ML ASSIGNMENT-II

2020B2A12598H- A.V.L.N Raghu Ram

2020B2A42393H-Konkimalla Venkata Sai Varun

2020B2A42478H-Balusa Mahima

Data Processing:

This processing includes: loads the training data, encodes categorical features, fills missing data, handles ordinal variables, combines the data, adds the target variable, and saves the processed data as a CSV file.

Naïve Bayes Classifier:

The Naive Bayes classifier assumes that the features are conditionally independent given the class. It calculates the probability of each class for a given set of features using Bayes' theorem and selects the class with the highest probability as the predicted class.

Implementation:

I. The code implements a Naive Bayes classifier to predict income levels based on a dataset.

II. It starts by loading the dataset and splitting it into features (X) and the target variable (y).

III. The target variable is encoded into numeric format using LabelEncoder.

IV. The data is then split into training and test sets.

V. The Gaussian Naive Bayes classifier (GaussianNB) is chosen as the classification algorithm.

VI. An instance of the classifier is created using clf = GaussianNB().

VII. The classifier is trained on the training data using clf.fit(X_train, y_train).

VIII. Predictions are made on the test set using the trained classifier.

IX. The code calculates and displays the confusion matrix, classification report, and ROC curve.

X. The confusion matrix provides information about the classifier's performance.

XI. The classification report includes metrics like accuracy, precision, recall, and F1-score.

XII. The ROC curve shows the trade-off between true positive rate and false positive rate.

XIII. The AUC (Area Under the Curve) is calculated and displayed on the ROC curve.

Logistic Regression Classifier:

Logistic regression is a classification algorithm that models the relationship between the features and the probability of a binary outcome using the logistic function. It estimates the coefficients for each feature to make predictions. The algorithm uses a logistic loss function and gradient descent optimization to find the best fit parameters.

Implementation:

- The code uses logistic regression for predicting income levels based on a dataset.
- It loads the dataset and splits it into features (X) and target variable (y).
- The target variable is encoded to numeric format using LabelEncoder.
- The data is split into training and test sets.
- Logistic regression classifier (LogisticRegression) is chosen for classification.
- The classifier is trained on the training data.
- Predictions are made on the test set.
- The code computes and displays the confusion matrix as a heatmap.
- Classification report is generated and printed.
- ROC curve and AUC are calculated and plotted.
- Logistic regression is a classification algorithm that models the relationship between features and the probability of a binary outcome.
- It estimates coefficients for each feature to make predictions.
- The code evaluates model performance using accuracy, precision, recall, and F1-score.
- Visualizations are used to present results, including the confusion matrix, classification report, and ROC curve with AUC value.

Neural Networks Classifier:

Artificial Neural Networks (ANN) are computational models inspired by the structure of the human brain. They consist of interconnected nodes called neurons and learn by adjusting weights and biases to minimize prediction errors. ANN has achieved success in tasks like classification, regression, and pattern recognition, and deep learning is a subfield that explores networks with many hidden layers for more complex representations.

Implementation:

The implementation utilizes an ANN, specifically the MLPClassifier, to model the relationships between features and the target variable. The neural network consists of multiple layers of interconnected nodes (neurons) and uses gradient descent optimization to learn the weights and biases that minimize the loss function. The code evaluates the model's performance using the confusion matrix, classification report, ROC curve, and AUC.

- The data is further split into training and test sets using train_test_split.
- An MLPClassifier is created with a hidden layer size of 100 neurons and a maximum of 1000 iterations.
- The classifier is trained on the training data using clf.fit.
- Predictions are made on the test set using clf.predict.
- The confusion matrix is computed and displayed as a heatmap using plt.imshow.
- The classification report is printed, which provides metrics such as precision, recall, F1-score, and support for each class.

