# A

**MAJOR PROJECT REPORT**

**on**

# CREDIT CARD FRAUD DETECTION USING

# MACHINE LEARNING

**BACHELOR OF TECHNOLOGY**

**in**

# COMPUTER SCIENCE AND ENGINEERING

**Submitted by (MP-B24)**

## SATYA MATTAPARTHI:207Y1A05A5

**Under the Guidance of**

**Dr. A. BRAHMA REDDY**

**Associate Professor**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**MARRI LAXMAN REDDY**

**INSTITUTE OF TECHNOLOGY AND MANAGEMENT (AUTONOMOUS)**

**(Affiliated to JNTU-H, Approved by AICTE New Delhi and Accredited by NBA & NAAC With ‘A’ Grade)**

**MARCH 2024**



|  |  |
| --- | --- |
| **CERTIFICATE** | |
| This is to certify that the Major Project report titled **“Credit card fraud detection using machine learning”** is being submitted by **Satya Mattaparthi (207Y1A05A5**) and **Bimini Prashanth (207Y1A0591)** in IV B.Tech II Semester **Computer Science & Engineering** is a record Bonafide work carried out by them. The results embodied in this report have not been submitted to any other University for the award of any degree. | |
| **Internal Guide** | **HOD** |
| **Principal** | **External Examiner** |



**DECLARATION**

|  |
| --- |
| We hereby declare that the Major Project Report entitled, “**Credit Card Fraud Detection using Machine Learning”** submitted for the B. Tech degree is entirely our work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree. |
| **Date:**  **Satya Mattaparthi(207Y1A05A5)**  **Bimini Prashanth(207Y1A0591)** |



**ACKNOWLEDGEMENT**

We are happy to express my deep sense of gratitude to the Director of the college **Dr.**

**K. Venkateswara Reddy,** Professor, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for having provided me with adequate facilities to pursue my project. We would like to thank **Mr. Abdul Basith Khateeb,** Assoc. Professor and Head, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library. We are very grateful to my project guide **Dr. A. Brahma Reddy,** Professor, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for her extensive patience and guidance throughout my project work.

We sincerely thank my seniors and all the teaching and non-teaching staff of the Department of Computer Science for their timely suggestions, healthy criticism and motivation during the course of this work.

We would also like to thank my classmates for always being there whenever we needed help or moral support. With great respect and obedience, we thank my parents and sister who were the backbone behind my deeds. Finally, we express my immense gratitude with pleasure to the other individuals who have either directly or indirectly contributed my need at right time for the development and success of this work.



# CONTENTS

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **TITLE** | | **PAGE NO.** |
|  | COVER PAGE | | **i** |
|  | CERTIFICATE | | ii |
|  | DECLARATION | | iii |
|  | ACKNOWLEDGE | | iv |
|  | ABSTRACT | | viii |
|  | LIST OF FIGURES | | ix |
|  | SYMBOLS & ABBREVIATIONS | | x,xi |
| **1** | **INTRODUCTION** | | **1-2** |
|  | 1.1 | Motivation | 1 |
|  | 1.2 | Scope | 2 |
|  | 1.3 | Problem Definition | 2 |
|  | 1.4 | Objective | 2 |
|  | 1.5 | Limitations | 2 |
|  | 1.6 | Data Set | 2 |
| **2** | **LITERATURE SURVEY** | | **3-11** |
|  | 2.1 | Introduction: Previous study with at least 10 journal papers / models comparison | 3-7 |
|  | 2.2 Over view | | 7 |
|  | 2.3 | Existing System | 7 |
|  | 2.4 | Disadvantages of Existing System | 8-9 |
|  | 2.5 Proposed System | | 9 |
|  | 2.6 Advantages of Proposed System | | 9,10 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2.7 Summary | | 10 |
|  | 2.8 Performance Requirement | | 11 |
| **3** | **ANALYSIS** | | **12-20** |
|  | 3.1 | System Analysis | 12-14 |
|  | 3.2 | Feasibility Study | 15 |
|  | 3.3 | Functional Requirements | 16 |
|  | 3.4 | Software Requirements | 17-18 |
|  | 3.5 | Hardware Requirements | 19 |
|  | 3.6 | Summary | 20 |
| **4** | **DESIGN** | | **21-30** |
|  | 4.1 | System Architecture | 21 |
|  | 4.2 | UML Diagrams | 22-27 |
|  | 4.3 | Modules used in project | 28-29 |
|  | 4.4 | Summary | 30 |
| **5** | **IMPLEMENTATION** | | **31-42** |
|  | 5.1 | Language used | 31-34 |
|  | 5.2 | Method/Algorithm used | 35-36 |
|  | 5.3 | Summary | 37 |
|  | 5.4 | Modules implementation Code | 38-42 |
| **6** | **RESULTS AND DISCUSSIONS** | | 43-47 |
|  | 6.1 | Table and Graphs of results | 43-44 |
|  | 6.2 | Results comparison and graphs | 45 |
|  | 6.3 | Final Output | 46-47 |
| **7** | **TESTING AND VALIDATION** | | **48-52** |
|  | 7.1 | Types of Testing | 48-50 |
|  | 7.2 | Design of Test Cases Scenarios & Validation | 51 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 7.3 | Summary | 52 |

|  |  |  |
| --- | --- | --- |
| **8** | **CONCLUSION** | **53** |
| **9** | **FUTURE ENHANCEMENTS** | **54** |
| **10** | **REFERENCES** | **55** |
| **11** | **APPENDIX A** | **56-58** |

**ABSTRACT**

Whenever we hear the word Credit Card the first thing that pops in our mind is the frauds that are associated with these cards. Credit card has become an indispensable part of our lives. Although a credit card has many advantages when used in a proper manner but damages can be caused to it by many fraudulent activities as well. But in today’s advanced world these frauds can be detected with a vast knowledge of machine learning algorithms. The Credit Card Anomaly Detection Problem includes modeling past credit card transactions with the ones that turned out to befraud. After the implementation of this model we can use it further to identify, a new transaction that is occurring as fraudulent or not. Basically our focus here is to detect 100% fraud transactions that is being occur by minimizing the incorrect fraud classification.­­­ This detection process is a typical example of classifications. This process involve the analysis and the pre-processing of data sets as well as the utilization of multiple Anomaly detection algorithms such as Local Outlier Factor, Super Vector Machine and many such relevant algorithms.

In today's world this is the major concern, which demands the attention of the fields such as Machine Learning, Artificial Intelligence, Deep Learning etc. where the solution of this issue can be automated. Our aim is to predict the accuracy/precision of the fraud detection through different algorithms. Further this analysis can be used to implement the fraud detection model.

Keywords—Credit Card Fraud Classification, Fraud Detection Techniques, Python, Artificial Intelligence, Machine Learning Algorithm, Data Science, Dataset, Comparative Analysis.

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIG. NO** | **FIG. NAME** | **PAGE NO.** |

|  |  |  |
| --- | --- | --- |
| Fig 1 | System Architecture | 21 |
| Fig 2 | Class Diagram | 22 |
| Fig 3 | Use Case Diagram | 23-24 |
| Fig 4 | Sequence Diagram | 25 |
| Fig 5  Fig 6 | Activity Diagram  Component Diagram | 26  27 |
| Fig 7 | Graph 1 | 29 |
| Fig 8 | Graph 2 | 29 |
| Fig 9 | Graph 3 | 30 |
| Fig 10 | Output 1 | 36 |
| Fig 11 | Output 2 | 37 |
| Fig 12 | Output 3 | 37 |
| Fig 13 | Base paper 1 | 57 |
| Fig 14 | Base paper 2 | 58 |

**SYMBOLS AND ABBREVIATIONS**

**ML :** **Machine Learning:**

**CC :** **Credit Card:**

**LR : Logistic Regression**

**DT : Decision Tree:**

**RF : Random Forest:**

**GBM : Gradient Boosting Machine**

**ANN : Artificial Neural Network**

**PCA : Principal Component Analysis**

**SMOTE : Synthetic Minority Over-sampling Technique**

**ROC : Receiver Operating Characteristic curve**

**AUC : Area Under the Curve**

**F1 : F1-score**

**TPR : True Positive Rate**

**FPR : False Positive Rate**

**TPR : True Positive Rate**

**POS : Point of Sale data**

**BIN : Bank Identification Number**

**TS : Time Series data**

# 1.INTRODUCTION

# Credit card fraud is an overall term that can be used to define the fraud that may be carry out by any payment card such as credit card or debit card. The basic aim of these fraud is to purchase goodies without paying or to steal money from someone else’s account. The Payment Card Industry Data Security Standard (PCIDSS) is the data security standard created to help businesses process card payments securely and reduce card fraud. There is a rapid growth in the usage of Cards which has led to rise in the fraudulent activities. The process of credit card fraud detection involve the analysis and the pre-processing of data sets as well as the utilization of multiple Anomaly detection algorithms such as Local Outlier Factor, Super Vector Machine and many such relevant algorithms. In today's world this is the major concern, which demands the attention of the fields such as Machine Learning, Artificial Intelligence, Deep Learning etc. where the solution of this issue can be automated. Our aim is to predict the accuracy/precision of the fraud detection through different algorithms. Further this analysis can be used to implement the fraud detection model.

### Motivation

Predicting the success or popularity of apps on Google Play can be a highly lucrative end , driven by several motivations.

**1. Market Research:** Predicting app success helps companies understand market trends and consumer preferences. By successful and unsuccessful apps, developers can gain insights into what features, designs, and functionalities resonate with users.

