National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Ayusha Eknath Kashilkar
Student ID: x20239343

School of Computing
National College of Ireland

Supervisor: Dr.Abid Yaqoob

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ayusha Eknath Kashilkar
Student ID:	x20239343
Programme:	Data Analytics
Year:	2022
Module:	Msc Research Project
Supervisor:	Dr Abid Yaqoob
Submission Due Date:	15/12/2022
Project Title:	Configuration Manual
Word Count:	1022
Page Count:	12

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:	Ayusha Eknath Kashilkar
Date:	15th December 2022

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

Ayusha Eknath Kashilkar
x20239343

1 Introduction

In this configuration manual , the process from design – environment to implementation and results has been discussed thoroughly to determine the sentiment analysis techniques used with machine learning and deep learning models on amazon reviews of different sectors. The configuration manual also includes the details steps of programming parts like libraries, data transformation and modelling. Every model applied on reviews having multiple evaluation methods have been discussed in this document.

2 Environment setup

2.1 System specification

The Implementation was started on Juypter notebook from Anaconda on System of 16 GB ram and 512 SSD Windows 10, but due to limitation in processing and time taken the Implementation was shifted to Google Collaboratory also known as Google Colab. Google Colab is free and runs code on Google servers. It also offers GPU for faster execution. Since when google colab reconnects the run time , the data file saved locally over google colab are lost. Therefore the Dataset were uploaded into Google drive which was then used by Google Colab.

2.2 Technical specification

The implementation was done in python programming language was used in this research. Following are the packages that were used:-

1. Numpy
2. Pandas
3. Seaborn
4. Matplotlib
5. Sklearn
6. Ploty.express
7. NLTK
8. Worldcloud
9. String

10. NRCLEX
11. keras
12. Tensor flow

3 Dataset

The data was downloaded in zip format from Amazon official reviews dataset site - https://s3.amazonaws.com/amazon-reviews-pds/tsv/index.txt. The data was then unzipped on the system and .tsv file was extracted. The file was then changed into csv for preference of loading in python. There are total 5 datasets downloaded of different sectors of amazon review – Personal Appliance Care, mobile electronics, Electronics and Apparels. The data was then uploaded to google drive for later to be fetched from Google Colab.

4 Implementation

In this section, the steps of implementation will be discussed. For each dataset a Google colab file was created for implementation. In this section the screenshot will be from different google colab notebooks and will be mentioned in caption. For a gist of models and their dataset are as follows:-

1. Apparels dataset has implementation of Random forest model
2. Mobile electronics have implementation of Bi-LSTM
3. Personal Care Appliances has implementation of SVM
4. Electronics have implementation of best 2 models – RF and SVM Note:- the data processing and data transformation steps for all the 4 dataset are same steps.

4.1 Data Pre-processing

- The data was loaded from google drive to google colab using pandas read_csv function

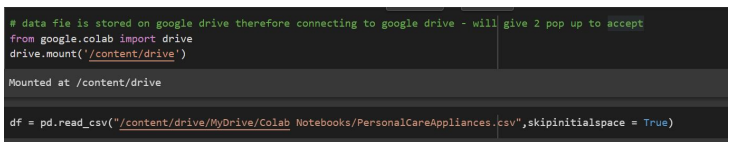

Figure 1: Loaded data

- The data had 7 unnamed columns with nan values. These columns were dropped using the drop function
- The null values in data frame were checked using isull() function with sum() function to get total count of null values per column. the null values were dropped.

```
df.isnull().sum()

marketplace          27
customer_id          27
review_id            40
product_id           45
product_parent       45
product_title        45
product_category     45
star_rating          45
helpful_votes        45
total_votes          45
vine                 50
verified_purchase    50
review_headline      54
review_body         396
review_date          90
dtype: int64
```

Figure 2: data cleaning ; null value removal

- As the review data contained space at beginning and end of the sentences. Strip() method from string library was used. From the reviews the emojis, stopwords and punctuation were removed

```
df['review_body'] = df['review_body'].str.strip()
```

Figure 3: data cleaning : whitespace removal

```
7] #remove emojis

   df['review_body'] = df.review_body.apply(lambda x: x.encode('ascii', 'ignore').decode('ascii')

8] #removing punctuation


   non_alpha = string.punctuation + '0123456789'

   df['review_body'] = df.review_body.apply(lambda x: x.translate(str.maketrans('','',non_alpha))
```

