**APPENDIX 1**

**Twitter Sentiment Analysis using Machine Learning Algorithms in python**

### A PROJECT REPORT

#### Submitted by

**HASHIM HAYATH BASHA**

#### in partial fulfillment for the award of the degree of

B. TECH

**IN**

## ARTIFICIAL INTILLIGENCE

## &

## DATA SCIENCE

## CHENNAI INSTITUTE OF TECHNOLOGY

## ANNA UNIVERSITY: CHENNAI 600 025

##### November & 2022

**APPENDIX 2**

**ANNA UNIVERSITY: CHENNAI 600025**

**BONAFIDE CERTIFICATE**

Certified that this project report **“Twitter Sentiment Analysis using Machine**

**Learning Algorithms in python”** is the bonafide work of “**HASHIM HAYATH**

**BASHA(210420243024) , MAURUS (2104202430..) and KISHORE VS**

**(210420243036)”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**HEAD OF THE DEPARTMENT SUPERVISOR**

<<Academic Designation>>

<<Department>>

Department of Artificial intelligence and data science <<Department>>

<<Full address of the Dept & Colle

**APPENDIX 3**

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO. ABSTRACT iii**

**LIST OF TABLE xvi**

**LIST OF FIGURES xviii**

**LIST OF SYMBOLS xxvii**

* 1. **INTRODUCTION**
  2. **RELATED ARTICLES**
  3. **RELATED WORK**
  4. **PROPOSED METHODOLOGY**
  5. **SYSTEM SPECIFICATION**
  6. **INPLEMENTATION AND RESULTS**
  7. **MODEL CREATION**
  8. **CONCLUSION AND FUTURE SCOPE**

## ABSTRACT

Twitter is a platform used by people to express their opinions and display sentiments on different occasions. Twitter is one of the non-traditional data sources with unlimited potential. It contains a large reserve of data sets which are easier to access and collect when compared to others. One of the available tools is twitter API which let us to collect data and various other information about the tweets. The tweets are collected through Tweepy and the tweets are being labelled through Text-Blob. To increase the processing of the tweets are done. The accuracy we will create an vocab using W2R (word to vector) model containing all the related words. The words are vectorized with the relationship among them. This makes the prediction easier. Then through implementation of CNN (conventional neural network) we insert the embedding layer which is formed with vectorized words and a LSTM (long-short-term-memory) layer. The accuracy value of the model is created and we can analyze the tweets of the particular domain.

**ACKNOWLEDGEMNET**

I thank the almighty,for the blessings that have been showered upon me to bring forth the success of the project would like to express my sincere gratitude to our chairman **Shri.P.SHRIRAM**, and all trust members of Chennai Institute of Technology for providing the facility and opportunity to do this project as a part of our undergraduate course.

We thank our Principal **Dr.A.RAMESH M.E.. Ph.D,** for his valuable suggestion and guidance for the development and completion of this project.

We sincerely thank our Head pf the Deparment **Dr.S.PAVITHRA M.E..** **Ph.D.,** Associate Professor, Department of Computer Science Engineering for having provided us valuable guidance, resources and timely suggestions through our work.

We sincerely thank our project guide **Dr BA sir M.TECH., Ph.D..**

Professor, Department of Compute Science Engineering for having provided us valuable guidance, resources and timely suggestions through our work

.

I express our deep sense of thanks to all faculty members in my department for their cooperation and interest shown at every stage of our endeavor in making a project work success.

Last but not least, our sincere thanks to our lovely parents and friends who had been the constant source of our strength throughout our life.

## LIST OF FIGURES

**FIGURE NO** **NAME OF THE FIGURES** **PAGE NO**

## LIST OF TABLES

**FIGURE NO NAME OF THE FIGURES PAGE**

1. SOFTWARE SPECIFICATIONS
2. HARDWARE SPECIFICATION
3. SAMPLE DISTRIBUTIONS OF CLASS

## LIST OF ABBREVIATIONS

**S. NO ABBREVIATIONS EXPANSION**

1.

2.

