Project Development Phase
Model Performance Test

Date	27 october 2023
Team ID	Team-592372
Project Name	**Project- Diabetes Prediction Using Machine Learning**
Maximum Marks	10 Marks

Model Performance Testing:

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

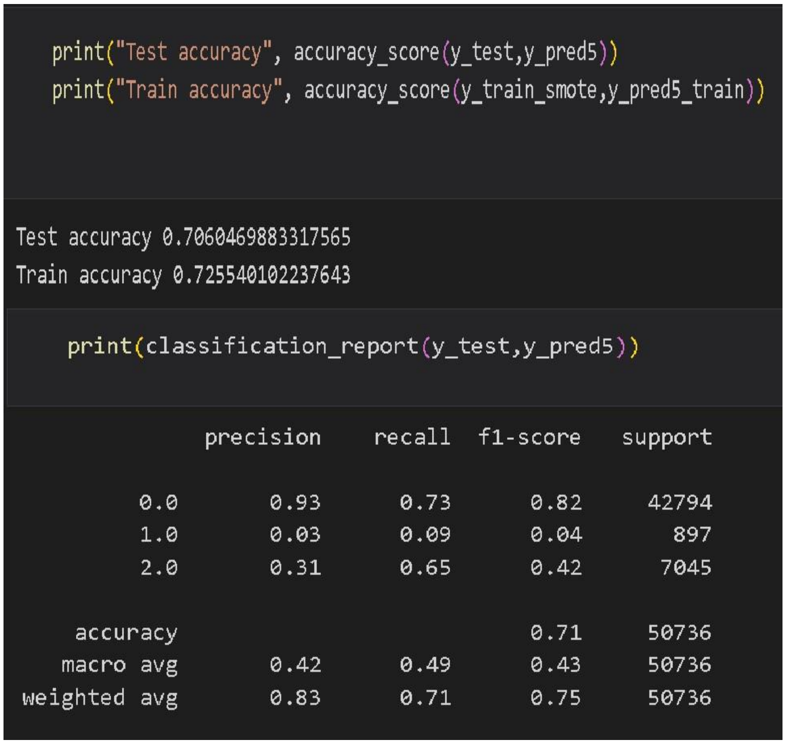
S.No.	**Parameter**	**Values**	**Screenshot** **(Random Forest Model)**

<table>
<tr>
<td>1.</td>
<td>Metrics</td>
<td>

Regression Model:
MAE - , MSE - , RMSE - , R2 score -

Classification Model:
Confusion Matrix - , Accuracy Score- & Classification Report -

</td>
<td>



```
print("Test accuracy", accuracy_score(y_test,y_pred5))
print("Train accuracy", accuracy_score(y_train_smote,y_pred5_train))
```

```
Test accuracy 0.7060469883317565
Train accuracy 0.725540102237643
```

```
print(classification_report(y_test,y_pred5))
```

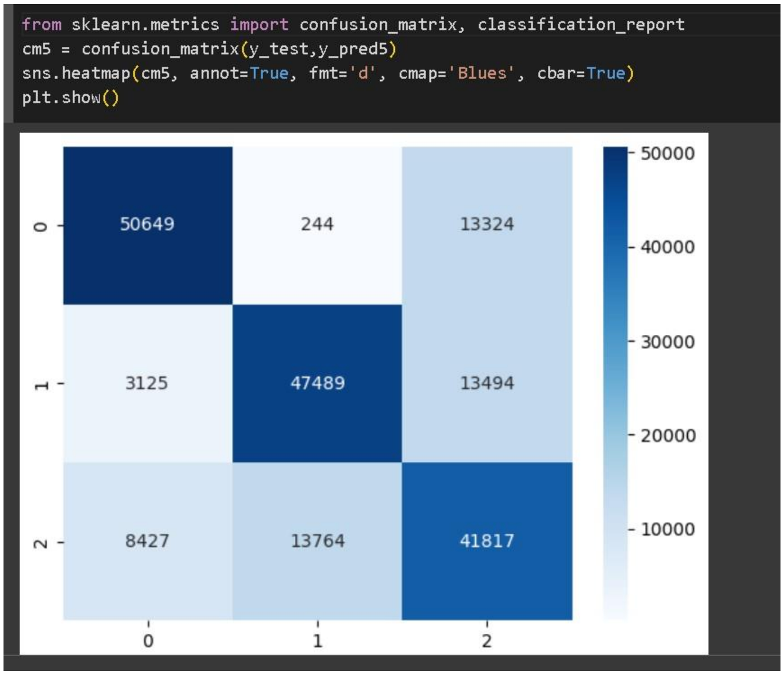
```
              precision    recall  f1-score   support

         0.0       0.93      0.73      0.82     42794
         1.0       0.03      0.09      0.04       897
         2.0       0.31      0.65      0.42      7045

    accuracy                           0.71     50736
   macro avg       0.42      0.49      0.43     50736
weighted avg       0.83      0.71      0.75     50736
```

