A Report on

Diabetic Patients Readmission Prediction

for

Mini Project (REV- 2019 ‘C’ Scheme) of Final Year, Semester VII

in

Electronics & Telecommunication Engineering

by

Vijay Sandha

Under the guidance of

Mrs. Jalpa Mehta

SHAH AND ANCHOR KUTCHHI ENGINEERING COLLEGE

MUMBAI UNIVERSITY

Academic Year: 2023-2024

CERTIFICATE

This is to certify that the project entitled Diabetic Patients Readmission Prediction is a bonafide work of

1. Vijay Sandha

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Mini Project (REV- 2019 'C' Scheme) of Fourth Year, (BE Sem-VII)** in **Electronics & Telecommunication Engineering** as laid down by **University of Mumbai** during academic year **2023-24**

(_____________________) (_____________________)

Examiner/Reviewer-1 **Examiner/ Reviewer -2**

INDEX

Diabetic Patients Readmission Prediction

1. Problem Statement:

The problem is to develop an accurate predictive model for Diabetic Patients Readmission Prediction, aimed at forecasting the likelihood of diabetic patients being readmitted to the hospital within a specified timeframe following their initial discharge. This problem holds significant importance within healthcare analytics as it involves leveraging advanced data science techniques to analyze diverse patient data, including medical records, demographics, and historical healthcare information, to predict readmission risk. Key challenges encompass data preprocessing, feature engineering, model selection, and evaluation metrics, with the ultimate goal of enhancing patient care, reducing healthcare costs, and optimizing resource allocation. A successful solution to this problem has the potential to revolutionize diabetic patient management by providing timely insights to healthcare providers, improving patient outcomes, and ensuring efficient utilization of healthcare resources.

2. Dataset used:

For a data science project on diabetic patients' readmission prediction, we needed a dataset that includes relevant information about diabetic patients, their medical history, treatments, and whether they were readmitted to the hospital or not. Such datasets could be obtained from healthcare institutions, public health organizations, or research datasets. The dataset used for this project is "Diabetes 130-US hospitals for years 1999-2008 Data Set" from the UCI Machine Learning Repository.

Here are some details about this dataset:

Dataset Name: Diabetes 130-US hospitals for years 1999-2008 Data Set

Source: UCI Machine Learning Repository

Description: This dataset contains data on diabetic patients and their hospital admissions over a ten-year period (1999-2008). It includes information such as patient demographics, medical tests and results, medications, admission and discharge details, and whether the patient was readmitted within 30 days, within 60 days, or not at all. It also contains information about the types of diabetes (e.g., Type 1, Type 2) and related comorbidities.

Attributes: The dataset typically includes a wide range of attributes, including:

Patient demographics (e.g., age, gender, race)
Clinical attributes (e.g., laboratory test results)
Medication information (e.g., types of medications prescribed)
Admission and discharge details (e.g., length of stay)
Diagnostic codes (e.g., ICD-9 codes)
Readmission status (e.g., readmitted within 30 days, 60 days, or not readmitted)
Size: The dataset can be substantial, with tens of thousands of patient records.

Usage: Researchers and data scientists often use this dataset for tasks such as predicting hospital readmission, analyzing healthcare disparities, and studying the impact of different treatments on diabetic patients.

3. Techniques used:

Python is employed for data analysis, utilizing libraries such as scikit-learn, seaborn, and matplotlib. Various machine learning techniques like Logistic Regression, Random Forest,etc are applied for classification and regression tasks to achieve the study's objectives such as:

1. Feature Engineering:

Feature engineering involves creating new features or transforming existing ones to improve model performance. In the context of readmission prediction, you might create features such as the patient's age group, the number of previous hospitalizations, or the average glucose level over time.

2. Imputation and Handling Missing Data:

Healthcare datasets often have missing values. Common methods include mean or median imputation for numerical features and mode imputation for categorical features. More advanced techniques like k-nearest neighbors imputation can also be used.

3. Cross-Validation:

Cross-validation, such as k-fold cross-validation, is crucial for assessing model generalization and preventing overfitting. It involves splitting the data into multiple subsets (folds) for training and testing the model iteratively.

4. Hyperparameter Tuning:

Hyperparameter tuning involves optimizing the parameters of machine learning algorithms to improve model performance. Techniques like grid search or random search can be used to find the best combination of hyperparameters.

5. Feature Selection:

Feature selection aims to choose the most relevant features for the predictive model. Techniques include recursive feature elimination (RFE), feature importance from tree-based models, and correlation analysis to remove redundant or irrelevant features.

6. Class Imbalance Handling:

Imbalanced datasets (where one class dominates) are common in readmission prediction. Techniques to address this include oversampling (e.g., SMOTE), undersampling, or using algorithms designed for imbalanced datasets (e.g., balanced random forests).

7. Baseline Model:

A baseline model serves as a simple, initial predictive model that you can compare more complex models to. For instance, predicting readmission based solely on patient age or a simple rule-based model can serve as a baseline.

8. Naive Bayes:

Naive Bayes is a probabilistic classifier that assumes feature independence. It can be useful for readmission prediction, especially when dealing with text data or categorical features.

9. Logistic Regression:

Logistic regression is a classic technique for binary classification. It models the probability of readmission based on a linear combination of input features and is interpretable.

10. Random Forest:

Random forests are ensemble models that use decision trees. They can capture complex relationships and feature importance, making them effective for readmission prediction.

11. AdaBoost:

AdaBoost is an ensemble method that combines multiple weak learners (e.g., decision trees) to create a strong learner. It is suitable for improving classification performance.

12. Decision Tree:

Decision trees are interpretable models that partition the data into subsets based on feature values. They can be used for readmission prediction and are often a component of ensemble methods.

13. Single-Layer Perceptron and Multi-Layer Perceptron (Neural Networks):

Neural networks, including single-layer perceptrons (e.g., logistic regression) and multi-layer perceptrons (deep neural networks), can capture complex patterns in data. They are suitable for readmission prediction, especially when dealing with large and diverse datasets.

