A project report

on

Predicting App Success Using Machine Learning

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

Batch No: B-7

R.SAI THEJA	**184G1A0581**
T.SHAHUL	**184G1A0585**
S.SUPRIYA	**184G1A05A2**
S.YASWANTH REDDY	**194G5A0513**

Under the Guidance of

Mrs.V. Kamakshamma, M.Tech., (Ph. D)

Assistant Professor

SR IT
Empowering Knowledge

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUA & Approved by AICTE)

(Accredited by NAAC with ‘A’ Grade & Accredited by NBA (EEE, ECE & CSE))

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu – 515701.

2021 - 2022

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUA & Approved by AICTE)

(Accredited by NAAC with 'A' Grade & Accredited by NBA (EEE, ECE, CSE))

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu – 515701

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Certificate

This is to certify that the Technical Seminar report entitled **PREDICTING APP SUCCESS USING MACHINE LEARNING** is the bonafide work carried out by **R.Sai Theja** bearing Roll Number **184G1A0581, T.Shahul** bearing Roll Number **184G1A0585, S.Supriya** bearing Roll Number **184G1A05A2, S.Yashwanth Reddy** bearing Roll Number **194G5A0513** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** duringthe academic year 2021-2022.

Signature of the Guide	**Head of the Department**
Mrs.V. Kamakshamma M.Tech.,(Ph.D)	Mr.P. Veera Prakash M.Tech ,(Ph.D)
Assistant Professor	Assistant Professor & HOD

Date:

Place: Rotarypuram **EXTERNAL EXAMINAR**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that I would like to express my indebted gratitude to my Guide **Mrs.V. Kamakshamma M.Tech(Ph.D), Assistant Professor**, **Computer Science & Engineering**, who has guided us a lot and encouraged us in every step of the project work. We thank her for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deepfelt gratitude to **Mr. Venkatesh, M.Tech(Ph.D), Assistant Profressor, Computer Science and Engineering,** project coordinator ,for his valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We are very much thankful to P**. Veera Prakash, M.Tech.(Ph.D), Assistant Professor & HOD, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey my special thanks to Dr**. G. Bala Krishna Ph.D, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, I thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported me in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey my gratitude to my family who fostered all the requirements and facilities that I need.

Project Associates

DECLARATION

We R.Sai Theja bearing reg no:184G1A0581,T.Shahul bearing reg no: 184G1A0585, S.Supriya bearing reg no:184G1A05A2,S.Yashwanth Reddy reg no:194G5A0513 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that dissertation entitled "PREDICTING APP SUCCESS USING MACHINE LEARNING" embodies that report of our project work carried out by us during IV year Batchelor of Technology under the guidance of Mrs. V.Kamakshamma,M.Tech(P.hD),Department of Computer Science and Engineering, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY,ANANTAPUR and this work is submitted for the partial fulfilment of the requirements for the award of the Batchelor of Technology degree.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

R.Sai Theja	Reg no: 184G1A0581
T.Shahul	Reg no: 184G1A0585
S.Supriya	Reg no: 184G1A05A2
S.Yashwanth Reddy	Reg no: 194G5A0513

CONTENTS

Page No.

LIST OF FIGURES

LIST OF ABBREVATIONS

GAE	Google App Engine
AWS	Amazon Web Services
API	Application Program Interface
CPN	Colored Petri Nets
SVM	Support Vector Machine
UML	Unified Modified Language
ER	Entity Relationship
DFD	Data Flow Diagram
OOP	Object Oriented Programming
XML	Extended Markup Language
DOM	Document Object Model
HTTP	Hypertext Markup Language

ABSTARCT

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 SVM, Decision Tree, XGBoost and Random Forest algorithms and, with that, the features that determine the best correlation for the rating of the applications were investigated, besides to compute and evaluate regression metrics using a Google Play Storedatabase. The work is structured as follows: in section II, the theoretical framework will be presented, where the main themes relevant to this work will be explained. Section III will discuss the materials and methods, where the step by step will be described according to the CRISP-DM to obtain knowledge of the database. In section IV, the results obtained with the execution of the algorithms will be structured and in section V, it will be the conclusion, which will be scored what was possible to understand with this research.

CHAPTER 1

INTRODUCTION

Cloud computing environments, especially the PaaS (Platform As A Service) environments, are one of the most cost-effective, efficient, and productive platforms to implement enterprise information systems including database and transaction applications, since they provide us with not only virtualized scalable hardware but also operating systems and middleware like database management systems.

By a PaaS environment, we can realize more simplified system operation and flexible application development simultaneously. In spite of these advantages, there are several problems to be overcome, when we try to run mission critical information systems in the PaaS environments. Among them, the most serious ones are “data integrity” and “performance estimation”. The former problem is caused by the relaxed data integrity principle “BASE”, while the latter is mainly caused by the virtualized and concealed hardware, which makes it difficult to obtain the performance related information accurately.

In order to implement stable large-scale information systems, performance prediction is one of the key success factors, since it requires too much cost and time to tune up the systems after the cutover. There have been developed many kinds of methodologies and techniques for system performance prediction. They are classified broadly into two approaches, that is, an analytic approach and simulation-based approach. While the former can predict the performance precisely for small-sized and simply- structured systems, the latter is suitable for ballpark prediction of large-scale and complex structured systems. In both approaches, it is important to build appropriate models to predict the performance.

Traditionally, these models have been built based on the configuration of the system to be evaluated, which is composed of hardware components (like CPUs, HDDs,and memory units), software components (like database management systems, transaction control systems, and operating systems), and network components (like communication lines, routers, and switches). However, these components are virtualized in the cloud environments, and we cannot obtain the required performance indices and parameters of them easily.

Therefore, we need another approach to predicting the performance of the systems in the cloud. Since the only information we can obtain regarding performance predictions the structure of each application to run in the cloud, we focus on this information to build the prediction models. From performance viewpoint, the application structure is divided into two parts, that is, the “cloud-dependent” part and “cloud-independent” part. While the former utilizes the various services provided by the cloud, the latter performs standalone functions which are executable without the cloud.

Generally, most of the processing time of each application is spent by the former part, and it is difficult to estimate this part. On the other hand, the latter part slightly affects the processing time, and can easily be estimated. Therefore, we focus on the former part to build the prediction models. This part is composed of the interactions between an application and the cloud services, which are provided in the form of API (Application Programming Interfaces).

Therefore, we firstly define each related application as a series of these APIs. However, these APIs are different between cloud services like Google App Engine (GAE), Amazon Web Services (AWS), or IBM Bluemix, and consequently we have to select the cloud service to be modeled. This paper proposes a modeling and simulation-based cloud performance prediction framework, focusing the application structure and API characteristics. As a modeling and simulation tool, we use the UPPAAL model checker. The paper assumes GAE transaction processing as a cloud platform; however, the approach is applicable to other platforms by replacing a set of APIs provided by them.

All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output Expect.

CHAPTER 2

LITERATURE SURVEY

[1] M. R. Islam, “Numeric rating of apps on google play store by sentiment analysis on user reviews,” in 2014 International Conference on Electrical Engineering and Information & Communication Technology. IEEE, Apr 2014.[Online]. Available: https://doi.org/10.1109/iceeict. 2014.6919058

The sudden eruption of sentiment analysis and opinion mining has opened new possibilities to improve our information gathering interests. We are always keen to know what others say about the devices or applications we are going to use. It’s observed that sometimes the numeric rating has vast difference than the reviews given bythe users. To remove this ambiguity a unified rating system has been proposed here. The starred rating and a generated numeric polarity of the reviews are combined to generate the final rating. The proposition is based on sentiment analysis and an optimized probabilistic approach described by a group of researchers. The approach is proved for its efficiency in a diverse corpus of writings where the targets are of differentcategories.

[2] S. Nishida and Y. Shinkawa. Data Integrity in Cloud Transactions, Proceedings of the 4th International Conference on Cloud Computing and Services Science, pages 457–462, Barcelona, Spain, April 3–5, 2014.

BASE is a new principle for transaction processing in cloud computing environments, which stands for Basically Available, Soft state, and Eventually consistent. Unlike the traditional principle of ACID for transactions, BASE pays greater attention to scalability and availability of systems, and less to the data integrity. Therefore, when migrating mission critical transaction systems into cloud environments, we have to carefully evaluate whether the provided data integrity levels acceptable. This paper presents a formal, simulation based, and model driven approach to evaluating data integrity in BASE transaction systems. We define the data integrity in cloud environments using a new concept for system states, referred to as superposed states. As a formalization tool, VDM++ and Colored Petri Nets (CPN) are used to express and simulate the BASE transaction systems accurately.

[3] S. Nishida and Y. Shinkawa. CPN Based GAE Performance Prediction Framework, Proceedings of the 9th International Conference on Software Engineering and Applications, pages 401–406, Vienna, Austria, August 29–31, 2014.

