2019

Sentiment analysis for TV show popularity prediction: case of Nation Media Group's NTV

Joshua M. Mutisya
Faculty of Information Technology (FIT)
Strathmore University

Sentiment Analysis for TV Show Popularity Prediction: Case of Nation Media Group's NTV

Joshua Mutinda Mutisya

Submitted in partial fulfilment of the requirements for the Degree of Master of Science in Information Technology at Strathmore University

Faculty of Information Technology

Strathmore University

Nairobi, Kenya

June, 2019

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

..…………….. [Name of Candidate]

……………….. [Signature]

……………….. [Date]

Approval

The thesis of Mutisya Joshua Mutinda was reviewed and approved by the following:

Dr Joseph Onderi-Orero, PhD

Senior Lecturer, Faculty of Information Technology

Strathmore University

Dr Joseph Onderi-Orero, PhD

Dean, Faculty of Information Technology

Strathmore University

Professor Ruth Kiraka

Dean, School of Graduate Studies

Strathmore University

Abstract

Media-houses play a vital role in informing the masses on the issues that matter. They are also a source of entertainment for many households. In Kenya, the public depends on media largely for entertainment and educational purposes. However, many media-houses find it difficult to make decisions on what the viewers actually wish to watch. This makes the media-houses to be in the dark, unaware of what viewers want and making decisions based on perceptions rather than data.

Most of the analytic tools used by media-houses in Kenya are used to provide insights on website-related activities such as site visits, number of article reads and read-depths. This is not a good way of measuring popularity and does not create a true reflection of how an audience perceives a given product.

In this study, we propose a predictive model that uses sentiment analysis to determine the popularity of a given TV show. This enables accurate decisions to be made based on the viewership trends over a specific period of time. Natural Language Processing is used to perform sentiment analysis on tweets derived from Twitter.

This solution involved tweets being derived from the social site Twitter through the use of the Twitter API. Once fetched, the tweets had their polarity measured through the use of a lexicon dictionary in order to remove neutral tweets. These tweets were then be labelled as either positive or negative using the Support Vector Machine classifier. Then the overall popularity score of a movie was calculated. The solution was able to not only show the polarity of derived tweets, but also assign overall popularity scores showing how positive or negative a TV show is.

Table of Contents

Declaration ii

Abstract iii

Table of Contents iv

List of Figures viii

List of Equations ix

List of Tables x

Acknowledgement xi

Dedication xii

Chapter 1 : Introduction 1

1.1 Background 1

1.2 Problem Statement 2

1.3 Objectives 3

1.3.1 Main Objective 3

1.3.2 Specific Objectives 3

1.4 Research Questions 3

1.5 Justification 4

1.6 Scope and Limitation 4

Chapter 2 : Literature Review 5

2.1 Introduction 5

2.2 The State of the Media in Kenya 5

2.3 Related works 7

2.3.1: Data Mining by Integrating TV ratings with Multimedia Contents 7

2.3.2 Diary-based Popularity Measurement 7

2.3.3 Forecasting Video access Patterns 9

2.3.4 Broadcast TV Industries 9

2.3.5 Sentiment analysis and Presidential Elections in the United States 10

2.3.6 Social Media Analytics for Network Television Ratings 11

2.3.7 Sentiment analysis for social media data 12

2.3.8 Prediction of Popular Tweets using Similarity Learning 13

2.3.9 Prediction of the 2017 French Election based on Twitter Analysis 14

2.3.10 Hate Speech Detection on Social Media 15

2.3.11 Prediction of Movie success using Twitter Sentiment Analysis 15

2.4 Sentiment Analysis 16

2.4.1 Supervised learning 18

2.4.2 Lexicon based approach 21

2.5 Twitter Data 23

2.6 Conceptual Framework 23

Chapter 3 : Research Methodology 25

3.1 Introduction 25

3.2 Research Design 25

3.3 Location of the Study 25

3.4 Target Population 25

3.5 Sampling 25

3.6 Data Collection 26

3.6.1 Datasets from Twitter API 26

3.6.2 Python programming 26

3.7 System Development Methodology 26

3.8 Research Quality 27

3.9 Ethical Considerations 29

Chapter 4 : System Design and Architecture 30

4.1 Introduction 30

4.2 Requirement Analysis 30

4.2.1 Functional Requirements 30

4.2.2 Non-Functional requirements 30

4.3 Use Case Diagram 31

4.3.1 Detailed use case description 32

4.4 System Architecture 34

4.5 Context Diagram 35

4.6 Sequence Diagram 36

4.7 Data flow diagram 37

Chapter 5 : System Implementation and Testing 39

5.1 Introduction 39

5.2 Designing the corpus 39

5.3 Text pre-processing 40

5.4 Feature extraction 42

5.5 Determining positive and negative tweets 42

5.6 Determining a TV show's overall score 44

5.7 Training the SVM model 44

5.8 Testing the model 45

Chapter 6 : Discussions 46

6.1 Experiments on Sentiment analysis 46

6.1.1 Using different classifiers 46

6.1.2 Experiment 1: SVM with different feature types 50

6.1.3 Experiment 2: Random Forest with different feature types 51

6.1.4 Experiment 3: KNN with different feature types..51

6.1.5 Experiment 4: Naïve Bayes with different feature types.................................52

Chapter 7 : Conclusion and recommendations...53

7.1 Conclusion...53

7.2 Recommendations 53

7.3 Future work 53

References..55

Appendices..62

List of Figures

Figure 2.1: Top 5 TV stations by viewership 5
Figure 2.2: Favourite TV programmes for viewers 6
Figure 2.3: Using Multimedia Contents 7
Figure 2.4: How Nielsen Determines TV Ratings 8
Figure 2.5: Overview of the Broadcast TV Programme Popularity Prediction 10
Figure 2.6: Flow chart used to gauge candidates' popularity 11
Figure 2.7: Proposed Framework for Social Media Analytics 12
Figure 2.8: Deep Learning on Data from Twitter 13
Figure 2.9: Framework of text classification 14
Figure 2.10: The process of sentiment analysis 17
Figure 2.11: Support Vector Machine 18
Figure 2.12: Artificial neural network model 21
Figure 2.13 Sentiment analysis using dictionary-based approach 22
Figure 2.14: Conceptual framework 24
Figure 3.1: Rapid Application Development Model 27
Figure 4.1: Use Case diagram...31
Figure 4.2: System Architecture 35
Figure 4.3: Context diagram 36
Figure 4.4: Sequence diagram...37
Figure 4.5: Level 0 Data flow diagram 38
Figure 5.1: Collecting TV Show related tweets from Twitter 39
Figure 5.2: Sample tweet pulled using the Twitter API...40
Figure 5.3: Sample of processed tweets...40
Figure 5.4: Code for cleaning tweets 41
Figure 5.5: Annotated tweets for supervised learning 42
Figure 5.6: VADER code snippet deriving the sentiment value of a text........................44
Figure 6.1: SVM ROC curve 47
Figure 6.2: Random Forest ROC curve...48
Figure 6.3: Naive Bayes ROC curve...49
Figure 6.4: KNN ROC curve 50

List of Equations

Equation 2.1 Similarity learning formula..14
Equation 2.2 Popularity formula for the French elections case 15
Equation 2.3 Linear function to build SVM hyperplane..18
Equation 2.4: Assigning input vector to the positive or negative class 19
Equation 2.5: Final determining formula for the decision surface..................................19
Equation 2.6: Calculating the posterior probability 19
Equation 2.7: Calculating the $P(d|c_i)$...20
Equation 2.8: Calculating the $P(c_i)$ 20
Equation 2.9: Calculating the $P(d_j|c_i)$..20
Equation 3.1: Accuracy rate formula 28
Equation 3.2: Error level formula 28
Equation 3.3: True positive rate...28
Equation 3.4: False positive rate 29
Equation 3.5: Precision 29
Equation 5.1: Calculating overall popularity score...44

List of Tables

Table 3.1: Confusion matrix 28

Table 5.1: The tweet labelling categories 41

Table 5.2 Performance of VADER against other lexicon dictionaries 43

Table 5.3 Confusion matrix for the model 45

Table 5.4: Values for the confusion matrix variables 45

Table 6.1: SVM performance 46

Table 6.2: Random Forest performance 47

Table 6.3: Naive Bayes performance 48

Table 6.4: K-Nearest neighbour performance 49

Table 6.5: Performance of classifiers 50

Table 6.6: Performance of SVM 51

Table 6.7: Performance of Random Forest 51

Table 6.8: Performance on K-nearest neighbour 51

Table 6.9: Performance on Naive Bayes 52

Acknowledgement

I would like to thank God Almighty for enabling me to successfully complete this research. My sincere appreciation also goes to my supervisor Dr Joseph Orero for his endless support and guidance, as well as all the lecturers who gave feedback during presentations which made this work better.

I wish to also thank Ms Misiko Andere, the Programmes Manager at NTV Kenya for providing insights on the processes of gauging the performance of a TV show to the station’s audience.

Dedication

To my dearest parents, Harrison Kiamba and Veronicah Mbau, and my sisters Linda Munee and Nancy Mwikali, thank you for your endless prayers and unyielding support.

Chapter 1 : Introduction

1.1 Background

The media landscape in the world as well as in Kenya is ever evolving, with new TV channels and products being unveiled to tap into the available audience (Kamdar, 2016). This is gradually moving the media market from being a blue ocean with lots of opportunities for many, to a red ocean, prompting media houses to become innovative enough to stay afloat. This has resulted in many media-houses retrenching their workers with the hope of remaining afloat (Simons, 2017).

However, insufficient consideration is made on the source of the drop in viewership income, since there are very few data insights on the matter. In Kenya, the situation is made worse due to few Cable TV services, which make use of information of pay-tv subscribers. The current number of cable TV subscriptions stands at 79,938 subscriptions (Kenya National Bureau of Statistics, 2018) Therefore, it is difficult to detect any changes in the viewers' perceptions and interests.

The machine-learning techniques, although not new in the technological landscape, have not been fully tapped into not only in Kenya but in the world in the media sector especially the TV segment. In the new era of media integration with technology, acquiring insights about audience's perspective towards a TV show also provides insights about which marketing strategy to approach (Miller, 2017). This would secure and perhaps enhance the marketing income generated by media-houses. However, few companies such as Netflix have made good use of artificial intelligence to determine which shows are more popular than others, resulting in the success of huge shows such as the political drama House of Cards (Vawdrey, 2015).

Currently, Kenyan media-houses are yet to deploy a concrete data-driven approach to make such decisions as indicated above. In as much as cable TV is not common, information on audience behaviour towards a given TV show or character could still be acquired through use of sentiment analysis from social media datasets. There are currently 4.3 million set top boxes in Kenya as at 2017 (Kenya National Bureau of Statistics, 2018). There is need to address the missing link in the audience behaviour that is yet to be captured. Thus, there is few evidence of data being used to check audience perspectives regarding a given TV show.

1.2 Problem Statement

The media industry is currently facing challenges that have never been faced before. Companies are trying to balance between reducing costs and maximising profits while at the same time keeping up with changes in technology (Cavanillas & Wahlster, 2015)

In an effort to determine the performance and viewership perception towards TV shows, mediahouses are find it an uphill task. Misiko Andere, the Programmes Manager at Nation Media Group's NTV explains in an interview that the process of determining the popularity of a given TV show entails use of focus groups selected by the research agency of choice. First, the programming team watches the selected TV show, then proceeds to giving the focus groups the opportunity to do the same.

These focus groups are selected on the basis of the target audience. For instance, if a TV show is targeting a teenage audience, a focus group with teenagers is collected. With the Nation Media Group not having much influence about the quality of focus groups provided, the results from the groups about the popularity of a given show could be prone to subjectivity. The same focus group is then given a call after an episode to provide feedback about how a programme performed. Another approach used is the free-of-charge SMS service where feedback can be sent to the station.

There are various media reports depicting the turbulent times the Kenyan media-houses are facing, in an effort to remain in business. In 2018, the Nation Media Group announced its plans to retrench about 140 employees in what they termed as 'a restructuring process' (Consumer Federation of Kenya, 2018). This notwithstanding, there is little use of big data insights about TV viewership within the media-house. Tapping into the diverse insights provided by artificial intelligence would ensure media houses make the right. This results in challenges making the right business decisions regarding programming and making money through advertisements.

In the case of the Nation Media Group (NMG), sentiment analysis is yet to be fully exploited to provide insights on viewership behaviour. This results in decisions being made on the basis of perception. This however, should not be the case. With the ability to mine data from various sources including social media at relatively low costs, machine learning algorithmic models can be designed to predict viewers' behaviour (Tsuchiya, 2014).

The development of a predictive model to gauge the popularity of TV shows using sentiment analysis is hereby proposed. This would go a long way in filling the knowledge gap that is present due to the absence of viewership data, resulting in surveys which are subject to bias (FluidSurveys, 2013). Currently, there is no available repository for viewership data at NMG, thus a better alternative for data sourcing would be online social media platforms e.g. Twitter.

1.3 Objectives

1.3.1 Main Objective

The purpose of the research was to design a prediction model that gauges the popularity of TV shows for effective decision-making in the TV department. This would be determined through sentiment analysis on data derived through the Twitter Search API.

1.3.2 Specific Objectives

i. To investigate the problems associated with the current methods used by media houses in gauging the popularity of a show.

ii. To evaluate algorithms used in data mining and prediction of TV show's popularity.

iii. To design a model that determines a show's popularity based on sentiment analysis on Twitter.

iv. To test the TV show popularity predictive model.

1.4 Research Questions

i. What are the problems encountered with the current methods used by media houses in gauging popularity of a show?

ii. Which algorithms are used in data mining and predicting popularity of TV shows?

iii. How a predictive model is developed based on sentiment analysis on Twitter?

iv. Which tests are carried out to validate the TV show popularity predictive model?

1.5 Justification

Currently, there is minimal use of data in making decisions regarding audience behaviour and attitude towards various TV products, casting doubt on the accuracy of such decisions. This is because such decisions lack any factual backing and are not sustainable. They do not provide the decision-makers with any predictive analysis to aid in making timely decisions. Designing a data model that provides insights based on algorithmic sentiment analysis would go a long way in ensuring accurate and credible decisions for the betterment of their audience and also an improvement in their revenues.

1.6 Scope and Limitation

The project was limited to making decisions based on user sentiments expressed as tweets posted on Twitter about a TV show. One assumption made in this research is that the popularity of a show is solely dependent on the sentiments captured in the tweets posted. There are many other factors that affect the popularity of a show such as the characters or any other external factor that could have an impact on the views. The analysis was also limited to only English tweets.

Chapter 2 : Literature Review

2.1 Introduction

The purpose of this chapter is to review any relevant literature and studies carried out in studying TV audience behaviour. The various approaches taken in different scenarios to determine popularity using sentiment analysis are reviewed in this chapter, as well as the different machine learning algorithms used in classifying text.

2.2 The State of the Media in Kenya

The media industry in Kenya is a thriving market, In the TV station field, Citizen TV leads with the highest share of viewers in Kenya. Other stations are KTN (11 percent), KTN News (9 percent) and NTV (8.5 percent) closing the top four watched TV stations in Kenya (GeoPoll, 2018). A 2017 report by PriceWaterhouseCoopers (PwC) places the TV entertainment and media market at USD865 million. It had also been estimated that the expenditure would cross USD3 billion by 2018 (PwC, 2013).

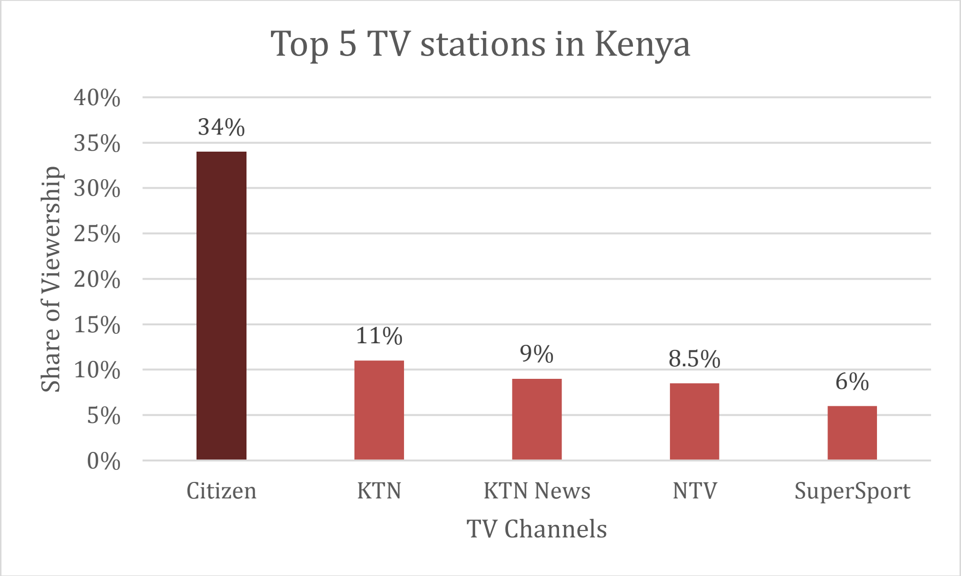


Figure 2.1: Top 5 TV stations by viewership (GeoPoll, 2018)

The media in Kenya is very vibrant, ranking at 96 in 180 countries in the World Press Freedom Index (Media Council of Kenya, 2018). According to a study by the Kenya Film Classification

Board (KFCB), about four in five respondents frequently watching TV at their homes. (Kenya Film Classification Board, 2018).

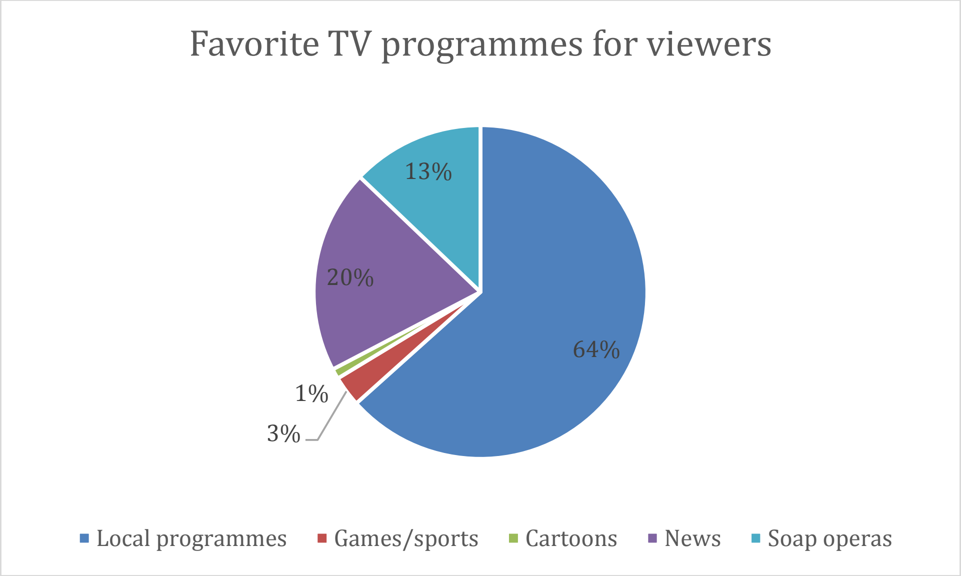


Figure 2.2: Favourite TV programmes for viewers (GeoPoll, 2018)

The survey also states that 64 percent of TV viewers prefer local programmes, whereas 20 percent prefer news. It is followed by soap operas (13 percent), games/sports (3 percent) and cartoon (one percent). The respondents also identified that Machachari was the most popular show on Citizen TV, whereas on NTV it was Churchill Show. For KTN, it was Mshamba, while KBC had Vioja Mahakamani.

