Car Price Prediction Project

Submitted by:

Aaron D'souza

ACKNOWLEDGMENT

Reference Paper :
https://www.temjournal.com/content/81/TEMJournalFebruary2019_113_118.pdf

Reference Paper :
https://www.researchgate.net/publication/319306871_Predicting_the_Price_of_Used_Cars_using_Machine_Learning_Techniques

Link:
https://github.com/krishnaik06/Gaussian-Trnasformaion/blob/master/Untitled1.ipynb

Link:
https://github.com/krishnaik06/Feature-Engineering-Live-sessions/blob/master/Outliers.ipynb

Link:
https://github.com/krishnaik06/Complete-Feature-Selection/blob/master/2-Feature%20Selection-%20Correlation.ipynb

INTRODUCTION

- ## Business Problem Framing

  A car price prediction has been a high interest research area, as it requires noticeable effort and knowledge of the field expert.

  Accurate car price prediction involves expert knowledge, because price usually depends on many distinctive features and factors.

  Typically, most significant ones are brand and model, age, horsepower and mileage. The fuel type used in the car as well as fuel consumption per mile highly affect price of a car due to a frequent changes in the price of a fuel.

  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.

- ## Conceptual Background of the Domain Problem

  Predicting the price of used cars in both an important and interesting problem.

  According to data obtained from the National Transport Authority In many developed countries, it is common to lease a car rather than buying it outright.

  A lease is a binding contract between a buyer and a seller (or a third party –usually a bank, insurance firm or other financial institutions) in which the buyer must pay fixed instalments for a pre-defined number of months/years to the seller/financier.

  Predicting the resale value of a car is not a simple task. It is trite knowledge that the value of used cars depends on a number of factors. The most important ones are usually the age of the car, its make (and model), the origin of the car (the original country of the manufacturer), its mileage (the number of kilometres it has run) and its horsepower.

- ## Review of Literature

  The number of cars registered between 2003 and 2013 has witnessed a spectacular increase of 234%.

  To build a model for predicting the price of used cars in India, we applied eight machine learning techniques (Linear Regression, Random Forest Regression, Bagging Regressor, XGB Regressor, ADA Boost Regressor, Regularization (Lasso), Regularization (Ridge), Gradient Boosting Regressor)

  The data used for the prediction was collected from the web portal Cars24.com using web scraper that was written in python programming language.

  Respective performances of different algorithms were then compared to find one that best suits the available data set.

- ## Motivation for the Problem Undertaken

  In this problem, we investigate the application of supervised machine learning techniques to predict the price of used cars in India. The predictions are based on historical data collected from websites.

  Different techniques like (Linear Regression, Random Forest Regression, Bagging Regressor, XGB Regressor, ADA Boost Regressor, Regularization (Lasso), Regularization (Ridge), Gradient Boosting Regressor) have been used to make the predictions.

  The predictions are then evaluated and compared in order to find those which provide the best performances. All the four methods provided comparable performance. In the future, we intend to use more sophisticated algorithms to make the predictions

Analytical Problem Framing

- ## Mathematical/ Analytical Modelling of the Problem

  As an aspiring Data Scientist, the goal is to create a model that will predict the used car prices with the available independent variables.

  This model will then be used by the management to understand how exactly the prices vary with the variables.

  The company can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns.

  Further, the model will be a good way for the management to understand the pricing dynamics of a new market.

  Based on all the independent variables the model needs to predict dependent variable (Price)

  A total of 8 regression models were used in order to predict the target variable Price.

  - Linear Regression.
  - Random Forest Regression.
  - Bagging Regressor.
  - XGB Regressor.
  - ADA Boost Regressor.
  - Regularization (Lasso).

- Regularization (Ridge)
- Gradient Boosting Regressor.

- ## Data Sources and their formats

  The sample data was scrapped from Cars24.com by using selenium web-driver in python programming language.

  A total of three data types in the data set Int and Object.

  The data collected was saved in .CSV (comma-separated values) format.

  The data set has a total of 10 columns:

  - Name: Name of the car
  - Variant: Variant of the car
  - Fuel: Fuel type (Petrol or Diesel)
  - Kilometers: Kilometers driven per car
  - Purchase date: Original purchase year of the car
  - Owners: Number of previous owners
  - Transmission: Transmission type (Manual or Automatic)
  - Accidental: Whether the car is accidental or not
  - Price: Car Price in Rs
  - Location: Location of car

	Name	Variant	Fuel	Kilometers	Purchase Date	Owners	Transmission	Accidential	Price	Location
0	2016 Mahindra Scorpio S10 MANUAL	S10 MANUAL	Diesel	51,569 km	November 2016	1st Owner	MANUAL	Non-Accidental	₹10,18,199	Gurgaon
1	2012 Maruti Alto 800 LXI MANUAL	LXI MANUAL	Petrol	38,732 km	January 2012	1st Owner	MANUAL	Non-Accidental	₹2,77,999	Kolkata
2	2010 Hyundai i10 MAGNA 1.1 IRDE2 MANUAL	MAGNA 1.1 IRDE2 MANUAL	Petrol	38,567 km	July 2010	1st Owner	MANUAL	Non-Accidental	₹3,00,799	Bangaluru
3	2018 Honda WR-V 1.2 i-VTEC VX MT MANUAL	1.2 i-VTEC VX MT MANUAL	Petrol	17,665 km	July 2018	1st Owner	MANUAL	Non-Accidental	₹7,57,599	Mumbai
4	2015 Hyundai Eon SPORTZ MANUAL	SPORTZ MANUAL	Petrol	15,346 km	July 2015	3rd Owner	MANUAL	Non-Accidental	₹3,33,499	Chennai

Fig.1 Data-frame Sample

- ## Data Preprocessing Done

Step – 1

We can drop the Accidental column as it only has one value and it will not have any significant affect on the target variable.

```
# We can drop the Accidential column as it has only one value
df = df.drop("Accidential",axis=1)
```

Step – 2

We create two new columns Brand and Car_name from Name column.

```
# Extracting Brand name from Name column
df["Brand"] = df["Name"].str.split().str[1]
```

```
# Extracting car name from Name column
df["Car_Name"] = df["Name"].str.split().str[2]
```

Step – 3

We can drop the name column now as we have already created two new columns from it.

```
# We can drop Name as it is not necessary now
df = df.drop("Name",axis=1)
```

Step – 4

Now we need to extract Transmission type(Manual or Automatic) from Variant column to create a new column name Transmission

```
# Creating Transmission column from Variant column
df["Transmission"] = df["Variant"].str.split().str[-1]
```

Step – 5

We need to remove the string km from Kilometres column as we only need the numerical data.

```
# dropping km text from Kilometers column
df["Kilometers"] = df["Kilometers"].str.split().str[0]
```

Step – 6

We can remove the month name and only keep the year data.

```
# dropping Month Name text from Purchase date column
df["Purchase_Date"] = df["Purchase Date"].str.split().str[1]
```

Step – 7

Removing the owner text from owner column as we only need the numerical data (1, 2 or more)

```
# dropping Owner text from Owners column
df["Owners"] = df["Owners"].str.split().str[0]
```

Step – 8

We need to remove the rupee symbol from the Price column.

```
# Removing the price symbol
df["Price"] = df["Price"].str.split("₹").str[1]
```

Step – 9

Checking for null values

```
df.isnull().sum()
```

```
Car_Name         14
Brand            14
Variant           0
Fuel              0
Kilometers        0
Purchase_Date     0
Owners            0
Location          0
Transmission      0
Price            14
dtype: int64
```

We can drop the null values as there are only 14 null values, so it won't have a significant effect on the target variable.

Step – 10

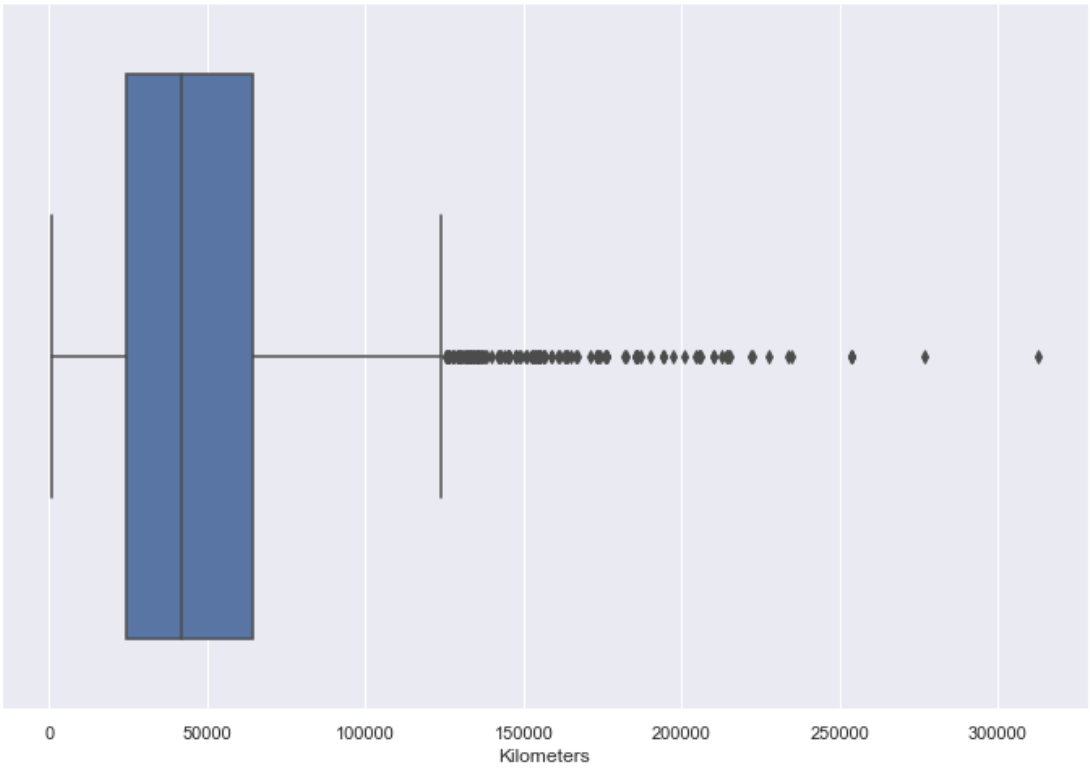
All the data-types are Objects so we need to convert (Kilometres, Purchase_date, Owners and Price to Float)

```
# Converting Object dtype to float dtype
df["Purchase_Date"] = df["Purchase_Date"].astype(float)
```

```
# Converting Object dtype to float dtype
df["Owners"] = df["Owners"].astype(float)
```

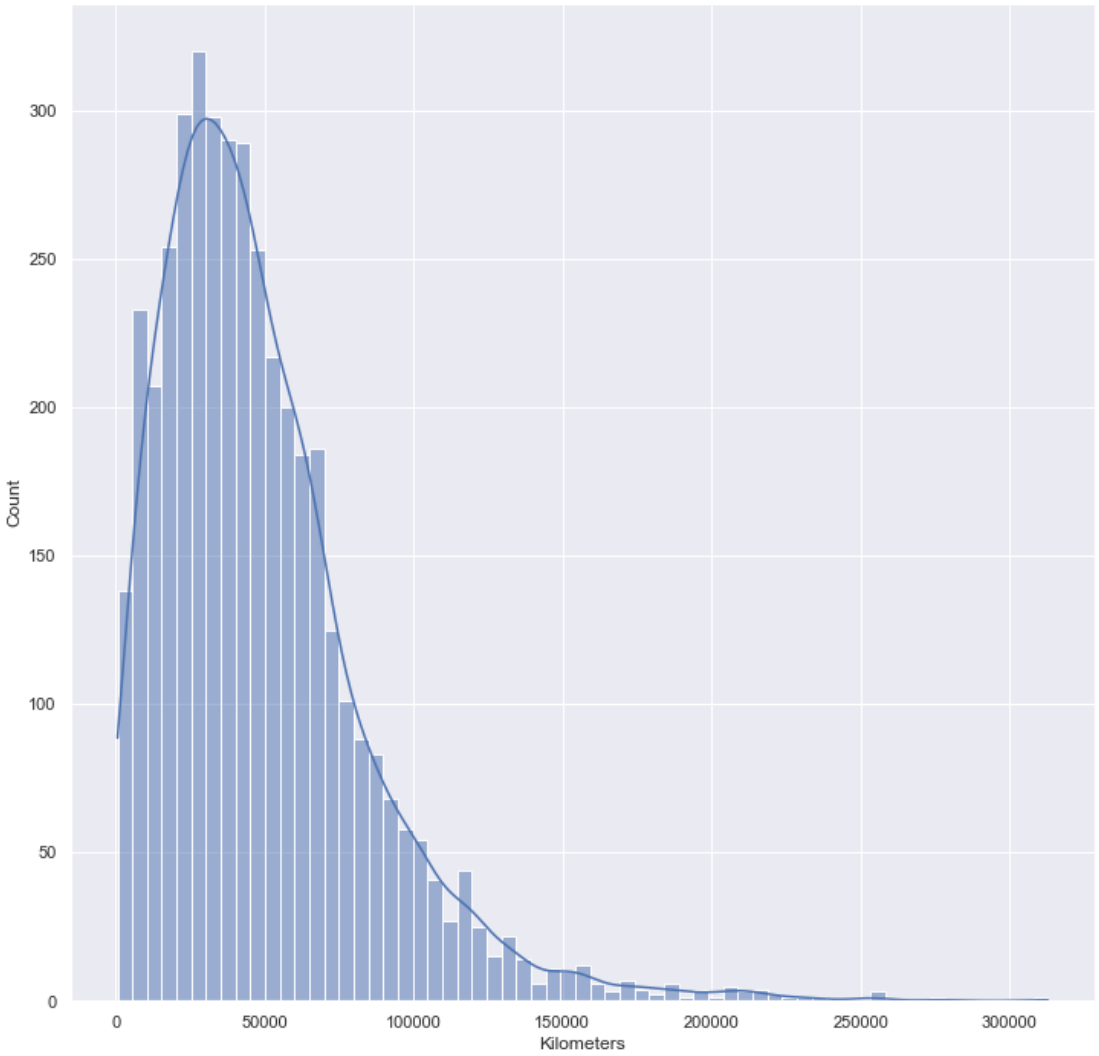
Step – 11

We need to handle outliers and transformations. Fortunately Except Kilometres we don't have to worry about other independent variables as they are categorical in nature. Using box-plot to visually detect outliers.