Google App Engine (GAE) is one of the most popular PAAS type cloud platform for database transaction systems. When we plan to run those systems on GAE, performance prediction is one of the obstacles, since only a little performance information on GAE is available. In addition, the structure of GAE is not opened to general public. This paper proposes a Colored Petri Net (CPN) based simulation framework, based on the performance parameters obtained through the measurement by user written programs. The framework is build focusing on the application structure, which consists of a series of GAE APIs, and GAE works as a mechanism to produce the probabilistic process delay. The framework has high modularity to plug-in any kinds of applications easily.

[4] G. Kesavaraj and S. Sukumaran, "A study on classification techniques in data mining," in Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on. IEEE, 2013, pp. 1–7.

Data mining is a process of inferring knowledge from such huge data. Data Mining has three major components Clustering or Classification, Association Rules and Sequence Analysis. By simple definition, in classification/clustering analyze a set of data and generate a set of grouping rules which can be used to classify future data. Data mining is the process is to extract information from a data set and transform it into an understandable structure. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems.

The actual data mining task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown interesting patterns. Data mining involves six common classes of tasks. Anomaly detection, Association rule learning, Clustering, Classification, Regression, Summarization.

Classification is a major technique in data mining and widely used in various fields. Classification is a data mining (machine learning) technique used to predict group membership for data instances. In this paper, we present the basic classification techniques. Several major kinds of classification method including decision tree induction, Bayesian networks, k-nearest neighbor classifier, the goal of this study is to provide a comprehensive review of different classification techniques in data mining.

The actual data mining task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown interesting patterns. Data mining involves six common classes of tasks. Anomaly detection, Association rule learning, Clustering, Classification, Regression, Summarization. Classification is a major technique in data mining and widely used in various fields. Classification is a data mining (machine learning) technique used to predict group membership for data instances. In this paper, we present the basic classification techniques. Several major kinds of classification method including decision tree induction, Bayesian networks, k-nearest neighbor classifier, the goal of this study is to provide a comprehensive review of different classification techniques in data mining. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. Data mining involves six common classes of tasks. Anomaly detection, Association rule learning, Clustering, Classification, Regression, Summarization. Classification is a major technique in data mining and widely used in various fields.

[5] G. Kesavaraj and S. Sukumaran, "A study on classification techniques in data mining," in Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on. IEEE, 2013, pp. 1–7.

Data mining is a process of inferring knowledge from such huge data. Data Mining has three major components Clustering or Classification, Association Rules and Sequence Analysis. By simple definition, in classification/clustering analyze a set of data and generate a set of grouping rules which can be used to classify future data. Data mining is the process is to extract information from a data set and transform it into an understandable structure. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems.

The actual data mining task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown interesting patterns. Data mining involves six common classes of tasks. Anomaly detection, Association rule learning, Clustering, Classification, Regression, Summarization. Classification is a major technique in data mining and widely used in various fields. Classification is a data mining (machine learning) technique used to predict group membership for data instances. In this paper, we present the basic classification techniques. Several major kinds of classification method including decision tree induction, Bayesian networks, k-nearest neighbor classifier, the goal of this study is to provide a comprehensive review of different classification techniques in data mining. In this paper, we focus on an interactive recommendation system which can help users to correct their own ratings. Thereby, we propose a method to determine whether the ratings from users are consistent to their own preferences (represented as a set of dominant attribute values) or not and eventually to correct these ratings to improve recommendation.

[6] X. Amatriain, J. M. Pujol, N. Tintarev, and N. Oliver, "Rate it again: increasingrecommendation accuracy by user re-rating," in Proceedings of the third ACMconference on Recommender systems. ACM, 2009, pp. 173–180.

In most of the recommendation systems, user rating is an important user activity that reflects their opinions. Once the users return their ratings about items the systems have suggested, the user ratings can be used to adjust the recommendation process. However, while rating the items users can make some mistakes (e.g., naturalnoises). As the recommendation systems receive more incorrect ratings, the performance of such systems may decrease.

The author focus on an interactive recommendation system which can help users to correct their own ratings. Thereby, we propose a method to determine whether threating's from users are consistent to their own preferences (represented as a set of dominant attribute values) or not and eventually to correct these ratings to improve recommendation. The proposed interactive recommendation system has been particularly applied to two user rating datasets (e.g., MovieLens and Netflix) and it has shown better recommendation performance (i.e., lower error ratings).

2.2 Existing Method

The increasing growth of machine learning, computer techniques divided into traditional methods and machine learning methods. This section describes the related works of google play store app rating and how machine learning methods are better than traditional methods. The existing method in this project have a certain flow and also SVM regression is used for model development. But it requires large memory and result is not accurate.

Disadvantages

Accuracy low

Operating cost is high

Difficult to handle

2.3 Proposed Method

In the proposed system, ten features have been evaluated to make this comparison more unique. Our introduced algorithms were conducted based on the obtained outcomes were compared to other works to show the percentage of improvement, while decrease in performance also noted in one occasion Random Forest Regression. The highest increment was noticed for previous works which was about percentage improvement were calculated for google play store app ratings is used to determine whether an google play store app ratings are not to know this we used the machine learning based methods such as Decision Tree, Support Vector Machine and XG Boost Regression techniques to figure out.

Advantages

Accuracy is maximum

Prediction is accurate

Block Diagram

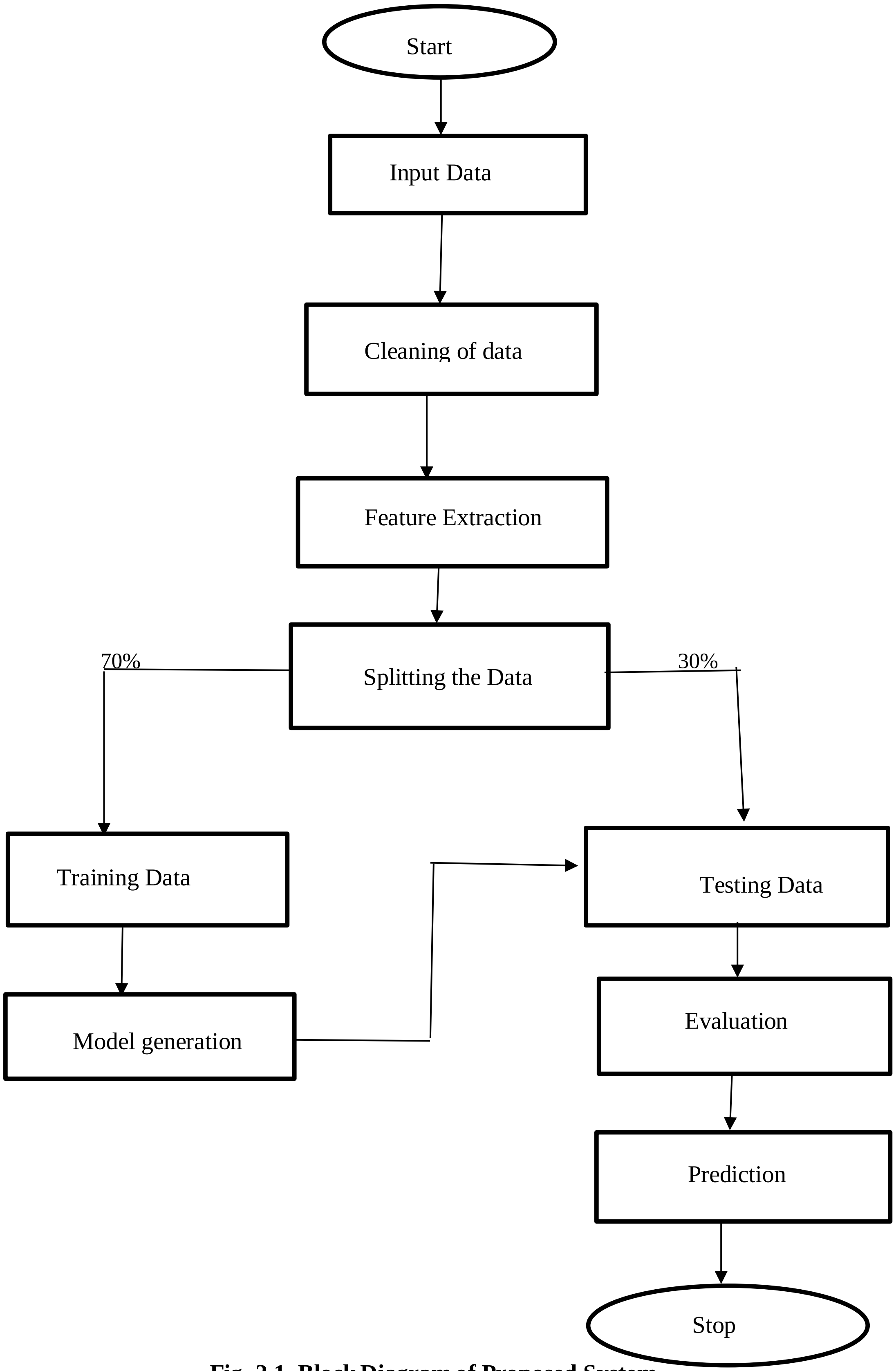


Fig .2.1. Block Diagram of Proposed System

CHAPTER 3
SYSTEM REQUIREMENTS SPECIFICATIONS

3.1 Functional and Non-Functional requirements

Requirement’s analysis is very critical process that enables the success of a systemor software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated nto the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1) Authentication of user whenever he/she logs into the system
2) System shutdown in case of a cyber-attack
3) A verification email is sent to user whenever he/she register for the first time on some software system.

