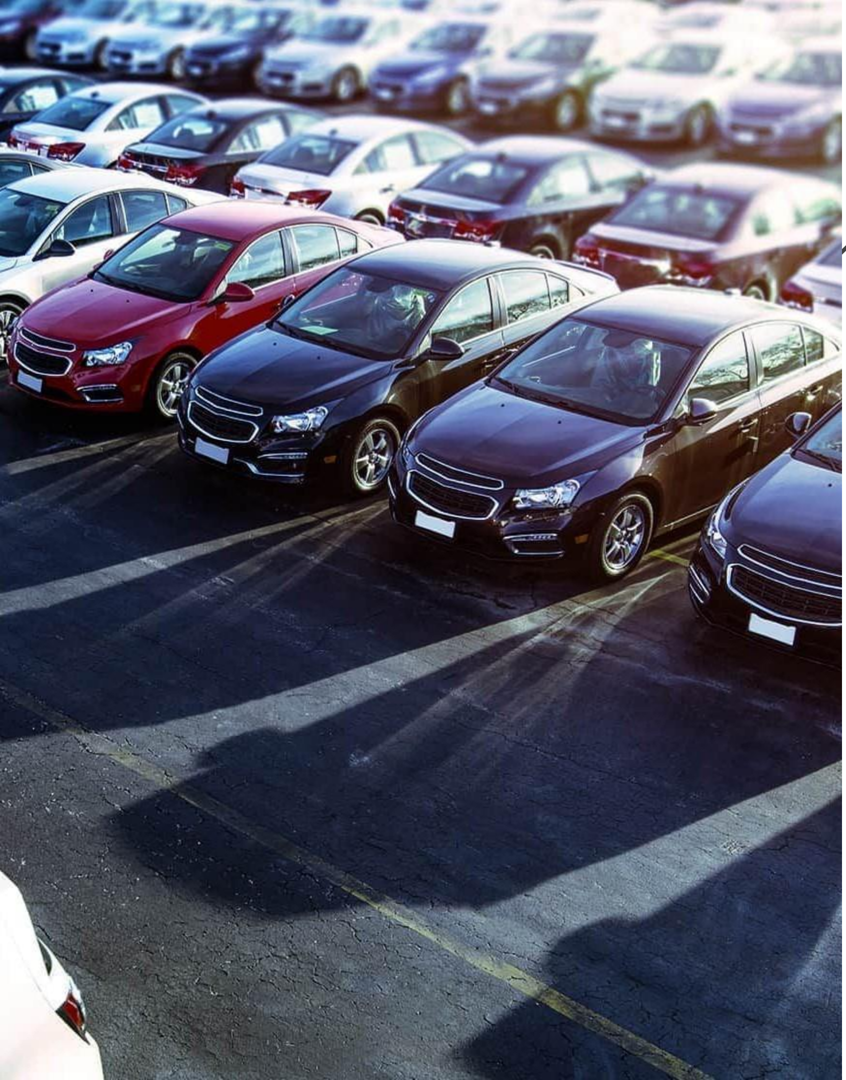


Price Prediction of Used Cars Using Machine Learning

By

Taufiqul Alam
Md. Tamjid Hosain
Summiya Sunjida Kashpia

01

Description of Dataset

Data set

- The dataset that we used in our model is taken from Kaggle.
- In our dataset there were 2059 rows and 20 attributes.
- The attributes of the cars that were given in our dataset were:

  Make, Model, Price, Year, Kilometer Driven, Fuel type, Transmission, Location, Color, Number of Owners, Seller Type, Engine Capacity, Max Power, Max Torque, Length, Width, Height, Drivetrain, Seating Capacity and Fuel Tank Capacity.