2.3 Related works

2.3.1: Data Mining by Integrating TV ratings with Multimedia Contents

In Japan, TV ratings are provided by Video Research Ltd, an agency that specializes in TV ratings which started audience measurement in 1962. Here, the researchers proposed the use of various concepts of multimedia in order to identify how viewers relate with the video and speech content of a given TV show. The TV ratings were acquired from Video Research, an agency that specializes in TV ratings. By using a systematic random sampling, a total of 6,600 households were surveyed in Kanto, Kansai and Nagoya, Japan (Shin'ichi & Hinami, 2017).

The ratings were based on samples from the households selected. Using this data, they are able to detect an audience's behaviour with its TV sets through what they refer to as 'Boundary of a TV show'. This refers to a user tuning into a station at the start and off at the end. Another factor considered is what the researchers termed as 'Transition'. This refers to the act of switching from one channel to another perhaps due to lack of interest. The model used is as shown below.

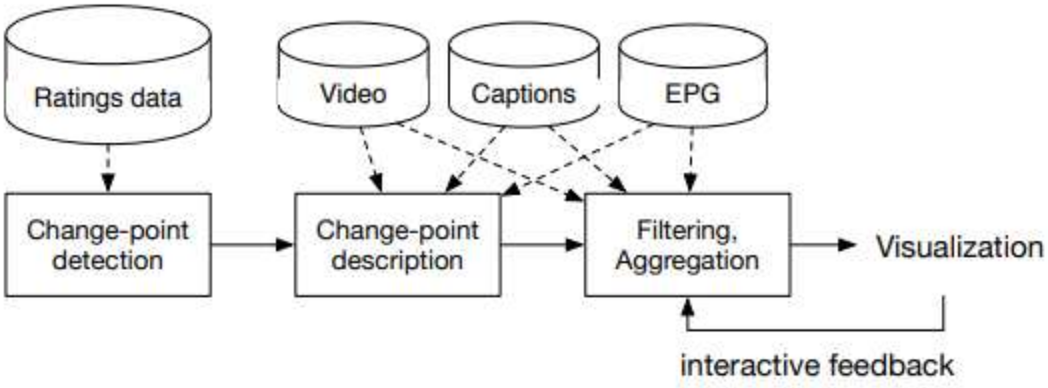


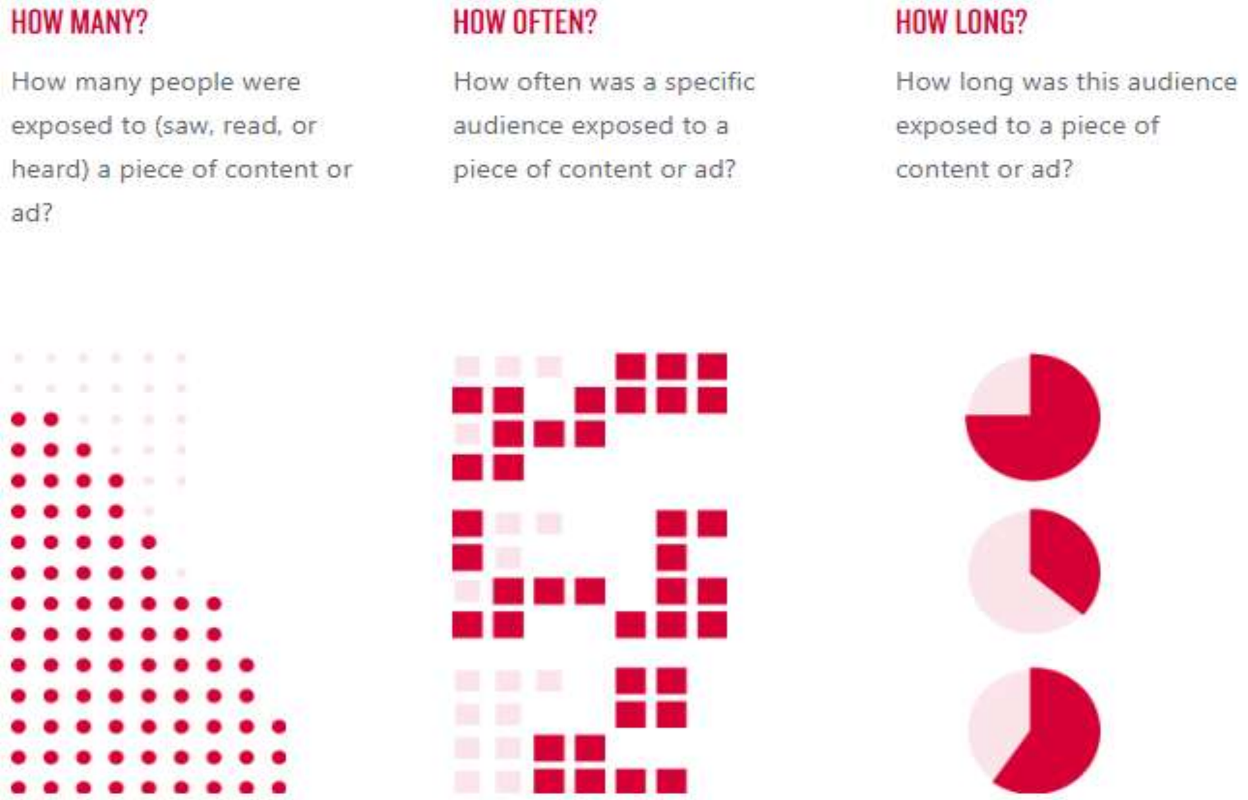
Figure 2.3: Using Multimedia Contents (Shin'ichi & Hinami, 2017)

As illustrated above, the ratings data would provide data on the point in a show which viewers change their tuning to another show. Then, other insightful data would be collected from multimedia elements such as video and captions. The final product would be aggregated to provide a visualization which can be analysed.

2.3.2 Diary-based Popularity Measurement

The Nielsen Company provides TV companies with insights about how the audience perceives of the TV programmes. A set of households are provided with a diary and urged to record their viewing habits for a given period of time. Once the insights are picked, a programme is allocated a share of the audience. Thus, for instance, if 10 percent of the households said they watch Programme X, then a conclusion is made that the programme's household rating is 10 percent. The analysis only makes use of households, not head counts. Thus, the analysis provides an aggregate figure, thus implying that the share would be higher than the rating (Halbrooks, 2018).

Another crucial means of measurement used by Nielsen is the use of people meters. These are devices that capture special sounds called watermarks, that are found in the audio belonging to the specific TV programs and advertising segments that are being measured. With each program having a unique watermark, which is also inaudible to the human eye. Nielsen is able to track the programme being watched, how often and the duration it took to watch it (Nielsen, 2019).



HOW MANY?

How many people were exposed to (saw, read, or heard) a piece of content or ad?

HOW OFTEN?

How often was a specific audience exposed to a piece of content or ad?

HOW LONG?

How long was this audience exposed to a piece of content or ad?

Figure 2.4: How Nielsen Determines TV Ratings (Nielsen, 2019)

2.3.3 Forecasting Video access Patterns

Big data plays an important role in a company’s decision-making and making forecasts about the various aspects of the business entity. A good example of how data is used to make forecasts is describing and forecasting video access patterns (Hassani & Silva, 2015).

The study carried out on the YouTube online sharing platform analysed a set of datasets to check and predict which videos would be accessed more in the near future. They categorised the qualities into rarely accessed and frequently accessed videos. For frequently accessed videos, they generated components that summarised popular videos and using these components, they could predict future video views for individuals. For rarely-accessed videos, a clustering method was used to classify and generate popularity bursts (Gursun et al., 2010).

2.3.4 Broadcast TV Industries

In this method, there are three crucial steps taken. One, patterns of the popularity are monitored and the Dynamic Time Warping calculated. Secondly, using the Random Forest Regression, the data is grouped into four predictive models. Finally, to obtain the final analysis, a Gradient Boost decision Tree, which produces decision trees, is used to generate the final results of the analysis done (Zhu et al., 2015).

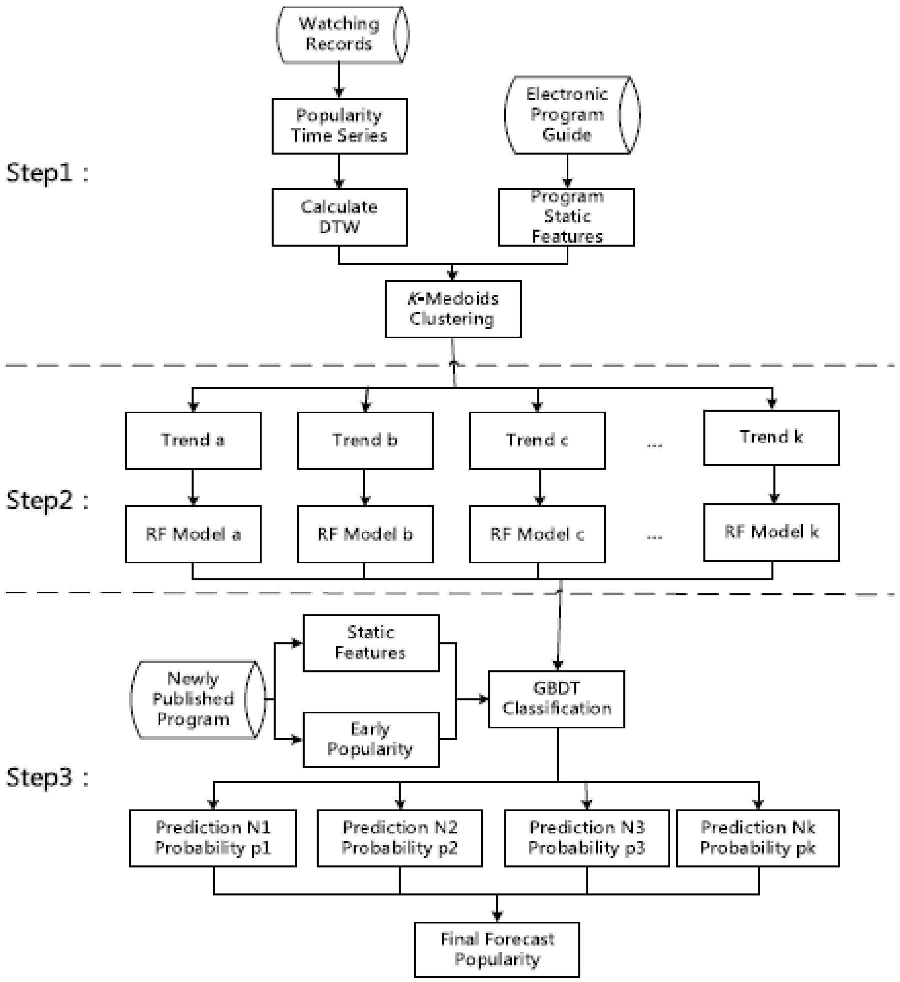


Figure 2.5: Overview of the Broadcast TV Programme Popularity Prediction (Zhu et al., 2015)

2.3.5 Sentiment analysis and Presidential Elections in the United States

In this case, (Wicaksono et al., 2017) propose a system where the winner of the US presidential election would be predicted by analysing data from social media. In the study, focus was given to the popularity of 2012 presidential candidates; Republican Mitt Romney and Democrat Barack Obama. This procedure was broken out into pre-processing, analysing the data gathered, and visualizing the findings.

To perform the sentiment analysis, the binary multinomial naïve Bayes classifier was used on the data gathered from Twitter, assigning polarity scores for each tweet. Corpus was also used in this study to aid in the analysis. For purposes of this study, the pre-processing and sentiment analysis was borrowed from this study.

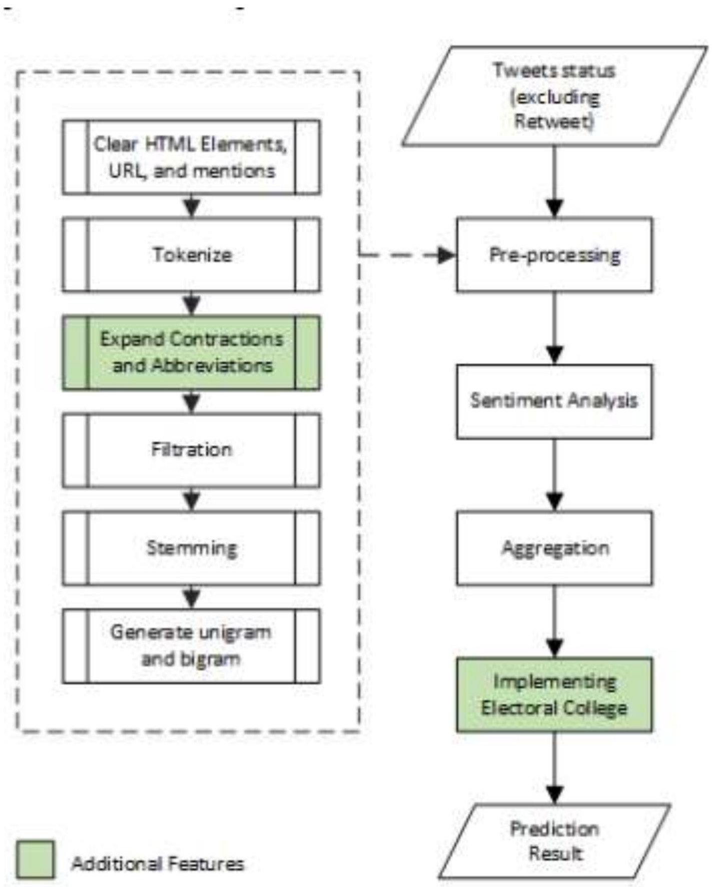

Tweets status (excluding Retweet)
Pre-processing
Sentiment Analysis
Aggregation
Implementing Electoral College
Prediction Result
Clear HTML Elements, URL, and mentions
Tokenize
Expand Contractions and Abbreviations
Filtration
Stemming
Generate unigram and bigram
🟩 Additional Features

Figure 2.6: Flow chart used to gauge candidates' popularity (Wicaksono et al., 2017)

2.3.6 Social Media Analytics for Network Television Ratings

In this study, social media data for 38 programmes was collected for a period of five weeks. Here, what was extracted was the number of tweets, followings, followers, Facebook likes and mentions of each program (Oh et al., 2015). The key purpose of the assignment was to identify the relationship between posts by the official account of a programme and its rating performance. Here, an analytics framework was proposed, which entails collecting data, cleaning data, extracting features and carrying out analysis.

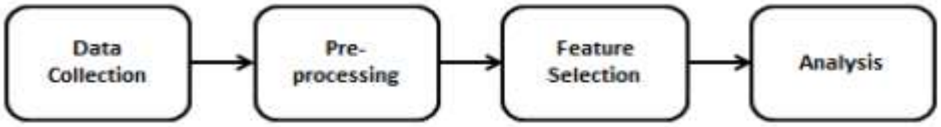


Figure 2.7: Proposed Framework for Social Media Analytics (Oh et al., 2015)

The data collected, which entailed the collection of tweets from the official accounts of the programme, number of followers and related TV programmes, analysis using Ordinary Least OLS regression model was carried out. The challenge with this approach in the Kenyan context is that most TV shows do not have their own social media accounts, and if they do, they do not commission large following. It would be wrong to gauge popularity based on only the state or existence of a TV show's social media account.

2.3.7 Sentiment analysis for social media data

In this study, (Ramadhani & Goo, 2017) intended to analyse data. Here, the first step is text mining, whereby 3 steps are involved:

i. Retrieving information from social media.
ii. Performing partial analysis and identifying any relevant data points.
iii. Digging into the data to retrieve insights.

A dataset of about 4,000 tweets was used on a neural network which entailed a hundred neurons, 3-layer architecture as well as a stochastic descent. The layers provide a step-by-step method of clearing words, sentences and finally dictating the popularity.

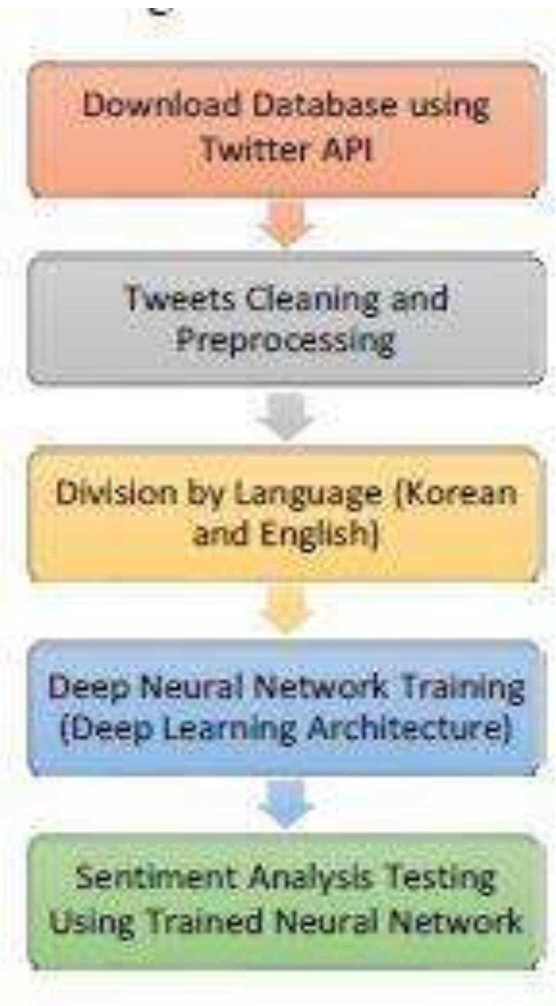

Download Database using Twitter API
Tweets Cleaning and Preprocessing
Division by Language (Korean and English)
Deep Neural Network Training (Deep Learning Architecture)
Sentiment Analysis Testing Using Trained Neural Network

Figure 2.8: Deep Learning on Data from Twitter (Ramadhani & Goo, 2017)

2.3.8 Prediction of Popular Tweets using Similarity Learning

The study suggests the use of retweets to determine the popularity of a tweet. In this case, if the number of retweets (*retweet_count)*) a tweet gets is zero, then that tweet is deemed unpopular. If a tweet receives more than zero retweets, then the tweet is said to be popular (Ahmed et al., 2013).

