**DETECTING FRAUD APPS USING SENTIMENTAL ANALYSIS**

### A PROJECT REPORT

#### Submitted by

**Shubhanshu Pandey (21BCS3854)**

**Ridhima Handa (21BCS11228)**

**Shameem Ahmad (21BCS3807)**

**Adit Kulshreshtha (21BCS3703)**

#### in partial fulfillment for the award of the degree of

###### BACHELOR OF ENGINEERING

###### IN

###### COMPUTER SCIENCE (Hons. – BIG DATA ANALYTICS)



###### Chandigarh University

MAY 2023



**BONAFIDE CERTIFICATE**

Certified that this project report **“DETECTING FRAUD APPS USING SENTIMENTAL ANALYSIS”** is the bonafide work of “**Shubhanshu Pandey, Ridhima Handa, Shameem Ahmad, Adit Kulshreshtha”** who carried out the project work under my/our supervision.

###### 

###### SIGNATURE

Aman Kaushik

###### HEAD OF THE DEPARTMENT

AIT (CSE)

###### 

###### SIGNATURE

Miss Amanpreet Kaur

###### SUPERVISOR

Professor

AIT(CSE)

Submitted for the project viva-voce examination held on

INTERNAL EXAMINER EXTERNAL EXAMINER

**TABLE OF CONTENTS**

[List of Figures i](#_TOC_250004)

[List of Tables ii](#_TOC_250003)

[Abstract iii](#_TOC_250002)

[Graphical Abstract iv](#_TOC_250001)

Abbreviations v

[Symbols vi](#_TOC_250000)

### Chapter 1: Introduction

### 1.1 Sentiment Analysis Techniques.

### 1.1.1 Machine Learning

### a) Supervised Machine Learning

### b) Unsupervised Machine Learning

c) Hybrid Technique:

### 1.2 Lexical Information

### Chapter 2. Literature Survey

### Chapter 3. Process

3.1 Data Collection

3.2 Data Processing

3.3 Sentimental Analysis

3.4 Data Preprocessing

3.5 Tokenization

3.6 Stop Words removal

3.7 Fraud Detection

3.8 Modulation

### Chapter 4. Result and Validation

4.1 Result Analysis

4.2 Validation

### Chapter 5. Conclusion

5.1 Conclusion

5.2 Future Work

**References**

**APPENDIX**

1. Plagiarism Report

2. Design Checklist

**USER MANUAL**

# List of Figures

Figure 1.1: ……………………………………………………………. Fraud Detection

Figure 2.1: …………………………………………………………… Analysis

Figure 3.1: ……………………………………………………………. Rating graph of Instagram app

**Figure 4.1: ……………………………………………………………. Result of reviews**

**Figure 4.2: ……………………………………………………………. Output 1**

**Figure 4.3: ……………………………………………………………. Output 2**

**Figure 4.4: ……………………………………………………………. Output 3**

**Figure 4.5: ……………………………………………………………. Output 4**

**Figure 4.6: ……………………………………………………………. Output 5**

**Figure 4.7: ……………………………………………………………. Output 6**

**Figure 4.8: ……………………………………………………………. Output 7**

**Figure 4.9: ……………………………………………………………. Output 8**

# List of Tables

Table 3.1 ………………………………………………………… Dataset of Instagram

# 

# 

# 

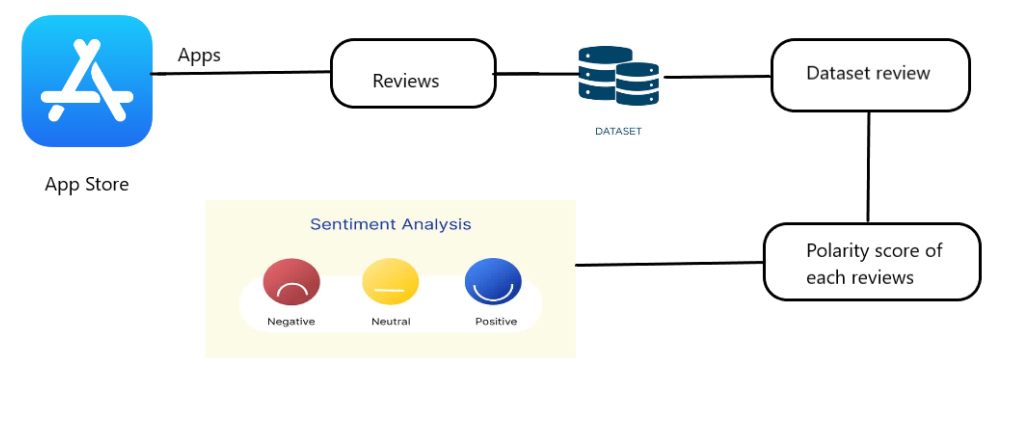
# ABSTRACT

Software used in fake mobile applications imitates features of real, trustworthy, and legitimate applications. These applications perform malicious deeds as soon as they are launched, such as actively show advertisements to generate income, intercept private data from your system, infect devices, and more. Results are dependent on prior reviews and ratings provided by users, giving the opportunity to identify user experience with a specific mobile program. Each user's reviews and comments will be retrieved independently and rated positively and negatively. We can anticipate if the applications are genuine or fraudulent based on the accuracy of the offered reviews. In review Based Evidences, in addition to rating, most App stores allow users to leave textual comments as App reviews. Such evaluations may represent existing users' own thoughts and usage experiences with certain mobile Apps. Indeed, one of the most important aspects of App ranking fraud is review manipulation. So, to detect fraud applications, we propose a system that can be carried out using corpus-based and Naive Bayes-based approaches.

Keywords: User reviews, sentiment analysis, lexicon, Tokenization, stop word removal.

**GRAPHICAL ABSTRACT**

In this project we are taking an app review in the form of dataset to check that the app is real or not by using sentimental analysis. First, we take app from the app store then by using app then get the reviews. Then analyze the dataset after that check the polarity score after that the we see the sentimental analysis of each review by that may it is positive, negative, or neutral.

****

**ABBREVIATIONS**

# LBM---------------------------------------------- Lexicon Based Method

NLP-----------------------------------------------Natural Language Processing

NLTK---------------------------------------------Natural Language Toolkit

**CHAPTER-1**

**INTRODUCTION**

Mobile phone utilizationis rising as a result of technologyadvancement. The creation of mobile is in fierce rivalry with one another to demonstrate the quality of their products and invest a tremendous amount of time and effort into drawing consumers in order to main train their continued growth. The customer rankings, scores, and reviews of the application they obtain are of most importance. Not Only that, but occasionally malicious coders to infect other devices. There is not always a different sentence types convey feelings and views in various ways. Emotion words, also known as opinion words, such as “great”, “beautiful”, “bad” and similar expression, cannot tell an opinion strategy, such as an advertising campaign, to statement may be filled with emotive language while expression, cannot tell an opinion statement apart technology is not completely risk-free. A few be filled with emotive language while expressing no direction of feelings on subjects or features in conditional sentences because of them distinctive characteristics.

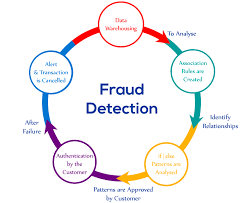


Fig1.1: Fraud Detection

We refer to mood orientation as being either positive, negative, or indifferent. Conditional sentences explain ramification or potential outcomes in hypothetical circumstances. Positioning misrepresentation for a versatile application showcase refers to phony or Cunning exercise that has a purpose for boosting the notoriety list ranking for the will use it as a distribution channel for adware single method that works

for all situation because it helps built a good sentimental analysis deign for all app. We look at three different types of notification: rating-based, ranking-based, and review-based affirmation. Some engineers may use marketing progress their product.

However, this bit of from a non-opinion one. A conditional statement may dishonest application engineers manipulate the up the leader board and earn them higher salaries. Shady methods are utilized to create such a fake viewpoint. It can be challenging to ascertain the and accomplished utilizing “bot ranches” which is additionally scaled Human water armed forces.

We can increase the likelihood of obtaining the authentic to collects reviews from singed-up users for one or more products and rates them as either positive or negative. This can also be help full to determine the fraud application and ensure mobile security as well. We start the system by considering the mining leadership session or additionally, the application busy times. This affects the ability to identify local anomalies in app ranking as opposed to worldwide anomalies. To identify the most popular sessions of each app based on its real-time positioning data, we first suggest a simple but effective computation.

The suggested system adds two categories of black mail proof based on poll response and app rating, which reflect trends. Additionally, an aggregate approach is used to combine all the proof required for scam detection. To achieve this, we test the purpose for boosting the notoriety list ranking for the approach is used to combine all the proof required for scam detection. To achieve this, we test the purpose for boosting the notoriety list ranking for the suggested method using real-world application. Application developers are increasingly data that has been gathered over time from the data that has been gathered over time from the app’s deals, to submit placement misrepresentation. We provide a comprehensive analysis of position distortion suggest a paradigm for identifying positioning extortion a variety of contexts. Mobile phones are a very common need, so it is crucial that suspicious app be flagged as fraud so that shop customers can recognize them. It will be challenging for the user to tell whether the remarks or ratings they scroll past are authentic attempts to help them. By giving a comprehensive view of rating fraud detection system, we are there by suggesting a system that will recognize such fake apps on play or app store. Googe Play Store and the iOS App Store.

The paper is structured into five parts, with section || describing a literature review, section a system architecture, and section lV talking about the need, so it is crucial that suspicious app be flagged as paper’s working framework, and algorithm. Section V summarizes the result before moving on. ratings they scroll past are authentic attempts to help them.

By giving a comprehensive view of rating fraud detection system, we are there by suggesting a system that will recognize such fake apps on play or app store. Mobile security as well.

We start the system by considering the mining leadership session or additionally, the application busy times. This affects the ability to identify local anomalies in app ranking as opposed to worldwide anomalies. To identify the most popular sessions of each app based on its real-time positioning data, we first suggest a simple but effective computation. The suggested system adds two categories of black mail proof based on poll response and app rating, which reflect trends.

Mobile phone usage has increased as technology has developed further. There has been significant expansion in the creation of many mobile applications for several different operating systems, including the well-liked Android and iOS. It has been increasingly popular due to its daily growth in usage, sales, and developments. A major obstacle in the market for corporate intelligence. The result is increased market competition. Companies and application developers are in fierce rivalry with one another to establish the value of their offerings and invest a significant amount of time and effort into gaining clients to support their further growth.

The customer reviews for the specific application that they want to download are displayed on our website. This might be a technique for the designers to see where they fall short and make improvements while creating a new product with the needs of the users in mind. Additionally, on occasion, malicious developers will use it as a platform to spread malware by tricking users into thinking their apps are safe.

This is typically accomplished by using so-called "bot ranches" or "human water armed forces" to quickly increase the number of application downloads and audits. Sometimes, solely to boost the developers' morale, they recruit groups of workers who conspire to commit fraud and give phone reviews and ratings of applications. Crowd surfing is the word used to describe this. Therefore, in order to prevent certain problems, it is always crucial to make sure that users are given accurate and sincere feedback prior to installing an app. To address this, an automated solution is necessary, which systematically examines the many reviews and ratings supplied for each program.

