Corona Tweet Classification System

PROJECT REPORT

OF MAJOR PROJECT

BACHELOR OF COMPUTER APPLICATIONS (B.C.A.)

SUBMITTED BY

Ahmad Faraz Ansari

Batch Year – 2019 -22

Enrollment Number – U1946007

PROJECT SUPERVISOR – Er. Shreya Agrawal

CENTER OF COMPUTER EDUCATION & TRAINING

Institute of Professional Studies,

University of Allahabad, Prayagraj

Uttar Pradesh

ACKNOWLEDGEMENT

Project is an important milestone in the completion of any Professional Course. As a student of B.C.A, I got the golden opportunity to do this work. It gives me immense pleasure to express my feelings of deep gratitude towards the subjects without whom, it would have been very difficult to accomplish this mammoth task.

I wish to express my thanks, to my parents, my Project Supervisor and Project in-charge **Er. Shreya Agrawal.** I am also thankful to my Course Coordinator **Prof. Ashish Khare,** who provided me this golden opportunity to work on this wonderful project called “**Corona Tweet Classification System”,** which also helped me in doing lot of research, which, in return, gave me insight on so many new things that are going to help me in the foreseeable future. I would like to thank all those who have helped me in providing direction, information and advice at all stages in this Project.

I take this opportunity to express my special gratitude to the members of **Institute of Professional Studies (IPS), University of Allahabad** for providing me this wonderful opportunity to work on this project.

Ahmad Faraz Ansari

B.C.A. 6th Semester

Enrollment No. – U1946007

CERTIFICATE

This is to certify that **Ahmad Faraz Ansari** student of **B.C.A. 6th Semester** of **Centre of Computer Education, Institute of Professional Studies, University of Allahabad, Prayagraj** has successfully completed his project entitled **Corona Tweet Classification System** under the guidance of **Er. Shreya Agrawal** during the academic year 2019 - 22 as per guidelines given by **University of Allahabad, Prayagraj**.

Er. Shreya Agrawal
(Project Supervisor and Project in-charge)

DECLARATION

I, **Ahmad Faraz Ansari**, hereby declare that the project report entitled "**Corona Tweet Classification System**" has been submitted to **University of Allahabad** in partial fulfilment of the requirement for the award of degree of B.C.A., is a record of Bonafede Project work carried out by me, under the guidance of **Er. Shreya Agrawal**.

I further declare that this project has not been submitted and will not be submitted, either in part or full, for the award of any other degree or diploma in this institute or any other institute or university.

The work contained in the report is original and has been done by me under the general supervision of my supervisor.

I have followed the guidelines provided by the University of Allahabad in writing this report.

Date: 20-05-2022 **AHMAD FARAZ ANSARI**

Place: Prayagraj B.C.A. 6th Semester

ABSTRACT

Corona Tweet Classification System is tweet classification system based on Machine-learning. It takes user inputs as tweet and then predict the sentiment of that tweet i.e., whether the tweet is covid-positive or covid-negative. After the prediction, system displays the predicted sentiment i.e., covid-positive or covid-negative, tokenized tweet and bunch of evaluation metrics (It is used to check model accuracy).

This report reflects the idea of taking user's input into consideration and performing classification and establishing conclusion on interested topics using Machine – Learning algorithm. Support Vector Machines in Machine – Learning is tuned up using supervised learning to obtain outputs for classification.

Keywords : Classification, sentiment, Support vector machines, supervised learning.

SYNOPSIS

INTRODUCTION

Corona Tweet Classification System is a Machine Learning based Tweet Classification System.

It developed using Python 3 for platform-independency.

It is one of the most accurate and precise Tweet Classification System.

The proposed System is able to classify the Tweets based on the magnitude of emotions like Positivity or Negativity or Neutrality underlying it.

PROBLEM DEFINITION

Supervised Machine Learning and Classification.

Based on the input (Tweet), the System can classify it in the given classes.

ALGORITHM

Algorithm used by the System is **Support Vector Machine (SVM).**

OBJECTIVES

The prime objectives for the Corona Tweet Classification System are as follows:

1. To create a System capable of Classifying Tweets.
2. Creation of System that has great chance of accuracy and precision.
3. Classification is done on the basis of severeness of the underlying emotion.

OVERALL DESCRIPTION

The proposed Corona Tweet Classification System will take on other Classification System based by tackling the basic underlying problem.

It will take care of all the user resources without requiring any user Interaction.

User is not supposed to get into the details of System's Working or its underlying complexities and its model (abstraction).

USER CHARACTERISTICS AND ASSUMPTIONS

1. User is supposed to have basic knowledge of a computer.
2. User has to provide a valid input to gain a valid output.
3. User should always try to provide a valid input i.e., Tweet.
4. User is required to be patient especially if he/she is using the System on an older hardware.

USER REQUIREMENTS

1. System should be able to handle all logic effectively.
2. System should not throw any unchecked exceptions.
3. System should be able to determine output i.e., the positivity or negativity of Tweet without any ambiguity.
4. System should not crash without providing any concrete result.

REQUIREMENTS ANALYSIS

The System is designed to work on any machine with newer hardware to harness the maximum power of modern Hardware.

Hardware Requirements

1. 4GB of RAM or more.
2. i3 - 7th Generation Processor or later.
3. No External GPU Required.

Software Requirements

1. Python 3 Interpreter has to be pre-installed.
2. PyCharm or Jupyter Notebook pre-installed
3. Any 64-bit Desktop OS (Microsoft Windows 10 or later is recommended).