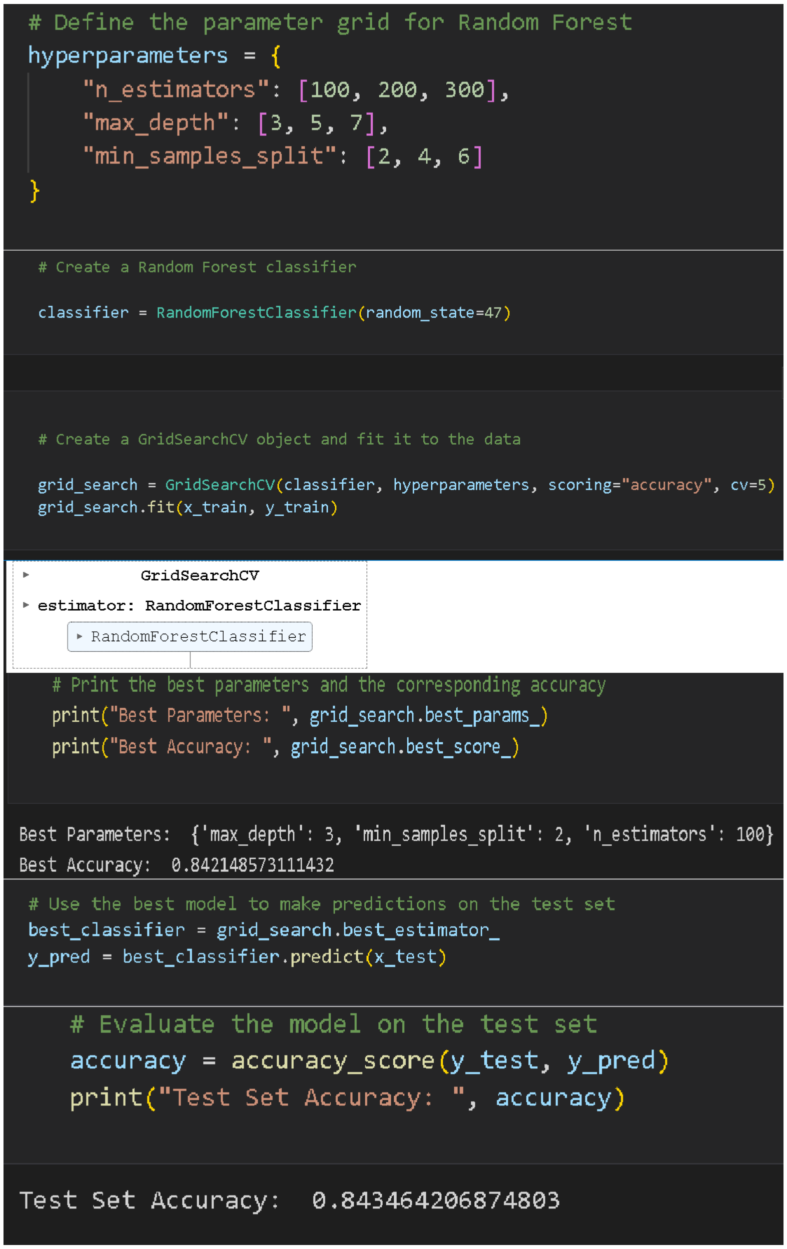
Classification Model:

</td>
</tr>
</table>



```
from sklearn.metrics import confusion_matrix, classification_report
cm5 = confusion_matrix(y_test,y_pred5)
sns.heatmap(cm5, annot=True, fmt='d', cmap='Blues', cbar=True)
plt.show()
```

	0	1	2
0	50649	244	13324
1	3125	47489	13494
2	8427	13764	41817

2.	Tune the Model	Hyperparameter Tuning - Validation Method -	

```
# Define the parameter grid for Random Forest
hyperparameters = {
    "n_estimators": [100, 200, 300],
    "max_depth": [3, 5, 7],
    "min_samples_split": [2, 4, 6]
}
```

```
# Create a Random Forest classifier

classifier = RandomForestClassifier(random_state=47)
```

```
# Create a GridSearchCV object and fit it to the data

grid_search = GridSearchCV(classifier, hyperparameters, scoring="accuracy", cv=5)
grid_search.fit(x_train, y_train)
```

```
▸          GridSearchCV
▸ estimator: RandomForestClassifier
      ▸ RandomForestClassifier
```

```
# Print the best parameters and the corresponding accuracy
print("Best Parameters: ", grid_search.best_params_)
print("Best Accuracy: ", grid_search.best_score_)
```

```
Best Parameters:  {'max_depth': 3, 'min_samples_split': 2, 'n_estimators': 100}
Best Accuracy:  0.842148573111432
```

```
# Use the best model to make predictions on the test set
best_classifier = grid_search.best_estimator_
y_pred = best_classifier.predict(x_test)
```

```
# Evaluate the model on the test set
accuracy = accuracy_score(y_test, y_pred)
print("Test Set Accuracy: ", accuracy)
```

```
Test Set Accuracy:  0.843464206874803
```