Sentiment Classification of Current Public Opinion on BREXIT: Naive Bayes Classifier Model vs Python's TextBlob Approach

MSc Research Project
Data Analytics

Bhupender Singh Shekhawat
Student ID: X17170214

School of Computing
National College of Ireland

Supervisor: Prof. Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Bhupender Singh Shekhawat
Student ID:	X17170214
Programme:	MSc Data Analytics
Year:	2018/19
Module:	Research Project
Supervisor:	Prof. Vladimir Milosavljevic
Submission Due Date:	16th September 2019
Project Title:	Sentiment Classification of Current Public Opinion on BREXIT: Naive Bayes Classifier Model vs Pythons TextBlob Approach
Word Count:	8716
Page Count:	31

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:	16th September 2019

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):	

Sentiment Classification of Current Public Opinion on BREXIT: Naive Bayes Classifier Model vs Pythons TextBlob Approach

Bhupender Singh Shekhawat
X17170214

Abstract

Sentiment Analysis is playing a crucial role in technological world due to tremendous growth in the field of social media. The motivation regarding sentiment analysis comes from the fact that social media platforms like twitter provide a great platform which is used by general public to express their opinions about a product or an event. Such opinions provide an opportunity to researchers to work on data mining based on public reviews and opinion and provide critical insights helpful for organizations in better decision making. This paper discusses comparison in performances of Naive Bayes Classifier Model and Python's TextBlob library by carrying out sentiment classification on current public opinion on BREXIT based on different geographical locations. In order to achieve the objectives, natural language processing concepts including regular expression library and count vectorization have been used. Also, Natural Language Toolkit library along with TextBlob library are used to clean the data and provide polarity score to the tweets respectively. Naive Bayes Classification algorithm is then introduced into the model after training it on Sentiment140 Twitter dataset to provide an accuracy comparison to that of the TextBlob. Therefore, useful insights are produced considering visualizations obtained from Tableau. Moreover, the results of this research provide an insight about public opinions of different countries about BREXIT. Also, the results will help British and Irish governments to formulate their foreign policies and internal policies in order to maintain their relationship with their major business friendly countries.

1 Introduction

The exit of Britain from European Union is termed as BREXIT. However, since March 2019, the uncertainty over BREXIT deal has been pushing the BREXIT date further. Now, the United Kingdom has been provided an opportunity to work upon the BREXIT deal until 31st Oct 2019 before BREXIT happens. BBC (2019b)

Moreover, there have been a lot of speculations on the aftereffects of BREXIT deal. It is expected that such speculations along with many other political and social events associated with BREXIT happened in the recent past will play an important role on present public sentiments about BREXIT. Hence, this research aims at discovering the public sentiments using social media platform twitter and producing visualizations based on insights obtained from sentiment analysis of the retrieved dataset. Also, it is considered that BREXIT will certainly impact the United Kingdom's economy and the associated countries(business friendly countries) with it. Therefore, it becomes extremely important to determine the public reaction in the countries which plays an important part in positively contributing towards the United Kingdom's economy. Such analysis will provide critical insights to United Kingdom government for deciding their internal as well as foreign policies. Until now, only a few researches have been done on BREXIT and they have been mainly focussing on determining the public's opinion about it. Therefore, this research will focus on identifying the impact of BREXIT on countries playing an important part in the United Kingdom's economy. For it, data mining techniques Naive Bayes Classifier algorithm, NLTK library along with natural language processing concepts are used to carry out sentiment analysis. In order to fetch real time tweets, Twitter streaming API has been used, once sentiment analysis was carried out then insights related to the retrieved dataset were visualized using Tableau.

1.1 Motivation and Background

With surge in number of users of social media platforms like Twitter, it has now become very common for people to express their opinions about certain event, product or organization on such platforms. This has brought sentiment classification at the centre of interest for many researchers and scholars. Sentiment classification or sentiment analysis in text classification on social media platform like Twitter is defined as a process of finding out public opinion about an event, product or topic using techniques like machine learning. In it, public opinions are classified into categories like Positive, Negative and Neutral. Sentiment classification helps organizations to gain insightful knowledge from retrieved data for swift decisions on crucial moments. Sentiment Analysis on BREXIT has been an interesting topic for research. However, so far very few researches have been done on this topic and previous researches have mainly been focussing on identifying the opinion swing among public. As the BREXIT date (31st Oct 2019) is approaching closer, there are a lot of speculations and discussions going on about it on social media platforms like twitter. A lot of organizations are also interested to have an idea about BREXIT impacts globally, Osborn and Barry (2016) published article about possible impact of BREXIT on Information Technology Industry. Therefore, it is expected directly or indirectly BREXIT will have certain impact on businesses. This brings a need for sentiment analysis about BREXIT on social media platform like Twitter and in order to make an effort to gain insights from public opinions from residents of major business partner countries to United Kingdom. Therefore, conducting real time sentiment analysis on BREXIT for public based in major business partners countries of UK will provide an opportunity to the United Kingdom government to gain some useful insights in order to formulate their business and foreign policies in a better sense.

1.2 Research Question

Can sentiment classification on BREXIT using Naive Bayes Classifier Model and Pythons TextBlob approach assist/support United Kingdom and Irish government in gaining important insights from real time sentiment analysis of their major business partner countries.

In order to solve the research question, certain objectives have been formulated, implemented, evaluated and finally results are illustrated.

1.3 Objectives

- Literature search on sentiment classification based on Naive Bayes Classifier Model and Python's TextBlob approach.
- Implementation, Evaluation and Results on Naive Bayes Classifier Model.
- Implementation, Evaluation and Results on Python's TextBlob approach.
- Comaprison of Naive Bayes Classifier Model and Python's TextBlob approach .

2 Related Work

In order to discover the topic to perform research on sentiment analysis, the papers related to sentiment analysis were initially studied which further provided the basis for deciphering text classification and opinion mining that helped in forming the core foundation of text understanding about BREXIT using twitter as the topic for this research. Also, these papers have helped to explore Naive Bayes Classifier Algorithm, python's Textblob library, along with concepts from Natural Language Processing such as tokenization, Count Vectorization and Python's regular expression library as the machine learning technique to carry out this research. Moreover, special credit goes to the online research paper-based libraries such as IEEE explore and Research Gate for providing an abundance of literature resources. As far as sentiment analysis is concerned, there have been numerous researches being done in this field. Therefore, it becomes extremely important to throw some light on key previous work which have been done in this field and then proposing a new solution at the end of the discussion.

2.1 A Review on Sentiment Analysis on Twitter Data

As shown by Ramanathan and Meyyappan (2019), Sentiment analysis is playing a crucial role in the technological era because of presence of wide range of applications supporting business and platforms like social media. The advantage of sentiment analysis is, it helps in determining the present opinion of public about a product or field. The insights retrieved from such analysis on public opinion assists business to work upon improving their product quality. Domain specific ontology is a type of analysis done using common sense. The part of speech (POS) tagging turns out to be crucial in identifying the entities, this is further supported by comparing entities based on knowledge gained from domain specific ontology. Sentiment lexicon approach is a branch of sentiment analysis which is used for determining the sentiment scores of entities. Semantic orientation is combined belonging to respective domain specific features. Also, as one of the advantages it is found that using machine learning algorithms along with features as conceptual semantic improves overall performance of the model. However, as an improvement contextual and conceptual semantic sentiment analysis can be utilized to improve the performance of model. This paper has helped in understanding on how part of speech tagging could be used along with conceptual semantic sentiment analysis, which assist in improving overall performance of the selected machine learning algorithm. Although, automatic sentiment analysis is a very good way to analyze sentiments of public about a product or topic. But, at the same time another challenge related to sentiment scores comes up. As sentiment scores allocated to all words stay same irrespective of domain of the research. There is possibility for obtaining less accurate results because lexicon-based approach performs differently in different domains. Ikoro et al. (2018) resolved this problem and accuracy of the research is improved by using two sentiment lexicons altogether. As a part of it, initially a lexicon was used to obtain the words containing sentiments and negative words. Later, another lexicon was used to classify remaining data. However, use of machine learning algorithm would have given more accurate results. This approach helped in understanding how accuracy of the final model is improved by using two lexicons together instead of going for traditional single sentiment analysis approach. In automatic sentiment analysis vocabulary is built based on set of words being assigned.In order to improve user experience, artificial intelligence assistance is an emerging technology being used for carrying out sentiment analysis. It focuses on evaluating user experience and emotions while understanding user tendency through opinion mining. The author Park and Seo (2018) tried to identify which artificial intelligence assistant statistically performs well among the three chosen artificial assistants. Users opinion regarding the chosen three artificial intelligence assistants were divided into three categories positive, negative and neutral by using lexicon Valence aware dictionary along with VADER (sentiment reasoner). In order to identify the statistical stability of the three artificial intelligence assistants test like Mann- Whitney, independent sample T test, Krushal Wallis test were used. Improper optimization of natural language processing turned out to be a limitation of this work. This work has provided knowledge of another lexicon approach to carry out sentiment analysis. In El Rahman et al. (2019), the authors have taken a different approach to perform sentiment analysis on retrieved unstructured data from twitter. Supervised and unsupervised algorithms were used for performing sentiment analysis. Initially, data is retrieved using twitter API which was then pre-processed. In the next step data was pre-processed and cleaned. In the model building phase first unsupervised lexicon model was used to classify collected tweets data as pre-processed data did not had class labels assigned to it. Tweets were classified in positive, negative and neutral categories by matching the words of the tweets with a predefined library. In order to classify the tweets 1,0, -1 sentiment scores were assigned to them. In the next step, supervised models were implemented for training purpose. The used supervised models were Naive Bayes Classifier Model, Support Vector Machine, Maximum Entropy Classifier, Decision Tree and Bagging. As an advantage the use of multiple machine learning algorithms have improved the performance of model over other machine learning models. This model was unable to classify tweets automatically which turned out to be its limitation. This work has provided very critical knowledge regarding the models which can be incorporated for performing sentiment analysis.

After a thorough literature review on sentiment analysis using different techniques. These papers helped in deciding Natural language processing, NLTK library, Naive Bayes Classifier algorithm, and Python's TextBlob approach as core foundation to carry out work in this Research.

2.2 A Review on Sentiment Classification Using Natural Language Processing (NLP)

As articulated by Lobur et al. (2011), the natural language processing (NLP) is the domain in machine learning which is used in text analytics. NLTK which is called Natural language toolkit is a part of python's library belonging to natural language processing. Natural language processing not only deals with text analytics, but it also plays an important part with research based on analysis on human languages. Preparing models for research based on human languages comes in computational linguistics. The major advantage of using NLTK is that it allows even a beginner programmer to understand concepts of natural language processing. Thus, saving a lot of time from gathering information about it. Numerous advantages for using NLTK are, it contains 60 corpora belonging to real world data, collections of grammar, models which have been trained, functions which provides a path for performing general natural language processing tasks. The Table1 depicts the common functions performed in natural language processing.

Table 1: Functions of NLTK Library

Language Processing Task	**NLTK Modules**	**Functionality**
Accessing Corpora	nltk.corpus	standardized interfaces to corpora
String Processing	nltk.tokenize, nltk.stem	tokenizers,and stemmers
Collocation discovery	nltk.collocations	t-test, point-wise mutual information
Part of speech tagging	nltk.tag	n-gram, backoff, Brill, HMM,TnT
Classification	nltk.classify, nltk.cluster	naive Bayes
Chunking	nltk.chunk	regular expression
Parsing	nltk.parse	feature-based, probabilistic
Semantic interpretation	nltk.sem, nltk.inference	model checking
Evaluation metrics	nltk.metrics	precision,recall
Ptobability and estimation	nltk.probability	frequency distributions,
Applications	nltk.app,nltk.chat	WordNet browser, chatbots
Linguistic fieldwork	nllk.toolbox	manipulating data

The corpora used in NLTK are generally divided into different categories for assisting its users. Though in other programming languages, natural language processing tasks can be accomplished. The major points which keeps python apart from other languages are as follows.

