Model Development Phase Template

Date	18th June 2024
Team ID	SWTID1720109498
Project Title	Blueberry Yield Predictor
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

This document outlines the process and results of developing and validating regression models to predict the outcomes for the Wild Blueberry Pollination Simulation dataset. The goal of this phase is to build accurate and reliable models by selecting relevant features, evaluating various algorithms, and fine-tuning hyperparameters. This report includes detailed evaluation metrics, visualization, and insights into the performance of each model, ensuring a comprehensive understanding of their predictive capabilities.

Initial Model Training Code:

```
# Importing and building Linear Regression model
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
model.score(X_test, y_test)

y_preds = model.predict(X_test)

print("Regression metrics on the test set")
print(f"R2 score: {r2_score(y_test, y_preds)}")
print(f"MAE: {mean_absolute_error(y_test,y_preds)}")
print(f"MSE: {mean_squared_error(y_test, y_preds)}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_preds))}")
```

```
# Importing and building the Random forest regressor model
from sklearn.ensemble import RandomForestRegressor

model2 = RandomForestRegressor()
model2.fit(X_train,y_train)

y_preds2 = model2.predict(X_test)

print("Regression metrics on the test set")
print(f"R2 score: {r2_score(y_test, y_preds2)}")
print(f"MAE: {mean_absolute_error(y_test,y_preds2)}")
print(f"MSE: {mean_squared_error(y_test, y_preds2)}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_preds2))}")
```

```
# Importing and building Decision tree regressor model
from sklearn.tree import DecisionTreeRegressor

model3 = DecisionTreeRegressor()
model3.fit(X_train,y_train)
y_preds3 = model3.predict(X_test)

print("Regression metrics on the test set")
print(f"R2 score: {r2_score(y_test, y_preds3)}")
print(f"MAE: {mean_absolute_error(y_test,y_preds3)}")
print(f"MSE: {mean_squared_error(y_test, y_preds3)}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_preds3))}")
```

```
# Importing and building XGB Regressor model
from xgboost import XGBRegressor
model4 = XGBRegressor()
model4.fit(X_train,y_train)
model4.score(X_test,y_test)

y_preds4 = model4.predict(X_test)

print("Regression metrics on the test set")
print(f"R2 score: {r2_score(y_test, y_preds4)}")
print(f"MAE: {mean_absolute_error(y_test,y_preds4)}")
print(f"MSE: {mean_squared_error(y_test, y_preds4)}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_preds4))}")
```

Model Validation and Evaluation Report:

Model	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	R2 Score	Screenshorts of the evaluation report

Linear Regression	111.5204986497179	21684.627147497344	0.9850297685259484	`print("Regression metrics on the test set")`
				`print(f"R2 score: {r2_score(y_test, y_preds)}")`
				`print(f"MAE: {mean_absolute_error(y_test,y_preds)}")`
				`print(f"MSE: {mean_squared_error(y_test, y_preds)}")`
				`print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_preds))}")`
				Regression metrics on the test set
				R2 score: 0.9850297685259484
				MAE: 111.5204986497179
				MSE: 21684.627147497344
				RMSE: 147.25701052071287

Random Forest Regressor	133.30059127843145	30238.378584205282	0.9791245879522628	`print("Regression metrics on the test set")`
				`print(f"R2 score: {r2_score(y_test, y_preds2)}")`
				`print(f"MAE: {mean_absolute_error(y_test,y_preds2)}")`
				`print(f"MSE: {mean_squared_error(y_test, y_preds2)}")`
				`print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_preds2))}")`
				Regression metrics on the test set
				R2 score: 0.9791245879522628
				MAE: 133.30059127843145
				MSE: 30238.378584205282
				RMSE: 173.8918588784572

Decision Tree Regressor	177.92750329411768	51633.51043756918	0.9643542128803687	`print("Regression metrics on the test set")`
				`print(f"R2 score: {r2_score(y_test, y_preds3)}")`
				`print(f"MAE: {mean_absolute_error(y_test,y_preds3)}")`
				`print(f"MSE: {mean_squared_error(y_test, y_preds3)}")`
				`print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_preds3))}")`
				Regression metrics on the test set
				R2 score: 0.9643542128803687
				MAE: 177.92750329411768
				MSE: 51633.51043756918
				RMSE: 227.23008259816564

XGB Regressor	119.19604783486517	25823.161181161377	0.9821726840104634	`print("Regression metrics on the test set")`
				`print(f"R2 score: {r2_score(y_test, y_preds4)}")`
				`print(f"MAE: {mean_absolute_error(y_test,y_preds4)}")`
				`print(f"MSE: {mean_squared_error(y_test, y_preds4)}")`
				`print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_preds4))}")`
				Regression metrics on the test set
				R2 score: 0.982172684010463
				MAE: 119.19604783486517
				MSE: 25823.161181161377
				RMSE: 160.69586547625107

This bar chart compares the performance of four regression models—Linear Regression, Random Forest, Decision Tree, and XGBoost—using three evaluation metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2).