

# **ONLINE FRAUD DETECTION**

**A Project Report Submitted  
In Partial Fulfillment of the Requirements  
For the Degree of**

## **BACHELOR OF TECHNOLOGY**

**In  
Computer Science & Engineering  
by**

**SHUBHAM KUSHWAHA**

**(Roll No.-2001870100111)**

**SOMESH YADAV**

**(Roll No.-2001870100116)**

**YUVRAJ SINGH YADAV**

**(Roll No.-2001870100135)**

**MOHD. YASIR**

**(Roll No.-2001870100072)**

**Under the Supervision of  
MR. ASHUTOSH MISHRA  
(Assistant Professor)**



**To the  
DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**FEROZE GANDHI INSTITUTE OF ENGINEERING & TECHNOLOGY  
RAEBARELI**

**(Affiliated to Dr. A.P.J. Abdul Kalam Technical University)**

**CERTIFICATE**

This is to certify that **Shubham Kushwaha**( Roll No.-2001870100111) , **Somesh Yadav** (Roll No.-2001870100116) , **Yuvraj Singh Yadav**( Roll No.-2001870100135) , **Mohd Yasir**(Roll No.-2001870100072) has carried out the research work presented in the report entitled “**Online Fraud Detection**” for the award of **Bachelor of Technology** Department of **Computer Science and Engineering** from Dr. A.P.J. Abdul kalam Technical University , Lucknow under Mr. Ashutosh Mishra supervision. The project report embodies result of original work and studies carried out by the student himself and the content of report do not form the basis for the award of any other degree to the candidate or to any body else from this or any other University/Institution.

Signature

(Dr. Shruti Tripathi)

(Head of Department)

Signature

(Mr. Ashutosh Mishra)

(Assistant Professor)

Date:-

## **DECLARATION**

We hereby declare that this submission is our work and that , to the best of our knowledge and belief , contains no material previously published or written by another person nor material which to a substantial 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:-Shubham kushwaha

Roll No. :-2001870100111

Date:-

Signature:-

Name:- Somesh Yadav

Roll No. :-20001870100116

Date:-

Signature:-

Name:- Yuvraj Singh Yadav

Roll No. :-2001870100135

Date:-

Signature:-

Name:-Mohd Yasir

Roll No. :-2001870100072

Date:-

## **ACKNOWLEDGEMENT**

It gives a great sense of pleasure to present the report of B.tech Project undertaken during B.tech Final year. We owe Special Debt of gratitude to our project guide Mr. Ashutosh Mishra Department of Computer Science and Engineering, Feroze Gandhi Institute of Engineering and Technology, Raebareli for his constant support and guide throughout the course of our work. His sincerity, thoroughness and perseverance has been a constant source of inspiration to us. It is only his cognizant efforts that our endavour have been light of the day.

We also take the opportunity to acknowledge the contribution of Dr, Shruti Tripathi, Head Department of Computer Science and Engineering , Feroze Gandhi Institue of Engineering and Technology , Raebareli for his full support and assistance during developement of project.

We also do not like to miss the opportunity the contribution of all faculty members of the department for their kind assistance and cooperation during the developement of the project. Last but not the least, we acknowledge our friends for thier contribution in the completion of project.

## **ABSTRACT**

Online Fraud Detection utilising machine Learning techniques to enhance the existing security techniques in the banking sector. It recognises the fraud transaction taking place in the banking sector.

### **Key Functionality:-**

- Analyse the transaction that take place on the daily basis in the banking sector
- Categorise the daily transaction
- Recognises the Fraud Transaction
- Appropriate action taken on the transaction category
- Increase user trust and experience

## **TABLE OF CONTENTS**

CONTENT	PAGE NO.
Certificate	i
Declaration	ii
Abstract	iii
Acknowledgement	iv
Table of Contents	v
List of figure	vi
List of snapshots	vii
Chapter 1: Introduction	
1.1 Overview	1
1.2 Objective of the Project	2-3
1.3 Concept of Machine Learning	4-7
1.4 Machine Learning Techniques	
1.4.1 LogisticRegression	8-10
1.4.2 XGBoost	11-14
1.4.3 Support vector machine(SVC)	15-19
1.4.4 Random Forest Classifier	20-23
Chapter 2: Tools and Technology Used	
2.1 Hardware Requirement	24
2.2 Software Requirement	24-27
2.3 Github Codespace	28-29
Chapter 3: Technical Contents	30-33
Chapter 4: Training and Implementation of Machine Model	34-38
Chapter 5: Snapshots	40
Chapter 6: Conclusion	41
Chapter 7: References	42

**LIST OF FIGURE**

<b>FIGURE</b>	<b>Page</b>
Figure-i	4
Figure-ii	5
Figure-iii	6
Figure-iv	7
Figure-v	7
Figure-vi	12
Figure-vii	19
Figure-viii	20

**LIST OF SNAPSHOT**

<b>SNAPSHOT</b>	<b>PAGE</b>
Snapshot-i	29
Snapshot-ii	39
Snapshot-iii	39
Snapshot-iv	40
Snapshot-v	40
Snapshot-vi	41
Snapshot-vii	41



## **CHAPTER-1: INTRODUCTION**

### **1.1 OVERVIEW**

Online Fraud Detection is a project that utilises machine learning concept to enhance security mechanism of the existing Banking system. On integrating this technique with existing security checks, it enables banking system to give real time analysis of the ongoing transaction and classify it as a fraud transaction or a valid transaction.

On the basis of classification the appropriate action is taken like if transaction is fraud then it is immediately abort or cancelled and the customer is blocked.

In the case of a valid transaction it is further proceeded to the next step.

It not only increase the security of the existing banking sector but also increase the trust and belief of the user on availing banking services. It gives assurance to the user that there thing are safe and secure.

## 1.2 OBJECTIVE OF THE PROJECT

The purpose of an online fraud detection system is to fulfill specific goals and objectives related to identifying and preventing fraudulent activities in digital and online environments.

Here's a detailed explanation of the purpose of such a system:

1. **Security Enhancement:**-The primary purpose is to bolster the security of online transaction and interactions. By detecting and preventing fraudulent activities, the system helps protect individuals and businesses from financial losses and reputational damage caused by fraudulent transactions and activities.
2. **Fraud Loss Protection:**- Online fraud can result in substantial financial losses for businesses and individuals. The purpose of the system is to minimize these financial losses by identifying and blocking fraudulent transactions in real-time.
3. **User Trust and Confidence:**- A robust fraud detection system builds trust and confidence among users of online services. When users feel that their transactions are secure, they are more likely to engage in online activities, which benefits both businesses and the digital economy as a whole.
4. **Real Time Detection:**- The system's purpose extends to the swift identification of fraudulent activities as they occur. Real-time detection allows for immediate action to be taken, reducing the impact of fraud and potentially apprehending perpetrators.
5. **Adaptability:**- Online fraud tactics are constantly evolving. The purpose of the system includes adaptability and continuous improvement to stay ahead of emerging fraud schemes. This ensures that the system remains effective over time.

6.Compliances and Regulations:- Any industries and regions have specific regulations and compliance requirements related to fraud prevention and data security. The system's purpose also includes ensuring that businesses comply with these regulations.

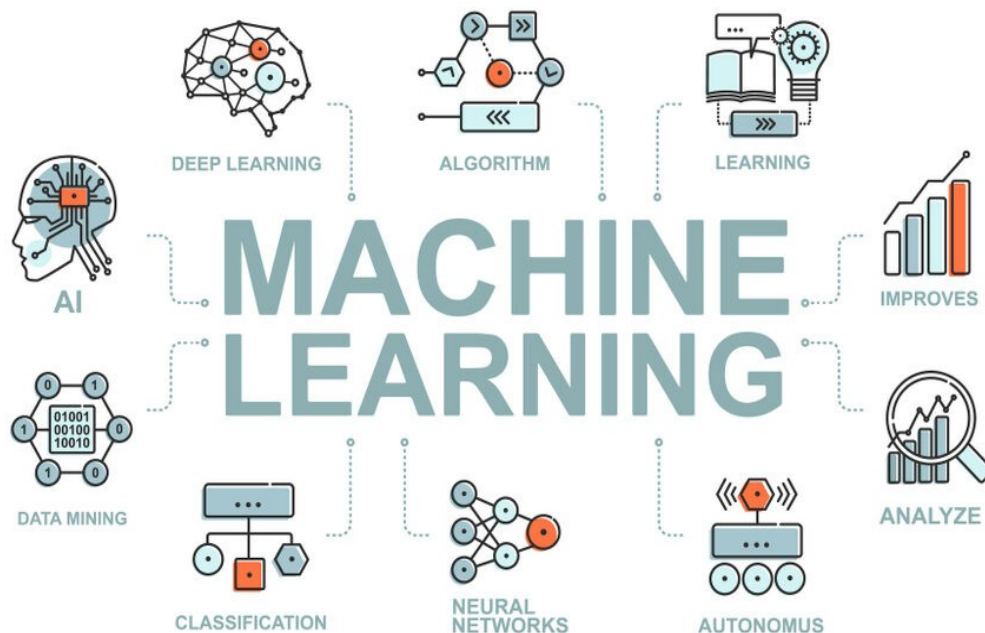
7.Data Security:- The system's purpose includes safeguarding sensitive user data. By preventing fraudulent access to user accounts and personal information, it contributes to data protection and privacy.