- Better reading ability.
- User-friendly object-oriented technique.
- Ease of extensibility.
- Better Unicode assistance.
- A functionality rich library.

NLTK has vast source of libraries which are being updated with new functionalities over the period. This paper has provided deep understanding regarding functioning of NLTK library. Tasks such as summarization of text, extraction of information, machine translation are performed by NLP as depicted in work from Zitnik et al. (2017) here, the author has carried out sentiment analysis using natural language processing toolkit nutIE in order to detect the language of text and extract meaning out of it. For it, first the language dataset has been cleaned in the pre-processing stage which was then followed by language detection and evaluation of results. Major limitation of this work is that it does not compares performance of this library with other natural language toolkit libraries. Though, as an advantage this library can be used for natural language processing courses for educational purpose. Moreover, this work has provided understanding of use of natural language processing in language detection which is used in this research project.

2.3 A Review on Feature Extraction for Sentiment Classification

As stated by Zhang et al. (2010) , the one of the most important model utilized for categorization of object is Bag of Words (BoW). The concept behind Bow model is forming visual words by quantizing every extracted key point. After this each picture is shown using visual words histogram. Joachimsu (1998), also worked upon BoW model. He showed that BoW model depicts count of every word present in a textual data. Ma et al. (2018) showed that a matrix depicting count of words in textual data is created in BoW model. Afterwards, frequency of occurrence of these words are used as features for the purpose of training the classifier. Luong and Manning (2015) conducted a research where it is observed that BoW model performed considerably well in comparison with other models on Chinese English language translation data. All these works have helped in understanding the concept behind BoW Model for feature extraction. S.Vijayarani and R.Janani (2016),emphasized on various steps being taken while preprocessing the data-set. Various steps which were taken for pre-processing dataset are stop words removal, determination of sentence boundary, tokenization and stemming. Tokenization is one of the most important steps while pre-processing a dataset. It works in a manner that textual data is divided into small tokens. Each token represents a word from the textual document or language. There are numerous libraries available in python such as NLTK word tokenize, Mila tokenizer, TextBlob tokenizer etc which are used for tokenization. This work has helped in understanding the in-depth functioning of TextBlob library for pre-processing phase.

2.4 A Review on Using Naive Bayes Algorithm For Sentiment Classification

The Naive Bayes Classifier is a probability-based algorithm which is mainly used for text classification purpose. It works on the concept of Bayes probability theorem. According to it the probability of presence of a specific component in a class is random as compared to presence of some other part. In Rana and Singh (2016), the authors have tried to carry out sentiment analysis on reviews on drama by using Naive Bayes Classifier algorithm, Support Vector Machine(SVM) and Synthetic words approach. Reviews were first pre-processed and then data mining is performed by using Naive Bayes Classifier algorithm and Support Vector Machine(SVM) which is followed by comparison of results. In results it is observed that Support Vector Machine gives better accuracy as compared to Naive Bayes Classifier algorithm. The comparison among Naive Bayes Classifier algorithm and SVM was an advantage as it brought up another approach for text classification over the traditional approach. However, the accuracy was obtained on reviews based on drama. Therefore, the major limitation of this work is, for reviews based on other topics the accuracies may vary and that can make Naive Bayes a better model as compared to Support Vector Machines for text classification. This work has not only helped in understanding Naive Bayes Classifier algorithm but also porter stemming algorithm which plays an important part for removing suffixes from words as a part of pre-processing of text. Ibrahim and Yusoff (2017) tried to test the accuracy of Naive Bayes Classifier algorithm on different size of datasets. Sentiments were classified in positive, negative and neutral categories and 5 different datasets with dataset size 5,10,25,50 and 100 tweets were used. In order to train the model five users were used to classify the words in positive, negative and neutral categories. The training results were then given to Naive Bayes classifier algorithm which produced accuracy results of 46%,78%,89%,87% and 79% for 5, 10,25,50 and 100 tweets dataset respectively. The advantage of this work is that it removes the confusion that the Naive Bayes Classifier is a weak model as compared to Support Vector Machine. However, the fact that this work was done on small groups of datasets, it turns out to be its limitation as works done on bigger datasets produces more accurate results. It has provided an understanding about functioning and performance of Naive Bayes Classifier algorithm while choosing different amount of dataset for this research. Most work done on sentiment analysis have been mainly focussing on documents or datasets from English language. Therefore, the advantage of the work done by Sarkar (2018) is that it resolved the limitation of no work on a different language. Here, the author has used combined supervised and unsupervised techniques to determine sentiment scores from Bengali language.Here, the author has implemented Multinomial Naive Bayes and Character n gram approach. In order to remove noisy data, the characters from every tweet have been tokenized using character n gram approach which are then used in multinomial Naive Bayes Classifier for classifying the tweets. It is observed that multinomial naive Bayes with character n gram possess better accuracy as compared to multinomial naive Bayes with word n gram. Less training data and wrongly labelled parts of the data were limitation of this work, which can be work upon to improve performance of the model. This work provided understanding to deal with tweets written in different languages on BREXIT and in-depth functioning of Multinomial Naive Bayes Classifier algorithm. Permatasari et al. (2018) proposed a new approach in which just Bag of words were not used in feature selection. Apart from it, they used ensemble features which included bag of words with lexicon-based features, twitter specific features, textual features and part of speech feature. In order to implement the model first extraction of ensemble features was done from training and test data after which features results were then fed into Naive Bayes Classifier Model which then labelled the tweets in Positive and Negative classes. In result it is observed that Naive Bayes Model with bag of words feature performed well as compared to Naive Bayes Model with ensemble features. The advantage of this work was that it has removed the misconception that ensemble feature always performs well. However, the only limitation of this work is that the author tested the model only on movie reviews, therefore, on different datasets ensemble features with Naive Bayes Classifier algorithm may perform better than bag of words features with Naive Bayes Classifier algorithm. This work has provided knowledge pertaining to bag of words features used for sentiment analysis. In Matharasi1 (2017), author has conducted sentiment analysis on twitter data using Naive Bayes Classifier algorithm with unigram approach. Before using Naive Bayes classifier firstly, the dataset was cleaned then the Naive Bayes Classifier model was first trained and then the stability of the output is tested by using cross validation, holdout method, k- fold cross validation and leave one out validation method. In these methods training dataset was divided into training datasets and validation datasets which were then used to evaluate performance of the algorithm. Later, Naive Bayes classifier algorithm was used to calculate the sentiment scores which were classified in positive, negative and neutral categories. The classifier performed reasonably well but had some errors in output which turned out to be the limitation of this work. The major advantage of this work is that it deals with Naive Bayes Classifier algorithm on categorical data. This work has also helped in understanding different validation methods. Moreover, the implementation of Naive Bayes Classifier algorithm is understood with a new approach.

2.5 A Review on TextBlob Approach Algorithm for Sentiment Classification

One of the pythons library which uses API for accessing methods in order to perform Natural language processing is called as TextBlob. A common challenge for work based on sentiment analysis are miss spelled words. This problem is addressed by Manushree et al. (2017). Here, authors have compared TextBlob and SentiWordNet approach. Firstly, the dataset was pre-processed by removing stop words and unrequired data which could result in added computational cost in performance of models. It was followed by aspect selection and based on it sentence extraction was done. Both the models were then used to calculate sentiment polarity and categorize the reviews in positive, negative and neutral categories. This work just focussed on sentiment analysis of miss spelled words in English language. The advantage of this work was that it performed sentiment analysis on miss spelled words. However, limitation of this work is that it was unable to perform sentiment analysis on miss spelled words in other language. Moreover, this work has helped in gaining depth understanding regarding implementation of TextBlob approach for the research project. Shobana et al. (2018), also carried out research work to perform sentiment analysis on twitter tweets using pythons textblob library. Firstly, tweets were fetched using twitter streaming API. It is followed by cleaning the tweets dataset in pre-processing stage. The authors have then performed feature extraction which is then followed by training the Naive Bayes model. Afterwards, classifier is used for classifying tweets in positive, negative and neutral classes. The major limitation of this work is that it is unable to test sentiment score for slangs and short words in the form of abbreviations used in a text message. This work has also provided understanding of using python's textblob library. The pre-processing of the textual data is of very importance in sentiment analysis as it reduces the size of textual data which is given as input to the model. Various steps are followed while pre-processing the textual data. The pre-processing tasks performed for cleaning textual data include determination of boundary of sentences, removal of stop words from natural language, stemming and tokenization. Tokenization involves splitting a sentence into tokens of each word belonging to the respective sentences. S.Vijayarani and R.Janani (2016) carried out work on certain tokenization tools including TextBlob in order to test the performance of selected tokenization tools. In the results it is observed that TextBlob performed significantly well in order to tokenize and read the tokenized words. Advantage of this work is that it compared various good tokenization tools and it distinguished TextBlob from them. However, it was unable to read tokenized special characters which turned out as its limitation. This work has helped in understanding major limitation of TextBlob.

2.6 A Review on Python's Regular Expression (REGEX/RE) Library

Chapman and Stolee (2016), concentrated mainly on regex, which is also called as regular expression. It is reflection of specific words search which helps in identification of text through recognition of patterns in place of exact strings. REGEX library is commonly utilized for parsing textual data belonging to general language. Regex are also called as Python's module. Even though regex is considered as versatile and powerful library it could be difficult to understand, this is one of its limitations. According to Spishak and Dietl (2012), The major advantage of python's regular expression library is that it has variety of applications as it has powerful ability to fetch meaningful information from a given sentence. Regular expression is applicable in preprocessing the data, MY SQL injection, generation of test cases and intrusion detection in networks etc. According to Ganesh and Artzi (2012) and Yeole and Meshram (2011), The major advantage of regular expression library is that it has fast processing speed in terms of code execution, and it has very compressed code which reduces efforts of writing long codes for pre-processing. All these works have helped in understanding pythons regular expression library for pre-processing of dataset.As understood from these works,the one of the major Advantage of using python's regular expresssion library is its ability for fast processing and code compressibility.

2.7 A Review on Sentiment Classification on Public Opinions About BREXIT.

So far, there have been a very few researches done based on sentiment analysis on BREXIT. The authors Lansdall et al. (2017) in their work, carried out sentiment analysis on BREXIT by collecting data from twitter. They have categorized the tweets in positive, negative and neutral categories using LARS algorithm. This work was based on comparing the public mood swing before and after BREXIT until early 2017. It could have been more informative if the authors would have considered a wider aspect into their analysis like possible impact of BREXIT on United Kingdoms economy. Advantage of this work is that it has given a fair idea about variation in mood swing in public sentiments before and after BREXIT.' Moreover, this work has provided a case study-based understanding on BREXIT.The European Union initiated a venture under the name of SSIX (Social sentiments financial indexes). Vasiliu et al. (2016) used natural language processing to determine public sentiments on BREXIT by categorizing the retrieved tweets into positive, negative and neutral sentiments and assigning sentiment scores to them. The major advantage of using SSIX platform was that it was able to detect and understand text of other European languages. However, this work had certain limitations such that it was unable to retrieve location, age and gender gap bias associated to tweeter users. Moreover, This work has provided understanding of application of natural language processing for sentiment analysis on BREXIT. In Khatua (2016), the author has collected over 2.7 million tweets before BREXIT referendum to carry out sentiment analysis. tweets were categorized in positive, negative and neutral categories. Hierarchical clustering analysis (HCA) was used to calculate sentiment scores. In the results author was successful to predict outcome of the referendum. The advantage of this research was that apart from public sentiments it was also able to find out the topics people were talking about like possible impact of BREXIT on UK and USA relation.