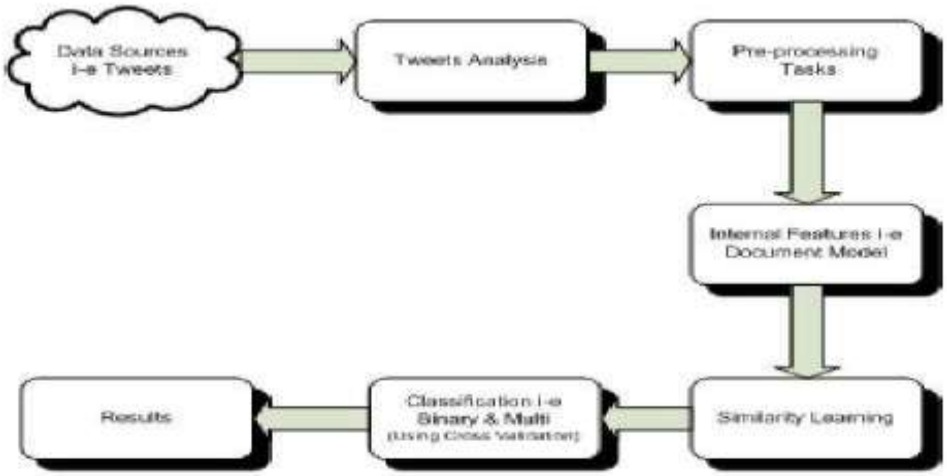

Data Sources i-e Tweets
Tweets Analysis
Pre-processing Tasks
Internal Features i-e Document Model
Similarity Learning
Classification i-e Binary & Multi (Using Cross Validation)
Results

Figure 2.9: Framework of text classification (Ahmed et al., 2013)

Similarity learning algorithm operates by bringing close elements that share certain aspects in common, which are referred to as target neighbours, and setting aside those with different aspects, referred to as impostors. The formula used for analysis is as shown below;

$$s_A(x,y) = \frac{x^T A y}{N(x,y)}$$

Equation 2.1 Similarity learning formula

As indicated by the formula above, *N(x, y)* refers to the normalization, whereas A refers to the similarity matrix. This approach cannot be applied when gauging popularity of TV shows since

2.3.9 Prediction of the 2017 French Election based on Twitter Analysis

Tweets based on the 2017 French election were collected through the Twitter API and analysed according to ‘good’, ‘bad’ or ‘neutral’. The more the good tweets attributed to a candidate, the more popular he/she is. Little concern is given to neutral networks, which can be restrictive for a candidate who enjoys a strong following from a small crowd (Wang & Gan, 2017). The proposed method to determine popularity is as shown below;

$$popularity(a) = \left[\frac{pos(a)}{pos(a)+neg(a)}\right]\left[\frac{N(a)}{N(a)+N(b)}\right]$$

Equation 2.2 Popularity formula for the French elections case

As indicated by the formula above, *pos(a)* refers to the number of positive tweet related to a presidential candidate *a*, *n(b)* refers to negative tweets about candidate *b*, whereas *n(a)* refers to negative tweets about candidate *a*.

2.3.10 Hate Speech Detection on Social Media

In this research, (Mugambi, 2017) proposes the use of unigram features and term frequency inverse document frequency (TF-IDF) weighting to label and train a model that could determine which tweet posted on Twitter was hate speech or on-hate speech. First, tweets were gathered using the Twitter API. The tweets were then represented with the TF-IDF weighting and then applied to the SVM classifier.

2.3.11 Prediction of Movie success using Twitter Sentiment Analysis

In this study, the popularity of movies is determined by carrying out sentiment analysis on tweets related to the movie title. Eight movies were predicted. The metrics used were hit, flop or average. Since the number of tweets used to perform the study were many thus required lots of labour to manually label them, 200 tweets were selected on a random basis (Jain, 2013).

To provide the success score, a PT-NT ratio was developed. For starters PT is defined as the total number of positive tweets whereas NT was defined as the number of negative tweets. Therefore, the PT-NT ratio was defined as the total positive tweets divide by the total number of negative tweets.

In addition to this, a threshold was set. For a movie to qualify as a hit, its PT-NT ratio needed to be more than or equal to 5. On the other hand, for it to qualify as average, the same ratio needed to be less than five but more than 1.5, whereas less than 1.5 was considered a flop (Jain, 2013). The prediction of the popularity was done by first training the Naïve Bayes classifier through the use of the NLTK toolkit, while the feature used was word count.

2.4 Sentiment Analysis

A sentiment refers to an attitude, an emotion or a feeling held by a person towards another person or object. Sentiment analysis, therefore, refers to the process of automatically obtaining attitudes and emotions from speech, text, and database by using Natural Language Processing (Kharde & Sonawane, 2016). This is achieved by grouping the attitudes as either “negative”, “positive” or “neutral”.

A sentiment can be represented mathematically as (o, f, so, h, t). Each of these attributes is interpreted below:

o = object, which refers to a given entity such as TV show, person etc.

f = feature of the object, which points to a specific attribute of the object

so = polarity of the sentiment, whether positive, negative or neutral

h = holder of the opinion, referring to the entity that is providing the sentiment

t = time when the opinion was expressed

A lot of studies have been done on opinion analysis. One by (Pak & Paroubek, 2010) proposed an approach where tweets are classified as positive, negative and objective (neutral). A twitter corpus was created by fetching tweets using the Twitter API and annotating by using emoticons.

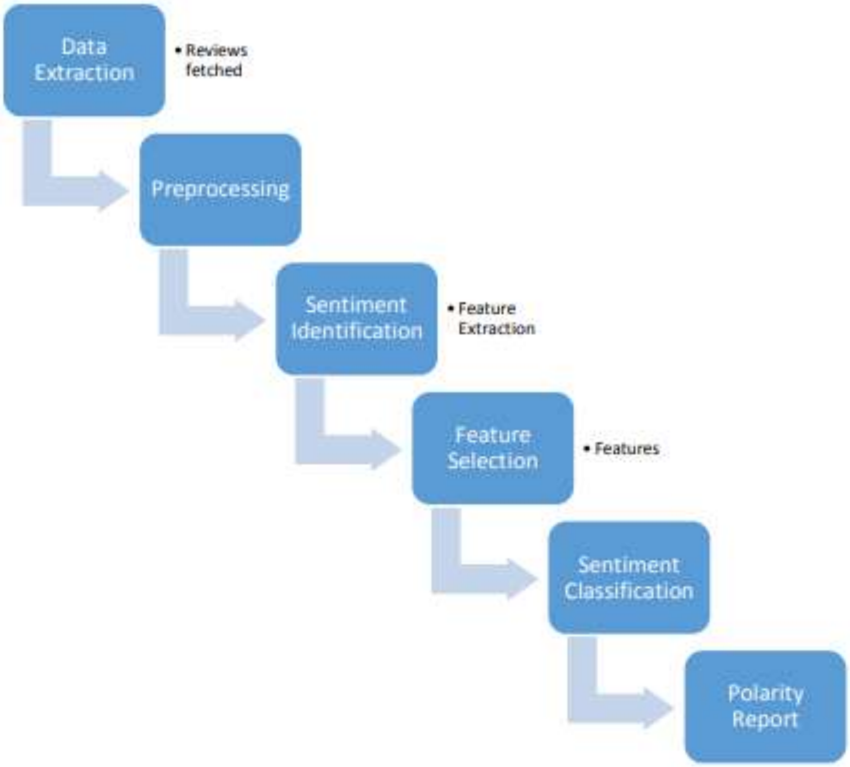


Figure 2.10: The process of sentiment analysis ((Rajput & Solanki, 2016))

In the first step, identified as data extraction, the data is selected from various sources or by using an application programming interface (API).

In the second step, called text preprocessing, data cleaning is carried out which entails the following steps:

a) Eradicating non-English words – this is because most lexicon dictionaries have sentimental English words.

b) Removing uniform resource locators, punctuation marks and hashtags – this goes a long way in reducing the noise in the corpus.

c) Removing extra letters from words – this is for the purpose of ensuring that the final word is one found in the lexicon e.g. *merrryyyyyy* reduced to *merry*.

d) Removing stop words - these are words that have very little significance to the message of a statement e.g. ‘an’, ‘a’ or ‘the’.

e) Stemming – this entails ensuring that only the root part of the word is left e.g. *waking* is recorded as *wake.*

2.4.1 Supervised learning

2.4.1.1 Support Vector Machines

Support Vector Machines (SVM) is one of the techniques under the supervised learning and plays a crucial role on various data mining operations, and are known to be powerful enough to apply real world scenarios such as face recognition and text categorisation (Nayak et al., 2015). A SVM is a hyperplane *h* that separates the negative instances from positive ones. Results from research shows that SVM and naïve Bayes have the highest accuracy in opinion mining.

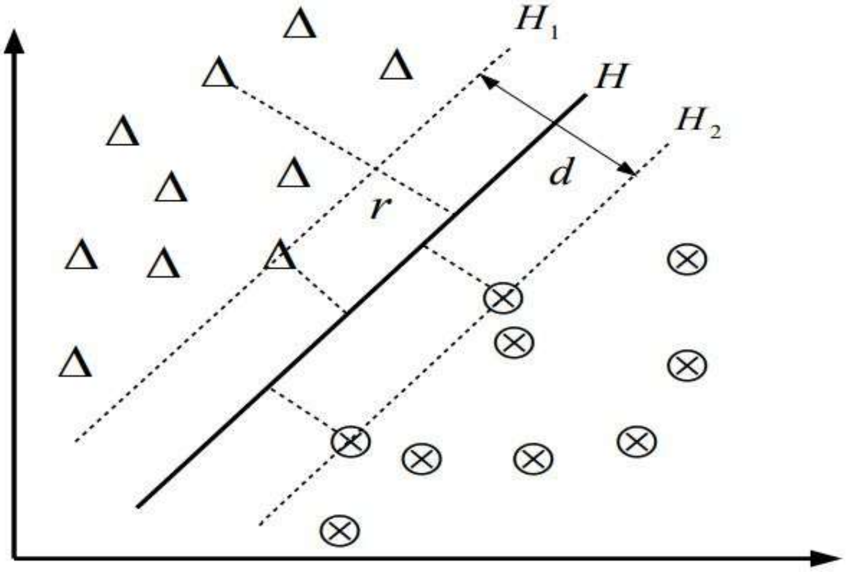


Figure 2.11: Support Vector Machine (Nayak et al., 2015)

As shown in Figure 2.11, SVM builds two-class classifiers. Let the set of training examples *D* be

$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$,

Where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a r-dimensional input vector in a real-valued space and denotes the positive class as 1, and the negative class as -1 (Liu, 2011). In order to build a hyperplane, we build a linear function of the form

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b$$

Equation 2.3 Linear function to build SVM hyperplane

An input vector x_i is assigned to the positive or negative class I as shown below

$$y_i = \begin{cases} 1 & \text{if} \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 0 \\ -1 & \text{if} \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b < 0 \end{cases}$$

Equation 2.4: Assigning input vector to the positive or negative class

This implies that SVM determines the hyperplane shown below called the decision surface.

$$\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$$

Equation 2.5: Final determining formula for the decision surface

In a research that involved analysis of Twitter data, (Zainuddin & Selamat, 2014) used term-weighting schemes to extract the needed features for the classifier. The TF-IDF, binary occurrences and term occurrences are tested to determine which one delivered the best results after being submitted in to the SVM classifier. The TF-IDF provided the best results in feature-weighting i.e. extracting the most important features to provide an input to the classifier.

2.4.1.2 Naïve Bayes

It is a probabilistic classifier and a set of supervised learning approaches employed in classification and clustering. Therefore, given a document d and a set of classes {}, the posterior probability that the document belongs to either of the classes is calculated and the document is then assigned to the class with the highest value of probability (Bai & Nie, 2004)

According to (Liu, 2011) the posterior probability is calculated by applying the equation below

$$P(c_i \mid d) = \frac{P(d \mid c_i)P(c_i)}{P(d)}$$

Equation 2.6: Calculating the posterior probability

By assuming conditional independence among words in a class, then *P(d|c$_i$)* can be solved as shown below

$$P(d \mid c_i) = \prod_{j=1}^{m} P(d_j \mid c_i)$$

Equation 2.7: Calculating the *P(d|c_i)*

In Equation 2.2, $P(c_i)$ can be determined as the share of training sets in class c_i.

$$P(c_i) = \frac{N_i}{N}$$

Equation 2.8: Calculating the $P(c_i)$

As shown above, N refers to the total number of training documents and N_i is the number of training documents found in the class c_i. So, $P(d_j|c_i)$ is calculated as

$$P(d_j \mid c_i) = \frac{1 + count(d_j, c_i)}{|V| + N_i}$$

Equation 2.9: Calculating the $P(d_j|c_i)$

Naïve Bayes combines ease of use with efficiency with accuracy, and also does a good job with numerical and textual data (Swamy & Hanumanthappa, 2013) . It is used in various areas such as disease diagnosis, filtering of spam emails in client servers as well as in image recognition.

2.4.1.3 Artificial Neural Network

This is a collection of neurons which act on provided input to provide desired output. This type of learning is most suitable for systems whose inputs change rapidly (MathWorks, 2016). ANNs are able to undergo learning by altering the weights associated with each node. There are two ANN topologies namely Feedforward and Feedback. The Feedforward topology does not have any feedback loops and is mostly preferable for pattern recognition, whereas the feedback topology loops its feedbacks.

(Hsieh et al., 2013) carried out a study on the use of artificial neural networks to predict ratings of television content using social media content. In the study, a TV action drama that airs once a week was used for analysis. Data in terms of pages posts, comments on the posts, likes, shares and fan posts were used for the study.

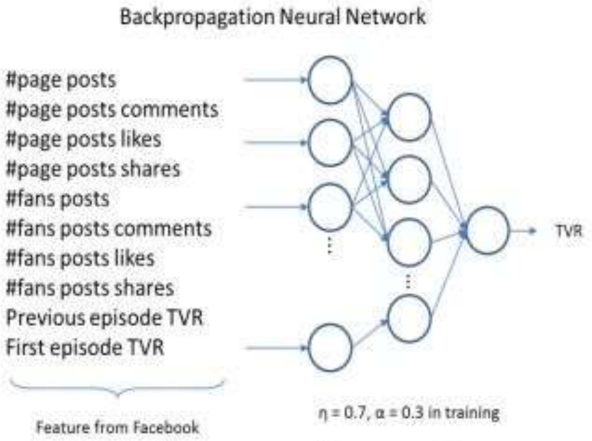


Figure 2.12: Artificial neural network model (Hsieh et al., 2013)

2.4.2 Lexicon based approach

The lexicon based approach is based on the notion that the contextual sentiment orientation is the sum of the individual sentiment values for each of the words (Palanisamy et al., 2013). An example of such library is the SentiwordNet, which labels sentiments as either positive or negative, scoring at the range of 0 to 1.There are two states involved; the desired, which basically depicts positive sentiments, and the undesired, which employs negative opinions.

Another form of library used is the AFINN – 111 which labels sentiments on the range of -5 to +5 as negative to positive. However, a major challenge is posed by the complexity that comes with natural languages, where some negation aspects could be missed. There are two approaches used in compiling the word list namely dictionary based and corpus based approaches.

In this approach, data collected is matched to a set of sentimental words in a dictionary and then determined as either positive or negative or neutral (Hardeniya & Borikar, 2016). Sentiment analysis is carried out in two steps namely subjectivity detection and polarity assignment. The dictionary-based approach first begins by carrying out the text pre-processing operation to remove features such as punctuation marks and stop words.

Then stemming is also done. There exists a threshold which is used to gauge the polarity of a sentiment. If the score exceeds the threshold, then the statement is considered to be positive, and negative if it fails to do so. If the variation from the threshold is zero, the statement is classified as neutral (Hardeniya & Borikar, 2016). One of the shortcomings of this approach is the inability to identify meaning of sentimental words based on the context they are used.

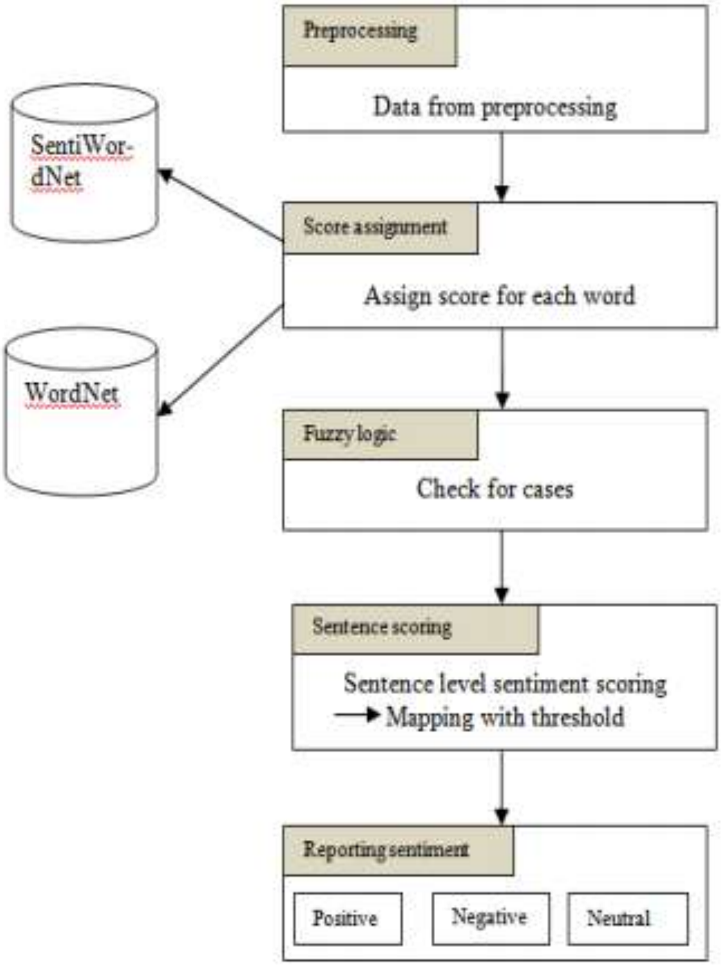


Figure 2.13 Sentiment analysis using dictionary-based approach (Hardeniya & Borikar, 2016)

2.5 Twitter Data

A study by (Asur & Huberman, 2010) showed that content on online communities can be useful when one wishes to make future predictions that do better than those in artificial markets such as opinion polls and surveys. This saves time and resources since one does not need to put in place market mechanisms.

Twitter, a microblog, was founded in 2006. According to the (Global Web Index, 2019), it has an estimated 326 million active users. In order to tap into the data provided by these users, Twitter provides a search API, with one able to retrieve tweets, even in real-time for free over the last seven days (Twitter, 2019).

A major drawback is that one is unable to retrieve older tweets beyond seven days unless they pay for the premium service.

2.6 Conceptual Framework

Below is the proposed conceptual framework to predict the popularity of a TV show. Tweets related to selected TV shows for analysis will be fetched from Twitter using the Twitter API. They will then undergo a process of text preprocessing whereby unnecessary symbols, punctuation marks and stop words will be removed. The new list is then taken through a lexicon dictionary to assign each of the tweets a polarity score of how positive, negative or neutral it is. Here, all the tweets with a neutral score are dropped. The positive and negative tweets then form a corpus to be used for learning. These tweets are then labelled as positive, negative or neutral. The training algorithm of choice is the SVM algorithm. The tweets are then be converted into token counts through the countvectorizer and then fed into the SVM algorithm to determine the popular and non-popular TV shows. These results will be placed in a repository, where the user can derive to view the popularity results.

