Project Development Phase
Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID35768
Project Name	Project - University Admit Eligibility Predictor
Maximum Marks	10 Marks

Model Performance Testing:

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

S.No.	Parameter	Values	Screenshot

1.	Metrics	**Regression Model**
		
		MAE -0.0390254623838967,
		MSE -0.0029806758228552222,
		RMSE -0.05459556596331997,
		R2 score -0.835933486388181.
			`from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error`
			`import numpy as np`
			`print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))`
			`print('Mean Squared Error:', mean_squared_error(y_test, y_pred))`
			`print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))`
			`print('R2 Error:', r2_score(y_test, y_pred))`
			
			Mean Absolute Error: 0.03909254623838967
			Mean Squared Error: 0.0029806758228552222
			Root Mean Squared Error: 0.05459556596331997
			R2 Error: 0.8359334863688181

2.	Tune the model	**Hyperparameter Tuning –** GridSearchCv with Repeated 10 Folds is used to find the set of hyperparameters for the given training set. **Validation Method -**	

Hyperparameter Tuning and Validation Method:

```
from sklearn.model_selection import RepeatedKFold
from sklearn.model_selection import GridSearchCV
```

```
# Hyperparameter Tuning + CV
grid = dict()
grid['n_estimators'] = [10, 50, 100, 500]
grid['learning_rate'] = [0.0001, 0.001, 0.01, 0.1, 1.0]
grid['subsample'] = [0.5, 0.7, 1.0]
grid['max_depth'] = [3, 7, 9]

cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)

grid_search = GridSearchCV(estimator=model, param_grid=grid, n_jobs=-1, cv=cv)

grid_result = grid_search.fit(X_train, y_train)
# summarize the best score and configuration
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_)) # summarize all scores that were evaluated
```

```
Best: 0.767087 using {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 500, 'subsample': 0.5}
```

```
best_model = grid_result.best_estimator_
```

```
y_pred = best_model.predict(X_test)
```

```
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
import numpy as np
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
print('R2 Error:', r2_score(y_test, y_pred))
```

```
Mean Absolute Error: 0.03909254623838967
Mean Squared Error: 0.0029806758228552222
Root Mean Squared Error: 0.05459556596331997
R2 Error: 0.8359334863688181
```