CHAPTER-1
INTRODUCTION

1.INTRODUCTION

1.1 MOTIVATION

Detecting fraud in banking transactions using machine learning is motivated by the need to prevent financial losses, maintain customer trust, comply with regulations, and counterthe rising sophistication of fraudsters. Advanced technologies enable the analysis of vastdatasets to identify patterns, enhancing the security of financial transactions.

1.2 PROBLEM DEFINITION

The problem to be addressed in the Fraud Detection in Banking Transactions using Machine Learning project is the identification and prevention of fraudulent activities within financial transactions. The goal is to develop a robust and accurate system that cananalyze transactional data, detect patterns indicative of fraud, and take proactive measuresto mitigate risks, safeguarding both financial institutions and their customers.

1.3 OBJECTIVE OF PROJECT

The project aims to develop an advanced Fraud Detection system for Banking Transactions using Machine Learning. The primary objective is to design and implement models capable of analyzing transactional data, identifying patterns indicative of fraud, and enabling real-time monitoring. By assigning risk scores to transactions, the system aims to proactively detect and prevent fraudulent activities. Seamless integration with existing banking systems is a key focus, ensuring practical implementation. Continuous evaluation and improvement are crucial components, allowing the system to adapt to evolving fraud tactics and enhance overall accuracy. The ultimate goal is to provide financial institutions with a robust tool to safeguard against financial losses, maintain customer trust, and meet regulatory compliance standards.

CHAPTER-2
LITERATURE SURVEY

1. LITERATURE SURVEY

Statistical methods can be used for fraud detection. Here the statistical distribution of the dataset is analyzed for anomalous behavior of the fraudulent by using Linear DiscriminantAnalysis and Logistic regression [1]. The authors used a variety of data mining techniquesin real-time fraud detection using historical data [1].

The research work [2] describes the methods to detect fraud by using KNN algorithm and outlier finding mechanism. The model helps in the detection of malicious behavior of the fraudulent.

The authors in [3] used an ensemble technique including the Random Forest model to analyze the normal transactions and compare the performance of the fraudulent transaction detection method by neural networks.

Fraud detection in [4] presented the method for credit card transactions and analyzed thedata using Wale-algorithm optimized backpropagation. The authors in [4][6] have analyzed already classified results for detecting credit card fraud using an imbalanced dataset. K means clustering is used for sampling groups of fraudulent transaction samples. Authors also used genetic algorithms for group fraudulent transactions.

The researcher used multiple machine learning algorithms such as KNN, Logistic regression, and Naïve Bayes for analyzing the available dataset. An enhanced study of this has demonstrated, the represents that KNN outperforms the other two methods [2][6]. The performance was assessed by precision, recall, Mathew correlation coefficient, and balanced classification rate specificity.

A unique fraud detection technique is proposed in [7] using Big data technology with a new method known as Scalable Real-time Fraud Finder (SCARFF) using different data analysis tools such as Spark, Cassandra, and Kafka. Real-time data analysis is possible with a large amount of transaction data. The advantage of this system proposed holds higher accuracy, fault tolerance, and scalability. In [8] the researcher presented a featureengineering method to minimize the number of false positive rates, that are normally usedin the anomaly-detecting algorithm [16].

CHAPTER-3

SYSTEM ANALYSIS

3.SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

The current banking systems lack effective fraud detection algorithms, exposing vulnerabilities to financial losses and reputational damage. Without advanced mechanisms, slo+w check verification processes contribute to counterfeits. This paper proposes an AI-based model to expedite verification and minimize fraud impact, emphasizing the need for system upgrades and improved accuracy.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM:

- Machine learning-based systems may generate false alarms, causing inconvenience for legitimate transactions.
- Analyzing sensitive customer data raises privacy issues.
- Fraudsters evolve tactics, and models may struggle to adapt rapidly, leading to detection gaps

3.2 PROPOSED SYSTEM:

The proposed system leverages a hybrid approach incorporating Random Forest, K- Nearest Neighbors (KNN), and Logistic Regression algorithms for robust fraud detectionin banking transactions. This ensemble model combines the strengths of these techniquesto enhance accuracy and adaptability, offering a comprehensive solution to counteract evolving fraudulent activities.

3.2.1ADVANTAGES OF PROPOSED SYSTEM:

1.Hybrid model (Random Forest, KNN, Logistic Regression) boosts accuracy.2.Adapts to evolving tactics for dynamic fraud detection.

3.Comprehensive solution for banking transactions

CHAPTER-4

SYSTEM REQUIREMENTS

4 SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS:

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

SOFTWARE REQUIREMENTS:

- **Operating System:** Windows
- **Coding Language:** Python
- **Front-End :** Python
- **Back-End :** Flask

4.1 SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is putforth 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 ensurethat 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 anddevelopment 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 demandon 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

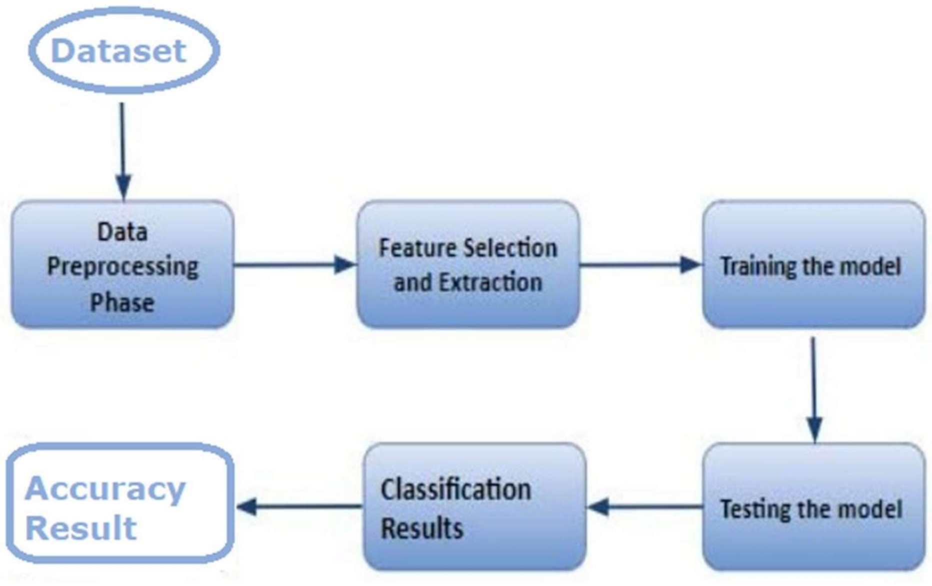
The aspect of study is to check the level of acceptance of the system by the user.This includes the process of training the user to use the system efficiently. The user mustnot feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on 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, ashe is the final user of the system.

CHAPTER-5

SYSTEM DESIGN

5. SYSTEM DESIGN

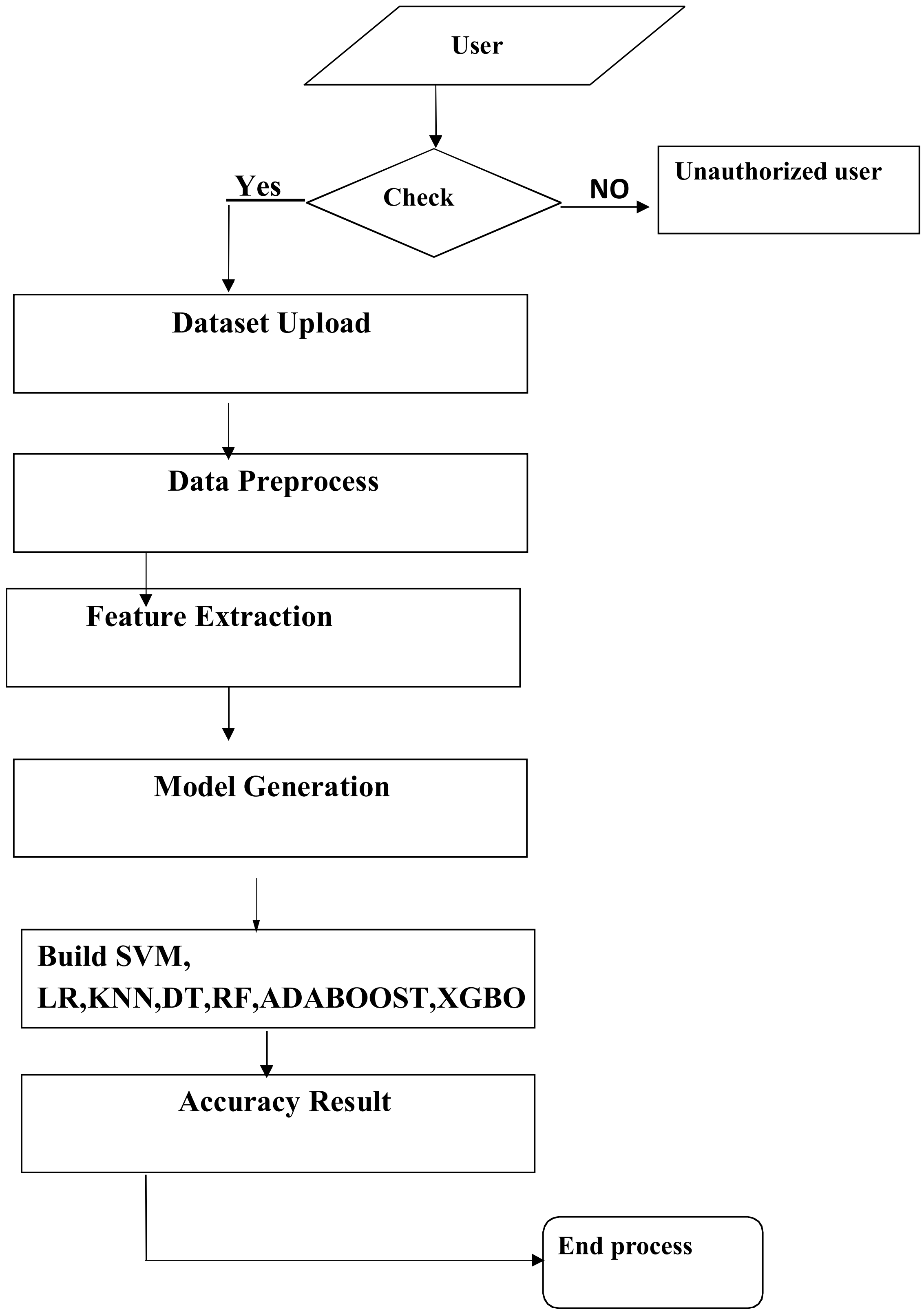
5.1 SYSTEM ARCHITECTURE:



5.2 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be usedto represent a system in terms of input data to the system, various processing carried outon this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flowsin the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and thetransformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

USER:


User
Check
Yes
NO
Unauthorized user
Dataset Upload
Data Preprocess
Feature Extraction
Model Generation
Build SVM,
LR,KNN,DT,RF,ADABOOST,XGBO
Accuracy Result
End process

5.3 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 processmay 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 asfor 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.

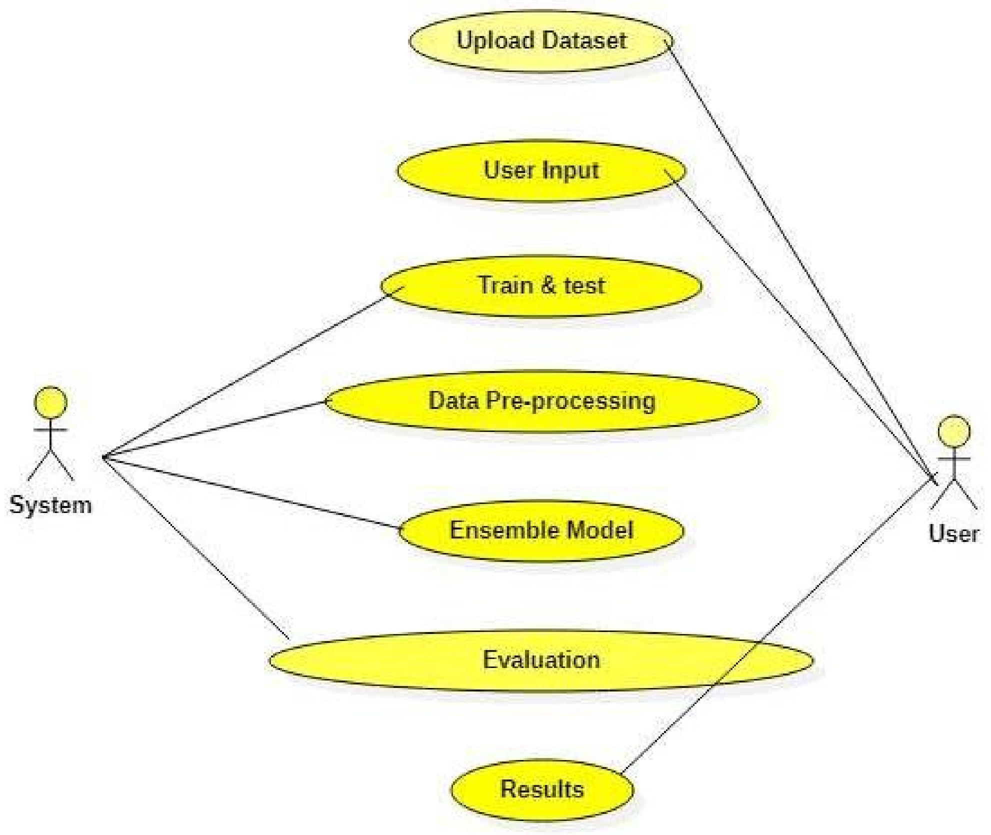
GOALS:

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 candevelop 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, patternsand components.
7. Integrate best practices.

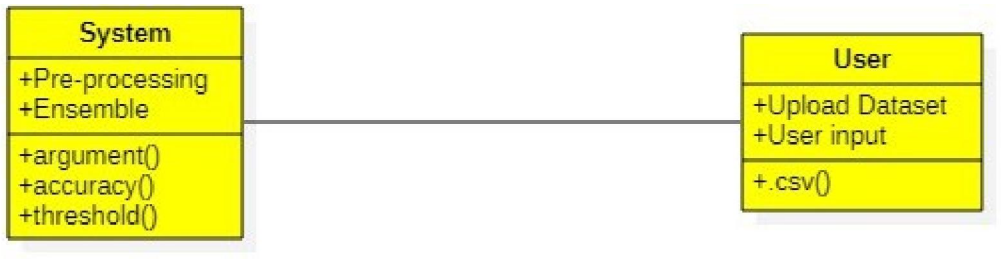
5.3.1 Use case diagram:

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. Themain 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.



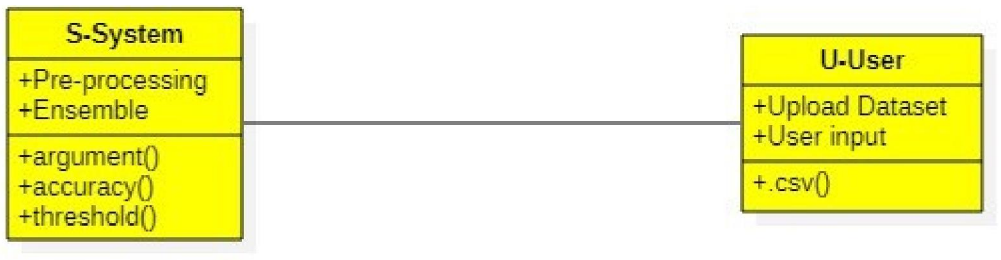
5.3.2 Class diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram maybe capable of providing certain functionalities. These functionalities provided by the classare termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



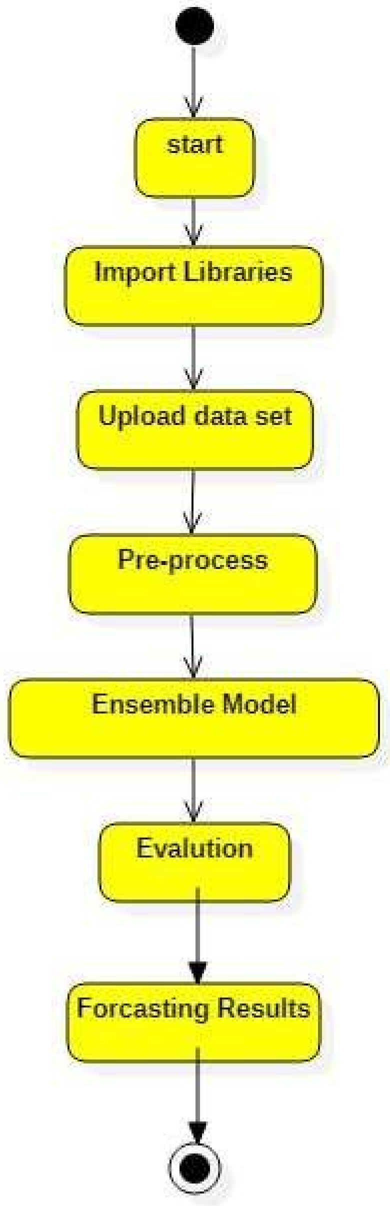
5.3.3 Object diagram:

The object diagram is a special kind of class diagram. An object is an instance ofa class. This essentially means that an object represents the state of a class at a given pointof time while the system is running. The object diagram captures the state of different classes in the system and their relationships or associations at a given point of time.



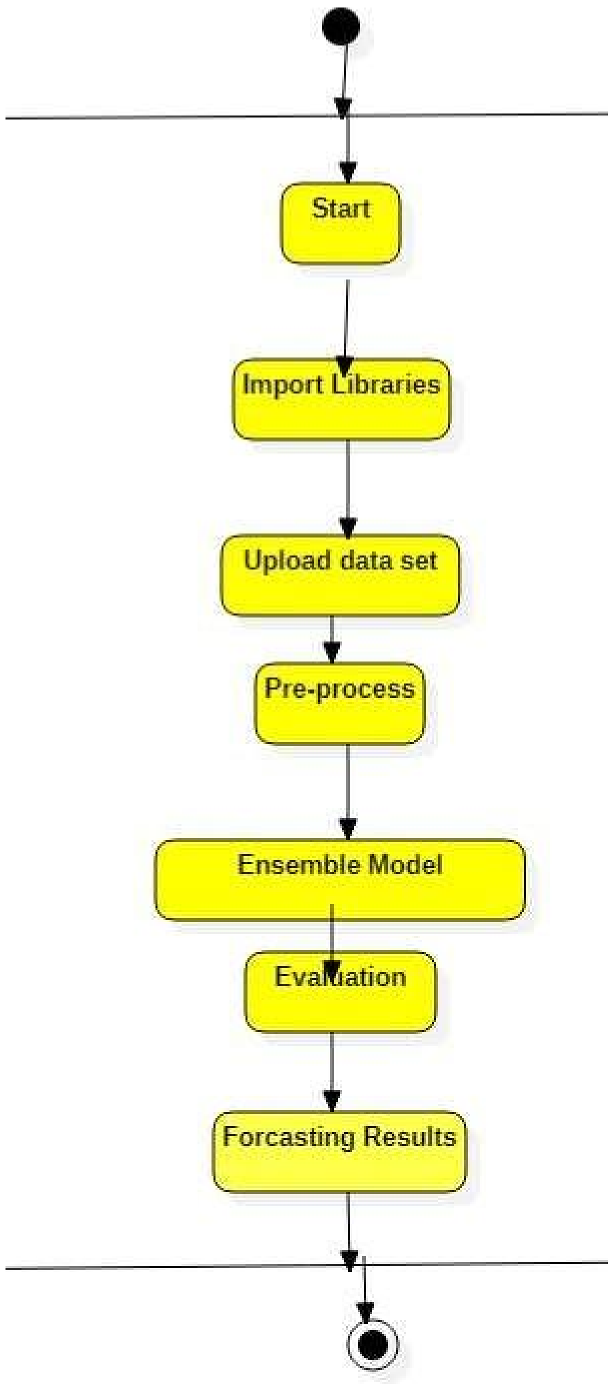
5.3.4 State diagram:

A state diagram, as the name suggests, represents the different states that objectsin the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.



