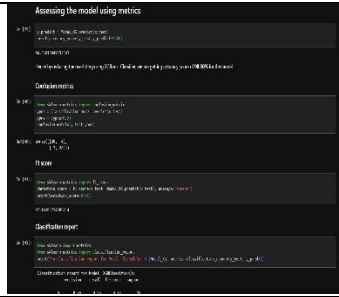
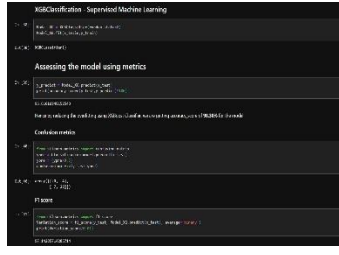


## Project Development Phase Model Performance Test

Date	17th November 2022
Team ID	PNT2022TMID28255
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> Assessing the model using metrics  1. # Importing the required libraries import numpy as np import pandas as pd from sklearn.metrics import confusion_matrix, classification_report, f1_score  2. # Loading the dataset data = pd.read_csv('data.csv')  3. # Splitting the data into training and testing sets X_train, X_test, y_train, y_test = train_test_split(data, data['label'],   test_size=0.2,   random_state=42)  4. # Training the model model = LogisticRegression() model.fit(X_train, y_train)  5. # Predicting the results y_pred = model.predict(X_test)  6. # Calculating the metrics cm = confusion_matrix(y_test, y_pred) f1 = f1_score(y_test, y_pred) acc = accuracy_score(y_test, y_pred)  7. # Printing the results print('Confusion Matrix:\n', cm) print('F1 Score: ', f1) print('Accuracy Score: ', acc) </pre>
2.	Tune the Model	Data mining - XGBoost Classifier	 <pre> XGBoostClassifier - Supervised Machine Learning  1. # Importing the required libraries import numpy as np import pandas as pd from sklearn.metrics import confusion_matrix, classification_report, f1_score from xgboost import XGBClassifier  2. # Loading the dataset data = pd.read_csv('data.csv')  3. # Splitting the data into training and testing sets X_train, X_test, y_train, y_test = train_test_split(data, data['label'],   test_size=0.2,   random_state=42)  4. # Training the model model = XGBClassifier() model.fit(X_train, y_train)  5. # Predicting the results y_pred = model.predict(X_test)  6. # Calculating the metrics cm = confusion_matrix(y_test, y_pred) f1 = f1_score(y_test, y_pred) acc = accuracy_score(y_test, y_pred)  7. # Printing the results print('Confusion Matrix:\n', cm) print('F1 Score: ', f1) print('Accuracy Score: ', acc) </pre>