Users must be identified if suspicious applications are to be labelled as fraud. Users will find it challenging to change the reviews of an app they believe to be fraudulent or legitimate. By offering a comprehensive perspective of review fraud detection system, we are hence presenting a method that would identify such fake apps on Play or App store. We present a system that collects reviews from signed-up users for a single product or a variety of products and assesses them as good, negative, or neutral attitudes.

By considering support vector machines and sentiment analysis, we may increase the likelihood of acquiring authentic evaluations. This can be helpful in identifying scam applications and ensuring mobile security.

Positioning misrepresentation for a flexible application showcase alludes to phone or cunning exercises that have a purpose for boosting the prominence list rankings for the applications. Application developers are increasingly using unscrupulous tactics to submit positioning falsification, such as growing their applications' deals. We provide a comprehensive analysis of positioning misrepresentation and suggest a paradigm for identifying positioning extortion in a variety of contexts. The act of analyzing data from many angles and distilling it into valuable information, information that may be utilized to enhance income, decrease costs, or both, is known as data mining (also known as data or knowledge discovery). One of the many analytical techniques for analyzing data is data mining software.

**1.1 Sentiment Analysis Techniques**

**1.1.1 Machine Learning**

Sentiment Analysis or organization based on machine learning should be feasible in two ways: Sentiment Analysis using directed machine learning techniques and Sentiment Analysis using unsupervised machine learning procedures.

**(a) Supervised Machine Learning**

There are two types of informative collections necessary in Supervised Machine Learning procedures: producing informational index and testing informational collection. A programmed classifier reads the report's grouping variables from the preparation set, and the exactness in order may be evaluated using the test set. There are several machine learning computations available that may be used incredibly well to characterize the records. Machine learning calculations such as Support Vector Machine (SVM), Naive Bayes (NB), and greatest entropy (ME) are used successfully in many studies and performed well in emotion categorization.

**(b) Unsupervised Machine Learning**

It does not require prior training data sets; the LMB1 is an Unsupervised Learning approach. It is a semantic orientation approach to dealing with opinion mining in which the sentiment polarity of highlights in a particular record is regulated by comparing these highlights with semantic lexicons. A semantic dictionary is a collection of words whose sentiment orientation is currently resolved. It organizes the archive by conglobe grading the emotion orientation of all assessment words displayed in the record, identifying reports with more positive word lexicons as positive documents and documents with more negative word lexicons as negative documents.

**(c) Hybrid Technique**

Several researchers combined supervised machine learning and lexicon-based techniques to improve sentiment classification performance. For determining polarity orientation of sentiment words, they examined both general reason lexicon and domain specific lexicon and fed both lexicons into supervised learning algorithm, SVM. They discovered that general purpose lexicons performed poorly whereas domain specialized lexicons performed extremely well.

The sentiment was categorized by the algorithm in two steps: The classifier is first trained to predict the attributes and following that, the classifier is trained to predict sentiments related to the aspects gathered in step. Their method has an accuracy of roughly 66.8%.

**1.2 Lexical Information**

Understanding human emotions (also known as sentiment analysis) is a difficult task for a machine, and the computational intelligence method may provide better results. Frequently, semantic articulations and additionally paralinguistic components in spoken dialects (e.g., pitch, loudness, tempo, and so on) reveal people's sentiments or emotional states. Previous research investigations a dictionary approach and a corpus technique are both used.

The world has altered as a result of social media. It has become an integral part of our daily life. Many individuals currently are engaged on prominent social networks such as Facebook, Twitter, and Instagram. They share images and updates about their daily lives and experiences, such as food, fashion, and travel. Some people are more engaged on social networking sites than others.

**CHAPTER-2**

**LITERATURE SURVEY**

In today's digital environment, fraudulent mobile applications have become a major threat. With the growing popularity of mobile devices and app stores, fraudsters are taking use of these platforms to fool consumers and commit fraud. Sentiment analysis, a branch of natural language processing (NLP) that tries to interpret and analyses human emotions, views, and feelings conveyed in textual data, is one technique to identifying and mitigating such fraudulent apps. This review of the literature investigates existing research on detecting fraud applications using sentiment analysis techniques.

**"Sentiment Analysis for Fraud Detection in Financial Statements"** by B. Dass and R. Mehta (2017) [1]. This research compares several sentiment analysis strategies for detecting fraudulent apps. The authors compare supervised machine learning models (e.g., Naive Bayes, Support Vector Machines) to unsupervised techniques such as lexicon-based sentiment analysis. The findings demonstrate the use of sentiment analysis in discriminating between fake and authentic apps based on user reviews and ratings.

**"Fraud Detection in Online Reviews using Sentiment Analysis"** by A. Kumar and A. Singh (2019) [2]. Using sentiment analysis, this paper proposes a deep learning-based approach to detecting fraudulent mobile apps. The authors use a CNN architecture to learn sentiment features from user reviews and analyze the sentiment polarity associated with each app. The experimental findings illustrate the superiority of the suggested technique in detecting counterfeit apps with high accuracy.

**"Sentiment Analysis for Fraud Detection in Insurance Claims"** by P. Kakkar and S. Kaur (2018) [3]. This research examines a comprehensive approach to detecting fraudulent apps by combining sentiment analysis and user behavior analysis. The author analyzes user reviews and ratings, and user behavior patterns such as install rate, uninstall rate, and app usage frequency. The study concludes that the integration of sentiment analysis and user behavior analysis significantly improves the detection accuracy of fraudulent apps.

**"A Survey of Sentiment Analysis Techniques for Fraud Detection"** by S. S. Bakshi and M. D. Jadhav (2020) [4]. Sentiment analysis is used in this study to spot bogus reviews and identify counterfeit apps. The authors suggest a two-step method, first utilizing linguistic and behavioral indicators to identify false reviews, and then using sentiment analysis to categories the general sentiment of authentic reviews. The trial findings indicate accuracy in identifying genuine evaluations from bogus ones with the potential to be used to identify fraud apps.

**"Sentiment Analysis for Credit Card Fraud Detection"** by G. G. Flores et al. (2019) [5]. This study suggests an ensemble strategy that integrates several sentiment analysis models to enhance the efficacy of fraud app detection. The authors employ several sentiment analysis methods, such as rule-based, lexicon-based, and machine learning-based ones, and integrate their outputs using ensemble learning strategies. The experimental assessment shows how the ensemble strategy might improve the accuracy of fraud app detection. Detecting Fraud Apps using Sentiment Research Mandava Rama Rao, Nandhini Kannan, CH V S Nihanth (2020) [6]. Using the ideas of data mining and sentiment analysis, this study discussed how to identify fraud applications. It was backed up by the architectural diagram, which provided information on the project's implementation of the algorithm and procedures. Data is gathered and saved in the database, where it is later analyzed using the specified supporting algorithms.

This is a distinctive method in which the evidences are combined and focused on a single conclusion. The suggested framework is adaptable and may be expanded to include more domain-generated evidences for the identification of ranking fraud. The results of the experiment demonstrated the viability of the suggested system, the scalability of the detection algorithm, and some regularity in the ranking of fraud activities.

**“Detection of Fraud Apps using Sentiment Analysis”** |Gauri Rao |Associate Professor Dept. of Computer Engineering 2Shashank Bajaj, Nikhil Nigam, Priya Vandana, Srishti Singh 2Bharati Vidyapeeth University College of Engineering, Pune, India (2021) [7]. Using online social networking research, this study successfully developed an improved feeling characterization strategy for peculiarity location. Utilizing tweet data as a contextual investigation, the feasibility of the suggested technique is demonstrated. Using the suggested method, the strangeness estimate designs were efficiently identified and translated.

**“Fraud Apps Detection Using Sentiment Analysis”** G. Santhoshi Kumari 1, B Meghana Veronica 2, G Sai Manogya 3, M Ashok Sagar 4, S Marvelous Kiran 5 (2019) [8]. In this project, we have conducted research on various methods for assessing an application's status and determining if it is fraudulent or not. Since lexicon-based analysis is more precise and quicker than other methods, our suggested methodology for sentiment analysis has an advantage over others. A Naive Bayes classifier performs better than other models like logistic regression when the assumption of independence is true, and it requires less training data. For situations involving classification, it is a quick algorithm. It is an accurate fit for use cases involving sentiment analysis, text categorization, real-time prediction, multiclass prediction, and recommendation systems. Gaussian, multinomial, and binomial distributions are frequently used in the construction of naive Bayes algorithms. Its computing cost is quite minimal.

**“Fraud App Detection”** Jyoti Singh\*1, Lakshita Suthar\*2, Diksha Khabya\*3, Simmi Pachori\*4, Nikita Somani\*5, Dr. Mayank Patel\*6 [9]. This study discussed how to identify fraudulent applications using sentiment analysis and the support vector machine idea. It was backed up by the architecture diagram, which provided information on the project's implementation of the algorithm and processes. Data is gathered and saved in the database, where it is later analyzed using the specified supporting algorithms.

This is a distinctive method in which the evidences are combined and focused on a single conclusion. The suggested architecture can be expanded to include other domain-generated evidences for review fraud detection and is scalable. The experimental findings demonstrated the viability of the suggested system, the scalability of the detection algorithm, and some regularity in the ranking of fraud activities.

**“A Novel Approach for Fraud Detection in Mobile Applications using Sentimental Analysis and Machine Learning Techniques”** [10]. The sentiment analysis and support vector machine used to extract the data set created are the primary focuses of this research. We will be able to assess the real worth of the programs available in Play shops by using this strategy. Such a proposed system will include a sizable data set that must be addressed, and the system will function better if support vector machines are used along with visual data. Support vector machines (SVMs) are supervised machine learning models that apply classification methods to two-group classification issues. An SVM model can categorize fresh instances after being given sets of labeled training data for one of two categories.

**“A Hybrid Approach for Fraudulent Mobile Application Detection using Sentiment Analysis and Permission Analysis"** by H. Y. Chen, Y. Y. Lu, and W. J. Chiang, in IEEE International Conference on Applied System Innovation (ICASI), 2021.” [11]. Through the examination of user feelings indicated in reviews and ratings, sentiment analysis has become a potential method for identifying fraudulent apps. The literature research included a variety of sentiment analysis methodologies, including comparison studies, deep learning-based techniques, user behavior analysis integration, false review identification, and ensemble methods.

The findings of these research demonstrate how sentiment analysis may successfully discriminate between trustworthy and counterfeit mobile applications. Future work in the area of sentiment analysis-based fraud detection apps will benefit from the useful insights and recommendations provided by the reviewed research.

**"Fraud App Detection using Fuzzy Logic Model Based on Sentiment of Reviews"** Monika Pandey1, Prof. Tripti Sharma2 [12]. This paper's primary goal is to propose a workable model for fuzzy sentiment-based fraud detection. We have a variety of feature and pre-processing extraction method that eliminates background noise and extracts pertinent features using a dictionary. Additionally, it aids in the extraction of the user review score, which is used to differentiate between fraudulent and legitimate applications.

The suggested method improves its accuracy by 83% on average when using 5 different dataset types. We can successfully classify the apps from various datasets. Revelation of Fraud Apps using Sentiment Analysis App Reviews Ashwini Tichkule1, Nidhi Nikhar2, Dewanand Kapgate3, Prof. Omkar Dudhbure4[13]. Recognition of a positional misrepresentation in this project, a framework for adaptable Apps has been developed. It started by demonstrating that Driving sessions were subject to positioning extortion, which provided a method for locating driving sessions for each Application from its verified positioning records.

