National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Nishant Bharti
Student ID: x21148686

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Nishant Bharti
Student ID:	x21148686
Programme:	Data Analytics
Year:	2018
Module:	MSc Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	20/12/2018
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	8

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:	30th January 2023

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

Nishant Bharti
x21148686

1 Introduction

This configuration manual discusses the step-by-step process that was involved in this project, from setting up the environment to implementing and evaluating it. The goal of this evaluation was to determine whether or not the performance of a transformer-based model could be improved by applying an aspect-based sentiment analysis technique to it. In this setup manual, you will find information on the programming language that is used, as well as the configuration of the system and any required libraries.

Discussion centers on the findings of this study as well as the several experiments conducted and the assessment criteria used for each of them.

2 Environment Setup

2.1 System Specification

Jupyter notebook and Google Collaboratory were used in order to carry out the implementation of this study. It is a web-based platform that is free of charge and is based on the Jupyter notebook. It provides resources for running Python applications on Google servers and gives users free access to high-end GPUs for the purpose of implementing machine learning models. Because of the faster GPU, the amount of time spent waiting while the code is executing is significantly reduced.

2.2 System Specification

Python language was used in this project, with all the packages and libraries listed below.

- NumPy
- Matplotlib
- Pandas
- NLTK
- Keras
- pytorch
- tensorflow
- Seaborn
- Scikit-Learn

2.3 Data Source

The data in this project is captured from website Kaggle. Four data set have been used in this project. The demonetization tweets data contains 14941 tweets captured in the year 2016 and contains the information across 16 columns holding data of tweets, source, id etc. Other three dataset contains data pf digital wallets that are googlepay,phonepay and paytm. Data was captured from year 2019-2021.

3 Implementation

In this section, all the steps are noted right from start to end implementation proposed in this project. Libraries loading, data preparation and implementing models.

3.1 Importing Libraries

In figure 1. all the necessary libraries are loaded used for implementing this project.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O

#Importing Libraries
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from textblob import Word
from sklearn.preprocessing import LabelEncoder
from collections import Counter


from keras.preprocessing.text import Tokenizer
import matplotlib.pyplot as plt
import re
from keras.models import Sequential
from keras_preprocessing.sequence import pad_sequences
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
from tensorflow.keras.optimizers import Adam

```

Figure 1: Importing Libraries

3.2 Data Loading and Pre-processing

In figure 2. and 3 data loading process is done for both the topics: sentiment analysis and lstm model. Data for EDA and Depp learning are same dataset.

Data Pre-processing was carried on to remove unwanted columns, stop words, removing outliers, null values etc. Below snippets show some of the steps carried out to in data

```
data1 = pd.read_csv("C:\\Users\\Nishant Bharti\\Desktop\\thesis\\demonetization-tweets.csv", encoding = 'unicode_escape')
data1.head()
```

	Unnamed: 0	X	text	favorited	favoriteCount	replyToSN	created	truncated	replyToSID	id	replyToUID	statusSource
0	1	1	RT @rssurjewala: Critical question: Was PayTM ...	False	0	NaN	2016-11-23 18:40:30	False	NaN	8.014957e+17	NaN	<a href="http://twitter.com /download/android" ...
1	2	2	RT @Hemant_80: Did you vote on #Demonetization...	False	0	NaN	2016-11-23 18:40:29	False	NaN	8.014957e+17	NaN	<a href="http://twitter.com /download/android" ...
2	3	3	RT @roshankar: Former FinSec, RBI Dy Governor,...	False	0	NaN	2016-11-23 18:40:03	False	NaN	8.014955e+17	NaN	<a href="http://twitter.com /download/android" ...
3	4	4	RT @ANI_news: Gurugram (Haryana): Post office ...	False	0	NaN	2016-11-23 18:39:59	False	NaN	8.014955e+17	NaN	<a href="http://twitter.com /download/android" ...
4	5	5	RT @satishacharya: Reddy Wedding! @mail_today ...	False	0	NaN	2016-11-23 18:39:39	False	NaN	8.014954e+17	NaN	<a href="http://cpimharyana.com" rel="nofollow...

Figure 2: Loading data for sentiment analysis

```
#loading data
df_1 = pd.read_csv("/content/GooglePayIndia.csv")
df_2 = pd.read_csv("/content/PaytmIndia.csv")
df_3 = pd.read_csv("/content/PhonePayIndia.csv")
```

Figure 3: loading data for EDA and LSTM model

pre-processing.

