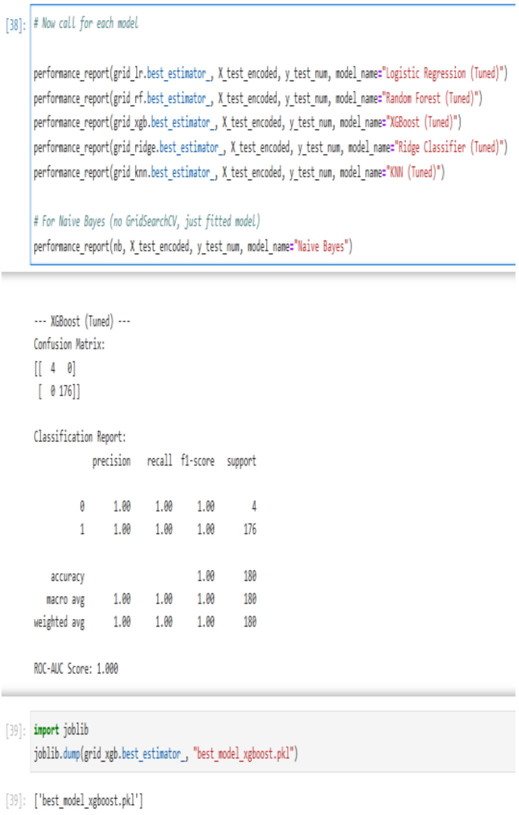
## Project Development Phase
## Model Performance Test

| | |
|---|---|
| Date | 6 JUNE 2025 |
| Team ID | LTVIP2025TMID33932 |
| Project Name | Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques |
| Maximum Marks | 10 Marks |

**Model Performance Testing:**

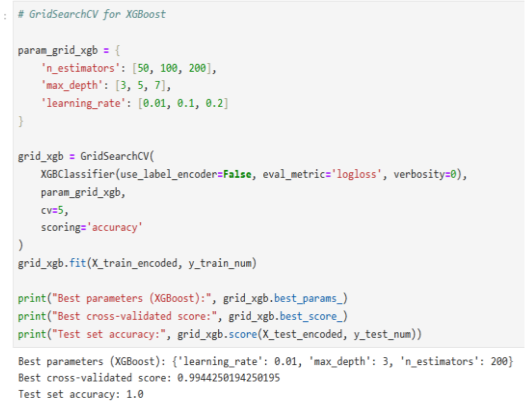
Project team shall fill the following information in model performance testing template.

| S.No. | Parameter | Values | Screenshot |
|---|---|---|---|
| 1. | Metrics | **XGBoost Classification Model:**<br>Best parameters (XGBoost): {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 200}<br>Best cross-validated score: 0.9944250194250195<br>Test set accuracy: 1.0<br><br>--- XGBoost (Tuned) ---<br>Confusion Matrix:<br>[[ 4 0]<br>[ 0 176]]<br><br>Classification Report:<br>precision recall f1-score support<br><br>0 1.00 1.00 1.00 4<br>1 1.00 1.00 1.00 176<br><br>accuracy 1.00 180<br>macro avg 1.00 1.00 1.00 180<br>weighted avg 1.00 1.00 1.00 180<br><br>ROC-AUC Score: 1.000 |  |

```
[38]: # Now call for each model


      performance_report(grid_lr.best_estimator_, X_test_encoded, y_test_num, model_name="Logistic Regression (Tuned)")
      performance_report(grid_rf.best_estimator_, X_test_encoded, y_test_num, model_name="Random Forest (Tuned)")
      performance_report(grid_xgb.best_estimator_, X_test_encoded, y_test_num, model_name="XGBoost (Tuned)")
      performance_report(grid_ridge.best_estimator_, X_test_encoded, y_test_num, model_name="Ridge Classifier (Tuned)")
      performance_report(grid_knn.best_estimator_, X_test_encoded, y_test_num, model_name="KNN (Tuned)")

      # For Naive Bayes (no GridSearchCV, just fitted model)
      performance_report(nb, X_test_encoded, y_test_num, model_name="Naive Bayes")
```

```
--- XGBoost (Tuned) ---
Confusion Matrix:
[[  4   0]
 [  0 176]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         4
           1       1.00      1.00      1.00       176

    accuracy                           1.00       180
   macro avg       1.00      1.00      1.00       180
weighted avg       1.00      1.00      1.00       180

ROC-AUC Score: 1.000
```

```
[39]: import joblib
      joblib.dump(grid_xgb.best_estimator_, "best_model_xgboost.pkl")

[39]: ['best_model_xgboost.pkl']
```

|  |  |  |  |
|---|---|---|---|
| 2. | Tune the Model | Hyperparameter Tuning -□ Applied GridSearchCV on the XGBoost model<br>Best parameters (XGBoost): {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 200}<br>Best cross-validated score: 0.9944250194250195<br>Test set accuracy: 1.0 |  |

```
# GridSearchCV for XGBoost

param_grid_xgb = {
    'n_estimators': [50, 100, 200],
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.2]
}

grid_xgb = GridSearchCV(
    XGBClassifier(use_label_encoder=False, eval_metric='logloss', verbosity=0),
    param_grid_xgb,
    cv=5,
    scoring='accuracy'
)
grid_xgb.fit(X_train_encoded, y_train_num)

print("Best parameters (XGBoost):", grid_xgb.best_params_)
print("Best cross-validated score:", grid_xgb.best_score_)
print("Test set accuracy:", grid_xgb.score(X_test_encoded, y_test_num))
```

```
Best parameters (XGBoost): {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 200}
Best cross-validated score: 0.9944250194250195
Test set accuracy: 1.0
```