Configuration Manual

MSc Research Project
Data Analytics

Jaiprakash Chandraker
Student ID: x22213546

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Jaiprakash Chandraker
Student ID:	x22213546
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Vladimir Milosavljevic
Submission Due Date:	10/08/2024
Project Title:	Configuration Manual
Word Count:	2000
Page Count:	13

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	21st August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Jaiprakash Chandraker
x22213546

1 Introduction

The manual includes step-by-step guidelines on how to operate depending on the software and library versions, required computer specifications, RAM volume, and memory space needed for the project. The tutorial contains information on how to install Google Collab and how to link Google Drive to it. The manual also states clearly that one should prefill the project with libraries. These specifications are required to predict the sentiment with the help of the Support Vector Machine (SVM) model Pisner and Schnyer (2020).

2 Software and Hardware Requirements

This part provides a comprehensive overview of software and hardware specifications needed for a local machine.

2.1 Hardware Specification of Local Machine

Figure 1: details about software specifications of the local machine to implement or execute the project. Figure 2: shows about hardware specifications as per the minimum requirement for executing the project.

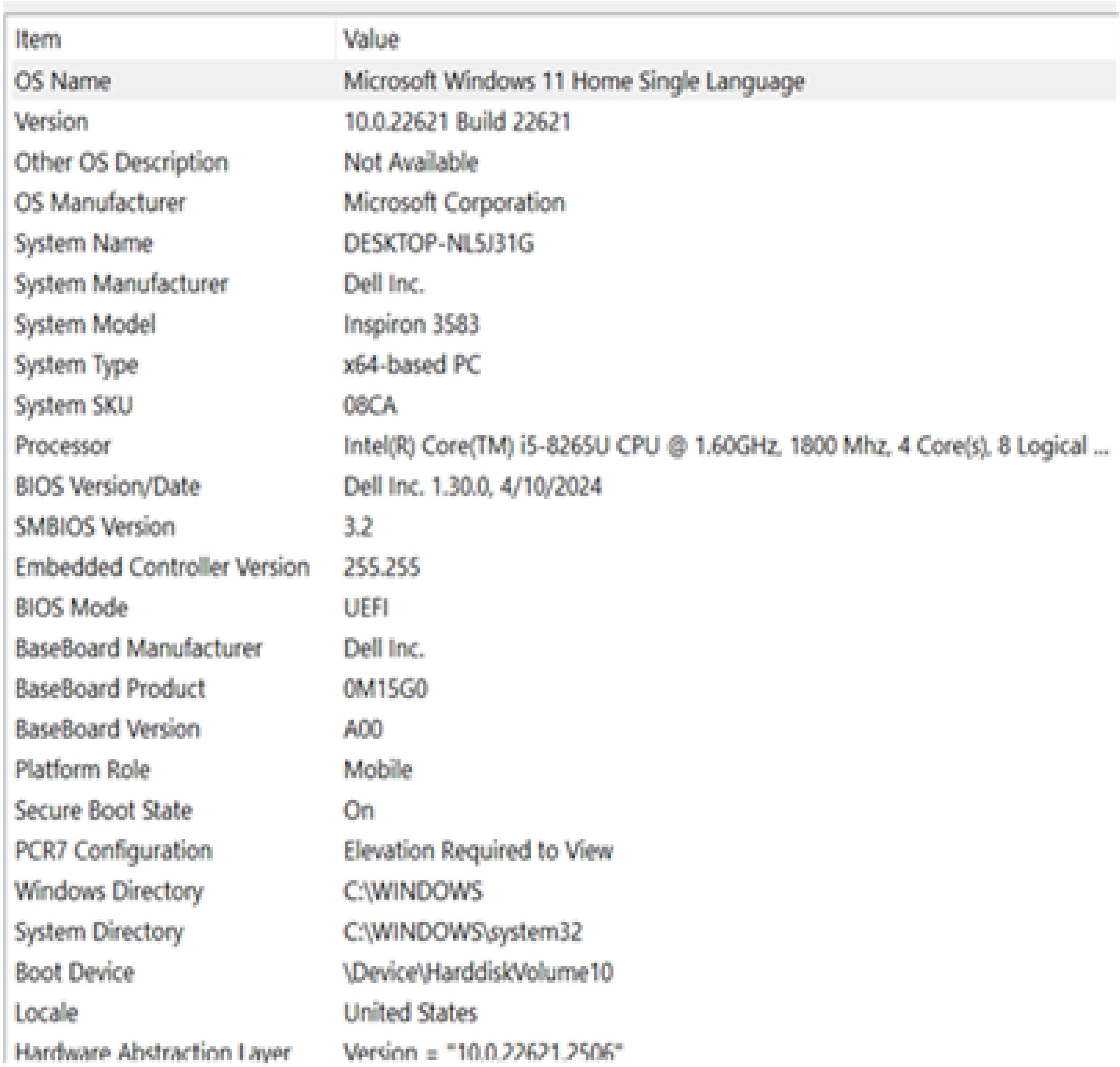
3 Cloud Environment Specifications

3.1 Cloud Service Provider and Environment Details

This project uses a Google Cloud Platform (GCP) environment with an n1-standard-4 instance, providing 4 vCPUs and 15 GB of memory, coupled with a 100 GB persistent disk for storage.

