Domain: Machine Learning **Technology: Python**

PROJECT REPORT

ON

ADVANCED PREDICTION OF PERFORMANCE OF A STUDENT IN AN UNIVERSITY USING MACHINE LEARNING TECHNIQUES

Submitted in the partial fulfillments of the requirements

For the degree of B. tech

By

Aditi Bhardwaj(1706810019)

Shreya Singh(1706810277)

Gazala Malik(1706810104)

Under supervision of:-

Dr. Mukesh Rawat

(professor, department of CSE)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Meerut Institute Of Engineering and Technology

Meerut-250005

Domain: Machine Learning **Technology: Python**

DECLARATION

I hereby declare that this submission is my own work and that to the best of my knowledge and belief. It contains neither material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Signature:

Name: **Shreya Singh**

Roll No: **1706810277**

Date:

Signature:

Name: **Aditi Bhardwaj**

Roll No: **1706810019**

Date:

Signature:

Name: **Gazala Malik**

Roll No: **1706810104**

Date:

Domain: Machine Learning **Technology: Python**

CERTIFICATE

This is to certify that Project Report entitled – ADVANCED PREDICTION OF PERFORMANCE OF A STUDENT IN AN UNIVERSITY USING MACHINE LEARNING TECHNIQUES which is submitted by Aditi Bhardwaj(1706810019), Shreya Singh(1706810277), Gazala Malik(17068104), in partial fulfillment of the requirement for the award of degree B. Tech. in Department of CSE, Of Dr. A.P.J. Abdul Kalam Technical University, U.P., Lucknow., is a record of the candidate own work carried out by him/her under my/our supervision. The matter embodied in this Project report is original and has not been submitted for the award of any other degree.

Date: **Supervisor:**

Dr.Mukesh Rawat

(Professor,CSE Department)

MIET,Meerut

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech Final Year. We owe special debt of gratitude to our guide Prof. (Dr.) Mukesh Rawat , Department of CSE, Meerut Institute of Engineering and Technology, Meerut for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavours have seen light of the day. We also take the opportunity to acknowledge the contribution of Dr. Sunil Kumar, Department of CSE, Meerut Institute of Engineering And Technology, Meerut for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature : Signature :

Name : Aditi Bhardwaj Name :Shreya Singh

Roll No. 1706810019 Roll No. : 1706810277

Date: Date:

Signature :

Name : Gazala Malik

Roll No. :1706810104

Date :

Domain: Machine Learning **Technology: Python**

ABSTRACT

Predicting academic performance is an important task for the students in university, college, and school, etc. Machine Learning is a field of computer science that makes the computer to learn itself without any help of external programs. The dataset used in this project is stored in a cloud server and accessed using queries as and when required. There are two approaches for machine learning techniques one is supervised learning and the other one is unsupervised learning. In unsupervised learning, K-means clustering are being used and in supervised, ensemble techniques like Random Forest and XgBoost algorithm are implemented. Nowadays evaluating the student performance of any organization is going to play a vital role to train the students. All of the above algorithms were combined and used for student evaluation and a possible suggestion to the student is provided to improve their career.

KEYWORDS: Predicting Academic Performance of Students, Machine Learning, K-Means, XG Boost, Random Forest, Ensemble, and Cloud Server.

INTRODUCTION

Student's academic performance is a crucial part of an academic institution. This is considered as one of the important measures for many superior universities. Some researchers stated that the student's academic performance can be measured through learning assessment and co-curriculum activities. Though, the majority of researchers have mentioned that the student's past performances, achievements, and grades can play a vital role to predict the student's success rate.

Predominantly, most of the higherlevel institutions use grade as the main measure to assess student's performance. In addition, course structure, student behavior and extracurricular activities will affect the student's academic performance. The student's academic program can be well planned during their sophomore period of studies in an institution to analyze the performance of students.

Domain: Machine Learning **Technology: Python**

At present, machine learning algorithms are most popular to evaluate student's academic performance that has been extensively applied in the education sector. The topic of explanation and prediction of academic performance is widely researched. The ability to predict student performance is very important in educational environments. Increasing student success is a long term goal in all academic institutions. If educational institutions can predict students' academic performance early before their final examination, then extra effort can be taken to arrange proper support for the low performing students to improve their studies and help them to success.

On the other hand, identifying attributes that affect course success rate can assist in courses improvement. Newly developed web-based educational technologies and the application of quality standard offer researchers' unique opportunities to study how students learn and what approaches to learning lead to success.

Many organizations are also using cloud based infrastructure to enable seamless accessibility of their systems from anywhere around the world and for cheap. We will also incorporate these technologies in our project by storing the dataset in a cloud server.

LITERATURE REVIEW

[1] Vairachilai S, Vamshidharreddy, "Student's Academic Performance Prediction Using Machine Learning Approach", *IJAST*, vol. 29, no. 9s, pp. 6731 - 6737, Jun. 2020.

Predicting academic performance is an important task for the students in university, college, and school, etc. The factors which affect the student's academic performance are class quizzes, assignments, lab exams, mid, and final exams. The student's academic performance should be informed to the class teacher in advance that will decrease the student's dropout and increase the performance. In this paper, machine learning classification algorithms such as decision tree, Support Vector Machine (SVM), and Naive Bayes are implemented to predict the student's academic performance. The performance of an algorithm has been evaluated based on confusion matrix, accuracy, precision, recall, and F1 score. The obtained result shows that the Naive Bayes classification algorithm performs better.

Summary: This journal discusses about SVM and Naïve Bayes for students' performance prediction using students score/marks.

[2] Sharma, Himani & Kumar, Sunil. (2016). A Survey on Decision Tree Algorithms of Classification in Data Mining. International Journal of Science and Research (IJSR). 5.

As the computer technology and computer network technology are developing, the amount of data in information industry is getting higher and higher. It is necessary to analyze this large amount of data and extract useful knowledge from it. Process of extracting the useful knowledge from huge set of incomplete, noisy, fuzzy and random data is called data mining. Decision tree classification technique is one of the most popular data mining techniques. In decision tree divide and conquer technique is used as basic learning strategy. A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node. This paper focus on the various algorithms of Decision tree (ID3, C4.5, CART), their characteristic, challenges, advantage and disadvantage.

Summary: In this paper, we learn about Decision Tree, types of Decision tree (ID3, C4.5, CART etc). It also discusses about the advantages and disadvantages of Decision Tree.

[3] Kaushik, Manju & Mathur, Bhawana. (2014). Comparative Study of K-Means and Hierarchical Clustering Techniques. International journal of Software and Hardware Research in Engineering. 2. 93-98.

Clustering is a process of keeping similar data into groups. Clustering is an unsupervised learning technique as every other problem of this kind; it deals with finding a structure in a collection of unlabeled data. Many types of clustering methods are— hierarchical, partitioning, density –based, model-based, grid –based, and soft-computing methods. In this paper compare with k-Means Clustering and Hierarchical Clustering Techniques. Strength and weakness of both Clustering Techniques and their methodology and process.

Summary: In this paper, we learn clustering algorithms like Kmeans and Agglomerative clustering and their comparisons.

[4] Kabakchieva D (2012) Student performance prediction by using data mining classification algorithms. IJCSMR 1: 686-690.

This paper presents the results from data mining research, performed at one of the famous and prestigious Bulgarian universities, with the main goal to reveal the high potential of data mining applications for university management and to contribute to more efficient university enrolment campaigns and to attracting the most desirable students. The research is focused on the development of data mining models for predicting student performance, based on their personal, pre-university and university-performance characteristics. The dataset used for the research purposes includes data about students admitted to the university in three consecutive years. Several well-known data mining classification algorithms, including a rule learner, a decision tree classifier, a neural network and a Nearest Neighbour classifier, are applied on the dataset. The performance of these algorithms is analyzed and compared.

Domain: Machine Learning **Technology: Python**

SCOPE:

- Education organizations can use them to improve students' performance.
- Government agencies can decide policies and regulations.
- Ed-Tech industries can use them to set a better curriculum which suits them well.

EXISTING METHOD

Earlier works involves using older Machine Learning algorithms like Logistic regression. They suffer from very low accuracies. There are Deep Learning based predictions as well which uses neural networks for prediction but they have high complexities and they fail to individually identify the important features for prediction.