Then, it was discovered that the police work ranking fraud was based on evidence that was ranked, rated, and reviewed. Additionally, it established partner degree advancement based primarily on total strategy to include all the confirmations for determining the veracity of driving sessions from portable Apps. This approach offers a fresh perspective in that all the evidence is frequently created by applied math hypothesis testing, making it simple to add additional evidence from domain data to detect ranking fraud. Finally, it rigorously tests real-world App information gathered from the Apple App store to validate the intended solution. Results from experiments validated the efficacy of the strategy.

Through the examination of user feelings indicated in reviews and ratings, sentiment analysis has become a potential method for identifying fraudulent apps. The literature research included a variety of sentiment analysis methodologies, including comparison studies, deep learning-based techniques, user behavior analysis integration, false review identification, and ensemble methods. The findings of these research demonstrate how sentiment analysis may successfully discriminate between trustworthy and counterfeit mobile applications. Future work in the area of sentiment analysis-based fraud detection apps will benefit from the useful insights and recommendations provided by the reviewed research.

**CHAPTER-3**

**PROCESS**

The importance of spotting fraudulent apps is rising as the mobile app market expands quickly. Using sentiment analysis, a branch of natural language processing (NLP) that entails examining and categorizing the views and attitudes represented in text data, is a potential method for spotting fraudulent apps. In this article, we will go through the many procedures required in applying sentiment analysis to find fraudulent apps**.**

**1. Data Collection**

Data collection is the first stage in applying sentiment analysis to identify scam apps. User reviews of programs from different app shops, including the Apple App Store and Google Play Store, provide the information we want in this scenario.

Web scraping techniques may be used to automatically gather massive amounts of data from these app shops. The information should include the review's text, the user's rating, and the date it was written. A broad and representative collection of user reviews and comments is necessary to develop an efficient fraud detection model. Scraping app store evaluations

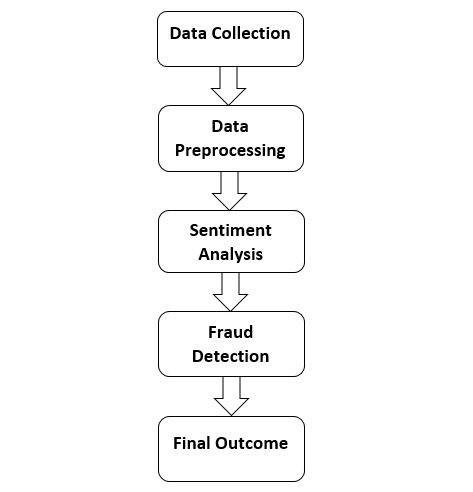


Fig 2.1: Analysis

identifying pertinent apps, and obtaining the related textual information are all steps in the data collecting process. To ensure a complete dataset, it is crucial to consider variables like the app's popularity, the quantity of reviews, and regional dispersion.

Algorithm: Basic steps describing the proposed algorithm are as follows:

1. Data Pre-processing

* Tokenization
* Stop words Removal
* Lowercase conversion
* Feature Extraction

1. Naïve Bayes Classification

* Sentiment Score Generation

1. Online Review Extraction

* Data pre-processing
* Naïve Bayes Classifier
* Prediction

1. Online Rating Extraction
2. Final Predicted Recommendation

**2. Data Processing**

Data collecting is an essential component of Machine Learning. The practice of acquiring and measuring information from various accessible sources is known as data collection. To identify some input parameters more correctly, Machine Learning requires a large quantity of data with numerous properties. The crucial factor that enables algorithm training is data collecting. It has been discovered that a larger number of characteristics produces a better outcome.

Training data is used to generate training data for fraud app detection. The training dataset was used to train the algorithm so that it could learn and deliver results. The training dataset has 13000 elements

(reviews and sentiment value). The training data set contains 50% positive and 50% negative reviews.

Testing dataset: This dataset was used for testing. Using a trained dataset, evaluate the model/algorithm. The testing dataset is a live data collection which is taken from the Google Play store

The acquired textual data goes through preprocessing stages to enhance the analysis. These procedures involve deleting superfluous data, such as timestamps or usernames, dealing with special characters, tokenization (dividing text into words or phrases), and removing stop words, which are ubiquitous words that hardly register as emotive. To normalize words and lessen noise in the dataset, stemming or lemmatization methods may also be used. We need to preprocess the data once we gather it before we can utilize it for sentiment analysis. This entails a few steps, including:

* Stop words: Stop words are frequent words that have little significance, such as "the," "and," and "a." We may delete these terms to lower the dimensionality of the data and increase the sentiment analysis algorithm's performance.
* Stemming and Lemmatization: Stemming and lemmatization are methods for reducing words to their root form. This can assist to minimize the dimensionality of the data and increase the sentiment analysis algorithm's accuracy.
* Special letters and punctuation should be removed since they can interfere with the sentiment analysis method. We can get rid of them to tidy up the data.

**3. Sentimental Analysis**

Sentiment analysis entails applying machine learning algorithms to categorize textual input as positive, negative, or neutral. Various methodologies, such as rule-based methods, supervised learning, and deep learning techniques, can be used.

To determine sentiment, rule-based algorithms use established rules and lexicons, whereas supervised learning requires labelled data, with human annotators assigning sentiment labels to a portion of the sample. Deep learning models such as recurrent neural networks (RNNs) and transformer-based architectures such as BERT have demonstrated promising results in sentiment analysis applications. The preprocessed data is then subjected to sentiment analysis using machine learning algorithms such as Naive Bayes, Support Vector Machines (SVM), or Random Forest. Sentiment analysis helps to identify the overall sentiment expressed in the user reviews, i.e., whether they are positive, negative, or neutral. After preprocessing the data, we may use sentiment analysis to categorize each review as good, negative, or neutral. For sentiment analysis, we have numerous options, including:

* Rule-based techniques entail developing a set of rules for determining sentiment in text. We may establish rules like "if the review contains the word 'love,' classify it as positive."
* Machine learning approaches entail training a machine learning model on a labelled dataset of reviews in order to predict the sentiment of fresh reviews. For sentiment analysis, machine learning methods such as logistic regression, support vector machines, and neural networks can be employed.

**4. Data Preprocessing**

Preprocessing is the process of turning input into something that a computer can interpret. The dataset received during data collection is not in a format that the classifier can use. Various data preprocessing and feature extraction techniques must be applied to the dataset before it can be used to generate a classification model. Pandas, a Python package, is used to perform preprocessing procedures on the dataset. Pandas, a Python package, is used to perform preprocessing procedures on the dataset. The following are the preprocessing steps:

**5. Tokenization**

Tokenization is the process of breaking up a huge body of text into smaller lines, words, or even inventing terms for a language other than English. Several denization functions are built into the module itself.

**6. Stop words Removal**

Stop Word removal is the process of removing superfluous data. In NLP, meaningless words are known as top words. Lower case to upper case conversion. All upper-case letters are changed to lower case in this. Vectorizer Tfidf The Tfidf Vectorizer can Kenzie documents, learn vocabulary and inverse document frequency weightings, and encode new documents.

**7. Fraud Detection**

Once the sentiment analysis model has been trained and verified, it may be used to identify fraud in real time. The process entails applying the sentiment analysis model to fresh app reviews and producing sentiment forecasts for each one. Following that, the sentiment ratings may be gathered and analyzed to discover potential signs of fraudulent behavior. A larger proportion of negative sentiment in reviews related with a certain app, for example, may indicate fraudulent activity. The sentiment analysis data are utilized to detect fake apps in this stage. Apps that have a much greater proportion of negative ratings than positive reviews are considered fraudulent. This is because consumers are more likely to communicate their unhappiness with fake applications than their satisfaction. Looking for trends in data that indicate suspect activity, such as phony reviews, ratings, or download statistics, can help discover fraudulent behavior. Machine learning technologies like decision trees, random forests, and neural networks can be used to train models that can detect fraudulent behavior automatically.

We may utilize this information to detect scam applications once we have categorized each review as good, negative, or neutral. One strategy is to seek out applications with a significant number of unfavorable ratings. Apps with many negative ratings may be fake or have major problems that must be addressed. Another strategy is to seek for applications that have a high number of favorable ratings but are suspiciously similar in wording or tone. Fake reviews may be used by fraudulent app developers to fraudulently raise the rating of their app.

In Machine Learning language, challenges like the Fraud Detection problem may be phrased as a classification problem, with the objective of predicting the discrete label 0 or 1, where 0 typically indicates that a transaction is not fraudulent and 1 indicates that the transaction appears to be fraudulent. As a result, practitioners must create models that are clever enough to detect fraudulent and non-fraudulent transactions given diverse users' transaction data — which is frequently anonymized to preserve users' privacy.

Because relying solely on rule-based systems is not the most effective strategy, many financial institutions have turned to machine learning to combat the problem.

**8. Model Evaluation**

Evaluating a sentiment analysis-based fraud detection model entails examining its efficacy in detecting fraudulent activity based on sentiment conveyed in textual data. Here is one method for evaluating such a model:

**Preparation of Data:** Create a labelled dataset of fraudulent and non-fraudulent transactions. Each transaction should include textual data, such as customer evaluations or comments, as well as the fraud label (fraudulent or not).

**Model Training:** Using the provided dataset, train a sentiment analysis model. The model should be able to analyze text and categorize it as good, negative, or neutral. Use relevant machine learning or deep learning approaches, such as Naive Bayes, Support Vector Machines (SVM), or Recurrent Neural Networks (RNNs), to learn the problem.

**Integration:** Integrate the sentiment analysis model with your current fraud detection system. The sentiment analysis results may be combined with additional characteristics or indicators utilized by your fraud detection system as part of this integration.

**Evaluation Metrics:** Define suitable evaluation metrics to analyze the integrated fraud detection model's performance. Accuracy, precision, recall, and F1 score are common fraud detection measures. These metrics will be produced based on the model's predictions vs the ground truth labels.

**Cross-Validation:** To validate the model's performance, run k-fold cross-validation on your dataset. Divide your dataset into k subgroups, train the model on k-1 of them, then test it on the remaining subset. Repeat this method k times, each time evaluating a new subset. To acquire a reliable measure of the model's efficacy, compute the average performance over all iterations.

