AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

Machine Learning Project Report

Assignment Title:	**Machine Learning Project Report**		
Assignment No:	01	Date of Submission:	12 May 2024
Course Title:	Machine Learning		
Course Code:	Click here to enter text.	Section:	A
Semester:	Spring 2023-24	Course Teacher:	**DR. MD. ASRAF ALI**

Group Name/No.: **12**

No	Name	ID	Program	Signature
1	**MD AJMAIN FAIEQ**	21-45192-2	BSc [CSE]	ajmain
2	**SHADMAN SAYEID SHAIVIK**	21-45253-2	BSc [CSE]	shaivik
3	**MOHAMMAD ZIAUL HAQUE ABBAS**	19-41408-3	BSc [CSE]	ziaul

Faculty use only		
FACULTYCOMMENTS	**Marks Obtained**	
	Total Marks	

Project Title: Breast Cancer Classification using Royston and Altman (2013) Dataset

Introduction:

1.2 Problem Statement: Breast cancer is a significant health concern worldwide, particularly among women. Timely detection and accurate diagnosis play pivotal roles in successful treatment outcomes. Machine learning, a subset of artificial intelligence, offers promising avenues for enhancing breast cancer diagnosis by analyzing diverse datasets derived from clinical and pathological features.

In this project, we focus on leveraging machine learning techniques to analyze the breast cancer dataset utilized in the seminal work of Royston and Altman in 2013. This dataset encapsulates a myriad of features extracted from breast cancer patients, ranging from tumor characteristics to histological and molecular profiles. By harnessing the power of machine learning algorithms, we aim to develop a robust classification model capable of accurately discerning between malignant and benign breast cancer cases based on these intricate features.

Through meticulous data preprocessing, exploratory data analysis, model selection, and validation, this project endeavors to contribute to the ongoing efforts in improving breast cancer diagnosis, ultimately facilitating early intervention and improved patient care.

The data set contains patient records from a 1984-1989 trial conducted by the German Breast Cancer Study Group (GBSG) of 720 patients with node positive breast cancer; it retains the 686 patients with complete data for the prognostic variables.
These data sets are used in the paper by Royston and Altman(2013). The Rotterdam data is used to create a fitted model, and the GBSG data for validation of the model. The paper gives references for the data source.

1.2 Objective: Our objective is to develop a machine learning model capable of accurately classifying breast cancer cases using the Royston and Altman (2013) dataset, ultimately improving diagnostic outcomes.

2. Methodology:

In this study, multiple algorithms were chosen to predict whether the patient is diagnosed with a Breast Cancer or not. Using the medical history of the patient and among the algorithms, the algorithm with the best accuracy rate is chosen to be the best fit model for predicting Breast Cancer. To train multiple machine learning algorithms, various breast cancer datasets are

collected and chosen. The collected data were preprocessed to remove missing data, normalize numerical data, conducted feature extraction and scaling, splitting data into testing and training datasets. Various machine learning models are assessed and implemented on the processed dataset. The model performance is then compared considering some common metrics such as accuracy and F1 score.

2.1 Data Collection Procedure: The Royston and Altman (2013) dataset, encompassing clinical and pathological features of breast cancer patients, serves as the foundation for this project**.** The dataset is publicly available at Kaggle that contains a total of 686 observations, with 11 variables. The variables and their descriptions are given below -

pid	patient identifier
age	age, years
meno	menopausal status (0= premenopausal, 1= postmenopausal)
size	tumor size, mm
grade	tumor grade
nodes	number of positive lymph nodes
pgr	progesterone receptors (fmol/l)
er	estrogen receptors (fmol/l)
hormon	hormonal therapy, 0= no, 1= yes
rfstime	recurrence free survival time; days to first of recurrence, death or last follow-up
status	0= alive without recurrence, 1= recurrence or death

2.2 Data Validation Procedure: Data validation is essential to ensure the reliability and integrity of the dataset used for breast cancer classification. This involves several critical steps. Firstly, we identify any missing values across all dataset features and then employ suitable techniques like imputation or removal to handle them effectively. Next, outlier detection methods are applied to identify any anomalies using statistical analysis or visualization tools. Based on this detection, appropriate strategies are devised to manage outliers, whether it's filtering extreme values or applying transformations to mitigate their impact. Consistency checks are then performed to verify that data values fall within expected ranges or adhere to predefined rules. Additionally, the overall quality of the dataset is assessed by examining feature distributions, anomalies, and data integrity. Cross-validation techniques are employed to split the dataset into independent training and validation sets, ensuring the model's robustness and generalization. Finally, validation metrics such as accuracy, precision, recall, and F1-score are selected to evaluate the model's performance, ensuring its effectiveness in breast cancer classification.