Figure 4: data cleaning : emojis and punctuation removal

4.2 Sentiment Analysis

NCRLex:-
The emotions were detected using the NCRLex library. As shown in the below Figure 5 the function emotion was created to detect emotion. As emotion are detected on the score the value of 0.0 is set to no emotion which are the emotions not being in any other sentiment quite neutral.

```
import nltk
nltk.download('punkt')

def emotion(x):
    text = NRCLex(x)
    if text.top_emotions[0][1] == 0.0:
        return "No emotion"
    else:
        return text.top_emotions[0][0]
df['Emotion'] = df['review_body'].apply(emotion)
df.head()
```

_votes	total_votes	vine	verified_purchase	review_headline	review_body	review_date	Emotion
0	0	N	Y	★ THESE REALLY DO WORK GREAT WITH SOME TWEAKING ★	These Really Do Work Great But You Do Need To ...	14-01-2013	No emotion
1	2	N	Y	Favorite for winter. Very warm!	I love this dress Absolute favorite for winter...	04-03-2014	positive
0	0	N	Y	Great Socks for the money.	Nice socks great colors just enough support fo...	12-07-2015	trust
					I bought this for my		

Figure 5: NCRLex model

VADER:-

1. Tokenization:-
   Using the word_tokenize and sent_tokenize function from nltk tokenize library, as shown in figure 6

```
df['tokenized_word'] = df['review_body'].apply(word_tokenize)

df['tokenized_sent'] = df['review_body'].apply(sent_tokenize)
```

```
df['tokenized_word']
```

```
1          [I, love, dress, Absolute, favorite, winter, H...
2          [Nice, socks, great, colors, enough, support, ...
3          [I, bought, husband, WOW, slick, hat, High, qu...
4               [Perfect, dress, customer, service, awesome]
5                 [Excellent, feet, skinny, years, old, boy]
                                  ...
1048570    [The, shirt, extremely, small, I, normally, we...
1048571    [I, order, shirt, times, order, get, size, wou...
1048572    [I, like, shirts, I, reorder, size, I, need, T...
1048573    [I, work, retail, worn, uniform, specially, or...
1048574    [Great, quality, beautiful, material, But, exp...
Name: tokenized_word, Length: 765155, dtype: object
```

```
df['tokenized_sent']
```

```
1          [I love dress Absolute favorite winter Heavy m...
2          [Nice socks great colors enough support wearin...
3          [I bought husband WOW slick hat High quality c...
4                   [Perfect dress customer service awesome]
5                     [Excellent feet skinny years old boy]
                                  ...
1048570    [The shirt extremely small I normally wears la...
1048571    [I order shirt times order get size would fit ...
1048572    [I like shirts I reorder size I need Thanks aw...
1048573    [I work retail worn uniform specially ordered ...
1048574    [Great quality beautiful material But expect s...
Name: tokenized_sent, Length: 765155, dtype: object
```

Figure 6: Tokenization

2. Lemmatizer
   WordnetLemmatizer is used from nltk.stem library for lemmatization of review_body The output was saved in other column. As shown in Figure 7

```
import nltk
nltk.download('omw-1.4')
nltk.download('wordnet')

w_tokenizer = nltk.tokenize.WhitespaceTokenizer()
lemmatizer = nltk.stem.WordNetLemmatizer()

def lemmatize_text(text):
    return [lemmatizer.lemmatize(w) for w in w_tokenizer.tokenize(text)]

df['text_lemmatized'] = df.review_body.apply(lemmatize_text)
```

Figure 7: Lemmatizer function

3. VADER :-

The Sentiment Intensity Analyzer of VADER has been used to detect sentiments the polarity score with compound value. If the compound value is greater than 0 then sentiment is mapped as positive or else mapped as negative. As shown in Figure 8

```
import nltk
nltk.download('vader_lexicon')

sia = SentimentIntensityAnalyzer()

df["sentiment_label"] = df["review_body"].apply(lambda x: "pos" if sia.polarity_scores(x)["c

df.head(20)
```

	tokenized_sent	text_lemmatized	date	month	year	review_len	word_count	sentiment_label
,	[I love dress Absolute favorite winter Heavy m...	[I, love, dress, Absolute, favorite, winter, H...	04	03	2014	252	45	pos
,	[Nice socks great colors enough support wearin...	[Nice, sock, great, color, enough, support, we...	12	07	2015	65	10	pos
,	[I bought husband WOW slick hat High quality c...	[I, bought, husband, WOW, slick, hat, High, qu...	03	06	2015	131	19	pos
,	[Perfect dress	[Perfect dress						