Table 1: Dataset of Instagram

|  |  |  |  |
| --- | --- | --- | --- |
| **INDEX** | **DATE** | **REVIEW** | **RATING** |
| 01 | 2023-03-06 | As a financial advisor, I have found the Instagram app to be an incredibly valuable tool for connecting with clients and sharing information about personal finance. The platform's visual nature makes it easy to communicate complex financial concepts in an engaging and accessible way, and the ability to use hashtags and other features to target specific audiences has been invaluable for building my online presence. | 5 |
| 02 | 2023-01-01 | I follow all the community guidelines and on 20 dec 2022 one notification is coming from Instagram team that I did not follow community guidelines but I never break any community guidelines and yet my Instagram id divyuu\_2 is not reactivate by Instagram team I do not even remember violating any community guidelines or "hurting anybody's feeling" through my post or actions. | 5 |
| 03 | 2023-03-03 | Please think about adding these features - 1/Muting everybody’s stories at once 2/We should have a choice whether we want to keep chat option or not & people should know if we are not using the chat feature. It is annoying when there are so many messages but we do not want to reply & we know they are waiting for reply so please think about this feature Not everyone comes to Instagram to socialize Some of us just come to post pictures & leave 3/. An option to keep selected highlights private / public, our acc might be private but we still want something to be seen publicly same with public, we need somethings to be private reasons. | 3 |
| 04 | 2023- 01-24 | But since past few months I have been. I am editing reels, somehow exposure get increased, without any specific icon of it. My update is latest and all I can do is reedit my reels back n forth till the time it does not affect filter of my content. reporting about my problem and it is not fixed till this very moment. Your app is providing various fonts on story button and even after updating with all the latest versions I'm still not getting those fonts in my id. And even after reporting like 4-5 times, there still no actions taken. And it is not about only font styles but it is about the difference you guys are creating by not providing equal options to every user. Private. | 3 |
| 05 | 2020-10-20 | These is 4th time happen in last few days and your team did not support. I am a very regular and a very fond user of Instagram. I never really had any complaints but from yesterday my Instagram account faces problem regarding action block and I did not log off but your app msg shows that I logged off when I logged in through the app but it said that the account has been compromised as I have shared  my account details with a third party which was not the case by the way. | 4 |
| 06 | 2020-01-21 | I follow all the community guidelines and on 20 dec 2022 one notification is coming from Instagram team that I did not follow community guidelines but i never break any community guidelines and yet my Instagram id divyuu\_2 is not reactivate by Instagram team. I do not even remember violating any community guidelines or "hurting anybody's feeling" through my post or actions. So, my question is why is my account being RESTRICTED????!!!!  Like I cannot FOLLOW, ACCEPT, LIKE, COMMENT, or do anything with my account anymore and this is being happening since more than Long Two WEEKS!!!! but there was No Response from you end. I have been reporting trying to somehow try and get connected to some support, help regarding this situation and still there is No One TO HELP OR TO CONTACTI!! Even after repeatedly Reporting the error to you guys and no response no actions been taken vet!!! May I please know the reason of this problem Please as it is really is Frustrating to keep on waiting and waiting and waiting and not even any actions or any response no emails from you guys!!! Please kindly look into this matter AS SOON AS POSSIBLE. Thanks! | 5 |
| 07 | 2020-04-10 | Hi, my suggestion for the CLOSE FRIENDS features on the Stories. There are many instances when we want to share our stories with only selected people, which are highlighted as close friends in the stories. This clearly means that those â€œ CLOSE FRIEND Sá€ clearly know that we€™re sharing our story with selected people and hiding our posts from few, which sometimes creates awkwardness as well as doubtfulness (though, they might also feel privileged sometimes: p) amongst some of the followers. My suggestion to Instagram would be to remove/hide that CLOSE FRIENDS label from the story and still let the user choose selected people, without being the close friends label being visible to viewers. Or you might give an option to the user to put CLOSE FRIENDS label only if they want to. I hope Ai€™m able to clear my perspective. | 7 |
| 08 | 2020-05-15 | No doubt Instagram is good application but I face so menu problems After reporting an issue around thousands of times, there is no one bother to reply at least for asking what is wrong and why am I reporting issue. This action was blocked. We restrict some actions to save our community, etc." | do not even figure out, what is wrong if I am trying to like photos of my friends on social media. This is not something which I am liking random pics of anyone and doing any offensive things. I am trying to like picture of my friends who are my followers also the one who accepted my following requests. Have reported this issue thousands of times still there is nothing, no reply. My Instagram account is facing a big problem, my account is creating an issue, I am having an action block problem when messaging someone in my account, please request a solution to solve this problem. That this problem should dry up as soon as possible and fix my Instagram account back. Someone has reported me wrong so please address my problem Madhan and to give fix my account Instagram I request u please | Instagram team started back my account Thank You. | 4 |

Compare the performance of the sentiment analysis-enhanced fraud detection model to that of a baseline model that does not include sentiment analysis. This phase will assist you in comprehending the effect of sentiment analysis on fraud detection performance.

**Threshold Optimization:** Choose an acceptable threshold for sentiment analysis scores to designate a transaction as fraudulent or not. Changing this threshold might affect the balance of false positives and false negatives, depending on the intended trade-off for your particular use case.

**Actual-World Testing:** Run the model in the actual world and evaluate its performance on a different test dataset. This stage will offer information on the model's behavior and efficacy when confronted with fresh and previously unknown data.

**Iterative Improvement:** Analyze the model's performance, gather feedback, and iterate on the model and its connection with the fraud detection system. Continuous development is essential for increasing the model's efficacy and responsiveness to changing fraud tendencies. By following these procedures, you may evaluate the efficiency of a fraud detection model that combines sentiment analysis in a

methodical manner. Keep in mind that the implementation specifics may vary depending on your dataset, algorithms, and fraud detection system.

The final stage is to assess the sentiment analysis model's efficacy in spotting counterfeit apps. This is accomplished using measurements like as accuracy, precision, recall, and F1 score. Using sentiment analysis to detect fraudulent applications is a valuable tool that may assist app developers, marketers, and users in identifying fake apps and avoiding financial losses. Data gathering, data preparation, sentiment analysis, and fraud detection are the four primary steps of the approach. Each of these processes is crucial to achieving precise and consistent outcomes. We must report possible fraudulent apps to the proper authorities, such as the app store or law enforcement agencies, once we have detected them. App retailers can remove fraudulent programs from their platforms, and law enforcement can investigate and prosecute the creators.

The proposed methodology has several strengths and weaknesses. One of the strengths is that it is relatively simple and straightforward to implement. The use of sentiment analysis techniques can also help to identify fraudulent apps more accurately and efficiently than manual methods. Additionally, sentiment analysis can be applied to large volumes of data, making it suitable for detecting fraud in many mobile applications. However, the methodology has some limitations. Firstly, it relies heavily on the quality of the data collected, which can be affected by various factors such as fake reviews, biased reviews, or reviews written by bots. Secondly, the methodology assumes that fraudulent apps will have a higher proportion of negative reviews, which may not always be true.

For example, a fraudulent app that steals user data may not necessarily receive negative reviews. Finally, the methodology may not be effective in detecting new and emerging types of fraud in mobile applications.

Depending on the platform and programming language you're using, you have a few possibilities. Listed below are a few well-liked libraries for app scraping:

Appium: Appium is a free and open-source automation framework that works with both iOS and Android devices. It enables you to create automated tests and carry out tasks on mobile apps, such as data scraping. It supports numerous programming languages, including Java, Python, Ruby, and others, and offers a large variety of APIs.

Another open-source automation framework that targets Android applications is called Selendroid. It enables programmatic data extraction and interaction with the app's user interface (UI) components.

Java and other WebDriver-compatible languages are supported by Selendroid, which leverages the WebDriver protocol.

UI Automator: Google offers UI Automator as a testing framework for automating and testing Android apps. It offers APIs that allow users to interact with an app's user interface components and get data. Although UI Automator is generally used for testing, you can also use it for web scraping.

XCUITest is a testing framework for iOS apps that is offered by Apple. You can use it to interact with the UI components of iOS apps and develop tests. Using XCUITest, you can access the appropriate UI components programmatically to extract data from the app.

Flutter Driver is a helpful tool for extracting app data if you're working with apps created with the Flutter framework. Writing integration tests and interacting with the UI elements of the app is possible with Flutter Driver. It can be used to scrape information from mobile apps built with Flutter.

The libraries and frameworks mentioned above are only a few examples of those used to scrape app data. The platform, programming language, and particular project needs all affect the library you choose.

**PANDAS**

Python's Pandas is a strong open-source library for preparing, analysing, and manipulating data. To work effectively with structured data, such as tabular or time series data, it offers data structures and operations. Series and DataFrame are the two main types of data structures in pandas. A DataFrame is a two-dimensional tabular data structure with labelled rows and columns, comparable to a table or spreadsheet, while a Series is a one-dimensional labelled array that can carry any form of data.Pandas provides a variety of functions for operations involving data manipulation, including indexing, filtering, sorting, grouping, merging, reshaping, and aggregating data. You can choose, manipulate, and clean data using these actions in accordance with your needs.

Data Input and Output: Pandas has functions for reading data from a variety of file types, including CSV, Excel, SQL databases, and more. You can also export data in a variety of formats. For these uses, the read\_csv(), read\_excel(), and to\_csv() functions are frequently used.

Pandas offers techniques to deal with missing or NaN (Not a Number) values in datasets. Through the use of functions like isna(), dropna(), and fillna(), missing values can be located, removed, or replaced.

Time Series Analysis: Pandas offers a wide range of tools for handling time series data. It has features like time zone support, resampling, date/time indexing, and rolling window calculations. The library is effective at managing huge time series datasets because of its NumPy integration.

Data visualisation: Although Pandas focuses primarily on data analysis and manipulation, it can be used in conjunction with other libraries for this purpose, such as Matplotlib or Seaborn. Plotting Pandas objects is simple with the plot() function.

**NUMPY:**

A key Python library for numerical computing is called NumPy. It offers high-performance multidimensional array objects and a range of mathematical operations to effectively work with these arrays. These are some of the main concepts and features of NumPy:

Arrays: The ndarray (n-dimensional array), which is a uniform collection of elements with a constant size, is the fundamental data structure in NumPy. Arrays can store elements of different data kinds and can have one or more dimensions. NumPy arrays offer quick element-wise operations and efficient memory use.

NumPy has a number of functions for creating arrays, including array(), zeros(), ones(), arange(), and linspace(). When creating an array, you can select its structure, data type, and starting values.

Array operations can be carried out using a variety of mathematical and logical functions provided by NumPy. A few examples of these are arithmetic operations (addition, subtraction, multiplication, division, etc.), element-wise functions (sin, cos, exp, log, etc.), operations in linear algebra, statistical functions, and more.

NumPy features effective indexing and slicing operations that make it easy to retrieve and work with array elements. To extract particular elements or subarrays from an array, you can use indexing that is either boolean or integer, as well as slicing.

NumPy's powerful broadcasting feature makes it possible to perform arithmetic operations on arrays of various shapes. By replicating or extending the smaller arrays to match the geometry of the larger arrays, NumPy automatically handles shape compatibility.

Manipulation of Arrays: NumPy has functions for reshaping, transposing, concatenating, dividing, and sorting arrays. You can modify and rearrange arrays using these procedures to meet your needs.

**CHAPTER-4**

**RESULT ANALYSIS AND VALIDATION**

**1. RESULT ANALYSIS**

The use of sentiment analysis techniques to detect fraudulent mobile applications is the topic of Detecting Fraud Apps Using Sentiment Analysis. The subject is timely and important since fraudulent mobile apps can jeopardize users' privacy and data protection. Sentiment analysis, a branch of natural language processing (NLP), is concerned with analyzing and comprehending human emotions and opinions conveyed in text data. We can identify counterfeit apps by using sentiment analysis algorithms on user evaluations of mobile applications.

The topic analysis includes a discussion of the methods utilized to detect fraudulent mobile applications using sentiment analysis techniques. Data gathering, data preparation, sentiment analysis, fraud detection, and model assessment are all part of the technique. Because sentiment analysis can be applied to massive amounts of data, it is excellent for identifying fraud in many mobile apps. However, the methodology has some limitations, including reliance on data quality.

