**RSIP CAREER PLUS AI 003**

**PROJECT TITLE**

**SENTIMENTAL ANALYSIS OF TWITTER DATA USING DEEP LEARNING**

**CATEGORY : DEEP LEARNING**

**SKILLS REQUIRED : PYTHON, PYTHON WEB FRAME**

**WORKS,NLP.**

**BY**

**A.Susitha**

**A.Swetha**

**B.Yalini**

**R.Nagalakshmi**

**M.Varshini**

**TWITTER SENTIMENT ANALYSIS USING DEEP LEARNING**

**INTRODUCTION:**

**Overview**

The analysis of sentiment on social networks, such as Twitter or Facebook, has become a powerful means of learning about the users’ opinions and has a wide range of applications. Twitter has been growing in popularity and nowadays, it is used every day by people to express opinions about different topics, such as products, movies, music, politicians, events, social events, among others. Twitter sentiment classification aims to classify the sentiment polarity of a tweet as positive, negative or neutral. In this project, we develop a deep learning system for Twitter sentiment classification.

**Purpose**

The study of public opinion can provide us with valuable information. Sentiment Analysis, also called Opinion Mining, is an useful tool within natural language processing that allow us to identify, quantify, and study subjective information. Due to the fact that quintillion of bytes of data is produced every day, this technique gives us the possibility to extract attributes of this data such as negative or positive opinion about a subject, also information about which subject is being talked about and what characteristics hold the persons or entities expressing that opinion.

**LITERATURE SURVEY:**

This section summarizes some of the scholarly and the research works in the field of deep learning to analyse sentiments on the Twitter and preparing prediction model for various applications. As the available social platforms are shooting up, the information is becoming vast and can be extracted to turn into business objectives ,social campaigns, marketing and other promotional strategies. The benefit of social media to know public opinions and extract their emotions are considered.

**Problem:**

Every day massive amount of data is being generated by social media users which can be used to analyze their opinion about any event, movie, product or politics. Thus in order to trace it, an automated process of analysing text data and sorting into positive, negative or neutral is proposed.

**THEORITICAL ANALYSIS:**

**Block Diagram:**

Getting Data Twitter

from

Feature Extraction

Tweets Preprocess

get

Evaluation

Classifier

Feature vector

**Software Designing**

There are a high number of frameworks that can be used for machine learning tasks, however, we are going to use ***Keras*** because it offers consistent and simple APIs, minimizes the number of user actions required and more importantly, it is easy to learn and use. We will also make use of the ***Natural Language Toolkit (NLTK),*** that provides many corpora and lexical resources that will come in handy for tagging, parsing, and semantic reasoning, and ***Scikit-learn***, that provides useful tools for data mining and data analysis.

**EXPERIMENTAL INVESTIGATION:**

**Text Processing**

**Steps**

1.Gathering data

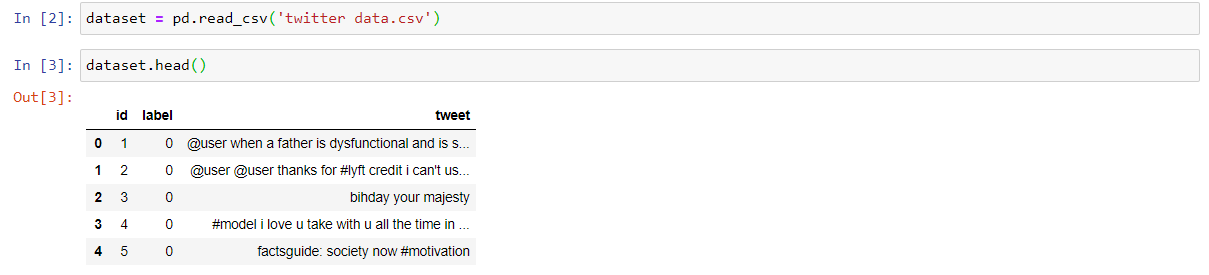
2.Import the Dataset

3.Text Cleaning or Pre-processing

* Remove Punctuations, Numbers
* Convert each word into its lower case
* Stemming
* Splitting Data into Training and Test set

**1.Gathering data**

The data set which we have taken is Twitter Sentiment Analysis. The train set consists of three columns namely ID, Tweet Statement and category.

