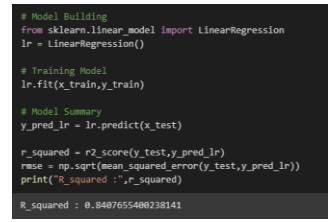
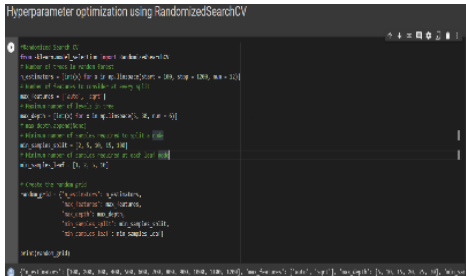


## Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID21102
Project Name	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	<b>Regression Model:</b> R2 score – 0.8407655400238141	 <pre># Model Building from sklearn.linear_model import LinearRegression lr = LinearRegression()  # Training Model lr.fit(x_train,y_train)  # Model Summary y_pred_lr = lr.predict(x_test)  r_squared = r2_score(y_test,y_pred_lr) rmse = np.sqrt(mean_squared_error(y_test,y_pred_lr)) print("R_squared :",r_squared)  R_squared : 0.8407655400238141</pre>
2.	Tune the Model	Hyperparameter Tuning - 0.8016218757946758	 <pre>Hyperparameter optimization using RandomizedSearchCV  # Importing Libraries from sklearn.linear_model import LogisticRegression from sklearn.metrics import accuracy_score from sklearn.model_selection import RandomizedSearchCV from sklearn.datasets import load_iris  # Load the dataset iris = load_iris() X = iris.data[:, :4] # Features y = iris.target # Target  # Define the parameter grid param_grid = {     'C': [0.1, 1, 10],     'max_depth': [None, 5, 10],     'max_features': ['auto', 'sqrt'],     'min_samples_split': [2, 5, 10],     'min_samples_leaf': [1, 2, 4] }  # Create the model logit = LogisticRegression()  # Create the RandomizedSearchCV object rs = RandomizedSearchCV(logit, param_grid, cv=5, n_iter=10, verbose=1)  # Fit the model rs.fit(X, y)  # Print the best parameters print("Best parameters found: ", rs.best_params_)  # Print the best score print("Best score: ", rs.best_score_)  # Print the accuracy of the best model print("Accuracy of the best model: ", accuracy_score(y, rs.predict(X)))</pre>