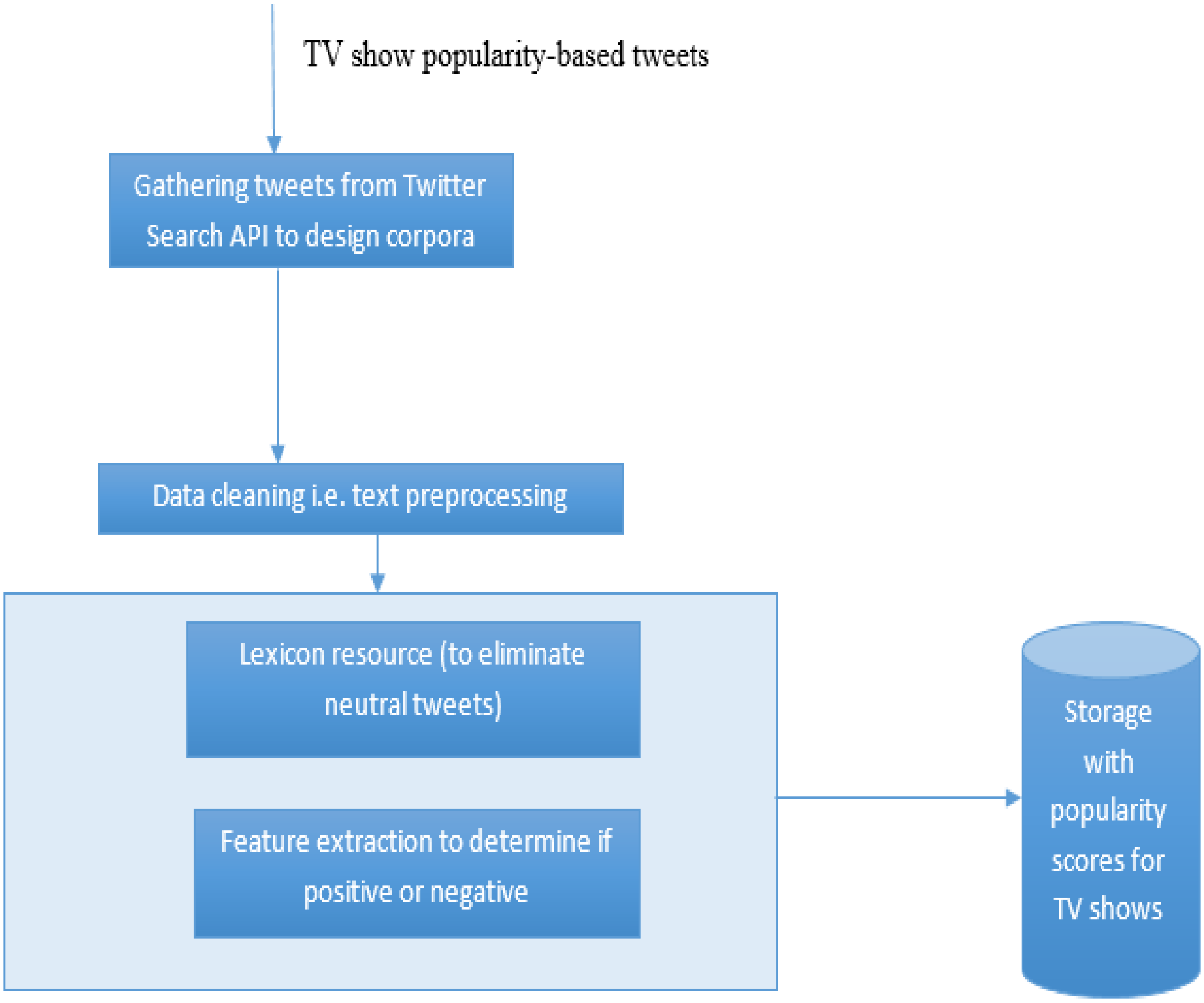

TV show popularity-based tweets
Gathering tweets from Twitter Search API to design corpora
Data cleaning i.e. text preprocessing
Lexicon resource (to eliminate neutral tweets)
Feature extraction to determine if positive or negative
Storage with popularity scores for TV shows

Figure 2.14: Conceptual framework

Chapter 3 : Research Methodology

3.1 Introduction

The purpose of the research methodology is to state the preferred way of doing the research. Research methodology refers to the procedural steps used to solve a problem, and figures out issues such as the methods of data collection, processing and presentation (Pitchai et al., 2013).

3.2 Research Design

This project took an experimental angle of research. This involves having a set of variables and manipulating them while checking the outcome. The main goal was to create predictive models to determine the popularity of TV shows, and determine which one is most accurate. The models were trained by using data derived from the Twitter API.

Using the Twitter Data API, tweets related to the TV shows were obtained. Thereafter, the selected tweets were cleaned and key attributes obtained. The classified positive, neutral and negative tweets were used to generate popularity scores for the popularity shows.

3.3 Location of the Study

The study was carried out in Nairobi, Kenya. Insights into how media-houses determine the choice of a TV show worthy to be aired were derived from the Nation Media Group's NTV television channel. The Nation Media Group is a media house with operations in print, broadcast and digital media, with audiences in Kenya, Uganda, Tanzania and Rwanda (Nation Media Group, 2018).

3.4 Target Population

The targeted population for this study was tweets related to TV shows aired. The sample is tweets made by Twitter users who have viewed selected TV shows across different Kenyan channels.

3.5 Sampling

Purposive sampling was used to select the sample to use. Purposive sampling, also known a Judgement sampling is whereby the researcher makes judgement on which sample best suits his/her needs (Sharma, 2017). The TV shows selected for analysis are only those which had a renowned twitter hashtag associated with them. The advantages of this sampling method are:

i) It is economical since it saves on time and resources.

ii) It avoids any unnecessary items from entering into the sample by chance.

iii) It gives better results if the investigator is unbiased and has the capacity of keen observation and sound judgment.

3.6 Data Collection

A semi-structured interview with the Programmes Manager at NTV Kenya, Ms Misiko Andere was useful in getting crucial insights on the topic under study, and highlighted the need to have such a predictive model as proposed by this research designed to help bridge the existing knowledge gap on TV shows popularity.

3.6.1 Datasets from Twitter API

The Twitter API was used to gather tweets based on the hashtags of the TV shows in the sample. Each of the TV shows selected has a specific hashtag that it has been assigned to by the management teams of the TV crew for purposes of enhancing audience engagement and getting viewership feedback. It is important to note that the information provided by Twitter API is used to enhance relevance, instead of completeness. This implies that there are some tweets that the API may fail to capture (Twitter Inc., 2018).

3.6.2 Python programming

Python is a programming language that enables programmers do various tasks including database interactions (Mueller, 2014). In this study, the programming language was used to retrieve tweets from the Twitter search API, carry out text pre-processing for the tweets fetched and carrying out the relevant machine learning operations through it provision of libraries.

3.7 System Development Methodology

The Rapid Application Development Model is the most preferred for this project. The model makes it possible for users and developers to freely experience the prototypes as they are being developed and make any recommendations for possible modifications (Ghahrai, 2017).

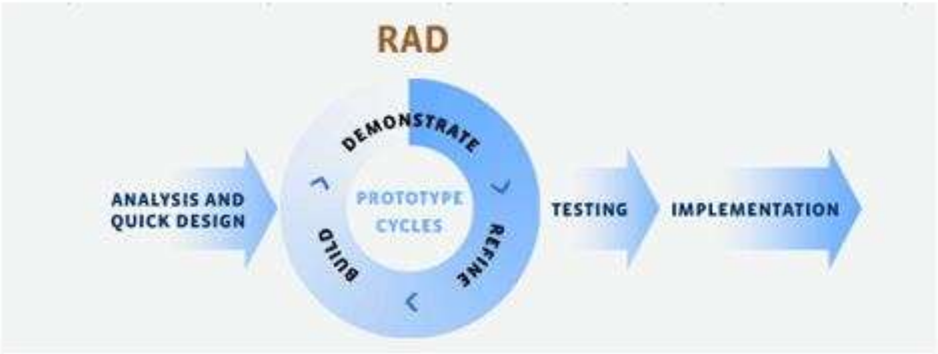


Figure 3.1: Rapid Application Development Model (Ghahrai, 2017)

The steps taken in this model are (Ghahrai, 2017):

i. Analysis and quick design - In this stage, the scope of the system was defined and the various functions which need to be addressed are highlighted.
ii. Prototyping cycles - This stage involved defining the process and data flows of the system through designing use-case, context and sequence diagrams and designing a conceptual framework.
iii. Testing - The designed prototype underwent a series of validation tests to verify if it was working and meeting the user’s expectations.
iv. Implementation-The developed prototype is rolled out and an iterative process of carrying out revisions can be made to make it a better product.

3.8 Research Quality

To gauge the quality of the research, the researcher will answer the following questions;

i. Did the research meet the set objectives?
ii. Can the research be applied in other related fields and still be useful?
iii. Did the methods of data collection provide accurate information?
iv. Are there any external factors that could have provided an alternative end?

There was no tweet without a sentiment to avoid any data gaps. For the purpose of testing the system, a confusion matrix was used. It entails information about real and predicted classifications done (Visa et al., 2011).

Table 3.1: Confusion matrix

		Actual class	
		Positive sentiment	Negative sentiment
Predicted class	Positive sentiment	*TN*	*FP*
	Negative sentiment	*FN*	*TP*

TN refers to the number of correct negative predictions

FP refers to the number of incorrect positive predictions

TP refers to the number of correct negative predictions

FN refers to the number of incorrect negative predictions

The formula for calculating the accuracy of the prediction model is as shown in Equation 3.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation 3.1: Accuracy rate formula

On the other hand, the formula for calculating the error level of the model is as shown in Equation 3.2.

$$Error = \frac{FP + FN}{FP + FN + TP + TN}$$

Equation 3.2: Error level formula

The share of positive sentiments which were identified and classified accurately (also identified as recall) is calculated using the equation:

$$True\ positive\ rate = \frac{TP}{TP + FP}$$

Equation 3.3: True positive rate

The share of negative sentiments which were incorrectly identified and classified as positive is calculated using the equation:

$$False\ positive\ rate = \frac{FP}{FP +\ TN}$$

Equation 3.4: False positive rate

For the share of positive sentiments which were correct, the precision value is calculated as

$$\text{Precision} = Precision = \frac{TP}{TP + FP}$$

Equation 3.5: Precision

3.9 Ethical Considerations

The purpose and the means to achieve the goals set in the proposal was moral and legal. There was no falsification or fabrication of data for purpose of bending the truth to fit a certain idea. Also, no work of another researcher or study was used without full or appropriate credits being provided. There was also no ghost-writing in this thesis, and the work submitted was that of the researcher and no one but the researcher.

Chapter 4 : System Design and Architecture

4.1 Introduction

System design refers to the process of identifying and defining the various elements (hardware, software, people and communication flows) that constitute the proposed system, as well as their functions (Dennis et al., 2012). System architecture on the other hand defines the overall structures of an information system and the various interaction paths across the system (Garlan, 2000).

For purposes of designing the systems, UML diagrams are used to illustrate the flow of processes and commands for the system as well as describe broadly what each component of the system does. Use case diagrams, sequence, context and data flow diagrams were used to provide more insights into this.

4.2 Requirement Analysis

The main objective of the research is to determine the popularity of TV shows using Twitter. The requirements to be met are grouped into functional and non-functional requirements.

4.2.1 Functional Requirements

i. The proposed system should enable users to enter TV show hashtags or names to be able to pick related tweets.
ii. The system can fetch tweets through the Search API based on the specific hashtags or names of TV programmes.
iii. The system should perform text pre-processing and putting them in a well tabulated comma-delimited format.
iv. The system should classify the cleaned tweets as either positive or negative.
v. The system should display to the user the popularity scores of the TV show.

4.2.2 Non-Functional requirements

4.2.2.1 Scalability

The proposed model should be able to handle as many tweets as possible without its functionality being compromised.

4.2.2.2 Usability

The proposed system should be as simple as possible for users to interact with and achieve its main objective.

4.2.2.3 Reliability

The proposed system should have as little breakdown episodes as possible, so that users can easily use the application without any interruptions. Also, its results should be accurate so as to efficiently aid in long-term managerial solutions.

4.3 Use Case Diagram

The use case diagram as shown below illustrates how the main actor will be interacting with the suggested system for determining TV show popularity.

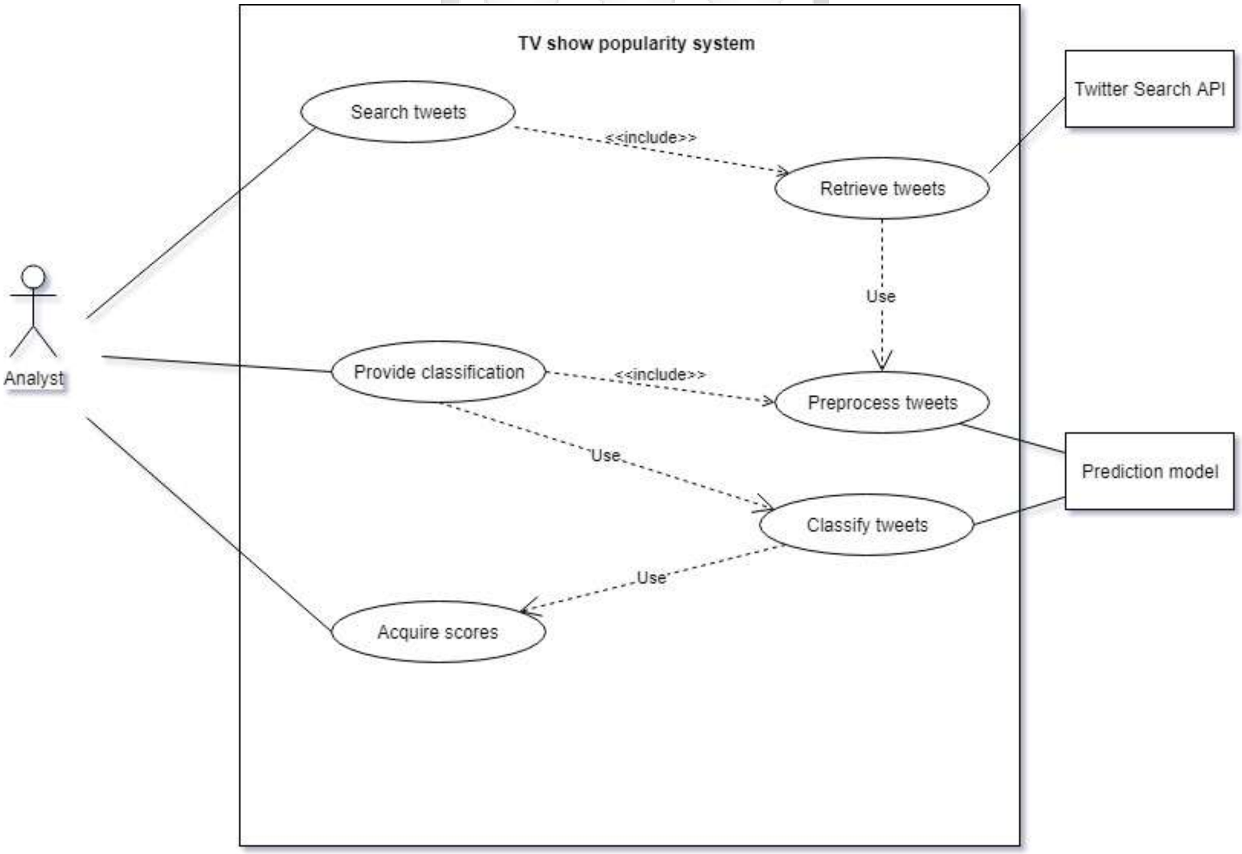


Figure 4.1: Use Case diagram

4.3.1 Detailed use case description

The detailed use case description provides a deeper description of the various processes illustrated in the use case diagram.

Use case: Search tweets, Retrieve tweets

Primary actor

Analyst

Twitter search API

Prediction model

Pre-conditions

Reliable internet access

Main success scenario

Actor	**System responsibility**
1. Actor enters the hashtag of the TV show under study	
	2. Pass the hashtag entered as a metric to be used to fetch tweets
	3. Retrieve tweets from the Twitter search API using the hashtag of the TV show
	4. Store the fetched tweets.
5. Actor views the gathered tweets	
	6. Undergo pre-processing on the collected tweets
	7. Annotate and classify the tweets as either positive or negative
	8.Provide the aggregated popularity score of the TV show
9. Receives the popularity score of the TV show	
10. Exit the system	

a. Scenario 2: Collecting TV show related tweets

Use case: Search tweets, Retrieve tweets

Primary actor

Twitter search API

Prediction model

Pre-conditions

The TV show whose tweets are being retrieved is pre-set as per the hashtags entered.

Actor **System responsibility**

1. Actor selects the hashtag for the TV show to be determined.
2. The system fetches the tweets related to the selected TV show hashtag for analysis, and then proceeds to classifying them as either positive (1) or negative (0) or neutral (-1)
3. The system then selects the calculated sentiments and use the model to determine the popularity score
4. The system then saves the popularity score for the TV show the fetched tweets.
5. Actor views the gathered tweets

b. Scenario 2: Predicting the popularity of a TV show

Use case: Search tweets, Retrieve tweets

Primary actor

Analyst

Twitter search API

Prediction model

Pre-conditions

The tweets related to a TV show have been fetched and put in a .csv file

Actor	**System responsibility**
1. Actor selects the hashtag for the TV show to be determined.	
	2. The system fetches the tweets related to the selected TV show hashtag for analysis, and then proceeds to classifying them as either positive (1) or negative (0) or neutral (-1)
	3. The system then selects the calculated sentiments and use the model to determine the popularity score
	4. The system then saves the popularity score for the TV show the fetched tweets.
5. Actor views the gathered tweets	

4.4 System Architecture

As illustrated by the system architecture below, the user, who is in this case is labelled as the analyst, enters the Twitter hashtag associated with a TV show, or the name of the show. The tweets collector, tasked with aiding in the process of deriving tweets, engages the Twitter Search API to gather the tweets associated with the hashtag.

After this, the data collected is then put in a structured format, and stored in a database. The cleaned tweets would then undergo text pre-processing to removing any abbreviations, punctuation marks and stop words. The tweets proceed to the SVM classifier to be marked as positive, negative or neutral, and an aggregate score provided.