Make	Model	Price	Year	Kilometer	Fuel Type	Transmiss	Location	Color	Owner	Seller Typ	Engine	Max Powe	Max Torqu	Drivetrain	Length	Width	Height	Seating Ca	Fuel Tank
Honda	Amaze 1.2	505000	2017	87150	Petrol	Manual	Pune	Grey	First	Corporate	1198 cc	87 bhp @	109 Nm @	FWD	3990	1680	1505	5	35
Maruti Su	Swift DZir	450000	2014	75000	Diesel	Manual	Ludhiana	White	Second	Individual	1248 cc	74 bhp @	190 Nm @	FWD	3995	1695	1555	5	42
Hyundai	i10 Magna	220000	2011	67000	Petrol	Manual	Lucknow	Maroon	First	Individual	1197 cc	79 bhp @	112.7619 N	FWD	3585	1595	1550	5	35
Toyota	Glanza G	799000	2019	37500	Petrol	Manual	Mangalore	Red	First	Individual	1197 cc	82 bhp @	113 Nm @	FWD	3995	1745	1510	5	37
Toyota	Innova 2.4	1950000	2018	69000	Diesel	Manual	Mumbai	Grey	First	Individual	2393 cc	148 bhp @	343 Nm @	RWD	4735	1830	1795	7	55
Maruti Su	Ciaz ZXi	675000	2017	73315	Petrol	Manual	Pune	Grey	First	Individual	1373 cc	91 bhp @	130 Nm @	FWD	4490	1730	1485	5	43
Mercedes	CLA 200 Pe	1898999	2015	47000	Petrol	Automati	Mumbai	White	Second	Individual	1991 cc	181 bhp @	300 Nm @	FWD	4630	1777	1432	5	
BMW	X1 xDrive2	2650000	2017	75000	Diesel	Automati	Coimbato	White	Second	Individual	1995 cc	188 bhp @	400 Nm @	AWD	4439	1821	1612	5	51
Skoda	Octavia 1.	1390000	2017	56000	Petrol	Automati	Mumbai	White	First	Individual	1798 cc	177 bhp @	250 Nm @	FWD	4670	1814	1476	5	50
Nissan	Terrano XI	575000	2015	85000	Diesel	Manual	Mumbai	White	First	Individual	1461 cc	84 bhp @	200 Nm @	FWD	4331	1822	1671	5	50

Data set

- We used recursive feature elimination to discard all the irrelevant features. After using this algorithm, the remaining 12 attributes of the car were:

  Make, Price, Year, Kilometer Driven, Fuel type, Transmission, Number of Owners, Seller Type, Engine Capacity, Drivetrain, Seating Capacity, Fuel Tank Capacity, and Price.

Make	Price	Year	Kilometer	Fuel Type	Transmiss	Owner	Seller Typ	Engine	Drivetrain	Seating Ca	Fuel Tank
Honda	505000	2017	87150	Petrol	Manual	First	Corporate	1198 cc	FWD	5	35
Maruti Su:	450000	2014	75000	Diesel	Manual	Second	Individual	1248 cc	FWD	5	42
Hyundai	220000	2011	67000	Petrol	Manual	First	Individual	1197 cc	FWD	5	35
Toyota	799000	2019	37500	Petrol	Manual	First	Individual	1197 cc	FWD	5	37
Toyota	1950000	2018	69000	Diesel	Manual	First	Individual	2393 cc	RWD	7	55
Maruti Su:	675000	2017	73315	Petrol	Manual	First	Individual	1373 cc	FWD	5	43
Mercedes	1898999	2015	47000	Petrol	Automati	Second	Individual	1991 cc	FWD	5	
BMW	2650000	2017	75000	Diesel	Automati	Second	Individual	1995 cc	AWD	5	51
Skoda	1390000	2017	56000	Petrol	Automati	First	Individual	1798 cc	FWD	5	50
Nissan	575000	2015	85000	Diesel	Manual	First	Individual	1461 cc	FWD	5	50

Preprocessing the Dataset

- In the dataset we had 2059 rows. But out of those rows there were 185 rows with null values. So the rows with null values were removed and at the end we had 1874 rows in our dataset.

- All the numerical values in the engine attribute ended with the string ‘cc’. To treat it as numerical data, the string ‘cc’ had to be removed and after removing it the values were converted to integers. So, in the end, the values in the engine attribute were all numerical data.

- The pricing values for the cars listed in the dataset were excessively high (from thousands to crores). Due to this large range of target values, a scaling method called MinMaxScaler was used to bring the values to a comparable range which is between 0 and 1.

