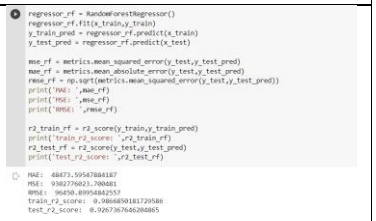
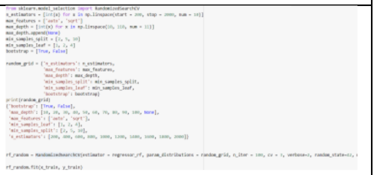
Project Development Phase
Model Performance Test

Date	18 November 2022
Team ID	PNT2022TMID02157
Project Name	Project – Car resale value prediction
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	**Regression Model:** MAE - 48473.59547884187, MSE - 9302776023.700481, RMSE - 96450.89954842557, R2 score – Training - 0.9866850181729586 R2 score – Training – 0.9267367646204865	
2.	Tune the Model	Hyperparameter Tuning – n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf, bootstrap Validation Method - Cross Validation	

```
regressor_rf = RandomForestRegressor()
regressor_rf.fit(x_train,y_train)
y_train_pred = regressor_rf.predict(x_train)
y_test_pred = regressor_rf.predict(x_test)

mse_rf = metrics.mean_squared_error(y_test,y_test_pred)
mae_rf = metrics.mean_absolute_error(y_test,y_test_pred)
rmse_rf = np.sqrt(metrics.mean_squared_error(y_test,y_test_pred))
print('MAE: ',mae_rf)
print('MSE: ',mse_rf)
print('RMSE: ',rmse_rf)

r2_train_rf = r2_score(y_train,y_train_pred)
print('train_r2_score: ',r2_train_rf)
r2_test_rf = r2_score(y_test,y_test_pred)
print('test_r2_score: ',r2_test_rf)
```

```
MAE:  48473.59547884187
MSE:  9302776023.700481
RMSE:  96450.89954842557
train_r2_score:  0.9866850181729586
test_r2_score:  0.9267367646204865
```

```
from sklearn.model_selection import RandomizedSearchCV
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
min_samples_split = [2, 5, 10]
min_samples_leaf = [1, 2, 4]
bootstrap = [True, False]

random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
print(random_grid)
{'bootstrap': [True, False],
 'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None],
 'max_features': ['auto', 'sqrt'],
 'min_samples_leaf': [1, 2, 4],
 'min_samples_split': [2, 5, 10],
 'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]}


rf_random = RandomizedSearchCV(estimator = regressor_rf, param_distributions = random_grid, n_iter = 100, cv = 3, verbose=2, random_state=42,

rf_random.fit(x_train, y_train)
```