Review Based Analysis of Mobile Phones

Project Team

Asjid Ahmed	p17-6124
Muhammad Suleman	p17-6147

Session 2017-2022

Supervised by

Mr. Zeshan Khan

Department of Computer Science

National University of Computer and Emerging Sciences
Peshawar, Pakistan

June, 2022

Student's Declaration

We declare that this project titled "*Review Based Analysis of Mobile Phones*", submitted as requirement for the award of degree of Bachelors in Computer Science, does not contain any material previously submitted for a degree in any university; and that to the best of our knowledge, it does not contain any materials previously published or written by another person except where due reference is made in the text.

We understand that the management of Department of Computer Science, National University of Computer and Emerging Sciences, has a zero tolerance policy towards plagiarism. Therefore, We, as authors of the above-mentioned thesis, solemnly declare that no portion of our thesis has been plagiarized and any material used in the thesis from other sources is properly referenced.

We further understand that if we are found guilty of any form of plagiarism in the thesis work even after graduation, the University reserves the right to revoke our BS degree.

Asjid Ahmed	Signature: ________________
Muhammad Suleman	Signature: ________________

Verified by Plagiarism Cell Officer
Dated:

Certificate of Approval

The Department of Computer Science, National University of Computer and Emerging Sciences, accepts this thesis titled *Review Based Analysis of Mobile Phones*, submitted by Asjid Ahmed (p17-6124), and Muhammad Suleman (p17-6147), in its current form, and it is satisfying the dissertation requirements for the award of Bachelors Degree in Computer Science.

Supervisor

Mr. Zeshan Khan Signature: ________________

Mr. Zeshan Khan

FYP Coordinator

National University of Computer and Emerging Sciences, Peshawar

Dr. Hafeez-Ur-Rehman

HoD of Department of Computer Science

National University of Computer and Emerging Sciences

Acknowledgements

We are thankful to Allah Almighty who is the most Merciful and Beneficent for giving us strength and courage to choose such challenging project and for completing this project with great dignity and grace. Lots of people played a significant role in the completion of this project. We firstly say thanks to Mr. Zeshan Khan , who motivated us to select this project. He encouraged and showed us the way the project can be implemented. He has been very kind, supportive and patient to us since the beginning of our project. His guidance and experience brought us on the right track whenever we faced any difficulty. We are highly thankful to all of our teachers who had been guiding us throughout our project work. Their knowledge, guidance, and training enabled us to carry out this development work more efficiently. We would like to thank all our friends and family for their support, motivation, and suggestions about the project.

Asjid Ahmed

Muhammad Suleman

Abstract

Review Based Analysis plays a very important role in mobile selection. Many techniques exist for the analysis of mobile phones. But feature based analysis and fake reviews detection requires understanding the context and the actual meaning of the words. Two reviews, conveying same message, can be written quite differently. The review analysis requires human intelligence and effort. We propose an automatic review analyzer application that analyzes and gives the top suggestions. The system based on the artificial intelligence will verify the reviews and allocate rating accordingly. In the end, the evaluation is done to prove that our approach gives much realistic behavior and can be used by the buyers as well as the sellers.

Contents

1 Preliminaries and Introduction **0**

1.1 Motivation 0

1.2 Objective 1

1.3 Scope: 1

1.4 Limitations: 1

2 Review of Literature **2**

2.1 Techniques 2

2.1.1 KNN 2

2.1.2 Naive Bayes 3

2.1.3 SVM(Support Vector Machine) 3

2.1.4 BagOfWords 3

2.1.5 Word2Vec 3

2.1.6 LDA(Topic Modelling) 3

2.1.7 TextBlob 4

2.2 Grading 5

2.2.1 Polarity Score 5

2.2.2 Single Feature 5

2.2.3 Multiple Features 5

2.3 Datasets 6

2.3.1 Online Datasets 6

2.3.2 FAST-NU Dataset 6

2.4 Conclusion: 8

3 Implementation **9**

3.1 Data . . . 9
3.2 Models . . . 11
3.2.1 Pre-Processing . . . 11
3.2.2 Model Training . . . 15
3.2.3 Evaluation . . . 16
3.2.4 Ensemble (Majority Voting) . . . 18
3.3 Methodology . . . 19
3.4 Analysis and Design . . . 20
3.5 Front-End . . . 23

4 Results **24**
4.1 Polarity Results . . . 24
4.1.1 Polarity Scores for Features . . . 24
4.2 Front-end Results . . . 26

5 Discussion **28**

6 Conclusions and Future Work **29**

References **33**

List of Figures

2.1 Polarity scaling 5
2.2 Pie Chart 7
2.3 Data-set Sample 7

3.1 Data Scraping 1 9
3.2 Data Scraping 2 10
3.3 Pre-processing 1 12
3.4 Pre-processing 2 13
3.5 Pre-processing 3 14
3.6 Training Model (Naive Bayes) 15
3.7 Training Model (SVM) 15
3.8 Naive Bayes Confusion Matrix 16
3.9 Naive Bayes Heat Map 16
3.10 SVM Confusion Matrix 17
3.11 SVM Heat Map 17
3.12 Majority Voting Ensemble 18
3.13 Methodology 19
3.14 Use Case Diagram 21
3.15 Flow Chart 22
3.16 Front-End 23

4.1 Polarity Score with features 24
4.2 Sum of polarity scores 25
4.3 Results for single feature 26
4.4 Results for multiple features 27

Chapter 1

Preliminaries and Introduction

Mobile phones have been one of the most essential components for us during the last decades. The fact about human psychology is that our beliefs are highly motivated and influenced by the thinking and evaluation of other people. When the consumer wants to buy a specific product like a mobile phone. He/She will review a similar product and then he will read reviews. However, the reviews may not always be truthfully provided and fake reviews generally exit. More than 80% of the data which are generated are in text format[4]. Analyzing these problems and working on different techniques had helped us to develop such a web application that provides sentiment analysis using machine learning algorithms which provide us to help get rid of the problem of selecting a mobile phone. It is predicted that, sometime in the near future, machine intelligence will surpass human intelligence, so by automating review based analysis we can get the best results in future[23].

1.1 Motivation

Users visit E-Commerce sites to shop products and also for the opinion of other buyers and users of products. Customer reviews are helping consumers to decide which products are good to buy and also manufacturing companies to understand the buying behavior of consumers, qualities and defects of their products.[19]. So we developed a web application that provides best mobile phone suggestions based on specific selected features. It will save time by automating analysis of reviews.

1.2 Objective

The objective is to develop a platform for selection of mobile phones . In order to achieve that goal, the primary objectives should be fulfilled.

- Polarity of reviews.
- Training model on real data.
- Specific features selection.
- Analysis of the specific features of the mobile phone specifications.
- Analyzing the reviews based on the specific features.
- Detection and removal of fake reviews on opinions.