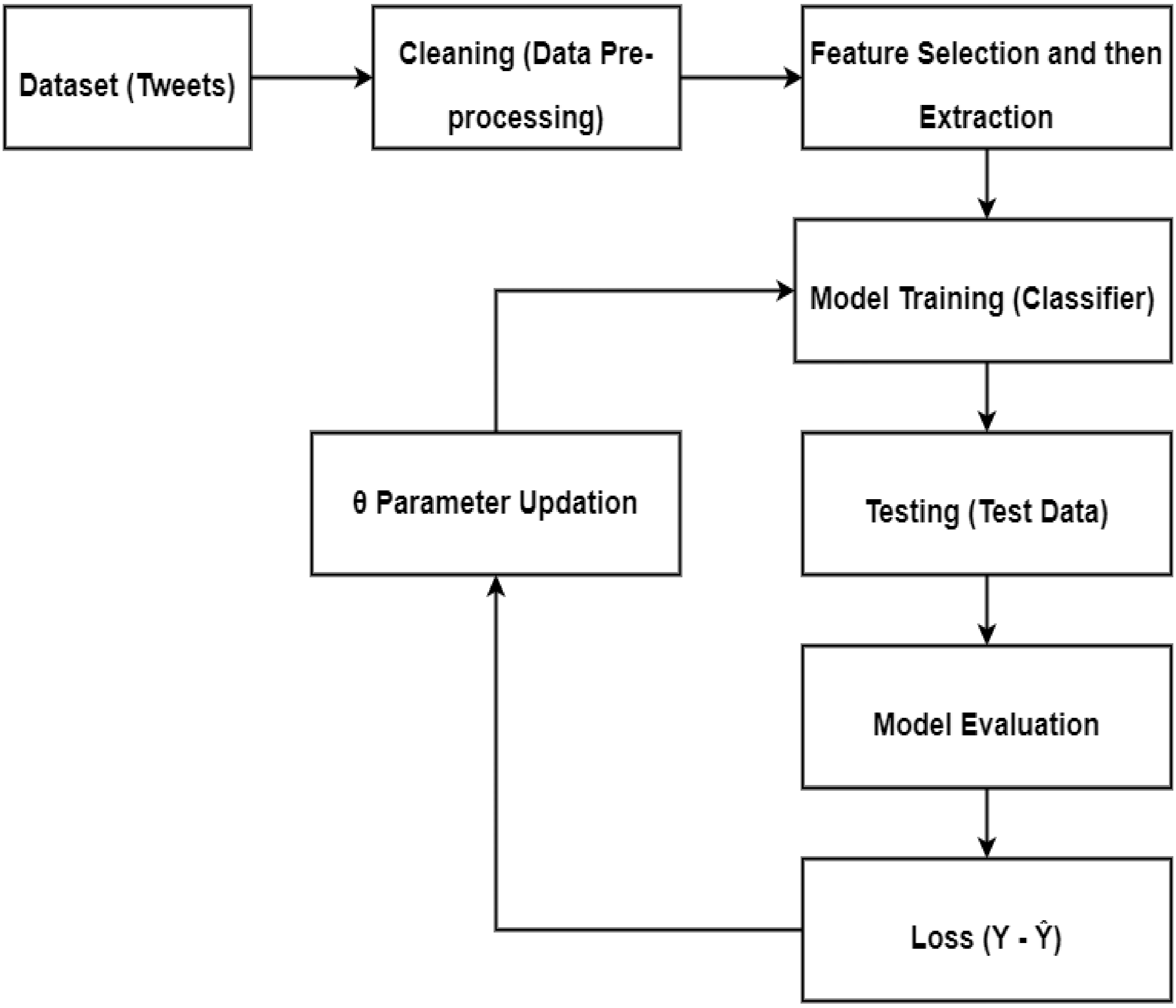
ATTRIBUTES RELATED TO APPLICATION

Corona Tweet Classification System, has many attributes that makes it outstanding if compared to other Classification System in the same genre.

Following Points provide some of the insights related to the Classification System.

1. Adaptability: Corona Tweet Classification System, can be easily adapted by new Users without any hassle.
2. Availability: System can be made available to general User everywhere in the world. It can be accessed by any user irrespective of their location (Some countries excluded).
3. Accuracy: System can determine the outcome of the Tweet i.e., positivity or negativity with maximum precision and accuracy.
4. Maintainability- No extra maintenance is required by User.
5. Portability: Since the System is itself built using Python (An interpreted platform independent language) so that means it can be ported to any desktop platform including macOS, Linux, Windows. It requires no extra overhead to be ported to any other platform.
6. Reusability: The proposed Classification System can be used any number of times as it does not require to evaluate all the Tweets each time, a new Tweet is provided as input.
7. Cost: Corona Tweet Classification System is free to use.

MACHINE LEARNING PIPELINE DIAGRAM


Dataset (Tweets)
Cleaning (Data Pre-processing)
Feature Selection and then Extraction
Model Training (Classifier)
θ Parameter Updation
Testing (Test Data)
Model Evaluation
Loss (Y - Ŷ)

MILESTONE

S. No.	Project Activity	Estimated Start Date	Estimated End Date
1.	Synopsis Submission	22/03/2022	25/03/2022
2.	Presentation Submission	23/03/2022	25/03/2022

MEETING WITH THE SUPERVISOR

Date of Meet	Mode	Comments by the Supervisor	Signature of the Supervisor
10/03/2022	Offline		
24/03/2022	Offline		

REFERENCES

1. https://app.diagrams.net/
2. Kaggle.com/pre-processing-in-NLP
3. Statistical Analysis using Machine Learning

TABLE OF CONTENTS

PROJECT REPORT

INTRODUCTION

Corona Tweet Classification System or also known as Covid Tweet Classification System, is a Machine-learning based tweet classification system.

It is developed entirely using Python 3 for platform-independency.

Classification of tweets is done using SVM algorithm.

Proposed System can classify tweets based on the underlying emotion i.e., whether the tweet is covid-positive or covid-negative.

The main objective of this project includes:

1. Creating a system that can classify tweets using machine-learning.
2. Classification of tweets is to be done on the basis of severeness of the underlying emotion.

PROPOSED SYSTEM

The Proposed, as mentioned before, is able classify user tweets (input) into Positive or Negative Class.

It is simpler and less-resource/time consuming than other classifiers available in the same genre. It is very quick, responsive and user-friendly.

It will take care of all the system resources without requiring any user Interaction. User is not supposed to get into the details of underlying software technicalities and its model (abstraction).

User-Interface is Jupyter Notebook Based.

MODULES

Classification is a supervised learning technique used for categorizing a given set of data into classes. Following are the modules developed regarding this project.

1. **Importing**
2. **Processing**
3. **Feature Extraction**
4. **Training**
5. **Testing**
6. **Evaluation Metrics**

1. Importing

In this module, all the required packages and the dataset are imported.

It is necessary to place the importing line at the top of the file for very obvious reasons.

Following are the packages name along with their purpose.

1. pandas – this package is necessary for working on the dataset
2. numpy – this package is necessary for array conversion
3. sklearn – this package is responsible for classification of tweets, splitting of dataset and calculation of metric scores.
4. re – this package is used for regular expression operations
5. string – this package is used for string related operations
6. nltk – this package is responsible for removing stop-words, Lemmatization of words, Tokenization of words and calculating the polarity scores.

2. Processing

Processing is the second step towards classification of tweet.

We require our dataset to be clean i.e., free from all unnecessary things that have no use in our model. All unnecessary columns are dropped from the dataset to save time and resources.

After that, each tweet is cleaned using regular expressions (removing symbols, stickers and hyperlinks etc.) and then tweets are tokenized and all punctuations and stop-words are removed.

3. Feature Extraction

We are extracting two features from the tweets that are useful for model training.

First, the length of the tweet and second, the polarity score (amount of positivity and negativity of the tweet).

4. Training

Training of model means feeding the machine learning algorithm with sufficient training data to learn from. It is very important to provide the best data to the ML algorithm to get the most accurate prediction in return.

ML Algorithm used for this project is called **Support Vector Machines (SVM).** This is a supervised learning algorithm primarily used for classification.

5. Testing

Testing of model means testing the model that has been trained using unknown data i.e., testing data. The predictions made by the models are then compared in further steps to check model accuracy. If the predictions made by the model are not satisfactory or accurate then the necessary measures has to be taken to increase model accuracy.

6. Evaluation Metrics

Evaluation is always good in any field. In the case of machine learning, it is the best practise. There are many metrics used for evaluating machine learning model. In this project, F1 Score and Jaccard Index are used for very specific reasons.