### 1.3 CONCEPT OF MACHINE LEARNING

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

#### Features of Machine learning

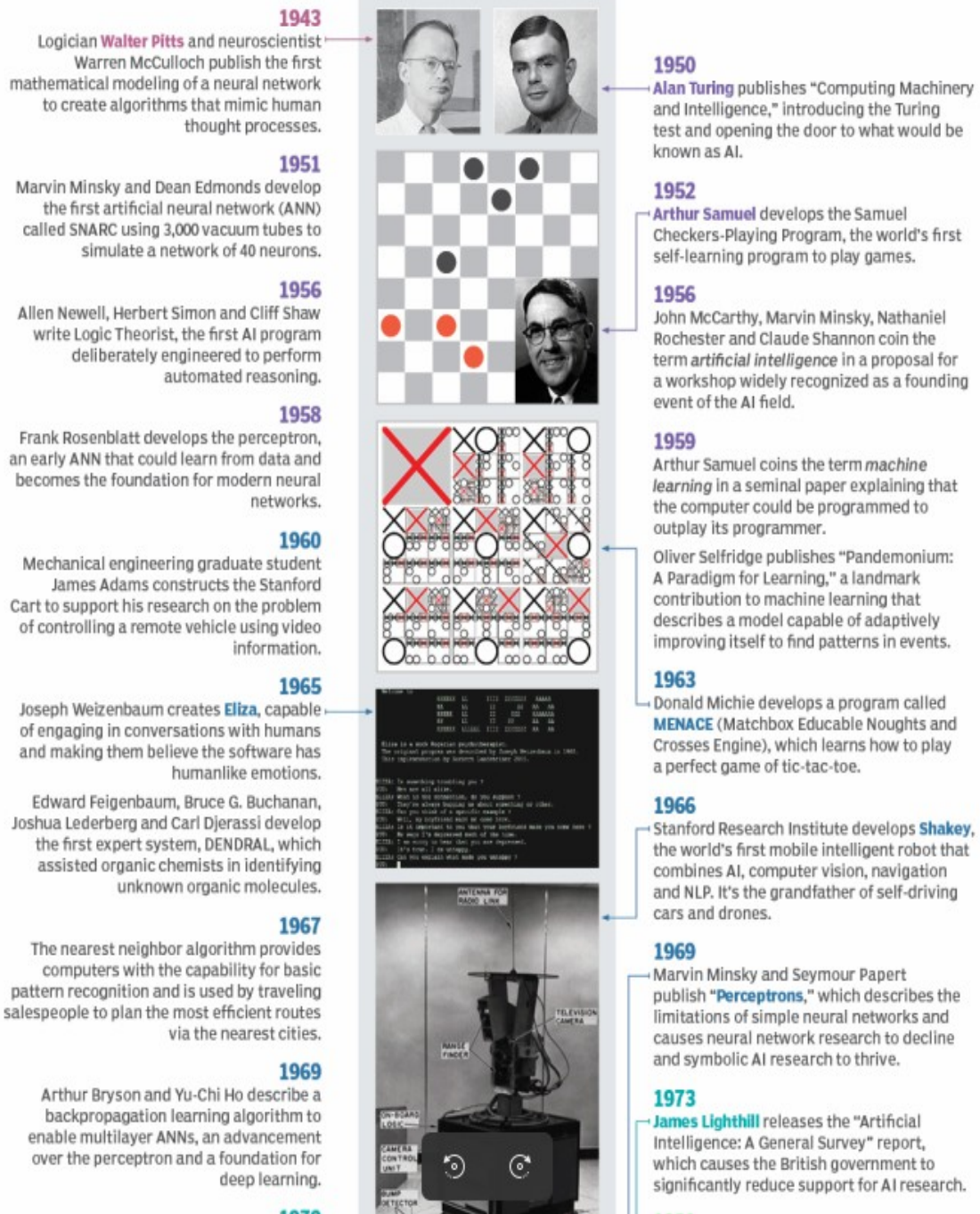
- Machine learning is data driven technology. Large amount of data generated by organizations on daily bases. So, by notable relationships in data, organizations makes better decisions.
- Machine can learn itself from past data and automatically improve.
- From the given dataset it detects various patterns on data.
- For the big organizations branding is important and it will become more easy to target relatable customer base.
- It is similar to data mining because it is also deals with the huge amount of data.



( Figure-i)

# Machine learning's long legacy

The history of machine learning dates from the early esoteric beginnings of neural networks to recent breakthroughs in generative AI that democratize new and controversial ways to create content.



(Figure-ii)





(Figure-iii)

## ML challenges and associated pitfalls

	<b>Dealing with risk</b> <ul style="list-style-type: none"> <li>• ML bias and other ethical issues.</li> <li>• Cyber attacks such as data poisoning.</li> <li>• Employees potentially displaced by AI.</li> <li>• Hallucinations embedded in business/legal documents.</li> </ul>		<b>Ensuring ML adoption</b> <ul style="list-style-type: none"> <li>• Emphasis on implementation over adoption.</li> <li>• Lack of focus on governance, change management and UX.</li> <li>• ML tools used improperly or not at all.</li> </ul>
	<b>Framing the ML problem</b> <ul style="list-style-type: none"> <li>• Putting the "solution" before the problem.</li> <li>• Lack of a compelling business case.</li> <li>• Stalled projects due to underresourcing.</li> </ul>		<b>Addressing data literacy</b> <ul style="list-style-type: none"> <li>• Users lack sophisticated statistical background.</li> <li>• Lack of confidence slows adoption.</li> </ul>
	<b>Investing in data quality</b> <ul style="list-style-type: none"> <li>• Neglecting the data prep phase.</li> <li>• Resistance to upfront data investment.</li> <li>• Low-quality data leads to rework later.</li> </ul>		<b>Accepting uncertainty</b> <ul style="list-style-type: none"> <li>• Comfort with deterministic software development projects.</li> <li>• Unwillingness to fail fast and move on.</li> </ul>

ICONS: DAVIDDA/JETTY IMAGES

©2021 TECHTARGET. ALL RIGHTS RESERVED. TechTarget

(Figure-iv)

## Machine learning models and their training algorithms

Supervised learning	Unsupervised learning	Semi-supervised learning	Reinforcement learning
<p>Data scientists provide input, output and feedback to build model (as the definition).</p> <p><b>EXAMPLE ALGORITHMS:</b></p> <p><b>Linear regressions</b></p> <ul style="list-style-type: none"> <li>■ Sales forecasting.</li> <li>■ Risk assessment.</li> </ul> <p><b>Support vector machines</b></p> <ul style="list-style-type: none"> <li>■ Image classification.</li> <li>■ Financial performance comparison.</li> </ul> <p><b>Decision trees</b></p> <ul style="list-style-type: none"> <li>■ Predictive analytics.</li> <li>■ Pricing.</li> </ul>	<p>Use deep learning to arrive at conclusions and patterns through unlabeled training data.</p> <p><b>EXAMPLE ALGORITHMS:</b></p> <p><b>Apriori</b></p> <ul style="list-style-type: none"> <li>■ Sales functions.</li> <li>■ Word associations.</li> <li>■ Searcher.</li> </ul> <p><b>K-means clustering</b></p> <ul style="list-style-type: none"> <li>■ Performance monitoring.</li> <li>■ Searcher intent.</li> </ul> <p><b>Artificial neural networks</b></p> <ul style="list-style-type: none"> <li>■ Generate new, synthetic data.</li> <li>■ Data mining and pattern recognition.</li> </ul>	<p>Builds a model through a mix of labeled and unlabeled data, a set of categories, suggestions and example labels.</p> <p><b>EXAMPLE ALGORITHMS:</b></p> <p><b>Generative adversarial networks</b></p> <ul style="list-style-type: none"> <li>■ Audio and video manipulation.</li> <li>■ Data creation.</li> </ul> <p><b>Self-trained Naïve Bayes classifier</b></p> <ul style="list-style-type: none"> <li>■ Natural language processing.</li> </ul>	<p>Self-interpreting but based on a system of rewards and punishments learned through trial and error, seeking maximum reward.</p> <p><b>EXAMPLE ALGORITHMS:</b></p> <p><b>Q-learning</b></p> <ul style="list-style-type: none"> <li>■ Policy creation.</li> <li>■ Consumption reduction.</li> </ul> <p><b>Model-based value estimation</b></p> <ul style="list-style-type: none"> <li>■ Linear tasks.</li> <li>■ Estimating parameters.</li> </ul>

©2021 TECHTARGET. ALL RIGHTS RESERVED. TechTarget

( Figure-v)

## 1.4 MACHINE LEARNING TECHNIQUES

### 1.4.1 LOGISTIC REGRESSION

Logistic regression is a supervised machine learning algorithm used for classification tasks where the goal is to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyze the relationship between two data factors. The article explores the fundamentals of logistic regression, it's types and implementations.

#### Key Points:

- Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).

#### Types of Logistic Regression

On the basis of the categories, Logistic Regression can be classified into three types:

1. **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

2. **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as “cat”, “dogs”, or “sheep”

3. **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as “low”, “Medium”, or “High”.

#### Assumptions of Logistic Regression

We will explore the assumptions of logistic regression as understanding these assumptions important to ensure that we are using appropriate application of the model. The assumption include:

1. **Independent observations:** Each observation is independent of the other. meaning there is no correlation between any input variables.



2.Binary dependent variables: It takes the assumption that the dependent variable must be binary or dichotomous, meaning it can take only two values. For more than two categories SoftMax functions are used.

3.Linearity relationship between independent variables and log odds: The relationship between the independent variables and the log odds of the dependent variable should be linear.

4.No outliers: There should be no outliers in the dataset.

5.Large sample size: The sample size is sufficiently large

### **Terminologies involved in Logistic Regression**

Here are some common terms involved in logistic regression:

- Independent variables:** The input characteristics or predictor factors applied to the dependent variable's predictions.
- Dependent variable:** The target variable in a logistic regression model, which we are trying to predict.
- Logistic function:** The formula used to represent how the independent and dependent variables relate to one another. The logistic function transforms the input variables into a probability value between 0 and 1, which represents the likelihood of the dependent variable being 1 or 0.
- Odds:** It is the ratio of something occurring to something not occurring. It is different from probability as the probability is the ratio of something occurring to everything that could possibly occur.
- Log-odds:** The log-odds, also known as the logit function, is the natural logarithm of the odds. In logistic regression, the log odds of the dependent variable are modeled as a linear combination of the independent variables and the intercept.
- Coefficient:** The logistic regression model's estimated parameters, show how the independent and dependent variables relate to one another.
- Intercept:** A constant term in the logistic regression model, which represents the log odds when all independent variables are equal to zero.

## How does Logistic Regression work?

The logistic regression model transforms the **linear regression** function continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1. This function is known as the logistic function.