## SYSTEM SPECIFICATION

### SOFTWARE REQUIREMENT

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO | SOFTWARE | VERSION | URL |
| 1 | ANACONDA | 3.7 | https://www.anaconda.com/ |
| 2 | NUMPY | 1.11.3 | https://numpy.org/ |
| 3 | KERAS | 2.3.0 | https://keras.io/ |
| 4 | TENSOR FLOW | 2.0 | https://www.tensorflow.org/ |
| 5 | MATPLOTLIB | 3.5 | https://matplotlib.org/ |
| 6 | NLTK | 3.9 | https://www.nltk.org/ |
| 7 | PANDAS | 3.9 | https://pandas.pydata.org/ |
| 8 | TWEEPY | 4.12.1 | https://pypi.org/project/tweepy/ |
| 9 | WORLDCLOUD | 1.8.2.2 | https://pypi.org/project/wordcloud/ |
| 10 | GENSIM | 4.2.0 | https://pypi.org/project/gensim/ |

#### ANACONDA

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.). that aims to simplify package management and deployment. Package versions are managed by the package management system anaconda. The Anaconda distribution includes data-science packages suitable for Windows, Linux and MacOS.

#### TENSORFLOW

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library. and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

#### KERAS

Kera’s is an open-source neural-network library written in ion li la capable of running on top of TensorFlow, Microsoft Cognitive Toolkit. It focuses on being user-friendly, modular, and extensible. Kera’s contains numerous implementations of commonly used neural-network building blocks such layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.

#### NUMPY

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

#### MATPLOTLIB

Matplotlib is a comprehensive library for creating static, animated. They are interactive visualizations in python.

#### PANDAS

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make the working with “relational” or “labeled” data both easy and intuitive. the name is derived from the term panel data, that is an econometrics term for multidimensional structured data sets.

**TWEEPY**

Tweepy is an open source Python package that gives you a very convenient way to access the Twitter API with Python. Tweepy includes a set of classes and methods that represent Twitter's models and API endpoints, and it transparently handles various implementation details, such as: Data encoding and decoding.

**WORLDCLOUD**

Wordcloud is basically a visualization technique to represent the frequency of words in a text where the size of the word represents its frequency. In order to work with wordclouds in python, we will first have to install a few libraries using pip.

**GENSIM**

Gensim = “Generate Similar” is a popular open source natural language processing (NLP) library used for unsupervised topic modeling. It uses top academic models and modern statistical machine learning to perform various complex tasks such as − Building document or word vectors.

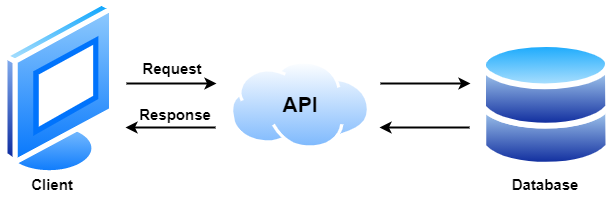
### HARDWARE REQUIREMENT

|  |  |
| --- | --- |
|  |  |
| PROCESSOR | INTEL CORE |
| OPERATING SYSTEM | WINDOWS 11(64-BIT) |
| RAM | 8 GB |
| HARD DISK | 40 GB |
| MONITOR | 15 VGA COLOUR |
| MOUSE | LOGITECH |
| GRAPHICS | DIRECTX 9 WITH WDDM 1.0 |

**DATASET DESCRIPTION**

### Proposed System

Here we will implement our system from the data collection process itself. We will be using twitter API for data collection for sentiment analysis of twitter data. Major obstacle in twitter analysis is domain dependence and world knowledge. To overcome these, we will collect data directly from the topics where the tweets are tested.



We will be collecting 1.6 M data from the twitter API. These datasets are divided into training datasets and testing datasets. Using the Online Text Blob library, we will append the texts along with their respective polarity of the texts. Before we Use the Text blob we will convert the emojis in the text and convert them into the appropriate meanings. This is one of the reasons why twitter is considered one with the most potential to develop the Natural Language processing where we can include other information present in the tweets.

The Preprocessing of data does not end here as they are various steps involved in making the data adaptable to our model that is being created. The first step always practiced in NLP is Removal of Stop words which are commonly used words to make sure phrase structure does not get affected. Then Tokenization is done where the text is splitted to array of words. Then Stemming is done to reduce each word in the array to reduce it to their respective root words. After the preprocess is done the dataset is divided into testing dataset and training dataset.