```
plt.figure(figsize=(10,4))
plt.xlim(-100, 3000)
flierprops = dict(marker='o', markerfacecolor='purple', markersize=6,
                  linestyle='none', markeredgecolor='black')
plt.title("Checking Outliers in Google thumbsUpCount Column")
sns.boxplot(x=google_data.thumbsUpCount, flierprops=flierprops)

plt.figure(figsize=(10,4))
plt.title("Checking Outliers in Paytm thumbsUpCount Column")
plt.xlim(paytm_data.thumbsUpCount.min(), paytm_data.thumbsUpCount.max()*1.1)
sns.boxplot(x=paytm_data.thumbsUpCount, flierprops=flierprops)

plt.figure(figsize=(10,4))
plt.title("Checking Outliers in PhonePay thumbsUpCount Column")
plt.xlim(Phonepe_data.thumbsUpCount.min(), Phonepe_data.thumbsUpCount.max()*1.1)
sns.boxplot(x=Phonepe_data.thumbsUpCount, flierprops=flierprops)
```

```
<AxesSubplot:title={'center':'Checking Outliers in PhonePay thumbsUpCount Column'}, xlabel='thumbsUpCount'>
```

Figure 4: Detecting Outliers

Before and after cleaning dataset, we obtain 5 rows and 22 columns, as shown in below figure 5.

```python
import re, sys
def clean_tweet(tweet):
    tweet = re.sub('http\S+\s*', '', tweet)  # remove URLs
    tweet = re.sub('RT|cc', '', tweet)  # remove RT and cc
    tweet = re.sub('#\S+', '', tweet)  # remove hashtags
    tweet = re.sub('@\S+', '', tweet)  # remove mentions
    tweet = re.sub('[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"""), '', tweet)  # remove punctuations
    tweet = re.sub('\s+', ' ', tweet)  # remove extra whitespace
    tokens = tweet.split(" ")
    tweet = [ x for x in tokens if len(x) < 25 ]
    tweet = " ".join(tweet)
    return tweet
```

Figure 5: Cleaning tweets dataset

	index	Unnamed: 0	X	text	favorited	favoriteCount	replyToSN	created	truncated	replyToSID	...	statusSource	screenName	r
0	0	1	1	RT @rssurjewala: Critical question: Was PayTM ...	False	0	NaN	2016-11-23 18:40:30	False	NaN	...	<a href="http://twitter.com /download/android" ...	HASHTAGFARZIWAL	
1	7	8	8	RT @Joydeep_911: Calling all Nationalists to j...	False	0	NaN	2016-11-23 18:38:20	False	NaN	...	<a href="http://twitter.com /download/android" ...	KARUNASHANKEROJ	
2	8	9	9	RT @sumitbhati2002: Many opposition leaders ar...	False	0	NaN	2016-11-23 18:38:09	False	NaN	...	<a href="http://twitter.com /download/android" ...	sumitbhati2002	
3	10	11	11	Many opposition leaders are with @narendramodi...	False	1	NaN	2016-11-23 18:37:47	False	NaN	...	<a href="http://twitter.com /download/android" ...	sumitbhati2002	
4	11	12	12	RT @Joydas: Question in Narendra Modi App wher...	False	0	NaN	2016-11-23 18:37:25	False	NaN	...	<a href="http://twitter.com /download/android" ...	MonishGavand	

5 rows × 22 columns

Figure 6: Cleaned dataset

3.3 Feature Extraction

While performing EDA on digital payments data, feature extraction task were carried out s shown in below figure 7 and data after feature extraction in figure 8,

In figure 7, all three data of digital payment, feature extraction are applied together and prepared a cleaned data.

3.4 Modelling and Evaluation

The process of data modeling is an act of training machine learning model to predict the values from the features and adjusting it according to the business needs. Deep learning method are going to be used such as LSTM and Mutlinomial Naive Bayes. The model measures accuracy of 95% accuracy and on similar line, T Srinivas et al. (2019) discussed similar model with the satisfactory results.

In figure 9, data is split into data training set and testing set and multinomial Naive Bayes theorem is applied.

```
cat_category = [feature for feature in google_data.columns if google_data[feature].dtypes == "O"]
google_data[cat_category].isnull().sum()
```

```
reviewId                   0
userName                   1
userImage                  0
content                    5
reviewCreatedVersion    4241
at                         0
replyContent           28106
repliedAt              28106
dtype: int64
```

```
google_data = google_data.drop(columns=["reviewCreatedVersion", "repliedAt"])
google_data["replyContent"] = google_data["replyContent"].fillna("No_reply/No_data")
```

```
cat_category = [feature for feature in paytm_data.columns if paytm_data[feature].dtypes == "O"]
paytm_data[cat_category].isnull().sum()
```

```
reviewId                   0
userName                   0
userImage                  0
content                    2
reviewCreatedVersion    23665
at                         0
replyContent           69777
repliedAt              69777
dtype: int64
```

```
paytm_data = paytm_data.drop(columns=["reviewCreatedVersion","repliedAt"])
paytm_data["replyContent"] = paytm_data["replyContent"].fillna("No_reply/No_data")
```