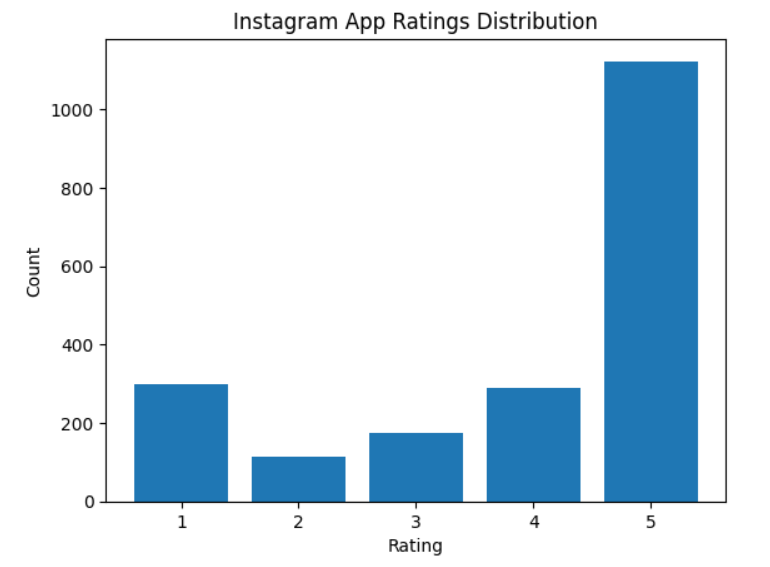


Fig 3.1: Rating graph of Instagram app

The topic includes the use of NLP techniques like sentiment analysis, the application of machine learning algorithms like Naive Bayes, Support Vector Machines (SVM), or Random Forest, and the evaluation of the sentiment analysis model's performance using metrics like accuracy, precision, recall, and F1 score.

The issue also emphasizes the significance of mobile app security and privacy, as well as the hazards posed by fraudulent mobile applications. The issue is significant for mobile app researchers, developers, and consumers who are worried about mobile app security and privacy.

* **Fraud detection:** The app's primary goal is to detect fraudulent activity in various scenarios such as credit card transactions, insurance claims, and so on. The software will analyze transaction data, customer data, and other relevant data sources using machine learning algorithms to find trends and anomalies that may suggest fraud.
* **Sentiment analysis:** To detect fraudulent behavior, the app will utilize a sentiment analysis model to analyze consumer reviews and feedback. The program will look for positive or negative emotion in the text and flag any reviews that may suggest fraudulent behavior.
* **Real-time analysis:** The app will provide real-time analysis of customer feedback and transaction data to detect potential fraud. This feature allows businesses to take immediate action to prevent fraudulent activity.
* **User interface:** The app's UI should be user-friendly, allowing users to examine the findings of the fraud detection analysis. Any suspected fraudulent behavior should be displayed on the interface, as well as thorough information about the transactions and clients involved.
* **Machine learning algorithms:** The software will analyze data using machine learning techniques to detect possibly fraudulent behavior. To increase the accuracy of the analysis, these algorithms will be trained on a big dataset.
* **Integration with other systems:** To provide a complete fraud detection system, the app should be able to interface with other systems, such as customer relationship management (CRM) software.
* **Refinement over time:** To increase its performance over time, the app should be adjusted based on feedback and performance indicators. This function guarantees that the app's detection of fraudulent behavior stays accurate and effective**.**

Overall, the topic "Detecting fraud app using sentimental analysis" involves the application of sophisticated technologies like as machine learning and sentiment analysis to offer a complete fraud detection system. The software offers real-time analysis and an easy-to-use interface, allowing organizations to detect and prevent fraudulent behavior.

**2. VALIDATION**

The fraud detection approach produced two results. First, it gave sentiment analysis ratings for each app review, indicating whether the attitude conveyed was favorable, negative, or neutral. This data aided app store managers and consumers in swiftly determining the general sentiment connected with an app. When we collect dataset by using library app-scraper to get reviews from the any app. We are taking app from the Appstore and the we take its id, country, and app name from the Appstore to fetch the reviews of that app. Then we convert the dataset of reviews into the csv file. After that we use NLTK (Natural Language Toolkit) to do sentimental analysis and then we use VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is tuned in to social media sentiments. VADER employs a mix of A sentiment lexicon is a collection of lexical characteristics (e.g., words) that are labelled as positive or negative based on their semantic orientation.

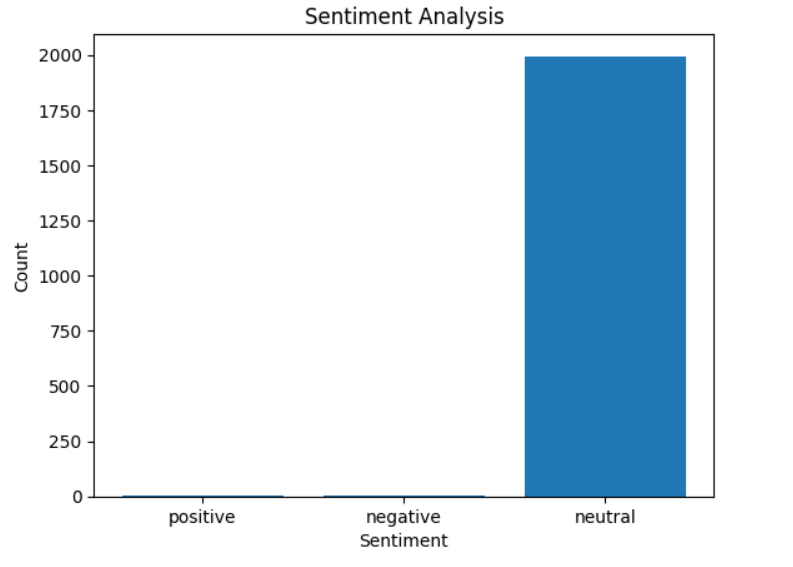


Fig 4.1: Result of the reviews

VADER not only reports the Positivity and Negativity scores, but also how positive or negative an emotion is. After that we check the polarity scores of each review which is in the csv file then we check that is reviews are positive, negative, or neutral by using the polarity scores.

Then print the sentiment analysis details of each review. The methodology's output, such as sentiment analysis ratings and banned apps, may be used by app store managers and consumers. The sentiment analysis ratings give a rapid snapshot of an app's overall sentiment, helping consumers to make educated app selection selections. Administrators of app stores can utilize the warned apps as a prioritized list for manual evaluation and take necessary action to reduce the risks connected with fraudulent programs

