ChooseYourProblemAndData

August 8, 2023

1 Assigment 8: Choose Your ML Problem and Data

In this unit's lab, you will implement a model to solve a machine learning problem of your choosing. First, you will have to make some decisions, such as which model to choose and which data preparation techniques may be necessary, and formulate a project plan accordingly.

In this assignment, you will select a data set and choose a predictive problem that the data set supports. You will then inspect the data with your problem in mind and begin to formulate your project plan. You will create this project plan in the written assignment that follows.

1.0.1 Import Packages

Before you get started, import a few packages. You can import additional packages that you have used in this course that you may need for this task.

[2]:
```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

1.1 Step 1: Choose Your Data Set and Load the Data

You will have the option to choose one of four data sets that you have worked with in this program:

- The "adult" data set that contains Census information from 1994: `adultData.csv`
- Airbnb NYC "listings" data set: `airbnbListingsData.csv`
- World Happiness Report (WHR) data set: `WHR2018Chapter2OnlineData.csv`
- Book Review data set: `bookReviewsData.csv`

Note that these are variations of the data sets that you have worked with in this program. For example, some do not include some of the preprocessing necessary for specific models.

Load the Data Set The code cell below contains filenames (path + filename) for each of the four data sets available to you.

Task: In the code cell below, use the same method you have been using to load the data using `pd.read_csv()` and save it to DataFrame `df`.

You can load each file as a new DataFrame to inspect the data before choosing your data set.

[3]:
```
# File names of the four data sets
adultDataSet_filename = os.path.join(os.getcwd(), "data", "adultData.csv")
airbnbDataSet_filename = os.path.join(os.getcwd(), "data", "airbnbListingsData.
↪csv")
WHRDataSet_filename = os.path.join(os.getcwd(), "data",␣
↪"WHR2018Chapter2OnlineData.csv")
bookReviewDataSet_filename = os.path.join(os.getcwd(), "data", "bookReviewsData.
↪csv")

df=pd.read_csv(adultDataSet_filename)
df.head()
```

[3]:
```
    age         workclass  fnlwgt  education  education-num  \
0  39.0         State-gov   77516  Bachelors             13
1  50.0  Self-emp-not-inc   83311  Bachelors             13
2  38.0           Private  215646    HS-grad              9
3  53.0           Private  234721       11th              7
4  28.0           Private  338409  Bachelors             13

       marital-status         occupation   relationship   race sex_selfID  \
0       Never-married       Adm-clerical  Not-in-family  White Non-Female
1  Married-civ-spouse    Exec-managerial        Husband  White Non-Female
2            Divorced  Handlers-cleaners  Not-in-family  White Non-Female
3  Married-civ-spouse  Handlers-cleaners        Husband  Black Non-Female
4  Married-civ-spouse     Prof-specialty           Wife  Black     Female

   capital-gain  capital-loss  hours-per-week native-country income_binary
0          2174             0            40.0  United-States         <=50K
1             0             0            13.0  United-States         <=50K
2             0             0            40.0  United-States         <=50K
3             0             0            40.0  United-States         <=50K
4             0             0            40.0           Cuba         <=50K
```

1.2 Step 2: Choose Your Predictive Problem and Label

Now that you have chosen your data set, you can:

1. Choose what you would like to predict (i.e. the label)
2. Identify your problem type: is it a classification or regression problem?

Task: In the markdown cell below, state what you are predicting (the label) and whether this is a classification or regression problem.

The label I would like to predict is the "income_binary" label. This is a classification problem because the variable has two categories: the income is less than or equal to 50,000 dollars, or more than 50,000 dollars.

1.3 Step 3: Inspect Your Data

In the code cell below, use some of the techniques you have learned in this course to take a look at your data. As you are investigating your data, consider the following to help you formulate your project plan:

1. What are my features?
2. Which model (or models) should I select that is appropriate for my machine learning problem and data?
3. Which data preparation techniques may be needed for my model (e.g. perform one-hot encoding)?
4. Which techniques should I use to evaluate my model's performance and improve my model?

Note: You will use this notebook to take a glimpse at your data to help you start making some considerations. In the written assignment you will outline your project plan, and in the lab assignment you will perform a deeper exploratory analysis of the data before implementing data preparation and feature engineering techniques.

Task: Use the techniques you have learned in this course to inspect your data.

Note: You can add code cells if needed by going to the Insert menu and clicking on Insert Cell Below in the drop-drown menu.

[4]:
```
# YOUR CODE HERE
df.head()
```

[4]:
```
    age         workclass  fnlwgt  education  education-num  \
0  39.0         State-gov   77516  Bachelors             13
1  50.0  Self-emp-not-inc   83311  Bachelors             13
2  38.0           Private  215646    HS-grad              9
3  53.0           Private  234721       11th              7
4  28.0           Private  338409  Bachelors             13

       marital-status         occupation   relationship   race  sex_selfID  \
0       Never-married       Adm-clerical  Not-in-family  White  Non-Female
1  Married-civ-spouse    Exec-managerial        Husband  White  Non-Female
2            Divorced  Handlers-cleaners  Not-in-family  White  Non-Female
3  Married-civ-spouse  Handlers-cleaners        Husband  Black  Non-Female
4  Married-civ-spouse     Prof-specialty           Wife  Black      Female