The limitation of this work is that author did not researched on a broader scale using geographical location of twitter user to find out what are the sentiments of people living in business-friendly countries of UK. This work has also provided knowledge about scale at which sentiment analysis can broaden the research for topics like BREXIT.

2.8 Identified Limitations in Previous Works Based on Sentiment Classification on BREXIT

After a thorough literature survey on sentiment analysis of BREXIT, it is observed that there has not been much work done on this topic. There have been few researches on BREXIT and all of them were conducted in year 2016. However, those researches were mainly focussed on analysing public mood swing based on public sentiments before and after BREXIT referendum. There was just one research which concentrted on finding out trending topics related to BREXIT being discussed on social media platform like twitter. So far, there has not been any work done on analysing the public sentiments based on locations and obtaining insights belonging to the major trade partners countries(Business friendly countries which are contributing positively in United Kingdom's economy/GDP.) of the United Kingdom. This problem is unique as it will assist United Kingdom and Irish government to formulate their business, internal and foreign policies based on insights obtained from sentiment analysis. Also, so far there has not been any research done on sentiment classification comparing performance of Naive Bayes Classifier model with python's TextBlob approach. Therefore, both these problems are unique.

3 Methodology

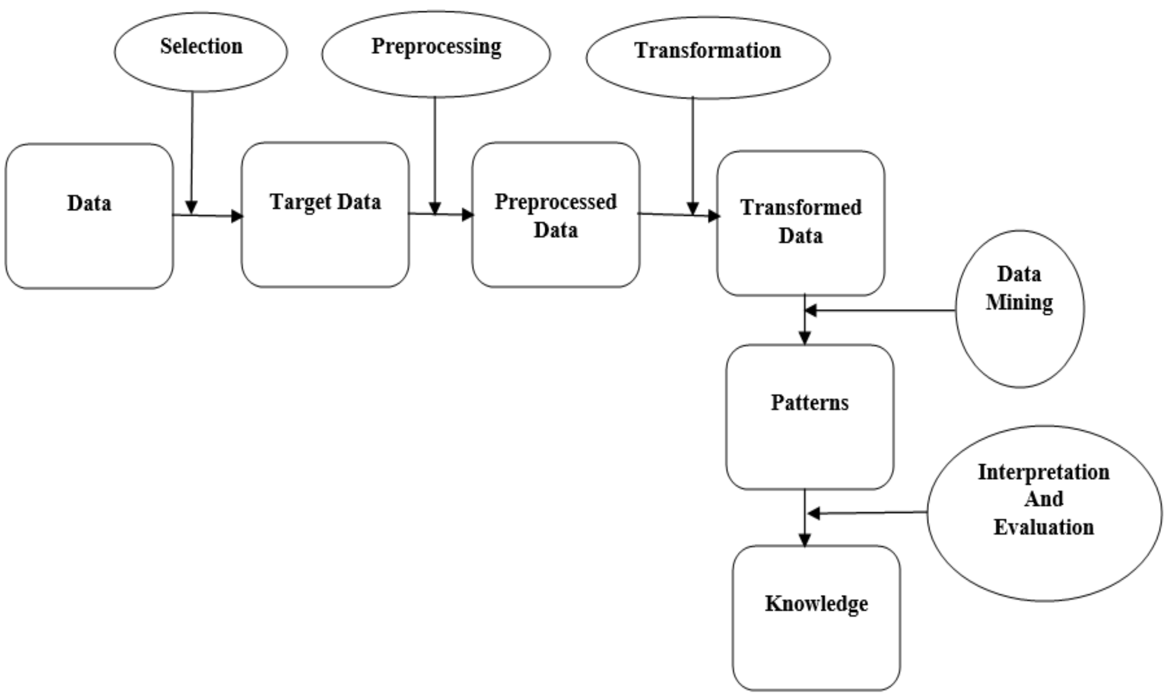

Figure 1: Knowledge Discovery and Data Mining

As articulated by Goebel (2014), this project uses modified KDD (Knowledge discovery and data mining) methodology. As depicted in Fig 1, the methodology explains the process and the KDD concept used for sentiment analysis on BREXIT data obtained from twitter.

- Dataset Preparation:- During this phase dataset has been collected and prepared for research.
- Data Pre-Processing:- In this phase different natural language processing (NLP) techniques have been used to clean dataset. This step is very important in order to prepare dataset for next steps.
- Data Transformation:- in this phase the pre processed dataset has been transformed in a format suitable to implement data mining techniques.
- Data Mining:- this phase is used to implement data mining models.
- Interpretation/Evaluation:-During this phase interpretation of patterns using visualizations is done. Also, performance of implemented model is tested using data mining concepts.
- Knowledge:- Using visualizations and model evaluation results, knowledge about dataset and model performance has been gained.

4 Design Specification

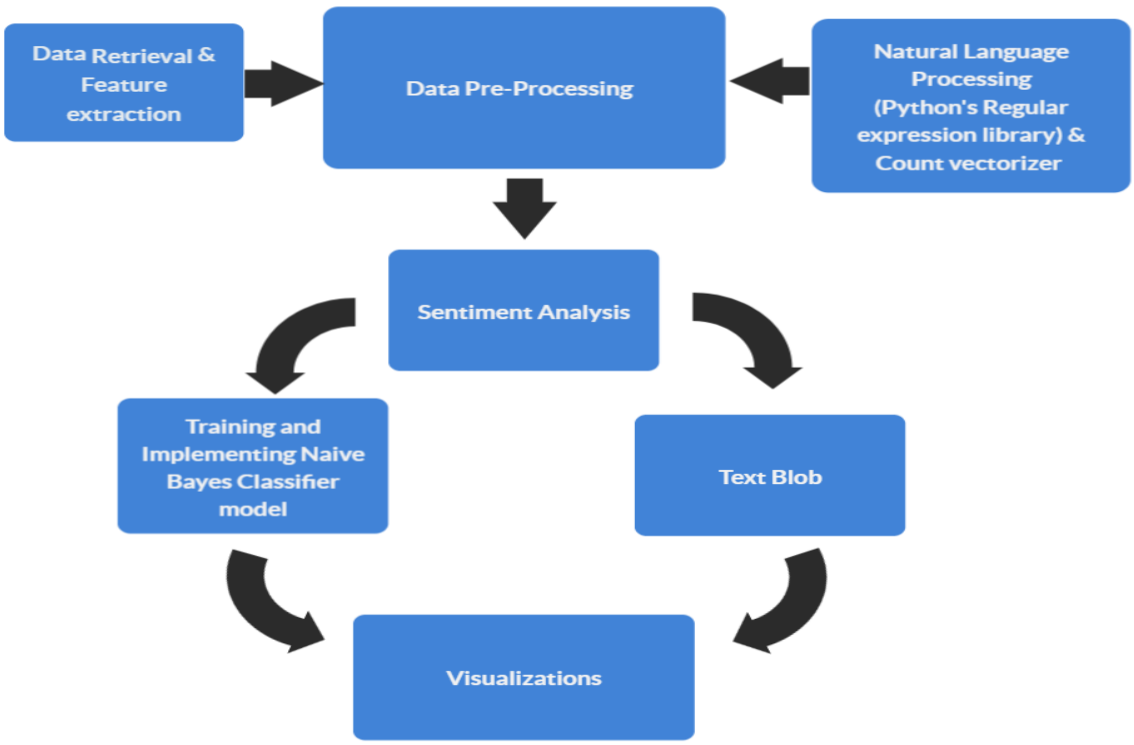

Figure 2: Project Workflow

This section explains the workflow of the research project using Naive Bayes Classifier algorithm and Python's TextBlob approach. Two separate datasets have been used for training and testing Naive Bayes Classifier Model. Also, for implementation purpose python 3.0 has been used. The workflow for this research is illustrated in figure 2.

- Created dataset using Twitter streaming API.
- Used natural language processing(NLP) consisting python's regular expression library to clean the dataset and count vectorization to convert data into small pieces of tokens.
- Conducted sentiment analysis.
- Firstly, sentiment analysis was conducted using Naive Bayes Classifier algorithm
- In the next step sentiment analysis was incorporated using Python's Textblob library.
- Finally, results containing insights from retrieved dataset were visualized using business intelligence tool Tableau.
- After implementation performances of both approaches were compared

5 Implementation

5.1 Data Preparation

In order to prepare dataset for this research following data sources have been used.

5.1.1 Twitter

For fetching tweets data from twitter. Initially, an API request was made to twitter which was later approved. Afterwards, as explained by Shah et al. (2018) python's tweepy library which is specifically developed for retrieving tweets data from twitter is used along with twitter streaming API and authentication keys (Consumer_key, consumer token key , access token and access token secret.) provided by twitter.
For authentication purpose, the Tweepy library uses the OAuthHandler function for verification of authentication keys. Once, authentication request is approved it starts fetching the tweets.

- **Python's Tweepy Library:-**

```
import tweepy,pandas as pd
import sys
import jsonpickle
import os,random


auth = tweepy.AppAuthHandler('xOCDelyewVjVLvqUhVPOFnisD', 'sd6YM3RScVq8qz9yG0P9GmZBuPNG195Z4bLj\

api = tweepy.API(auth, wait_on_rate_limit=True,wait_on_rate_limit_notify=True)

if (not api):
    print ("Can't Authenticate")
    sys.exit(-1)
```

Figure 3: Pythons Tweepy Library

Figure 3 shows use of pythons tweepy library to retrieve tweets from twitter. Shah et al. (2018) explained the use of tweepy library in his work. Once API strategy was invoked, a tweepy class instance was then sent back to the requester. It includes information sent back to us by twitter which was later used within our application.

Tweets based on BREXIT were fetched by different user account by using.
tweets = api.home_timeline()
for tweet in tweets:
print(tweet.text)

The tweepy function user_timeline() was used to fetch recent tweets of users by using.
tweets = api.user_timeline()
for tweet in tweets:
print(tweet.text)

- **Pandas Libraray:-**

```
data = pd.DataFrame([twts,places,dates,likes,retweets],
                    index=['tweets','place','date','likes','retweets'])
```

Figure 4: Pythons Pandas Library

Pandas is a python library used for analyzing data through manipulation. This library was used to provide final shape to the collected tweets dataset.The pandas library was used to fetch all the collected tweets in a data frame. As shown in figure 4.

5.1.2 UkTradeinfo.com

UKTradeInfo.com is an open data source. It is used to retrieve dataset showcasing the United Kingdom's (UK) economy statistics in terms of net contribution to GDP by major trading partners of UK. The collected dataset comprised of figures explaining net imports, exports and contribution to UK economy (in GBP) by respective countries.

5.2 Data Pre-Processing

The raw dataset retrieved from twitter is cleaned in the preprocessing stage using natural language processing concepts as stated by Jettakul et al. (2018) and Pal et al. (2015). In order to calculate sentiment scores, it is essential to clean the dataset such that machine easily understands the text. Cleaning dataset using natural language processing involves a science. The detailed steps used while pre-processing the dataset is as depicted in figure 2.

