ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

Fraud Apps Detection Using Sentiment Analysis

G. Santhoshi Kumari [1], B Meghana Veronica [2], G Sai Manogya [3], M Ashok Sagar [4], S Marvillous Kiran [5].

1 Assistant Professor, Dept. of CSE, NS Raju Institute of Technology Visakhapatnam, Andhra Pradesh, India
2,3,4,5 Student, Dept of CSE, NS Raju Institute of Technology Visakhapatnam, Andhra Pradesh, India
santoshikumari.cse@nsrit.edu.in, meghanaveronica225@gmail.com, manogyavarma999@gmail.com, ashoksagarvincy@gmail.com, marvillouskiran@gmail.com.

Abstract

Fake mobile application is software that mimic functionality of valid, reliable and genuine applications. Once this application is installed, they perform malicious actions like aggressively display advertisements to get revenue from that, intercept sensitive data from your system, infect devices and so on. Most of the times the user cannot differentiate between the fake and legitimate applications hence before downloading any app people always enquire about the opinion of the app by the users. In this project we are introducing a platform where people can enquire about the application before downloading it.

Basically, we will analyze the reviews using sentiment analyses which is a text classification technique which analysis text and tells whether the sentimentis positive, neutral or negative. And we implement machine learning algorithms such as Navi Bayes, random forest. Analyzing the rating and reviews together involving both user and admins comments, we can determine whether the app is genuine or not.

For every user's reviews and comments will be fetched separately and analyzed for positive and negative rating. According to the evaluate accuracy of the given reviews we can predict the applications as Genuine or Fraud. In review Based Evidences, besides rating, most of the App stores also allow users to write some textual comments as App reviews. Such reviews can reflect the personal perceptions and usage experiences of existing users for particular mobile Apps. Indeed, review manipulation is one of the most important perspective of App ranking fraud. So, we are proposing a system that can be carried out by using corpus based and Naive Bayes based approach to detect fraud applications.

Keywords: Fraud Apps Detection, Sentiment Analysis, Technological development.

1. Introduction

Sentiment is an emotion or attitude prompted by the feelings of the customer. Sentiment analysis is also referred to as opinion mining, as opinions are collected from customer is mined to reveal the rating of the app. The process of Sentiment analysis comes under machine learning.

Cite this article as: G.Santhoshi Kumari, B Meghana Veronica, G Sai Manogya, M Ashok Sagar & S.Marvillous Kiran, "Fraud Apps Detection Using Sentiment Analysis", International Journal of Research in Advanced Computer Science Engineering, (IJRACSE), Volume 6 Issue 12, May 2021, Page 9-21.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

Information is gathered and is analyzed to determine the sentiment about the information such as negative or positive sentiment. Before purchasing the app people always enquire about the opinion of the app by the other users.

The process of Sentiment analysis uses natural language processing (NLP) to collect and examine the opinion or sentiment of the sentence. It is popular as many people prefer to take some advice from the users. As the amount of opinions in the form of reviews, blogs, etc. are increasing continuously, it is beyond the control of manual techniques to analyze huge amount of reviews & to aggregate them to an efficient decision. Sentiment analysis performs these tasks into automated processes with less user support.

It is not always possible to have a one technique to fit in all solution because different types of sentences express sentiments /opinions in different ways. Sentiment words (also called as opinion words) (e.g., great, beautiful, bad, etc.) cannot distinguish an opinion sentence from anon-opinion one. A conditional sentence may contain many sentiment words or sentences, but express no opinion. The type of sentences, i.e., conditional sentences, it has some unique characteristics which make it hard to determine the orientation of sentiments on topics/features in such of the sentences. By sentiment orientation, we mean positive, negative or neutral opinions. Conditional sentences are sentences which describe implications or hypothetical situations & their consequences. In English language, a variety of conditional connectives can be used to form these sentences. A conditional sentence contains two clauses: the condition clause and the consequent clause, that are dependent on each other. Their relationship has significant implications on whether the sentence describes an opinion.