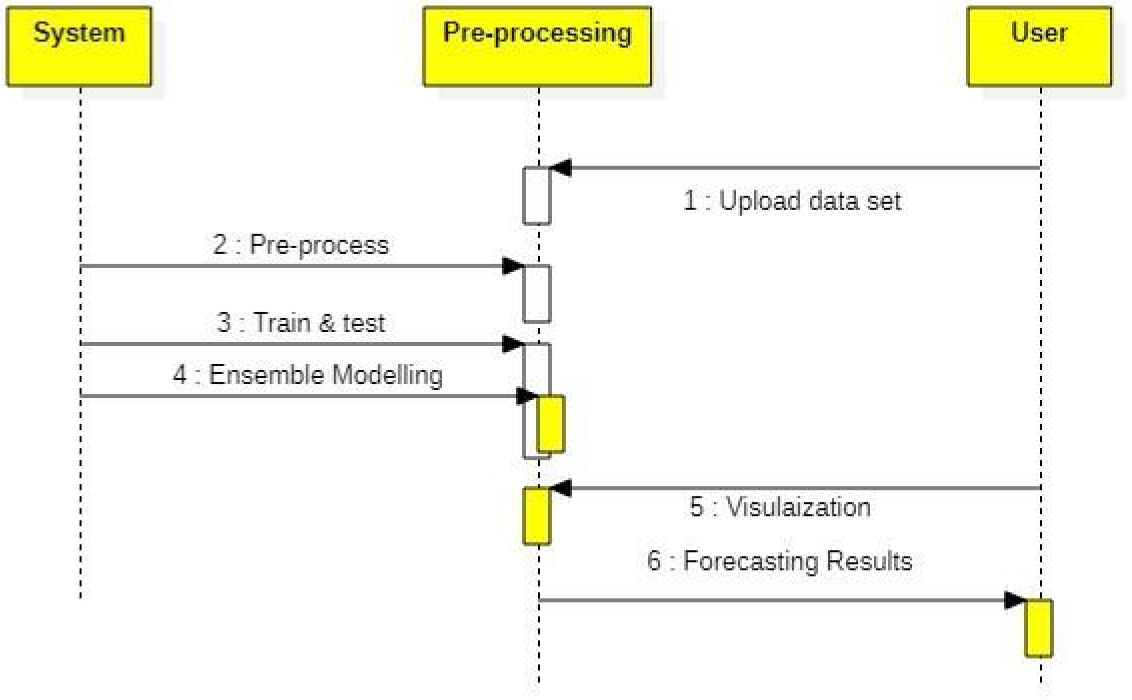
5.3.5 Activity diagram:

The process flows in the system are captured in the activity diagram. Similar to astate diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.



5.3.6 Sequence diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step.Different objects in the sequence diagram interact with each other by passing "messages".



5.4 IMPLEMENTATION:

5.4.1 MODULES:

Data Collection:

The data collection module serves as the foundational step, acquiring relevant informationfor fraud detection. It involves retrieving transactional data from diverse sources, such asbanking records and logs. Ensuring data quality and comprehensiveness is crucial for thesubsequent stages, establishing a reliable foundation for accurate fraud analysis.

Data Preprocessing:

This module handles dataset preparation, addressing missing data, outliers, and encoding categorical variables. It ensures data uniformity by normalizing and scaling features for effective analysis in subsequent stages.

SMOTE (Synthetic Minority Over-sampling Technique):

SMOTE is employed to counter dataset imbalance, generating synthetic samples for the minority class. This enhances the dataset's representation and improves algorithm performance in detecting fraud.

Ensemble Model Building:

The ensemble model module combines outputs from individual algorithms, utilizing a boosting method. This integration enhances overall model performance and adaptability.

Model Training and Evaluation:

The module involves dividing the dataset, training the model on a designated set, and evaluating its performance using key metrics such as accuracy, precision, recall, and F1 score.

5.4.2 CODING:

FRAUD_ANALYSIS:

```
# Importing Libraries

import numpy as np import pandas as pd
import matplotlib.pyplot as pltimport seaborn as sns
%matplotlib inline

df = pd.read_csv(r'C:\Users\Sunjeevi\PycharmProjects\FDUBML\venv\FraudDetectio
nBankTransaction\Dataset\Dataset.csv')

# Identifying any null valuesdf.isnull().sum().sum() df.info()
df.describe() sns.heatmap(df.corr(), annot=True)
sns.catplot(x="isFraud", y="amount", col="type", data=df) sns.lmplot(x="oldbalanceOrg",
y="amount", hue="isFraud", col="type",data=df)
sns.lmplot(x="newbalanceOrig", y="amount", hue="isFraud", col="type",data=df)

sns.lmplot(x="oldbalanceDest", y="amount", hue="isFraud", col="type",data=df)
sns.lmplot(x="newbalanceDest", y="amount", hue="isFraud", col="type",data=df)
df[df['isFraud'] == 0]['amount'].nlargest(10)
df[df['isFraud'] == 1]['amount'].nlargest(10)
fraud = df[df['isFraud'] == 1]['amount'].max() not_fraud = df[df['isFraud'] == 0]['amount'].max()
print("Highest amount for fraud transaction:", fraud)
print("Highest amount for noraml transaction:", not_fraud) print("\nHigest Fraud Transaction is", fraud
- not_fraud, "times morethan the highest normal transaction")
df[df['amount'] == fraud] df[df['amount'] == not_fraud]
```

FRAUD_MODELS:

```
# Importing Analysis Librariesimport numpy as np
import pandas as pd
import matplotlib.pyplot as pltimport seaborn as sns
%matplotlib inline
# for saving the modelimport pickle
# Importing ML Libraries
from sklearn.model_selection import train_test_splitfrom sklearn import metrics
from sklearn.linear_model import LogisticRegressionfrom sklearn.neighbors import
KNeighborsClassifierfrom sklearn.ensemble import RandomForestClassifierfrom sklearn import svm
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_selection import mutual_info_regressionfrom sklearn.metrics import
mean_squared_error, r2_score
from sklearn.metrics import confusion_matrix, classification_report,accuracy_score,
mean_absolute_error
df = pd.read_csv(r'C:\Users\Sunjeevi\PycharmProjects\FDUBML\venv\FraudDetectio
nBankTransaction\Dataset\Dataset.csv')
df.head(5)
# Label encoding for 'type' column
from sklearn.preprocessing import LabelEncoderl = LabelEncoder()
df['type_code'] = l.fit_transform(df['type'])


# Dropping isFlaggedFraud since it is false in all rows and also dropping'type' since we now have
type_code
X = df.drop(['isFlaggedFraud', 'isFraud', 'type', 'nameOrig','nameDest'], axis=1)
y = df['isFraud']
```

```
def make_mi_scores(X, y):
        mi_scores = mutual_info_regression(X, y)
        mi_scores = pd.Series(mi_scores, name="MI Scores", index=X.columns)mi_scores =
        mi_scores.sort_values(ascending=False)
        return mi_scores
mi_scores = make_mi_scores(X, y)mi_scores[:10]
def plot_mi_scores(scores):
        scores = scores.sort_values(ascending=True)width = np.arange(len(scores))
        ticks = list(scores.index) plt.barh(width, scores) plt.yticks(width, ticks) plt.title("Mutual
        Information Scores")
plt.figure(dpi=100, figsize=(10, 18))plot_mi_scores(mi_scores)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,random_state=123)
print('Training dataset shape:', X_train.shape, y_train.shape)print('Testing  dataset  shape:',
X_test.shape,  y_test.shape) lr = LogisticRegression()
lr.fit(X_train, y_train) lr_pred = lr.predict(X_test)
print('MAE:', metrics.mean_absolute_error(y_test, lr_pred)) print('MSE:',
metrics.mean_squared_error(y_test, lr_pred)) print('RMSE:',
np.sqrt(metrics.mean_squared_error(y_test, lr_pred)))print(classification_report(y_test, lr_pred))
knc = KNeighborsClassifier(n_neighbors=3)knc.fit(X, y)
knc_pred = knc.predict(X_test)
print('MAE:', metrics.mean_absolute_error(y_test, knc_pred)) print('MSE:',
metrics.mean_squared_error(y_test, knc_pred)) print('RMSE:',
np.sqrt(metrics.mean_squared_error(y_test, knc_pred)))print(classification_report(y_test, knc_pred))
rf = RandomForestClassifier()rf.fit(X_train, y_train) rf_pred = rf.predict(X_test)
print('MAE:', metrics.mean_absolute_error(y_test, rf_pred)) print('MSE:',
metrics.mean_squared_error(y_test, rf_pred)) print('RMSE:',
np.sqrt(metrics.mean_squared_error(y_test, rf_pred)))
```

5.5 SOFTWARE ENVIRONMENT

What is Python:-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programminglanguage.