5.2.1 Use of Natural language processing(Pythons Regular Expression Library)

As explained by Goyvaerts (2006), pythons regular expression (RE) library has been used to remove unnecessary data from text messages of tweets. Figure 5, depicts pre-processing

```
data['text'][:2].apply(lambda x: re.sub('(@[^ ]+|#[^ ]+|\\n|http[^ ]+|[^\w ])','',x.lower())).values
```

```
array(['   awww thats a bummer  you shoulda got david carr of third day to do it d',
       'is upset that he cant update his facebook by texting it and might cry as a result  school today also blah'],
      dtype=object)
```

Figure 5: Data Preprocessing using Regular Expression Library

of a tweet using regular expression library. The unnecessary data removed from tweets involved.

- **URLS:** A lot of users use different hyperlink urls in their tweets. Removing such urls was necessary as they did not contribute towards calculation of sentiment score. Also, such urls brings in data redundancy which adds additional computational processing burden.
- **Removal of usernames:** In twitter usernames starts with @ which is of no use in sentiment analysis. Therefore, such usernames starting with @ were removed.
- **Removal of special characters:** There are various special characters being used by twitter users which needed to be cleaned to make dataset easily readable by the machine. The special characters removed were Stop(.), inverted commas ( ) , exclamation marks(!), special characters like @ , commas(,).
- **Removal of hastags:** Many twitter users express their topic of discussion with (eg:- #BREXIT, #ENGVsAUS). These # are of no use in calculating sentiment scores. Therefore, hash # were removed from the dataset.
- **White Spaces:** Many users on twitter leave unnecessary white spaces which were removed while cleaning.

5.2.2 Count Vectorizer

The process of processing textual data into numerical form is called as count vectorization. It is a type of encoding. It comes in the last stage of pre-processing.

- Depending on the size of vocabulary, different vectors are created.
- When a specific word is detected in the vocabulary then 1 is assigned as a count for that word.
- Every time when a word repeats in a vocabulary, its count is increased by 1.
- Zeros represent all those words which doesn't occur even once in vocabulary.

Count Vectorizer has also helped in performing tokenization.

- **Tokenization:-** One of the crucial steps performed as a part of natural language processing (NLP) is tokenization. As stated by Garg (2015) in this stage each word of a textual document is splitted from sentence in the forms of tokens and all the created tokens collectively forms a feature set. Sentences were tokenized into tokens of each word to form feature set. Eg:- Sentence:- "this is a sentence" Feature set after tokenization:- 'this' ,'is', 'a','sentence'

5.3 Sentiment Analysis

Once the dataset was pre-processed, in the next stage sentiment scores were calculated by using Naive Bayes Classifier Model and Python's TextBlob Library. Both the approaches were implemented as follows.

5.3.1 Naive Bayes Classifier Model

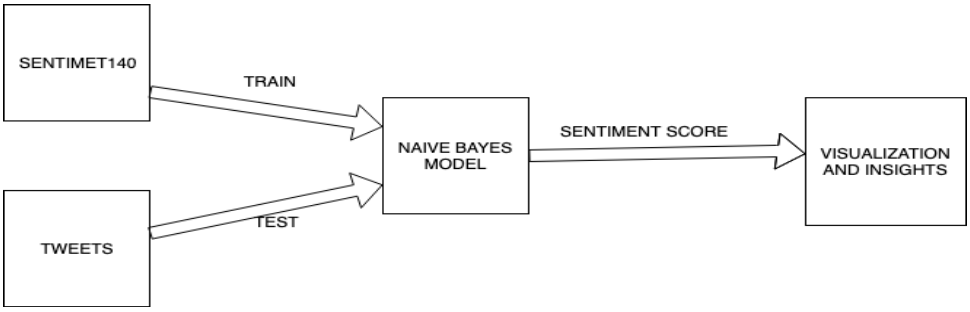

Figure 6: Naive Bayes Classifier Model

The Naive Bayes Classifier algorithm is a probability-based machine learning algorithm, which is utilized for text analysis. its main concept follows Bayes theorem, which states that the occurrence of an article within a class is random to presence of other components. Mathematically Bayes theorem says that probability of event A given that event B has occurred is given by,

$$P(A/B)=(P(B/A).P(A))/P(B) \tag{1}$$

where,
A : hypothesis
B : evidence

- **Training Naive Bayes Classifier Model:-** In order to train Naive Bayes Classifier Model Sentiment140 (2019) dataset has been used from open source data website Kaggle. Sentiment140 is a well-known dataset which comprises of tweets depicting reviews and opinions regarding different topics and products. It is comprised of over 1.6 million tweets. This dataset has helped in calculating sentiment scores of its tweets which helped in training the Naive Bayes Classifier Model. Initially, model was utilized to train on 15000 tweets. Afterwards, 50,000 tweets were used to train the model. Following numerical values were assigned to tweets while calculating sentiment scores.
  - 0 for Negative polarity tweets.

  - 2 for Neutral polarity tweets.
  - 4 for Positive polarity tweets.

- **Testing Naive Bayes Classifier Model:-** The Naive Bayes Classifier Model was tested on the collected dataset of 2.18 million tweets as shown in figure 6.

```
res_ = model_NB.predict(our_data.text)

NB_res = ['POSITIVE' if(i==4) else 'NEGATIVE' for i in res_]
NB_res[:5]
```

```
['POSITIVE', 'POSITIVE', 'NEGATIVE', 'POSITIVE', 'NEGATIVE']
```

```
[ ] res_
```

```
array([4, 4, 0, ..., 0, 4, 4])
```

Figure 7: Testing Naives Bayes Classifier Model

The sk learn library from python is used to implement Naive Bayes Classifier model. In order to label collected tweets dataset, python's predict() function has been used. Once, prediction was completed then newly labelled dataset along with sentiment scores of respective tweets were obtained.

5.3.2 Sentiment Analysis using TextBlob Library

TextBlob is a library from python which is used for processing data. The functioning of TextBlob library is as shown in figure 7. Once data was cleaned it was passed through TextBlob library in order to generate sentiment scores. This approach classifies polarity

```
def analyze_sentiment(self, tweet):
    analysis = TextBlob(self.clean_tweet(tweet))

    if analysis.sentiment.polarity > 0:
        return 1
    elif analysis.sentiment.polarity == 0:
        return 0
    else:
        return -1
```

Figure 8: TextBlob Approach

of textual data in positive, neutral and negative categories with '1','0' and '-1'. The sentiment scores for collected tweets is calculated as shown in figure 7.

5.4 Visualizations

Once sentiment scores of the tweets were calculated, Tableau is used as the business intelligence tool in order to build graphs and interactive dashboards showcasing trends and patterns. Insights obtained pertaining to the collected tweets data is as follows.

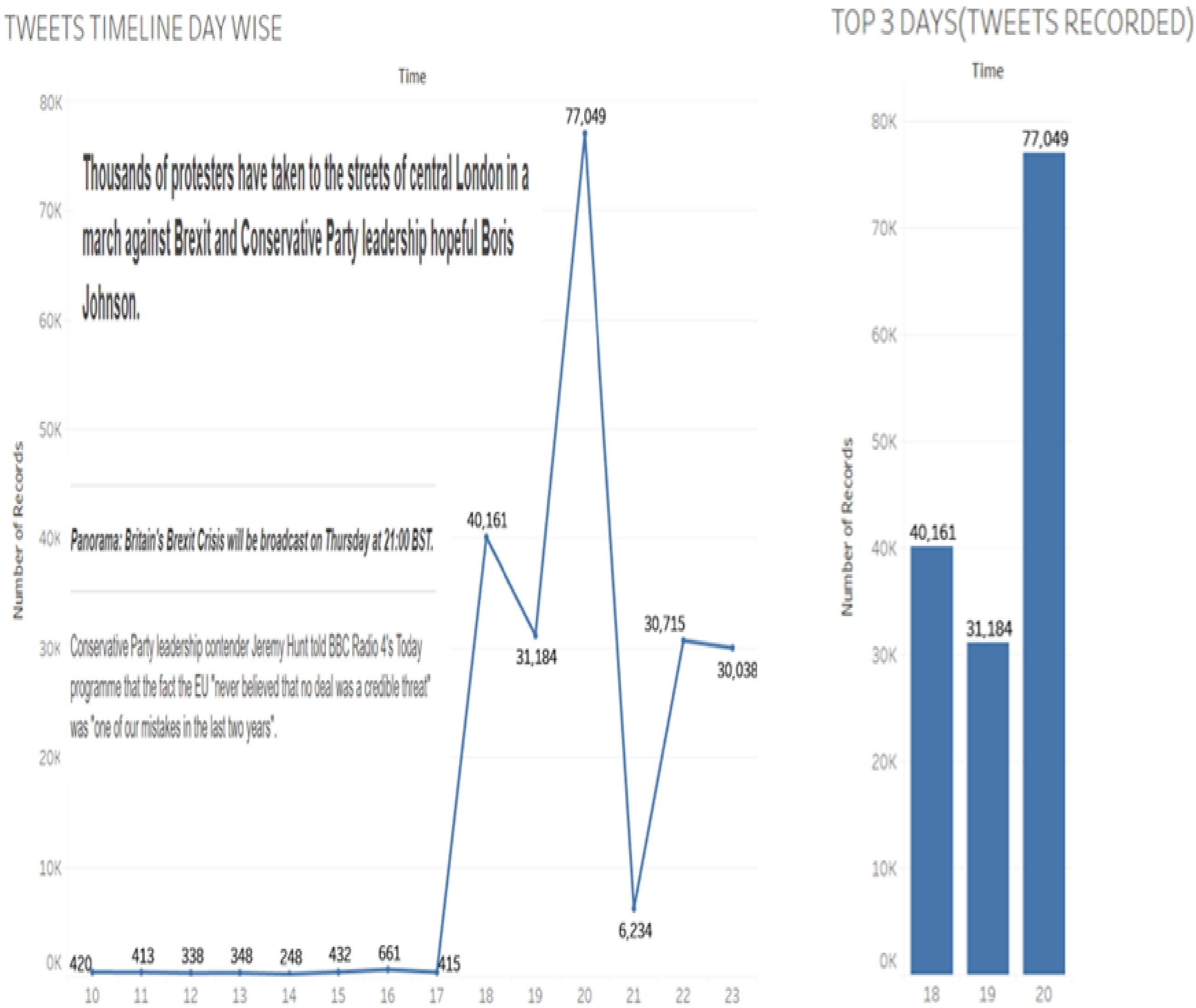

Figure 9: Most number of recorded tweets

Figure 9 depicts visualizations built from collected tweets between 10th July 2019 to 23rd July 2019. It is observed that there is a sudden surge in the number of tweets on 18th and 20th July 2019. As shown in figure, on 18th July 2019 the conservative party minister of parliament Mr. Jeremy Hunt held a show on BBC radio in order to clear general rumors about BREXIT among public and explained the positive side of it. (BBC; 2019c) However, on 20th July 2019, on London's street more than thousand people carried out a rally to show their protest against BREXIT in order to put forward their opinion before the newly appointed Prime Minister of England Mr. Boris Johnson. BBC (2019a) Therefore, it is interesting to find out, what impact does these two events had on sentiment of people belonging to important countries for United Kingdom.