- The categorical variables in the dataset were Make, Fuel Type, Transmission, Owner, Seller Type, Drivetrain, Seating Capacity, and Fuel Tank Capacity. All the categorical values were encoded into a binary vector representation using one-hot encoding.

02

Target and Motivation

Target

Based on the given attributes the target variable that our model aims to predict is the selling price of the cars.

The features of the car that will be utilized to predict the selling price of the cars are: Make, Year, Kilometer Driven, Fuel type, Transmission, Number of Owners, Seller Type, Engine Capacity, Drivetrain, Seating Capacity and Fuel Tank Capacity.

Make	Price	Year	Kilometer	Fuel Type	Transmiss	Owner	Seller Typ	Engine	Drivetrain	Seating Ca	Fuel Tank
Honda	505000	2017	87150	Petrol	Manual	First	Corporate	1198 cc	FWD	5	35
Maruti Su	450000	2014	75000	Diesel	Manual	Second	Individual	1248 cc	FWD	5	42
Hyundai	220000	2011	67000	Petrol	Manual	First	Individual	1197 cc	FWD	5	35
Toyota	799000	2019	37500	Petrol	Manual	First	Individual	1197 cc	FWD	5	37
Toyota	1950000	2018	69000	Diesel	Manual	First	Individual	2393 cc	RWD	7	55
Maruti Su	675000	2017	73315	Petrol	Manual	First	Individual	1373 cc	FWD	5	43
Mercedes	1898999	2015	47000	Petrol	Automatic	Second	Individual	1991 cc	FWD	5	
BMW	2650000	2017	75000	Diesel	Automatic	Second	Individual	1995 cc	AWD	5	51
Skoda	1390000	2017	56000	Petrol	Automatic	First	Individual	1798 cc	FWD	5	50
Nissan	575000	2015	85000	Diesel	Manual	First	Individual	1461 cc	FWD	5	50

Motivation

The price of brand-new cars has significantly increased in recent years as a result of the global economic crisis. As a result, used car purchases are becoming more and more popular. However, figuring out a used car's fair market value can be difficult because it depends on a number of things, including the overall condition brand, model, year, mileage, transmission, etc. Uninformed buyers often fall prey to inflated price and end up paying a price which is not worth the car’s value.

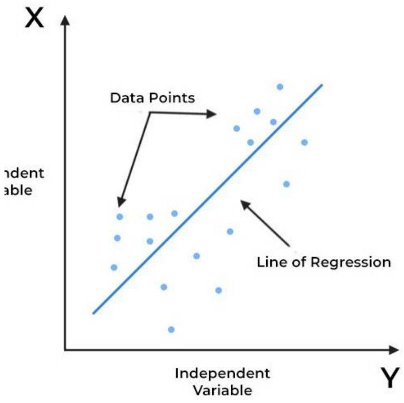
To address this issue and help buyers and sellers make informed decisions, we aim to develop a system that accurately predicts the prices of used cars based on their different attributes. This will empower buyers to assess the value of a used car and negotiate a fair

price, ensuring they get the best deal possible. Likewise, sellers can set realistic prices based on the provided information, fostering transparency and fairness in the used car market.

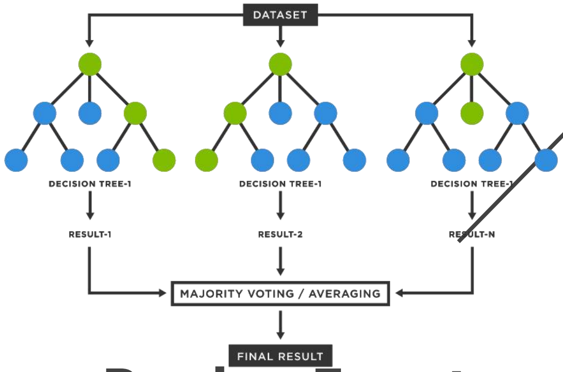
03

Designing the Model

Regression Algorithms Used



Linear Regression



Random Forest Regression

Linear Regression (Training Data)

Splitting Training and Testing Data

```
[20] X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1, random_state = 2)
```