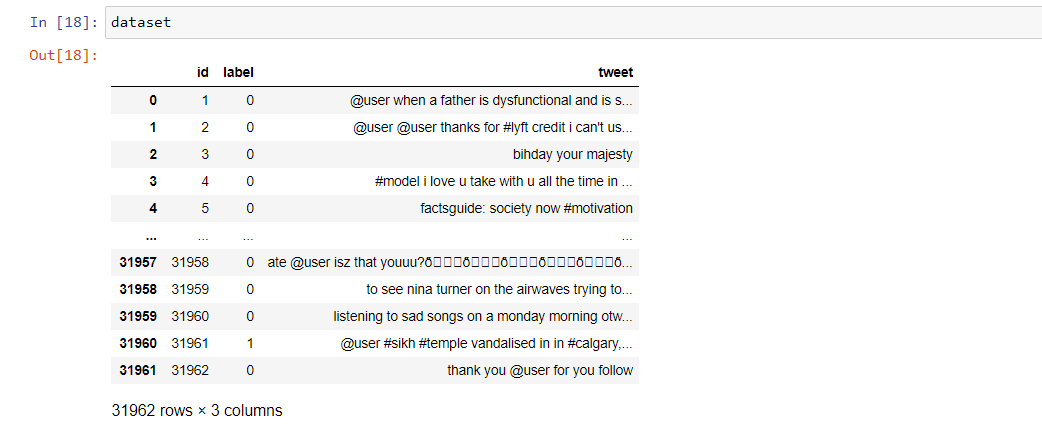


**2.Import the libraries**

The dataset train.csv is imported using Pandas Library.Various libraries such as numpy and matplotlib.pyplot



**Import the dataset**



**3.Text Cleaning or Pre-processing**

“Re” is the library which is used to replace the selected special characters with desired parameter. “NLTK” – Natural language Tool Kit is the library used for stemming using a special class in the library.

**Remove Punctuations, Numbers**

Punctuations, Numbers doesn’t help much in processing the given text, so we will be using **re** library to replace all the punctuations numbers with a space while excluding alphabets. As in the dataset the reviews are present in **Tweet** column, we are declaring a variable called tweet and assigning the second row of the column to declared variable. Then using **re** library we are substituting all the other special characters with a space excluding alphabets.

**Convert each word into its lower case**

Every word in the taken tweet should be lower cased, because if we have a word in different cases the machine will think both are different words.