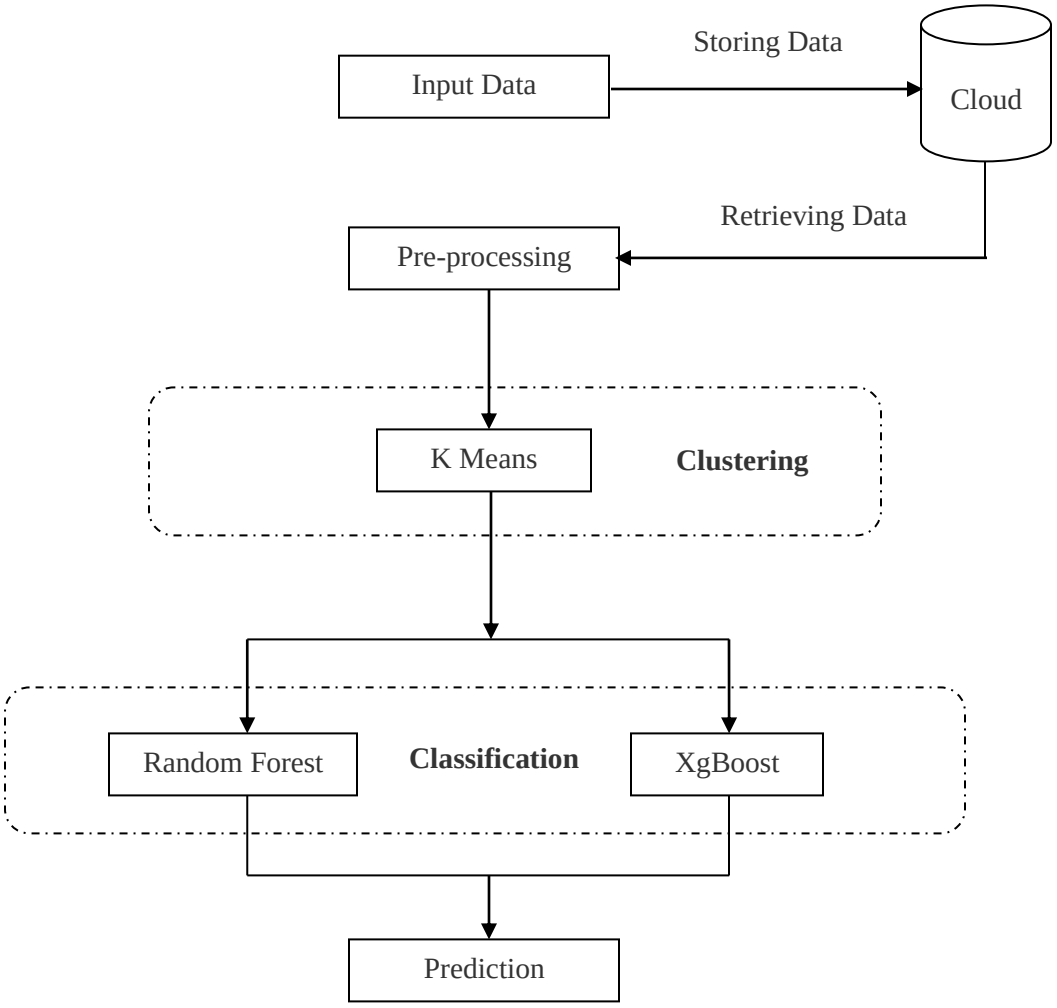
DISADVANTAGES:

- Low accuracy.
- High Variance.
- Incurs bias in classification.
- High Complexity.

PROPOSED METHOD

In our research & extensive literature survey, we found that Random Forest works fine for Student's performance prediction with a great accuracy but it can be further increased by other tree based algorithm like XgBoost. Xgboost, when tuned properly can generate significant increase in performance. Also, we have used both unsupervised and supervised learning methods. The data is stored in a cloud server which makes it easy to access from anywhere and this system generates predictions and provides suggestions to the student.

Flow of the project:



ADVANTAGES:

- Higher Accuracy.
- Low variance in classification.
- Bias due to assumption about dataset are minimum or even nonexistent.
- Low Complexities.
- Easy access of data**.**

APPLICATIONS:

- Schools.
- Universities and Colleges.
- Education department.
- Ed-Tech industries.
- Government agencies.

HARDWARE & SOFTWARE REQUIREMENTS

H/W CONFIGURATION:

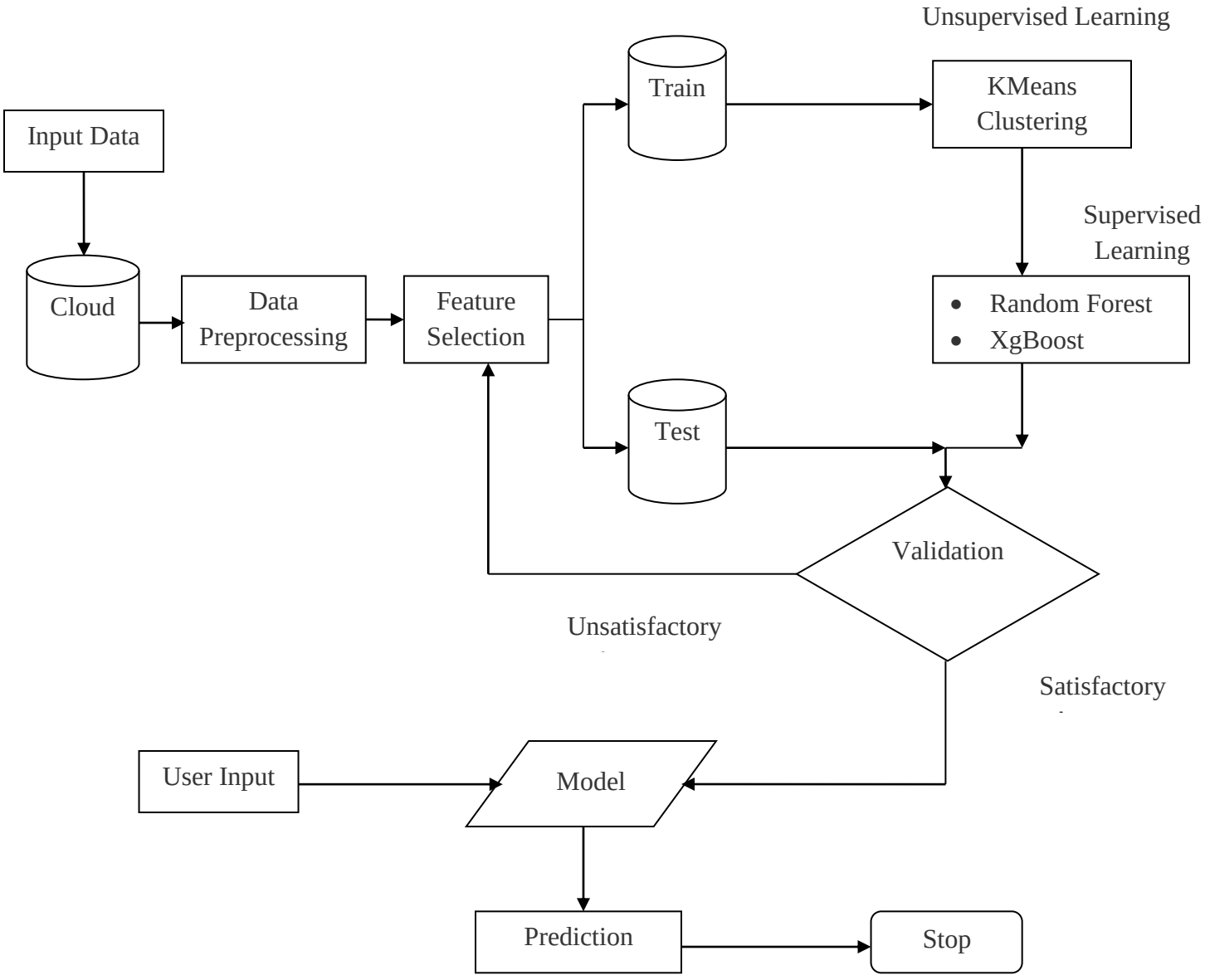
- Processor - I3/Intel Processor
- RAM - 4GB (min)
- Hard Disk - 128 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - Any

S/W CONFIGURATION:

- Operating System : Windows 7+
- Server side Script : Python 3.6+
- IDE : PyCharm IDE
- Libraries Used : Pandas, Numpy, Sci-Kit Learn, Matplotlib, Seaborn,Flask, Pickle.
- Dataset : Students' Academic Performance Dataset.