1.3 Scope:

We are providing the way of selecting mobile phones and analysis based on reviews ranking. Our analysis will be on some selective features.

1.4 Limitations:

- Works only on selective features.
- Computationally expensive.
- Accuracy may vary according to data-set.
- Can't reach human accuracy[3].

Chapter 2

Review of Literature

2.1 Techniques

Most important thing to choose was the right technique on which we would train our model. After analyzing many literature we came up with following techniques used in review based analysis.

- KNN
- Naive Bayes
- SVM
- BagOfWords
- Word2Vec
- LDA
- TextBlob

2.1.1 KNN

KNN[21] is widely used because its simple to understand and easy to implement but as data-set grows the efficiency and speed of model is highly decreased. The features must be homogeneous and it cannot deal with outliers.

2.1.2 Naive Bayes

Naive Bayes is used for Finding the polarity for each review. It works well for high-dimensional data such as text classification. Independent features make Naive Bayes faster. It performs better than other classifiers in terms of accuracy.[22]

2.1.3 SVM(Support Vector Machine)

SVM[13] is mostly used for text classification. In the algorithm, each data will be plotted as a point in n-dimensional space with the value of each feature, n is the number of features.

2.1.4 BagOfWords

A most popular model which is widely used for sentiment analysis in NLP(Natural Language Processing)[7]. It is used for document classification where the frequency of occurrence of each word is considered as a feature for training a classifier.[15]

2.1.5 Word2Vec

The Word2Vec model is used to extract the notion of relatedness across words or products[9]. The Word2vec algorithm uses a NN(neural network) model to learn word associations from a large corpus of text. After training the model, it can detect synonyms of words or suggest additional words for a sentence completion.

2.1.6 LDA(Topic Modelling)

Topic modelling[20] is one of the most powerful techniques in text mining for data mining, latent data discovery, and finding relationships among data and text documents[12].

There are various methods for topic modelling; Latent Dirichlet Allocation (LDA) is one of the most popular in this field. Researchers have proposed various models based on the LDA in topic modelling. It is a generative statistical model that explains a set of observations through unobserved groups, and each group explains why some parts of the data are similar. It creates topics from similar words in a document.

2.1.7 TextBlob

TextBlob[16] is a Python library for processing textual data. It provides a simple API for diving into some very useful natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and polarity scores[10].

2.2 Grading

2.2.1 Polarity Score

The polarity score is the score of a sentence between -1 to +1 which is scaled into three classes in our case. Positive if the polarity is above 0, negative if below 0 and neutral on 0.[5]

```
def polarity_scaling(polarity):
    """
        Scaling of polarity into three classes i.e. negative, positive and neutral.
    """
    # 0  nutral
    if polarity  == 0:
        return 0
    # greater then 0  positive
    elif polarity > 0:
        return 1
    else:
    # less then 0  negative
        return -1
```

Figure 2.1: Polarity scaling

2.2.2 Single Feature

Top five maximum polarity scores are considered as the best suggestions.

$$\max(\text{Polarity Scores of selected feature})$$

2.2.3 Multiple Features

Average of top five maximum polarity scores are considered as the best suggestions.

$$\max(\ \text{Average}(\text{Polarity Scores of multiple selected features})\ )$$

2.3 Datasets

2.3.1 Online Datasets

- Amazon Date-set was based on star rating rating and reviews. Top 5 brands were compared in this data-set.[11]
- GSMArena
- Twitter This data-set was used for comparing only 5 brands on 4 features i.e battery life, screen quality, operating system and brand.[25]

2.3.2 FAST-NU Dataset

This data-set has multiple sources. It has mobile specifications and reviews of more than 6500+ mobiles of 100+ different brands.

Brands	No. of Mobiles	No. of Reviews
Apple	98	6,000+
Samsung	1,313	70,000+
Oppo	202	14,000+
Huawei	376	26,000+
Infinix	75	5,000+