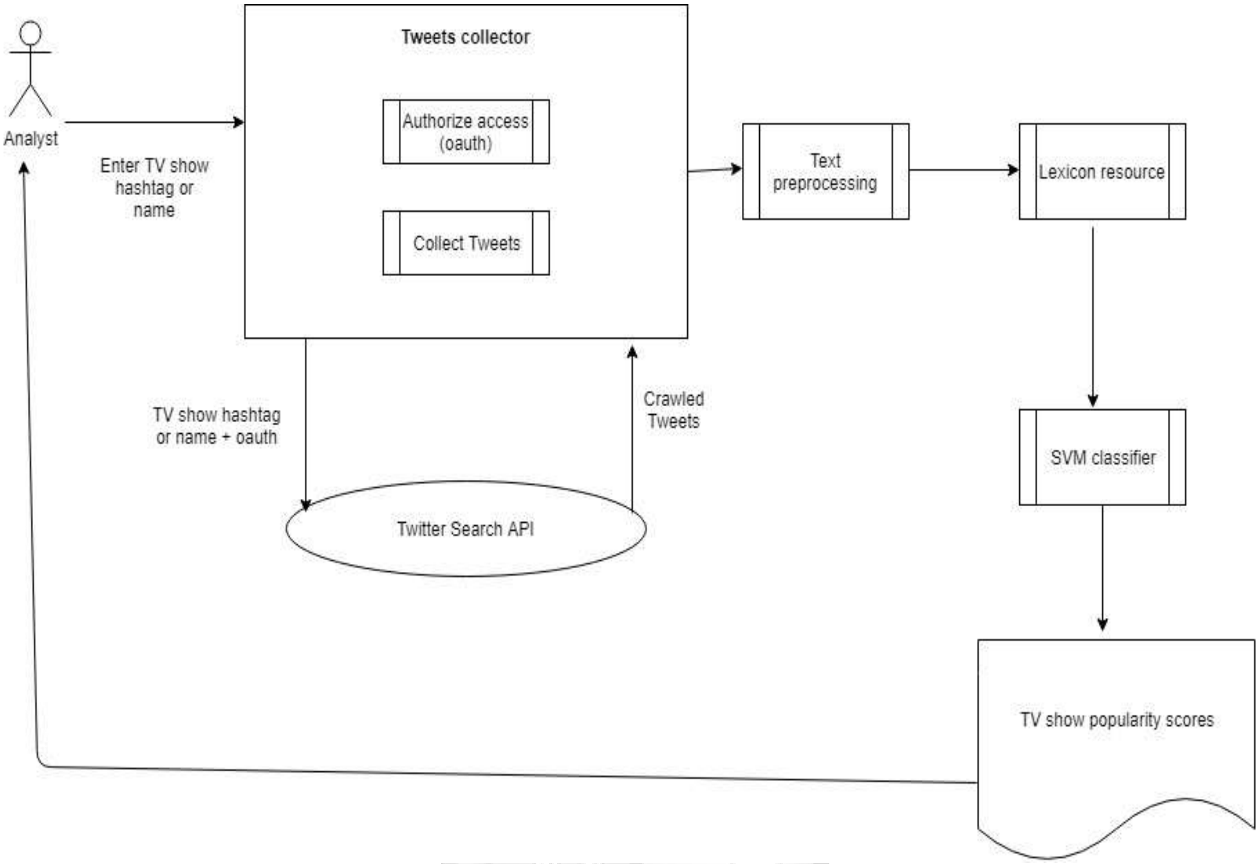


Figure 4.2: System Architecture

4.5 Context Diagram

The context diagram is used to show the environment in which a system operates, as well as the entities that interact with it. The input values and output results are also illustrated here. The analyst and the Twitter Search API are the major entities that interact with the prediction model. The analyst requests the popularity score of a given TV show, based on the hashtag and name of show. The popularity prediction model then goes ahead and sends a request to the Twitter Search API. On authorising access, tweets related to the TV show are fetched, and classified as positive or negative. The annotated tweets are then sent back to the analyst.

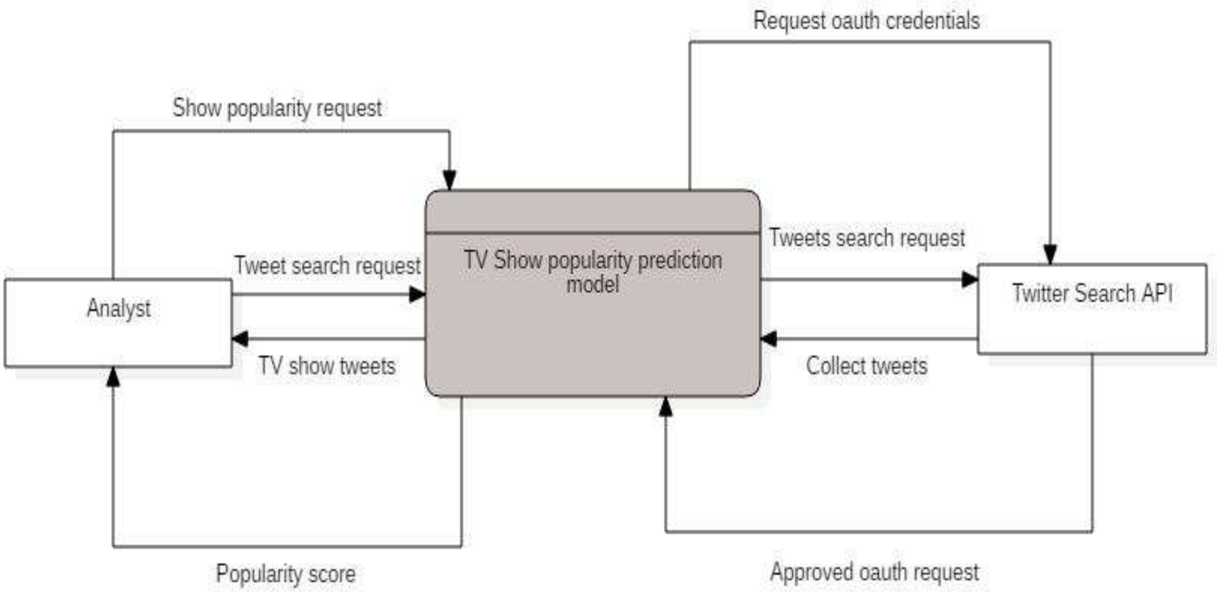


Figure 4.3: Context diagram

4.6 Sequence Diagram

As shown below, the context diagram illustrates how the user gets to interact with the system and also how the various parts of the system communicate with one another. The user interacts with the system through the user interface. The user enters the hashtag of the select TV show, which is then passed to the Twitter Search API for it to be used to search for the relevant tweets. The acquired tweets are then put in a .csv file, which the user orders for cleaning to be carried out. The function clean_tweets() calls the preprocessor to get rid of any abbreviations, stop words etc. After the data cleaning,

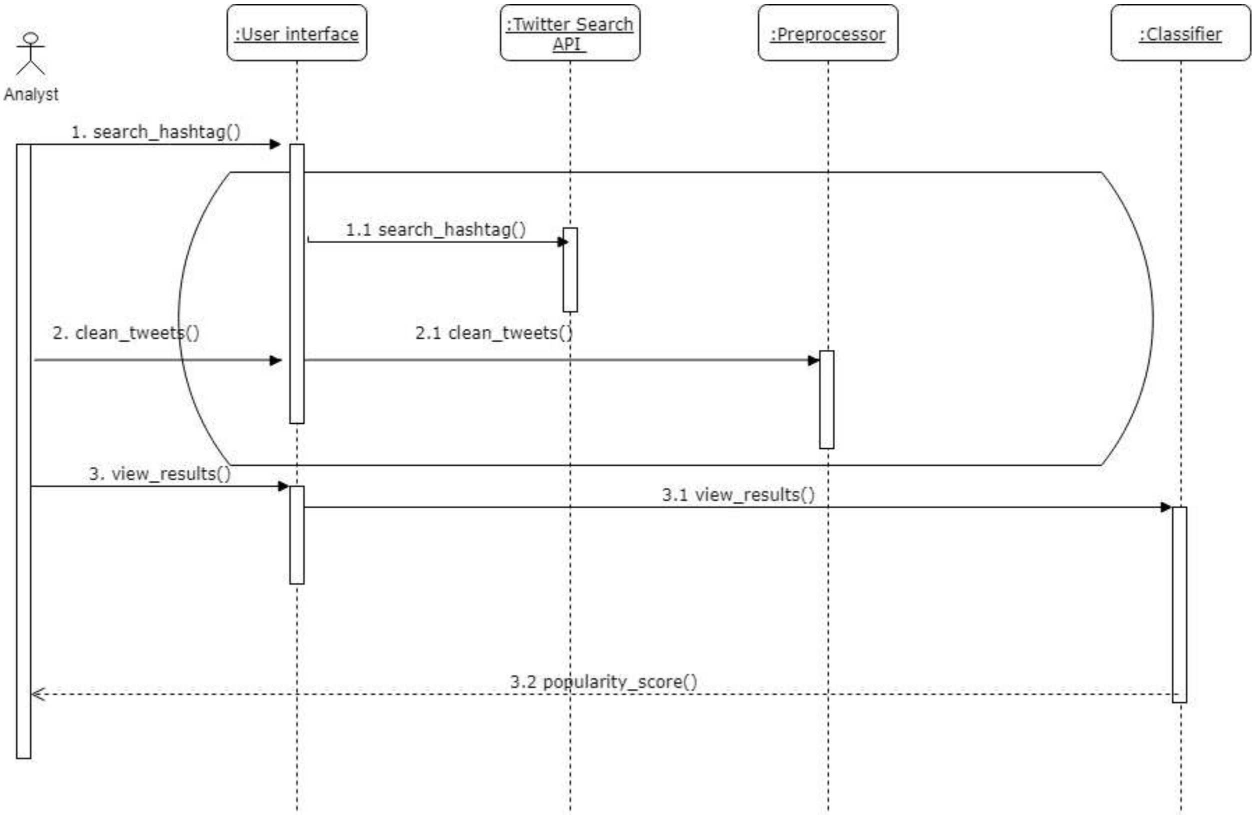


Figure 4.4: Sequence diagram

4.7 Data flow diagram

The diagram shown below, a level 0 data flow diagram, illustrates the entities, processes, data flow paths and data stores involved in the system. The first entity, the user, sends a request to get tweets related to a specific TV show. This request is sent to the process called **Collect tweets,** which then sends a message to the Twitter Search API, requesting for its oauth credentials to be approved. The API then returns a set of tweets, which are put in the data store labelled **D1: Retrieved tweets**. It is these tweets that undergo a series of pre-processing operations, and then placed in the data store called **D2: Clean tweets.** The **Classify tweets** process then labels the cleaned tweets by classifying them as either positive or negative. By doing so, the user is able to receive a popularity score of the TV show under analysis.

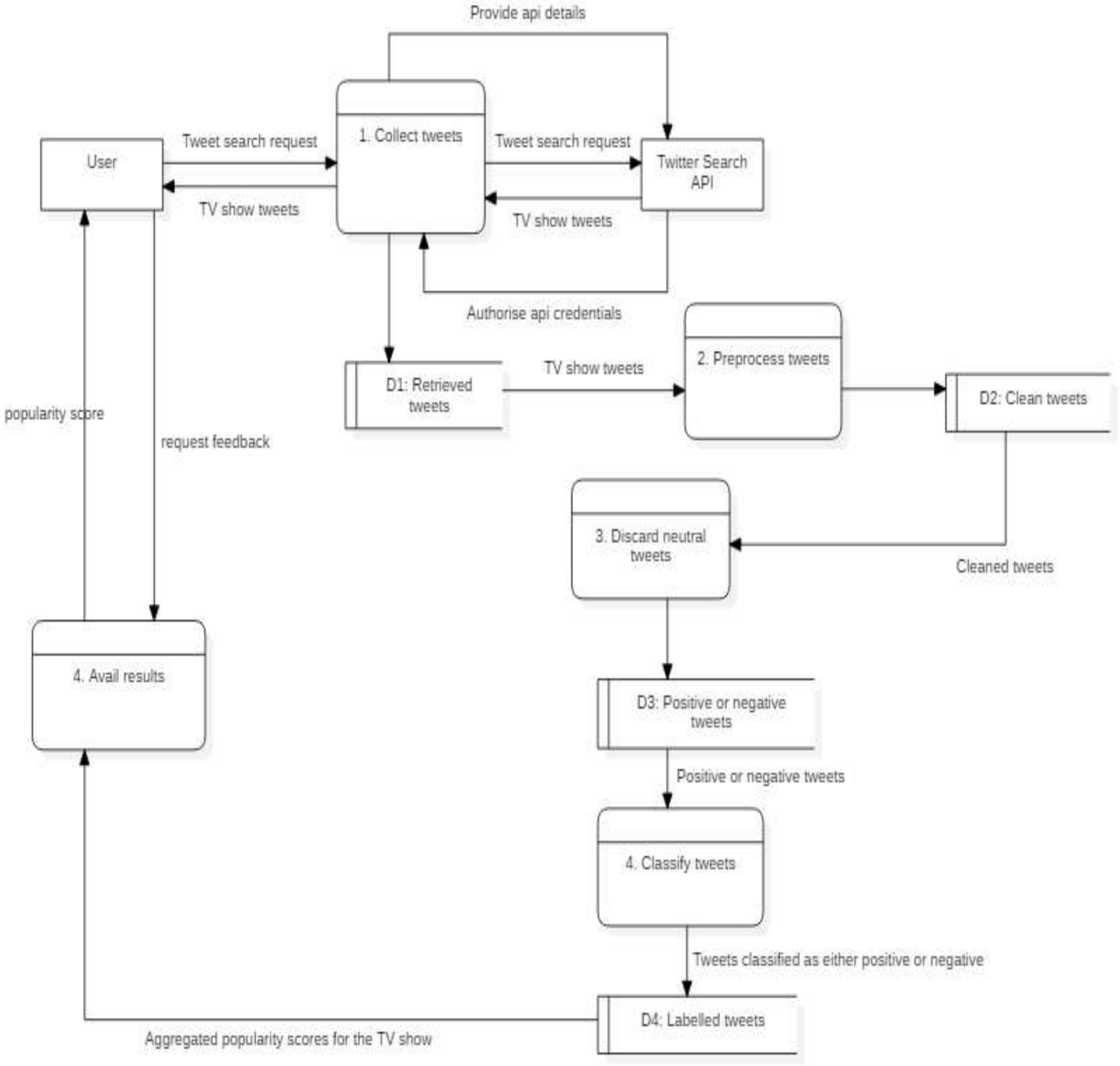


Figure 4.5: Level 0 Data flow diagram

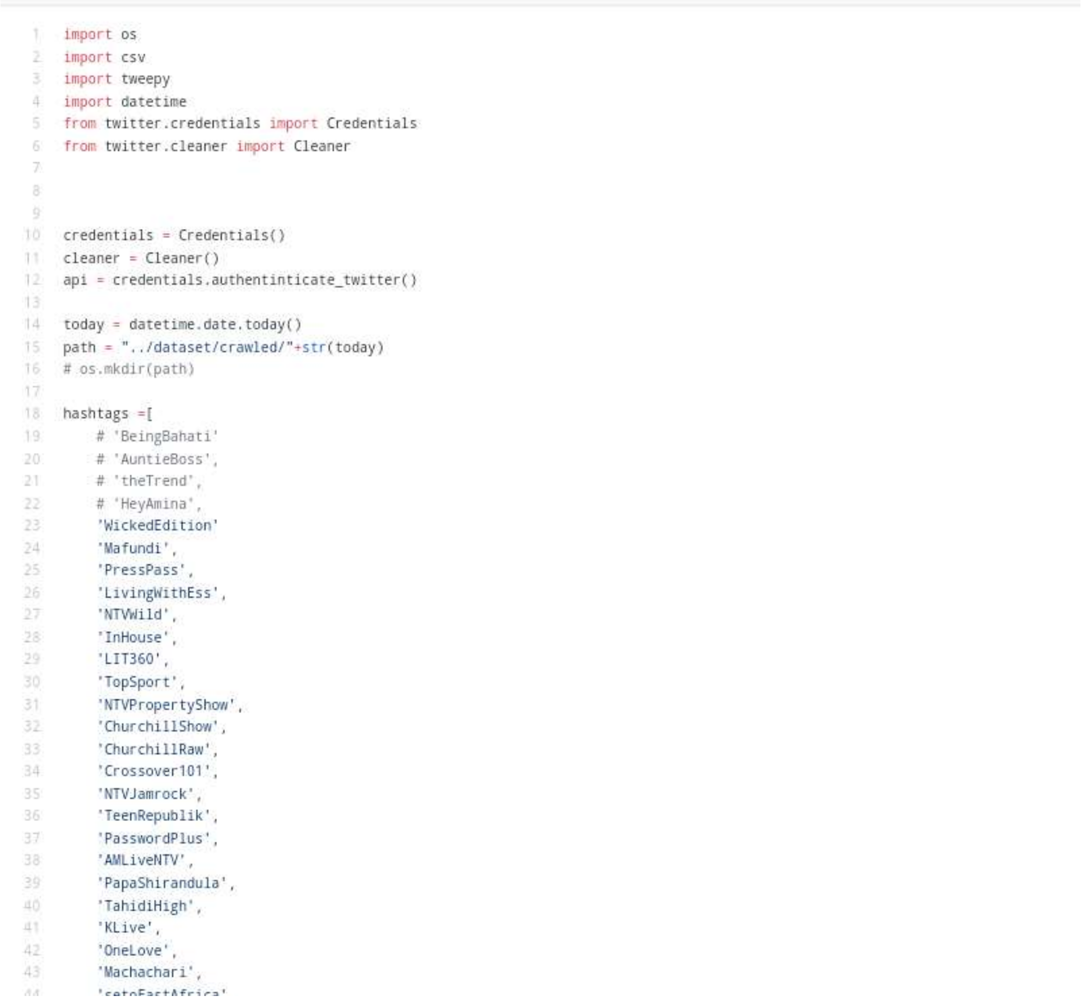
Chapter 5 : System Implementation and Testing

5.1 Introduction

The chapter seeks to explain how the solution is developed and tested. The first step entails constructing the TV show popularity corpus to be used in the machine learning process. The second process involves text pre-processing, where the corpus content is cleaned for training.