Let the independent input features be:

$$X = \begin{bmatrix} x_{11} & \dots & x_{1m} \\ x_{21} & \dots & x_{2m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nm} \end{bmatrix}$$

and the dependent variable is Y having only binary value i.e. 0 or 1.

$$Y = \begin{cases} 0 & \text{if Class 1} \\ 1 & \text{if Class 2} \end{cases}$$

then, apply the multi-linear function to the input variables X.

$$z = (\sum_{i=1}^n w_i x_i) + b$$

Here  $x_i$  is the  $i$ th observation of X,  $w_i = [w_1, w_2, w_3, \dots, w_m]$  is the weights or Coefficient, and  $b$  is the bias term also known as intercept. simply this can be represented as the dot product of weight and bias.

$$z = w \cdot X + b$$

## Logistic Regression Equation

The odd is the ratio of something occurring to something not occurring. it is different from probability as the probability is the ratio of something occurring to everything that could possibly occur. so odd will be:

$$\frac{p(x)}{1-p(x)} = e^z$$

Applying natural log on odd. then log odd will be:

$$\begin{aligned} \log \left[ \frac{p(x)}{1-p(x)} \right] &= z \\ \log \left[ \frac{p(x)}{1-p(x)} \right] &= w \cdot X + b \\ \frac{p(x)}{1-p(x)} &= e^{w \cdot X + b} \quad \dots \text{Exponentiate both sides} \\ p(x) &= e^{w \cdot X + b} \cdot (1 - p(x)) \\ p(x) &= e^{w \cdot X + b} - e^{w \cdot X + b} \cdot p(x) \\ p(x) + e^{w \cdot X + b} \cdot p(x) &= e^{w \cdot X + b} \\ p(x)(1 + e^{w \cdot X + b}) &= e^{w \cdot X + b} \\ p(x) &= \frac{e^{w \cdot X + b}}{1 + e^{w \cdot X + b}} \end{aligned}$$

then the final logistic regression equation will be:

$$p(X; b, w) = \frac{e^{w \cdot X + b}}{1 + e^{w \cdot X + b}} = \frac{1}{1 + e^{-w \cdot X - b}}$$

## Implementation of Logistic Regression(Example)

```
# import the necessary libraries
from sklearn.datasets import load_breast_cancer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# load the breast cancer dataset
X, y = load_breast_cancer(return_X_y=True)

# split the train and test dataset
X_train, X_test, \
    y_train, y_test = train_test_split(X, y,
                                       test_size=0.20,
                                       random_state=23)

# LogisticRegression
clf = LogisticRegression(random_state=0)
clf.fit(X_train, y_train)

# Prediction
y_pred = clf.predict(X_test)

acc = accuracy_score(y_test, y_pred)
print("Logistic Regression model accuracy (in %):", acc*100)
```

### 1.4.2 XGBoost

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for “Extreme Gradient Boosting” and it has become one of the most popular and widely used machine learning algorithms due to its ability to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

One of the key features of XGBoost is its efficient handling of missing values, which allows it to handle real-world data with missing values without requiring significant pre-processing. Additionally, XGBoost has built-in support for parallel processing, making it possible to train models on large datasets in a reasonable amount of time.

XGBoost can be used in a variety of applications, including Kaggle competitions, recommendation systems, and click-through rate prediction, among others. It is also highly customizable and allows for fine-tuning of various model parameters to optimize performance.

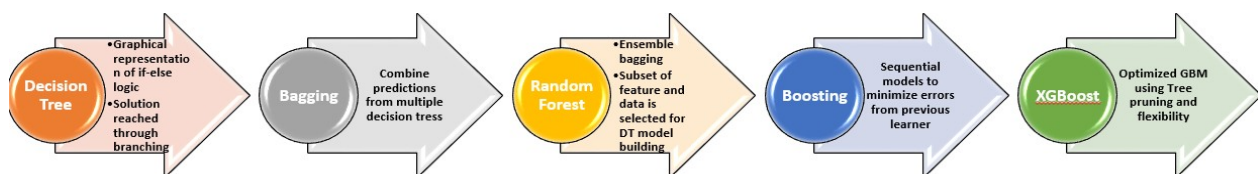
XgBoost stands for Extreme Gradient Boosting, which was proposed by the researchers at the University of Washington. It is a library written in C++ which optimizes the training for Gradient Boosting.

#### **Advantages of XGBoost:**

1. **Performance:** XGBoost has a strong track record of producing high-quality results in various machine learning tasks, especially in Kaggle competitions, where it has been a popular choice for winning solutions.
2. **Scalability:** XGBoost is designed for efficient and scalable training of machine learning models, making it suitable for large datasets.
3. **Customizability:** XGBoost has a wide range of hyperparameters that can be adjusted to optimize performance, making it highly customizable.
4. **Handling of Missing Values:** XGBoost has built-in support for handling missing values, making it easy to work with real-world data that often has missing values.
5. **Interpretability:** Unlike some machine learning algorithms that can be difficult to interpret, XGBoost provides feature importances, allowing for a better understanding of which variables are most important in making predictions.

### Disadvantages of XGBoost:

1. Computational Complexity: XGBoost can be computationally intensive, especially when training large models, making it less suitable for resource-constrained systems.
2. Overfitting: XGBoost can be prone to overfitting, especially when trained on small datasets or when too many trees are used in the model.
3. Hyperparameter Tuning: XGBoost has many hyperparameters that can be adjusted, making it important to properly tune the parameters to optimize performance. However, finding the optimal set of parameters can be time-consuming and requires expertise.
4. Memory Requirements: XGBoost can be memory-intensive, especially when working with large datasets, making it less suitable for systems with limited memory resources.



(Figure-vi)

### Parameters in XGBoost

- **Learning Rate (eta):** An important variable that modifies how much each tree contributes to the final prediction. While more trees are needed, smaller values frequently result in more accurate models.
- **Max Depth:** This parameter controls the depth of every tree, avoiding overfitting and being essential to controlling the model's complexity.
- **Gamma:** Based on the decrease in loss, it determines when a node in the tree will split. The algorithm becomes more conservative with a higher gamma value, avoiding splits that don't appreciably lower the loss. It aids in managing tree complexity.
- **Subsample:** Manages the percentage of data that is sampled at random to grow each tree, hence lowering variance and enhancing generalization. Setting it too low, though, could result in underfitting.
- **Colsample Bytree:** Establishes the percentage of features that will be sampled at random for growing each tree.

- **n\_estimators**: Specifies the number of boosting rounds.
- **lambda (L2 regularization term) and alpha (L1 regularization term)**: Control the strength of L2 and L1 regularization, respectively. A higher value results in stronger regularization.
- **min\_child\_weight**: Influences the tree structure by controlling the minimum amount of data required to create a new node.
- **scale\_pos\_weight**: Useful in imbalanced class scenarios to control the balance of positive and negative weights.

### What Makes XGBoost “eXtreme”?

XGBoost extends traditional gradient boosting by including regularization elements in the objective function, XGBoost improves generalization and prevents overfitting.

**Preventing Overfitting**:-The learning rate, also known as shrinkage, is a new parameter introduced by XGBoost. It is represented by the symbol “eta.” It quantifies each tree’s contribution to the total prediction. Because each tree has less of an influence, an optimization process with a lower learning rate is more resilient. By making the model more conservative, regularization terms combined with a low learning rate assist avoid overfitting.

XGBoost constructs trees level by level, assessing whether adding a new node (split) enhances the objective function as a whole at each level. The split is trimmed if not. This level growth along with trimming makes the trees easier to understand and easier to create.

The regularization terms, along with other techniques such as shrinkage and pruning, play a crucial role in preventing overfitting, improving generalization, and making XGBoost a robust and powerful algorithm for various machine learning tasks.

**Tree Structure**:-Conventional decision trees are frequently developed by expanding each branch until a stopping condition is satisfied, or in a depth-first fashion. On the other hand, XGBoost builds trees level-wise or breadth-first. This implies that it adds nodes for every feature at a certain depth before moving on to the next level, so growing the tree one level at a time.

**Determining the Best Splits**: XGBoost assesses every split that might be made for every feature at every level and chooses the one that minimizes the objective function as much as feasible (e.g., minimizing the mean squared error for regression tasks or cross-entropy for classification tasks). In contrast, a single feature is selected for a split at each level in depth-wise expansion.

**Prioritizing Important Features**: The overhead involved in choosing the best split for each feature at each level is decreased by level-wise growth. XGBoost eliminates the need to revisit

and assess the same feature more than once during tree construction because all features are taken into account at the same time.

### 1.4.3 SUPPORT VECTOR MACHINE(SVC)

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.

SVM algorithms are very effective as we try to find the maximum separating hyperplane between the different classes available in the target feature.

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

#### Support Vector Machine Terminology

- 1.**Hyperplane:**Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space. In the case of linear classifications, it will be a linear equation i.e.  $wx+b=0$ .
- 2.**Support Vectors:**Support vectors are the closest data points to the hyperplane, which makes a critical role in deciding the hyperplane and margin.
- 3.**Margin:** Margin is the distance between the support vector and hyperplane. The main objective of the support vector machine algorithm is to maximize the margin. The wider margin indicates better classification performance.
- 4.**Kernel:** Kernel is the mathematical function, which is used in SVM to map the original input data points into high-dimensional feature spaces, so, that the hyperplane can be easily found out even if the data points are not linearly separable in the original input space. Some of the common kernel functions are linear, polynomial, radial basis function(RBF), and sigmoid.



5.**Hard Margin:**The maximum-margin hyperplane or the hard margin hyperplane is a hyperplane that properly separates the data points of different categories without any misclassifications.

6.**Soft Margin:**When the data is not perfectly separable or contains outliers, SVM permits a soft margin technique. Each data point has a slack variable introduced by the soft-margin SVM formulation, which softens the strict margin requirement and permits certain misclassifications or violations. It discovers a compromise between increasing the margin and reducing violations.

7.**C:**Margin maximisation and misclassification fines are balanced by the regularisation parameter C in SVM. The penalty for going over the margin or misclassifying data items is decided by it. A stricter penalty is imposed with a greater value of C, which results in a smaller margin and perhaps fewer misclassifications.

8.**Hinge Loss:**A typical loss function in SVMs is hinge loss. It punishes incorrect classifications or margin violations. The objective function in SVM is frequently formed by combining it with the regularisation term.

9.**Dual Problem:**A dual Problem of the optimisation problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The dual formulation enables the use of kernel tricks and more effective computing.

### Types of Support Vector Machine

Based on the nature of the decision boundary, Support Vector Machines (SVM) can be divided into two main parts:

- Linear SVM:**Linear SVMs use a linear decision boundary to separate the data points of different classes. When the data can be precisely linearly separated, linear SVMs are very suitable. This means that a single straight line (in 2D) or a hyperplane (in higher dimensions) can entirely divide the data points into their respective classes. A hyperplane that maximizes the margin between the classes is the decision boundary.

- Non-Linear SVM:**Non-Linear SVM can be used to classify data when it cannot be separated into two classes by a straight line (in the case of 2D). By using kernel functions, nonlinear SVMs can handle nonlinearly separable data. The original input data is

ransformed by these kernel functions into a higher-dimensional feature space, where the data points can be linearly separated. A linear SVM is used to locate a nonlinear decision boundary in this modified space.

