

## Project Development Phase Model Performance Test

Date	17th November 2022
Team ID	PNT2022TMID43513
Project Name	Project – Detecting Parkinson’s Disease 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
1.	Metrics	<b>Classification Model:</b> Confusion Matrix, F1 Score, Accuracy Score & Classification Report	<pre> from sklearn.metrics import confusion_matrix, f1_score, accuracy_score from sklearn.metrics import classification_report  # Confusion matrix cm = confusion_matrix(y_test, y_pred)  # F1 score f1 = f1_score(y_test, y_pred)  # Accuracy score acc = accuracy_score(y_test, y_pred)  # Classification report print(classification_report(y_test, y_pred)) </pre>
2.	Tune the Model	Data mining - XGBoost Classifier	<pre> from sklearn.model_selection import GridSearchCV from xgboost import XGBClassifier  # GridSearchCV grid = GridSearchCV(XGBClassifier(), param_grid, cv=5) grid.fit(X_train, y_train)  # Best parameters best_params = grid.best_params_  # Accuracy score accuracy = grid.best_score_ </pre>

### 1) Metrics Parameter screenshot

## ▼ XGBClassification - Supervised Machine Learning

```
✓ [31] Model_XG = XGBClassifier(random_state=0)
0s      Model_XG.fit(x_train,y_train)

      XGBClassifier()
```

## ▼ Assessing the model using metrics

```
✓ [32] y_predict = Model_XG.predict(x_test)
0s      print(accuracy_score(y_test,y_predict)*100)

      98.30508474576271
```

Hence by reducing the overfitting using XGBoost Classifier, we are getting accuracy\_score of **98.30%** for the model

## ▼ Confusion metrics

```
✓ [33] from sklearn.metrics import confusion_matrix
0s      ypre = Classification_model.predict(x_test)
      ypre = (ypre>0.5)
      confusion_matrix(y_test,ypre)

      array([[18,  6],
            [ 6, 29]])
```

## 2) Tune the model Parameter screenshot

```
[32] y_predict = Model_XG.predict(x_test)
print(accuracy_score(y_test,y_predict)*100)

98.30508474576271
```

Hence by reducing the overfitting using XGBoost Classifier, we are getting accuracy\_score of **98.30%** for the model

#### ▼ Confusion metrics

```
[33] from sklearn.metrics import confusion_matrix
ypre = Classification_model.predict(x_test)
ypre = (ypre>0.5)
confusion_matrix(y_test,ypre)

array([[18,  6],
       [ 6, 29]])
```

#### ▼ F1 score

```
[34] from sklearn.metrics import f1_score
Variation_score = f1_score(y_test, Model_XG.predict(x_test), average='binary')
print(Variation_score/0.01)

98.59154929577464
```

#### ▼ Classification report

```
[35] from sklearn import metrics
from sklearn.metrics import classification_report
print("\n Classification report for Model_XG:\n\n"% (Model_XG, metrics.classification_report(y_test, y_pred)))

Classification report for Model_XGClassifier():
precision    recall  f1-score   support
```