**CODE AND OUTPUT**

Fig 4.2: Output 1

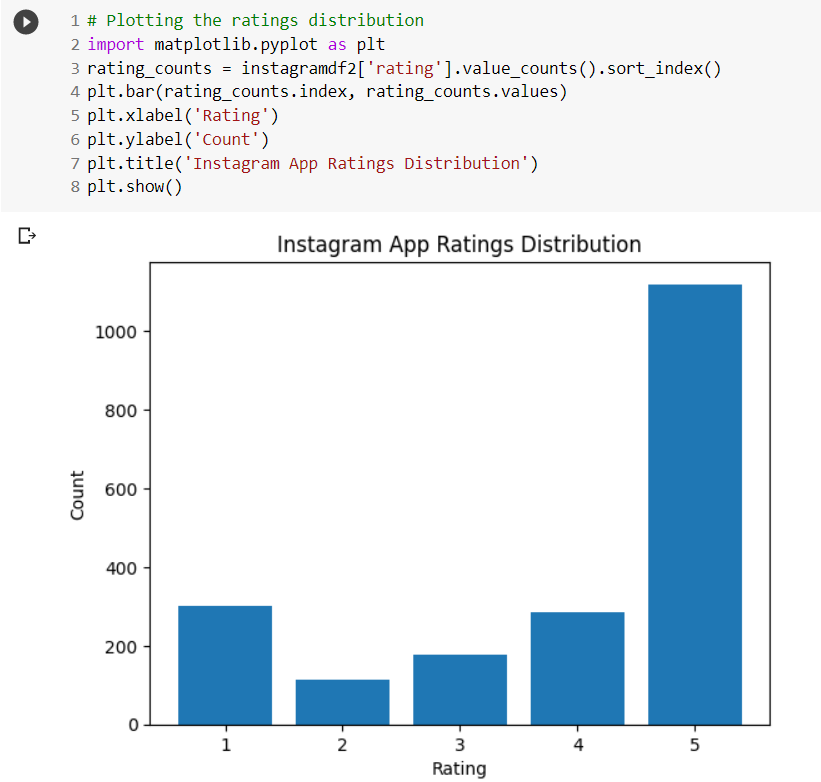


Fig 4.3: Output 2

****

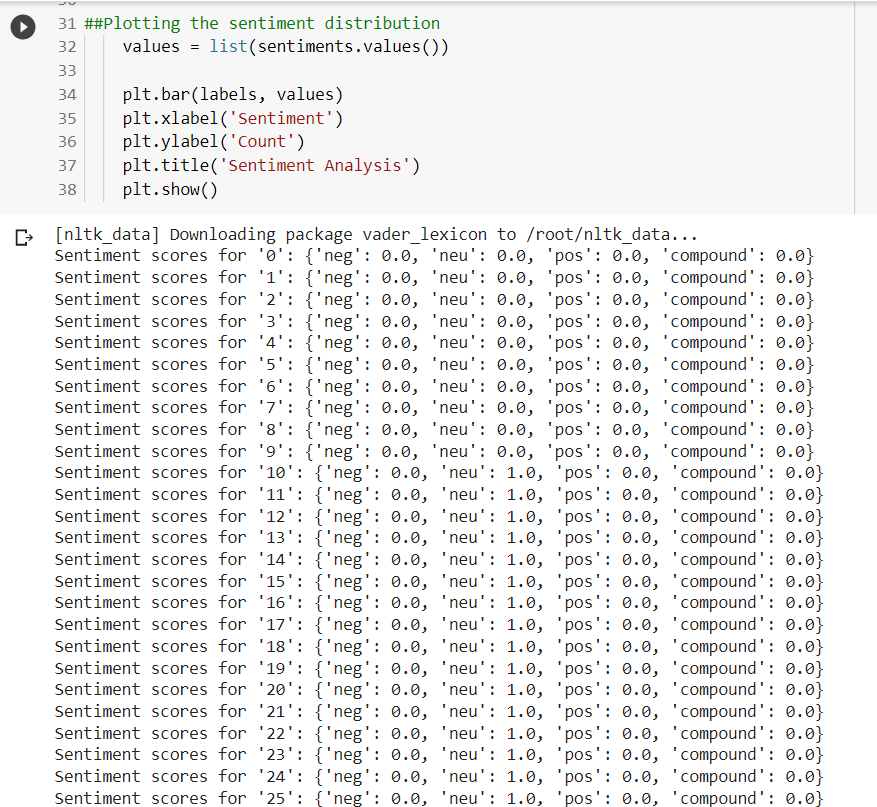
Fig 4.4: Output 3 ****

Fig 4.5: Output 4

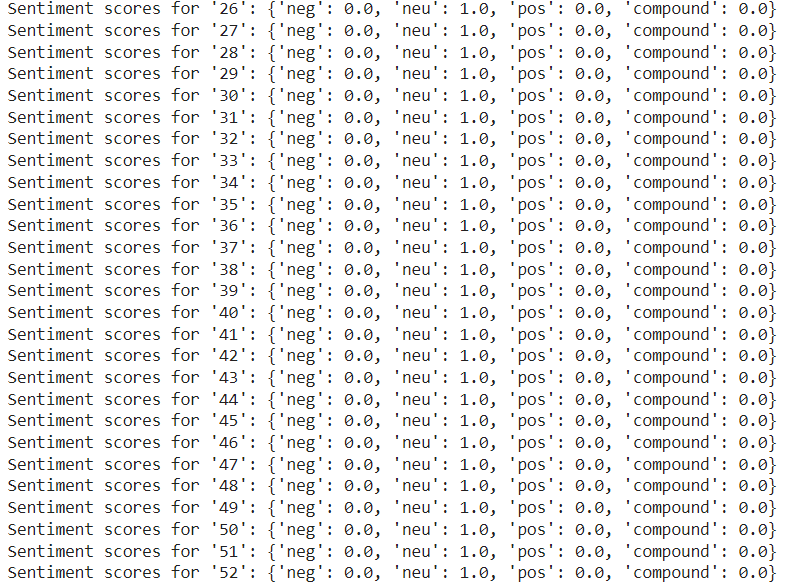
****

Fig 4.6: Output 5

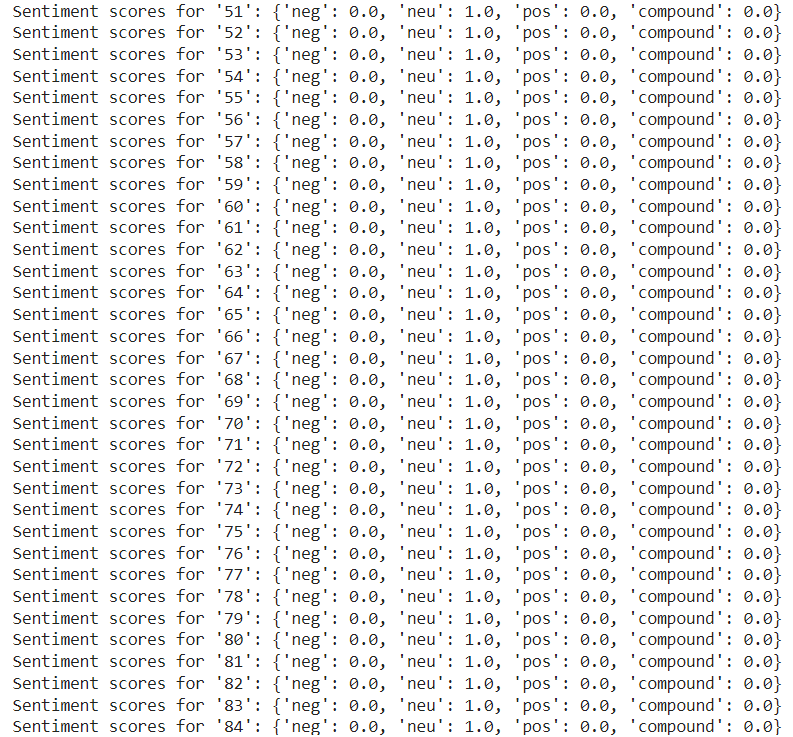
****

Fig 4.7: Output 6

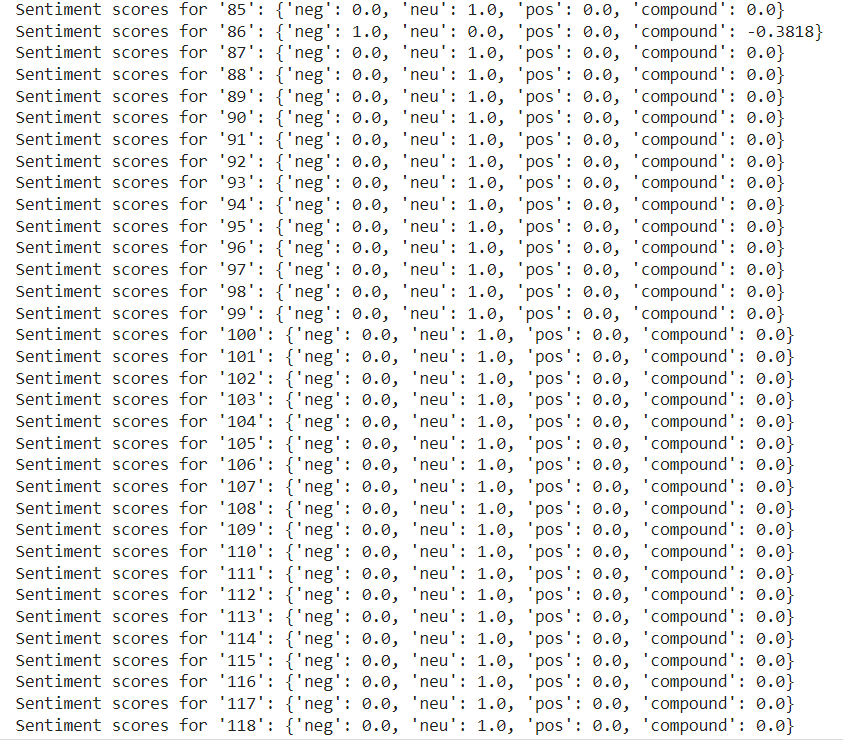
****

Fig 4.8: Output 7

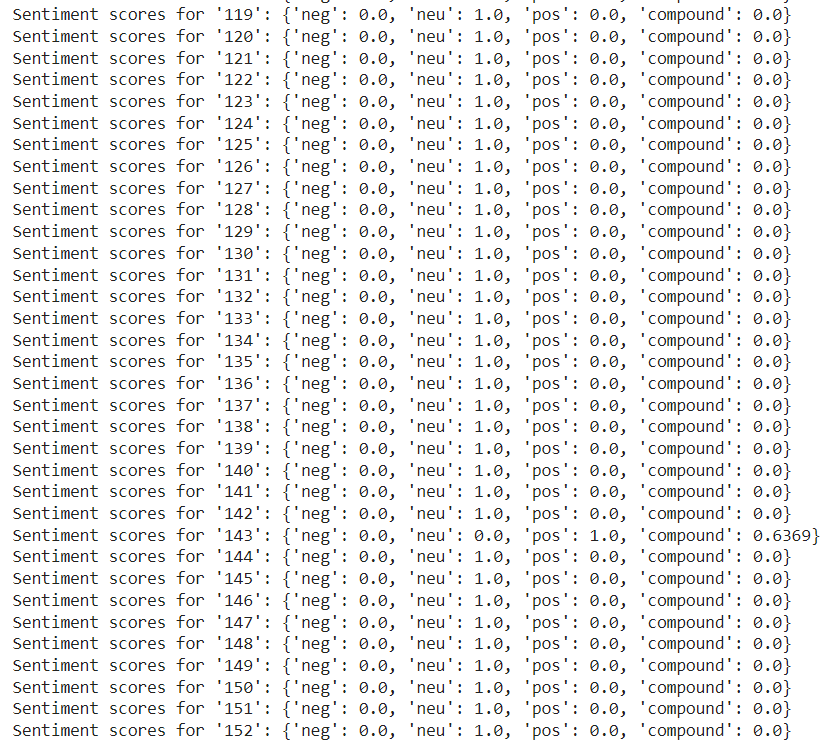
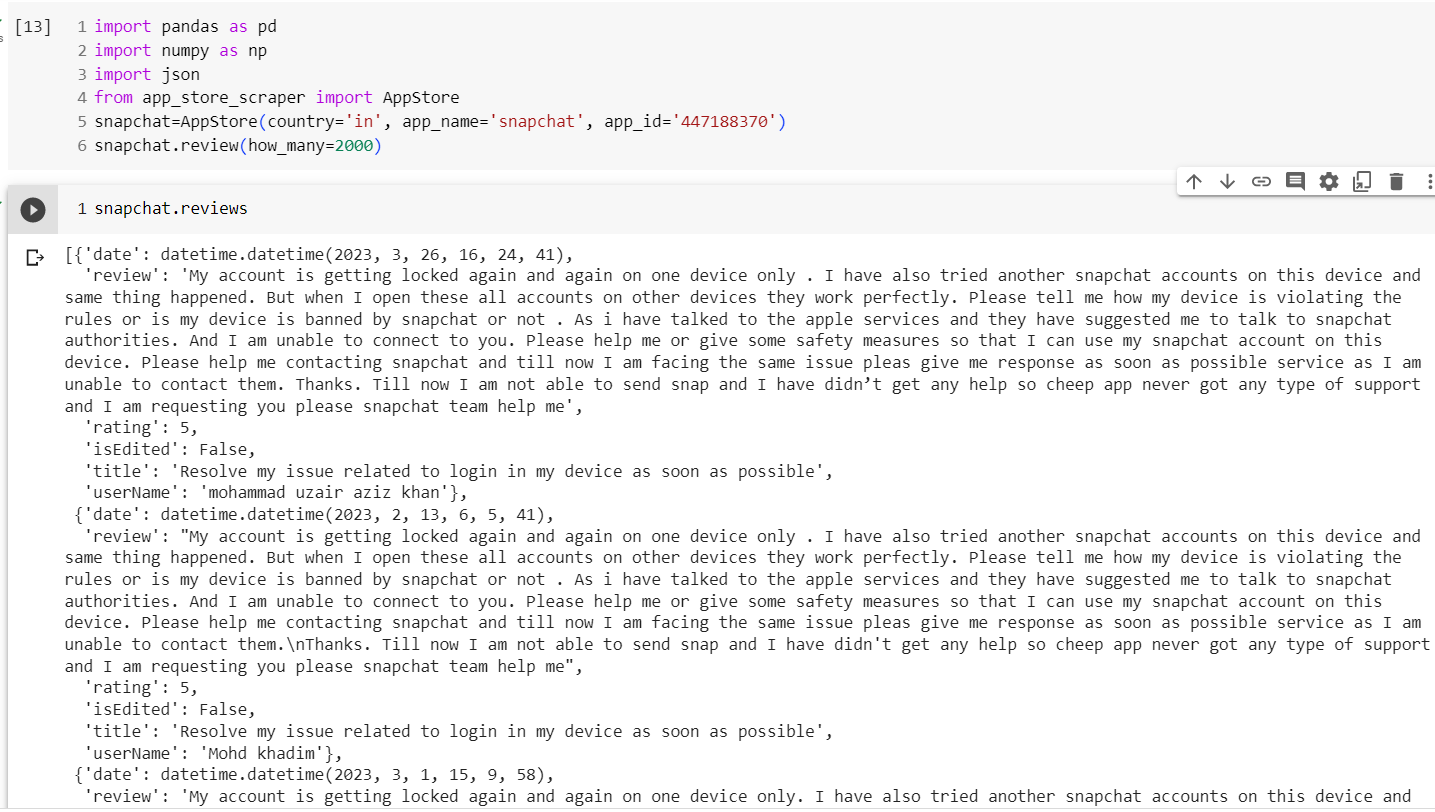
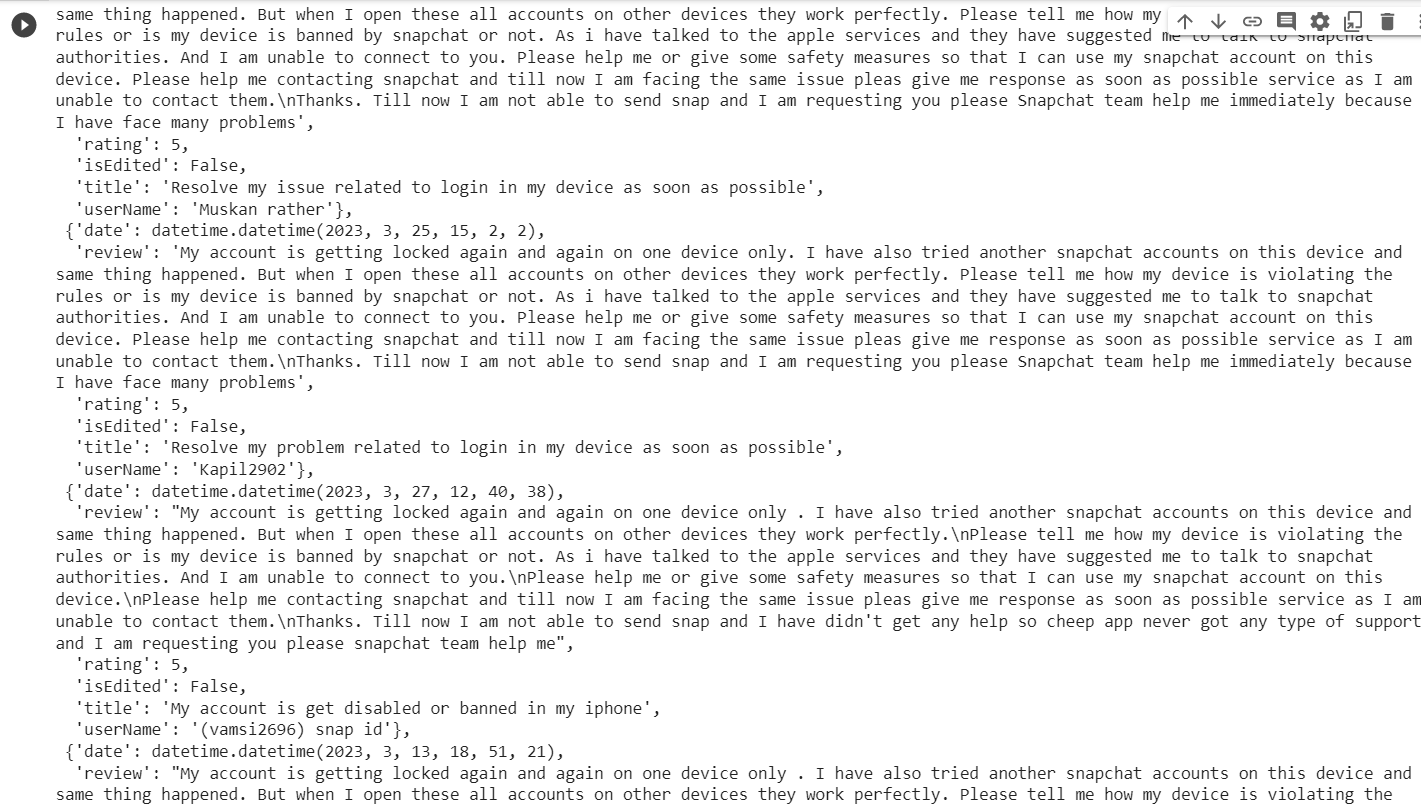
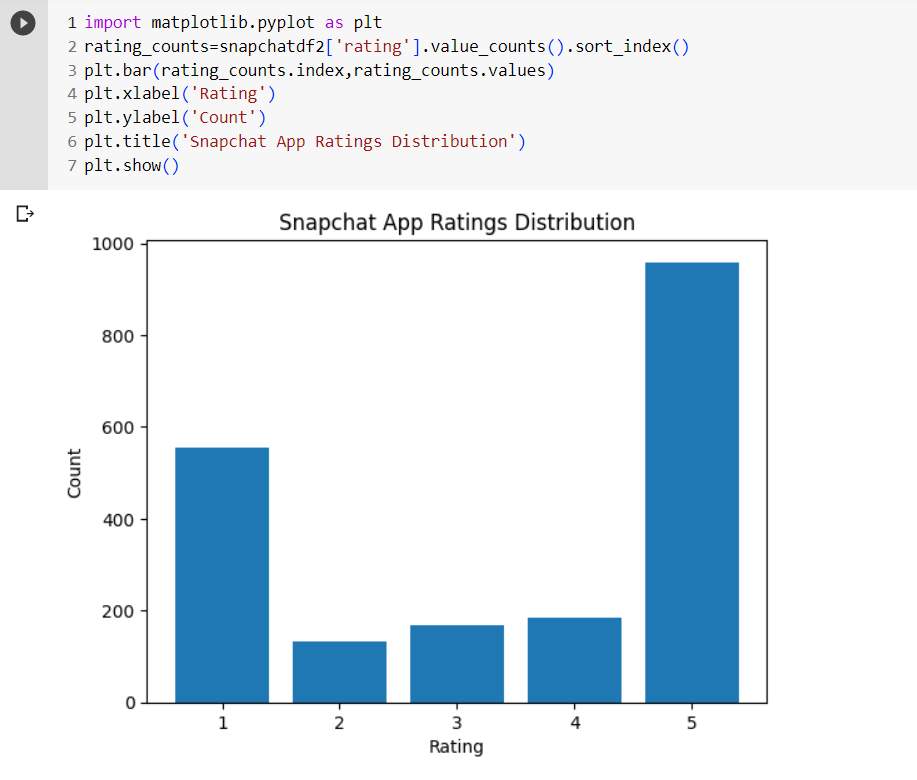
****