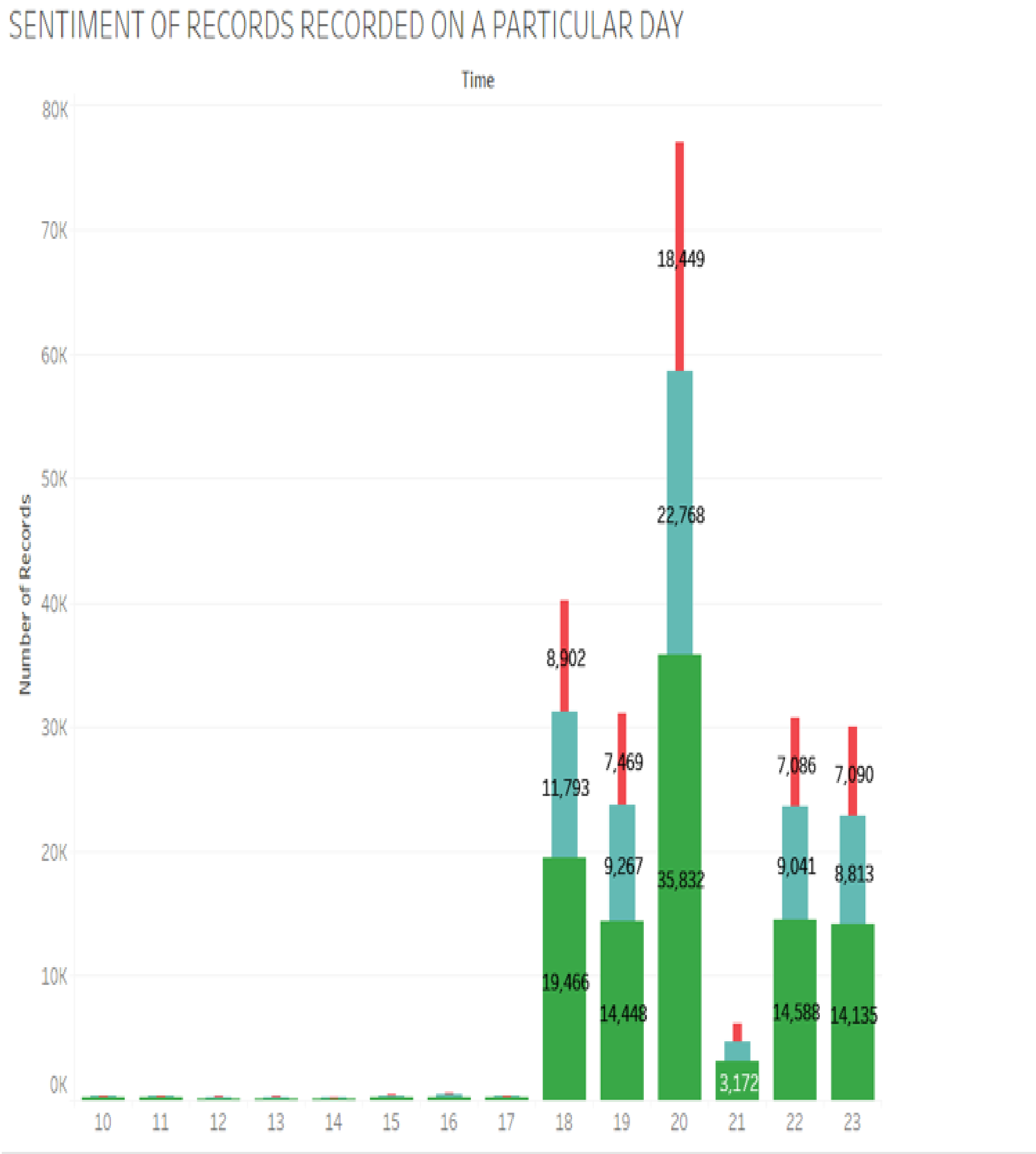

Figure 10: Sentiment wise number of recorded tweets

Figure 10 demonstrates the count of category wise sentiments of all the tweets received on the respective days (10th to 23rd July 2019). The tweets were categorized into Positive, Negative and Neutral classes.

5.4.1 Dashboard1

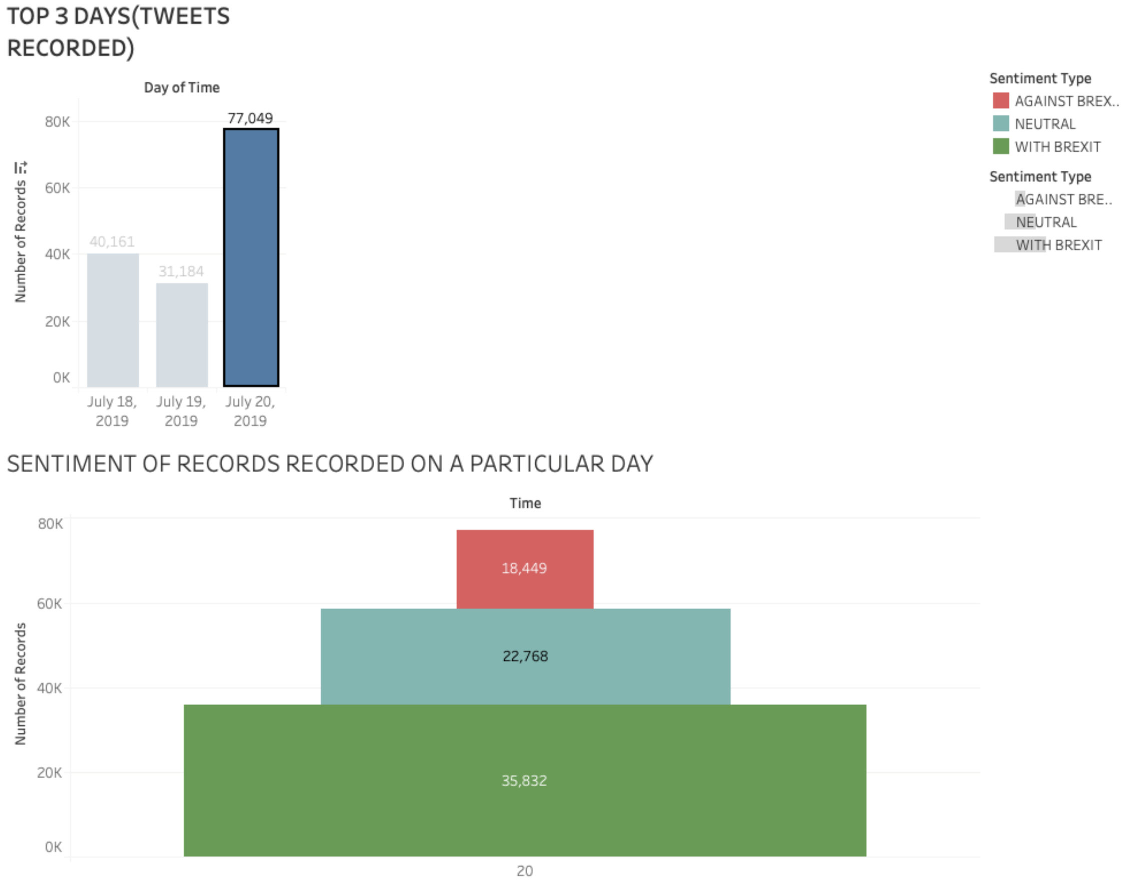

Figure 11: Dashboard for Cross Checking Data Accuracy

The figure 11 depicts a dashboard which was created to verify the accuracy of cleaned tweets dataset used for analysis. It consists of two linked graphs; the first graph depicts the top 3 days for which tweets have been gathered. Whereas, second graph has been built to showcase tweets count based on sentiment type on respective days for duration of 10th July to 23rd July 2019. When statistics for 20th July 2019 were selected then corresponding respective tweets count based on sentiment type (Positive = 'With Brexit', Neutral, Negative = 'Against Brexit') were reflected on the graph 'sentiments of records recorded on a particular day'. Therefore, for cross verifying the number of tweets gathered on 20th July and count of sentiment type of the tweets we can just add up the tweet count for sentiment type and then check whether the total number comes equal to count of the tweets obtained on 20th July 2019.

Total Count of Tweets in a day = Count of (With Brexit + Neutral + Against Brexit) Tweets

Therefore, Total Count of Tweets gathered on 20th July 2019 = 18,449 + 22,768 + 35,832

Total Count of tweets gathered on 20th July page2019 = 77,049

As, above calculation matches with total number of tweets obtained on 20th July 2019. It is evident that the cleaned tweets dataset is free of redundancy.

5.4.2 Dashboard 2

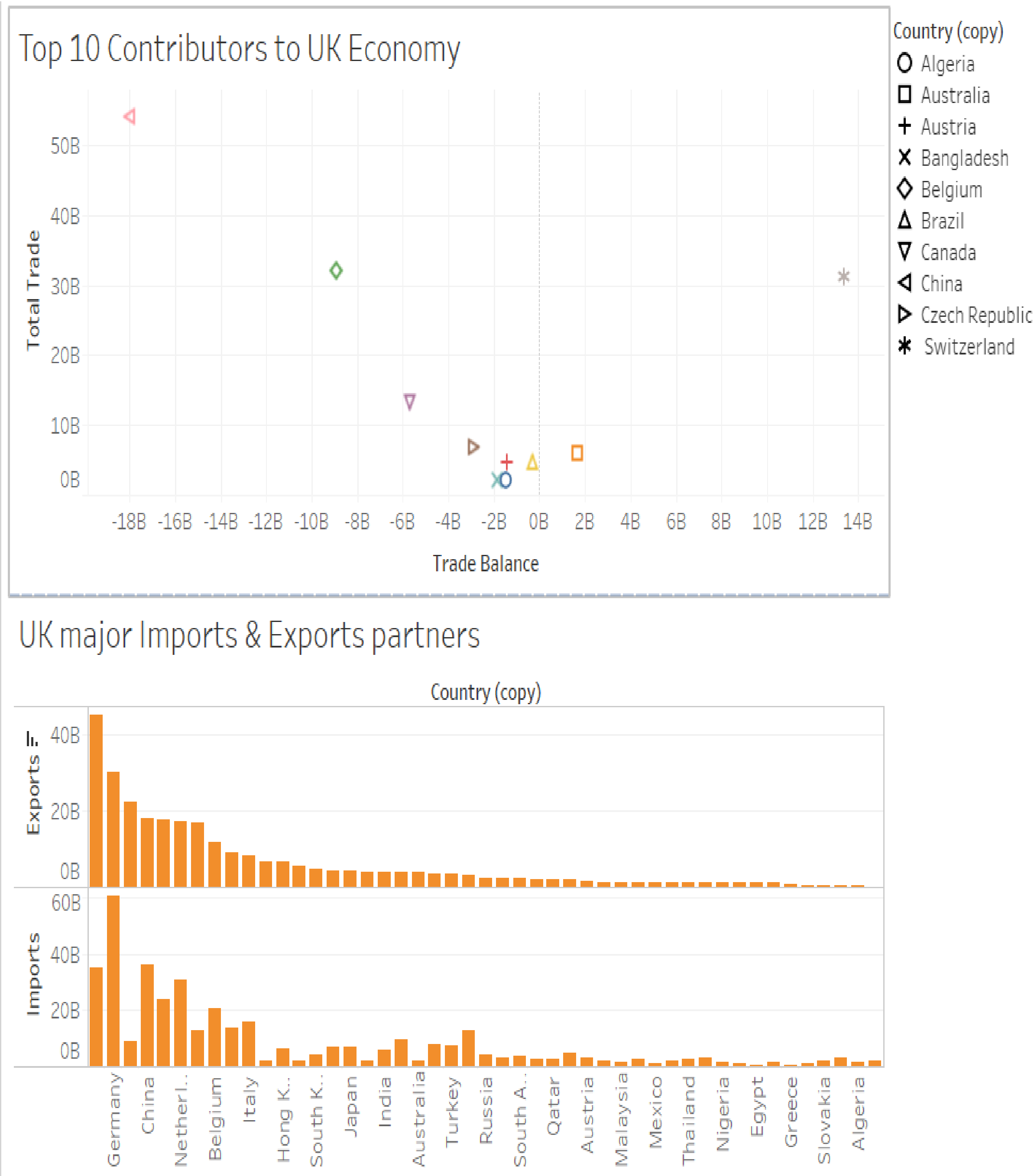

Figure 12: Major Contributors to United Kingdoms(UKs) Economy

The dashboard 2 built in figure 12 depicts United Kingdom's economy at a glance. The graphs used in dashboard have been built using the data obtained from Uktradeinfo.com. the first graph (Scatterplot) in the dashboard represents top 10 countries contributing in United Kingdom's (UK) economy in billion GBP. Whereas, the second graph (Bar plot) is showcasing Imports and Exports trade (in billion GBP) of UK with respective countries.