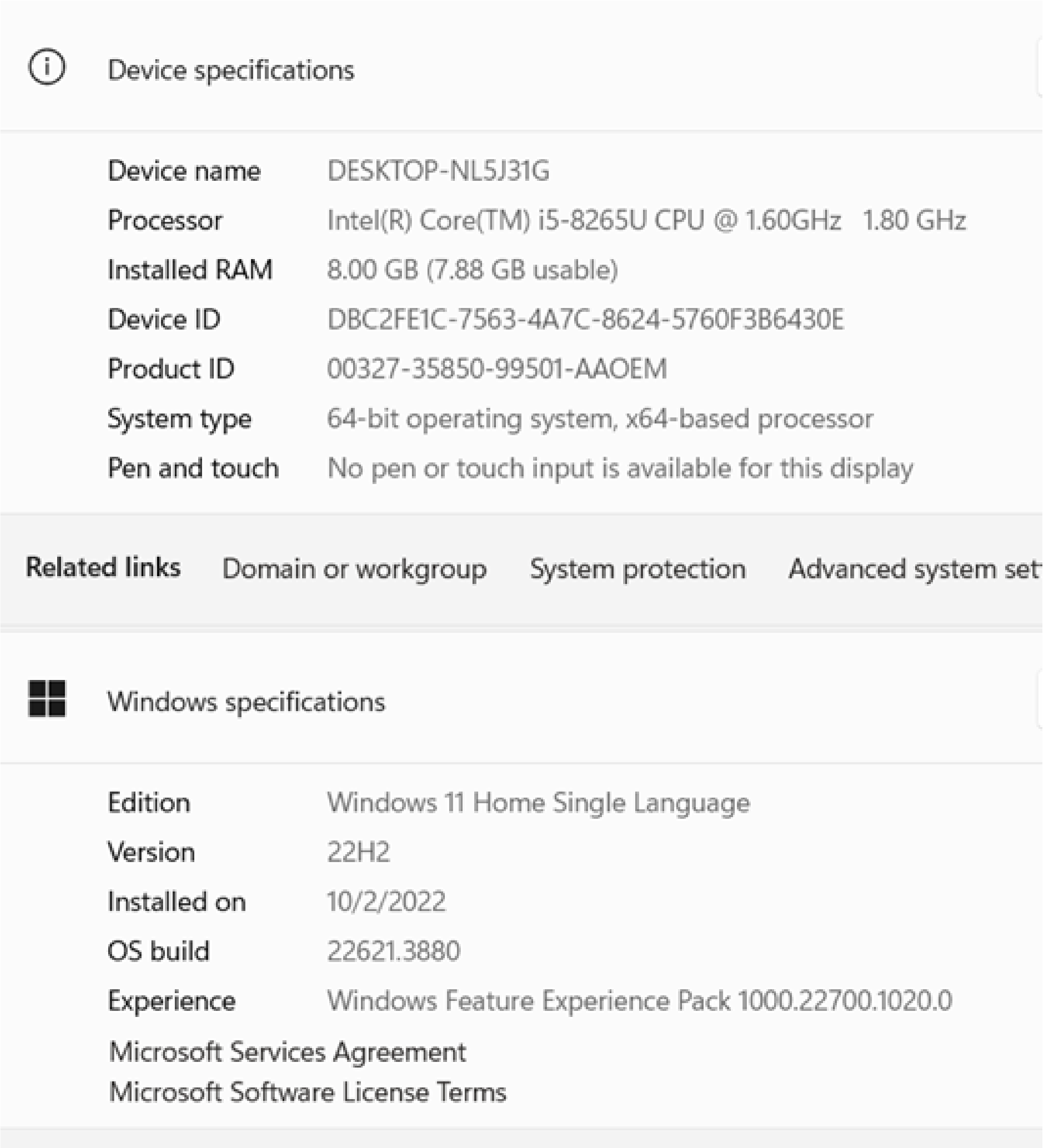
3.2 Operating System and Software Setup

The cloud instance is running Windows 10 with Python 3.8.10 pre-installed. Additional packages such as TensorFlow, PyTorch, and other machine learning libraries were installed using pip.



Item	Value
OS Name	Microsoft Windows 11 Home Single Language
Version	10.0.22621 Build 22621
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	DESKTOP-NL5J31G
System Manufacturer	Dell Inc.
System Model	Inspiron 3583
System Type	x64-based PC
System SKU	08CA
Processor	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz, 1800 Mhz, 4 Core(s), 8 Logical ...
BIOS Version/Date	Dell Inc. 1.30.0, 4/10/2024
SMBIOS Version	3.2
Embedded Controller Version	255.255
BIOS Mode	UEFI
BaseBoard Manufacturer	Dell Inc.
BaseBoard Product	0M15G0
BaseBoard Version	A00
Platform Role	Mobile
Secure Boot State	On
PCR7 Configuration	Elevation Required to View
Windows Directory	C:\WINDOWS
System Directory	C:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume10
Locale	United States
Hardware Abstraction Layer	Version = "10.0.22621.2506"

Figure 1: Software Specifications



Device specifications

Device name	DESKTOP-NL5J31G
Processor	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
Installed RAM	8.00 GB (7.88 GB usable)
Device ID	DBC2FE1C-7563-4A7C-8624-5760F3B6430E
Product ID	00327-35850-99501-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

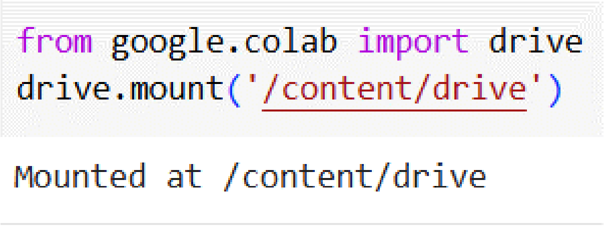
Related links Domain or workgroup System protection Advanced system set

Windows specifications

Edition	Windows 11 Home Single Language
Version	22H2
Installed on	10/2/2022
OS build	22621.3880
Experience	Windows Feature Experience Pack 1000.22700.1020.0