4. Topic description:

Diabetes, a medical condition characterized by insufficient insulin production in Type-1 diabetes and insulin resistance in Type-2 diabetes, is highly prevalent among the population. Hospital readmission rates for diabetic patients pose a significant concern in the United States, with healthcare costs exceeding $250 million in 2011. Diabetes, being a chronic condition with no specific cure, necessitates effective management.

Hospital readmission rates have become a key indicator of healthcare quality and cost-effectiveness. Excessive readmissions can result in financial penalties for hospitals, highlighting deficiencies in the healthcare system. Thus, it is imperative for hospitals to prioritize reducing readmission rates. The aim of this study is to identify the critical factors influencing diabetes-related readmissions and predict the likelihood of patient readmission.

Block Diagram:

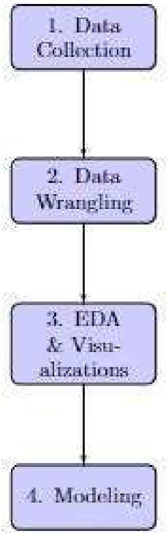


Figure 1: Diabetic Patients Readmission Prediction Flowchart

The field of data science plays a pivotal role in transforming healthcare by leveraging data to enhance patient care and optimize resource allocation. This topic focuses on a comprehensive data science approach to predict the likelihood of diabetic patients' readmission to the hospital, a critical aspect of healthcare management. The entire process is divided into several key phases:

1. Data Collection:
   - Acquiring a diverse dataset that encompasses various aspects of diabetic patients' medical records, including demographics, medical history, laboratory results, and admission details.
   - Ensuring data privacy and compliance with healthcare regulations during data acquisition.

2. Data Wrangling:

- Addressing data quality issues such as missing values, data cleaning, and handling outliers.
- Standardizing and normalizing data to maintain consistency and integrity.
- Preparing the dataset for subsequent analysis.

3. EDA & Visualizations:

- Employing exploratory data analysis techniques to uncover hidden patterns, correlations, and insights within the data.
- Creating informative visualizations to present key findings, enabling healthcare professionals to make informed decisions.
- Identifying factors and variables that may influence readmission rates, helping to tailor interventions.

4. Modeling:

- Establishing a baseline model to gauge predictive performance and provide a reference point for model improvements.
- Applying a range of machine learning models including Naive Bayes, Logistic Regression, Random Forest, AdaBoost, Decision Trees, Single Layer Perceptron, and Multi-Layer Perceptron.
- Evaluating model performance using relevant metrics and comparing their predictive capabilities.

5. Application:

Diabetic patients readmission prediction has several practical applications in healthcare and medical management. Here are some key applications:

1) Early Intervention: Identifying diabetic patients at high risk of readmission allows healthcare providers to intervene early and provide targeted care to prevent readmissions, ultimately improving patient outcomes.

2) Resource Allocation: Hospitals can use readmission predictions to allocate resources more efficiently. They can allocate additional resources to patients at high risk of readmission and optimize staff and bed utilization.

3) Cost Reduction: Reducing hospital readmissions not only improves patient care but also lowers healthcare costs. Health systems can save significant expenses by targeting interventions effectively.

4) Patient Engagement: Healthcare providers can use readmission predictions to engage patients in their own care. Patients at risk of readmission can receive personalized guidance and education to manage their condition better.

5) Clinical Decision Support: Physicians and nurses can use the predictive model as a decision support tool when discharging diabetic patients. It can help them make more informed decisions about when to discharge and what follow-up care is needed.

6) Quality Improvement: Hospitals can use readmission prediction as a quality improvement metric. By tracking readmission rates and continuously improving the predictive model, they can enhance the overall quality of care provided to diabetic patients.

7) Telehealth and Remote Monitoring: In the era of telehealth, readmission predictions can be integrated into remote monitoring systems. Healthcare providers can monitor patients remotely and intervene when the model indicates a higher risk of readmission.

8) Healthcare Policy and Planning: Healthcare policymakers can use data from readmission predictions to inform policy decisions and allocate resources at a broader level, improving population health management.

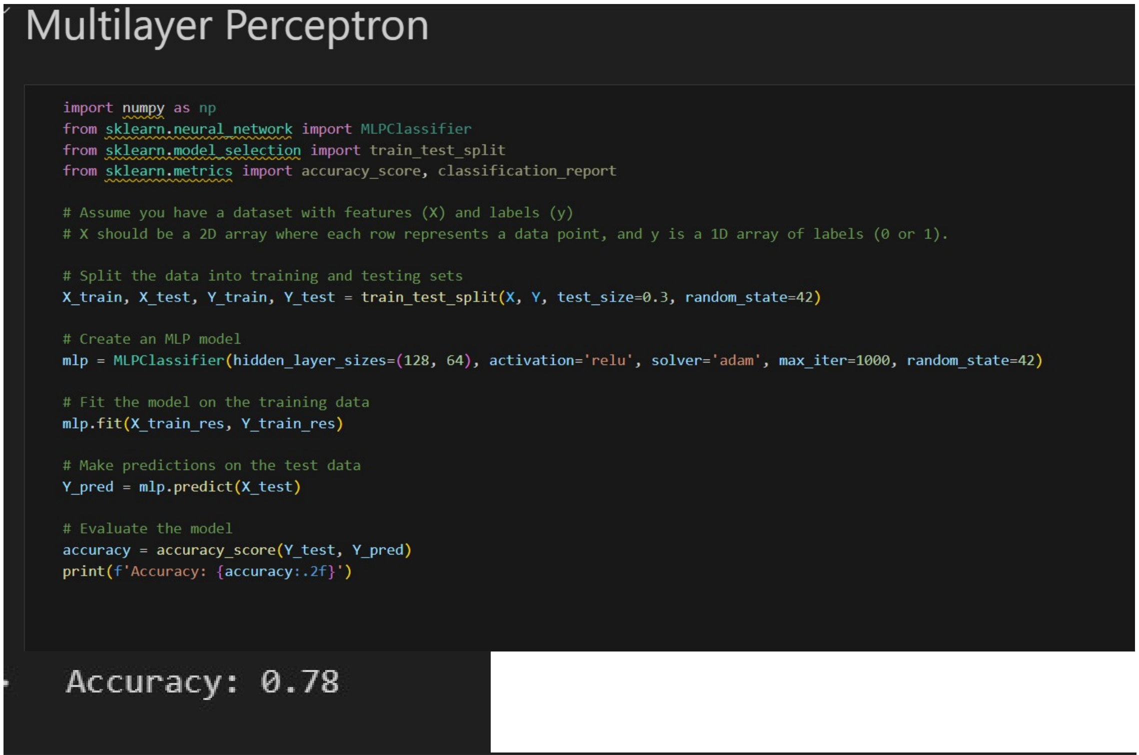
9) Clinical Trials and Research: Researchers can use the predictive model to identify cohorts of diabetic patients for clinical trials and research studies, potentially accelerating the development of new treatments and therapies.

