

<b>Date</b>	<b>7 November 2022</b>
<b>Team ID</b>	<b>PNT2022TMID01716</b>
<b>Project Name</b>	<b>PROJECT-CAR RESALES VALUE PREDICTION</b>
<b>Maximum Marks</b>	<b>2 Marks</b>

## 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 np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
import pickle

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

	dateCrawled	name	seller	offerType	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage	dateCreated	nroPictures	postalCode	lastSeen
0	24-03-2016 11:52	Golf_3_1.6	privat	Angebot	480	test	NaN	1993	manuel	0	golf	150000.00	0	benzin	volkswagen	NaN	24-03-2016 00:00	0.00	70435	07-04-2016 03:16
1	24-03-2016 10:58	A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	2011	manuel	190	NaN	125000.00	5	diesel	audi	ja	24-03-2016 00:00	0.00	66954	07-04-2016 01:46
2	14-03-2016 12:52	Jeep_Grand_Cherokee_Overland	privat	Angebot	9800	test	suv	2004	automatik	163	grand	125000.00	8	diesel	jeep	NaN	14-03-2016 00:00	0.00	90480	05-04-2016 12:47
3	17-03-2016 16:54	GOLF_4_1.4_317ER	privat	Angebot	1500	test	kleinwagen	2001	manuel	75	golf	150000.00	6	benzin	volkswagen	nein	17-03-2016 00:00	0.00	91074	17-03-2016 17:40
4	31-03-2016 17:25	Skoda_Fabia_1.4_TDI_PD_Classic	privat	Angebot	3600	test	kleinwagen	2008	manuel	69	fabia	90000.00	7	diesel	skoda	nein	31-03-2016 00:00	0.00	60437	06-04-2016 18:17

## 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()
```

	dateCrawled	name	seller	offerType	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage	dateCreated	nrOfPictures	postalCode	lastSeen
0	24-03-2016 11:52	Golf_3_1.6	privat	Angebot	480	test	NaN	1993	manuell	0	golf	150000.00	0	benzin	volkswagen	NaN	24-03-2016 00:00	0.00	70435	07-04-2016 03:16
1	24-03-2016 10:58	A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	2011	manuell	190	NaN	125000.00	5	diesel	audi	ja	24-03-2016 00:00	0.00	66954	07-04-2016 01:46
2	14-03-2016 12:52	Jeep_Grand_Cherokee_Overland	privat	Angebot	9800	test	suv	2004	automatik	163	grand	125000.00	8	diesel	jeep	NaN	14-03-2016 00:00	0.00	90480	05-04-2016 12:47
3	17-03-2016 16:54	GOLF_4_1.4_3T7ER	privat	Angebot	1500	test	Kleinwagen	2001	manuell	75	golf	150000.00	6	benzin	volkswagen	nein	17-03-2016 00:00	0.00	91074	17-03-2016 17:40
4	31-03-2016 17:25	Skoda_Fabia_1.4_TDI_FDI_Classic	privat	Angebot	3600	test	Kleinwagen	2008	manuell	69	fabia	90000.00	7	diesel	skoda	nein	31-03-2016 00:00	0.00	60437	06-04-2016 10:17

```
print(df.shape)
(371539, 20)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 371539 entries, 0 to 371538
Data columns (total 20 columns):
 #   Column              Non-Null Count  Dtype
---  -
 0  dateCrawled          371539 non-null object
 1  name                 371538 non-null object
 2  seller               371538 non-null object
 3  offerType            371538 non-null object
 4  price                371538 non-null object
 5  abtest               371532 non-null object
 6  vehicleType          333604 non-null object
 7  yearOfRegistration    371517 non-null object
 8  gearbox              351348 non-null object
 9  powerPS              371525 non-null object
10  model                353865 non-null object
11  kilometer            371537 non-null float64
12  monthOfRegistration    371511 non-null object
13  fuelType             338177 non-null object
14  brand                374338 non-null object
15  notRepairedDamage     209576 non-null object
16  dateCreated           371537 non-null object
17  nrOfPictures          372537 non-null float64
18  postalCode           371537 non-null object
19  lastSeen             371235 non-null object
dtypes: float64(2), object(18)
memory usage: 56.7+ MB
```