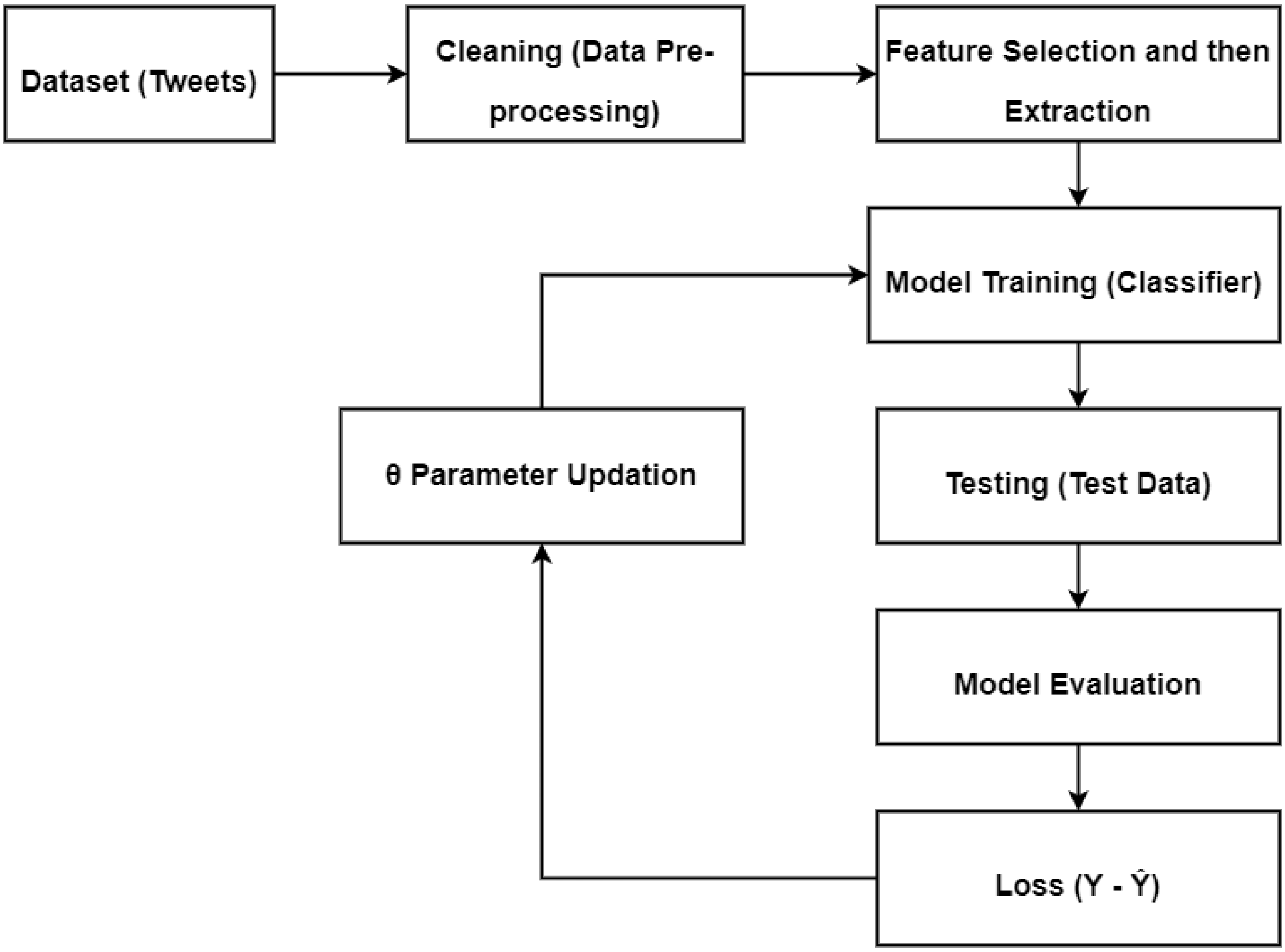
F1 Score : It is used to test the accuracy of model. Its range is [0, 1].

F1 Score = $\mathbf{2} * \left(\frac{\boldsymbol{Precision} * \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}}\right)$

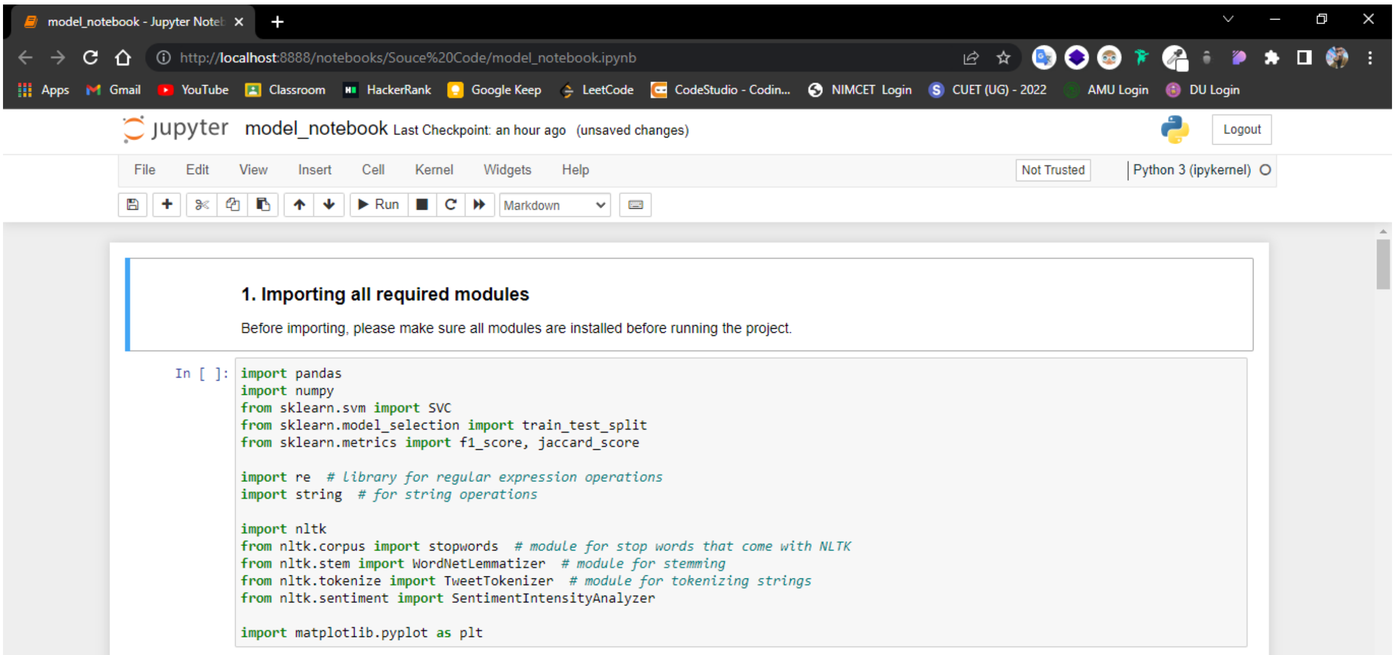
Jaccard Index : it is used in understanding the similarity between sample sets. The measurement emphasises similarity between finite sample sets. Its range is [0, 1].