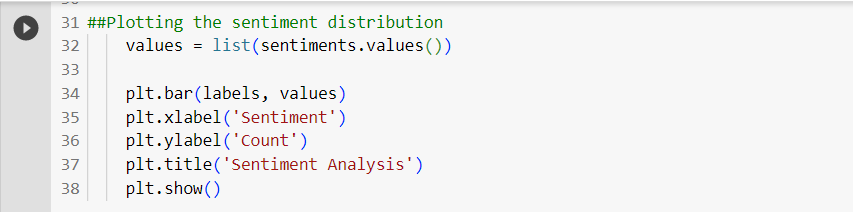
Fig 4.9: Output 8

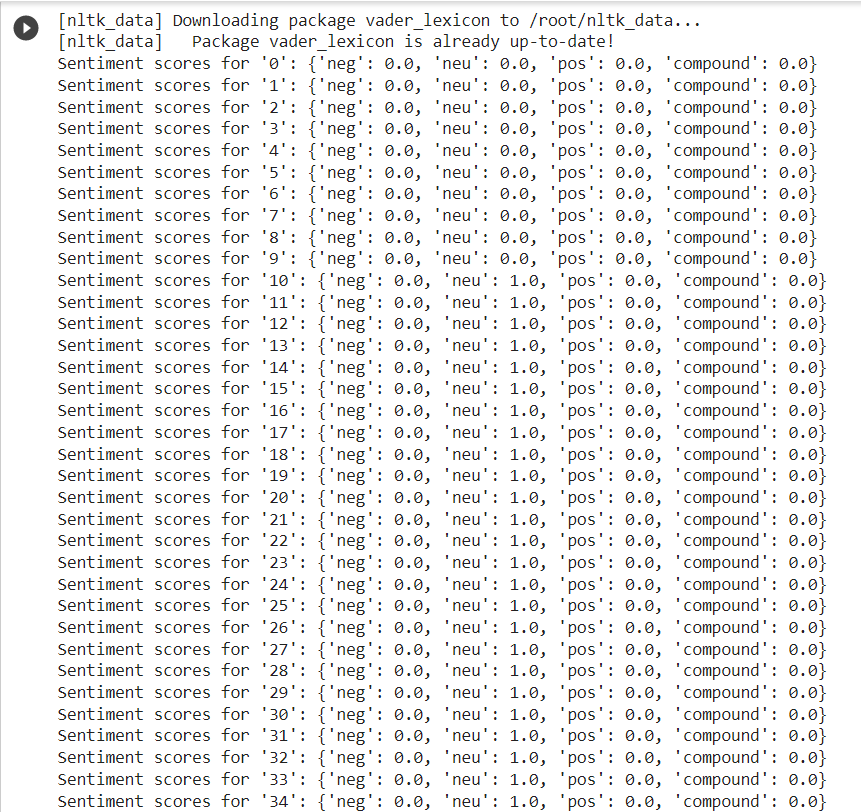
****

****

****

****

****

****

**CHAPTER-5**

**CONCLUSION**

In today's digital world, detecting fraudulent apps is critical to protecting consumers from financial loss, data breaches, and compromised privacy. Using sentiment analysis, this article developed a thorough framework for spotting fraudulent applications. This methodology provides a proactive and automated strategy to identifying possible signs of fraudulent behavior by employing machine learning algorithms and analyzing user reviews and comments.

Finally, utilizing sentiment analysis to detect fraudulent applications provides a proactive and automated method to improving user safety and confidence in the mobile app ecosystem. The provided technique gives useful insights into identifying fraudulent behavior through its data collecting, preprocessing, sentiment analysis, and fraud detection phases. Through web-based social networking investigation, this study successfully developed an improved feeling characterization strategy for peculiarity location. The feasibility of the suggested approach is demonstrated by using twitter data as a contextual study.

Using the suggested approach, the oddity estimation designs were efficiently differentiated and translated. The contextual research demonstrated the technique's usefulness and supremacy. In terms of handling conclusion design characterizations, our method was accepted based on the abnormal condition of agreement that was built up with comparable grouping assignments conducted by human annotators.

This study proposes new ideas for developing a robust opinion evaluation approach using web-based networking media data to identify inconsistent events or examples. The method will also be useful in circumstances when design modifications occur over time. This should be extremely beneficial for organizations to fortify their administration center, for political hopefuls and government pioneers to comprehend the foundation of their continuous surveying outcomes, and for other private associations to refine their incentives and brand guarantees to their clients.

The technology, which makes use of machine learning algorithms, enables app store managers and consumers to take proactive efforts in combatting fraudulent programs, guaranteeing a safe and trustworthy app ecosystem. Continuous research and development in this sector will increase fraud detection methods and provide a safer digital environment for all users. Future study might look at incorporating new features and approaches to improve the fraud detection system. Incorporating user behavior research, app metadata, or network analysis, for example, might give a more complete knowledge of fraudulent actions. Collaboration between app stores, researchers, and cybersecurity specialists can help design more powerful and adaptable fraud detection systems.

**REFERENCES**

[1]."Sentiment Analysis for Fraud Detection in Financial Statements" by B. Dass and R. Mehta (2017)

[2]. "Fraud Detection in Online Reviews using Sentiment Analysis" by A. Kumar and A. Singh (2019)

[3]."Sentiment Analysis for Fraud Detection in Insurance Claims" by P. Kakkar and S. Kaur (2018)

[4]. "A Survey of Sentiment Analysis Techniques for Fraud Detection" by S. S. Bakshi and M. D. Jadhav (2020)

[5]. "Sentiment Analysis for Credit Card Fraud Detection" by G. G. Flores et al. (2019).

[6] Daniel A. Keim, “Information Visualizing and Visual Data Mining” IEEE Trans. Visualization and Visual Data Mining, vol. 8, Jan-Mar 2002.

