REPORT ON SALES OF AUTO  
 ----------------------------------------

Import data:-  
🡪First we have to load the autos data for analysing and visualisation

🡪For that we need to import python libraries like pandas , numpy, seaborn, mathplotlib,scipy as

Import numpy as np

Import pandas as pd

Import mathplotlib as plt

Import seaborn as sns

Import scipy as s

Import scipy.stat as ss

LOAD DATA:-

* After that load the data from the excel or csv file by copying the path as copypath
* But this data has a typical encoding error so the data should be in UTF-8 format and saved and while loading the data we have to load the data as

Data=pd.read\_csv(r”copypath”,encoding=utf-8)

* Now to make the data not to be effected by the analysis changes so we have to take a copy of the data as

Datac=data.copy()

ANALYSIS-1

1. Perform general Data analysis

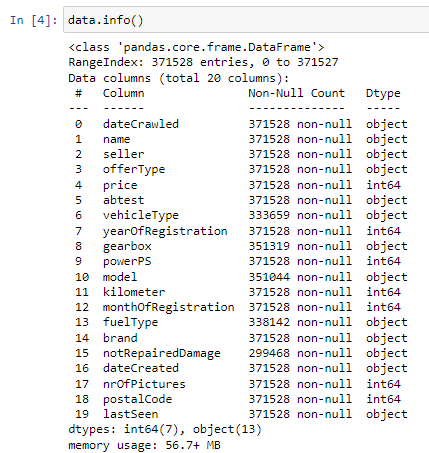
To perform general data we have to find the general information about the data like

Number of columns in the data   
Number of rows in the data  
Number of non null values in every column in the data  
Number of duplicate rows in the data  
what are the data types of each column in the data

Theese are the basic data analysis should be done on the data as follows

To know the basic iformation about the data we use the code as datac.info()

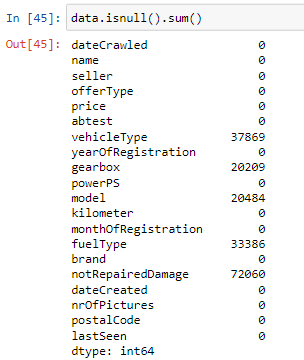
Which will gives u the all values of data columns,rows,datatype of columns,non-null values in the column size of the data as follows



🡪Here we can confirm that there are total count of 18 columns in the autos sales dataset

🡪And there are 371528 rows in the dataset also there are some null values in some columns which are

FINDING NULL VALUES:-



So for we have 37869 null values in vehicle type column

And 20209 null values in gearbox column ,20484 null values in model column

33386 null values in fueltype column and 72060 null values in notrepaired damage column

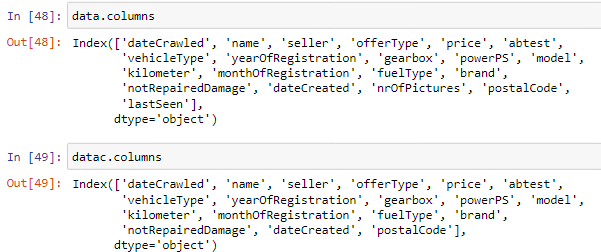
🡪here all the null valued columns have same object-datatype  
REMOVING UNNECESSARY COLUMNS:-

For our data we have unnecessary columns which we can remove by using drop method as

datac.drop(["nrOfPictures"],axis=1,inplace=True)

which can remove the column in the data permanently in the dataset.

🡪The result is “nrofpictures” column has removed permanantly from datac. If we observe both the cells of our outputs from the below picture

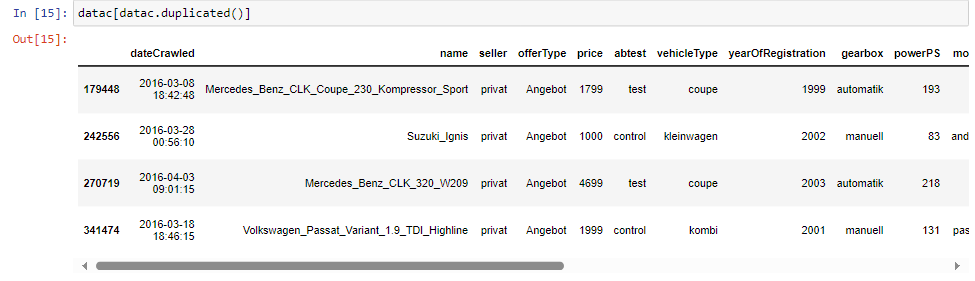


REMOVING DUPLICATED ROWS:-

* If we have duplicated rows we can find them by using the code as

datac.duplicated()

it will gives you the result as



i.e., we have four(4) no of duplicated rows present In the datac set

🡪 to remove these duplicated rows in the data use the code as

datac.drop\_duplicates(inplace=True)

* If we check the data these duplicated 4 rows will be removed permanently from the datac

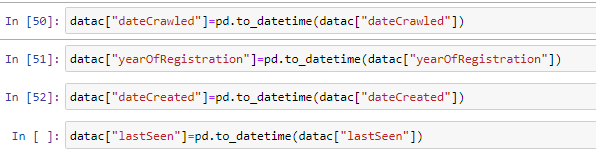
REMOVING OR REPLACING NULL VALUES:-  
From the above data all the null values are from object type so all are categorical columns for the categorical columns if the data contain less than 50% of its datapoint we can replace the null values with the mode of the column here all the columns are having null values very less than the 50% of the data of that particular column .So we can replace the data with its mode using the code as

datac.fillna({"notRepairedDamage":datac["notRepairedDamage"].mode()[0],"fuelType":datac["fuelType"].mode()[0],"vehicleType":datac["vehicleType"].mode()[0],"gearbox":datac["gearbox"].mode()[0],"model":datac["model"].mode()[0]},inplace=True)

it will replace the null values with their respective column modes.

CHANGING DATA TYPE:-

🡪Here we obtain some of the columns are in object type instead of being its original type.so we can change their data type with their corresponding type as



Here all the columns are changed to their corresponding data type

----------->Finally all the basic data analysis have been done!<-------------------

2)Can you tell me the Distribution of Vehicles based on Year of Registration with the help of a plot

🡪 To represent any distribution of any plot we can use histogram or bar plot

Because the smoothened curve on the plots can represents the perfect distribution of that plots

* We have to show how many no of vehicles have registered at that particular year so first we have to extract the data of registered year column and fit in a plot
* But the problem here is there are many years out of bound
* So we have to handle this by ranging the data to a certain value

And replacing the other bounded data to to the mode of its data values in the data by using the code

datac["yearOfRegistration"] = datac["yearOfRegistration"].where((datac["yearOfRegistration"] >= 1950) & (datac["yearOfRegistration"] <= 2023),datac["yearOfRegistration"].mode()[0])

here we replaced the data of the column name year of registration which is not lying between 1950 and 2023 will be replaced with the mode of the data

using the where function with applying a condition

* Hence the data is converted to the particular minor ranged data
* To plot the data we use the code as

plt.figure(figsize=(20, 10))

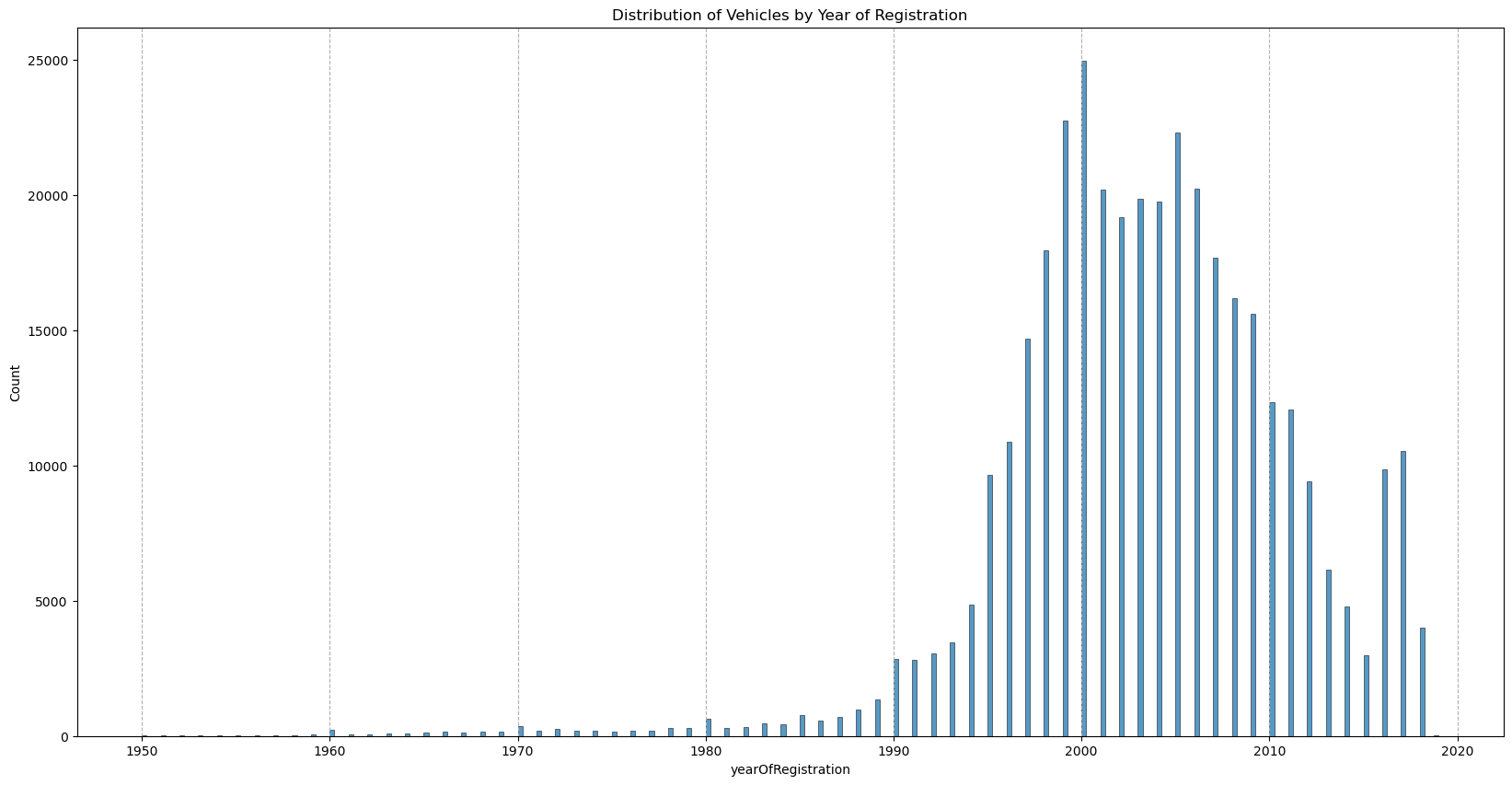
sns.histplot(data=datac, x=datac["yearOfRegistration"])

plt.title("Distribution of Vehicles by Year of Registration")

plt.grid(axis='x', linestyle='--')

plt.show()

🡪using this code the mathplotlib plots the distribution of the yearofregistration as



This image represents that the distribution of vehicles by the year of registration

At x-axis we consider the year of registration and on y-axis we consider the count of vehicles at that particular year on bar graph

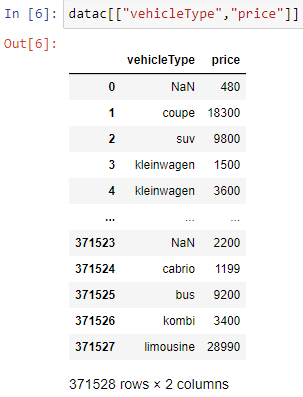
From this distribution we can conclude that a max no of vehicles are registered between the years 1980-2018

And there are very less count of vehicles which are registered on that particular years

1. Create a plot based on the Variation of the price range by the vehicle type

🡪 To create a plot variation of price differ from one vehicle type to another vehicle we can select the bar plot

* Because in bar plot we can differentiate the count difference from one vehicle to another vehicle type
* Before that we should take the consideration based on vehicle type and price by using the code



Here both the columns are considered from the data to fit in the bar plot

* To fit the bar plot we use the code as

sns.barplot(datac,x=datac["vehicleType"],y=datac["price"])

plt.title("Variation of the price range by the vehicle type")

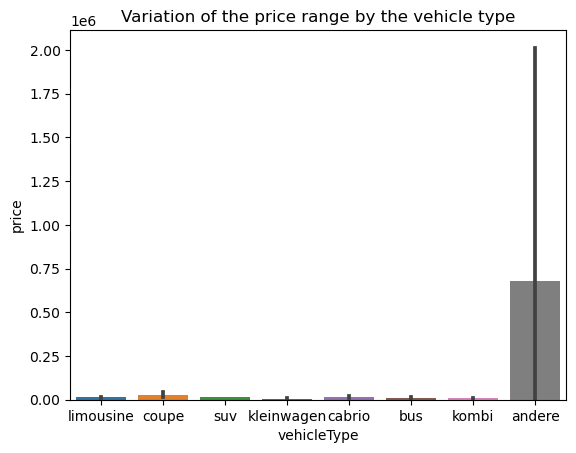
plt.xlabel("vehicleType")

plt.ylabel("price")

plt.show()

which represent the bar plot of vehicletype as x-axis and price on y-axis

as shown below



From the figure we can say that the “andere” vehicle have the highest price of $0.75 \*10^6 and all other vehicleare veary least price in the range of $0.12\*10^6-$0.0001\*10^6

And kleinwagen vehicle have the least cost price among all the vehicletypes

1. Find out Total count of vehicles by type available on ebay for sale.As well as create a visualization for the client

To create a plot on the count of vehicles by its type we can use the count plot for the data

* Because in count plot we can get the count frequency on y-axis and vehicle type on x-axis
* Before that we should take the consideration based on vehicle type by using the code
* Datac[“vehicletype”]
* To plot the data of the count of vehicle type we should use the code as below

datac["vehicleType"].value\_counts()

sns.countplot(x=datac["vehicleType"])

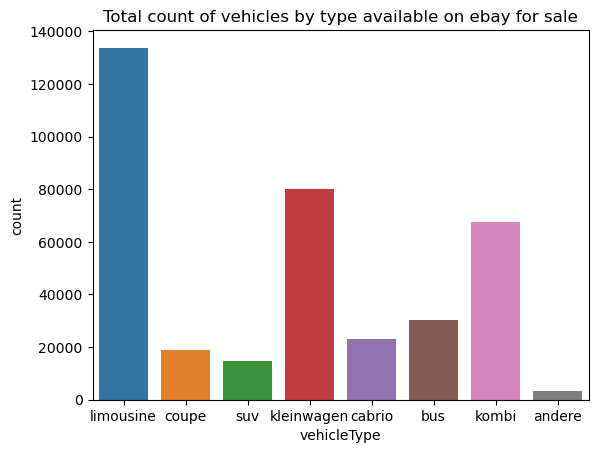
plt.title("Total count of vehicles by type available on ebay for sale")

plt.xlabel("vehicleType")

plt.ylabel("count")

plt.show()

will gives the bar plot about the count of each vehicle on showing the count of values in it



From the figure we can define the total count of vehicles by its type which are available on ebay

From the figure we can say that limousine have the highest count of 138900 cars saled

Also andere car has least count osf saled which is around 5000

5)Is there any relationship between dollar\_price and kilometer? (Explain with appropriate analysis)

Yes, there a relation between the column price and kilometre which can be explained as below

To know the relation ship between the columns.

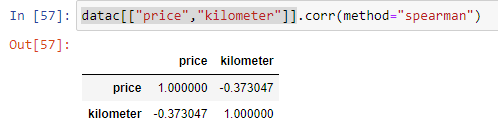
In the statistics there is a concept called correlation ,here we can find the relation between two or moer columns

But to know the relation ship between any two columns we have to access the columns as



Hence the two columns are accessed from the data

Later we cue the correlation concept in he code as



To represent this information in a plot we can use the heat map which can shows your result in a better effective and understanding

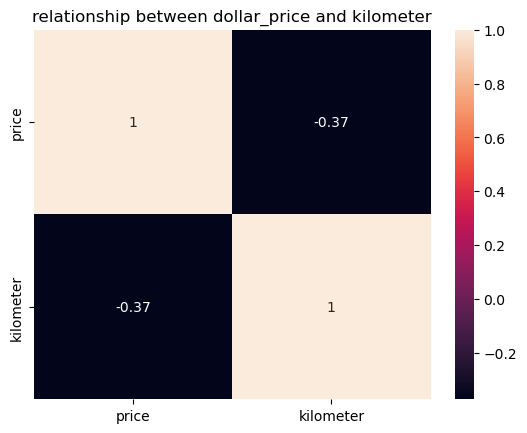
For this we have to use the code as

sns.heatmap(datac[["price","kilometer"]].corr(method="spearman"),annot=True)

plt.title("relationship between dollar\_price and kilometer")

plt.show()

this code will gives you the plot as



From the figure we can define that price and kilometre are negatively moderately correlated

And the correlation between the price column and kilometre is -0.37

Why because the correlation always between -1&1

If the correlation lies between 0 to 0.5 we can say that moderately positively correlated

If the correlation lies between 0 to -0.5 we can say that moderately negatively correlated

If the correlation lies between 0.5-1 we can say that highly positively correlated

If the correlation lies between -0.5 to -1 we can say that highly negatively correlated

From our figure we can say there is an inversely proportional relation between price and kilometre due to its negative correlation