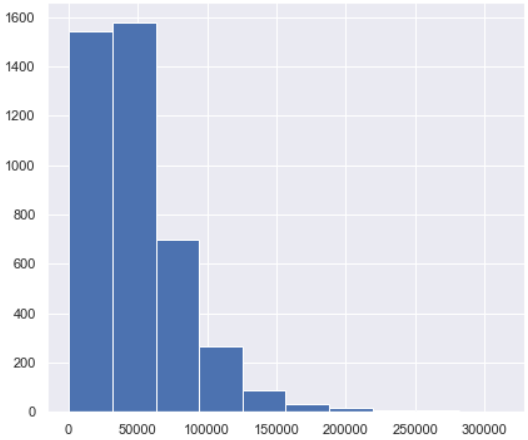
Step – 12

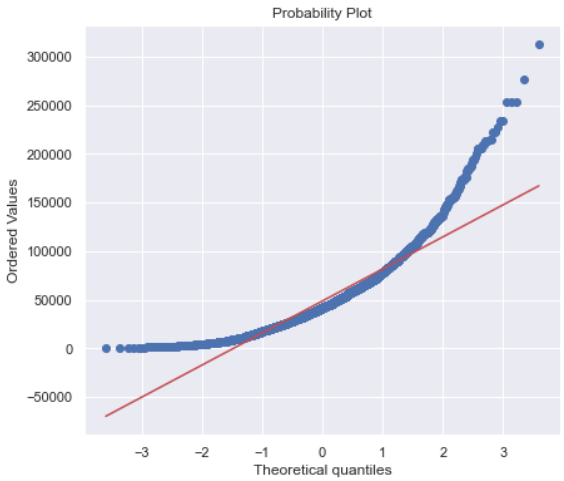
Using hist-plot to understand the distribution of the variable, The distribution for Kilometres seems to be right-skewed



Step – 13

Checking the distribution using Q-Q plot, we can clearly see that the distribution is right-skewed so we need to reduce the skewness and try to make the plot normally distributed.





Step – 14

Checking Outliers and replacing them.

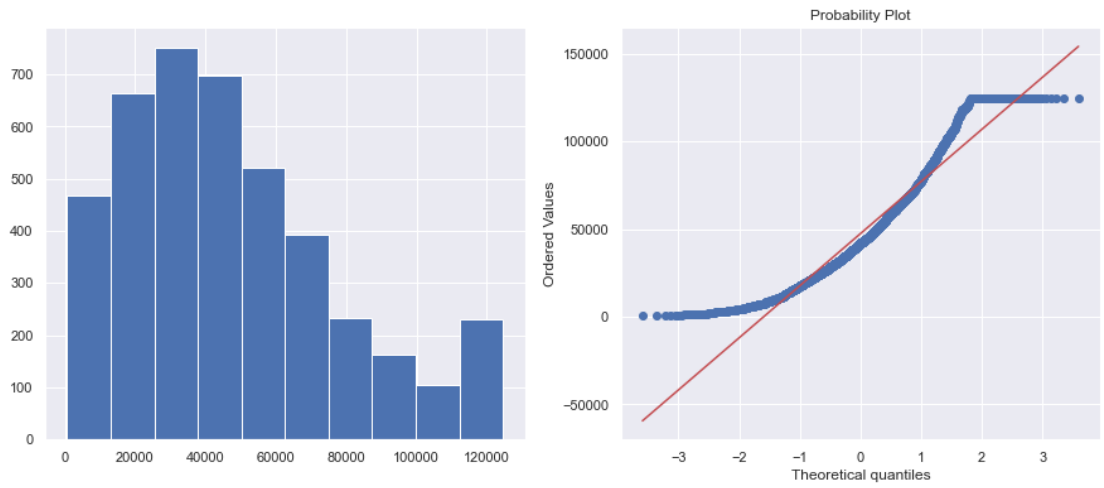
```
1  outSkew("Kilometers")
```

```
(-36053.875, 124617.125)
```

```
1  # Anything above 124617.12 is considered as an outlier
2
3  df.loc[df["Kilometers"] >= 124617.12,"Kilometers"] = 124617.12
```

Step – 15

Checking the Q-Q plot after outlier removal. We can clearly observe that the plots looks more like normally distributed and the skewness has been reduced significantly.



Step – 16

Trying to further reduce the skewness using Square root transformations

```
# Checking Skewness
df["Kilometers"].skew()
```

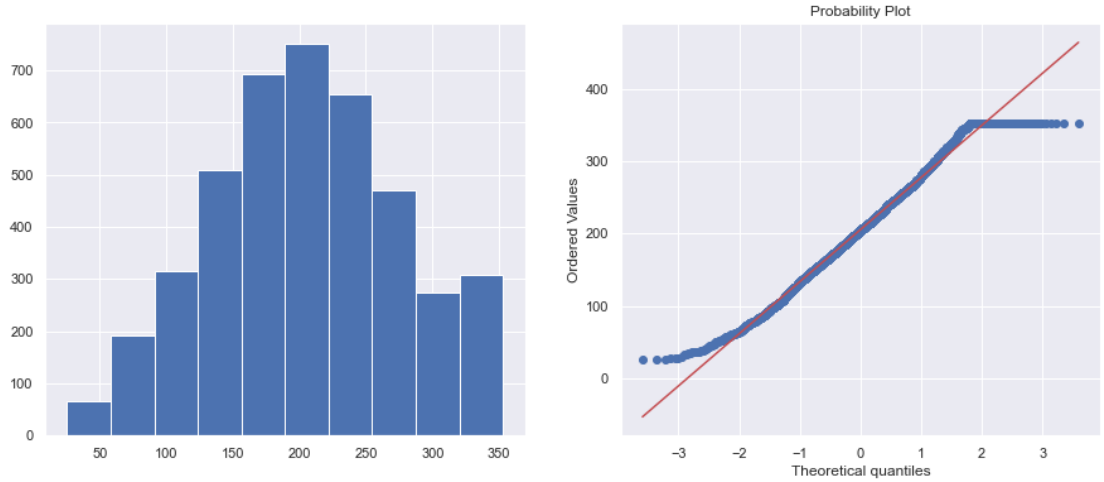
```
0.8101712014304718
```

```
# Applying Square root transformation
sRoot("Kilometers")
```

```
0.0575185935035886
```

Step – 17

After using square root transformation let's check the Q-Q plot.



Step – 18

We need to encode the categorical data.

Using Target encoding on Car_name and Variant columns.

```
# Applying Target encoding on Car_Name and Variant Column

from category_encoders import TargetEncoder
```

```
encoder = TargetEncoder()
```

```
df[["Car_Name","Variant"]] = encoder.fit_transform(df[["Car_Name","Variant"]], df["Price"])
```

Step – 19

Using pandas get_dummies method on rest of the categorical columns.

```
# Using Pandas .get_dummies on rest of the columns

df = pd.get_dummies(data=df,drop_first=True)
```

Step – 20

Using Train Test Split on Data-frame

Train Test Split

```
from sklearn.model_selection import train_test_split
```

```
# Seprating independent and dependent feature from train dataset
x = df.drop("Price",axis=1)
y = df["Price"]
```

Step – 21

Scaling the data using StandardScalar() method

```
# Scaling the data using Standard Scaler
from sklearn.preprocessing import StandardScaler
```

```
sc = StandardScaler()
```

```
# using fit transform of test data
scale_x = sc.fit_transform(x)
```

- ## Data Inputs- Logic- Output Relationships

  Data Correlation:

  When checked the effect of independent variables on the target variable Price, Car Name, Variant and Purchase Year are having a significant effect on the target variable (Price)

```
[ ]: 1 # We can clearly see the features which are highly correlated with the target variable
     2 cor["Price"].sort_values(ascending=False)
```

```
[ ]: Price                1.000000
     Car_Name             0.901571
     Variant              0.897978
     Purchase_Date        0.444294
```

- ## Hardware and Software Requirements and Tools Used

  Hardware used:

  - OS: Windows 10 Home Single Language 64 bit
  - Ram: 8 GB
  - Processor: Intel I5

  Software used:

  - Jupyter Notebook

Model/s Development and Evaluation

- ## Identification of possible problem-solving approaches (methods)

  Given the number of inputs the Machine learning Algorithm has to predict the Price(target) column.

  Target Variable is Price. The target variable is continuous in nature.

  The best approach is to solve this as a regression problem using various regression algorithms and find out which algorithm gives the best results

- ## Testing of Identified Approaches (Algorithms)

  ### Linear Regression:

  Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting.

```
# Taking best random state as 30
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=30)
mod_1 = LinearRegression()
mod_1.fit(X_train,y_train)
train_score_1 = mod_1.score(X_train,y_train)
pred_1 = mod_1.predict(X_test)
test_score_1 = r2_score(y_test,pred_1)

print("The training accuracy is :",train_score_1)
print("The testing accuracy is :",test_score_1)
print("\n")
```

```
The training accuracy is : 0.9041909178791092
The testing accuracy is : 0.9014225442405491
```

Random Forest Regression:

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

```
# Taking the best random state as 7
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=7)
mod_2 = RandomForestRegressor()
mod_2.fit(X_train,y_train)
train_score_2 = mod_2.score(X_train,y_train)
pred_2 = mod_2.predict(X_test)
test_score_2 = r2_score(y_test,pred_2)

print("The training accuracy is :",train_score_2)
print("The testing accuracy is :",test_score_2)
print("\n")
```

```
The training accuracy is : 0.9940768993988104
The testing accuracy is : 0.9778899250462563
```

Bagging Regressor:

A Bagging regressor is an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction.

```
# The best random state is 4
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=4)
mod_6 = BaggingRegressor()
mod_6.fit(X_train,y_train)
train_score_6 = mod_6.score(X_train,y_train)
pred_6 = mod_6.predict(X_test)
test_score_6 = r2_score(y_test,pred_6)

print("The training accuracy is :",train_score_6)
print("The testing accuracy is :",test_score_6)
print("\n")
```

```
The training accuracy is : 0.9915902142046472
The testing accuracy is : 0.9667423130019536
```

XGB Regressor:

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks.

```
# Taking the best random state as 2
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=2)
mod_7 = XGBRegressor()
mod_7.fit(X_train,y_train)
train_score_7 = mod_7.score(X_train,y_train)
pred_7 = mod_7.predict(X_test)
test_score_7 = r2_score(y_test,pred_7)

print("The training accuracy is :",train_score_7)
print("The testing accuracy is :",test_score_7)
print("\n")
```

```
The training accuracy is : 0.9957523755766945
The testing accuracy is : 0.975206187029716
```

ADA Boost Regressor:

Ada Boost algorithm, short for Adaptive Boosting, is a Boosting technique that is used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights to incorrectly classified instances.

```
# Taking the best random state as 2
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=2)
mod_4 = AdaBoostRegressor()
mod_4.fit(X_train,y_train)
train_score_4 = mod_4.score(X_train,y_train)
pred_4 = mod_4.predict(X_test)
test_score_4 = r2_score(y_test,pred_4)

print("The training accuracy is :",train_score_4)
print("The testing accuracy is :",test_score_4)
print("\n")
```

```
The training accuracy is : 0.8618706462585168
The testing accuracy is : 0.8631424975050204
```

Regularization (Lasso):

Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters).

```
# Taking the best random state as 3
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=3)
mod_8 = Lasso()
mod_8.fit(X_train,y_train)
train_score_8 = mod_8.score(X_train,y_train)
pred_8 = mod_8.predict(X_test)
test_score_8 = r2_score(y_test,pred_8)

print("The training accuracy is :",train_score_8)
print("The testing accuracy is :",test_score_8)
print("\n")
```

```
The training accuracy is : 0.9038680911072616
The testing accuracy is : 0.9014942687184432
```

Regularization (Ridge):

Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values.

```
# taking the best random state as 11

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=11)
mod_9 = Ridge()
mod_9.fit(X_train,y_train)
train_score_9 = mod_9.score(X_train,y_train)
pred_9 = mod_9.predict(X_test)
test_score_9 = r2_score(y_test,pred_9)

print("The training accuracy is :",train_score_9)
print("The testing accuracy is :",test_score_9)
print("\n")
```

```
The training accuracy is : 0.9063007485759471
The testing accuracy is : 0.8921586316788612
```

Gradient Boosting Regressor:

Gradient boosting is a machine learning technique for regression, classification and other tasks, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

```
# using the best random state as 14
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=14)
mod_3 = GradientBoostingRegressor()
mod_3.fit(X_train,y_train)
train_score_3 = mod_3.score(X_train,y_train)
pred_3 = mod_3.predict(X_test)
test_score_3 = r2_score(y_test,pred_3)

print("The training accuracy is :",train_score_3)
print("The testing accuracy is :",test_score_3)
print("\n")
```

```
The training accuracy is : 0.9502964296908767
The testing accuracy is : 0.9441175613366892
```

- ## Key Metrics for success in solving problem under consideration

R2 Score:

R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

Mean Square Error:

The mean squared error (MSE) tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them.