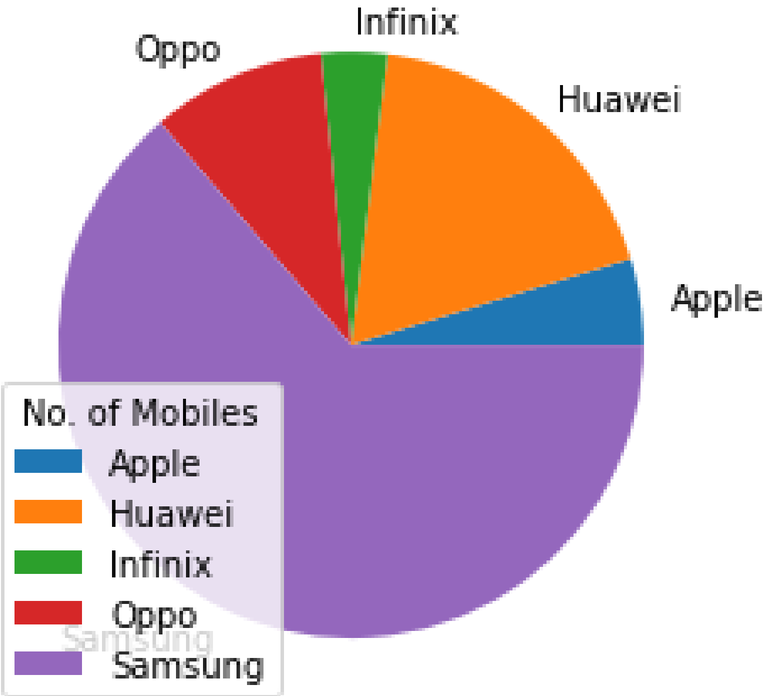


Figure 2.2: Pie Chart

Name	reviewer	review	Date	OS	cpu	Storage	Hits	Screen_siz	Screen_res	RAM
Apple iPhone 13 Pro Max	your moder	its has 1tb storage bruh the thing is most people use :	2 hours ag	iOS 15 up	Hexa-core	128GB/256	3 792 981	6.7"	1284x2778	6 GB RAM
Apple iPhone 13 Pro	John	John 12 Dec 2021Lol stop with the lies!	13-Dec-21	iOS 15 up	Hexa-core	128GB/256	1 533 621	6.1"	1170x2532	6 GB RAM
Apple iPhone 13	your moder	so SUS Beacuse iphone 13 needs an upgrade man like	2 hours ag	iOS 15 up	Hexa-core	128GB/256	1 847 081	6.1"	1170x2532	4 GB RAM
Apple iPhone 13 mini	John	Anonymous 07 Dec 2021ok but I wanted to buy a	12-Dec-21	iOS 15 up	Hexa-core	128GB/256	808 297 hi	5.4"	1080x2340	4 GB RAM
Apple iPad mini (2021)	Aj	Can somebody help me to buy the better one overall	5 hours ag	iPadOS 15	Hexa-core	64GB/256	472 867 hi	8.3"	1488x2266	4 GB RAM
Apple iPad 10.2 (2021)	Anonymous	If you want iPad with powerful performance donâ€™	09-Dec-21	iPadOS 15	Hexa-core	64GB/256	344 187 hi	10.2"	1620x2160	3 GB RAM
Apple Watch Edition Series 7	Anonymous	Inst this the aple watch 7?	08-Oct-21	watchOS 8	Dual-core	32GB stora	58 410 hits	1.9"	484x396 pi	
Apple Watch Series 7	Anonymous	Very poor battery life. Substandard charger. The fas	########	watchOS 8	Dual-core	32GB stora	106 039 hi	1.9"	484x396 pi	
Apple Watch Series 7 Aluminum	Anonymous	with this battery life is for junk only	16-Oct-21	watchOS 8	Dual-core	32GB stora	187 498 hi	1.9"	484x396 pi	
Apple iPad Pro 12.9 (2021)	Anonymous	To the Apple please improve ipad os . I know your m1	10-Dec-21	iPadOS 14.	Octa-core	128GB/256	844 567 hi	12.9"	2048x2732	8/16 GB RA
Apple iPad Pro 11 (2021)	Anonymous	Anonymous 18 Oct 2021do you want it to have a	########	iPadOS 14.	Octa-core	128GB/256	692 050 hi	11.0"	1668x2388	8/16 GB RA
Apple iPhone 12 Pro Max	Anonymous	NeoGul 04 Dec 2021Anyone here had they phone st	07-Dec-21	iOS 14.1 u	Hexa-core	128GB/256	11 276 43	6.7"	1284x2778	6 GB RAM
Apple iPhone 12 Pro	KC2FYA		23 hours a	iOS 14.1 u	Hexa-core	128GB/256	4 158 611	6.1"	1170x2532	6 GB RAM
Apple iPhone 12	Anonymous	Anonymous 12 Dec 2021mines at 85% after a year A	6 hours ag	iOS 14.1 u	Hexa-core	64GB/128	5 478 172	6.1"	1170x2532	4 GB RAM
Apple iPhone 12 mini	Anonymous	tri 29 Nov 2021i just upgrade to 12 mini from 11.. i	08-Dec-21	iOS 14.1 u	Hexa-core	64GB/128	3 843 010	5.4"	1080x2340	4 GB RAM
Apple iPad Air (2020)	Anonymous	Anonymous 05 Oct 2020just pleeeease stop crying	13-Dec-21	iPadOS 14.	Hexa-core	64GB/256	1 420 752	10.9"	1640x2360	4 GB RAM
Apple iPad 10.2 (2020)	Lolol	I'm still using it after 12 months and I only charge it	########	iPadOS 14	Hexa-core	32GB/128	1 079 356	10.2"	1620x2160	3 GB RAM
Apple Watch SE	Sid	Sid 02 Dec 2021I've been using the SE cellular varian	09-Dec-21	watchOS 7	Dual-core	32GB stora	479 599 hi	1.78"	448x368 pi	1 GB RAM
Apple Watch Series 6 Aluminum	EKispert	Anonymous 03 Jun 2021Series 6 watch cost 136 US I	25-Jul-21	watchOS 7	Dual-core	32GB stora	253 270 hi	1.78"	448x368 pi	1 GB RAM
Apple Watch Series 6	anan	can iphone series 6 support to version 6 s plus 12.5.5	25-Oct-21	watchOS 7	Dual-core	32GB stora	419 519 hi	1.78"	448x368 pi	1 GB RAM
Apple Watch Edition Series 6	range	Anonymous 05 May 2021The design has become so	########	watchOS 7	Dual-core	32GB stora	144 015 hi	1.78"	448x368 pi	1 GB RAM
Apple iPhone SE (2020)	kaveh	really good phone battery life is ok be consider batte	12-Dec-21	iOS 13 up	Hexa-core	64GB/128	6 864 903	4.7"	750x1334	3 GB RAM
Apple iPad Pro 12.9 (2020)	Takis	Sheesh 15 Jul 2021A12z is more powerful than a14N	########	iPadOS 13.	Octa-core	128GB/256	1 395 177	12.9"	2048x2732	6 GB RAM
Apple iPad Pro 11 (2020)	Devonte	Najib1312 23 Jul 2021Guys pls help me choose.	20-Aug-21	iPadOS 13.	Octa-core	128GB/256	1 269 773	11.0"	1668x2388	6 GB RAM
Apple iPhone 11 Pro Max	John	Anonymous 20 Oct 2021I personally thinks itâ€™s th	11-Dec-21	iOS 13 up	Hexa-core	64GB/256	9 853 722	6.5"	1242x2688	4 GB RAM
Apple iPhone 11 Pro	Anonymous	11pro is the best compect phone till now	########	iOS 13 up	Hexa-core	64GB/256	6 418 120	5.8"	1125x2436	4 GB RAM
Apple iPhone 11	ping koi balah angui	Anonymous 12 Dec 2021I'm on ios 14.4 version but	7 hours ag	iOS 13 up	Hexa-core	64GB/128	10 768 85	6.1"	828x1792	4 GB RAM

Figure 2.3: Data-set Sample

2.4 Conclusion:

We studied several techniques. One good option to choose the best technique was to use each technique for whole data but we concluded that by using ensemble learning we can achieve better results by combining results of SVM and Naive Bayes Classifier as the data is diverse[18].

