National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Gandam Suresh Kailash
Student ID: 23177667

School of Computing
National College of Ireland

Supervisor: Dr Anu Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Gandam Suresh Kailash	
Student ID:	x23177667	
Programme:	Data Analytics	
Year:	2024	
Module:	MSc Research Project	
Supervisor:	Dr Anu Sahni	
Submission Due Date:	12/12/2024	
Project Title:	Configuration Manual	
Word Count:	1024	
Page Count:	10	

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:	Gandam Suresh Kailash
Date:	11th December 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

Gandam Suresh Kailash
x23177667

1. INTRODUCTION

This Research is about Opinion mining on newspaper headlines regarding the US elections using NLP,SVM and Deep learning. The Following steps contains the step by step process of the tools required for the research and how tools are used and installed. Finally provided with the execution of the code step by step process.

2. SYSTEM CONFIGURATION

2.1. HARDWARE SPECIFICATION

- **OPERATING SYSTEM :** MacOS(Ventura13.5.1)
- **PROCESSOR:** M1Processor with build in 10 CPU and GPU
- **HARD DRIVE:** SSD(256GB) • **RAM:**8GB

2.2. SOFTWARE SPECIFICATION

- PYTHON
- GOOGLE COLLAB

3. Installation and Environment Setup

Fig 1 - Python Version

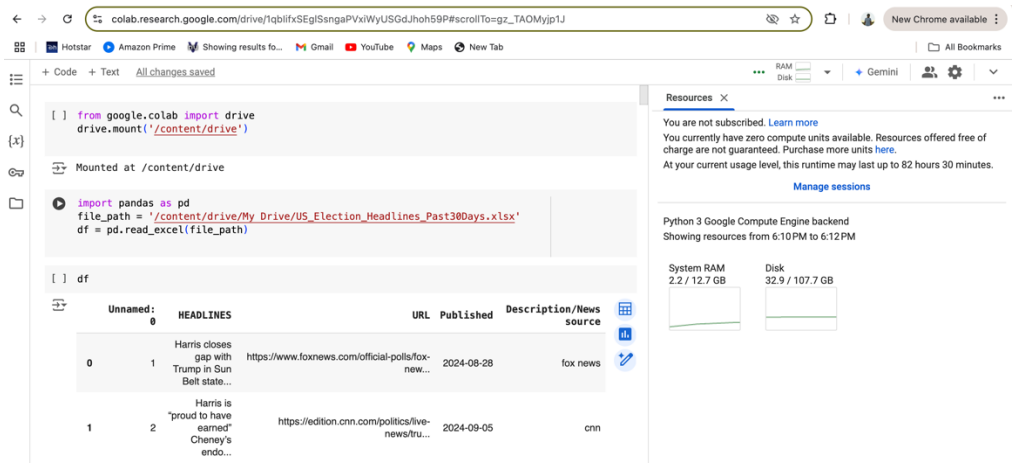

Fig 2 : GOOGLE Collab working Space

4. DATA COLLECTION

The dataset is collected from multiple mainstream news article sources which are U.S based media sources. The collected news articles are exactly started collecting 2 months before the U.S election polling dates which was on November 5. News articles headlines are collected from the various media such as CNN, Fox, Hindustan Times, BBC, ABC, Al Jazeera, The New York Times, Reuters, The Guardian, Forbes, and Washington News etc.

5. IMPLEMENTATION

5.1. Libraries Used in this research.

- **Numpy:** For numerical operations.
- **Pandas:** For data manipulation.
- **scikit-learn:** For implementing SVM and Random Forest models.
- **spaCy:** For named entity recognition (NER).
- **NLTK:** For basic text preprocessing.
- **transformers:** For BERT and RoBERTa implementation.
- **matplotlib:** For data visualization.
- **wandb:** For experiment tracking and visualization.

```
import gensim
from gensim.models import Word2Vec, KeyedVectors
import gensim.downloader as api
!pip install transformers datasets torch
from transformers import BertTokenizer, BertModel, BertForSequenceClassification
pip install torch
```

Fig 3.

```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('punkt_tab')
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
import re
import spacy
#spacy.cli.download("en_core_web_sm")
from sklearn.feature_extraction.text import TfidfVectorizer
!pip install vaderSentiment
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

```
#3 ml models
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from xgboost import XGBClassifier
from transformers import Trainer, TrainingArguments
from datasets import Dataset
from transformers import logging
logging.set_verbosity_error()
```

Loading...