As there are more than millions of apps on the App store, there is many competitions between apps to be on top of the leader board on the basis of popularity. As leader board is the most important way for promoting apps. The higher rank on the leader board leads to huge number of downloads & million doll or of profit. Apps give advertisement to promote their apps on the leader board. Many apps use fraudulent means to boost their ranking on the leader board of the App store. There are various means to increase downloads & ranking of the app which is done by "bot farms" or "human water armies", human water armies are a group of internet ghostwriters who are paid to post fake reviews. The app is said to fraud on the basis of 3 parameters: Ranking, Rating & Review of the app. In ranking based we check the historical ranking of the app, there are 3 different ranking phases, rising phase, maintaining phases & recession phase. The apps ranking rising to peak position on leader board (i.e. rising phase), to keep at the peak position on the leader board (i.e. maintaining phase), & finally decreasing till the end of event (i.e. recession phase). There views are taken from the dataset and are converted into tokens on which sentiment analysis is performed.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

1.2 Sentiment Analysis Techniques

1.2.1 Machine Learning

Machine learning based Sentiment Analysis or arrangement should be possible in two different ways:

Sentiment Analysis by utilizing directed machine learning strategies and

Sentiment Analysis by utilizing unsupervised machine learning procedures.

(a) Supervised Machine Learning

In Supervised Machine learning procedures, two sorts of informational collections are required: preparing informational index and test informational collection. A programmed classifier takes in the grouping variables of the report from the preparation set and the exactness in order can be assessed utilizing the test set. Various machine learning calculations are accessible that can be utilized extremely well to characterize the records. The machine learning calculations like Support Vector Machine (SVM), Naive Bayes(NB) and greatest entropy(ME) are utilized effectively in numerous examinations and they performed well in the feeling characterization.

(b) Unsupervised Machine Learning

Lexicon Based Method is an Unsupervised Learning approach since it does not require prior training data sets. It is a semantic orientation way to deal with opinion mining in which sentiment polarity of highlights show in the given record are controlled by contrasting these highlights and semantic lexicons. Semantic dictionary contains arrangements of words whose sentiment orientation is resolved as of now. It arranges the archive by conglome rating the sentiment orientation of all assessment words display in the record, reports with more positive word lexiconsis characterized as positive document and the documents with more negative word lexicons is classified as negative document.

Hybrid Technique: A few researchers combined the supervised machine learning and lexicon-based techniques jointly to enhance sentiment classification performance. They considered both general reason lexicon and domain specific lexicon for identifying polarity orientation of sentiment words and feed these lexicons into supervised learning algorithm, SVM. They found that general purpose lexicon performed very poor while domain specific lexicon performed very well.

The system classified the sentiment in two steps: First the classifier is trained to predict the aspects and In Next the classifier is trained to predict the sentiments related to the aspects collected in step. Their system yielded around 66.8%accuracy.

1.3 Lexical Information

Understanding human emotions (also called sentiment analysis) is a hard job for a machine, for which the computational intelligence technique may offer enhanced results. Regularly, semantic articulations and additionally paralinguistic includes in talked dialects (e.g., pitch, loudness, tempo, etc.) reveal the sentiments or emotional states of individuals. Prior research studies have developed sentiment lexicons using a

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

dictionary technique and a corpus technique. Social media has changed the world. It has become an everyday part of our lives. Many people are nowadays active on several popular social networks such as Face book, twitter, Instagram, etc. They share photos and posts on their daily life and experiences such as their food, their clothes, and their trips. Some people are more active on social networks, while others are less so.

2. System Analysis

2.1 Existing System

A general approach to protect the data confidentiality is to encrypt the data before outsourcing. Searchable encryption schemes enable the client to store the encrypted data to the cloud and execute keyword search over cipher text domain. So far, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search, multi-keyword Boolean search, ranked search, multi-keyword ranked search, etc. Among them, multi-keyword ranked search achieves more and more attention for its practical applicability. Recently, some dynamic schemes have been proposed to support inserting and deleting operations on document collection. These are significant works as it is highly possible that the data owners need to update their data on the cloud server.

