

Natural Language Processing  
**CSCE 5290, Fall 2022**

---



**UNT**<sup>TM</sup>  
**UNIVERSITY OF NORTH TEXAS**

Group 14:

- **Amough Mittal**
  - **Arun Sai Kumar Gutala**
  - **Neha Gummalla**
  - **Madi Arnold**
-

---

# **Project Proposal**

## **Project Title: Sentimental Analysis for Monarchy in UK**

### **1. Motivation**

Sentiment analysis is a study of subjective information present in any form of data which will give out different expressions used in any particular statement. Automating such a process gives out deep hidden insights on the data which is possible only with Natural Language Processing.

Our main goal is to find out sentiment analysis on how people reacted to the recent event 'Queen Elizabeth II death' and if possible find the most probable issues that might rise in the future.

### **2. Significance**

This project can be used for any particular event that occurs around the world and see how people are reacting to it. The expressions of people can be evaluated to see how the country/company is going to progress further in the future. The benefit of knowing things early can be taken as a precaution and avoid it totally which can turn over a catastrophe. This project can be contributed towards Artificial intelligence and further improve

### **3. Objectives**

The main objective of our project is to know about the expressions of people in the UK and around the world on how they feel about the Monarchy in the UK. The following project will help to get a score of expressions and also see how many people are unaffected by it. This score can be used by the officials to see what can be done to increase rating in any particular index. The project may also reveal the emotions of the people and lifestyle changes(in a positive or negative manner) they make after the Queen's death. The reign of the King can be predicted with the tweets of the people and how long it can go without breaking the Country / Government.

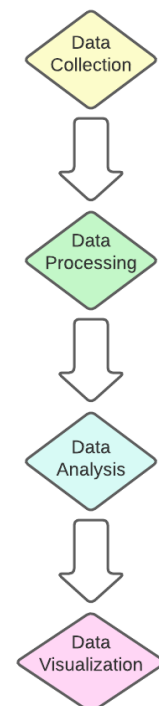
---

## 4. Features

- This project evaluates any kind of tweet dataset presented and gives a score based on the data.
- Visualization -- that includes charts generated from the data using Google Charts API.

### Content

```
Columns: ['id', 'conversation_id', 'created_at', 'date',  
'time', 'timezone', 'user_id', 'username', 'name', 'place',  
'tweet', 'language', 'mentions', 'urls', 'photos',  
'replies_count', 'retweets_count', 'likes_count', 'hashtags',  
'cashtags', 'link', 'retweet', 'quote_url', 'video',  
'thumbnail', 'near', 'geo', 'source', 'user_rt_id', 'user_rt',  
'retweet_id', 'reply_to', 'retweet_date', 'translate',  
'trans_src', 'trans_dest']
```



## 5. Related Work (Background)

### Sentiment Analysis

Sentimental analysis is a technique, which is used in determination of positivity or negativity of the data. It is used in real life segments like product reviews, spam email filtering, emotions and much more.

### Types of sentimental analysis:

1. Graded Sentiment Analysis: It categories the text into 5 levels like very negative, negative, neutral, positive or very positive.
2. Emotion Detection: It determines emotions like happiness, anger and frustration.

- 
3. Aspect-based: If there is a need to find the positivity or negativity in terms of opinion, then it comes under this category.
  4. Multilingual: Compared to other types, this is a complex model. In the case of multilingual sentences or idioms, it is hard to find meaning and perform analysis.

#### **Algorithms used:**

1. Linear Regression: It is a supervised learning used to perform regression tasks. It implements the model that finds a linear relationship between all the variables.
2. Naive Bayes: It is also a supervised learning which uses Bayes theorem for classification. Sentimental analysis is done considering the probability of the words.
3. Support Vector Machine: This algorithm is supervised learning that is implemented by finding a best fit line in a n-dimensional space to classify the data.

#### **Previous work:**

There are many projects that implement sentiment analysis like product reviews, movie reviews, paper reviews, customer feedback and much more.

There is a project that implements the sentimental analysis model on the twitter data.

Overview of the project: In this project is builded to classify the positivity or negativity of the tweets. The data is collected from the Twitter API and uses Naive Bayes classifier.

In this project firstly the preprocessing of the text is done by removing all the stop words, repeated words and punctuations, lower casing all the text and stemming. Next the train and test dataset are formed and the TF-IDF vectorizer is used in transforming the dataset. Finally the prediction is performed. The model is built using Naive Bayes and Support Vector Machine separately and accuracy of both models are compared.

---

## 6. Dataset

In this project we are using the tweets that have been generated on the Queen Elizabeth 2 death. Using this we are going to perform the sentimental analysis.