Domain: Machine Learning **Technology: Python**

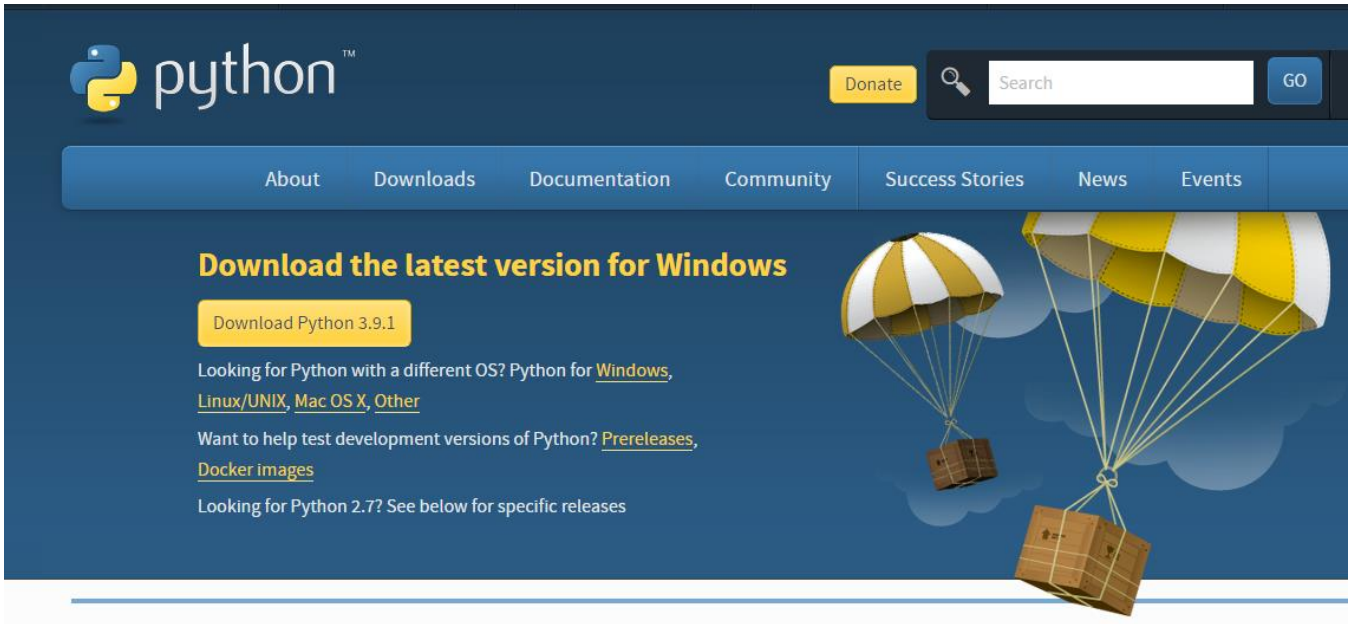
ARCHITECTURE



SOFTWARE INSTALLATION FOR MACHINE LEARNING PROJECTS:

Installing Python:

1. To download and install Python visit the official website of Python https://www.python.org/downloads/ and choose your version.



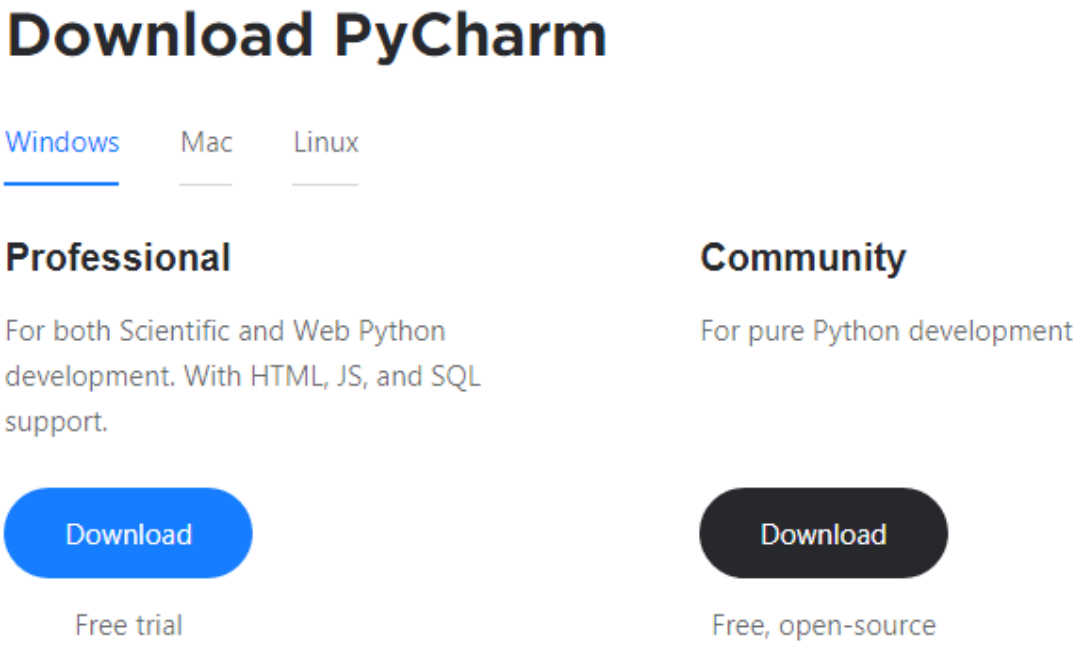
2. Once the download is complete, run the exe for install Python. Now click on Install Now.

3. You can see Python installing at this point.

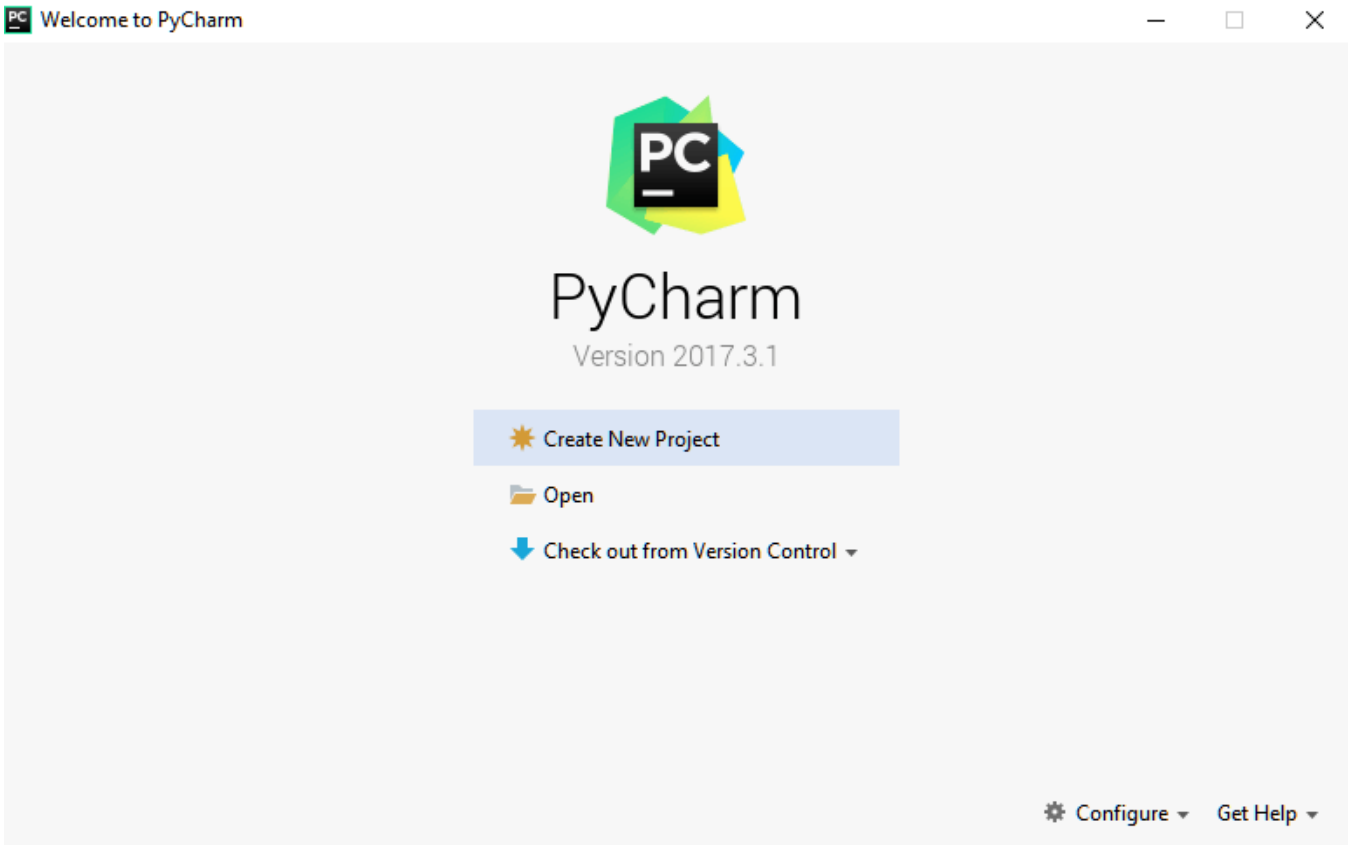
4. When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

