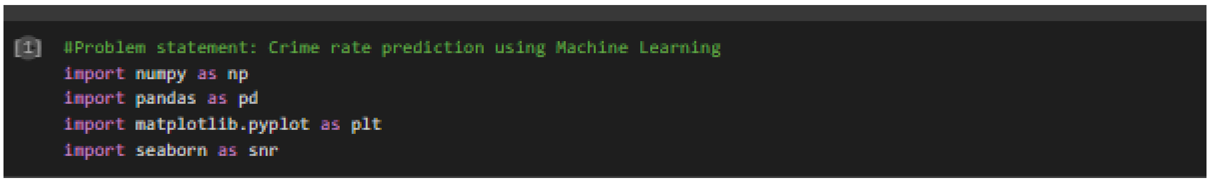
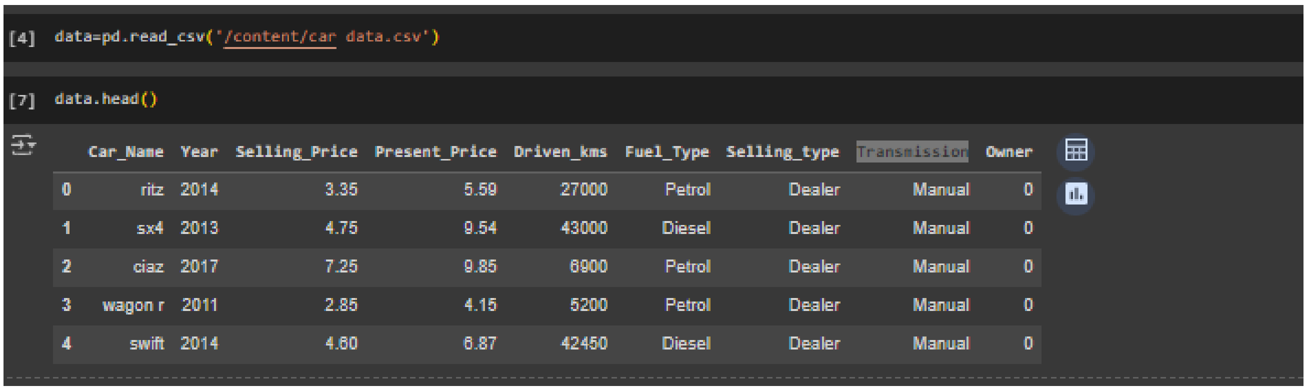
Project on Car price prediction by Machine Learning

- **Aim:-**To create a Data science Project, where we will be predicting the The price of a car depends on a lot of factors like the goodwill of the brand of the car, features of the car, horsepower and the mileage it gives and many more. Car price prediction is one of the major research areas in machine learning. So if you want to learn how to train a car price prediction model then this project is for you.
- **Steps to be taken in the project is sub-divided into the following sections. These are:**
    - Importing the libraries such as 'numpy', 'pandas','sklearn. model' etc.
    - Loading Dataset as a CSV file for training & testing the models.
    - Splitting the data set into independent & dependent sets.
    - Checking if still any null values or any other data_types other than float and integers are present into the dataset or not.
    - Importing the train_test_split model from sklearn.model for splitting data into train & test sets.
    - Applying the different kinds of ML Algorithms .which gives Best accuracy of model.
    - Also checking with new data set for predicting the values.
- **Steps of creating ML model:-**
- **Importing numpy as np & pandas as pd for loading and reading the data-set & using matplotlib.pyplot and Seaborn for visualization of data.**



```
[1] #Problem statement: Crime rate prediction using Machine Learning
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as snr
```

- **Loading the csv-dataset in the variable name 'data' Then viewing the data with data.head()**

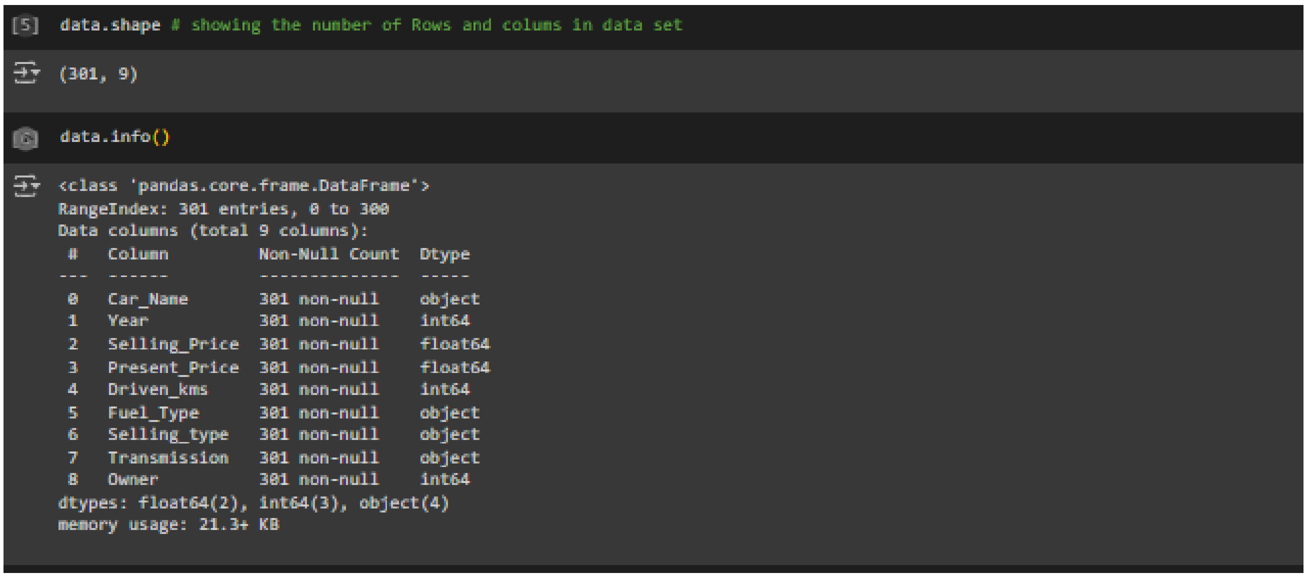


```
[4] data=pd.read_csv('/content/car data.csv')
```

```
[7] data.head()
```

	Car_Name	Year	Selling_Price	Present_Price	Driven_kms	Fuel_Type	Selling_type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

- **Checking the data such as number of columns, rows and type of data(float,integer) with help of data.info()**

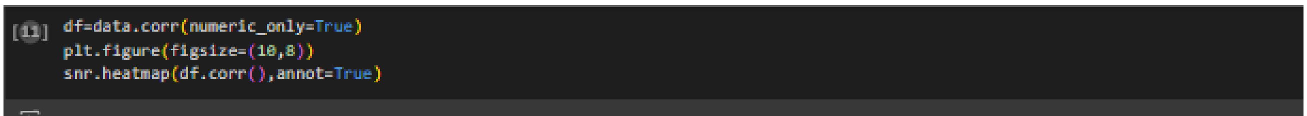


```
[5] data.shape # showing the number of Rows and colums in data set

(301, 9)
```

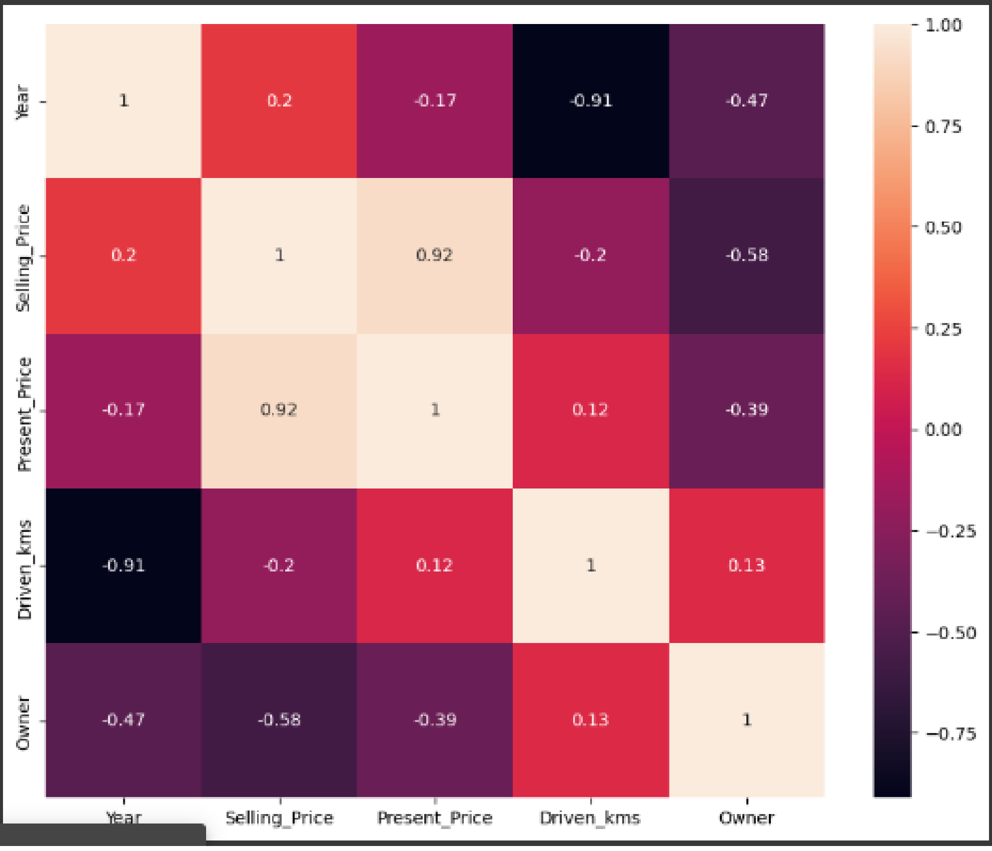
```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Car_Name        301 non-null    object
 1   Year            301 non-null    int64
 2   Selling_Price   301 non-null    float64
 3   Present_Price   301 non-null    float64
 4   Driven_kms      301 non-null    int64
 5   Fuel_Type       301 non-null    object
 6   Selling_type    301 non-null    object
 7   Transmission    301 non-null    object
 8   Owner           301 non-null    int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

We observed 301 row and 9 columns

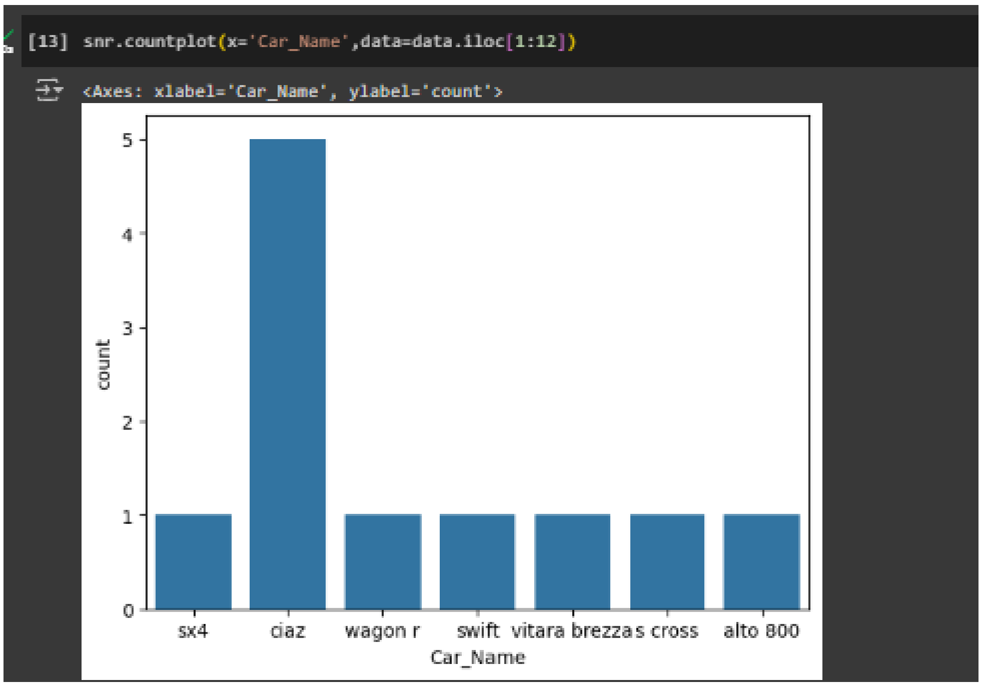
Correlation bw data by using heatmap



```
[11] df=data.corr(numeric_only=True)
     plt.figure(figsize=(10,8))
     snr.heatmap(df.corr(),annot=True)

<Axes: >
```

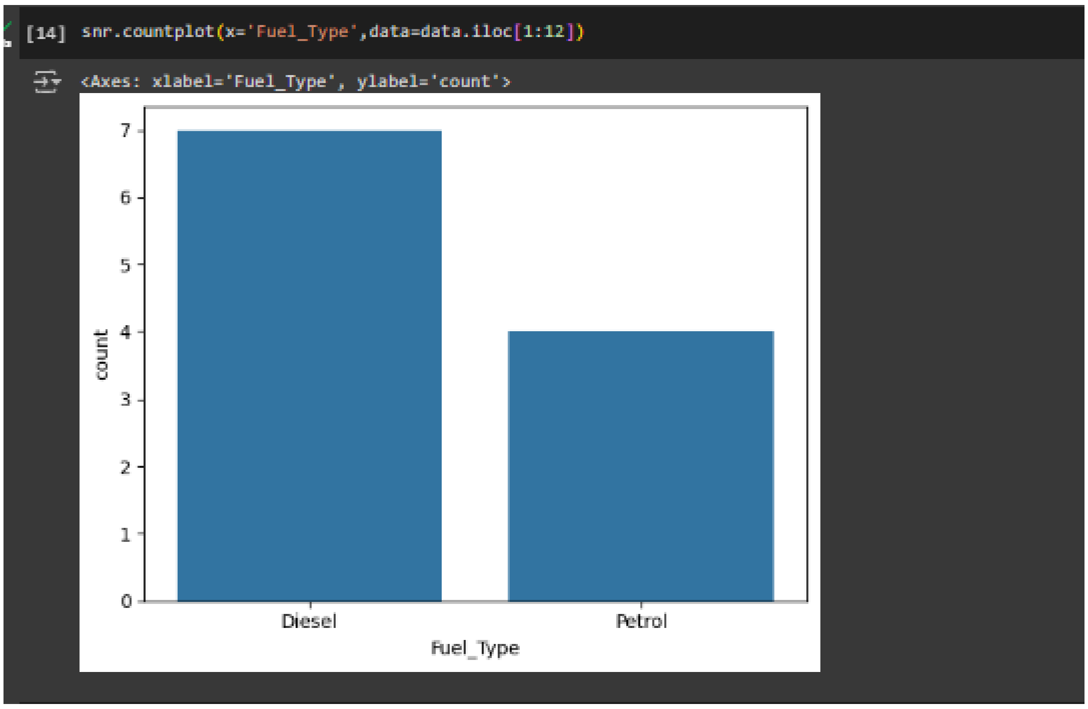


❖ **Visualising the various factor for price.**



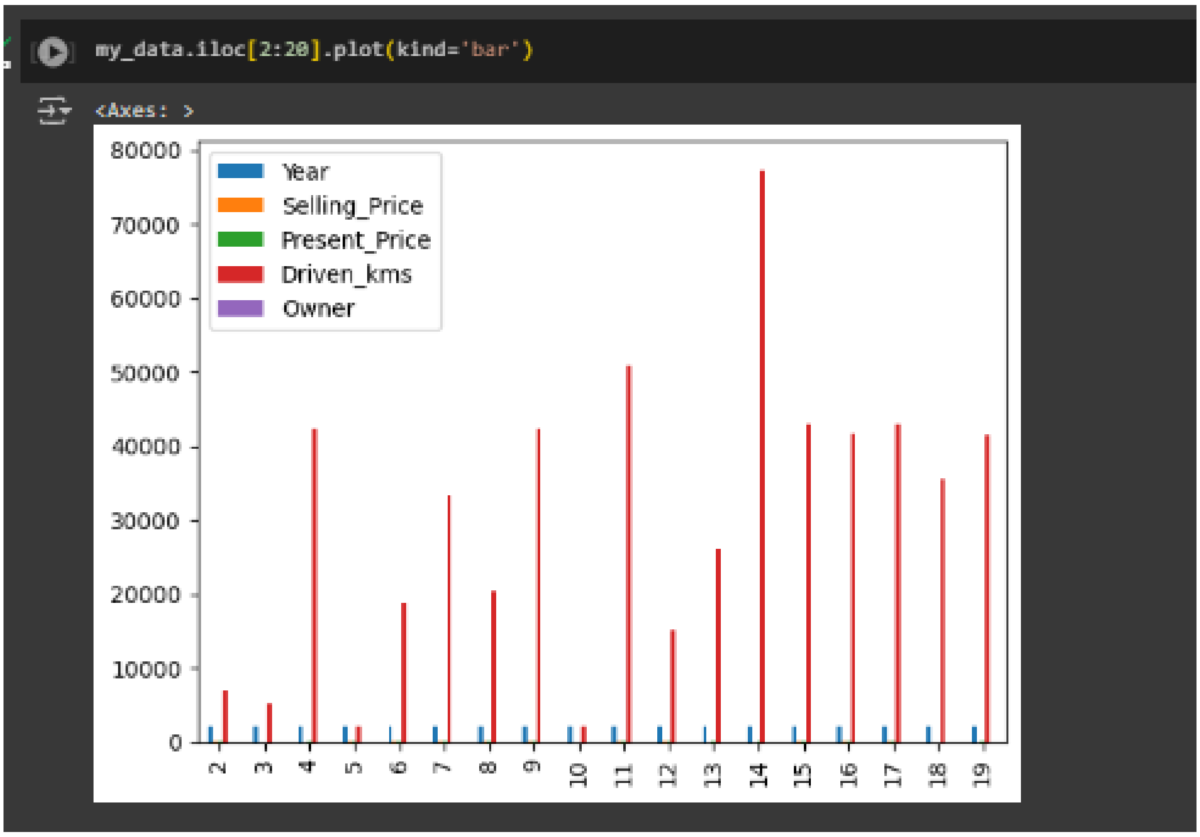
```
[13] snr.countplot(x='Car_Name',data=data.iloc[1:12])
<Axes: xlabel='Car_Name', ylabel='count'>
```

We observed the above graph ciaz car company manufacturing more than other.

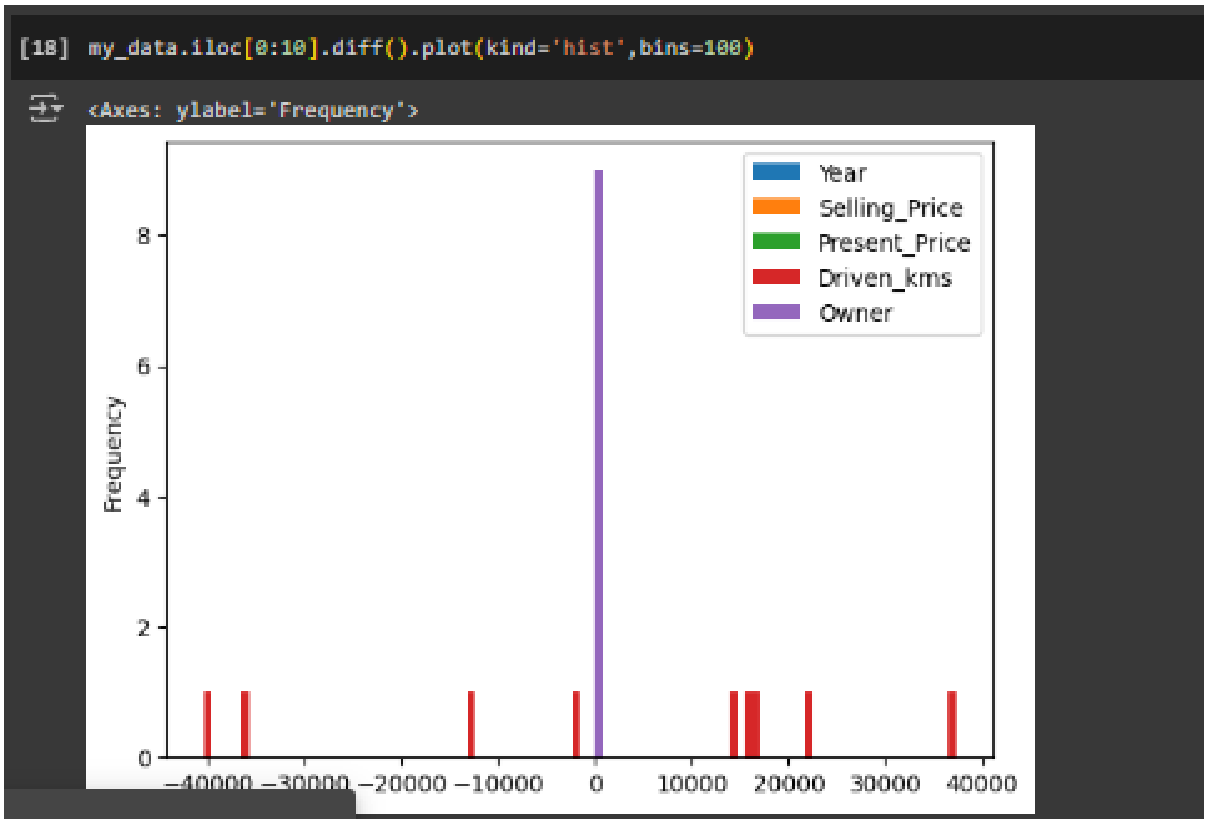


```
[14] snr.countplot(x='Fuel_Type',data=data.iloc[1:12])
<Axes: xlabel='Fuel_Type', ylabel='count'>
```

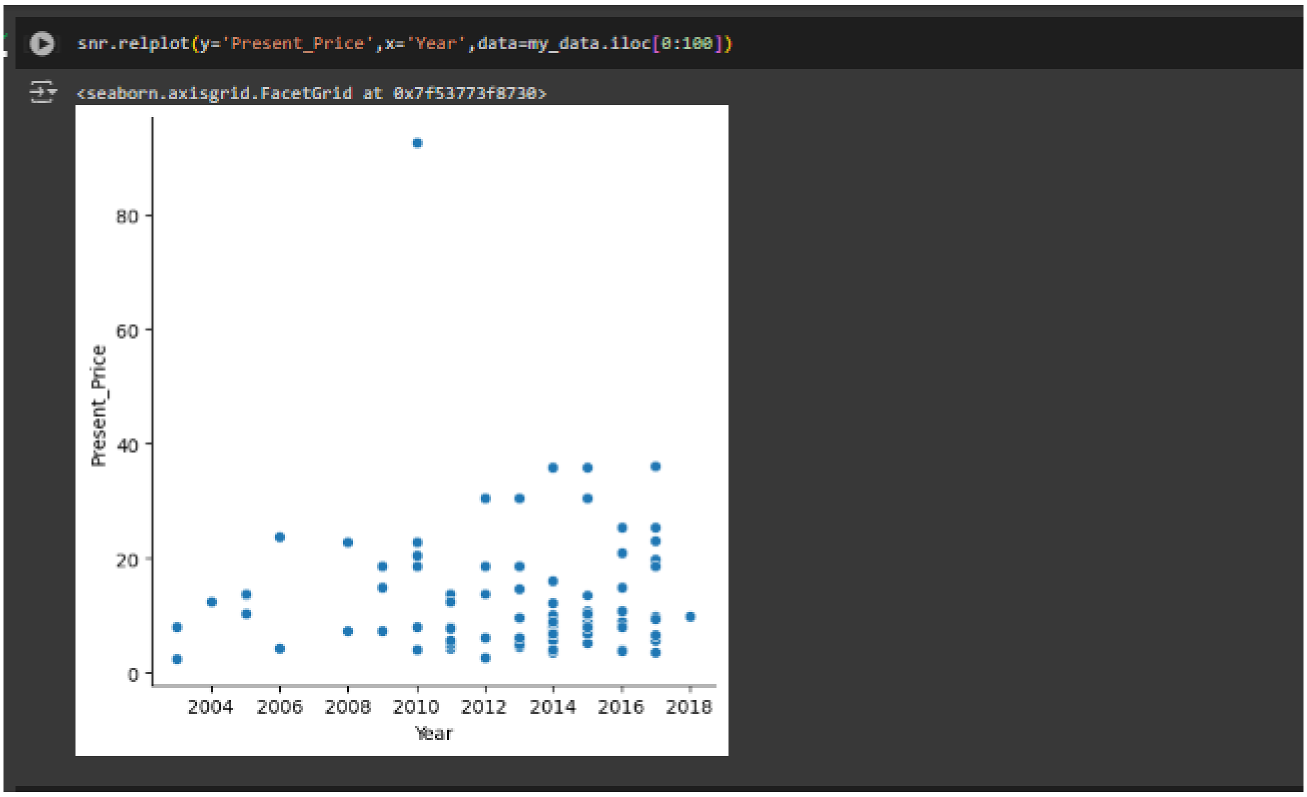
We observed the above graph diesel car manufacture more than Petrol.



```
my_data.iloc[2:20].plot(kind='bar')
<Axes: >
```



```
[18] my_data.iloc[0:10].diff().plot(kind='hist',bins=100)
<Axes: ylabel='Frequency'>
```

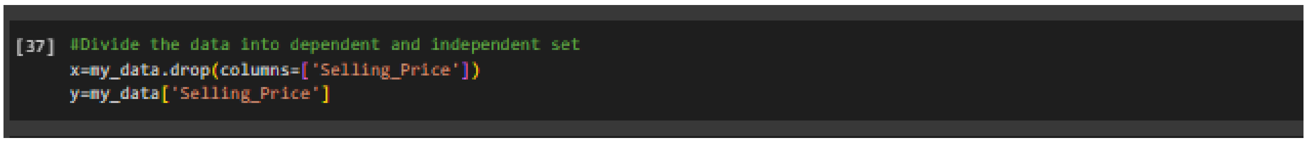


```
snr.relplot(y='Present_Price',x='Year',data=my_data.iloc[0:100])
```

```
<seaborn.axisgrid.FacetGrid at 0x7f53773f8730>
```

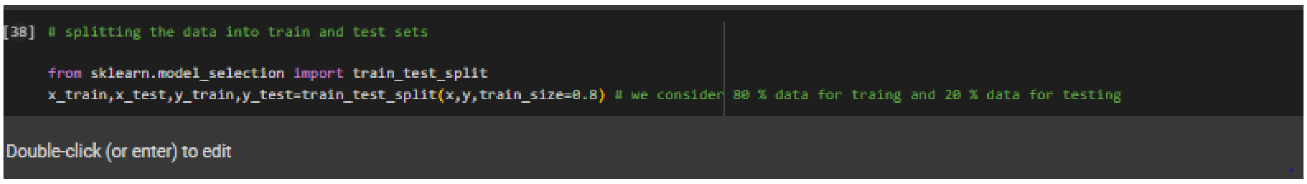
As per Visualisation the data, Car price not dependent on a particular area ,it depends on almost all feature.

- **Splitting the dataset into dependent & independent sets**



```
[37] #Divide the data into dependent and independent set
     x=my_data.drop(columns=['Selling_Price'])
     y=my_data['Selling_Price']
```

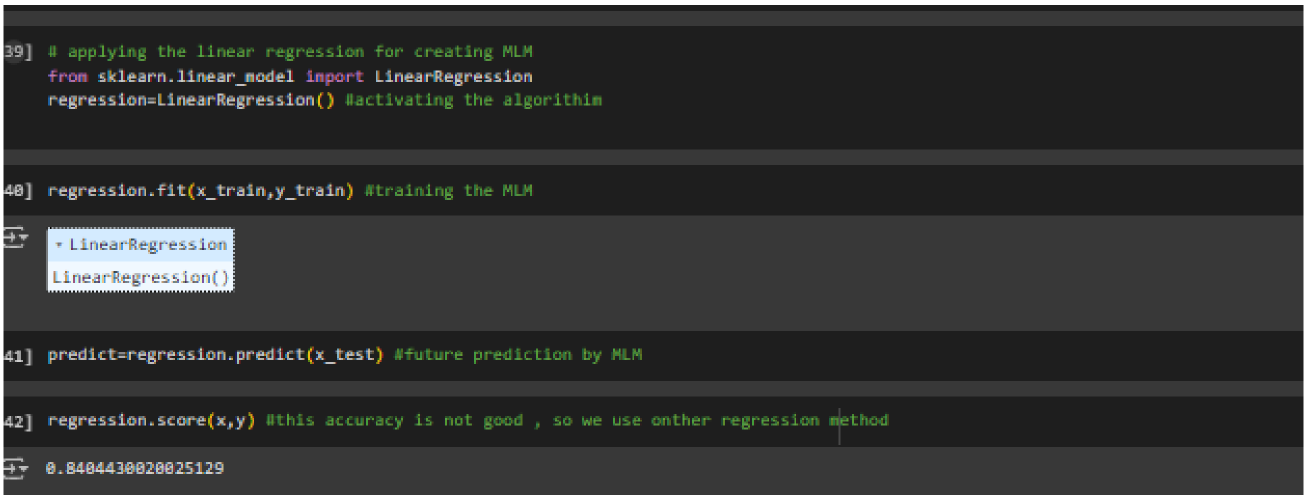
- **Importing train_test_split from sklearn.model library for splitting the data into train and test sets. (we consider train dataset).**



```
[38] # splitting the data into train and test sets

     from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.8) # we consider 80 % data for traing and 20 % data for testing
```

Double-click (or enter) to edit

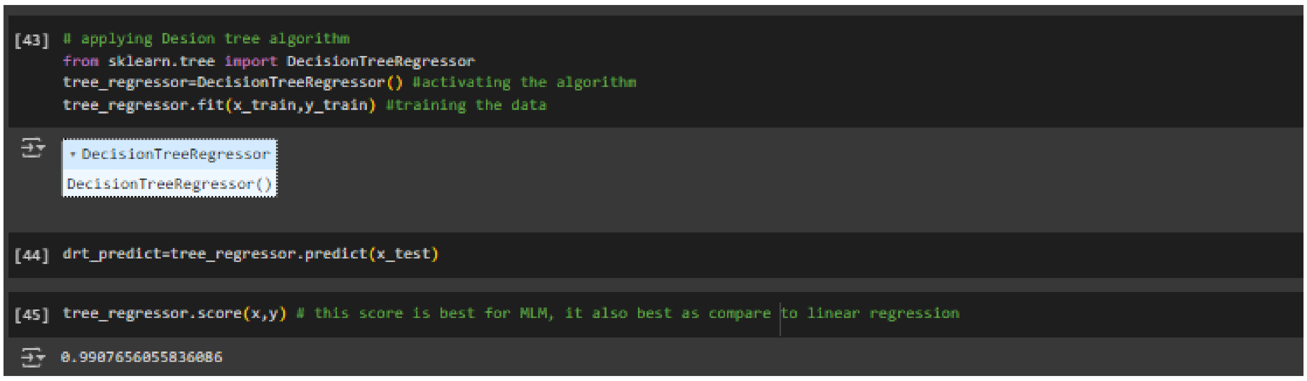
- Importing LinearRegression algorithm from sklearn Libaray & then activating the Machine learning Model .Then used regression.fit() to training the model by providing train & test sets as x & y. And then predicted the trained model with help of MLM & the checked score as regression.score(x,y)



```
39] # applying the linear regression for creating MLM
    from sklearn.linear_model import LinearRegression
    regression=LinearRegression() #activating the algorithim

40] regression.fit(x_train,y_train) #training the MLM

    ▾ LinearRegression
    LinearRegression()

41] predict=regression.predict(x_test) #future prediction by MLM

42] regression.score(x,y) #this accuracy is not good , so we use onther regression method

    0.8404430020025129
```

In the above model we can see that the accuracy obtained is 84% which is not good , we can try using different models to see if we can get better accuracy than this or not.

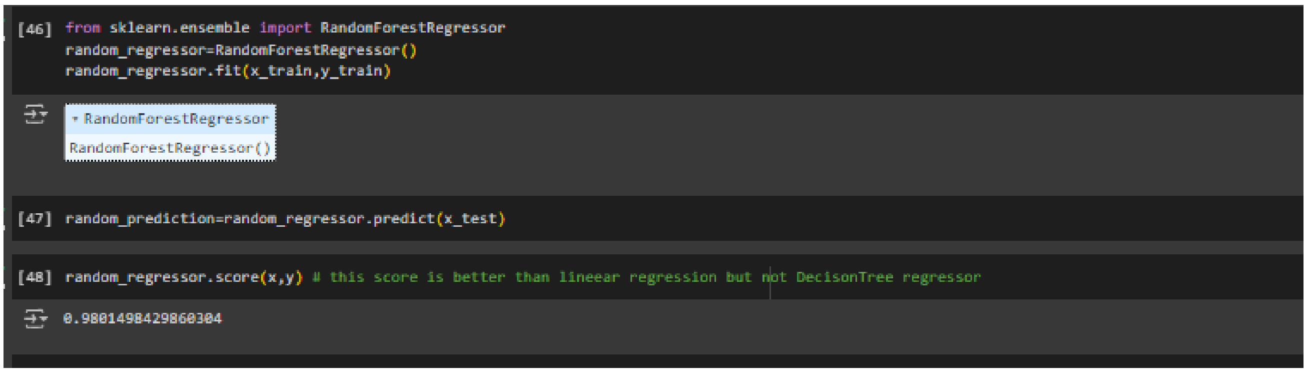
- **Now applying new algorithm DecisionTreeRegressor,then checked score.**



```
[43] # applying Desion tree algorithm
     from sklearn.tree import DecisionTreeRegressor
     tree_regressor=DecisionTreeRegressor() #activating the algorithm
     tree_regressor.fit(x_train,y_train) #training the data

     ▾ DecisionTreeRegressor
     DecisionTreeRegressor()

[44] drt_predict=tree_regressor.predict(x_test)

[45] tree_regressor.score(x,y) # this score is best for MLM, it also best as compare to linear regression

     0.9987656055836086
```

In the above model we can see that the accuracy obtained is 99% ,is more accuracy than **LinearRegressionModel** which is good .but we can try using **RandomForestRegressor** to see if we can get better accuracy than this or not.

- **Applying RandomForestRegressor algorithm**



```
[46] from sklearn.ensemble import RandomForestRegressor
     random_regressor=RandomForestRegressor()
     random_regressor.fit(x_train,y_train)
```

```
▾ RandomForestRegressor
RandomForestRegressor()
```

```
[47] random_prediction=random_regressor.predict(x_test)
```

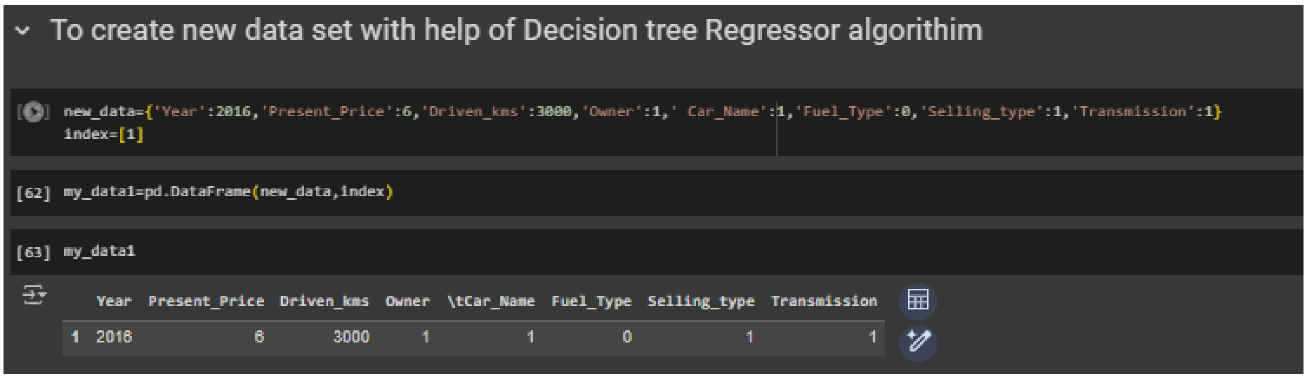
```
[48] random_regressor.score(x,y) # this score is better than lineear regression but not DecisonTree regressor
```

```
0.9801498429860304
```

In the above model we can see that the accuracy obtained 98% ,which is better than LinearRegression but less than Decision Tree Regressor.

Final conclusion ,Decision Tree Regressor(99%) is the best algorithm for this dataset.

- ➢ **Now Prediction of new dataset by using Decision Tree regressor:-**



To create new data set with help of Decision tree Regressor algorithim

```
new_data={'Year':2016,'Present_Price':6,'Driven_kms':3000,'Owner':1,' Car_Name':1,'Fuel_Type':0,'Selling_type':1,'Transmission':1}
index=[1]
```

```
[62] my_data1=pd.DataFrame(new_data,index)
```

```
[63] my_data1
```

```
   Year  Present_Price  Driven_kms  Owner  \tCar_Name  Fuel_Type  Selling_type  Transmission
1  2016              6        3000      1           1          0             1             1
```

Conclusion:- In this new model we analysed the data we found the Car price depends on fuel type,selling type,km & Transmission

Thank you