```
#4 visualization
from matplotlib import pyplot as plt
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

Fig 4

Fig 3 & 4: Libraries required for research.

5.2 Dataset

This Dataset was initially collected in xlxs file.

- `US_Election_Headlines_Past30Days.xlsx - Before preprocessing`
- `headlines_for_manual_review_done.xlsx - After preprocessing`

5.3 The Flow of the Implementation

- Using the headlines available in the file name `US_Election_Headlines_Past30Days.xlsx` all the pre- processing are done and save in the cleaned_text column which we can see in the fig 5
- Named entity recognition has been performed in the fig 6.
- The inclusion of sentiment score which we can see in the fig 7.
- Categorizing the sentiment scores and adjusting them to put them into the correct criteria which they correctly fit into, which we can see in the fig 8.
- Visualization of the expression and Frames without the model fig 9
- Model Implementation
- Results

	Unnamed: 0	HEADLINES	URL	Published	Description/News source	cleaned_text
0	1	Harris closes gap with Trump in Sun Belt state...	https://www.foxnews.com/official-polls/fox-new...	2024-08-28	fox news	harris close gap trump sun belt statestrump le...
1	2	Harris is “proud to have earned” Cheney’s endo...	https://edition.cnn.com/politics/live-news/tru...	2024-09-05	cnn	harris proud earned cheneys endorsement campai...
2	3	Vance says it's "the best thing in the world" ...	https://edition.cnn.com/politics/live-news/tru...	2024-09-05	cnn	vance say best thing world cheney announced su...
3	4	Liz Cheney says she is voting for Harris for p...	https://edition.cnn.com/politics/live-news/tru...	2024-09-05	cnn	liz cheney say voting harris president
4	5	Trump suggests he could win 50% of Jewish vote...	https://www.foxnews.com/politics/trump-suggest...	2024-09-05	fox news	trump suggests could win 50 jewish vote presid...

Fig5: Data are loaded and videos are downloaded using Id’s

```
patterns = [
    {"label": "PERSON", "pattern": "Trump"},
    {"label": "PERSON", "pattern": "Donald Trump"},
    {"label": "PERSON", "pattern": "Trump's"},
    {"label": "PERSON", "pattern": "Kamala Harris"},
    {"label": "PERSON", "pattern": "Harris"},
    {"label": "PERSON", "pattern": "trump"}
    ]
ruler.add_patterns(patterns)

# Function to classify headlines based on NER
def classify_headline(headline):
    doc = nlp(headline)
    for ent in doc.ents:
        if ent.text.lower() in ['trump', 'donald trump', 'trumps']:
            return 'Trump'
        elif ent.text.lower() in ['kamala harris', 'harris']:
            return 'Kamala Harris'
    return 'Other'  # If no match
```

Fig 6: NER is performed with headlines

```
                                          cleaned_text  vader_sentiment
542  donald trump radically reshaped story america ...          0.0000
543                trump either white house big house          0.0000
544  tiny village india kamala harris ancestral roo...          0.3612
545           u gaza policy hurting harris black voter         -0.4019
546  raising rhetoric trump hit back kamala harris ...         -0.5574
```

Fig 7: Addition of sentiment score

```python
def sentiment_to_label(score):
    if score <= -0.6:
        return 1  # Very Negative
    elif score <= -0.2:
        return 2  # Negative
    elif score <= 0.2:
        return 3  # Neutral
    elif score <= 0.6:
        return 4  # Positive
    else:
        return 5  # Very Positive

# Apply the mapping function to convert sentiment scores into labels
df['sentiment'] = df['vader_sentiment'].apply(sentiment_to_label)

# Show the first few rows
print(df[['cleaned_text', 'vader_sentiment', 'sentiment']].head())
```