Microsoft Services Agreement
Microsoft Software License Terms

Figure 2: Hardware Specification



```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Figure 3: Mounting Google Drive to Colab

3.3 Configuration in Google Colab

3.4 Accessing Configuration Files

Google Colab provides a flexible environment for managing configuration files and setting parameters required for the project. configuration files are accessed and used.

3.5 Mounting Google Drive

To access configuration files stored in Google Drive, the drive must be mounted in Google Colab. This allows direct access to files as if they were local.(Figure:3)

3.6 Installation Libraries on Google Colab

This section provides the steps for installing the necessary libraries and utilities for the sentiment analysis project using Google Colab, along with a brief explanation of the importance of each library. In this Research project Installed necessary libraries are shown below

- pip install Numpy
- pip install pandas
- pip install seaborn
- pip install wordcloud
- pip install matplotlib
- pip install sci-kit-learn

3.7 Importance of Libraries

3.7.1 re (Regular Expressions)

The re-module helps to support regular expressions in Python, allowing for string search, matching patterns, and manipulation. It is used for tasks such as parsing of text, data extraction, and validation He (2012).

3.7.2 Numpy (Numerical Python)

NumPy is a fundamental package for numerical calculation in Python. It provides calculation for arrays and matrices, along with a wide variety of mathematical functions to operate on these arrays efficiently Balahur (2013).

3.7.3 Pandas

pandas is a powerful data manipulation and analysis library. It helps to read structures like Excel, CSV, and JSON. After reading files It converts into a data frame for proper structure, It's a sorting problem out of data manipulation, data alignment, reshaping, merging, etc Abid et al. (2019).

3.7.4 Seaborn

Seaborn is a data visualization library for statistic graph plotting in Python. It gives wonderful default styles and colors to make statistical plots more attractive and meaningful.

3.7.5 WordCloud

Wordclouds are tools that utilize data visualizations to depict textual information. Words' dimensions in the picture reveal how often or importantly they appear. You can use a word cloud to emphasize significant textual data points. The data originating from social networking sites can also be evaluated using word clouds.

3.7.6 Matplotlib. plot

Matplotlib is a powerful plotting library in Python used for creating static, animated, and interactive visualizations. Matplotlib's primary purpose is to provide users with the tools and functionality to represent data graphically, making it easier to analyze and understand.

3.7.7 Sklearn. svm.LinearSVC

The LinearSVC class from sklearn.svm uses Support Vector Classification for linear kernels. It is used for performing multiclass classification tasks and is known for its efficiency with big datasets Abid et al. (2019).

3.7.8 Sklearn.model selection. train test split

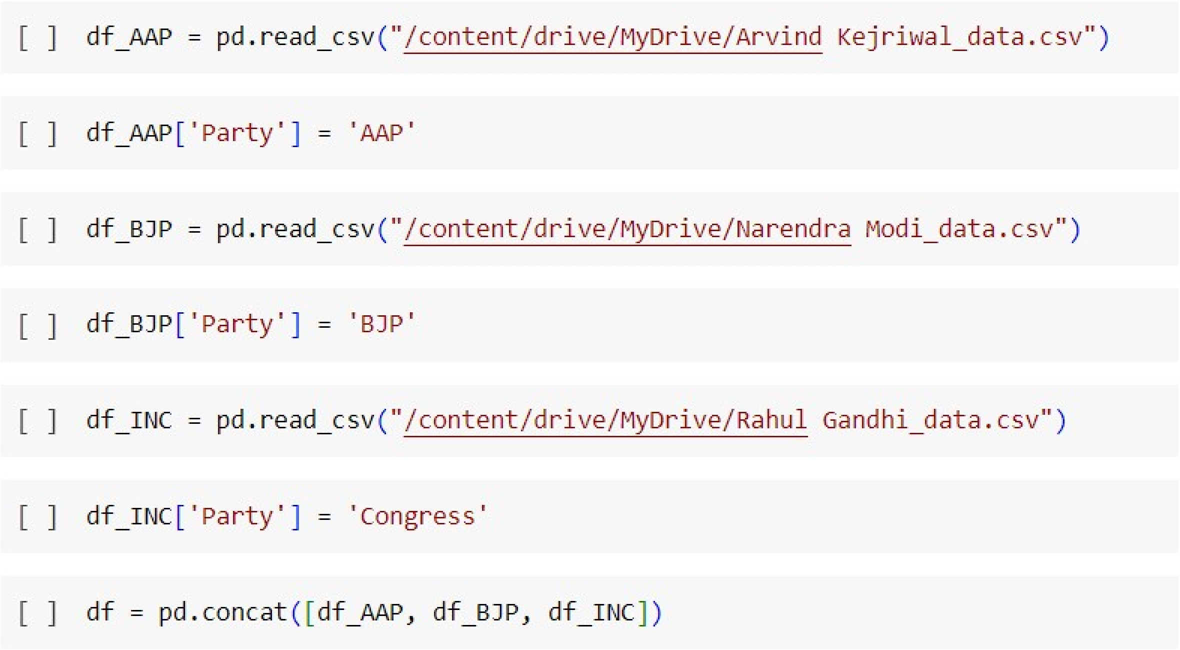
The train test split function from sklearn.model selection is used to split a dataset into training and testing subsets. This is helpful for evaluating the performance of a machine-learning model for useful data.

3.7.9 Sklearn. feature extraction. text.TfidfVectorizer

The TfidfVectorizer class from sklearn.feature extraction.text converts a collection of raw data into a matrix of TF-IDF (Term Frequency-Inverse Document Frequency) features. It is commonly used in data mining and information extraction to transform text data into numerical features.

3.7.10 Sklearn. metrics.confusion matrix

The confusion matrix function from sklearn.metrics computes the confusion matrix to evaluate the accuracy of a classification. It provides insight into the performance of the