Linear Regression

```
[21] lin_reg_model = LinearRegression()
```

```
[22] lin_reg_model.fit(X_train, Y_train)
```

Model Evaluation

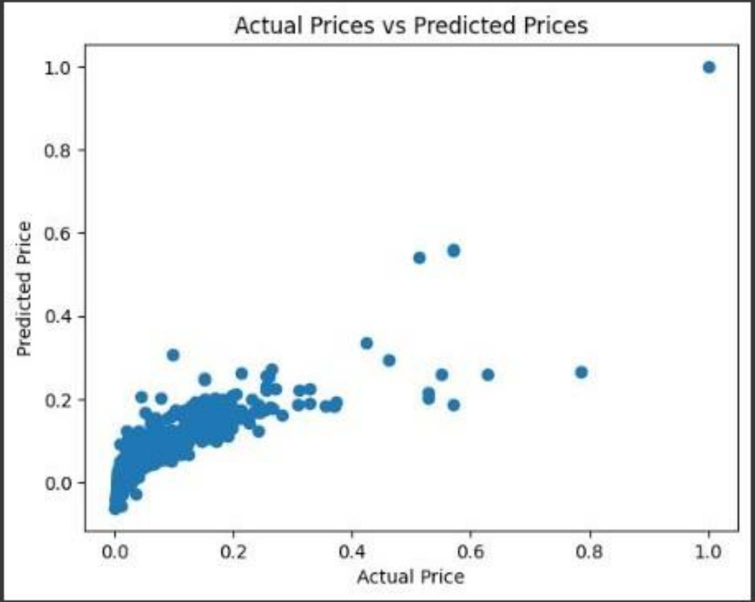
```
[70] # Prediction on Training Data
     training_data_prediction = lin_reg_model.predict(X_train)
```

```
[71] # R Squared Error
     error_score = metrics.r2_score(Y_train, training_data_prediction)
     print("R squared Error : ", error_score)
```

```
R squared Error :  0.7607724473925945
```

Linear Regression
(Scatter Plot for Training Data)

```
[72] plt.scatter(Y_train, training_data_prediction)
     plt.xlabel("Actual Price")
     plt.ylabel("Predicted Price")
     plt.title("Actual Prices vs Predicted Prices")
     plt.show()
```

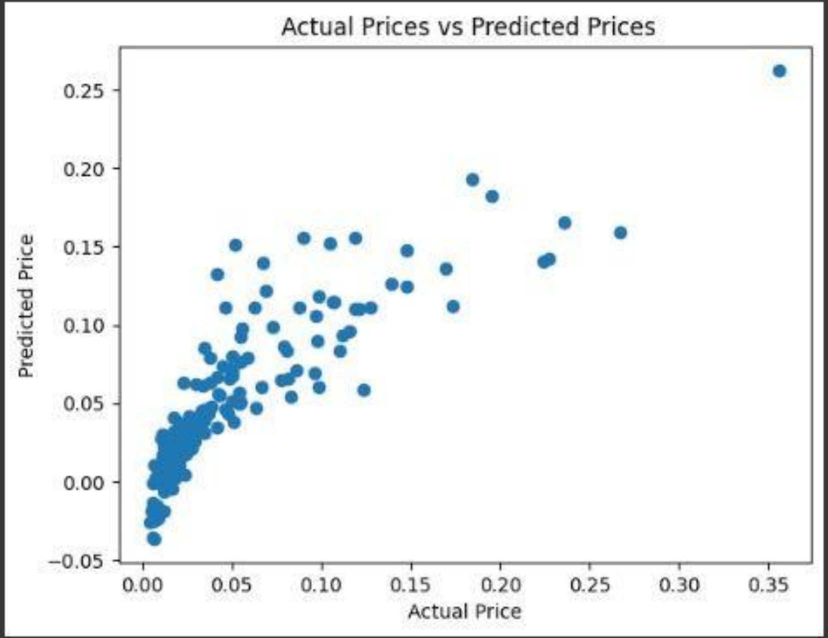


Linear Regression (Testing Data)

```
[73] # prediction on Test data
     test_data_prediction = lin_reg_model.predict(X_test)

[94] # R squared Error
     error_score = metrics.r2_score(Y_test, test_data_prediction)
     print("R squared Error : ", error_score)

     rmse = sqrt(metrics.mean_squared_error(Y_test,test_data_prediction))
     print("Root Mean squared Error : ", rmse)

     R squared Error :  0.848614052666839
     Root Mean squared Error :  0.02072945033644719
```

