

Project Development Phase Model Performance Test

Date	10 February 2025
Team ID	LTVIP2025TMID42458
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	<p>Regression Model: MAE - , MSE - , RMSE - , R2 score -</p> <p>Classification Model: Confusion Matrix - , Accuray Score- & Classification Report -</p>	<pre> # Import Libraries from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score, confusion_matrix, accuracy_score, classification_report # Data Preprocessing X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42) # Model Training model = RandomForestRegressor() model.fit(X_train, y_train) # Model Evaluation # Regression Metrics mae = mean_absolute_error(y_test, model.predict(X_test)) mse = mean_squared_error(y_test, model.predict(X_test)) rmse = sqrt(mse) r2 = r2_score(y_test, model.predict(X_test)) # Classification Metrics y_pred = model.predict(X_test) cm = confusion_matrix(y_test, y_pred) acc = accuracy_score(y_test, y_pred) cl_report = classification_report(y_test, y_pred) # Print Results print("MAE: ", mae) print("MSE: ", mse) print("RMSE: ", rmse) print("R2 Score: ", r2) print("Confusion Matrix: ") print(cm) print("Accuracy Score: ", acc) print("Classification Report: ") print(cl_report) </pre>
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	<pre> # Hyperparameter Tuning param_grid = { 'n_estimators': [50, 100, 200], 'max_depth': [None, 10, 20, 30], 'min_samples_split': [2, 5, 10] } # Create model rf = RandomForestClassifier(random_state=42) # Grid Search grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=5, n_jobs=-1) # Best model best_rf = grid_search.best_estimator_ print("Best Parameters: ", grid_search.best_params_) # Predict using the tuned model y_pred_tuned = best_rf.predict(X_test_scaled) # Model Performance (After Tuning) print("Accuracy: ", accuracy_score(y_test, y_pred_tuned)) print("Precision: ", precision_score(y_test, y_pred_tuned)) print("Recall: ", recall_score(y_test, y_pred_tuned)) print("F1 Score: ", f1_score(y_test, y_pred_tuned)) print("Confusion Matrix: ") print(confusion_matrix(y_test, y_pred_tuned)) print("Classification Report: ") print(classification_report(y_test, y_pred_tuned)) # Model Performance (After Tuning) accuracy = accuracy_score(y_test, y_pred_tuned) precision = precision_score(y_test, y_pred_tuned) recall = recall_score(y_test, y_pred_tuned) f1_score = f1_score(y_test, y_pred_tuned) # Print Results print("Accuracy: ", accuracy) print("Precision: ", precision) print("Recall: ", recall) print("F1 Score: ", f1_score) </pre>