```
df_AAP = pd.read_csv("/content/drive/MyDrive/Arvind Kejriwal_data.csv")
```

```
df_AAP['Party'] = 'AAP'
```

```
df_BJP = pd.read_csv("/content/drive/MyDrive/Narendra Modi_data.csv")
```

```
df_BJP['Party'] = 'BJP'
```

```
df_INC = pd.read_csv("/content/drive/MyDrive/Rahul Gandhi_data.csv")
```

```
df_INC['Party'] = 'Congress'
```

```
df = pd.concat([df_AAP, df_BJP, df_INC])
```

Figure 4: The data is stored in CSV files located on Google Drive.

classification model by showing the counts of true positive, true negative, false positive, and false negative predictions.

3.7.11 Sklearn. metrics.classification report

The classification report function from sklearn.metrics creates a text report showing the main classification metrics, including precision, recall, F1-score, and support for each class. It is useful for evaluating the quality of predictions made by a classification model.

4 Applications used to run/execute the project

To execute or run the implemented code to complete the project, below list of applications used

- Google Colab Python 3.12.4
- Visual Studio Code
- Microsoft Excel
- Microsoft Power BI

5 Data Files

This section details the steps and code required to import and prepare data for analysis. These Data files are datasets containing information about three political parties: Arvind Kejriwal (AAP), Narendra Modi (BJP), and Rahul Gandhi (Congress). (Figure 4:)

6 Coding files

Real-Time Sentiment Analysis and Interactive Dashboard Deployment for Social Media Discourse.ipynb.

This is the file used for Sentiment Analysis for the project. It's divided into multiple sections.

- Import Libraries
- Read and Load Data Set
- Exploratory Data Analysis
- Sentiment Analysis
- Data Visualization
- Transforming dataset – Using TF-IDF Vectorizer
- Function for Model Evaluation
- Model Building
- Testing a Model

6.1 Import Libraries

(Figure 5: Import Libraries)

6.2 Exploratory Data Analysis

(Figure 6: Data Preprocessing)

7 Sentiment Analyzer

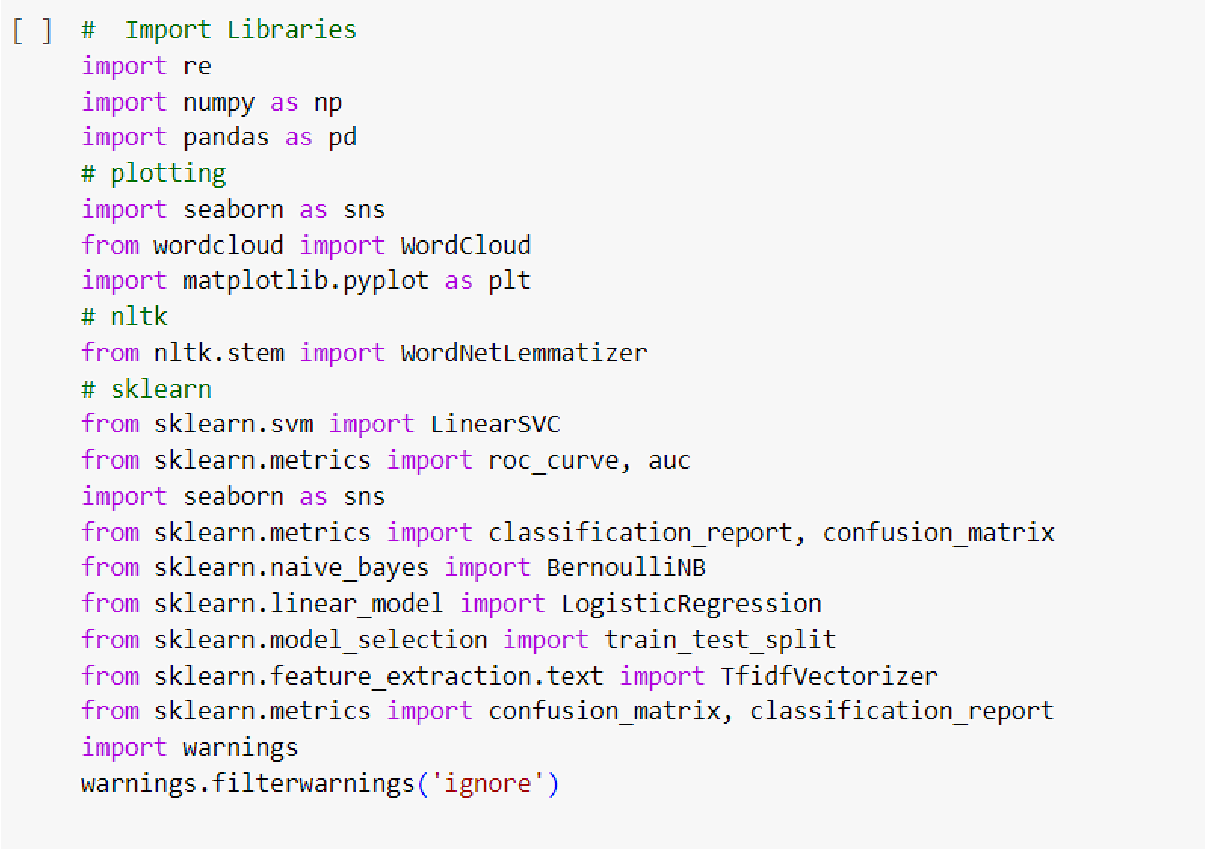
The function is used for Sentiment text upon tweets. Gives a score and is divided into three parts positive, negative, and neutral Balahur (2013). (Figure 7: Sentiment Analyzer Function)

8 Data Visualization

(Figure 8: Data Visualization)

9 Transforming dataset – Using TF-IDF Vectorizer

It will transform using TF – IDF Vectorizer to the matrix for extracting features. [2](Figure 9: TF – IDF Vectorizer 9)