5.2 Designing the corpus

Tweets were collected using Twitter API and a set of predefined hashtags which represent the selected TV shows. Below is the code snippet for retrieving tweets using the Twitter API:

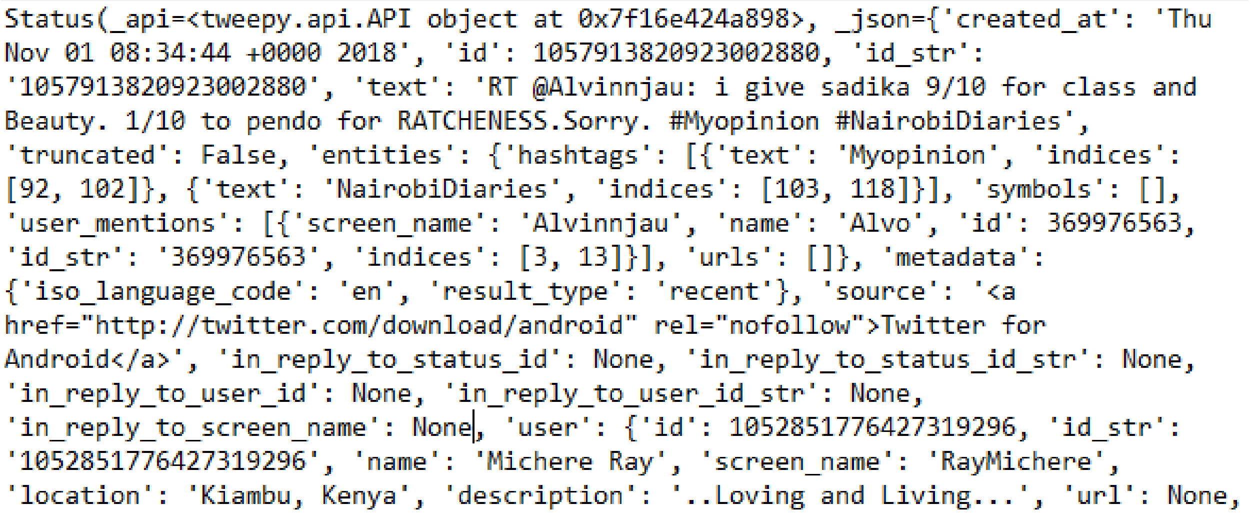


```
import os
import csv
import tweepy
import datetime
from twitter.credentials import Credentials
from twitter.cleaner import Cleaner



credentials = Credentials()
cleaner = Cleaner()
api = credentials.authentinticate_twitter()

today = datetime.date.today()
path = "../dataset/crawled/"+str(today)
# os.mkdir(path)

hashtags =[
    # 'BeingBahati'
    # 'AuntieBoss',
    # 'theTrend',
    # 'HeyAmina',
    'WickedEdition'
    'Mafundi',
    'PressPass',
    'LivingWithEss',
    'NTVWild',
    'InHouse',
    'LIT360',
    'TopSport',
    'NTVPropertyShow',
    'ChurchillShow',
    'ChurchillRaw',
    'Crossover101',
    'NTVJamrock',
    'TeenRepublik',
    'PasswordPlus',
    'AMLiveNTV',
    'PapaShirandula',
    'TahidiHigh',
    'KLive',
    'OneLove',
    'Machachari',
    'setoEastAfrica'
```

Figure 5.1: Collecting TV Show related tweets from Twitter

5.3 Text pre-processing

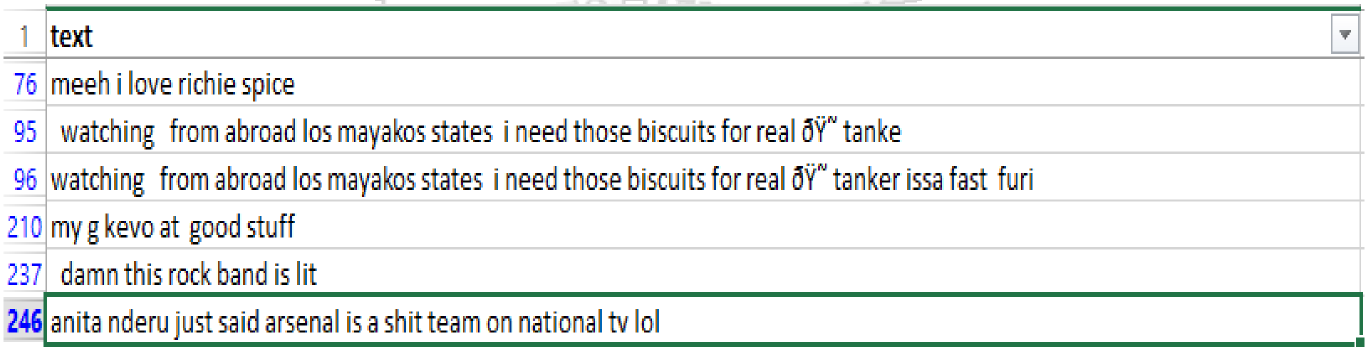
On collecting tweets using the Twitter API, the following is the data derived for a single tweet related to the show '*Nairobi Diaries'* identified by the hashtag *#NairobiDiaries.*



```
Status(_api=<tweepy.api.API object at 0x7f16e424a898>, _json={'created_at': 'Thu
Nov 01 08:34:44 +0000 2018', 'id': 1057913820923002880, 'id_str':
'1057913820923002880', 'text': 'RT @Alvinnjau: i give sadika 9/10 for class and
Beauty. 1/10 to pendo for RATCHENESS.Sorry. #Myopinion #NairobiDiaries',
'truncated': False, 'entities': {'hashtags': [{'text': 'Myopinion', 'indices':
[92, 102]}, {'text': 'NairobiDiaries', 'indices': [103, 118]}], 'symbols': [],
'user_mentions': [{'screen_name': 'Alvinnjau', 'name': 'Alvo', 'id': 369976563,
'id_str': '369976563', 'indices': [3, 13]}], 'urls': []}, 'metadata':
{'iso_language_code': 'en', 'result_type': 'recent'}, 'source': '<a
href="http://twitter.com/download/android" rel="nofollow">Twitter for
Android</a>', 'in_reply_to_status_id': None, 'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None, 'in_reply_to_user_id_str': None,
'in_reply_to_screen_name': None, 'user': {'id': 1052851776427319296, 'id_str':
'1052851776427319296', 'name': 'Michere Ray', 'screen_name': 'RayMichere',
'location': 'Kiambu, Kenya', 'description': '..Loving and Living...', 'url': None,
```

Figure 5.2: Sample tweet pulled using the Twitter API

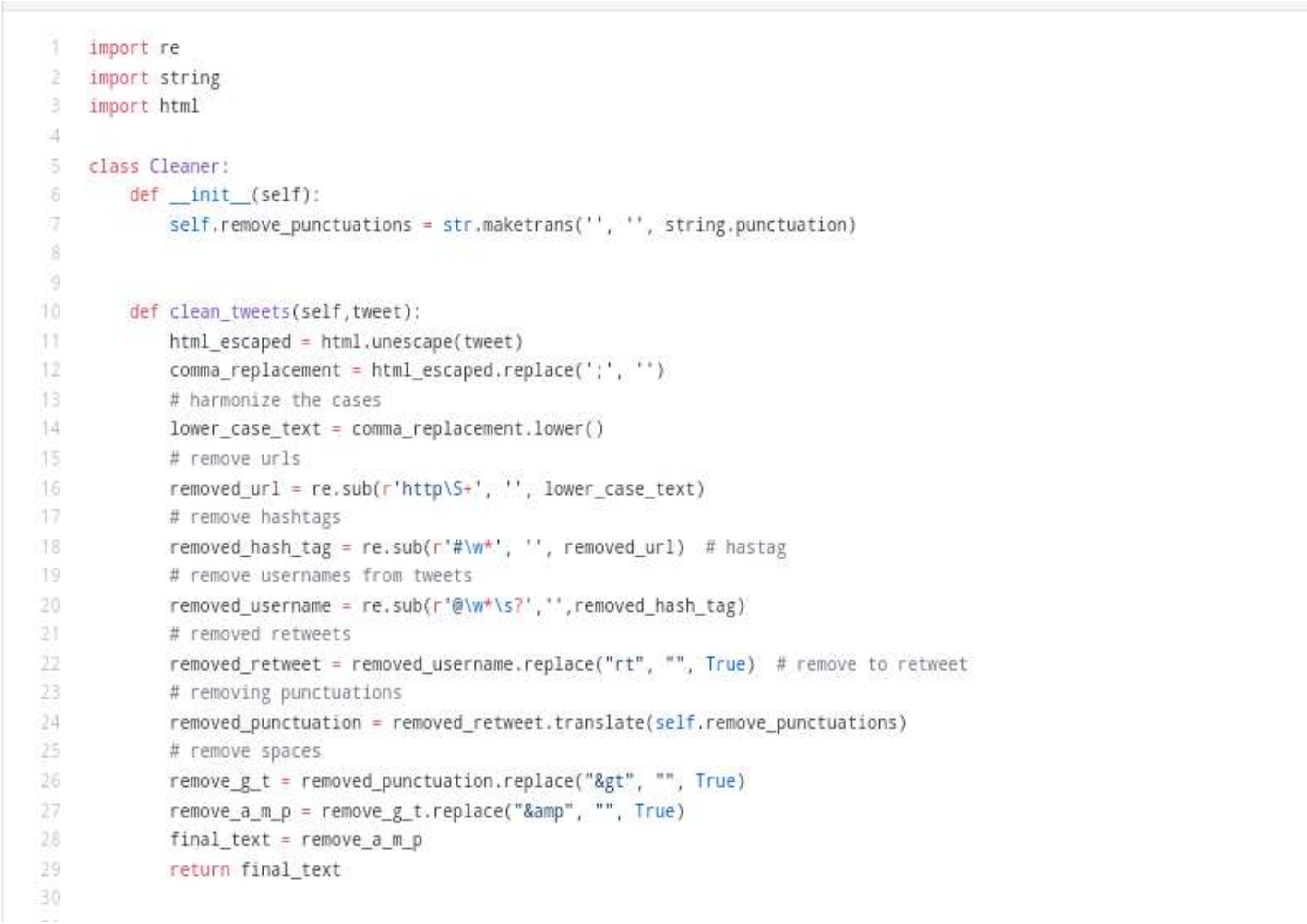
However, for purposes of building a corpora suitable to achieve the goal of this study, the focus is on the tweet text that is necessary to determine the sentiment communicated. In the processing process, information such as retweets, likes, hashtags etc. are removed. It therefore resulted in a sample of tweets like the ones below



1	text
76	meeh i love richie spice
95	watching from abroad los mayakos states i need those biscuits for real ðŸ˜ tanke
96	watching from abroad los mayakos states i need those biscuits for real ðŸ˜ tanker issa fast furi
210	my g kevo at good stuff
237	damn this rock band is lit
246	anita nderu just said arsenal is a shit team on national tv lol

Figure 5.3: Sample of processed tweets

The text pre-processing operation involves removing unnecessary punctuations, putting all text in lower case as well as removing hashtags. The code to carry out the operation is as shown in Figure 3.4.



```python
import re
import string
import html

class Cleaner:
    def __init__(self):
        self.remove_punctuations = str.maketrans('', '', string.punctuation)


    def clean_tweets(self,tweet):
        html_escaped = html.unescape(tweet)
        comma_replacement = html_escaped.replace(';', '')
        # harmonize the cases
        lower_case_text = comma_replacement.lower()
        # remove urls
        removed_url = re.sub(r'http\S+', '', lower_case_text)
        # remove hashtags
        removed_hash_tag = re.sub(r'#\w*', '', removed_url)  # hastag
        # remove usernames from tweets
        removed_username = re.sub(r'@\w*\s?','',removed_hash_tag)
        # removed retweets
        removed_retweet = removed_username.replace("rt", "", True)  # remove to retweet
        # removing punctuations
        removed_punctuation = removed_retweet.translate(self.remove_punctuations)
        # remove spaces
        remove_g_t = removed_punctuation.replace("&gt", "", True)
        remove_a_m_p = remove_g_t.replace("&amp", "", True)
        final_text = remove_a_m_p
        return final_text

```

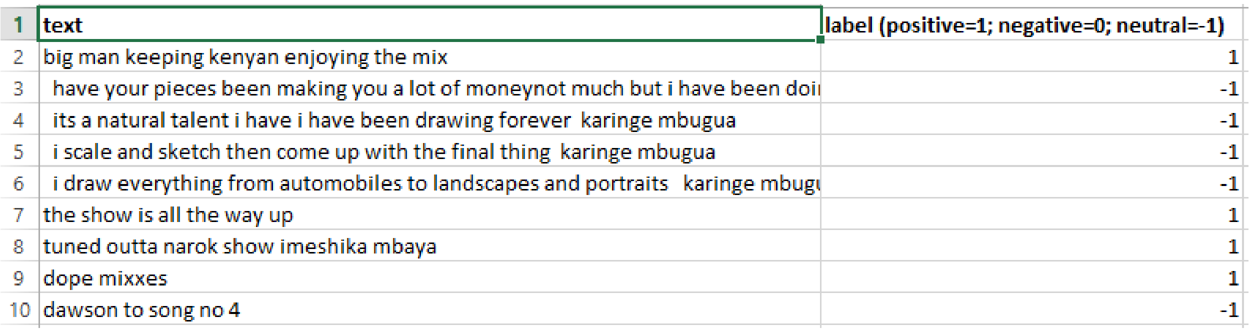
Figure 5.4: Code for cleaning tweets

After cleaning the tweets using the code above, there were three main categories in which the tweets were grouped. They are as shown in Table 5.1.

Table 5.1: The tweet labelling categories

Positive	Positive opinion towards the show
Negative	Negative opinion towards the show
Neutral	Neither positive or negative opinion

The tweets were then labelled as 1 for positive, 0 for negative and -1 for neutral. This is for the purpose of carrying out the supervised learning, as depicted in Figure 5.6.



1	text	label (positive=1; negative=0; neutral=-1)
2	big man keeping kenyan enjoying the mix	1
3	have your pieces been making you a lot of moneynot much but i have been doi	-1
4	its a natural talent i have i have been drawing forever karinge mbugua	-1
5	i scale and sketch then come up with the final thing karinge mbugua	-1
6	i draw everything from automobiles to landscapes and portraits karinge mbug	-1
7	the show is all the way up	1
8	tuned outta narok show imeshika mbaya	1
9	dope mixxes	1
10	dawson to song no 4	-1

Figure 5.5: Annotated tweets for supervised learning

5.4 Feature extraction

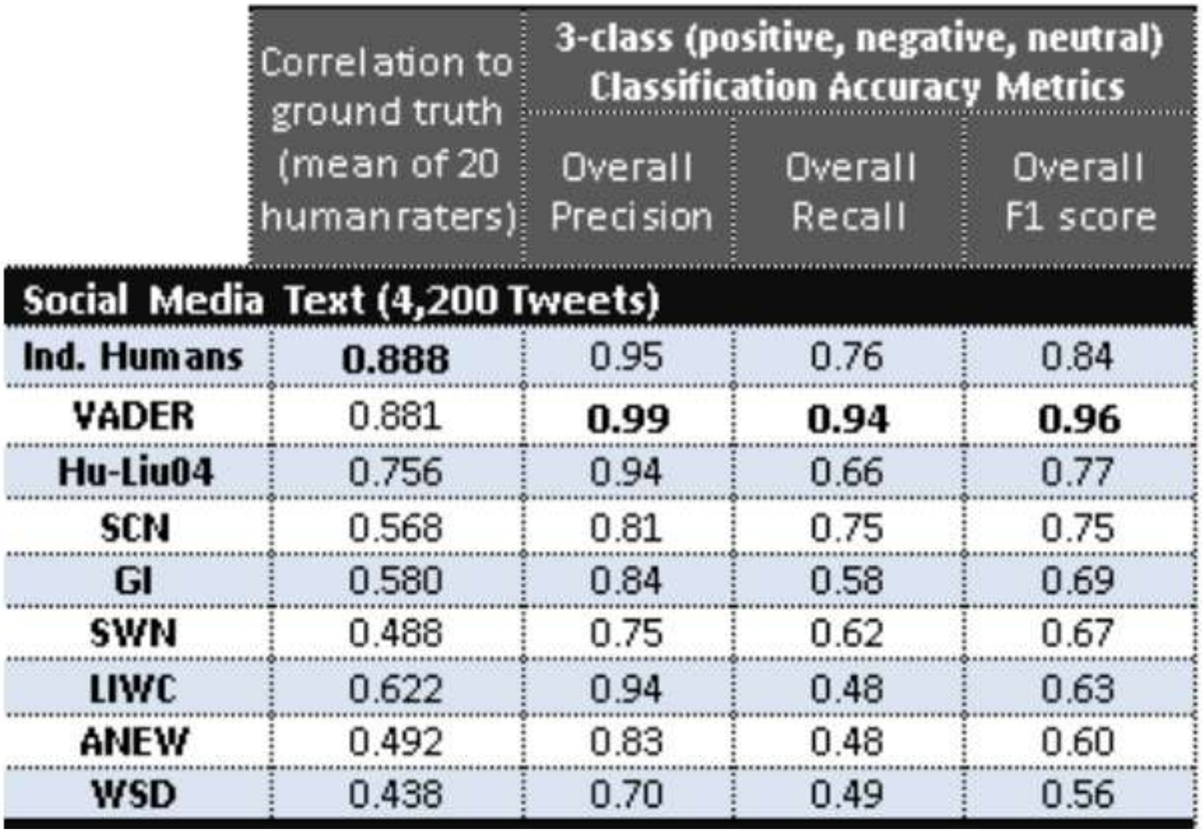
Feature extraction refers to the process of extracting special features from data input in order to be able to feed into the classifier (Scherer & Rao, 2011). When handling text, there is need to convert to integers since machine learning algorithms do not understand text. The same case applies to tweets. In order to convert the text to integers, the CountVectorizer method makes it possible to parse text and convert into integers then count the number of appearances in the corpus. On the other hand, the Term Frequency-inverse document frequency (TF_IDF) transformer normalizes the findings made by the CountVectorizer, focussing more on the more important words and degrading those occurring frequently but with less influence such as 'a', 'the'(Scikit-learn, 2007). In order to do both the CountVectorizer and the TF-IDF transform operations, the two operations were combined in one pipeline as shown below:

```
Pipeline([('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),

          ('svm', SVC(kernel="linear", C=1))])
```

5.5 Determining positive and negative tweets

The research gathered tweets based on the hashtags associated with the selected TV shows. While this approach delivered the required data for analysis, there was also the existence of large sets of tweets that were neutral. In some cases, the hashtags are misused to push content which does not relate to the TV show e.g. business advertisements.