Linear Regression
(Scatter Plot for Testing Data)

```
[75] plt.scatter(Y_test, test_data_prediction)
     plt.xlabel("Actual Price")
     plt.ylabel("Predicted Price")
     plt.title(" Actual Prices vs Predicted Prices")
     plt.show()
```



Linear Regression
(K-Fold Cross Validation)

CROSS VALIDATION (Linear Regression)

[97]

```
scores = cross_val_score(LinearRegression(), X, Y, cv=10)
accuracies = cross_val_score(LinearRegression(), X, Y,scoring='neg_root_mean_squared_error', cv=10)
print("Cross-validation scores:", scores)
print('\n')
print("R2:", scores.mean())
print("RMSE score:", -accuracies.mean())
```

```
Cross-validation scores: [0.7211052  0.55787067 0.46623215 0.74512589 0.79184665 0.65818223
 0.70801287 0.71603444 0.75313744 0.58246746]


R2: 0.67000150076733
RMSE score: 0.03844546835472814
```

Random Forest Regression (Training Data)

Splitting Training and Testing Data

```
[19] X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1, random_state = 2)
```

Random Forest Regression

```
[38] rf = RandomForestRegressor(n_estimators=100, max_depth=10, random_state=42)
```

```
[39] rf.fit(X_train,Y_train)
```

Model Evaluation

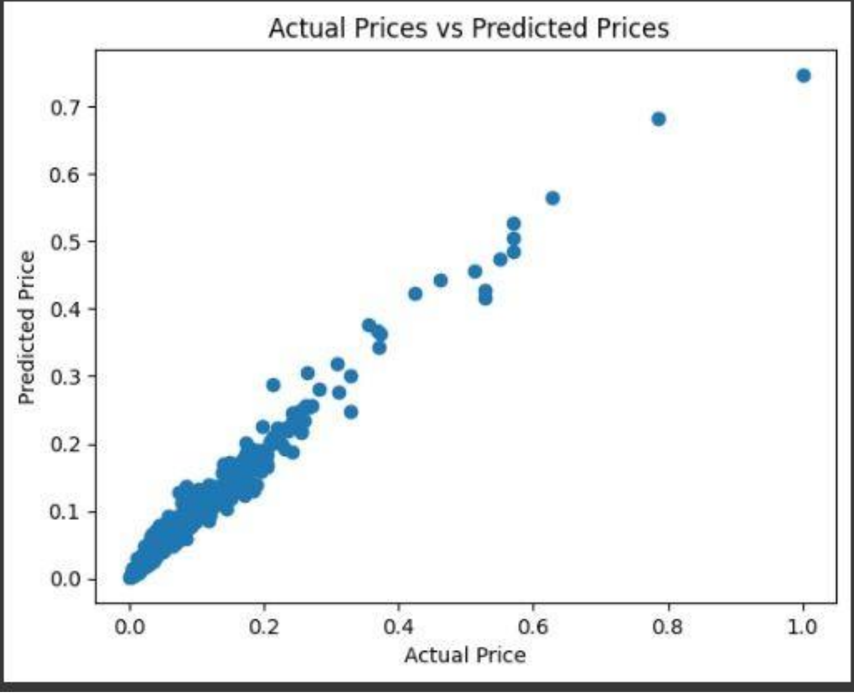
```
[40] # prediction on Training data
     training_data_prediction = rf.predict(X_train)
```

```
[49] # R squared Error
     error_score = metrics.r2_score(Y_train, training_data_prediction)
     print("R squared Error : ", error_score)
```

```
R squared Error :  0.9698186374464624
```

Random Forest Regression (Scatter Plot for Training Data)

```
plt.scatter(Y_train, training_data_prediction)
plt.xlabel("Actual Price")
plt.ylabel("Predicted Price")
plt.title(" Actual Prices vs Predicted Prices")
plt.show()
```