The dataset comprises of around 200000 tweets. The details of each tweets are recorded and are categorised into 36 columns. These columns gives the details such as the timestamp of the tweet, the name of the user posted, names of the user, like count, retweet count, source, user ID, translate, 'id', 'conversation\_id', 'created\_at', 'date', 'timezone', 'user\_id', 'username', 'place', 'tweet', 'language', 'mentions', 'urls', 'photos', 'replies\_count', 'hashtags', 'cashtags', 'link', 'retweet', 'quote\_url', 'video', 'thumbnail', 'near', 'geo', 'user\_rt\_id', 'user\_rt', 'retweet\_id', 'reply\_to', 'retweet\_date', 'trans\_src', and 'trans\_dest'. The size of the data is about 98.72 MB.

The dataset has a unique hashtags of 26857 and proportion of empty hashtags is 62.558%. It has a unique cashtags of 55 and the proportion of empty cashtags is 99.958%. It also has a unique Mentions Count of 7482 and the proportion of empty Mentions is 91.14999%. It has a unique Urls Count of 37070 and the proportion of empty Urls is 71.594%. It has a unique Photos count of 68241 and the proportion of empty Photos is 63.8839%. It also has a unique Reply\_to Count is 8764 and proportion of empty Reply\_to is 92.683%. It has 530 unique timezones. The maximum user-id value is 1568164808561950720. Maximum replies count of a tweet is 8102. Maximum retweets count is 16910. The maximum likes count is 151799. The total number of videos is 1 and the maximum uro count is 766 and the maximum photo count of 204.

Among the tweets 83.8% are from the en region. The regions where most of the tweets have been observed are - 4.5% from cy region. 1.7%, 1.6%, 1.4%, belongs to es, und, and qme regions.

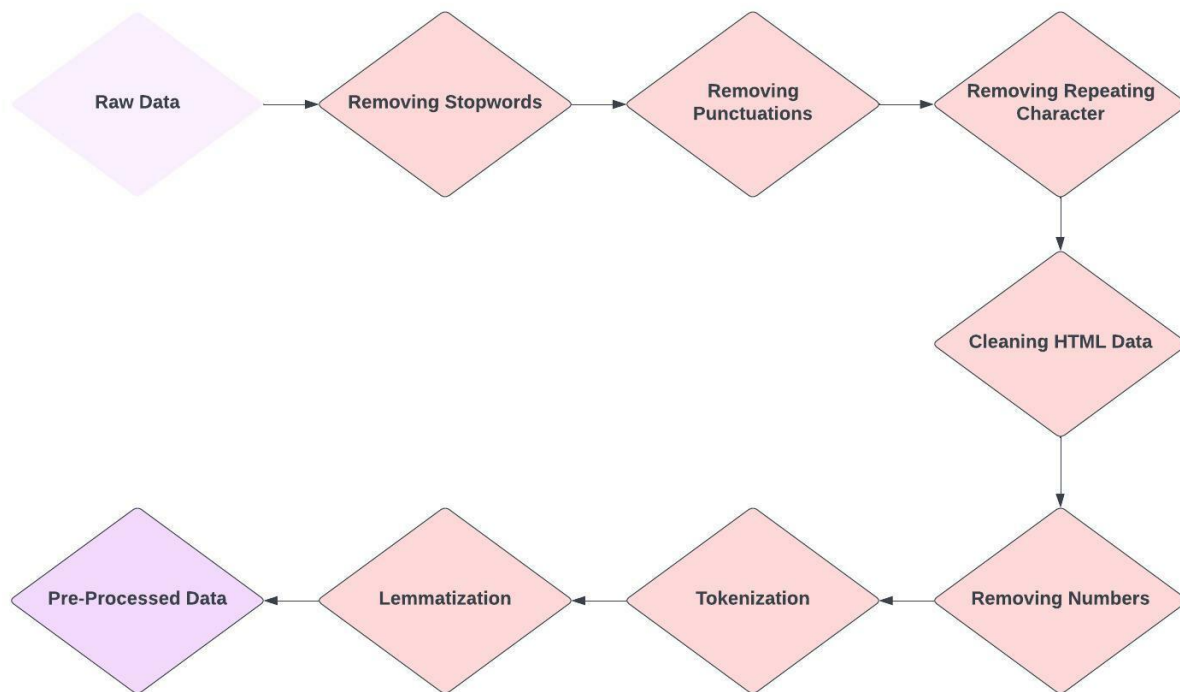
Index	128
date	1522600
user_id	1522600
username	1522600
name	1522600
tweet	1522600
language	1522600
mentions	1522600
replies_count	380650
retweets_count	380650
likes_count	761300
hashtags	1522600
retweet	190325
video	380650
reply_to	1522600
url_count	380650
photo_count	380650
dtype	int64