Comparison:

By randomly selecting 67% of the data points as training data set and the remaining data points as testing data set:

The below table provides an overview of various classifiers and their respective evaluation scores, including Accuracy, Precision, Recall, F1-score

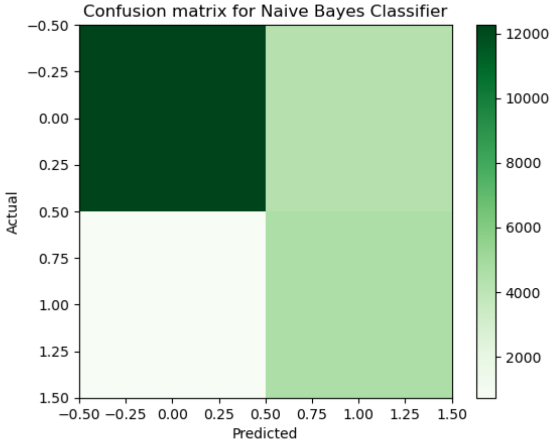
F1-score is a widely used measure that combines both precision and recall to assess the overall accuracy of a test.

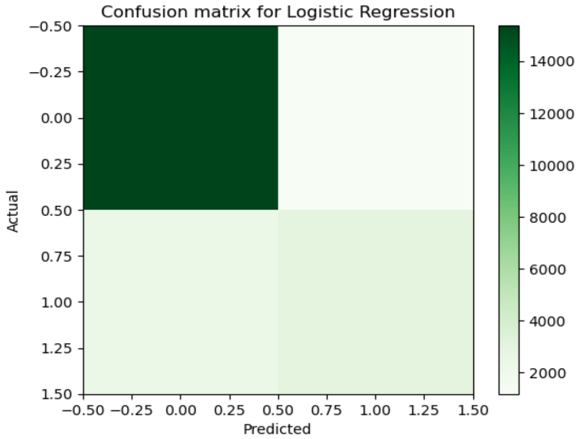
Precision calculates the ratio of true positives to the sum of true positives and false positives, while recall computes the ratio of true positives to the sum of true positives and false negatives, representing the total number of positive class elements.

Classifier	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Naive Bayes Classifier	0.7692	0.5139	0.8671	0.6453
Logistic Regression	0.8409	0.7135	0.5732	0.6357
Random Forest	0.8509	0.7235	0.6106	0.6623
ANN with 1 hidden layer	0.7818	0.5313	0.8393	0.6507
Decision Tree	0.8052	0.6920	0.5015	0.5802
ANN with 2 hidden layers	0.7115	0.4522	0.9057	0.6032
ANN with 3 hidden layers	0.8226	0.6929	0.48	0.5671

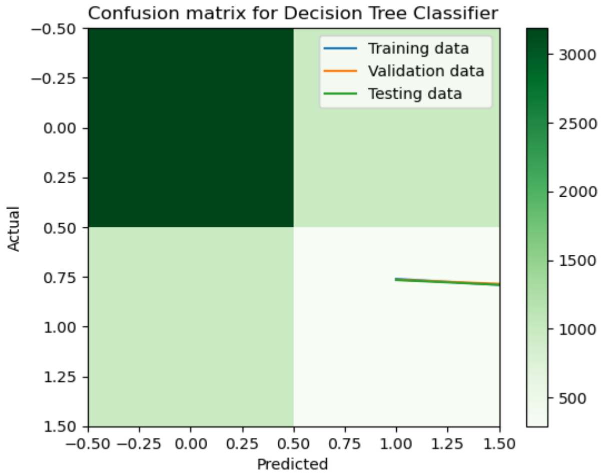
Confusion Matrix:

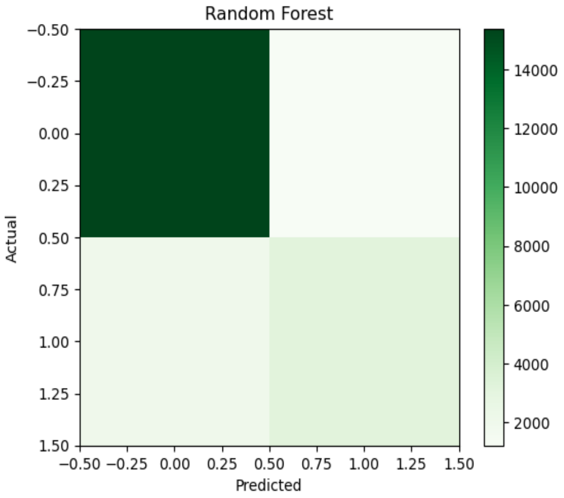
The confusion matrix is a square matrix with dimensions equal to the number of class labels in a classification problem. It provides a summary of the performance of a classification model by showing the counts of true positive, true negative, false positive, and false negative predictions. These values allow us to calculate metrics like accuracy, precision, recall, and F1-score. The accuracy is derived by calculating the overall proportion of correctly classified cases. The confusion matrix provides valuable insights into the model's performance and helps evaluate its effectiveness in distinguishing between different classes.