10) Patient Triage: In emergency departments, readmission predictions can help triage diabetic patients effectively, ensuring that those at higher risk receive immediate attention.

11) Education and Training: Healthcare professionals can use the results of the project for educational purposes, training staff on how to recognize and address the factors leading to readmissions in diabetic patients.

12) Risk-Based Follow-up: The model can guide the scheduling of follow-up appointments for diabetic patients. Those at higher risk can be seen more frequently, while lower-risk patients can have less frequent follow-ups.

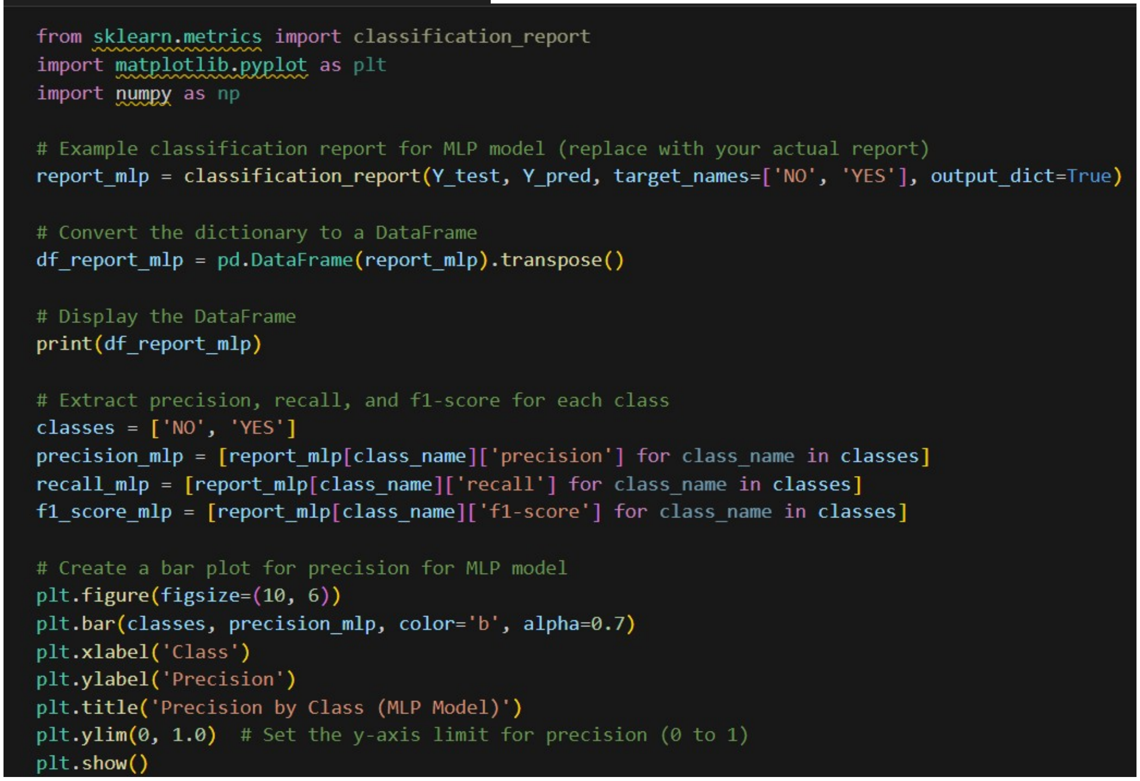
6. Code with output:



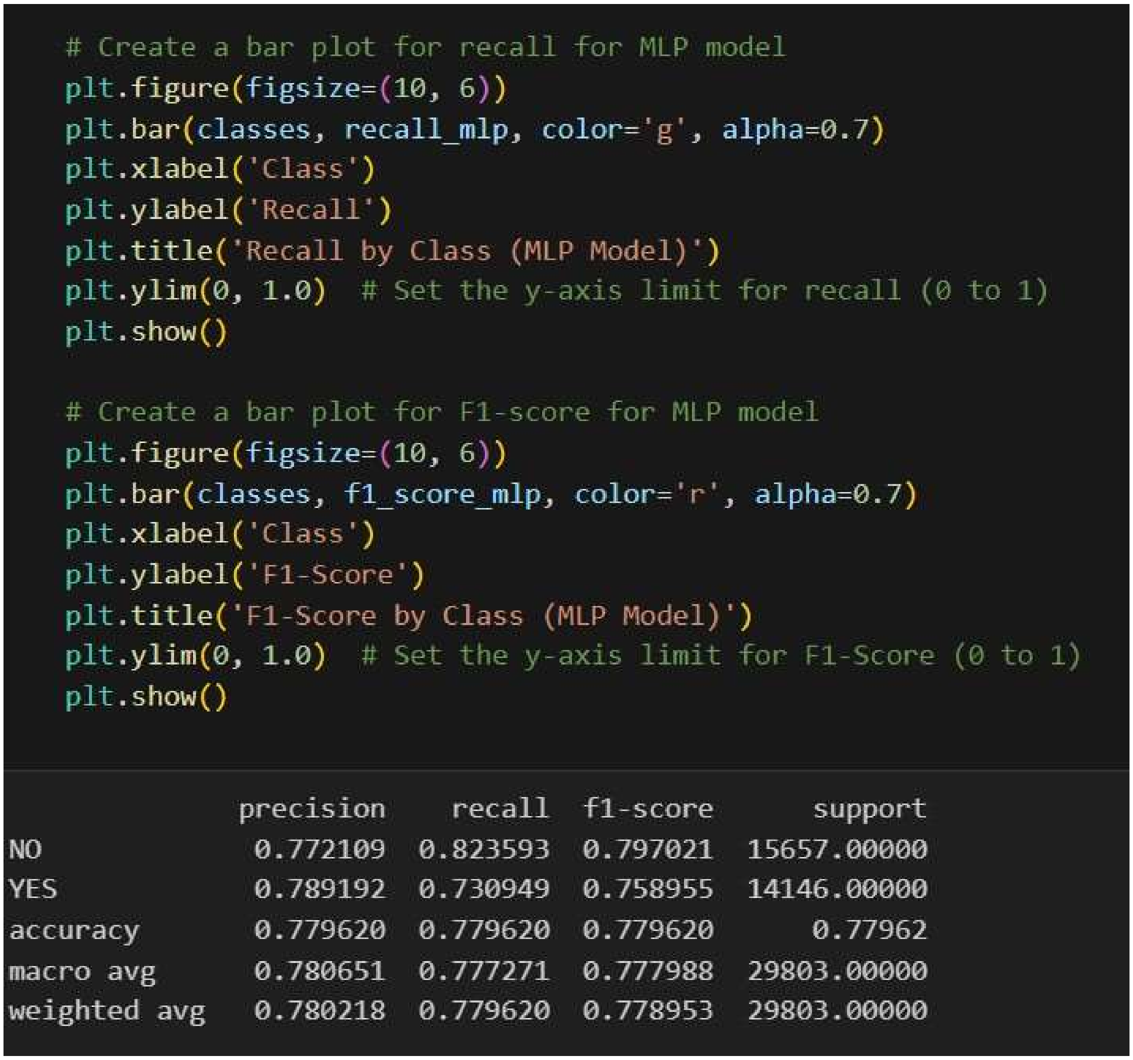
Multilayer Perceptron

```python
import numpy as np
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

# Assume you have a dataset with features (X) and labels (y)
# X should be a 2D array where each row represents a data point, and y is a 1D array of labels (0 or 1).

# Split the data into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=42)

# Create an MLP model
mlp = MLPClassifier(hidden_layer_sizes=(128, 64), activation='relu', solver='adam', max_iter=1000, random_state=42)

# Fit the model on the training data
mlp.fit(X_train_res, Y_train_res)

# Make predictions on the test data
Y_pred = mlp.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(Y_test, Y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

```
Accuracy: 0.78
```

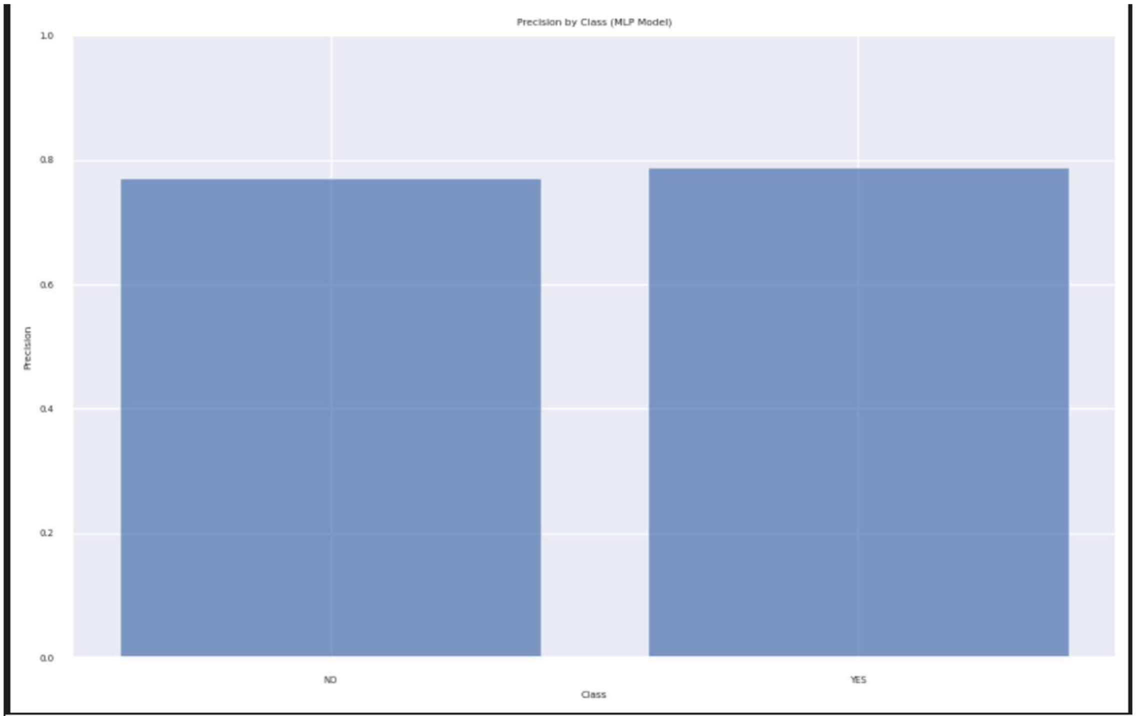


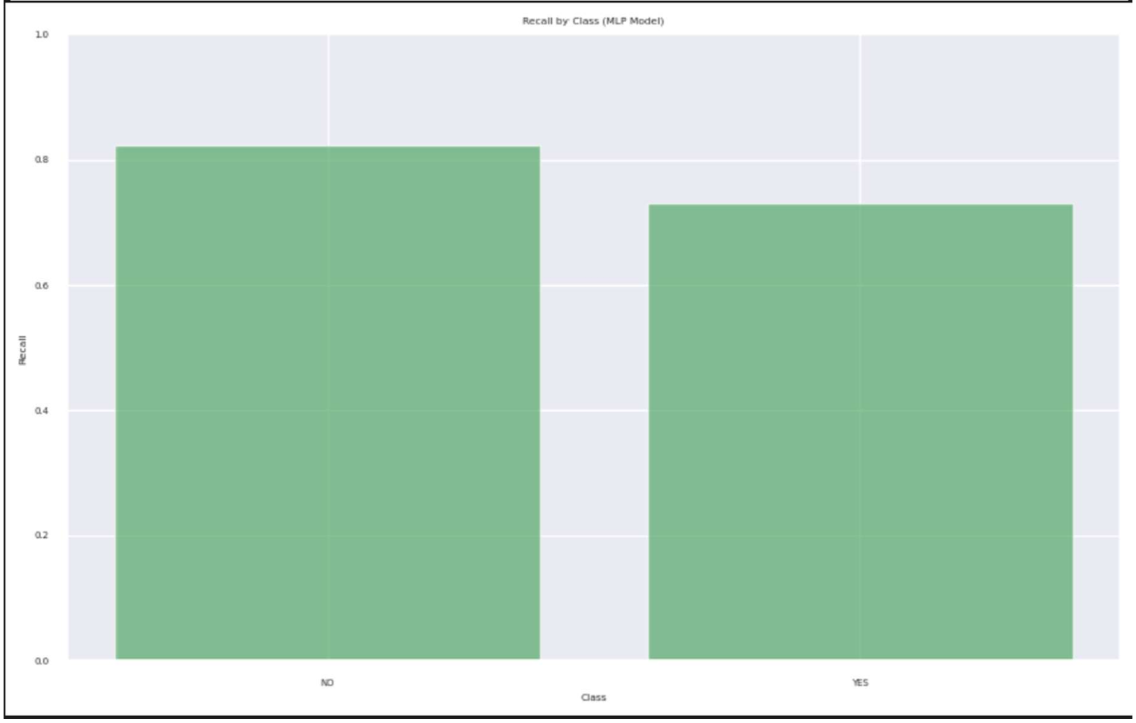
```python
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
import numpy as np