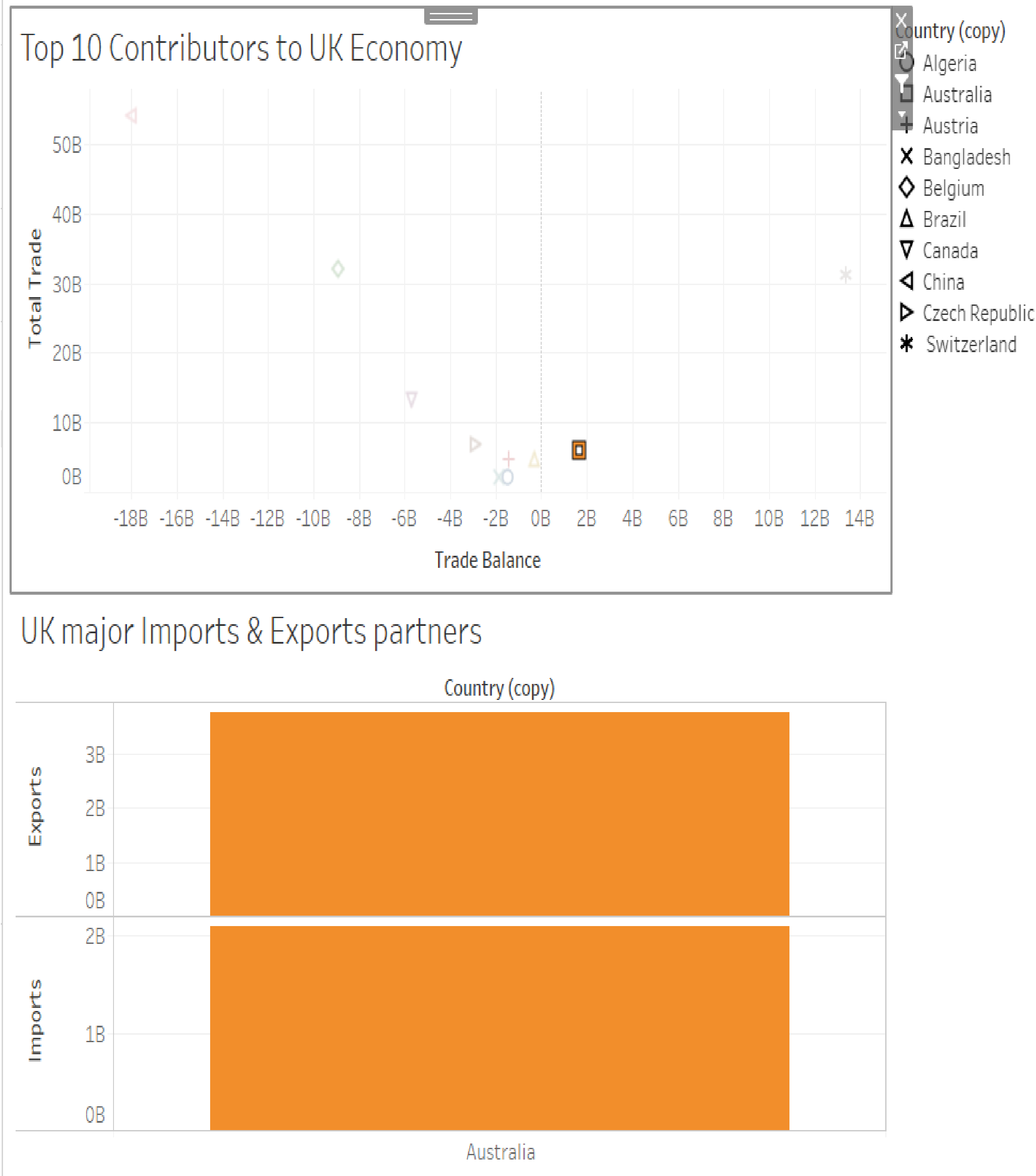

Figure 13: Australia's contribution in UK's Economy

A country's contribution in UK's economy has been calculated based on difference in total Exports (in billion GBP) and Imports (in billion GBP) between UK and respective countries. Therefore, a country's positive contribution in UK economy is determined by finding the difference between its total export and import with UK. As, depicted in the dashboard 2(figure 13), overall contribution of Australia in UK's economy is positive. When Australia is selected in the scatter plot depicting difference in total trade and trade balance, corresponding exports and imports figures are illustrated in dashboard 2(figure

13).

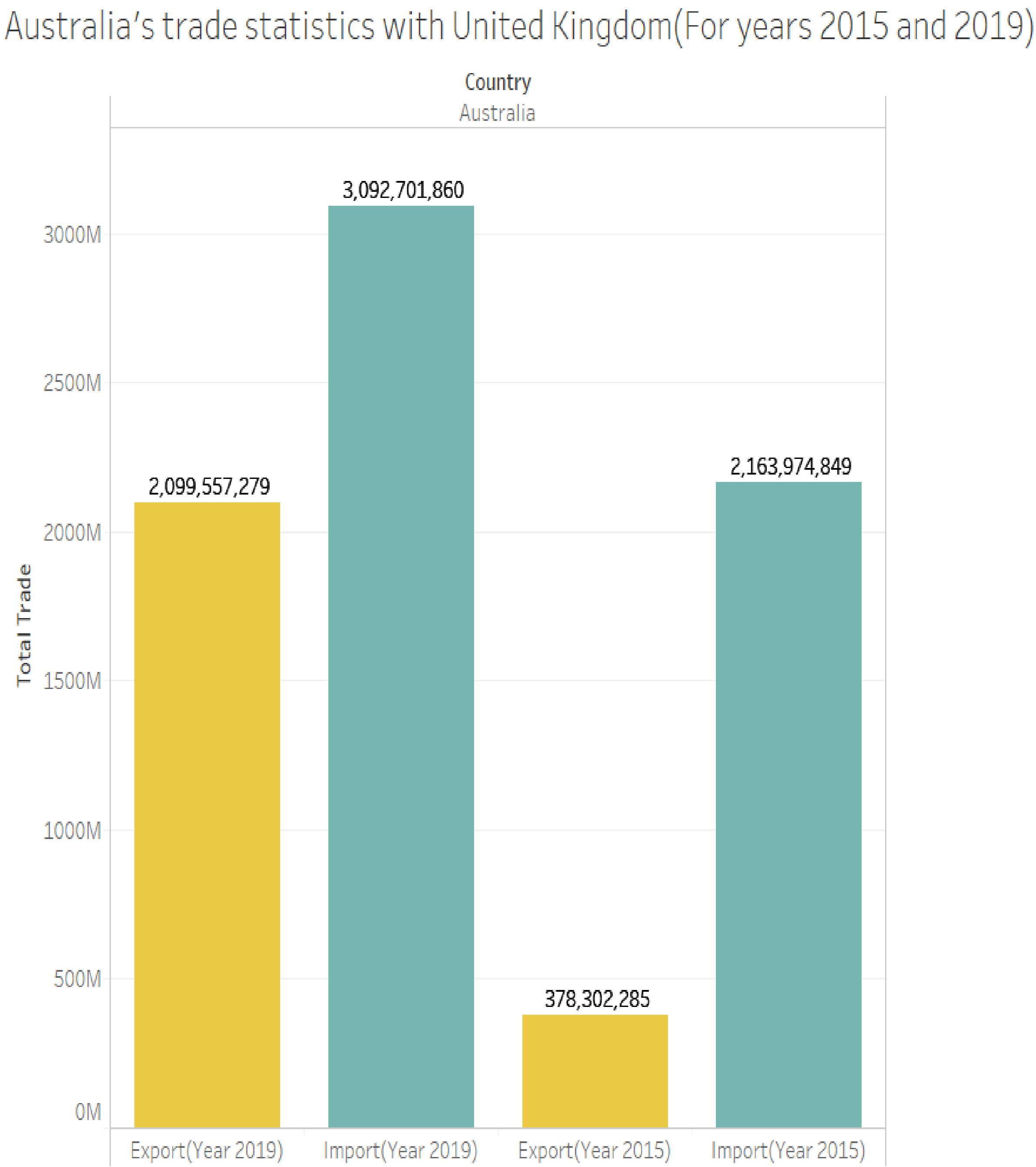

Figure 14: Australias trade statistics with United Kingdom

Figure 14 illustrates Australia's trade statistics with United Kingdom before and after BREXIT referendum. This visualization has been prepared from the data retrieved from UKtradeinfo.com. It is observed that Australia's trade statistics with UK have skyrocketed since year 2015. Therefore, it is now evident that Australia is one of the extremely important countries for UK and it is very important for UK government to gain insights related to the impact on Australian public due to the events held on 18th and 20th July 2019.

