REPORT ON CAR PRICE PREDICTION

S. NO.	Topic	Page No.
1	Introduction	2
2	Dataset	3
3	Date Cleaning and Preprocessing	10
5	Model Building	12

1. Introduction

The automotive sector is booming in different places, so to get the most out of it, both car buyers and sellers need to make appropriate decisions that require them to do some analysis and visualization so that they can understand the different types of cars and brands in terms of prices and other factors if someone is new to buying Car or wants to buy a new car.

Business Problems We are Trying to Solve

- Problems we generally face during buying a car like:
    - We are not aware of general factors that influence the car prices
    - Car market monopoly.
    - Tremendous Paperwork
    - Here, we are making predictions of the selling price of the car on the basis of the previously cars.
    - We are also making a conclusion what are the factors affecting the prices of the car.

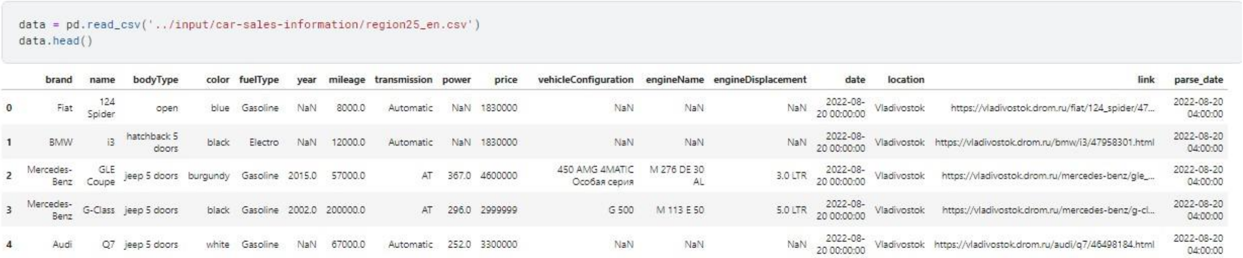
Main Goal

- Create an analytical framework to understand
    - Key factors impacting car price.
- Develop a modelling framework
    - To estimate the price of a car that is up for sale

About Dataset

The data is taken from a popular website in Russia with ads for the sale of cars. Data is collected hourly from the first hundred pages. The only filter is the region to search for. The date and time of selection is indicated in the "parse_date" column.

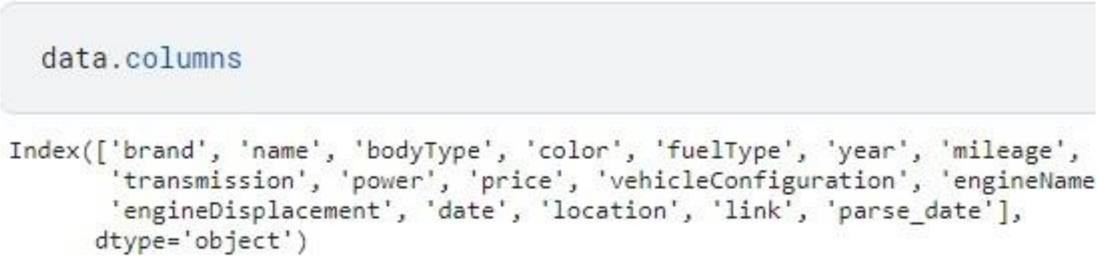
This is what the dataset looks like,



```
data = pd.read_csv('../input/car-sales-information/region25_en.csv')
data.head()
```

	brand	name	bodyType	color	fuelType	year	mileage	transmission	power	price	vehicleConfiguration	engineName	engineDisplacement	date	location	link	parse_date
0	Fiat	124 Spider	open	blue	Gasoline	NaN	8000.0	Automatic	NaN	1830000	NaN	NaN	NaN	2022-08-20 00:00:00	Vladivostok	https://vladivostok.drom.ru/fiat/124_spider/47...	2022-08-20 04:00:00
1	BMW	i3	hatchback 5 doors	black	Electro	NaN	12000.0	Automatic	NaN	1830000	NaN	NaN	NaN	2022-08-20 00:00:00	Vladivostok	https://vladivostok.drom.ru/bmw/i3/47958301.html	2022-08-20 04:00:00
2	Mercedes-Benz	GLE Coupe	jeep 5 doors	burgundy	Gasoline	2015.0	57000.0	AT	367.0	4600000	450 AMG 4MATIC Особая серия	M 276 DE 30 AL	3.0 LTR	2022-08-20 00:00:00	Vladivostok	https://vladivostok.drom.ru/mercedes-benz/gle_...	2022-08-20 04:00:00
3	Mercedes-Benz	G-Class	jeep 5 doors	black	Gasoline	2002.0	200000.0	AT	296.0	2999999	G 500	M 113 E 50	5.0 LTR	2022-08-20 00:00:00	Vladivostok	https://vladivostok.drom.ru/mercedes-benz/g-cl...	2022-08-20 04:00:00
4	Audi	Q7	jeep 5 doors	white	Gasoline	NaN	67000.0	Automatic	252.0	3300000	NaN	NaN	NaN	2022-08-20 00:00:00	Vladivostok	https://vladivostok.drom.ru/audi/q7/46498184.html	2022-08-20 04:00:00

It has lots of rows and columns, which is 1513200 rows spread across 17 columns.
Here is the list of columns this dataset has,



```
data.columns
```

```
Index(['brand', 'name', 'bodyType', 'color', 'fuelType', 'year', 'mileage',
       'transmission', 'power', 'price', 'vehicleConfiguration', 'engineName
       'engineDisplacement', 'date', 'location', 'link', 'parse_date'],
      dtype='object')
```

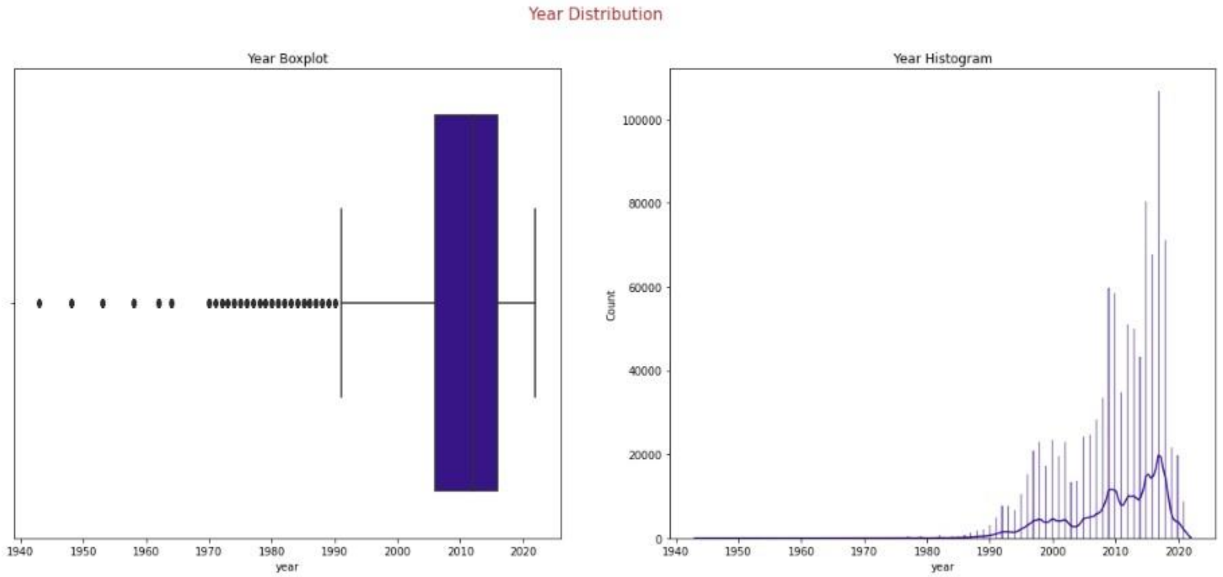
2. EDA & Business Implication

EDA stands for exploratory data analysis where we explore our data and grab insights from it. EDA helps us in getting knowledge in form of various plots and diagrams where we can easily understand the data and its features.

Analysis of year

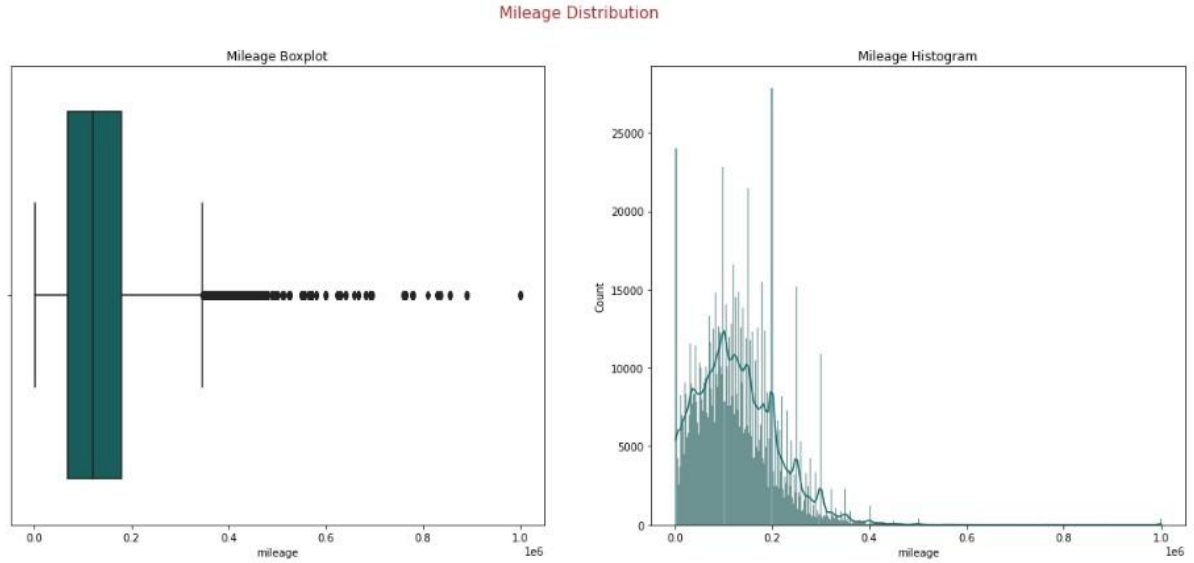
Observation:

- We can see from the graph that in april 2015 and june and july of 2014 most houses are sold.



- In the dataset, we can see the sales is high from 2010 to 2019.

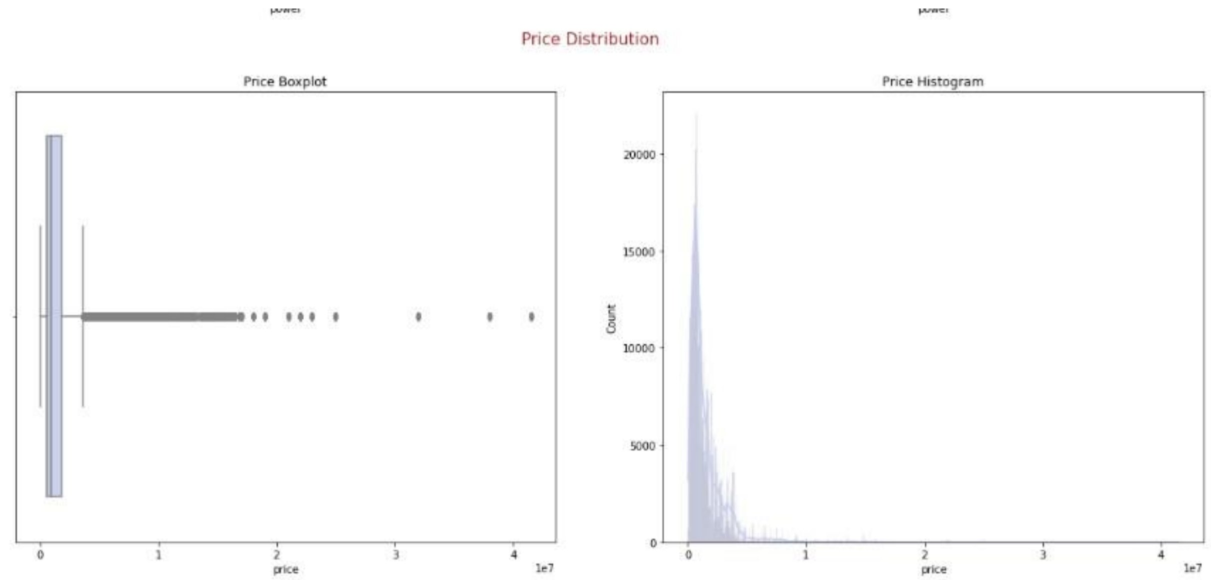
Analysis of mileage:



Observation:

This is the distribution of mileage and outliers.

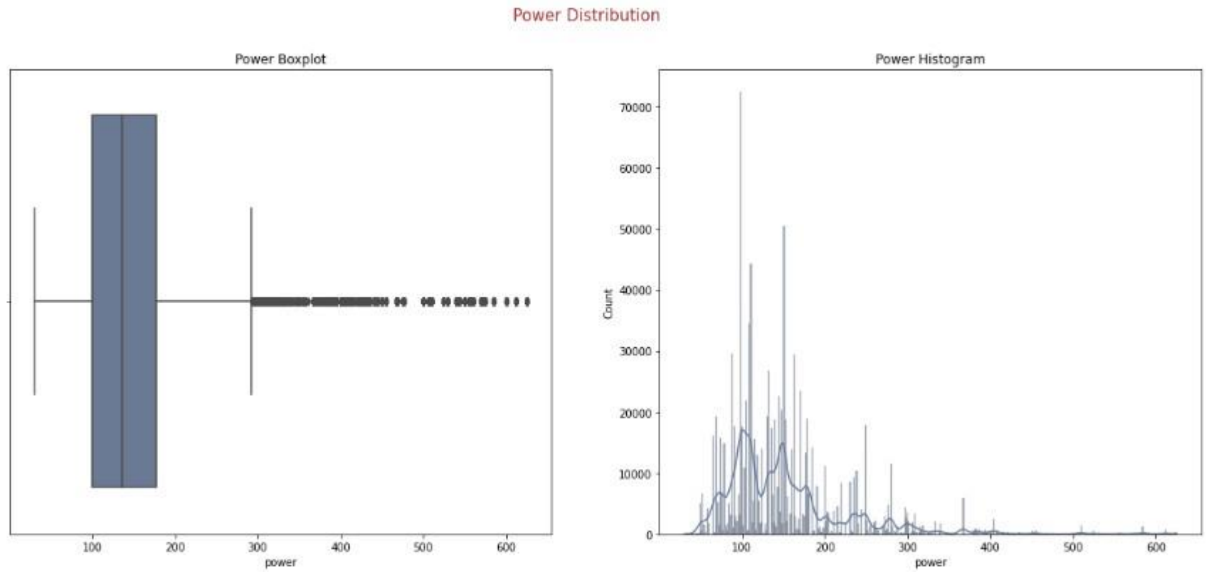
Analysis of price:



Observation:

This is the distribution of price and outliers.

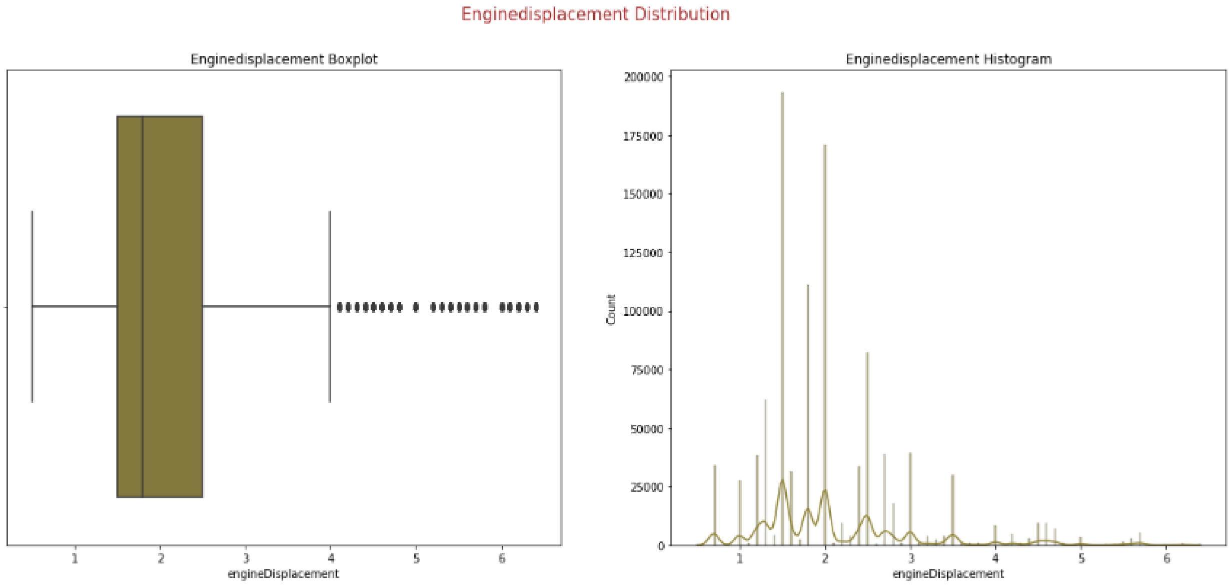
Analysis of power:



Observation:

This is the distribution of power and outliers.

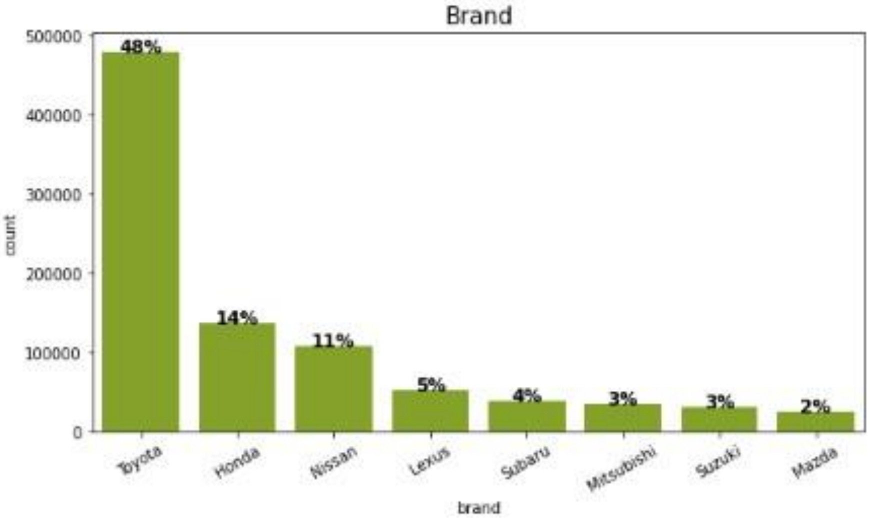
Analysis of engine displacement:



Observation:

This is the distribution of engine and outliers.

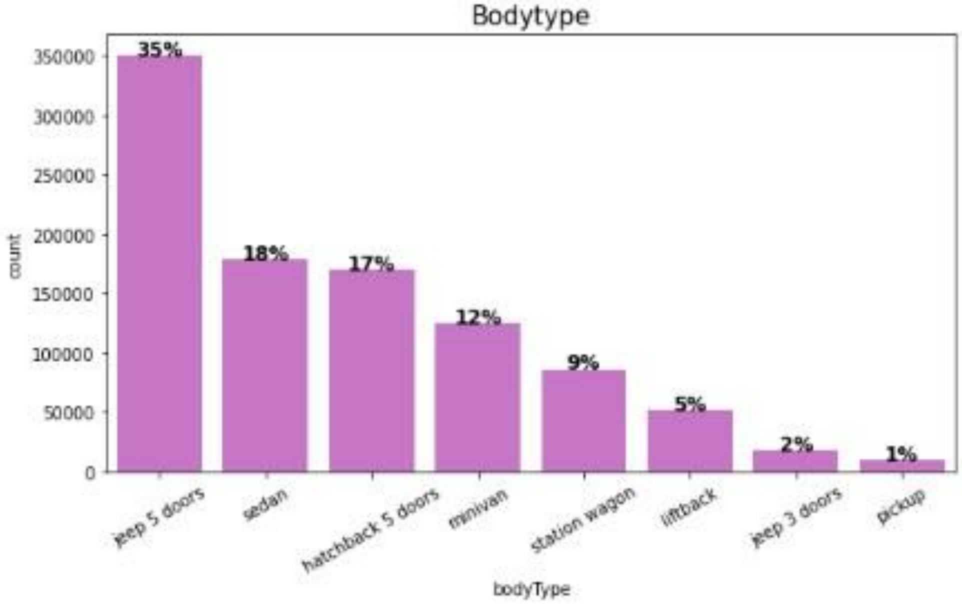
Analysis of Brand:



Observation:

Toyota is the largest Brand in our dataset

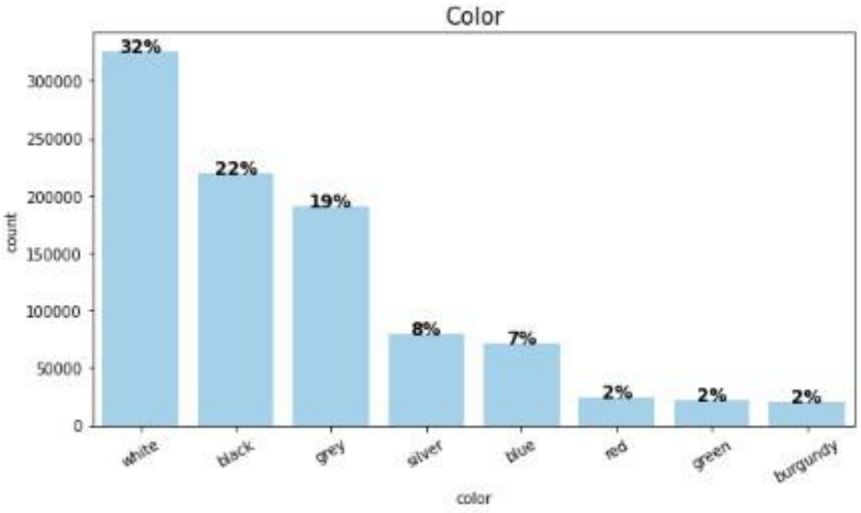
Analysis of Body type:



Observation:

Jeep5 doors is the largest Body type in our dataset

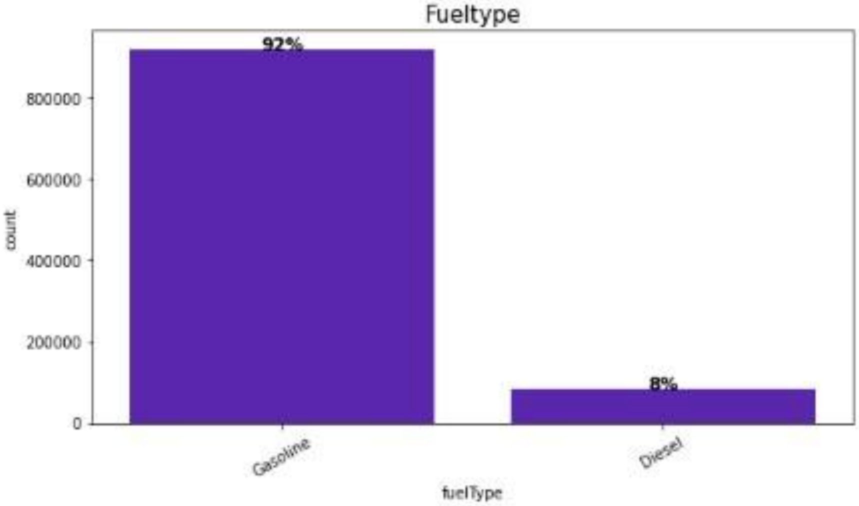
Analysis of Color:



Observation:

White is the largest Color in our dataset

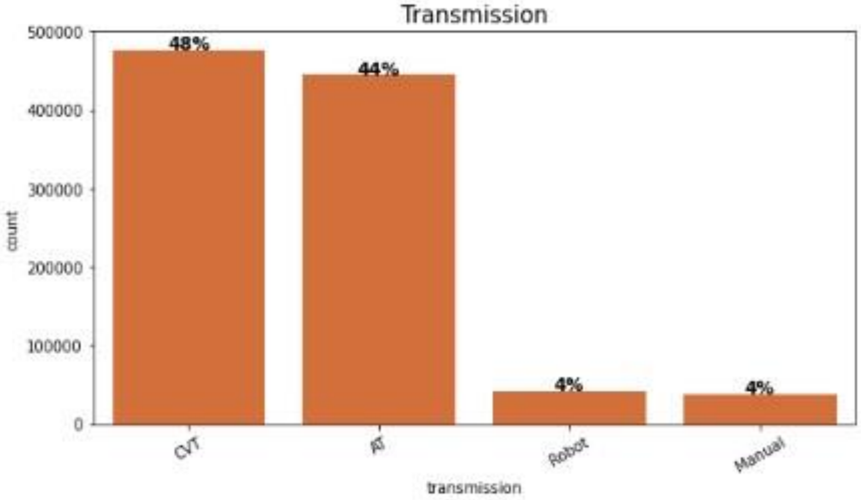
Analysis of Fuel type:



Observation:

This is Fuel Type of cars in dataset

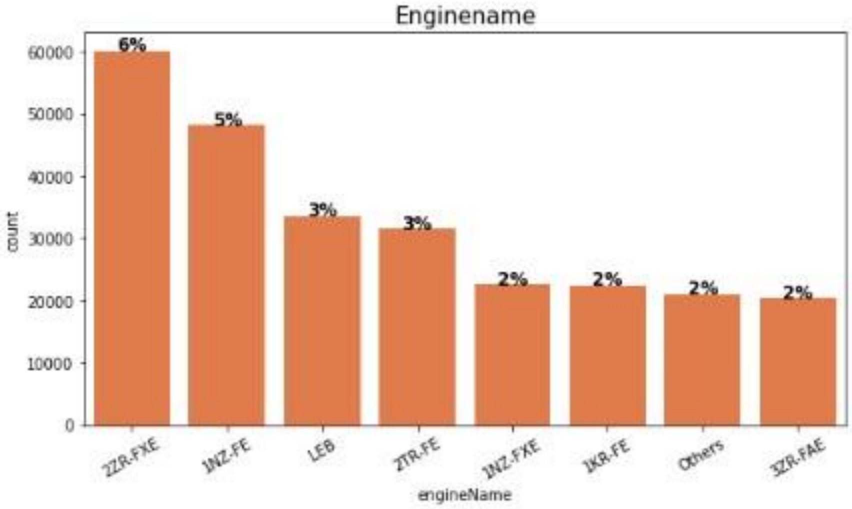
Analysis of transmission:



Observation:

CVT is the largest transmission in our dataset

Analysis of engineName:



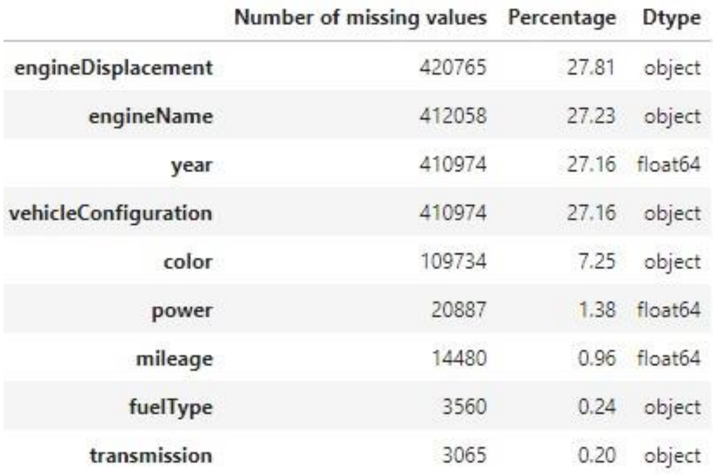
Observation:

2ZR-FXE is the largest engine in our dataset

3. Data Cleaning & Pre-processing

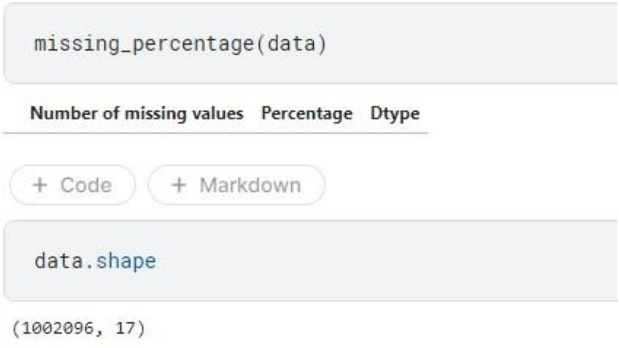
Data Cleaning is an important phase in any data science project, if our data is clean then only we can provide it to our machine learning model. Uncleaned Data can further lead our model with low accuracy. And, If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset.

The approach used for identifying and treating missing values and outlier treatment:-



	Number of missing values	Percentage	Dtype
engineDisplacement	420765	27.81	object
engineName	412058	27.23	object
year	410974	27.16	float64
vehicleConfiguration	410974	27.16	object
color	109734	7.25	object
power	20887	1.38	float64
mileage	14480	0.96	float64
fuelType	3560	0.24	object
transmission	3065	0.20	object

We have a lot of missing values but we have above 1.5M rows so we try to drop these rows after that we got 1M rows so it's good for me that a lot of rows.



```
missing_percentage(data)
```

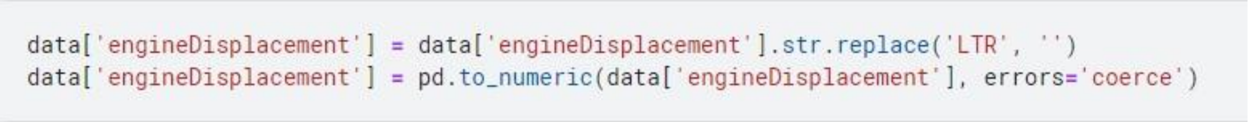
Number of missing values	Percentage	Dtype

```
data.shape
```

```
(1002096, 17)
```

Handle engineDisplacement column:-

To convert engineDisplacement column to numaric we remove LTR from this column to can convert it



```
data['engineDisplacement'] = data['engineDisplacement'].str.replace('LTR', '')
data['engineDisplacement'] = pd.to_numeric(data['engineDisplacement'], errors='coerce')
```

Drop unused columns?

['date', 'location', 'parse_date', 'link', 'name', 'vehicleConfiguration'] these columns I dropped.

4. Modeling Building

Model Selection and Why?

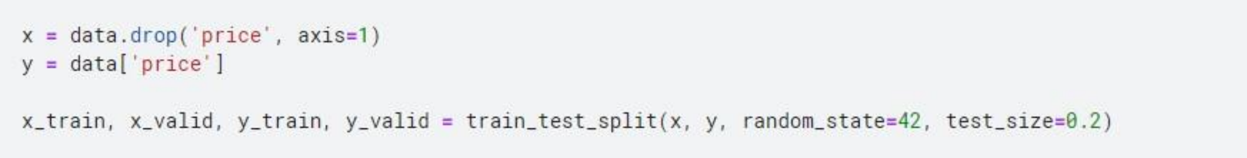
After cleaning and processing the data then comes the modeling part which includes building Machine Learning models, let’s first understand in brief what Machine Learning is?

Machine Learning is a technique that analyzes past data and tries to extract meaningful insights and patterns from them which can be further used to perform predictions in future. For example, classifying whether a tumor is benign or malignant, predicting stock prices, etc. One such application which we’re using right here is predicting house prices. Before making predictions first we need to build a model and train it using past data.

First, we need to separate the dataset into two parts: features (property attributes) and labels (prices) which is the required format for any model to be trained on.

Then the data needs to be split into 2 sets

1. Training set - This will be the part of the dataset which the model will be using to train itself, the size should be at least 80% of the total data we’ve.
2. Validation set - This set is used for validating our model’s performance for a different set of hyperparameters. After taking out the train set, the remaining set can be split into validation and test set.

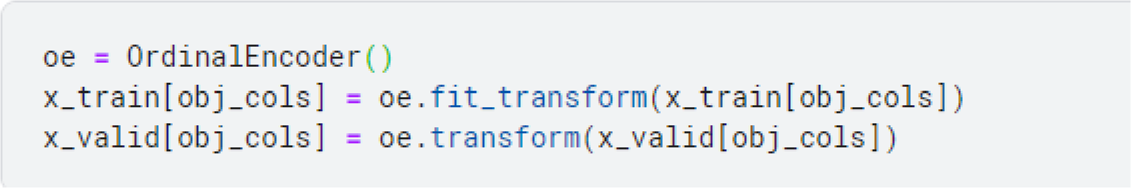


```
x = data.drop('price', axis=1)
y = data['price']

x_train, x_valid, y_train, y_valid = train_test_split(x, y, random_state=42, test_size=0.2)
```

We need to build different regression algorithms and using the validation set we can determine which model to keep for making final predictions.

Before we train we encode cat columns.



```
oe = OrdinalEncoder()
x_train[obj_cols] = oe.fit_transform(x_train[obj_cols])
x_valid[obj_cols] = oe.transform(x_valid[obj_cols])
```

Initially, we’ve tried **Linear Regression** and **RandomForestRegressor**.

We can see that the train and valid scores are 0.609 and 0.614 respectively.

Let’s look at some other algorithms, and how they are performing as compared to linear regression.

So **KNN** is giving a very good score of 0.997on the train set and 0.997 in valid soo it’s great thing to reach its accuracy.

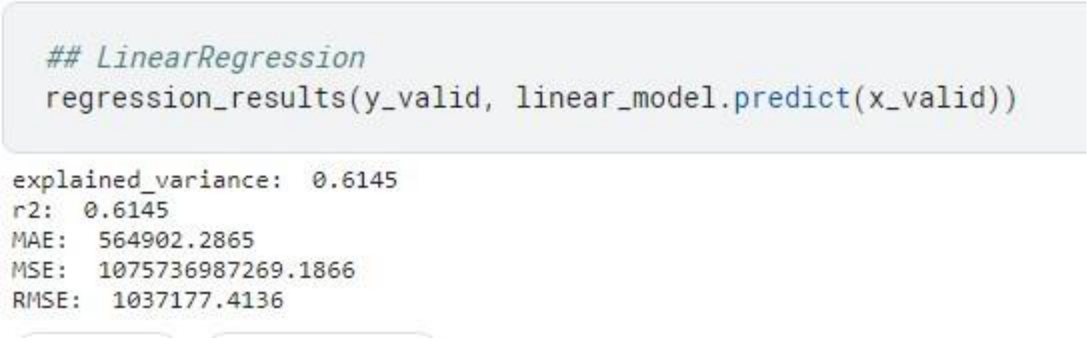
Performance Metrices

Just building is not enough, as we need to evaluate it using different metrics based on the problem we're solving. Model Validation helps us to understand how well the model is generalizing on the real-world data, the data which it has not seen during the training phase.

For regression problems the evaluation metrics we’ve used are:

- **RMSE** (Root Mean Squared Error)
- **explained_variance**
- **MSE** (Mean Squared Error)
- **MAE** (Mean Absolute Error)
- **Adjusted R^2**

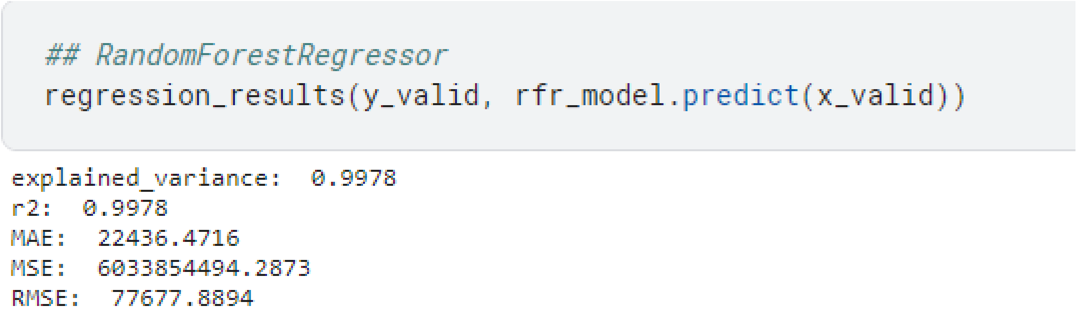
LinearRegession:



```
## LinearRegression
regression_results(y_valid, linear_model.predict(x_valid))
```

```
explained_variance:  0.6145
r2:  0.6145
MAE:  564902.2865
MSE:  1075736987269.1866
RMSE:  1037177.4136
```

RandomForestRegression



```
## RandomForestRegressor
regression_results(y_valid, rfr_model.predict(x_valid))
```

```
explained_variance:  0.9978
r2:  0.9978
MAE:  22436.4716
MSE:  6033854494.2873
RMSE:  77677.8894
```