Python allows programming in Object-Oriented and Procedural paradigms. Pythonprograms generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google,Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which canbe used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python :-

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have towrite the complete code for that manually.

2. Extensible

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

3. Embeddable

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

4. Improved Productivity

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

5. IOT Opportunities

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

6. Simple and Easy

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

7. Readable

Because it is not such a verbose language, reading Python is much like reading English.
This is the reason why it is so easy to learn, understand, and code. It also doesnot need curly

braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

8. Free and Open-Source

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

Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

9. Interpreted

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

Any doubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is donein other languages. Python also has an awesome standard library support, so you don'thave to search for any third-party libraries to get your job done. This is the reason thatmany people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leveragethe free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in themost popular programming language category.

3. Python is for Everyone

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

Disadvantages of Python

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

1. Speed Limitations

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

2. Weak in Mobile Computing and Browsers

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

3. Design Restrictions

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

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python's database access
layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

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

History of Python : -

What do the alphabet and the programming language Python have in common? Right,both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in theNetherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatestachievement of ABC was to influence the design of Python.Python was conceptualizedin the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido vanRossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "Iremembered all my experience and some of my frustration with ABC. I decided to tryto design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC partsthat I liked. I created a basic syntax, used indentation for statement grouping instead ofcurly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start bylooking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can oftenbe misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learningmethods, it's more helpful to think of machine learning as a means of building modelsof data.

Fundamentally, machine learning involves building mathematical models to helpunderstand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previouslyseen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to whichthis type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essentialto using these tools effectively, and so we will start with some broad categorizations ofthe types of approaches we'll discuss here.

Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined,it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference toany label, and is often described as "letting the dataset speak for itself." These models include tasks such as *clustering* and *dimensionality reduction.* Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earthbecause they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Thenthe question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from datato perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, butother aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way togo. The reason behind is that ML has not been able to overcome number of challenges.The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessingand feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features ofdata points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to bedeployed in real life.

Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researcherswe are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real- world applications of ML

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention

Recommendation of products to customer in online shopping

How to Start Learning Machine Learning?

Arthur Samuel coined the term **"Machine Learning"** in 1959 and defined it as a **"Field of study that gives computers the capability to learn without beingexplicitly programmed".**

And that was the beginning of Machine Learning! In modern times, Machine Learningis one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a *344%* growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to startlearning it? So this article deals with the Basics of Machine Learning and also the pathyou can follow to eventually become a full-fledged Machine Learning Engineer. Nowlet's get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talentedMachine Learning Engineer. Of course, you can always modify the steps according toyour needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, MultivariateCalculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if youwant to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using variousonline resources and courses such as **Fork Python** available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML(Which is the fun part!!!) It's best to start with the basics and then move on to the morecomplicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels),so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning –** This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning –** This involves learning optimal actions through trial and error.

Advantages of Machine learning :-

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce websitelike Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It usesthe results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve thealgorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional andmulti-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where itdoes apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning :-

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, ittakes quite some time to recognize the source of the issue, and even longer to correct it.

Python Development Steps: -

Guido Van Rossum published the first version of Python code (version 0.9.0) atalt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object orientedand had module system.Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced.

This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.xbefore the next major release as Python 3.0 (also known as "Python 3000" and "Py3K")was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists

The rules for ordering comparisons have been simplified. E.g. a heterogeneous listcannot be sorted, because all the elements of a list must be comparable to each other.

- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to havethe "old" behaviour.
- Text Vs. Data Instead Of Unicode Vs. 8-bit

Purpose:-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- **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 also acknowledges that speed of development is important. Readable and tersecode is part of this, and so is access to powerful constructs that avoid tedious repetitionof code. Maintainability also ties into this may be an all but useless metric, but it doessay something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Modules Used in Project :-

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used

for machine learning applications such as neural networks. It is used for both researchand production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It wasreleased under the Apache 2.0 open-source license on November 9, 2015.

Numpy

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

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

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

Pandas

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

Matplotlib

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

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

Scikit – learn

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

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- 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 also acknowledges that speed of development is important. Readable and tersecode is part of this, and so is access to powerful constructs that avoid tedious repetitionof code. Maintainability also ties into this may be an all but useless metric, but it doessay something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac:

Python a versatile programming language doesn't come pre-installed on your computerdevices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability withits notable use of great whitespace.
The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not comepre-packaged with Windows.

How to Install Python on Windows and Mac:

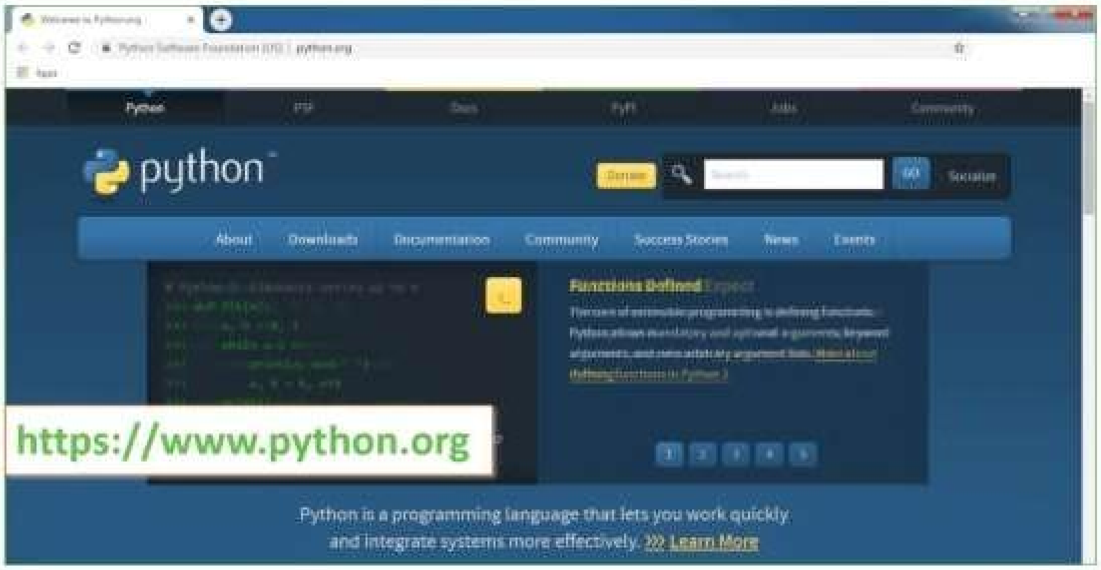
There have been several updates in the Python version over the years. The question is howto install Python? It might be confusing for the beginner who is willing to start learningPython but this tutorial will solve your query. The latest or the newest version of Pythonis version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know aboutyour **System Requirements**. Based on your system type i.e. operating system and basedprocessor, you must download the python version. My system type is a **Windows 64-bitoperating system**. So the steps below are to install python version 3.7.4 on Windows 7device or to install Python 3. Download the Python Cheat sheet here.The steps on how toinstall Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

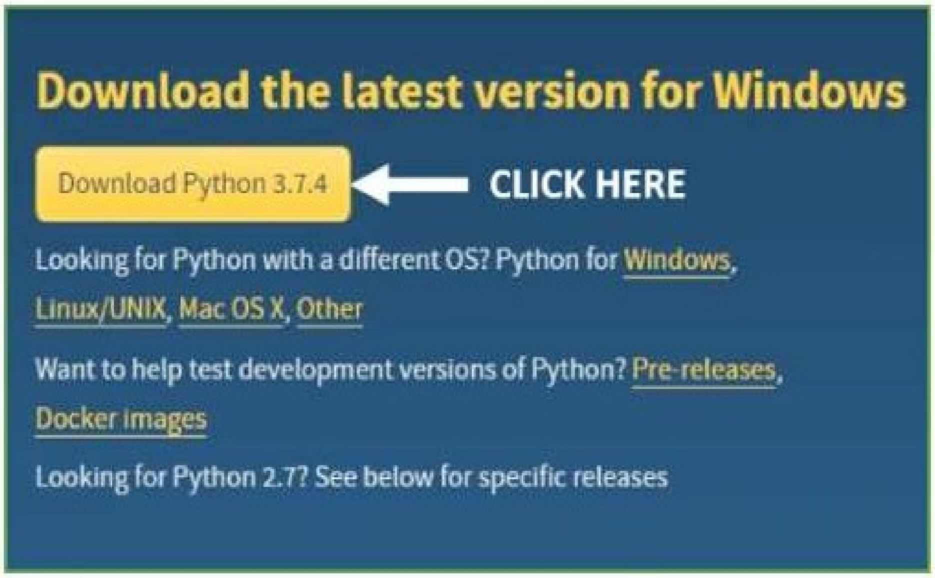
Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or anyother web browser. OR Click on the following link: **https://www.python.org**


python
About
Downloads
Documentation
Community
Success Stories
News
Events
Functions Defined
https://www.python.org
Python is a programming language that lets you work quickly
and integrate systems more effectively. >>> Learn More