Non-Functional requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non- behavioral requirements. They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

Examples of non-functional requirements:

Emails should be sent with a latency of no greater than 12 hours from such an activity .The processing of each request should be done within 10 seconds the site should load in 3 seconds whenever of simultaneous users are > 10000.

3.1.1 H/W Configuration

- Processor - I3/Intel Processor
- Hard Disk -160 GB
- RAM - 8 GB

3.1.2 S/W Configuration

• Operating System	: Windows 7/8/10 .
• Server side Script	: HTML, CSS & JS.
• IDE	: Pycharm.
• Libraries Used	: Numpy, IO, OS, Flask, keras.
• Technology	: Python 3.6+.

3.2 Modules

3.2.1 User Upload

Upload dataset which is downloaded from the Kaggle

View Data

View data after pre-processing (cleaned dataset)

Input

User will give the input values of app, category, rating, reviews, size, installs, type, price, content rating, genres, last updated and current version these parameters and user can select the algorithm.

Result History

After giving the inputs, model will predict the result which it was set according to performance, it will predict that the google play store areNot.

3.2.2 Take Dataset

The dataset for the cleaned is collected from the Kaggle website (kaggle.com). The size of overall dataset is 7.63kb. The Data set columns app, category, rating, reviews, size, installs, type, price, content rating, genres, last updated, current version and android version.

3.2.3 Pre-processing

- In pre-processing first of all we will check whether there is any Nan values.
- If any Nan values is present, we will fill the Nan values with different fillna techniques like bfill, ffill, mode, and mean.
- Here we used the ffill (front fill) technique on our project.

3.2.4 Training The Data

Irrespective of the algorithm we select the training is the same for every algorithm**.** Given a dataset we split the data into two parts training and testing, the reason behind doing this is to test our model/algorithm performance just like the exams for a student the testing is also exam for the model.

We can split data into anything we want but it is just good practice to split the data such that the training has more data than the testing data, we generally split the data into 70% training and 30% testing. And for training and testing there are two variables X and Y in each of them, the X is the features that we use to predict the Y target and same for the testing also. Then we call the fit () method on any given algorithm which takes two parameters i.e., X and Y for calculating the math and after that when we call the predict () giving our testing X as parameter and checking it with the accuracy score giving the testing Y and predicted X as the two parameters will get our accuracy score and same steps for the classification report and the confusion matrix, these are just checking for how good our model performed on a given dataset.

XGB Regression

In XGB Regression we are fitting the X_train and Y_train in our model. Based on this we are testing the data which is fitted based on that we get the accuracy.

Support Vector Machine

In Support Vector Regression we are fitting the X_train and Y_train in our model. Based on this we are testing the data which is fitted based on that we get theaccuracy.

Random Forest Regression

In Random Forest Regression we are fitting the X_train and Y_train in our model. Based on this we are testing the data which is fitted based on that we get theaccuracy.

Decision Tree Regression

In Decision Tree Regression we are fitting the X_train and Y_train in our model. Based on this we are testing the data which is fitted based on that we get theaccuracy. By using this models we will find the accuracy_score.

CHAPTER 4
ALGORITHMS

4.1 Decision Tree

Decision tree learning is a supervised machine learning technique for inducing a decision tree from training data. A decision tree (also referred to as a classification tree or a reduction tree) is a predictive model which is a mapping from observations about an item to conclusions about its target value. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data). In Decision Trees, for predicting a class label for a record we start from the root of the tree.

4.2 Random Forest

Random forest is a supervised learning algorithm which is used for both classifications well as regression Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. In data science speak, the reason that the random forest model works so well is: A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. The low correlation between models is the key.

4.3 Support Vector Machine

A support vector machine (SVM) is a type of deep learning algorithm that performs supervised learning for classification or regression of data groups. In AI and machine learning, supervised learning systems provide both input and desired output data, which are labeled for classification. SVM's are very good when we have no idea on the data. Works well with even unstructured and semi structured data like text, Images and trees. The kernel trick is real strength of SVM Unlike in neural networks, SVM is not solved for local optima. It scales relatively well to high dimensional data.

4.4 XGBoost

XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leaf's that contains a continuous score. XGBoost is a scalable and accurate implementation of gradient boosting machines and it has proven to push the limits of computing power for boosted trees algorithms as it was built and developed for the sole purpose of model performance and computational speed. Each tree learns from its predecessors and updates the residual errors. Hence, the tree that grows next in the sequence will learn from an updated version of the residuals. The base learners in boosting are weak learners in which the bias is high, and the predictive power is just a tad better than random guessing.

CHAPTER 5
SYSTEM DESIGN

5.1 Input Design

In an information system, input is the raw data that is processed to produce output During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc. Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties –

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles. –
- What are the inputs needed for the system?
- How end users respond to different elements of forms and screens.

5.1.1 Objectives for Input Design

The objectives of input design are

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

5.2 Output Design

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

5.2.1 Objectives of Output Design

The objectives of input design are**:**

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

5.3 UML Diagrams

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling 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, UML.The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven.

5.3.1 Goals

The Primary goals in the design of the UML are as follows:
Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models. Provide extendibility and specialization mechanisms to extend the core concepts. Be independent of particular programming languages and development process. Provide a formal basis for understanding the

modelling language. Encourage the growth of OO tools market. Support higher level development concepts such as collaborations, frameworks, patterns and components Integrate best practices.

5.4 Use Case Diagram

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.

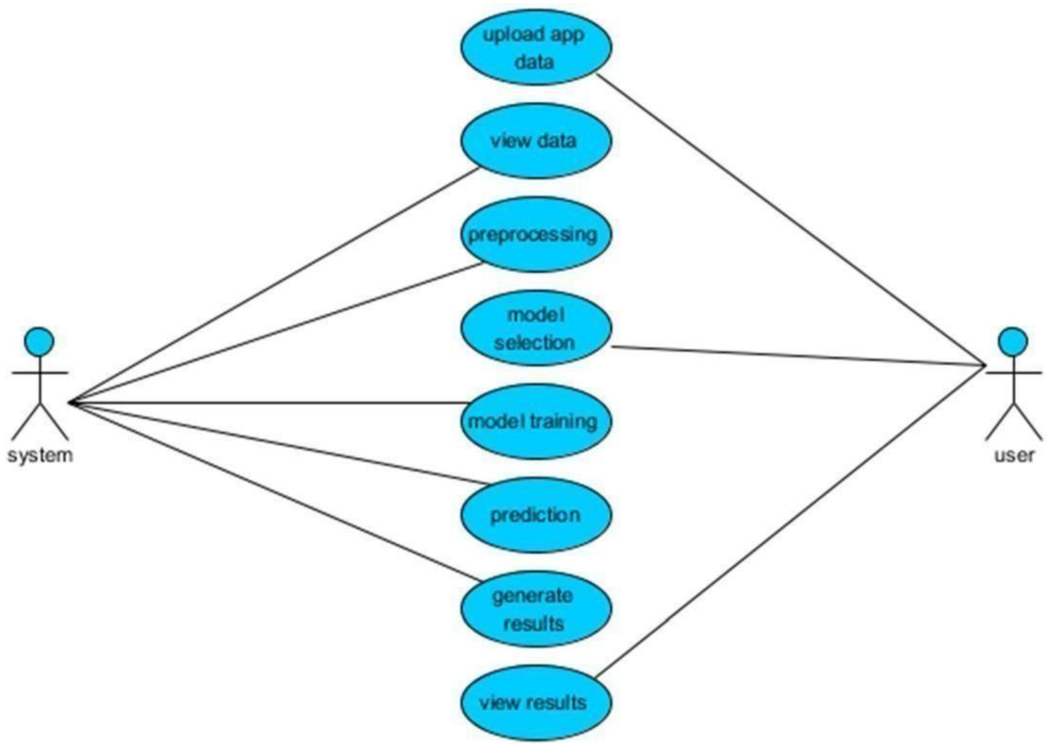


Fig.5.4. Use Case Diagram

5.5 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.

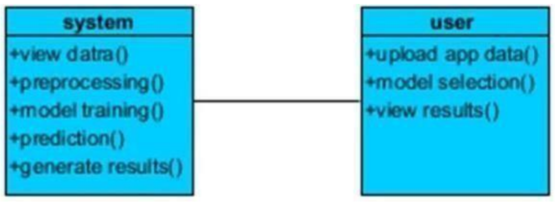


Fig.5.5. Class Diagram

5.6 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.