Random Forest Regression
(Testing Data)

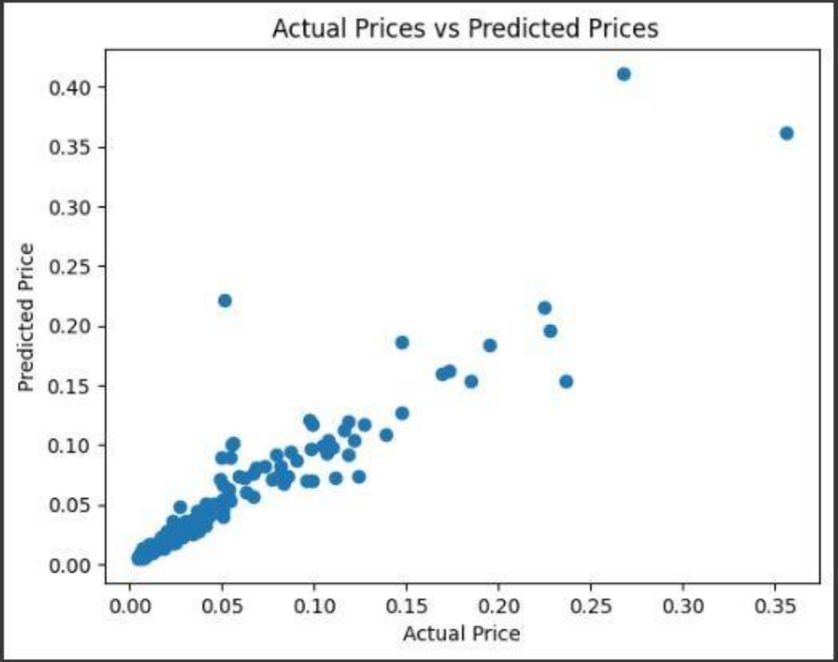
```
[73] # prediction on Test data
     test_data_prediction = lin_reg_model.predict(X_test)
```

```
[94] # R squared Error
     error_score = metrics.r2_score(Y_test, test_data_prediction)
     print("R squared Error : ", error_score)

     rmse = sqrt(metrics.mean_squared_error(Y_test,test_data_prediction))
     print("Root Mean squared Error : ", rmse)
```

```
R squared Error :  0.848614052666839
Root Mean squared Error :  0.02072945033644719
```

Random Forest Regression (Scatter Plot for Testing Data)

```
[183] plt.scatter(Y_test, test_data_prediction)
      plt.xlabel("Actual Price")
      plt.ylabel("Predicted Price")
      plt.title(" Actual Prices vs Predicted Prices")
      plt.show()
```



Random Forest Regression
(K-Fold Cross Validation)

CROSS VALIDATION (Random Forest)

```
rf = RandomForestRegressor(n_estimators=100, max_depth=10, random_state=42)
scores = cross_val_score(rf, X, Y, cv=10)
accuracies = cross_val_score(rf, X, Y,scoring='neg_root_mean_squared_error', cv=10)
print("Cross-validation scores:\n", scores)
print('\n')
print("R2:", scores.mean())
print("RMSE score:", -accuracies.mean())
```

```
Cross-validation scores:
 [0.9388275  0.74806957 0.71070942 0.83947818 0.90922816 0.84192961
 0.79404785 0.94468888 0.89191764 0.79119726]


R2: 0.8410094066439795
RMSE score: 0.026802310926536167
```

04

Result and Comments

Result

We used two evaluation metrics: **R-Squared score** and **RMSE value**.

1. R-squared error measures the proportion of variance in the dependent variable that can be explained by the independent variables in regression model. The R-squared score ranges from 0 to 1. Higher values suggest a better fit between the model and the data.

2. RMSE value is a measure of the average distance between the predicted and actual values of the target variable. Lower rmse value indicates better predictive accuracy, as they indicate smaller average prediction errors.

Result

In application of both the regression algorithms, we got the following results of our evaluation metrics. The following table represents the R-squared score and RMSE value during a 90% split.