Chapter 3

Implementation

3.1 Data

During the initial stage of the project, we scraped data from an online website GSMArena[2] which had reviews of almost 10,000 different mobiles with different brands including 100+ brands and 6500+ mobiles.

```python
def extract_smartphone_infos(network, smartphone):
    smartphone_dict = dict()
    smartphone = smartphone.find("a")
    url_smartphone = f"https://www.gsmarena.com/{str(smartphone['href'])}"
    logger.debug("url_smartphone : %s", url_smartphone)
    smartphone_dict["Link"] = url_smartphone
    smartphone_dict["Image"] = str(smartphone.find("img")["src"])
    soup_smartphone = network.get_soup(url_smartphone)
    smartphone_dict["Name"] = str(
        soup_smartphone.find("h1").find_all(text=True, recursive=False)[0]
    )
    logger.info(f"Processing model {smartphone_dict['Name']}")

    if soup_smartphone.select("td", {"class": "nfo"}):
        smartphone_dict["Release date"] = soup_smartphone.find(
            "span", {"data-spec": "released-hl"}
        ).text.strip()
        smartphone_dict["Weight"] = soup_smartphone.find(
            "span", {"data-spec": "body-hl"}
        ).text.strip()
        smartphone_dict["OS"] = soup_smartphone.find(
            "span", {"data-spec": "os-hl"}
        ).text.strip()
        smartphone_dict["Storage"] = soup_smartphone.find(
            "span", {"data-spec": "storage-hl"}
        ).text.strip()
        smartphone_dict["Fans"] = str(
            soup_smartphone.find("li", {"class": "help-fans"})
            .find("strong")
            .find(text=True)
        ).strip()
        smartphone_dict["Popularity"] = str(
            soup_smartphone.find("li", {"class": "help-popularity"}).find_all(
                text=True
            )[2]
        ).strip()
        smartphone_dict["Hits"] = str(
            soup_smartphone.find("li", {"class": "help-popularity"}).find_all(
                text=True
            )[4]
        ).strip()
        ecran = soup_smartphone.find("li", {"class": "help-display"}).find_all(
            text=True
        )
        if ecran:
            try:
                logger.debug("Screen : %s", ecran)
                smartphone_dict["Screen_size"] = str(ecran[2]).strip()
                smartphone_dict["Screen_resolution"] = str(ecran[3]).strip()
            except Exception as e:
                logger.debug("Screen : %s", e)
        ram = soup_smartphone.find("li", {"class": "help-expansion"}).find_all(
            text=True
        )
```

Figure 3.1: Data Scraping 1

```python
def extract_brand_name(brand):
    return brand["href"].rsplit("-", 1)[0]


def extract_brand_infos(network, brand):
    index_page = 1
    brand = brand["href"].rsplit("-", 1)
    brand_name = str(brand[0])
    brand_id = str(brand[1].split(".")[0])
    logger.info(f"Processing brand {brand_name}")
    url_brand_base = f"https://www.gsmarena.com/{brand_name}-f-{brand_id}-0"
    smartphone_list = []

    while True:
        url_brand_page = f"{url_brand_base}-p{index_page}.php"
        logger.debug(url_brand_page)
        index_page = index_page + 1
        soup_page = network.get_soup(url_brand_page)
        logger.debug(f"Page URL : {url_brand_page}")

        if soup_page.find("div", {"class": "section-body"}).select("li"):
            smartphones = soup_page.find(
                "div", {"class": "section-body"}
            ).find_all("li")
            soup_page.decompose()
            for smartphone in smartphones:
                smartphone_dict = extract_smartphone_infos(network, smartphone)
                smartphone_list.append(smartphone_dict)
        else:
            soup_page.decompose()
            logger.error(
                "%s : td class=section-body not found", url_brand_page
            )
            return smartphone_list
```

Figure 3.2: Data Scraping 2

For Data Scraping we used the python library BeautifulSoup[24]. Data for mobile phones consists of features, reviews' details and ratings.

We labeled our data into three categories.

Labels are:

- Positive: Polarity score above 0
- Neutral: Polarity score equals to 0
- Negative: Polarity score less than 0

3.2 Models

We trained different models that gave different accuracy measures but SVM and Naive Bayes gave us comparatively good results. In some cases SVM had better accuracy but in other cases Naive Bayes was better as the data had diversity so we ensemble the results of both models to generate our final output. We used Majority Voting Ensemble for that purpose.

3.2.1 Pre-Processing

For training any model on a data-set we have to do some pre-processing. Pre-processing involves data cleaning, tokenization, removing stop words, POS tagging and more[14]. In our case we performed multiple techniques to get the required data.

First of all we had removed special characters, single characters and multiple spaces as shown below.

```python
def pre_process(sentence):
    """
        i) Remove special characters.
        ii) Remove single characters.
        iii) Replace multiple spaces with single spaces.
    """


    # Remove all the special characters
    processed_sentence = re.sub(r'\W', ' ', str(sentence))

    # remove all single characters
    processed_sentence = re.sub(r'\s+[a-zA-Z]\s+', ' ', processed_sentence)

    # Remove single characters from the start
    processed_sentence= re.sub(r'\^[a-zA-Z]\s+', ' ', processed_sentence)

    # Substituting multiple spaces with single space
    processed_sentence= re.sub(r'\s+', ' ', processed_sentence, flags=re.I)

    # Removing prefixed 'b'
    processed_sentence = re.sub(r'^b\s+', '', processed_sentence)


    return processed_sentence
```

Figure 3.3: Pre-processing 1

- Setting lower case for all the text.
- Stemming each sentence.
- Tokenizing.
- Removing Stop-words.

```
def updated_sentence(sentence):
    """
        i) Lower cases the sentence.
        ii) Sentence stemming.
        iii) Tokenization of sentence.
        iv) Stop-words removal.
        v) Returns list after all above operations.
    """
    sentence = sentence.lower()
    new = " "
    p = PorterStemmer()
    processed = pre_process(sentence)
    stemmed = p.stem(processed)
    tokens = tokenize(stemmed.strip())
    lst=list(tokens)
    lst2 = []
    for i in lst:
        #using nltk dic for stopwords
        if  i not in stopwords :
            lst2.append(i)
    return (new.join(lst2))
```

Figure 3.4: Pre-processing 2

- Removing none keyword from text.
- Removing tabs and new lines.
- Polarity calculation.
- Creating review-polarity tuple.

```python
def reviews_list_polarity():
    """
        i) Removes none keyword from sentence.
        ii) Removes tabs and newlines from sentence.
        iii) Calculates polarity of each sentence.
        iv) Creates a tuple (review,polarity).
        v) Adds tuples to a dictionary.
    """

    individual_review = []
    for_reviews = []
    for_polarity = []
    updated_review_Dict = {}
    super_list_2 = []
    super_list = listing()
    for remove_none_type in super_list:
        if remove_none_type is not None:
            super_list_2.append(remove_none_type)
    for select_list in super_list_2:
        for i in select_list:
            news =i.strip()
            new1=news.split(".")

            lst_list = []
            lst_list_2 = []


            for j in new1:
                j = updated_sentence(j)
                temp =j.replace("\r\n" ,"")
                pol = TextBlob(temp).sentiment.polarity
                polarity=polarity_scaling(pol)
                tup = (temp ,polarity)
                lst_list.append(tup)
            for remove_tuple in lst_list:
                if remove_tuple != ('',0):
                    lst_list_2.append(remove_tuple)
            if lst_list_2 != []:
                for_reviews.append(lst_list_2[0][0])
                for_polarity.append(lst_list_2[0][1])
    updated_review_Dict["reviews"]=for_reviews
    updated_review_Dict["polarity"]=for_polarity
    return updated_review_Dict
```