   capital-gain  capital-loss  hours-per-week native-country income_binary
0          2174             0            40.0  United-States         <=50K
1             0             0            13.0  United-States         <=50K
2             0             0            40.0  United-States         <=50K
3             0             0            40.0  United-States         <=50K
4             0             0            40.0           Cuba         <=50K
```

[5]:
```
df.shape
```

[5]:
```
(32561, 15)
```

[6]:
```
df.dtypes
```

[6]:
```
age                float64
workclass           object
fnlwgt               int64
education           object
education-num        int64
marital-status      object
occupation          object
relationship        object
race                object
sex_selfID          object
capital-gain         int64
capital-loss         int64
hours-per-week     float64
native-country      object
income_binary       object
dtype: object
```

[7]:
```
df.isnull().sum()
```

[7]:
```
age                162
workclass         1836
fnlwgt               0
education            0
education-num        0
marital-status       0
occupation        1843
relationship         0
race                 0
sex_selfID           0
capital-gain         0
capital-loss         0
hours-per-week     325
native-country     583
income_binary        0
dtype: int64
```

[8]:
```
df.describe()
```

[8]:
```
                age        fnlwgt  education-num  capital-gain  capital-loss  \
count  32399.000000  3.256100e+04   32561.000000  32561.000000  32561.000000
mean      38.589216  1.897784e+05      10.080679    615.907773     87.303830
std       13.647862  1.055500e+05       2.572720   2420.191974    402.960219
min       17.000000  1.228500e+04       1.000000      0.000000      0.000000
25%       28.000000  1.178270e+05       9.000000      0.000000      0.000000
50%       37.000000  1.783560e+05      10.000000      0.000000      0.000000
75%       48.000000  2.370510e+05      12.000000      0.000000      0.000000
max       90.000000  1.484705e+06      16.000000  14084.000000   4356.000000

       hours-per-week
```

```
count    32236.000000
mean        40.450428
std         12.353748
min          1.000000
25%         40.000000
50%         40.000000
75%         45.000000
max         99.000000
```

[9]:
```
df["income_binary"].value_counts()
```

[9]:
```
<=50K    24720
>50K      7841
Name: income_binary, dtype: int64
```

[10]:
```
for col in df.select_dtypes(include="object").columns:
    print(col,df[col].unique())
```

```
workclass ['State-gov' 'Self-emp-not-inc' 'Private' 'Federal-gov' 'Local-gov'
nan
 'Self-emp-inc' 'Without-pay' 'Never-worked']
education ['Bachelors' 'HS-grad' '11th' 'Masters' '9th' 'Some-college' 'Assoc-
acdm'
 'Assoc-voc' '7th-8th' 'Doctorate' 'Prof-school' '5th-6th' '10th'
 '1st-4th' 'Preschool' '12th']
marital-status ['Never-married' 'Married-civ-spouse' 'Divorced' 'Married-spouse-
absent'
 'Separated' 'Married-AF-spouse' 'Widowed']
occupation ['Adm-clerical' 'Exec-managerial' 'Handlers-cleaners' 'Prof-
specialty'
 'Other-service' 'Sales' 'Craft-repair' 'Transport-moving'
 'Farming-fishing' 'Machine-op-inspct' 'Tech-support' nan
 'Protective-serv' 'Armed-Forces' 'Priv-house-serv']
relationship ['Not-in-family' 'Husband' 'Wife' 'Own-child' 'Unmarried' 'Other-
relative']
race ['White' 'Black' 'Asian-Pac-Islander' 'Amer-Indian-Inuit' 'Other']
sex_selfID ['Non-Female' 'Female']
native-country ['United-States' 'Cuba' 'Jamaica' 'India' nan 'Mexico' 'South'
 'Puerto-Rico' 'Honduras' 'England' 'Canada' 'Germany' 'Iran'
 'Philippines' 'Italy' 'Poland' 'Columbia' 'Cambodia' 'Thailand' 'Ecuador'
 'Laos' 'Taiwan' 'Haiti' 'Portugal' 'Dominican-Republic' 'El-Salvador'
 'France' 'Guatemala' 'China' 'Japan' 'Yugoslavia' 'Peru'
 'Outlying-US(Guam-USVI-etc)' 'Scotland' 'Trinadad&Tobago' 'Greece'
 'Nicaragua' 'Vietnam' 'Hong' 'Ireland' 'Hungary' 'Holand-Netherlands']
income_binary ['<=50K' '>50K']
```

[12]:
```
class_counts=df["income_binary"].value_counts()
total_samples=len(df)
percentage_0=(class_counts[0]/total_samples)*100
```

```python
percentage_1=(class_counts[1]/total_samples)*100

print("Percentage of class 0 (<=50K): {:.2f}%".format(percentage_0))
print("Percentage of class 1 (>50K): {:.2f}%".format(percentage_1))
```

```
Percentage of class 0 (<=50K): 75.92%
Percentage of class 1 (>50K): 24.08%
```

[ ]:

```python

```