i . f_i$$

To change the normalization factor Z and specify the number of features as N is as follows:

$$P(C|x) = \frac{\exp(\sum_{i=1}^{N} w_i . f_i)}{\sum_c \exp(\sum_{i=1}^{N} w_i . f_i)}$$

The final equation for computing the probability of y being of class c given x is:

$$P(C|x) = \frac{\exp(\sum_{i=1}^{N} w_i . f_i\ (c, x))}{\sum_{c\prime\in C} \exp(\sum_{i=1}^{N} w_i . f_{i(c',x)})}$$

Chapter 4: Implementation and Testing

This section describes the technologies used in Tweets Sentiment Analysis Using Logistic Regression Algorithm. Tweets Sentiment Analysis Using Logistic Regression Algorithm is a web application that uses flask python framework. HTML, Twitter Bootstrap CSS, JavaScript are used to develop front-end and Python, NLTK, Scikit-learn and Matplotlib are used to develop back-end. HTML is used for presentation technology. JavaScript are implemented to show the result of the application in a dynamic way.

All the algorithms for the application are written in Python. Algorithms used in Sentiment Analysis of Twitter Data Using Logistic Regression is Predictive analysis model. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables. The algorithm is implemented using python programming language.

4.1 Description of Major Function

The major function in the application are:

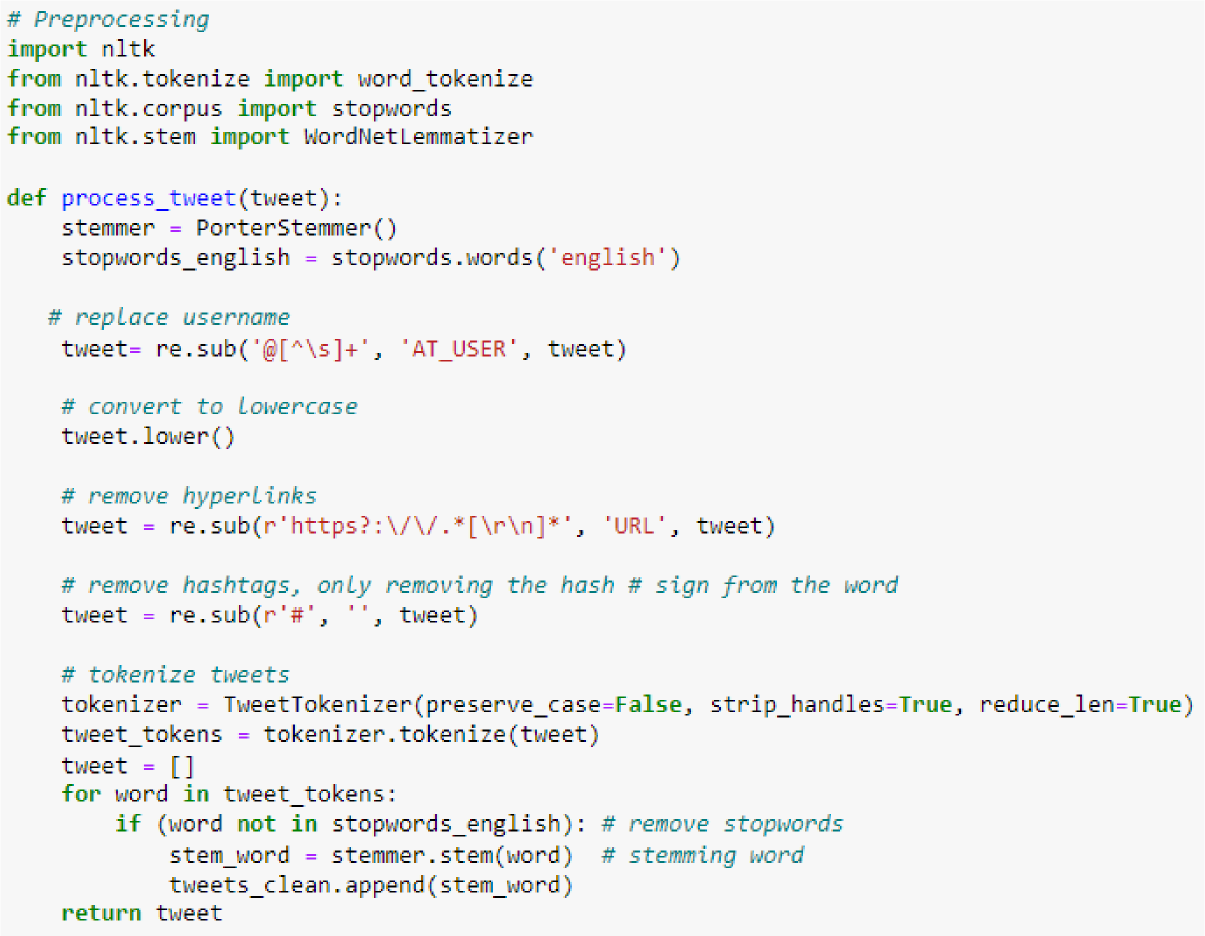
4.1.1 Preprocessing

This is the function which is run for processing the tweets.

Input: It takes the inputs as tweets.

Process: It call other function like remove URL, filter stop words, etc.

Output: It gives the list of the process tweets.



```
# Preprocessing
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

def process_tweet(tweet):
    stemmer = PorterStemmer()
    stopwords_english = stopwords.words('english')

   # replace username
    tweet= re.sub('@[^\s]+', 'AT_USER', tweet)

    # convert to lowercase
    tweet.lower()

    # remove hyperlinks
    tweet = re.sub(r'https?:\/\/.*[\r\n]*', 'URL', tweet)

    # remove hashtags, only removing the hash # sign from the word
    tweet = re.sub(r'#', '', tweet)

    # tokenize tweets
    tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True, reduce_len=True)
    tweet_tokens = tokenizer.tokenize(tweet)
    tweet = []
    for word in tweet_tokens:
        if (word not in stopwords_english): # remove stopwords
            stem_word = stemmer.stem(word)  # stemming word
            tweets_clean.append(stem_word)
    return tweet
```

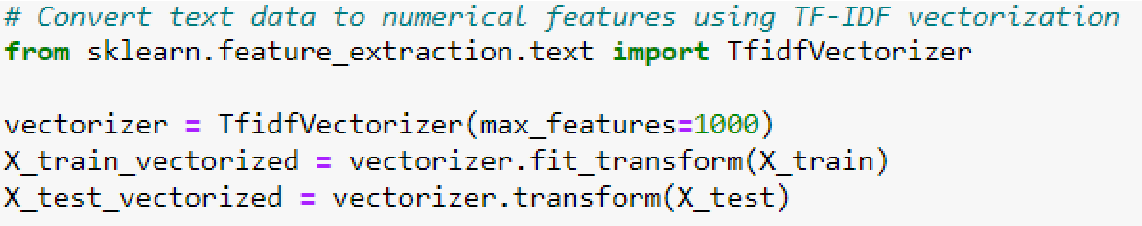
Figure 4.1: Data Preprocessing

4.1.2 Feature Extraction

This function is implemented after the preprocessing data.

Input: This takes pre-processed data as in input.

Process: It then uses the method, extract_feature() to process the taken input, process and extract the feature.



```
# Convert text data to numerical features using TF-IDF vectorization
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=1000)
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)
```

Figure 4.2: Feature Extraction

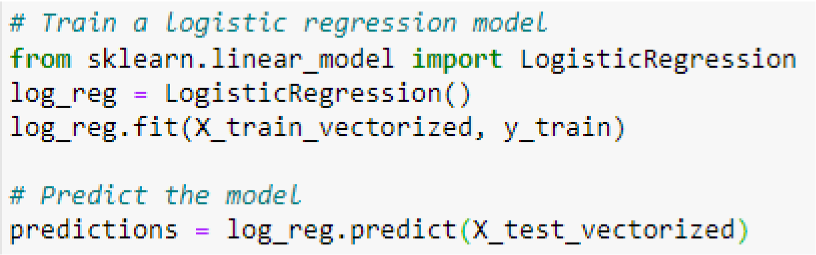
4.1.3 Model Training

After feature extraction the model is trained using logistic regression algorithm, after that model is used for prediction.

Input: It takes input from the feature extractor.

Process: It classify the tweets as positive or negative and return the list of tweets with its sentiment value.

Output: It gives classified tweets with positive and negative tweets value.



