
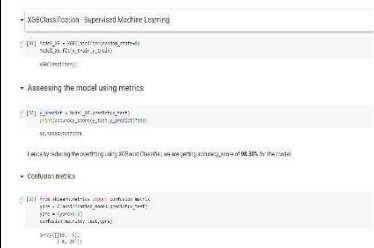


ProjectDevelopmentPhase ModelPerformanceTest

Date	17thNovember2022
TeamID	PNT2022TMID20255
ProjectName	Project– DetectingParkinson’sDiseaseusingMachineLearn ing
MaximumMarks	10Marks

ModelPerformanceTesting:

Projectteamshallfillthefollowinginformationinmodelperformancetestingtemplate.

S. No.	Parameter	Values	Screenshot
1.	Metrics	Classification Model:Confusion Matrix, F1 Score,AccuracyScore&Classific ationReport	
2.	TunetheModel	Data mining - XGBoostClassifier	

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)TunethemodelParameterscreenshot

```
[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 %s:\n%s\n" % (Model_XG, metrics.classification_report(y_test, y_pred)))

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