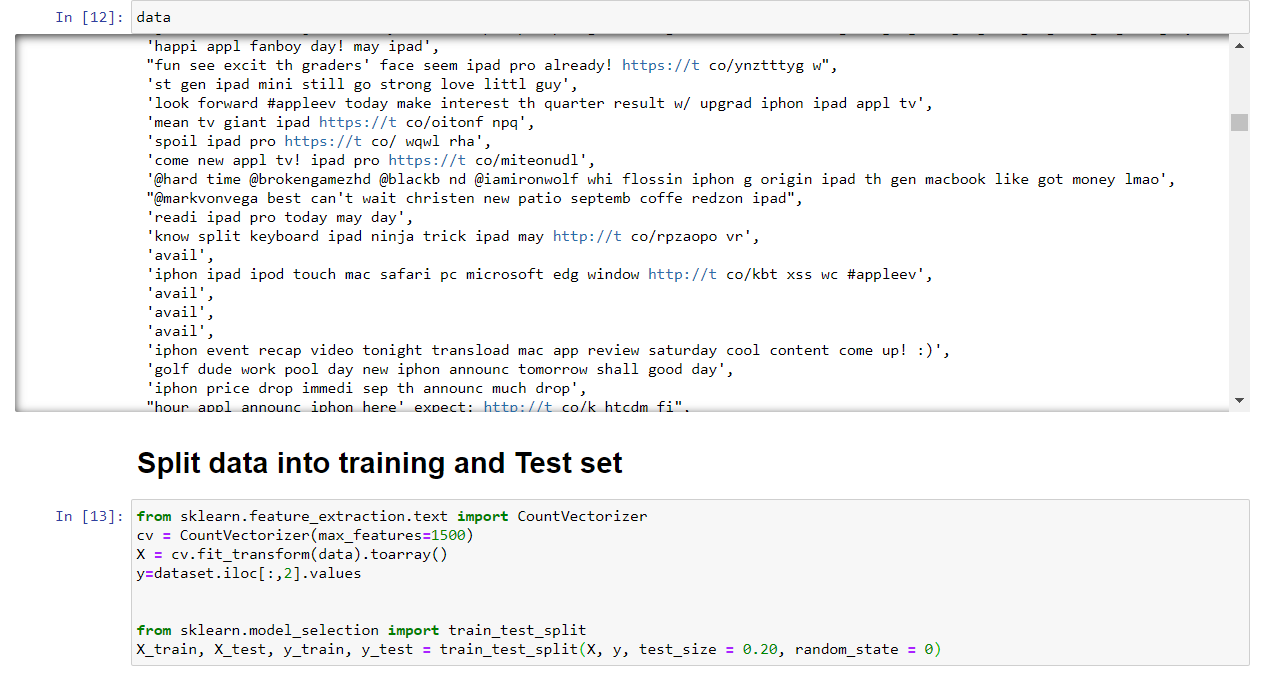
**Stemming**

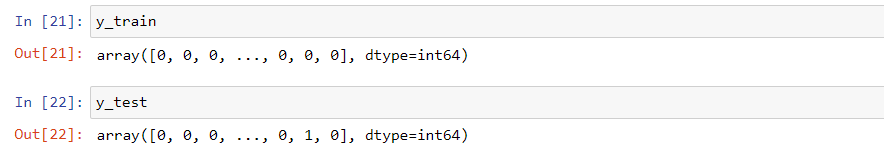
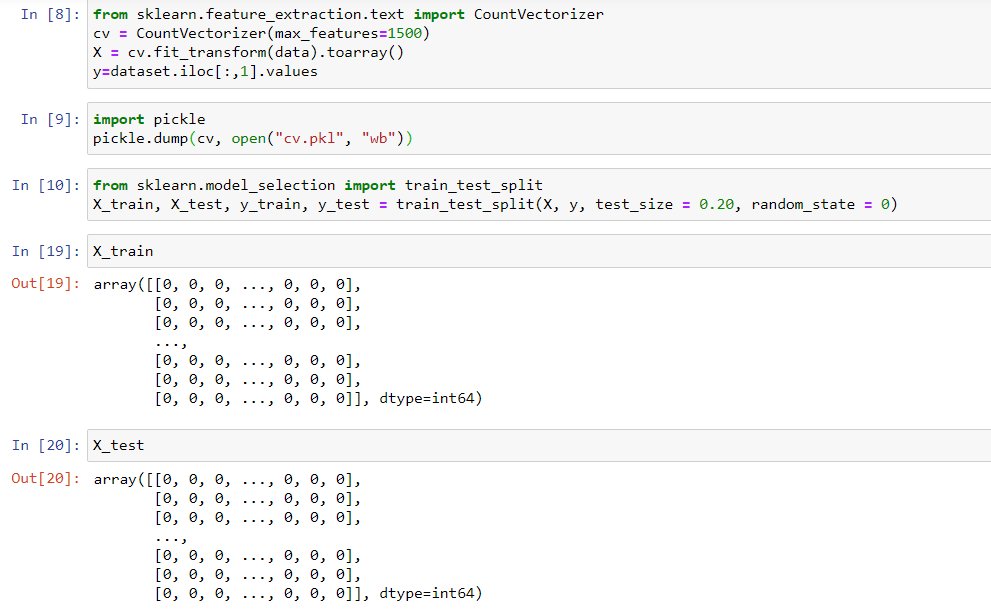
Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers.

****

**Splitting Data into Training and Test set**

For this, we need class train\_test\_split from sklearn.cross\_validation. Split can be made 70/30 or 80/20 or 85/15 or 75/25, here we choose 80/20 via “test\_size”. X is the bag of words; y is 0 or 1 (positive or negative).



****

**Model Building**

**Importing Libraries:**

The first step is to define the functions and classes we intend to use in this. We will use two classes from the Keras library to define our model.

**Initializing the model :**

Keras has 2 ways to define a neural network**:**

* Sequential
* Function API

We will use the **Sequential** constructor to create a model, which will then have layers added to it using the **add()** method.

**Adding Input layer:**

This step is to add a dense layer (input layer) where you will be specifying the number of inputs to the neural network, activation function and weights initializer and number of connection to the hidden layer as the arguments. We use add() method to add dense layers.

**Adding Hidden layer:**

This step is to add a dense layer (Hidden layer) where you will be specifying the number neurons to the next layer, activation function and weight initializer as the arguments.