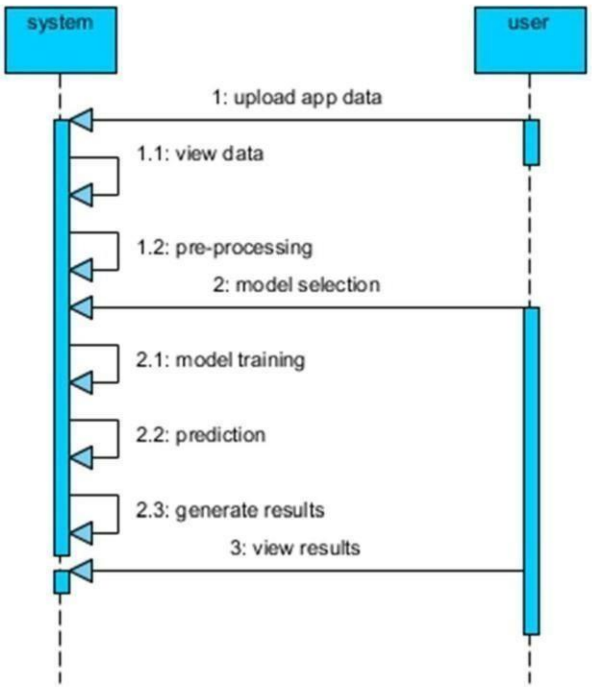


Fig.5.6. Sequence Diagram

5.7 Collaboration Diagram

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

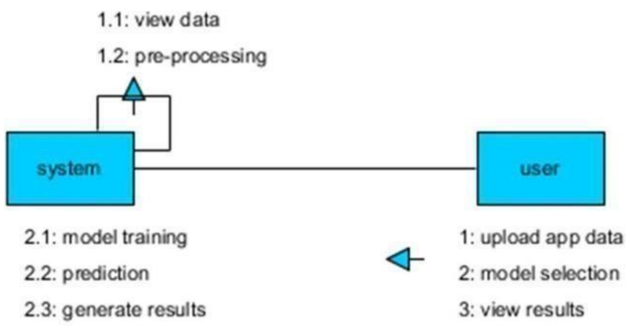


Fig.5.7. Collaboration Diagram

5.8 Deployment Diagram

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

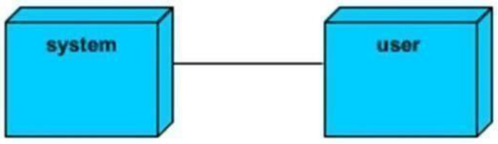


Fig.5.8. Deployment Diagram

5.9 Activity Diagram

Activity diagrams are graphical representations of work flows 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.

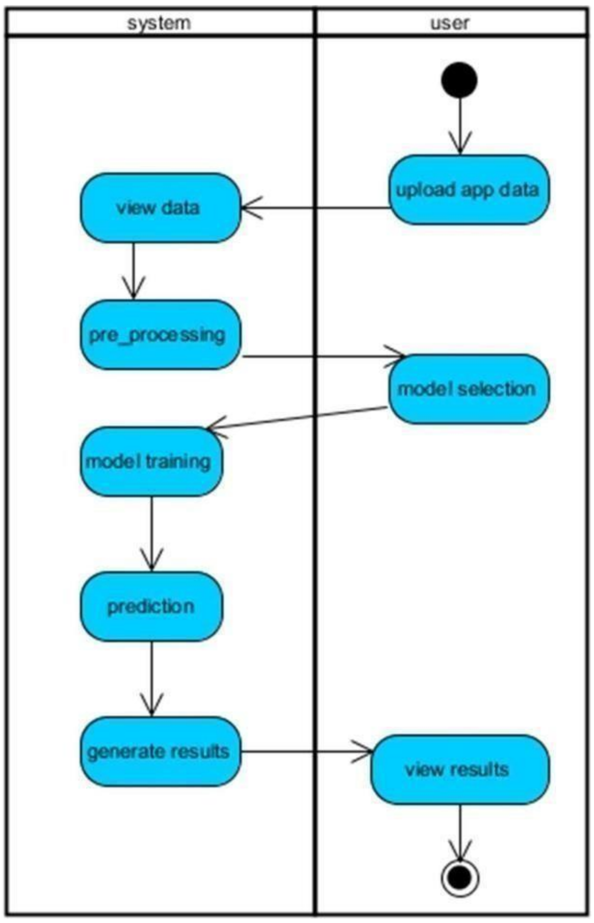


Fig.5.9. Activity Diagram

5.10 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.

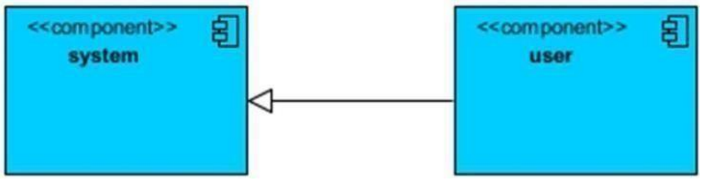


Fig.5.10. Component Diagram

5.11 ER Diagram

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

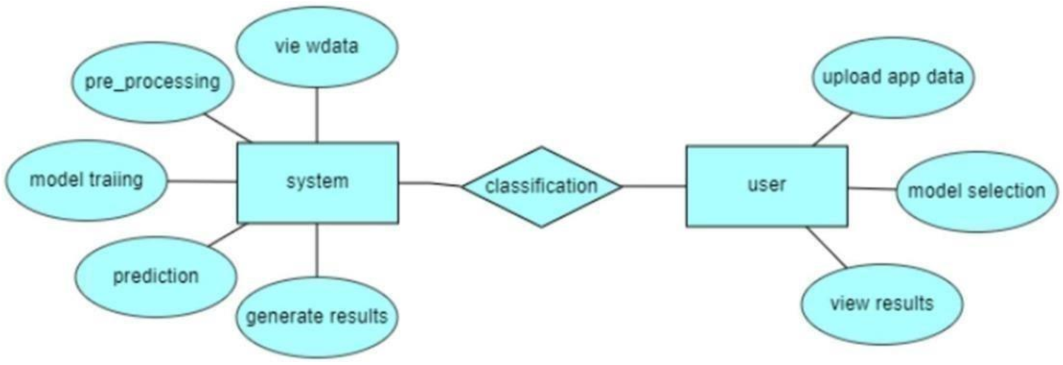
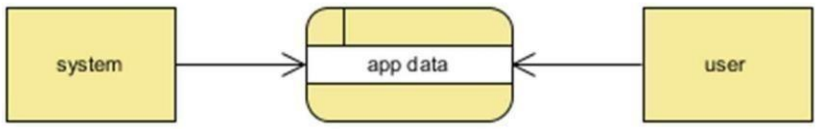


Fig.5.11.ER Diagram

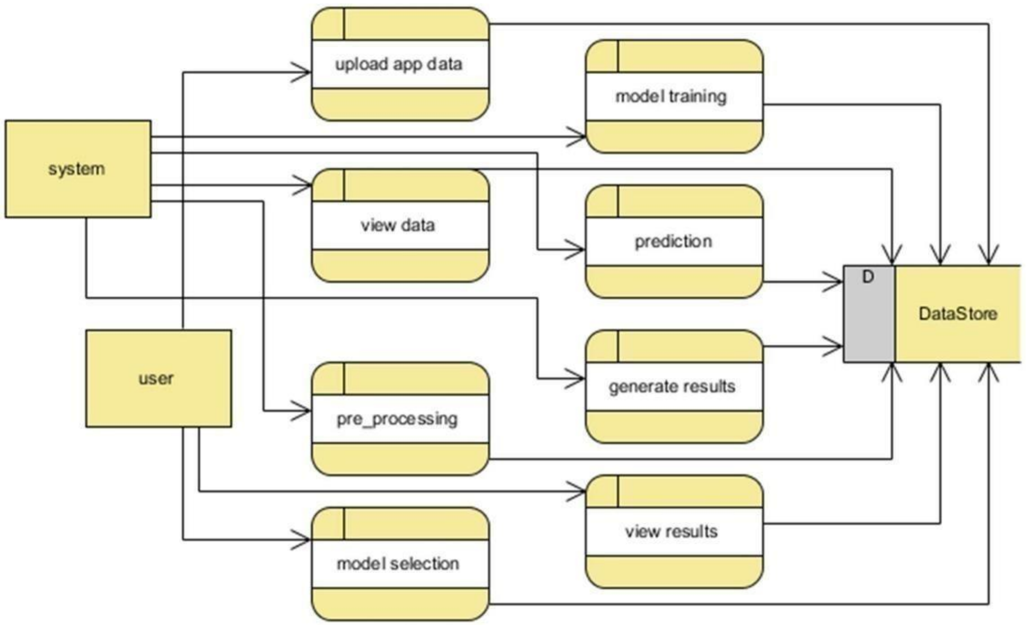
5.12 DFD Diagram

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows information enters and leaves the system, what changes the information andwhere information is stored. The purpose of a DFD is to show the scope and boundariesof a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning system.