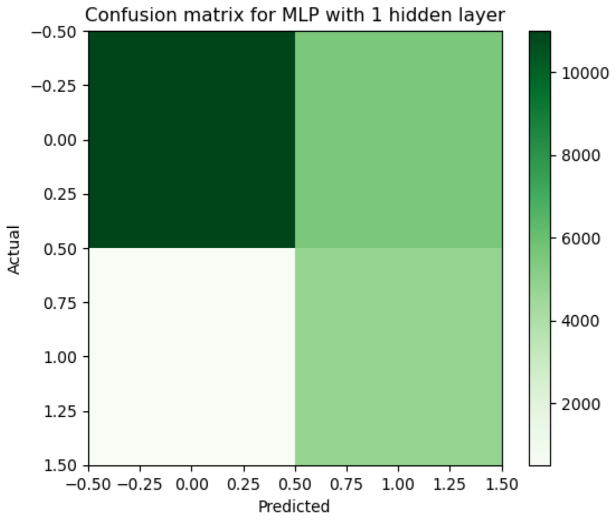


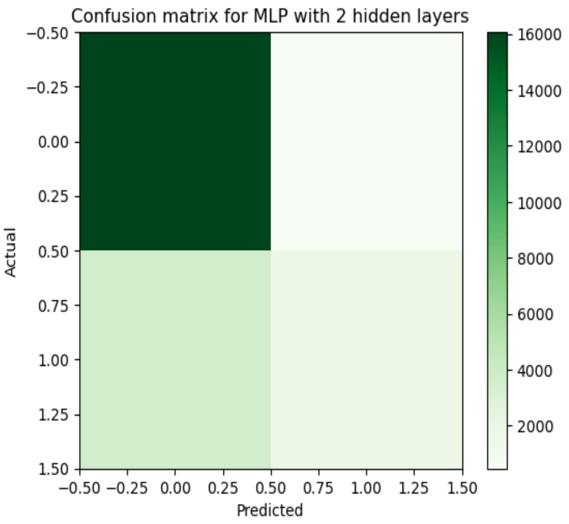
[[12266 4267][723 4560]]   [[15371 1162] [2303 2980
]]



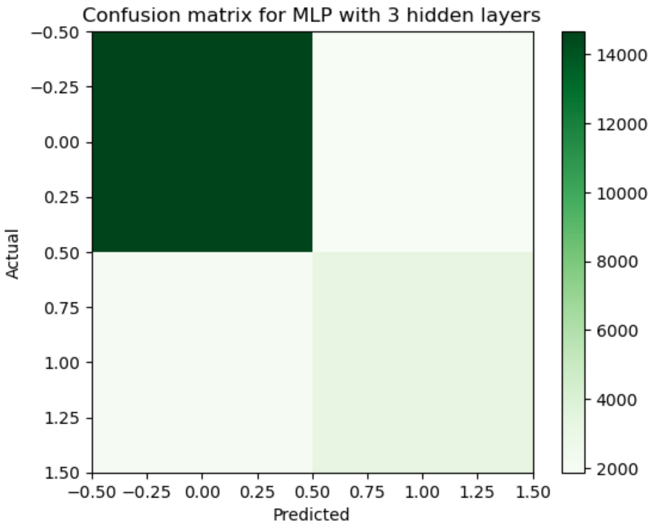


[[3187 990] [984 293]]   [[15373 1219] [2034 3190]]





[[10995 5538] [508 4775]]   [[16065   468] [3654 1629]]