2.1.1 Disadvantages Of Existing System

Huge cost in terms of data usability. For example, the existing techniques on keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical.

Existing System methods not practical due to their high computational overhead for both the cloud sever and user.

2.2 Proposed System

This paper proposes a secure tree-based search scheme over the encrypted cloud data, which supports multi-keyword ranked search and dynamic operation on the document collection. Specifically, the vector space model and the widely-used "term frequency (TF) × inverse document frequency (IDF)" model are combined in the index construction and query generation to provide multi-keyword ranked search. In order to obtain high search efficiency, we construct a tree-based index structure and propose a "Greedy Depth-first Search" algorithm based on this index tree. The secure KNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. To resist different attacks in different threat models, we construct two secure search schemes: the basic dynamic multi-keyword ranked search (BDMRS) scheme in the known ciphertext model, and the enhanced dynamic multi-keyword ranked search (EDMRS) scheme in the known background model.

2.2.1 Advantages of Proposed System

Due to the special structure of our tree-based index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the deletion and insertion of

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

documents. We design a searchable encryption scheme that supports both the accurate multi-keyword ranked search and flexible dynamic operation on document collection. Due to the special structure of our tree-based index, the search complexity of the proposed scheme is fundamentally kept to logarithmic. And in practice, the proposed scheme can achieve higher search efficiency by executing our "Greedy Depth-first Search" algorithm. Moreover, parallel search can be flexibly performed to further reduce the time cost of search process.

3. Requirements and Specifications

3.1 Scope of The System

This project is intended to implement a system that find ranking, rating and review behaviors for investigating review-based evidences, rating based evidences and ranking based evidences and then aggregation based on optimization to combine all the evidences for detection of fraud.

3.2 Objective of The System

To design a system which may detect fake apps by considering different evidences indicating their true behavior. To find apps are real or not. To increase the classification accuracy of a system.

3.3 Functional Requirements

In Software engineering, a functional requirement defines a function of a software system or its components. A function is described as set of inputs, the behavior and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements which impose constants on the design or implementation.

3.3.1 Functional Requirements For This Project

The functional requirements describe the interactions between the system and its environment independent of its implementations.

MODULE 1: Data Collection Details

Data collection is an important part of Machine Learning. Data collection is the process of gathering and measuring information for different available sources. Machine Learning requires a huge set of data having multiple attributes, to be able to classify some input parameters more accurately. Data collection is the important aspect that makes the algorithm training possible. It has been observed that greater number of attributes yields a better result. Training data for fraud app detection is obtained from Training data Training dataset was used for training the algorithm so that algorithm learns and produce results. Training dataset consist of13000 entries (reviews, sentiment value). Training data set consists 50% of positive and 50% of negative reviews. Testing dataset: Testing dataset was used for evaluating the model/algorithm with trained dataset. Testing dataset is real time data set which is extracted from google play store.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

MODULE 2: Data Preprocessing

The process of converting data to something a computer can understand is called Preprocessing. The dataset which is obtained in data collection is not in the form which can be used by the classifier. Various Data preprocessing and feature extraction techniques must be performed on the dataset to make it suitable for generation of classification model. The python library Pandas is used toper form the preprocessing techniques on the dataset. Preprocessing steps are: Tokenization

Tokenization basically refers to splitting up a large body of text in to smaller lines, words or even creating words for a non-English language. Various to kenization functions are in built into module itself.

MODULE 3: Stop words Removal

Stop Word Removal is a process of filtering out useless data. In NLP, useless words are referred to ass top words. Lower case conversion In this all the upper-case letters are converted to lower case. Tfidf Vectorizer The Tfidf Vectorizer will to kenize documents, learn the vocabulary and inverse document frequency weightings, and allow you to encode new documents.