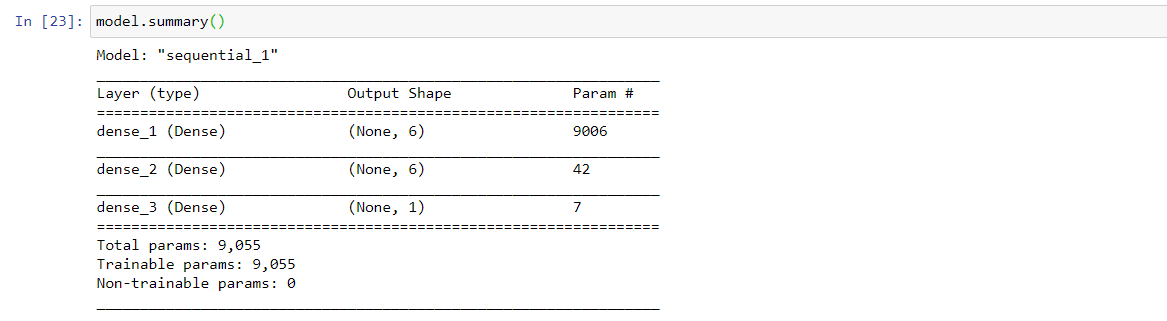
**Adding an Output Layer :**

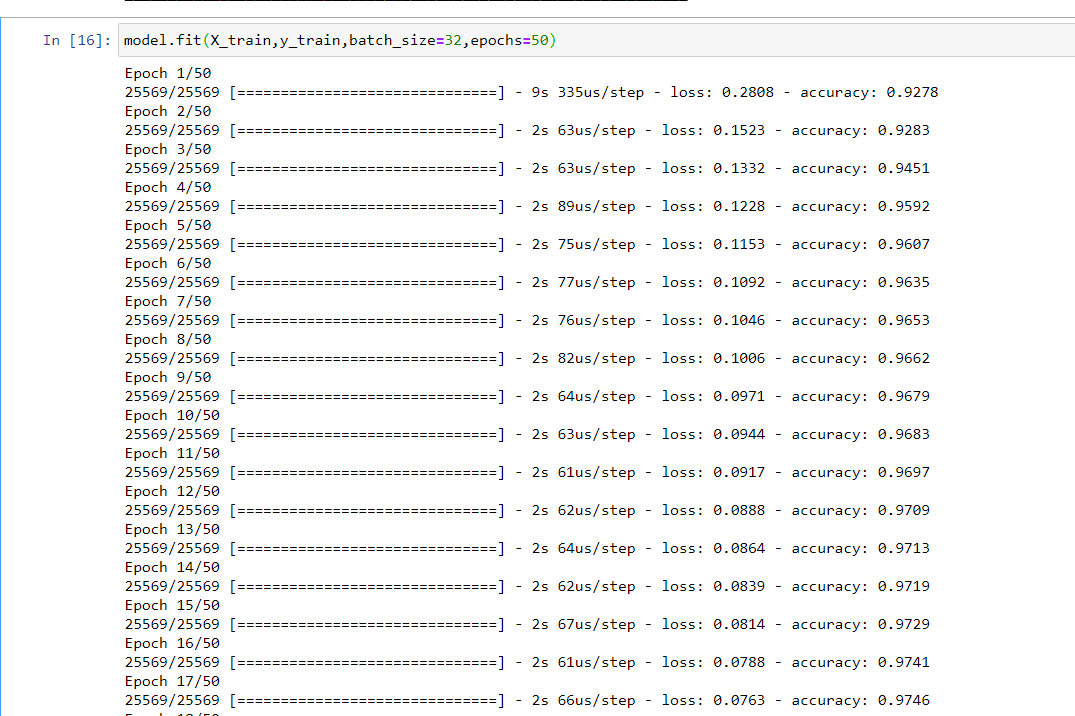
This step is to add a dense layer (output layer) where you will be specifying the number of classes your dependent variable has, activation function and weight initializer as the arguments.



**Configuring the learning process:**

Compilation requires 3 arguments: an optimizer, a loss function, and a list of metrics.

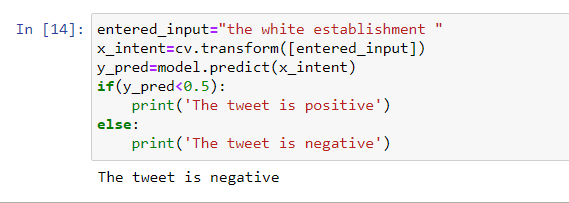
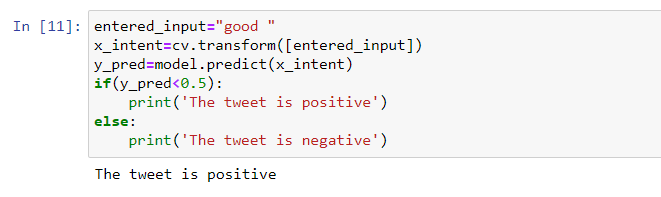




**Training the model and saving the :**

Training begins by calling the **fit()**method. The arguments are batch size as you are using “**adam**” (bath gradient descent and epochs: no: of times the model should get trained.