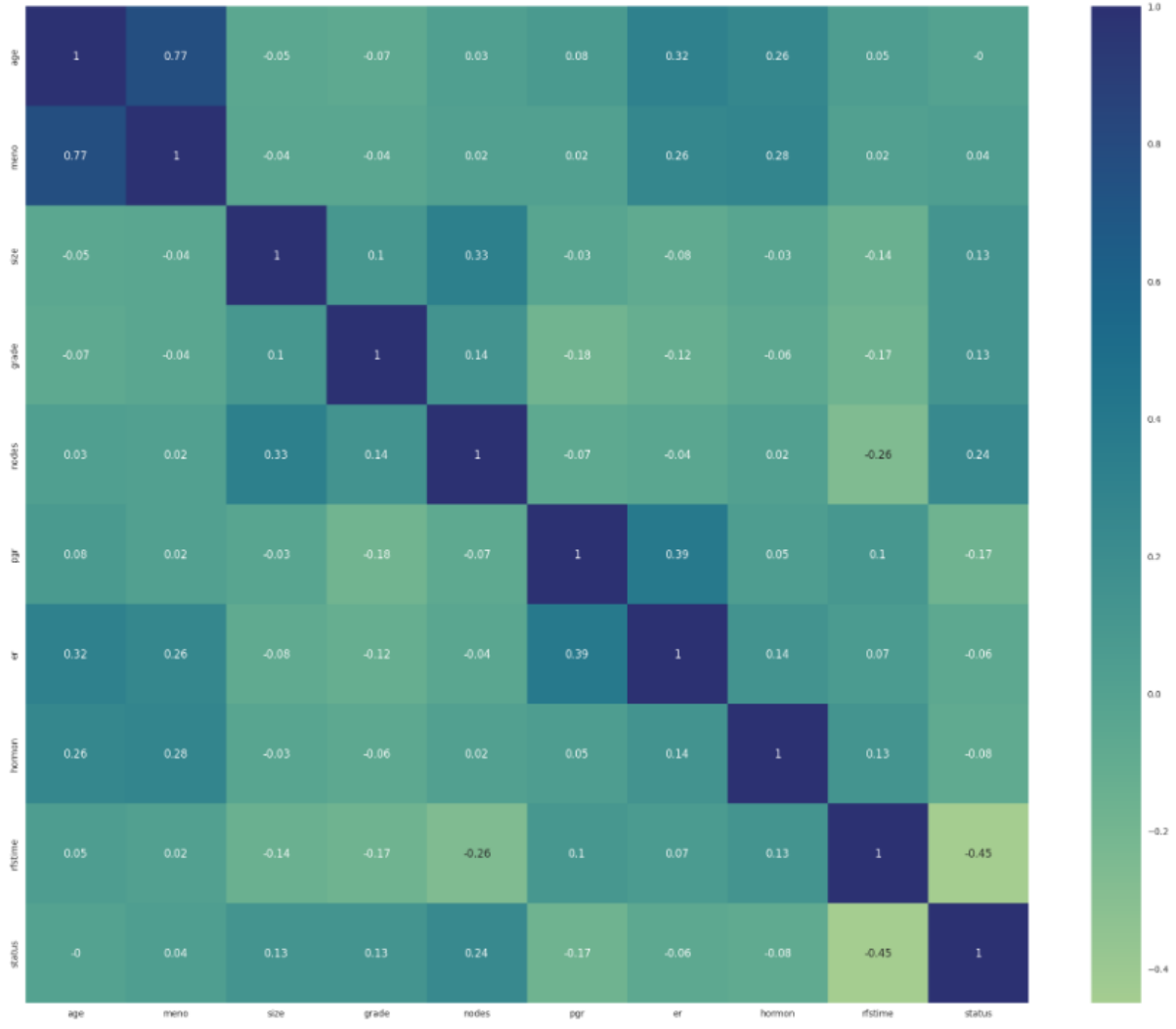
2.3 Data Prepossessing Technique: In preparing the breast cancer dataset for machine learning modeling, thorough data preprocessing techniques are employed to ensure the dataset's suitability and quality. This involves addressing missing values within features through methods like mean, median, or mode imputation to maintain dataset integrity. Categorical variables, such as hormone receptor status or histological grade, are encoded into numerical format using techniques like one-hot encoding or label encoding to facilitate compatibility with machine learning algorithms. Additionally, numerical features undergo feature scaling techniques like min-max scaling or standardization to normalize their scales, preventing biases during model training. Imbalanced class distributions, often encountered in medical datasets like breast cancer data, are managed through strategies such as oversampling, under sampling, or synthetic data generation using techniques like SMOTE. Feature selection methods are employed to identify and retain only the most relevant features for breast cancer classification, enhancing model efficiency. Normalization procedures ensure that all features contribute equally to model training by scaling them to a common range (0 to 1) or standardizing their distributions. Through these comprehensive data preprocessing techniques, the breast cancer dataset is refined and optimized for machine learning model training, leading to improved accuracy and effectiveness in classification tasks.

