Date	10 November 2022
Team ID	PNT2022TMID40489
Project Name	PROJECT-CAR RESALES VALUE PREDICTION
Maximum Marks	2 Marks

A training model is a dataset that is used to train an algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model.

This iterative process is called "model fitting". The accuracy of the training dataset or the validation dataset is critical for the precision of the model.

Model training is the process of feeding an algorithm with data to help identify and learn good values for all attributes involved.

```
import seaborn as sns from
matplotlib import * import
sys from pylab import *
plt.figure(figsize=[11,5])
sns.distplot(new df['price'
M Welcome to Project! Delighted : X | 3 IBM
                             X (O) IBM.ipynb - Colaboratory
← → C 🕯 colab.research.google.com/drive/11j9sNX357ox7oXpLi7LX8hTzwrveQq0h#scrollTo=pHbQ7D2r3bsF
                                                                         £ ☆ □ D :
 CO & IBM.ipynb 🌣
                                                                  Comment A Share D
    sns.distplot(new_df['price'])
                                                                     ↑ ↓ ⊕ □ ‡ [ i :
Q
      {x}
0
B IBM.ipynb
plt.figure(figsize=[17,5])
plt.subplot(1,3,1)
```

```
sns.distplot(new df['yearOfRegistration'])
plt.title('Car Year Distribution Plot')
plt.subplot(1,3,2)
sns.distplot(new df['powerPS'])
plt.title('Car power Plot')
 M Welcome to Project! Delighted t × 3 IBM
                                                X CO IBM.ipynb - Colaboratory
                                                                         × • What is Model Training | Oden To × +
 ← → C  

colab.research.google.com/drive/1lj9sNX357ox7oXpLi7LXBhTzwrveQq0h#scrollTo=aOWsCtps5gQz
                                                                                                                            B & 0 0 :
 CO & IBM.ipynb 🌣
                                                                                                                □ Comment 🎎 Share 🌣 🕟
       File Edit View Insert Runtime Tools Help <u>Last edited on November 7</u>
      + Code + Text
                                                                                                                     Connect - Editing ^
       • pit.title( kilometer distribution Piot )
 \equiv
                                                                                                                      ↑ ↓ © □ ‡ 🗓 🖥
 a
          /usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version.
           Warnings.warnings, Futurewarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version.
 {x}
           warnings.warn(msg, FutureWarning) /usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version.
 Car Year Distribution Plot
                                                             Car power Plot
                                                                                            kilometer Distribution Plot
                                                                                 0.00020
             0.08
                                               0.010
                                               0.008
             0.06
                                                                                 0.00015
                                              0.006
                                                                                 0.00010
                                               0.004
             0.02
                                                                                 0.00009
 1950 1960 1970 1980 1990 2000 2010 2020
                                                                                                                                    Show all X
  BM.ipynb
                                 ★ 77*F
Mostly cloudy
new df.describe()
                                                 X O IBM.ipynb - Colaboratory
                                                                                                                                       0 ×
 ► Welcome to Project! Delighted to × S IBM
                                                                          × • What is Model Training | Oden Tr × +
 ← → C 🕯 colab.research.google.com/drive/1lj9sNX357ox7oXpLi7tX8hTzwrveQq0h#scrollTo=98WNw6XN7CYq
                                                                                                                              ₽☆□□:
                                                                                                                 □ Comment ♣ Share 🌣 🕟
       File Edit View Insert Runtime Tools Help Last edited on November 7
      + Code + Text
                                                                                                                      Connect - PEditing A
                                                                                                                       ↑ ↓ © 目 章 □ 盲 :
 Q
       new_df.describe()
 {x}
                    price yearOfRegistration powerPS kilometer
            count 278363.00
                            278363.00 278363.00 278363.00
 6420 87
                                  2003 37
                                           129 25 125331 24
            min
                   100.00
                                   1951.00
                                            51.00
                                                    5000.00
            50%
                  3600.00
                                  2004.00
                                            116.00 150000.00
                           2008.00 150.00 150000.00
            max 150000.00
                                  2016.00
                                           871.00 150000.00
 45
       [ ] plt.figure(figsize=[20,7])
 plt.subplot(1,3,1)
plt.title("Price and Year")
sns.scatterplot(x='yearOfRegistration',y='price',data=new_df)
 >-
  IBM.ipynb
                                  11/14/2022 D
plt.figure(figsize=[20,7])
plt.subplot(1,3,1)
```