3.4 Non-Functonal Requirements

In System engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system rather than specific behavior. In general, functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be.

3.5 System Requirements

3.5.1 Software Requirements

Operating system: - Windows XP.
Coding Language: J2EE
Data Base: MYSQL

3.5.2 Hardware Requirements

System: Pentium IV 2.4 GHz.
Hard Disk: 40 GB.
Floppy Drive: 1.44 Mb.
Monitor: 15 VGA Colour.
Mouse: Logitech.
Ram: 512 Mb.

3.6 Software Development Environment

3.6.1 An Introduction to Python

Python is a popular object-oriented programming language having the capabilities of high-level programming language. Its easy to learn syntax and portability capability makes it popular these days.

The chapter describes about the software tool that is used in our project. Python was developed by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands. It was written as the successor of programming language named ‘ABC’. It’s first version was released in 1991. The name Python was picked by Guido van Rossum from a TV show named Monty Python’s Flying Circus. It is an open source programming language which means that we can freely download it and use it to develop programs.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

4. System Design

4.1 Introduction

The design of a system is essentially a blueprint or a plan for a solution for the system. Here we consider a system to be a set of components with clearly defined behavior that interacts with each other in a fixed defined manner to produce some behavior or services for its environment.

4.1.1 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

4.1.2 Objectives

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

4.1.3 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2.Select methods for presenting information.
3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

4.2 Database Design

A general theme begins a database is to handle information as an integrated whole. A database is a collection of inter-related data stored with minimum redundancy to server many users quickly and efficiently. The general objective is to make information access easy, quick, expensive and flexible for the user. In database design several specific objectives are consider:

4.2.1 Control Redundancy

Redundant data occupies space and therefore, is wasteful. If versions of the same data are in different phase of updating, a system often gives conflicting information. A unique aspect of database design is storing data only once, which controls redundancy and improves system performance.

4.2.2 Data Independence

An important database objective is changing hardware and storage procedures for adding raw new data without having to rewrite application programs.

4.2.3 Accuracy and Integrity

The accuracy and database ensure the data quality content remain constant. Integrity controls detects data inaccuracy where occur.

4.2.4 Privacy and Security

For the data to remain private, security measures must be taken to an unauthorized access. Database security means that data are protected from various forms of destructions. Uses must be positively identifies and actions monitored. Managing the database require a Database Administrator (DBA) whose key functions are to be managing data activities, The database structure and the DBMS. In addition, a managerial background the DBA needs a technical knowledge to deal with database designer.

5. Implementation Code

```
import pandas as pd
from sklearn.model_selection import
    train_test_split
import joblib
from sklearn.feature_extraction.text import
    CountVectorizer

# In[2]:
df = pd.read_csv('frauddataset.csv')
# In[3]:
```

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

```
print(df.head())
# In[4]:

def preprocess_data(df):
    # Remove package name as it's not relevant
    df = df.drop('package_name', axis=1)

    # Convert text to lowercase
    df['review'] =
      df['review'].str.strip().str.lower()
    return df

# In[5]:
df = preprocess_data(df)
# Split into training and testing data
x = df['review']
y = df['polarity']
x, x_test, y, y_test = train_test_split(x,y,
    stratify=y, test_size=0.25,
    random_state=42)
# Vectorize text reviews to numbers
vec = CountVectorizer(stop_words='english')
x = vec.fit_transform(x).toarray()
x_test = vec.transform(x_test).toarray()
# In[8]:

from sklearn.tree import
    DecisionTreeClassifier
from xgboost import XGBClassifier
from sklearn.naive_bayes import
    MultinomialNB
from sklearn.metrics import confusion_matrix,
    accuracy_score
from sklearn.metrics import
    classification_report
model = MultinomialNB()
print(model.fit(x, y))

# In[9]:
print(model.score(x_test, y_test))
p=model.score(x_test, y_test)
# In[10]:

print(model.predict(vec.transform(['I love this
    app !!!'])))
pred = model.predict(x_test)

# In[11]:
print(accuracy_score(y_test,pred))
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))

# Save model
joblib.dump(model, 'model.pkl')
xgb = XGBClassifier(max_depth=10,
n_estimators=400, learning_rate=0.3,
objective='binary:logistic')
xgb.fit(x, y)
q=xgb.score(x_test, y_test)
pred = xgb.predict(x_test)
print(accuracy_score(y_test,pred))
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
dt = DecisionTreeClassifier()
dt.fit(x, y)
r=dt.score(x_test, y_test)
pred = dt.predict(x_test)

print(accuracy_score(y_test,pred))
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))

import matplotlib.pyplot as plt
# creating the dataset
data = {'MultinomialNB':p, 'XGBClassifier':q,
    'DecisionTreeClassifier':r}
courses = list(data.keys())
values = list(data.values())
```