**2. Monetization:** For developers and app owners, predicting the success of an app can directly impact revenue potential. High-performing apps are more likely to attract advertisers, gain paid users, or generate revenue through in-app purchases.

**3. Resource Allocation:** Predictive models can help allocate resources more effectively. For instance, if an app is predicted to be highly successful, developers may invest more resources in its development, marketing, and support.

1. **Competitive Advantage:** Predicting app success can give companies a competitive edge by allowing them to release apps that are better positioned to succeed in the market. This can lead to increased market share and brand visibility.
2. **Risk Mitigation:** Predictive analytics can help identify potential risks associated with app development and deployment. By forecasting potential challenges or failures, developers can take proactive measures to mitigate these risks.

**6.User Experience Optimization:** Understanding what makes an app successful can lead to better user experiences. By predicting user preferences and behaviour, developers can tailor their apps to meet user needs more effectively, leading to higher user satisfaction and retention.

Overall, predicting the success of apps on Google Play is essential for both developers and businesses looking to maximize their investments, optimize resource allocation, and stay competitive in the rapidly evolving mobile app market.

### Scope

### The scope of credit card fraud detection using machine learning in Python is quite broad and encompasses various aspects of data processing, feature engineering, model selection, and evaluation. Python provides a rich ecosystem of libraries and frameworks for implementing each of these steps, including sci kit-learn, TensorFlow, PyTorch, and Keras for machine learning, as well as pandas and NumPy for data manipulation and analysis. Additionally, tools like Flask or Django can be used for building APIs to integrate the fraud detection model into existing systems.

### Problem Definition

### typically involves identifying fraudulent transactions within a dataset of credit card transactions. Develop a machine learning model to accurately detect fraudulent credit card transactions in a given dataset. The goal is to minimize the number of false positives (legitimate transactions incorrectly classified as fraudulent) and false negatives (fraudulent transactions incorrectly classified as legitimate) while maximizing the overall accuracy of the model. a machine learning model capable of accurately identifying fraudulent credit card transactions with high precision and recall, thereby helping financial institutions reduce losses due to fraud and enhance security for their customers.

### Top of Form

### Objectives

### The objective of credit card fraud detection using machine learning with Python is to develop a predictive model that can effectively identify fraudulent transactions within credit card data. By achieving these objectives, machine learning-based credit card fraud detection systems can play a crucial role in safeguarding financial transactions and minimizing the impact of fraudulent activity on individuals and businesses.

### Limitations

### Credit card fraud datasets are often highly imbalanced, with a small proportion of fraudulent transactions compared to legitimate ones. This imbalance can lead to biased models that favor the majority class and struggle to accurately detect fraudulent transactions. Fraudulent patterns and techniques evolve over time, leading to concept drift. ML models trained on historical data may become less effective at detecting new types of fraud if they are not regularly updated and retrained with fresh data. The quality of the data used for training ML models is crucial. Incomplete, inaccurate, or noisy data can hinder model performance and lead to unreliable predictions. Additionally, missing or incorrect labels in the dataset can impact the model's ability to learn effectively. Addressing these limitations requires a combination of advanced modeling techniques, robust data preprocessing, ongoing monitoring and model maintenance, collaboration between domain experts and data scientists, and a holistic approach to fraud detection that combines ML with other methods such as rule-based systems and anomaly detection.

### Data Set

### 

### For credit card fraud detection using machine learning with Python, you can use various datasets. One

### commonly used dataset is the Credit Card Fraud Detection dataset available on Kaggle. Here's how

### you can access it:

### Go to the Kaggle website. Search for "Credit Card Fraud Detection" in the dataset search bar.

### Choose the dataset with the appropriate features and size that suits your needs. Download the

### dataset, and you can then use it in your Python environment for building your machine learning

### model. Alternatively, you can also use the "creditcard.csv" dataset available in the "Anomaly Detection"

### repository on GitHub. This dataset contains credit card transactions labeled as fraudulent or genuine.

This will load the dataset into a pandas Data Frame, which you can then use for building your fraud detection

model using machine learning algorithms in Python. Remember to preprocess the data and handle any class

imbalance before training your model.

# LITERATURE SURVEY

# 

* + - * 1. **Introduction: Previous Study with journal papers/models comparison**

**Mobile Application Rating Prediction via Feature Oriented Matrix Factorization**

With the proliferation of mobile application (app) markets (e.g., Google Play, Apple App Store), predicting user preferences on apps becomes a challenging problem. Different from previous work, we assume that a user likes an app because he/she likes certain features of the app (e.g., permission, genre, topic). A user likes an app because he likes several features of the app rather than all the features. Our contribution is to transform the user-app rating prediction problem to user-feature rating prediction problem, and utilize the prediction result to derive user ratings on apps through two effective integration strategies.

**Securing Android App Markets via Modelling and Predicting Malware Spread between Markets**

The Android ecosystem has recently dominated mobile devices. Android app markets, including official Google Play and other third

party markets, are becoming hotbeds where malware originates and spreads. Android malware has been observed to both propagate within markets and spread between markets. If the spread of Android malware between markets can be predicted, market administrators can take appropriate measures to prevent the outbreak of malware and minimize the damages caused by malware. In this paper, we make the first attempt to protect the Android ecosystem by modelling and predicting the spread of Android malware between markets. In this paper, we conducted the first

attempt to explore the mechanism of malware propagation and spread. Based on the understanding, we studied the spread of Android malware within and between markets from a huge number of Android apps infected with malware, and proposed comprehensive models to simulate the spread behavior of Android malware.

**Android Application Market Prediction Based on User Ratings Using KNN**

—Google play store is a digital distribution service operated and developed by Google. It provides various applications that can be downloaded directly on Android devices, allowing users to browse and download their desired applications. When they search for an app, users can see the list of applications with the name and its rating on its side. The most review of the application category is the education category, and the highest rating is the book and reference category, followed by the game word category. In contrast, the lowest rating goes to the dating category. From the results shown above, KNN is a fairly good performance, which has an accuracy of 88.68%, recall of 87.46%, the precision of 89.5%, and RMSE of 0.213 in predicting the android application market.

**Rating Prediction of Google Play Store Apps with Application of Data Mining**

**Techniques**

Software development is based on implementation standards. In the case of selling and accepting software by customers, it has been a challenge to develop applications for marketplaces. App stores have features such as the number of downloads, comments, and ratings on ratings. From this, the difficulties were the fields (previously listed) and their way of analyzing the problem, thus resulting in characteristics that define the pattern of success in apps. Based on this scenario, this work aimed to create two inference engines from the KNN and Random Forest algorithms and, with that.

**App Popularity Prediction by Incorporating Time- Varying Hierarchical Interactions**

App popularity prediction is a significant task in mobile service development, which predicts an app’s future popularity based on its current behaviors. It provides benefits from app development to targeted investment. Popularity is affected by two factors, i.e., internal ones like reviews and external ones like interaction among apps. However, most related studies only explore internal factors but neglect external ones. In fact, external factor plays an important role in popularity prediction modelling since it is the promoting and/or inhibiting influence resulted by app interaction. This paper proposes DPOP - a popularity prediction model that innovatively integrates time-varying hierarchical interaction to enhance prediction performance. First, Hierarchical Interaction Graph is constructed based on the relationship and influence among apps.

# 2.1 TABLE-SUREVY TABLE-2.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.NO  . | Year | Author | Title | Algorithm | Input | Output |
| 1 | 201  7 | Tingting Liang,  Liang Chen,  Xingde Yin | Mobile Application Rating Prediction via Feature- Oriented Matrix Factorizatio n | Feature- Oriented Matrix Factorizatio n | Users Apps, Ratings, Sparsity, Permissions | 7,536,11,390 59,608  ,99.93%, 84 |
| **2** | 2019 | Guozhu Meng, Matthew Patrick , Yinxing Xue , Yang Liu  , Jie Zhang SKLOS | Securing Android App Markets via Modelling and Predicting Malware Spread between Markets | Spread Model of Android Malware | Name, Region, Number, Day Range, Name, Number, Day Range | APK20, U.S, 5,241  2,999, DOMOB,  5,718,  2,973 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 2020 | Green Arther Sandag, Fidy Gara | Android Application Market Prediction Based on User Ratings Using KNN | K-Nearest Neighbour Algorithm | KNN  Accuracy(%) Recall(%) Precission(%) RMSE | 1, 80,55,  81,3,  81,49  0,294 |
| 4 | 2021 | R. G. da Silva, J. de O. L Magalhae s, I. R. R.  Silva, R.  A. de A. Fagundes,  E. A. de  O. Lima and M. A.  ˜ MACIE | Rating Prediction of Google Play Store Apps with Application of Data Mining Techniques | Cross- Industry Standard Process of Data Mining, KNN | Metric, W, p- value | Acuracia0 4.320464057827488 e-08 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | 2022 | Yixuan Zhang, Jiaqi Liu | App Popularity Prediction by Incorporatin g Time- Varying Hierarchical Interactions | Mobile App Analysis | Level 1 label, App number, Level 2 label, Average downloads, Average reviews | Shopping, 120, 5  38,901,915,  571 |

## OVERVIEW

There is a wide variety of information available on internet and it has been booming in the world, including textual sources, people’s thoughts on review sites, blog and other social media platforms. Evaluation- based prediction systems make unstructured data automatically converted into structured information through public opinions. Many papers worked on app related datasets with the Main focus on security problems, customer responses, software performances and its version power. However, little study has done on forecasting the best app of google play store with the review and rating based.