5.4.3 Dashboard 3

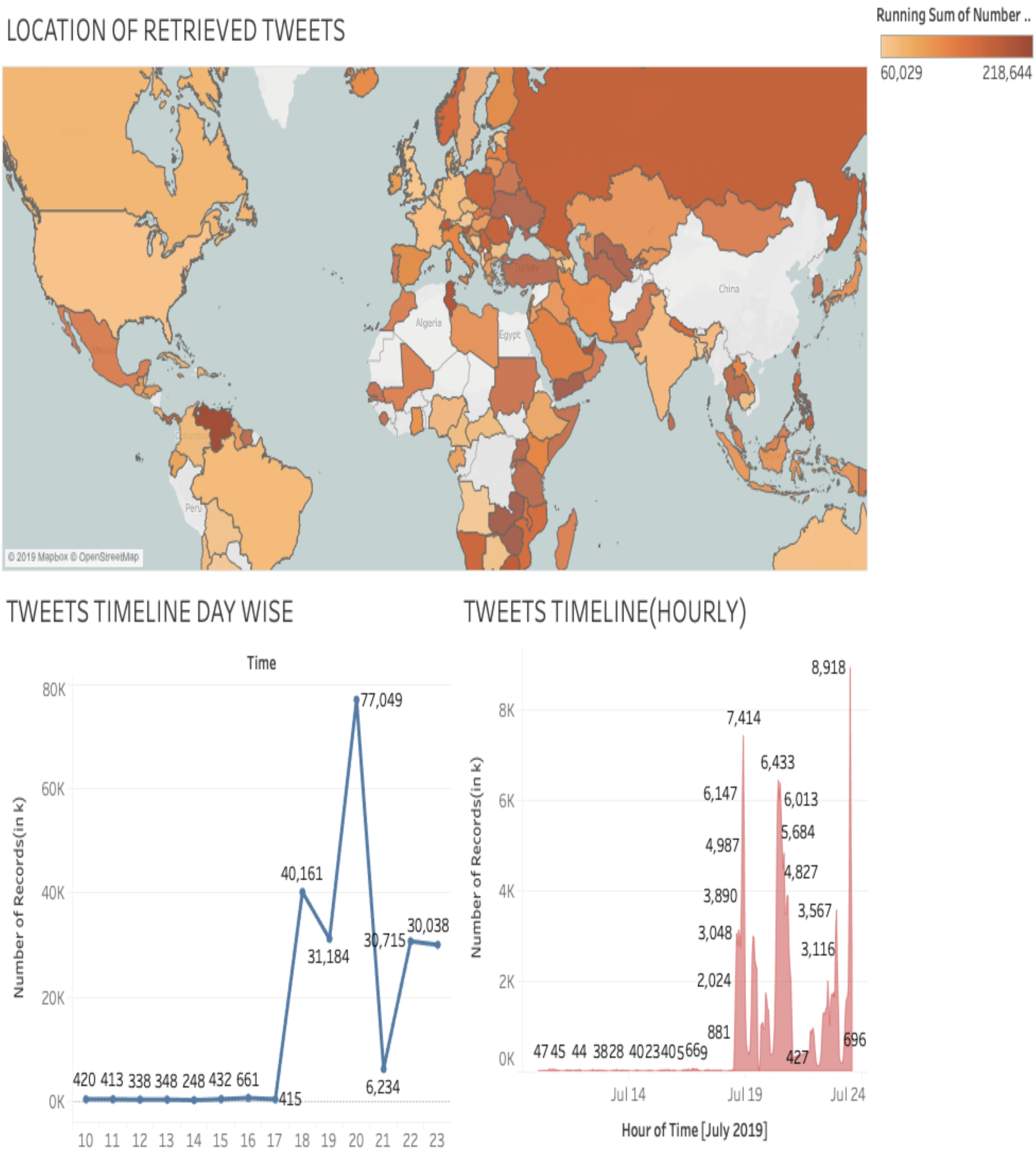

Figure 15: Location and Count of Tweets based on Day and Time.

The dashboard 3 shown in figure 15 was built using sentiment scores of the retrieved tweets. It was built for illustrating the different regions of the world from which tweets have been gathered for the duration between 10th July 2019 to 23rd July 2019. The geographical world map depicts the locations from which tweets have been fetched and the count of total number of tweets obtained is illustrated in line graph on day basis and on area graph on hourly basis for each day. It is interesting to see that, Australia is also among the locations from where tweets have been retrieved for 13days duration.

5.4.4 Dashboard 4

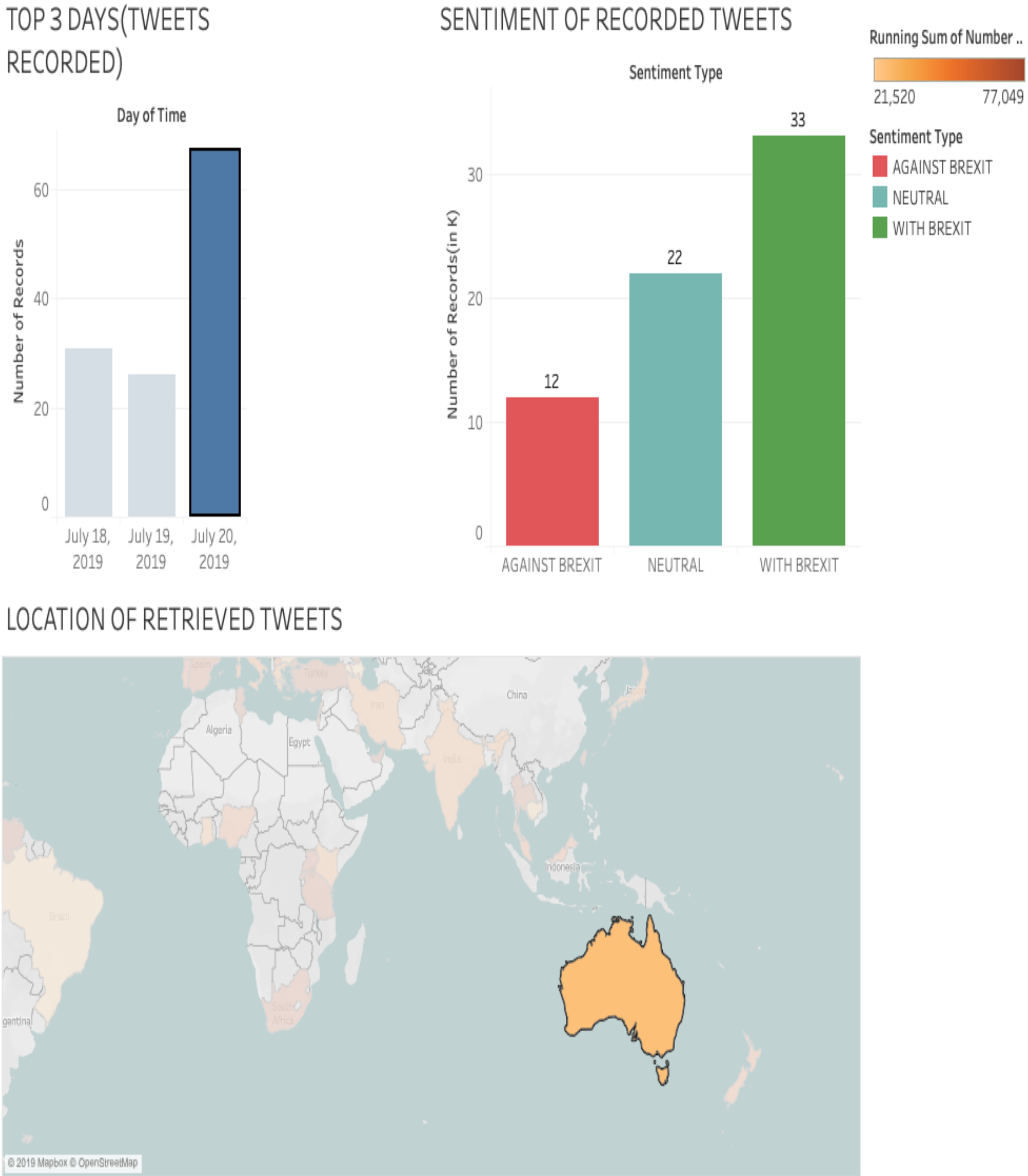

Figure 16: Top 3 days Dashboard with location and sentiment types.

The dashboard 4 in figure 16 depicts a drill down approach based on top 3 days(in tweets count) on which tweets have been fetched. It consists of graphs belonging to count of tweets based on top 3 days, Overall count of sentiment type of the tweets and geographical world map. So, far it is observed that Australia was also among the locations from where tweets have been fetched regarding BREXIT for duration of 10th July 2019 to 23rd July 2019. Therefore, when 20th July is selected in dashboard, it drills down to next step by showing that tweets from top 3 days (in terms of tweet count) also consists of proportion of tweets fetched from Australia. In the next step when Australia is selected, it has provided statistics pertaining to number of tweets supporting BREXIT, Neutral in opinion and Against BREXIT through sentiment type graph. Therefore, by using filters in dashboard and selecting 20th July and Australia has helped in gaining important insights which can help the United Kingdom (UK) government to understand how reaction or sentiments of general public in Australia were when protest against BREXIT was carried out in central London on 20th July 2019. Similarly, insights for sentiment reaction of public in Australia or other important countries can be gained for 18th July 2019 when Conservative Party member Jeremy Hunt held a program on BBC radio for clearing doubts about BREXIT among general public. All such insights are helpful for UK government to formulate their strategy for making future internal and foreign policies.

6 Evaluation

In order to evaluate performance of both the approaches the results were compared. Accuracy of both the models have been considered as main comparison criteria.

6.1 Accuracy of Naive Bayes Classifier Model

The accuracy of Naive Bayes Classifier Model is calculated using Confusion Matrix. As stated by Simon (2010) a confusion matrix helps in understanding the parameters which assists in testing stability of a machine learning model. A typical example of a confusion matrix is as shown in figure 17.

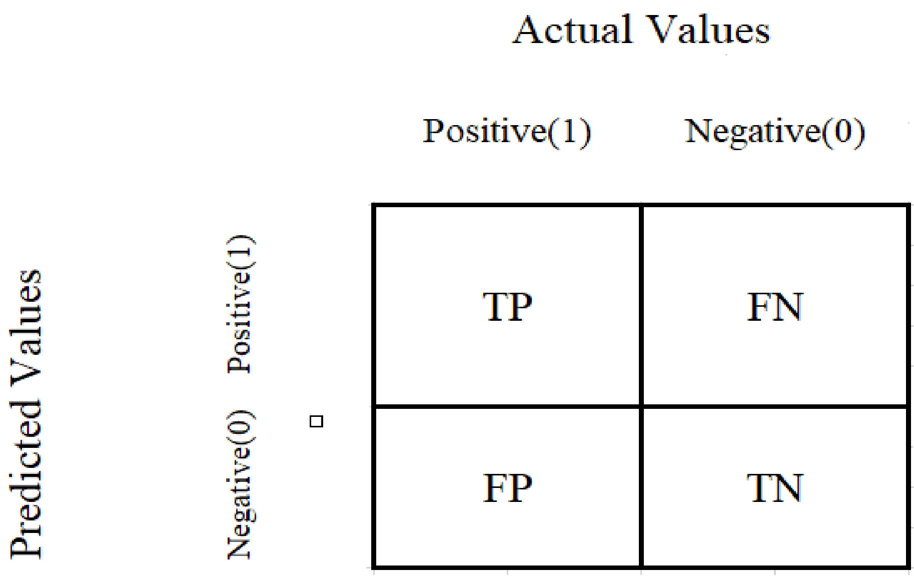

Figure 17: Confusion Matrix

As shown in above figure, there are four components used for calculating the parameters required to evaluate a machine learning model's performance.

- **True Positive Rate (TP):** When predicted values are predicted correctly then it is called as True Positive Rate. For example: - Predicting England will win the football match and it actually won it.

- **False Positive Rate (FP):** When the prediction made turns out to be wrong then it is known as False Positive Rate. For example: - Predicting England will win the match but it lost it.
- **False Negative Rate (FN):** When a negative prediction is made, and it actually turns out to be negative then its called as False Negative Rate. For example: - Predicting England will not win the football match but it actually won it.
- **True Negative Rate (TN):** When the prediction made is negative in respect to the actual values then it is termed as True Negative. For example: Predicting England will not win the football match and it actually lost.

Based on these components of confusion matrix, following parameters are calculated to evaluate a machine learning models performance.