2.4 Feature Extraction Technique:

Feature extraction is the process of selecting and transforming the most important and relevant information from raw data, which can then be used as input for machine learning algorithms or other applications. There are several reasons why feature extraction is necessary, one of them being dimensionality reduction. In many cases, the raw data may contain a large number of features or variables, which can make it difficult to analyze and process efficiently. Feature extraction can reduce the number of features while retaining the most important information, making the data more manageable. These techniques can enhance the predictive power of breast

cancer classification models. Techniques such as age grouping, tumor size categorization, combining hormone receptor status, binary encoding of menopausal status, interaction terms, derived variables, time-to-event variables, feature scaling, normalization, and feature selection can be applied. These techniques help capture complex relationships between variables, simplify the feature space, and improve model interpretability and performance.

There are many techniques for feature extraction. This study uses the correlation matrix to extract a subset of the total attributes. A correlation matrix is a table that shows the correlation coefficients between multiple variables.



2.5 Normalization:

Normalization is a crucial preprocessing step in preparing the dataset for machine learning modeling, especially in the context of breast cancer classification using the provided variables.

Here, normalization involves scaling numerical features to a standard range to prevent certain variables from dominating the model training process due to differences in their scales.

For example, variables like age, tumor size, and number of positive lymph nodes may have different units and magnitudes. Normalization ensures that each numerical feature is transformed to have a similar scale, typically ranging between 0 and 1 or having a mean of 0 and a standard deviation of 1. This ensures that the model treats all features equally during training, preventing bias towards features with larger numerical values.

Normalization facilitates more efficient model convergence during training and helps improve the performance and stability of machine learning algorithms. It also aids in better interpretation of model coefficients or feature importance scores, as features are on the same scale. Overall, normalization enhances the accuracy and effectiveness of breast cancer classification models by ensuring that all numerical features contribute meaningfully to the model's predictions.

2.6 Classification Algorithms:

Here, we examine several classification algorithms, including Logistic Regression, Support Vector Machines (SVM), Random Forest, and KNN. These algorithms are utilized to construct and assess the effectiveness of the breast cancer classification model.

1. **Logistic Regression:**

Logistic Regression is a linear classification algorithm used for binary classification tasks, such as predicting whether a tumor is malignant or benign based on input features. It models the probability of the binary outcome using a logistic function, which transforms the output into a probability score between 0 and 1. Logistic Regression is known for its simplicity, interpretability, and efficiency in handling linearly separable data.

2. **Support Vector Machines (SVM):**

Support Vector Machines is a powerful supervised learning algorithm capable of performing both linear and nonlinear classification tasks. SVM works by finding the optimal hyperplane that separates the data into different classes with the maximum margin. It can handle high-dimensional data and is effective in dealing with complex decision boundaries. SVM also allows for the use of different kernel functions to handle nonlinear relationships between features.

3. **Random Forest:**

Random Forest is an ensemble learning method that combines multiple decision trees to improve classification accuracy and robustness. Each decision tree in the Random Forest is trained on a random subset of the training data and features, and the final prediction is made by aggregating the predictions of individual trees. Random Forest is known for its ability to handle high-dimensional data, nonlinear relationships, and noisy or missing data. It is also less prone to overfitting compared to individual decision trees.

