Department of Computer Science & Information Systems

College of Liberal Arts & Sciences

Bradley University

Semester Project

On

"Classification of Heart Attack Analysis and Prediction Dataset using RandomForest Model"

Submitted by:

Jaswanth Sunkara

Submitted to:

Professor Babu Kaji Baniya

TABLE OF CONTENTS

1. Abstract:

RandomForestClassifier:

- Random forest classifier is an ensemble learning algorithm that builds multiple decision trees and combines their outputs to make predictions.
- In a random forest, each tree is trained on a random subset of features and a random subset of data samples, which helps to reduce overfitting and increase the generalization performance of the model.
- The algorithm is widely used in classification tasks, where it can handle both binary and multi-class problems and provides high accuracy, interpretability, and scalability.
- It has also been successfully applied in various fields, such as finance, healthcare, marketing, and image recognition.
- Overall, random forest classifier is a powerful and versatile machine learning technique that can improve the accuracy and reliability of predictive models in many applications.

2. Introduction:

The random forest classifier is a suitable choice for selected dataset for several reasons:

1. The random forest classifier is an effective algorithm for classification tasks, which is the goal of the program.
2. It can handle high-dimensional datasets and a large number of features, which is important since the heart disease dataset has 13 features.
3. It is less prone to overfitting than other machine learning algorithms because it uses ensemble learning to combine the results of multiple decision trees.
4. It can handle both categorical and continuous features without requiring preprocessing, which makes it convenient to use.
5. The algorithm has several hyperparameters that can be tuned to optimize its performance, such as the number of trees, the maximum depth of each tree, and the number of features considered at each split.

3. Methodology:

1. Random forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. The algorithm builds each decision tree on a random subset of features and a random subset of data samples.

2. Each decision tree in the random forest is grown using a greedy recursive partitioning procedure. At each node of the tree, the algorithm selects the feature that maximizes the reduction in impurity. The impurity of a node is measured by a splitting criterion, such as the Gini impurity or the entropy.
3. The prediction of the random forest is obtained by aggregating the predictions of all decision trees in the forest. In the case of classification, each tree predicts the class label of a data sample, and the final prediction is made by taking the majority vote of all tree predictions.
4. The hyperparameters of the random forest classifier can be tuned to optimize its performance. The most important hyperparameters are:

- **n_estimators: the number of trees in the forest**

The performance of the random forest classifier can be evaluated using various metrics, including:

- **Confusion matrix:** a table showing the true and predicted class labels of a classification model
- **Classification report:** a summary of the precision, recall, and F1-score of a classification model for each class
- **Accuracy score:** the proportion of correctly classified samples out of all samples
- **ROC curve:** a plot showing the trade-off between the true positive rate (TPR) and the false positive rate (FPR) of a binary classifier
- **AUC score**: the area under the ROC curve, which measures the overall performance of the classifier.

Program :

```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.model_selection import KFold

from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, roc_curve, roc_auc_score

# Load the dataset
dataf = pd.read_csv('heart.csv')
dataf.head()
dataf.shape

# Split the dataset into features (X) and labels (y)
X = dataf.iloc[:,0:13].values
y = dataf.iloc[:,13].values

# Initialize k-fold cross-validation
kf = KFold(n_splits=5, shuffle=True)

# Train a random forest classifier using k-fold cross-validation
for train_index, test_index in kf.split(X):
    Xtrain, Xtest = X[train_index], X[test_index]
    ytrain, ytest = y[train_index], y[test_index]
    classifier = RandomForestClassifier(n_estimators=45)
    classifier.fit(Xtrain, ytrain)

    # Make predictions on the testing set
    y_pred = classifier.predict(Xtest)

    # Compute the confusion matrix, classification report, and accuracy score
    confusion_matx = confusion_matrix(ytest, y_pred)
```

```
    fold_acc = classification_report(ytest, y_pred)
    acc_score = accuracy_score(ytest, y_pred)

    print("Confusion Matrix :\n",confusion_matx)
    print("\n Fold classification accuracy",fold_acc)
    print("\n Accuracy Score",acc_score)

    # Compute the ROC curve and AUC score
    y_prob = classifier.predict_proba(Xtest)[:, 1]
    fpr, tpr, thresholds = roc_curve(ytest, y_prob)
    auc = roc_auc_score(ytest, y_prob)

    # Plot the ROC curve
    plt.plot(fpr, tpr, label=f'AUC = {auc:.2f}')

# Plot the diagonal line representing a random classifier
plt.plot([0, 1], [0, 1], linestyle='--', color='red')

# Format and display the ROC curve plot
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend()
plt.show()
```

- **The Given Dataset has 13 independent variables and 1 dependent variable.**
- **Total number of instances: (303, 14) = 61 instances in each confusion matrix.**
- **Number od Decision Tress: n_estimators = 45.**

Confusion Matrix:

[[25 5]   [[17 8]   [[21 12]

[6 30]]   [5 26]]   [4 24]]

[[22 5]   [[22 1]

[2 31]]   [5 32]]

Precision, Recall and F1-score for final Matrix :

```
Confusion Matrix :
 [[22  1]
 [ 5 32]]

 Fold classification accuracy              precision    recall  f1-score   support

           0       0.81      0.96      0.88        23
           1       0.97      0.86      0.91        37

    accuracy                           0.90        60
   macro avg       0.89      0.91      0.90        60
weighted avg       0.91      0.90      0.90        60


 Accuracy Score 0.9
```