---

## 7. Detail design of Features

### A. Data pre-processing

- 1) Removal of Punctuations: Sentimental Analysis is sometimes difficult to implement because of the presence of non-useful data(referred as noise) with the useful data. The Non-useful data in our dataset includes Punctuations, emojis, stopwords, etc. We are removing the punctuations, emojis etc, using the Regular Expressions.
- 2) Removal of Stopwords: Stopwords are words that are present in a high volume in most of the documents and contribute very little to the meaning of the text. These words usually harm the accuracy of the model and it's better to remove them before training. Stopwords are generally filtered out for processing in any natural language processing context. Examples of Stopwords are 'the', 'are', etc.
- 3) Removing Repeating Characters: Characters which are appearing repeatedly are removed.
- 4) Cleaning HTML Data: HTML Data is Cleaned.
- 5) Removing Numbers: Numbers have been removed using Regular Expressions.
- 6) Tokenization: Tokenization is the process of breaking a stream of text data into words. This is usually the first step of the NLP Pipeline. After Tokenization we can perform other NLP Processes like Lemmatization.
- 7) Lemmatization: Lemmatization is the process to get the base form for all the tokens we got after Tokenization. We have used WordNetLemmatizer from the NLTK library.



**Fig:** Workflow diagram

## **B. Positive and Negative / True Positive and True Negative score**

Positive Score refers to the score we get that the tweet is positive(or sentimental/empathetic) towards the queen's death. Negative Score is the score we get that the tweet is negative(or showing anger towards the monarchy). True Positive Score is the score we get on the basis of the number of times the model correctly predicts the positive class. True Negative Score is the score we get on the basis of the number of times the model correctly predicts the negative class.

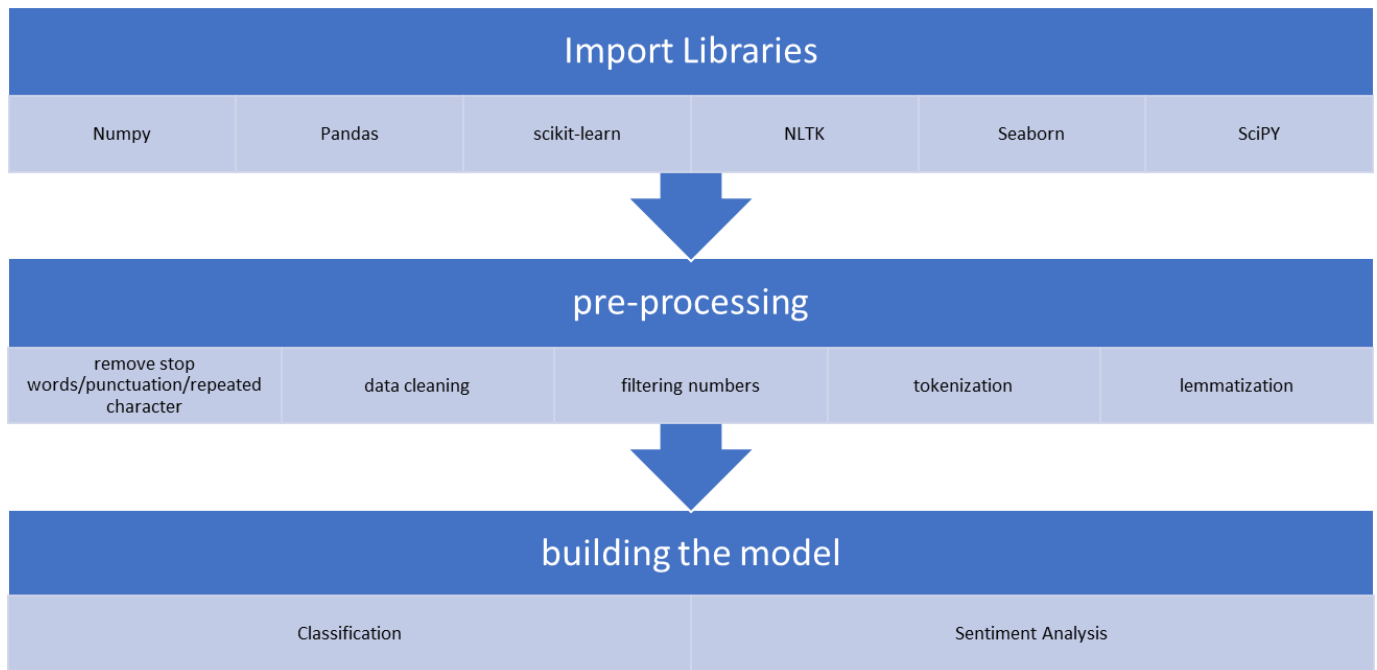
## **C. Classification of statement**

We ask users to enter a tweet(text input), we can classify if the tweet is positive or negative using our model.

