Car Price Prediction

Using the scraped data from multiple websites

Submitted by:

P.SaiDeekshitReddy

ACKNOWLEDGMENT

I would like to thank Flip Robo Technologies, for giving me this golden opportunity to work on this valuable project. I got to learn a lot from this project about Data Scraping, Data Wrangling, and practical implementations of using machine learning modules.

I take this opportunity to express my profound gratitude and deep regards to my mentor Ms. Khusboo Garg for her exemplary guidance, monitoring and constant encouragement throughout the course of this assignment. The blessing, help and guidance given by her time to time shall carry me a long way in the journey of life on which I am about to embark.

Lastly, I thank almighty, my parents, brother, sister and friends for their constant encouragement without which this assignment would not be possible.

References:

1. https://www.cardekho.com/
2. https://www.cars24.com/

INTRODUCTION

• Business Problem Framing

The pre-owned vehicle market is a developing business with a market esteem that has almost multiplied itself in earlier years. The ascent of online sites and other instruments like it has made it more straightforward for the two purchasers and merchants to improve comprehension of the variables that decide the market worth of a pre-owned vehicle. In light of a set of variables, Machine Learning calculations might be used to conjecture the cost of any vehicle. The informational collection will remember data for an assortment of vehicles. There will be data with respect to the vehicle's specialized components, for example, the motor kind, fuel type, total driven kilometers, and that's only the tip of the iceberg, for each vehicle.

There is no all-inclusive instrument for building up the retail cost of utilized vehicles in light of the fact that unique sites utilize various techniques to make it. By utilizing measurable models to expect to value, it is conceivable to acquire a fundamental value gauge without entering every one of the subtleties into the ideal site. The fundamental motivation behind this study is to think about the precision of two distinct expectation models for assessing a pre-owned vehicle's retail cost. Subsequently, we offer a Machine Learning-based philosophy at anticipating the costs of secondhand vehicles in light of their attributes.

This philosophy can help purchasers hoping to buy a pre-owned vehicle in making more informed decisions. Clients can now search for all vehicles in a district without actual endeavors, whenever and from any area.

With the Coronavirus sway on the lookout, we have seen lot of changes in the vehicle market. Presently some vehicles are sought after subsequently making them exorbitant and some are not popular consequently less expensive. With the adjustment of market due to Coronavirus 19 effect, people/sellers are facing issues with their past Car Price valuation AI/Machine Learning models. Along these lines, they are

searching for new AI models from new information. Here we are building the new car price valuation model.

The primary point of this venture is to create a dataset with the help of web scraping and anticipate the cost of trade-in vehicle in view of different elements.

• Conceptual Background of the Domain Problem

The costs of new vehicles in the business is fixed by the producer for certain extra expenses brought about by the Government as assessments. Along these lines, clients purchasing another vehicle can be guaranteed of the money/investment they contribute to be commendable. Be that as it may, because of the expanded cost of new vehicles and the ineptitude of clients to purchase new vehicles because of the absence of assets, utilized vehicles deals are on a worldwide increment. There is a requirement at a pre-owned vehicle cost expectation framework to successfully decide the value of the vehicle utilizing an assortment of highlights. Despite the fact that there are sites that offers this assistance, their expectation technique may not be awesome. Additionally, various models and frameworks might contribute on anticipating power for a pre-owned vehicle's genuine market esteem. It is essential to realize their genuine market esteem while both trading.

To have the option to anticipate utilized vehicles market worth can help the two purchasers and merchants. Utilized Vehicle merchants are one of the greatest objective gathering that can be keen on consequences of this review. On the off chance that pre-owned vehicle merchants better get what makes a vehicle attractive, what the significant highlights are for a pre-owned vehicle, then, at that point, they might think about this information and proposition a superior assistance.

