CLEANING THE DATASET

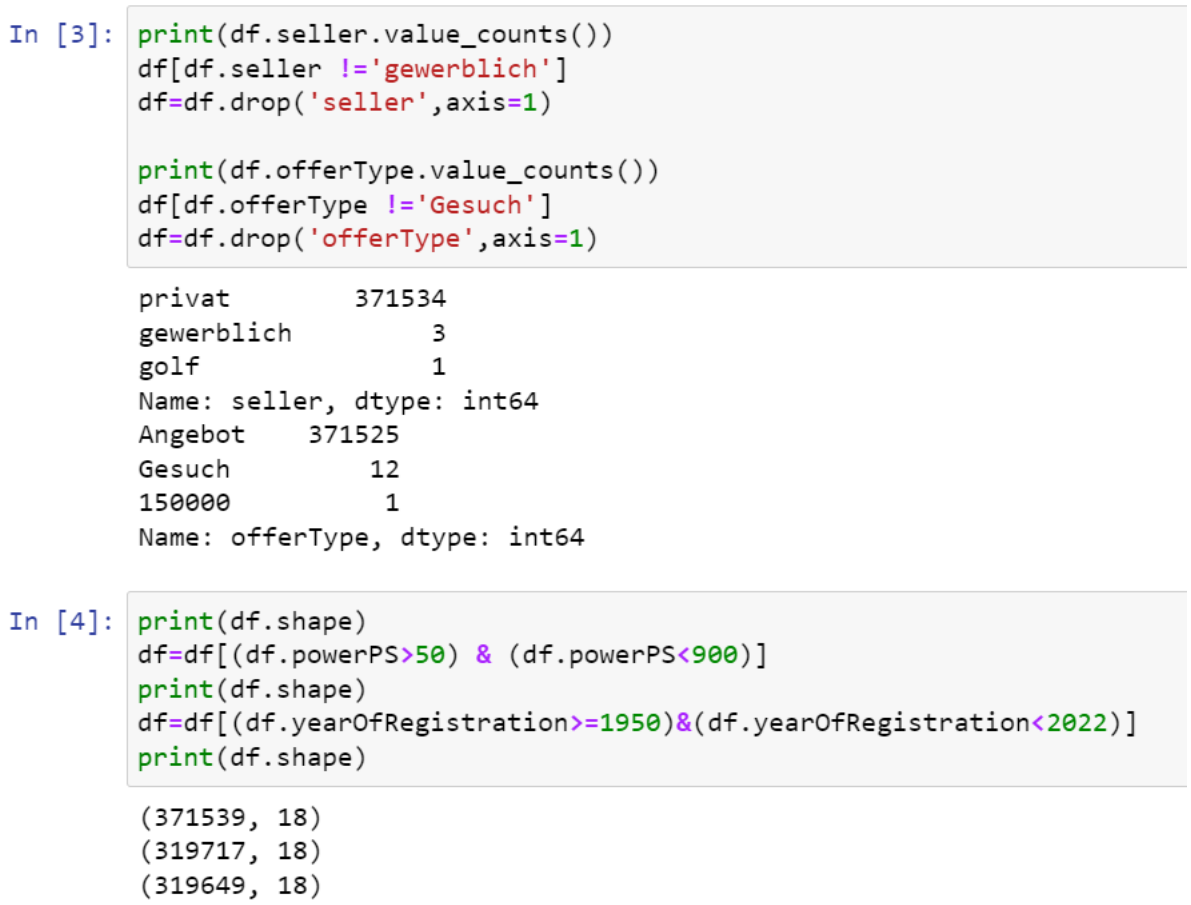
Team ID	**PNT2022TMID16122**
Project Name	**Car Resale value Prediction**

CLEANING THE DATASET

```
print(df.seller.value_counts())
df[df.seller !='gewerblich']
df=df.drop('seller',axis=1)

print(df.offerType.value_counts())
df[df.offerType !='Gesuch']
df=df.drop('offerType',axis=1)
```

```
print(df.shape)
df=df[(df.powerPS>50) & (df.powerPS<900)]
print(df.shape)
df=df[(df.yearOfRegistration>=1950)&(df.yearOfRegistration<2022)]
print(df.shape)
```



```
In [3]: print(df.seller.value_counts())
        df[df.seller !='gewerblich']
        df=df.drop('seller',axis=1)

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

        privat        371534
        gewerblich         3
        golf               1
        Name: seller, dtype: int64
        Angebot     371525
        Gesuch          12
        150000           1
        Name: offerType, dtype: int64

In [4]: print(df.shape)
        df=df[(df.powerPS>50) & (df.powerPS<900)]
        print(df.shape)
        df=df[(df.yearOfRegistration>=1950)&(df.yearOfRegistration<2022)]
        print(df.shape)

        (371539, 18)
        (319717, 18)
        (319649, 18)
```