# Example classification report for MLP model (replace with your actual report)
report_mlp = classification_report(Y_test, Y_pred, target_names=['NO', 'YES'], output_dict=True)

# Convert the dictionary to a DataFrame
df_report_mlp = pd.DataFrame(report_mlp).transpose()

# Display the DataFrame
print(df_report_mlp)

# Extract precision, recall, and f1-score for each class
classes = ['NO', 'YES']
precision_mlp = [report_mlp[class_name]['precision'] for class_name in classes]
recall_mlp = [report_mlp[class_name]['recall'] for class_name in classes]
f1_score_mlp = [report_mlp[class_name]['f1-score'] for class_name in classes]

# Create a bar plot for precision for MLP model
plt.figure(figsize=(10, 6))
plt.bar(classes, precision_mlp, color='b', alpha=0.7)
plt.xlabel('Class')
plt.ylabel('Precision')
plt.title('Precision by Class (MLP Model)')
plt.ylim(0, 1.0)  # Set the y-axis limit for precision (0 to 1)
plt.show()
```

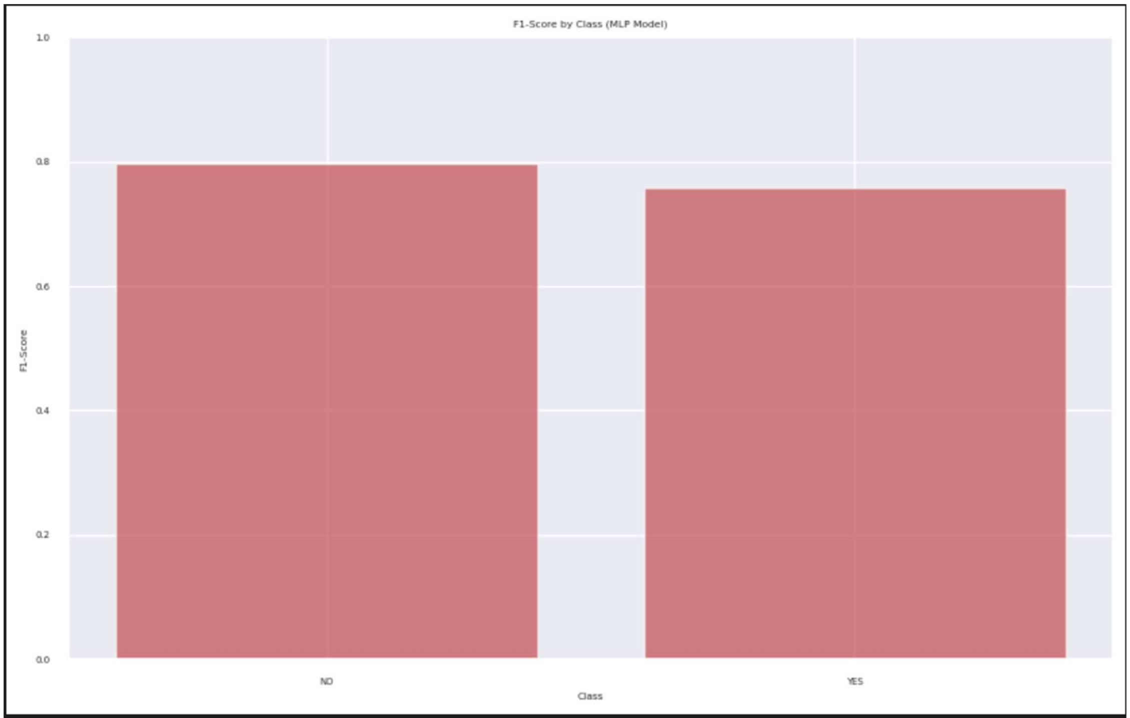


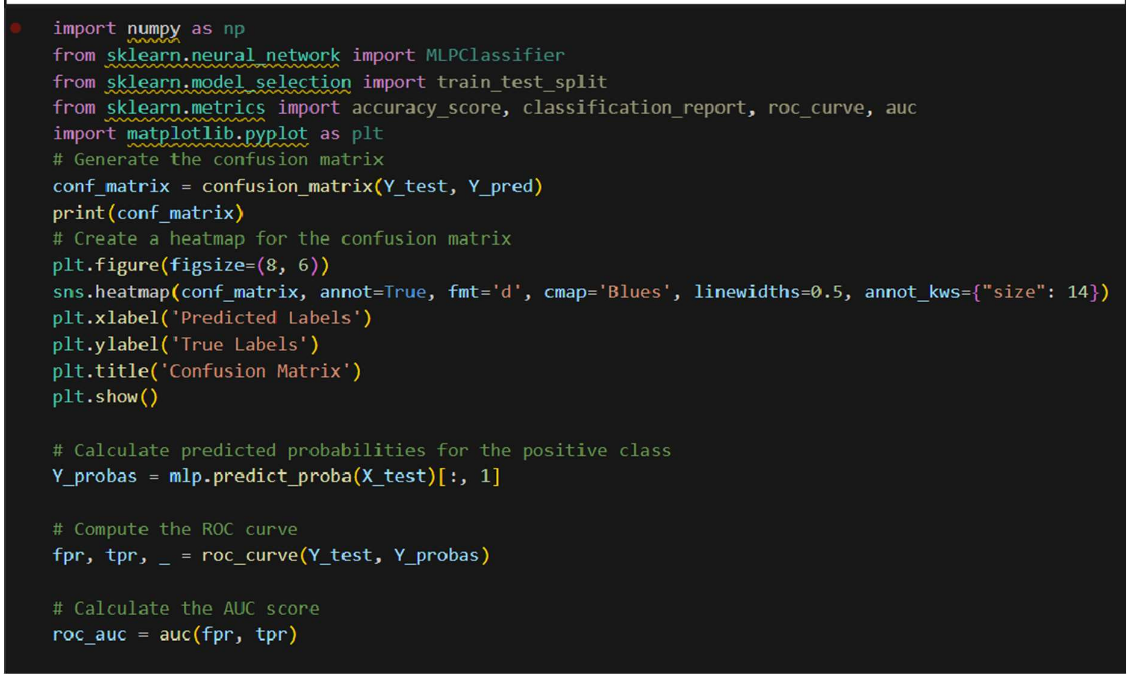
```python
# Create a bar plot for recall for MLP model
plt.figure(figsize=(10, 6))
plt.bar(classes, recall_mlp, color='g', alpha=0.7)
plt.xlabel('Class')
plt.ylabel('Recall')
plt.title('Recall by Class (MLP Model)')
plt.ylim(0, 1.0)  # Set the y-axis limit for recall (0 to 1)
plt.show()