```
[ ] #  Import Libraries
    import re
    import numpy as np
    import pandas as pd
    # plotting
    import seaborn as sns
    from wordcloud import WordCloud
    import matplotlib.pyplot as plt
    # nltk
    from nltk.stem import WordNetLemmatizer
    # sklearn
    from sklearn.svm import LinearSVC
    from sklearn.metrics import roc_curve, auc
    import seaborn as sns
    from sklearn.metrics import classification_report, confusion_matrix
    from sklearn.naive_bayes import BernoulliNB
    from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import train_test_split
    from sklearn.feature_extraction.text import TfidfVectorizer
    from sklearn.metrics import confusion_matrix, classification_report
    import warnings
    warnings.filterwarnings('ignore')
```

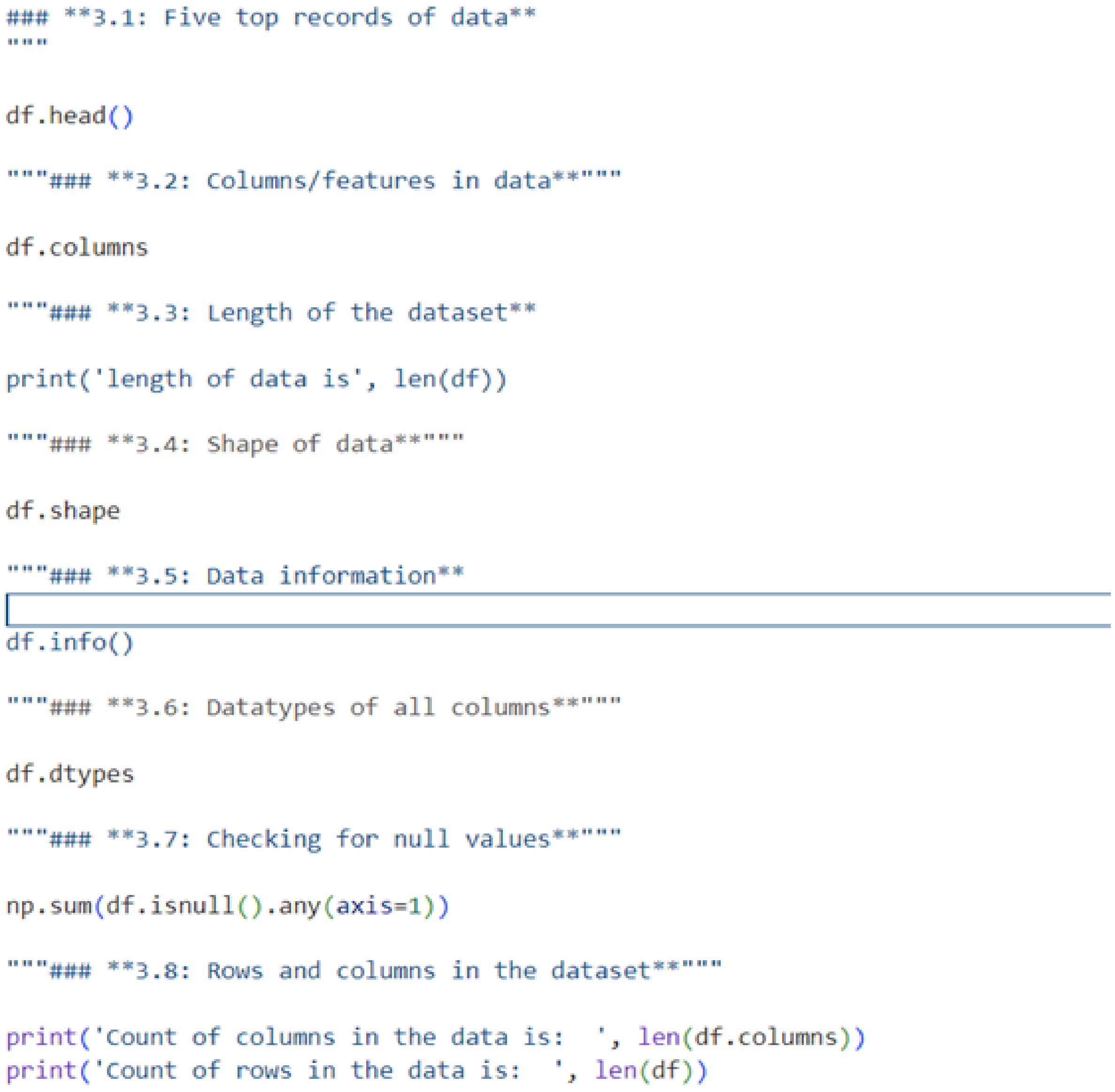
Figure 5: Import Libraries

10 Function for Model Evaluation

Model Evaluation function for Confusion Matrix, heatmap, prediction values, and actual values Balahur (2013). (Figure 11: Model Evaluation 10)

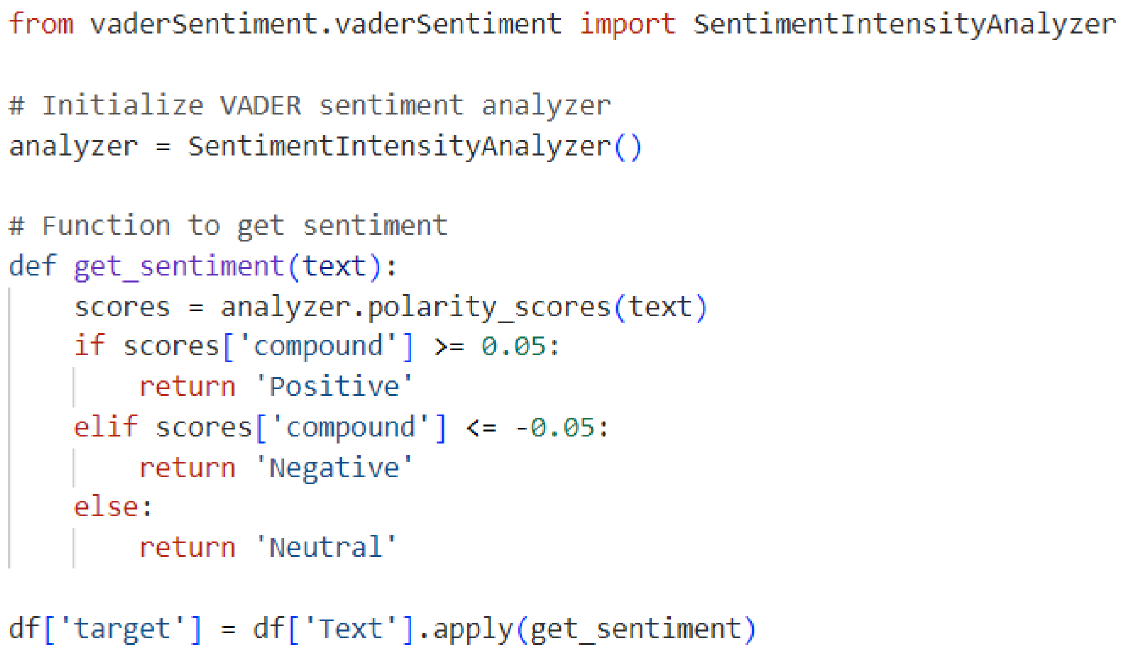
10.1 Testing a Model

(Figure 12: SVM Model Test)



```
### **3.1: Five top records of data**
"""

df.head()

"""### **3.2: Columns/features in data**"""

df.columns

"""### **3.3: Length of the dataset**

print('length of data is', len(df))

"""### **3.4: Shape of data**"""

df.shape

"""### **3.5: Data information**

df.info()

"""### **3.6: Datatypes of all columns**"""

df.dtypes

"""### **3.7: Checking for null values**"""

np.sum(df.isnull().any(axis=1))

"""### **3.8: Rows and columns in the dataset**"""

print('Count of columns in the data is:  ', len(df.columns))
print('Count of rows in the data is:  ', len(df))
```

Figure 6: Data Preprocessing



```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# Initialize VADER sentiment analyzer
analyzer = SentimentIntensityAnalyzer()