Fig 8: Categorization of headlines using the sentiment scores

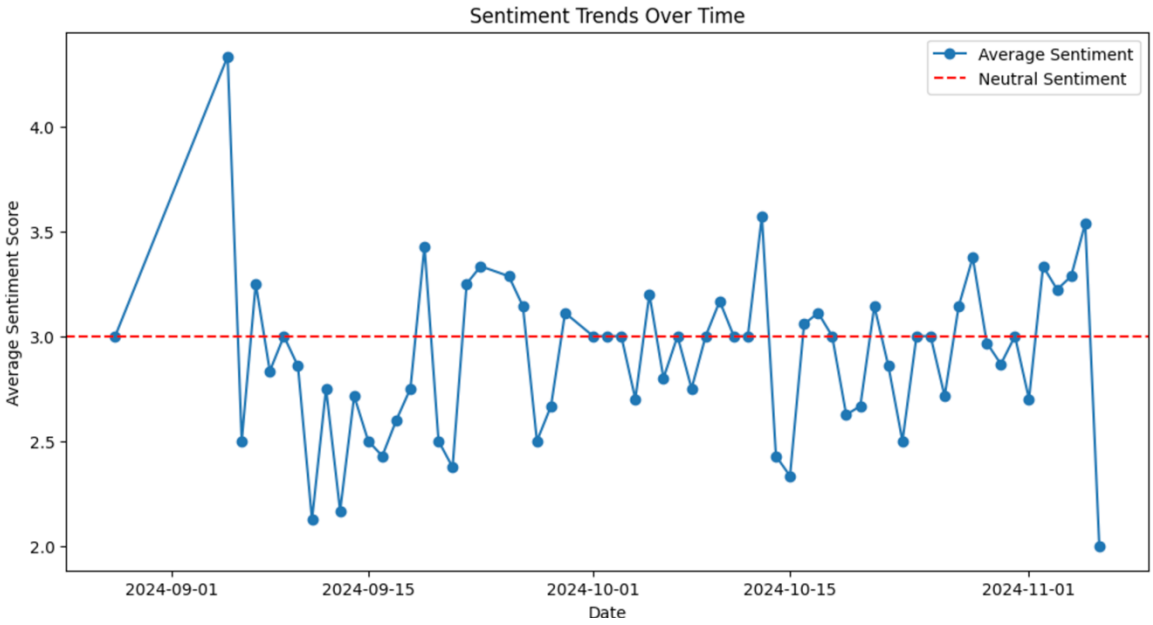

Fig 9: Average sentiment trends of US election candidates.

MODEL IMPLEMENTATION

There are three main three models used in this research

1. SVM + Word2Vec
2. BERT Transformer Model
3. RoBERTa Transformer

1. MODEL TRAINING SVM +Word2Vec

```
# Predict and evaluate
y_pred = svm_model.predict(X_test)
print("SVM with Word2Vec Classification Report:")
print(classification_report(y_test, y_pred))
```

```
SVM with Word2Vec Classification Report:
              precision    recall  f1-score   support

           1       0.36      0.71      0.48         7
           2       0.61      0.47      0.53        36
           3       0.61      0.61      0.61        38
           4       0.42      0.48      0.45        23
           5       0.75      0.50      0.60         6

    accuracy                           0.54       110
   macro avg       0.55      0.55      0.53       110
weighted avg       0.56      0.54      0.54       110
```

Fig 10 : SVM + Word2Vec Model Training

BERT TRANSFORMER MODEL:

```
<ipython-input-92-2d5e085ada47>:50: FutureWarning: `tokenizer` is deprecated and will be removed in version 5.0.0 for `Trainer.__init__`. Use `pr
  trainer = Trainer(
{'train_runtime': 2320.6139, 'train_samples_per_second': 0.565, 'train_steps_per_second': 0.036, 'train_loss': 1.3734676724388486, 'epoch': 3.0}
Classification Report:

               precision    recall  f1-score   support

Very Negative       0.00      0.00      0.00         7
     Negative       0.47      0.19      0.27        36
      Neutral       0.39      0.95      0.55        38
     Positive       1.00      0.13      0.23        23
Very Positive       0.00      0.00      0.00         6

     accuracy                           0.42       110
    macro avg       0.37      0.25      0.21       110
 weighted avg       0.50      0.42      0.33       110

Trainer Metrics:

{'test_loss': 1.323750615119934, 'test_runtime': 49.6049, 'test_samples_per_second': 2.218, 'test_steps_per_second': 0.141}
```

Fig 11 :BERT Transformer Model Training Results

RoBERTa Transformer:

[84/84 36:58, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1
1	1.623800	1.601348	0.345455	0.133508	0.345455	0.189816
2	1.519100	1.399246	0.345455	0.119339	0.345455	0.177396
3	1.426700	1.347226	0.354545	0.337638	0.354545	0.209867

Fig12 Roberta Model Training Results

The predicted value of the models is plotted in that comparing to all the models. The SVM in integration with Word2Vec has the better performance in terms of BERT and RoBERTa.
We experimented with multiple models such as :

- RBF Kernel in SVM represented in fig.13
- Polynomial kernel with SVM classification represented in fig.14
- Logistic Regression with Word2Vec is represented in fig.15
- Random forest regression model is show in fig 16
- At last Word cloud for the positive score is shown in fig 17.

RBF Kernel SVM Classification Report:

	precision	recall	f1-score	support
1	1.00	0.14	0.25	7
2	1.00	0.17	0.29	36
3	0.38	0.97	0.54	38
4	0.80	0.17	0.29	23
5	0.00	0.00	0.00	6
accuracy			0.44	110
macro avg	0.64	0.29	0.27	110
weighted avg	0.69	0.44	0.36	110

Fig 13 RBF Kernel with SVM Classification

Polynomial Kernel SVM Classification Report:

	precision	recall	f1-score	support
1	1.00	0.14	0.25	7
2	1.00	0.08	0.15	36
3	0.37	0.97	0.53	38
4	0.80	0.17	0.29	23
5	0.00	0.00	0.00	6
accuracy			0.41	110
macro avg	0.63	0.27	0.24	110
weighted avg	0.68	0.41	0.31	110

Fig 14 RBF Polynomial kernel with SVM Classification

Logistic Regression with Word2Vec Classification Report:

	precision	recall	f1-score	support
1	0.31	0.71	0.43	7
2	0.59	0.44	0.51	36
3	0.64	0.66	0.65	38
4	0.40	0.35	0.37	23
5	0.38	0.50	0.43	6
accuracy			0.52	110
macro avg	0.46	0.53	0.48	110
weighted avg	0.54	0.52	0.52	110

Fig.15 Logistic Regression with Word2Vec classification

Random Forest with Word2Vec Classification Report:

	precision	recall	f1-score	support
1	1.00	0.14	0.25	7
2	0.57	0.36	0.44	36
3	0.43	0.84	0.57	38
4	0.55	0.26	0.35	23
5	0.00	0.00	0.00	6
accuracy			0.47	110
macro avg	0.51	0.32	0.32	110
weighted avg	0.51	0.47	0.43	110