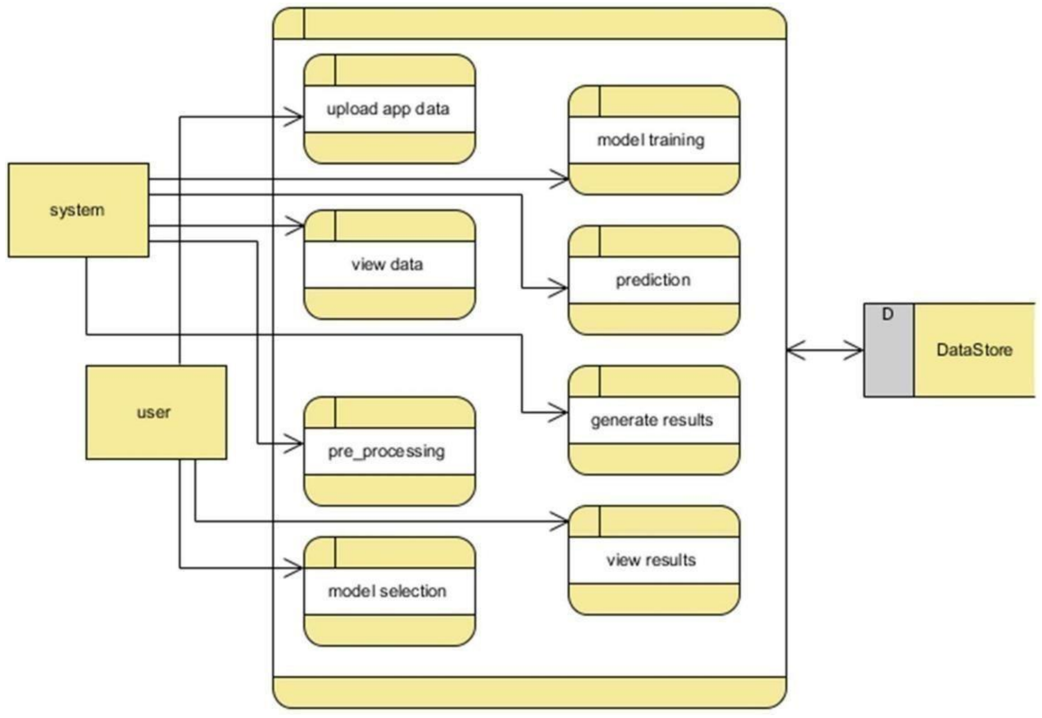
Context level diagram



Level 1 diagram



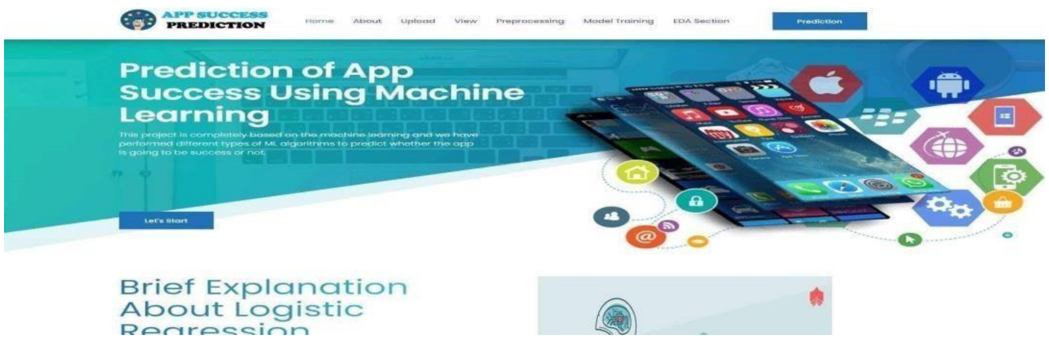
Level 2 diagram



CHAPTER 6

OUPUT SCREENSHOTS

Home page

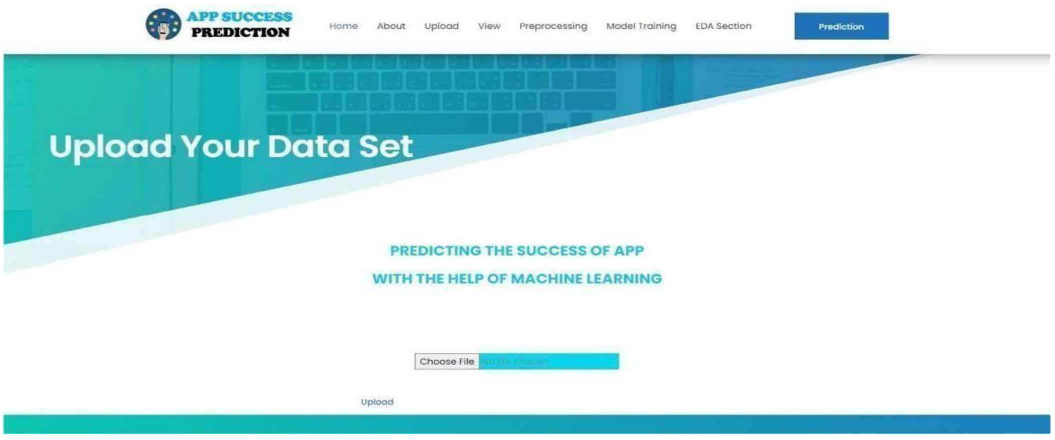


Prediction of App Success Using Machine Learning

This project is completely based on the machine learning and we have performed different types of ML algorithms to predict whether the app is going to be success or not.

Brief Explanation About Logistic Regression

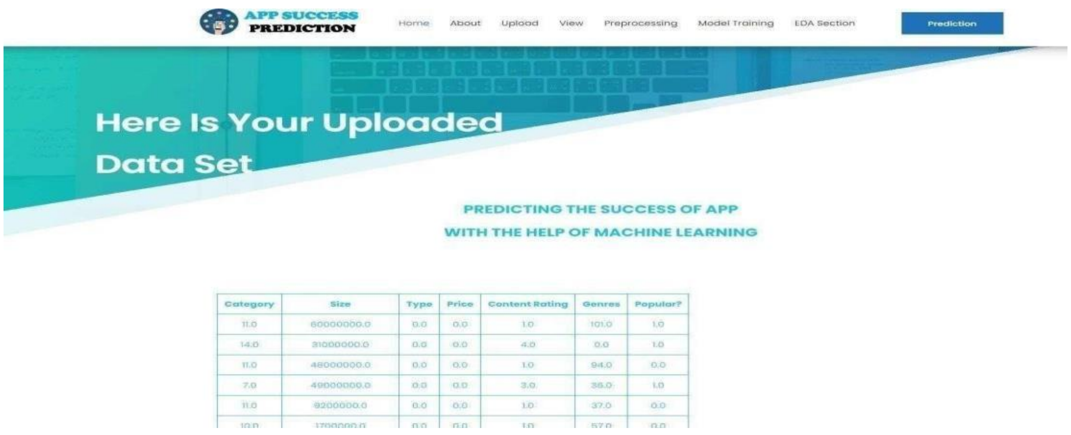
About page



Upload Your Data Set

PREDICTING THE SUCCESS OF APP
WITH THE HELP OF MACHINE LEARNING

View Data

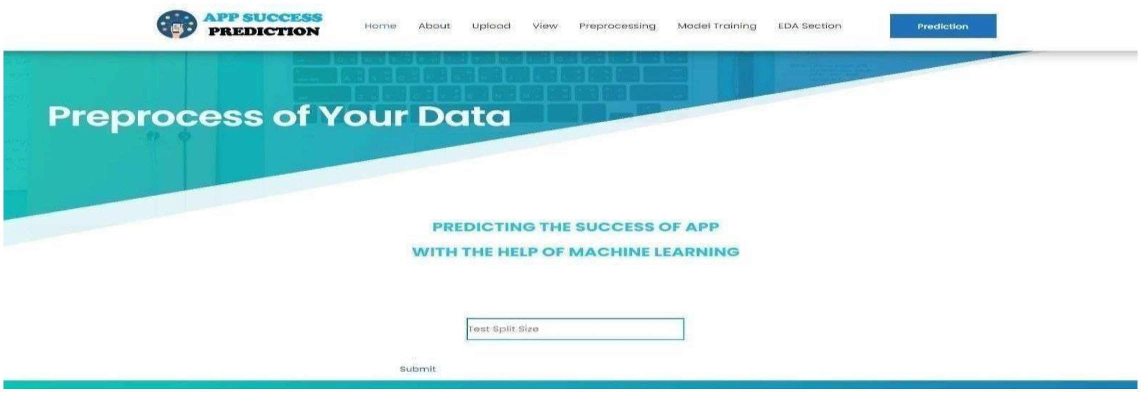


Here Is Your Uploaded Data Set

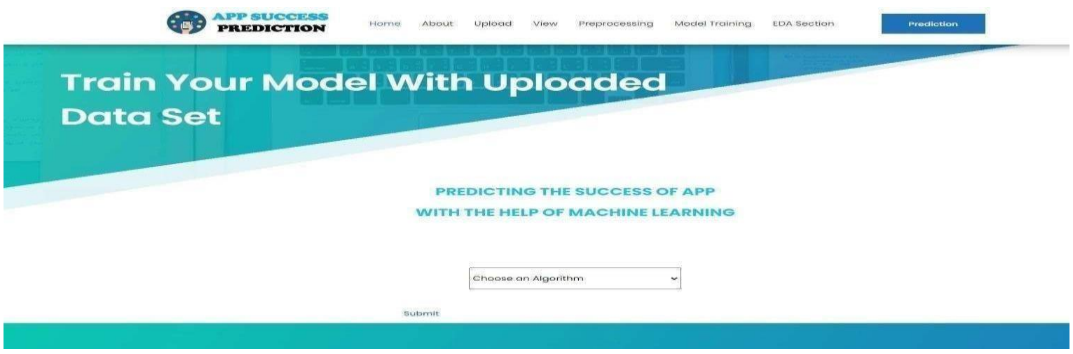
PREDICTING THE SUCCESS OF APP
WITH THE HELP OF MACHINE LEARNING