Jaccard Index, J(A, B) = $|\mathbf{A} \cap \mathbf{B}| \ / |\mathbf{A} \cup \mathbf{B}|$

MACHINE LEARNING PIPELINE DIAGRAM


Dataset (Tweets)
Cleaning (Data Pre-processing)
Feature Selection and then Extraction
Model Training (Classifier)
θ Parameter Updation
Testing (Test Data)
Model Evaluation
Loss (Y - Ŷ)

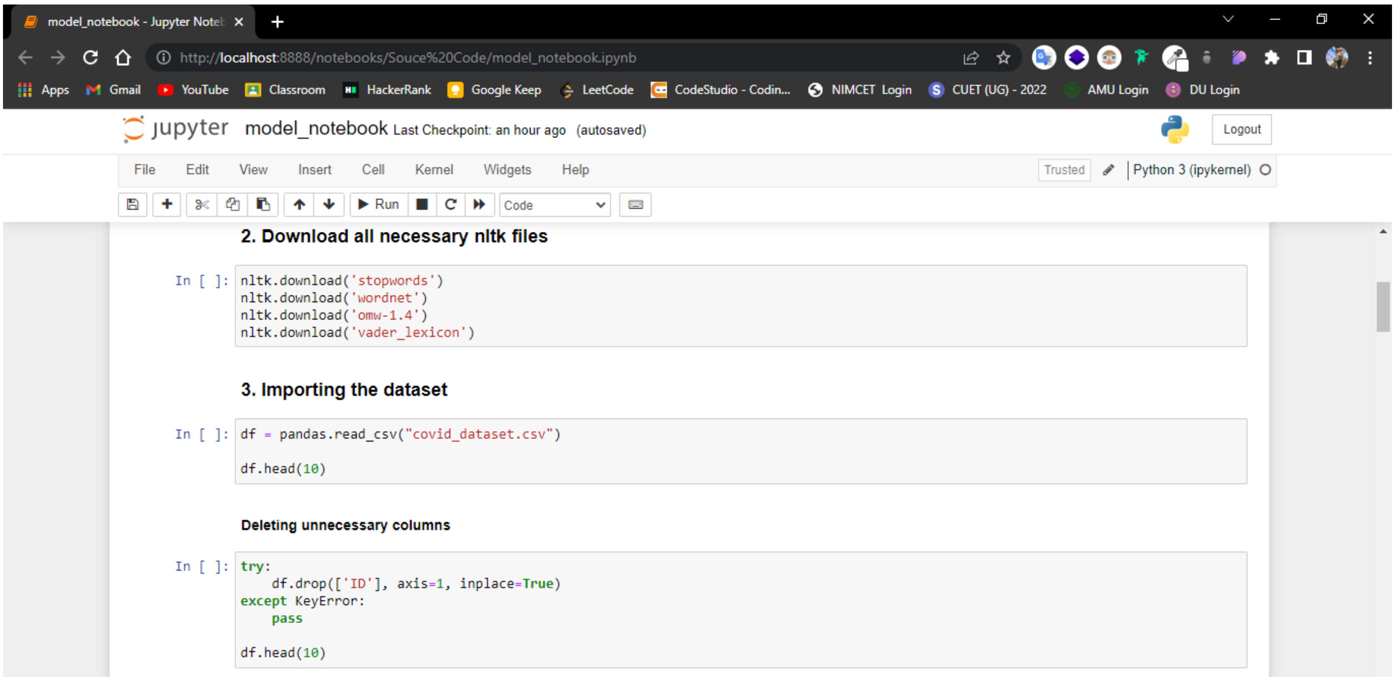
CODING



1. Importing all required modules

Before importing, please make sure all modules are installed before running the project.

```
import pandas
import numpy
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score, jaccard_score

import re  # library for regular expression operations
import string  # for string operations

import nltk
from nltk.corpus import stopwords  # module for stop words that come with NLTK
from nltk.stem import WordNetLemmatizer  # module for stemming
from nltk.tokenize import TweetTokenizer  # module for tokenizing strings
from nltk.sentiment import SentimentIntensityAnalyzer

import matplotlib.pyplot as plt
```



2. Download all necessary nltk files

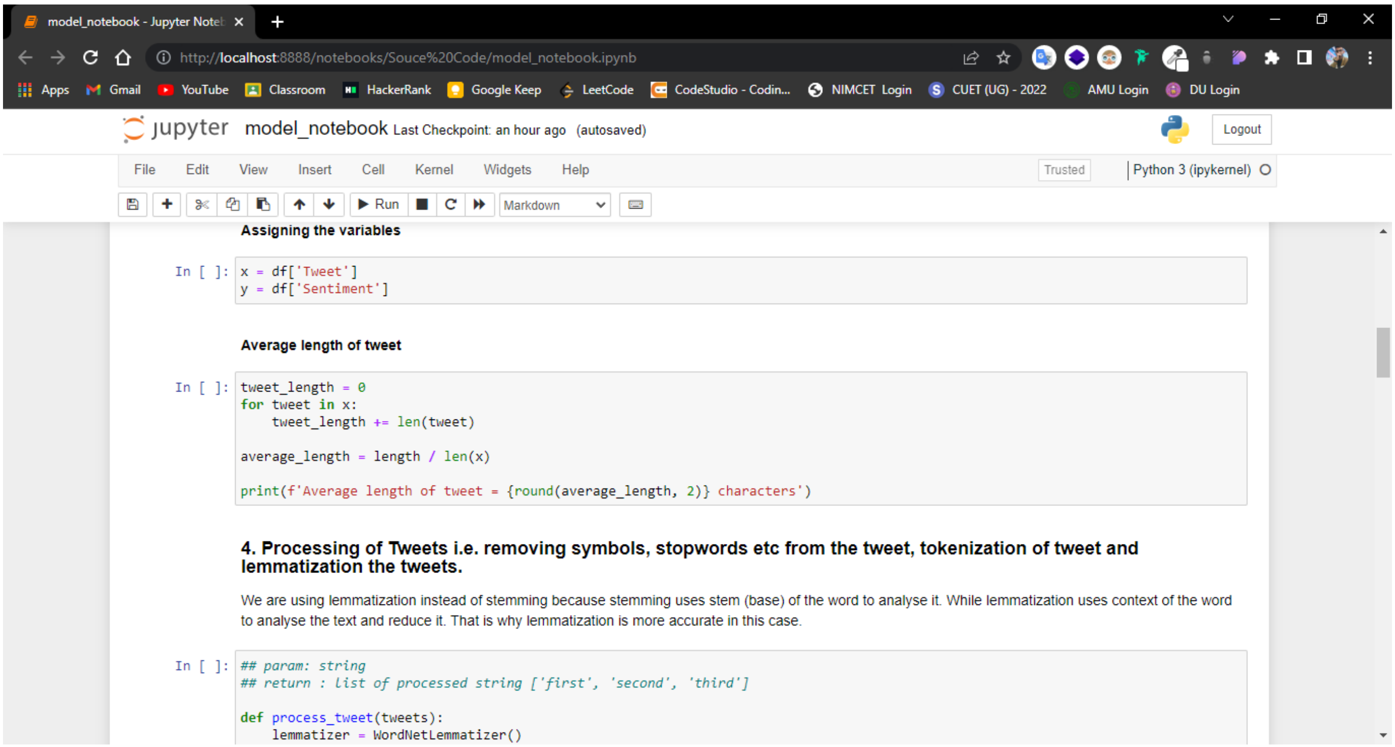
```
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('vader_lexicon')
```

3. Importing the dataset

```
df = pandas.read_csv("covid_dataset.csv")

df.head(10)
```

Deleting unnecessary columns

```
try:
    df.drop(['ID'], axis=1, inplace=True)
except KeyError:
    pass

df.head(10)
```



model_notebook Last Checkpoint: an hour ago (autosaved)