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

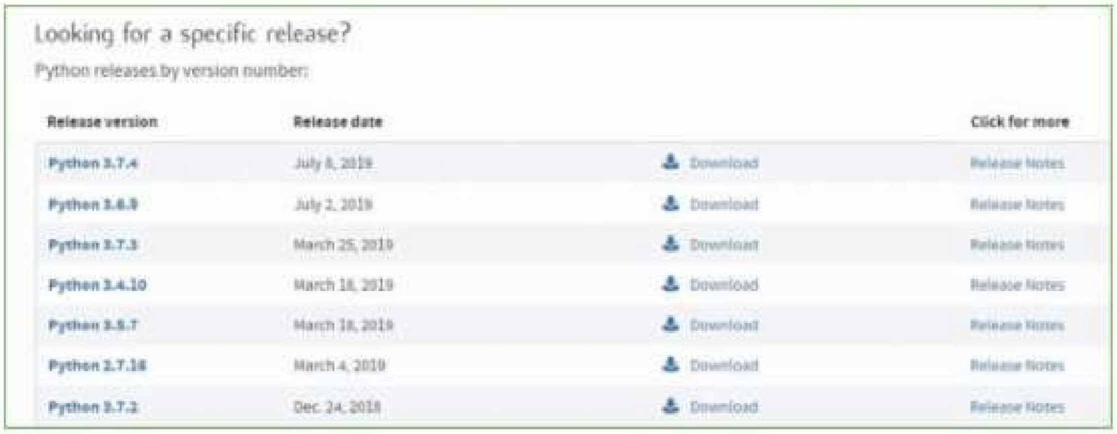

Download the latest version for Windows

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

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

Step 3: You can either select the Download Python for windows 3.7.4 button in YellowColor or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

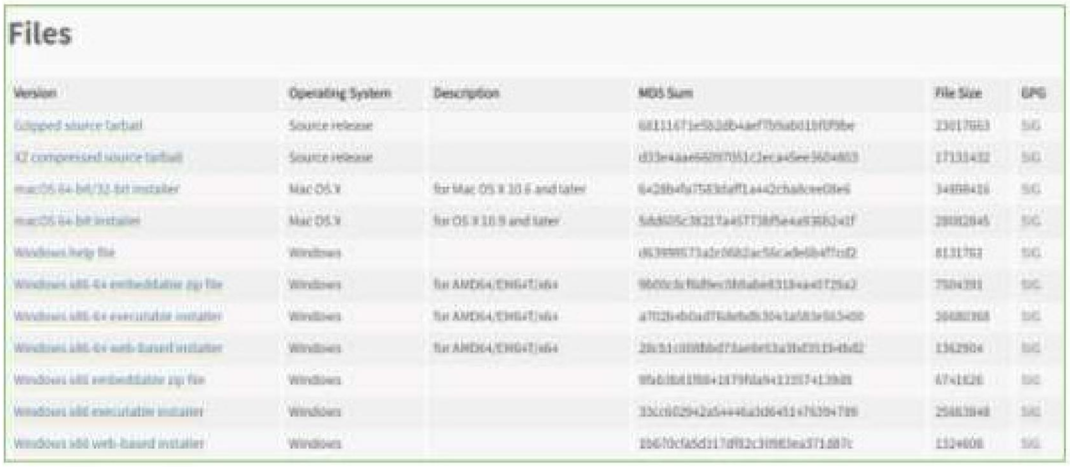

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		[illegible]	23017663	SIG
XZ compressed source tarball	Source release		[illegible]	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	[illegible]	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	[illegible]	28082845	SIG
Windows help file	Windows		[illegible]	8131763	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	[illegible]	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	[illegible]	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	[illegible]	1362904	SIG
Windows x86 embeddable zip file	Windows		[illegible]	6741626	SIG
Windows x86 executable installer	Windows		[illegible]	25663848	SIG
Windows x86 web-based installer	Windows		[illegible]	1324608	SIG

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86web-based installer.

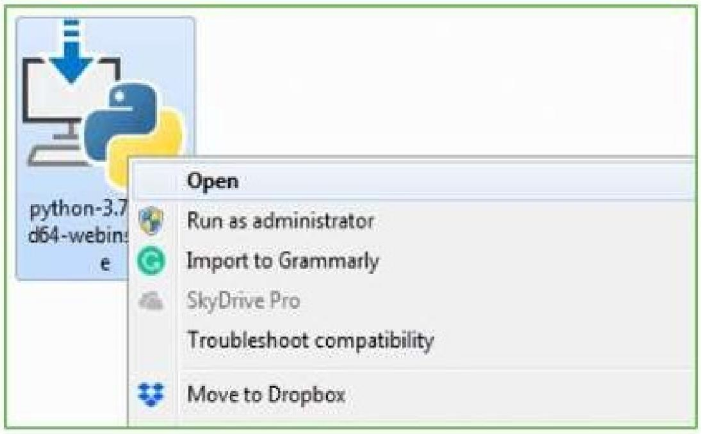
•To download Windows 64-bit python, you can select any one from the three options:Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regardingwhich version of python is to be downloaded is completed. Now we move ahead with thesecond part in installing python i.e. Installation

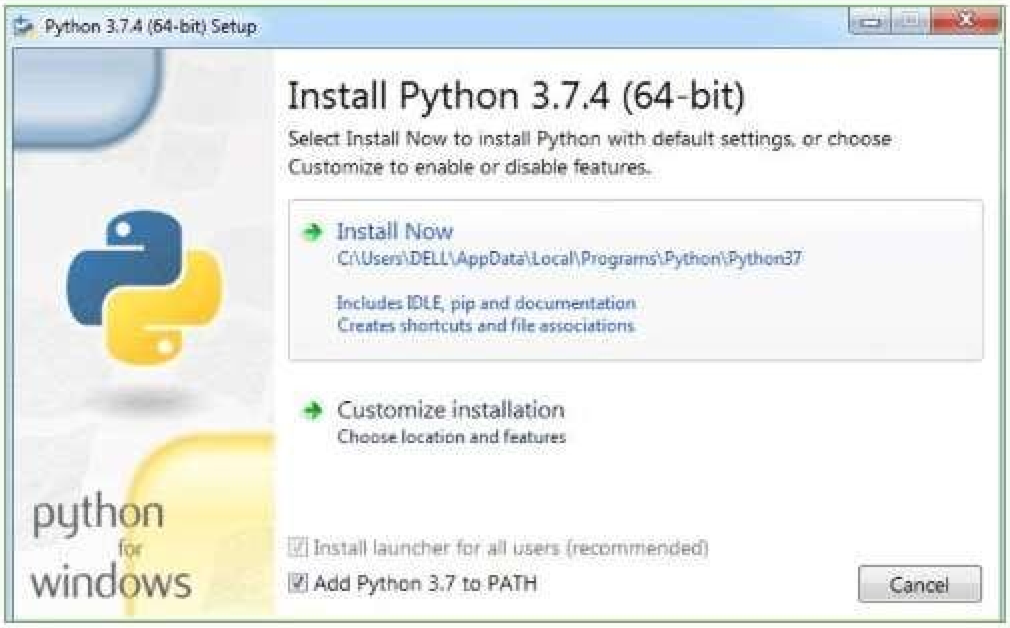
Note: To know the changes or updates that are made in the version you can click on theRelease Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 toPATH.


Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

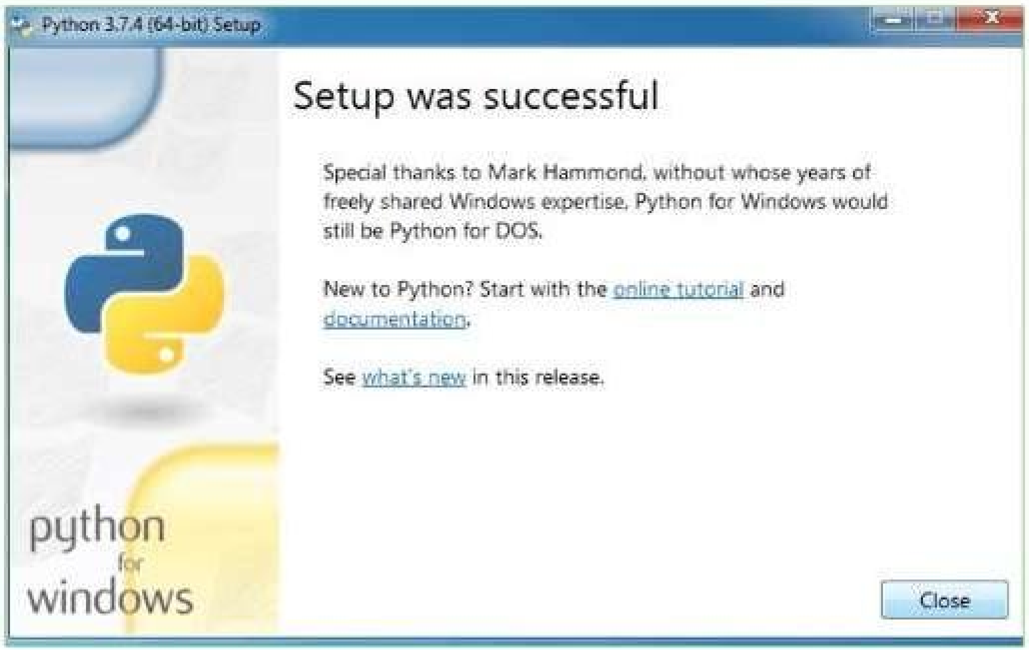
Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

Step 3: Click on Install NOW After the installation is successful. Click on Close.