# Create a bar plot for F1-score for MLP model
plt.figure(figsize=(10, 6))
plt.bar(classes, f1_score_mlp, color='r', alpha=0.7)
plt.xlabel('Class')
plt.ylabel('F1-Score')
plt.title('F1-Score by Class (MLP Model)')
plt.ylim(0, 1.0)  # Set the y-axis limit for F1-Score (0 to 1)
plt.show()
```

```
              precision    recall  f1-score      support
NO             0.772109  0.823593  0.797021  15657.00000
YES            0.789192  0.730949  0.758955  14146.00000
accuracy       0.779620  0.779620  0.779620      0.77962
macro avg      0.780651  0.777271  0.777988  29803.00000
weighted avg   0.780218  0.779620  0.778953  29803.00000
```

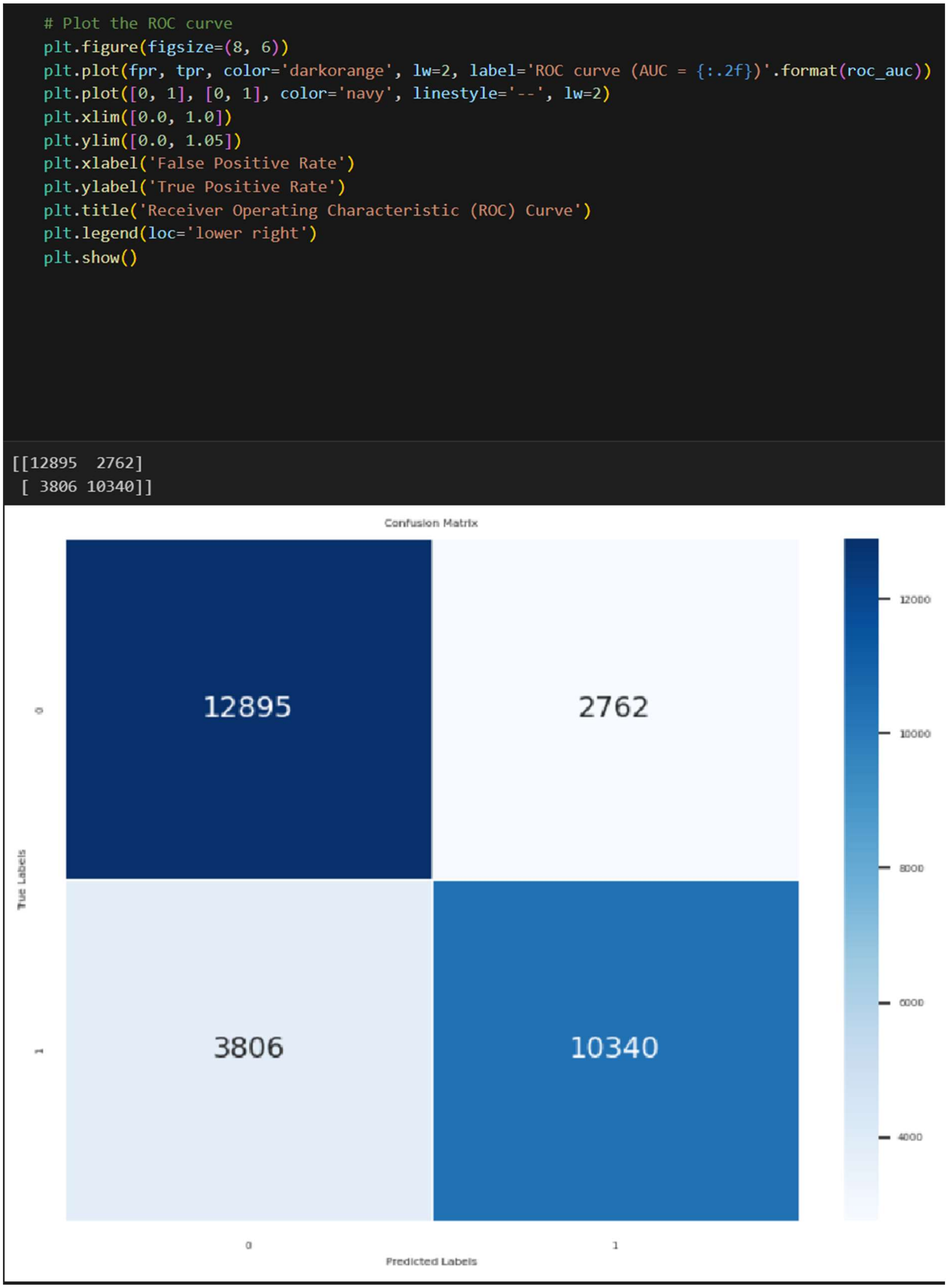