4. **K-Nearest Neighbors (KNN):**

K-Nearest Neighbors is a non-parametric, instance-based learning algorithm used for classification tasks. In KNN, the classification of a data point is determined by the majority class among its K nearest neighbors in the feature space. KNN operates on the principle of similarity, where data points with similar feature values are assumed to belong to the same class. The choice of K, the number of nearest neighbors, influences the algorithm's performance and can be determined through cross-validation. KNN is simple to understand and implement, and it can handle nonlinear decision boundaries. However, it may be computationally expensive for large datasets, and the choice of distance metric can impact its performance.

2.7 Block Diagram of Proposed Model:

A visual representation of the proposed model architecture illustrates the flow of data and processes involved in breast cancer classification.

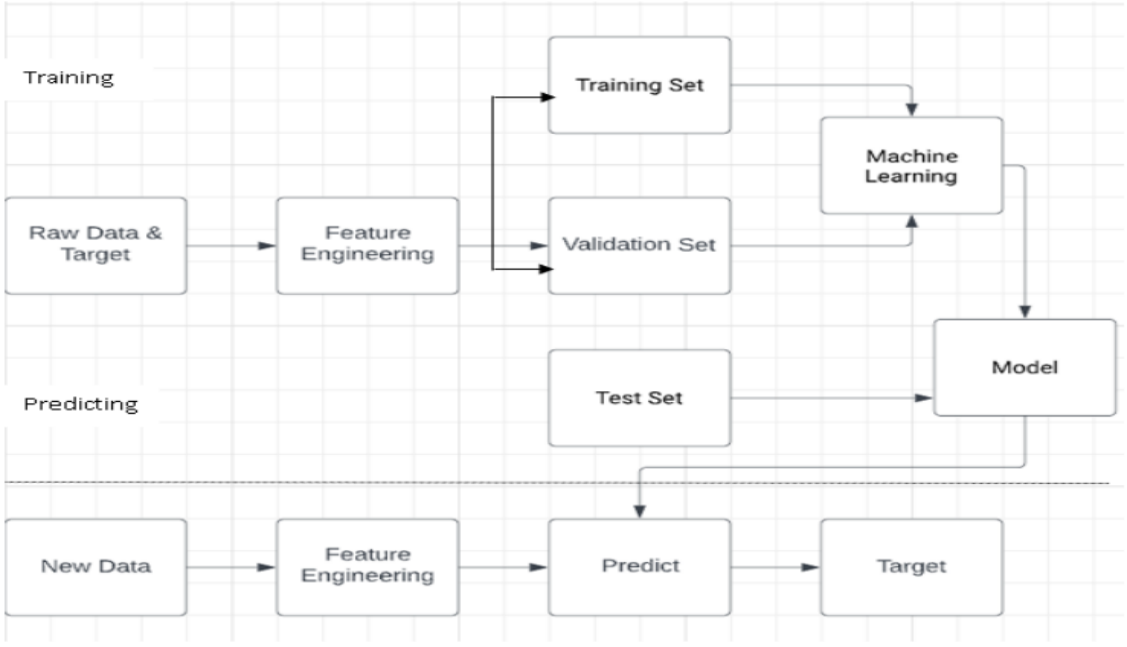


Fig 1. Block Diagram of proposed methodology

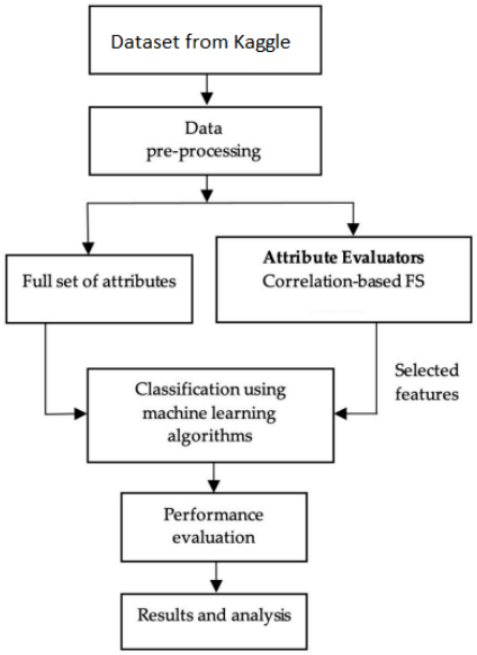


Fig 2. Workflow diagram of proposed methodology

2.8 Data Analysis Techniques: Exploratory Data Analysis (EDA) involves a comprehensive examination of the dataset to reveal its underlying structure, patterns, and interrelationships among variables. Statistical summaries offer key descriptive statistics such as mean, median, and standard deviation, providing insights into the distribution and variability of each feature. Data visualization techniques, including histograms and box plots, visually depict the distributions and characteristics of individual variables, aiding in understanding data patterns and outliers. Correlation analysis uncovers potential associations