```
cat_category = [feature for feature in Phonepe_data.columns if Phonepe_data[feature].dtypes == "O"]
Phonepe_data[cat_category].isnull().sum()
```

Figure 7: Feature Extraction

```
google_data = google_data.sample(frac=1).reset_index(drop=True)
paytm_data = paytm_data.sample(frac=1).reset_index(drop=True)
Phonepe_data = Phonepe_data.sample(frac=1).reset_index(drop=True)

data = Phonepe_data.append([paytm_data[:11735 ], google_data[:11735]], ignore_index=True)

data = data.rename(columns={"at": "review_created_at"})
data.head()
```

	Unnamed: 0	reviewId	userName	userImage	content	score	thumbsUpCount	review_created_at	r
0	10443	gp:AOqpTOHJeSKMH2iQ7M36qxCCeq-f3egXa-K65lA-7mx...	Aaditri Gupta	https://play-lh.googleusercontent.com/a-/AOh14...	Very useful	5	0	2021-11-11 11:36:07	No_r
1	9813	gp:AOqpTOEzVsnW5UCy6_gQLH7j8vidVltUGA0AifqsPCe...	Universe Knowledge	https://play-lh.googleusercontent.com/a-/AOh14...	Nice app	4	0	2021-11-11 18:43:06	No_r
2	9461	gp:AOqpTOHgeUbJMZjBS5HYYMZVlH9swB3lQxmBJhnJZ7H...	Salma Sallu	https://play-lh.googleusercontent.com/a/AATXAJ...	Hhh	5	0	2021-11-11 22:42:32	No_r
3	7804	gp:AOqpTOGnKlDVUZ6Pwl_BQr1uhEH4tNa47uPgXjuQuuC...	Tapan Kumar Sahu	https://play-lh.googleusercontent.com/a-/AOh14...	Auto pay failed transfer but mainas Bank account	1	0	2021-11-13 07:51:55	ir
4	7142	gp:AOqpTOFDXblvWDaLfBi2CQfHuNlvZ8NxeQw80uOqzRp...	sushmitha thigala	https://play-lh.googleusercontent.com/a-/AOh14...	Nice app	5	0	2021-11-13 15:52:54	No_r

Figure 8: Data After Feature Extraction

```
print ("Multinomial Naive Bayes")
mnb_model = MultinomialNB()
print (mnb_model)
y = y_train.astype('int')
t = y_test.astype('int')
mnb_model.fit(X_train_vectorized, y)
test_predictedValues = mnb_model.predict(X_test_vectorized)
train_predictedValues = mnb_model.predict(X_train_vectorized)

trainingAccuracyScore = mnb_model.score(X_train_vectorized, y) * 100
testingAccuracyScore = mnb_model.score(X_test_vectorized, t) * 100
print ("\nTraining Accuracy Score - %.2f " % (trainingAccuracyScore), "%")
print ("Testing Accuracy Score - %.2f " % (testingAccuracyScore), "%")
print ("\n Following is the Classification Report for Multinomial Naive Bayes - ")
print (classification_report(t, test_predictedValues))

# finding the true positive rates and false positive rates using the roc_curve function
test_fpr, test_tpr, test_thresholds = roc_curve(t, test_predictedValues)
test_reqAUCScore = "%.2f " % (auc(test_fpr, test_tpr) * 100)
print ("\nTesting AUC Score - " , test_reqAUCScore, "%")

train_fpr, train_tpr, train_thresholds = roc_curve(y, train_predictedValues)
train_reqAUCScore = "%.2f " % (auc(train_fpr, train_tpr) * 100)
print ("Training  AUC Score - " , train_reqAUCScore, "%")

plt.plot(test_fpr, test_tpr, color='green', label = "Testing ROC Curve")
plt.plot(train_fpr, train_tpr, color='red', label = "Training ROC Curve")
plt.plot([0, 1], [0, 1], color='blue', linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic for Multinomial Naive Bayes model')
plt.legend(loc="lower right")
plt.show()
```

Figure 9: Split and run Multinomial Naive Bayes Model

The below figure 10 and 11 , shows the evaluation results of the trained Multinomial Naive Bayes.

Second phase of modelling was to train LSTM model and capture evaluation and results of the same. Figure 12 and Figure 13 shows the data split, training and accuracy of the model. The model reports with 82% accuracy, whereas similar model was applied in Shobana and Murali (2021) which had slightly better results.

References

Shobana, J. and Murali, M. (2021). Adaptive particle swarm optimization algorithm based long short-term memory networks for sentiment analysis, *Journal of Intelligent & Fuzzy Systems* **40**(6): 10703–10719.

T Srinivas, A. S., Govinda, K., Ramasubbareddy, S. and Swetha, E. (2019). Sentimental analysis of demonetization over twitter data using machine learning, *Journal of Computational and Theoretical Nanoscience* **16**(5-6): 2055–2058.