## EXISTING SYSTEM

The google play store is one of the famous platforms which consider the central hub of apps and game content. Today, it is one of the topmost app markets in the world and quite popular among millions of people. To make the apps successful on the google play store, developers work individually or in a group, however, popular apps are generally estimated by their rate of installation, the number of comments and the number of ratings. As the app store market is growing, the demand for Android app creators is also increasing.

The google play store allows a user to install a different kind of apps as per their interests which are created with the Android software development kit and have access to rate and comment on apps according to their experiences. A customer can discover the type of application for example both paid or unpaid programs that are available on the google play store platform. The outstanding growth of mobile application market has a great impact on today’s generation and digital technology.

### Disadvantages of Existing system

**Data Quality and Bias**: Existing prediction models heavily rely on historical app data. However, this data may be biased towards certain types of apps or demographics, leading to skewed

predictions. Biases in data collection and sampling methods can result in inaccurate predictions, as the models may not capture the full diversity of user preferences and behaviors.

**Limited Scope of Features:** Many existing prediction models focus on a limited set of features

such as app ratings, downloads, and user reviews. While these features provide valuable insights, they may not fully capture the complex dynamics that influence app success, such as user engagement, retention, and monetization strategies. This can result in oversimplified predictions that fail to account for the multifaceted nature of app success.

**Inability to Adapt to Changing Trends**: Existing prediction models may struggle to adapt to rapidly changing market trends and user preferences. The dynamic nature of the mobile app ecosystem requires models to continuously update and evolve to remain accurate. Failure to incorporate emerging trends and technologies can lead to outdated predictions that do not reflect the current state of the market.

**Lack of Contextual Information**: Predictive models often lack contextual information about individual apps and their target audiences. Factors such as app category, target demographic, competition, and marketing strategies can significantly influence an app's success but may not be adequately accounted for in existing models. This can result in predictions that are overly generalized or irrelevant to specific app contexts.

**Overreliance on Quantitative Metrics**: While quantitative metrics such as app ratings and downloads provide valuable insights, they may not capture the qualitative aspects of app success, such as user satisfaction, brand reputation, and user experience. Existing prediction models may overlook these qualitative factors, leading to incomplete or inaccurate predictions of app success.

**Ethical Considerations**: Predictive models raise ethical concerns related to privacy, fairness, and transparency. The use of user data to train and deploy prediction models raises privacy issues, particularly regarding data collection, storage, and usage. Moreover, biased predictions can perpetuate existing inequalities and discrimination, leading to unfair outcomes for certain app developers or user groups.

## PROPOSED SYSTEM

Data scientists use several of machine learning algorithms and techniques in order to achieve the best results. To efficiently acquire and the large amount of data, applying ML different algorithms will be beneficial for predicting and examine the best results from the given dataset. For forecasting the best software on the google play store, the major factors will be considered app reviews, number of installs, content ratings, Interactive Element and apps name.

### Advantages of Proposed system

**Advanced Data Analysis Techniques:** The proposed system leverages advanced data analysis techniques such as machine learning algorithms, natural language processing, and deep learning to extract valuable insights from diverse data sources. By incorporating a wider range of features and data points, the system can provide more accurate predictions of app success.

**Real-Time Updates**: Unlike traditional prediction models that rely on static datasets, the proposed system can continuously update and evolve based on real- time data streams. This allows the system to adapt to changing market trends, user preferences, and competitive dynamics, ensuring that predictions remain relevant and up-to-date.

**Personalized Recommendations:** The proposed system can generate personalized recommendations for app developers based on their specific needs and objectives. By analyzing individual app profiles, target demographics, and business goals, the system can offer tailored insights and strategies to maximize the success of each app.

**Multi-Dimensional Analysis**: The proposed system takes a multi-dimensional approach to app success prediction, considering a wide range of factors such as app features, user engagement metrics, competitor analysis, and market trends. This holistic perspective provides a more comprehensive understanding of app success drivers and enables more accurate predictions.

**Ethical Considerations**: The proposed system incorporates ethical considerations such as privacy, fairness, and transparency into its design and implementation. By adhering to strict data privacy standards, mitigating biases, and ensuring transparency in the prediction process, the system upholds ethical principles and promotes trust among users and stakeholders.

**Scalability and Flexibility**: The proposed system is designed to be scalable and flexible, allowing it to accommodate a growing volume of app data and evolving user behaviors. Whether analyzing individual apps or entire app categories, the system can scale seamlessly to meet the needs of different stakeholders and adapt to

changing market conditions.

**Actionable Insights**: The proposed system not only predicts app success but also provides actionable insights and recommendations for app developers to improve their app's performance. By identifying strengths, weaknesses, and opportunities, the system empowers developers to make informed decisions and optimize their app's chances of success on Google Play.

### Summary

In summary, the proposed system for predicting app success on Google Play offers several advantages, including advanced data analysis techniques, real-time updates, personalized recommendations, multi-dimensional analysis, ethical considerations, scalability, flexibility, and actionable insights. By leveraging these advantages, the system can help app developers make more informed decisions and enhance the overall success of their apps on Google Play.

## PERFORMANCE REQUIREMENT

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirements specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirements specifications because they are the people who finally use the system. This is because of the requirements had to be known during the initial stages so that the system can be designed, according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The System should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

# ANALYSIS

**TABLE 3.1-System Analysis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO.** | **Year** | **Title** | **Adavantage** | **Disadvantage** | **Conclusion** |
| 1 | 2017 | Mobile Application Rating Prediction via Feature- Oriented Matrix Factorization | As it integrates feature information to analyze the details of user preference, it can generalize better as the feature rating data is denser, and improve the interpretation of the prediction of app ratings. | Our contribution is to transform the user-app rating prediction problem to user-feature rating prediction problem, and utilize the prediction result to derive user ratings on apps through two effective integration strategies. | The previous work about app recommendatio n mainly focus on users or apps, and features of apps are generally considered as additional information to improve the quality of user  preferences prediction. |
| 2 | 2019 | Securing Android App Markets via Modelling and Predicting Malware Spread between Markets | The experimental results show our approach can depict and simulate the growth of Android malware on a large scale, and predict the infection time and order among markets with  0.89 and 0.66  precision, respectively. | Family labeling accuracy. In this paper, we propose a family specific growth model and spread model between markets. Therefore, the fitness of models to some extent, relies on the accuracy of family labeling of Android malware | We conducted the first attempt to explore the mechanism of malware propagation and spread.  Based on the understanding, we studied the spread of Android malware within and between markets from a huge number of Android apps  infected with malware. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3 | 2020 | Android Application Market Prediction Based on User Ratings Using KNN | This research is designed with Knowledge Discovery in Database (KDD). KDD is  a form of an automatic analysis and | Further research is recommended to get higher accuracy. Further research is also recommended to develop this  machine learning into software or | The most review of the application category is the education category, and the highest  rating is the book and |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | repositories | web-based system. | reference |
| modeling from | For researchers, it | category, |
| big data, which | is suggested to use | followed by the |
| is an organized | other algorithms | game word |
| process of identifying | and compare them to get better results | category**.** |
| valid, new, | on the next |  |
| useful, and | research. |  |
| understandable |  |  |
| patterns from a |  |  |
| big and complex |  |  |
| dataset. |  |  |
| 4 | 2021 | Rating | Most of the | Looks like there | we hope you enjoyed learning about Data Mining process on a dataset, and we enjoyed rattling your brain. We also had an amazing time writing this article and will continue sharing about Data Mining**.** |
| Prediction of | data scientist | are missing |
| Google Play | use python | values in |
| Store Apps | because of the | “Rating”, “Type”, |
| with | great built-in | “Content Rating” |
| Application | library | and “Android |
| of Data Mining Techniques | functions and the decent  community. Python now | Ver”. But most of these missing  values in Rating column. First, we |
|  | has 70,000 | checked to see if |
|  | libraries. | there were any |
|  | Python is | missing |
|  | simplest | values/NaN |
|  | programming | values. |
|  | language to |  |
|  | pick up |  |
|  | compared to |  |
|  | other language. |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | 2022 | App Popularity Prediction by Incorporating Time- Varying Hierarchical Interactions | The attention mechanism in the modules assigns weights for different factors during the encoding process, which is useful to improve prediction performance and provide abundant explanation for  the prediction results | The prediction performance of DeePOP variants laterally reflect the importance of different modules for popularity modeling. It can be observed that the DeePOPInn\_Ti has the worst performance among variants, with DeePOP-Inn. | This paper proposes DeePOP - a popularity prediction model that innovatively integrates time- varying hierarchical interaction to enhance prediction performance. |

## FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understandingof the major requirements for the system is essential.

#### ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, The developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

#### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

#### SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by theusers solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of system

## FUNCTIONAL REQUIREMENTS

#### GRAPHICAL USER INTERFACE WITH THE USER.

GUI is an interface that allows users to interact with different electronic devices using icons and other visual indicators. The graphical user interfaces were created because command line interfaces were quite complicated and it was difficult to learn all the commands in it.

#### ELEMENTS IN GRAPHICAL USER INTERFACE

**WINDOW:** This is the element that displays the information on the screen. It is very easy to manipulate a window. It can be opened or closed with the click of an icon. Moreover, it can be moved to any area by dragging it around. In multitasking environment, multiple windows can be open at the same time, all of them performing different tasks. There are multiple types of windows in a graphical user interface, such as container window, browser window, text terminal window, child window, message window etc.