Figure 3.5: Pre-processing 3

3.2.2 Model Training

We trained two different models on our data-set which includes SVM and Naive Bayes.

```
data=pd.DataFrame(reviews_list_polarity())
# spiting data for train data
x = data['reviews']
y = data['polarity']#text blob computed
```

```
vec = TfidfVectorizer()
matrix_X = vec.fit_transform(x)
```

```
train_x=matrix_X[:8000]
train_y=y[:8000]
NB_model = MultinomialNB()
NB_model.fit(train_x,train_y)
```

```
MultinomialNB()
```

```
pred_x=matrix_X[8000:]
pred_y=y[8000:]
pred=NB_model.predict(pred_x)
```

Figure 3.6: Training Model (Naive Bayes)

```
train_x_svm=matrix_X[:8000]
train_y_svm=y[:8000]
svm_model = SVC(kernel='linear')
svm_model.fit(train_x_svm,train_y_svm)
```

```
SVC(kernel='linear')
```

```
pred_x_svm=matrix_X[8000:]
pred_y_svm=y[8000:]
pred_svm=svm_model.predict(pred_x_svm)
```

Figure 3.7: Training Model (SVM)

3.2.3 Evaluation

For the evaluation of the models, confusion matrix[17] and heat map is used that will give us accuracy measures of both models.

```
Naive Bayes Matrix , accuracy and f1-score


NB confusion matrix :
 [[ 1477 11727  6750]
 [     2 75741  3624]
 [     1 15144 41689]]
              precision    recall  f1-score   support

          -1       1.00      0.07      0.14     19954
           0       0.74      0.95      0.83     79367
           1       0.80      0.73      0.77     56834

    accuracy                           0.76    156155
   macro avg       0.85      0.59      0.58    156155
weighted avg       0.79      0.76      0.72    156155

Accuracy of Model:  0.7614677724056226
```

Figure 3.8: Naive Bayes Confusion Matrix

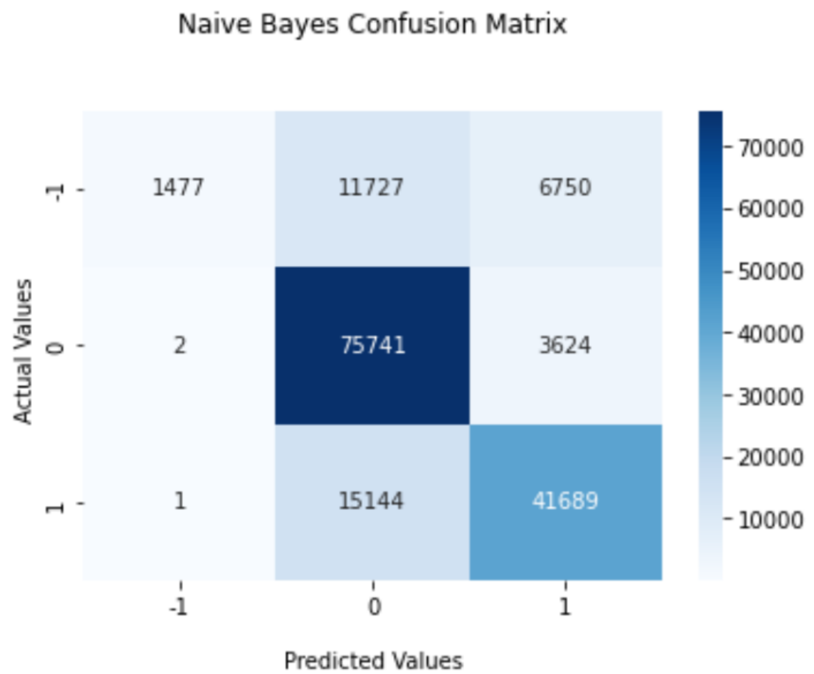


Figure 3.9: Naive Bayes Heat Map

```
SVM Matrix , accuracy and f1-score


SVm confusion matrix :
 [[12530  5935  1489]
 [  107 78837   423]
 [  831  7875 48128]]
              precision    recall  f1-score   support

          -1       1.00      0.07      0.14     19954
           0       0.74      0.95      0.83     79367
           1       0.80      0.73      0.77     56834

    accuracy                           0.76    156155
   macro avg       0.85      0.59      0.58    156155
weighted avg       0.79      0.76      0.72    156155

Accuracy of Model :  0.8933111331689667
```