Category	Size	Type	Price	Content Rating	Genres	Popular?
11.0	60000000.0	0.0	0.0	1.0	101.0	1.0
14.0	31000000.0	0.0	0.0	4.0	0.0	1.0
11.0	48000000.0	0.0	0.0	1.0	94.0	0.0
7.0	49000000.0	0.0	0.0	3.0	36.0	1.0
11.0	9200000.0	0.0	0.0	1.0	37.0	0.0
10.0	1700000.0	0.0	0.0	1.0	57.0	0.0

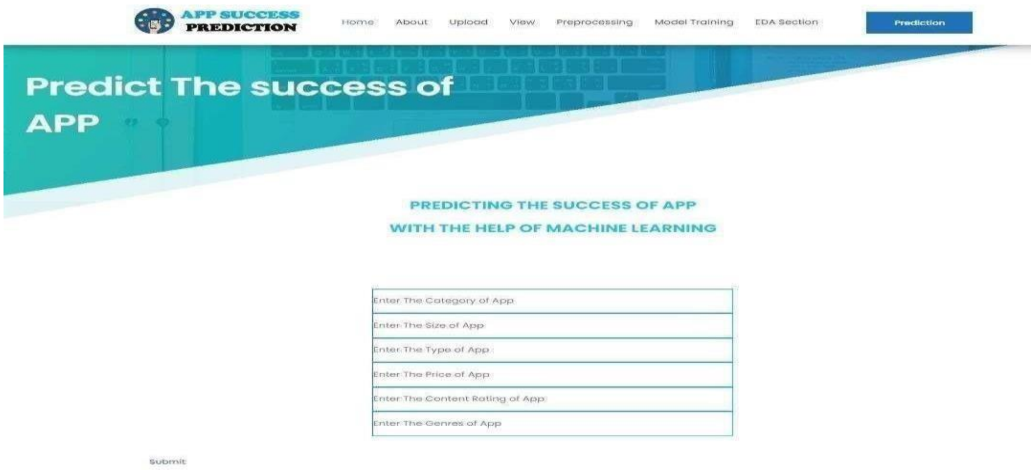
Preprocessing



Model Training



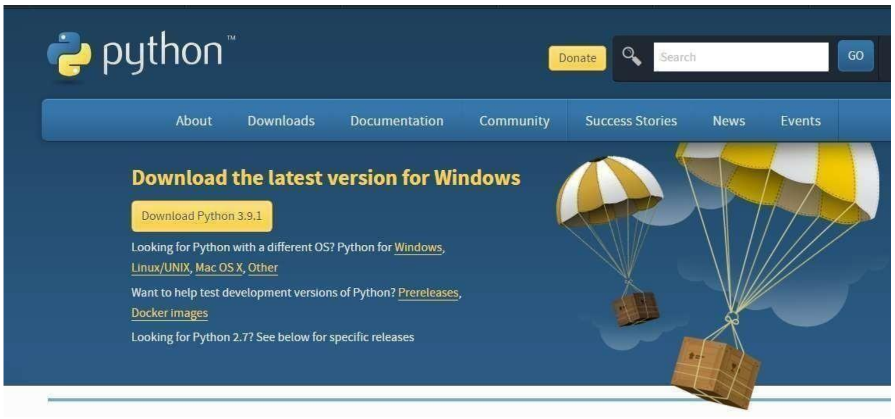
Prediction



6.1 Bibliography

Software Installation for Machine Learning

Installing Python



Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

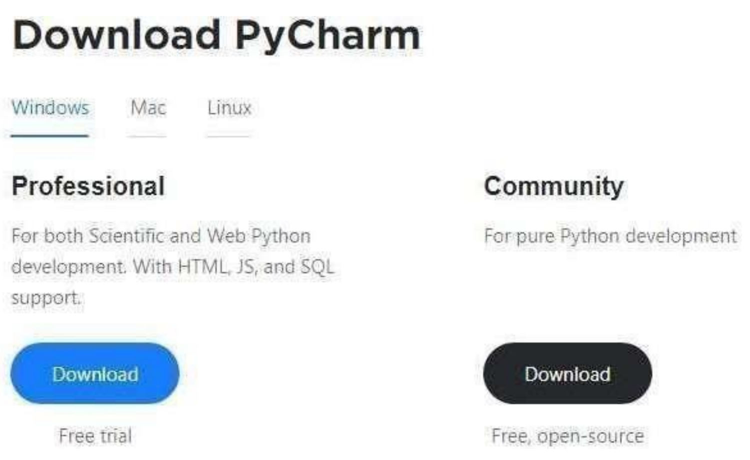
Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

1. Once the download is complete, run the exe for install Python. Now click on Install Now.
2. You can see Python installing at this point.
3. When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

Installing PyCharm

1. Once the download is complete, run the exe for install PyCharm. The setupwizard should have started. Click "Next".
2. On the next screen, Change the installation path if required. Click "Next".
3. On the next screen, you can create a desktop shortcut if you want and click on "Next".
4. Choose the start menu folder. Keep selected Jet Brains and click on "Install".
5. Wait for the installation to finish. Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the "Run PyCharm Community Edition" box first and click "Finish".
6. After you click on "Finish," the Following screen will appear.


Download PyCharm
Windows
Mac
Linux
Professional
For both Scientific and Web Python development. With HTML, JS, and SQL support.
Download
Free trial
Community
For pure Python development
Download
Free, open-source

6.2 Introduction To Python

6.2.1 What Is a Script?

Up to this point, I have concentrated on the interactive programming capability of Python. This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode

Scripts are reusable

Basically, a script is a text file containing the statements that comprise a Python program. Once you have created the script, you can execute it over and over without having to retype it each time.

Scripts are editable

Perhaps, more importantly, you can make different versions of the script by modifying the statements from one file to the next using a text editor. Then you can execute each of the individual versions. In this way, it is easy to create different programs with a minimum amount of typing.

You will need a text editor

Just about any text editor will suffice for creating Python script files.

You can use *Microsoft Notepad, Microsoft WordPad, Microsoft Word,* or just word.

Difference between a script and program

Scripts are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or byte code, whereas the applications they control are traditionally compiled to native machine code.

Program

The program has an executable form that the computer can use directly to execute the instructions.

The same program in its human-readable source code form, from which executable programs are derived (e.g., compiled)

6.2.2 Python

What is Python? Chances you are asking yourself this. You may have found this book because you want to learn to program but don't know anything about programming languages. Or you may have heard of programming languages like C, C++, C#, or Java and want to know what Python is and how it compares to "big name" languages. Hopefully I can explain it for you.

6.2.3 Python concepts

If you're not interested in the how's and whys of Python, feel free to skip to the next chapter. In this chapter I will try to explain to the reader why I think Python is one of the best languages available and why it's a great one to start programming with.

• Open-source general-purpose language.
• Object Oriented, Procedural, Functional
• Easy to interface with C /Java/Fortran
• Easy-is to interface with C++ (via SWIG)
• Great interactive environment
• Great interactive environment

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

6.3 History of Python

Python was developed by Guido van Possum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and UNIX shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Possum still holds a vital role in directing its progress.

6.4 Python Features

- Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read − Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain − Python's source code is fairly easy-to maintained.
- A broad standard library − Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows.
- Interactive Mode − Python has support for an interactive mode which allowsinteractive testing and debugging of snippets of code.

- Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable − you can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases − Python provides interfaces to all major commercial database
- GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable − Python provides a better structure and support for large programs than shell scripting.
- Apart from the above-mentioned features, Python has a big list of good features, few are listed below –
- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.
- In Python, we don't need to specify the data-type of the variable. When we assign some to the variable, it automatically allocates the memory to the variable at run time. Suppose we assigned integer value 15 to **x,** then we don't need to write **int x = 15.** Just write x = 15.The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language our code.