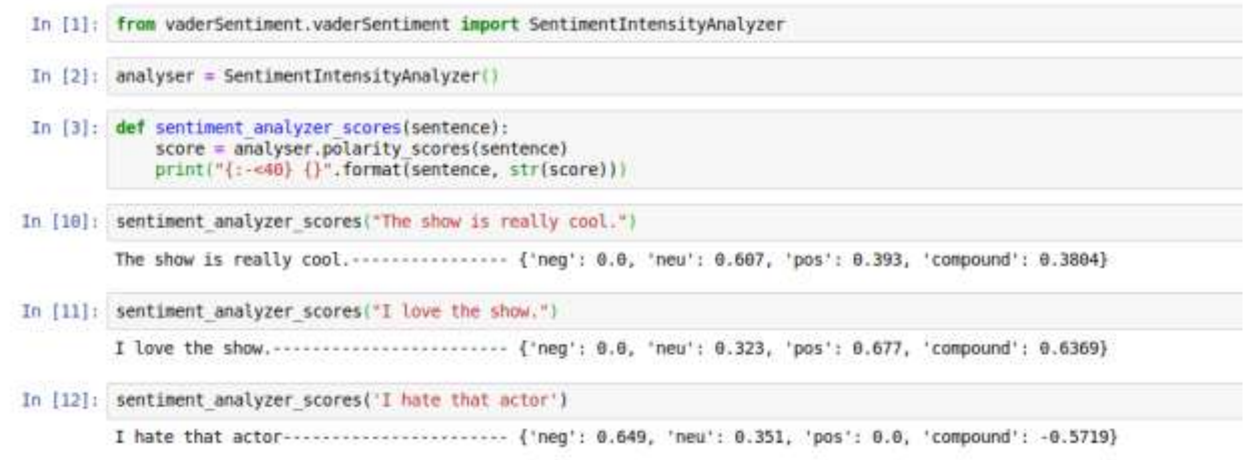
To acquire only the positive or negative tweets for analysis, the fetched tweets were through the VADER lexicon resource. The Valence Aware Dictionary for sEntiment Reasoning is a lexicon resource with over 7500 features that labels sentiments in the range of -4 (negative) to 4 (positive) (Hutto & Gilbert, 2014). The compound opinion value of a tweet would be calculated by adding up the sentiment values of each of the words and ranging from -1 (most negative) to +1 (most positive). The choice of VADER was made after experiments showed it perform better than the other lexicon techniques, including individual human raters as shown in Table 5.2. Tweets selected as either positive or negative would then proceed to undergo classification



	Correlation to ground truth (mean of 20 human raters)	3-class (positive, negative, neutral) Classification Accuracy Metrics		
		Overall Precision	Overall Recall	Overall F1 score
Social Media Text (4,200 Tweets)				
Ind. Humans	**0.888**	0.95	0.76	0.84
VADER	0.881	**0.99**	**0.94**	**0.96**
Hu-Liu04	0.756	0.94	0.66	0.77
SCN	0.568	0.81	0.75	0.75
GI	0.580	0.84	0.58	0.69
SWN	0.488	0.75	0.62	0.67
LIWC	0.622	0.94	0.48	0.63
ANEW	0.492	0.83	0.48	0.60
WSD	0.438	0.70	0.49	0.56

Table 5.2 Performance of VADER against other lexicon dictionaries (Hutto & Gilbert, 2014)

The snippet below illustrates how the VADER lexicon resource was used. The sentiment values should sum up to 1. For instance, the text ‘I love the show’ scores a negative value of 0, neutral value of 0.323 and positive value of 0.677. The compound score 0.6369 is closer to 1, thus qualifies as a positive text.



```
In [1]: from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

In [2]: analyser = SentimentIntensityAnalyzer()

In [3]: def sentiment_analyzer_scores(sentence):
            score = analyser.polarity_scores(sentence)
            print("{:-<40} {}".format(sentence, str(score)))

In [10]: sentiment_analyzer_scores("The show is really cool.")

         The show is really cool.---------------- {'neg': 0.0, 'neu': 0.607, 'pos': 0.393, 'compound': 0.3804}

In [11]: sentiment_analyzer_scores("I love the show.")

         I love the show.------------------------ {'neg': 0.0, 'neu': 0.323, 'pos': 0.677, 'compound': 0.6369}

In [12]: sentiment_analyzer_scores('I hate that actor')

         I hate that actor----------------------- {'neg': 0.649, 'neu': 0.351, 'pos': 0.0, 'compound': -0.5719}
```

Figure 5.6: VADER code snippet deriving the sentiment value of a text

5.6 Determining a TV show’s overall score

To calculate the aggregate score of a selected TV show, it was calculated as the total number of positive tweets, divide by the total number of positive and negative tweets. Once the analyst enters the hashtag related to a given TV show, the system first labels them as either positive or negative. After, the overall score is calculated using the formula below:

$$Overall\ score = \frac{total\ positive\ tweets}{total\ positive\ and\ negaive\ tweets}$$

Equation 5.1: Calculating overall popularity score

5.7 Training the SVM model

Machine learning algorithms do not analyse plain text. To be put in a language that is understood by the classifier. Python provides a library called scikit-learn, which enables one to remove some characters such as punctuation marks in a process called tokenization, as well as carry out feature extraction, a process that entails converting words into integers in order to be inputted into the classifier (Brownlee, 2017).

On completing the pre-processing and annotations as earlier mentioned, the model could now be trained. SVM with bigram feature gave the best results during the experiments. The process of token vectorization and transformation can be combined into one command under the pipeline method. The corpus was split into two: 70 percent for training the model and 30 percent for testing the model. A snippet of the training code used to test the SVM model is as shown below.

```
def
svm_accuracy(X,
y):
                      X_train, X_test, y_train, y_test = train_test_split(X, y,
                  test_size=0.3, random_state=1)
                      svm = Pipeline([('vect', CountVectorizer()), ('tfidf',
                  TfidfTransformer()),
                                            ('svm', SVC(kernel="linear", C=1))])
                      svm = svm.fit(X_train, y_train)
                      ypred = svm.predict(X_test)
```

5.8 Testing the model

The dataset for training was made up of 7,500 tweets of the initial tweets. The model was tested using the confusion matrix as shown

Table 5.3 Confusion matrix for the model

	Actual 0 (negative)	Actual 1 (positive)
Predicted 0(negative)	3311	474
Predicted 1(positive)	400	3315

The values for true positive, true negative, false positive and false negative were as shown below:

Table 5.4: Values for the confusion matrix variables

True Positive (TP)	3315
False Positive (FP)	474
False Negative (FN)	400
True Negative (TN)	3311

Chapter 6 : Discussions

This chapter analyses the results of the research with reference to the set objectives. The purpose of the research is to design a prediction model that determines the popularity of TV shows for effective decision-making in the TV production department. SVM delivered better results in the sentiment analysis.

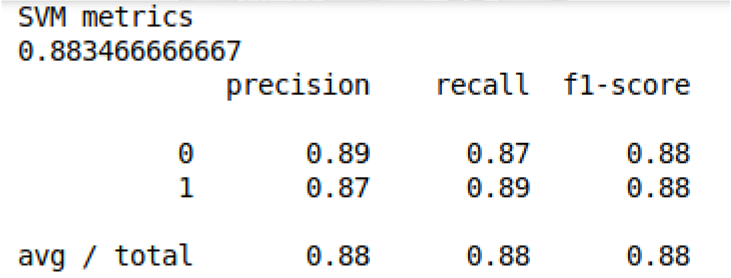
6.1 Experiments on Sentiment analysis

6.1.1 Using different classifiers

The performance of four different classifiers was measured in terms of sentiment analysis. These machine learning methods were SVM, K-nearest neighbour, Naïve Bayes and Random Forest. The experiments delivered the following results

Using SVM, it delivered results as shown in Table 6.1.

Table 6.1: SVM performance



```
SVM metrics
0.883466666667
             precision    recall  f1-score

          0       0.89      0.87      0.88
          1       0.87      0.89      0.88

avg / total       0.88      0.88      0.88
```

The Receiver Operating Characteristic (ROC) curve which shows how the SVM Classifier performed, is as shown in Figure 6.1

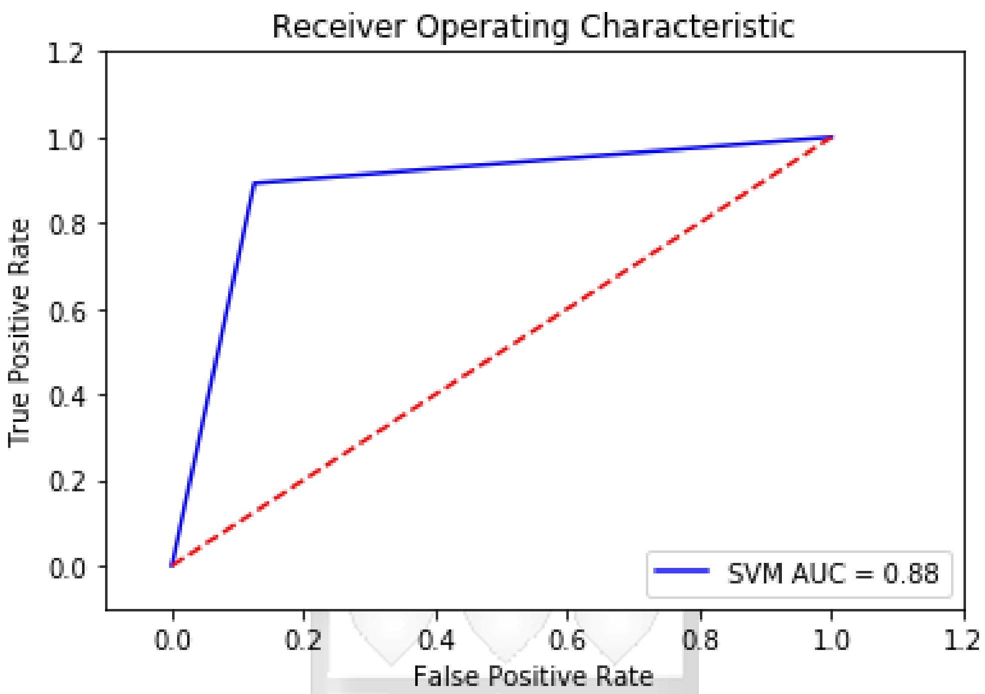
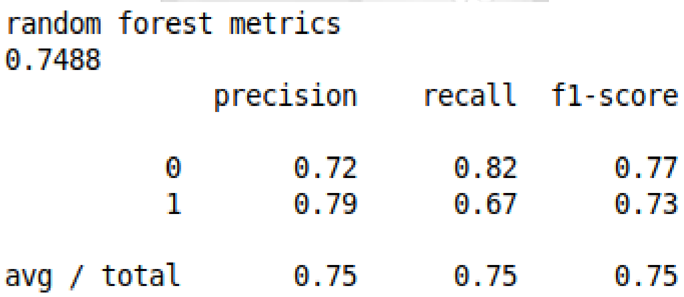


Figure 6.1: SVM ROC curve

Using Random Forest, it delivered results as shown in Table 6.2.

Table 6.2: Random Forest performance



```
random forest metrics
0.7488
             precision    recall  f1-score

          0       0.72      0.82      0.77
          1       0.79      0.67      0.73

avg / total       0.75      0.75      0.75
```

The ROC curve which shows how the random forest classifier performs, is as shown in Figure 6.2

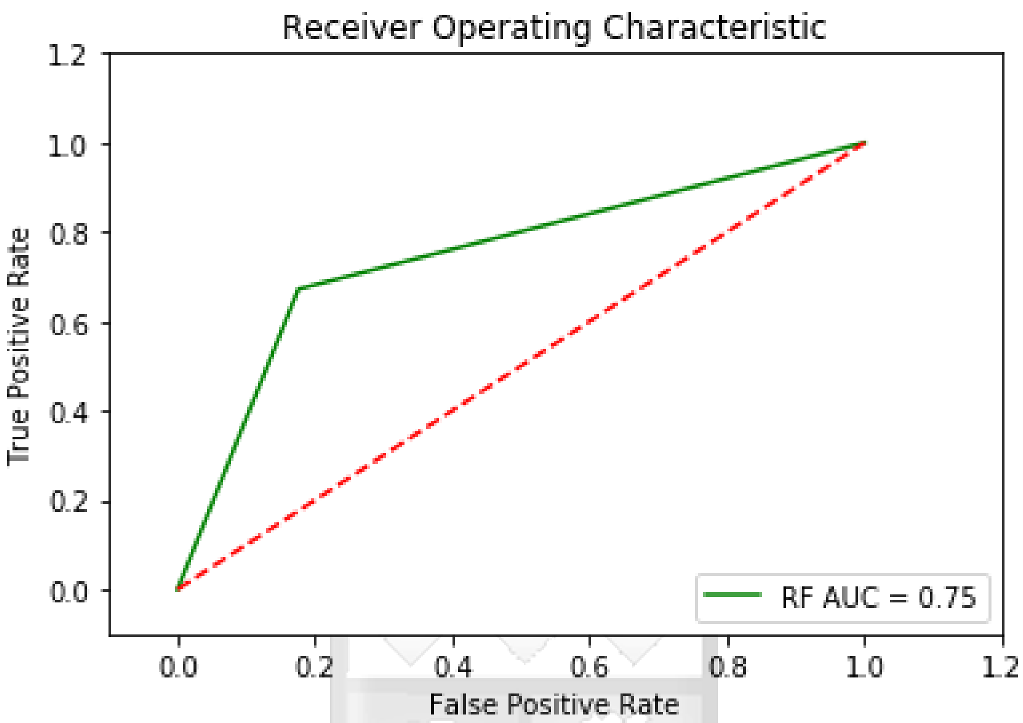
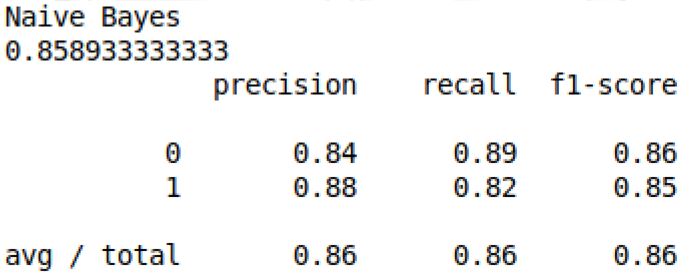


Figure 6.2: Random Forest ROC curve

Using Naïve Bayes, it delivered results as shown in Table 6.3.

Table 6.3: Naive Bayes performance



```
Naive Bayes
0.858933333333
             precision    recall  f1-score

          0       0.84      0.89      0.86
          1       0.88      0.82      0.85

avg / total       0.86      0.86      0.86
```

The ROC curve which shows how the naïve bayes classifier performs, is as shown in Figure 6.3

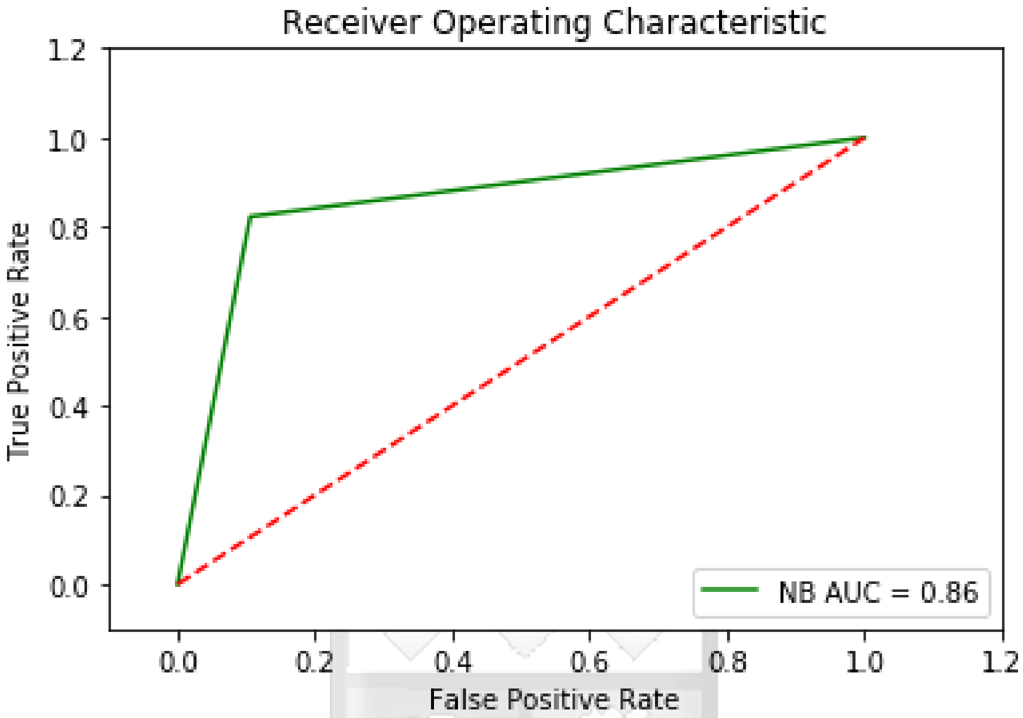
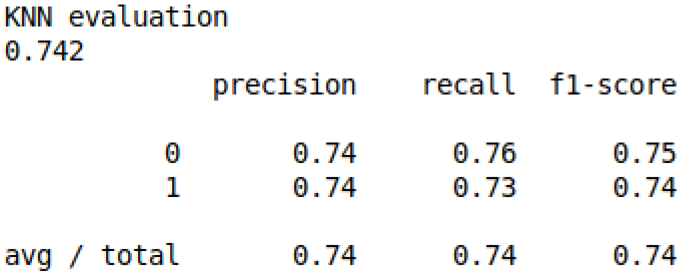


Figure 6.3: Naive Bayes ROC curve

Using K-nearest neighbour, it delivered results as shown in Table 6.3.

Table 6.4: K-Nearest neighbour performance



```
KNN evaluation
0.742
             precision    recall  f1-score

          0       0.74      0.76      0.75
          1       0.74      0.73      0.74

avg / total       0.74      0.74      0.74
```

The ROC curve which shows how the KNN classifier performs, is as shown in Figure 6.4

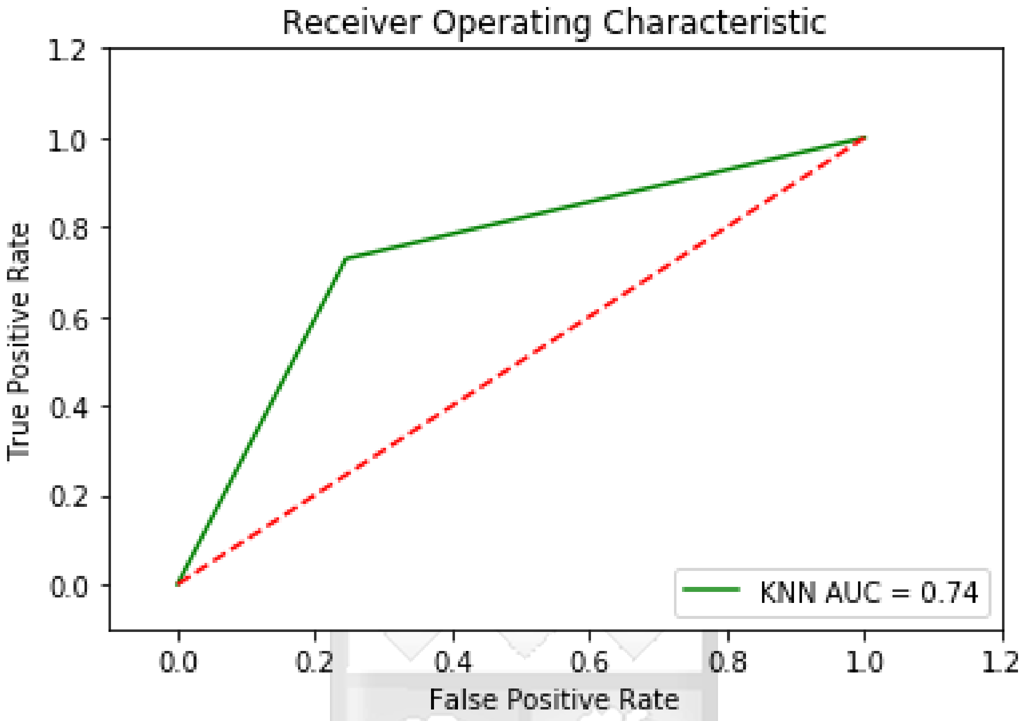


Figure 6.4: KNN ROC curve

Table 6.6 represents the summary of the classifiers' performances

Table 6.5: Performance of classifiers

Classifier	Accuracy	Precision	Recall	F-score
SVM	0.88	0.88	0.88	0.88
Random Forest	0.75	0.75	0.75	0.75
K-nearest neighbour	0.74	0.74	0.74	0.74
Naïve Bayes	0.86	0.86	0.86	0.86

6.1.2 Experiment 1: SVM with different feature types

The features used in this experiment were unigram, bigram and trigram. The results showed that the best performance of the classifier was when using bigram feature.