Assigning the variables

In [ ]:
```
x = df['Tweet']
y = df['Sentiment']
```

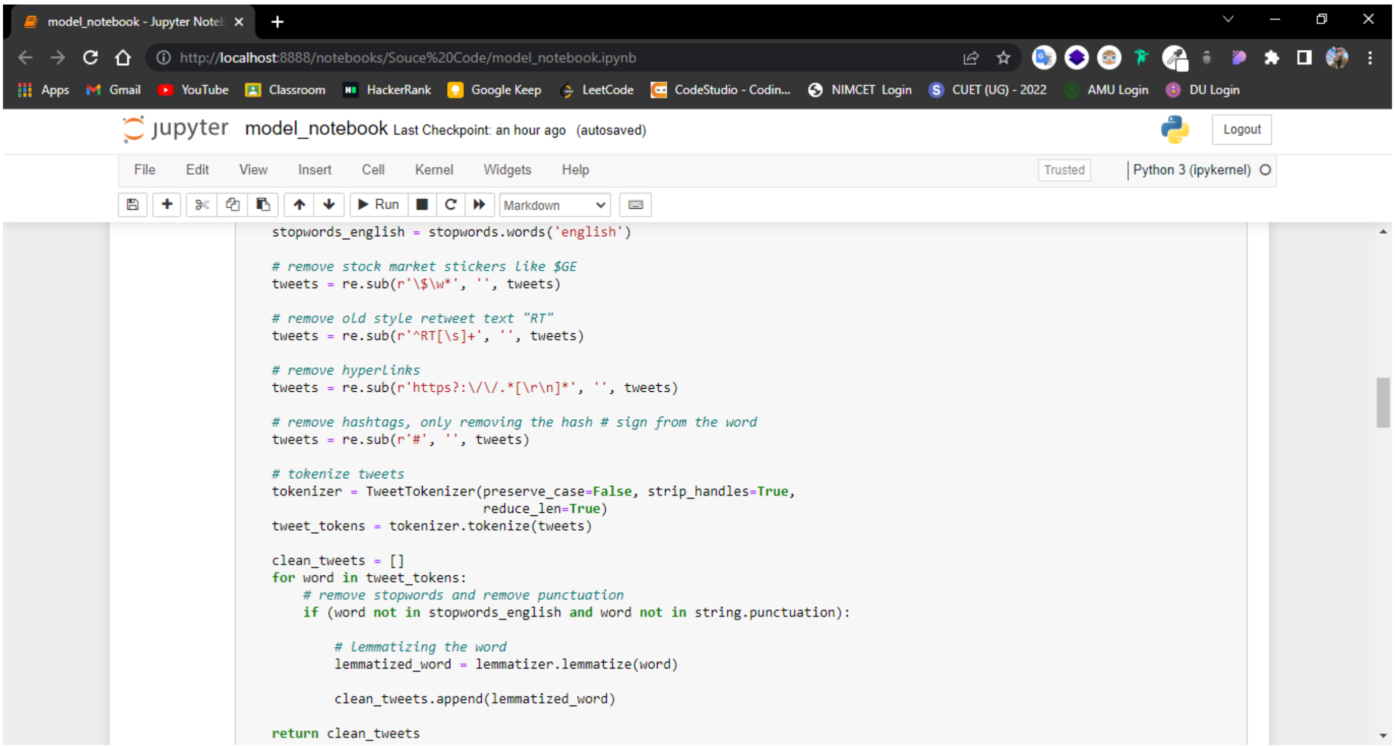
Average length of tweet

In [ ]:
```
tweet_length = 0
for tweet in x:
    tweet_length += len(tweet)

average_length = length / len(x)

print(f'Average length of tweet = {round(average_length, 2)} characters')
```

4. Processing of Tweets i.e. removing symbols, stopwords etc from the tweet, tokenization of tweet and lemmatization the tweets.

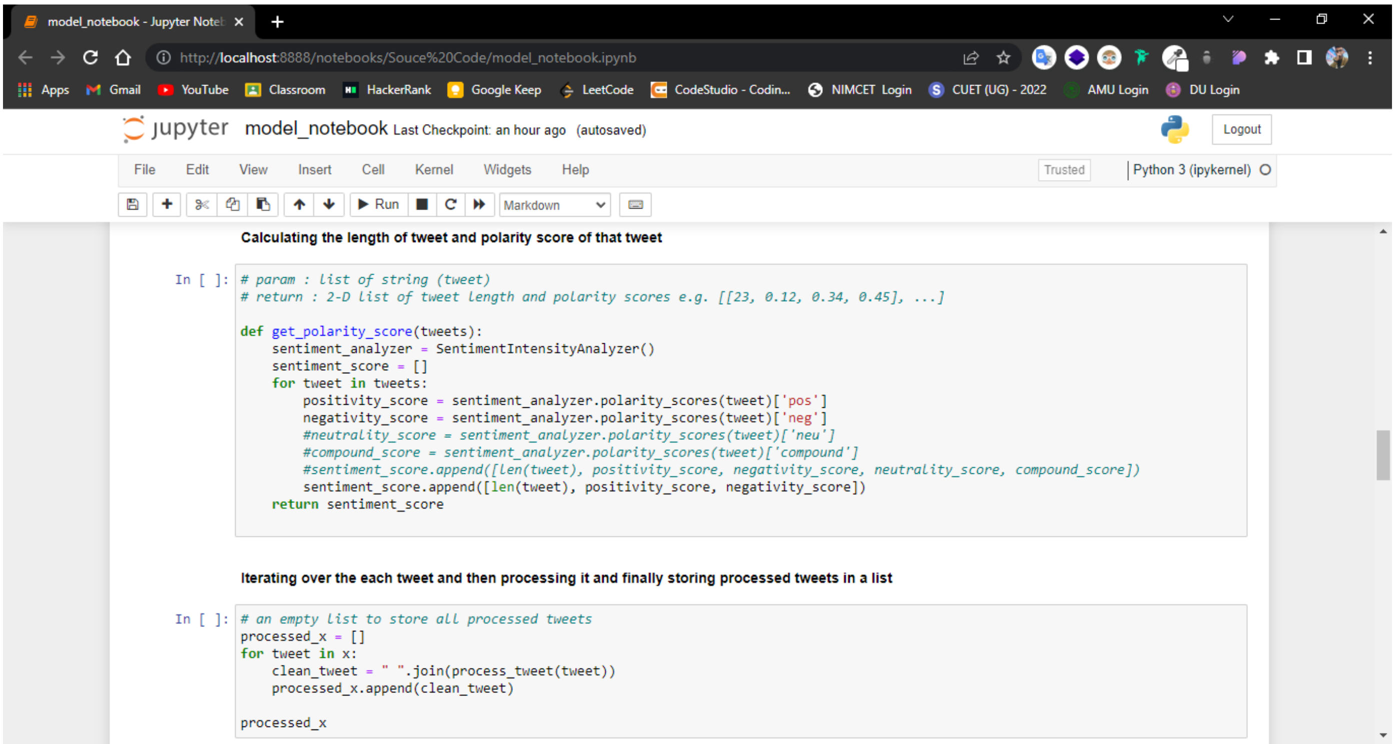
We are using lemmatization instead of stemming because stemming uses stem (base) of the word to analyse it. While lemmatization uses context of the word to analyse the text and reduce it. That is why lemmatization is more accurate in this case.