Python 3.7.4 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

With these above three steps on python installation, you have successfully and correctlyinstalled Python. Now is the time to verify the installation.

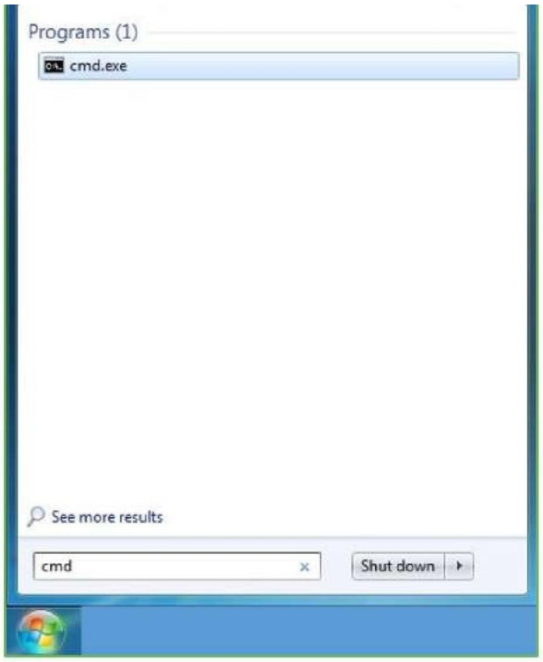
Note: The installation process might take a couple of minutes.Verify the Python Installation

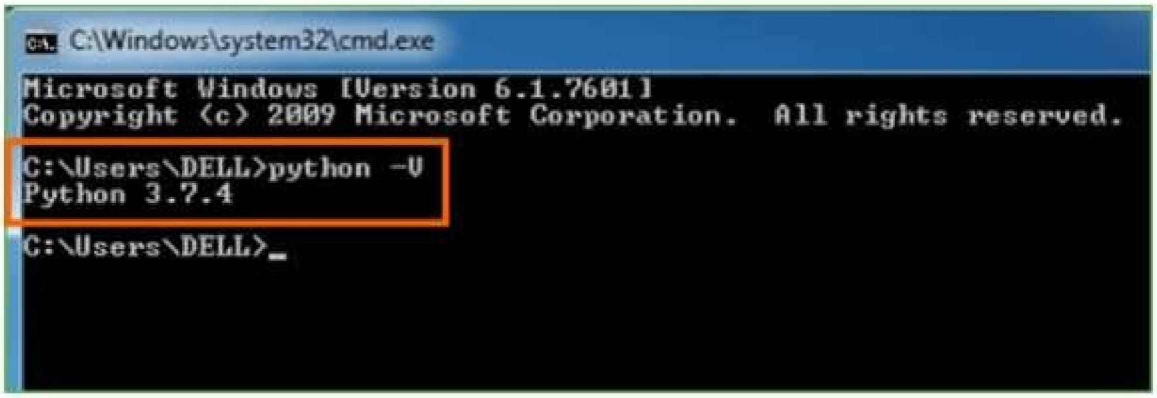
Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and pressEnter.





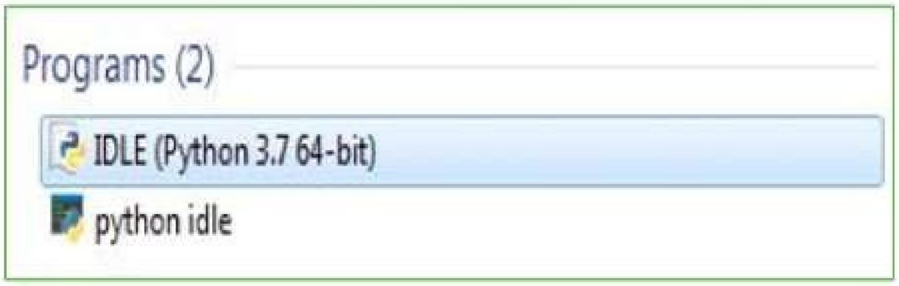
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must firstuninstall the earlier version and then install the new one.
Check how the Python IDLE works

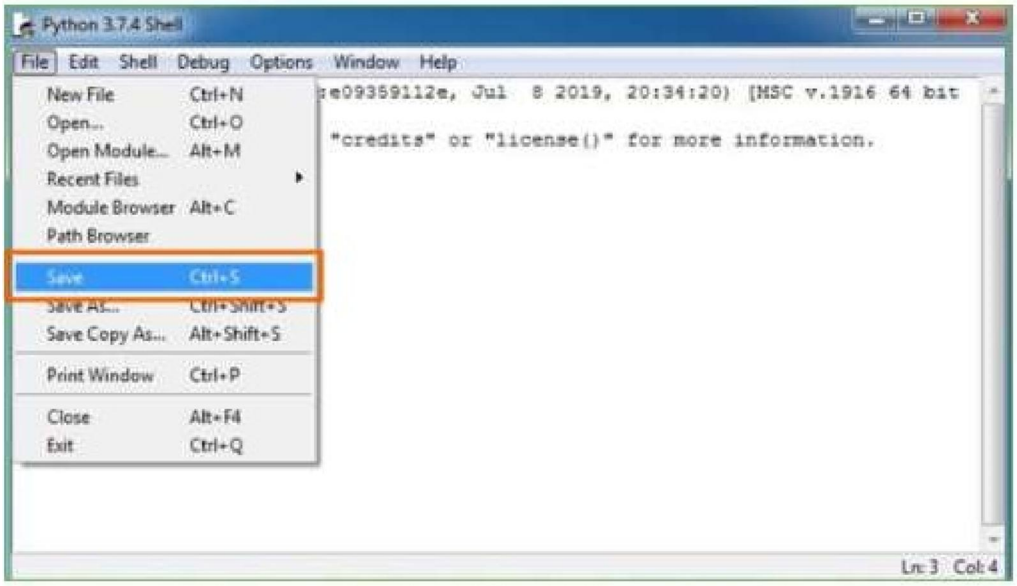
Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle".



Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File >Click on Save**



```
e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit
"credits" or "license()" for more information.
```

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here Ihave named the files as Hey World.
Step 6: Now for e.g. **enter print**

CHAPTER-6
SYSTEM TEST

6. SYSTEM TEST

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

6.1 TYPES OF TESTS

6.1.1 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 individualsoftware 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 constructionand 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.

6.1.2 Integration testing

Integration tests are designed to test integrated software components todetermine 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 thatalthough 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 ofcomponents.

6.1.3 Functional test

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 coveragepertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete,additional tests are identified and the effective value of current tests is determined.

6.1.4 System Test

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

6.1.5 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.

6.1.6 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 mostother 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.

6.1.7 Unit Testing

Unit testing is usually conducted as part of a combined code and unit testphase 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 bewritten 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.

6.1.8 Integration Testing

Software integration testing is the incremental integration testing of two ormore integrated software components on a single platform to produce failures caused byinterface 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 companylevel – interact without error.

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

Acceptance Testing

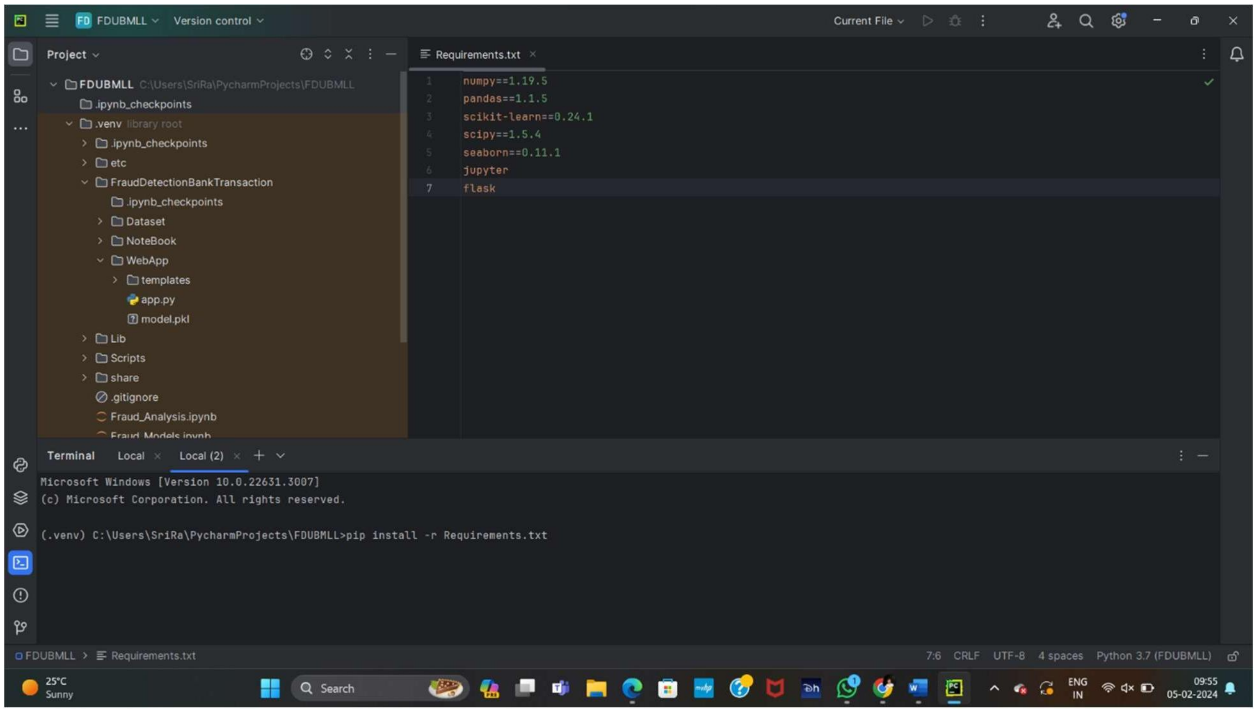
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