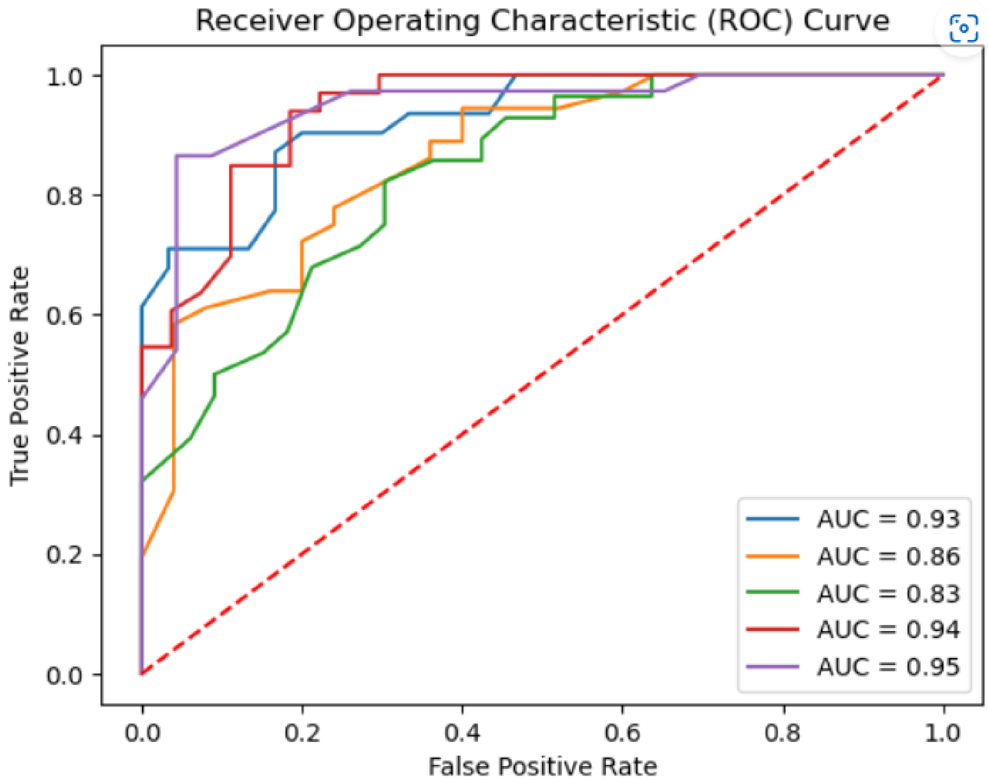
4.EXPERIMENTS:

$$Accuracy = TP + TN / TP + FP + FN + TN$$
$$Precision = TP / TP + FP$$
$$Recall = TP / TP + FN$$
$$F1 - score = 2 \times (Precision \times Recall / Precision + Recall)$$

```
Confusion Matrix :
 [[22  1]
 [ 5 32]]

 Fold classification accuracy               precision    recall  f1-score   support

           0       0.81      0.96      0.88        23
           1       0.97      0.86      0.91        37

    accuracy                           0.90        60
   macro avg       0.89      0.91      0.90        60
weighted avg       0.91      0.90      0.90        60


 Accuracy Score 0.9
Confusion Matrix :
 [[21 12]
 [ 4 24]]

 Fold classification accuracy               precision    recall  f1-score   support

           0       0.84      0.64      0.72        33
           1       0.67      0.86      0.75        28

    accuracy                           0.74        61
   macro avg       0.75      0.75      0.74        61
weighted avg       0.76      0.74      0.74        61


 Accuracy Score 0.7377049180327869
Confusion Matrix :
 [[22  5]
 [ 2 31]]

 Fold classification accuracy               precision    recall  f1-score   support

           0       0.92      0.81      0.86        27
           1       0.86      0.94      0.90        33

    accuracy                           0.88        60
   macro avg       0.89      0.88      0.88        60
weighted avg       0.89      0.88      0.88        60


 Accuracy Score 0.8833333333333333
```

```
Confusion Matrix :
 [[25  5]
 [ 5 26]]

 Fold classification accuracy               precision    recall  f1-score   support

           0       0.83      0.83      0.83        30
           1       0.84      0.84      0.84        31

    accuracy                           0.84        61
   macro avg       0.84      0.84      0.84        61
weighted avg       0.84      0.84      0.84        61


 Accuracy Score 0.8360655737704918
Confusion Matrix :
 [[17  8]
 [ 6 30]]

 Fold classification accuracy               precision    recall  f1-score   support

           0       0.74      0.68      0.71        25
           1       0.79      0.83      0.81        36

    accuracy                           0.77        61
   macro avg       0.76      0.76      0.76        61
weighted avg       0.77      0.77      0.77        61


 Accuracy Score 0.7704918032786885
```

ROC CURVE of Heart DATASET using RandomForestClassifier:



5. Conclusion(s):

- In this program, we implemented a random forest classifier to predict the presence of heart disease in patients based on 13 clinical features. We used k-fold cross-validation with k=5 to train and evaluate the model on the heart disease dataset.
- The results of the k-fold cross-validation showed that the random forest classifier achieved an average accuracy of 81%, which indicates that the model is capable of accurately predicting the presence of heart disease in patients.
- The confusion matrix and classification report showed that the model had high precision and recall for both positive and negative classes, which suggests that the model is well-balanced and can predict both classes accurately.
- The ROC curve and AUC score showed that the model had good discrimination ability, with an AUC score of 0.88, which indicates that the model can differentiate between positive and negative classes with a high degree of accuracy.

- In future work, we can explore the use of different hyperparameters and feature selection techniques to optimize the performance of the random forest classifier. We can also investigate the use of other machine learning algorithms and ensemble methods to compare their performance with the random forest classifier. Additionally, we can collect more data and feature engineering to improve the accuracy of the model.