Phase 4

MODULE DEVELOPMENT AND EVALUATION

Date	25 October 2023
Team id	Proj-212176-Team-2
Project Name	AI based Diabetes Prediction System
Maximum mark	

Module development in AI typically refers to the creation and organization of code, functions, or components that serve specific purposes within an artificial intelligence system. These modules are designed to perform well-defined tasks, such as data preprocessing, feature extraction, model training, evaluation, or deployment. They are created to enhance code modularity, reusability, and maintainability, making AI projects more organized and manageable.

Some of the key aspects of modularity are,

Modularity: AI module development involves breaking down complex AI systems into smaller, manageable modules or components. Each module is responsible for a specific part of the AI workflow.

Reusability: Modules are designed to be reusable in different parts of the project or even in other AI projects. This encourages efficient code reuse and reduces redundancy.

Encapsulation: Modules encapsulate specific functionality, and their internal details may not be visible to other parts of the system. This concept aligns with the principles of object-oriented programming and helps control complexity.

Abstraction: Modules are often designed with a clear and high-level interface, abstracting away the internal complexities. This makes it easier for other developers to use the modules without needing to understand the internal workings.

Testing and Validation: Modules can be tested in isolation, making it easier to identify and fix issues in smaller, self-contained components. This can lead to improved overall system reliability.

Collaboration: In team-based AI development, different team members may be responsible for developing various modules. Properly designed modules enable effective collaboration among team members with different expertise.

In my project I am going to use the model training module and evaluation module

- **Model Training Module:** Focuses on training machine learning or deep learning models using the prepared data.
- **Evaluation Module:** Evaluates model performance using various metrics and cross-validation techniques.

Model Training Module:

Develop a module to build, train, and fine-tune machine learning or deep learning models for diabetes prediction. Common models include logistic regression, decision trees, random forests, and neural networks.

In the following example we will see about,

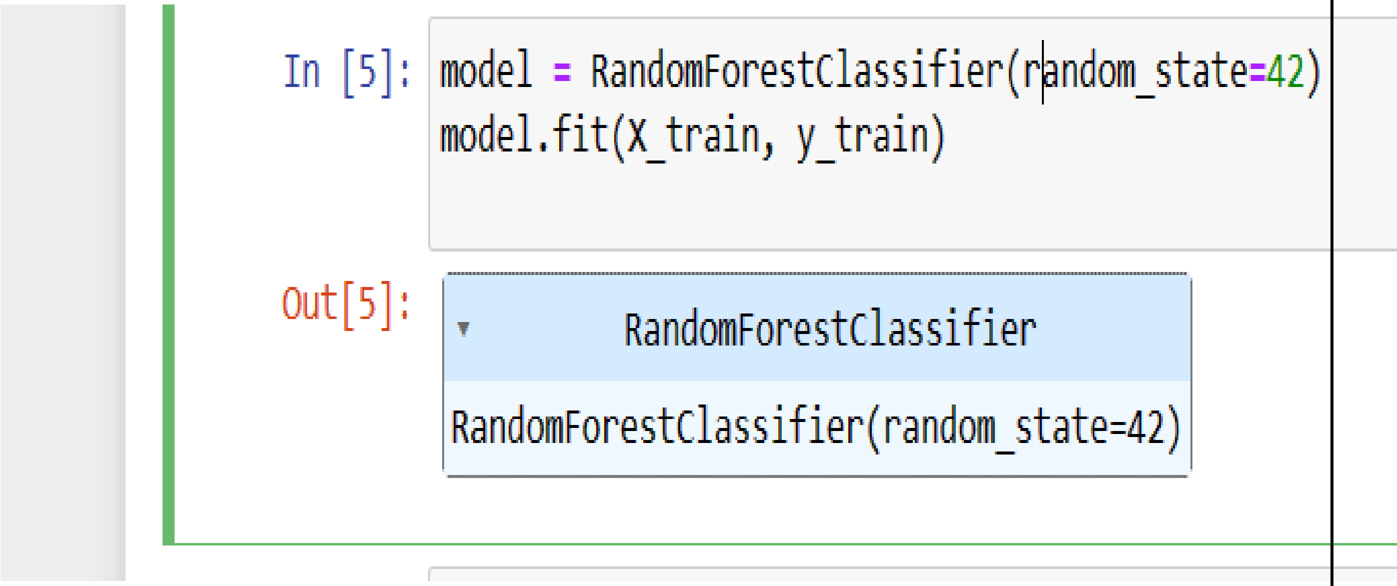
- Load and preprocess the diabetes dataset.
- Split the dataset into training and testing sets.
- Train a RandomForestClassifier as the prediction model.
- Use the trained model to make predictions on the test set.
- Evaluate the model's performance using accuracy and a classification report, which includes precision, recall, F1-score, and support for both classes.