Figure 3.10: SVM Confusion Matrix

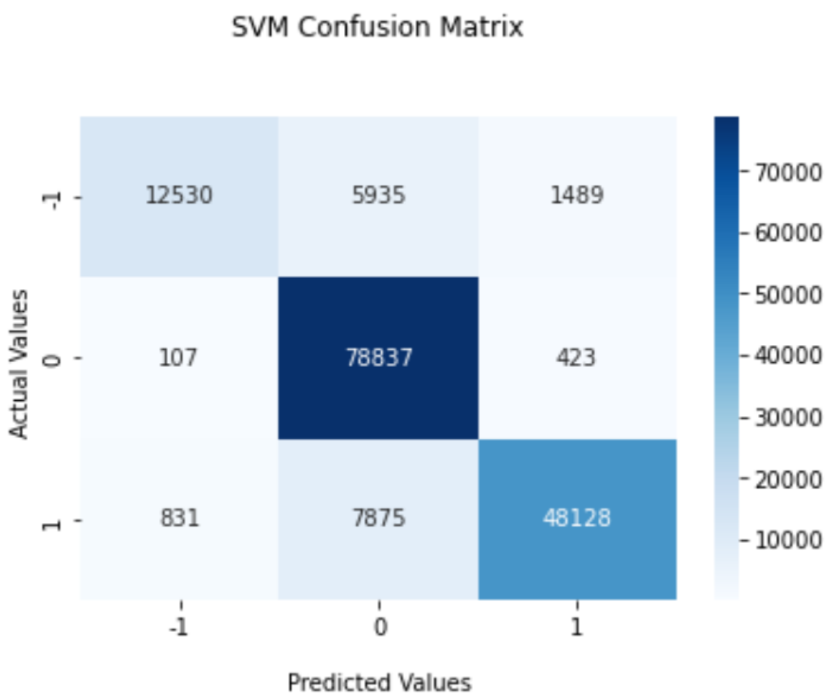


Figure 3.11: SVM Heat Map

3.2.4 Ensemble (Majority Voting)

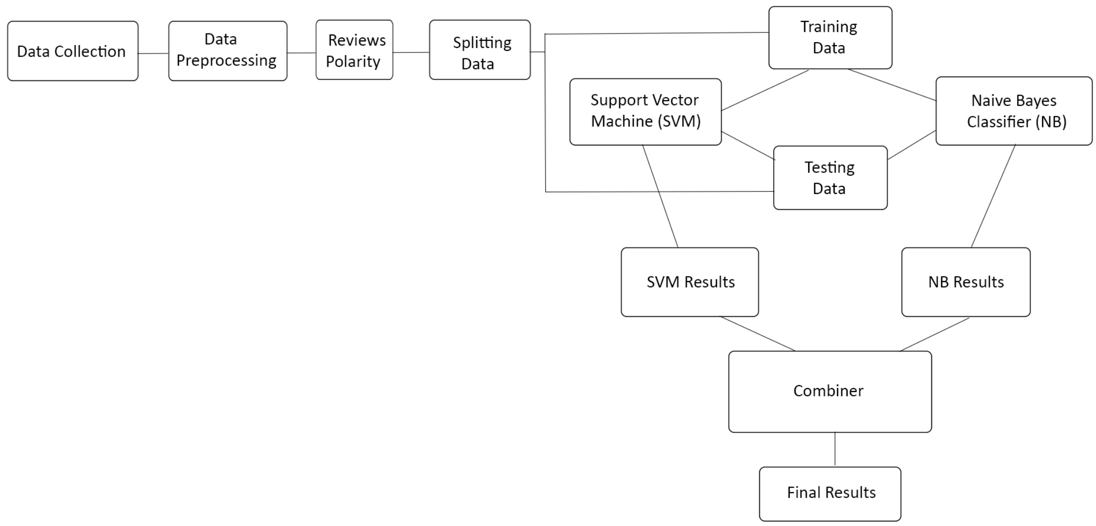
Using majority voting ensemble on the results of both models[1].

```
models = [('NB',NB_model),('svm',svm_model)]
ensemble = VotingClassifier(estimators=models)
results=model_selection.cross_val_score(ensemble, matrix_X[:8000], y[:8000])
print("Ensembler Majority voting ",results.mean(),results)
```

```
Ensembler Majority voting  0.8242499999999999 [0.7825    0.830625 0.8175    0.83625   0.854375]
```

Figure 3.12: Majority Voting Ensemble

3.3 Methodology



Architecture of Review Based Analysis of Mobile Phones

Figure 3.13: Methodology

3.4 Analysis and Design

Use Cases:

1) Register/Login

Name: Register/Login

Goal: User login or user registration.

Actors: User

Pre-conditions: For user login, user must be registered.

Basic Flow: User will register himself and after registration he will login.

Post conditions: After login user will select the features.

2) Provides Requirements

Name: Provides Requirements

Goal: User will provide specific requirements.

Actors: User

Pre-conditions: User will login.

Basic Flow: User will select the features from given list of features.

Post conditions: Selected features will be processed.

3) View Results:

Name: View Results

Goal: User will get results based on his requirements.

Actors: User

Pre-conditions: At least one feature must be selected.

Basic Flow: User will see the top results according to his search.

Post conditions: User will be redirected to feedback.

4) Submit Feedback:

Name: Submit Feedback

Goal: User will submit his experience with system.

Actors: User

Pre-conditions: View Results.

Basic Flow: User will enter his opinion in a text box and submit.

Post conditions: User will be redirected to main page.