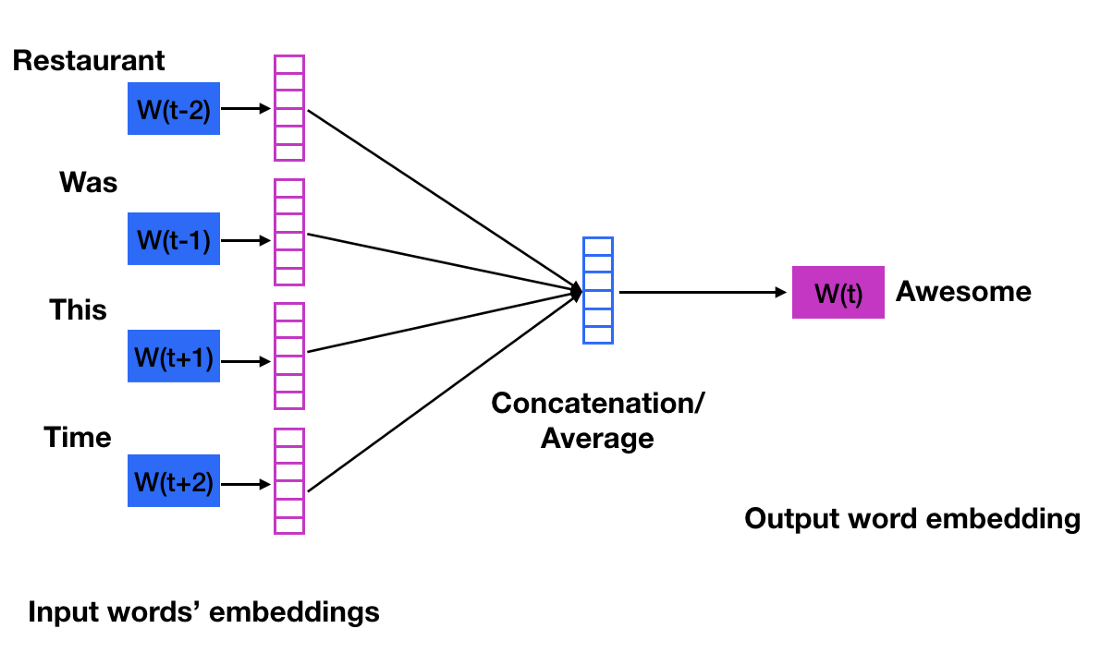
#### Word2Vec Model

Word Embeddings

Word embeddings are words that have been translated into real number vectors in a way that preserves their semantic meaning. The techniques used in my earlier entries of BOW and TFIDF failed to capture the meaning between the words; instead, they treated each word separately as a feature. This model considers the words around a given word inside a specific window size when training it. The term "embedding vectors" can be derived in a variety of ways. One such approach is Word2vec, which makes use of a neural embeddings model to learn that.

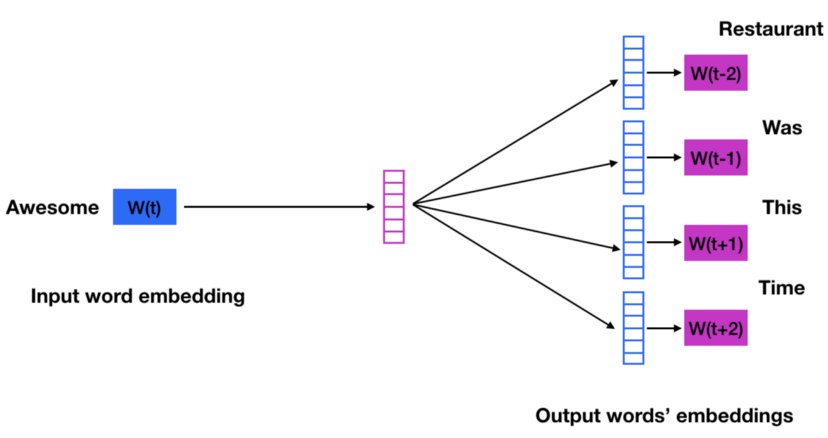
CBOW

Here the model predicts the word under consideration given context words given within a specific window. The hidden layer has the number of dimensions in which the current word needs to be represented at the output layer.