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

CHAPTER-7
SCREENSHOTS

7.SCREENSHOTS

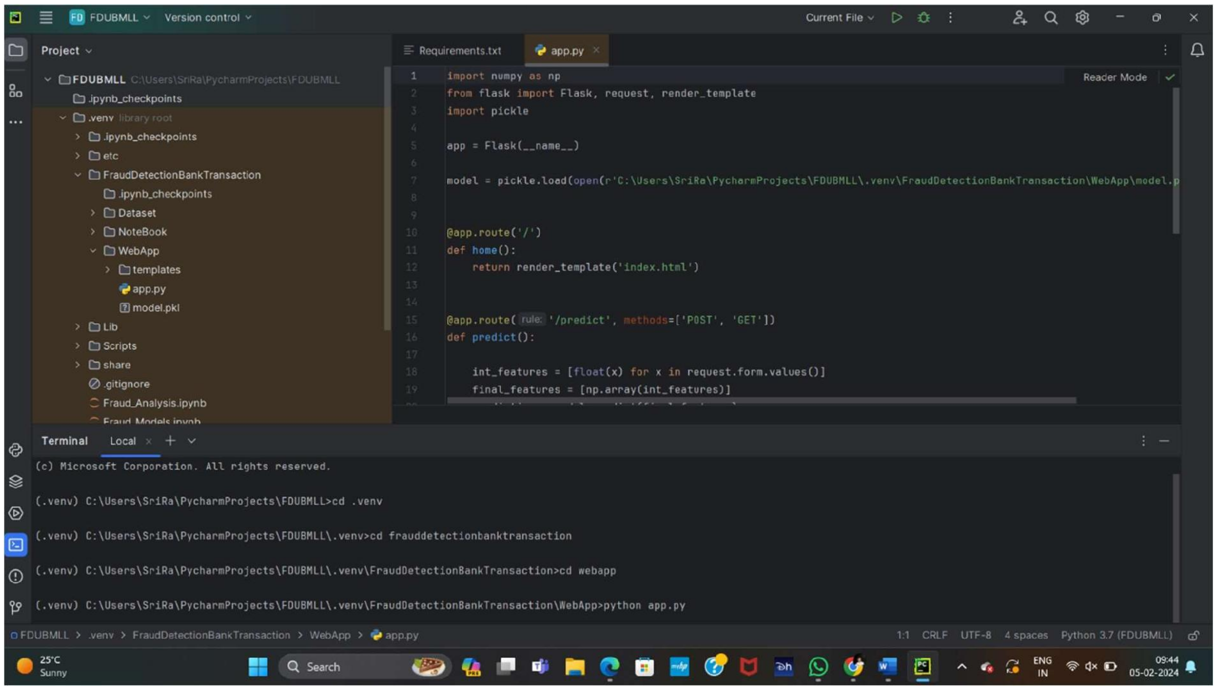
1. Click the FDUMLL project & install the Requirements.txt



```
numpy==1.19.5
pandas==1.1.5
scikit-learn==0.24.1
scipy==1.5.4
seaborn==0.11.1
jupyter
flask
```

```
Microsoft Windows [Version 10.0.22631.3007]
(c) Microsoft Corporation. All rights reserved.

(.venv) C:\Users\SriRa\PycharmProjects\FDUBMLL>pip install -r Requirements.txt
```

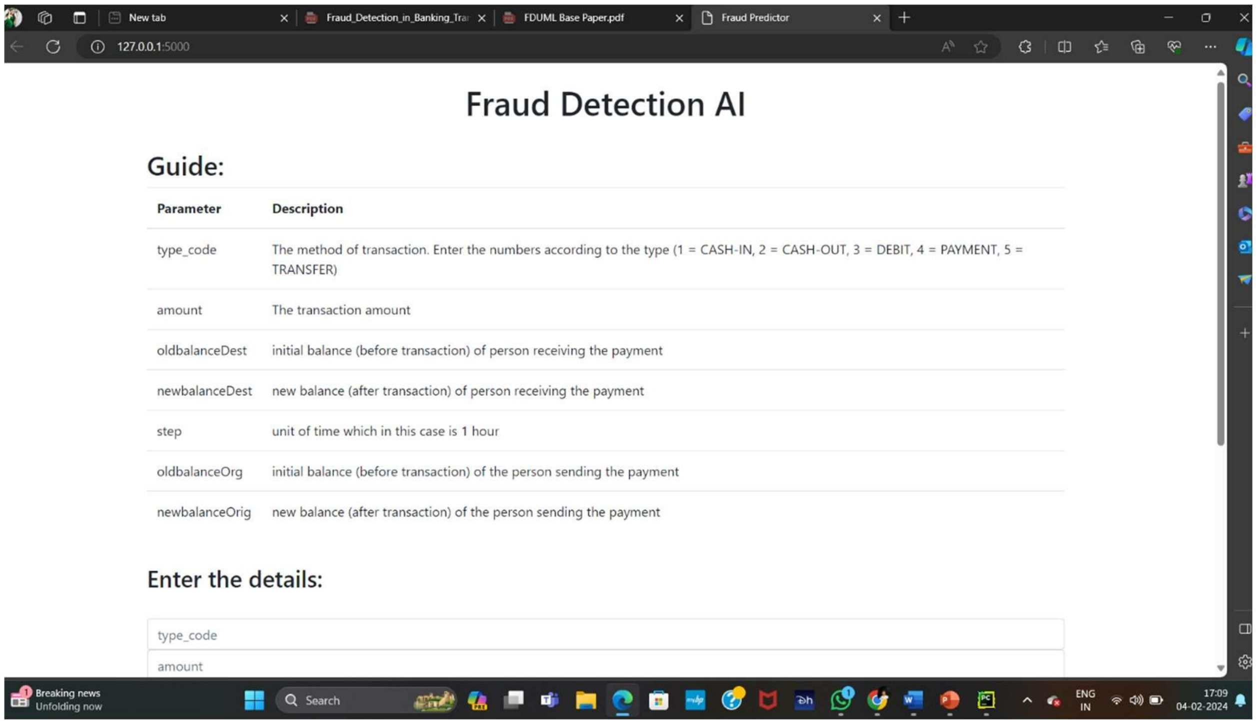
2. Enter the Commands to Run and predict the Values



```
import numpy as np
from flask import Flask, request, render_template
import pickle

app = Flask(__name__)

model = pickle.load(open(r'C:\Users\SriRa\PycharmProjects\FDUBMLL\.venv\FraudDetectionBankTransaction\WebApp\model.p


@app.route('/')
def home():
    return render_template('index.html')


@app.route('/predict', methods=['POST', 'GET'])
def predict():

    int_features = [float(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
```

```
(c) Microsoft Corporation. All rights reserved.

(.venv) C:\Users\SriRa\PycharmProjects\FDUBMLL>cd .venv

(.venv) C:\Users\SriRa\PycharmProjects\FDUBMLL\.venv>cd frauddetectionbanktransaction

(.venv) C:\Users\SriRa\PycharmProjects\FDUBMLL\.venv\FraudDetectionBankTransaction>cd webapp

(.venv) C:\Users\SriRa\PycharmProjects\FDUBMLL\.venv\FraudDetectionBankTransaction\WebApp>python app.py
```

3. Open the Browser & Run the Fraud predictor Values



127.0.0.1:5000

Fraud Detection AI

Guide:

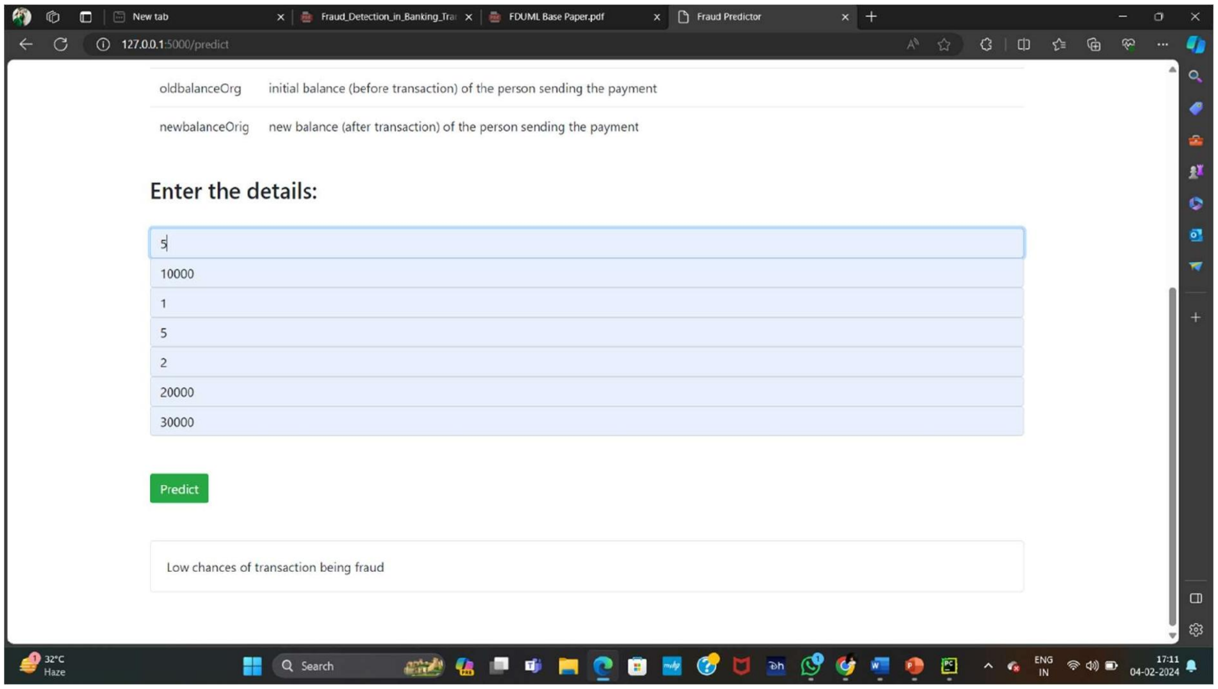
Parameter	Description
type_code	The method of transaction. Enter the numbers according to the type (1 = CASH-IN, 2 = CASH-OUT, 3 = DEBIT, 4 = PAYMENT, 5 = TRANSFER)
amount	The transaction amount
oldbalanceDest	initial balance (before transaction) of person receiving the payment
newbalanceDest	new balance (after transaction) of person receiving the payment
step	unit of time which in this case is 1 hour
oldbalanceOrg	initial balance (before transaction) of the person sending the payment
newbalanceOrig	new balance (after transaction) of the person sending the payment