plt.title("Price and Year")

```
sns.scatterplot(x='yearOfRegistration',y='price',data=new df)
plt.subplot(1,3,2)
plt.title("price and kilometer")
sns.scatterplot(x='price', y='kilometer', data=new df)
plt.subplot(1,3,3)
sns.scatterplot(y='price', x='powerPS', data=new df)
plt.title("Price and powerPS")
 M Welcome to Project! Delighted to X | S IBM
                                      x CO IBM.ipynb - Colaboratory
                                                         ← → C 🖟 colab.research.google.com/drive/11j9sNX357ox7oXpLi7LXBhTzwrveQq0h#scrollTo=m_pa0d_-7Rxm
                                                                                                  ፼☆□□:
 CO & IBM.ipynb 🌣
                                                                                       ■ Comment 😃 Share 🌣
      File Edit View Insert Runtime Tools Help <u>Last edited on November 7</u>
     Text(0.5, 1.0, 'Price and powerPS')
                                                                                             ↑ ↓ ◎ 目 ☆ ☑ 前 :
 Q
      0
 {x}
          140000
                                          14000
 120000
 ()
                            1990
                                2000
                                                                                                       Show all X
 BM.ipynb
                          log price = np.log(new df['price'])
new df['log price'] = log price
new df.head()
    seller offerType price vehicleType yearOfRegistration gearbox powerPS
                                                         model kilometer monthOfRegistration fuelType
                                                                                             brand notRepairedDamage 1
  1 privat
          Angebot 18300
                         coupe
                                       2011
                                            manual
                                                     190
                                                                125000
                                                                                     diesel
                                                                                              audi
                                                                                                           Yes
                                                        declared
           Angebot 9800
                                       2004 automatic
                                                     163
                                                                125000
                                                                                                      not-declared
  2 privat
                           SUV
                                                         grand
                                                                                 8
                                                                                     diesel
                                                                                              ieep
                        small car
  3 privat
           Angebot
                 1500
                                       2001
                                            manual
                                                     75
                                                          golf
                                                                150000
                                                                                     petrol volkswagen
                                                                90000
  4 privat
           Angebot
                 3600
                        small car
                                       2008
                                            manual
                                                     69
                                                          fabia
                                                                                     diesel
                                                                                             skoda
                                                                                                           No
           Angebot
                        limousine
                                                     102
                                                                150000
plt.figure(figsize=[20,7])
plt.subplot(1,3,1)
plt.title("Price and Year")
```

```
sns.scatterplot(x='yearOfRegistration',y='log price',data=new df)
plt.subplot(1,3,2)
plt.title("price and kilometer")
sns.scatterplot(x='kilometer',y='log price',data=new df)
plt.subplot(1,3,3)
sns.scatterplot(y='price', x='log price', data=new df)
plt.title("Price and powerPS")
M Welcome to Project! Delighted to X S IBM
                                        × • What is Model Training | Oden Te × | +
 母☆□□:
                                                              Comment A Share
    File Edit View Insert Runtime Tools Help Last edited on November 7
   + Code + Text
                                                                 Connect - Fediting A
                                                                 ↑ ↓ co □ □ □ [ i :
    ● Text(0.5, 1.0, 'Price and powerPS')
Q
                                                            Price and powerPS
{x}
1980 1990 2000 2010
 B IBM.ipynb
                  new df= new df.drop(['price'],axis=1)
new_df['monthOfRegistration']=new_df['monthOfRegistration'].astype(int)
labels= ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType',
've hicleType'] 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[
['log price','yearOfRegistration','powerPS','kilometer','
monthOfRegistration']
+ [x+" labels" for x in labels]]
print(labeled.columns)
```

```
notRepairedDamage : LabelEncoder()
 model : LabelEncoder()
 brand : LabelEncoder()
 fuelType : LabelEncoder()
 vehicleType : LabelEncoder()
 Index(['log price', 'yearOfRegistration', 'powerPS', 'kilometer',
         'monthOfRegistration', 'gearbox_labels', 'notRepairedDamage_labels',
         'model_labels', 'brand_labels', 'fuelType_labels',
         'vehicleType labels'],
        dtype='object')
plt.figure(figsize=[15,7])
sns.heatmap(new df.corr(), annot=True)
                                x CO IBM.ipynb - Colaboratory
 M Welcome to Project! Delighted to X | 3 IBM
                                                × • What is Model Training | Oden Tr × +
← → C 🖟 colab.research.google.com/drive/11j9sNX357ox7oXpLi7LXBhTzwrveQq0h#scrollTo=25OtPQNKDCuc
                                                                                  @ # O D :
CO & IBM.ipynb 🌣
                                                                         Comment A Share
    File Edit View Insert Runtime Tools Help All changes saved
                                                                             Connect - Fediting A
   + Code + Text
    sns.heatmap(new_dt.corr(), annot=Irue)
⊨
                                                                             ↑ ↓ © □ ‡ 🖟 🖹 :
       <matplotlib.axes._subplots.AxesSubplot at 0x7fa7f3b952d0>
{x}
1
                               1
>_
 BM.ipynb
                      10/21 PM D C 😭 📵 👊 🚳 🐧 🚳 🐧 🔞 🥞 🦠 O 🚱 🖓 A R ENG 🖘 40 🗈 11/14/2022 D
Mostly cloudy
Y =labeled.iloc[:,0].values
X = labeled.iloc[:,1:].values Y = Y.reshape(-1,1) from
sklearn.model selection import train test split, cross val score X train,
X test, Y train, Y test = train test split(X,Y,test size=0.3, ran
dom state=3) from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2 score
regressor= RandomForestRegressor (n estimators=1000, max_depth=10,
random state=34)
regressor.fit(X train, np.ravel (Y train, order='C'))
y pred=regressor.predict(X test)
                                              print(r2 score
(Y test, y pred))
                        y pred=regressor.predict(X test)
print(r2 score (Y test, y pred))
```