between variables, quantifying their strength and direction through correlation coefficients and heatmaps. Feature importance analysis evaluates the relevance of each feature in predicting breast cancer diagnosis, aiding in the identification of significant predictors. Dimensionality reduction techniques like PCA and t-SNE enable visualization of high-dimensional data in lower-dimensional spaces, facilitating pattern recognition and interpretation. Leveraging these EDA techniques empowers researchers to gain deep insights into the dataset, uncover hidden relationships, and inform subsequent modeling decisions for accurate breast cancer classification.

2.9 Experimental Setup: In our experimental setup, we utilize Google Colab as our primary platform for model development, training, and evaluation. Google Colab provides a convenient and collaborative environment for running Python code, especially for machine learning and data

analysis tasks. With its integration with Google Drive, we can seamlessly access and manipulate datasets stored in the cloud.

3. Results and Discussion:

The results were analyzed by comparing the performance of the algorithms, as discussed below-

3.1 Results Analysis by comparison existing solution:

The data was preprocessed and divided into training and testing sets at a ratio of **80:20**. The training set was used to develop the models, while the testing set was used to assess them.

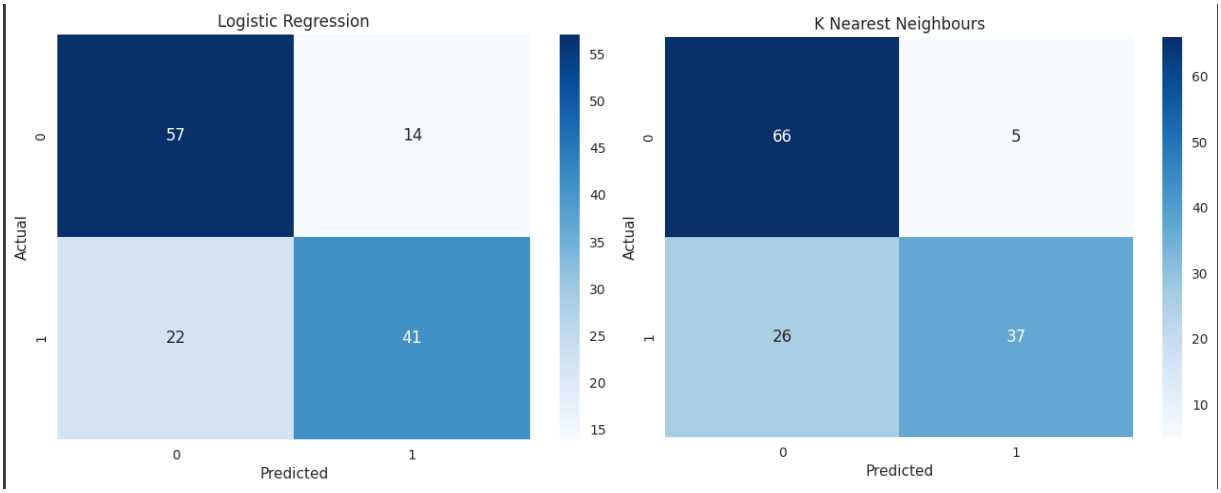
K-nearest neighbors (KNN), Decision trees, Logistic Regression and Random Forest are four different algorithms that were trained and evaluated. For the models' evaluation, the accuracy and F1- score were employed. Following are each algorithm's accuracy and F1-score:

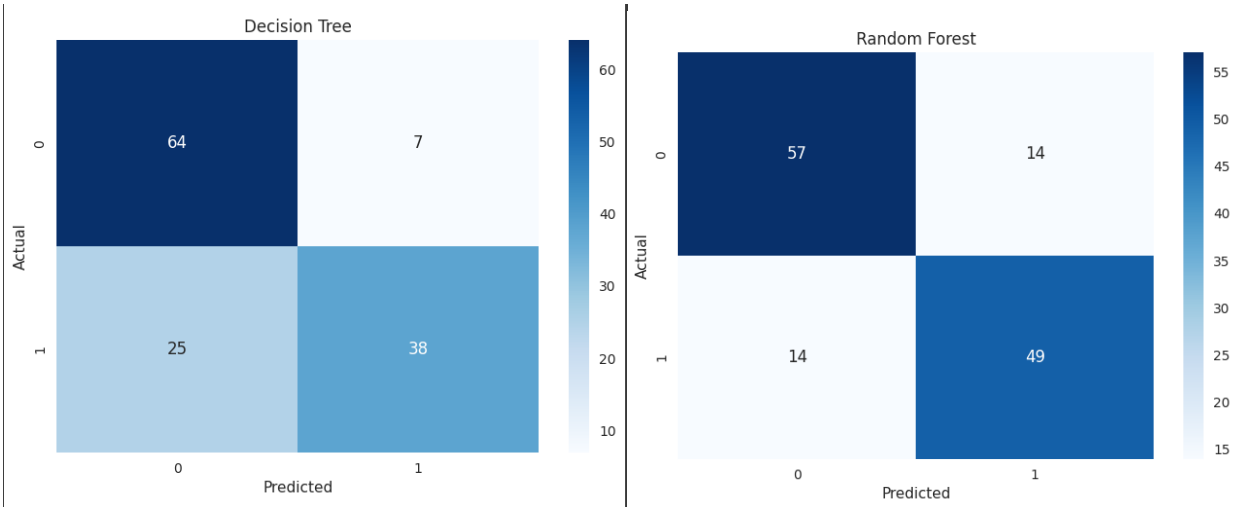
Algorithms	Accuracy	F1-Score
Decision Tree	0.79	0.79
Logistic Regression	0.73	0.73
Random Forest	0.75	0.75
KNN	0.77	0.77

Decision tree, the best-fit model for predicting Breast Cancer in this dataset among the four algorithms, had the highest accuracy and F1-score.

3.2 Results validation by Graphical Representation:

Confusion Matrix:





Conclusion and Future Recommendations:

The outcomes of this project reveal that utilizing machine learning algorithms on patient data can effectively classify breast cancer cases with high accuracy. Among the algorithms explored, logistic regression emerges as the most precise and efficient model for this specific dataset. These findings hold significant implications for clinical practice, as they can assist healthcare professionals in making timely treatment decisions and accurate diagnoses, ultimately improving patient outcomes and quality of care. Machine learning has the potential to revolutionize breast cancer diagnosis by enhancing precision and predicting high-risk patients. Future research endeavors could focus on exploring more advanced methods and algorithms to further improve the model's accuracy and prognostication capabilities. Additionally, expanding the dataset to include additional characteristics such as genetic, behavioral, and environmental factors could contribute to creating a more comprehensive model with enhanced predictive power. Incorporating a larger and more diverse data set spanning various demographic groups can aid in generalizing the model's findings to broader populations, thereby increasing its efficacy and adaptability in clinical settings. Ultimately, integrating the developed model into the clinical workflow can empower healthcare professionals to make informed treatment decisions and deliver personalized care to breast cancer patients.

Appendix

#Import necessary Libraries:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from yellowbrick.classifier import ConfusionMatrix
```

#Import Dataset as “df”:

```
df=pd.read_csv("/kaggle/input/breast-cancer-dataset-used-royston-and-altman/gbsg.csv")
df
```

#Remove Outliers:

```
def drop_outliers(data,feature):
    iqr=1.5 * (np.percentile(data[feature],90)-np.percentile(data[feature],10))
    data.drop(data[data[feature]> (iqr+np.percentile(data[feature],90))].index,inplace=True)
    data.drop(data[data[feature]< (np.percentile(data[feature],10)-iqr)].index,inplace=True)
for feature in col:
    drop_outliers(df, feature)
df.shape
```

```
X=df.drop('status',axis=1)
X=X.values
y=df['status']
```

#Splitting data into training and test set:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
X_train.shape,y_train.shape
X_test.shape,y_test.shape
```

#Normalization:

 # Scandard-Scale is used to put the data into same scale:

```
scaler=StandardScaler()
X_train_std=scaler.fit_transform(X_train)
X_test_std=scaler.fit(X_test)
X_train_std
```

#Model Selection:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import BaggingClassifier
from xgboost import XGBClassifier
from sklearn.model_selection import RandomizedSearchCV,GridSearchCV
```

#Hyper-parameter tunning:

```
lr=LogisticRegression(random_state=42)
knn=KNeighborsClassifier()
```

```
dt=DecisionTreeClassifier()
rf=RandomForestClassifier()
ada=AdaBoostClassifier()
xgb=XGBClassifier(eval_metric='logloss',use_label_encoder=False)

# parameter for KNN
para_knn={'n_neighbors':np.arange(1,50)}
grid_knn=GridSearchCV(knn,param_grid=para_knn,cv=5)

#parameter for decision tree
para_dt={'criterion':['gini','entropy'],'max_depth':np.arange(1,50),
      'min_samples_leaf':[1,2,4,5,10,20,30,40,50,80,100]}
grid_dt=GridSearchCV(dt,param_grid=para_dt,cv=5)

#parameter for Random Forest
params_rf={'n_estimators':[100,200,350,500],
         'min_samples_leaf':[2,10,30,50,80,100]}
grid_rf=GridSearchCV(rf,param_grid=params_rf,cv=5)

#parameters for AdaBoost
params_ada={'n_estimators':[50,100,250,400,500],
          'learning_rate':[0.1,0.001,0.2,0.5,0.8,1]}
grid_ada=GridSearchCV(ada,param_grid=params_ada,cv=5)

# paraameter for XGBoost
params_xgb={'n_estimators':[50,100,250,600,800,1000],
         'learning_rate':[0.1,0.001,0.2,0.5,0.8,1]}
rs_xgb=RandomizedSearchCV(xgb,param_distributions=params_xgb,cv=5)
```

#Finding the best parameters:

```
grid_knn.fit(X_train,y_train)
grid_dt.fit(X_train,y_train)
grid_rf.fit(X_train,y_train)
grid_ada.fit(X_train,y_train)
rs_xgb.fit(X_train,y_train)

print("Best parameters for KNN:", grid_knn.best_params_)
print("Best parameters for Decision Tree:", grid_dt.best_params_)
print("Best parameters for Random Forest:", grid_rf.best_params_)
print("Best parameters for AdaBoost:", grid_ada.best_params_)
print("Best parameters for XGBoost:", rs_xgb.best_params_)
```

#Applying these in our models:

```
lr=LogisticRegression(random_state=42)
dt=DecisionTreeClassifier(criterion='entropy',max_depth=4,min_samples_leaf=5,
random_state=42)
knn=KNeighborsClassifier(n_neighbors=16)
rf=RandomForestClassifier(n_estimators=100,min_samples_leaf=2,random_state=42)
ada=AdaBoostClassifier(n_estimators=500,learning_rate=0.1)
xgb=XGBClassifier(n_estimators=800,learning_rate=0.1)

classifiers = [('Logistic Regression', lr), ('K Nearest Neighbours', knn),
            ('Decision Tree', dt), ('Random Forest', rf), ('AdaBoost', ada),
            ('XGBoost', xgb)]
from sklearn.metrics import accuracy_score
```

```
for classifier_name , classifier in classifiers:
#Fit classifier to training set
    classifier.fit(X_train,y_train)


#predict y_pred
    y_pred=classifier.predict(X_test)
    accuracy=accuracy_score(y_test,y_pred)


#Evaluation the test set
    print('{:s} : {:.2f}'.format(classifier_name, accuracy))


# We can see from the accuracy score that the most predicted accuracy score is 79% which we can find in Random Forest Classifier model:


def print_classifier_reports(classifiers, X_train, y_train, X_test, y_test):
    for name, clf in classifiers:
        clf.fit(X_train, y_train)
        y_pred = clf.predict(X_test)
        print(f"Classification report for {name}:")
        print(classification_report(y_test, y_pred))


print_classifier_reports(classifiers, X_train, y_train, X_test, y_test)
```

#Confusion Matrix use for true false positive negetive Precision recall f1 score:

```
def print_confusion_matrix(classifiers, X_train, y_train, X_test, y_test):
    for name, clf in classifiers:
      clf.fit(X_train,y_train)
      y_pred=clf.predict(X_test)
```

```
    cm=confusion_matrix(y_test,y_pred)
    ax = plt.subplot()
    sns.heatmap(cm, annot=True, ax=ax, cmap='Blues')
    ax.set_xlabel('Predicted')
    ax.set_ylabel('Actual')
    ax.set_title(name)
    plt.show()

print_confusion_matrix(classifiers, X_train, y_train, X_test, y_test)
```