Date	7 November 2022
Team ID	PNT2022TMID01545
Project Name	PROJECT-CAR RESALES VALUE PREDICTION
Maximum Marks	2 Marks

Collect dataset:

Machine Learning has become a tool used in almost every task that requires estimation. So we need to build a model to estimate the price of used cars. The model should take car-related parameters and output a selling price. On sprint-1 the selling price of a used car depends on certain features datasets are collected from different open sources like kaggle.com, data.gov, UCI machine learning repository, the dataset which contains a set of features through which the resale price of the car can be identified is to be collected as

- seller
- offerType
- price
- vehicleType
- yearOfRegistration
- gearbox
- powerPS
- model
- kilometer
- monthOfRegistration
- fuelType
- brand
- notRepairedDamage

ML is a data hunger technology, it depends heavily on data, without data, it is impossible. It is the most crucial aspect that makes algorithm training possible. Collects Data, Import necessary packages, Pre-process images, and passes on to Network Model and Saves Model Weights. The libraries can be imported,

import pandas as pd import numpy as mp import matplotlib a from sklearn.prepro import pickle																			
<pre>df = pd.read_csv("/ df.head()</pre>	content/drive/MyDrive/Colab Not	rtebooks/s	eutos.csv")																
dateCrawled	name	seller	offerType	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage	dateCreated	nrOfPictures	postalCode	lastSeen
dateCrawled 0 24-03-2016 11:52	name Golf_3_1.6		offerType Angebot	price 480	abtest test	vehicleType NaN	yearOfRegistration	gearbox manuell	powerPS 0		kilometer 150000.00	monthOfRegistration 0		brand volkswapen	- 2	dateCreated 24-03-2016 00:00			lastSeen 07-04-2016 03:16
		privat		480					powerPS 0 190	golf		monthOfRegistration 0 5		volkswagen	NaN		0.00	70435	
0 24-03-2016 11:52 1 24-03-2016 10:58	Gott_3_1.6 A5_Sportback_2.7_Tdi	privat privat	Angebot	480 18300	test	NaN	1993 2011	manuell	0 190	golf NaN	150000.00	monthOfRegistration 0 5	benzin	volkswagen audi	NaN ja	24-03-2016 00:00	0.00	70435 66954	07-04-2016 03:16
0 24-03-2016 11:52 1 24-03-2016 10:58	Golf_3_1.6 A5_Sportback_2.7_Tdi	privat privat privat	Angebot Angebot	480 18300	test test	NaN	1993 2011	manuell manuell	0 190	golf NaN grand	150000.00 125000.00	monthOfRegistration 0 5 8	benzin diesel	volkswagen audi jeep	NaN ja NaN	24-03-2016 00:00 24-03-2016 00:00	0.00 0.00 0.00	70435 66954 90480	07-04-2016 03:16 07-04-2016 01:46

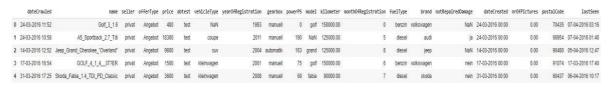
Pre-Process The Data:

Pre-processing the dataset that includes:

- Handling the null values.
- Handling the categorical values if any.
- Normalize the data if required.
- Identify the dependent and independent variables.

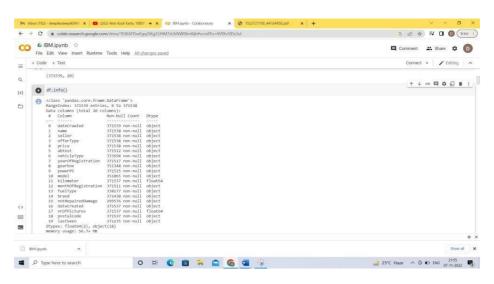
Data cleaning and wrangling methods are applied on the *used cars* data file. Before making data cleaning, some explorations and data visualizations were applied on data set. This gave some idea and guide about how to deal with missing values and extreme values. After data cleaning, data exploration was applied again in order to understand cleaned version of the data.

df = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/autos.csv")
df.head()



print(df.shape)
(371539, 20)

df.info()



```
df['powerPS'].unique()
                          epikadeepik09 🌣 | 🚾 (262) Vetri Kodi Kattu 1080 + 🖈 🖒 IBM.ipynb - Colaboratory 🗴 🔇 1523727100_441IJARSE.pdf 🗴 | 🛨
  \leftarrow \  \  \, \rightarrow \  \  \, \mathbf{C} \quad \text{(a colab.research.google.com/drive/1EtKAFl5wEgq2tKg2UHM7sUkNWIXml6jk#scrollTo=xm9jX4nhtMjU}
                                                                                                                                                                                                                                                  G 🖻 🖈 🗊 🔲 📵 Error
  CO & IBM.ipynb 🌣
                                                                                                                                                                                                                                                   □ Comment 🎎 Share 🌣 📵
             File Edit View Insert Runtime Tools Help All changes saved
                                                                                                                                                                                                                                                            Connect - / Editing ^
          + Code + Text
                  df[powerPs].unique()

array(['0', '190', '163', '75', '69', '102', '109', '50', '125', '101', '105', '140', '115', '131', '60', '136', '160', '231', '90', '118', '193', '99', '113', '218', '122', '129', '70', '306', '95', '61', '177', '80', '170', '55', '143', '64', '286', '232', '150', '156', '82', '204', '155', '54', '185', '87', '180', '86', '84', '303', '224', '235', '200', '178', '265', '77', '110', '144', '120', '164', '164', '165', '88', '194', '305', '197', '179', '250', '45', '313', '41', '165', '98', '130', '114', '211', '56', '326', '201', '213', '58', '167', '300', '103', '45', '258', '120', '68', '66', '299', '74', '52', '510', '147', '65', '310', '71', '97', '239', '295', '233', '5', '300', '103', '245', '258', '292', '320', '63', '81', '148', '354', '44', '145', '230', '288', '266', '457', '104', '409', '188', '333', '186', '117', '141', '55', '26', '27', '192', '22', '138', '35', '457', '140', '409', '188', '333', '186', '117', '141', '55', '26', '27', '30', '30', '30', '313', '124', '313', '124', '313', '219', '241', '167', '94', '182', '166', '288', '277', '384', '288', '272', '219', '22', '133', '121', '388', '36', '344', '128', '133', '124', '133', '124', '133', '124', '134', '129', '241', '166', '288', '221', '133', '124', '138', '36', '27', '38', '35', '38', '152', '396', '348', '108', '121', '388', '365', '175', '96', '226', '27', '139', '121', '388', '36', '155', '96', '226', '27', '139', '121', '388', '36', '152', '396', '381', '125', '396', '348', '108', '121', '388', '365', '175', '36', '212', '368', '366', '171', '38', '38', '152', '38', '38', '152', '396', '328', '171', '38', '38', '351', '38', '351', '35', '36', '38', '351', '35', '36', '361', '318', '361', '38', '351', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '37', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38', '38',
            odf['powerPS'].unique()
                                                                                                                                                                                                                                                             ↑ ↓ © □ ‡ 🖟 🗎 :
 Q
 {x}
  ()
  [] IBM.ipynb
                                                                                                                                                                                                                Type here to search
                                                                            df=df[df['powerPS'].str.isnumeric().fillna(False)]
print(df.seller.value counts())
df[df.seller != 'gewerblich']
print(df.offerType.value counts())
df[df.offerType != 'Gesuch']
M Inbox (782) - deepikadeepik09® x | 10 (262) Muthu Movie Songs | 4 x CO | BM.ipynb - Colaboratory x 🚷 1523727100_441IJARSE.pdf x +
                                                                                                                                                                                                                                        G Q 🖻 ☆ 🗐 🔲 📵 (Error 🚦
  ← → C  a colab.research.google.com/drive/1EtKAFI5wEgq2tKg2UHM7sUkNWIXml6jk#scrollTo=2tCAaWQlsuCt
                                                                                                                                                                                                                                       O HI C B 7 C 6 W 9
df['powerPS']=df['powerPS'].astype(int)
df=df[(df.powerPS > 50) & (df.powerPS < 900)]</pre>
print(df.shape)
df=df[df['yearOfRegistration'].str.isnumeric().fillna(False)]
```

```
df=df[(df.yearOfRegistration > 1950) & (df.yearOfRegistration < 2017)]</pre>
print (df.shape)
df.drop(['name', 'abtest', 'dateCrawled', 'nrOfPictures', 'lastSeen', '
postalCode', 'dateCreated'], axis='columns', inplace=True)
df.info()
  <class 'pandas.core.frame.DataFrame'>
  Int64Index: 308923 entries, 1 to 371538
  Data columns (total 13 columns):
       Column
                             Non-Null Count
                                               Dtype
                             _____
  seller
   0
                             308923 non-null
                                               object
     offerType
                             308923 non-null
                                               object
   1
                             308923 non-null
                                               object
   2
      price
   3
      vehicleType
                             297510 non-null
                                               object
                             308923 non-null int64
   4
      yearOfRegistration
   5
      gearbox
                             303629 non-null
                                               object
   6
      powerPS
                             308923 non-null int64
   7
      model
                             297134 non-null
                                               object
     kilometer
                             308923 non-null float64
     monthOfRegistration 308923 non-null
                                               object
   10 fuelType
                             293046 non-null
                                               object
   11 brand
                             308923 non-null
                                               object
   12 notRepairedDamage
                             265507 non-null
                                               object
  dtypes: float64(1), int64(2), object(10)
  memory usage: 33.0+ MB
new df=df.copy()
new df = new df.drop duplicates(['price', 'vehicleType', 'yearOfRegistr
'gearbox', 'powerPS', 'model', 'kilometer', 'monthOfRegistration', 'fue
lType',
'notRepairedDamage'])
new df.gearbox.replace(('manuell', 'automatik'), ('manual', 'automatic'
), inplace=True)
new df.fuelType.replace(('benzin', 'andere', 'elektro'), ('petrol', 'ot
hers', 'electric'), inplace=True)
new df.notRepairedDamage.replace(('ja', 'nein'),('Yes', 'No'), inplace=
True)
new df.vehicleType.replace(('kleinwagen', 'cabrio', 'kombi', 'andere'),
 ('small car', 'convertible', 'combination', 'others'), inplace=True)
```