In [ ]:
```
## param: string
## return : list of processed string ['first', 'second', 'third']

def process_tweet(tweets):
    lemmatizer = WordNetLemmatizer()
```



model_notebook Last Checkpoint: an hour ago (autosaved)

```
    stopwords_english = stopwords.words('english')

    # remove stock market stickers like $GE
    tweets = re.sub(r'\$\w*', '', tweets)

    # remove old style retweet text "RT"
    tweets = re.sub(r'^RT[\s]+', '', tweets)

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

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

    # tokenize tweets
    tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True,
                               reduce_len=True)
    tweet_tokens = tokenizer.tokenize(tweets)

    clean_tweets = []
    for word in tweet_tokens:
        # remove stopwords and remove punctuation
        if (word not in stopwords_english and word not in string.punctuation):

            # lemmatizing the word
            lemmatized_word = lemmatizer.lemmatize(word)

            clean_tweets.append(lemmatized_word)

    return clean_tweets
```

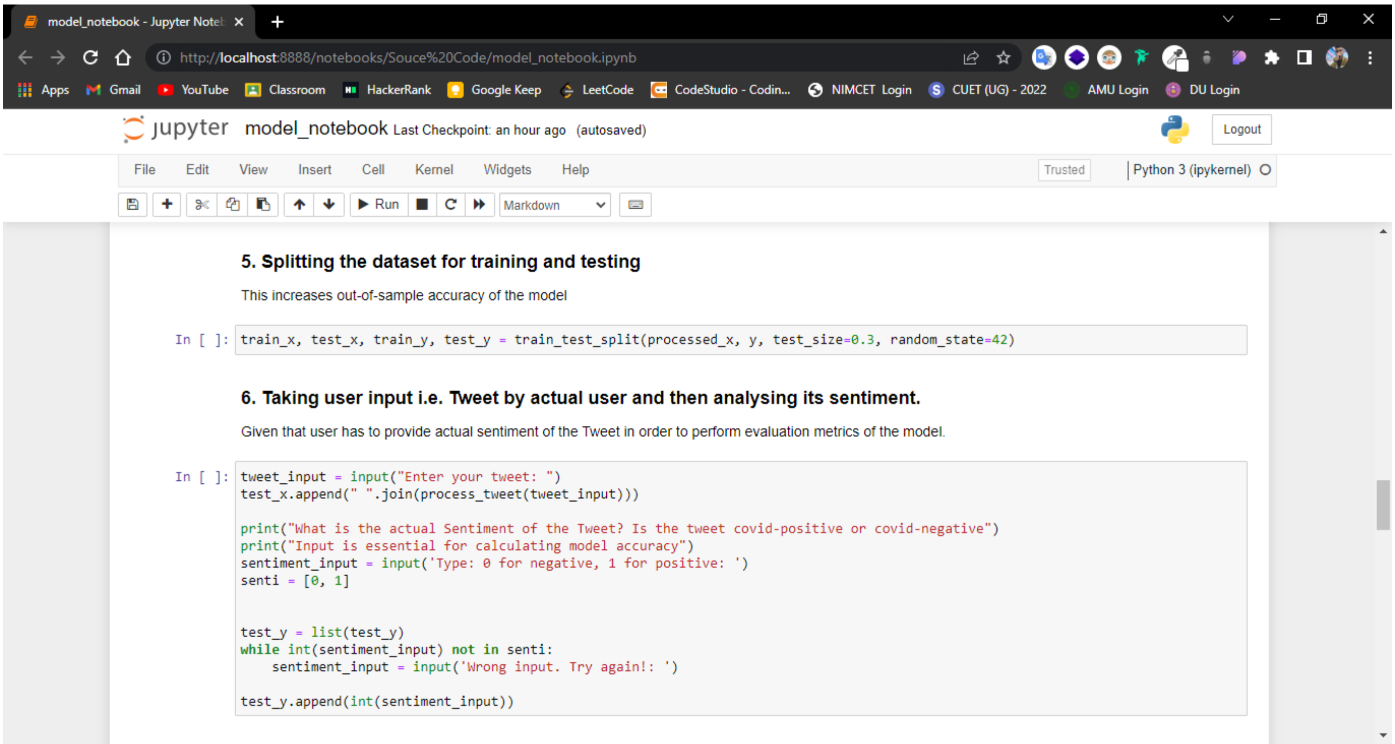

model_notebook - Jupyter Notebook
http://localhost:8888/notebooks/Souce%20Code/model_notebook.ipynb
jupyter model_notebook Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown

Calculating the length of tweet and polarity score of that tweet

```
In [ ]: # param : list of string (tweet)
        # return : 2-D list of tweet length and polarity scores e.g. [[23, 0.12, 0.34, 0.45], ...]

        def get_polarity_score(tweets):
            sentiment_analyzer = SentimentIntensityAnalyzer()
            sentiment_score = []
            for tweet in tweets:
                positivity_score = sentiment_analyzer.polarity_scores(tweet)['pos']
                negativity_score = sentiment_analyzer.polarity_scores(tweet)['neg']
                #neutrality_score = sentiment_analyzer.polarity_scores(tweet)['neu']
                #compound_score = sentiment_analyzer.polarity_scores(tweet)['compound']
                #sentiment_score.append([len(tweet), positivity_score, negativity_score, neutrality_score, compound_score])
                sentiment_score.append([len(tweet), positivity_score, negativity_score])
            return sentiment_score
```

Iterating over the each tweet and then processing it and finally storing processed tweets in a list

```
In [ ]: # an empty list to store all processed tweets
        processed_x = []
        for tweet in x:
            clean_tweet = " ".join(process_tweet(tweet))
            processed_x.append(clean_tweet)

        processed_x
```


model_notebook - Jupyter Notebook
http://localhost:8888/notebooks/Souce%20Code/model_notebook.ipynb
jupyter model_notebook Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown

5. Splitting the dataset for training and testing

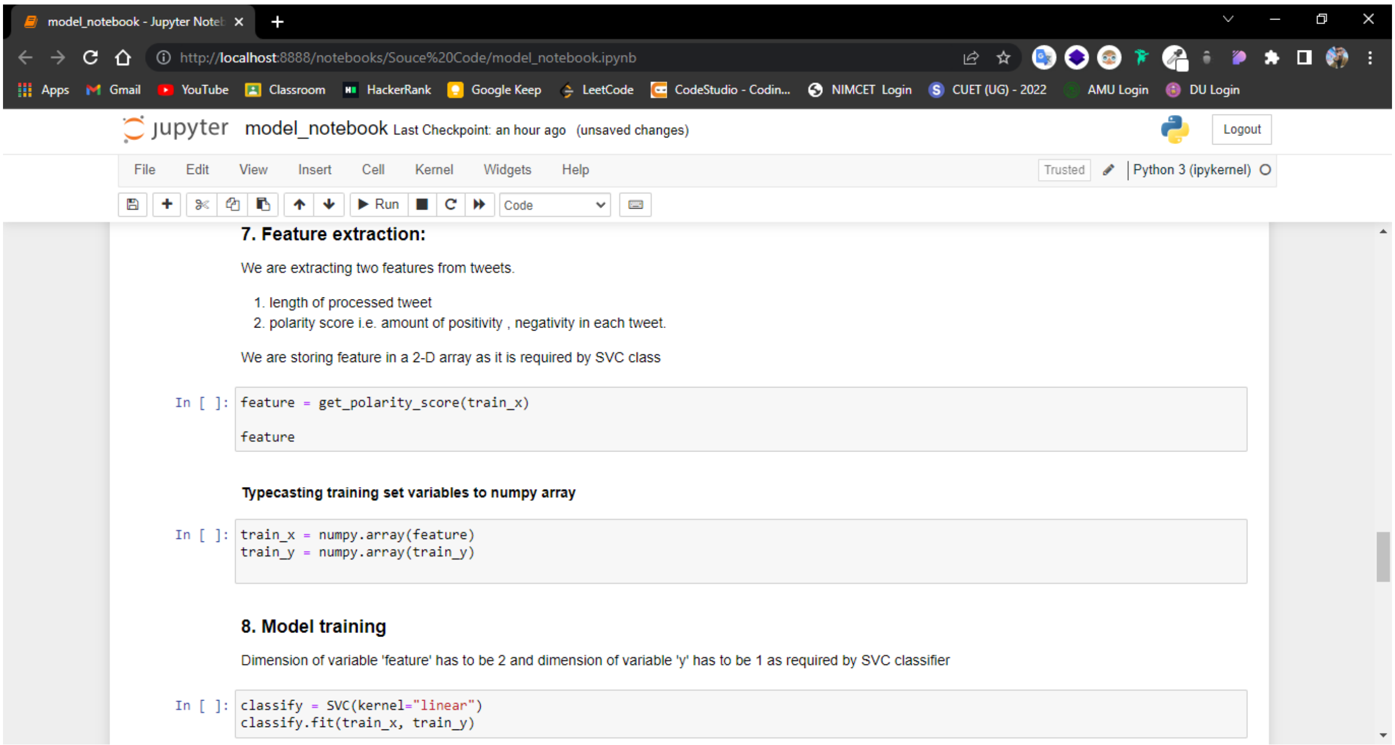
This increases out-of-sample accuracy of the model

```
In [ ]: train_x, test_x, train_y, test_y = train_test_split(processed_x, y, test_size=0.3, random_state=42)
```