Model Name	R-squared score	RMSE score
Linear Regression	0.745906	0.026856
Random Forest Regression	0.848614	0.020729

Table 1: Evaluation metric scores during a 90 percent split

The following table represents the R-squared score and RMSE value during 10-fold cross validation.

Result

Model Name	R-squared score	RMSE score
Linear Regression	0.670001	0.038445
Random Forest Regression	0.841009	0.026802

Table 2: Evaluation metric scores during 10-fold cross-validation

Result
From Research papers

These are the R-squared values we obtained from research papers of car price prediction using linear regression and random forest regression. We can see that the results we achieved in our model are similar to the results they obtained in the research paper.

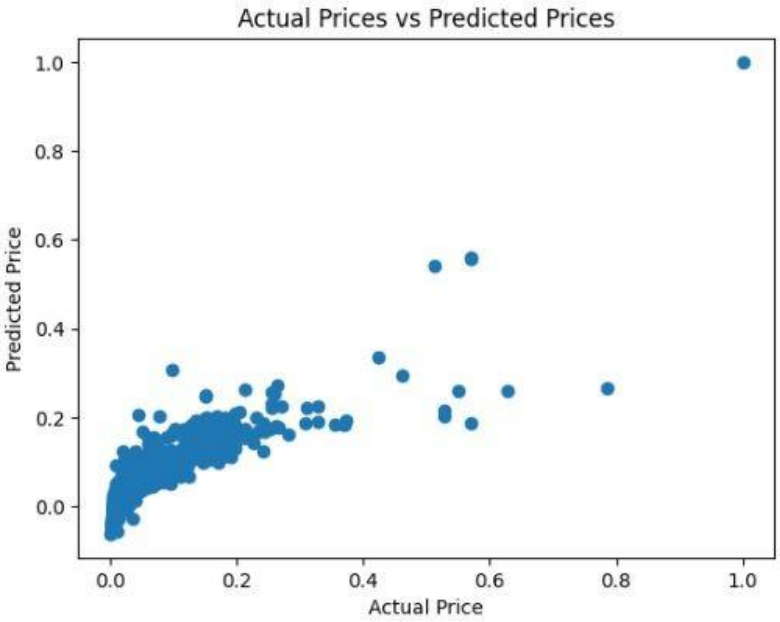
The following table represents the R-squared values of both models from research paper 1.

Model Name	R-squared score
Linear Regression	0.625564
Random Forest Regression	0.911812

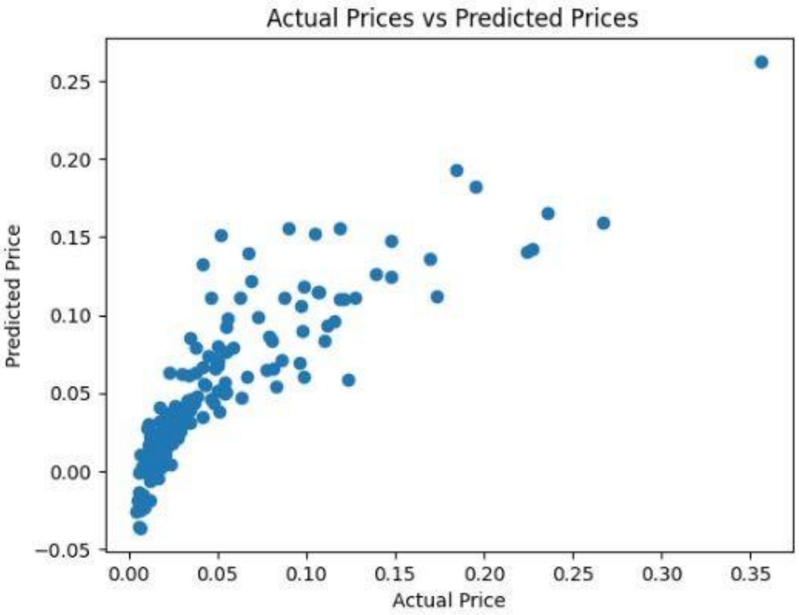
The following table represents the R-squared values of both models from research paper 2.