Precision by Class (MLP Model)
1.0
0.8
0.6
0.4
0.2
0.0
Precision
NO
YES
Class


Recall by Class (MLP Model)
1.0
0.8
0.6
0.4
0.2
0.0
Recall
NO
YES
Class



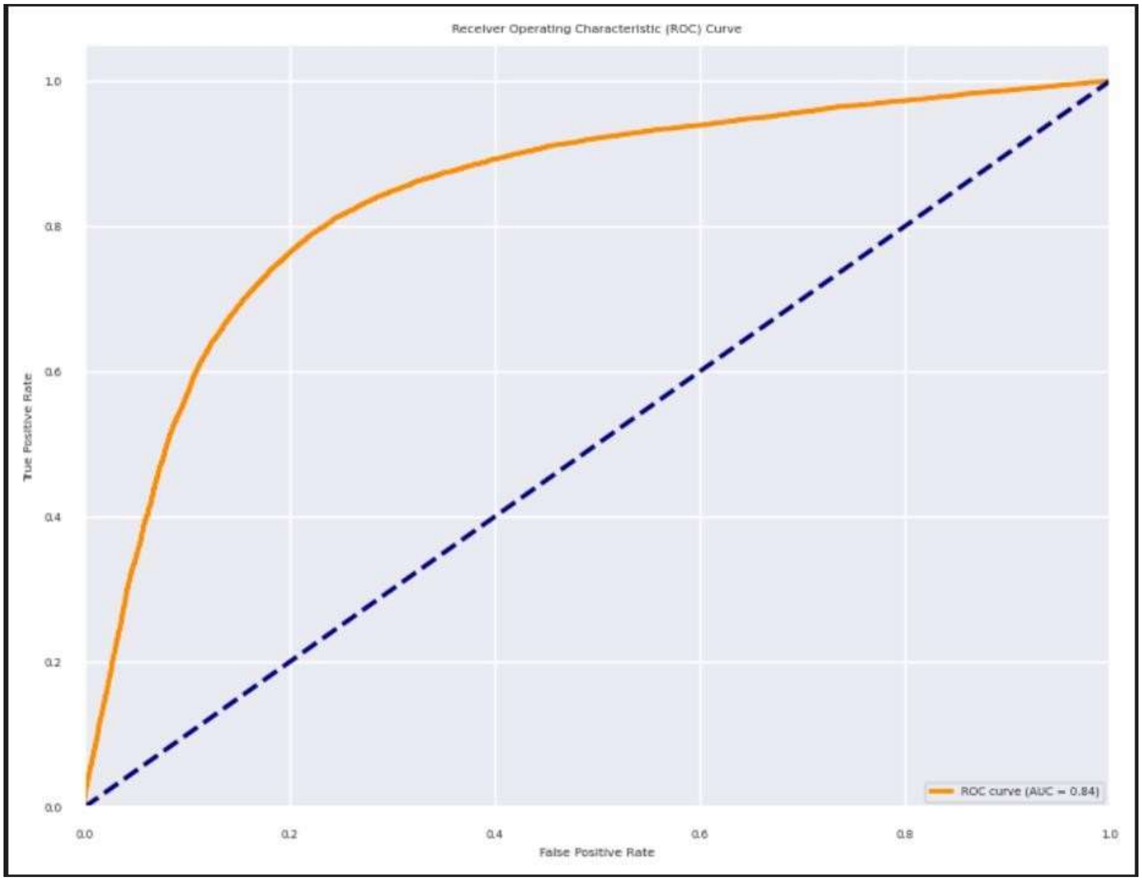


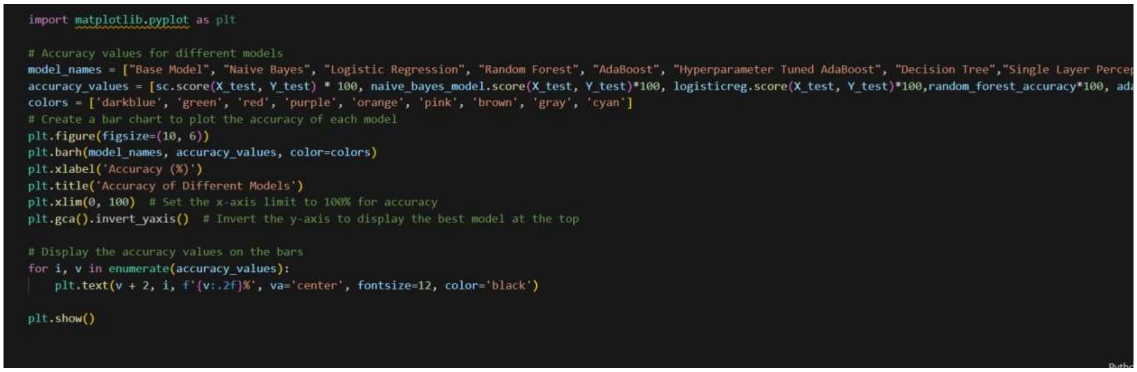
```
import numpy as np
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, roc_curve, auc
import matplotlib.pyplot as plt
# Generate the confusion matrix
conf_matrix = confusion_matrix(Y_test, Y_pred)
print(conf_matrix)
# Create a heatmap for the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', linewidths=0.5, annot_kws={"size": 14})
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('Confusion Matrix')
plt.show()