Installing PyCharm:

1. To download PyCharm visit the website https://www.jetbrains.com/pycharm/download/ and Click the "DOWNLOAD" link under the Community Section.



2. Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.
3. On the next screen, Change the installation path if required. Click “Next”.
4. On the next screen, you can create a desktop shortcut if you want and click on “Next”.
5. Choose the start menu folder. Keep selected JetBrains and click on “Install”.
6. Wait for the installation to finish.
7. 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”.
8. After you click on "Finish," the Following screen will appear.

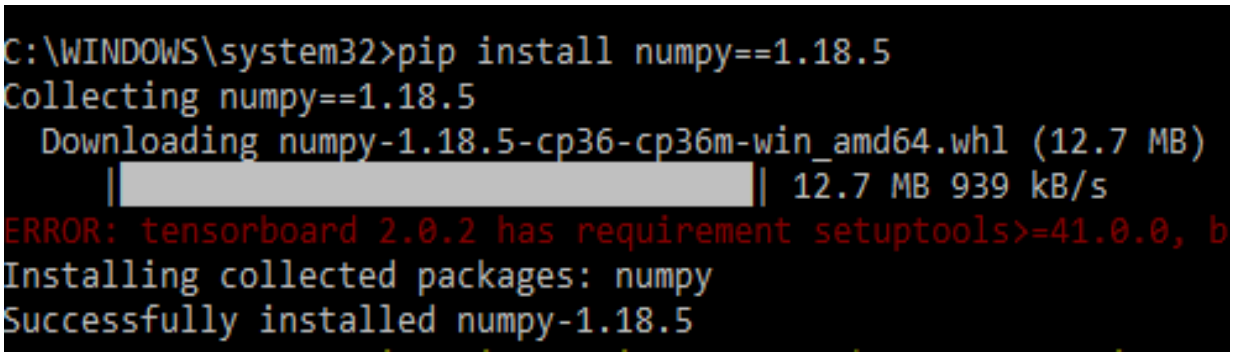


9. You need to install some packages to execute your project in a proper way.

10. Open the command prompt/ anaconda prompt or terminal as administrator.

11. The prompt will get open, with specified path, type “pip install package name” which you want to install (like numpy, pandas, seaborn, scikit-learn, matplotlib.pyplot)

Ex: pip installnumpy



MODULES

System

User

1. System:

1.1 Takes Dataset:

The system takes in the .csv data uploaded by the user and load it in to a cloud server which acts as an interconnection for communication purposes using queries.

1.2 Preprocessing:

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. Cleaning the data refers to removing the null values, filling the null values with meaningful value, removing duplicate values, removing outliers, removing unwanted attributes. If dataset contains any categorical records means convert those categorical variables to numerical values.

Here, we are removing rows with null values, Ordinal encoding the predictor variable, Label encoding the target variable.

1.3 Splitting Dataset:

The dataset is split into test and train dataset with a test size as entered by the user.

1.4 Model Training:

The models:

- K Means Clustering:
  - The algorithm will categorize the items into k groups of similarity.
  - To calculate that similarity, we will use the euclidean distance as measurement.
  - First we initialize k points, called means, randomly.
  - We categorize each item to its closest mean and we update the mean's coordinates, which are the averages of the items categorized in that mean so far.

- We repeat the process for a given number of iterations and at the end, we have our clusters.

- Random Forest:
  - Random forest is a supervised learning algorithm.
  - The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method.
  - The general idea of the bagging method is that a combination of learning models increases the overall result.
  - It is also one of the most used algorithms, because of its simplicity and diversity.
  - It can be used for both classification and regression tasks.

- XgBoost:
  - Can be run on both single and distributed systems (Hadoop, Spark)
  - XG Boost is used in supervised learning (regression and classification problems).
  - Supports parallel processing.
  - Cache optimization.
  - Efficient memory management for large datasets exceeding RAM.
  - Has a variety of regularizations which helps in reducing over fitting.
  - Auto tree pruning – Decision tree will not grow further after certain limits internally.
  - Can handle missing values.
  - Has inbuilt Cross-Validation.
  - Takes care of outliers to some extent.

1.5 Prediction:

The system takes input from the user for a student and predicts using the best model among the trained ones.

2. User

2.1 Upload Data:

The user uploads a .csv dataset from the webb application which contains students performance parameters.

2.2 View Data:

The user views the data in the webapp after it is cleaned. The user can also search any record by typing any keyword in the search box.

2.3 Input Test Size:

The user enters the desired test dataset size in percentages in the webapp which will be used by the system for splitting the data.

2.4 Model Testing:

The user tests all possible models trained by the system using the testing dataset and views their accuracies.

2.5 Prediction:

The user enters the information about a student which is used by the system to predict the performance of that student.

ALGORITHM:

K Means Clustering:

There is an algorithm that tries to minimize the distance of the points in a cluster with their centroid – the k-means clustering technique.

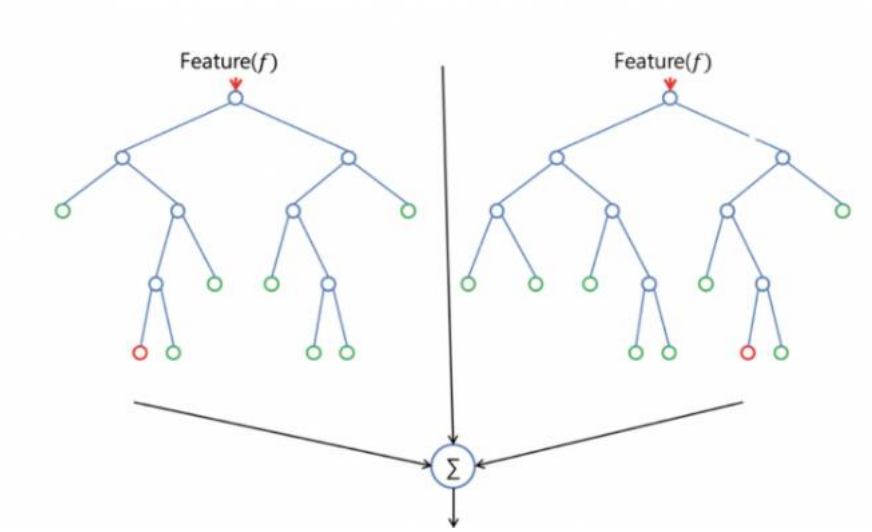
The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.

- Choose the number of clusters k.
- Select k random points from the data as centroids.
- Assign all the points to the closest cluster centroid.
- Recompute the centroids of newly formed clusters.
- Repeat steps 3 and 4.

Random Forest:

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity. It can be used for both classification and regression tasks. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Let's look at random forest in classification, since classification is sometimes considered the building block of machine learning. Below you can see how a random forest would look like with two trees:



XgBoost:

XG Boost is the most widely used algorithm in machine learning, whether the problem is a classification or a regression problem. It is known for its good performance as compared to all other machine learning algorithms.

XG Boost or extreme gradient boosting is one of the well-known gradient boosting techniques (ensemble) having enhanced performance and speed in tree-based (sequential decision trees) machine learning algorithms. It is the most common algorithm used for applied machine learning in competitions and has gained popularity through winning solutions in structured and tabular data. It is open-source software. Earlier only python and R packages were built for XG Boost but now it has extended to Java, Scala, Julia and other languages as well.