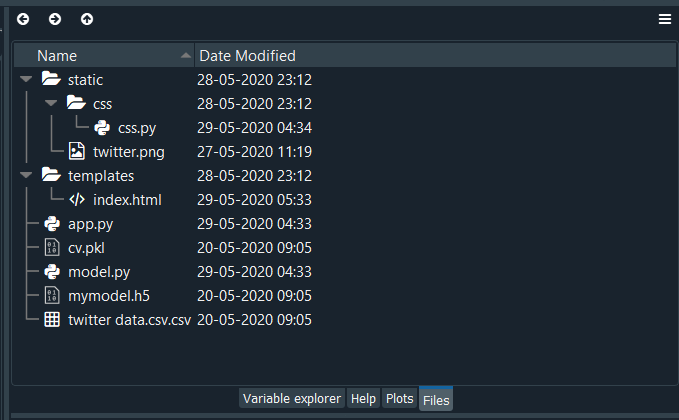


****

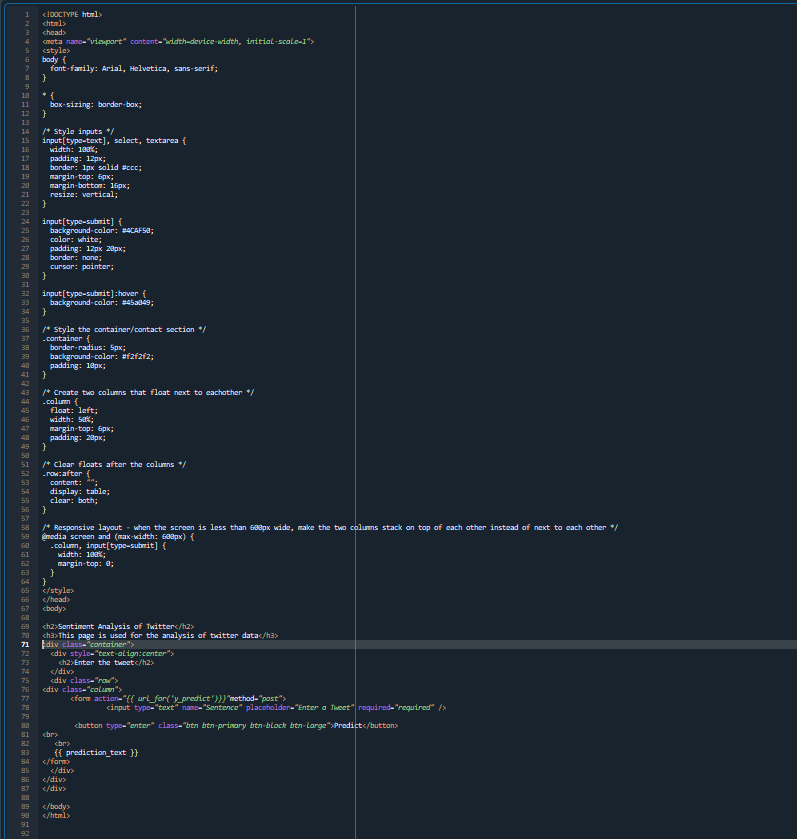
**Building HTML Page:**

This is the basic HTML page for our Project. H1 tag is used to give heading to the project. user has to enter the tweet , so we have to add 1 (one) text input fields in the web page.A button is used to send these values to the model files this functionality will be written in the python file app.py. the model predicts the value and is displayed on the {{ **y\_pred** }}field and respected emoji will be displayed.