**MENU:** A menu contains a list a choice and it allows users to select one from them. A menu bar is displayed horizontally across the screen such as pulldown menu. When any option is clicked in this menu, then the pulldown menu appears. Another type of menu is the context menu that appears only when the user performs a specific action. An example of this is pressing the right mouse button. When this is done, a menu will appear under the cursor.

**ICONS:** Files, programs, web pages etc. can be represented using a small picture in a graphical user interface. This picture is known as an icon. Using an icon is a fast way documents, run programs etc. because clicking on them yields instant access.

**CONTROLS:** Information in an application can be directly read or influences using the graphical control elements. These are also known as widgets. Normally, widgets are used to display lists of similar items, navigate the system using links, tabs etc.

## SOFTWARE REQUIREMENTS

#### PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamicbinding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

#### FLASK

Flask is a web frame work, it is a Python module that lets you develop web applications easily.It is had a small and easy-to-extend core: it is a microframework that doesn’t include an ObjectRelational Manager or such features. Flask is a web framework that provides libraries to buildlightweight web applications in python. It is developed by Armin Ronacher who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

#### WINDOWS 7

Windows 7 is an operating system that Microsoft has produced for use on personal computers. It is the follow-up to the Windows Vista Operating System, which was released in 2006. An operating system allows your computer to manage software and perform essential tasks. It is also a Graphical User Interface (GUI) that allows you to visually interact with your computer’s functions in a logical, fun, and easy way.

#### WINDOWS XP

Windows XP is an operating system that lets you use different types of applications or software. For example, it allows you to use a word processing application to write a letter and a spreadsheet application to track your financial information. Window XP is a graphical user interface (GUI). It has pictures (graphical) that you use (user) to communicate (interface) with the computer. This type of system is popular because it is logical, fun, and easy to use.

#### WINDOWS 8

Windows 8, code-named Midori, is a version of Microsoft Windows that was released on October 26, 2012. This iteration contains some of the biggest changes Microsoft has made their operating system since Windows 95. It features new programming and technology that makes run faster than the previous versions. It also has a more streamlined look and feel.

Windows 8 includes a tablet interface, called Metro, which is compatible with touch screendisplays and stills offers access to the traditional Windows desktop.

#### WEB BROWSER (CHROME)

Chrome is a free Internet browser officially released by Google on December 11, 2008. Its features include synchronization with Google services and accounts, tabbed browsing, and automatic translation and spell check of web pages. It also features an integrated address bar/search bar, called the omnibox. Chrome works quite well with Google sites and services such as YouTube and Gmail. It also manages its system resources differently than other browsers. Your computer can access RAM memory much faster than data on a hard disk, SSD, or other long-term storage device, which is why RAM capacity is critical for system performance. Every computing device has RAM, whether it’s adesktop computer (running Windows, MacOS, or Linux), a tablet or smartphone (running Android or iOS), or even an IoT computing device (like a smart TV). The series was designed by Intel and launched in November 2000. Pentium 4 clock speeds were over 2.0 GHz .Intel shipped Pentium 4 processors until August 2008. Pentium 4 variants included code named Willamette, Northwood, Presscott and Cedar Mill with clock speeds that varied from 1.3-3.8 GHz.

## HARDWARE REQUIREMENTS

#### PROCESSOR: (PENTIUM IV OR HIGHER)

Pentium 4 was a series of single-core central processing units (CPU) for desktop PCs and laptops. The series was designed by Intel and launched in November 2000. Pentium 4 clock speeds were over 2.0 GHz .Intel shipped Pentium 4 processors until August 2008. Pentium 4 variants included code named Willamette, Northwood, Presscott and Cedar Mill with clock speeds that varied from 1.3-3.8 GHz.

#### RAM: (256 MB)

RAM (random access memory) is a computer's short-term memory, where the data that the processor is currently using is stored. Your computer can access RAM memory much faster than data on a hard disk, SSD, or other long-term storage device, which is why RAM capacity is critical for system performance. Every computing device has RAM, whether it’s adesktop computer (running Windows, MacOS, or Linux), a tablet or smartphone (running Android or iOS), or even an IoT computing device (like a smart TV).

#### SPACE ON HARD DISK: (MINIMUM 512MB)

Hard disks are flat circular plates made of aluminum or glass and coated with a magnetic material. Hard disks for personal computers can store terabytes (trillions of bytes) of information. Data are stored on their surfaces in concentric tracks. A small electromagnet, called a magnetic head, writes a binary digit (1 or 0) by magnetizing tiny spots on the spinning disk in different directions and reads digits by detecting the magnetization direction of the spots.

### Summary

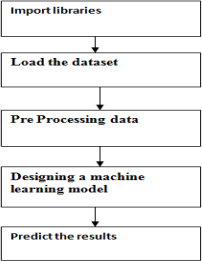
Google Play employs sophisticated analysis techniques to predict app recommendations for users. This analysis involves gathering and processing various types of user data, including app installs, ratings, reviews, and user interactions. By employing collaborative filtering, the system identifies patterns among users with similar behavior to recommend apps of interest. Content-based filtering analyzes app characteristics such as category, description, and developer information to make personalized recommendations based on user preferences. Additionally, machine learning models are utilized to continuously refine and improve recommendation accuracy based on large datasets. Natural language processing techniques extract semantic information from app descriptions and reviews to further enhance recommendation quality. Through these comprehensive analysis methods, Google Play ensures that users receive tailored app suggestions that align with their preferences and interests, ultimately enhancing the overall user experience.

Google Play's app prediction analysis encompasses a multifaceted approach that integrates numerous techniques and data sources. Beyond the fundamental methods like collaborative and content-based filtering, Google Play leverages user context, engagement signals, and multi-armed bandit algorithms to refine recommendations. Additionally, feature engineering plays a crucial role, transforming raw data into meaningful features for machine learning models. Continuous improvement is achieved through A/B testing, where different recommendation strategies are rigorously evaluated.

# SYSTEM DESIGN

## SYSTEM ARCHITECTURE

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages.



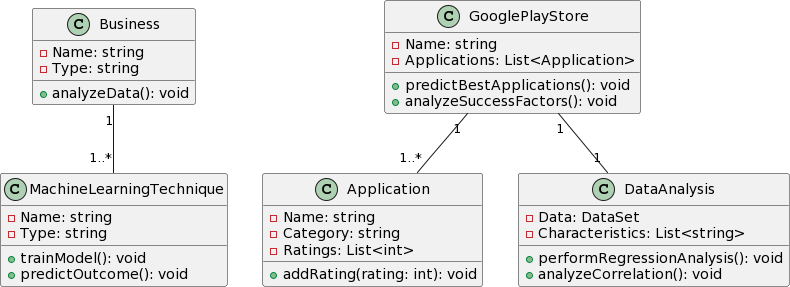
**FIGURE 4.1- System Architecture**

## UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general- purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

#### CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**Figure 4.2.1-Class Diagram**

#### USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

**Actors:** Business, Data Analyst

**Description:** This use case involves predicting application ratings on the Google Play Store platform using machine learning techniques such as KNN and Random Forest regression.

Perform Statistical Analysis:

**Actors:** Data Analyst

**Description:** This use case involves performing statistical analysis on Google Play Store applications, including hypothesis testing, correlation analysis, and regression metrics analysis.

Analyze Success Factors:

**Actors:** Data Analyst

**Description:** This use case involves analyzing the characteristics responsible for the success of applications on the Google Play Store platform.

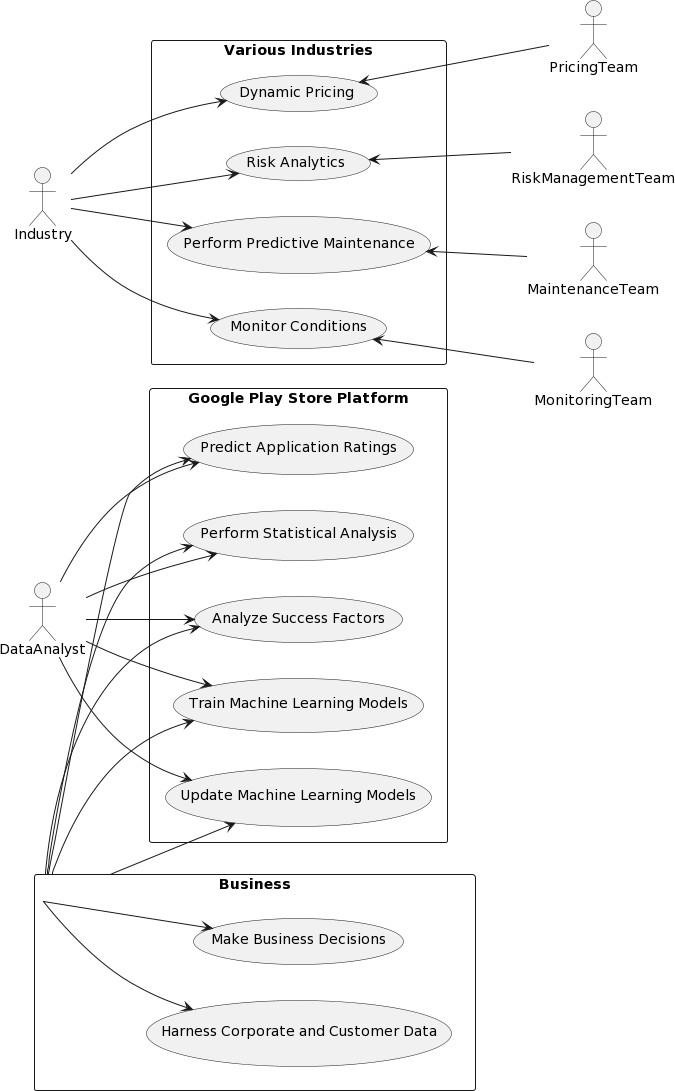
Train Machine Learning Models:

**Actors:** Data Analyst

**Description:** This use case involves training machine learning models using datasets from the Google Play Store platform.