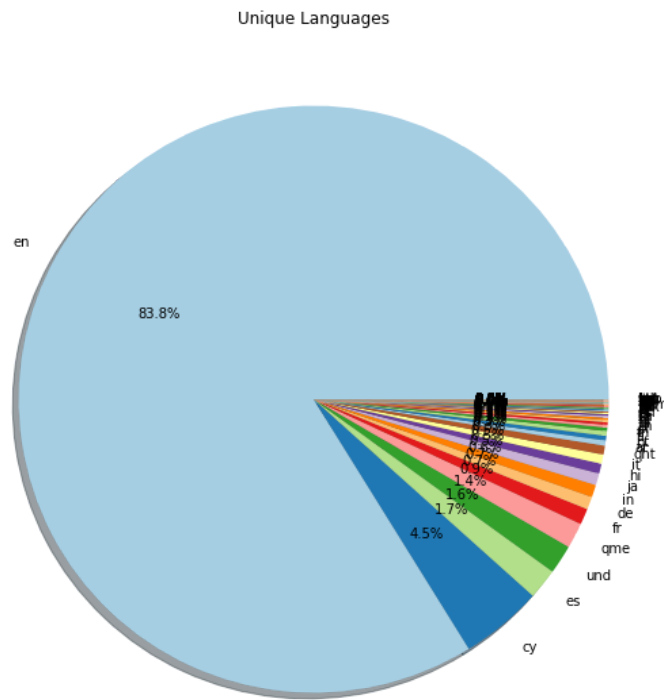
---

## 8. Analysis

Visualizing the data is important before building a model. There are many ways to visualize. Examples are a flow chart, a histogram, a word cloud, a pie chart, or more. These are all ways to perform exploratory data analysis on the dataset. We have shown a flowchart with the flowchart of the data model. We also have visualized the data by showing the different languages that are present in the tweets. This pie chart helps to see the amount of tweets per language in the dataset.







The above pie chart is displaying the portion of languages in the dataset.

## 9. Implementation / Work completed

Here is the implementation of our code:

---

```
[ ] #Importing numpy library
    import numpy as np
    #Importing pandas library
    import pandas as pd
    #Importing NLTK library
    from nltk.stem import WordNetLemmatizer
    from nltk.tokenize import RegexpTokenizer
    import string
    #Importing Regular Expressions for various operations
    import re
    #Importing Scipy
    import scipy.stats as stats
    #Importing pylab
    import pylab
    #Importing Matplot
    import matplotlib.pyplot as plt
    #Importing Seaborn
    import seaborn as sns
    from transformers import AutoTokenizer
```

```
[ ] nltk.download('wordnet')
    !python3 -m nltk.downloader wordnet
    !unzip /root/nltk_data/corpora/wordnet.zip -d /root/nltk_data/corpora/
```

**Step 1:** First the code starts by importing all the libraries and packages needed to perform the code. We are going to need a lot of libraries since we will be using different pre-processing models and plots for our code.

### Important functions

```
def stopwordsRemovalFn(text):  
    return " ".join([word for word in str(text).split() if word not in stop_words])  
  
def punctuationRemovalFn(text):  
    english_punctuations = string.punctuation  
    translator = str.maketrans('', '', english_punctuations)  
    return text.translate(translator)  
  
def repeatingCharacterRemovalFn(text):  
    return re.sub(r'(.+)+', r'1', text)  
  
def cleaningHTMLDataFn(data):  
    return re.sub('((www.[^s]+)|(https?://[^\s]+))', ' ', data)  
  
def removingNumbersFn(data):  
    return re.sub('[0-9]+', '', data)  
  
def lemmatizeDataFn(data):  
    text = [lm.lemmatize(word) for word in data]  
    return data
```

**Step 2:** The code is detailed and will need functions to perform properly. Each function will have a definition to define the properties of the function to tell the computer what to do.

```
queenTweetsDf = pd.read_csv("queen.csv")  
queenTweetsDf.info()
```

**Step 3:** This is the upload of the dataset that we are using. The dataset we are using is tweets about the queen.

```
plt.figure(figsize=(10,10))  
color_palette = sns.color_palette("Paired")  
plt.pie(queenTweetsDf.language.value_counts(), labels=queenTweetsDf.language.value_counts().index, colors=color_palette)  
plt.title("Unique Languages")  
plt.show()
```