Figure 8: NCRLex model

4.3 Exploratory Data Analysis

- Emotions were mapped with year to find the trend of emotion as shown in Figure 9

```
df.groupby(['year','Emotion'])['Emotion'].count().unstack().plot(legend=True)
plt.title('Year and Emotion count')
plt.xlabel('Year')
plt.ylabel('Emotion count')
plt.show()
```

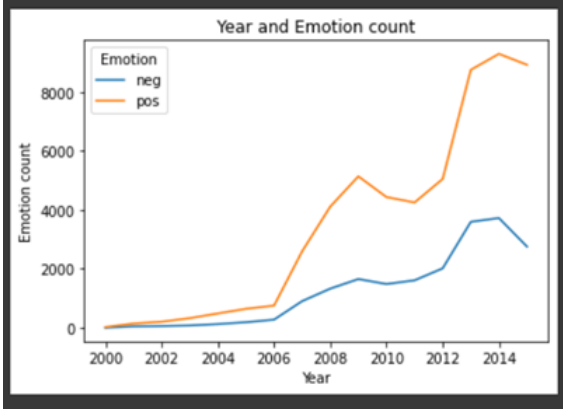

Figure 9: Trend of emotion and year

- Created word cloud with positive words from reviews as shown in Figure 10

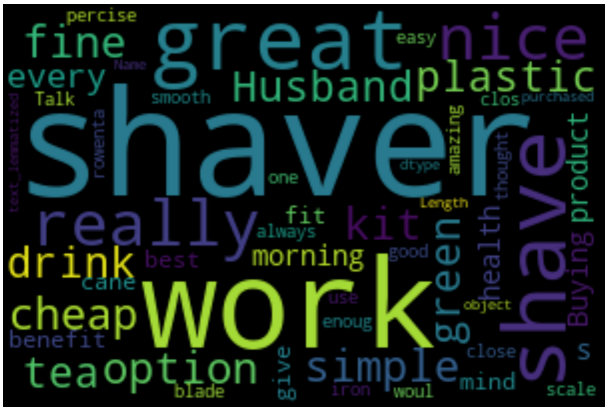

Figure 10: word cloud for positive words

- Create word cloud for negative words. As shown in Figure 11

Figure 11: Word cloud for negative words

- As shown in Figure 12 Here I have calculated the number of positive sentiments but have rating low as 1. which contradicts the start rating and sentiments.
- The same was checked for negative sentiments with rating of 5

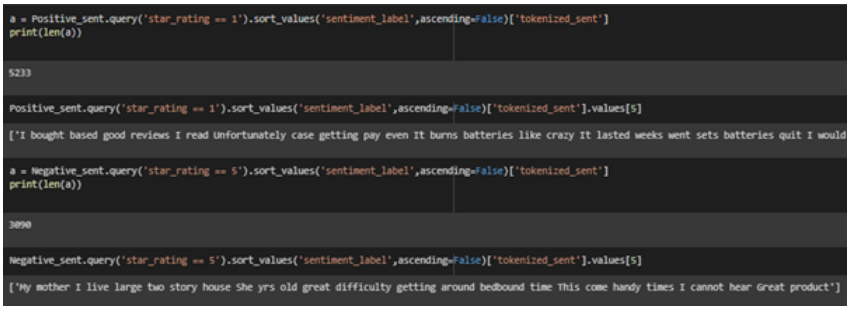

Figure 12: EDA for ratinga nd sentiments

Therefore we have not taken star ratings in consideration of creating model.