XG Boost falls under the category of Boosting techniques in Ensemble Learning. Ensemble learning consists of a collection of predictors which are multiple models to provide better prediction accuracy. In Boosting technique the errors made by previous models are tried to be corrected by succeeding models by adding some weights to the models. Unlike other boosting algorithms where weights of misclassified branches are increased, in Gradient Boosted algorithms the loss function is optimized. XG Boost is an advanced implementation of gradient boosting along with some regularization factors.

Domain: Machine Learning **Technology: Python**

STEPS FOR EXECUTING THE PROJECT

1. Import all the Libraries/packages.
2. Load the Students' Academic Performance dataset.
3. Store then into the cloud based server
4. Load the data from the server as and when required.
5. Perform exploratory data analysis.
6. Preprocess the datasets.
7. Remove the Null values.
8. Check for unbalanced data.
9. Normalize the datasets.
10. Split the dataset.
11. Train all datasets on all classification algorithms mentioned below and record their accuracies.
12. The algorithms are:
    a. Unsupervised:
        i. KMeans Clustering.
    b. Supervised:
        i. Random Forest.
        ii. XgBoost.
13. The best model (XgBoost, in this case) is used for prediction of performance.

SYSTEM DESIGN

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, UML.

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 objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

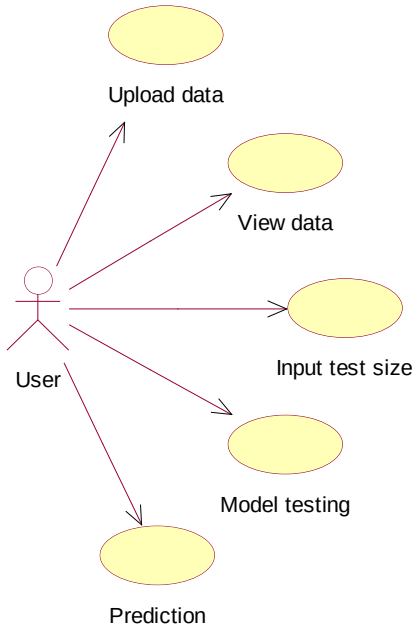
GOALS:

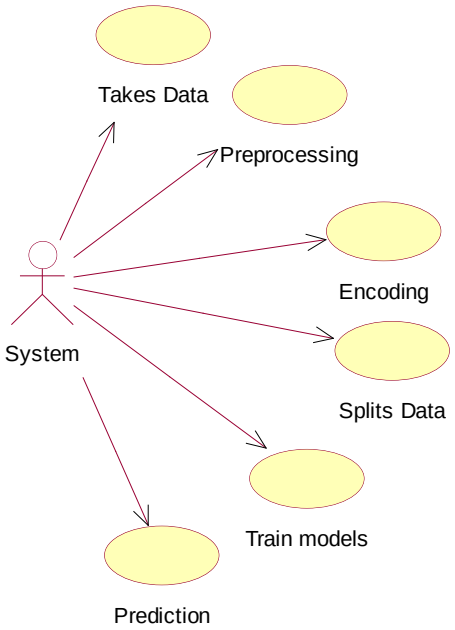
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

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.



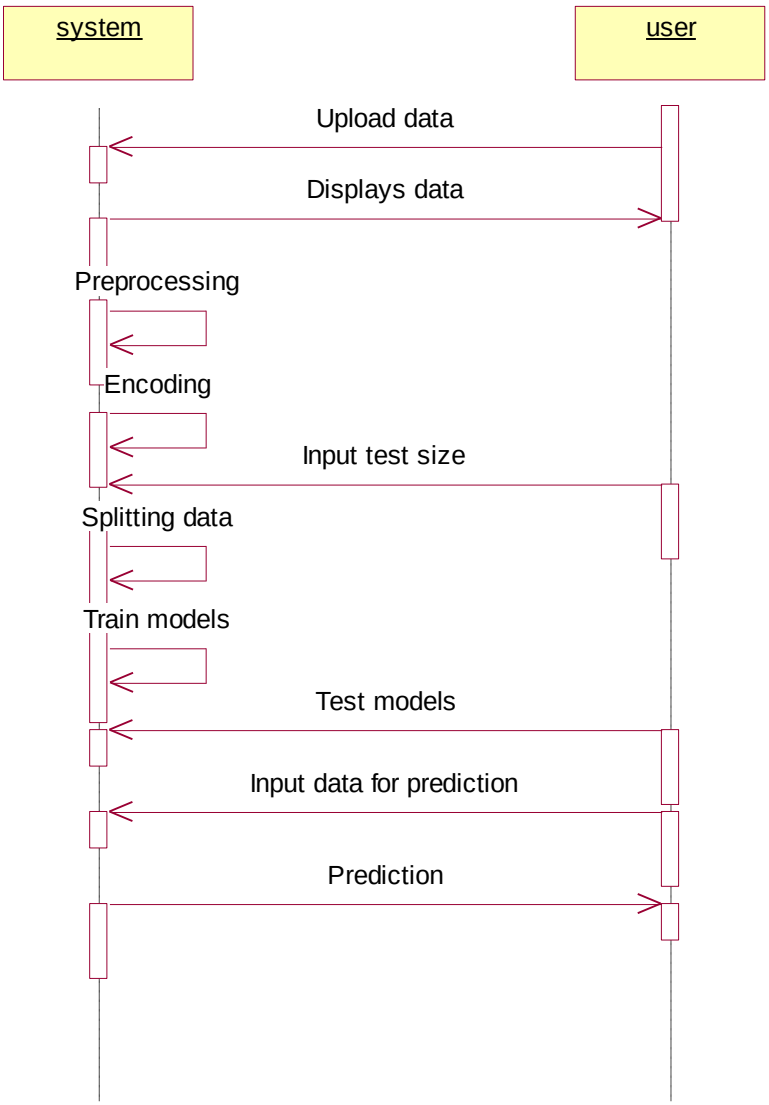


CLASS DIAGRAM:

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.

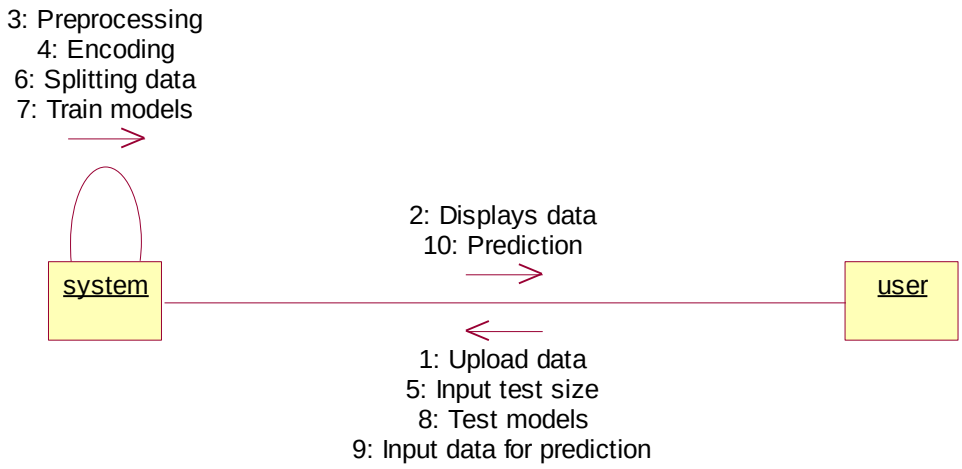
SEQUENCE DIAGRAM:

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.



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 where as the collaboration diagram shows the object organization.