Update Machine Learning Models:

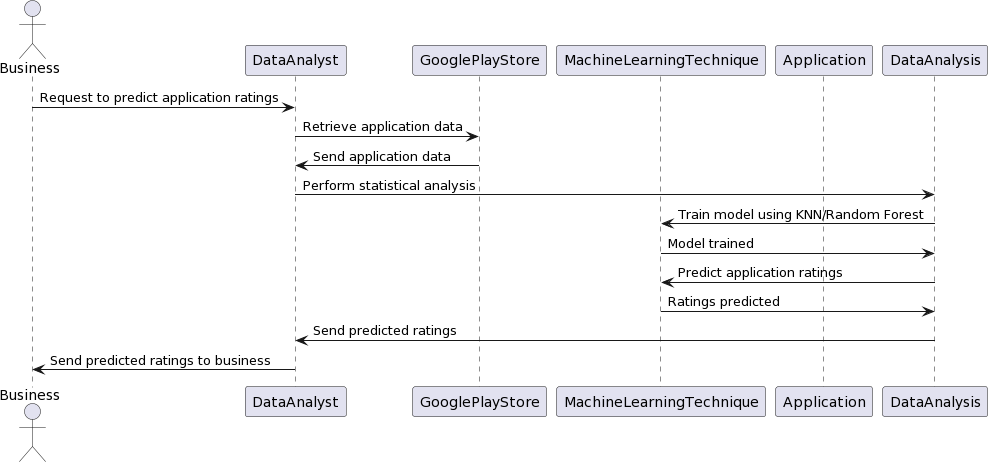
.



**FIGURE 4.2.2-Use Case Diagram**

## SEQUENCE DIAGRAM

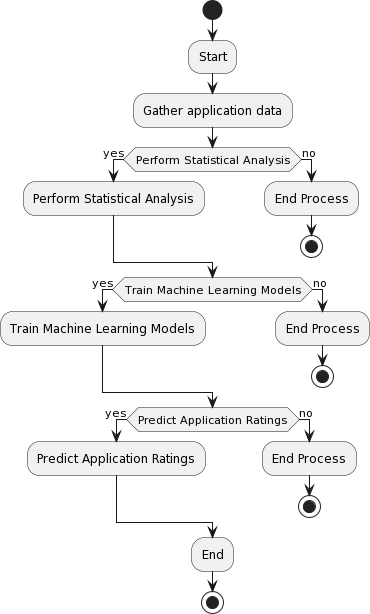
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**Figure-4.2.3-Sequence Diagram**

#### ACTIVITY DIAGRAM

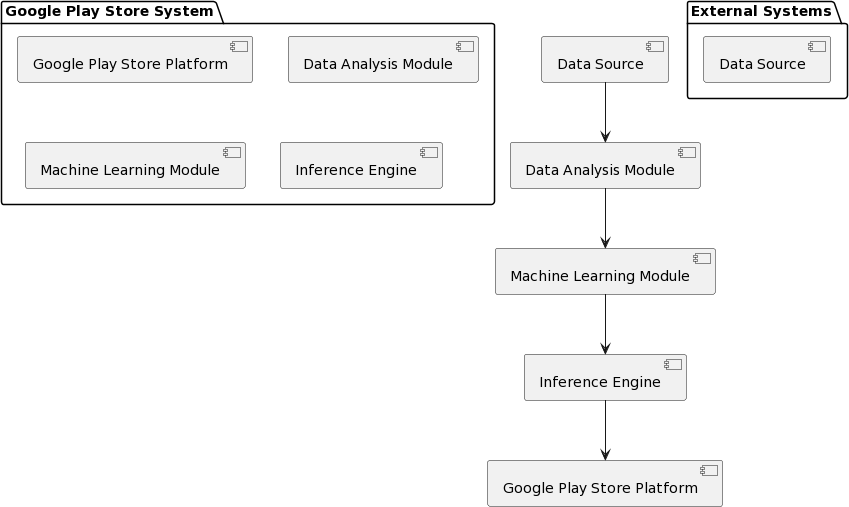
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**Figure-4.2.4-Activity Diagram**

**Component Diagram**

A component diagram serves as a visual representation of the system architecture and the interactions between its key components. At the heart of the system lies the Google Play Store Platform component, which forms the foundation for hosting applications and managing user interactions. The Data Analysis Module is responsible for collecting and processing data related to application ratings, user reviews, and download statistics. This data is then fed into the Machine Learning Module, which houses various machine learning algorithms and models such as KNN and Random Forest. These algorithms are trained using historical data to predict application ratings based on various factors. The predictions are generated by the Inference Engine component, which applies the trained models to incoming data and produces the anticipated ratings. External data sources, represented by the Data Source component, provide additional input data for analysis and prediction, enriching the system's capabilities.



**Figure-4.2.5 Component Diagram**

## MODULES USED IN PROJECT

#### TENSORFLOW

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

#### NUMPY

Numpy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

>A powerful N-dimensional array object

>Sophisticated (broadcasting) functions

>Tools for integrating C/C++ and Fortran code

>Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi- dimensionalcontainer of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

#### PANDAS

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Pythonwith Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

#### MATPLOTLIB

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can beused in Python scripts, the Python and I Python shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots andthumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly whencombined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc via an object oriented interface or via a set of functions familiar to MATLAB users.

#### SCIKIT – LEARN

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and isdistributed under many Linux distributions, encouraging academic and commercial use.

### Summary

In the realm of predicting application ratings on the Google Play Store platform, various modules serve distinct functions to facilitate accurate analysis and forecasting. Firstly, the Data Collection Module is tasked with gathering pertinent data encompassing application metadata, user ratings, reviews, and download statistics. Subsequently, the Data Preprocessing Module refines this data through tasks like normalization, handling missing values, and feature engineering to prepare it for analysis. The Statistical Analysis Module then delves into the data, employing techniques such as hypothesis testing, correlation analysis, and regression analysis to discern patterns and influential factors affecting application ratings. Following this, the Machine Learning Module takes center stage, housing algorithms like KNN and Random Forest, which are trained on historical data to predict ratings based on application characteristics. Once models are trained, the Inference Engine applies them to incoming data, generating predictions that guide decision-making. Moreover, a Feedback Loop Module ensures model refinement by incorporating user feedback and updating models dynamically. Finally, the Deployment and Integration Module ensures seamless incorporation of predictive models into the Google Play Store platform, facilitating real-time prediction and decision-making processes. Collectively, these modules constitute a comprehensive framework for analyzing data, training predictive models, and enhancing decision-making in the realm of Google Play apps prediction.

By leveraging these modules, businesses and data analysts can effectively analyze application data, train predictive models, and make informed decisions to enhance the performance and success of applications

on the Google Play Store platform. In the domain of Google Play apps prediction, several modules play pivotal roles in analyzing data,

training machine learning models, and making accurate predictions. Here's a summary of the key modules used in this context:

# IMPLEMENTATION

## LANGUAGE USED

#### PYTHON

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Pythonprograms generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Machine Learning.

#### ADVANTAGES OF PYTHON EXTENSIVE LIBRARIES

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers,

threading,databases, CGI, email, image manipulation, and more. So, we donot have to write the complete code for that manually.

#### EXTENSIBLE

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

#### EMBEDDABLE

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

#### IMPROVED PRODUCTIVITY

The language’s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### IOT OPPORTUNITIES

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

#### SIMPLE AND EASY

When working with Java, you may have to create a class to print ‘Hello World’. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### READABLE

Because it is not such a verbose language, reading Python is much like reading English.This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

#### OBJECT-ORIENTED

This language supports both the procedural and object- oriented programming paradigms.

While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

#### FREE AND OPEN-SOURCE

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### PORTABLE

When you code your project in a language like C++, you may need to make some changes to itif you want to run it on another platform. But it isnot the same with Python. Here, you need to code only once, and you can run it anywhere.

#### INTERPRETED

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

#### ADVANTAGES OF PYTHON OVER OTHER LANGUAGES

**LESS CODING**

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third- party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### AFFORDABLE

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used soil gives you better community support. The 2019 Git hub annual survey showed us that Python has overtaken Java in the most popular programming language category.

#### PYTHON IS FOR EVERYONE

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations.

#### DISADVANTAGES OF PYTHON

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let us now see the downsides of choosing Python over another language.

#### SPEED LIMITATIONS

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### WEAK IN MOBILE COMPUTING AND BROWSERS

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle. The reason it is not so famous despite the existence of python is that it isn’t that secure.

#### DESIGN RESTRICTIONS

As you know, Python is dynamically-typed. This means that you don’t need to declare the type of variable while writing the code. It uses duck- typing. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

#### UNDERDEVELOPED DATABASE ACCESS LAYERS

Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and

ODBC (Open DataBase Connectivity), Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### SIMPLE

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

## METHODS/ALGORITHMS USED

#### RANDOM FOREST REGRESSOR ALGORITHM:

Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap. An ensemble method is a technique thatcombines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model.

#### KNN:

KNN classifier is a machine learning algorithm used for classification and regression problems. It works by finding the K nearest points in the training dataset and uses their class to predict the class or value of a new data point.