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

```
fig = plt.figure(figsize = (10,5))

# creating the bar plot
```

6. Testing

6.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1.1 Testing Objectives

All field entries must work properly. Pages must be activated from the identified link. The entry screen, messages and responses must not be delayed. Testing cannot show the absence of defects, it can only show that software errors are present. Once code has been generated the testing begins.

The purpose of testing is more than just debugging and detecting of bugs. Testing is usually performing for the following:

- For improving and assuring software quality.
- For estimation reliability.
- For verification and validation.

The objection that should kept in mind while testing is being executed as follows:

- It should be easily predictable.
- It should be fixed.
- It should follow certain constraints a rule.

6.1.2 Testing Approaches

Field testing will be performed manually and functional tests will be written in detail.

The different types of Tests are as follows:

1) Unit Testing
2) Integration Testing
3) Functional Test
4) System Test
5) White Box Testing
6) Black Box Testing
7) Acceptance Testing

6.2 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results. Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

6.3 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

6.4 Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.5 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

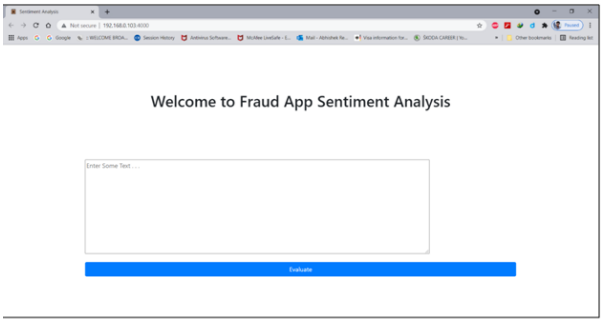
6.6 White Box Testing

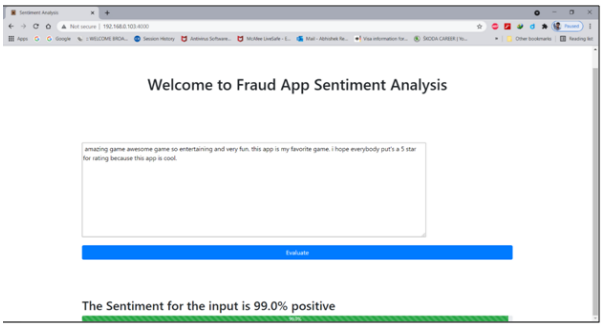
White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

7. Screenshots

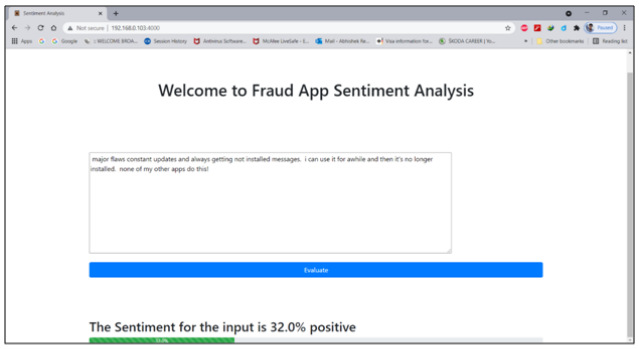
Main Page:

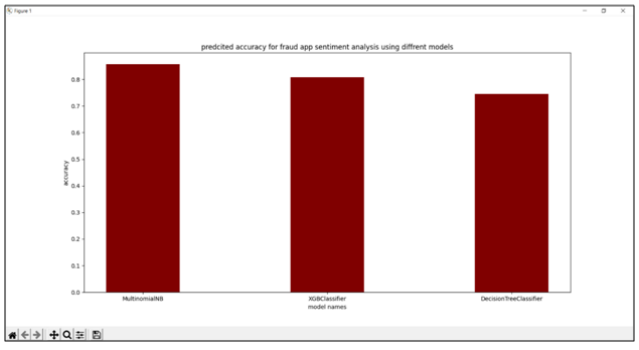




Description

From the above figure, the window clearly tells that the evaluate of given review produces 99.0% positive and High accuracy rate.





Description

From the above figure, we can see the predicted accuracy for fraud app sentiment analysis using different models i.e. Multinomial NB, XGB Classifier, and Decision Tree Classifier. Among those three models Multinomial NB produced high accuracy.

8. Conclusion

In this project, we have conducted a survey regarding different methodologies used in reviewing the status of application and predicting whether it is fraud or not. Our proposed methodology deals with sentiment analysis which has an advantage over the other methods due to fact that lexicon-based analysis is more accurate and faster than other approaches.

When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you would like less training data. It is a fast algorithm for classification problems. It is an honest fit real time prediction, multiclass prediction, recommendation system, text classification and sentiment analysis use cases. Naive Bayes Algorithm are often built using Gaussian, Multi nomial and binomiall distribution. it's very low computational cost.

Reference

1. Safrin,Raheesa, K.R.Sharmila, TSShriSubangi, and E.A.Vimal."Sentiment analysis on online product review. "Int.Res.J.Eng.Technol4,no.04(2017).
2. Shahana, P.H., and Bini Omman. "Evaluation

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

of features on sentimental analysis."Procedia Computer Science 46(2015):15851592.

3. "Muhammad Taimoor Khan, Mehr Durrani, Armughan Ali, Irum Inayat, Shehzad Khalid & Kamran Habib Khan"Sentiment analysis and complex natural language" (2016)a springer open journal.
4. Narayanan, Ramanathan, BingLiu, and Alok Choudhary. "Sentiment analysis of conditional sentences. "In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume1-Volume1, pp.180-189. Association for Computational Linguistics, 2009.
5. Li, Gang, and Fei Liu. "A clustering-based approach on sentiment analysis." In 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering, pp.331-337.IEEE,2010.
6. Xu, Jianlin, YifanYu, ZhenChen, BinCao, WenyuDong, YuGuo, and JunweiCao. "MobSafe: cloud computing based forensic analysis for massive mobile applications using data mining. "Tsinghua science and technology18, no.4(2013): 418-427.
7. Tichkule, Ashwini, Nidhi Nikhar, Dewanand Kapgate, and Omkar Dudhbure. "Revelation of Fraud Apps using Sentiment Analysis App Reviews.
8. Rohini,V .,and M.Thomas. "Comparison of Lexicon based and Naïve Bayes Classifierin Sentiment Analysis."International Journal for Scientific Research & Development3,no.4(2015).
9. Guzman, Emitza, and Walid Maalej."How do users like this feature? A fine grained sentiment analysis of app reviews. "In 2014 IEEE 22nd international requirements engineering conference (RE), pp.153-162.IEEE, 2014.
10. Zhou, Xujuan, Xiaohui Tao, Jianming Yong, and Zhenyu Yang. "Sentiment analysis on tweets for social events. "In Proceedings of the 2013 IEEE17th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp.557-562. IEEE, 2013.