gearbox : LabelEncoder()

```
df ev = pd.DataFrame(np.exp(y pred), columns=['Predicted Price'])
# We can also include the Actual price column in that data frame (so we ca
n manually compare them)
#Y test=Y test.reset index(drop=True)
df ev['Actual Price'] = np.exp(Y test)
# we can calculate the difference between the targets and the predictions
df ev['Residual'] = df ev['Actual Price'] - df ev['Predicted Price']
df ev['Difference%'] = np.absolute(df ev['Residual']/df ev['Actual
Price']
*100)
pd.set option('display.float format', lambda x: '%.2f' %
x) df ev.sort values(by=['Difference%']) df ev.tail(5)
          Predicted Price Actual Price Residual Difference%
    83504
                               5790.00
                                                      14.57
                  4946.32
                                         843.68
    83505
                               5200.00 1022.08
                                                      19.66
                  4177.92
    83506
                 11025.04
                              12499.00
                                       1473.96
                                                      11.79
    83507
                  7967.92
                               9800.00 1832.08
                                                      18.69
    83508
                   564.48
                                400.00
                                        -164.48
                                                      41.12
from sklearn.linear model import LinearRegression
lr = LinearRegression() lr.fit(X train, Y train)
y pred lr = lr.predict(X test) r squared
r2 score(Y test, y pred lr)
                           print("R squared
:",r squared)
from sklearn.ensemble import
GradientBoostingRegressor gbt =
GradientBoostingRegressor() gbt.fit(X train, Y train)
y pred gbt = gbt.predict(X test) r squared =
r2 score(Y test, y pred gbt) print("R squared
:", r squared)
df ev = pd.DataFrame(np.exp(y pred gbt), columns=['Predicted Price'])
df_ev['Actual Price'] = np.exp(Y_test) df_ev['Residual'] = df_ev['Actual
               df ev['Predicted
                                   Price']
                                                 df ev['Difference%']
np.absolute(df ev['Residual']/df ev['Actual Price']
*100)
```

pd.set_option('display.float_format', lambda x: '%.2f' % x) df ev.sort values(by=['Difference%']) df ev.tail(5) M Welcome to Project! Delighted to X | 3 IBM × © IBM.ipynb - Colaboratory × • What is Model Training | Oden Tr × + ← → C 🕯 colab.research.google.com/drive/1lj9sNX357ox7oXpLi7LX8hTzwrveQq0h#scrollTo=vLbxXSj2MVDs ❷☆□□: CO 💪 IBM.ipynb 🌣 Comment A Share D File Edit View Insert Runtime Tools Help All changes saved Connect - Fediting A + Code + Text ↑ ↓ © **目 ‡** [**i** : df_ev.tail(5) Q Predicted Price Actual Price Residual Difference% {x} 83504 5554.97 5790.00 235.03 83505 3859 59 5200 00 1340 41 25.78 9829.38 12499.00 2669.62 21.36 83506 83507 8553.73 9800.00 1246.27 640.32 400.00 -240.32 60.08 83508 [] filename = 'resale_model.sav' pickle.dump(gbt, open(filename, 'wb')) <> >_ BM.ipynb Show all X ■ Q ■ D © 😭 🖪 👊 🚇 🚇 🔘 🧿 🚱 🖓 ^ 🐧 NS 🖘 (4) D 10/27 PM 11/14/2022 D

filename = 'resale model.sav'

pickle.dump(gbt, open(filename, 'wb'))