### Advantages of SVM

- Effective in high-dimensional cases.
- Its memory is efficient as it uses a subset of training points in the decision function called support vectors.
- Different kernel functions can be specified for the decision functions and its possible to specify custom kernels.

### Implementation of Support Vector Machine(Example)

```
from sklearn.datasets import load_breast_cancer
import matplotlib.pyplot as plt
from sklearn.inspection import DecisionBoundaryDisplay
from sklearn.svm import SVC

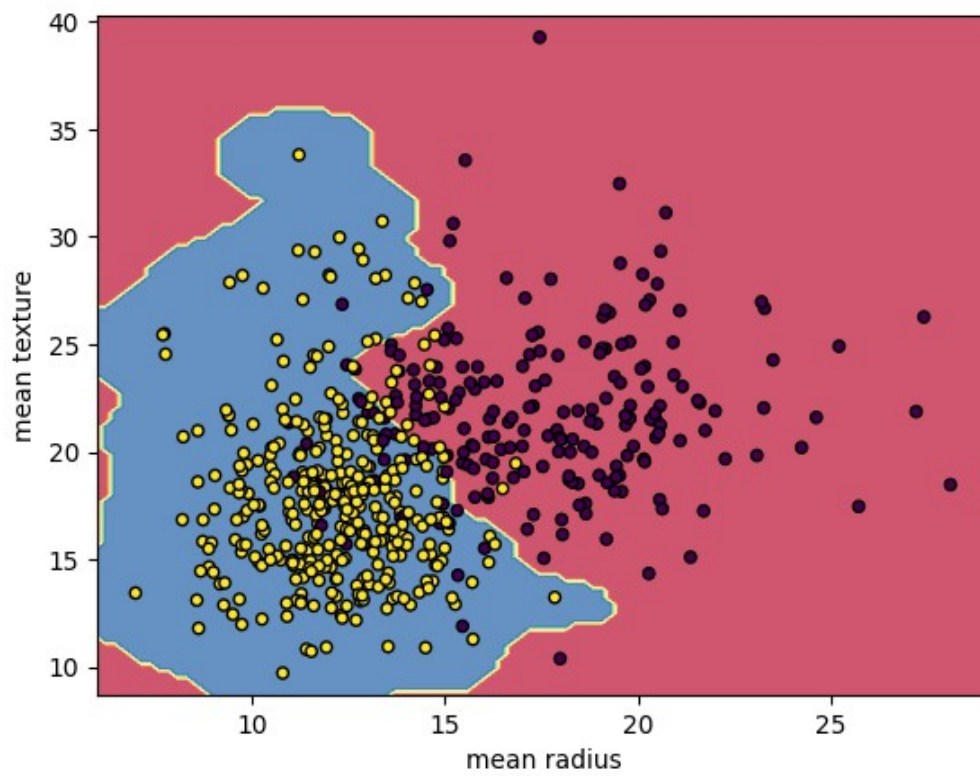
# Load the datasets
cancer = load_breast_cancer()
X = cancer.data[:, :2]
y = cancer.target

#Build the model
svm = SVC(kernel="rbf", gamma=0.5, C=1.0)
# Trained the model
svm.fit(X, y)

# Plot Decision Boundary
DecisionBoundaryDisplay.from_estimator(
    svm,
    X,
    response_method="predict",
    cmap=plt.cm.Spectral,
    alpha=0.8,
    xlabel=cancer.feature_names[0],
    ylabel=cancer.feature_names[1],
)

# Scatter plot
plt.scatter(X[:, 0], X[:, 1],
            c=y,
            s=20, edgecolors="k")
plt.show()
```

Output



*Breast Cancer Classifications with SVM RBF kernel*

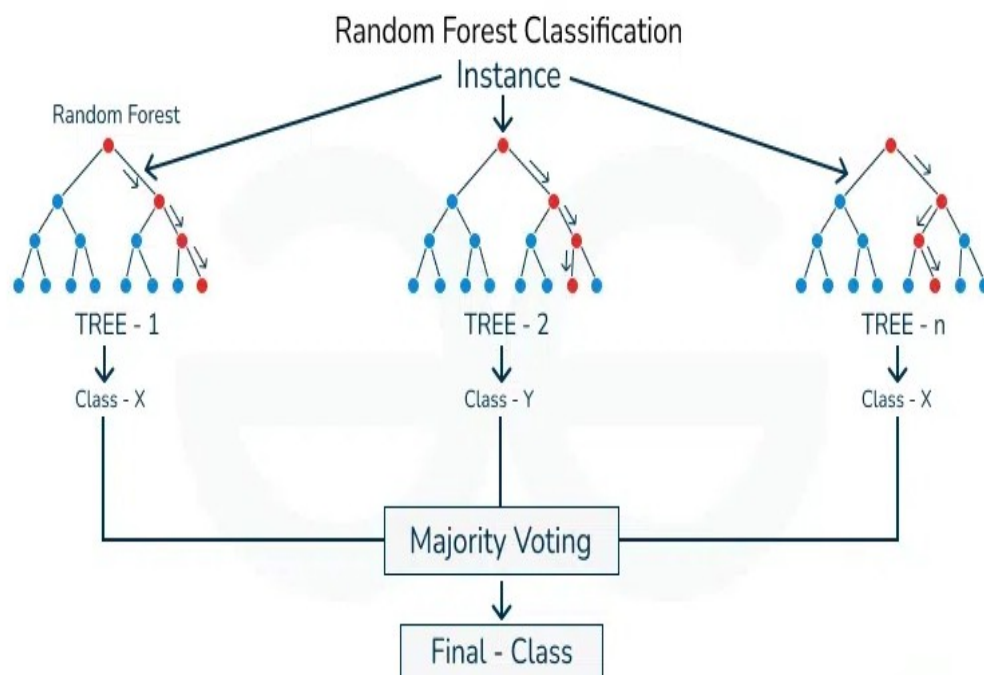
(Figure-vii)

#### 1.4.4 RANDOM FOREST CLASSIFIER

The Random Forest or Random Decision Forest is a supervised Machine learning algorithm used for classification, regression, and other tasks using decision trees. Random Forests are particularly well-suited for handling large and complex datasets, dealing with high-dimensional feature spaces, and providing insights into feature importance. This algorithm's ability to maintain high predictive accuracy while minimizing overfitting makes it a popular choice across various domains, including finance, healthcare, and image analysis, among others.

The Random forest classifier creates a set of decision tree from a randomly selected subset of the training set. It is a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction.

Additionally, the random forest classifier can handle both classification and regression tasks, and its ability to provide feature importance scores makes it a valuable tool for understanding the significance of different variables in the dataset.



(Figure-viii)

### Random Forest Classifier Parameters

- **n\_estimators:** Number of trees in the forest.
  - More trees generally lead to better performance, but at the cost of computational time.
  - Start with a value of 100 and increase as needed.
- **max\_depth:** Maximum depth of each tree.
  - Deeper trees can capture more complex patterns, but also risk overfitting.
  - Experiment with values between 5 and 15, and consider lower values for smaller datasets.
- **max\_features:** Number of features considered for splitting at each node.
  - A common value is 'sqrt' (square root of the total number of features).
  - Adjust based on dataset size and feature importance.
- **Criterion:** Function used to measure split quality ('gini' or 'entropy').
  - Gini impurity is often slightly faster, but both are generally similar in performance.
- **min\_samples\_split:** Minimum samples required to split a node.
  - Higher values can prevent overfitting, but too high can hinder model complexity.
  - Start with 2 and adjust as needed.
- **min\_samples\_leaf:** Minimum samples required to be at a leaf node.
  - Similar to min\_samples\_split, but focused on leaf nodes.
  - Start with 1 and adjust as needed.
- **bootstrap:** Whether to use bootstrap sampling when building trees (True or False).
  - Bootstrapping can improve model variance and generalization, but can slightly increase bias.

### Advantages of Random Forest Classifier

- The ensemble nature of Random Forests, combining multiple trees, makes them less prone to overfitting compared to individual decision trees.
- Effective on datasets with a large number of features, and it can handle irrelevant variables well.

- Random Forests can provide insights into feature importance, helping in feature selection and understanding the dataset.

### Disadvantages of Random Forest Classifier

- Random Forests can be computationally expensive and may require more resources due to the construction of multiple decision trees.
- The ensemble nature makes it challenging to interpret the reasoning behind individual predictions compared to a single decision tree.
- In imbalanced datasets, Random Forests may be biased toward the majority class, impacting the predictive performance for minority classes.

### Implementation(Example)

```
# importing random forest classifier from assemble module
from sklearn.ensemble import RandomForestClassifier
import pandas as pd
# creating dataframe of IRIS dataset
data = pd.DataFrame({'sepalwidth': iris.data[:, 0], 'sepalwidth': iris.data[:, 1],
                    'petallength': iris.data[:, 2], 'petalwidth': iris.data[:, 3],
                    'species': iris.target})

# creating a RF classifier
clf = RandomForestClassifier(n_estimators = 100)

# Training the model on the training dataset
# fit function is used to train the model using the training sets as parameters
clf.fit(X_train, y_train)

# performing predictions on the test dataset
y_pred = clf.predict(X_test)

# metrics are used to find accuracy or error
from sklearn import metrics
print()

# using metrics module for accuracy calculation
print("ACCURACY OF THE MODEL:", metrics.accuracy_score(y_test, y_pred))
```

Output: ACCURACY OF MODEL=0.98930254862

## **CHAPTER-2: TOOLS AND TECHNOLOGY USED**

### **2.1 Hardware Requirement**

- 4 GB RAM or above recommended.
- Recommended: SSD (Solid State Drive) for faster read/write speeds.
- Minimum: 2 GB of available disk space
- Processor (CPU):
  - **For Mac** : M1 or higher Chips.
  - **For Windows : Intel Core i5 or higher processor.**

### **2.2 Software Requirement**

- **Operating System** : Windows 7 or above / macOS 10.14 or later.
- **IDE** : Github Codespace
- Python
- OnlineFraud.CSV(data)

### **Python Library Used**

#### **Pandas**

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis/manipulation tool available in any language. It is already well on its way toward this goal.

pandas is well suited for many different kinds of data:

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet
- Ordered and unordered (not necessarily fixed-frequency) time series data.

- Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
- Any other form of observational / statistical data sets. The data need not be labeled at all to be placed into a pandas data structure

Here are just a few of the things that pandas does well:

- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let Series and DataFrame etc. automatically align the data for you in computations
- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
- Make it easy to convert aggregated, differently-indexed data in other Python and NumPy data structures into DataFrame objects
- Intelligent label-based slicing, fancy indexing and subsetting of large data sets
- Intuitive merging and joining data sets
- Flexible reshaping and pivoting of data sets
- Hierarchical labeling of axes (possible to have multiple labels per tick)
- Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving / loading data from the ultrafast HDF5 format
- Time series-specific functionality: date range generation and frequency conversion, moving window statistics, date shifting, and lagging.

## Matplotlib

Matplotlib is a powerful plotting library in Python used for creating static, animated, and interactive visualizations. Matplotlib's primary purpose is to provide users with the tools and functionality to represent data graphically, making it easier to analyze and understand. It was originally developed by John D. Hunter in 2003 and is now maintained by a large community of developers.

Key Features of Matplotlib:

1. Versatility: Matplotlib can generate a wide range of plots, including line plots, scatter plots, bar plots, histograms, pie charts, and more.



2.Customization: It offers extensive customization options to control every aspect of the plot, such as line styles, colors, markers, labels, and annotations.

3.Integration with NumPy: Matplotlib integrates seamlessly with NumPy, making it easy to plot data arrays directly.

4.Publication Quality: Matplotlib produces high-quality plots suitable for publication with fine-grained control over the plot aesthetics.

5.Extensible: Matplotlib is highly extensible, with a large ecosystem of add-on toolkits and extensions like Seaborn, Pandas plotting functions, and Basemap for geographical plotting.

6.Cross-Platform: It is platform-independent and can run on various operating systems, including Windows, macOS, and Linux.

7.Interactive Plots: Matplotlib supports interactive plotting through the use of widgets and event handling, enabling users to explore data dynamically.

### Different Types of Plots in Matplotlib

Matplotlib offers a wide range of plot types to suit various data visualization needs. Here are some of the most commonly used types of plots in Matplotlib:

- Line Graph
- Stem Plot
- Bar chart
- Histograms
- Scatter Plot
- Stack Plot
- Box Plot
- Pie Chart
- Error Plot
- Violin Plot
- 3D Plots

**Seaborn(sns)**

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on top matplotlib library and is also closely integrated with the data structures from pandas.

Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

**Different categories of plot in Seaborn**

Plots are basically used for visualizing the relationship between variables. Those variables can be either completely numerical or a category like a group, class, or division. Seaborn divides the plot into the below categories –

Relational plots: This plot is used to understand the relation between two variables.

Categorical plots: This plot deals with categorical variables and how they can be visualized.

Distribution plots: This plot is used for examining univariate and bivariate distributions

Regression plots: The regression plots in Seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.

Matrix plots: A matrix plot is an array of scatterplots.

Multi-plot grids: It is a useful approach to draw multiple instances of the same plot on different subsets of the dataset.

**Scikit-learn**

Scikit-learn has emerged as a powerful and user-friendly Python library. Its simplicity and versatility make it a better choice for both beginners and seasoned data scientists to build and implement machine learning models.

Scikit-learn is an open-source python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface. It is an open-source machine-learning library that provides a plethora of tools for various machine learning tasks such as classification, Regression, Clustering, and many more.

## 2.3 Github Codespace

A codespace is a development environment that's hosted in the cloud. You can customize your project for GitHub Codespaces by committing configuration files to your repository (often known as Configuration-as-Code), which creates a repeatable codespace configuration for all users of your project.

Each codespace you create is hosted by GitHub in a Docker container, running on a virtual machine. You can choose from a selection of virtual machine types, from 2 cores, 8 GB RAM, and 32 GB storage, up to 32 cores, 64 GB RAM, and 128 GB storage.

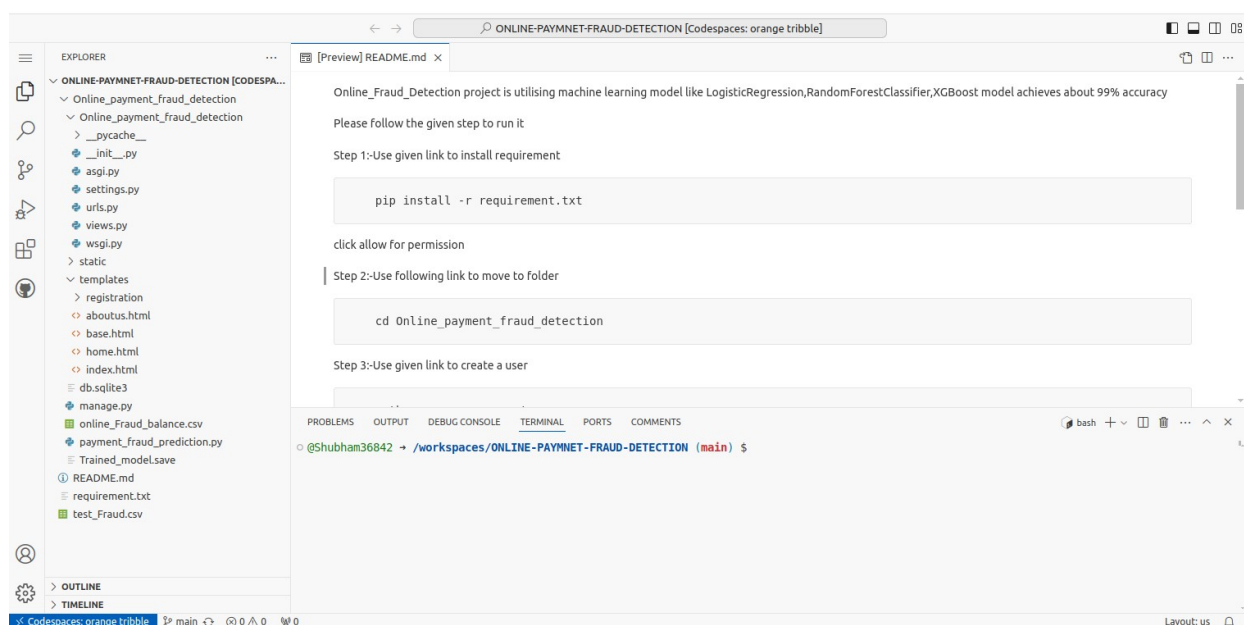
By default, the codespace development environment is created from an Ubuntu Linux image that includes a selection of popular languages and tools, but you can use an image based on a Linux distribution of your choice and configure it for your particular requirements. Regardless of your local operating system, your codespace will run in a Linux environment. Windows and macOS are not supported operating systems for the remote development container.

### Benefits of GitHub Codespaces

Reasons for choosing to work in a codespace include:

- Use a preconfigured development environment- You can work in a development environment that has been specifically configured for the repository. It will have all of the tools, languages, and configurations you need to work on that project. Everyone who works on that repository in a codespace will have the same environment. This reduces the likelihood of environment-related problems occurring and being difficult to debug. Each repository can have settings that will give contributors a ready-to-use, fit-for-purpose environment, and the environment on your local machine will be unchanged.
- Access the resources you need - Your local computer may not have the processing power, or storage space, you need to work on a project. GitHub Codespaces allows you to work remotely on a machine with adequate resources.

- Work anywhere - All you need is a web browser. You can work in a codespace on your own computer, on a friend's laptop, or on a tablet. Open your codespace and pick up from where you left off on a different device.
- Choose your editorn- Work in the browser in the VS Code web client, or choose from a selection of desktop-based applications.
- Work on multiple projects - You can use multiple codespaces to work on separate projects, or on different branches of the same repository, compartmentalizing your work to avoid changes made for one piece of work accidentally affecting something else you're working on.
- Pair program with a teammate - If you work on a codespace in VS Code, you can use Live Share to work collaboratively with other people on your team.
- Publish your web app from a codespace- Forward a port from your codespace and then share the URL, to allow teammates to try out the changes you've made to the application before you submit those changes in a pull request.
- Try out a framework- GitHub Codespaces reduces the setup time when you want to learn a new framework.



(Snapshot-i)

## **CHAPTER -3: TECHNICAL CONTENTS**

### **FRONTEND TECHNOLOGY**

#### **HTML**

HyperText Markup Language (HTML) is the standard markup Language for documents designed to be displayed in a web browser. It defines the content and structure of web content. It is often assisted by technologies such as Cascading Style Shee(CSS) and scripting language such as JavaScript.

Web browser receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page systematically and originally included cues for its appearance.

#### **CSS(Cascading style sheet)**

Cascading Style Sheets (CSS) is a style sheet language used for specifying the presentation and styling of a document written in a markup language such as HTML or XML(including XML dialects such as SVG,MathML or XHTML).CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accesibility provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

The name *cascading* comes from the specified priority scheme to determine which declaration applies if more than one declaration of a property match a particular element. This cascading priority scheme is predictable.

Online\_payment\_fraud\_detection > templates > home.html > div.form > form > table > tr > td

```

1  {% extends "base.html" %}
2  {% block home %}class="activ"{% endblock %}
3  {% block base %}
4  <div class="form">
5  <form method="GET">
6  <table>
7  <tr>
8  <td>
9      <label for="step"> Step:</label>
10 </td>
11 <td>
12     <input required type="number" name="step" id="" value="{{step}}">
13 </td>
14 <td>
15     <label for="name"> Type of transaction:</label>
16 </td>
17 <td><select name="transaction" id="transaction">
18     <option value="CASH_OUT">CASH_OUT</option>
19     <option value="CASH_IN">CASH_IN</option>
20     <option value="PAYMENT">PAYMENT</option>
21     <option value="TRANSFER">TRANSFER</option>
22 </select>
23 </td>
24 <td>
25     <label for="amount"> Enter the Amount:</label>
26 </td>
27 <td><input required type="number" name="amount" id="" value="{{amt}}">
28 </td>
29 </tr>
30 <tr>
31 <td>
32     <label for="Customerid">Customerid:</label>
33 </td>
34 <td><input required type="text" name="Customerid" value="{{cid}}">

```

Online\_payment\_fraud\_detection > templates > index.html > div#carouselExampleIndicators.carousel.slide > div.carousel-inner > div.carousel-item > div.co

```

1  {% extends "base.html" %}
2  {% block index %}class="activ"{% endblock %}
3  {% block base %}
4  <div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">
5  <div class="carousel-indicators">
6  <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active"
7  aria-current="true" aria-label="Slide 1"></button>
8  <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1"
9  aria-label="Slide 2"></button>
10 <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2"
11 aria-label="Slide 3"></button>
12 </div>
13 <div class="carousel-inner">
14 <div class="carousel-item active">
15 <div style="padding: 30px;" class="corousel">
16 
18 <div style="margin-top:40px;margin-right:100px;width:350px;text-align:center;">
19 <h4>
20 <b style="font-size:30px;">Welcome to our beautiful project which detect scams and adds se
21 to minimize the risk using <br><br> MACHINE LEARNING!!!</b>
22 </h4>
23 </div>
24 </div>
25 </div>
26 <div class="carousel-item">
27 <div style="padding: 30px;" class="corousel">
28 
31 <div style="margin-top:60px;margin-right: 40px;width: 280px;text-align: center;">
32 <h4>
33 <b style="font-size:22px;"> We are providing

```

## BACKEND TECHNOLOGY

### DJANGO

Django is a Python framework that makes it easier to create web sites using Python. Django takes care of the difficult stuff so that you can concentrate on building your web applications. Django emphasizes reusability of components, also referred to as DRY (Don't Repeat Yourself), and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).