# Function to get sentiment
def get_sentiment(text):
    scores = analyzer.polarity_scores(text)
    if scores['compound'] >= 0.05:
        return 'Positive'
    elif scores['compound'] <= -0.05:
        return 'Negative'
    else:
        return 'Neutral'

df['target'] = df['Text'].apply(get_sentiment)
```

Figure 7: Sentiment Analyzer Function

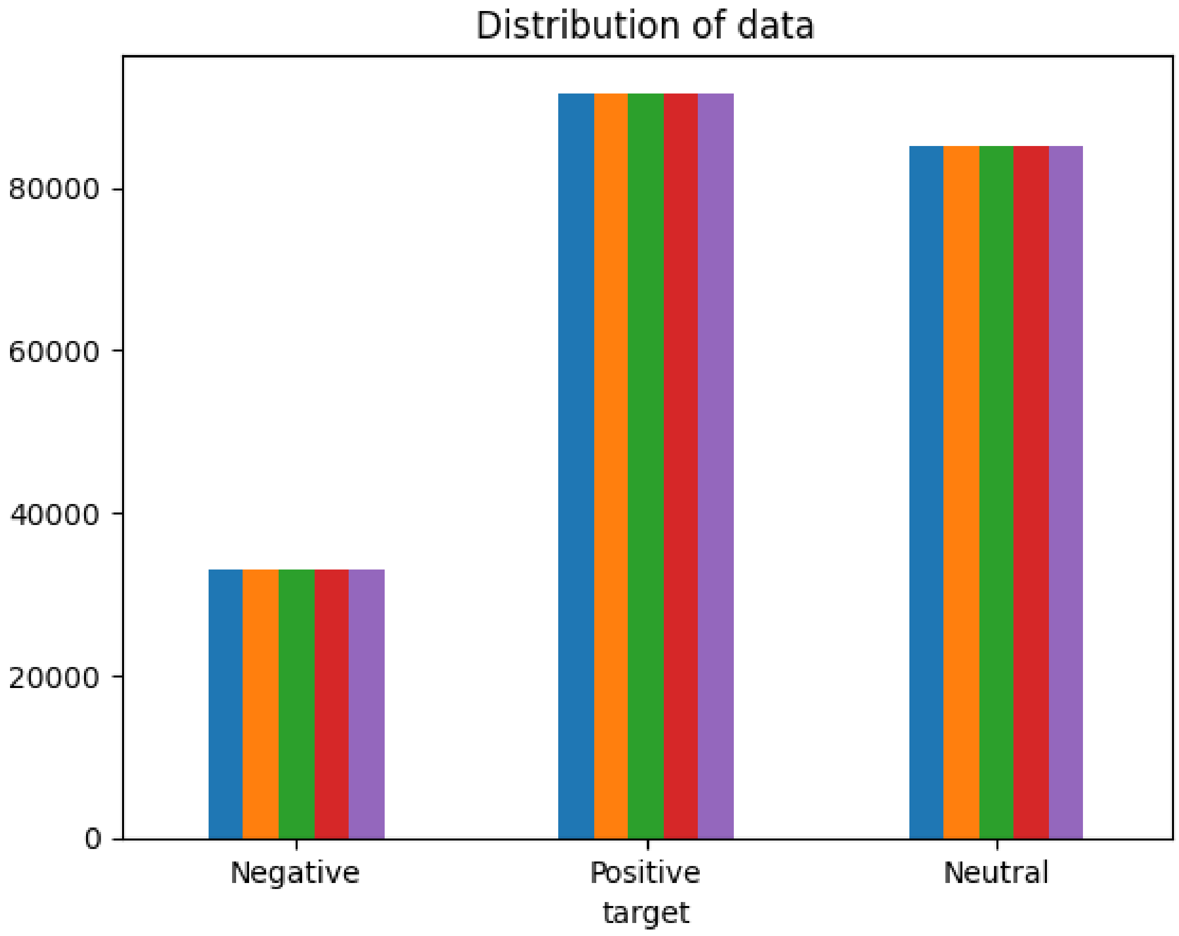
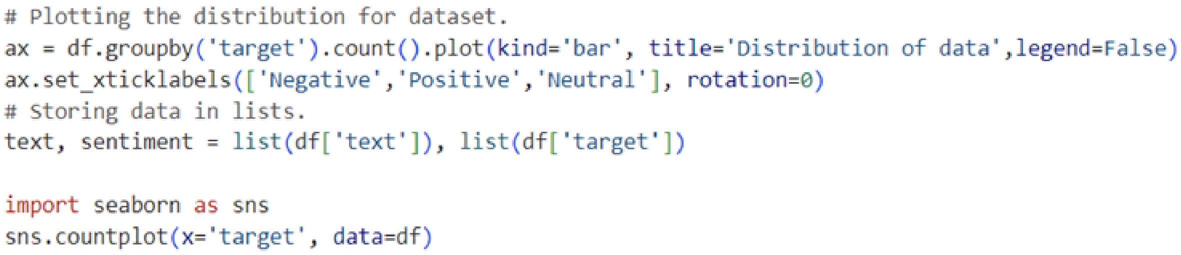
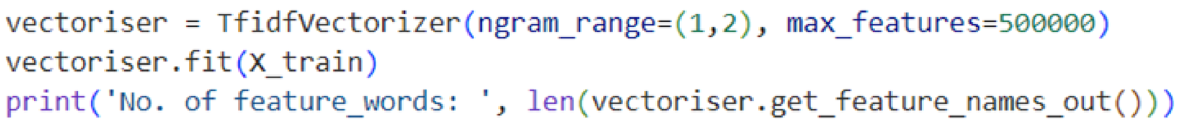


Figure 8: Data Distribution



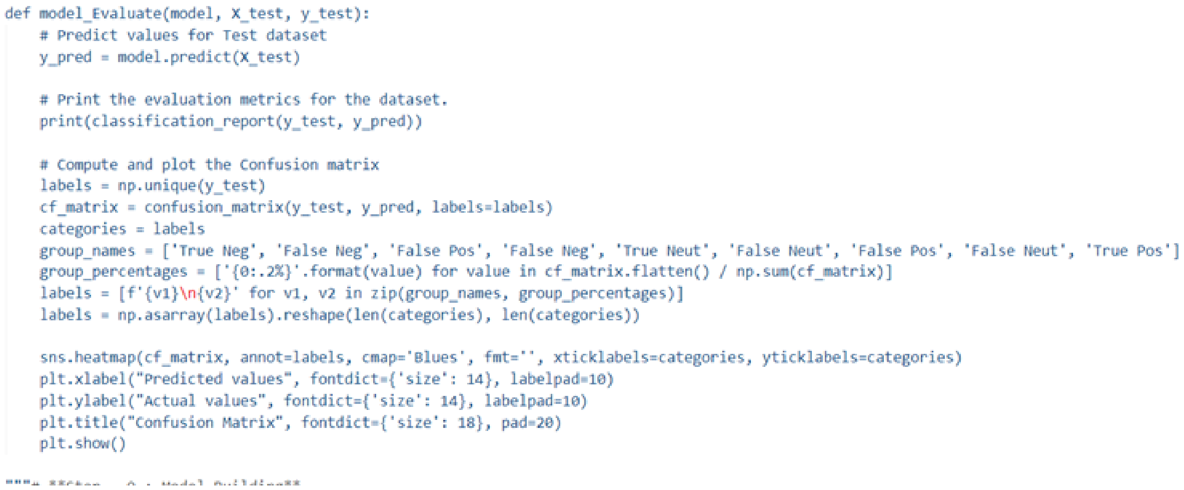
```
# Plotting the distribution for dataset.
ax = df.groupby('target').count().plot(kind='bar', title='Distribution of data',legend=False)
ax.set_xticklabels(['Negative','Positive','Neutral'], rotation=0)
# Storing data in lists.
text, sentiment = list(df['text']), list(df['target'])