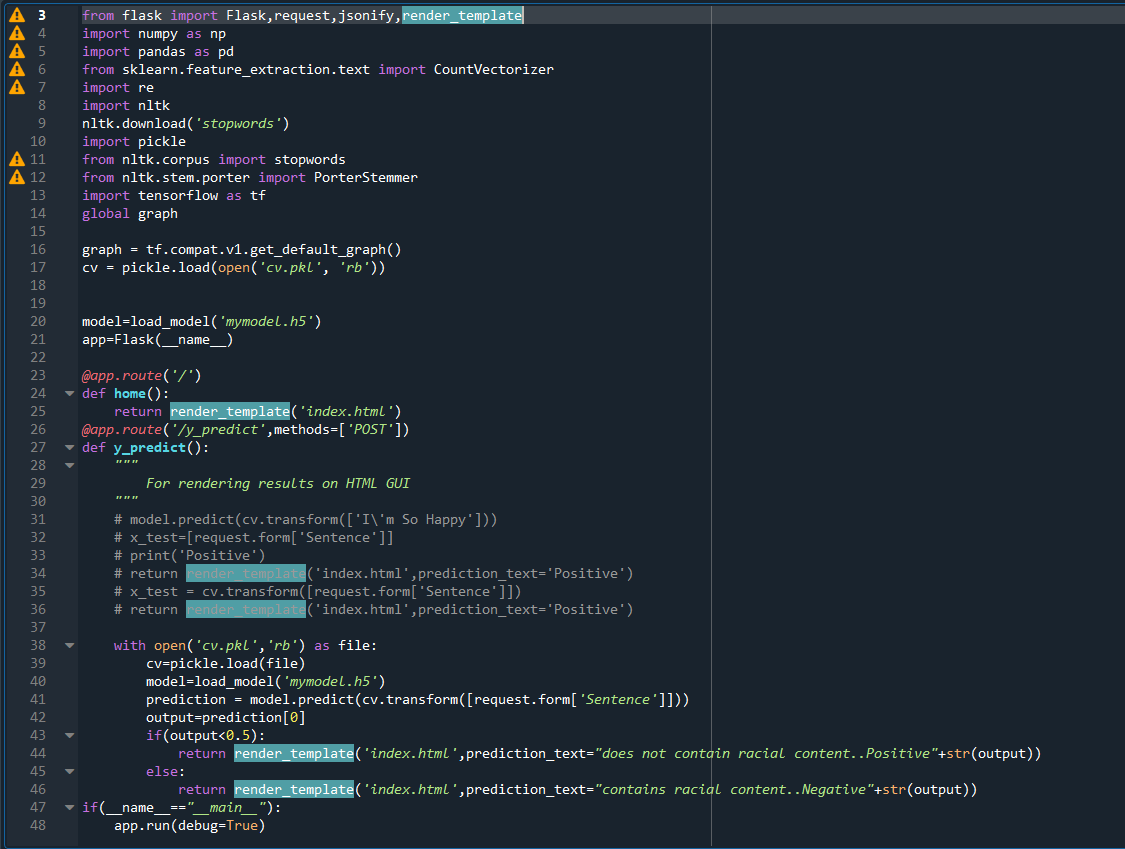
**FILE STRUCTURE:**

****

**HTML CODE:**

****

**PYTHON CODE:**

****

**FLOWCHART:**

**Training Data**

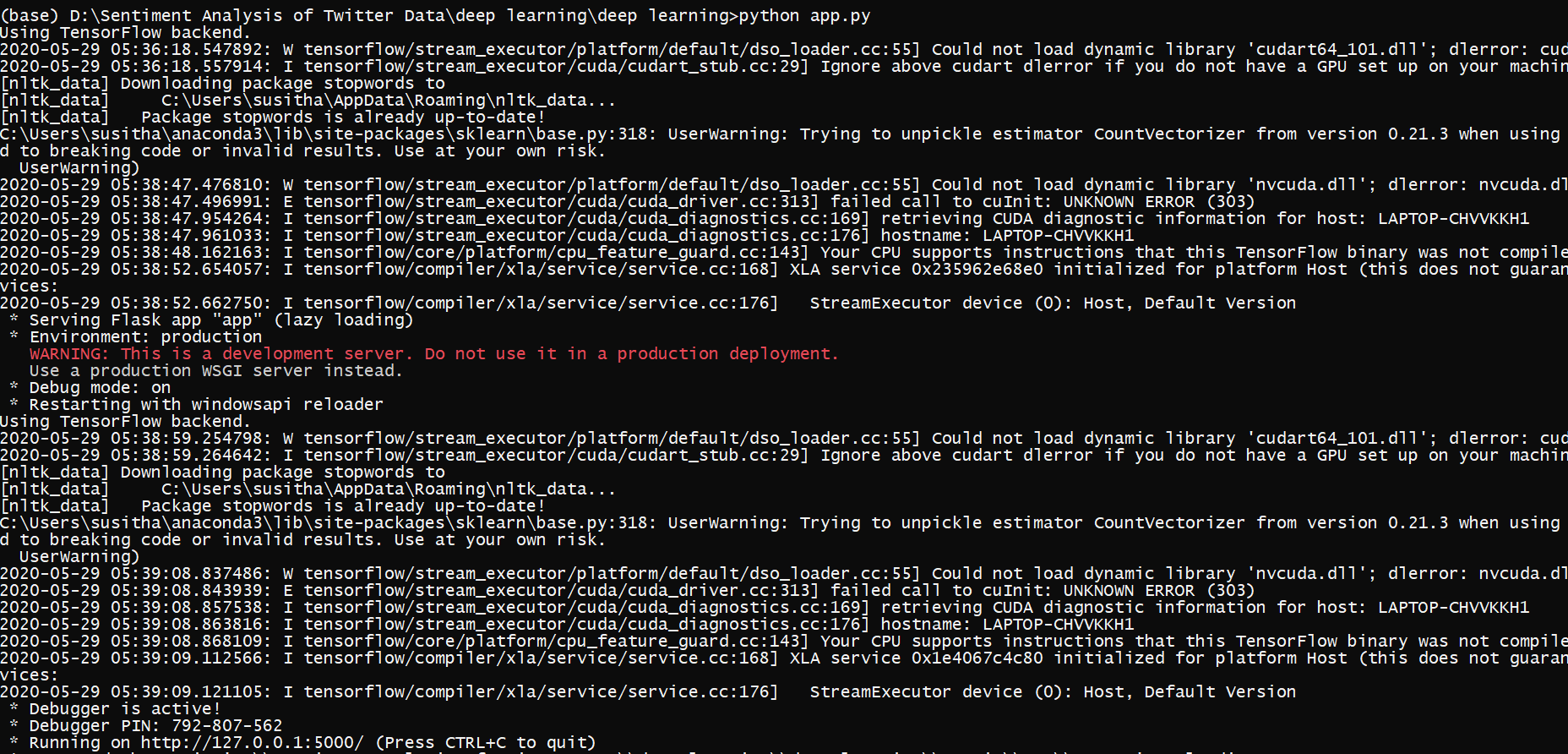
**Training Data**

**Feature Representation**

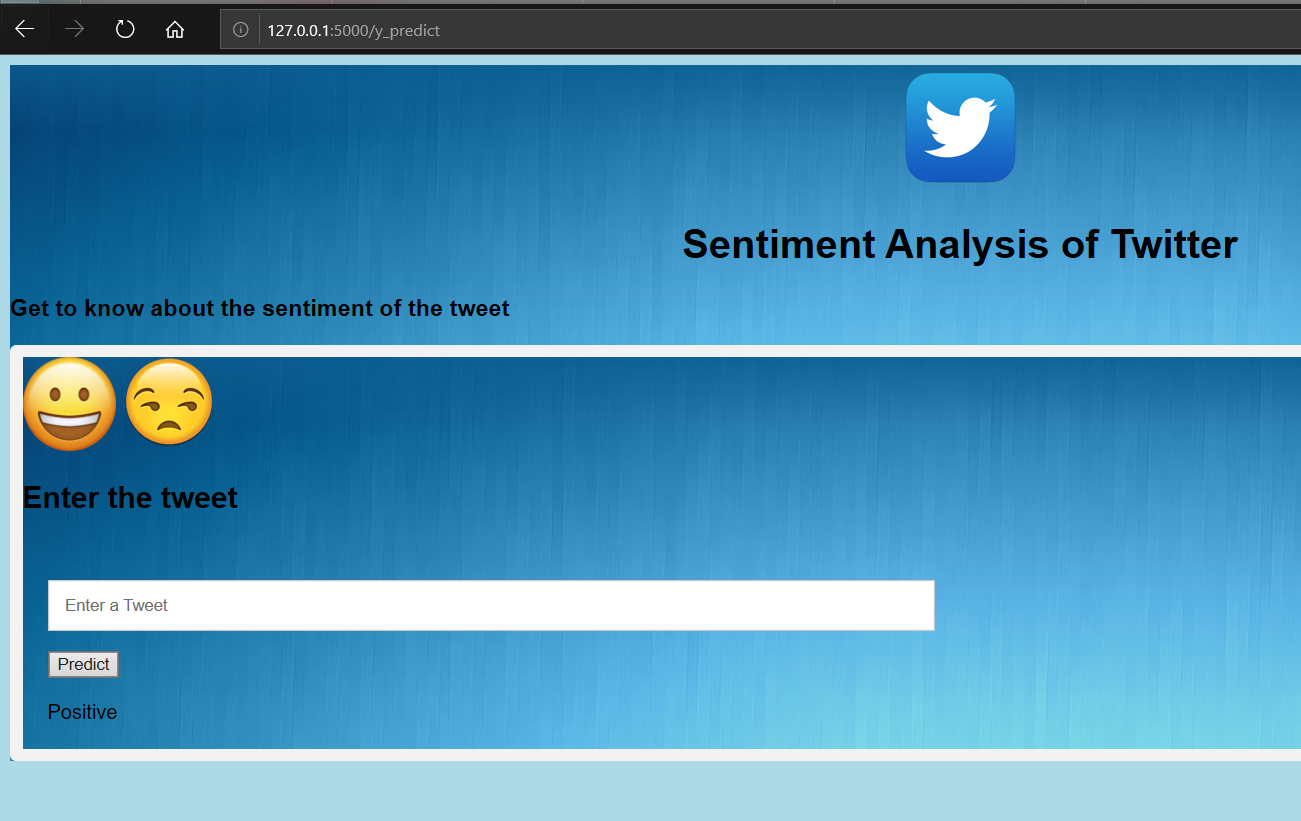
**Learning Algorithm**

**Sentiment Classifier**

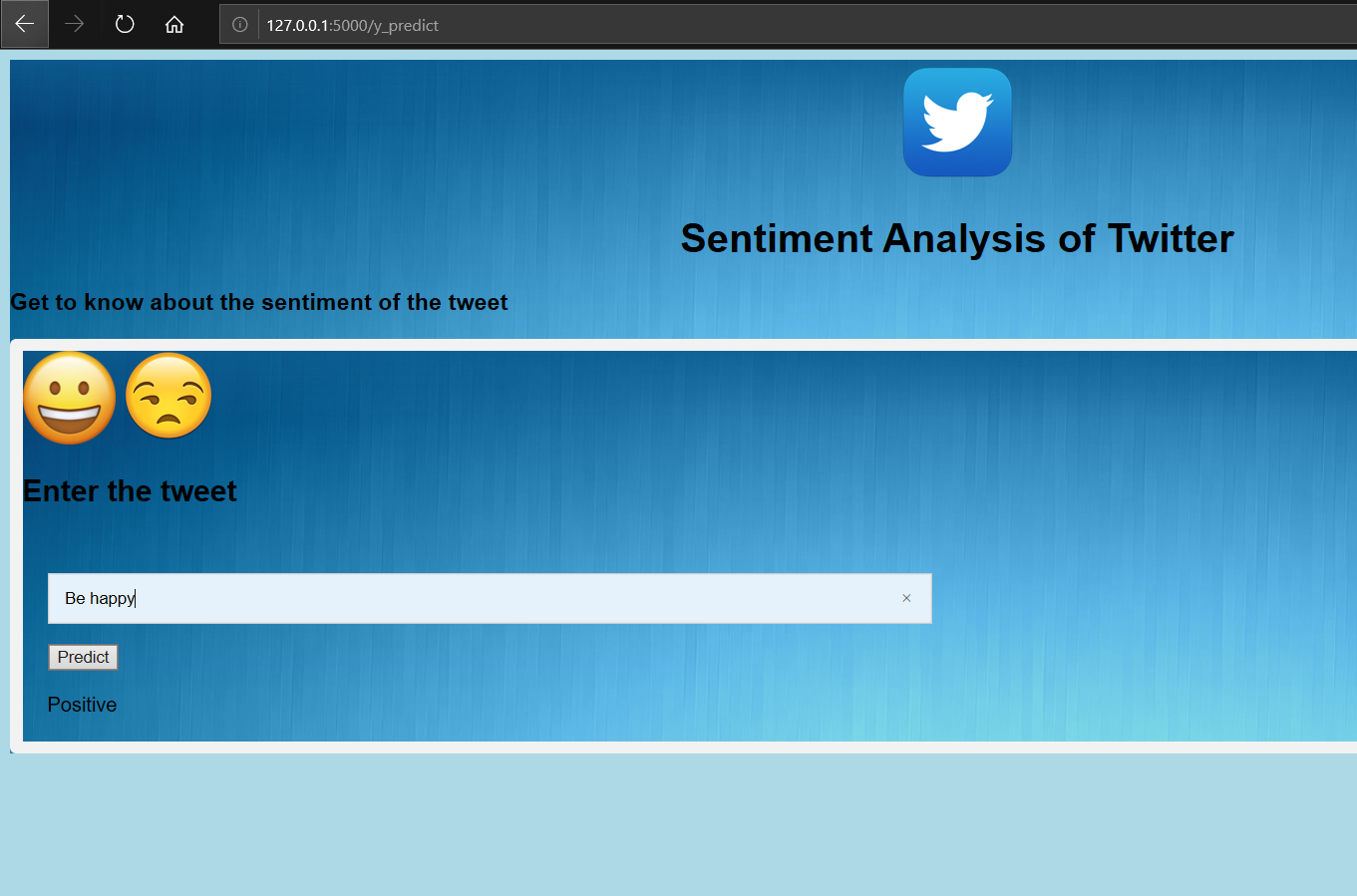
**OUTPUT:**

****

**PAGE:**

****

**OUTPUT:**

****

**RESULT:**

Sentiment Analysis is an interesting way to think about the applicability of Natural Language Processing in making automated conclusions about text. It is being utilized in social media trend analysis and, sometimes, for marketing purposes. The results from sentiment analysis help businesses understand the conversations and discussions taking place about them.They can quickly identify any negative sentiments being expressed and turn poor customer experiences into good ones.

|  |  |
| --- | --- |
|  |  |

**ADVANTAGES:**

* Sentiment analysis is a useful tool for any organisation or group for which public sentiment or attitude towards them is important for their success-whichever way that success is defined.