```
Multinomial Naive Bayes
MultinomialNB()

Training Accuracy Score - 95.87  %
Testing Accuracy Score - 92.05  %

 Following is the Classification Report for Multinomial Naive Bayes -
              precision    recall  f1-score   support

           0       0.93      0.94      0.93      1895
           1       0.91      0.88      0.90      1238

    accuracy                           0.92      3133
   macro avg       0.92      0.91      0.92      3133
weighted avg       0.92      0.92      0.92      3133


Testing AUC Score -  91.41  %
Training  AUC Score -  95.63  %
```

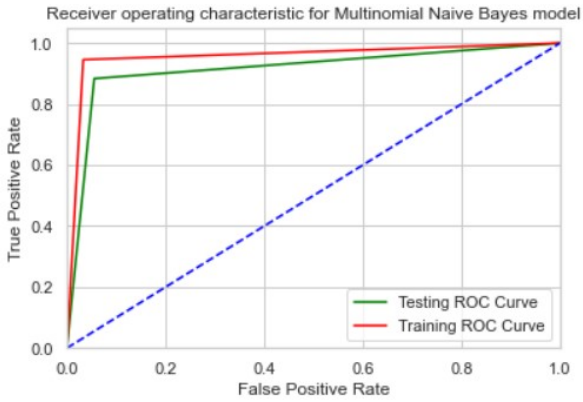

Figure 10: Results of Multinomial Naive Bayes

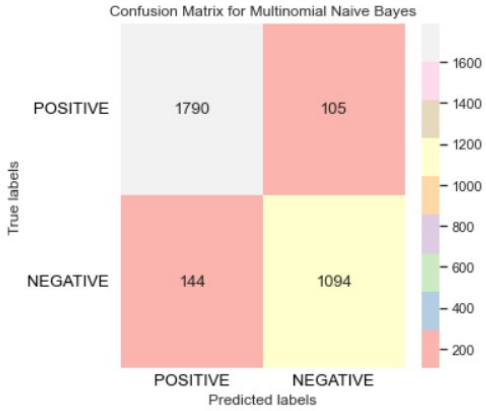

Figure 11: Confusion Matrix of Multinomial Naive Bayes

```
data_model = data[['content','score']]
token = Tokenizer(num_words=5000,split=' ')
token.fit_on_texts(data_model['content'].values)
X=token.texts_to_sequences(data_model['content'].values)
X = pad_sequences(X)
Y = pd.get_dummies(data_model['score'])
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.3,random_state = 1)
```

```
modelf = Sequential()
modelf.add(Embedding(5000, 240, input_length = X.shape[1]))
modelf.add(SpatialDropout1D(0.2))
modelf.add(LSTM(176, dropout=0.2, recurrent_dropout=0.2))
modelf.add(Dense(5,activation='softmax'))
```

```
print(modelf.summary())
```

Figure 12: Split and train LSTM Model

```
modelf.compile(loss = 'binary_crossentropy', optimizer= Adam(lr=0.01), metrics = ['accuracy'])
modelf.fit(x_train,y_train, epochs = 10, batch_size=1024, verbose = 'auto')
modelf.evaluate(x_test,y_test),
```

```
Epoch 1/10
47/47 [==============================] - 198s 4s/step - loss: 0.3345 - accuracy: 0.6774
Epoch 2/10
47/47 [==============================] - 190s 4s/step - loss: 0.2644 - accuracy: 0.7585
Epoch 3/10
47/47 [==============================] - 197s 4s/step - loss: 0.2481 - accuracy: 0.7715
Epoch 4/10
47/47 [==============================] - 197s 4s/step - loss: 0.2377 - accuracy: 0.7803
Epoch 5/10
47/47 [==============================] - 195s 4s/step - loss: 0.2284 - accuracy: 0.7892
Epoch 6/10
47/47 [==============================] - 204s 4s/step - loss: 0.2209 - accuracy: 0.7979
Epoch 7/10
47/47 [==============================] - 201s 4s/step - loss: 0.2141 - accuracy: 0.8039
Epoch 8/10
47/47 [==============================] - 205s 4s/step - loss: 0.2090 - accuracy: 0.8090
Epoch 9/10
47/47 [==============================] - 206s 4s/step - loss: 0.2039 - accuracy: 0.8145
Epoch 10/10
47/47 [==============================] - 201s 4s/step - loss: 0.1988 - accuracy: 0.8200
634/634 [==============================] - 28s 43ms/step - loss: 0.3215 - accuracy: 0.7391

[0.321541965007782, 0.7390574812889099]
```

Figure 13: Results of LSTM model