

Project name	Car resale value prediction
Team ID	PNT2022TMID11620
Sprint number	1

In sprint 1 we have

1. Imported the necessary libraries

```
import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
import pickle
from sklearn.model_selection import cross_val_score, train_test_split
```

2. Reading the data

```
df = pd.read_csv("Data/autos.csv", header=0, sep=',', encoding='Latin1',)
```

3. Cleaning the entire data

```
df[df.seller != 'gewblich']
df=df.drop('seller',1)

print(df.offerType.value_counts())

df[df.offerType != 'Gesuch']
df = df.drop('offerType',1)

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

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

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

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

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

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)
```

4. Conversion of German data to English

```
new_df.gearbox.replace(('manuell','automatik'),('manual','automatic'), inplace=True)
new_df.fuelType.replace(('benzin','andere','elecktro'),('petrol','others','electic'),
inplace=True)
new_df.vehicleType.replace(('kleinwagen','cabrio','kambi','andere'),('small
car','convertible','combination','others'), inplace=True)
new_df.notRepairedDamage.replace(('ja','nein'),('Yes','No'), inplace=True)
```

5. Exporting the pre-processed data

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

6. Creating the .npz files

```
l=['gearbox','notRepairedDamage','fuelType','vehicleType','model','brand']

m={}
for i in l:
    m[i]=LabelEncoder()
    m[i].fit(new_df[i])
    tr=m[i].transform(new_df[i])
    np.save(str('classes'+i+'.npz'),m[i].classes_)
    print(i,":",m[i])
    new_df.loc[:,i+'_labels']=pd.Series(tr,index=new_df.index)

l2=new_df[['price','yearOfRegistration','powerPS','kilometer','monthOfRegistration'] +
[x+"_labels" for x in l]]
```

7. Splitting the train and testing data









```
print(l2.columns)

Y=l2.iloc[:,0].values
X=l2.iloc[:,1:].values

Y=Y.reshape(-1,1)

X_train,X_test,Y_train,Y_test=train_test_split(X,Y, test_size=0.3, random_state=3)
```

Output:

 autos_preprocessed.csv	20-11-2022 16:25	Microsoft Excel Comma ...	20,001 KB
 classesbrand.npy	20-11-2022 16:25	NPY File	1 KB
 classesfuelType.npy	20-11-2022 16:25	NPY File	1 KB
 classesgearbox.npy	20-11-2022 16:25	NPY File	1 KB
 classesmodel.npy	20-11-2022 16:25	NPY File	4 KB
 classesnotRepairedDamage.npy	20-11-2022 16:25	NPY File	1 KB
 classesvehicleType.npy	20-11-2022 16:25	NPY File	1 KB
 preprocess_data.py	17-11-2022 01:37	Python Source File	3 KB

```
df=df.drop('seller',1)
notRepairedDamage : LabelEncoder()
fuelType : LabelEncoder()
vehicleType : LabelEncoder()
model : LabelEncoder()
brand : LabelEncoder()
Index(['price', 'yearOfRegistration', 'powerPS', 'kilometer',
      'monthOfRegistration', 'gearbox_labels', 'notRepairedDamage_labels',
      'fuelType_labels', 'vehicleType_labels', 'model_labels',
      'brand_labels'],
      dtype='object')
```