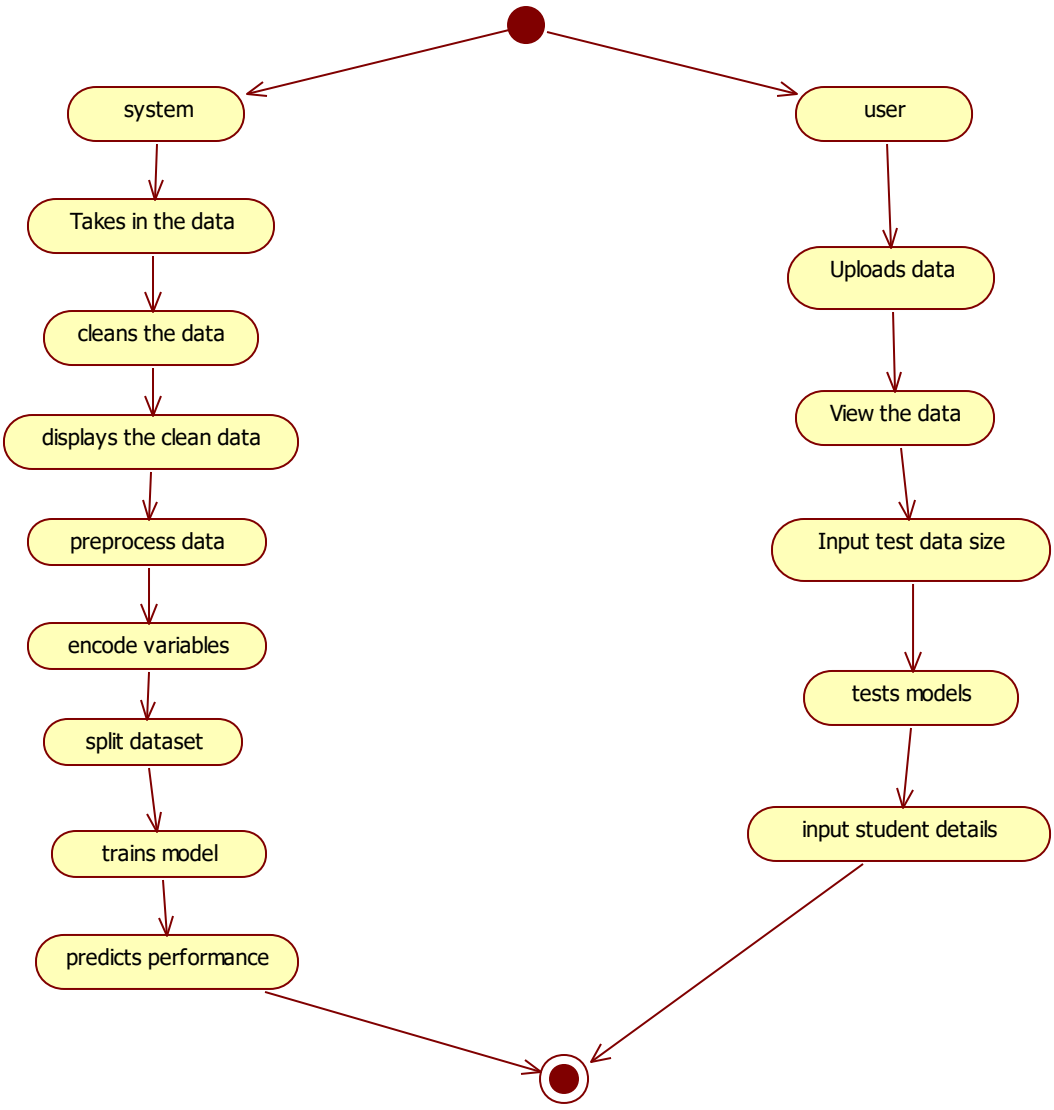


DEPLOYMENT DIAGRAM

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 used to deploy the application.

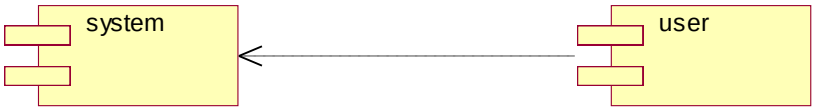
ACTIVITY DIAGRAM:

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.



COMPONENT DIAGRAM

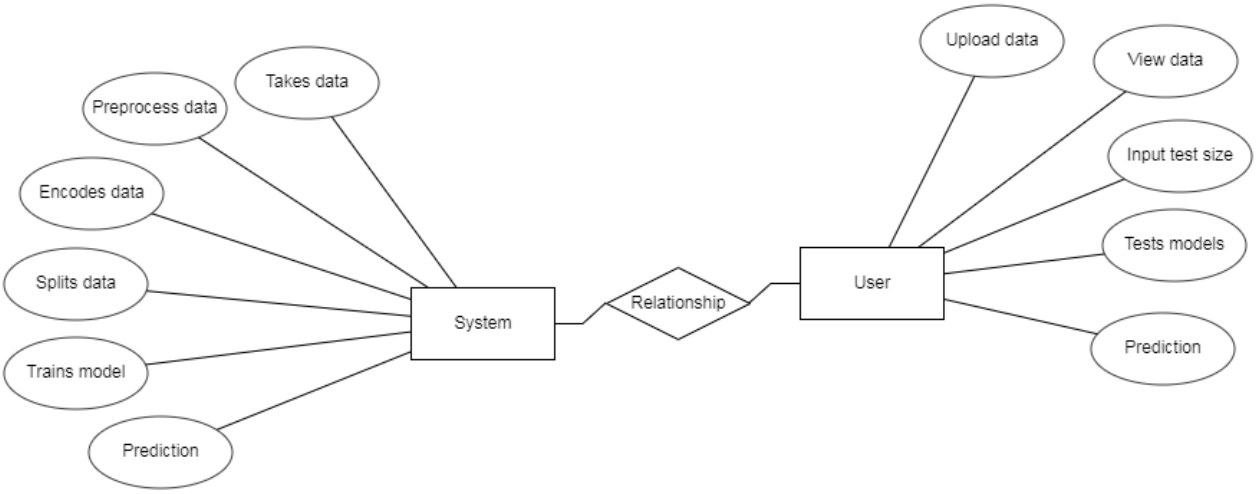
A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned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



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.

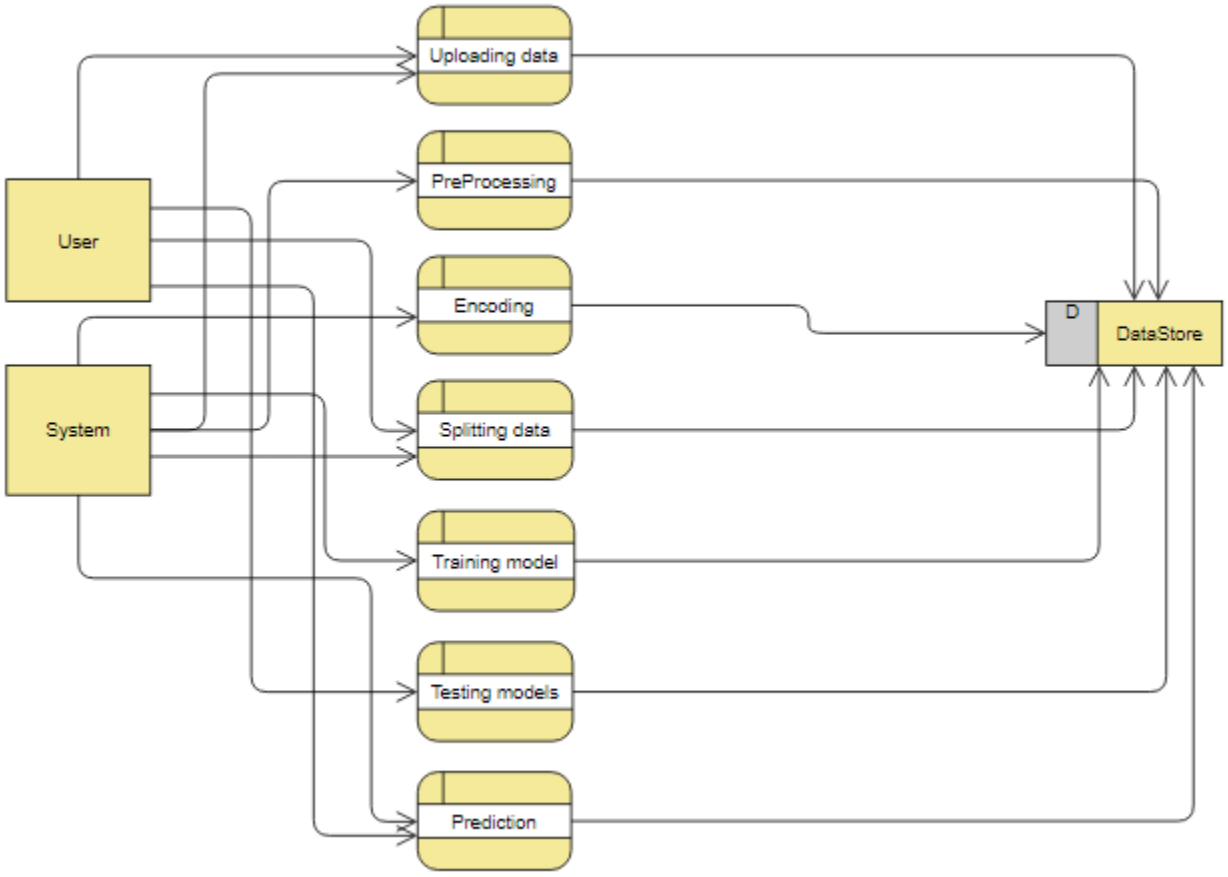
An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.