#### DECISION TREE:

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

#### SVM:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

**Collaborative Filtering:** This technique involves making predictions about the interests of a user by collecting preferences or information from many users (collaborating). Google Play may analyze user interactions such as app installs, uninstalls, ratings, and reviews to recommend similar apps to users who have similar behavior or preferences.

**Content-Based Filtering:** This method recommends items to users based on the characteristics of the items themselves. In the context of Google Play, content-based filtering might involve analyzing the features and metadata of apps (such as app category, description, keywords, developer information) to recommend apps that are similar to ones a user has previously interacted with.

**Matrix Factorization:** This technique is often used in collaborative filtering. It decomposes the user-item interaction matrix into lower-dimensional matrices to capture latent factors or features. By doing so, it can identify patterns and similarities among users and items, enabling personalized recommendations.

**Reinforcement Learning:** In some cases, reinforcement learning techniques might be used to optimize app recommendations over time based on user feedback. Reinforcement learning algorithms learn to maximize a

reward signal (e.g., user engagement or satisfaction) by iteratively adjusting the recommendation strategy.

**Machine Learning Models:** Google Play likely employs various machine learning models to improve recommendation accuracy. These models may include neural networks, decision trees, support vector machines, or ensemble methods trained on large datasets of user interactions and app metadata.

**Natural Language Processing (NLP):** NLP techniques may be employed to analyze app descriptions, user reviews, and other textual data to extract semantic information, sentiment, or user preferences, which can then be used to improve recommendation accuracy.

### Summary:

It's important to note that Google Play's recommendation system is likely a combination of several of these techniques, and the specific algorithms and approaches may evolve over time as Google refines its recommendation system to better serve users. Additionally, the exact details of Google Play's recommendation algorithms are proprietary and not publicly disclosed.

Google Play employs a sophisticated recommendation system that utilizes a combination of algorithms to predict app suggestions for users. Collaborative filtering analyzes user interactions such as installs, ratings, and reviews to recommend similar apps to those with similar behavior. Content-based filtering focuses on app characteristics like category, description, and developer information to suggest apps based on user preferences. Matrix factorization techniques decompose user-item interaction matrices to identify patterns and similarities among users and apps, enabling personalized recommendations.

Machine learning models, including neural networks and decision trees, are trained on large datasets to improve recommendation accuracy. Reinforcement learning may optimize recommendations based on user feedback, while natural language processing extracts semantic information from

app descriptions and reviews. Deep learning models further enhance the system's ability to learn complex patterns in user-app interactions. Together, these algorithms enable Google Play to provide tailored app suggestions that enhance user experience and engagement.

### Sample Code

import pandas as pd import seaborn as sns

import matplotlib.pyplot as plt import numpy as np

import re

import plotly.express as px

df=pd.read\_csv(r'C:\Users\sridevi madhavi\OneDrive\Desktop\python\GooglePlayStoreApps\_Prediction- main\googleplaystore.csv')

df.head(5) df.shape df.describe() df.isnull()

#Plot of null values plt.figure(figsize = (8,6))

sns.heatmap(df.isnull(), cbar=False , cmap = 'magma') #Numerical representation

df.isnull().sum()

# Here we found that there is too much null value in Rating

df.drop(['App','Last Updated','Current Ver','Android Ver'],axis=1,inplace=True) ##We dropped the unnecessary columns like..

df.dropna(inplace=True)

##We dropped unnecessay rows which were having null values

df.head(5)

##Our data set after dropping null values and unnessary rows and columns

df.shape

### Our data set now have 9366 rows and 9 columns

df['Size'] = df.Size.apply(lambda x: x.strip('+'))

df['Size'] = df.Size.apply(lambda x: x.replace(',', ''))

df['Size'] = df.Size.apply(lambda x: x.replace('M', ''))

df['Size'] = df.Size.apply(lambda x: x.replace('k', 'e-3')) df['Size'] = df.Size.replace('Varies with device', 0) df['Size'] = pd.to\_numeric(df['Size'])

df.loc[df.Size == 0, 'Size'] = df.Size.median() df.rename(columns={"Size": "Size\_MB"}, inplace=True)

df['Installs'] = df.Installs.apply(lambda x: x.strip('+')) df['Installs'] = df.Installs.apply(lambda x: x.replace(',', '')) df['Installs'] = pd.to\_numeric(df['Installs'])

df['Price']=df['Price'].apply(lambda x: x.replace('$',''))

###Our data after cleaning some unnecessay strings df.head(5)

sns.set(rc={'figure.figsize':(7,6.8)}) plt.tight\_layout()

plt.show()

plt.figure(figsize=[11, 4]) sns.set\_context('talk')

sns.countplot(x='Rating', data = df,palette="YlOrBr") plt.xticks(rotation=90)

plt.ylabel('Installs') plt.show()

plt.figure(figsize=[8,6])

sns.countplot(x='Category',hue='Type',data=df,order=df.Category.value\_counts().iloc[:21].index,palette='rocket') #plt.figure(figsize=[11, 20])

sns.color\_palette("magma", as\_cmap=True) sns.set\_context('talk') plt.xticks(rotation=90,fontsize=9) plt.ylabel('Installs')

Content\_Ratings = df['Content Rating'].value\_counts() Content\_Ratings

Figure = px.pie(labels=Content\_Ratings.index, values=Content\_Ratings.values, title="Content Ratings", names=Content\_Ratings.index, color\_discrete\_sequence=px.colors.sequential.RdBu)

Figure.update\_traces(textposition='outside', textinfo='percent+label') Figure.show()

plt.hist(x=df['Rating'],bins=30,density=True) #Top 15 Genres and their No of installs

top\_15 = df.groupby(['Genres']).agg({'Installs': "sum"}) df1 = top\_15.reset\_index()

df2 = df1.sort\_values(by=['Installs'], ascending=False) df3 = df2.reset\_index()

df3.drop('index',axis = 1,inplace = True) Genres = df3['Genres'].head(15)

Installs = df3['Installs'].head(15)

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

sns.barplot(Genres, Installs) plt.xticks(rotation=75) plt.xlabel("Genres") plt.ylabel("No. of Installs")

plt.title("Top 15 most installed Genres") plt.show()

plt.figure(figsize=[11, 4]) sns.set\_context('talk') sns.countplot(x='Rating', data = df) plt.xticks(rotation=65)

plt.show()

a=df.nlargest(100, 'Size\_MB')['Category'] plt.figure(figsize=[7,6]) sns.countplot(x=a,data=df,palette='rocket') plt.xticks(rotation=90,fontsize=9) plt.ylabel('Size\_MB')

plt.hist(x=df['Size\_MB'],bins=30,density=True)

from fitter import Fitter, get\_common\_distributions, get\_distributions f= Fitter(df["Size\_MB"],

distributions=["lognorm","norm"])

f.fit() f.summary()

df.describe()

# Hypothesis testing import scipy.stats as stats

Paid\_mean=df[df['Type']=='Paid']['Rating'] Free\_mean=df[df['Type']=='Free']['Rating'] stats.ttest\_ind(Paid\_mean,Free\_mean,equal\_var=False)

ttest,pval =stats.ttest\_ind(Paid\_mean,Free\_mean,equal\_var=False) print("p-value",pval)

if pval <0.05:

print("We reject null hypothesis") else:

print("We accept null hypothesis")

Family=df[df['Category']=='FAMILY']['Size\_MB']

Category\_mean=df['Size\_MB'].mean()

tset, pval = stats.ttest\_1samp(a=Family, popmean=Category\_mean) print('p-value',pval)

if pval < 0.05: # alpha value is 0.05 or 5% print("We reject null hypothesis")

else:

print("We accept null hypothesis")

X=df[['Rating','Reviews','Size\_MB','Price']] y=df['Installs']

from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression from sklearn.linear\_model import Ridge

from sklearn.linear\_model import Lars from sklearn.linear\_model import Lasso

from sklearn.linear\_model import LassoLars

from sklearn.ensemble import RandomForestRegressor from sklearn.metrics import mean\_squared\_error

from sklearn.metrics import mean\_absolute\_error from sklearn.metrics import explained\_variance\_score from sklearn.metrics import r2\_score

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=101) lr = LinearRegression(n\_jobs=3)

lr.fit(X\_train,y\_train) #ridge

r = Ridge(alpha=0.3) r.fit(X\_train,y\_train) #lars

lrs = Lars() lrs.fit(X\_train, y\_train) #lasso

ls = Lasso(alpha=0.3) ls.fit(X\_train, y\_train) #lasso lars

lslrs = LassoLars(alpha=0.3) lslrs.fit(X\_train, y\_train)

# Random Forest Regressor rfg=RandomForestRegressor()

rfg.fit(X\_train,y\_train)

models = {"Linear Regression":lr,"Ridge":r,"Lars":lrs,"Lasso":ls,"LassoLars":lslrs,"Random Forest Regressor":rfg}

train\_mse = {} val\_mse = {}

for i,model in enumerate(models): print("Model: {}\n".format(model))

pred = models[model].predict(X\_train) print("Training Scores:")

train\_mse[model] = mean\_squared\_error(y\_train,pred) print("Mean Sqaured Error: {}".format(train\_mse[model]))

print("Mean Absolute Error: {}".format(mean\_absolute\_error(y\_train,pred))) print("Explained Variance Score: {}".format(explained\_variance\_score(y\_train,pred))) print("R2 Score: {}\n".format(r2\_score(y\_train,pred)))

pred = models[model].predict(X\_test) print("Validation Scores:")

val\_mse[model] = mean\_squared\_error(y\_test,pred) print("Mean Sqaured Error: {}".format(val\_mse[model]))