import seaborn as sns
sns.countplot(x='target', data=df)
```

Figure 9: Data Visualization



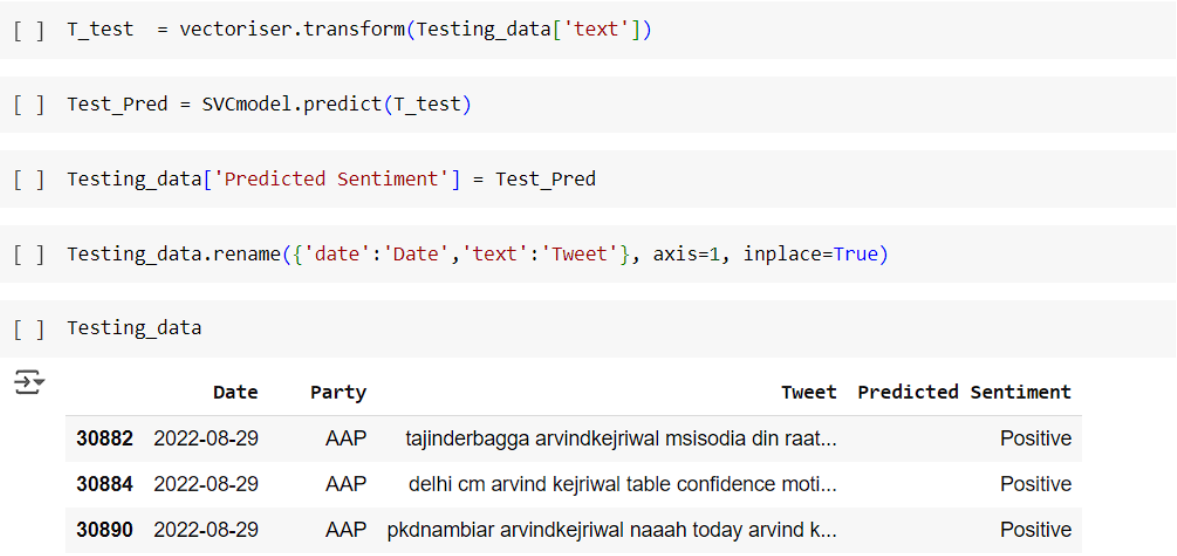
```
vectoriser = TfidfVectorizer(ngram_range=(1,2), max_features=500000)
vectoriser.fit(X_train)
print('No. of feature_words: ', len(vectoriser.get_feature_names_out()))
```

Figure 10: TF – IDF Vectorizer



```
def model_Evaluate(model, X_test, y_test):
    # Predict values for Test dataset
    y_pred = model.predict(X_test)

    # Print the evaluation metrics for the dataset.
    print(classification_report(y_test, y_pred))

    # Compute and plot the Confusion matrix
    labels = np.unique(y_test)
    cf_matrix = confusion_matrix(y_test, y_pred, labels=labels)
    categories = labels
    group_names = ['True Neg', 'False Neg', 'False Pos', 'False Neg', 'True Neut', 'False Neut', 'False Pos', 'False Neut', 'True Pos']
    group_percentages = ['{0:.2%}'.format(value) for value in cf_matrix.flatten() / np.sum(cf_matrix)]
    labels = [f'{v1}\n{v2}' for v1, v2 in zip(group_names, group_percentages)]
    labels = np.asarray(labels).reshape(len(categories), len(categories))

    sns.heatmap(cf_matrix, annot=labels, cmap='Blues', fmt='', xticklabels=categories, yticklabels=categories)
    plt.xlabel("Predicted values", fontdict={'size': 14}, labelpad=10)
    plt.ylabel("Actual values", fontdict={'size': 14}, labelpad=10)
    plt.title("Confusion Matrix", fontdict={'size': 18}, pad=20)
    plt.show()

"""# **Step - [illegible] : Model Building**
```

Figure 11: Model Evaluation



```
[ ] T_test  = vectoriser.transform(Testing_data['text'])
```

```
[ ] Test_Pred = SVCmodel.predict(T_test)
```

```
[ ] Testing_data['Predicted Sentiment'] = Test_Pred
```

```
[ ] Testing_data.rename({'date':'Date','text':'Tweet'}, axis=1, inplace=True)
```