Django follows the MVT design pattern (Model View Template).

- Model - The data you want to present, usually data from a database.
- View - A request handler that returns the relevant template and content - based on the request from the user.
- Template - A text file (like an HTML file) containing the layout of the web page, with logic on how to display the data


```
Online_payment_fraud_detection > Online_payment_fraud_detection > views.py
1  from django.http import HttpResponseRedirect
2  from django.shortcuts import render, redirect
3  import pandas as pd
4  import joblib
5  models=joblib.load('Trained_model.save')
6  def homepage(request):
7      try:
8          if request.method == "GET":
9              step = request.GET.get('step')
10             txn = request.GET.get('transaction')
11             amt = int(request.GET.get('amount'))
12             cid = request.GET.get('Customerid')
13             rid = request.GET.get('Recipientid')
14             nb = int(request.GET.get('New Balance'))
15             ob = int(request.GET.get('Old Balance'))
16             rnb = int(request.GET.get('Recipient New Balance'))
17             rob = int(request.GET.get('Recipient Old Balance'))
18             cash_out=debit=payment=transfer=0
19             if txn=='CASH_OUT':
20                 cash_out=1
21             elif txn=='CASH_IN':
22                 debit=1
23             elif txn=='PAYMENT':
24                 payment=1
25             else:
26                 transfer=1
27             prediction=[]
28             test = pd.DataFrame(data=[[step,amt,ob,nb,rob,rnb,cash_out,debit,payment,transfer]],columns=['step', '
29             prediction.append(models[0].predict(test))
30             prediction.append(models[2].predict(test))
31             prediction.append(models[3].predict(test))
32             if prediction.count(0)>prediction.count(1):
33                 print("NoFraud")
34             else:
35                 print("Fraud")
```

Online\_payment\_fraud\_detection > Online\_payment\_fraud\_detection >  urls.py

```

1  """
2  URL configuration for Online_payment_fraud_detection project.
3
4  The `urlpatterns` list routes URLs to views. For more information please see:
5  |   https://docs.djangoproject.com/en/5.0/topics/http/urls/
6  |   Examples:
7  |   Function views
8  |       1. Add an import:  from my_app import views
9  |       2. Add a URL to urlpatterns:  path('', views.home, name='home')
10 |   Class-based views
11 |       1. Add an import:  from other_app.views import Home
12 |       2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
13 |   Including another URLconf
14 |       1. Import the include() function: from django.urls import include, path
15 |       2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
16 |   """
17
18 from django.contrib import admin
19 from django.urls import path , include
20 from Online_payment_fraud_detection import views
21
22 urlpatterns = [
23     path("admin/", admin.site.urls),
24     path("aboutus",views.aboutuspage),
25     path("accounts/",include("django.contrib.auth.urls")),
26     path("",views.indexpage,name='index'),
27     path("home",views.homepage),
28     path("hiw",views.hiwpage),
29 ]
30

```

Online\_payment\_fraud\_detection > Online\_payment\_fraud\_detection >  settings.py

```

25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     "django.contrib.admin",
35     "django.contrib.auth",
36     "django.contrib.contenttypes",
37     "django.contrib.sessions",
38     "django.contrib.messages",
39     "django.contrib.staticfiles",
40 ]
41
42 MIDDLEWARE = [
43     "django.middleware.security.SecurityMiddleware",
44     "django.contrib.sessions.middleware.SessionMiddleware",
45     "django.middleware.common.CommonMiddleware",
46     "django.middleware.csrf.CsrfViewMiddleware",
47     "django.contrib.auth.middleware.AuthenticationMiddleware",
48     "django.contrib.messages.middleware.MessageMiddleware",
49     "django.middleware.clickjacking.XFrameOptionsMiddleware",
50 ]
51
52 ROOT_URLCONF = "Online_payment_fraud_detection.urls"
53
54 TEMPLATES = [
55     {
56         "BACKEND": "django.template.backends.django.DjangoTemplates",
57         "DIRS": [BASE_DIR, "templates"],
58     },
59 ]

```



## CHAPTER-4: TRAINING AND IMPLEMENTATION OF MACHINE MODEL

### 1.Importing Libraries and Reading dataframe

```
[ ] #importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Reading dataframe
url=('https://raw.githubusercontent.com/someshyadav9404/ONLINE-PAYMNET-FRAUD-DETECTION/main/Online_payment_fraud_detection/online_Fraud_balance.csv')
data=pd.read_csv(url,index_col=0)
data.head()
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrg	nameDest	oldbalanceDest	newbalanceDest	isFraud
993448	45	CASH_OUT	36467.43	C798504793	218603.0	182135.57	C1761667313	430274.32	466741.75	0
1712685	160	CASH_OUT	11177.72	C1298707190	0.0	0.00	C498599199	12264876.20	12276053.91	0
3868318	283	CASH_IN	76605.92	C1493839655	2663.0	79268.92	C461216810	0.00	0.00	0
4118275	302	PAYMENT	8909.38	C631523702	6220.0	0.00	M1102374371	0.00	0.00	0
3951753	287	TRANSFER	223730.40	C734008057	223730.4	0.00	C1763626980	0.00	0.00	1

```
[ ] data.sample(10)
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrg	nameDest	oldbalanceDest	newbalanceDest	isFraud
2207350	186	CASH_IN	143367.61	C334387716	64197.00	207564.61	C1445557695	993006.68	849639.07	0
1030556	72	CASH_OUT	622235.32	C615309889	622235.32	0.00	C755984599	3377968.96	4000204.28	1
6019701	456	CASH_OUT	4017972.88	C826677509	4017972.88	0.00	C1483869567	0.00	4017972.88	1

### 2. Performing EDA(Exploratory data analysis)

```
[ ] # checking dataframe
data.shape
```

```
(16000, 10)
```

```
# checking missing values
data.isnull().sum()
```

```
step      0
type      0
amount    0
nameOrig  0
oldbalanceOrg  0
newbalanceOrg  0
nameDest  0
oldbalanceDest  0
newbalanceDest  0
isFraud   0
dtype: int64
```

No missing values

```
[ ] data.duplicated().sum()
```

```
0
```

No duplicate value present

```
obj = (data.dtypes == 'object')
object_cols = list(obj[obj].index)
print("Categorical variables:", len(object_cols))
```

```
int_ = (data.dtypes == 'int')
num_cols = list(int_[int_].index)
print("Integer variables:", len(num_cols))
```

```
fl = (data.dtypes == 'float')
fl_cols = list(fl[fl].index)
print("Float variables:", len(fl_cols))
```

```
Categorical variables: 3
Integer variables: 2
Float variables: 5
```

```
# checking for number of values unique in coloumn
data.nunique()
```

```
step      726
type      5
amount    11875
nameOrig  16000
oldbalanceOrg  9147
newbalanceOrg  3574
nameDest  15845
oldbalanceDest  7447
newbalanceDest  8990
isFraud    2
dtype: int64
```

### ✓ Checking for unique values

```
data['type'].value_counts()
```

```
type
CASH_OUT    6811
TRANSFER    4717
PAYMENT     2600
CASH_IN     1822
DEBIT        50
Name: count, dtype: int64
```

```
data['nameDest'].value_counts()
```

```
nameDest
C1988180949    3
C1443408255    3
C1980653895    3
C1812798312    2
C123020135     2
..
M852513508     1
C1293124691    1
C195893407     1
C573200870     1
M647033739     1
Name: count, Length: 15845, dtype: int64
```

```
data['step'].value_counts()
```

```
step
212    80
205    80
355    80
15     78
235    78
..
28      4
54      4
662     2
112     2
545     1
Name: count, Length: 726, dtype: int64
```

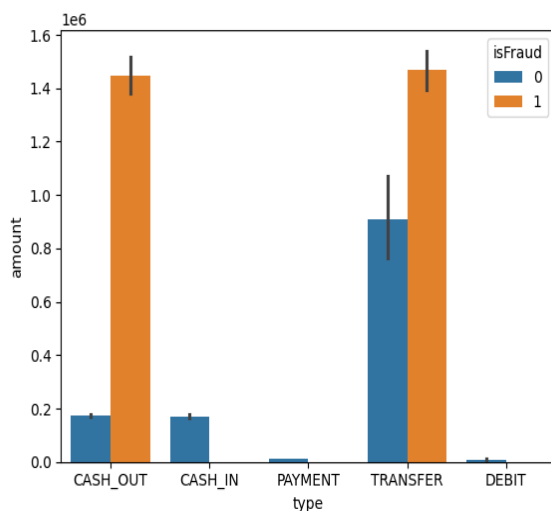
```
data['isFraud'].value_counts()
```

```
isFraud
0    8000
1    8000
Name: count, dtype: int64
```

### 3. Data Visualisation(creating graph to analyse data)-using seaborn and matplotlib

```
sns.barplot(x='type', y='amount', data=data, hue='isFraud')
```