Model Name	R-squared score
Linear Regression	0.764600
Random Forest Regression	0.931100

Results
Scatter plot for Linear Regression

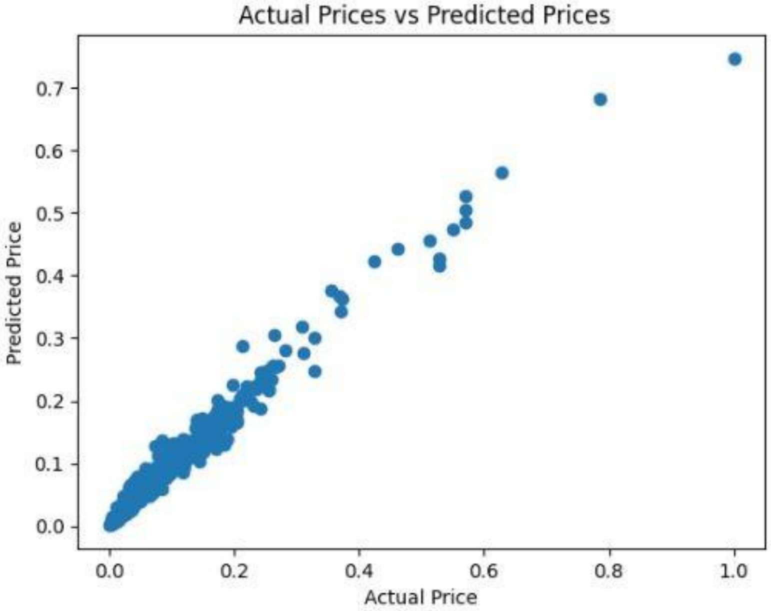


Training Data

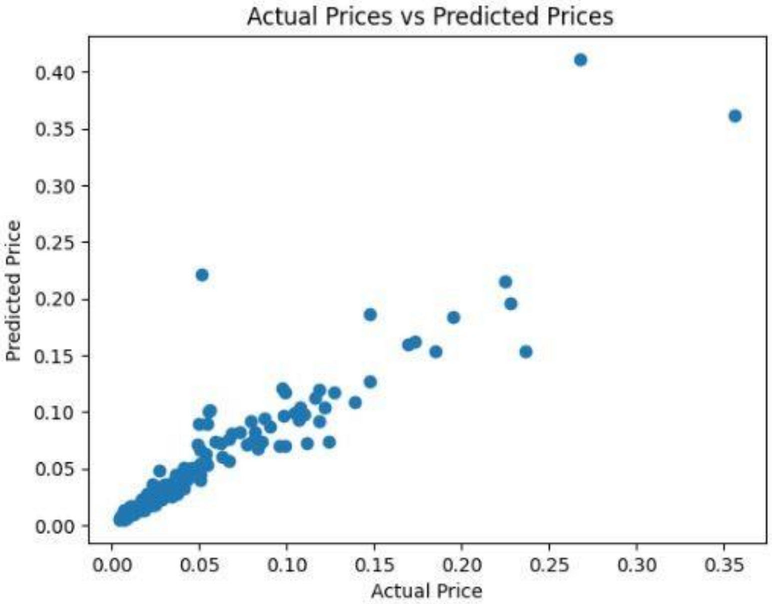


Testing Data

Results
Scatter plot for Random Forest Regression



Training Data



Testing Data

Comments

Based on our evaluation metric we can see that both the models performed with great accuracy in predicting car prices. But if we compare both the linear regression model and the random forest regression model, we can clearly observe that the random forest model performed much better than the linear regression model.

The R-squared score in forest regression model is higher than that of the linear regression model in both types of evaluation. In addition, the RMSE score in the random forest regression model is less than the linear regression model in both types of evaluation. This indicates that the random forest regression model performed with better accuracy than the linear regression model.

We also see that the results we achieved in our model are similar to the results they obtained in the research paper. So we obtained desirable results from our model.

In a car price prediction problem, the goal is to accurately predict car prices with minimal error, consistently providing predictions that closely match the actual prices. It would perform well across different car brands, models, and features, without being overly sensitive or biased. It would generalize effectively to new, unseen car instances and would identify and explain the most significant features and their impact on the pricing of cars.

THANK YOU