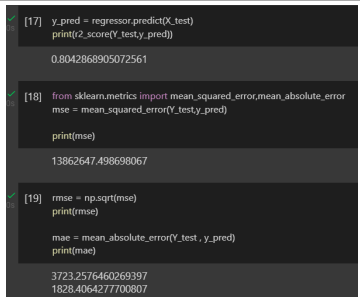



## Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID21521
Project Name	Project - Car Resale Value Prediction
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	<b>Regression Model: Random Forest</b>  MAE - 1828.4064277700807 MSE -13862647.498698067 RMSE -3723.2576460269397 Accuracy - 80.43%	 <pre> [17] y_pred = regressor.predict(X_test) print(f2_score(Y_test,y_pred))  0.8042868905072561  [18] from sklearn.metrics import mean_squared_error,mean_absolute_error mse = mean_squared_error(Y_test,y_pred)  print(mse)  13862647.498698067  [19] rmse = np.sqrt(mse) print(rmse)  3723.2576460269397 print(mae)  1828.4064277700807           </pre>
2.	Tune the Model	<b>Hyperparameter Tuning</b>  1) Learning Rate: [0.01, 0.03, 0.05, 0.07] 2) Max features: ['auto','sqrt'] 3) Number of Estimators: [10,20,30,50]  4) min_samples_leaf : [2,4,6]  <b>Validation Method:</b> Grid Search Cross Validation  <b>Best Parameters:</b> Learning Rate – 0.07 Number of Estimators - 300	 <pre> [26] X_estimators = [10,20,30,50] max_features = ['auto','sqrt'] max_depth = [None for x in range(10,50, num = 10)] min_samples_split = [2, 4, 6] min_samples_leaf = [1, 4, 6] bootstrapping = [True,False]  random_grid = {'n_estimators': X_estimators,                'max_features': max_features,                'max_depth': max_depth,                'min_samples_split': min_samples_split,                'min_samples_leaf': min_samples_leaf }  [27] from sklearn.model_selection import RandomizedSearchCV # Random = RandomizedSearchCV(Regressor = regressor,param_distributions = random_grid, #                             n_iter = 10,cv = 5,verbose=2)  # Random = RSCV(X_train,Y_train)  RandomizedSearchCV(cv=5,                   estimator=RandomForestRegressor(max_depth=10,                                                     n_estimators=100,                                                     random_state=34),                   param_distributions={'max_depth': [10, 15, 21, 27, 32, 38,                                                     44, 50],                                       'max_features': ['auto', 'sqrt'],                                       'min_samples_leaf': [2, 4, 6],                                       'min_samples_split': [4, 6, 10],                                       'n_estimators': [10, 20, 30, 50]},                   verbose=2)           </pre>

## Screenshots

### 1) Metrics

#### Accuracy

```
✓ [17] y_pred = regressor.predict(X_test)
0s      print(r2_score(Y_test,y_pred))

0.8042868905072561
```

#### Mean squared error

```
✓ [18] from sklearn.metrics import mean_squared_error,mean_absolute_error
0s      mse = mean_squared_error(Y_test,y_pred)

      print(mse)

13862647.498698067
```

#### Root mean squared error & mean absolute error

```
✓ [19] rmse = np.sqrt(mse)
0s      print(rmse)

      mae = mean_absolute_error(Y_test , y_pred)
      print(mae)

3723.2576460269397
1828.4064277700807
```

### 2) Hyperparameter tuning

```
✓ [24] n_estimators = [10,20,30,50]
0s      max_features = ['auto', 'sqrt']
      max_depth = [int(x) for x in np.linspace(10, 50, num = 8)]
      min_samples_split = [4, 8, 10]
      min_samples_leaf = [2, 4, 6]
      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 }
```

```
✓ [27] from sklearn.model_selection import RandomizedSearchCV
0s      rf_random = RandomizedSearchCV(estimator = regressor,param_distributions = random_grid,
      n_iter = 10, cv = 5, verbose=2)
```

```
✓ 18s [▶] rf_random.fit(X_train, Y_train)
```

## Output :

```
[-] Fitting 5 folds for each of 10 candidates, totalling 50 fits
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
estimator.fit(X_train, y_train, **fit_params)
[CV] END max_depth=21, max_features=auto, min_samples_leaf=2, min_samples_split=4, n_estimators=50; total time= 1.3s
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
estimator.fit(X_train, y_train, **fit_params)
[CV] END max_depth=21, max_features=auto, min_samples_leaf=2, min_samples_split=4, n_estimators=50; total time= 1.2s
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
estimator.fit(X_train, y_train, **fit_params)
[CV] END max_depth=21, max_features=auto, min_samples_leaf=2, min_samples_split=4, n_estimators=50; total time= 1.2s
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
estimator.fit(X_train, y_train, **fit_params)
[CV] END max_depth=21, max_features=auto, min_samples_leaf=2, min_samples_split=4, n_estimators=50; total time= 1.2s
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
estimator.fit(X_train, y_train, **fit_params)
[CV] END max_depth=15, max_features=auto, min_samples_leaf=2, min_samples_split=4, n_estimators=10; total time= 0.2s
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
estimator.fit(X_train, y_train, **fit_params)
[CV] END max_depth=15, max_features=auto, min_samples_leaf=2, min_samples_split=4, n_estimators=10; total time= 0.2s
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
estimator.fit(X_train, y_train, **fit_params)
[CV] END max_depth=15, max_features=auto, min_samples_leaf=2, min_samples_split=4, n_estimators=10; total time= 0.2s
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
estimator.fit(X_train, y_train, **fit_params)
[CV] END max_depth=15, max_features=auto, min_samples_leaf=2, min_samples_split=4, n_estimators=10; total time= 0.2s
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
estimator.fit(X_train, y_train, **fit_params)
```

```
RandomizedSearchCV(cv=5,
                    estimator=RandomForestRegressor(max_depth=10,
                                                    n_estimators=1000,
                                                    random_state=34),
                    param_distributions={'max_depth': [10, 15, 21, 27, 32, 38,
                                                    44, 50],
                                        'max_features': ['auto', 'sqrt'],
                                        'min_samples_leaf': [2, 4, 6],
                                        'min_samples_split': [4, 8, 10],
                                        'n_estimators': [10, 20, 30, 50]},
                    verbose=2)
```