It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

6.5 Dynamic vs Static

Types Python is a dynamic-typed language. Many other languages are static typed, such as C/C++ and Java. A static typed language requires the programmer to explicitly tell the computer what type of “thing” each data value is. For example, in C if you had a variable that was to contain the price of something, you would have to declare the variable as a “float” type. This tells the compiler that the only data that can be used for that variable must be floating point number, i.e. a number with a decimal point.If any other data value was assigned to that variable, the compiler would give an error when trying to compile the program.

Python, however, doesn’t require this. You simply give your variables names and assign values to them. The interpreter takes care of keeping track of what kinds of objects your program is using. This also means that you can change the size of the values as you develop the program. Say you have another decimal number (a.k.a. a floating-point number) you need in your program.

If you declare a variable to be a float but later on assign a value that is too big to it, your program will fail; you will have to go back and change that variable to be a double. With Python, it doesn’t matter. You simply give it whatever number you want and Python will take care of manipulating it as needed. It even works for derived values.

For example, say you are dividing two numbers. One is a floating point number and one is an integer. Python realizes that it’s more accurate to keep track of decimals so it automatically calculates the result as a floating point number With a static typed language, you have to decide the memory size the variable cant ake when you first initialize that variable. A double is a floating point value that can handle a much larger number than a normal float (the actual memory sizes depend on the operating environment).

If you declare a variable to be a float but later on assign a value that is too big to it, your program will fail; you will have to go back and change that variable to be a double. With Python, it doesn’t matter. You simply give it whatever number you want and Python will take care of manipulating it as needed. It even works for derived values.

For example, say you are dividing two numbers. One is a floating point number

and one is an integer. Python realizes that it’s more accurate to keep track of decimals so it automatically calculates the result as a floating point number.

Variables

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory. Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

6.6 Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types −

- Numbers
- String
- List
- Tuple
- Dictionary

Python Numbers

Number data types store numeric values. Number objects are created when you assign a value to them.

Python Strings

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of string scan be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type. The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses. The main differences between lists and tuples are: Lists are enclosed in brackets ([]) and their elements and size can be changed, while tuples are enclosed in parentheses (()) and cannot be updated. Tuples can be thought of as read-only lists.

Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned.

Different modes in python

Python has two basic modes: Normal and Interactive.

The Normal mode is the mode where the scripted and finished pie files are run in the Python interpreter. Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole procedure.

6.7 20 Python libraries

1. Requests. The most famous http library written by Kenneth remits. It’s a must have for every python developer.
2. Scrappy. If you are involved in web scraping then this is a must have library for you. After using this library you won’t use any other.
3. Python. A guy toolkit for python. I have primarily used it in place of tinder. You will really love it.
4. Pillow. A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.
5. SQL Alchemy. A database library. Many love it and many hate it.

6. Beautiful Soup. I know it's slow but this xml and html parsing library is very useful for beginners.

7. Twisted. The most important tool for any network application developer. It has a very beautiful ape and is used by a lot of famous python developers.

8. Numbly. How can we leave this very important library? It provides some advance math functionalities to python.

9. Skippy. When we talk about numbly then we have to talk about spicy. It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.

10. Matplotlib. A numerical plotting library. It is very useful for any data scientist or any data analyzer.

11. Pygmy. Which developer does not like to play games and develop them? This library will help you achieve your goal of 2d game development.

12. piglet. A 3d animation and game creation engine. This is the engine in which the famous python port of mine craft was made

13. Pit. A GUI toolkit for python. It is my second choice after python for developing GUI's for my python scripts.

14. Pit. Another python GUI library. It is the same library in which the famous Bittorrent client is created.

15. Scaly. A packet sniffer and analyzer for python made in python.

16. Pywin32. A python library which provides some useful methods and classes for interacting with windows.

17. Notch. Natural Language Toolkit – I realize most people won't be using this one, butit's generic enough. It is a very useful library if you want to manipulate strings. But its capacity is beyond that. Do check it out.

18. Nose. A testing framework for python. It is used by millions of python developers. Itis a must have if you do test driven development.

19. Simply. Simply can do algebraic evaluation, differentiation, expansion, complex numbers, etc. It is contained in a pure Python distribution.

20. Python. I just can't stress enough how useful this tool is. It is a python prompt on steroids. It has completion, history, shell capabilities, and a lot more. Make sure that you take a look at it.

21. Notch. Natural Language Toolkit – I realize most people won't be using this one, but it’s generic enough. It is a very useful library if you want to manipulate strings. But its capacity is beyond that. Do check it out.

6.8 Numpy

Numpy’s main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In numbly dimensions are called *axes*. The number of axes is *rank*.

- Offers Matlab-ish capabilities within Python
- Fast array operations
- 2D arrays, multi-D arrays, linear algebra etc.

Matplotlib

High quality plotting library.

CHAPTER 7
PYTHON CLASS AND OBJECT

These are the building blocks of OOP. Class creates a new object. This object can be anything, whether an abstract data concept or a model of a physical object, e.g. a chair. Each class has individual characteristics unique to that class, including variables and methods. Classes are very powerful and currently “the big thing” in most programming languages.

Hence, there are several chapters dedicated to OOP later in the book. Previously, you learned how to use functions to make your program do something. Now will move into the big, scary world of Object-Oriented Programming (OOP). To be honest, it took me several months to get a handle on objects When I first learned C and C++, I did great; functions just made sense forme.Having messed around with BASIC in the early ’90s, I realized functions were just like subroutines so there wasn’t much new to learn.However, when my C++ course started talking about objects, classes, and all the new features of OOP, my grades definitely suffered.

Once you learn OOP, you’ll realize that it’s actually a pretty powerful tool. Plus many Python libraries and APIs use classes, so you should at least be able to understand what the code is doing.One thing to note about Python and OOP: it’s not mandatory to use objects in your code in a way that works best; maybe you don’t need to have a full-blown class with initialization code and methods to just return a calculation. With Python, you can get as technical as you want.As you’ve already seen, Python can do just fine with functions. Unlike languages such as Java,you aren’t tied down to a single way of doing things; you can mix functions and classes as necessary in the same program. This lets you build the code objects are an encapsulation of variables and functions into a single entity. Objects get their variables and functions from classes. Classes are essentially a template to create your objects

Here’s a brief list of Python OOP ideas:

- The class statement creates a class object and gives it a name. This creates a new namespace.
- Assignments within the class create class attributes. These attributes are accessed by qualifying the name using dot syntax: ClassName.Attribute.

- Class attributes export the state of an object and its associated behavior. These attributes are shared by all instances of a class.
- Calling a class (just like a function) creates a new instance of the class. This is where the multiple copies part comes in.
- Each instance gets ("inherits") the default class attributes and gets its own namespace.This prevents instance objects from overlapping and confusing the program.
- Using the term self identifies a particular instance, allowing for per-instance attributes. This allows items such as variables to be associated with a particular instance.
- Assignments within the class create class attributes. These attributes are accessed by qualifying the name using dot syntax: ClassName.Attribute.
- Operator overloading simply means that objects that you create from classes can respond that are already defined within Python, such as addition, slicing, printing.

7.1 Inheritance

First off, classes allow you to modify a program without really making changes to it.To elaborate, by sub classing a class, you can change the behavior of the program by simply adding new components to it rather than rewriting the existing components. As we've seen, an instance of a class inherits the attributes of that class.However, classes can also inherit attributes from other classes. Hence, a subclassinherits from a superclass allowing you to make a generic superclass that is specializedvia subclasses. The subclasses can override the logic in a superclass, allowing you to change the behavior of your classes without changing the superclass at all.

7.2 Operator Overloads

Operator overloading simply means that objects that you create from classes can respond that are already defined within Python, such as addition, slicing, printing, etc Even though these actions can be implemented via class methods, using overloading the behavior closer to Python's object modeland the object interfaces are more consistent to Python's built-in objects, hence overloading is easier to learn and use.User-made classes can override nearly all of Python's built-in operation methods

7.2 Exceptions

I’ve talked about exceptions before but now I will talk about them in depth. Essentially,exceptions are events that modify program’s flow, either intentionally or due to errors.They are special events that can occur due to an error, e.g. trying to open a file that doesn’t exist, or when the program reaches a marker, such as the completion of a loop.Exceptions, by definition, don’t occur very often; hence, they are the "exception to the rule" and a special class has been created for them. Exceptions are everywhere in Python.Virtually every module in the standard Python library uses them, and Python itself will raise them in a lot of different circumstances.

Here are just a few examples:

• Accessing a non−existent dictionary key will raise a KeyError exception.

• Searching a list for a non−existent value will raise a Value Error exception

. • Calling a non−existent method will raise an Attribute Error exception.

• Referencing a non−existent variable will raise a Name Error exception.

• Mixing data types without coercion will raise a Type Error exception.