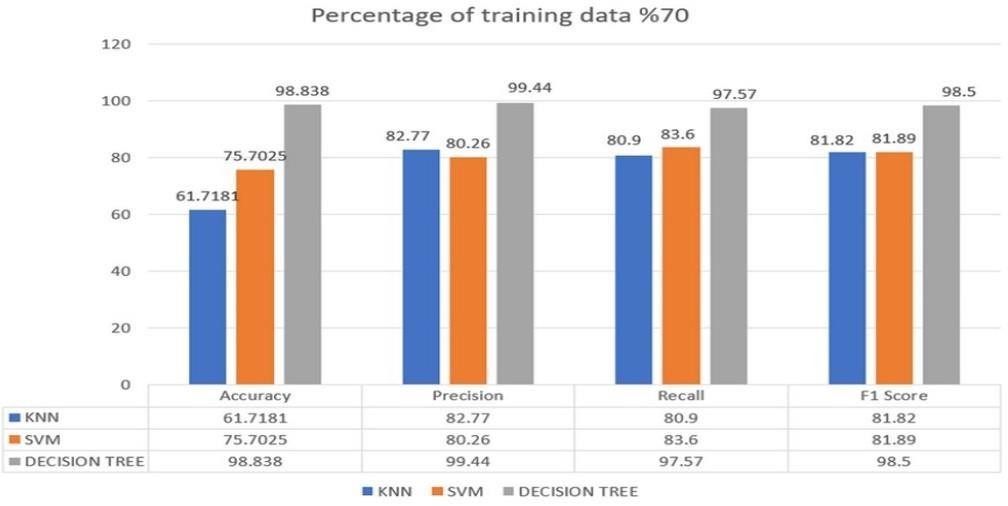
print("Mean Absolute Error: {}".format(mean\_absolute\_error(y\_test,pred))) print("Explained Variance Score: {}".format(explained\_variance\_score(y\_test,pred))) print("R2 Score: {}\n\n".format(r2\_score(y\_test,pred)))

model\_train\_error = pd.Series(data=list(train\_mse.values()),index=list(train\_mse.keys())) model\_val\_error = pd.Series(data=list(val\_mse.values()),index=list(val\_mse.keys()))

fig= plt.figure(figsize=(10,6)) model\_train\_error.sort\_values().plot.barh() plt.show()

fig= plt.figure(figsize=(10,6)) model\_val\_error.sort\_values().plot.barh() plt.show()

* + - 1. **Results and Discussions 6.1-Table and Graphs of results**

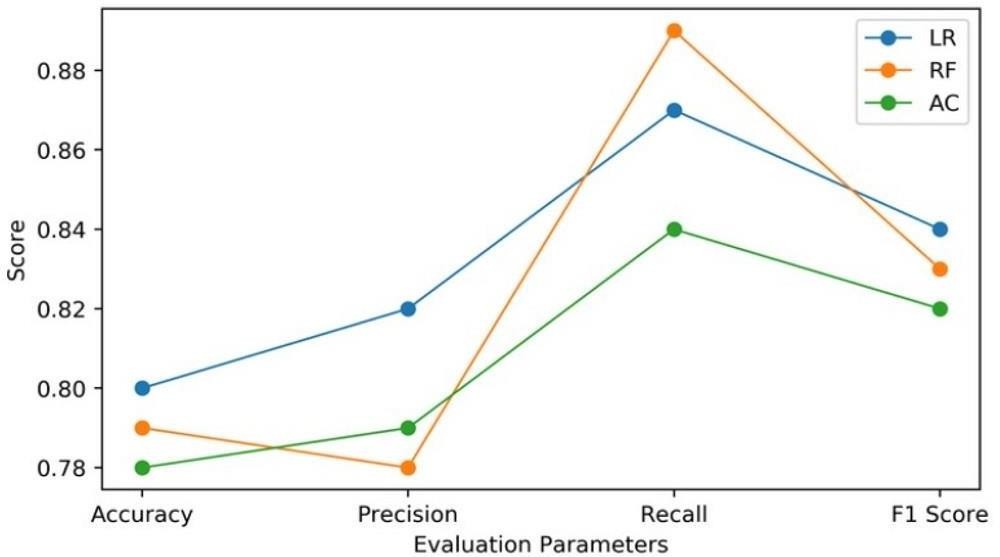


**Fig.6.1.1 Percentage of training data %70**



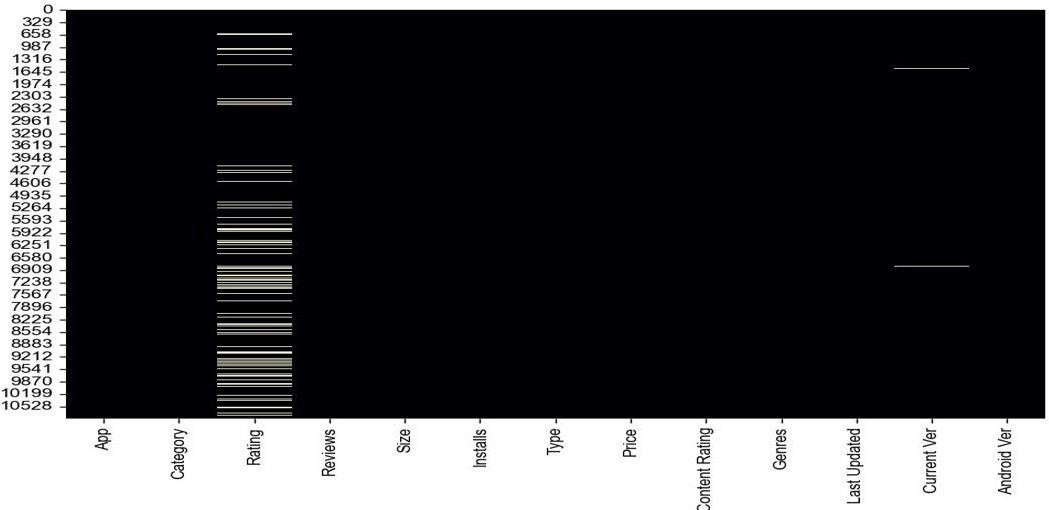
**Fig.6.1.2 Percentage of training data %80**

**6.2-Results comparison and Graphs**

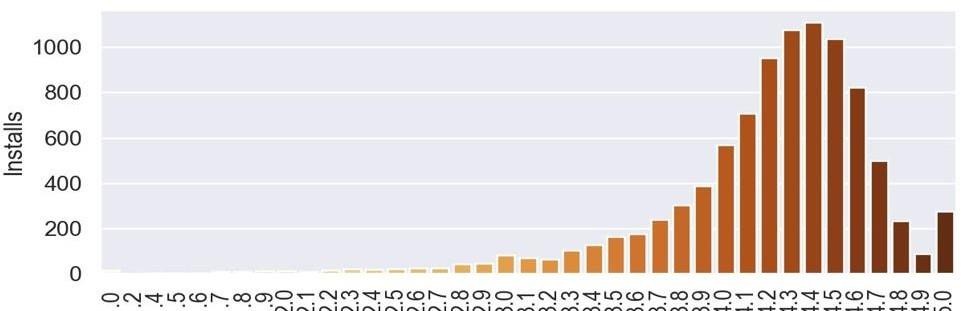


**Fig.6.1.3 Evaluation parameters**

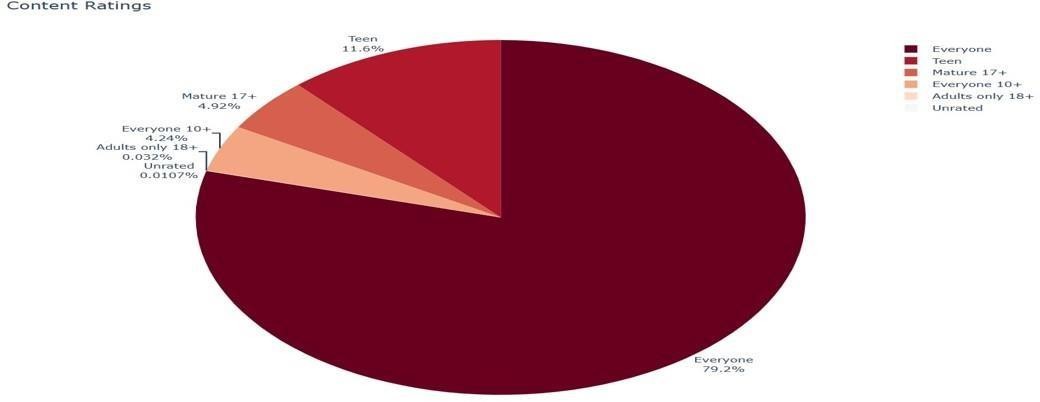
**6.3 Final Output**



**Fig.6.3.1 Rating scheme**



**Fig.6.3.2 Number of Installs**



**Fig.6.3.3 Content Rating**

# TESTING AND VALIDATION

## TYPES OF TESTING

A technique for programming testing coordinates the outline of programming experimentsinto an all around arranged arrangement of steps that outcome in fruitful improvement ofthe product. The procedure gives a guide that portrays the means to betaken, when, andhow much exertion, time, and assets will be required. The procedure joins test arranging, experiment configuration, test execution, and test outcome gathering and assessment. The procedure gives direction to the specialist and an arrangement of points of reference for thechief. Due to time weights, advance must bequantifiable and issues must surface as ahead of schedule as would be prudent.