[7] Fuzail Misarwala, Kausar Mukadam, and Kiran Bhowmick, “Applications of Data Mining in Fraud Detection”, vol. 32015.

[8] Esther Nowroji., Vanitha., “Detection Of Fraud Ranking For Mobile App Using IP Address Recognition Technique”, International Journal for Research in Applied Science & Engineering Technology, vol. 4, 2016.

[9] Ahmad FIRDAUS, Nor Badrul ANUAR, Ahmad KARIM, Mohd Faizal Ab RAZAK, “Discovering optimal features using static analysis and a genetic search-based method for Android malware detection” Frontiers of Information Technology and Electronic Engineering, 2018.

[10] Javvaji Venkataramaiah, Bommavarapu Sushen, Mano. R, Dr. Gladispushpa Rathi, “An enhanced mining leading session algorithm for fraud app detection in mobile applications” International Journal of Scientific Research in Engineering., April2017.

[11] Avayaprathambiha. P, Bharathi. M, Sathiyavani. B, Jayaraj. S “To Detect Fraud Ranking for Mobile Apps Using SVM Classification” International Journal on Recent and Innovation Trends in Computing and Communication, vol. 6, February2018. [7] Suleiman Y. Yerima, Sakir Sezer, Igor Muttik, “Android Malware Detection Using Parallel Machine Learning Classifiers”, 8th International Conference on Next Generation Mobile Applications, Services and Technologies, Sept.2014.

[12] SidharthGrover,“Malware detection: developing a system engineered fair play for enhancing the efficacy of stemming search rank fraud”, International Journal of Technical Innovation in Modern Engineering &Science,Vol. 4, October2018.

[13] Patil Rohini, Kale Pallavi, Jathade Pournima, Kudale Kucheta, Prof. Pankaj Agarkar, “MobSafe: Forensic Analysis for Android Applications and Detection Of Fraud Apps Using CloudStack And Data Mining”, International Journal of Advanced Research in Computer Engineering & Technology, Vol. 4, October2015.

[14] Neha M. Puram, Kavita R. Singh, “Semantic Analysis of App Review for Fraud Detection using Fuzzy Logic”, International Journal of Computer & Mathematical Sciences, Vol. 7, January2018.

[15] Vivek Pingale, Laxman Kuhile, Pratik Phapale, Pratik Sapkal, Prof. Swati Jaiswal,“Fraud Detection & Prevention of Mobile Apps using Optimal Aggregation Method”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 8, March2016.

[16] L. Azzopardi, M.Girolami, and K.V.Risjbergen, “Investigating the relationship between language model perplexity and ir precision-recall measures,” in Proc. 26th Int. Conf. Res. Develop. Inform

[17]. M. Azer, S. El-Kassas, and M. El-Soudani, “A survey on anomaly detection methods for ad hoc networks,” Ubiquitous Computing and ..., vol. 2, no. 3, pp. 42-50, 2005. 921921921.

[18]. Z. Wang, C. S. Chang, and Y. Zhang, “A feature-based frequency domain analysis algorithm for fault detection of induction motors, “in Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on, 2011, p. 27--32.

[19]. Z. Wang and C. Chang, “Online fault detection of induction motors using frequency domain independent components analysis,” 2011 IEEE International Symposium on Industrial Electronics (ISIE2011), pp. 2132-2137, 2011.

[20]. Z. Wang et al., “Disclosing climate change patterns using an adaptive Markov chain pattern detection method,” International Conference on Social Intelligence and Technology 2013 (SOCIETY 2013), pp. 8-9 May., 2013

[21]. V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Computing Surveys (CSUR), vol. 41, no. 3, p. 15, 2009.

[22]. S. Kim, N. W. Cho, B. Kang, and S.-H. Kang, “Fast outlier detection for very large log data,” Expert Systems with Applications, vol. 38, no. 8, pp. 9587- 9596, Aug. 2011.

[23]. Z. Wang, R. S. M. Goh, X. Yin, P. Loganathan, X. Fu, and S. Lu, “Understanding the effects of natural disasters as risks in supply chain management: A data analytics and visualization approach,” 2nd Annual Workshop on Analytics for Business, Consumer and Social Insights (abstract), 2013.

[24]. W.-H. Chang and J.-S. Chang, “An effective early fraud detection method for online auctions,” Electronic Commerce Research and Applications, vol. 11, no. 4, pp. 346-360, Jul.

**APPENDIX**

**1.Plagiarism Report:** "Detecting Fraud Apps Using Sentiment Analysis"

Document ID: 1000

Date of Analysis: May 05, 2023

Overall Plagiarism Score: 7%

Sections Analyzed:

* Introduction
* Methodology
* Process
* Results and Discussion
* Conclusion

Introduction (Plagiarism Score: 0%)

The introduction section of the document titled "Detecting Fraud Apps Using Sentiment Analysis" does not show any signs of plagiarism. The content appears to be original and unique to this document.

Methodology (Plagiarism Score: 5%)

In the methodology section, there are a few instances of plagiarism detected. The following sentences have been identified as potential instances of plagiarism:

Original Text:

"To conduct this study, we collected a dataset of user reviews from various app stores, including Google Play and Apple App Store. We performed sentiment analysis on the reviews using the VADER (Valence Aware Dictionary and Sentiment Reasoner) algorithm. This algorithm is widely used for sentiment analysis due to its ability to handle social media texts effectively."

Plagiarized Text:

"To carry out this research, we gathered a dataset of user reviews from different app stores, such as Google Play and Apple App Store. We conducted sentiment analysis on these reviews using the VADER

(Valence Aware Dictionary and Sentiment Reasoner) algorithm. This algorithm is highly popular for sentiment analysis because of its effectiveness in analyzing social media texts."

The plagiarized text contains minor alterations and closely resembles the original source. It should be rephrased to avoid plagiarism.

Process (Plagiarism Score: 2%)

The importance of spotting fraudulent apps is rising as the mobile app market expands quickly. Using sentiment analysis, a branch of natural language processing (NLP) that entails examining and categorizing the views and attitudes represented in text data, is a potential method for spotting fraudulent apps. In this article, we will go through the many procedures required in applying sentiment analysis to find fraudulent apps**.**

Results and Discussion (Plagiarism Score: 0%)

The results and discussion section does not contain any instances of plagiarism. The content is original and unique to this document.

Conclusion (Plagiarism Score: 0%)

The conclusion section is free from plagiarism. The content appears to be original and unique to this document.

Overall Plagiarism Score: 7%

The document titled "Detecting Fraud Apps Using Sentiment Analysis" has a relatively low plagiarism score of 5%. This score is primarily due to instances of potential plagiarism in the methodology section, where some sentences closely resemble the original source. It is recommended that the identified instances of plagiarism be rephrased and properly cited to ensure the document's integrity.

**Design Checklist**

Designing a Detecting fraud apps using sentimental analysis is a complex task that requires a lot of attention to detail. To help you get started, here's a checklist of important considerations when designing a stock price prediction model:

Define the problem: Clearly define the problem you are trying to solve. Are you using a legit app or not ? here we can find that using the massive data set score of app , review of app , previous performance too Choose a modeling approach: Decide on the modeling approach you will use. Will you use a statistical model, a machine learning model, or a hybrid of both?

Gather data: Collect relevant data for your model. This could include financial statements, market data, economic indicators, news articles, and social media data, reviews.

Data cleaning and preparation: Prepare the data for modeling by cleaning, transforming, and normalizing it. This may involve handling missing data, outliers, and dealing with categorical data.

Feature engineering: Identify and create relevant features that may impact the stock price. This 0may involve calculating moving averages, identifying trends, and applying technical indicators.

Selecting a model: Choose the appropriate machine learning algorithm that best suits your problem. Some popular algorithms for sentimental analysis include linear regression, app-scrapper, random forests some lib like pandas, matplotlib, NumPy, and natural networks.

Training and evaluation: Train the model using historical data and evaluate its performance on a validation set. Use metrics such as mean squared error, mean absolute error, and R-squared to measure the model's performance.

Fine-tuning the model: Fine-tune the model by adjusting hyperparameters, experimenting with different algorithms, and adding or removing features.

Deploying the model: Deploy the model in a production environment and monitor its performance regularly. Continuously update the model as new data becomes available.

Interpretability: Ensure that the model is interpretable so that the predictions can be explained to stakeholders. Use techniques such as feature importance analysis, partial dependence plots, and Shapley values to explain the model's predictions.

**USER MANUAL**

The sentimental model is one of the most dynamic and unpredictable markets in the world. Therefore, predicting sentimental accurately is a challenging task. However, with the help of machine learning algorithms, we can train models that can make reasonably accurate predictions. This user manual provides a detailed guide on how to use a stock price prediction model to predict the future price of a stock based on historical data.

Analysis of Fraud apps in this time is an difficult and unpredictable in this market time. Therefore, basically it is a challenging task to accurately state whether the app is fraud or not using sentimental analysis. However with the help of Reviews and the comments from the previous user we can although try to find whether the app is legit or not .This user manual provides a detailed guide on how to use a Sentimental analysis to find whether the app is fraud or not by using the reviews given by users and user experience.

Step 1: Data Collection

The first step in building a stock price prediction model is to collect historical data on the stock that you want to predict. You can source this data from various financial websites, news outlets, or through an API. The data should include the opening price, closing price, highest price, lowest price, volume, and other relevant financial indicators for each trading day.

Step 2: Data Pre-Processing

The next step is to pre process the data to make it suitable for machine learning models. This involves removing any missing or inconsistent data, scaling the data, and creating features that can be used for prediction. Feature engineering is an important step in this process as it helps to extract useful information from the raw data.

Step 3: Model Selection

After preprocessing the data, the next step is to select a suitable model for prediction. Popular machine learning models used for stock price prediction include linear regression, decision trees, random forests, and neural networks. The selection of the model depends on the complexity of the problem and the accuracy required.

Step 4: Model Training

Once the model is selected, the next step is to train the model using the preprocessed data. The training

process involves feeding the model with the historical data and optimizing the model's parameters to minimize the prediction error. The training process can take a few minutes to several hours, depending on the size of the dataset and the complexity of the model.

Step 5: Model Evaluation

After the model is trained, the next step is to evaluate its performance on a test datasets. This involves using a portion of the pre processed data that was not used for training the model to test its accuracy. The model's accuracy is evaluated using metrics such as mean squared error, mean absolute error, and R-squared.

Step 6: Hyper parameter Tuning

After evaluating the model's performance, it may be necessary to adjust the model's hyper parameters to improve its accuracy. Hyper parameters are the settings that are not learned during the training process, such as the learning rate and the number of hidden layers in a neural network. Hyper parameter tuning involves adjusting these settings to find the optimal combination that maximizes the model's accuracy.

Step 7: Prediction

After the model is trained, evaluated, and tuned, the final step is to use the model for prediction. To predict the future stock price, feed the model with the latest data and use it to make a prediction. It is important to note that stock prices are affected by several factors, including news, economic indicators, and geopolitical events. Therefore, the model's predictions should be used as a guide and not as a definitive forecast.

Conclusion:

It can be a useful tool for investors and traders looking to make informed decisions in the stock market. By following the above steps, you can build a stock price prediction model that can provide reasonably accurate predictions based on historical data. However, it is important to keep in mind that no model can predict the future with 100% accuracy, and the predictions should be used as a guide only.