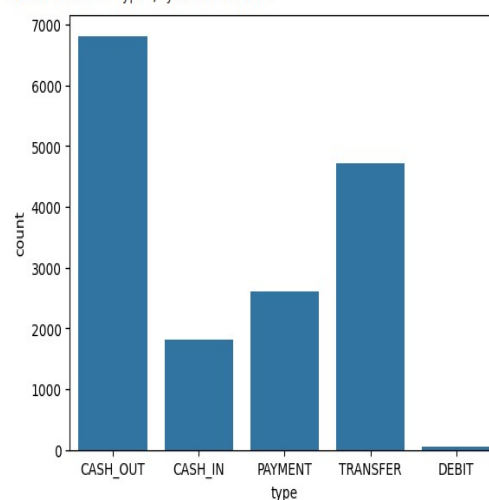
```
<Axes: xlabel='type', ylabel='amount'>
```



Fraud mostly occur in CASH\_OUT and TRANSFER

```
sns.countplot(x='type', data=data)
```

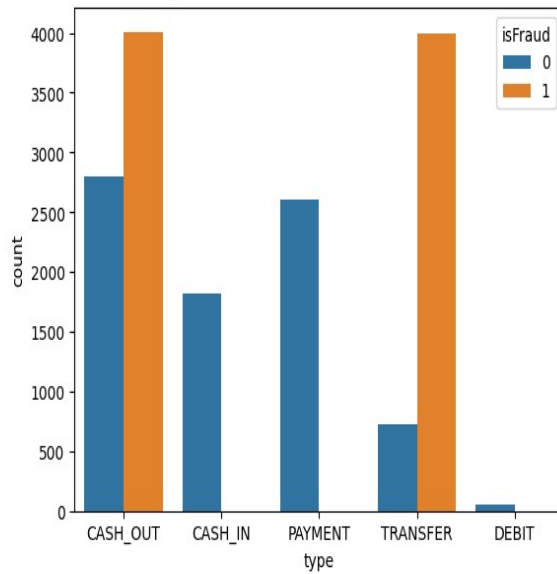
```
<Axes: xlabel='type', ylabel='count'>
```



Most of the Time CASH\_OUT is taking place

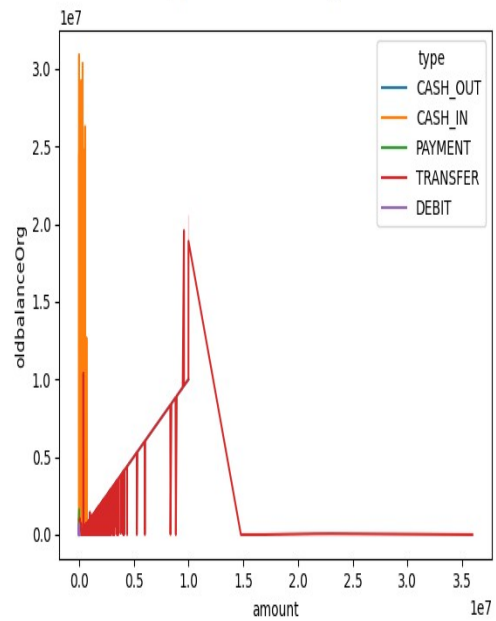
```
sns.countplot(x='type', data=data, hue='isFraud')
```

<Axes: xlabel='type', ylabel='count'>



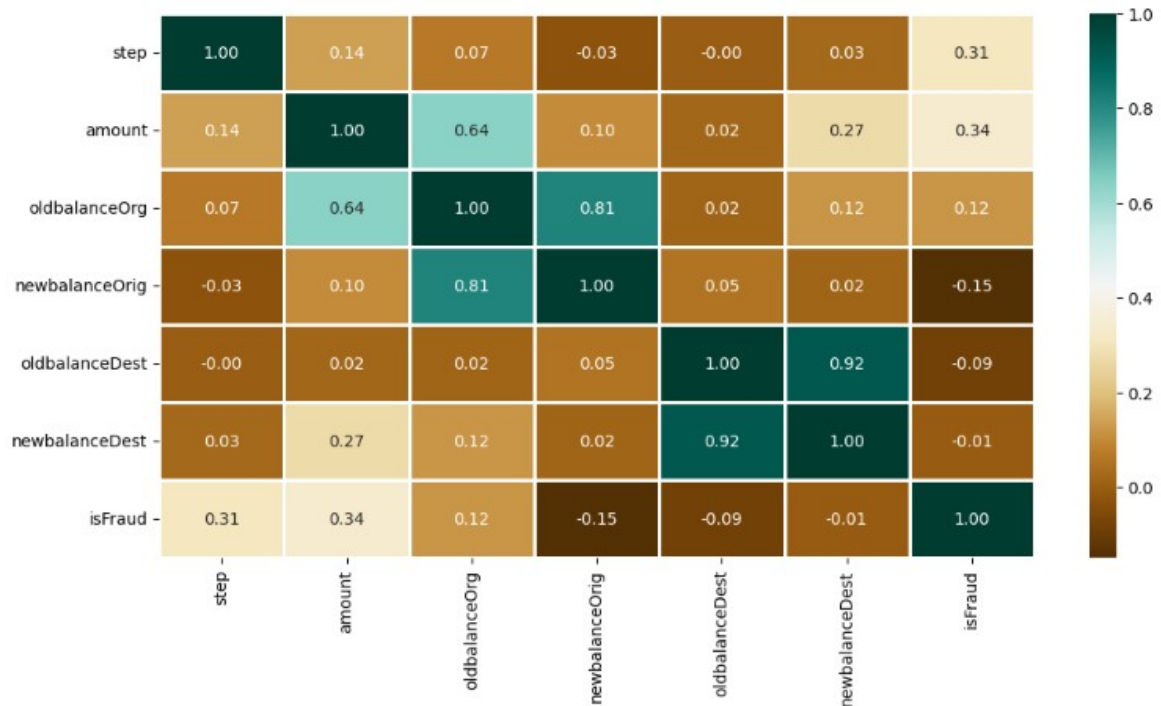
```
sns.lineplot(data=data, x='amount', y='oldbalanceOrg', hue='type')
```

<Axes: xlabel='amount', ylabel='oldbalanceOrg'>



```
plt.figure(figsize=(12, 6))
sns.heatmap(data1.corr(),
            cmap='BrBG',
            fmt='.2f',
            linewidths=2,
            annot=True)
```

<Axes: >



Coorelation among the features

#### 4. Encoding and dropping irrelevant features

```
# one hot encoding
type_new = pd.get_dummies(data['type'], drop_first=True)
data_new = pd.concat([data, type_new], axis=1)
data_new.head()
```

id	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	CASH_OUT	DEBIT	PAYMENT	TRANSFER
993448	45	CASH_OUT	36467.43	C798504793	218603.0	182135.57	C1761667313	430274.32	466741.75	0	True	False	False	False
1712685	160	CASH_OUT	11177.72	C1298707190	0.0	0.00	C498599199	12264876.20	12276053.91	0	True	False	False	False
3868318	283	CASH_IN	76605.92	C1493839655	2663.0	79268.92	C461216810	0.00	0.00	0	False	False	False	False
4118275	302	PAYMENT	8909.38	C631523702	6220.0	0.00	M1102374371	0.00	0.00	0	False	False	True	False
3951753	287	TRANSFER	223730.40	C734008057	223730.4	0.00	C1763626980	0.00	0.00	1	False	False	False	True

```
[ ] # dropping irrelevant columns
X = data_new.drop(['isFraud', 'type', 'nameOrig', 'nameDest'], axis=1)
y = data_new['isFraud']
```

```
[ ] # checking datashape
X.shape, y.shape
```

```
((16000, 10), (16000,))
```


```
[ ] # checking training features
X.columns
```

```
Index(['step', 'amount', 'oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest',  
      'newbalanceDest', 'CASH_OUT', 'DEBIT', 'PAYMENT', 'TRANSFER'],  
      dtype='object')
```

#### 5. Preparing Train, Test, Validation data for the model(Using sklearn train-test split)

```
[ ] #Performing test train split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42)
```

```
X_train.head()
```



	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	CASH_OUT	DEBIT	PAYMENT	TRANSFER	
	2739088	212	175903.88	641217.37	817121.26	640996.9	465093.02	False	False	False	False
	977960	44	29143.29	0.00	0.00	380362.2	409505.49	True	False	False	False
	1030394	61	1372301.24	1372301.24	0.00	0.0	1372301.24	True	False	False	False
	6362388	724	72389.42	72389.42	0.00	0.0	0.00	False	False	False	True
	5922160	404	4686.11	116373.00	111686.89	0.0	0.00	False	False	True	False

```
[ ] y_train.head()
```

```
2739088    0
977960     0
1030394    1
6362388    1
5922160    0
Name: isFraud, dtype: int64
```

#### NOTE:-

Train set:- This include dataset used during model training

Test set:- This include dataset used for evaluating the model

Validation set:- This include dataset used for validation of machine model

## 6. Preparing model and train it using Logistic regression Xgboost,SVC,Random forest

```

models = [LogisticRegression(), XGBClassifier(),
          SVC(kernel='rbf', probability=True),
          RandomForestClassifier(n_estimators=7,
                               criterion='entropy',
                               random_state=7)]

for i in range(len(models)):
    models[i].fit(X_train, y_train)
    print(f'{models[i]} : ')

    train_preds = models[i].predict_proba(X_train)[: , 1]
    print('Training Accuracy : ', ras(y_train, train_preds))

    y_preds = models[i].predict_proba(X_test)[: , 1]
    print('Validation Accuracy : ', ras(y_test, y_preds))
    print()

```

```

LogisticRegression() :
Training Accuracy :  0.9620545870257757
Validation Accuracy :  0.9615349620600442

XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, random_state=None, ...) :
Training Accuracy :  1.0
Validation Accuracy :  0.9989847371985205

SVC(probability=True) :
Training Accuracy :  0.9601104145208248
Validation Accuracy :  0.9591378861280409

RandomForestClassifier(criterion='entropy', n_estimators=7, random_state=7) :
Training Accuracy :  0.9999931279915559
Validation Accuracy :  0.9966494764807002