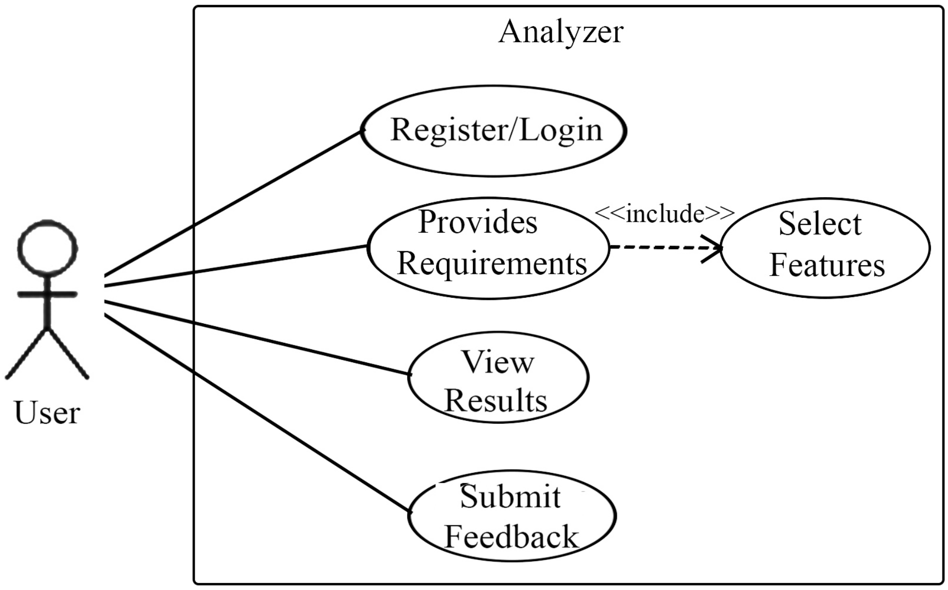


Figure 3.14: Use Case Diagram

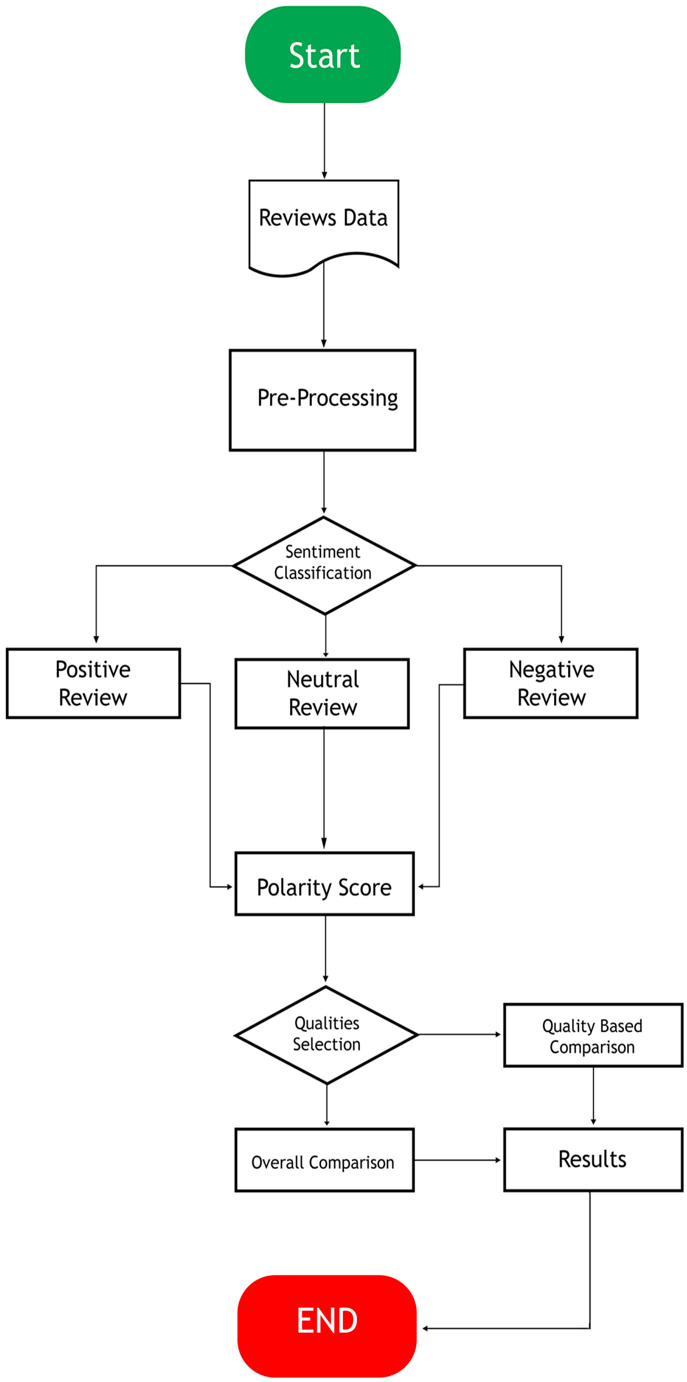


Figure 3.15: Flow Chart

3.5 Front-End

For front-end we used web framework React[8]. The front-end display of application is simple and consists of basic operations and search functionality.

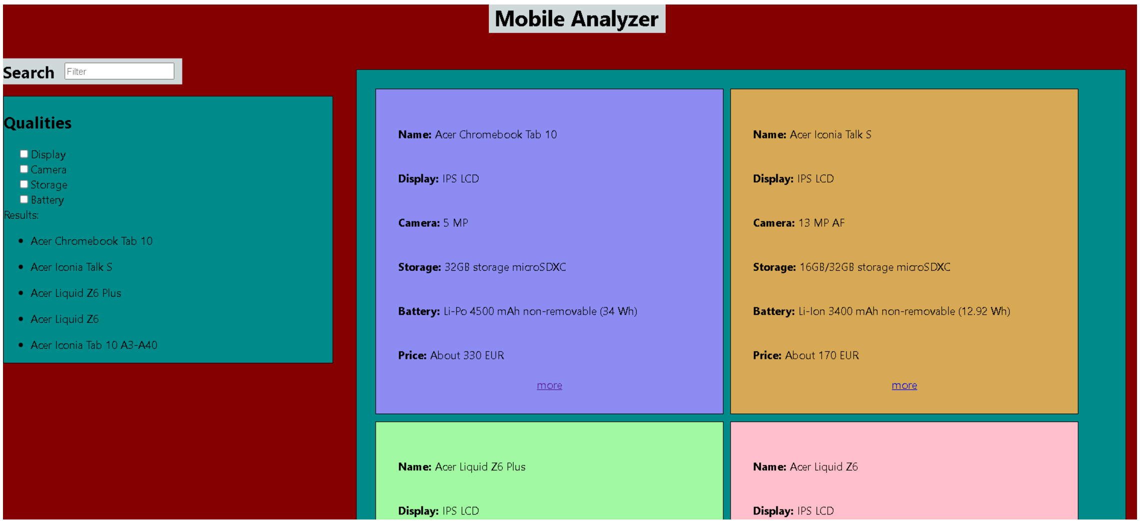


Figure 3.16: Front-End

Chapter 4

Results

4.1 Polarity Results

4.1.1 Polarity Scores for Features

Polarity score of each review according to features are calculated in positive(1), neutral(0) or negative(-1) and contained in a list. Each mobile phone has a list for polarity scores of all reviews.

	reviews	polarity	battery_polarity	display_polarity	storage_polarity	camera_polarity
0	[screen easily worthy device ipad users want t...	[1, 0, -1, 0, 0, 0, -1]	[0]	[1, 0, 0, -1, 0, 0]	[0, -1, -1]	[0]
1	[would pretty embarrasing use tablet purposes ...	[1, 1, 1, -1, -1, -1, 1, -1]	[1, 1]	[1]	[1, 1, 1, 1]	[-1, -1, -1, -1, 1, 0]
2	[indeed charging pot complet, bought phone dec...	[0, 1, -1, 0, 0, 0, 0, 1, 0, 0, 0]	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[-1, -1, 1, 1, 0, 1]	[1, 0]	[-1, 0]
3	[edge screen fell part part, excellent phone c...	[0, 1, 1]	[1]	[0, 0, 0, 1, 1]	[1, 1]	[0]
4	[nwith unique looks hdmi fhd screen speakers m...	[1, 1, 0, 1, 1]	[1, 1]	[1, 1, 1]	[1, 1, 1, 0]	[0]

Figure 4.1: Polarity Score with features

After getting all the polarities of the reviews, sum of all scores for each mobile phone is calculated.


```
{
    "Link": "https://www.gsmarena.com/acer_chromebook_tab_10-9139.php",
    "Name": "Acer Chromebook Tab 10",
    "Price": "69910",
    "polarity": "[1, 0, -1, 0, 0, 0, -1]",
    "battery_polarity": "[0]",
    "display_polarity": "[1, 0, 0, -1, 0, 0]",
    "storage_polarity": "[0, -1, -1]",
    "camera_polarity": "[0]",
    "battery_sum": "0",
    "display_sum": "0",
    "storage_sum": "-2",
    "camera_sum": "0"
},
```

```
{
    "Link": "https://www.gsmarena.com/acer_liquid_z6_plus-8305.php",
    "Name": "Acer Liquid Z6 Plus",
    "Price": "52962",
    "polarity": "[0, 1, -1, 0, 0, 0, 0, 1, 0, 0, 0]",
    "battery_polarity": "[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]",
    "display_polarity": "[-1, -1, 1, 1, 0, 1]",
    "storage_polarity": "[1, 0]",
    "camera_polarity": "[-1, 0]",
    "battery_sum": "0",
    "display_sum": "1",
    "storage_sum": "1",
    "camera_sum": "-1"
},
```

```
{
    "Link": "https://www.gsmarena.com/acer_iconia_talk_s-8306.php",
    "Name": "Acer Iconia Talk S",
    "Price": "36014",
    "polarity": "[1, 1, 1, -1, -1, -1, 1, -1]",
    "battery_polarity": "[1, 1]",
    "display_polarity": "[1]",
    "storage_polarity": "[1, 1, 1, 1]",
    "camera_polarity": "[-1, -1, -1, -1, 1, 0]",
    "battery_sum": "2",
    "display_sum": "1",
    "storage_sum": "4",
    "camera_sum": "-3"
},
```

```
{
    "Link": "https://www.gsmarena.com/acer_liquid_z6-8304.php",
    "Name": "Acer Liquid Z6",
    "Price": "25421",
    "polarity": "[0, 1, 1]",
    "battery_polarity": "[1]",
    "display_polarity": "[0, 0, 0, 1, 1]",
    "storage_polarity": "[1, 1]",
    "camera_polarity": "[0]",
    "battery_sum": "1",
    "display_sum": "2",
    "storage_sum": "2",
    "camera_sum": "0"
},
```


Figure 4.2: Sum of polarity scores

4.2 Front-end Results

Figure 4.3: Results for single feature

Figure 4.4: Results for multiple features

Final results are concluded on the basis of selected features. One or multiple features can be selected at the same time. The top five suggestions are calculated on top five polarity scores. In case of multiple features the average of polarity scores of the selected features are calculated and top five are presented as results.

