Model Development Phase Template

Date	15 June 2024
Team ID	739693
Project Title	Predicting the unpredictable: A Look into the World of Powerlifting
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

#importing and building the linear regression model:

```
lr = LinearRegression()
lr.fit(x_train,y_train)
y_pred1 = lr.predict(x_test)
```

```
mse = mean_squared_error(y_test,y_pred1)
 print("Mean Squared Error:", mse)
 rmse = np.sqrt(mse)
```

#importing and building the RandomRegression model:

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
rf.fit(x_train,y_train)
y_pred2 = rf.predict(x_test)
```

```
mse= mean_squared_error(y_test,y_pred2)
print("Mean Squared Error:", mse)
rmse = np.sqrt(mse)
```

```
print("RMSE value:(:.2f)",format(rmse))
```

```
print("Training Accuracy for RandomForest: {:.2f}",format(rf.score(x_train,y_train)*100),'%')
print("Testing Accuracy for RandomForest: {:.2f}",format(rf.score(x_test,y_test)*100),'%')
```

#importing and building DecisionTree Model:

```
from sklearn.tree import DecisionTreeRegressor
dt = DecisionTreeRegressor()
dt.fit(x_train,y_train)
y_pred3 = dt.predict(x_test)
```

```
mse= mean_squared_error(y_test,y_pred3)
print("Mean Squared Error:", mse)
rmse = np.sqrt(mse)
```

```
print("RMSE value:(:.2f)",format(rmse))
```

```
print("Training Accuracy for DecisionTree: {:.2f}",format(dt.score(x_train,y_train)*100),'%')
print("Testing Accuracy for DecisionTree: {:.2f}",format(dt.score(x_test,y_test)*100),'%')
```

#importing and building the XGboost Model:

```
import xgboost as xgb
xgb_model = xgb.XGBRegressor()
xgb_model.fit(x_train,y_train)
y_pred4 = xgb_model.predict(x_test)
```

```
mse= mean_squared_error(y_test,y_pred4)
print("Mean Squared Error:", mse)
rmse = np.sqrt(mse)
```

```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred1)
```

```
print("RMSE value:(:.2f)",format(rmse))
```

```
print("Training Accuracy for XGBoost: {:.2f}",format(xgb_model.score(x_train,y_train)*100),'%')
```

```
print("Testing Accuracy for XGBoost: {:.2f}",format(xgb_model.score(x_test,y_test)*100),'%')
```

Model Validation and Evaluation Report:

Model	RMSE	F1 score	Accuracy
Linear regression	21.814068813999864 |	21.814068813999864 |	Training Accuracy: 79.55925360084694 Testing Accuracy:| 79.95753541725657

Random forest	21.814068813999864	21.814068813999864 |	Training Accuracy: 98.32882732596175 Testing Accuracy: 87.81981945246893 |
Decision Tree	21.814068813999864 |	21.814068813999864 |	Training Accuracy: 100.0 Testing Accuracy: 77.16769363515307
XGBoost	21.814068813999864 |	21.814068813999864 |	Training Accuracy: 94.51461911481135 Testing Accuracy: 87.72510461710547