Mean Absolute Error:

Mean Absolute Error (MAE) is another loss function used for regression models. MAE is the sum of absolute differences between our target and predicted variables.

Root Mean Square Error:

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are.

Cross Validation(K-fold):

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

	Algorithm	Training_Acc	R2 Score	MSE	MAE	RMSE	Cross_validation
0	Linear Regression	0.904191	0.901423	1.705578e+10	82313.790679	130597.762250	0.895893
1	Random Forest Regression	0.994077	0.977890	4.465489e+09	40208.293432	66824.312897	0.963874
2	Gredient Boosting	0.950296	0.944118	1.196787e+10	64208.577737	253.394115	0.930256
3	ADA Boost	0.861871	0.863142	2.536318e+10	121810.555699	159258.207201	0.847103
4	Bagging Regressor	0.991590	0.966742	6.277652e+09	43227.498188	79231.638224	0.958628
5	XGB Regressor	0.995650	0.980635	3.416841e+09	38677.161939	58453.751142	0.963292
6	Lasso	0.903868	0.901494	1.942884e+10	87363.376719	139387.363009	0.892606
7	Ridge	0.906301	0.892159	2.050765e+10	85487.924861	143204.906819	0.892621

Fig.2 Result Table

- Visualizations

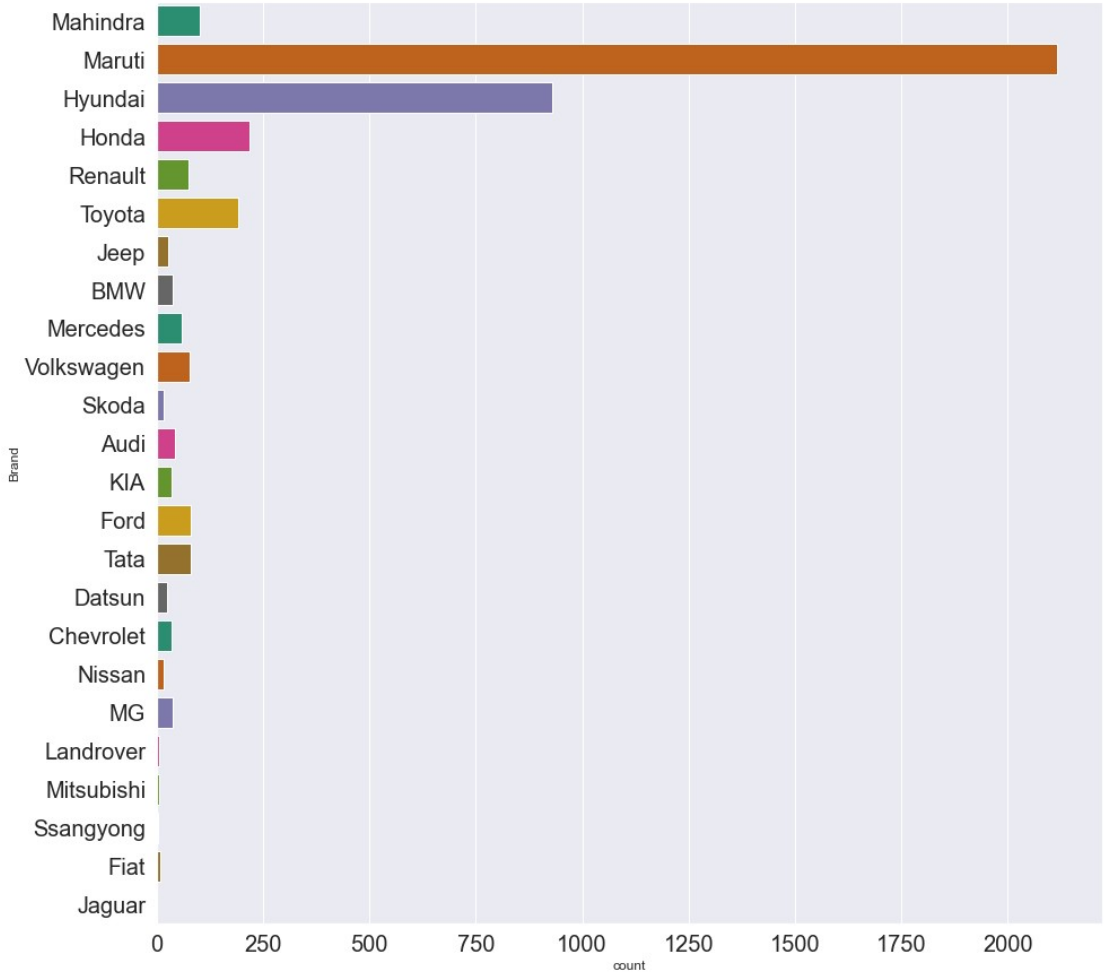


Fig.3 Count-plot for Brands

From the above plot, we can conclude that Maruti is the most trusted brand for buying cars in India.

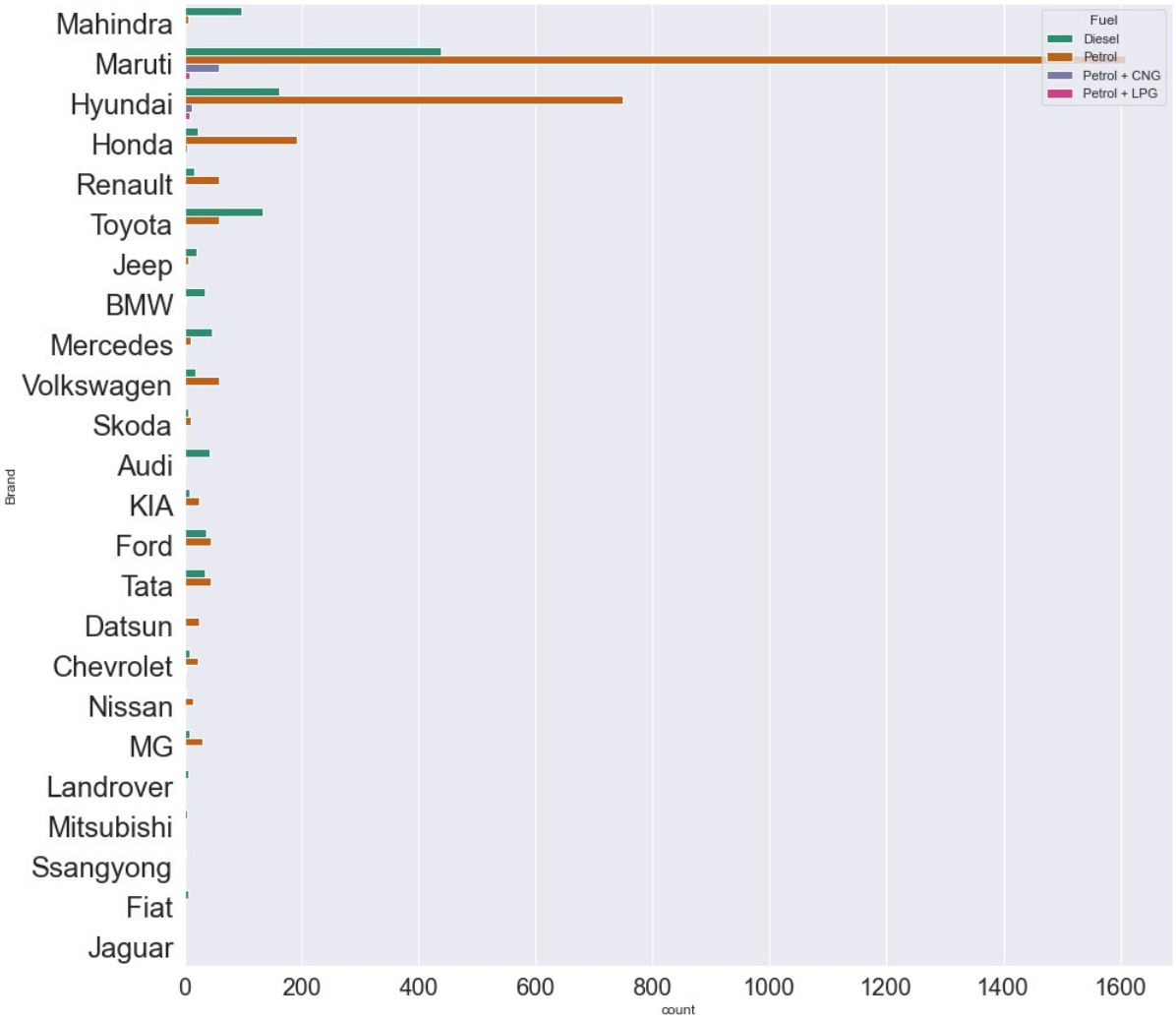


Fig.4 Count-plot for Brands with Fuel as hue

Most of the used cars which are for sale have fuel type Petrol. There are some rare cars with fuel type Petrol + CNG or Petrol + LPG. Maybe customers in India mostly prefer to buy petrol variant cars.

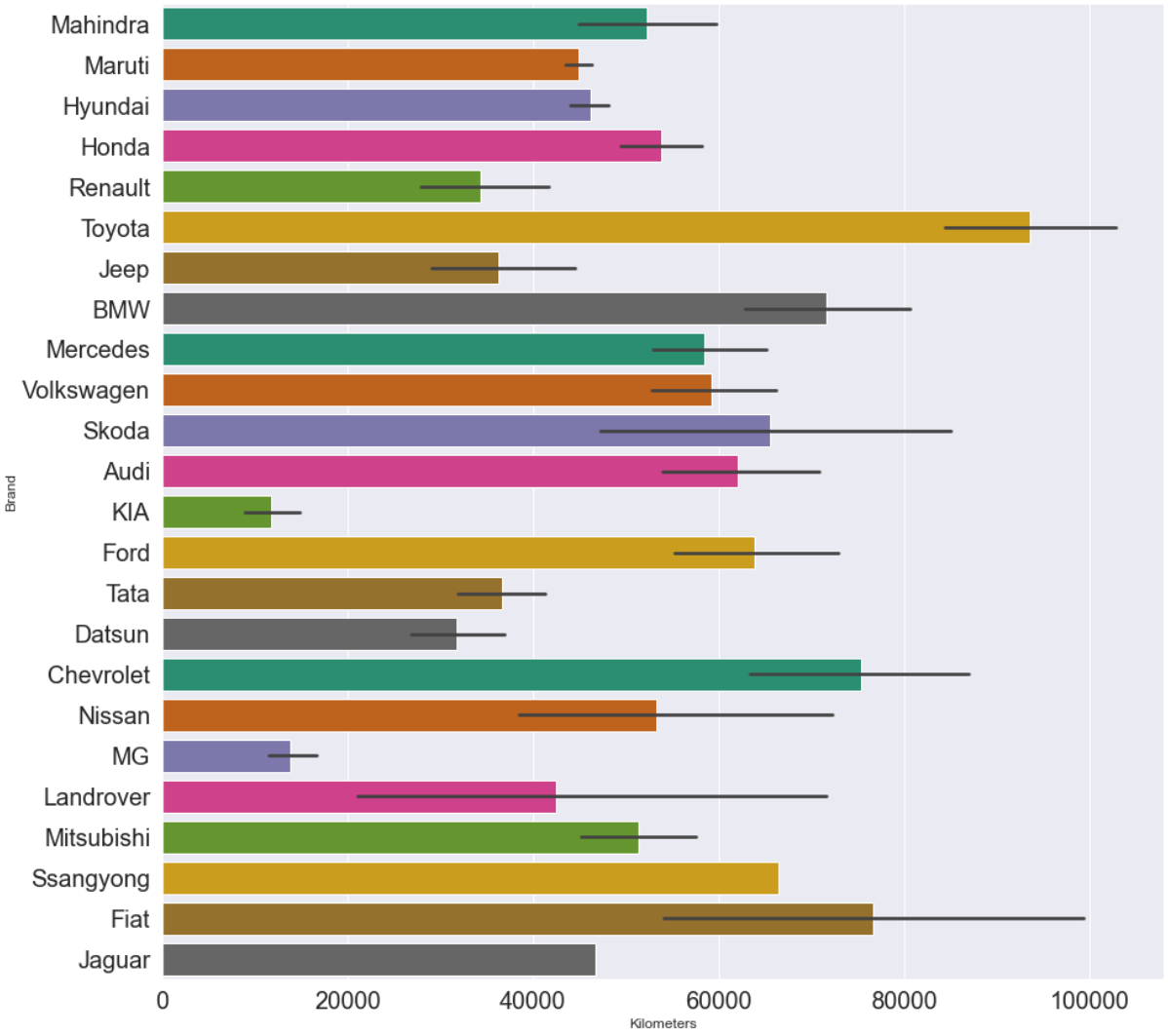


Fig.5 Brand Vs Km

Car brands Toyota, Chevrolet, BMW and Fiat have the highest running kilometres of approximately more that 70,000 km. MG and Kia have the lowest running kilometres less than 20,000 km

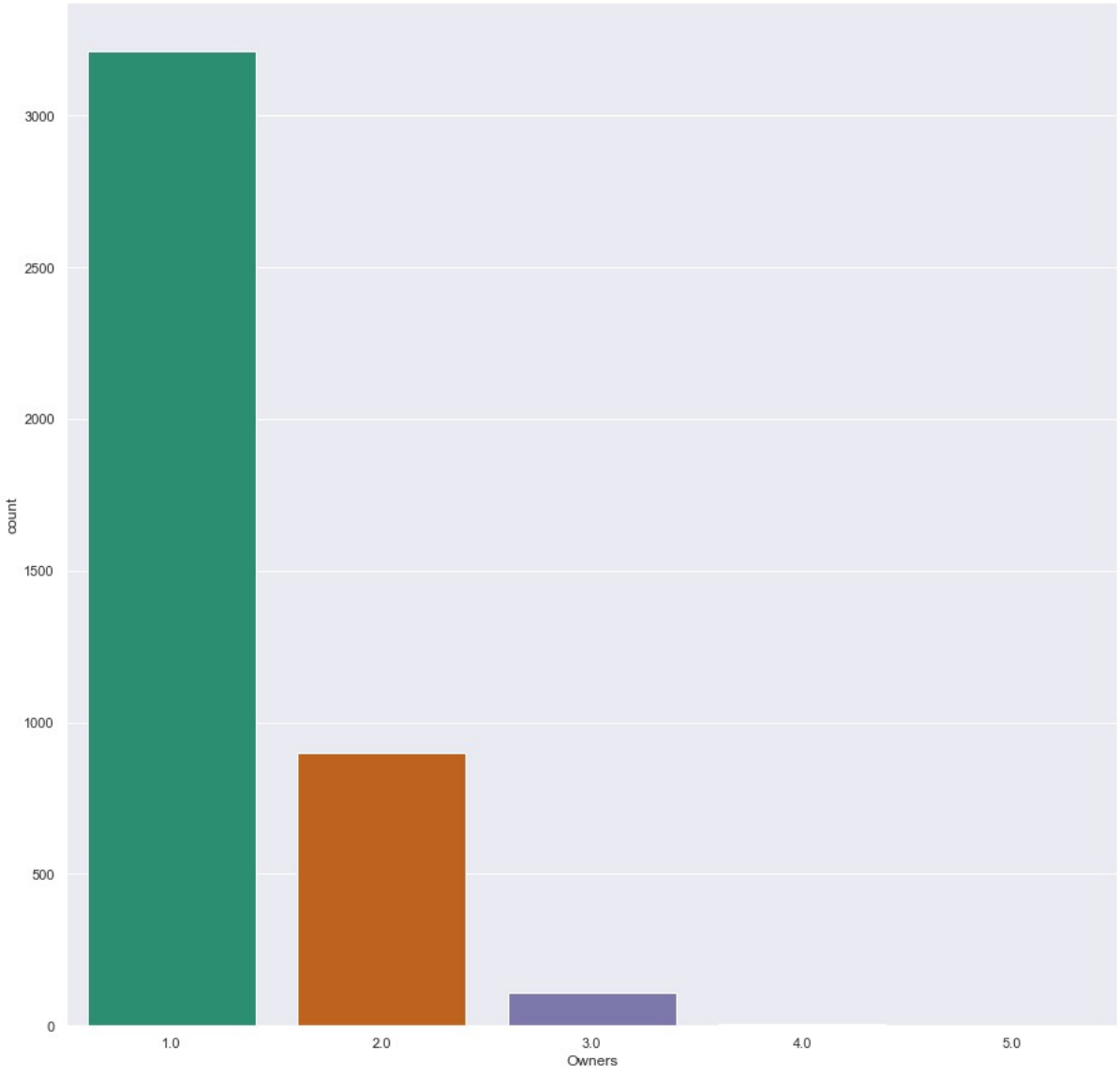


Fig.6 No of Owners

From the plot above we can clearly say that, most of the used cars are single owner cars.

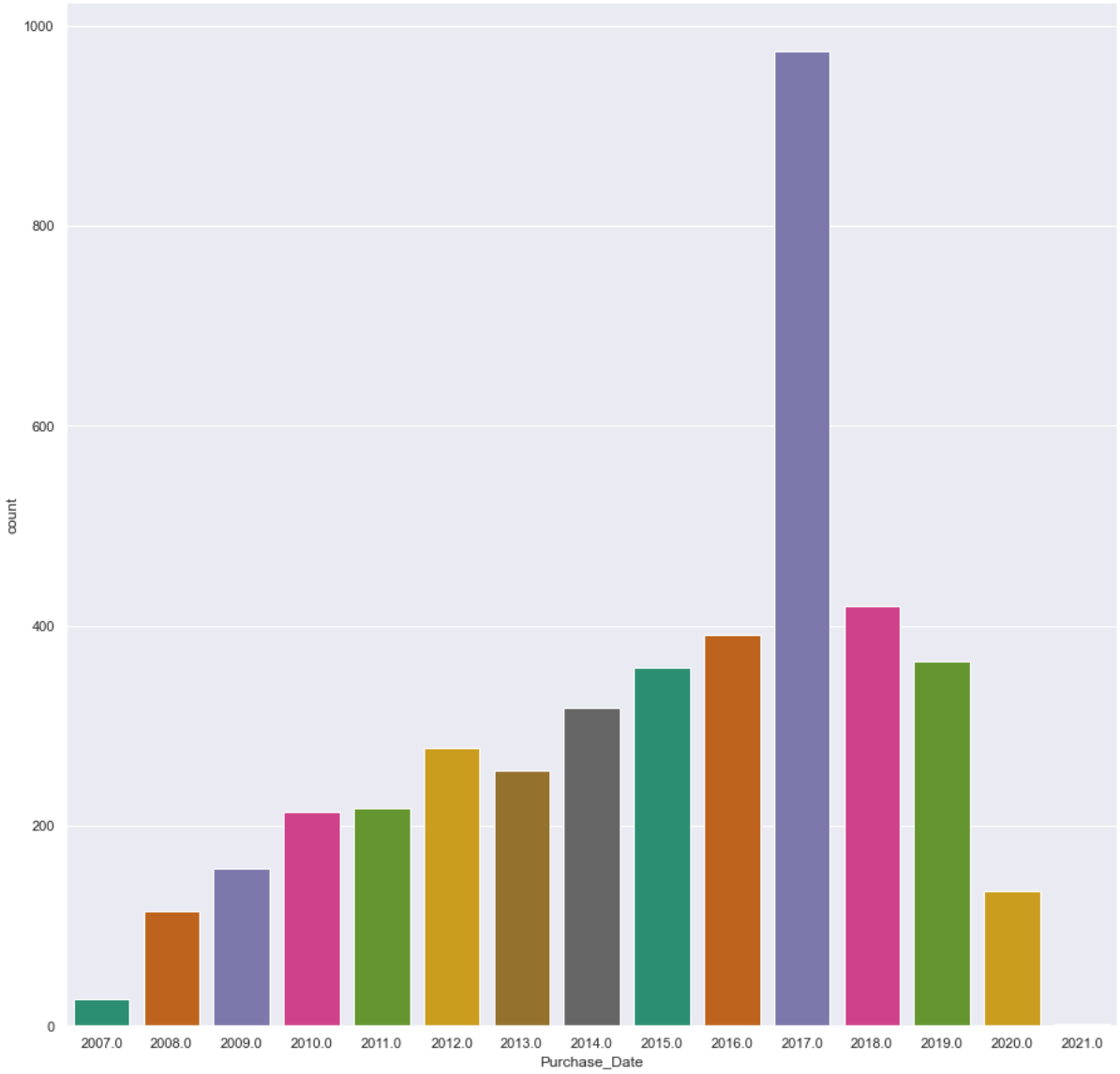


Fig.7 Purchase Year

Most of the cars were purchased in the year 2017

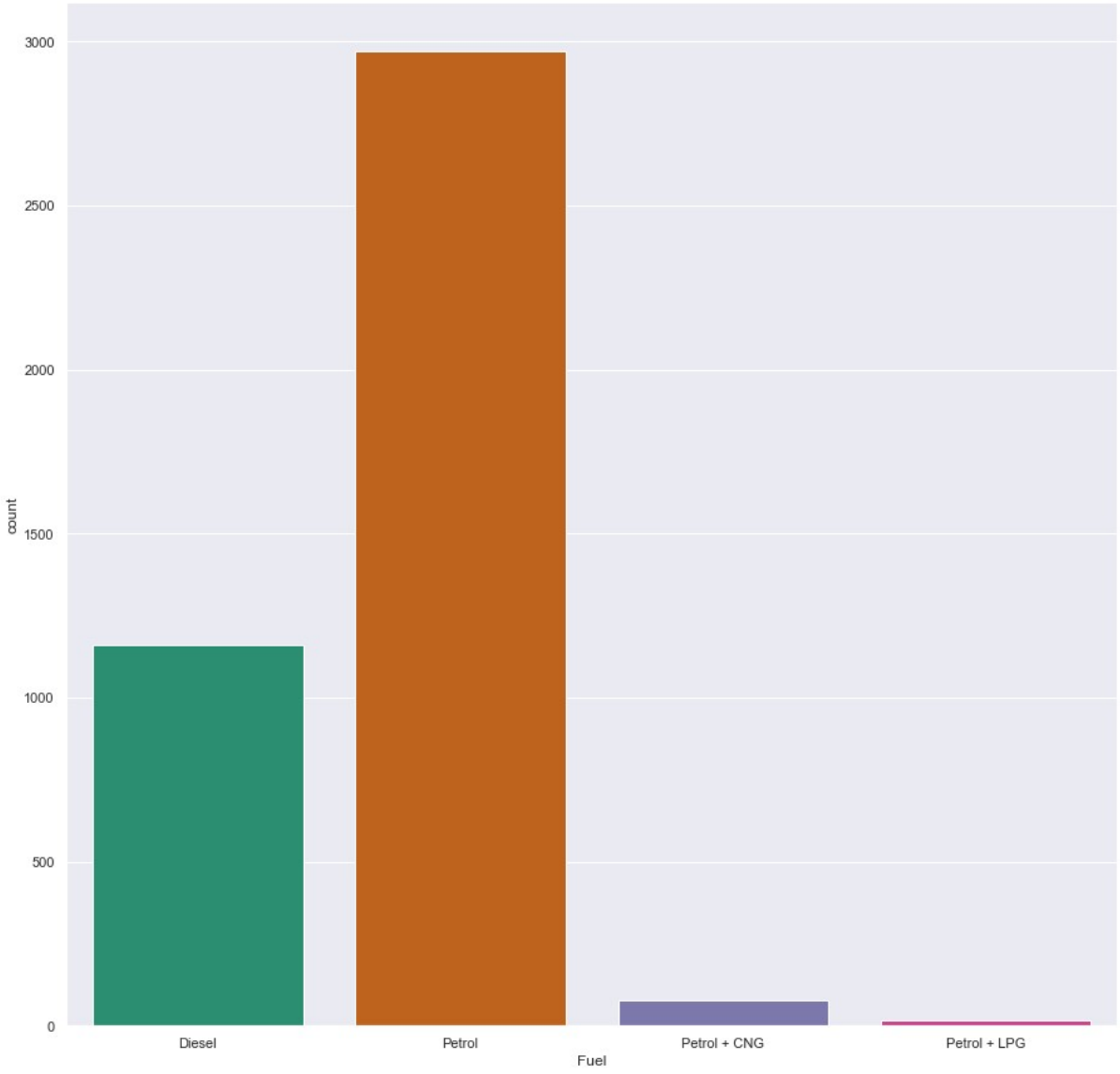


Fig.8 Fuel Type

From the above plot we can conclude that, most of the used cars are petrol variant.

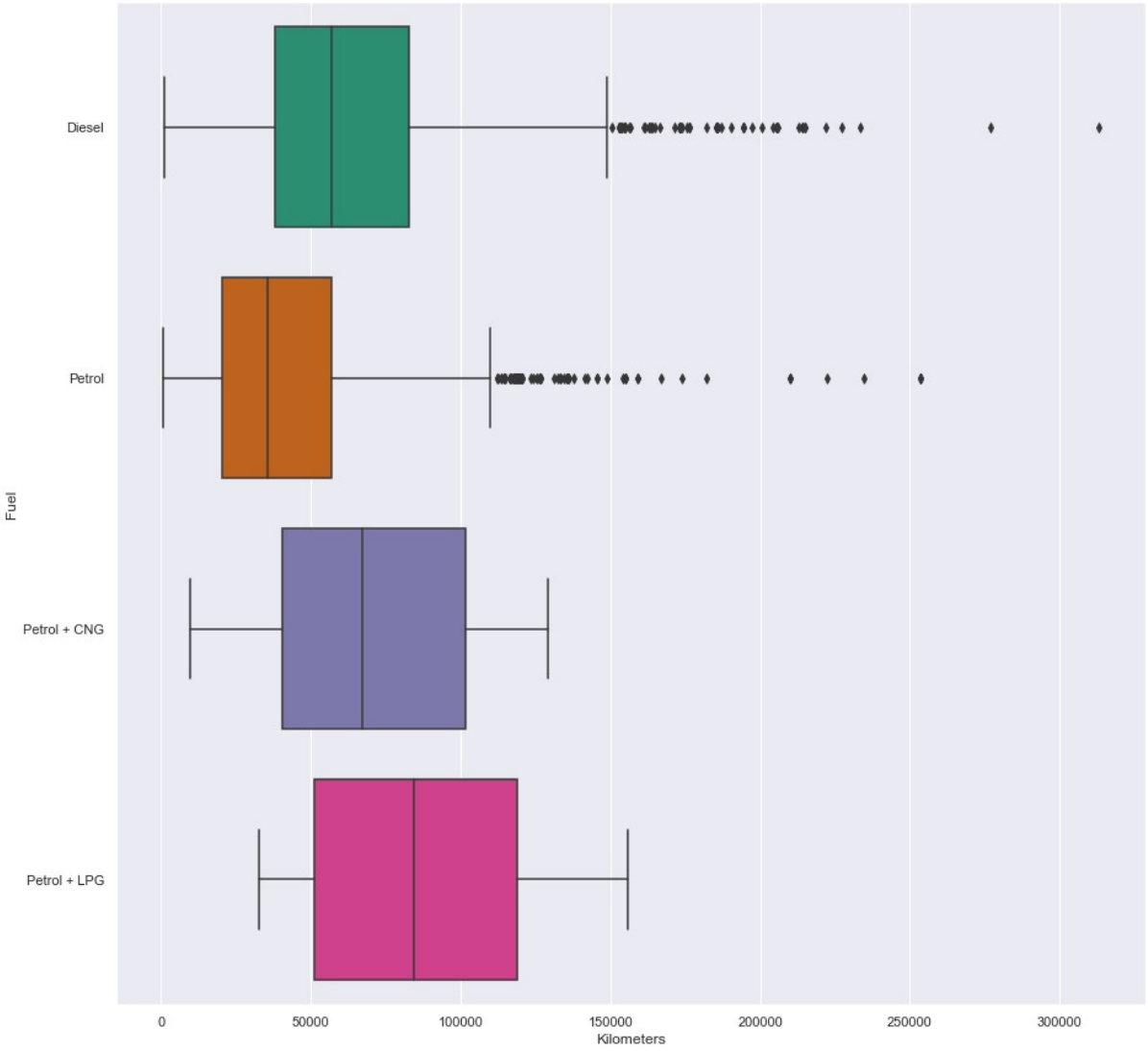


Fig.9 Fuel vs Km

Cars running on fuel type petrol have fewer running kilometres as compared to other fuel types. Fuel type Petrol + LPG have the highest running kilometres more than 80,000 km.

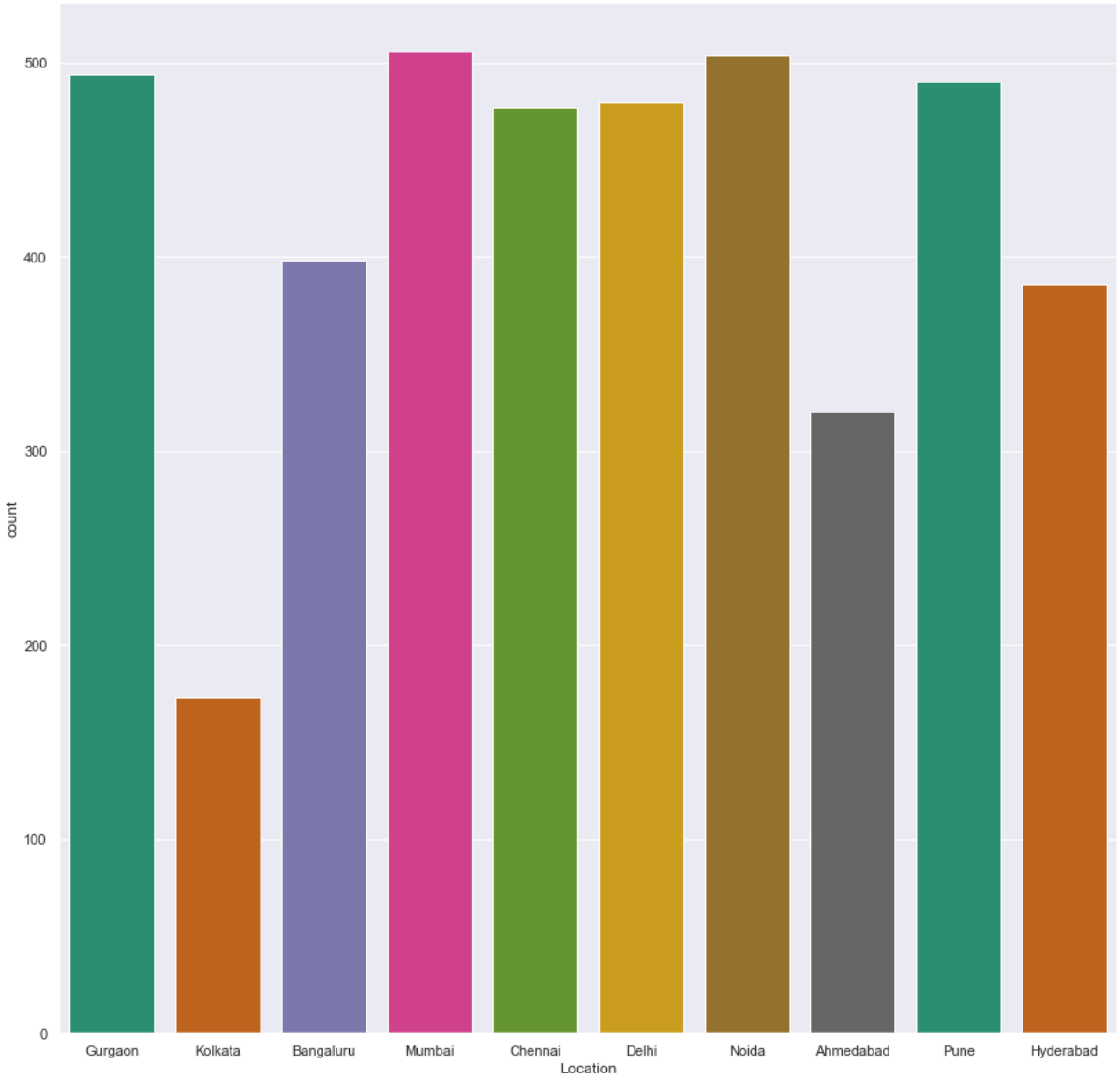


Fig.10 Fuel vs Km

Mumbai has the highest number of used cars recorded in this dataset. Kolkata has the lowest number of used cars.

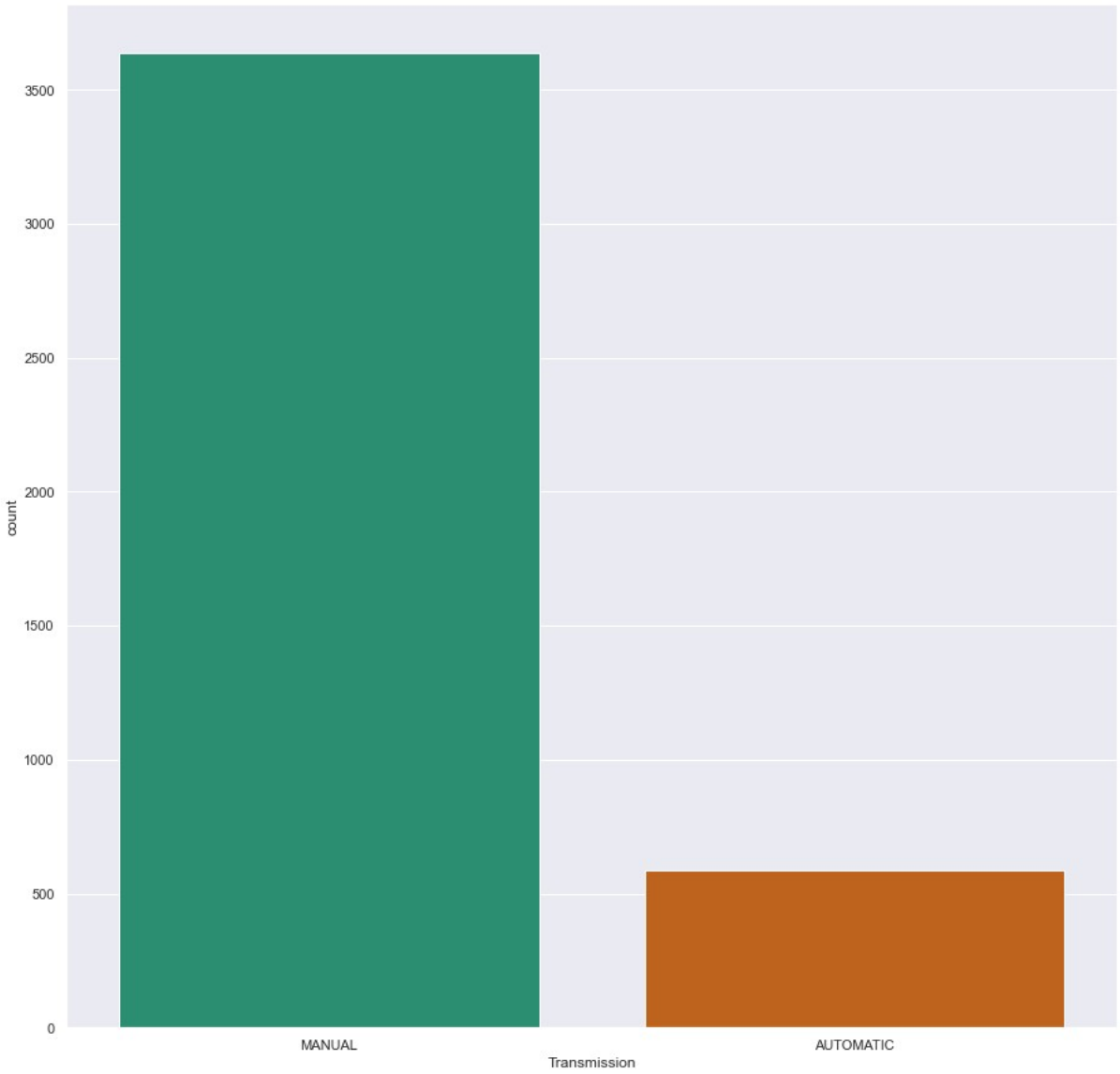


Fig.11 Transmission Type

Most of the customers prefer Manual transmission cars.

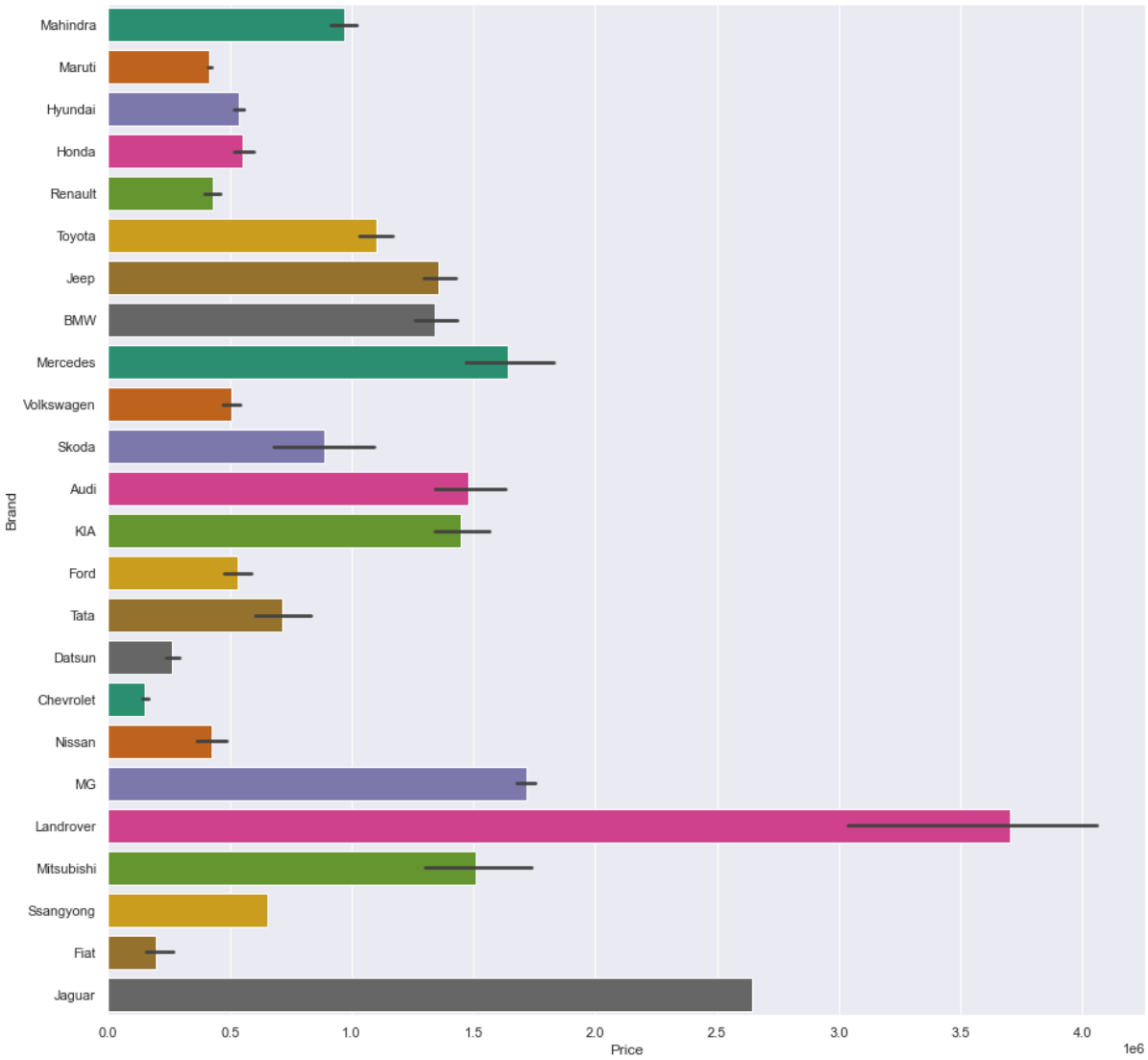


Fig.12 Brand vs Price

We can clearly observe that Luxury brands like Land-rover, Jaguar, and Mercedes are quite expensive. But there are also a lot of cars available for average customers.

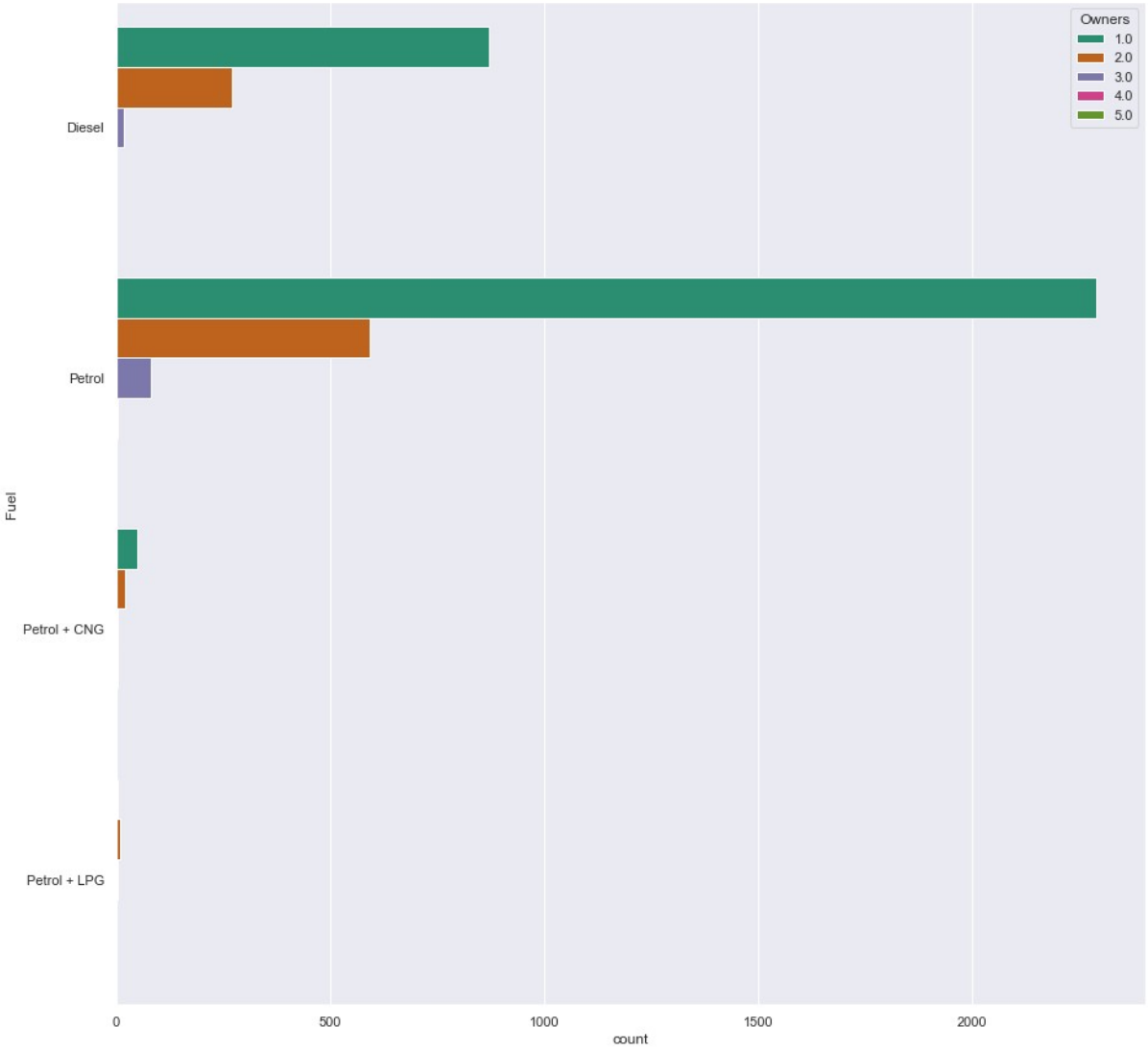


Fig.13 Fuel with hue as Owners

Most of the cars with only one previous owner, are petrol variant cars.

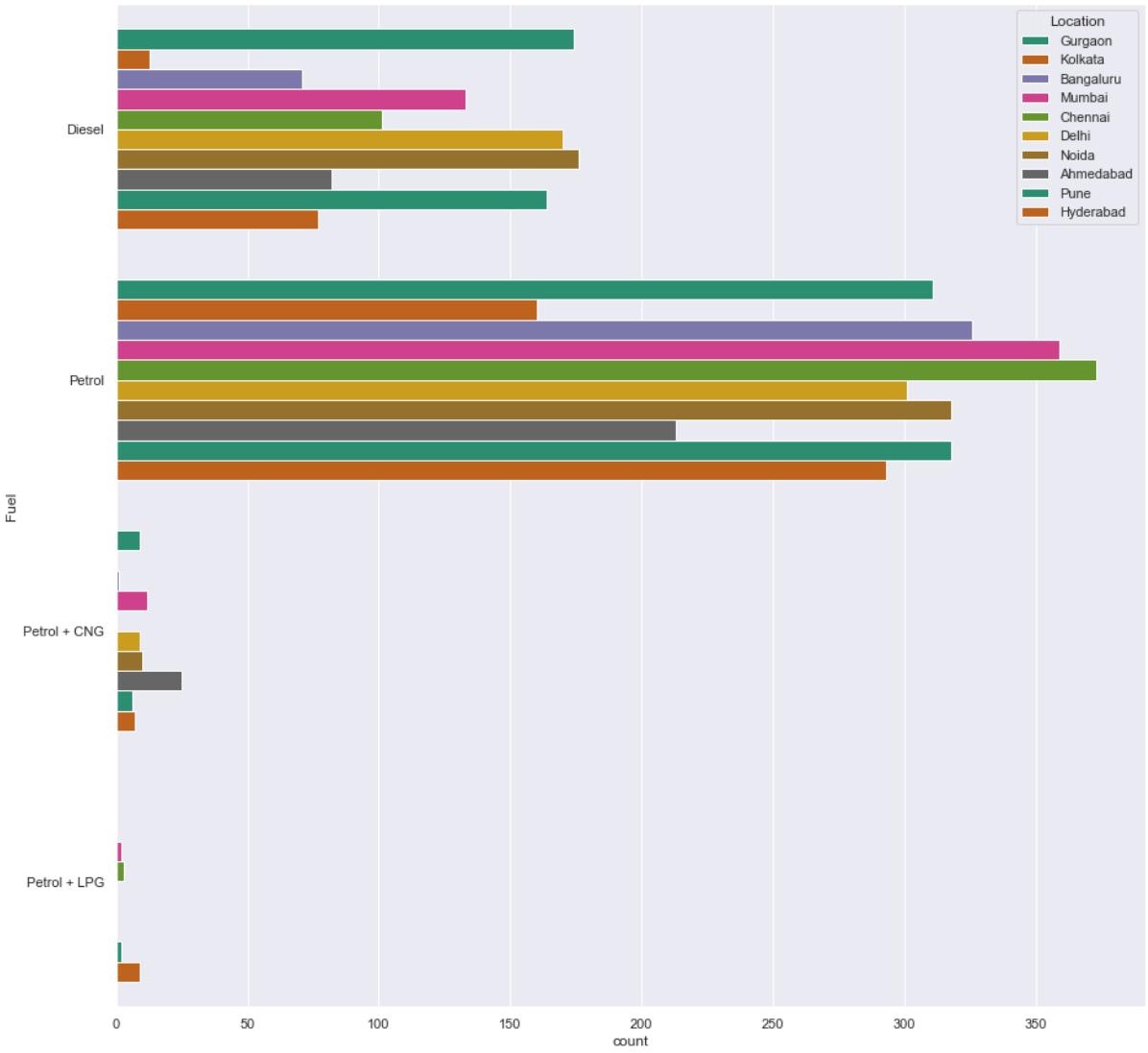


Fig.14 Fuel with hue as Location

Most of the cars with only one previous owner, are petrol variant cars.

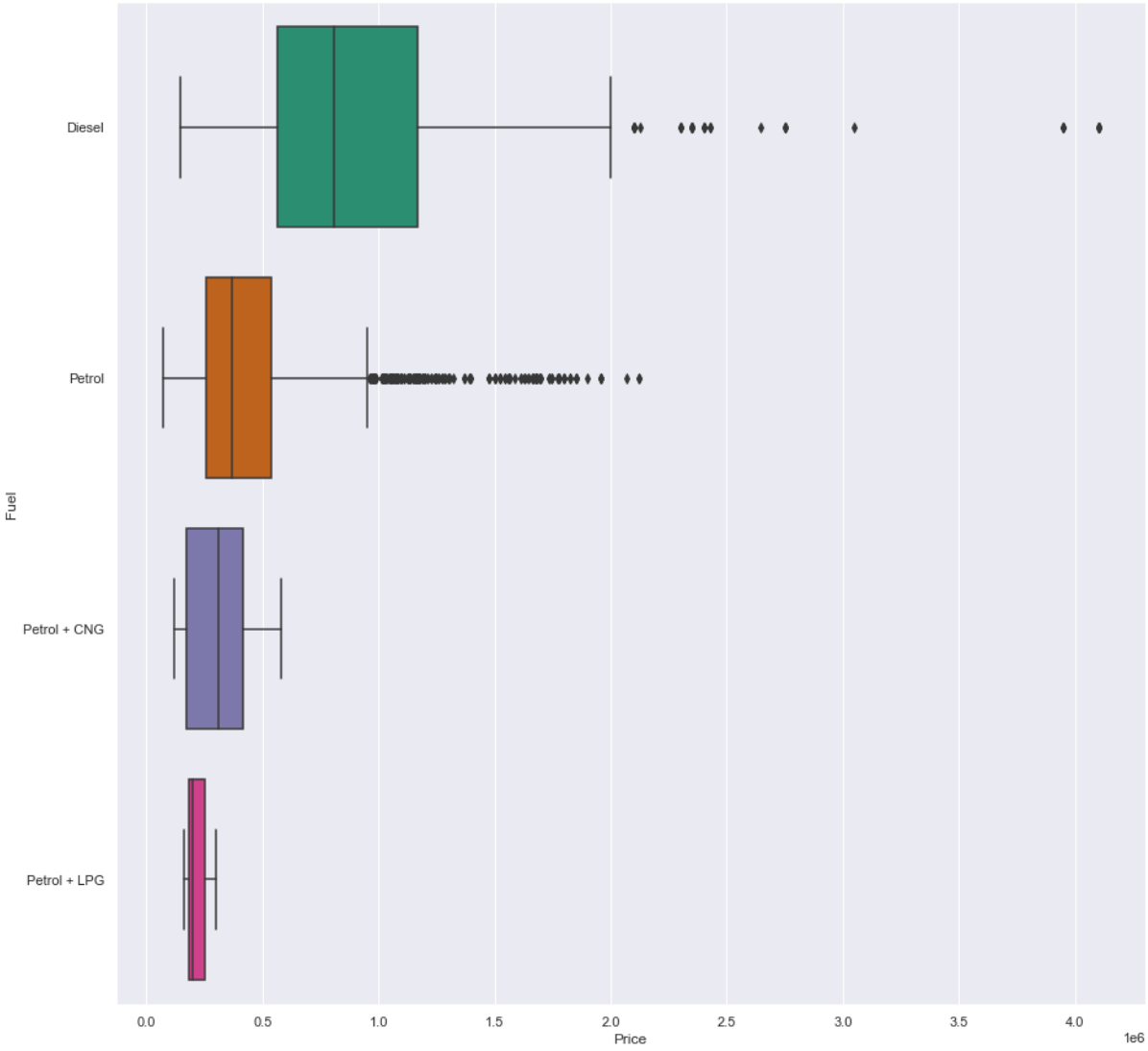


Fig.15 Fuel vs Price

Diesel cars are more expensive than any other fuel type including petrol, LPG and CNG.

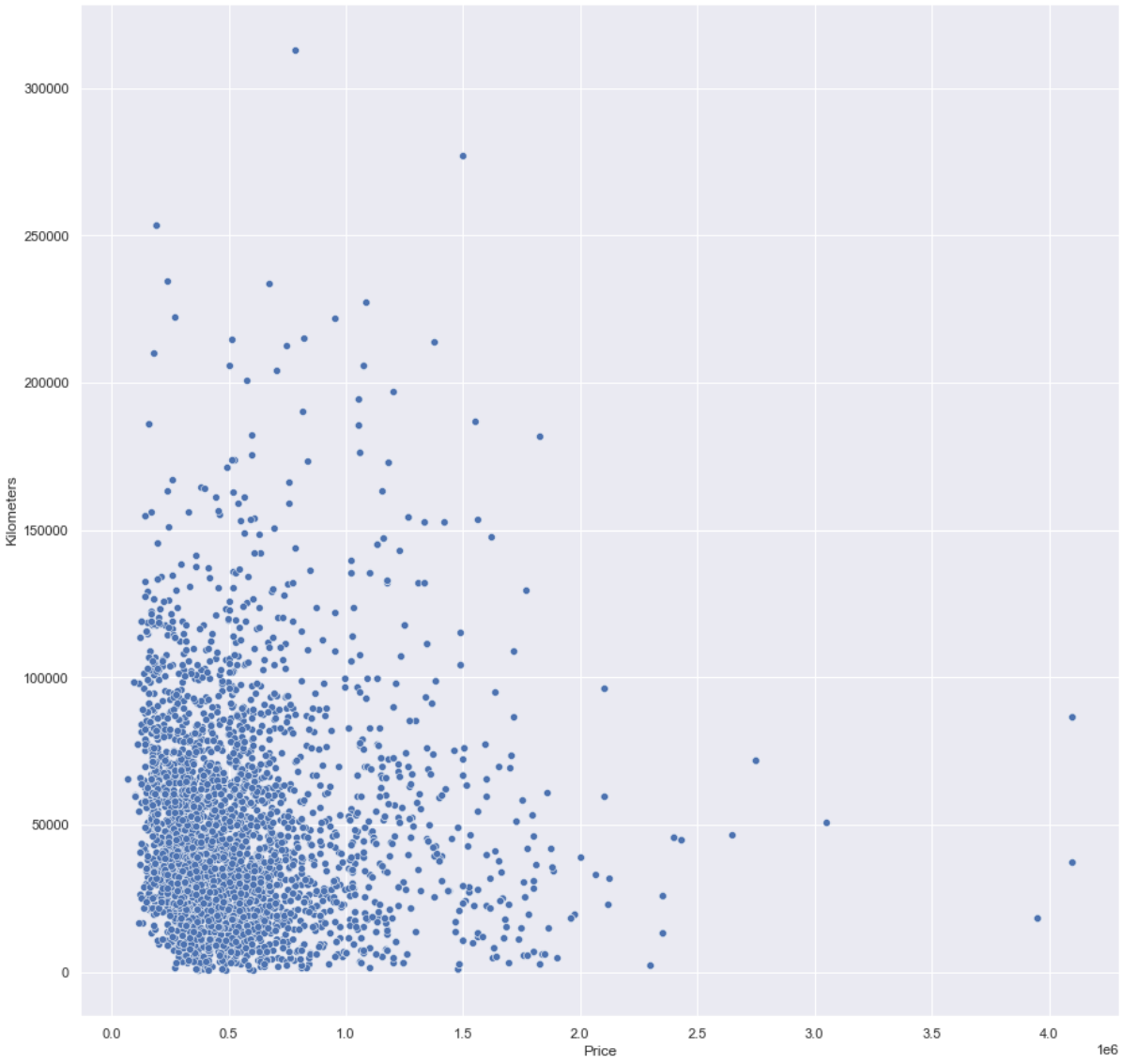


Fig.16 Km vs Price

From the above plot we can say that some of the used cars with less kilometres will be more expensive.

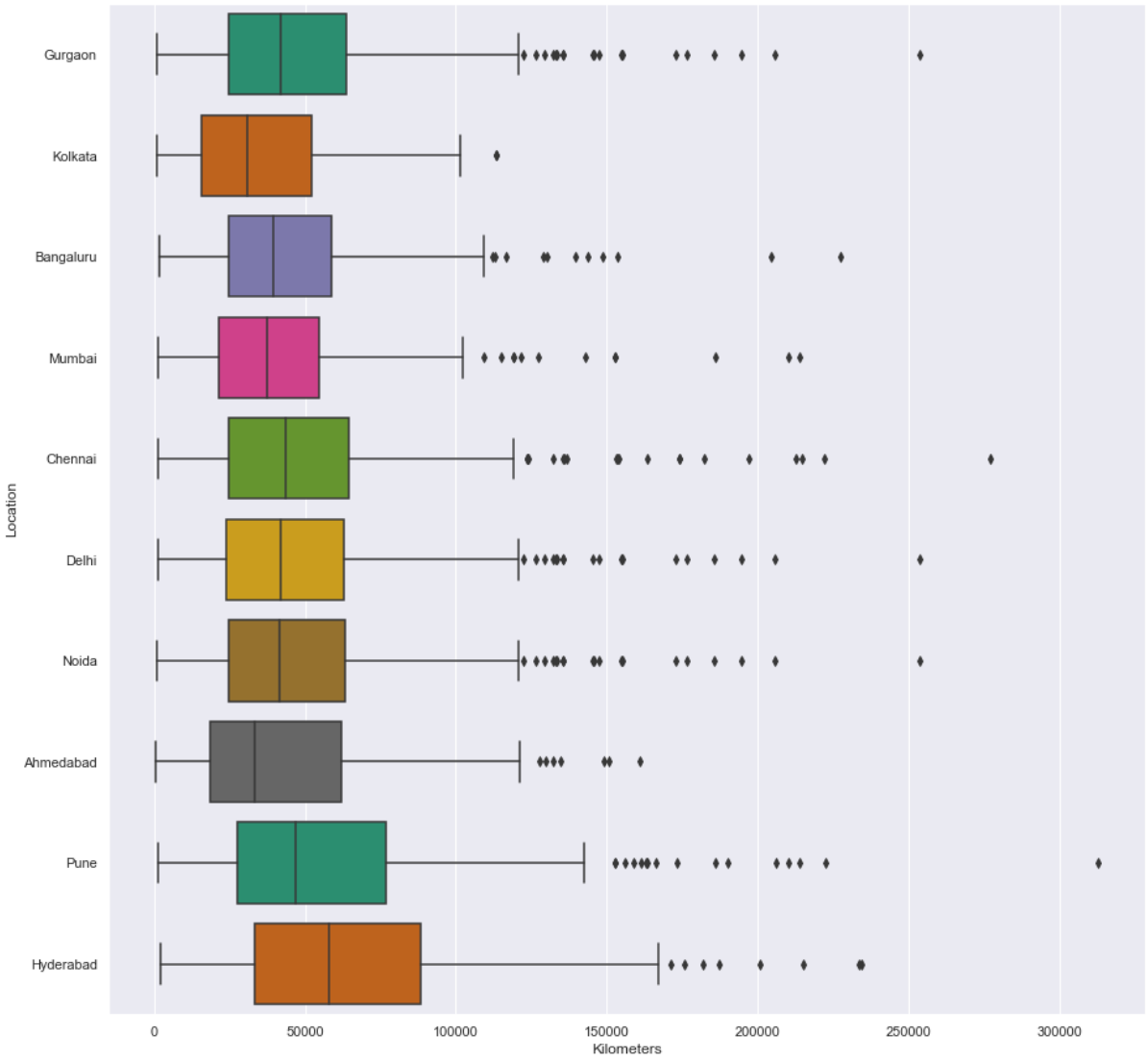


Fig.17 Location vs Km

Used cars from location Hyderabad have the highest running kilometres of more than 65,000 km

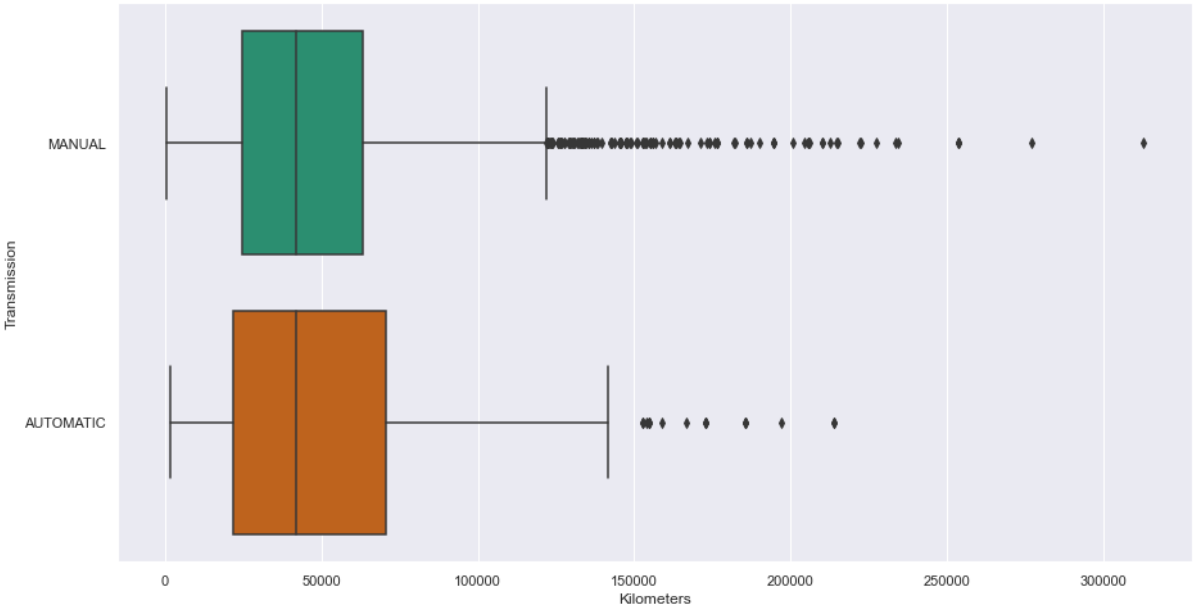


Fig.18 Transmission vs Km

Used Cars with transmission type Automatic have higher running kilometres as compared to transmission type Manual

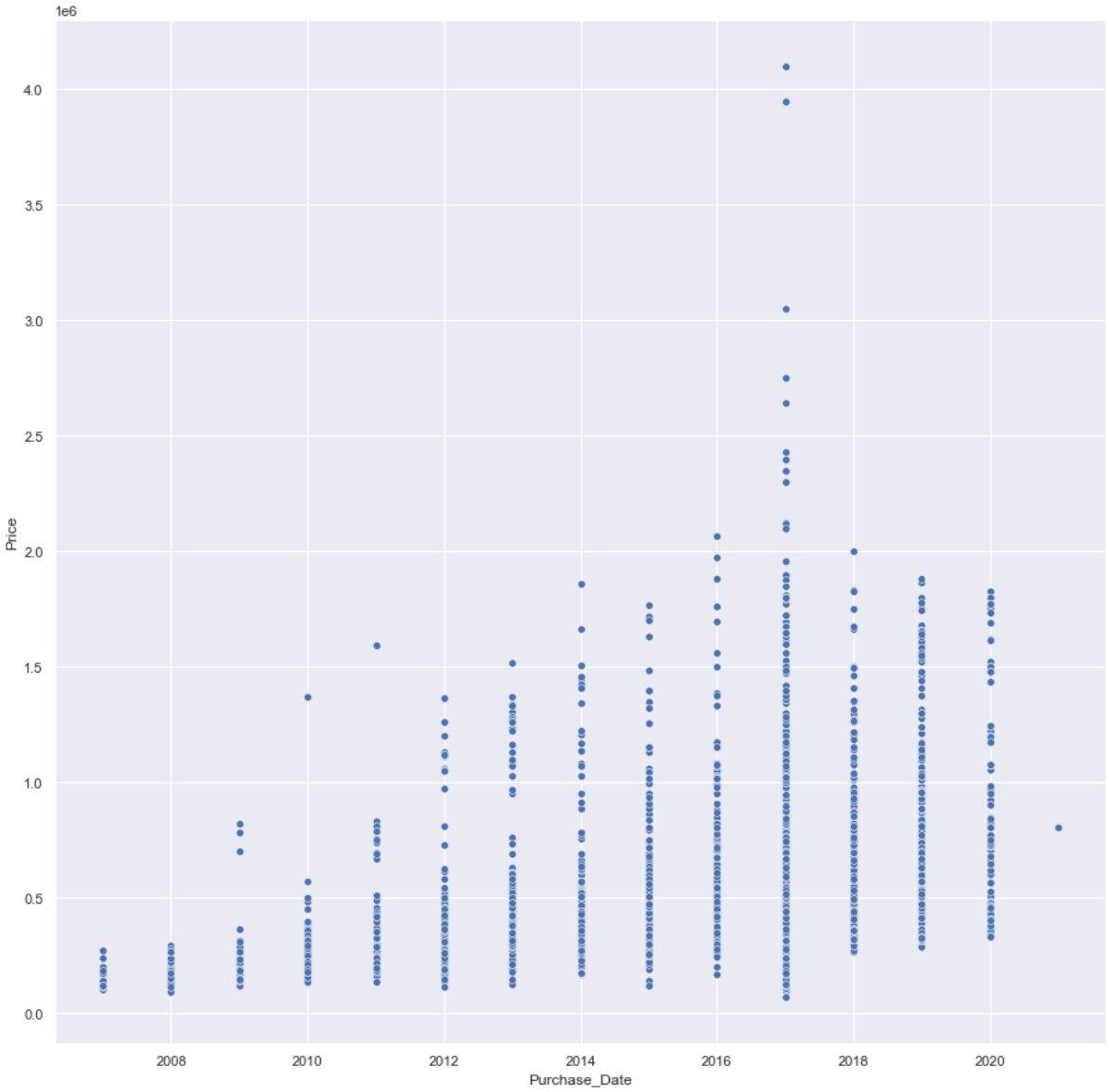


Fig.19 Purchase Year vs Price

We can clearly observe that, the newer the used car is the more expensive it will be.

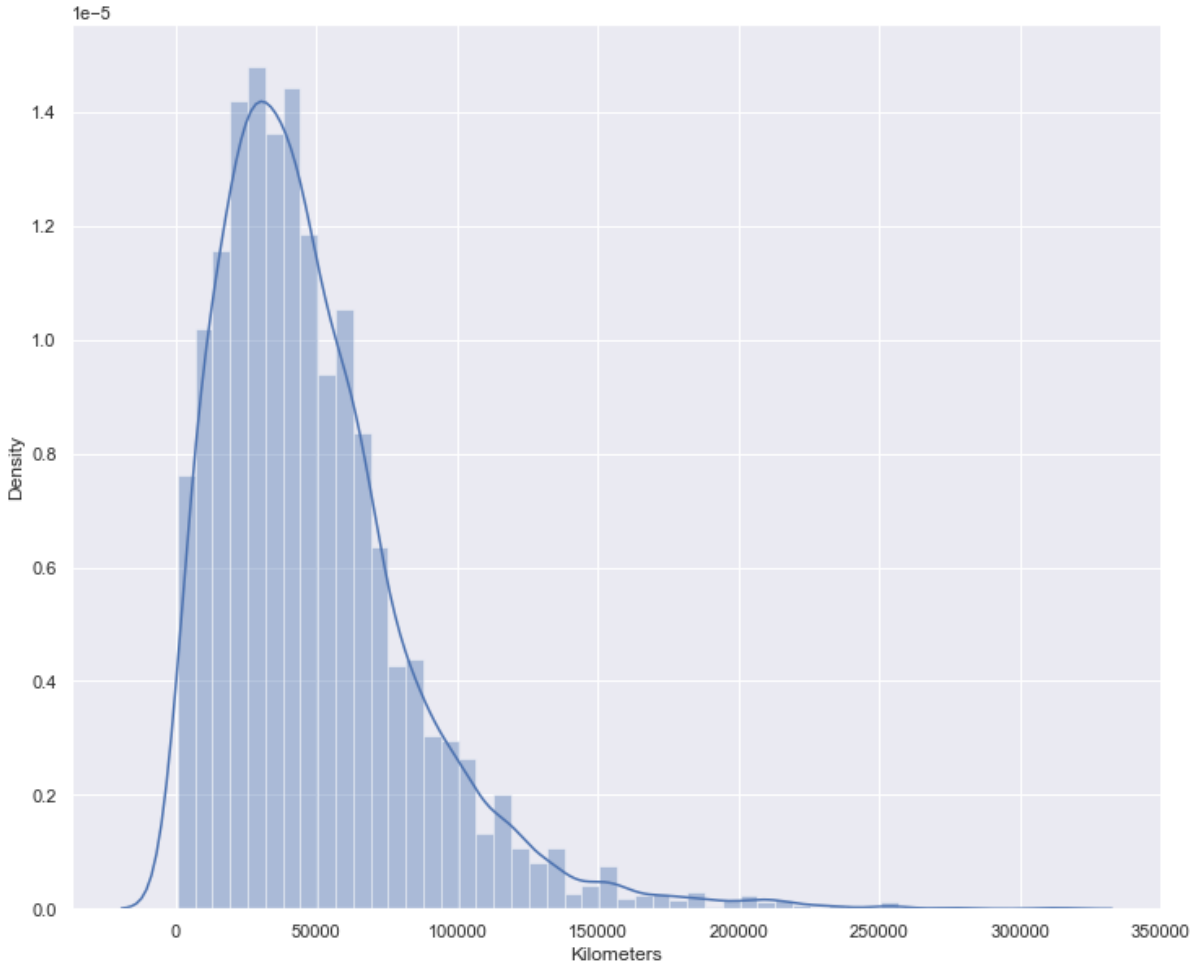


Fig.20 Kilometres

Average running kilometres per car is ~ 48,688. Lowest running km for a car is about 648 km

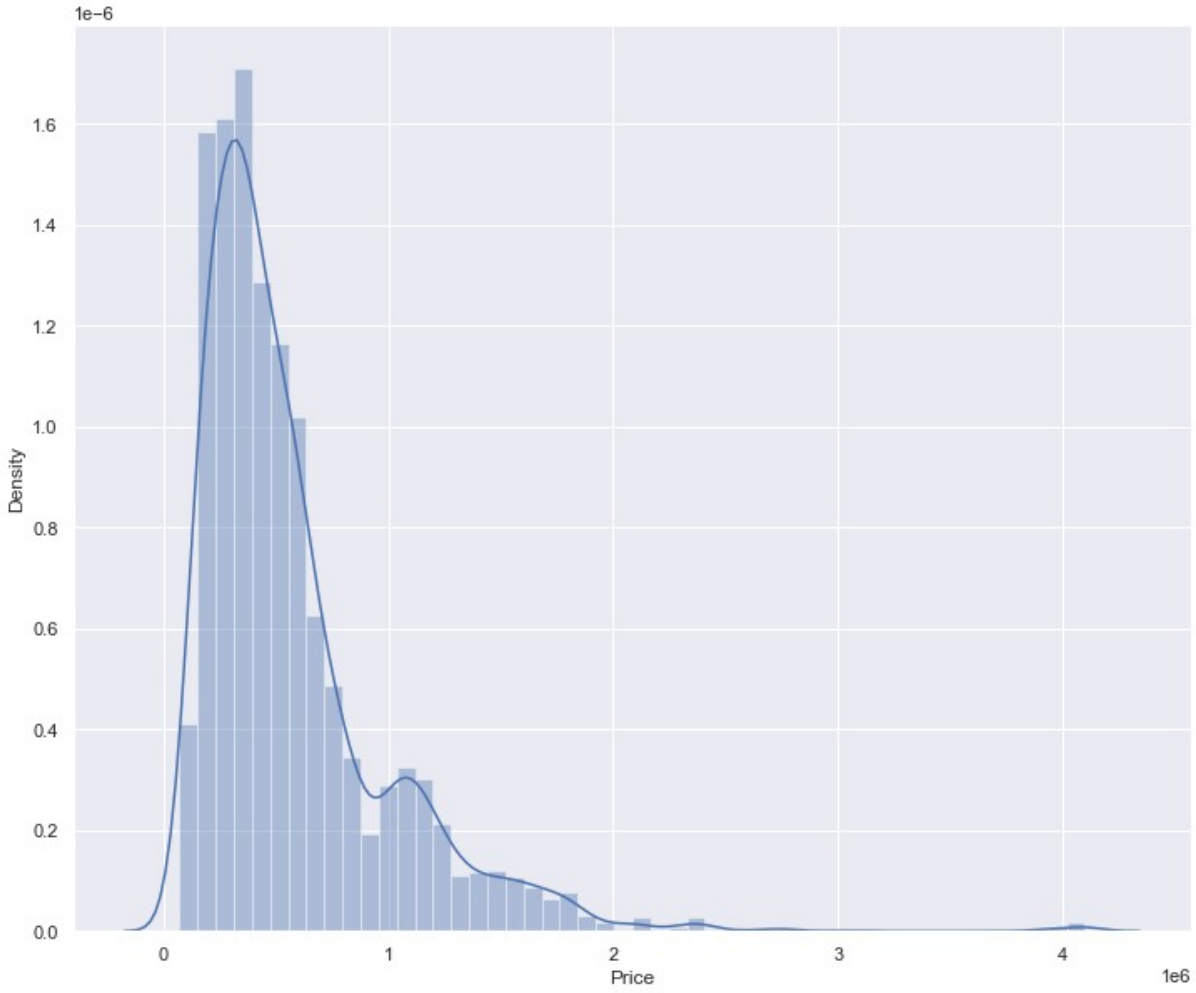


Fig.21 Price

Average price for used cars is about Rs ~ 5,72,921. The price variable looks normally distributed.

- ## Interpretation of the Results

  From all the Brands, Maruti Suzuki is still the most trusted brand by Indian customers.

  Most of the Indian customers are preferring car's with fuel type petrol.

  Many used cars are single owner(only one previous owner)

  Fuel type Petrol + LPG have the highest running kilometres more than 80,000 km

  Manual Transmission is high preference for Indian customers.

  Diesel cars are more expensive than any other fuel type including petrol, LPG and CNG.

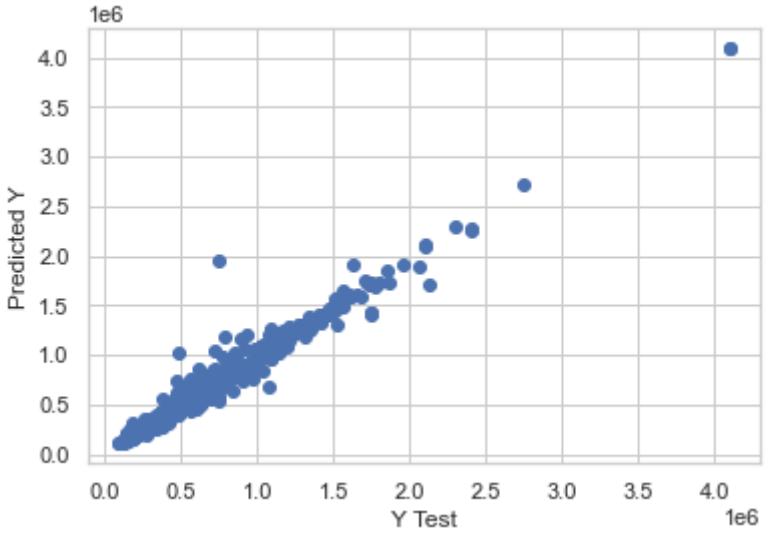
  The lesser the kilometres on car the more expensive it is.

  Used Cars with transmission type Automatic have higher running kilometres as compared to transmission type Manual.

CONCLUSION

- ## Key Findings and Conclusions of the Study

  In this project, eight different machine learning techniques have been used to forecast the price of used cars in Indian market. The first step I took, was to visualize the distribution of each feature and its effect on the Price (dependent variable). From the analysis, I conclude that some of the most useful features for predictions were "Car Name", "Variant", "Purchase Year", "Kilometers". The Gradient Boosting Regression Algorithm proved to be the best model for regression based on the Cross-Validation scores. An accuracy (r2 score) of 0.96 % was achieved by hyper parametric tuning of the model.



From the above plot, we can observe that when we plot the predicted values with the actual values we get a graph that looks some what linear in nature

- ## Limitations of this work and Scope for Future Work

  The main limitation of this study is the low number of records that have been used. As future work, we intend to collect more data and to use more advanced techniques like artificial neural networks, fuzzy logic and genetic algorithms to predict car prices.