- **Accuracy:** Accuracy of a machine learning model is calculated as below.

$$\boldsymbol{Accuracy{=}(TP{+}TN)/(TP{+}TN{+}FP{+}FN)} \tag{2}$$

- **Precision:** Precision is used for calculating accuracy of a positive class. It measures likelihood for prediction of classes which are positive. It is calculated as below

$$\boldsymbol{Precision{=}(TP/(TP{+}FP)} \tag{3}$$

- **Sensitivity (Recall):** Sensitivity or recall is defined as proportion of accurately classified positive classes. It also tells how a model behaves for a positive class. It is calculated as below.

$$\boldsymbol{Recall{=}(TP/(TP{+}FN)} \tag{4}$$

- **F1 Score:** F1 score helps in keeping a balance among precision and recall. It is calculated as below

$$\boldsymbol{F1\ Score{=}\ 2{*}(Precision{*}Recall/Precision{+}Recall)} \tag{5}$$

The confusion matrix calculated is shown in figure 18. From figure 18, it is clear that

```
[7]  # save best model to current working directory
     joblib.dump(grid, "twitter_sentiment.pkl")
     # load from file and predict using the best configs found in the CV step
     model_NB = joblib.load("twitter_sentiment.pkl" )
     # get predictions from best model above
     y_preds = model_NB.predict(X_test)
     print('accuracy score: ',accuracy_score(y_test, y_preds))
     print('\n')
     print('confusion matrix: \n',confusion_matrix(y_test,y_preds))
     print('\n')
     print(classification_report(y_test, y_preds))

     accuracy score:  0.73865

     confusion matrix:
      [[7587 2528]
      [2699 7186]]

                   precision    recall  f1-score   support

                0       0.74      0.75      0.74     10115
                4       0.74      0.73      0.73      9885

         accuracy                           0.74     20000
        macro avg       0.74      0.74      0.74     20000
     weighted avg       0.74      0.74      0.74     20000
```

Figure 18: Confusion Matrix for Naive Bayes Classifier Model

Naive Bayes Classifier Model achieved Accuracy of 74% accuracy with 74%, 75%,74% scores of precision, recall and F1 Score respectively.

6.2 Accuracy of TextBlob Approach

The accuracy calculation for Textblob approach is as shown in figure 19. Though, Textblob approach is part of natural language processing but it does not come under machine learning. Therefore, in the absence of confusion matrice, the accuracy for this approach is calculated using formulae.
Accuracy= Total number of Correct results/Total number of records.

TEXTBlob Accuracy

```
[8] from textblob import TextBlob

    blob_res = data['text'].apply(lambda x: TextBlob(re.sub('(@[^ ]+|#[^ ]+|\\n|http[^ ]+|[^\w ])','',x.lower()).strip()).sentiment.polarity>0)

[9] sum((data['labels']==4) == blob_res)/len(blob_res)

0.61458
```

Figure 19: Accuracy Calculation for TextBlob Approach

6.3 Comparison Between Naive Bayes Classifier Model and TextBlob Approach

As depicted in table 2, it is observed that Multinomial Naive Bayes Classifier Model(74%) achieved higher accuracy over TextBlob Approach(61%).

Table 2: Accuracy Comparison

Sr.no.	Model	Accuracy
1.	Naive Bayes Classifier	74%
2.	TextBlob	61.45%

6.4 Discussion

As illustrated in the evaluation section, the performance of Naive Bayes Classifier algorithm and Python's TextBlob approach have been compared on the basis of accuracy of both the models. It is clear from the obtained results that Naive Bayes Classifier(74% accuracy) algorithm provides an accurate model for text analysis as compared to python's TextBlob library(61% accuracy. Moreover, an important point to be noted is that TextBlob approach kept tweets with neutral sentiment polarity in neurtal classification category along with the tweets which it was unable to classify. Therefore, the neutral classification category had some additional tweets. Also, Naive Bayes Classifier algorithm works on the basis of classifying sentences into positive and negative classification categories. Therefore, this nature of Naive Bayes Classifier algorithm had an impact on the tweets with potential neutral classification. If both these issues observed in the respective techniques be resolved then the performance of the proposed model can be enhanced. Moreover, it is still a challange to deal with dual meaning sentences or slangs. If this issue is overcomed then the overall performance of both the approaches will have a significant boost. Also, as illustrated by Rana and Singh (2016), incorporation of other techniques such as Support Vector Machines(SVM) can help in building a better performing model by overcoming issues faced in Naive Bayes Classifier model. As SVM's inherits capability of classifying text into 'positive','negative' and 'neutral' categories. Therefore, SVM with better performance accuracy has potential to classify neutral sentences as well and thus overcoming the issue of Naive Bayes model not classifying neutral tweets. Ibrahim and Yusoff (2017) has also provided knowledge regarding selecting the appropriate dataset size for training the model. while implementing this reserach project 50,000 tweets were used from sentiment 140 dataset to train the naive bayes model. This dataset size was considered by keeping in mind the available computational resources. Therefore, with more available computational resources a bigger dataset size can also be incorporated in order to train the naive bayes model which can further provide better accuracy.Matharasi1 (2017) has helped in using K fold cross validation while training naive bayes classifier model. However, incorporation of character n gram approach along with it can enhance results. Also, during implementation TextBlob approach was able to handle miss spelled words as shown by by Manushree et al. (2017) in their work. S.Vijayarani and R.Janani (2016) showed the importance of tokenization which was being followed in this research project as well. The works from Chapman and Stolee (2016), Ganesh and Artzi (2012) and Yeole and Meshram (2011) were crucial for learning pre processing of tweets using python's regular expression library.

7 Conclusion and Future Work

The research question basically centers around determining the insights from the impacted regions of the world due to events related to BREXIT by utilizing Naive Bayes classifier algorithm, technology: Natural language processing (NLP), Python NLTK library, Python's TextBlob NLP library. Furthermore, the work focussed on the Twitter opinion mining which bifurcates the tweets based on three categories: positive, negative and neutral. This work can help organizations, associations or any governing body to centre about the opinion of the user about a rule or product. The sentiment scores have been obtained using Naive Bayes Classification algorithm and using TextBlob library of python, which were later compared in terms of accuracy. it was inferred that the Naive Bayes algorithm provided a strikingly contrasting accuracy of 74% as compared to 61.45% of TextBlob. The revelations of this investigation will support the British and Irish governments to refine their strategic plans in order to alleviate any repercussion of the BREXIT decision on their economies and lives of their inhabitants. As examined before, the significant restrictions of this research are the load it will add about while recovering the huge amount of data from twitter and cleaning it all concurrently. To alleviate this issue, other Artificial Intelligence approach like N-gram approach can be utilized to improve the general proficiency of the model. Additionally, the model is not able to divide the polarity and identify the words correctly as positive or negative tweets because of the double meaning words, these issues can likewise be worked upon in future to improve the productivity of the model. One point to be added for future work is to understand the N-gram approach and apply it to other languages such as Arabic, Latin, Mandarin, among others, as the tweets are tweeted in various languages, so in order to provide a detailed analysis on the type of language, i.e. making the model understand the language and then dividing it on the basis of positive and negative frontier, will be a commendable step towards future work. Besides this, the Google BERT is a new algorithm, can be used to provide higher accuracy for sentiment analysis. .

References

BBC (2019a). Anti-brexit protesters hold 'no to boris' march.

BBC (2019b). Representation in united kingdom.

BBC (2019c). Uk will have to face consequences of no deal.

Chapman and Stolee (2016). Exploring regular expression usage and context in python, *Proceedings of the 25th International Symposium on Software Testing and Analysis - ISSTA 2016* .

El Rahman, S. A., AlOtaibi, F. A. and AlShehri, W. A. (2019). Sentiment analysis of twitter data, *2019 International Conference on Computer and Information Sciences (ICCIS)* .

Ganesh and Artzi (2012). Hampi: A solver for word equations over strings, regular expressions and context-free grammars.

Garg, G. . (2015). Developing secured biometric payments model using tokenization, *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)* .

Goebel, V. (2014). Knowledge discovery in databases (kdd) and data mining (dm).

Goyvaerts, J. (2006). Regular expressions: The complete tutorial.

Ibrahim, M. N. M. and Yusoff, M. Z. M. (2017). The impact of different training data set on the accuracy of sentiment classification of nave bayes technique, *2017 IEEE Conference on Open Systems (ICOS)* **3**.

Ikoro, V., Sharmina, M., Malik, K. and Batista-Navarro, R. (2018). Analyzing sentiments expressed on twitter by uk energy company consumers, *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)* .

Jettakul, Thamjarat, Liaowongphuthorn, Udomcharoenchaikit, Vateekul and Boonkwan (2018). A comparative study on various deep learning techniques for thai nlp lexical and syntactic tasks on noisy data, *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)* .

Joachimsu, T. (1998). Text categorization with support vector machines: Learning with many relevant features.

Khatua, K. . (2016). Leave or remain? deciphering brexit deliberations on twitter, *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)* .

Lansdall, Dzogang and Cristianini (2017). Change-point analysis of the public mood in uk twitter during the brexit referendum, *In 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW 2016) Institute of Electrical and Electronics Engineers (IEEE)* .

Lobur, M., Romanyuk, A. and Romanyshyn, M. (2011). Using nltk for educational and scientific purposes, *CADSM2011, 23-25 February, 2011, Polyana-Svalyava (Zakarpattya), UKRAINE* .

Luong, M.-T. and Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains.

Ma, S., Xu Sun1, ., Wang, Y. and Lin, J. (2018). Bag-of-words as target for neural machine translation.

Manushree, Adarsh and Kumar (2017). A comparative method for different aspect based products features in online reviews of different languages, *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)* .

Matharasi1, S. (2017). Sentiment analysis of twitter data using nave bayes with unigram approach, *2International Journal of Scientific and Research Publications, Volume 7, Issue 5, May 2017 337 ISSN 2250-3153* .

Osborn, W. and Barry, J. (2016). Brexit: Implications for the it industry., pp. 36–37.

Pal, Dash and Saha (2015). An innovative lemmatization technique for bangla nouns by using longest suffix stripping methodology in decreasing order, *2015 International Conference on Computing and Network Communications (CoCoNet)* .

Park, C. and Seo, D. R. (2018). Sentiment analysis of twitter corpus related to artificial intelligence assistants, *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)* .

Permatasari, Fauzi, Adikara and Sari (2018). Twitter sentiment analysis of movie reviews using ensemble features based nave bayes, *2018 International Conference on Sustainable Information Engineering and Technology (SIET)* .

Ramanathan, V. and Meyyappan, T. (2019). Twitter text mining for sentiment analysis on peoples feedback about oman tourism., *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)* .

Rana, S. and Singh, A. (2016). Comparative analysis of sentiment orientation using svm and naive bayes techniques, *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)* **3**.

Sarkar, K. (2018). Using character n-gram features and multinomial naive bayes for sentiment polarity detection in bengali tweets, *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)* .

Sentiment140 (2019). Sentiment140 training dataset: Kaggle.com, *Home for Data Kaggle: Your Science.* .

Shah, Agarwal, Dubey and Correia (2018). Twitter analysis for disaster management, *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)* .

Shobana, Vigneshwara and Maniraj (2018). Twitter sentimental analysis, *International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878* **7**.

Simon, D. (2010). Analytic confusion matrix bounds for fault detection and isolation using a sum-of-squared-residuals approach, *IEEE Transactions on Reliability* .

Spishak and Dietl (2012). A type system for regular expressions, *FTfJP '12 Proceedings of the 14th Workshop on Formal Techniques for Java-like Programs* .

S.Vijayarani, D. and R.Janani, M. (2016). Text mining: Open source tokenization tools an analysis, *Advanced Computational Intelligence: An International Journal (ACII)* **3**.

Vasiliu, McDermott and Fernandez (2016). In or out? real-time monitoring of brexit sentiment on twitter.

Yeole and Meshram (2011). Analysis of different technique for detection of sql injection, *Proceedings of the International Conference Workshop on Emerging Trends in Technology - ICWET '11* .

Zhang, Y., Jin, R. and Zhou, Z.-H. (2010). Understanding bag-of-words model: A statistical framework.

Zitnik, S., Draskovic, D., Nikolic, B. and Bajec, M. (2017). nutie a modern open source natural language processing toolkit. 2017 25th telecommunication forum (telfor).