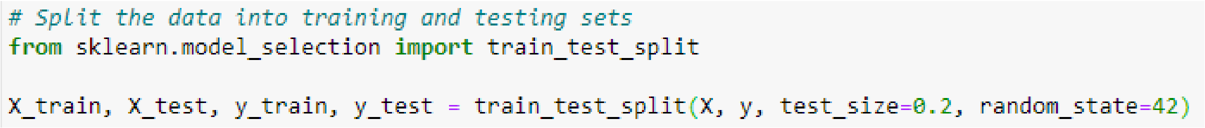
```
# Train a logistic regression model
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression()
log_reg.fit(X_train_vectorized, y_train)

# Predict the model
predictions = log_reg.predict(X_test_vectorized)
```

Figure 4.3: Model Training

4.2 Testing

Among the total data 80% of the data is used for training and 20% is used for testing. For testing hit and trial method is followed and the testing module from Scikit-learn is used for testing.



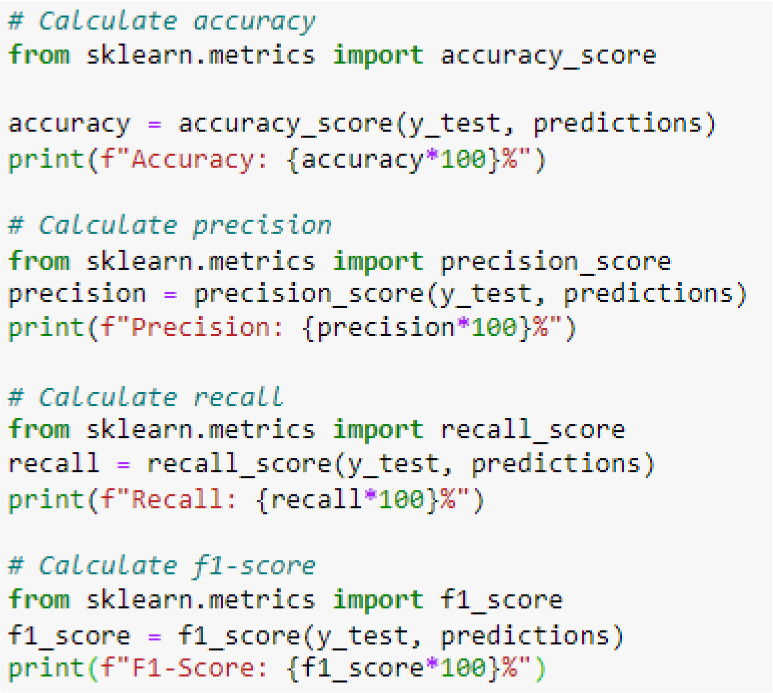
```
# Split the data into training and testing sets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 4.4: Train-Test Split

4.3 Model Evaluation

After training, the model’s performance was evaluated on the test dataset. Metrics such as accuracy, precision, recall and f1-score were calculated from the confusion matrix to assess its effectiveness in classifying positive and negative sentiments.



```
# Calculate accuracy
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test, predictions)
print(f"Accuracy: {accuracy*100}%")

# Calculate precision
from sklearn.metrics import precision_score
precision = precision_score(y_test, predictions)
print(f"Precision: {precision*100}%")

# Calculate recall
from sklearn.metrics import recall_score
recall = recall_score(y_test, predictions)
print(f"Recall: {recall*100}%")