```
[ ] Testing_data
```

	Date	Party	Tweet	Predicted Sentiment
30882	2022-08-29	AAP	tajinderbagga arvindkejriwal msisodia din raat...	Positive
30884	2022-08-29	AAP	delhi cm arvind kejriwal table confidence moti...	Positive
30890	2022-08-29	AAP	pkdnambiar arvindkejriwal naaah today arvind k...	Positive

Figure 12: SVM Model Test

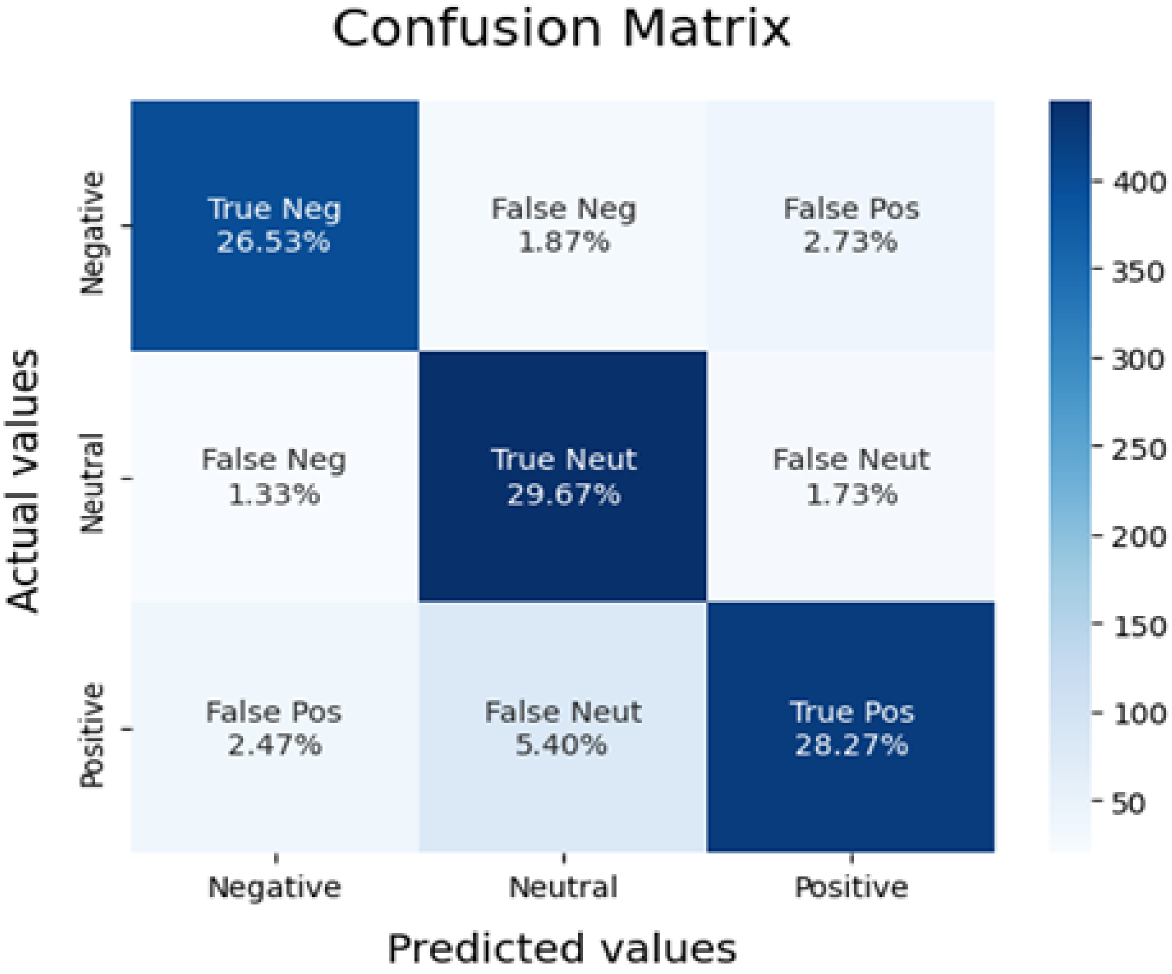
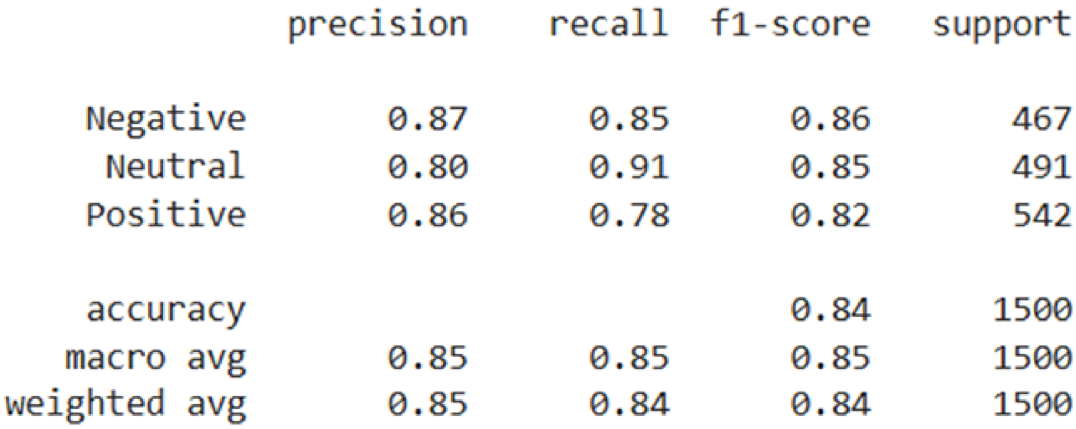


Figure 13: Confusion Matrix



```
              precision    recall  f1-score   support

    Negative       0.87      0.85      0.86       467
     Neutral       0.80      0.91      0.85       491
    Positive       0.86      0.78      0.82       542

    accuracy                           0.84      1500
   macro avg       0.85      0.85      0.85      1500
weighted avg       0.85      0.84      0.84      1500
```

Figure 14: Classification Report

References

Abid, F., Alam, M., Yasir, M. and Li, C. (2019). Sentiment analysis through recurrent variants latterly on convolutional neural network of twitter, *Future Generation Computer Systems* **95**: 292–308.

Balahur, A. (2013). Sentiment analysis in social media texts, *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pp. 120–128.

He, Y. (2012). Incorporating sentiment prior knowledge for weakly supervised sentiment analysis, *ACM Transactions on Asian Language Information Processing (TALIP)* **11**(2): 1–19.

Pisner, D. A. and Schnyer, D. M. (2020). Chapter 6 - support vector machine, *in* A. Mechelli and S. Vieira (eds), *Machine Learning*, Academic Press, pp. 101–121.
URL: *https://www.sciencedirect.com/science/article/pii/B9780128157398000067*