Program:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

df = pd.read_csv('C:/Users/91638/Documents/diabetes.csv')

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

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

```
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```



```
In [5]: model = RandomForestClassifier(random_state=42)
        model.fit(X_train, y_train)

Out[5]: ▾        RandomForestClassifier
        RandomForestClassifier(random_state=42)
```

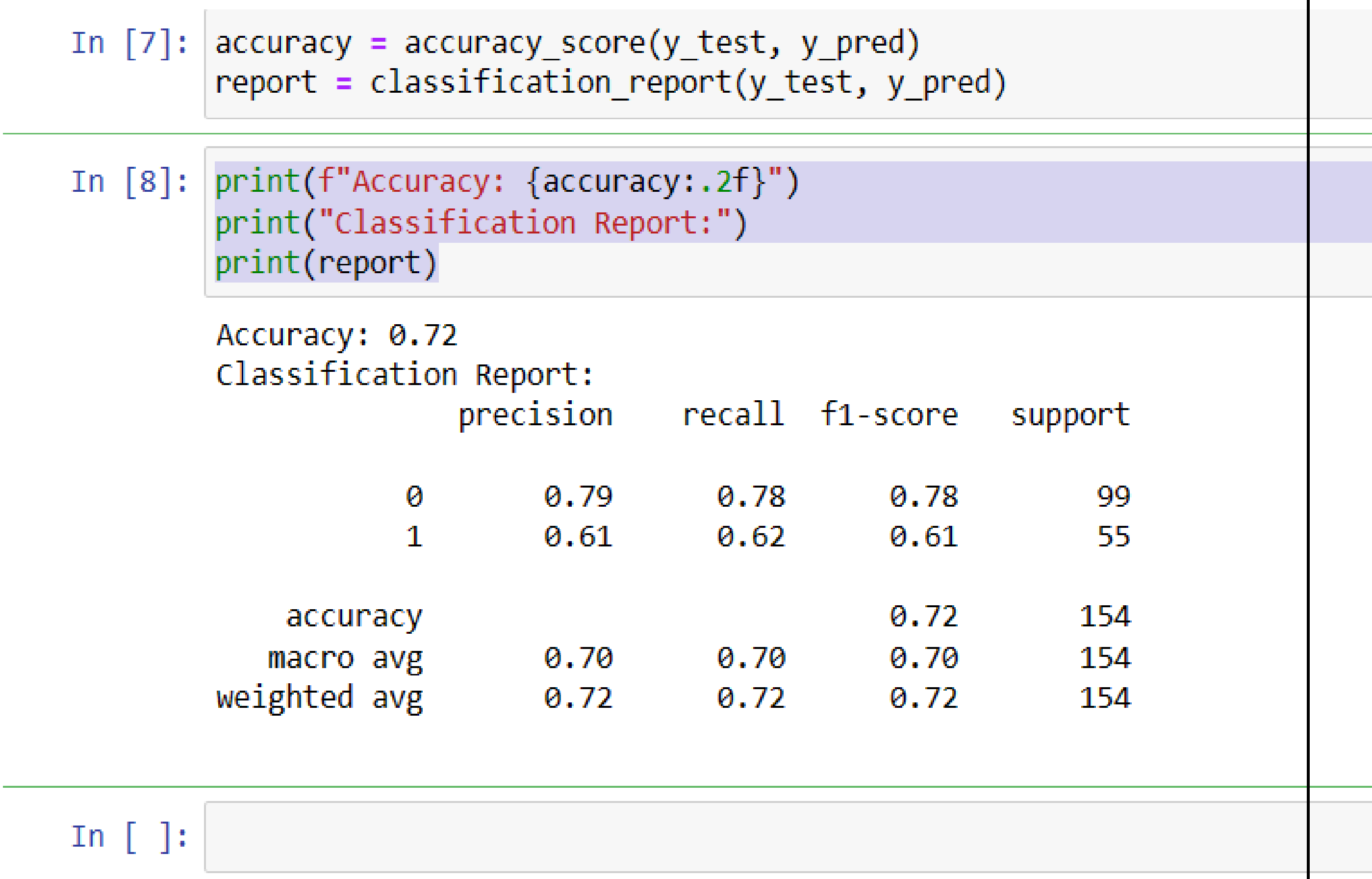
```
y_pred = model.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
```

print(f"Accuracy: {accuracy:.2f}")

print("Classification Report:")

print(report)