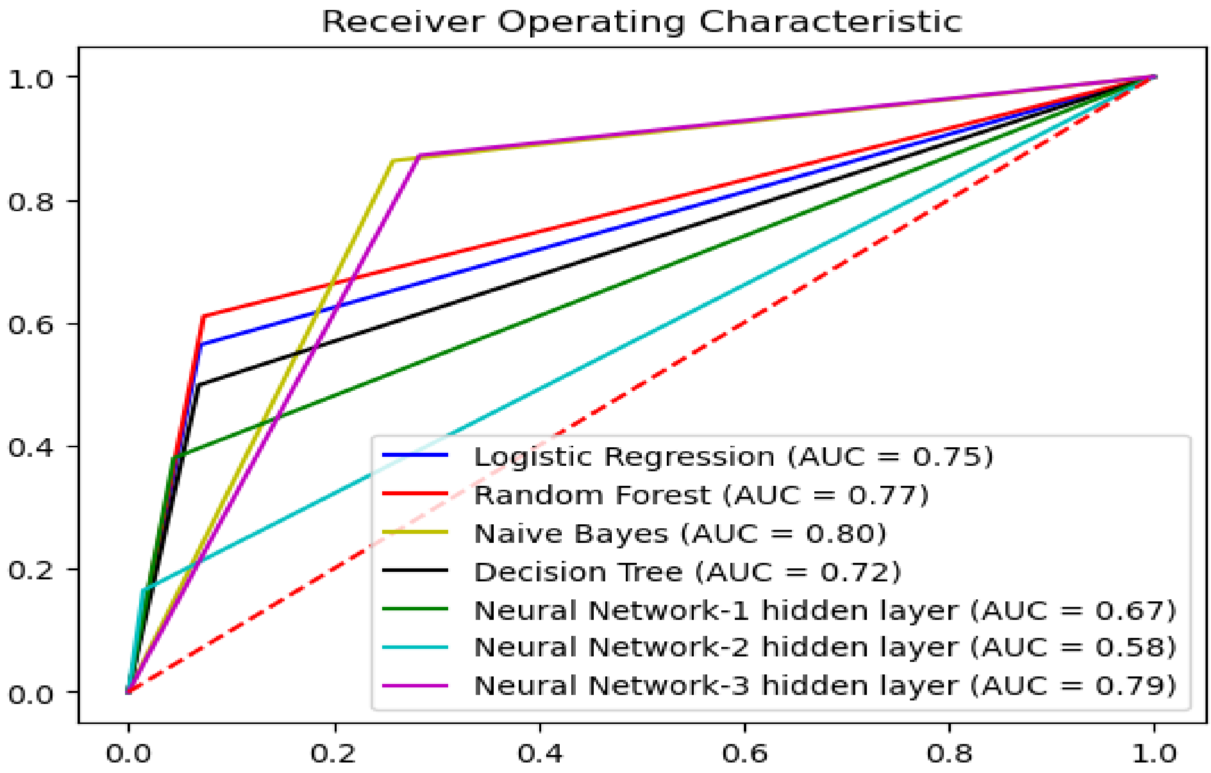
[[14658 1875] [1999 3284]]

ROC curve:

The ROC curve, short for Receiver Operating Characteristic curve, is a graphical plot that illustrates the performance of a binary classifier. It represents the relationship between the sensitivity (true positive rate) and the complement of specificity (false positive rate). By varying the classification threshold, the ROC curve shows how the trade-off between true positive rate and false positive rate changes.

The area under the ROC curve (AUC) is used as a measure of the classifier's performance. It quantifies the overall ability of the classifier to distinguish between the two classes. A higher AUC indicates a better classifier performance, as it suggests a larger area between the ROC curve and the random classifier line. The AUC value ranges from 0 to 1, with 1 representing a perfect classifier and 0.5 representing a random classifier.

Following Graph is obtained for all the classifiers:



Interpretation: Optimal Classifier:

The above observations depicts that Random Forest method is very good for accurately predicting the data with 85.09% accuracy. Also the F1 score and AUC under ROC curve is significantly good. Since it is an ensemble technique, it combines the predictions of many classifiers to give better accuracy. assigns class labels randomly without considering an y information from the input features. But sometimes it might be misleading due to factors su ch as imbalanced datasets, inappropriate evaluation metrics.

Meanwhile, Neural Networks(3-layers), being a modern and unconventional approach , demonstrate remarkable performance. The Neural Network model achieves an AUC curve of 0.79 and an impressive accuracy score of 82.26%.

Also Logistic Regression classifier remarkably performing well with good f1 scores and accuracy.