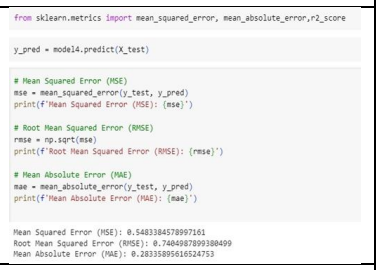
Project Development Phase
Model Performance Test

Date	9 November 2023
Team ID	593030
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	Hyperparameter Tuning - Validation Method -	NA

Regression model:

```
from sklearn.metrics import mean_squared_error, mean_absolute_error,r2_score
```

```
y_pred = model4.predict(X_test)
```

```
# 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}')
```

```
Mean Squared Error (MSE): 0.5483384578997161
Root Mean Squared Error (RMSE): 0.7404987899380499
Mean Absolute Error (MAE): 0.28335895616524753
```

XGBoost classifier:

XgBoost Classifier

```
import xgboost as xgb
from sklearn.metrics import accuracy_score, classification_report

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)

model4 = xgb.XGBClassifier()

model4.fit(X_train, y_train)
y_pred = model4.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print("Classification Report:\n", classification_rep)
```

```
Accuracy: 0.8491307947019867
Classification Report:
              precision    recall  f1-score   support

         0.0       0.86      0.98      0.92     85569
         1.0       0.00      0.00      0.00      1865
         2.0       0.54      0.19      0.28     14038

    accuracy                           0.85    101472
   macro avg       0.47      0.39      0.40    101472
weighted avg       0.80      0.85      0.81    101472
```

Confusion matrix:

```
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_xgboost)

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:
 [[83105     0  2464]
 [ 1693     0   172]
 [11586     0  2452]]
Confusion Matrix - Random Forest:
 [[82682    58  2829]
 [ 1670     0   195]
 [11242    16  2780]]
Confusion Matrix - Decision Tree:
 [[73149  1765 10655]
 [ 1361    57   447]
 [ 9013   568  4457]]
Confusion Matrix - XGBoost:
 [[83494     0  2075]
 [ 1673     0   192]
 [11369     0  2669]]
```