Chapter 5

Discussion

The main purpose of our project was to achieve such an automated system that can reduce time spent on mobile phone analysis. For achieving that goal we trained multiple models and after analyzing the results we achieved our goal by getting the results. The results shown in chapter 4 are fairly according to our goals. In the results the top suggestions are calculated on the basis of selected features and their polarity scores based on the reviews dropped by different reviewers. The mobile phones with higher sum of polarity scores are considered to be the best suggestions.

Chapter 6

Conclusions and Future Work

In Future, it would be interesting to perform further analysis based on the brands and added qualities. We can also look at building a model to predict the helpfulness of the review or extracting the top products of each brand. Some future works and modifications are described below:

- Now a days posting reviews with fake accounts is very common so in extension of our project we can filter out the fake reviews that are posted by fake accounts[6].
- More features can be added to the features list for analysis.
- Self generated features can be added on the basis of reviews. Topic modelling has gained popularity recently, so it can be used to generate topics from different documents.
- Other smart products like smart watches, Tabs and laptops can be added to data-set for analysis with dynamic feature selection option.

Bibliography

[1] Alwi M Bamhdi, Iram Abrar, and Faheem Masoodi. An ensemble based approach for effective intrusion detection using majority voting. *Telkomnika (Telecommunication Computing Electronics and Control)*, 19(2):664–671, 2021.

[2] Abdessamad Benlahbib, Achraf Boumhidi, and El Habib Nfaoui. Mining online reviews to support customers'decision-making process in e-commerce platforms: A narrative literature review. *Journal of Organizational Computing and Electronic Commerce*, pages 1–29, 2022.

[3] Khalifa Chekima and Rayner Alfred. Sentiment analysis of malay social media text. In *International Conference on Computational Science and Technology*, pages 205–219. Springer, 2017.

[4] Minu Choudhary and Prashant Kumar Choudhary. Sentiment analysis of text reviewing algorithm using data mining. In *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 532–538. IEEE, 2018.

[5] Gulsen Demiroz, Berrin Yanikoglu, Dilek Tapucu, and Yucel Saygin. Learning domain-specific polarity lexicons. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 674–679. IEEE, 2012.

[6] Huaxun Deng, Linfeng Zhao, Ning Luo, Yuan Liu, Guibing Guo, Xingwei Wang, Zhenhua Tan, Shuang Wang, and Fucai Zhou. Semi-supervised learning based fake review detection. In *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pages 1278–1280. IEEE, 2017.

[7] Doaa Mohey El-Din. Enhancement bag-of-words model for solving the challenges of sentiment analysis. *International Journal of Advanced Computer Science and Applications*, 7(1), 2016.

[8] Elad Elrom. Learn react basic concepts. In *React and Libraries*, pages 1–19. Springer, 2021.

[9] Ruhama Even. Factors involved in linking representations of functions. *The Journal of Mathematical Behavior*, 17(1):105–121, 1998.

[10] J Praveen Gujjar and HR Prasanna Kumar. Sentiment analysis: Textblob for decision making. *International Journal of Scientific Research & Engineering Trends*, (7):1097–1099, 2021.

[11] Md Niaz Imtiaz and Md Khaled Ben Islam. Identifying significance of product features on customer satisfaction recognizing public sentiment polarity: Analysis of smart phone industry using machine-learning approaches. *Applied Artificial Intelligence*, 34(11):832–848, 2020.

[12] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211, 2019.

[13] Abdullahil Kafi, M Shaikh Ashikul Alam, Sayeed Bin Hossain, Siam Bin Awal, and Hossain Arif. Feature-based mobile phone rating using sentiment analysis and machine learning approaches. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–6. IEEE, 2019.

[14] Shailender Kumar, Nazir Ahmad Jailani, Abhinav Raj Singh, and Shubham Panchal. Sentiment analysis on online reviews using machine learning and nltk. In *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1183–1189. IEEE, 2022.

[15] Alesia Locker. "because the computer said so!": Can computational authorship analysis be trusted? *Journal of Language Works-Sprogvidenskabeligt Studentertidsskrift*, 4(1):23–37, 2019.

[16] Steven Loria et al. textblob documentation. *Release 0.15*, 2(8), 2018.

[17] JR Maria Navin and R Pankaja. Performance analysis of text classification algorithms using confusion matrix. *International Journal of Engineering and Technical Research (IJETR)*, 6(4):75–8, 2016.

[18] Aytuğ Onan, Serdar Korukoğlu, and Hasan Bulut. Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*, 57:232–247, 2016.

[19] P Venkata Rajeev and V Smrithi Rekha. Recommending products to customers using opinion mining of online product reviews and features. In *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, pages 1–5. IEEE, 2015.

[20] Toqir A Rana, Yu-N Cheah, and Sukumar Letchmunan. Topic modeling in sentiment analysis: A systematic review. *Journal of ICT Research & Applications*, 10(1), 2016.

[21] Punit Rathore, Jayesh Soni, Nagarajan Prabakar, Marimuthu Palaniswami, and Paolo Santi. Identifying groups of fake reviewers using a semisupervised approach. *IEEE Transactions on Computational Social Systems*, 2021.

[22] Mohammed AM Salem and Ashraf YA Maghari. Sentiment analysis of mobile phone products reviews using classification algorithms. In *2020 International Conference on Promising Electronic Technologies (ICPET)*, pages 84–88. IEEE, 2020.

[23] Cecile Schultz. Future work: a qualitative review. 01 2017.

[24] Shambhavi Sudarsan, Atharva Shirode, Ninad Chavan, and Rizwana Shaikh. E-commerce website with image search and price prediction. In *Evolutionary Computing and Mobile Sustainable Networks*, pages 823–837. Springer, 2022.

[25] Ye Yiran and Sangeet Srivastava. Aspect-based sentiment analysis on mobile phone reviews with lda. In *Proceedings of the 2019 4th International Conference on Machine Learning Technologies*, pages 101–105, 2019.