df['yearOfRegistration']=df['yearOfRegistration'].astype(int)

```
new df['price'].unique()
array(['18300', '9800', '1500', ..., '18429', '24895', '10985'],
      dtype=object)
new df['price'] = new df['price'].astype(int)
new df = new df[(new df.price \geq 100) & (new df.price \leq 150000)]
new df['fuelType'].fillna (value='not-declared', inplace=True)
new df['gearbox'].fillna (value='not-declared', inplace=True)
new df['notRepairedDamage'].fillna (value='not-declared', inplace=True)
new df[ 'vehicleType'].fillna (value='not-declared', inplace=True)
new df['model'].fillna (value='not-declared', inplace=True)
new df['kilometer']=new df['kilometer'].astype(int)
new df.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 278363 entries, 1 to 371538
Data columns (total 13 columns):
     Column
                           Non-Null Count
                                            Dtype
     _____
                            _____
 0 seller
                            278363 non-null object
 1 offerType
                           278363 non-null object
                           278363 non-null int64
 2
     price
                            278363 non-null object
 3
     vehicleType
 4 yearOfRegistration 278363 non-null int64
                            278363 non-null object
 5
     gearbox
                            278363 non-null int64
 6
     powerPS
 7
     model
                            278363 non-null object
     kilometer
                            278363 non-null int64
    monthOfRegistration 278363 non-null object
 10 fuelType
                            278363 non-null object
                            278363 non-null object
 11 brand
 12 notRepairedDamage
                            278363 non-null object
dtypes: int64(4), object(9)
memory usage: 29.7+ MB
```

new df['price'].unique()

new_df.head() new_df.head()

	seller	offerType	price	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage
1	privat	Angebot	18300	coupe	2011	manual	190	not-declared	125000	5	diesel	audi	Yes
2	privat	Angebot	9800	suv	2004	automatic	163	grand	125000	8	diesel	jeep	not-declared
3	privat	Angebot	1500	small car	2001	manual	75	golf	150000	6	petrol	volkswagen	No
4	privat	Angebot	3600	small car	2008	manual	69	fabia	90000	7	diesel	skoda	No
5	privat	Angebot	650	limousine	1995	manual	102	3er	150000	10	petrol	bmw	Yes