**Step 4:** The code then plots the pie chart figure with the color palettes. The labels and titles are also given.

---

```

filteredColumns = ["tweet", "language"]
queenTweetsDf = pd.read_csv("queen.csv", usecols=filteredColumns)
#The content before filtering out the Tweets only containg English Language
print(queenTweetsDf)
print(queenTweetsDf[queenTweetsDf.language=='en'])
#The content after Filtering out the Tweets that only contain English Language
tweetInEn=queenTweetsDf[queenTweetsDf.language=='en']
print(tweetInEn)
data=tweetInEn.tweet
print(data)

```

---

**Step 5:** The code then reads the dataset and filters the tweets in the english language. It then outputs the results for the data in the english language.

```

#Printing the dictionary of stopwords downloaded from NLTK corpus
stop_words = set(stopwords.words('english'))
print(stop_words)

```

```

▶ data = data.apply(lambda text: stopwordsRemovalFn(text))
#Printing data after removing the stopwords
data.head()

```

**Step 6:** This is the stop words removal from the dataset. The code then reads the function and removes the stop words from the dataset.

```

▶ data=data.apply(lambda x: punctuationRemovalFn(x))
#Printing data after removing punctuation from the dataset
data.tail()

```

**Step 7:** This is the punctuation removal from the dataset. The code then reads the function and removes all punctuation from the dataset.

```

▶ data= data.apply(lambda x: repeatingCharacterRemovalFn(x))
#Printing data after removing repeating characters removal
data.tail()

```

---

---

**Step 8:** The code then removes all repeating characters from the dataset.

```
data= data.apply(lambda x: cleaningHTMLDataFn(x))  
#Printing data after removing URL or any HTML Data in the dataset  
data.tail()
```

**Step 9:** This is the cleaning of the dataset by removing all html and urls

```
[ ] data = data.apply(lambda x: removingNumbersFn(x))  
#Removing any numbers taht are in the dataset  
data.tail()
```

**Step 10:** This is the removal of all numbers from the dataset.

```
tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")  
#Printing tokens after tokenizing with the help of DistilBERT base model  
data = data.apply(tokenizer.tokenize)  
data.head()
```

**Step 11:** This is the tokenization of the dataset. It will lowercase all of the words in the dataset to help the code run smoothly.

```
[ ] data = data.apply(lambda x: lemmatizeDataFn(x))  
#Printing data after lemmatizing text with the help of lemmatize function from NL  
data.head()
```

**Step 12:** This is the lemmatization of the data. This will help group the words together without knowing their context for the sentiment analysis of the dataset.

---

## 10. Preliminary Results

Our preliminary results show that a large percentage of the data is going to be categorized as bad. The data is from tweets that have been tweeted after the recent passing of the Queen, so a large percentage of tweets will be categorized as bad. The tweets containing information about the death have a sad or shocked emotion that will be categorized into the bad category.

## 11. Project Management

### Implementation status report

#### • Responsibility/ Contributions (members/percentage)

Member Name	Contribution description	Overall Contribution
Amogh Mittal	Detail Design of Features, Objectives, Features, and Code(Dataset Implementation)	25%
Arun Sai Kumar Gutala	Dataset description & implementing code	25%
Neha Gummalla	Related work, project management	25%
Madi Arnold	Analysis, Implementation, and Preliminary Results	25%

### Work to be completed:

#### Description

In the Increment-2 we are planning to do the following:

- The machine learning models like Naive Bayes and SVM are going to be developed on the preprocessed data.
- Comparing the accuracies of both models.

---

• **Responsibility (Task, Person)**

Member Name	Contribution description	Overall Contribution
Amogh Mittal	Analysis, Details Design of Features, Code(Classification of a statement)	
Arun Sai Kumar Gutala	Implementation of machine learning model	
Neha Gummalla	Naive Bayes Algorithm, Accuracy	
Madi Arnold	Analysis, Implementation, and Results Analysis	

• **Issues/Concerns**

## 12. References

- Tweets after Queen Elizabeth II's Death  
[<https://www.kaggle.com/datasets/aneeshtickoo/tweets-after-queen-elizabeth-iis-death>]
- Towards Data Science [<https://towardsdatascience.com>]
- An Evolutionary-Based Sentiment Analysis Approach for Enhancing Government Decisions during COVID-19 Pandemic: The Case of Jordan  
[<https://www.mdpi.com/2076-3417/11/19/9080>]

Attachments:

1. GitHub Link -- [https://github.com/akgutala/NLP\\_Project](https://github.com/akgutala/NLP_Project)