* Our social media ,blogs and online forums millions of people are busily discussing and reviewing businesses,companies and organisations .And those opinions are being ‘listened to’ and analysed.
* Those being discussed are making use of this enormous amount data by using computer programs that don’t just locate all mentions of their products,services but also determines the emotions and attitudes behind the words being used.

**DISADVANTAGES:**

Sentiment analysis tools can identify and analyse many pieces of text automatically and quickly. But computer programs recognizing things –the sort of things a person would have little trouble identifying. And failing to recognize these can skew the results. So sentiment analysis tool do a really great job of analysing text for opinion and attitude ,but they are not perfect.

**APPLICATIONS:**

* Social media monitoring
* Customer Experience Management and voice of customer
* People analytics

**CONCLUSION**

We develop a deep learning system for message-level Twitter sentiment classification in this project. The effectiveness of this project has been verified in both positive/negative/neutral classification of tweets. **Sentiment analysis** allows businesses to identify customer **sentiment** toward products, brands or services in online conversations and feedback. Thus this tool can be used in a wide range of applications.

**FUTURE SCOPE:**

Sentiment analysis is already evolving rapidly from a very simple (positive, negative, neutral) to more granular and deep understanding. These new classifier really dimensionalize the nuances of human expression in meaningful ways. There is also a move away from document/record level analysis of the text towards entity/facet level - meaning every expression of opinion is captured so that we can really understand the root cause drivers of opinions. This requires machine learning approaches that are superceding more traditional rules based approaches.

**REFERENCES:**

* [www.kaggle.com](http://www.kaggle.com)
* [www.github.com](http://www.github.com)
* [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
* towardsdatascience.com
* pythonprogramming.net