

PROJECT DEVELOPMENT PHASE

SPRINT-2

PROJECT NAME:	CAR RESALE VALUE PREDICTION
TEAM ID:	PNT2022TMID05109
DATE:	07-11-2022

```
In [1]: import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
import pickle
```

```
In [2]: df = pd.read_csv("autos.csv",header=0,sep=',',encoding='Latin1',)
```

```
In [3]: df[df.seller != 'gewerblich']
df=df.drop( 'seller', 1)
df[df.offerType != 'Gesuch']
df=df.drop( 'offerType', 1)
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only
after removing the cwd from sys.path.

```
In [4]: df = df[ (df.powerPS > 50) & (df.powerPS < 900) ]
df = df[ (df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)]
```

```
In [5]: df.drop(['name','abtest','dateCrawled','nrOfPictures','lastSeen','postalCo
```

```
In [6]: new_df = df.copy()
new_df = new_df.drop_duplicates(['price','vehicleType','yearOfRegistration
```

```
In [7]: new_df.gearbox.replace(('manuell','automatik'),('manual','automatic'),inpl
new_df.fuelType.replace(('benzin','andere','elektro'),('petrol','others','
new_df.vehicleType.replace(('kleinwagen','cabrio','kombi','andere'),('smal
new_df.notRepairedDamage.replace(('ja','nein'),('Yes','No'),inplace=True)
```

```
In [8]: new_df = new_df[(new_df.price >= 100) & (new_df.price <= 150000)]
```

```
In [9]: 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 [10]: new_df.to_csv("autos_preprocessed.csv")

#label encoding the categorical data
labels = ['gearbox','notRepairedDamage','model','brand','fuelType','vehicl
```

```
In [11]: 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','
gearbox ; LabelEncoder()
notRepairedDamage ; LabelEncoder()
model ; LabelEncoder()
brand ; LabelEncoder()
fuelType ; LabelEncoder()
vehicleType ; LabelEncoder()
```

```
In [12]: print(labeled.columns)

Index(['price', 'yearOfRegistration', 'powerPS', 'kilometer',
      'monthOfRegistration', 'gearbox_labels', 'notRepairedDamage_label
s',
      'model_labels', 'brand_labels', 'fuelType_labels',
      'vehicleType_labels'],
      dtype='object')
```

```
dtype= object )
```

```
In [13]: Y = labeled.iloc[:,0].values  
X = labeled.iloc[:,1:].values
```

```
In [14]: Y = Y.reshape(-1,1)  
from sklearn.model_selection import cross_val_score , train_test_split  
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.3,random_
```

```
In [15]: from sklearn.ensemble import RandomForestRegressor  
from sklearn.metrics import r2_score  
regressor = RandomForestRegressor(n_estimators = 1000,max_depth=10,random_
```

```
In [16]: regressor.fit(X_train,np.ravel(Y_train,order='C'))
```

```
Out[16]: RandomForestRegressor(max_depth=10, n_estimators=1000, random_state=34)
```

```
In [17]: y_pred = regressor.predict(X_test)  
print(r2_score(Y_test,y_pred))
```

```
0.8042868905072561
```

```
In [ ]: filename = 'resale_model.sav'
```