FLIP ROBO

CAR PRICE PREDICTION PROJECT

Submitted by:

ASHIN DILEEP T P

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my Mentors for their efforts to make me understand the concepts of Data Science, because of that I could able to complete this project.

I also thank the following websites, https://towardsdatascience.com/, https://www.kaggle.com/, https://scikit-learn.org/, CarTrade.com, Cars24.com, Droom.com and Wikipedia for their information on their webpages that helped me in completing this project.

INTRODUCTION

- Business Problem

  With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model.

  This business problem is a ‘Regression problem’, which is similar to House price prediction problem’, predicting the Co2 emissions of vehicles etc.

- Conceptual Background of the Domain Problem

  Currently, the majority of motor vehicles worldwide are powered by gasoline, petrol or diesel. Other energy sources include ethanol, biodiesel, propane, compressed natural gas (CNG), electric batteries, and hydrogen (either using fuel cells or combustion). There are also cars that use a hybrid of different power sources.

  In our dataset majority of motor vehicles are powered by petrol and diesel. Along with it we have vehicles powered by LPG, CNG and hybrid fuel which is nothing but combination of Petrol with CNG or LPG.

- Review of Literature

Natural language processing (NLP) is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language. NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understanding.

Machine learning is a form of artificial intelligence which compose available computers with the efficiency to be trained without being veraciously programmed. Machine learning algorithms are broadly classified into three divisions, namely; Supervised learning, Unsupervised learning and Reinforcement learning. This project is of Supervised learning. Supervised learning is a learning in which we teach or train the machine using data which is well labelled that means some data is already tagged with correct answer. After that, machine is provided with new set of examples so that supervised learning algorithm analyses the training data and produces a correct outcome.

The automobile market is one of the most competitive in terms of pricing and same tends to vary significantly based on numerous factors; forecasting car price is an important module in decision making for both the buyers and investors in supporting budget allocation, finding the best car etc. Hence it becomes one of the prime fields to apply the concepts of machine learning to optimize and predict the prices with high accuracy. Therefore, we present various important features to use while predicting car prices with good accuracy. We can use regression models, using various features to have lower residual sum of squares error. While using features in a regression model some feature engineering is required for better prediction. Sometimes models are expected to be susceptible towards over fitting hence regularization models is used to reduce it.

Along with regression models, we train other models such as SVR, KNeighbors Regressor, Ada Boost Regressor, Gradient Boosting Regressor etc.

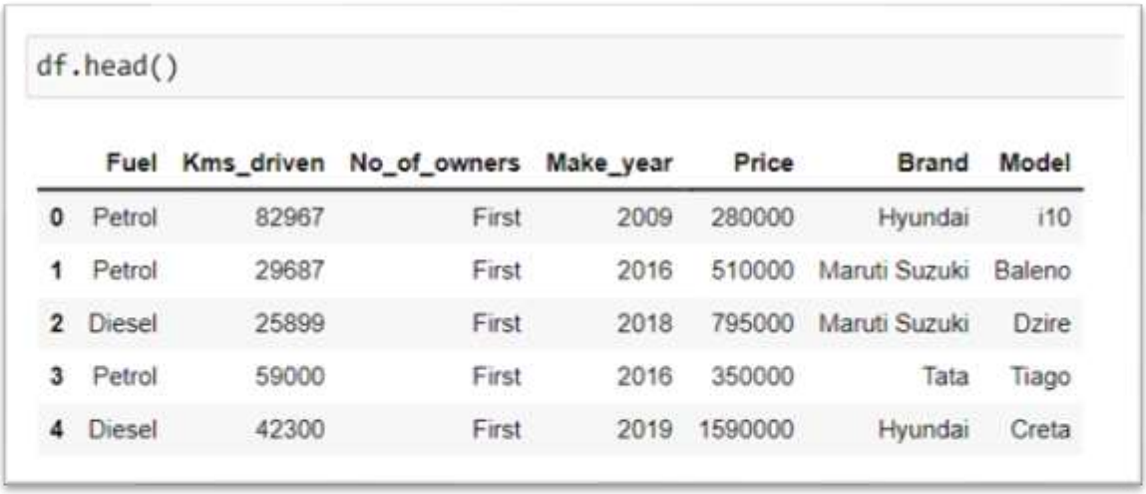
- The Objective for the Problem Undertaken

To build a car price valuation model. And also, to analyse what are the features that are affecting the used cars price in the market.

Analytical Solving Of Problem

- Data Description and it's Source

The data is collected from various websites such as CarTrade, Cars24, Droom. The data is in CSV file. The dataset contains 5307 rows and 7 columns. The dataset contains no missing values and there are no duplicate entries in the dataset.



```
df.head()
```

	Fuel	Kms_driven	No_of_owners	Make_year	Price	Brand	Model
0	Petrol	82967	First	2009	280000	Hyundai	i10
1	Petrol	29687	First	2016	510000	Maruti Suzuki	Baleno
2	Diesel	25899	First	2018	795000	Maruti Suzuki	Dzire
3	Petrol	59000	First	2016	350000	Tata	Tiago
4	Diesel	42300	First	2019	1590000	Hyundai	Creta

The following are the features/columns present in our dataset:

Fuel: Fuel Type

Kms_driven: kilometres Driven

No_of_owners: Total number of Owners

Make_year: Manufacturing Year

Price: Car Price

Brand: Brand Name

Model: Model Name

- Mathematical/ Analytical Modelling of the Problem

In this project we have used ‘shape’ function to check how many rows and columns are there in the dataset.
Then we have used ‘isnull’ function to check whether there are any null values in the dataset.
Then we have used ‘count plot’ to find out what is the count of each category in features.
Then we have used Pandas ‘describe()’ function to find out ‘mean’, ‘standard deviation’, ‘minimum value’, ‘median’, ‘75th percentile’, and ‘maximum value’ of continuous features.
Then we have used ‘scatter plot’ to find out what is the relationship between certain features and the target variable and how some features are affecting the target variable.

- Data Cleaning and Data Pre-processing

Brand and Model names are separated from Brand_Model column. After that Brand and Model features are inserted into dataset. From ‘Kms_driven’ column, ‘,’ and ‘km’ are removed and then it is converted to ‘int’ datatype. Month data is removed from ‘(Make_year) Manufacture Year’ column. In Fuel type column, Petrol + CNG and Petrol + LPG is considered as Hybrid. In No_of_owners column ‘Fifth owner’ is replaced as ‘4 or More’. Then Price column is converted into ‘int’ datatype. Model names are converted into lower case along with-it punctuation (if any) were removed from it.
Then converted categorical values into discrete values using Label Encoder. After that Brand column is converted into dummies format. Then transformed model_names into features using TfidfVectorizer model. Thereafter all these data are concatenated into single dataframe.

- Hardware and Software Requirements and Tools Used

  Some of the essential hardware requirements are as follows Windows OS, CPU with i3/i5 processor and 4GB/8Gb RAM.
  Some of the essential software that we need to do this project is 'Jupyter notebook' which is present on 'Anaconda framework'.
  Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.
  Some of the packages and libraries that are required for this project are:
  Pandas: Pandas are used to read our file from a csv file, to load the read file into a pandas dataframe and manipulate it for further use.
  Numpy: Numpy is used to convert our data into a format suitable to feed our models.
  Matplotlib and Seaborn: Matplotlib and Seaborn are libraries which will be helpful for data visualizations.
  Sklearn: Pre-processing module, feature_selection module, various Algorithms/Models and its Metrics are imported from Scikit-learn (sklearn) library.
  Joblib: Joblib library is used to save the best model for production and for future prediction.

Model/s Development and Evaluation

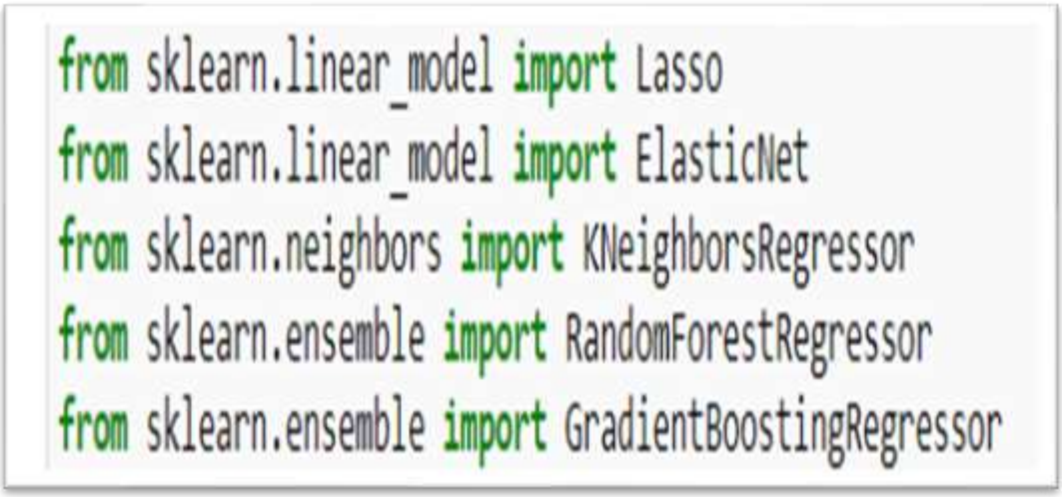
- Methodology of problem solving

After data cleaning the categorical values present in the columns such as ‘Fuel’ and ‘No_of_owners’ are converted into discrete values using Label Encoder. After that Brand column is converted into dummies format. Then transformed model_names into features using TfidfVectorizer model. Thereafter all these data are concatenated into single dataframe.

Then the features are stored in ‘x’ variable and label column (Price) is stored in ‘y’ variable. After that skewness present in the dataset is removed using ‘power transform’ function. We can import ‘power transform’ function from ‘sklearn.preprocessing’. After removing skewness let’s standardize our data using ‘StandardScaler’ function. We can import ‘StandardScaler’ function from ‘sklearn.preprocessing’. After that the x and y is split into train set and test set, which is then used to fit the regressor model.

- Selecting Algorithms

Since the problem is to build a model using Machine Learning in order to predict the price of the used cars i.e., regression problem. Hence let’s import algorithms/models such as Lasso, ElasticNet, K Neighbors Regressor, Random Forest Regressor, Gradient Boosting Regressor from sklearn library.



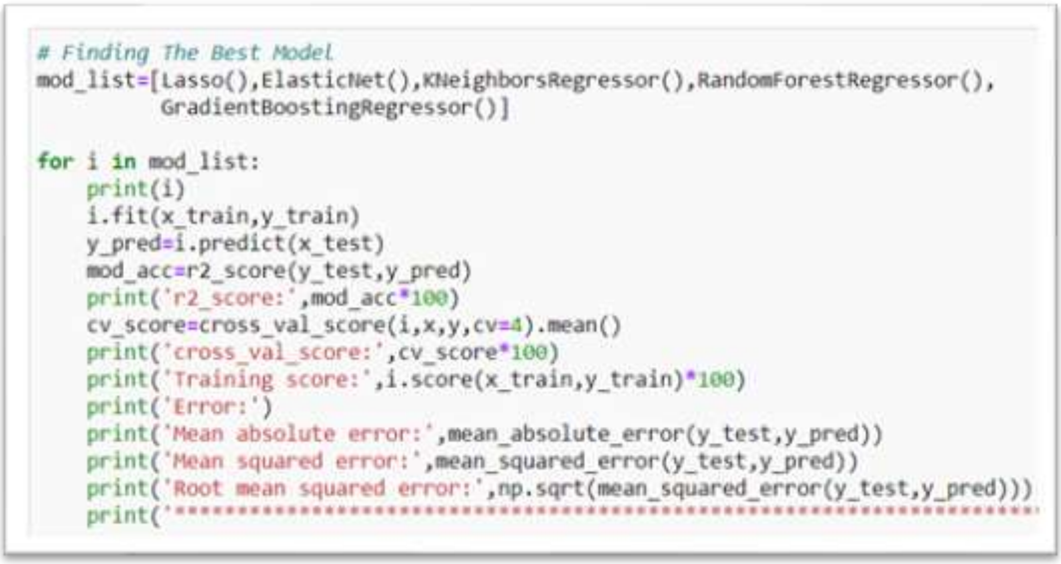
```
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
```

Before using machine learning algorithms/models we should always split our data into a training set and testing set.
Let’s split our data into train set and test set using ‘train_test_split’ function. We can import train_test_split function from sklearn.model_selection. Let the test size be 25% and remaining 75% be train set.

- Run and Evaluating selected Algorithms/Models

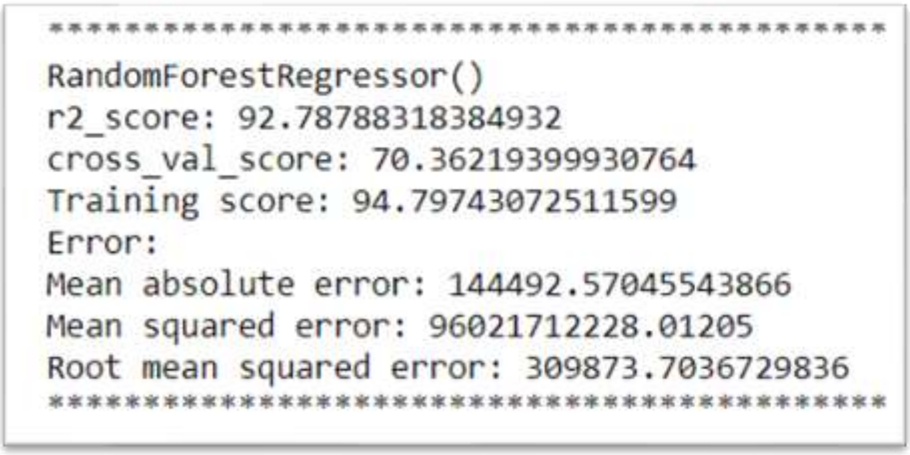
We can now train our model. We use the fit function to train the model.

Let’s run the code shown below and see which model gives us high ‘r2 score’ because higher the ‘r2 score’ better is the model performance.



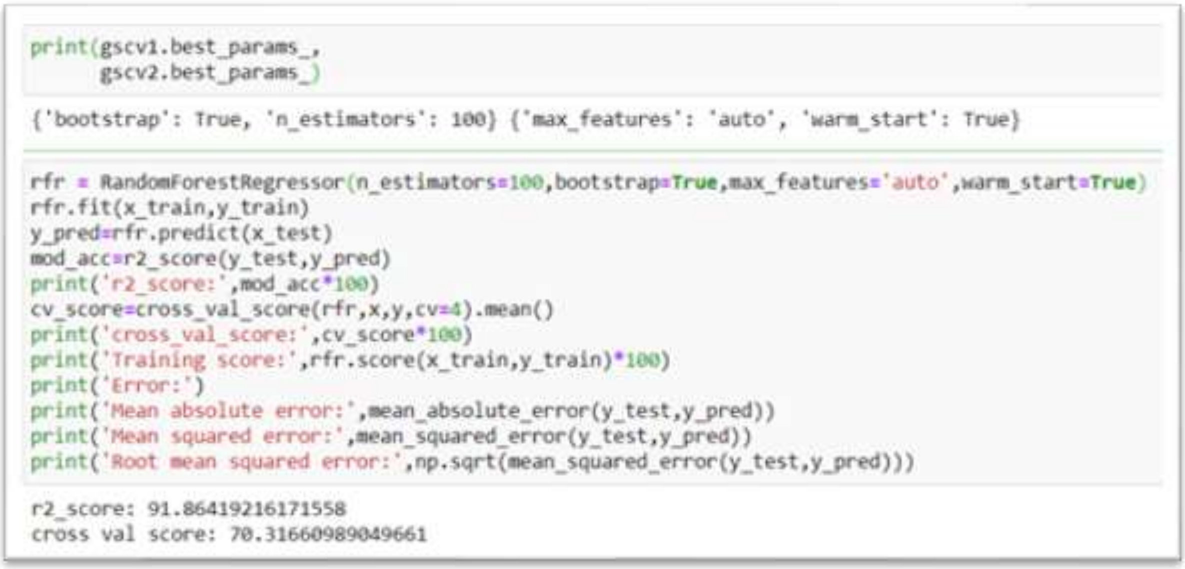
```
# Finding The Best Model
mod_list=[Lasso(),ElasticNet(),KNeighborsRegressor(),RandomForestRegressor(),
          GradientBoostingRegressor()]

for i in mod_list:
    print(i)
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    mod_acc=r2_score(y_test,y_pred)
    print('r2_score:',mod_acc*100)
    cv_score=cross_val_score(i,x,y,cv=4).mean()
    print('cross_val_score:',cv_score*100)
    print('Training score:',i.score(x_train,y_train)*100)
    print('Error:')
    print('Mean absolute error:',mean_absolute_error(y_test,y_pred))
    print('Mean squared error:',mean_squared_error(y_test,y_pred))
    print('Root mean squared error:',np.sqrt(mean_squared_error(y_test,y_pred)))
    print('**********************************************************************
```

After running the code shown above, we came to know that the r2 score of models such as Lasso, ElasticNet, K Neighbors Regressor, Random Forest Regressor, Gradient Boosting Regressor are 86, 81, 77, 92, 84 respectively. Hence, we can conclude that 'Random Forest Regressor' model is performing better with a r2 score of 92.78.



```
*******************************************
RandomForestRegressor()
r2_score: 92.78788318384932
cross_val_score: 70.36219399930764
Training score: 94.79743072511599
Error:
Mean absolute error: 144492.57045543866
Mean squared error: 96021712228.01205
Root mean squared error: 309873.7036729836
*******************************************
```

After that let’s consider Random Forest Regressor model for hyper-parameter tuning to find out which parameters of Random Forest Regressor model can be used so that model’s performance can be enhanced.
Finally, after performing hyper-parameter on Random Forest Regressor model we got an r2 score of 91.86.



```
print(gscv1.best_params_,
      gscv2.best_params_)
```

```
{'bootstrap': True, 'n_estimators': 100} {'max_features': 'auto', 'warm_start': True}
```

```
rfr = RandomForestRegressor(n_estimators=100,bootstrap=True,max_features='auto',warm_start=True)
rfr.fit(x_train,y_train)
y_pred=rfr.predict(x_test)
mod_acc=r2_score(y_test,y_pred)
print('r2_score:',mod_acc*100)
cv_score=cross_val_score(rfr,x,y,cv=4).mean()
print('cross_val_score:',cv_score*100)
print('Training score:',rfr.score(x_train,y_train)*100)
print('Error:')
print('Mean absolute error:',mean_absolute_error(y_test,y_pred))
print('Mean squared error:',mean_squared_error(y_test,y_pred))
print('Root mean squared error:',np.sqrt(mean_squared_error(y_test,y_pred)))
```

```
r2_score: 91.86419216171558
cross val score: 70.31660989049661
```

- Key Metrics for Evaluation of Algorithms/Models

There are various metrics available to evaluate selected Algorithms. Some of the metrics used in this project are as follows:

R2 score: R2 score is one of the best metrics to evaluate Algorithms, higher the ‘r2 score’ better is the model/algorithm performance. R2 score means the proportion of the variance in the dependent variable that is predictable from the independent variable(s). The r2 score varies between 0 and 100%. So, if it is 100%, the two variables are perfectly correlated, i.e., with no variance at all. A low value would show a low level of correlation, meaning a regression model that is not valid, but not in all cases.

‘Cross val score’: Cross-validation is a statistical method used to estimate the skill of machine learning models. This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k − 1 folds. Cross-validation score (Cross val score) is a metrics similar to r2 score where higher the ‘cross val score’ better is the model/algorithm performance.

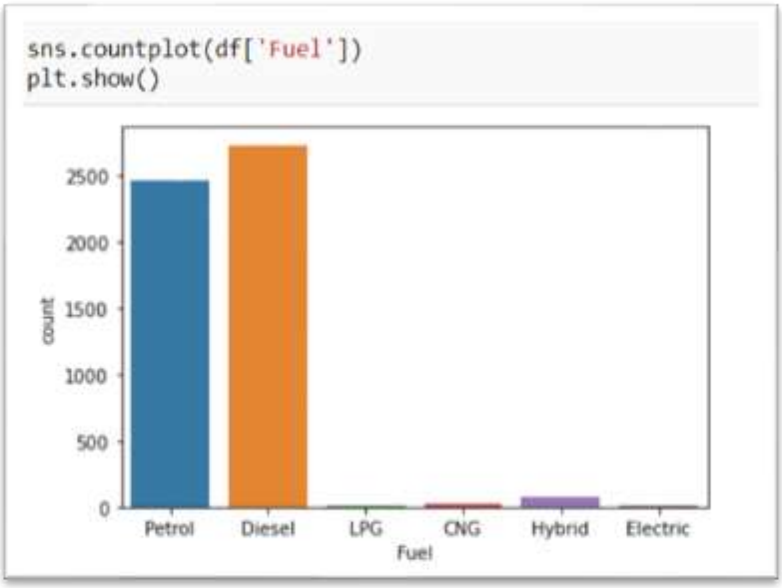
Mean absolute error: Absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation. MAE takes the average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group. Mean absolute error is one of the best metrics to evaluate Algorithms, lower the ‘mean absolute error’ better is the model/algorithm performance.

Mean squared error: MSE is the average of the squared error that is used as the loss function for least squares regression: It is the sum of the square of the difference between the predicted and actual target variables, divided by the number of data points.

Root mean squared error: RMSE is the square root of MSE. MSE is measured in units that are the square of the target variable, while RMSE is measured in the same units as the target variable. Similar to mean absolute error, lower the ‘root mean squared error’ better is the model/algorithm performance.

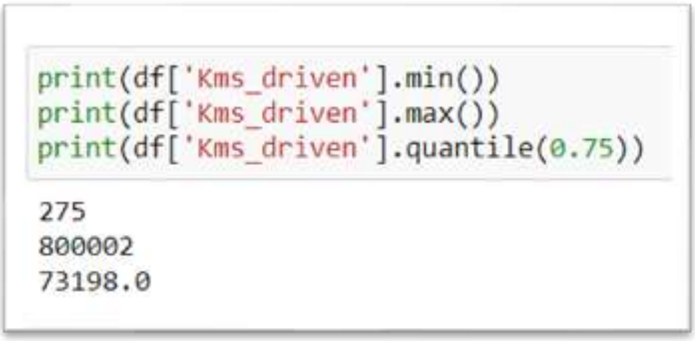
- Visualizations/EDA

Let's take a look into Visualizations /EDA to gain information about independent variable and the relationships between independent variable and price.



```
sns.countplot(df['Fuel'])
plt.show()
```

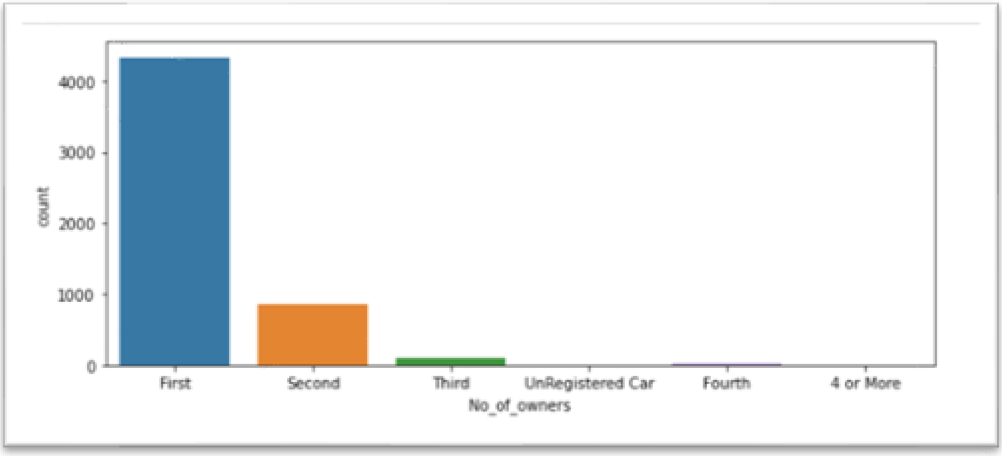
Majority of used cars fuel type is Petrol and Diesel.



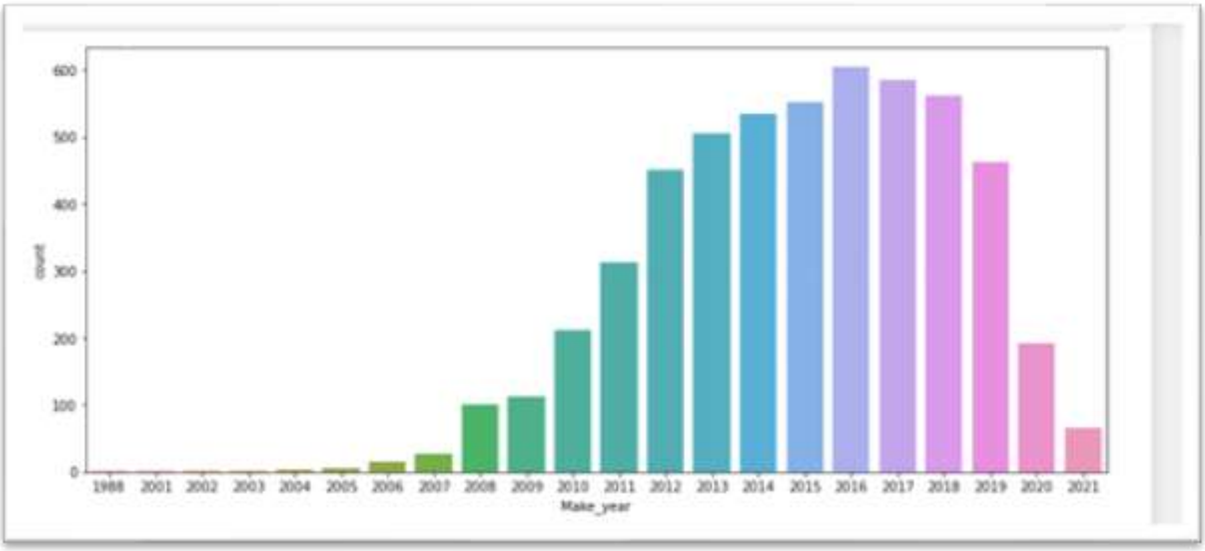
```
print(df['Kms_driven'].min())
print(df['Kms_driven'].max())
print(df['Kms_driven'].quantile(0.75))
```

```
275
800002
73198.0
```

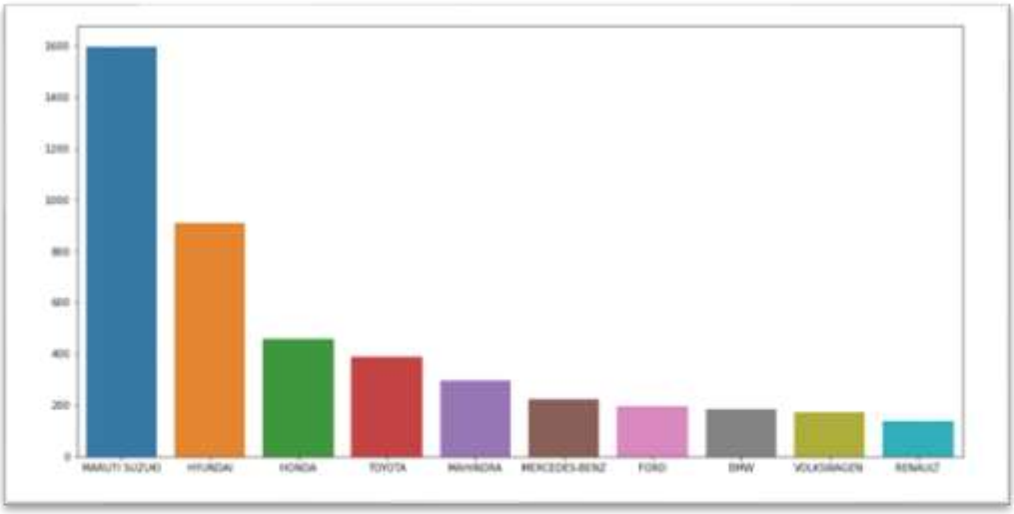
Used cars kilometres driven ranges from 275 km to 8-Lakh km. Majority of the cars have been driven below 70000 km.



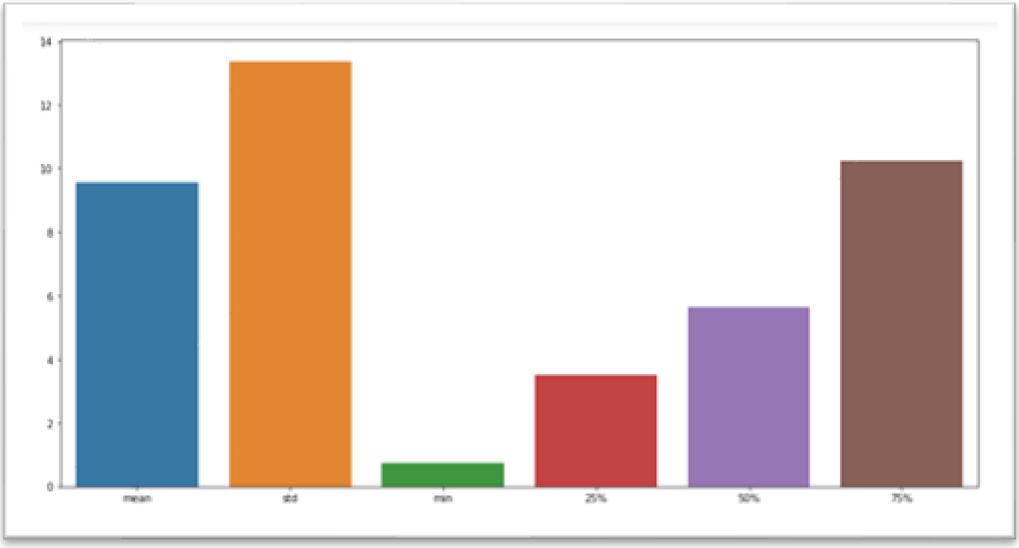
Majority of used cars available here are of 'First' ownership.



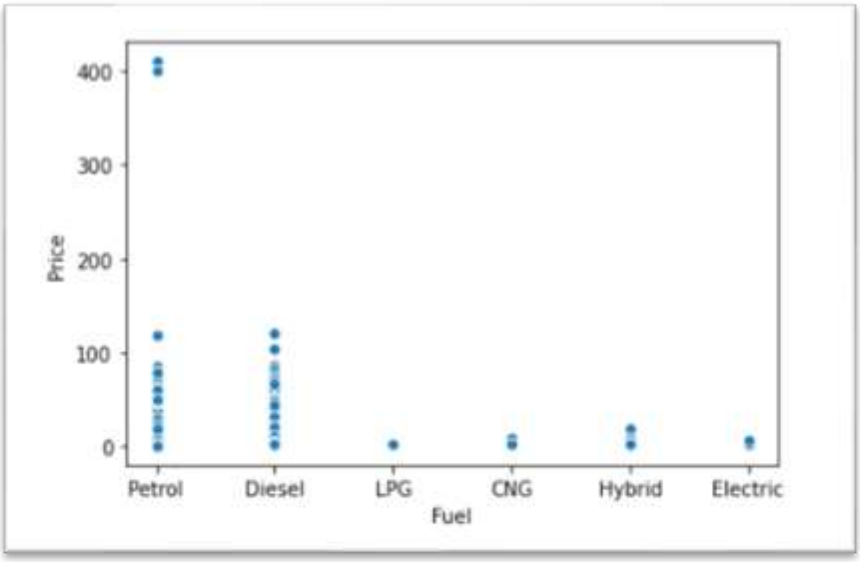
We have cars from 1988 to 2021 for sale. Out of which majority of the cars have been manufactured from 2011 to 2019.



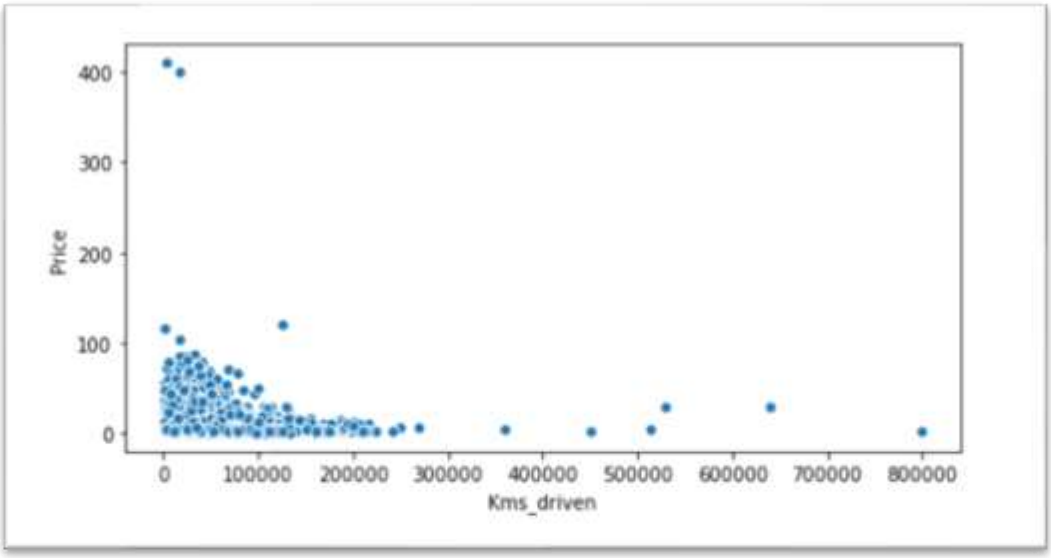
These are the brands, 'Maruti Suzuki', 'Hyundai', 'Honda', 'Toyota', 'Mahindra','Mercedes-Benz', 'Ford', 'BMW', 'Volkswagen', 'Renault', from which majority of car models are for sale.



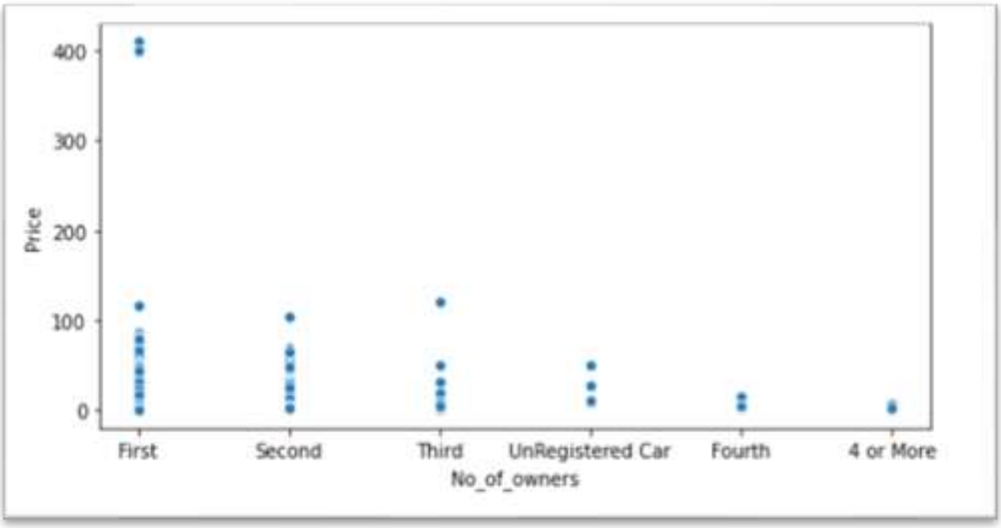
Majority of the car's sale price are below 5.65 Lakhs to 10.24 Lakhs.



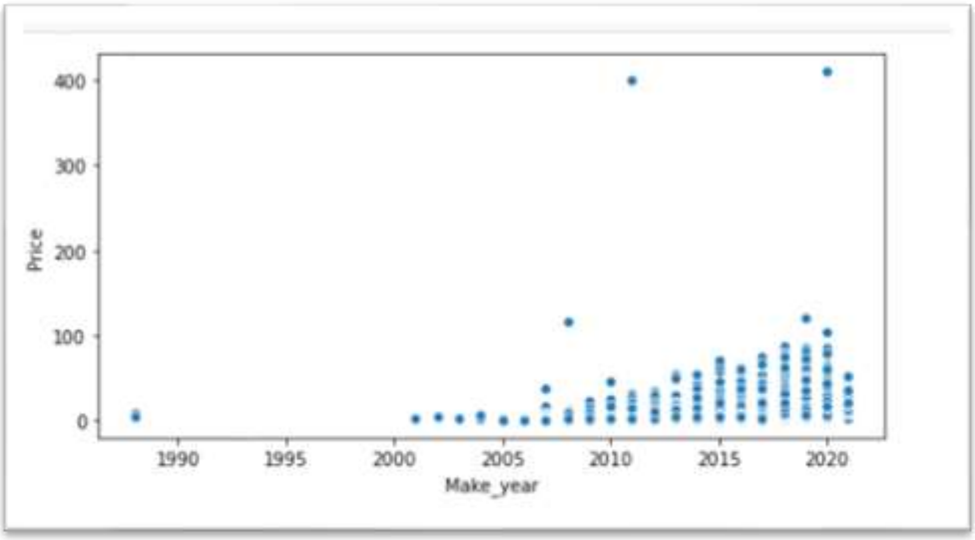
Average price of Diesel cars is high when compared to Petrol cars.



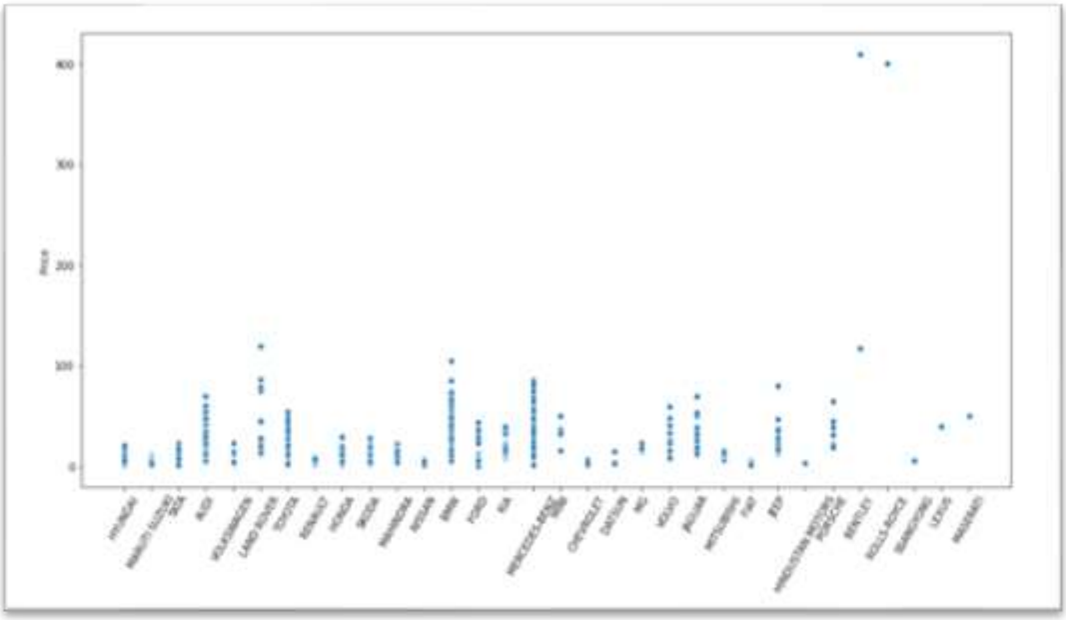
Average price of cars which have been driven for less is high when compared to cars which have been driven for more kilometres.



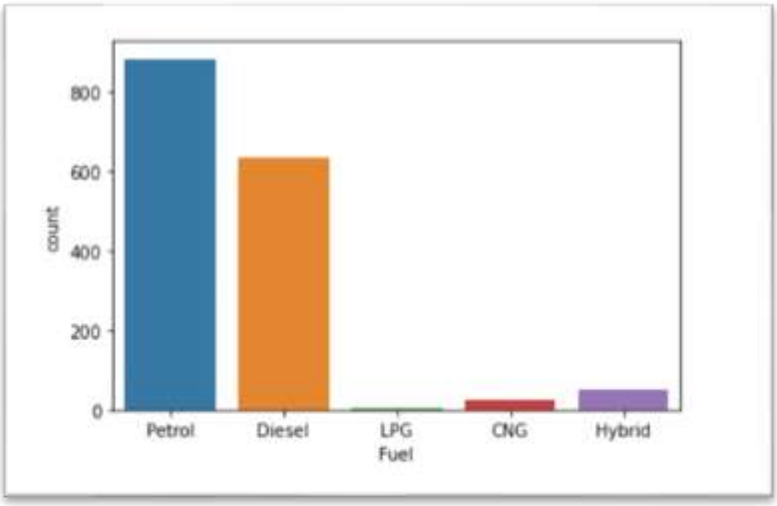
Average price of cars with a smaller number of previous owners is higher.



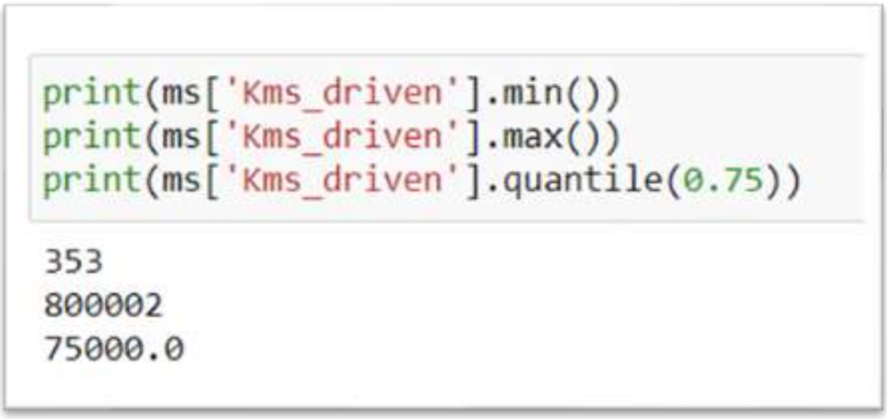
Average price of cars which have been manufactured recently is higher.



Average price of cars from brands such as 'Rolls-Royce', 'Bentley', 'Maserati', 'Land Rover' are higher.
Average price of cars from brands such as 'Chevrolet', 'Fiat', 'Hindustan Motors', 'Datsun' are lower.



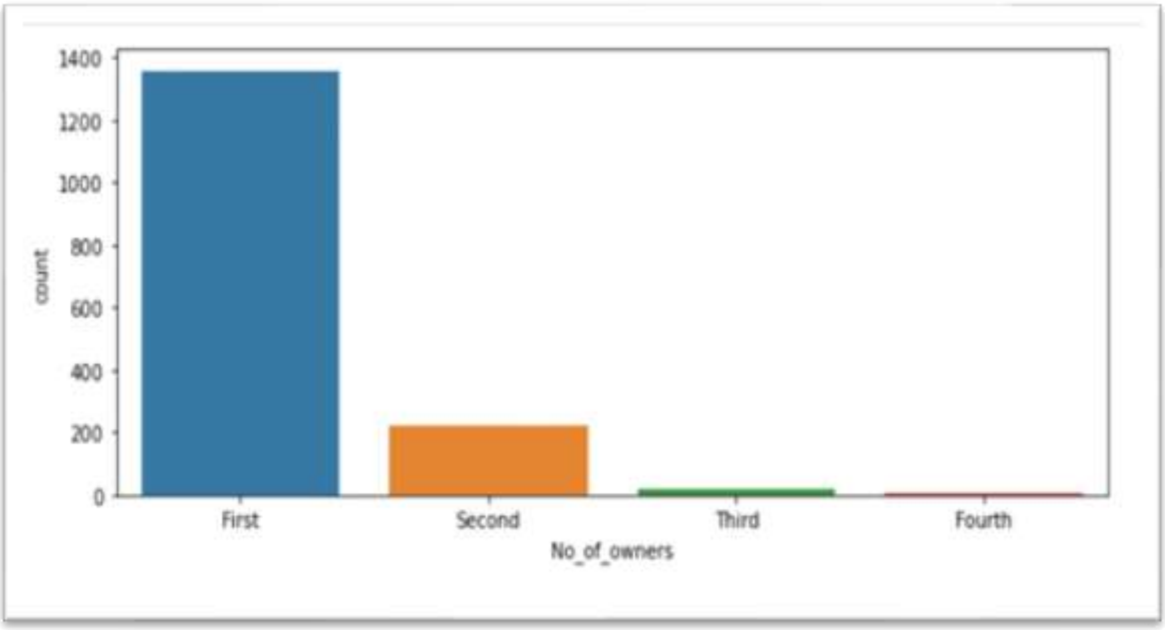
Majority of Maruti Suzuki used car's fuel type is Petrol and Diesel.



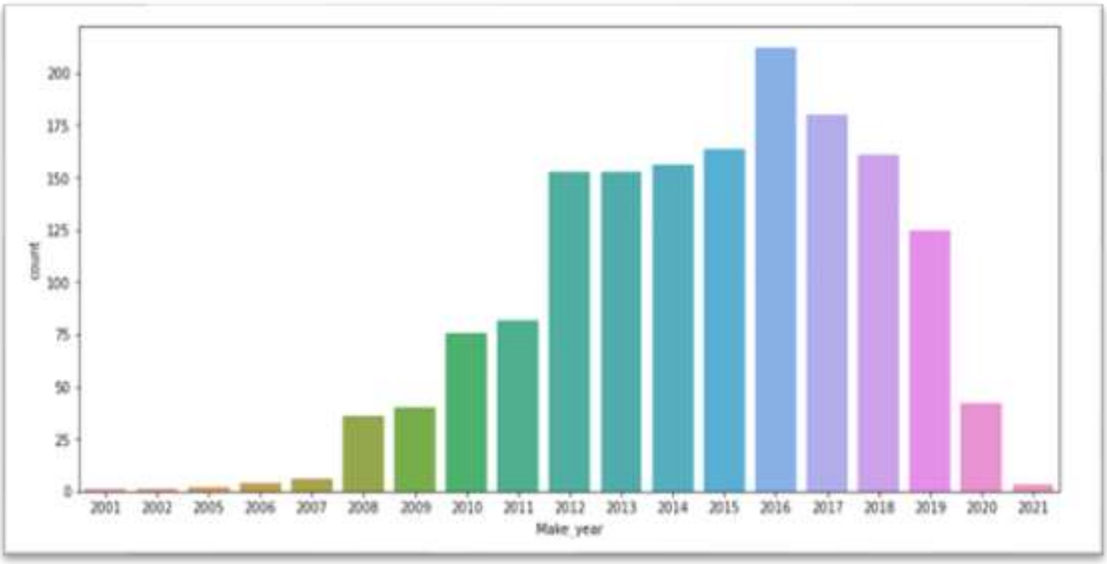
```
print(ms['Kms_driven'].min())
print(ms['Kms_driven'].max())
print(ms['Kms_driven'].quantile(0.75))
```

```
353
800002
75000.0
```

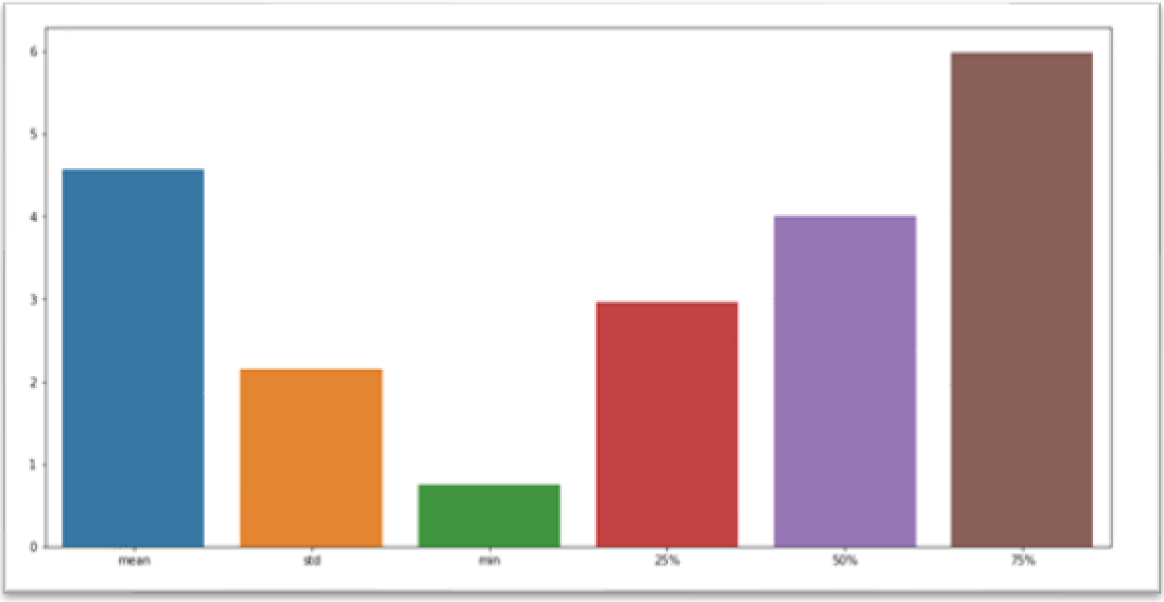
Maruti Suzuki used cars kilometres driven ranges from 350 km to 8-Lakh km. Majority of the cars have been driven below 75000 km.



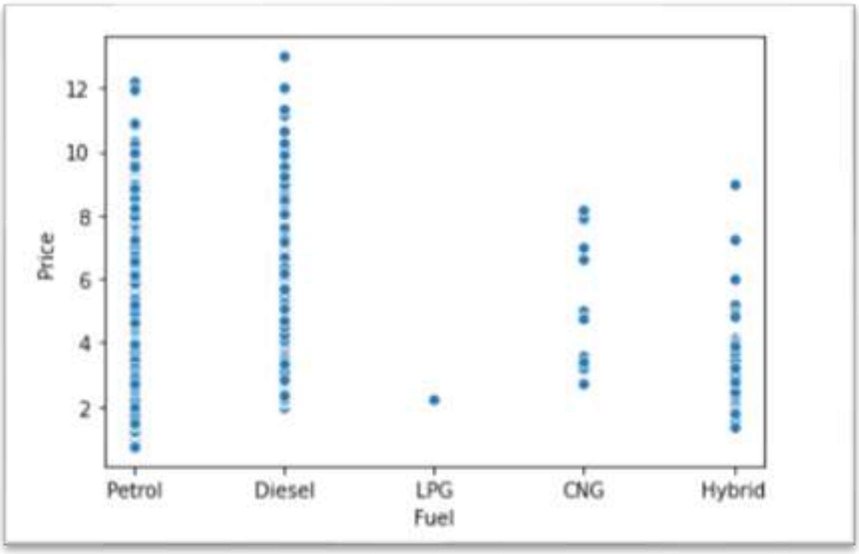
Majority of Maruti Suzuki used cars available here are of 'First' ownership.



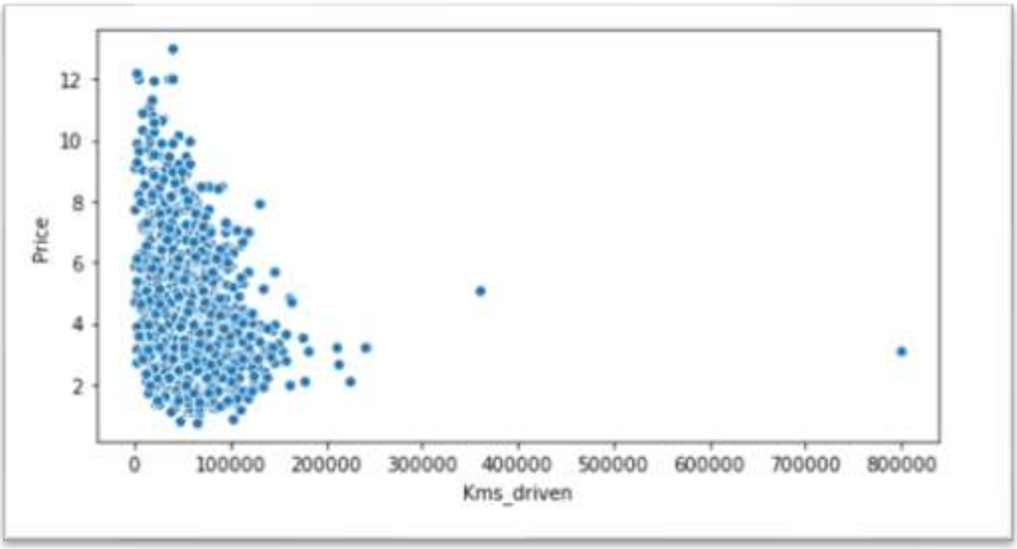
We have Maruti Suzuki used cars from 2001 to 2021 for sale. Out of which majority of the cars have been manufactured from 2012 to 2019.



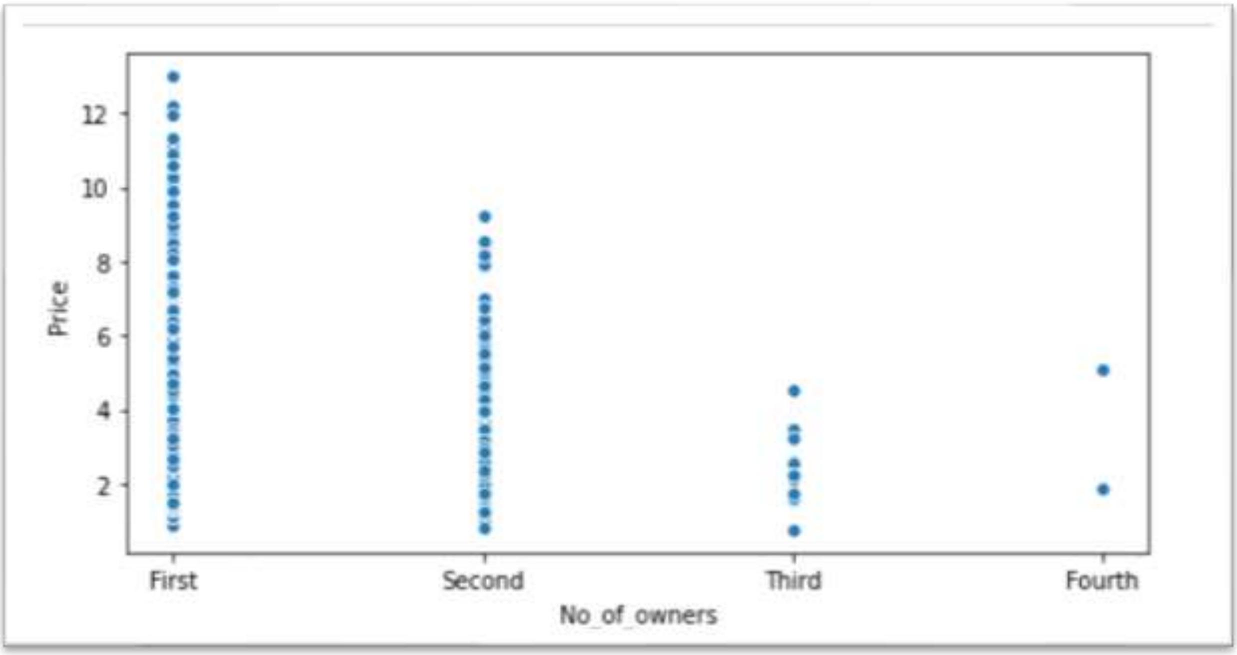
Majority of Maruti Suzuki car's sale price are below 3.99 Lakhs to 5.99 Lakhs.



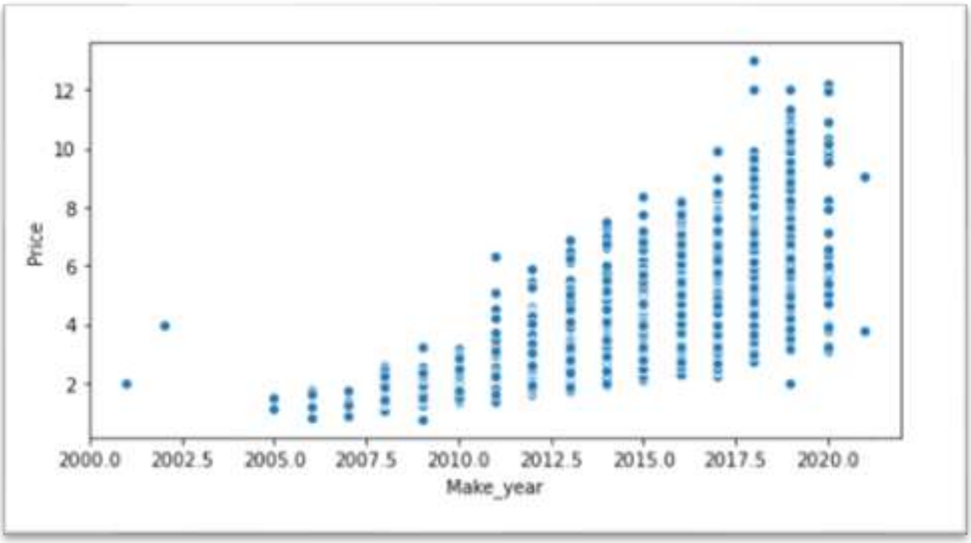
Average price of Maruti Suzuki Diesel cars is high when compared to Petrol cars.



Average price of Maruti Suzuki cars which have been driven for less is high when compared to cars which have been driven for more kilometres.



Average price of Maruti Suzuki used cars with a smaller number of previous owners is higher.



Average price of Maruti Suzuki cars which have been manufactured recently is higher.

- Interpretation of the Results

After doing data pre-processing, we have a dataset which contains 5307 rows and 458 columns.
After training and testing multiple algorithms, we came to know that Random Forest Regressor model is performing better with an r2 score and cross val score of 92.78 & 70.36 respectively. Then after performing hyper-parameter tuning the models final r2 score and cross val score was 91.86 and 70.31 respectively.

Conclusion

- Key Findings and Conclusions of the Study

  - Key Findings

    Majority of the cars have been driven below 70000 km.
    Majority of the car's sale price are below 5.65 Lakhs to 10.24 Lakhs.
    Average price of Diesel cars is high when compared to Petrol cars.
    Average price of cars which have been driven for less is high when compared to cars which have been driven for more kilometres.
    Average price of cars with a smaller number of previous owners is higher.
    Average price of cars which have been manufactured recently is higher.
    Average price of cars from brands such as 'Rolls-Royce', 'Bentley', 'Maserati', 'Land Rover' are higher.
    Average price of cars from brands such as 'Chevrolet', 'Fiat', 'Hindustan Motors', 'Datsun' are lower.

  - Conclusion

    Among selected algorithms/models Random Forest Regres sor model is performing better with an r2 score and cross val score of 91.86 and 70.31 respectively.

- Learning Outcomes of the Study in respect of Data Science

With the help of this project i.e., Car Price Prediction Project we came to know about the strengths/effectiveness of Data Science packages/libraries. With the help of Pandas library, we could able to read our data and manipulate it for further use. With the help of visualizations tools such as Matplotlib and Seaborn we could able to understand the relationship between features/attributes and the label and how some features are affecting the label. With the help of Scikit-Learn library we could able to import algorithms/models and then build algorithms/models which can be used to predict car price.
In this project the model which worked best is Random Forest Regressor. The reason for its performance is, Random Forest is a supervised learning algorithm, the "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. One of the biggest problems in machine learning is overfitting, but most of the time this won’t happen thanks to the random forest regressor. If there are enough trees in the forest, the regressor won’t overfit the model.

Limitations of this work and Scope for Future Work

Limitations of this work

The websites from which the data is scraped has not given certain information about cars such as mileage, engine size, max power, max-torque etc. Addition of these features into dataset would have been great.

Scope for Future Work

We need to scrape some more data because more the data better the model. And we should try to add more features such as mileage, engine size, max power, max-torque because these features will be beneficial to predict car price more accurately. Only four parameters are considered while hyper-parameter tuning the Random Forest Regressor model. In future we can consider other parameters as well and see what is the model performance.