File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.9

Run Format Code

In [2]:
```
import pandas as pd
```

In [3]:
```
import numpy as np
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
import pickle
```

In [4]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='yhSA46tGa0ele00Szl0tOfmNB7NDvvVT4hZOsE0B_67E',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'carresale-donotdelete-pr-ras7p9cygmm77y'
object_key = 'autos_preprocessed.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

dataset = pd.read_csv(body)
dataset.head()
```

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.9

Run Format Code

Out[4]:

	Unnamed: 0	price	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage
0	1	18300	coupe	2011	manual	190	not-declared	125000	5	diesel	audi	Yes
1	2	9800	suv	2004	automatic	163	grand	125000	8	diesel	jeep	not-declared
2	3	1500	small car	2001	manual	75	golf	150000	6	petrol	volkswagen	No
3	4	3600	small car	2008	manual	69	fabia	90000	7	diesel	skoda	No
4	5	650	limousine	1995	manual	102	3er	150000	10	petrol	bmw	Yes

In [6]:
```
dataset.head()
```

Out[6]:

	Unnamed: 0	price	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage
0	1	18300	coupe	2011	manual	190	not-declared	125000	5	diesel	audi	Yes
1	2	9800	suv	2004	automatic	163	grand	125000	8	diesel	jeep	not-declared
2	3	1500	small car	2001	manual	75	golf	150000	6	petrol	volkswagen	No
3	4	3600	small car	2008	manual	69	fabia	90000	7	diesel	skoda	No
4	5	650	limousine	1995	manual	102	3er	150000	10	petrol	bmw	Yes

In [6]:
```
print(dataset.seller.value_counts())
```

```
privat        371525
gewerblich         3
Name: seller, dtype: int64
```

In [8]:
```
df[df.offerType != 'Gesuch']
df=df.drop('offerType',1)
print (df.shape)
```

```
(371528, 18)
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2446718189.py:2: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument
```

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.9

Run Format Code

```
'labels' will be keyword-only.
  df=df.drop('offerType',1)
```

In [39]:
```
df=df[(df.powerPS> 50) & (df.powerPS < 900)]
print(df.shape)
df=df[(df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)]
print(df.shape)
```

```
(309171, 11)
(309171, 11)
```

In [50]:
```
new_df = df.copy()
new_df = new_df.drop_duplicates(['price','vehicleType','yearOfRegistration','gearbox','powerPS','model','kilometer','monthOfRegistration','fuelType','notRepairedDamage'])
```

In [52]:
```
new_df.gearbox.replace(('manuell','automatik'),('manual','automatic'), inplace = True)
```

In [54]:
```
new_df.fuelType.replace(('benzin','andere','elektro'),('petrol','others','electric'), inplace = True)
new_df.vehicleType.replace(('kleinwagen','cabrio','kombi','andere'),('small car','convertible','combination','others'), inplace = True)
new_df.notRepairedDamage.replace(('ja','nein'),('Yes','No'), inplace = True)
```

In [56]:
```
new_df=new_df[(new_df.price >=100) & (new_df.price <= 1500000)]
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")
```

```
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2212685035.py:2: SettingWithCopyWarning:
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/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df['notRepairedDamage'].fillna(value ='not-declared', inplace= True)
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2212685035.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.9

Run Format Code

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df['fuelType'].fillna(value ='not-declared', inplace= True)
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2212685035.py:4: SettingWithCopyWarning:
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/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df['gearbox'].fillna(value ='not-declared', inplace= True)
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2212685035.py:5: SettingWithCopyWarning:
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/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df['vehicleType'].fillna(value ='not-declared', inplace= True)
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2212685035.py:6: SettingWithCopyWarning:
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/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df['model'].fillna(value ='not-declared', inplace= True)
```

In [70]:

```
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)
```

```
gearbox : LabelEncoder()
notRepairedDamage : LabelEncoder()
```

```
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2853306416.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.9

Run Format Code

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df.loc[:, i +'_labels'] = pd.Series(tr, index = new_df.index)
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2853306416.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df.loc[:, i +'_labels'] = pd.Series(tr, index = new_df.index)
```

```
model : LabelEncoder()
brand : LabelEncoder()
```

```
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2853306416.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df.loc[:, i +'_labels'] = pd.Series(tr, index = new_df.index)
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2853306416.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df.loc[:, i +'_labels'] = pd.Series(tr, index = new_df.index)
```

```
fuelType : LabelEncoder()
vehicleType : LabelEncoder()
```

```
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2853306416.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df.loc[:, i +'_labels'] = pd.Series(tr, index = new_df.index)
C:\Users\RajiyaSan\AppData\Local\Temp\ipykernel_11756\2853306416.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3.9

Run Format Code

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_df.loc[:, i +'_labels'] = pd.Series(tr, index = new_df.index)
```

In [74]:
```
labeled = new_df[['price'
                  ,'yearOfRegistration'
                  ,'powerPS'
                  ,'kilometer'
                  ,'monthOfRegistration'
                 ]
                 + [ x +"_labels" for x in labels]]

print(labeled.columns)
```

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

In [134]:
```
X = labeled.iloc[:,0].values
y= labeled.iloc[:,1].values


X= X.reshape(-1,1)
#X= X.reshape(-1,1)
print (labeled.shape)
print (X.shape)
print (y.shape)
```

```
(278718, 11)
(278718, 1)
(278718,)
```

In [141]:
```
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_state =3)
```

In [142]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
regressor = RandomForestRegressor (n_estimators =1000, max_depth = 10, random_state = 34)
```

In [144]:
```
regressor.fit (X_train, np.ravel(Y_train, order = 'C'))
```

Out[144]:
```
RandomForestRegressor(max_depth=10, n_estimators=1000, random_state=34)
```

In [145]:
```
y_pred = regressor.predict (X_test)
```

In [146]:
```
print(r2_score(Y_test,y_pred))
```

```
0.29318977362082255
```

In [147]:
```
filename= 'resale_model.pkl'
pickle.dump(regressor, open (filename,'wb'))
```