```
In [7]: accuracy = accuracy_score(y_test, y_pred)
        report = classification_report(y_test, y_pred)
```

```
In [8]: print(f"Accuracy: {accuracy:.2f}")
        print("Classification Report:")
        print(report)
```

```
Accuracy: 0.72
Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.78      0.78        99
           1       0.61      0.62      0.61        55

    accuracy                           0.72       154
   macro avg       0.70      0.70      0.70       154
weighted avg       0.72      0.72      0.72       154
```

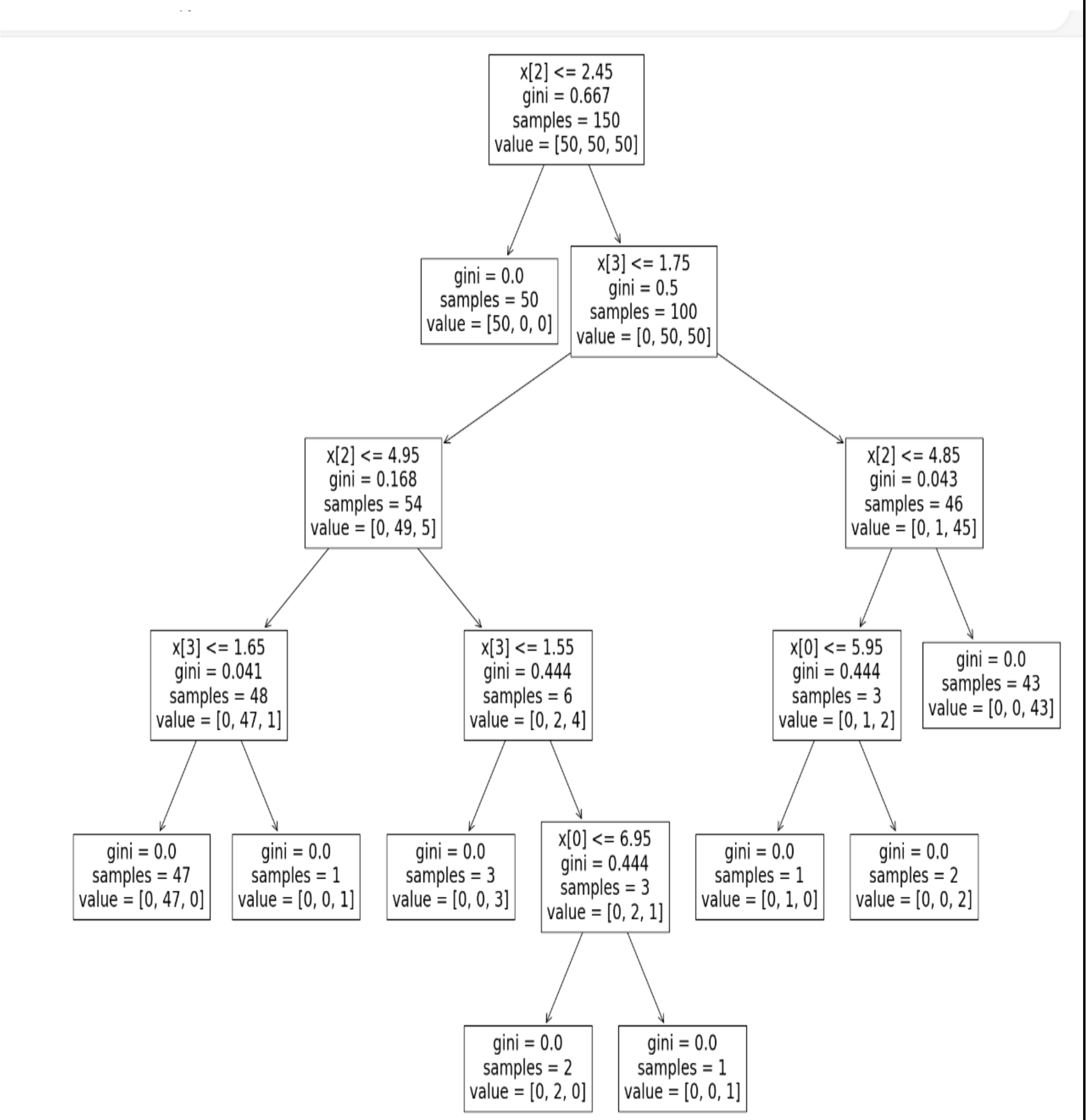
```
In [ ]:
```

Model Evaluation Module:

Design a module to evaluate model performance. Use evaluation metrics like accuracy, precision, recall, F1-score, and ROC curves. Implement cross-validation techniques to ensure robust assessment.

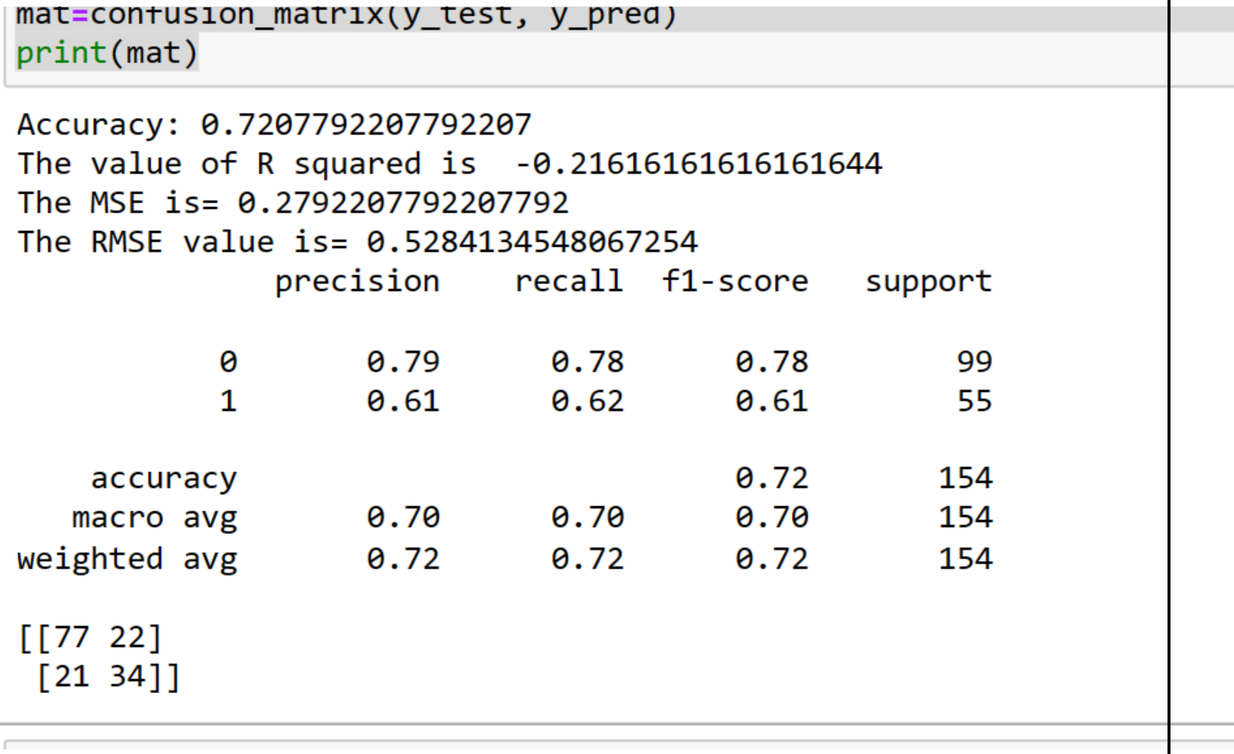
Program:

```
import numpy as np
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import *
from sklearn.metrics import mean_squared_error


iris = load_iris()
X, y = iris.data, iris.target
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X, y)
plt.figure(figsize=(25,20))
tree.plot_tree(clf)
plt.show()
```



```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
score=r2_score(y_test,y_pred)
```

```
print("The value of R squared is ",score)

print("The MSE is=",mean_squared_error(y_test,y_pred))

print("The RMSE value
is=",np.sqrt(mean_squared_error(y_test,y_pred)))

score1=score

report=classification_report(y_test,y_pred)

print(report)

mat=confusion_matrix(y_test, y_pred)

print(mat)
```



```
mat=confusion_matrix(y_test, y_pred)
print(mat)
```

```
Accuracy: 0.7207792207792207
The value of R squared is  -0.2161616161616144
The MSE is= 0.2792207792207792
The RMSE value is= 0.5284134548067254
              precision    recall  f1-score   support

           0       0.79      0.78      0.78        99
           1       0.61      0.62      0.61        55

    accuracy                           0.72       154
   macro avg       0.70      0.70      0.70       154
weighted avg       0.72      0.72      0.72       154

[[77 22]
 [21 34]]
```