5 Model Training and Testing

As shown in figure the X was transformed into numerical using the TFIDF vectorizer as shown in Figure 13 Y contains sentiments created from SIA VADER Y contains sentiments (emotion column) created from NCRLEX

```
v = TfidfVectorizer()
X = v.fit_transform(df['review_body'])
y = df["sentiment_label"]
Y = df['Emotion']
```

Figure 13: TFIDF vectorization

1. SVM
   The SVM model was applied on the personal care appliances and predicted accuracy of 87.40% as shown in below Figure 14. The models are same for both Vader and NRLEX just the Y value differs

```
from sklearn import svm
clf = svm.SVC(gamma=0.001, C=100.)
svm_model= clf.fit(X_train, y_train)

pred = svm_model.predict(X_test)

from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,pred)*100)

87.40241542670314
```

Figure 14: SVM model

The evaluation methods shown in Figure 15 - Confusion matrix and Figure 16 Classification report

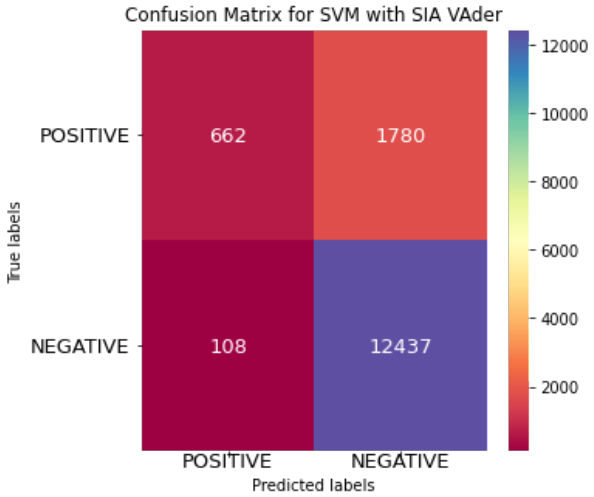

Figure 15: Confusion matrix

```
print(classification_report(y_test, pred))
```

	precision	recall	f1-score	support
neg	0.86	0.27	0.41	2442
pos	0.87	0.99	0.93	12545
accuracy			0.87	14987
macro avg	0.87	0.63	0.67	14987
weighted avg	0.87	0.87	0.85	14987

Figure 16: Classification report

2. Random forest
   The random forest model as shown in Figure 17. The models are same for both Vader and NRLEX just the Y value differs.

```
rf_model = RandomForestClassifier().fit(X_train, y_train)

pred = rf_model.predict(X_test)

from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,pred)*100)

94.13256137645314
```

Figure 17: Random Forest

The evaluation method :- confusion matrix (Figure 18) and classification report (Figure 19)

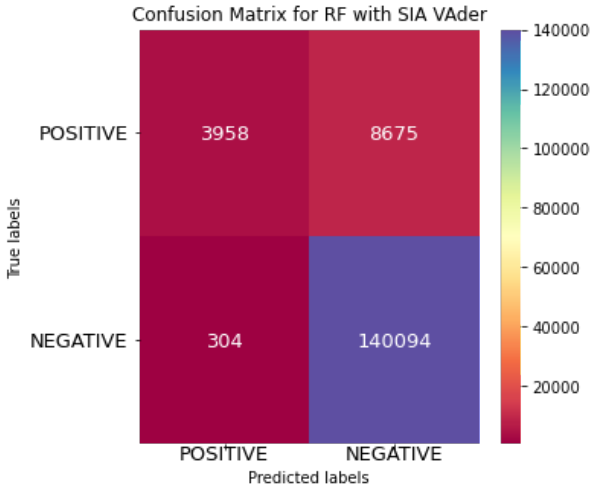

Figure 18: Confusion matrix

```
print(classification_report(y_test, pred))
```

	precision	recall	f1-score	support
neg	0.93	0.31	0.47	12633
pos	0.94	1.00	0.97	140398
accuracy			0.94	153031
macro avg	0.94	0.66	0.72	153031
weighted avg	0.94	0.94	0.93	153031

Figure 19: Classification report

3. Bi-direction LSTM
   Before applying Bi-LSTM the data needs to be transformed to input, therefore padding was applied as shown in figure 20. The models are same for both Vader and NRLEX just the Y value differs.

```
#padding

from  tensorflow.keras.preprocessing import sequence
max_review_lenght = 1500

X_train_pad = sequence.pad_sequences(X_train['text_tok'], maxlen = max_review_lenght, padding
X_test_pad = sequence.pad_sequences(X_test['text_tok'], maxlen = max_review_lenght, padding =
```