Skip Gram

Skip gram is opposite of CBOW where it predicts embeddings for the surrounding context words in the specific window given a current word. The input layer contains the current word and the output layer contains the context words.



#### Word2Vec Vectors

Word2Vec vectors are generated for each iteration by traversing through the training dataset. By simply using the model on each word in the training dataset we will be able to obtain the word embedding for those words in tweets. We will implement the average over the vectors of words in the sentence that will represent the sentence.

#### Convolutional Neural Network

Here after creating the embedding layer using the word2vec vectors, it is added as one of the layers in the CNN then a Dropout layer will be added with a value of 0.5. The dropout layer is added which helps prevent overfitting. Inputs not set to 0 are scaled up by 1/(1 – rate) such that the sum over all inputs is unchanged. Finally, an LSTM layer is being introduced along with a dense layer. The dense layer used to reduce the dimensionality of the output. In the end of the CNN modeling along with the Word2Vec vectors word embeddings we get the polarity or the sentiment of the tweets.

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

==========================================================

embedding (Embedding) (None, 300, 300) 12659700

dropout (Dropout) (None, 300, 300) 0

lstm (LSTM) (None, 100) 160400

dense (Dense) (None, 1) 101

==========================================================

Total params:12,820,201

Trainable params: 160,501

Non-trainable params: 12,659,700

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This classification and label as Proposed System. This classification and labeling of tweets make it more polarized towards the specific domain and the respective world knowledge through our suggestion and the way of preparation of our data.

### LITERATURE SURVEY

In CNN for situations understanding based on sentiment analysis of twitter data by Shi yang ,Liaoa, Junbo Wangb, Ruiyun Yua, Koichi Satob, Zixue Cheng they proved CNN's ability to extract a set of features from a global dataset is the primary justification for using it in image analysis and classification. information, and it can take into account how these qualities relate to one another. The aforementioned method can increase the accuracy in classification and evaluation. Text data features can also be piecemeal collected for natural language processing in order to Take into account how these elements interact, but without taking into account context or the entire sentence, the sentiment be misunderstood. We create a straightforward convolutional neural network model and evaluate it using the Results indicate that it performs more accurately in classifying Twitter sentiment than certain standard methods, including similar to the SVM and Naive Bayes techniques.

In A sentiment analysis study for twitter using the various model of convolutional neural network. Machine learning produces poor accuracy that is used in a larger range of applications. Consequently, the deep learning approach is being developed to increase sentiment analysis' accuracy. In order to increase accuracy performance, this study discusses several configuration options based on deep learning utilizing the Convolutional Neural Network (CNN) algorithm. To evaluate the effectiveness of the CNN models, a variety of parameters are offered, including the number of convolutional layers, the quantity of filters, and the size of the filters. The Word2Vec model for Indonesian has been utilized with the Indonesian-Sentiment-Analysis-Dataset, which consists of 10.806 tweets, as a word vector representation. The remaining 20% of the dataset is used for testing once the CNN models have been trained on 80% of it. Results from the proposed CNN models are compared and shown to be superior.

In Twitter sentiment analysis using distributed word and sentence representation by Dwarampudi Mahidhar Reddy, Dr. N V Subba Reddy and Dr. Prema K V they introduced usage of Long Short-Term Memory (LSTM) Networks and Convolutional Neural Networks (CNNs)for Distributed Representation of words. Which lets us to capture Capturing local co-occurrence statistics and gives good performance with small (100-300) dimension vectors that are important for downstream tasks. It makes the process faster as only non-zero counts matter. Instead of utilising conventional techniques or preparing text data, this research uses distributed representations of words and phrases. While the latter is used for the distributed representation of sentences, the first two are utilized for the distributed representation of words. This document has an accuracy rate of up to 81%. Out of the various techniques, it also offers the best and most effective approaches to generate distributed.

In Using Word2Vec to Process Big Text Data by Long Ma and Yanqing Zhang they proposed a method to decrease the dimension of the feature vector which will be used to make our embedding layer for the CNN model which we prepare. Processing large data sets takes time since they can have a variety of distinct data types and sophisticated structures in addition to their large volume of data. If the learning algorithm can choose useful features or reduce the feature dimension, it will be more effective when taking the data dimension into account. Continuous Bag of Words (CBOW) and Skip-gram are the two learning models that make up Word2Vec. Word2Vec creates word vectors from text data that can be represented as a substantial passage of text or perhaps the complete article. In our work, we trained the data using a Word2Vec model and then assessed the degree of word similarity.