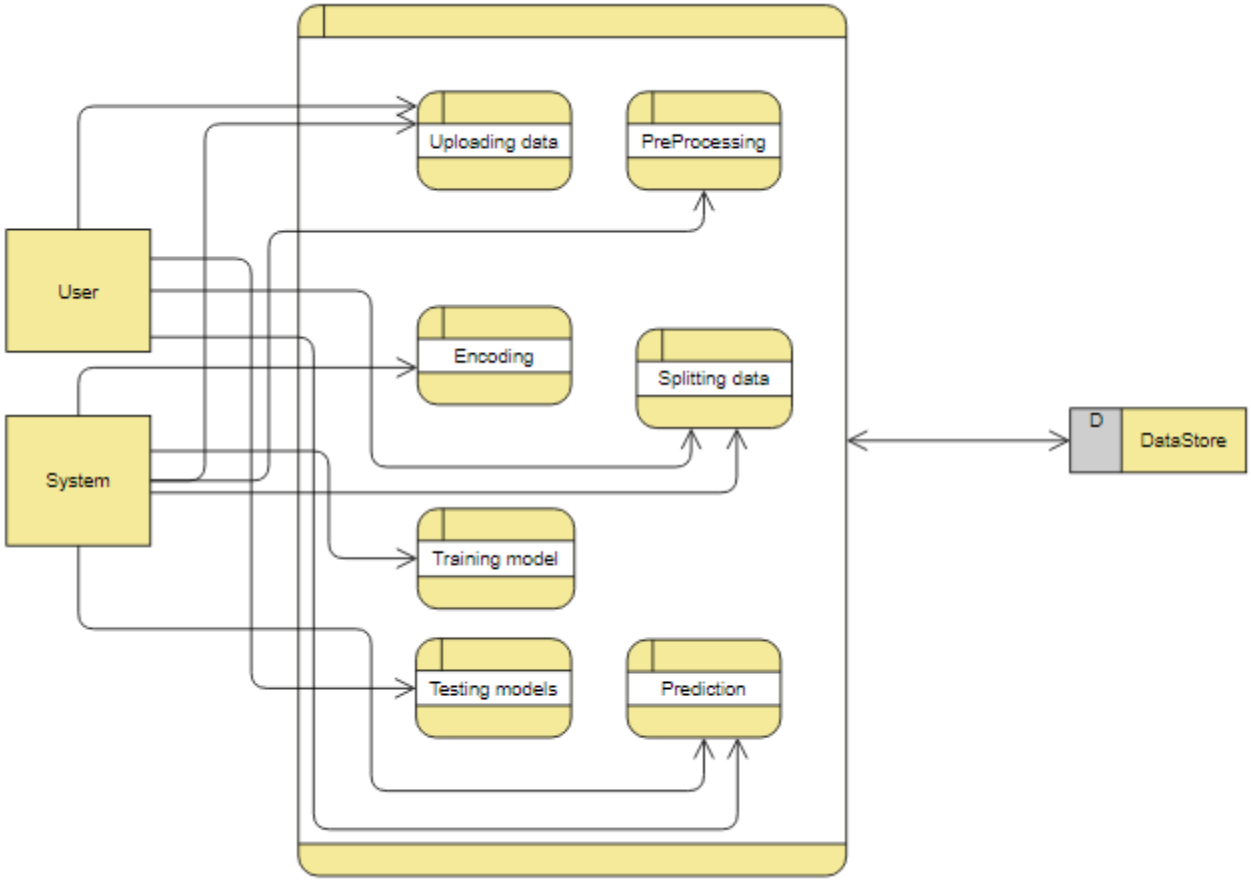
DFD DIAGRAM:

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 how information enters and leaves the system, what changes the information and

where information is stored. The purpose of a DFD is to show the scope and boundaries of 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 a system.



Domain: Machine Learning **Technology: Python**



INTRODUCTION TO PYTHON

Python

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 about any word processor if you want to.

Difference between a script and a program

Script:

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, where as 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)

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.

Python concepts

If you're not interested in thehows 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/ObjC/Java/Fortran

• Easy-ish to interface with C++ (via SWIG)

• 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.

History of Python

Python was developed by Guido van Rossum 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 Rossum still holds a vital role in directing its progress.

Python Features

Python's features include −

- **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-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode** − Python has support for an interactive mode which allows interactive 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 databases.
- **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.

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 a 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.

With a static typed language, you have to decide the memory size the variable can take 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.

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 strings can 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 and accessed using square braces ([]).

Different modes in python

Python has two basic modes: normal and interactive.

The normal mode is the mode where the scripted and finished .py 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

Some Python Libraries:

1. Pandas
2. Numpy
3. Matplotlib
4. Seaborn
5. OpenCV
6. Keras
7. TensorFlow
8. NLTK
9. Scikit-Learn
10. SciPY
11. BeautifulSoup
12. TextBlob
13. Pillow
14. Request
15. SQLAlchamy
16. PyTorch
17. Selenium

Pandas:

- Pandas provide us with many Series and DataFrames. It allows you to easily organize, explore, represent, and manipulate data.
- Smart alignment and indexing featured in Pandas offer you a perfect organization and data labeling.
- Pandas has some special features that allow you to handle missing data or value with a proper measure.
- This package offers you such a clean code that even people with no or basic knowledge of programming can easily work with it.
- It provides a collection of built-in tools that allows you to both read and write data in different web services, data-structure, and databases as well.
- Pandas can support JSON, Excel, CSV, HDF5, and many other formats. In fact, you can merge different databases at a time with Pandas.

Numpy:

- Arrays of Numpy offer modern mathematical implementations on huge amount of data. Numpy makes the execution of these projects much easier and hassle-free.
- Numpy provides masked arrays along with general array objects. It also comes with functionalities such as manipulation of logical shapes, discrete Fourier transform, general linear algebra, and many more.
- While you change the shape of any N-dimensional arrays, Numpy will create new arrays for that and delete the old ones.
- This python package provides useful tools for integration. You can easily integrate Numpy with programming languages such as C, C++, and Fortran code.
- Numpy provides such functionalities that are comparable to MATLAB. They both allow users to get faster with operations.

Matplotlib:

- Matplotlib can create such quality figures that are really good for publication. Figures you create with Matplotlib are available in hardcopy formats across different interactive platforms.
- You can use MatPlotlib with different toolkits such as Python Scripts, IPython Shells, Jupyter Notebook, and many other four graphical user interfaces.
- A number of third-party libraries can be integrated with Matplotlib applications. Such as seaborn, ggplot, and other projection and mapping toolkits such as basemap.
- An active community of developers is dedicated to helping you with any of your inquiries with Matplotlib. Their contribution to Matplotlib is highly praisable.
- Good thing is that you can track any bugs, new patches, and feature requests on the issue tracker page from Github. It is an official page for featuring different issues related to Matplotlib.

Seaborn:

Seaborn is built on top of Python's core visualization library Matplotlib. It is meant to serve as a complement, and not a replacement. However, Seaborn comes with some very important features. Let us see a few of them here. The features help in −

- Built in themes for styling matplotlib graphics
- Visualizing univariate and bivariate data
- Fitting in and visualizing linear regression models
- Plotting statistical time series data
- Seaborn works well with NumPy and Pandas data structures
- It comes with built in themes for styling Matplotlib graphics

In most cases, you will still use Matplotlib for simple plotting. The knowledge of Matplotlib is recommended to tweak Seaborn's default plots.

OpenCV:

- OpenCV is an ideal image processing package that allows you to both read and write images at the same time.
- Computer Vision allows you to rebuild, interrupt, and comprehend a 3D environment from its respective 2D environment.
- This package allows you to diagnose special objects in any videos or images. Objects such as faces, eyes, trees, etc.
- You can also save and capture any moment of a video and also analyze its different properties such as motion, background, etc.
- OpenCV is compatible with many operating systems such as Windows, OS-X, Open BSD, and many others.

