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

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

```
df.columns
```

```
    Index(['dateCrawled', 'name', 'offerType', 'price', 'abtest', 'vehicleType',
           'yearOfRegistration', 'gearbox', 'powerPS', 'model', 'kilometer',
           'monthOfRegistration', 'fuelType', 'brand', 'notRepairedDamage',
           'dateCreated', 'nrOfPictures', 'postalCode', 'lastSeen'],
          dtype='object')
```

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

```
    privat        371525
    gewerblich         3
    Name: seller, dtype: int64
    Angebot    371516
    Gesuch         12
    Name: offerType, dtype: int64
    /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: FutureWarning: In a
                                                                        ports until
                                                                        Warning: In a
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
print(df.shape)
df = df[(df.powerPS > 50) & (df.powerPS <900)]
print(df.shape)
print(df.shape)
df = df[(df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)]
print(df.shape)
```

```
    (371528, 18)
    (319709, 18)
    (319709, 18)
    (309171, 18)
```

```
df.drop(['name', 'abtest', 'dateCrawled', 'nrOfPictures', 'lastSeen','postalCode', 'dateC
new_df = df.copy()
```

```
df.columns
```

```
    Index(['price', 'vehicleType', 'yearOfRegistration', 'gearbox', 'powerPS',
           'model', 'kilometer', 'monthOfRegistration', 'fuelType', 'brand',
           'notRepairedDamage'],
          dtype='object')
```

```
new_df.columns
```

```
    Index(['price', 'vehicleType', 'yearOfRegistration', 'gearbox', 'powerPS',
           'model', 'kilometer', 'monthOfRegistration', 'fuelType', 'brand',
           'notRepairedDamage'],
          dtype='object')
```

```
new_df = new_df.drop_duplicates(['price', 'vehicleType', 'yearOfRegistration', 'gearbox',
```

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

```
    /usr/local/lib/python3.7/dist-packages/pandas/core/generic.py:6619: SettingWithCopyW
    A value is trying to be set on a copy of a slice from a DataFrame
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
                                                                      -docs/stable/u
      return self._update_inplace(result)
```

```
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)
```

```
    /usr/local/lib/python3.7/dist-packages/pandas/core/generic.py:6392: SettingWithCopyW
    A value is trying to be set on a copy of a slice from a DataFrame

    See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
      return self._update_inplace(result)
```

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

```
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' , 'monthOfRegistr
print(labeled.columns)
```

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

```
Y=labeled.iloc[:,0].values
X=labeled.iloc[:,1:].values
Y=Y.reshape(-1,1)


from sklearn.model_selection import cross_val_score, train_test_split
                                                                       ndom_state = 3)
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

Colab paid products - Cancel contracts here

✓ 0s completed at 10:17 PM ● ×

Automatic saving failed. This file was updated remotely or in another tab. Show diff