Fig.16 Random forest with Word2Vec model

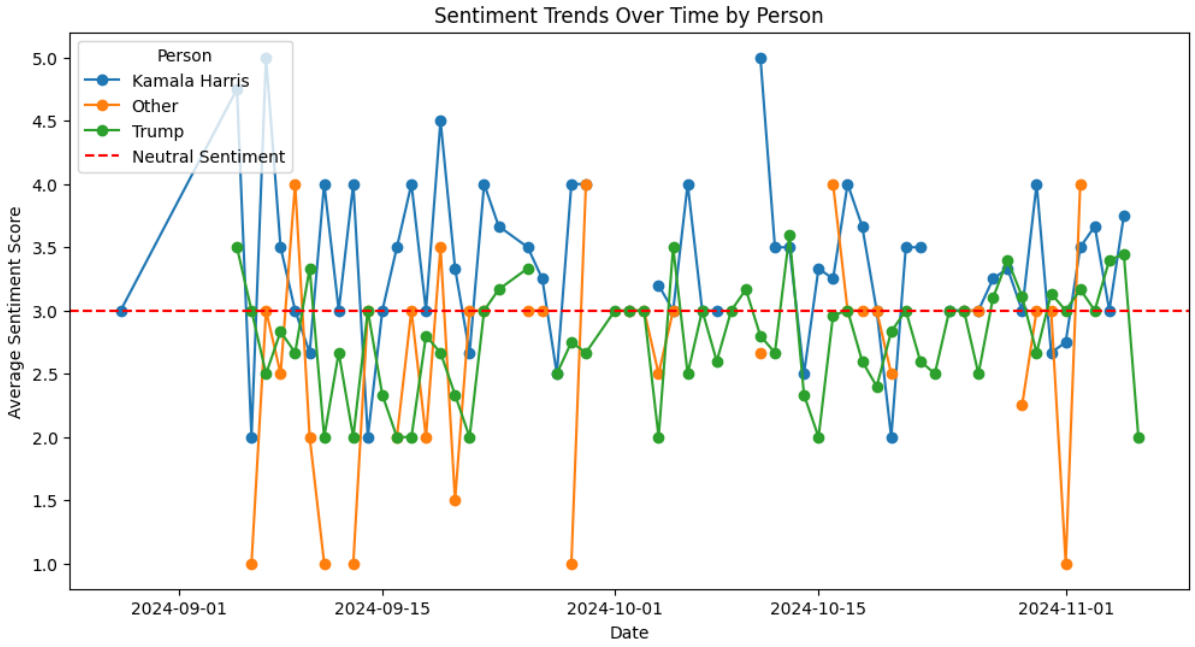

fig 17. Sentiment trend chart with the help of NER

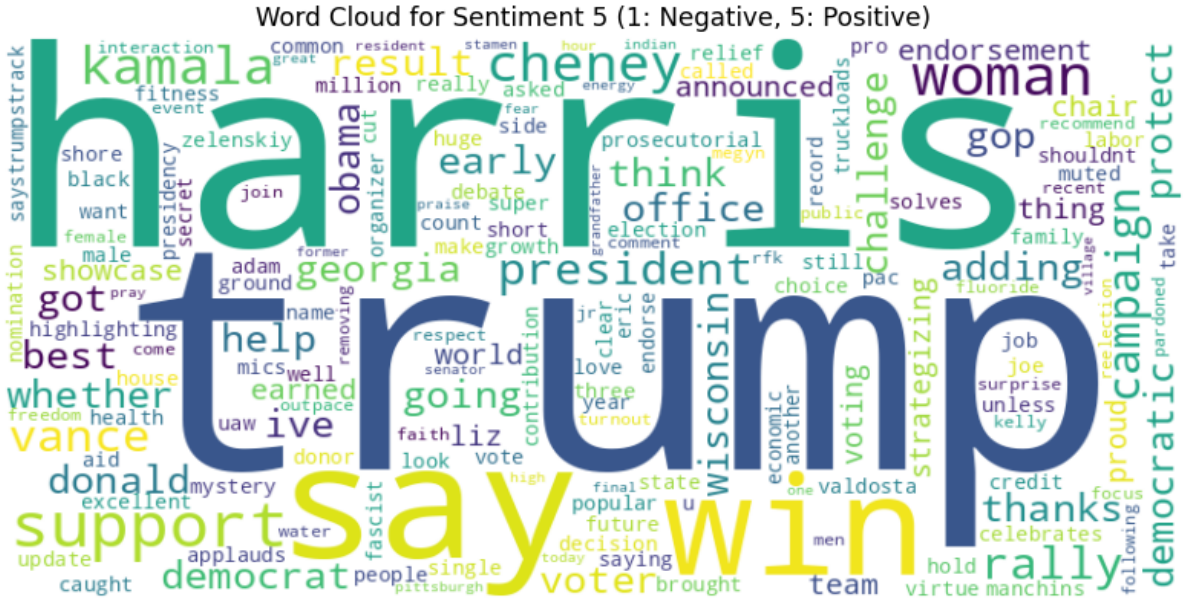

Fig 17 Word Cloud

6. Execution of the code

- As we are working with the google collab in order to retrieve the initial collected we need to mount the drive to the google collab.
- We are beginning with initial collected dataset artifact named as `US_Election_Headlines_Past30Days.xlsx` which at the we need to put inside the Data Frame for the further processing.
- Then we can proceed the data flow of the code with doing all the pre processing of the headlines in the google collab.
- After the pre-processing of the data, the pre-processed data are download for the manual analysis for checking the correctness of data whether it has fallen in the appropriate categorization of the sentiment and the correctness of the named entity recognition.
- After the manual work of correcting the headlines tagging to the appropriate person and adjusting the range of categorization criteria we once bring back dataset to the work with multiple models.
- The manipulated dataset is saved and produced in the name of headlines_for_manual_review_done.xlsx and mounted to the drive.
- The results of the multiple models are produced with precision, score, accuracy, recall, f1 score.