6. Taking user input i.e. Tweet by actual user and then analysing its sentiment.

Given that user has to provide actual sentiment of the Tweet in order to perform evaluation metrics of the model.

```
In [ ]: tweet_input = input("Enter your tweet: ")
        test_x.append(" ".join(process_tweet(tweet_input)))

        print("What is the actual Sentiment of the Tweet? Is the tweet covid-positive or covid-negative")
        print("Input is essential for calculating model accuracy")
        sentiment_input = input('Type: 0 for negative, 1 for positive: ')
        senti = [0, 1]


        test_y = list(test_y)
        while int(sentiment_input) not in senti:
            sentiment_input = input('Wrong input. Try again!: ')

        test_y.append(int(sentiment_input))
```


model_notebook - Jupyter Noteb
http://localhost:8888/notebooks/Souce%20Code/model_notebook.ipynb
Apps Gmail YouTube Classroom HackerRank Google Keep LeetCode CodeStudio - Codin... NIMCET Login CUET (UG) - 2022 AMU Login DU Login
jupyter model_notebook Last Checkpoint: an hour ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

7. Feature extraction:

We are extracting two features from tweets.

1. length of processed tweet
2. polarity score i.e. amount of positivity , negativity in each tweet.

We are storing feature in a 2-D array as it is required by SVC class

```
In [ ]: feature = get_polarity_score(train_x)

        feature
```

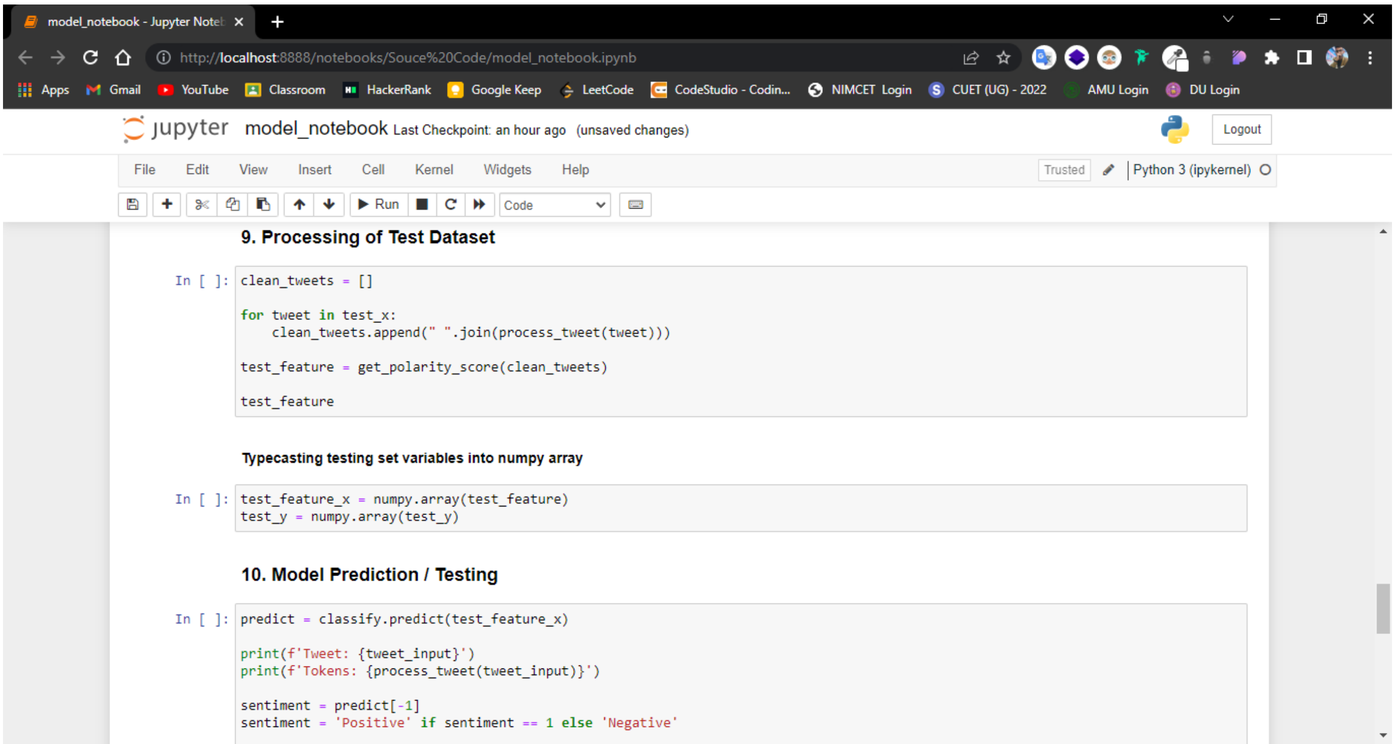
Typecasting training set variables to numpy array

```
In [ ]: train_x = numpy.array(feature)
        train_y = numpy.array(train_y)
```

8. Model training

Dimension of variable 'feature' has to be 2 and dimension of variable 'y' has to be 1 as required by SVC classifier

```
In [ ]: classify = SVC(kernel="linear")
        classify.fit(train_x, train_y)
```


model_notebook - Jupyter Noteb
http://localhost:8888/notebooks/Souce%20Code/model_notebook.ipynb
Apps Gmail YouTube Classroom HackerRank Google Keep LeetCode CodeStudio - Codin... NIMCET Login CUET (UG) - 2022 AMU Login DU Login
jupyter model_notebook Last Checkpoint: an hour ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