Table 6.6: Performance of SVM

Feature	Accuracy	Precision	Recall	F-score
Unigram	0.8866	0.89	0.89	0.89
Bigram	0.892	0.89	0.89	0.89
Trigram	0.878	0.88	0.88	0.88

6.1.3 Experiment 2: Random Forest with different feature types

The features used in this experiment were unigram, bigram and trigram. The results showed that the best performance of the Random Forest classifier was when using unigram feature.

Table 6.7: Performance of Random Forest

Feature	Accuracy	Precision	Recall	F-score
Unigram	0.74	0.74	0.73	0.73
Bigram	0.73	0.73	0.73	0.72
Trigram	0.73	0.73	0.73	0.73

6.1.4 Experiment 3: KNN with different feature types

The same experiment done on SVM and Random Forest classifiers was done to gauge KNN performance. The features used in this experiment were unigram, bigram and trigram. The results showed that the best performance of the KNN classifier was when using unigram and bigram features.

Table 6.8: Performance on K-nearest neighbour

Feature	Accuracy	Precision	Recall	F-score
Unigram	0.74	0.74	0.73	0.73
Bigram	0.74	0.74	0.71	0.71
Trigram	0.73	0.73	0.70	0.69

6.1.5 Experiment 4: Naïve Bayes with different feature types

The same experiment was done to gauge KNN performance. The features used in this experiment were unigram, bigram and trigram. The results showed that the best performance of the Naïve Bayes classifier was when using bigram feature.

Table 6.9: Performance on Naive Bayes

Feature	Accuracy	Precision	Recall	F-score
Unigram	0.874	0.86	0.88	0.85
Bigram	0.881	0.89	0.87	0.87
Trigram	0.872	0.88	0.88	0.88

Chapter 7 : Conclusion and recommendations

7.1 Conclusion

The research purposed to design a system that determines the popularity of a TV show based on sentiment analysis on Twitter data. To be able to successfully deliver on the set objectives, intense research on relevant works and literature done was done. An expert on TV programming matters at the Nation Media Group also provided actual insights into the decision-making process of which TV shows get aired. Research on the available machine learning algorithms was done to clearly understand how they can be applicable in the research.

Tweets related to TV shows were collected based on a set of predetermined Twitter hashtags using the Twitter API, and then underwent text pre-processing and labelling as either positive (1), negative (0) or neutral (-1). The corpus was then split into two; the training set, which was used to train the SVM model, and the testing set which was used to gauge the performance of the model.

A series of four experiments were carried out to determine which classifier delivered the best results. Indeed, SVM model with bigram features had the highest accuracy score, of 89 percent. The system was then deployed to carry out similar classifications on tweets obtained via the Twitter search API.

7.2 Recommendations

This thesis paper illustrates that SVM model can be useful when intending to get data on the popularity of a TV show using the available related tweets on Twitter. This approach compensates for the unexpected levels of subjectivity brought about by the use of focus groups in mediahouses to determine popularity. The researcher acknowledges that better results would have been yielded if the training dataset was larger.

7.3 Future work

Twitter dialect is a mix of English, Swahili and other dialects unique to persons in different areas. This implies that there is need to have a model that carries out classifications by using more than just the English language which is available in the lexical resources that are in English.

Also, the use of multimedia content to pass coded messages on twitter is on the rise, especially through use of emoticons, short videos, GIF content as well as memes. Such content was not included in building the corpus necessary for training the model. It would be useful if such type of content is considered.

In addition to this, the nature of tweets fetched by the search API are usually not comprehensive enough to effectively carry out labelling and classification. This could be as a result of the limitedness of the size of a tweet, which is currently at 280 characters. This presents an opportunity for research to be done on how to pre-formulate complete statements that have been pre-processed and cleaned, before carrying out classifications.

References

Ahmed, H., Qamar, A., & Razzaq, M. (2013). Prediction of popular tweets using Similarity Learning. *IEEE*. https://doi.org/10.1109/ICET.2013.6743524

Asur, S., & Huberman, B. A. (2010). Predicting the Future with Social Media. *IEEE*, 492–499. https://doi.org/10.1109/WI-IAT.2010.63

Bai, J., & Nie, J.-Y. (2004). Using Language Models for Text Classification. *Département d'informatique et de Recherche Opérationnelle*, 1–6.

Brownlee, J. (2017). How to Prepare Text Data for Machine Learning with scikit-learn. Retrieved from Machine Learning Mastery website: https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/

Cavanillas, J. M., Curry, E., & Wahlster, W. (2015). *New Horizons for a Data-Driven Economy*. Springer Nature.

Consumer Federation of Kenya. (2018). Storm brewing at Nation Media Group as mass layoffs beckon. Retrieved from http://www.cofek.co.ke/index.php/news-and-media/1878-storm-brewing-at-nation-media-group-as-mass-layoffs-beckon

Dennis, A., Wixom, B. H., & Roth, R. M. (2012). *System Analysis & Design*. John Wiley & Sons, Inc.

FluidSurveys. (2013). Avoiding Survey Bias. Retrieved from http://fluidsurveys.com/university/avoiding-survey-bias/

Garlan, D. (2000). Software Architecture: a Roadmap. *School of Computer Science, Carnegie Mellon University*.

GeoPoll. (2017). Kenya Q1 2017 Radio & TV Audience Ratings Report. Retrieved from https://knowledge.geopoll.com/kenya-media-measurement-kgmm-report-0-0

Ghahrai, A. (2018). Rapid Application Development (RAD). Retrieved from Testing Excellence website: https://www.testingexcellence.com/rapid-application-development-rad/

Global Web Index. (2019). Most famous social network sites 2019, by active users. Retrieved from Statista website: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

Gursun, G., Crovella, M., & Matta, I. (2010). Describing and Forecasting Video Access Patterns. *Department of Computer Science, Boston University*, 1–9.

Halbrooks, G. (2018). How to Understand Nielsen TV Ratings. Retrieved from Nielsen website: https://www.thebalance.com/how-to-understand-nielsen-tv-ratings-2315476

Hassani, H., & Silva, E. S. (2015). *Forecasting with Big Data: A Review*. 5–19. https://doi.org/10.1007/s40745-015-0029-9

Hinami, R., & Satoh, S. (2017). Audience Behavior Mining by Integrating TV Ratings with Multimedia Contents. *IEEE MultiMedia*, *24*(2), 44–54. https://doi.org/10.1109/MMUL.2017.25

Hsieh, W.-T., Chou, S. T., Cheng, Y.-H., & Wu, C.-M. (2013). Predicting TV Audience Rating with Social Media. *Workshop on Natural Language Processing for Social Media*, 1–5.

Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Association for the Advancement of Artificial Intelligenc*.

Jain, V. (2013). Prediction of Movie Success using Sentiment Analysis of Tweets. *The International Journal of Soft Computing and Software Engineering*, *3*, 308–313. https://doi.org/10.7321/jscse.v3.n3.46

Kamdar, S. (2017). 5 Differences Between Old and New Media. Retrieved from https://www.huffpost.com/entry/5-differences-between-old_b_9670634

Kenya Film Classification Board (last). (2018). *A Report on survey to establish stakeholders' opinion towards KFCB'S mandate and content classification function in Kenya*. Retrieved from http://kfcb.co.ke/wp-content/uploads/2018/06/KFCB-REPORT.pdf

Kenya National Bureau of Statistics. (2018). *Economic Survey 2018*. Nairobi, Kenya.

Kharde, V. A., & Sonawane, S. S. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. *Sentiment Analysis of Twitter Data: A Survey of Techniques*, *139*, 5–15.

Liu, B. (2011). *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*. University of Illinois, Chicago: Springer.

MathWorks. (2018). Machine Learning. Retrieved from MathWorks website: https://www.mathworks.com/discovery/machine-learning.html

Media Council of Kenya. (2018). *State of the Media Report*.

Miller, M. (2017). How Big Data predicts audience behaviour, drives marketing. Retrieved from International News Media Asociation website: https://www.inma.org/blogs/conference/post.cfm/how-big-data-predicts-audience-behaviour-drives-marketing

Mueller, J. P. (2014). *Beginning Programming with Python for Dummies*. Retrieved from https://doc.lagout.org/programmation/python/Beginning%20Programming%20with%20Python%20for%20Dummies%20%5BMueller%202014-09-22%5D.pdf

Mugambi, S. (2017). *Sentiment analysis for hate speech detection on social media: TF-IDF weighted N-Grams based approach* (Strathmore University). Retrieved from http://suplus.strathmore.edu/handle/11071/5657

Nation Media Group. (2019). Nation Media Group. Retrieved from http://www.nationmedia.com/who-we-are/

Nayak, J., Naik, B., & Behera, H. S. (2015). A Comprehensive Survey on Support Vector Machine in Data Mining Tasks: Applications & Challenges. *International Journal of Database Theory and Application*, *8*, 169–186.

Nielsen. (2018). Ratings Academy. Retrieved from Nielsen website: http://ratingsacademy.nielsen.com/media-overview/tv-panels-overview

Oh, C., Yergeau, S., Woo, Y., Wurtsmith, B., & Vaughn, S. (2015). Is Twitter Psychic? Social Media Analytics and Television Ratings. *2015 Second International Conference on Computing Technology and Information Management (ICCTIM)*. https://doi.org/DOI: 10.1109/ICCTIM.2015.7224610

Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Universit´e de Paris-Sud, Laboratoire*, 1320–1326.

Palanisamy, P., Yadav, V., & Elchuri, H. (2013). Serendio: Simple and Practical lexicon based approach to Sentiment Analysis. *Second Joint Conference on Lexical and Computational Semantics*, *2*, 543–548.

Pitchai, P., Veerapadran, C., & Rajasekar, S. (2013). *Research Methodology*.

PwC. (2013). *Kenyan entertainment and media outlook:2013 – 2017*. Retrieved from https://www.pwc.com/ke/en/assets/pdf/entertainment-and-media-outlook.pdf

Ramadhani, A. M., & Goo, H. S. (2017). Twitter sentiment analysis using deep learning methods. *IEEE*. https://doi.org/10.1109/INAES.2017.8068556

Scangroup. (2015). *Digital migration and its immediate implications to advertisers*. Retrieved from http://www.wpp-scangroup.com/pdf/DIGITAL%20MIGRATION[1].pdf

Scherer, R., & Rao, R. (2011). *Non-Manual Control Devices: Direct Brain-Computer Interaction*. Retrieved from https://www.igi-global.com/dictionary/feature-extraction/10960

Scikit-learn. (2007). Feature extraction. Retrieved from https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

Sharma, G. (2017). Pros and cons of different sampling techniques. *International Journal of Applied Research*, 749–752.

Simons, M. (2017). Journalism faces a crisis worldwide – we might be entering a new dark age. Retrieved from The Guardian website: https://www.theguardian.com/media/2017/apr/15/journalism-faces-a-crisis-worldwide-we-might-be-entering-a-new-dark-age

Swamy, M. N., & Hanumanthappa, M. (2013). Indian Language Text Representation and Categorization Using Supervised Learning Algorithm. *International Journal of Data Mining Techniques and Applications*, *2*, 251–257.

Tsuchiya, A. (2014). Television must mine bigger data or risk being Netflixed. Retrieved from The Guardian website: https://www.theguardian.com/media-network/2014/aug/04/tv-big-data-mine-customer-netflix

Twitter. (2019a). Search API. Retrieved from https://developer.twitter.com/en/docs/tweets/search/overview.html

Twitter. (2019b). Search Tweets. Retrieved from Twitter website: Twitter. (2019). Shttps://developer.twitter.com/en/docs/tweets/search/overview.html

Vawdrey, J. (2015). Using Data Science to Predict TV Viewer Behavior and Formulate a Hit TV Show. Retrieved from Pivotal website: https://content.pivotal.io/blog/using-data-science-to-predict-tv-viewer-behavior-and-formulate-a-hit-tv-show

Visa, S., Ramsay, B., Ralescu, A., & Knaap, E. van der. (2011). Confusion Matrix-based Feature Selection. *CEUR Workshop Proceedings*.

Wang, L., & Gan, J. (2017). Prediction of the 2017 French election based on Twitter data analysis. *IEEE*, 89–93. https://doi.org/10.1109/CEEC.2017.8101605

Wasson, C. S. (2006). *System Analysis, Design, and Development Concepts, Principles, and Practices*. Retrieved from http://www.zu.edu.jo/UploadFile/Library/E_Books/Files/LibraryFile_12159_26.pdf

Wicaksono, A. J., Suyoto, & Pranowo. (2017). A proposed method for predicting US presidential election by analyzing sentiment in social media. *IEEE*. https://doi.org/10.1109/ICSITech.2016.7852647

Zainuddin, N., & Selamat, A. (2014). Sentiment Analysis Using Support Vector Machine. *IEEE*. https://doi.org/10.1109/I4CT.2014.6914200

Zhu, C., Cheng, G., & Wang, K. (2015). Big Data Analytics for Program Popularity Prediction in Broadcast TV Industries. *IEEE*, *5*, 24593–24601. https://doi.org/10.1109/ACCESS.2017.2767104

Appendices

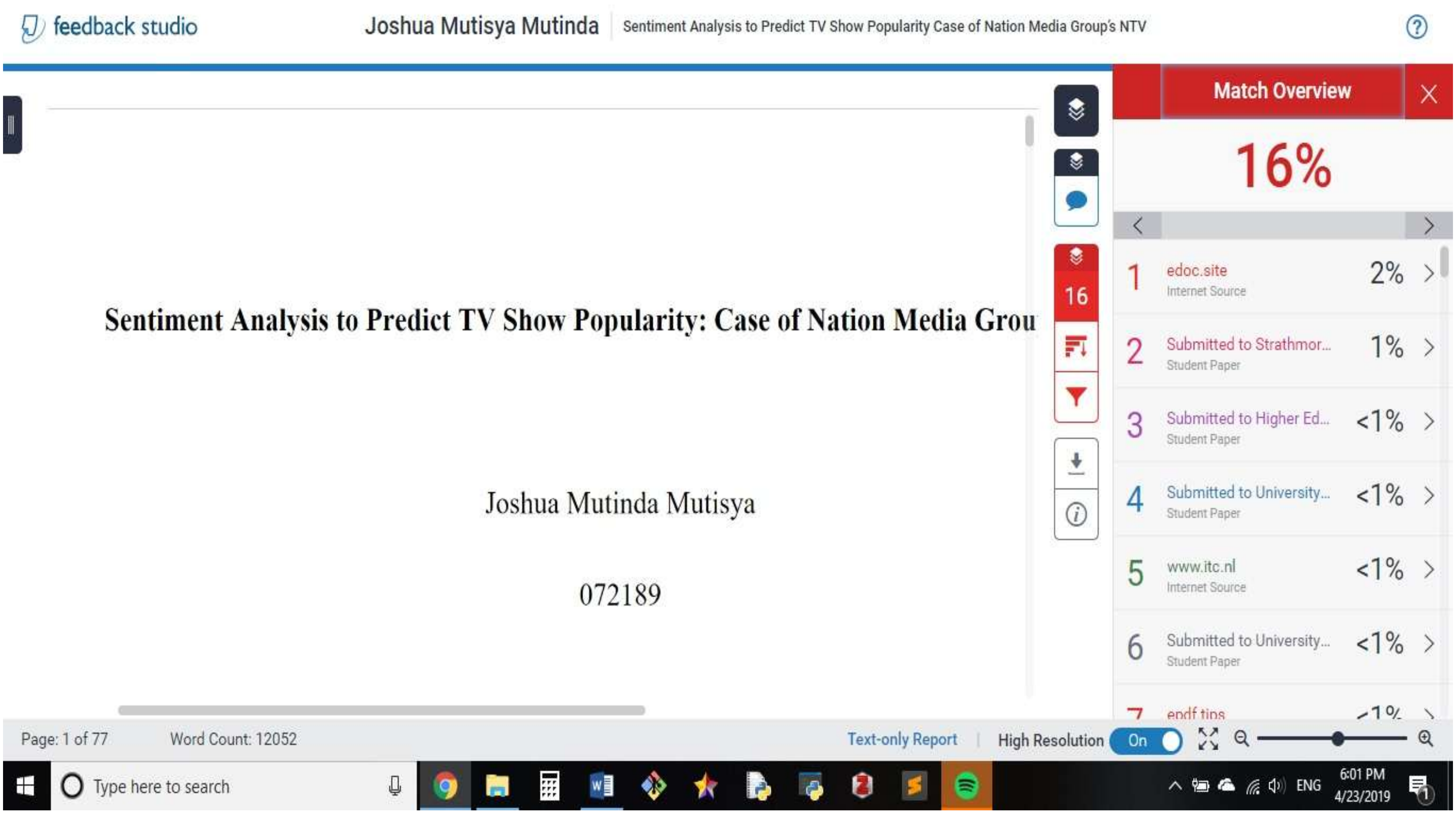
The code used to carry out the cleaning process and convert into a csv file:

```
import
re
         import string
         import html


         class Cleaner:
             def __init__(self):
                 self.remove_punctuations = str.maketrans('', '', string.punctuation)


             def clean_tweets(self,tweet):
                 html_escaped = html.unescape(tweet)
                 comma_replacement = html_escaped.replace(';', '')
                 # harmonize the cases
                 lower_case_text = comma_replacement.lower()
                 # remove urls
                 removed_url = re.sub(r'http\S+', '', lower_case_text)
                 # remove hashtags
                 removed_hash_tag = re.sub(r'#\w*', '', removed_url)  # hastag
                 # remove usernames from tweets
                 removed_username = re.sub(r'@\w*\s?','',removed_hash_tag)
                 # removed retweets
                 removed_retweet = removed_username.replace("rt", "", True)  # remove to
         retweet
                 # removing punctuations
                 removed_punctuation = removed_retweet.translate(self.remove_punctuations)
                 # remove spaces
                 remove_g_t = removed_punctuation.replace("&gt", "", True)
                 remove_a_m_p = remove_g_t.replace("&amp", "", True)
                 final_text = remove_a_m_p
                 return final_text
```

Below is the SVM model training code:

```
def
svm_accuracy(X,
y):
                          X_train, X_test, y_train, y_test = train_test_split(X, y,
                      test_size=0.3, random_state=1)
                          svm = Pipeline([('vect', CountVectorizer()), ('tfidf',
                      TfidfTransformer()),
                                          ('svm', SVC(kernel="linear", C=1))])
                          svm = svm.fit(X_train, y_train)
                          ypred = svm.predict(X_test)
                          print("SVM metrics")
                          print(metrics.accuracy_score(y_test, ypred))
                          print(metrics.classification_report(y_test, ypred))
```

Appendix B: Originality Report



feedback studio

Joshua Mutisya Mutinda | Sentiment Analysis to Predict TV Show Popularity Case of Nation Media Group's NTV

Sentiment Analysis to Predict TV Show Popularity: Case of Nation Media Grou

Joshua Mutinda Mutisya

072189

Match Overview

16%

1	edoc.site Internet Source	2%
2	Submitted to Strathmor... Student Paper	1%
3	Submitted to Higher Ed... Student Paper	<1%
4	Submitted to University... Student Paper	<1%
5	www.itc.nl Internet Source	<1%
6	Submitted to University... Student Paper	<1%
7	endf tips	<1%

Page: 1 of 77 Word Count: 12052