#### 7.1.1.UNIT TESTING:

Unit Testing is done on singular modules as they are finished and turned out to beexecutable. It is restricted just to the planner's prerequisites. It centers testing around the capacity or programming module. It Concentrates on the interior preparing rationale and

information structures. It is rearranged when a module is composed with high union reduces the quantity of experiments and allows mistakes to be all the more effectively anticipated and revealed.

#### BLACK BOX TESTING:

It is otherwise called Functional testing. A product testing strategy whereby the inwardworkings of the thing being tried are not known by the analyzer. For instance, in adiscovery test on a product outline the analyzer just knows the information sources andwhat the normal results ought to be and not how the program touches base at those yields. The analyzer does not ever inspect the programming code and does not require any further

learning of the program other than its determinations.

#### WHITE BOX TESTING :

It is otherwise called Glass box, Structural, Clear box and Open box testing . A product testing procedure whereby express learning of the inner workings of the thing being triedare utilized to choose the test information. Not at all like discovery testing, white box testing utilizes particular learning of programming code to inspect yields.

#### INTEGRATION TESTING:

Coordination testing guarantees that product and subsystems cooperate an entirety. It tests the interface of the considerable number of modules to ensure that the modules carry on legitimately when coordinated together. It is characterized as a deliberate procedure for developing the product engineering. In the meantime reconciliation is happening, lead teststo reveal blunders related with interfaces. Its Objective is to take unit tried modules and assemble a program structure in view of the recommended outline.

#### SYSTEM TESTING:

Framework testing includes in-house testing of the whole framework before conveyance to the client. Its point is to fulfill the client the framework

meets all necessities of the customer's determinations. This testing assesses working of framework from client perspective, with the assistance of particular report. It doesn't derequire any inward learning of framework like plan or structure of code.

#### ACCEPTANCE TESTING:

Acknowledgment testing, a testing method performed to decide if the product framework has met the prerequisite particulars. The principle motivation behind this test is to assess the framework's consistence with the business necessities and check inthe event that it is has met the required criteria for conveyance to end clients. It is a pre-conveyance testing in which whole framework is tried at customer's site on genuine information to discover

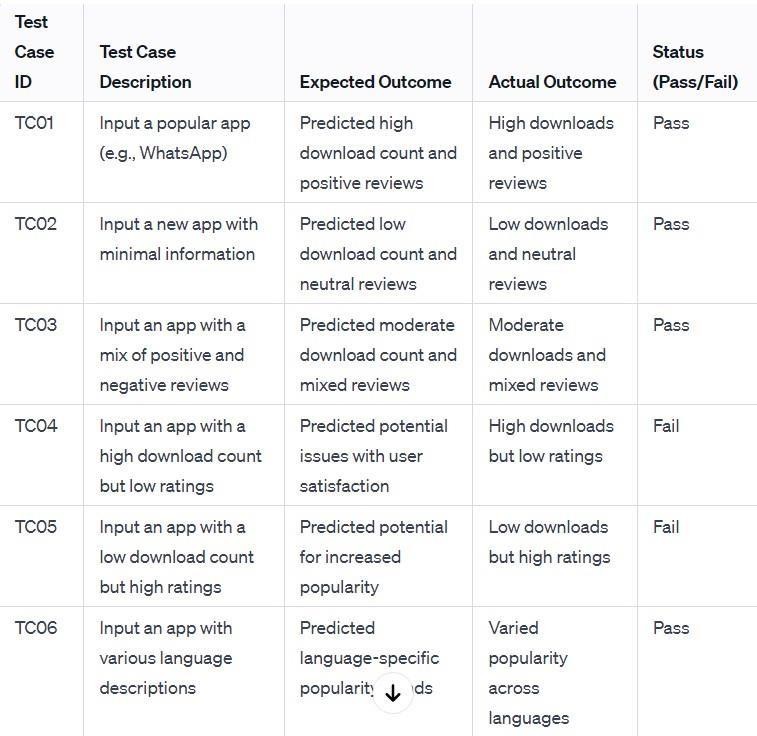
blunders. The acknowledgment test bodies of evidence are executed against

the test information or utilizing an acknowledgment test content and afterward the outcomes are contrasted and the normal ones.

## TESTCASES

Experiments include an arrangement of steps, conditions and sources of info that can be utilized while performing testing undertakings. The principle expectation of this action is to guarantee whether a product passes or bombs as far as usefulness and different perspectives. The way toward creating experiments can likewise help discoverissues in the prerequisites or plan of an application. Experiment goes about as the beginning stage for the test execution,and in the wake of applying an arrangement of information esteems, the application has a conclusive result and leaves the framework at some end point or otherwise called execution post condition.

* + 1. **Test Cases Table**



* 1. **Summary for Testing**

Testing in Google Play apps prediction is a critical process aimed at evaluating the effectiveness and accuracy of prediction models before deploying them into production. The summary of testing in Google Play apps prediction encompasses several key points.



Validation of Prediction Models



Evaluation of Performance Metrics



Feature Importance Analysis

Cross-Validation Techniques





Model Tuning and Optimization



Validation on New Data



Error Analysis and Interpretability

In summary, testing in Google Play apps prediction is a comprehensive process that involves validating prediction models, evaluating performance metrics, employing cross-validation techniques, analyzing feature importance, tuning model parameters, validating on new data, and conducting error analysis. By rigorously testing prediction models, developers can ensure their reliability, accuracy, and effectiveness in forecasting app success on Google Play.

## CONCLUSION

After undergoing these algorithms and process, we concluded that our hypothesis is true. Meaning you can predict the app ratings, however significant preprocessing must be donebefore you start the classification and regression processes. The Play Store apps data hasenormous potential to drive app-making businesses to success. Actionable insights can be drawn for developers to work on and capture the Android market.Thisshows that given theSize, Type, Price, Content Rating, and Genre of an app, we can predict about 92%accuracy if an app will have more than 100,000 installs and be a hit on the Google Play Store. User reviews are limited to identifying polarity and subjectivity. However, the massive increase in review- based data implies a requirementto focus also on performing predictions. We came to the conclusion that our hypothesis is correct after running through all of these algorithms and processes. As a result, it is possible to predict app ratings, but a large amount of preprocessing is required before the classification and regression processes can be started. The data collected from Google Play Store apps has huge potential to help app development companies succeed. Developers can use the information to their advantage to work on and conquer the Android market! In order to accurately estimate whether an app will have more than 100,000 downloads and be a success on the Google Play Store, we need to know the app's Size.

# FUTURE ENHANCEMENTS

This process is challenging yet fruitful, as user reviews are qualitative while ratings are essentially quantitative. The numeric scoring of apps within the Google App store could also be biased and overrated because higher ratings given by users potentially attract several new users disproportionately. This study therefore investigated the utilization of ensemble classifiers to predict numeric ratings for Google Play store apps supported the user reviews for those apps. Several ensemble classifiers were investigated to guage their performance on the reviews scraped from the Google App store. Random forest algorithm giving 90% accuracy. Future work includes the implementation of the deep learningtechnique to predict numeric rating.

By 2023, the global AI market will reach $70.94 billion.In this way, Android app developers will provide a more enhanced personalized experience to users. Future applications may integrate different AI features such as text, image classification, voice recognition, predictive maintenance, face detection, etc.

Only polarity and subjectiveness may be obtained from user reviews. Predictions are also important because of the enormous expansion in review-based data. This is a challenging but rewarding process, as user reviews are qualitative and ratings are mainly quantitative. Additionally, Google's numerical rating system may be distorted and amplified by the fact that higher ratings supplied by consumers may bring in disproportionately more new users. So this study investigated if ensemble classifiers might be used to predict numerical ratings for Google Play store apps based on user reviews. The Google App store assessments were used to test many ensemble classifiers. To predict numerical ratings in the future, deep learning technology will be applied.

# REFERENCES

1. **https://ieeexplore.ieee.org/document/8029770**
   1. **https://ieeexplore.ieee.org/document/9068071**
2. **https://ieeexplore.ieee.org/document/9051529**
3. **https://ieeexplore.ieee.org/document/9423823**
4. **https://ieeexplore.ieee.org/document/9043686**
5. Statista, Numerous accessible applications within the Google Play store from December 2009 to March 2019, https:/[/www.statista.com/statisti](http://www.statista.com/statistics/266210/numberof-available-applications-in-the-google-play-store/)c[s/266210/numberof-available-applications-in-the-google-play-store/,](http://www.statista.com/statistics/266210/numberof-available-applications-in-the-google-play-store/) Online: accessed twentytwo May 2019.
6. Statista, various mobile app downloads worldwide in 2017, 2018, and 2020 (in billions), https:/[/www.statista.com/statisti](http://www.statista.com/statistics/)c[s/](http://www.statista.com/statistics/) 271644/worldwide-free-and-paid-mobile-app-store downloads/, Online: accessed twenty-two May 2019.
7. Online shopping, pew internet, and American life project, Washington, DC, 2018, <http://www.pewinternet.org/Repor> ts/2008/online shopping/01-Summary-of-Findings.aspx Online: accessed eight August. 2014.
8. D. Pagano and W. Maalej, User feedback within the AppStore: an empirical study, in Proc. IEEE Int. Requirements Eng. Conf. (Rio de Janeiro, Brazil), July 2013, pp. 125–134.
9. T. Chumwatana, Using sentiment analysis technique for analyzing Thai client satisfaction from social media, 2015.
10. T. Thiviya et al ., Mobile apps feature extraction supported user reviews using machine learning, 2019.
    * + 1. **Appendix**