One use of exceptions is to catch a fault and allow the program to continue working we have seen this before when we talked about files.This is the most common way to use exceptions. When programming with the Python command line interpreter, you don’t need to worry about catching exceptions.Your program is usually short enough to not be hurt too much if an exception occurs.Plus, having the exception occur at the command line is a quick and easy way to tell if your code logic has a problem.

However, if the same error occurred in your real program, it will fail and stop working. Exceptions can be created manually in the code by raising an exception.It operates exactly as a system-caused exceptions, except that the programmer is doing it on purpose. This can be for a number of reasons. One of the benefits of using exceptions is that, by their nature, they don’t put any overhead on the code processing.Because exceptions aren’t supposed to happen very often, they aren’t processed until they occur.Exceptions can be thought of as a special form of the if/elf statements. You can realistically do the same thing with if blocks as you can with exceptions.However, as already mentioned, exceptions aren’t processed until they occur; if blocks are processed all the time.Proper use of exceptions can help the performance of your program.The more infrequent the error might occur, the better off you are to use exceptions; using if blocks requires Python to always test extra conditions before continuing. To make a custom exception, the programmer determines which base exception to use as the class toinherit from, e.g. making an exception for negative numbers or one for imaginary numbers would probably fall under the Arithmetic Error exception class.To make a custom exception, simply inherit the base exception and define what it will do. Exceptions can be created manually in the code by raising an exception.It operates exactly as a system-caused exceptions, except that the programmer is doing it on purpose. This can be for a number of reasons. One of the benefits of using exceptions isthat, by their nature, they don’t put any overhead on the code processing.Because exceptions aren’t supposed to happen very often, they aren’t processed until they occur.

7.3 Python Modules

Python allows us to store our code in files (also called modules). This is very useful for more serious programming, where we do not want to retype a long function definition from the very beginning just to change one mistake. In doing this, we are essentially defining our own modules, just like the modules defined already in the Python library.To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a *module*; definitions from a module can be *imported* into other modules or into the *main* module.

7.4 Python Code

As indicated above, code is usually developed in a file using an editor. To test the code, import it into a Python session and try torun it.Usually there is an error, so you go back to the file, make a correction, and test again. This process is repeated until you are satisfied that the code works. T His entire process is known as the development cycle.There are two types of errors that you will encounter. Syntax errors occur when the form of some command is invalid.This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons. Python will always give an error message for a syntax error.

7.5 Functions In Python

It is possible, and very useful, to define our own functions in Python. Generally speaking,if you need to do a calculation only once, then use the interpreter. But when you or others have need to perform a certain type of calculation many times, then define a function.

You use functions in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program called when needed. That means that a function is a piece of code written to carry out a specified task.

To carry out that specific task, the function might or might not need multiple inputs. When the task is carved out, the function can or cannot return one or more values. There are three types of functions in python:Help (), min (), print ().Namespaces in Python are implemented as Python dictionaries, this means it is a mapping from names (keys) to objects (values). The user doesn't have to know this to write a Python program and when using namespaces.

Some namespaces in Python:

- global names of a module
- local names in a function or method invocation
- built-in names: this namespace contains built-in functions (e.g. abs(), camp(), ...) .

7.6 Garbage Collection

Garbage Collector exposes the underlying memory management mechanism of Python, the automatic garbage collector. The module includes functions for controlling how the collector operates and to examine the objects known to the system, either pending collection or stuck in reference cycles and unable to be freed.

7.7 Python XML Parser

XML is a portable, open source language that allows programmers to develop applications that can be read by other applications, regardless of operating systemand/or developmental language.What is XML? The Extensible Markup Language XML is a markup language much like HTML or SGML.This is recommended by the World Wide Web Consortium and available as an open standard. XML is extremely useful for keeping track of small to medium amounts of data without requiring a SQL-based backbone.XML Parser Architectures and APIs the Python standard library provides a minimal but useful set of interfaces to work with XML.The two most basic and broadly used APIs to XML data are the SAX and DOM interfaces. Simple API for XML SAX: Here, you register call-backs for events of interest and then let the parser proceed through the document.This is useful when your documents are large or you have memory limitations, it parses the file as it reads it from disk and the entire file is never stored in memory.Document Object Model DOM API : This is a World Wide Web Consortium recommendation wherein the entire file is read into memory and stored in a hierarchical tree − based form to represent all the features of an XML document.SAX obviously cannot process information as fast as DOM can when working with large files. On the other hand, using DOM exclusively can really kill your resources, especially if used on a lot of small files.SAX is read-only, while DOM allows changes to the XML file. Since these two di fferent APIs literally complement each other, there is no reason why you cannot use them both.

7.8 Python Web Frameworks

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications.

Why are web frameworks useful?

Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.Common web framework functionality Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

1. URL routing
2. HTML, XML, JSON, and other output format tinplating
3. Database manipulation
4. Security against Cross-site request forgery (CSRF) and other attacks
5. Session storage and retrieval

Not all web frameworks include code for all of the above functionality. Frameworks fall on the spectrum from executing a single use case to providing every known web framework feature to every developer. Some frameworks take the “batteries-included" approach where everything possible comes bundled with the framework while others have a minimal core package that is amenable to extensions provided by other packages.

Comparing web frameworks

There is also a repository called compare-python-web-frameworks where the same web application is being coded with varying Python web frameworks, tinplating engines and object.

Web framework resources

- When you are learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.
- What is a web framework? Is an in-depth explanation of what web frameworks are and their relation to web servers?
- Jingo vs. Flash vs. Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.
- This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases.
- Python's web frameworks benchmarks is a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as well as retrieving datafrom the database and rendering it in a template. There wereno conclusive results but theoutput is fun to read about nonetheless.
- What web frameworks do you use and why are they awesome? Is a language agnostic Reedit discussion on web frameworks? It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks.
- This user-voted question & answer site asked "What are the best general purpose Pythonzweb frameworks usable in production?" The votes aren't as important as the list of the many frameworks that are available to Python developers.

Web frameworks learning checklist

1. Choose a major Python web framework (Jingo or Flask are recommended) and stick withit. When you're just starting it's best to learn one framework first instead of bouncing around trying to understand every framework.

2. Work through a detailed tutorial found within the resource’s links on theframework's page.
3. Studyopen-source examples built with your framework of choice so you can take parts of those projects and reuse the code in your application.
4. Build the first simple iteration of your web application then go to the deployment section to make it accessible on the web.

CHAPTER 8
SYSTEM STUDY

8.1 System Study

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.

8.2 Types Of Test

8.2.1 Unit Test

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.

8.2.2 Integration Test

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. Functional test Functional tests provide systematic demonstration, 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.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.

8.2.3 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.

8.2.4 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.

8.2.5 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

8.2.6 Unit Testing

Unit testing is usually conducted as part ofa 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.Test strategy and approach Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

8.2.7 Integration Testing

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. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.2.8 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CONCLUSION

The project has concluded that, to know the success or failure status of the application we used the machine learning based method like Random Forest, Support Vector Machine, XG Boost , Decision Tree classification techniques to get better accuracy. The algorithm works on success rate of application andit will gives the results whether the application is successful or fail.

REFERENCES

[1] M. R. Islam, "Numeric rating of apps on google play store by sentiment analysis on user reviews," in 2014 International Conference on Electrical Engineering and Information & Communication Technology. IEEE, apr 2014. [Online]. Available:https://doi.org/10.1109/iceeict. 2014.6919058

[2] S. Nishida and Y. Shinkawa. Data Integrity in Cloud Transactions, Proceedings of the 4th International Conference on Cloud Computing and Services Science, pages 457–462, Barcelona, Spain, April 3–5, 2014.

[3] S. Nishida and Y. Shinkawa. CPN Based GAE Performance Prediction Framework, Proceedings of the 9th International Conference on Software Engineering and Applications, pages 401–406, Vienna, Austria, August 29–31, 2014.

[4] D. Pritchett. BASE: An ACID Alternative, ACM QUEUE, Volume 6 Issue 3, pages48– 55, ACM, 2008.

[5] D. Sanderson, Programming Google App Engine, 1st ed. Sebastopol, Canada: O'Reilly Media, Inc, 2009.

[6] B. Beach, Pro PowerShell for Amazon Web Services: Devops for the Aws Cloud, 1st ed. New York City, US: APRESS, 2014.

[7] B. Gregg, Systems Performance: Enterprise and the Cloud, 1st ed. NewJersey, US: Prentice Hall, 2013.

[8] D. Jordan and C. Russell, Java Data Objects, 1st ed. Sebastopol, Canada: O'Reilly Media, Inc, 2003. [9] A. David et al., Uppaal SMC tutorial, International Journal on Software Tools for Technology Transfer, Volume 17, Issue 4, pages 397–415, Springer, 2015.

[9] U. Narayan Bhat, An Introduction to Queueing Theory: Modeling and Analysis in Applications, 1st ed. Boston, US: Birkhauser, 2008. ¨ [11] Commons Math:

The Apache Commons Mathematics Library, http://commons.apache.org/proper/commons-math/, 2015.