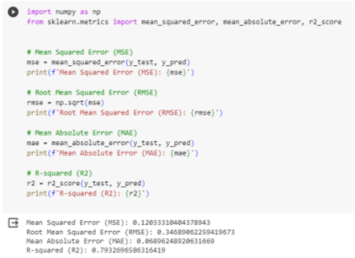
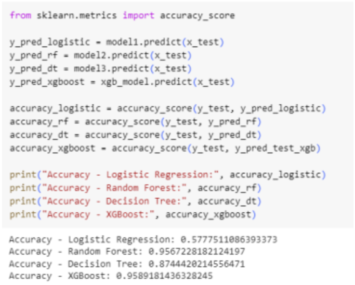
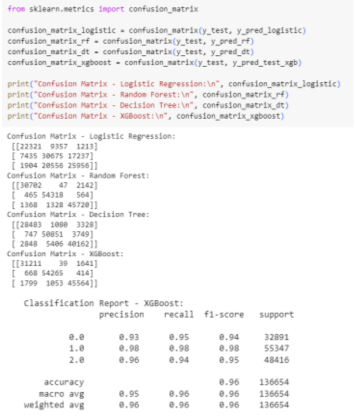
Project Development Phase
Model Performance Test

Date	8 November 2023
Team ID	592697
Project Name	Diabetes Prediction using ML
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE - , MSE - , RMSE - , R2 score - **Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	 
2.	Tune the Model	Validation Method -	

```
import numpy as np
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score


# Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error (MSE): {mse}')

# Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)
print(f'Root Mean Squared Error (RMSE): {rmse}')

# Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, y_pred)
print(f'Mean Absolute Error (MAE): {mae}')

# R-squared (R2)
r2 = r2_score(y_test, y_pred)
print(f'R-squared (R2): {r2}')
```

```
Mean Squared Error (MSE): 0.12033310404378943
Root Mean Squared Error (RMSE): 0.34689062259419673
Mean Absolute Error (MAE): 0.06896248920631669
R-squared (R2): 0.7932696586316419
```

```
from sklearn.metrics import accuracy_score

y_pred_logistic = model1.predict(x_test)
y_pred_rf = model2.predict(x_test)
y_pred_dt = model3.predict(x_test)
y_pred_xgboost = xgb_model.predict(x_test)

accuracy_logistic = accuracy_score(y_test, y_pred_logistic)
accuracy_rf = accuracy_score(y_test, y_pred_rf)
accuracy_dt = accuracy_score(y_test, y_pred_dt)
accuracy_xgboost = accuracy_score(y_test, y_pred_test_xgb)

print("Accuracy - Logistic Regression:", accuracy_logistic)
print("Accuracy - Random Forest:", accuracy_rf)
print("Accuracy - Decision Tree:", accuracy_dt)
print("Accuracy - XGBoost:", accuracy_xgboost)
```

```
Accuracy - Logistic Regression: 0.5777511086393373
Accuracy - Random Forest: 0.9567228182124197
Accuracy - Decision Tree: 0.8744420214556471
Accuracy - XGBoost: 0.9589181436328245
```

```
from sklearn.metrics import confusion_matrix

confusion_matrix_logistic = confusion_matrix(y_test, y_pred_logistic)
confusion_matrix_rf = confusion_matrix(y_test, y_pred_rf)
confusion_matrix_dt = confusion_matrix(y_test, y_pred_dt)
confusion_matrix_xgboost = confusion_matrix(y_test, y_pred_test_xgb)

print("Confusion Matrix - Logistic Regression:\n", confusion_matrix_logistic)
print("Confusion Matrix - Random Forest:\n", confusion_matrix_rf)
print("Confusion Matrix - Decision Tree:\n", confusion_matrix_dt)
print("Confusion Matrix - XGBoost:\n", confusion_matrix_xgboost)
```

```
Confusion Matrix - Logistic Regression:
 [[22321  9357  1213]
 [ 7435 30675 17237]
 [ 1904 20556 25956]]
Confusion Matrix - Random Forest:
 [[30702    47  2142]
 [  465 54318   564]
 [ 1368  1328 45720]]
Confusion Matrix - Decision Tree:
 [[28483  1080  3328]
 [  747 50851  3749]
 [ 2848  5406 40162]]
Confusion Matrix - XGBoost:
 [[31211    39  1641]
 [  668 54265   414]
 [ 1799  1053 45564]]

   Classification Report - XGBoost:
              precision    recall  f1-score   support

         0.0       0.93      0.95      0.94     32891
         1.0       0.98      0.98      0.98     55347
         2.0       0.96      0.94      0.95     48416

    accuracy                           0.96    136654
   macro avg       0.95      0.96      0.96    136654
weighted avg       0.96      0.96      0.96    136654
```