

## PROJECT DEVELOPMENT PHASE

### DELIVERY OF SPRINT-2

Date	17 November 2022
Team Id	PNT2022TMID21221
Project Name	Project – Car Resale Value Prediction

The model for the prediction is built. In this sprint-2, the data is preprocessed.

#### CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
import pickle

df=pd.read_csv(r"C:\Users\FATHIMASAF\Downloads\resale_predict\Flask\autos.csv",encoding = "Windows-1252")
df.head()
df.tail()

#printing different sellers
print(df.seller.value_counts())

#removing the seller "gewerblich"
df[df.seller != 'gewerblich']

#dropping the coloumn seller as all the entries are same
df = df.drop('seller',1)

#printing different offerType
print(df.offerType.value_counts())

#dropping the offerType 'Gesuch'
```

```
df[df.offerType != 'Gesuch']
```

```
#dropping the coloumn offerType since it has the same entries
```

```
df = df.drop('offerType',1)
```

```
print(df.shape)
```

```
#removing cars having power less that 50p and greater than 900p
```

```
df = df[(df.powerPS > 50) & (df.powerPS < 900)]
```

```
print(df.shape)
```

```
#Keeping all the cars which is registered between 1950 and 2017 and removing the rest
```

```
df = df[(df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)]
```

```
print(df.shape)
```

```
#removing irrelevant coloumns
```

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

```
#dropping the duplicates in the dataframe and storing it in a new dataframe
```

```
newdf = df.copy()
```

```
newdf = newdf.drop_duplicates(['price', 'vehicleType', 'yearOfRegistration', 'gearbox',  
'powerPS', 'model', 'kilometer', 'monthOfRegistration', 'fuelType',  
'notRepairedDamage'])
```

```
#replacing the german words with proper english words
```

```
newdf.gearbox.replace(('manuell','automatik'), ('manual', 'automatic'), inplace = True)
```

```
newdf.fuelType.replace(('benzin','andere','elektro'), ('petrol', 'others', 'electric'), inplace  
= True)
```

```
newdf.vehicleType.replace(('kleinwagen','cabrio','kombi','andere'),('small  
car','convertible','combination', 'others'), inplace = True)
```

```
newdf.notRepairedDamage.replace(('ja','nein'), ('yes', 'no'), inplace = True)
```

#Removing the outliers

```
newdf = newdf[(newdf.price >= 100) & (newdf.price < 15000)]
```

#filling NaN using fillna

```
newdf['notRepairedDamage'].fillna(value = 'not-declared', inplace = True)
```

```
newdf['fuelType'].fillna(value = 'not-declared', inplace = True)
```

```
newdf['gearbox'].fillna(value = 'not-declared', inplace = True)
```

```
newdf['vehicleType'].fillna(value = 'not-declared', inplace = True)
```

```
newdf['model'].fillna(value = 'not-declared', inplace = True)
```

#saving the cleaned dataset

```
newdf.to_csv("autos_preprocessed.csv")
```