```
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',
      '116', '184', '126', '88', '194', '305', '197', '179', '250', '45',
      '313', '41', '165', '98', '130', '114', '211', '56', '326', '201',
      '213', '58', '107', '83', '174', '100', '220', '85', '73', '192',
      '68', '66', '299', '74', '52', '510', '147', '65', '310', '71',
      '97', '239', '295', '203', '5', '300', '103', '245', '258', '292',
      '320', '63', '81', '148', '354', '44', '145', '230', '280', '260',
      '457', '104', '409', '188', '333', '196', '117', '141', '59',
      '132', '234', '158', '39', '272', '92', '51', '135', '53', '435',
      '209', '43', '146', '67', '106', '166', '276', '344', '349', '72',
      '249', '237', '111', '279', '23', '138', '256', '175', '96', '226',
      '26', '37', '301', 'andere', '334', '128', '133', '124', '33',
      '219', '241', '167', '94', '182', '196', '238', '277', '1398',
      '505', '228', '112', '48', '139', '1799', '40', '240', '79', '360',
      '243', '375', '252', '396', '340', '108', '121', '380', '345',
      '187', '666', '289', '222', '315', '198', '284', '450', '233',
      '387', '152', '347', '172', '57', '137', '212', '476', '314', '30',
      '7', '47', '173', '215', '370', '290', '207', '385', '205', '171',
      '750', '78', '91', '89', '325', '76', '151', '420', '350', '953',
      '93', 'c_klasse', '142', '500', '343', '199', '119', '164', '16',
      '264', '367', '100', 'focus', '300', '290', '286', '210', '400',
      '275', '46', '1001', '381', '225', '127', '400', '176', 'fortwo',
      '514', 'polo', '161', '603', '307', '35', '268', '206', '771',
      '355', '487', '416', '273', '507', '223', 'cooper', '255', '123',
      '154', '149', '7515', '1400', '620', '266', '480', '169', 'espace',
      '341', '31', '323', '1770', '300', '363', '350', '493', '330']
```

```
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']
```

```
[ ] 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']

[ ] print(df)
# Output:
# powerPS      375223
# seller      382
# gewerblich      3
# 24-03-2018 16:01
# Name: seller, dtype: object
# powerPS      375223
# seller      382
# gewerblich      3
# 24-03-2018 16:01
# Name: seller, dtype: object

[ ] df[df['yearOfRegistration'].str.isnumeric().fillna(False)]
print(df)
# Output:
# yearOfRegistration      375223
# seller      382
# gewerblich      3
# 24-03-2018 16:01
# Name: yearOfRegistration, dtype: object
```

```
df['powerPS']=df['powerPS'].astype(int)
df=df[(df.powerPS > 50) & (df.powerPS < 900)]
print(df.shape)
df=df[df['yearOfRegistration'].str.isnumeric().fillna(False)]
```

```
df['yearOfRegistration']=df['yearOfRegistration'].astype(int)
df=df[(df.yearOfRegistration > 1950) & (df.yearOfRegistration < 2017)]
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
---  -
0   seller                308923 non-null object
1   offerType             308923 non-null object
2   price                 308923 non-null object
3   vehicleType           297510 non-null object
4   yearOfRegistration    308923 non-null int64
5   gearbox               303629 non-null object
6   powerPS               308923 non-null int64
7   model                 297134 non-null object
8   kilometer             308923 non-null float64
9   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', 'yearOfRegistration',
'gearbox', 'powerPS', 'model', 'kilometer', 'monthOfRegistration', 'fuelType',
'notRepairedDamage'])
new_df.gearbox.replace(('manuell', 'automatik'), ('manual', 'automatic'), inplace=True)
new_df.fuelType.replace(('benzin', 'andere', 'elektro'), ('petrol', 'others', '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)
```

```
new_df['price'].unique()
```

```
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 >= 100) & (new_df.price <= 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  
2   price                 278363 non-null  int64  
3   vehicleType           278363 non-null  object  
4   yearOfRegistration     278363 non-null  int64  
5   gearbox               278363 non-null  object  
6   powerPS               278363 non-null  int64  
7   model                 278363 non-null  object  
8   kilometer             278363 non-null  int64  
9   monthOfRegistration    278363 non-null  object  
10  fuelType              278363 non-null  object  
11  brand                 278363 non-null  object  
12  notRepairedDamage     278363 non-null  object  
dtypes: int64(4), object(9)  
memory usage: 29.7+ MB
```

```
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