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

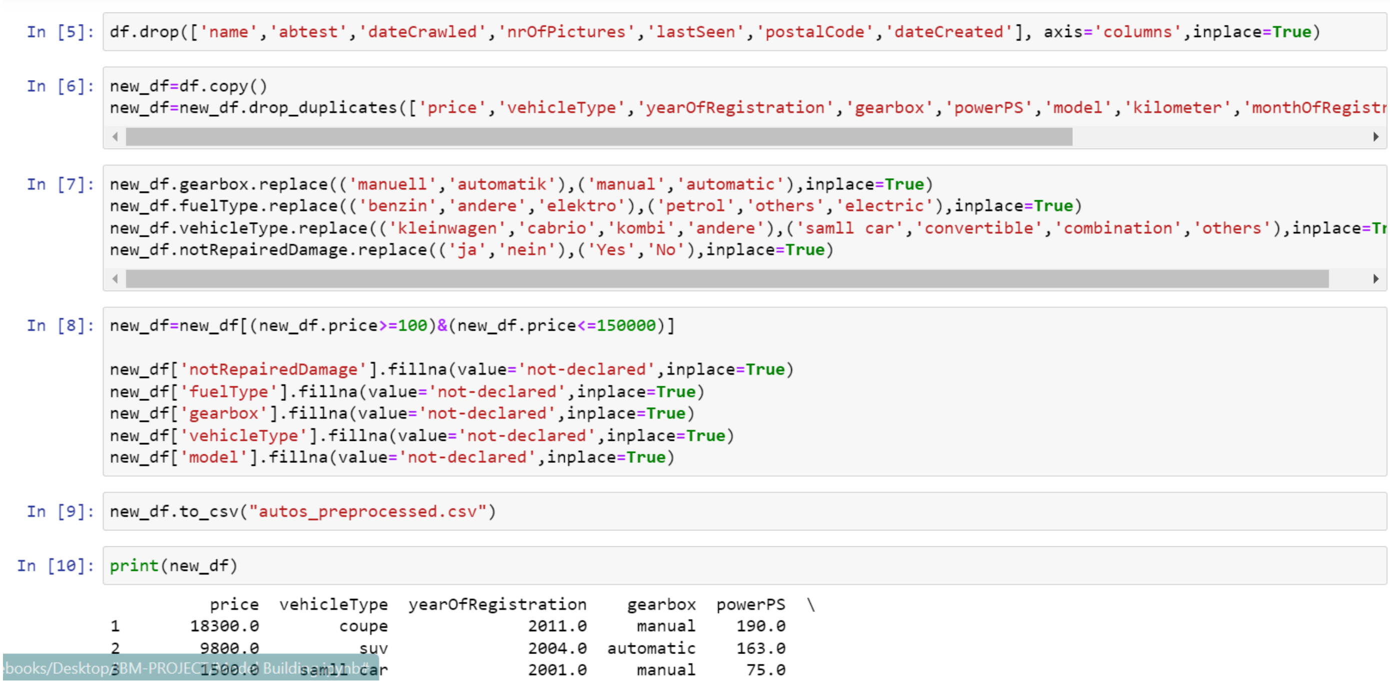
```
new_df=df.copy()
new_df=new_df.drop_duplicates(['price','vehicleType','yearOfRegistration','gearbox','power
PS','model','kilometer','monthOfRegistration','fuelType','notRepairedDamage'])
```

```
new_df.gearbox.replace(('manuell','automatik'),('manual','automatic'),inplace=True)
new_df.fuelType.replace(('benzin','andere','elektro'),('petrol','others','electric'),inpla
ce=True)
new_df.vehicleType.replace(('kleinwagen','cabrio','kombi','andere'),('samll
car','convertible','combination','others'),inplace=True)
new_df.notRepairedDamage.replace(('ja','nein'),('Yes','No'),inplace=True)
```

```
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)
```

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



```
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','monthOfRegistr

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

In [8]: 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)

In [9]: new_df.to_csv("autos_preprocessed.csv")

In [10]: print(new_df)

           price  vehicleType  yearOfRegistration    gearbox  powerPS  \
1        18300.0        coupe              2011.0     manual    190.0
2         9800.0          suv              2004.0  automatic    163.0
3         1500.0    samll car              2001.0     manual     75.0
```

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

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

labeled = new_df[['price','yearOfRegistration','powerPS','kilometer','monthOfRegistration']+[x+"_labels" for x in labels]]
print(labeled.columns)
```