Figure 20: Adding padding to the input

Creating input weights using GloVe as shown in Figure 21

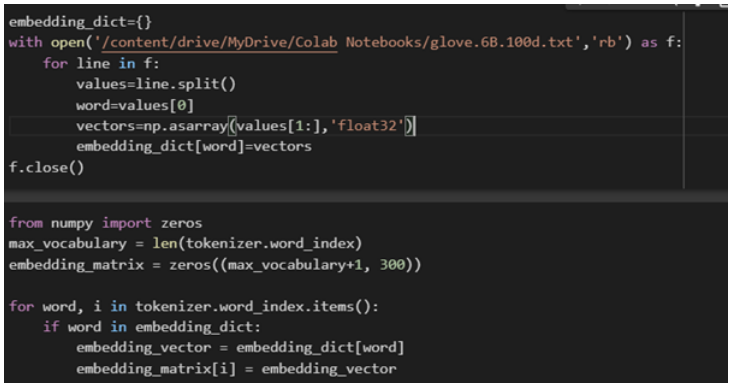

Figure 21: creating embedding matrix to the input weights

Saving the padded values into X_train and X_test for further processing as shown in Figure 22

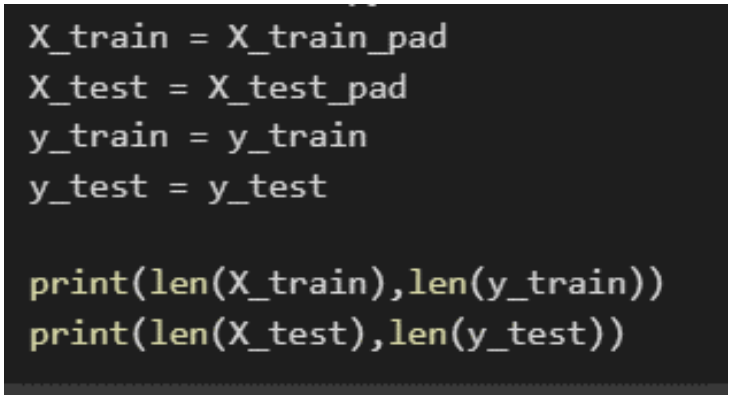

Figure 22: Saving padded values into Xtrain and Xtest

The Bi-LSTM model as shown in Figure 23

Figure 23: Bi-LSTM Model

The Evaluation method used :- Accuracy graph (Figure 24) and loss graph (Figure 25)

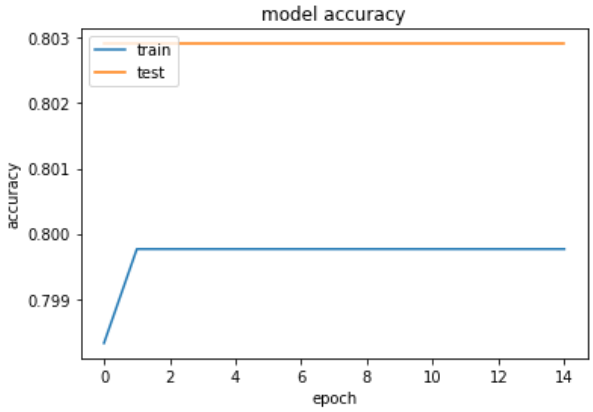

Figure 24: Accuracy graph

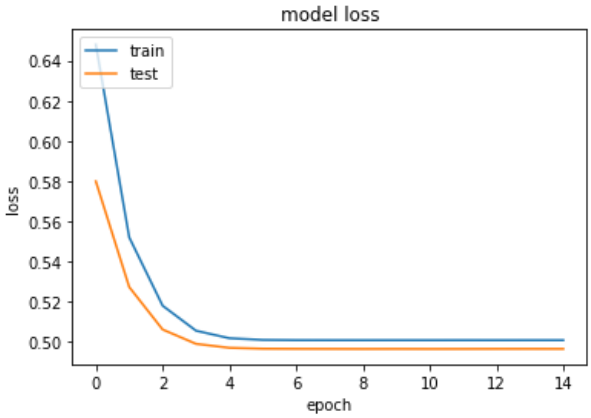

Figure 25: Loss graph