In addition, we clustering the similar words together and use the generated clusters to fit into a new data dimension so that the data dimension is decreased which will decrease time consumption and increase performance.

### EXISTING SYSTEM

The present work on sentiment analysis can be categorized from a variety of angles, including the method employed, the perspective on the text, the amount of text analysis detail, the rating level, etc. We identified machine learning, lexicon-based, statistical, and rule-based techniques from a technological perspective.

By training on a known dataset, the machine learning method employs different learning algorithms to ascertain the sentiment. The lexicon-based method involves determining a review's emotion polarity based on the semantic orientation of its words or sentences. A text's subjectivity and viewpoint are measured by its "semantic orientation."

The rule-based technique scans a document for opinion words before classifying it according to the proportion of positive and negative words. It takes into account a variety of classification criteria, including dictionary polarity, negation and boosting words, idioms, emoticons, and mixed viewpoints, among others.

Each review is represented by statistical models as a combination of latent features and ratings. In order to cluster head words into aspects and sentiments into ratings,

it is believed that aspects and their ratings can be represented by multinomial distributions. Another categorization is focused primarily on the organization of the text classification.

At the document, phrase, or word level. Sentence- or word-level classification can express a sentiment polarity for each sentence in a review or even for each word, as

opposed to document-level classification, which seeks to identify a sentiment polarity for the entire review.

According to our research, the majority of approaches concentrate on the document-level. Additionally, we may distinguish between techniques that aim to score a review globally versus techniques that gauge the strength of sentiment for various parts of a product.

Most approaches to global review classification that depend on machine learning simply take into account the polarity of the review (positive/negative) into account. More linguistic variables, such as intensification, negation, modality, and discourse structure, are used in solutions that aim for a more precise classification of reviews

(Such as three- or five-star ratings).

A thorough classification of existing approaches is shown in Figure 1. This grouping is not exhaustive. One solution can be used in multiple categories.

1. Rule or Lexion based approach - (how it works):

It counts number of positive or negative values in denote their polarity and sentiment to calculate the score.

- (Disadvantages):

* It does not care about combination of words
* quick but need constant maintence

1. Automated or Machine Learning approach

- Traditional Models:: gathering of a dataset with examples for positive,

negative, and neutral classes, processing data finally training the algorithm

Deep Learning Models::

* Naive Bayes sentiment analysis
* Deep Learning

# Sentiment analysis using NLP deep learning are able to learn patterns through multiple layers from unstructured and unlabeled data to perform sentiment analysis.

1. Hybrid approach
2. Named Entity based Sentiment Analyzer

Sentiment Analysis based on top named entities Targetted by finding sentences containing the named entities and performing sentiment analysis only on those sentences one by one.

### OBJECTIVE

To put into practice an algorithm that automatically categorizes material as positive, negative, or neutral.

Sentiment analysis is used to evaluate whether the general public has a

good, negative, or neutral view toward the topic at hand.

Native Models does not include world knowledge and domain specific considerations here in this model we use tweets of a specific category so that we can make more accurate predictions of sentiments.

Twitter sentiment analysis allows us to keep a track of what’s being said about your product/service on social media, and can help to detect angry customers or negative mentions before they escalate.

### USE CASES

Twitter sentiment analysis provides many exciting opportunities. Being able to analyze tweets in real time, and determine the sentiment that underlies each message, adds a new dimension to social media monitoring.

### APPLICATIONS

* Social Med ia Monitoring
* Customer Service
* Market Research
* Brand Monitoring

### CHALLENGES FOR SENTIMENT ANALYSIS

* Implicit Sentiment and Sarcasm
* Domain Dependency
* Thwarted Expectations
* World Knowledge and Pragmatics
* Subjectivity Detection
* Entity Identification
* Negation

**REFERENCES**

* **https://liris.cnrs.fr/Documents/Liris-6508.pdf**
* [**https://towardsdatascience.com/getting-started-with-data-collection-using**](https://towardsdatascience.com/getting-started-with-data-collection-using) **twitter-api-v2-in-less-than-an-hour-600fbd5b5558**