9. Processing of Test Dataset

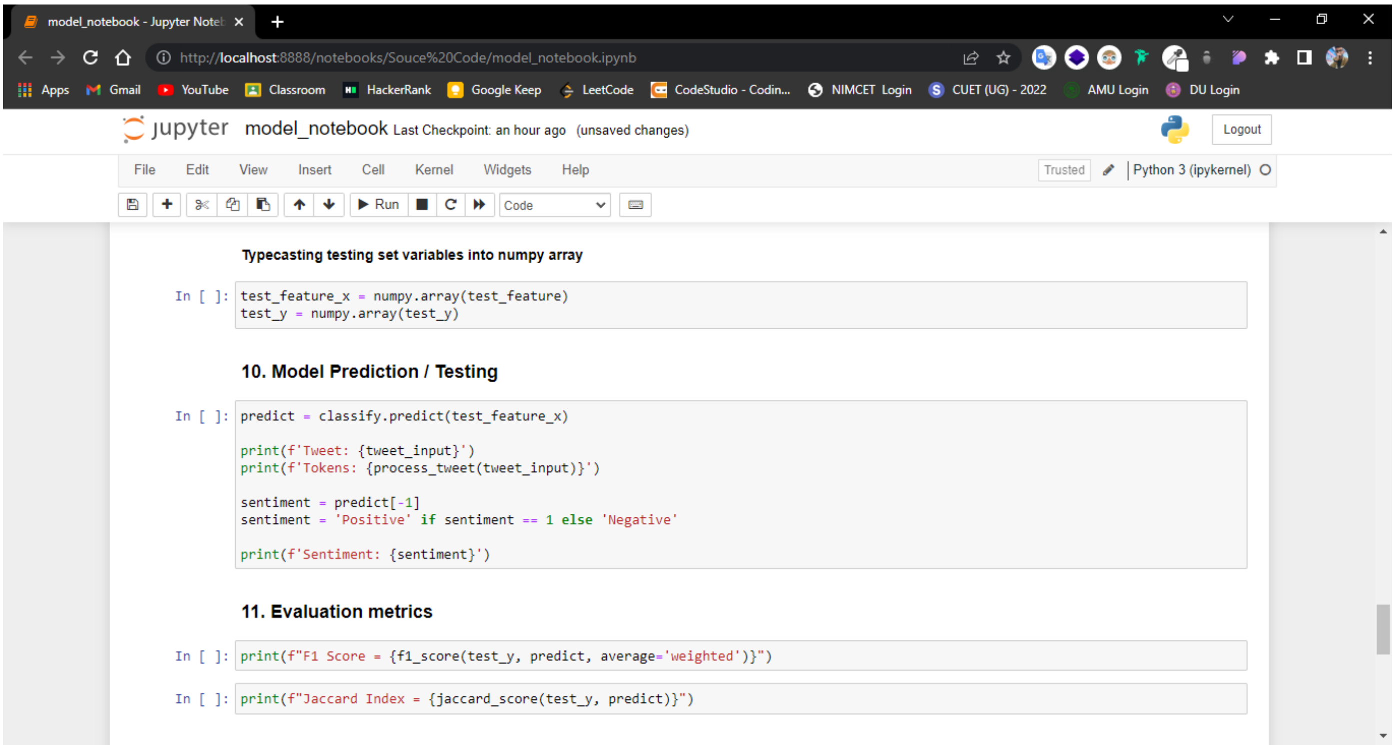
```
In [ ]: clean_tweets = []

        for tweet in test_x:
            clean_tweets.append(" ".join(process_tweet(tweet)))

        test_feature = get_polarity_score(clean_tweets)

        test_feature
```

Typecasting testing set variables into numpy array

```
In [ ]: test_feature_x = numpy.array(test_feature)
        test_y = numpy.array(test_y)
```

10. Model Prediction / Testing

```
In [ ]: predict = classify.predict(test_feature_x)

        print(f'Tweet: {tweet_input}')
        print(f'Tokens: {process_tweet(tweet_input)}')

        sentiment = predict[-1]
        sentiment = 'Positive' if sentiment == 1 else 'Negative'
```



Typecasting testing set variables into numpy array

```
In [ ]: test_feature_x = numpy.array(test_feature)
        test_y = numpy.array(test_y)
```

10. Model Prediction / Testing

```
In [ ]: predict = classify.predict(test_feature_x)

        print(f'Tweet: {tweet_input}')
        print(f'Tokens: {process_tweet(tweet_input)}')

        sentiment = predict[-1]
        sentiment = 'Positive' if sentiment == 1 else 'Negative'

        print(f'Sentiment: {sentiment}')
```

11. Evaluation metrics

```
In [ ]: print(f"F1 Score = {f1_score(test_y, predict, average='weighted')}")
```

```
In [ ]: print(f"Jaccard Index = {jaccard_score(test_y, predict)}")
```

TESTING

Software Testing is a Process of executing a program with the intent of finding errors during the run-time of program. It a feasible task to try and find the errors (whose presence is assumed) in a program, as it is a destructive process. I have tried to understand the proposed system by detailed study of the various operations that will be performed by a system.

System analysis is the process of studying an existing system to determine how it works and how it meets user needs. System analysis lays the groundwork for improvements to the system. The analysis involves an investigation, which is turn usually involves establishing a relationship with the client (User), for whom the analysis is done, and with the Administrator of the system.

This analysis phase is more of a thinking process. In this phase, I have improved logical aspects of the system.

To develop the system, one must deal with errors, bugs, defects etc. in more seamless way than ever, in order to preserve the integrity of Project and also to maintain the flow of maintenance.

I did thorough examination of the system processes, gathering Operational data, understanding the information flow, finding out weaknesses and evolving solutions for overcoming the weaknesses of the system so as to achieve the goals.

During the analysis phase, I dealt with:

1. Data Gathering
2. Feature selection and extraction
3. Data Analysis

Gathering the data for the completion of the Project was hard, given the complexity of the Project and also finding the correct dataset to train the model took much of the development time.

Selecting the right feature is a crucial part of the modelling. A considerable amount of time is dedicated to this phase, in considering which feature to choose and which to ignore. After thorough examination, five important features have been selected for the purpose of model training.

1. Length of tweet
2. Positivity score of tweets
3. Negativity score of tweets

Once the Feature selection and extraction was done, Data Analysis started, leading to thorough examination of the Project to make less prone to bugs, errors, defects etc.

CHALLENGES AND FUTURE SCOPES

“There is always room for improvements”

There are lot of things that can be added to the Project in future to make it more dynamic with respect to time.

Following are the abilities that can be added to the Project to make more modern and also visually – appealing.

1. Making the project executable (.exe) rather than ipython notebook (.ipynb) to eliminate the necessity of having python interpreter and jupyter notebook
   pre – installed in user’s system.
2. Making the user-interface GUI by utilizing the concepts of UI/UX (Colour Theory, Choosing right font style).
3. Making model more accurate by updating the parameter after each loss.

The challenge here will be adding the features in the Project without making the Project complex which can result in poor maintainability.

Challenges can be overcome by refactoring the Project from time to time to increase Code Maintainability.

CONCLUSION

The main objective of the project was to develop a Tweet Classifier which utilizes lesser resources but does not compromise with user-experience (UX).

I had taken a wide range of literature review in order to achieve all the tasks, where I came to know about some of the products that are existing in the market. I made detailed research in that path to uncover the loop holes that the existing systems are facing and to eradicate them in this Project. In the process of research, I came to know about the latest technologies and different algorithms, some of which I used in this Project and some of which are going to help me in foreseeable future.

MILESTONE

S. No.	Project Activity	Estimated Start Date	Estimated End Date
1	Synopsis Submission	22/03/2022	25/03/2022
2	Presentation Submission	23/03/2022	25/03/2022
3	Project Report Submission	01/05/2022	05/05/2022

MEETING WITH THE SUPERVISOR

Date of Meet	Mode	Comments by the Supervisor	Signature of the Supervisor
10/03/2022	Offline		
24/03/2022	Offline		
17/05/2022	Offline		

REFERENCES

1. https://app.diagram.net/
2. Kaggle.com/pre-processing-in-NLP
3. Kaggle.com/sentiment-analysis-using-logistic-regression
4. https://scikit-learn.org/stable/