• Review of Literature

With the recent arrival of internet portals, buyers and sellers may obtain an appropriate status of the factors that ascertain the market price of a used automobile. Lasso Regression, Multiple Regression, and Regression

Trees are examples of machine learning algorithms. We will try to develop a statistical model that can forecast the value of a pre-owned automobile based on prior customer details and different parameters of the vehicle. This project aims to compare the efficiency of different models' predictions to find the appropriate one. On the subject of used automobile price prediction, several previous studies have been conducted.

We did a background survey regarding the basic ideas of our project and used those ideas for the collection of information like the technological stack, algorithms, and shortcomings of our project which led us to build a better project.

Car Dekho

CarDekho is one of India's most popular automobile websites for new and secondhand car research. It offers precise automobile prices on the road, as well as authentic user and expert evaluations. It may also use the automobile comparison tool to compare different cars. This service also allows you to connect with local vehicle dealers to find the greatest deals.

Cars24

Cars24 is a website where used car sellers may list their vehicles for sale. It's an Indian start-up with a simple user interface that asks sellers for information like automobile model, mileage, registration year, and vehicle type (petrol, diesel). These enable the online model to estimate the price by running particular algorithms on provided parameters.

Motivation for the Problem Undertaken

Objective of the Project:

1. Data Collection : To scrape the data of at least 5000 used cars from various websites like olx, cardekho, cars24, autoportal, cartrade, etc.
2. Model Building : To build a supervised machine learning model for forecasting value of a vehicle based on multiple attributes.

Motivation Behind the Project:

There are a few major worldwide multinational participants in the automobile sector, as well as a number of merchants. By trade, international companies are mostly manufacturers, although the retail industry includes both new and used automobile dealers. The used automobile market has seen a huge increase in value, resulting in a bigger percentage of the entire market. In India, about 3.4 million automobiles are sold each year on the secondhand car market.

Analytical Problem Framing

• Data Sources and their formats

There has been a continuous paradigm of commodity exchanges in existence for a long time. Previously, these transactions were conducted through a barter system, which was ultimately converted into a monetary system. And, as a result of these considerations, any changes in the pattern of re-selling things were also affected. The resale of an object can be accomplished in two ways.

The first is offline, while the second is online. In offline transactions, there is a middleman who is extremely susceptible to corruption and making excessively lucrative deals. The second alternative is to sell it online, where there is a platform that allows the user to find out what price he may earn if he sells it.

We scrape the data for 4500+ cars from websites like cars24, car Dekho . And save it for Machine Learning Model.

We have extracted attributes like Brand, model name alongwith its manufacturing year; Variant; Fuel type; number of owners; location; Date of posting ad online; transmission; driven kilometers and lastly, price. Price is an int type column and is our target variable. Rest all attributes are of object datatype.

• Mathematical/ Analytical Modeling of the Problem

We find out that attributes like brand, model, manufacturing year, variant, total driven kilometers, date of posting ad online and location have very wide range of variables and it's not very smart tostudy their plots. We do not need to perform bivariate analysis on these.

• Data Preprocessing Done

We observe that there are 39 missing values in column ‘variant’ and since it is a column with values in categorical datatype, we will replace the null values with mode.

• Software Requirements and Tools Used

Python was the most popular technology for implementing machine learning ideas, owing to the fact that it has a large number of built-in algorithms in the form of bundled libraries. The following are some of the most important libraries and tools we used in our project:

1. NumPy:

NumPy is a Python module for array processing. It includes a high-performance multidimensional array object as well as utilities for manipulating them. It is the most important Python module for scientific computing. NumPy may be used as a multi-dimensional container of general data in addition to its apparent scientific applications.
NumPy allows any data types to be created, allowing NumPy to connect with a broad range of databases cleanly and quickly.

2. SciPy:

SciPy is a Python library for scientific and technical computing that is free and open-source. Optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers, and other activities used in research and engineering are all covered by SciPy modules.
SciPy is based on the NumPy array object, and it's part of the NumPy stack, which also contains Matplotlib, pandas, and SymPy, as well as a growing number of scientific computing libraries. Other apps with comparable users to NumPy include MATLAB, GNU Octave, and Scilab.
The SciPy stack is occasionally used interchangeably with the NumPy stack. The SciPy library is now available under the BSD license, with

an open community of developers sponsoring and supporting its development.

3. Scikit-Learn: -

Scikit-learn offers a standard Python interface for a variety of supervised and unsupervised learning techniques. It is provided under several Linux distributions and is licensed under a liberal simplified BSD license, promoting academic and commercial use. The library is being constructed.

4. JupyterNotebook:-

Jupyter Notebook is an open-source online software that lets you create and share documents with live code, equations, visualizations, and narrative prose. Data cleansing and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and more are all included.
Jupyter Notebook is an open-source online software that lets you create and share documents with live code, equations, visualizations, and narrative text. Data cleansing and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and more are all included.

Model/s Development and Evaluation

- **Identification of possible problem-solvingapproaches (methods)**

We go over the many techniques and datasets that were used to create this module.

The model will be trained using a dataset comprising over 5000 tuples. The value of a car is determined by factors such as the number of kilometers driven, the year of registration, the kind of gasoline used, and the financial strength of the owner. We created regressor methods and compared the two on different car models because this is a regression problem.

Anaconda seeks to address Python's dependency hell, where distinct projects have various dependency versions, so that project dependencies do not require separate versions, which might conflict.

- Testing of Identified Approaches (Algorithms)

The models used training and testing datasets are as followed:

1. SVR
2. Linear Regression
3. SGD Regressor
4. KNeighbors Regressor
5. Decision Tree Regressor
6. Random Forest Regressor
7. AdaBoostRegressor

- Run and Evaluate selected models

1. SVR:
   SVR is based on the same principles as SVM, with a few small exceptions. It tries to determine the curve given a set of data points. However, because it is a regression technique, rather of utilising the

curve as a decision boundary, the curve is used to identify a match between the vector and the curve's location. Support Vectors aid in establishing the most accurate match between data points and the function used to represent them.

```
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.3, random_state = 45)
svr = SVR()
svr.fit(xtrain,ytrain)
pred_train_svr=svr.predict(xtrain)
pred_test_svr=svr.predict(xtest)
print('SVR Regressor Score:',svr.score(xtrain,ytrain))
print('SVR Regressor r2_score:',r2_score(ytest,pred_test_svr))
print("Mean squared error of SVR Regressor:",mean_squared_error(ytest,pred_test_svr))
print("Root Mean Square error of SVR Regressor:",np.sqrt(mean_squared_error(ytest,pred_test_svr)))
```

```
SVR Regressor Score: -0.06325370730174851
SVR Regressor r2_score: -0.06820672247987791
Mean squared error of SVR Regressor: 297284446193.9865
Root Mean Square error of SVR Regressor: 545237.9720764012
```

The Accuracy of SVR is in negative which stats that this is not the correct model to apply here.

2. Linear Regression

Regression is a method for predicting a dependent component with the help of independent variables.

The method is commonly used to predict and calculate correlations between independent and dependent variables. The regression model establishes a linear or exponential connection between independent and dependent variables.

Linear regression is a type of regression analysis in which the independent(x) and dependent(y) variables can be constrained in a linear relationship. The red line in the graph above is known as the best fit straight line. We want to draw a line that best predicts the data points given the data points we have. The line may be represented using the linear equation below.

y = a0 + a1 * x # Linear Equation

```
lr= LinearRegression()
lr.fit(xtrain,ytrain)
lr.coef_
pred_train=lr.predict(xtrain)
pred_test=lr.predict(xtest)
print('Linear Regression Score:',lr.score(xtrain,ytrain))
print('Linear Regression r2_score:',r2_score(ytest,pred_test))
print("Mean squared error of Linear Regression:",mean_squared_error(ytest,pred_test))
print("Root Mean Square error of Linear Regression:",np.sqrt(mean_squared_error(ytest,pred_test)))
```

```
Linear Regression Score: 0.055020586349048384
Linear Regression r2_score: 0.061364713139148486
Mean squared error of Linear Regression: 261224410556.7481
Root Mean Square error of Linear Regression: 511101.1744818712
```

The accuracy of Linear Regression is only 6%

3. SGD Regressor

   The loss gradient is calculated each sample at a time, and the model is updated along the way using a decreasing strength schedule. SGD stands for Stochastic Gradient Descent (aka learning rate).

   The regularizer is a penalty applied to the loss function that decreases model parameters towards zero using either the squared euclidean norm L2 or the absolute norm L1 or a mix of the two (Elastic Net). The update is trimmed to 0.0 whenever the parameter update passes the 0.0 value due to the regularizer, allowing for the learning of sparse models and online feature selection.

```
sgd=SGDRegressor()
sgd.fit(xtrain,ytrain)
pred_train_sgd=sgd.predict(xtrain)
pred_test_sgd=sgd.predict(xtest)
print('SGD Regressor Score:',sgd.score(xtrain,ytrain))
print('SGD Regressor r2_score:',r2_score(ytest,pred_test_sgd))
print("Mean squared error of SGD Regressor:",mean_squared_error(ytest,pred_test_sgd))
print("Root Mean Square error of SGD Regressor:",np.sqrt(mean_squared_error(ytest,pred_test_sgd)))
```

```
SGD Regressor Score: 0.05307367896974513
SGD Regressor r2_score: 0.06076093246004877
Mean squared error of SGD Regressor: 261392444141.47607
Root Mean Square error of SGD Regressor: 511265.5319317703
```

The accuracy of SGD Regressor is also very poor, it's only 6%

4. KNeighbors Regressor

   Algorithm Calculating the average of the numerical goal of the K nearest neighbours is a straightforward implementation of KNN regression. An inverse distance weighted average of the K closest neighbours is another method. The distance functions used in KNN regression are the same as those used in KNN classification.

```
knr = KNeighborsRegressor()
knr.fit(xtrain,ytrain)
pred_train_knr=knr.predict(xtrain)
pred_test_knr=knr.predict(xtest)
print('K Neighbors Regressor Score:',knr.score(xtrain,ytrain))
print('K Neighbors Regressor r2_score:',r2_score(ytest,pred_test_knr))
print("Mean squared error of K Neighbors Regressor:",mean_squared_error(ytest,pred_test_knr))
print("Root Mean Square error of K Neighbors Regressor:",np.sqrt(mean_squared_error(ytest,pred_test_knr)))
```

```
K Neighbors Regressor Score: 0.748266387098428
K Neighbors Regressor r2_score: 0.6042476227806619
Mean squared error of K Neighbors Regressor: 110138818466.21764
Root Mean Square error of K Neighbors Regressor: 331871.68976310355
```

The accuracy of K Neighbors Regressor is 60% which is okay.

5. Decision Tree Regressor

   To get from observations about an item (represented in the branches) to inferences about the item's goal value, decision tree learning employs a decision tree (as a predictive model) (represented in the leaves). In statistics, data mining, and machine learning, it is one of the predictive modelling methodologies. Classification trees are tree models in which the goal variable can take a discrete set of values; in these tree structures, leaves indicate class labels and branches represent feature combinations that lead to those class labels. Regression trees are decision trees in which the target variable can take continuous values (usually real numbers). The objective is to build a model that predicts the value of a target variable from a set of input variables.

```
dtr=DecisionTreeRegressor(criterion='mse')
dtr.fit(xtrain,ytrain)
pred_train_dtr=dtr.predict(xtrain)
pred_test_dtr=dtr.predict(xtest)
print('Decision Tree Regressor Score:',dtr.score(xtrain,ytrain))
print('Decision Tree Regressor r2_score:',r2_score(ytest,pred_test_dtr))
print("Mean squared error of Decision Tree Regressor:",mean_squared_error(ytest,pred_test_dtr))
print("Root Mean Square error of Decision Tree Regressor:",np.sqrt(mean_squared_error(ytest,pred_test_dtr)))
```

```
Decision Tree Regressor Score: 0.9979397374843745
Decision Tree Regressor r2_score: 0.8024097840748301
Mean squared error of Decision Tree Regressor: 54989822361.62706
Root Mean Square error of Decision Tree Regressor: 234499.08818932978
```

6. Random Forest Regressor

   A regressor with a random forest. A random forest is a meta estimator that employs averaging to increase predicted accuracy and control over-fitting by fitting a number of classification decision trees on various sub-samples of the dataset.

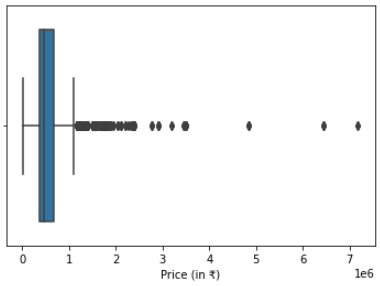
```
rf=RandomForestRegressor()
rf.fit(xtrain,ytrain)
pred_train_rf=rf.predict(xtrain)
pred_test_rf=rf.predict(xtest)
print('Random Forest Regressor Score:',rf.score(xtrain,ytrain))
print('Random Forest Regressor r2_score:',r2_score(ytest,pred_test_rf))
print("Mean squared error of Random Forest Regressor:",mean_squared_error(ytest,pred_test_rf))
print("Root Mean Square error of Random Forest Regressor:",np.sqrt(mean_squared_error(ytest,pred_test_rf)))
```

```
Random Forest Regressor Score: 0.9806729731863234
Random Forest Regressor r2_score: 0.8756911676636284
Mean squared error of Random Forest Regressor: 34595440751.717865
Root Mean Square error of Random Forest Regressor: 185998.49663832734
```

- Visualizations

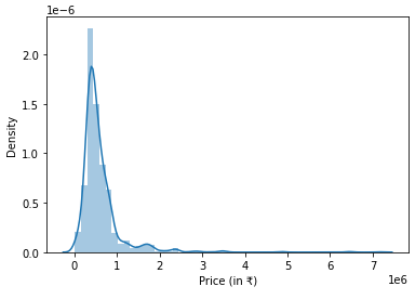
We have plotted histograms and distribution plot in univariate analysis, which interpreted that all the columns are equally important but the columns like brand, variant, location, date and total driven kilometers have a wide range of data spread hence we will not perform it's bivariate analysis.

```
In [13]: sn.boxplot(df['Price (in ₹)'])
Out[13]: <AxesSubplot:xlabel='Price (in ₹)'>
```



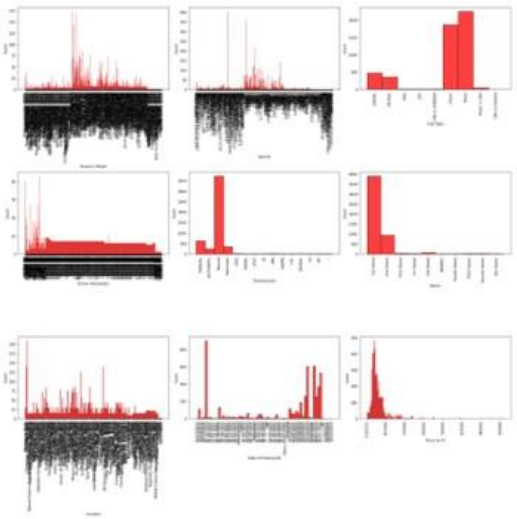
There are many outliers but since it's the target variable, hence we will not treat the outliers.

```
In [14]: sn.distplot(df['Price (in ₹)'])
Out[14]: <AxesSubplot:xlabel='Price (in ₹)', ylabel='Density'>
```



The data is very tightly distributed here and is almost normalized.

HISTOGRAM



~ Brands, Varients, Driven Kilometers & Location have a wide range of values in them.

~ Maximum Cars run on either Petrol or diesel. Only few goes for CNG and other fuels.

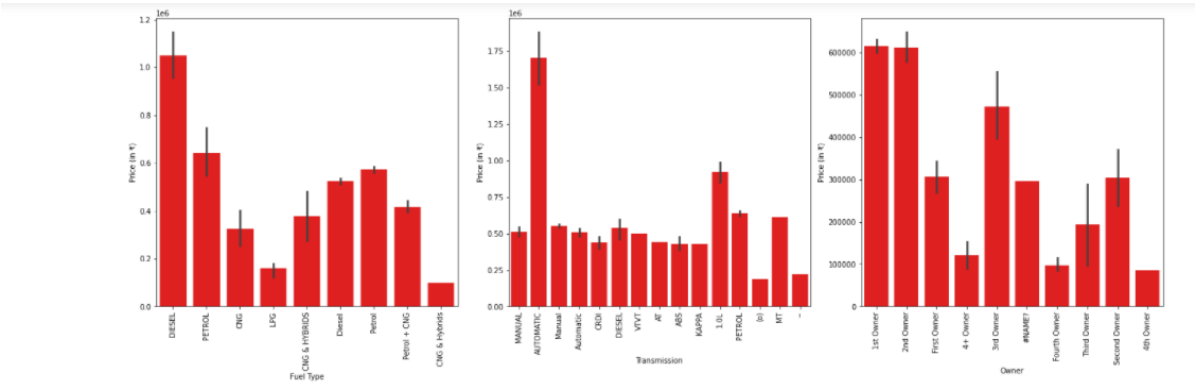
~ Maximum Cars have Manual transnission.

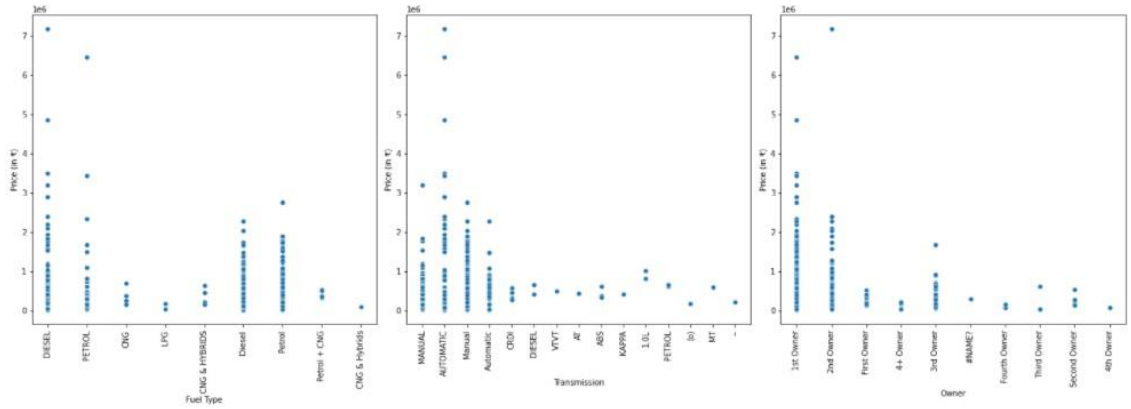
~ Maximum cars are being sold by their very 1st Owner.

~ We have collected the cars posted online in last one month, from 25th December 2021 to 27th January 2022.

~ Almost all the cars have a price rnging in between 270000 to 1165101.

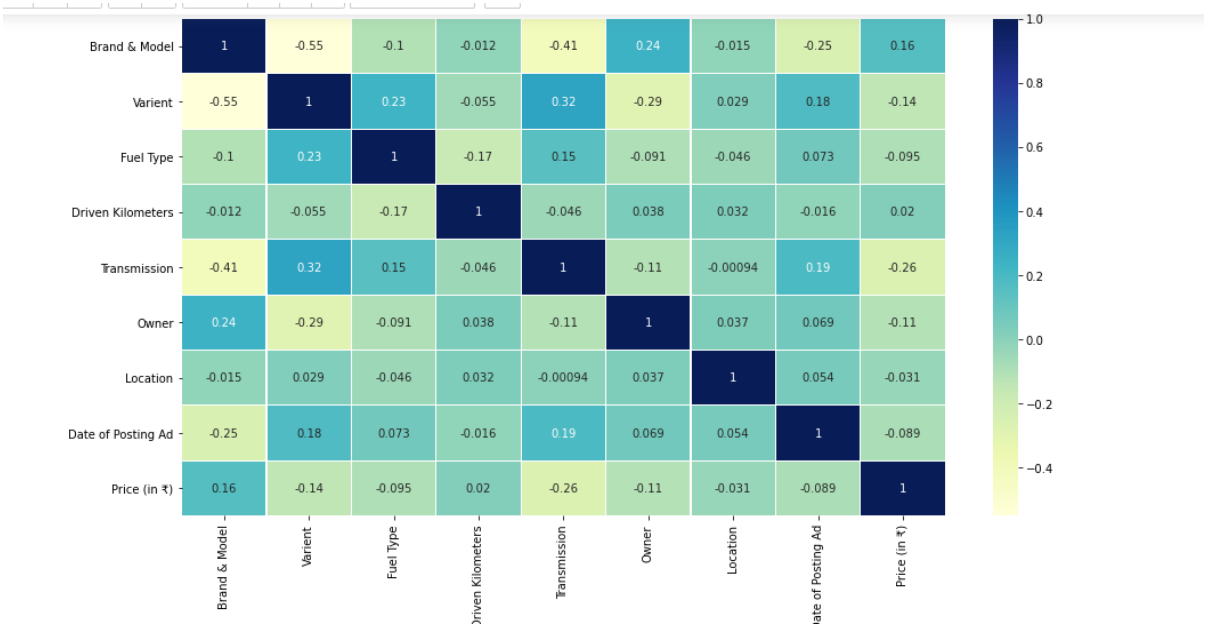
From bivariate analysis we conclude that, Since Brands, Varients, Driven Kilometers & Location have a wide range of values in them, we will not perform bivariate analysis for them as they will not give us any specific details. Now by plotting graph of Fuel type, Transmission and Owner against Price, we conclude that Car that uses Diesel, have automatic Transmission and Has only 1 owner is more likely to have a high price.





Just like bar graph, we can see that Price range is likely to be high for cars using Diesel as fuel, or having Automatic Transmission or is owned by only 1 Owner.

The multivariate analysis done by plotting heatmap says that there is no multicollinearity in the dataset.



CONCLUSION

- Key Findings and Conclusions of the Study

Car price prediction has picked researchers' interest since it takes a significant amount of work and expertise on the part of the field expert. For a dependable and accurate forecast, a large number of unique attributes are analyzed. We employed 6 different machine learning approaches to develop a model for forecasting the price of used automobiles. The respective performances of different algorithms were then compared to discover the one that best suited the existing data set. The final prediction model was implemented in a python programmed. Furthermore, the model was tested with test data, yielding an accuracy of 87.76 percent.

- Learning Outcomes of the Study in respect of Data Science

Using well-known algorithms from Python libraries, we were able to successfully construct machine learning algorithmic paradigms. On our dataset, we first do pre-processing and data cleaning. We trimmed the tuples that contained null values, which accounted for less than 1% of the total. The findings revealed a positive relationship between price and kilometers travelled, as well as year of registration and kilometers travelled.

Negative correlation is related to the notion of inverse proportion, whereas positive correlation is related to the concept of direct proportion. The model was trained using over 5000 tuples.

- Limitations of this work and Scope for Future Work

As a part of future work, we aim at the variable choices over the algorithms that were used in the project. We could only explore two algorithms whereas many other algorithms which exist and might be more accurate. More specifications will be added in a system or providing more accuracy in terms of price in the system i.e.

1) Horsepower

2) Battery power

3) Suspension

4) Cylinder

5) Torque

As we know technologies are improving day by day and there is also advancement in car technology also, so our next upgrade will include hybrid cars, electric cars, and Driverless cars.