Enter the details:

type_code

amount

4. Show the Output Values (Either High or Low)



127.0.0.1:5000/predict

oldbalanceOrg	initial balance (before transaction) of the person sending the payment
newbalanceOrig	new balance (after transaction) of the person sending the payment

Enter the details:

```
5
10000
1
5
2
20000
30000
```

Predict

Low chances of transaction being fraud

CHAPTER-8

CONCLUSION&FUTURE ENHANCEMENT

8.CONCLUSION&FUTURE ENHANCEMENT

9.1 Conclusion

The research proposes the integration of Random Forest, K-Nearest Neighbors (KNN), and Logistic Regression algorithms for fraud detection in banking applications. Utilizinga publicly available dataset from UCI, the analysis reveals a significant imbalance biasedtowards the majority of samples, addressed by the Synthetic Minority Over-sampling Technique (SMOTE). Implementation challenges with KNN and Random Forest method.The model demonstrates a commendable 97.74% performance rate. Notably, the analysishighlights a higher likelihood of fraudulent activity among individuals aged 19-25. This suggests the importance of considering demographic factors in fraud detection. The proposed ensemble of Random Forest, KNN, and Logistic Regression, offers an effectivesolution to address dataset imbalances and enhance fraud detection accuracy in banking applications.

FUTURE ENHANCEMENT:

In the future, advancements in fraud detection for banking transactions using machine learning will focus on real-time detection, advanced anomaly detection techniques, and enhanced interpretability. By incorporating behavioral analysis and ensemble learning methods, models will become more adept at identifying subtle patterns indicative of fraudulent activity while improving overall accuracy. Techniques such as adversarial robustness and incremental learning will ensure models remain resilient against evolving fraud tactics. Additionally, privacy-preserving approaches and cross-channel integration will enable more comprehensive fraud detection while safeguarding customer data. Continuous evaluation, global collaboration, and adherence to regulatory compliance will be paramount, ensuring that these systems remain effective, ethical, and trustworthy in safeguarding financial transactions.

REFERENCES

REFERENCES

[1] R. Rambola, P. Varshney and P. Vishwakarma, “Data Mining Techniques for Fraud Detection in Banking Sector,” 2018 4th International Conference on Computing Communication and Automation (ICCCA),Greater Noida, India, 2018, pp. 1-5, doi: 10.1109/CCAA.2018.8777535.

[2] N. Malini and M. Pushpa, “Analysis on credit card fraud identification techniques based on KNN and outlier detection,” 2017 Third international Conference on Advancesin Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2017, pp. 255-258, doi: 10.1109/AEEICB.2017.7972424.

[3] Ishan Sohony, Rameshwar Pratap, and Ullas Nambiar. 2018. Ensemble learning for credit card fraud detection. In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data (CoDS-COMAD ’18). Associationfor Computing Machinery, New York, NY, USA, 289–294. DOI:https://doi.org/10.1145/3152494.3156815

[4] C. Wang, Y. Wang, Z. Ye, L. Yan, W. Cai, and S. Pan, “Credit Card Fraud DetectionBased on Whale Algorithm Optimized BP Neural Network,” 2018 13th International Conference on Computer Science Education (ICCSE), Colombo, 2018, pp. 1-4, doi: 10.1109/ICCSE.2018.8468855

[5] I. Benchaji, S. Douzi and B. ElOuahidi, ”Using Genetic Algorithm to Improve Classification of Imbalanced Datasets for Credit Card Fraud Detection,” 2018 2nd CyberSecurity in Networking Conference (CSNet), Paris, 2018, pp. 1-5, doi: 10.1109/CSNET.2018.8602972.

[6] John O. Awoyemi, Adebayo Olusola Adetunmbi, and Samuel Adebayo Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI), pages 1–9, 2017.

[7] Fabrizio Carcillo, Andrea Dal Pozzolo, Yann-A¨el Le Borgne, Olivier Caelen, Yannis Mazzer, and Gianluca Bontempi. Scarff: a scalable framework for streaming credit card fraud detection with spark. Information Fusion, 41:182–194, 2018.

[8] Galina Baader and Helmut Krcmar. Reducing false positives in fraud detection: Combining the red flag

approach with process mining. International Journal of Accounting Information Systems, 2018.

[9] Ravisankar P, Ravi V, Raghava Rao G, and Bose, Detection of financial statement fraud and feature selection using data mining techniques, Elsevier, Decision Support Systems Volume 50, Issue 2, p491-500 (2011) SVM

[10] K. Seeja, and M. Zareapoor, "FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining," The Scientific World Journal, 2014, pp. 1-
8. KNN, SVM
[11] C. Tyagi, P. Parwekar, P. Singh, and K. Natla, "Analysis of Credit Card Fraud Detection Techniques," Solid State Technology, vol. 63, no. 6, 2020, pp. 18057-18069.
Credit card faud
[12] C. Chee, J. Jaafar, I. Aziz, M. Hassan, and W. Yeoh, "Algorithms for frequent itemsetmining: a literature review," Artificial Intelligence Review, vol. 52, 2019, pp. 2603–2621.
Litrature review AI
[13] S. Kiran, J. Guru, R. Kumar, N. Kumar, D. Katariya, and M. Sharma, "Credit card fraud detection using Naïve Bayes model based and KNN classifier," International Jounral of Advance Research, Ideas and Innovations in Technology, vol. 4, 2018, pp. 44-
47. KNN Naïve Byers
Pumsirirat, A.; Yan, L. Credit Card Fraud Detection Using Deep Learning based onAuto-Encoder and Restricted Boltzmann Machine. Available online:https://thesai.org/Downloads/Volume9No1/Paper_3
Credit_Card_Fraud_Detection_Using_Deep_ Learning.pdf (accessed on 23 February2021). DL

[14] PwC's Global Economic Crime and Fraud Survey 2020. Availableonline: https://www.pwc.com/fraudsurvey (accessed on 30 November 2020). Fraud server.

[15] Pourhabibi, T.; Ongb, K.L.; Kama, B.H.; Boo, Y.L. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. Decis. Support Syst. 2020, 133, 113303. Fraud detection.

[16] Lucas, Y.; Jurgovsky, J. Credit card fraud detection using machine learning: Asurvey. arXiv 2020, arXiv:2010.06479. Credit card fraud.

[17] Podgorelec, B.; Turkanovi´c, M.; Karakatiˇc, S. A Machine LearningBased Methodfor Automated Blockchain Transaction Signing Including Personalized Anomaly Detection. Sensors 20

[18] Amarasinghe, T.; Aponso, A.; Krishnarajah, N. Critical Analysis of Machine Learning Based Approaches for Fraud Detection in Financial Transactions. In Proceedings of the 2018 International Conference on Machine Learning Technologies (ICMLT'18), Nanchang, China, 21–23 June 2018; pp. 12–17. Machine learning for frauddetection.

[19] 20, 20, 147. Anomaly detection. Synthetic Financial Datasets for Fraud Detection. Available online: https://www.kaggle.com/ntnu-testimon/paysim1 (accessed on 30 November 2020). Frauddetection.

[20] Ma, T.; Qian, S.; Cao, J.; Xue, G.; Yu, J.; Zhu, Y.; Li, M. An Unsupervised Incremental Virtual Learning Method for Financial Fraud Detection. In Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–
Financial fraud detection.

[21] Puh, M.; Brki´c, L. Detecting Credit Card Fraud Using Selected Machine Learning Algorithms. In Proceedings of the 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 20–24 May 2019. Credit card fraud detection.

[22] Ryman-Tubb, N.F.; Krause, P.J.; Garn, W. How Artificial Intelligence and machinelearning research impacts payment card fraud detection: A survey and industry benchmark. Eng. Appl. Artif. Intell. 2018, 76, 130– 157. Credit card fraud detection.

[23] Xuan, S.; Liu, G.; Li, Z.; Zheng, L.; Wang, S.; Jiang, C. Random Forest for Credit Card Fraud Detection. In Proceedings of the 2018 IEEE 15th International Conference onNetworking, Sensing and Control (ICNSC), Zhuhai, China, 27–29 March 2018. RF.

[24] Huang, D.; Mu, D.; Yang, L.; Cai, X. CoDetect: Financial Fraud Detection with Anomaly Feature Detection. IEEE Access 2018, 6, 19161–19174. Financial fraud detection.