# Calculate f1-score
from sklearn.metrics import f1_score
f1_score = f1_score(y_test, predictions)
print(f"F1-Score: {f1_score*100}%")
```

Figure 4.5: Model Evaluation

Chapter 5: Result and Findings

5.1 Evaluation Metrics

Confusion Metrics: The confusion metrics provides a detailed breakdown of the model's predictions, showing the number of true positive, true negative, false positive, and false negatives.

The confusion matrix results provide a detailed view of the classification performance of the model. In this specific scenario, the model correctly identified 9216 instances as true positives, indicating that it accurately recognized as positive sentiment. It also correctly classified 5820 instances as true negatives, signifying its ability to correctly identify as negative sentiment. However, the model had 2930 false positives, indicating instances where negative sentiments were mistakenly classified as positive. Additionally, there were 2032 false negatives, highlighting cases where positive sentiments were misclassified as negative.

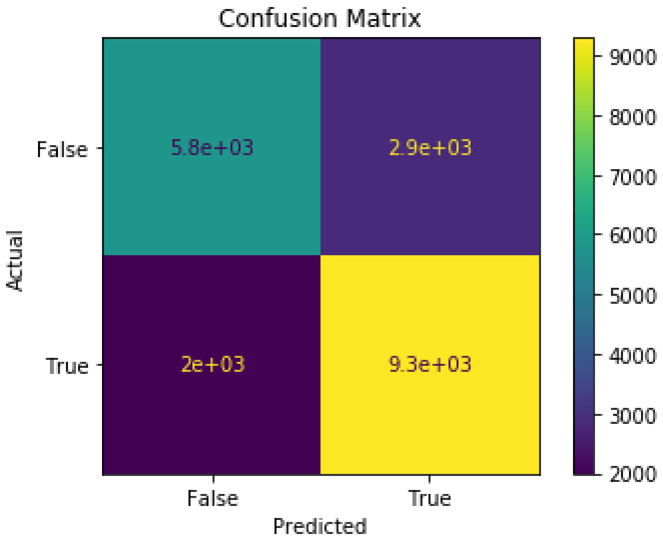


Figure 5.1: Confusion Matrix

Accuracy: Since we are working on a classification problem of classifying the tweets (positive and negative). Accuracy will be a good evaluator for the models. This is most commonly use metric to evaluate how well the model predicts. Accuracy is the ratio of the number of correct predictions made against the total number of prediction made.

$$\text{Accuracy} = \frac{(\text{True Positive} + \text{True Negative})}{\text{Total Number of Samples}}$$

Precision: Precision calculates the ratio of correctly predicted positive instances to the total number of instances predicted as positive by the model. It assesses how well the model performs when it predicts a positive sentiment.

$$\text{Precision} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Positive})}$$

Recall: Recall measures the model's ability to correctly identify positive instances (true positives) out of all the actual positive instances. It is also known as Sensitivity or True Positive Rate.

$$\text{Recall} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Negative})}$$

F1-Score: The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall, especially when you want to find a single metric to evaluate your model's performance.

$$\text{F1} - \text{Score} = \frac{2 * \text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

5.2 Model Performance

The performance of Tweet Sentiment Analysis Using Logistic Regression is measured. These metrics indicate that the model achieved a quit low accuracy in correctly classifying sentiments as positive or negative. The precision score reflects the model's ability to correctly identify positive and negative sentiments. The recall score indicates its ability to identify all the actual positive sentiments present in the dataset. F1 score indicates the model's ability to achieve both accurate positive sentiment identification and comprehensive capturing of actual positive sentiments

- Accuracy: 75.18%
- Precision: 75.87%
- Recall: 81.93%
- F1-Score: 78.78%

Chapter 6: Conclusion

Tweets Sentiment Analysis Using Logistic Regression Algorithm was successfully implemented using python programming language. Twitter is undoubtedly a place where most people express their thoughts and feelings. It is very important to have an appropriate model in prediction of positive or negative tweets. The accuracy of the model on training data set is 75.18% which is quite low and can be improved by providing more datasets.

References

[1] A. Go, R. Bhayani and L. Huang, "Twitter Sentiment Classification using Distant Supervision," *CS224N project report, Stanford* , p. 2009, 2009 Dec.

[2] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," *arXiv preprint cs/0205070,* 2002 May 28.

[3] P. Waykar, K. Wadhwani, P. More and A. Kollu, "Sentiment Analysis of Twitter tweets using supervised classification technique," *International Journal of Engineering Research and Applications,* pp. 32-34, 2016.

[4] I. S.T, L. Wikarsa, B. MComp, R. Turang and S. MKom, "Using logistic regression method to classify tweets into the selected topics," *international conference on advanced computer science and information systems (icacsis),* pp. 385-390, 2016 Oct.

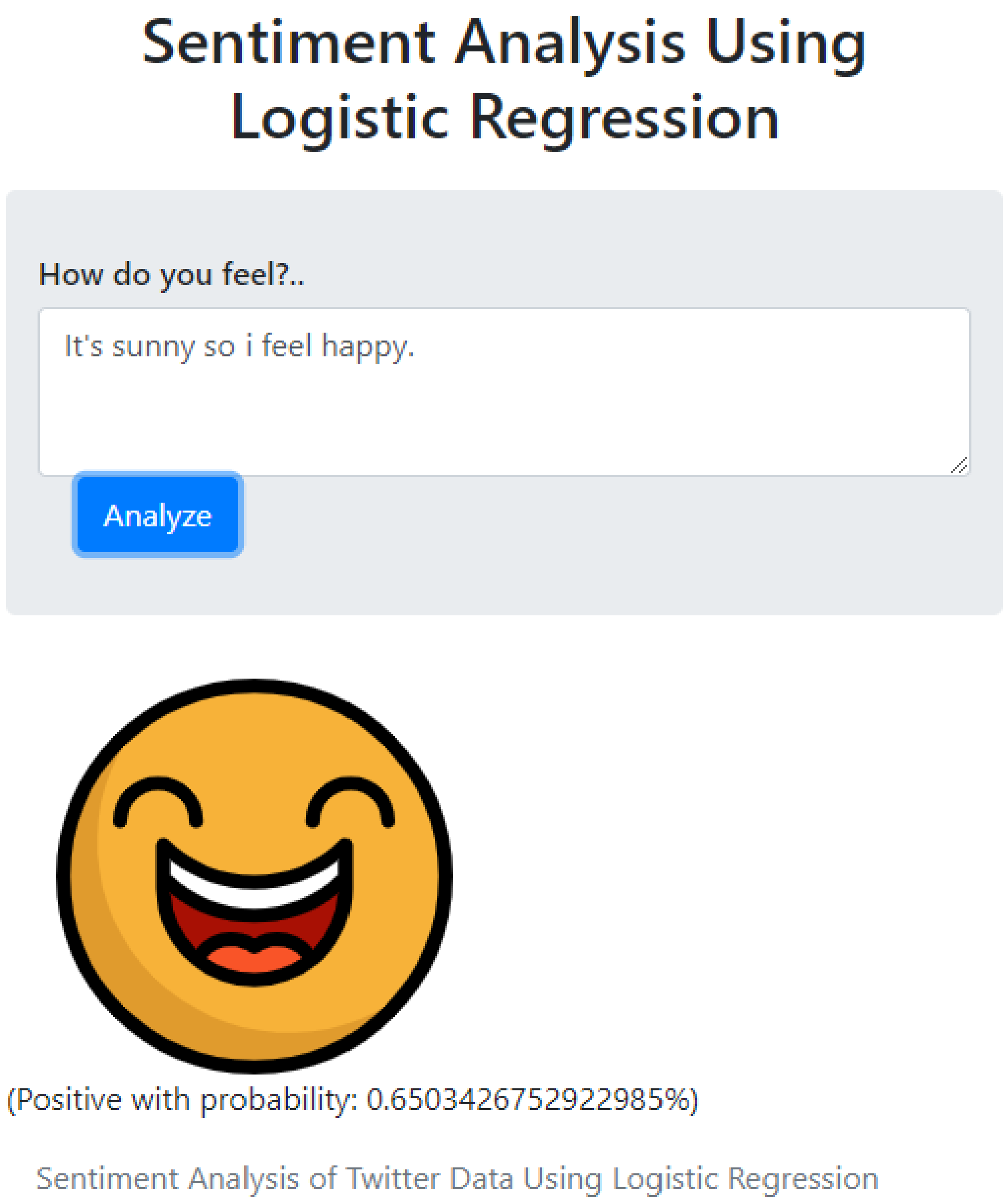
[5] A. Tyagi and N. Sharma, "Sentiment analysis using logistic regression and effective word score heuristic," *International Journal of Engineering and Technology (UAE),* pp. 20-23, 2018 Apr.

[6] A. Antim, "Application of ML in Text Analysis: Sentiment Analysis of Twitter Data Using Logistic Regression," 5 April 2023.

[7] G. Gautam and D. Yadav , "Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis," *2014 Seventh international conference on contemporary computing (IC3),* pp. 437-442, 2014.

[8] R. Janardhana, "how to build a twitter sentiment analyzer?," 8 May 2012. [Online]. Available: https://ravikiranj.net/posts/2012/code/how-build-twitter-sentiment-analyzer/.

Appendix



Sentiment Analysis Using Logistic Regression

How do you feel?..

It's sunny so i feel happy.

(Positive with probability: 0.6503426752922985%)

Sentiment Analysis of Twitter Data Using Logistic Regression

Figure (a): Sentiment Analysis Result