```

## 7. Save the model to used where it required

```

[ ] import joblib

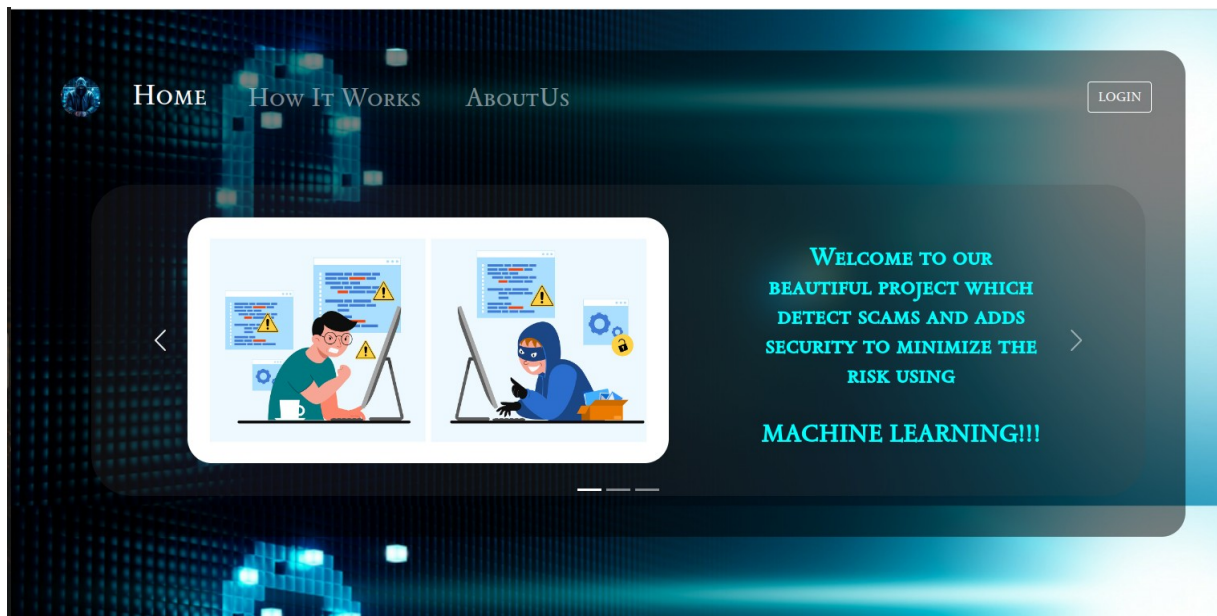
# save model with joblib
filename = 'joblib_model.save'
joblib.dump(models, filename)

['joblib_model.save']

```



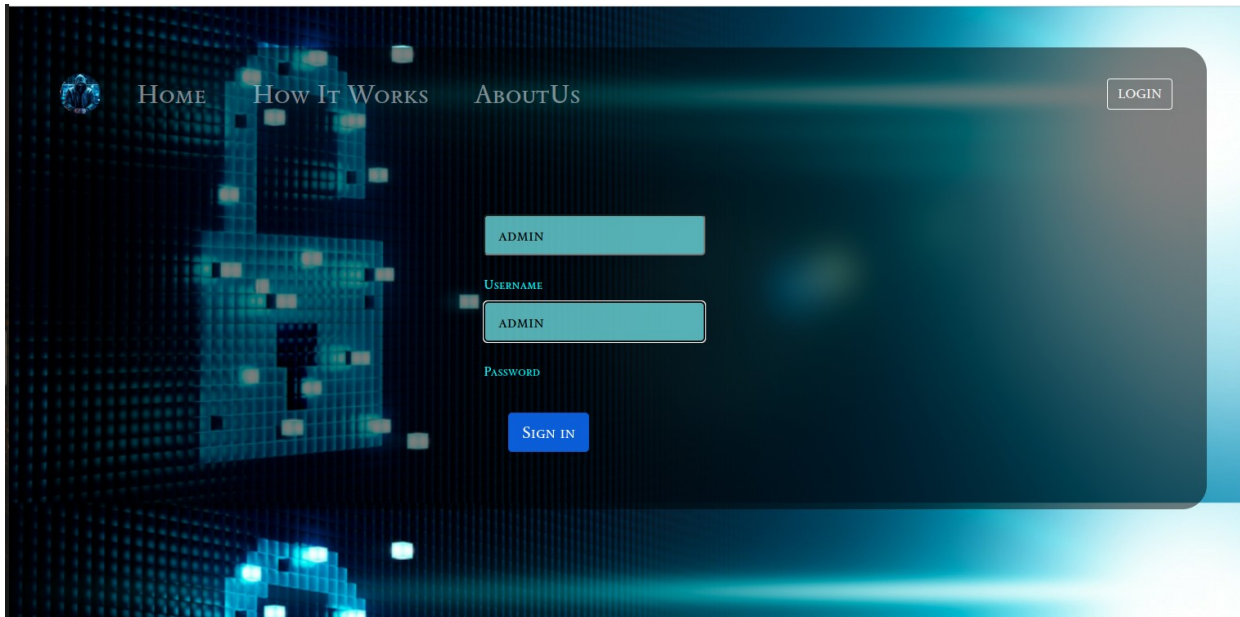
## SNAPSHOTS



(Snapshot-ii)  
(Home Page)

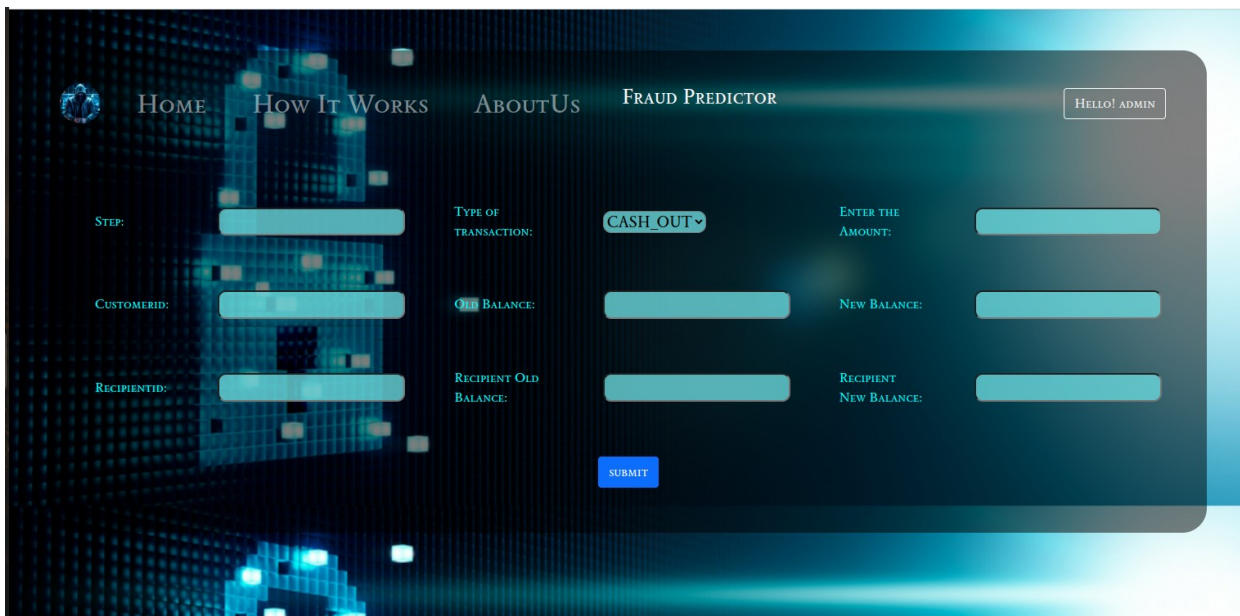


(Snapshot-iii)  
(Login Option)



The screenshot shows a login page with a dark blue background featuring a pixelated Bitcoin logo. The navigation bar at the top includes links for HOME, How It Works, and ABOUT US, along with a LOGIN button. The login form contains three input fields: ADMIN (for username), ADMIN (for password), and a SIGN IN button.

(Snapshot-iv)  
(Login Page)



The screenshot shows the Fraud Predictor page with a dark blue background and a pixelated Bitcoin logo. The navigation bar includes links for HOME, How It Works, ABOUT US, and FRAUD PREDICTOR, along with a HELLO! ADMIN button. The form contains several input fields and a dropdown menu for transaction type. The fields are organized into three rows: STEP, CUSTOMERID, and RECIPIENTID. The transaction type is set to CASH\_OUT. The form also includes fields for OLD BALANCE, NEW BALANCE, RECIPIENT OLD BALANCE, and RECIPIENT NEW BALANCE, along with a SUBMIT button.

(Snapshot-v)  
(Fraud Predictor page)

The screenshot shows the 'FRAUD PREDICTOR' interface. A green modal box displays 'HELLO! ADMIN' and 'YOUR INPUT DATA IS VALID'. The form fields are as follows:

Field	Value
STEP:	348
CUSTOMERID:	C508176191
RECIPIENTID:	C1002089279
TYPE OF TRANSACTION	CASH OUT
ENTER THE AMOUNT:	184155.52
OLD BALANCE:	11713081.45
RECIPIENT OLD BALANCE:	588112.9
RECIPIENT NEW BALANCE:	588112.9

A blue 'SUBMIT' button is located at the bottom center.

(Snapshot-vi)  
(Validating valid data)

The screenshot shows the 'FRAUD PREDICTOR' interface. A red modal box displays 'HELLO! ADMIN' and 'YOUR INPUT DATA IS INVALID'. The form fields are as follows:

Field	Value
STEP:	734
CUSTOMERID:	C1891630790
RECIPIENTID:	C2124431731
TYPE OF TRANSACTION	CASH OUT
ENTER THE AMOUNT:	10000000.0
OLD BALANCE:	1810044.85
RECIPIENT OLD BALANCE:	0.0
RECIPIENT NEW BALANCE:	0.0

A blue 'SUBMIT' button is located at the bottom center.

(Snapshot-vii)  
(Validating Fraud data)



## **CONCLUSION**

The development of our online fraud detection project marks a significant stride in combating digital fraud and safeguarding online transactions. Through meticulous analysis of user behavior patterns, machine learning algorithms, and real-time monitoring, we have created a robust system capable of swiftly identifying and mitigating fraudulent activities.

Our project not only enhances security measures but also minimizes financial losses and protects the integrity of online platforms. By leveraging advanced technology and proactive detection techniques, we empower businesses and consumers alike to navigate the digital landscape with confidence.

As we move forward, continuous refinement and adaptation will be essential to stay ahead of evolving fraudulent tactics. With ongoing collaboration and innovation, we can further strengthen our defenses and cultivate trust in the digital ecosystem.

Together, we can forge a safer and more secure online environment for all.

## **REFERENCES**

- [www.wikipedia.com](http://www.wikipedia.com)
- [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
- [www.kaggle.com](http://www.kaggle.com)
- <https://archive.ics.uci.edu/>
- [www.javapoint.com](http://www.javapoint.com)
- [www.w3school.com](http://www.w3school.com)
- [www.sanfoundry.com](http://www.sanfoundry.com)