NLTK (Natural Language Tool Kit):

- The text processing libraries of NLTK allow classification, tagging, tokenization, stemming, parsing, and semantic reasoning as well.
- NLTK contains a graphical illustration of data science. It also comes with a handbook for guiding through the principles of language processing for NLTK.
- It is open source and contains over fifty corpora and lexical resources such as open multilingual wordnet, question classification, SentiWordNet, SEMCOR, Stopwords Corpus, and many more.
- NLTK also features structure types, structure strings parsing, features different pathways, and re-entrance as well.
- This toolkit comes with a dynamic discussion forum where you can discuss and bring up any issues related to language NLTK.

Scikit-Learn:

- Scikit Learn comes with a clean and neat API. It also provides very useful documentation for beginners.

- It comes with different algorithms – classification, clustering, and regression. It also supports random forests, k-means, gradient boosting, DBSCAN and others
- This package offers easy adaptability. Once you get well with the general functionalities of Scikit Learn, switching to other platforms will be no problem at all.
- Scikit Learn offers easy methods for data representation. Whether you want to present data as a table or matrix, it is all possible with Scikit Learn.
- It allows you to explore through digits that are written in hands. You can not only load but also visualize digits-data as well.

SciPY:

- Scipy contains different modules. These modules are suitable for optimization, integration, linear algebra, and statistics, as well.
- It makes the best use of Numpy arrays for general data structures. In fact, Numpy is an integrated part of Scipy.
- Scipy can handle 1-d polynomials in two ways. Whether you can use poly1d class from numpy or you can use co-efficient arrays to do the job.
- High-level scipy contains not only numpy but also numpy.lib.scimath as well. But it is better to use them from their direct source.
- A supporting community of Scipy is always there to answer your regular questions and solve any issues if aroused.

Python class and objects

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.

The class is the most basic component of object-oriented programming. 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 for me.

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.

Inheritance

First off, classes allow you to modify a program without really making changes to it.

To elaborate, by subclassing 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 subclass inherits from a superclass allowing you to make a generic superclass that is specialized via 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.

Operator Overloads

Operator overloading simply means that objects that you create from classes can respond to actions (operations) that are already defined within Python, such as addition, slicing, printing, etc.

Even though these actions can be implemented via class methods, using overloading ties the behavior closer to Python's object model and 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

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 ValueError exception

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

• Referencing a non−existent variable will raise a NameError exception.

• Mixing datatypes without coercion will raise a TypeError 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/elif 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.

Exceptions also make code management easier: if your programming logic is mixed in with error-handling if statements, it can be difficult to read, modify, and debug your program.

User-Defined Exceptions

I won't spend too much time talking about this, but Python does allow for a programmer to create his own exceptions.

You probably won't have to do this very often but it's nice to have the option when necessary.

However, before making your own exceptions, make sure there isn't one of the built-in exceptions that will work for you.

They have been "tested by fire" over the years and not only work effectively, they have been optimized for performance and are bug-free.

Making your own exceptions involves object-oriented programming, which will be covered in the next chapter

. To make a custom exception, the programmer determines which base exception to use as the class to inherit 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.

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.

Testing 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 to run 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

he 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.

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 and 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 carried out, the function can or cannot return one or more values.

There are three types of functions in python:

help(),min(),print().

Python Namespace

Generally speaking, a **namespace** (sometimes also called a context) is a naming system for making names unique to avoid ambiguity. Everybody knows a namespacing system from daily life, i.e. the naming of people in firstname and familiy name (surname).

An example is a network: each network device (workstation, server, printer, ...) needs a unique name and address. Yet another example is the directory structure of file systems.

The same file name can be used in different directories, the files can be uniquely accessed via the pathnames. Many programming languages use namespaces or contexts for identifiers. An identifier defined in a namespace is associated with that namespace.

This way, the same identifier can be independently defined in multiple namespaces. (Like the same file names in different directories) Programming languages, which support namespaces, may have different rules that determine to which namespace an identifier belongs.

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(), cmp(), ...) and built-in exception names

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.

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 system and/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 callbacks 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 different APIs literally complement each other, there is no reason why you cannot use them both for large projects.

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 templating
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, templating 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.
- Frameworks is a really well done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.
- what is a web framework? is an in-depth explanation of what web frameworks are and their relation to web servers.
- Djangovs 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 data from the database and rendering it in a template. There were no conclusive results but the output is fun to read about nonetheless.

- What web frameworks do you use and why are they awesome? is a language agnostic Reddit 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 Python web 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 (Django or Flask are recommended) and stick with it. 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 resources links on the framework's page.
3. Study open 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.

2. SYSTEM STUDY

2.1 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 understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY

- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

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 the users 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 the system.

6. SYSTEM TESTING

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.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

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

Functional test

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

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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

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

SYSTEMTEST

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.

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.

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. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

6.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

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 and responses must not be delayed.

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.

6.2 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, e.g. 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.

6.3 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:

In this application, we have pre processed the data by removing the null values and encoding all the variables. We used 2 unsupervised and 2 supervised learning methods.

KMeans clustering and Agglomerative clustering are the 2 clustering algorithms which we have used here. Naïve Bayes and Decision Tree are the 2 supervised algorithms used for actual classification of the students performance.

The best model was the Decision Tree model with Hyper parameters tuning. Clustering algorithms cannot be explicitly used for classification. But, we can use them in conjunction with supervised techniques to be used for prediction. The dataset used was the Students' Academic Performance data on kaggle.

The accuracies for all models are shown below:

Model	Accuracy
KMeans Clustering	0.583333333
Agglomerative Clustering	0.485416667
Naïve Bayes	0.763888889
Decision Tree (untuned)	0.666666667
decision Tree (tuned)	0.777777778
Naïve Bayes + Kmeans Clustering	0.666666667

FUTURE SCOPE:

We should consider students performance prediction using ensemble techniques like random forest and other boosting and bagging techniques. We may also model a neural network which are high in complexities but offers high accuracy and automation of feature selection.

REFERENCES:

[1] Vairachilai S, Vamshidharreddy, "Student's Academic Performance Prediction Using Machine Learning Approach", IJAST, vol. 29, no. 9s, pp. 6731 - 6737, Jun. 2020.

[2] Kaviani, Pouria & Dhotre, Sunita. (2017). Short Survey on Naive Bayes Algorithm. International Journal of Advance Research in Computer Science and Management. 04.

[3] Sharma, Himani & Kumar, Sunil. (2016). A Survey on Decision Tree Algorithms of Classification in Data Mining. International Journal of Science and Research (IJSR). 5

[4] Altaher A, BaRukab O (2017) Prediction of student's academic performance based on adaptive Neuro-fuzzy inference. IJCSNS 17: 165-169.

[5] Kaushik, Manju & Mathur, Bhawana. (2014). Comparative Study of K-Means and Hierarchical Clustering Techniques. International journal of Software and Hardware Research in Engineering. 2. 93-98.

[6] Kabakchieva D (2012) Student performance prediction by using data mining classification algorithms. IJCSMR 1: 686-690.

[7] Baker, Ryan SJD, Yacef K (2009) The state of educational data mining in 2009: A review and future visions. JEDM 1: 3-16.

[8]. Ramesh V, Parkavi P, Ramar K (2013) Predicting student performance: A statistical and data mining approach. IJCA 63: 35-39

[9] R. R. Kabra,R. S. Bichkar, "Performance Prediction of Engineering Students using Decision Trees", International Journal of Computer Applications (0975 – 8887), Volume 36– No.11, December 2011

[10] Ajay Kumar Pal, Saurabh Pal, "Data Mining Techniques in EDM for Predicting the Performance of Students", International Journal of Computer and Information Technology (ISSN: 2279 – 0764),Volume 02– Issue 06, November 2013

[11]Agavanakis, Kyriakos & Karpetas, George & Taylor, Michael & Pappa, Evangelia & Michail, Christos & Filos, John & Trachana, Varvara & Kontopoulou, Lamprini. (2019). Practical machine learning based on cloud computing resources.

[12] Santhanam, Ramraj & Uzir, Nishant & Raman, Sunil & Banerjee, Shatadeep. (2017). Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets.