PROJECT DEVELOPMENT PHASE

DELIVERY OF SPRINT-1

DATE	29 OCTOBER 2022
TEAM ID	PNT2022TMID15627
PROJECT NAME	Car Resale Value Prediction
MAXIMUM MARK	4 Marks

Import library and load dataset in python

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn. preprocessing import LabelEncoder
import pickle
```

In [2]:
```
df = pd.read_csv("D:\IBM Dataset/autos.csv",header=0,sep=',',encoding='Latin1',
```

Cleaning the Dataset

In [3]:
```
df[df.seller != 'gewerblich']
df=df.drop( 'seller', 1)
df[df.offerType != 'Gesuch']
df=df.drop( 'offerType', 1)
```

```
C:\Users\arun\AppData\Local\Temp/ipykernel_10288/858333111.py:2: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'lab
els' will be keyword-only
  df=df.drop( 'seller', 1)
C:\Users\arun\AppData\Local\Temp/ipykernel_10288/858333111.py:4: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'lab
els' will be keyword-only
  df=df.drop( 'offerType', 1)
```

In [4]:
```
df = df[ (df.powerPS > 50) & (df.powerPS < 900) ]
df = df[ (df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)]
```

In [5]:
```
df.drop(['name','abtest','dateCrawled','nrOfPictures','lastSeen','postalCode','dateCreated'],axis='columns',inplace=True)
```

In [6]:
```
new_df = df.copy()
new_df = new_df.drop_duplicates(['price','vehicleType','yearOfRegistration','gearbox','powerPS','model','kilometer','monthOfRegistration','fuelType','notRepairedDamage']
```

In [7]:
```
new_df.gearbox.replace(('manuell','automatik'),('manual','automatic'),inplace=True)
new_df.fuelType.replace(('benzin','andere','elektro'),('petrol','others','electirc'),inplace=True)
new_df.vehicleType.replace(('kleinwagen','cabrio','kombi','andere'),('small car','convertible','combination','others'),inplace=True)
new_df.notRepairedDamage.replace(('ja','nein'),('Yes','No'),inplace=True)
```

Removing the outliers and filling NaN values using the fillna() function

In [8]:
```
new_df = new_df[(new_df.price >= 100) & (new_df.price <= 150000)]
```

In [9]:
```
new_df['notRepairedDamage'].fillna(value='not-declared',inplace=True)
new_df['fuelType'].fillna(value='not-declared',inplace=True)
new_df['gearbox'].fillna(value='not-declared',inplace=True)
new_df['vehicleType'].fillna(value='not-declared',inplace=True)
new_df['model'].fillna(value='not-declared',inplace=True)
```

In [23]:
```
new_df.to_csv("autos_preprocessed.csv")
```