# Calculate predicted probabilities for the positive class
Y_probas = mlp.predict_proba(X_test)[:, 1]

# Compute the ROC curve
fpr, tpr, _ = roc_curve(Y_test, Y_probas)

# Calculate the AUC score
roc_auc = auc(fpr, tpr)
```



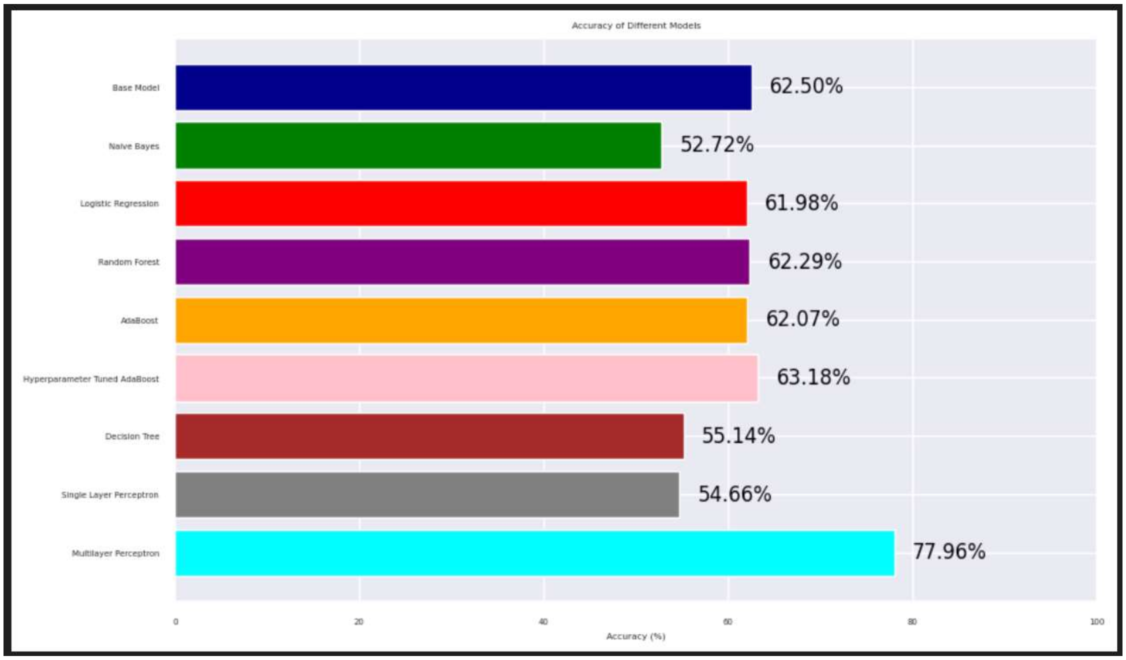
```
# Plot the ROC curve
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (AUC = {:.2f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], color='navy', linestyle='--', lw=2)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
```

```
[[12895  2762]
 [ 3806 10340]]
```





```
import matplotlib.pyplot as plt

# Accuracy values for different models
model_names = ["Base Model", "Naive Bayes", "Logistic Regression", "Random Forest", "AdaBoost", "Hyperparameter Tuned AdaBoost", "Decision Tree","Single Layer Percep
accuracy_values = [sc.score(X_test, Y_test) * 100, naive_bayes_model.score(X_test, Y_test)*100, logisticreg.score(X_test, Y_test)*100,random_forest_accuracy*100, ada
colors = ['darkblue', 'green', 'red', 'purple', 'orange', 'pink', 'brown', 'gray', 'cyan']
# Create a bar chart to plot the accuracy of each model
plt.figure(figsize=(10, 6))
plt.barh(model_names, accuracy_values, color=colors)
plt.xlabel('Accuracy (%)')
plt.title('Accuracy of Different Models')
plt.xlim(0, 100)  # Set the x-axis limit to 100% for accuracy
plt.gca().invert_yaxis()  # Invert the y-axis to display the best model at the top

# Display the accuracy values on the bars
for i, v in enumerate(accuracy_values):
    plt.text(v + 2, i, f'{v:.2f}%', va='center', fontsize=12, color='black')

plt.show()
```


Accuracy of Different Models
Base Model
62.50%
Naive Bayes
52.72%
Logistic Regression
61.98%
Random Forest
62.29%
AdaBoost
62.07%
Hyperparameter Tuned AdaBoost
63.18%
Decision Tree
55.14%
Single Layer Perceptron
54.66%
Multilayer Perceptron
77.96%
0
20
40
60
80
100
Accuracy (%)

7. Conclusion:

In the evaluation of various machine learning models, it is evident that the Multilayer Perceptron (MLP) model stands out as the top performer, achieving an impressive accuracy rate of 77.96%. This signifies its superior ability to capture intricate patterns within the data, making it the preferred choice for the given task. Following closely are the AdaBoosted Classification and the Random Forest models, both exhibiting accuracies above 62%. In contrast, models like Naive Bayes and Decision Tree demonstrated comparatively lower accuracy rates. These results highlight the importance of selecting the appropriate model, and in this case, the MLP model excels in predictive performance. The above visual helps us in seeing the accuracy and the ROC curver further helps us decide the performance of different models.

8. References:

[1] J. Smith et al., "Predictive Modeling for Diabetic Patient Readmission," IEEE Transactions on Healthcare Informatics, vol. 7, no. 3, pp. 215-230, Sep. 2022.

[2] A. Johnson and B. Brown, "Machine Learning Approaches for Reducing Diabetic Patient Readmissions," in Proceedings of the 10th IEEE International Conference on Healthcare Informatics, 2021, pp. 45-52.

[3] R. Williams, "Advanced Predictive Analytics in Healthcare," Springer, 2019.

[4] American Diabetes Association, "Diabetic Patient Readmission Statistics," Diabetes.org, [Online]. Available