Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №2
по дисциплине
«Технологии машинного обучения»
на тему
«Изучение библиотек обработки данных»

Выполнил:
студент группы ИУ5-63Б
Кривцов Н. А.

Москва — 2020 г.

0.1. mlcourse.ai - Open Machine Learning Course

Author: Yury Kashnitsky. Translated and edited by Sergey Isaev, Artem Trunov, Anastasia Manokhina, and Yuanyuan Pao.

#

Assignment #1 (demo) ##

Exploratory data analysis with Pandas

Same assignment as a Kaggle Kernel + solution.

In this task you should use Pandas to answer a few questions about the Adult dataset. (You don't have to download the data – it's already in the repository). Choose the answers in the web-form.

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

[1]:
```
import numpy as np
import pandas as pd
pd.set_option('display.max.columns', 100)
# to draw pictures in jupyter notebook
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
# we don't like warnings
# you can comment the following 2 lines if you'd like to
import warnings
warnings.filterwarnings('ignore')
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19:
FutureWarning: pandas.util.testing is deprecated. Use the functions in
 ↪the
public API at pandas.testing instead.
  import pandas.util.testing as tm
```

[2]:
```
data = pd.read_csv('/content/drive/My Drive/TMO/labs/datasets/adult.
 ↪data.csv')
data.head()
```

```
[2]:    age         workclass  fnlwgt  education  education-num  \
     0   39         State-gov   77516  Bachelors             13
     1   50  Self-emp-not-inc   83311  Bachelors             13
     2   38           Private  215646    HS-grad              9
     3   53           Private  234721       11th              7
     4   28           Private  338409  Bachelors             13

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

        capital-gain  capital-loss  hours-per-week native-country salary
     0          2174             0              40  United-States  <=50K
     1             0             0              13  United-States  <=50K
     2             0             0              40  United-States  <=50K
     3             0             0              40  United-States  <=50K
     4             0             0              40           Cuba  <=50K
```

1. How many men and women (*sex* feature) are represented in this dataset?

```
[3]: data['sex'].value_counts()
```

```
[3]: Male      21790
     Female    10771
     Name: sex, dtype: int64
```

2. What is the average age (*age* feature) of women?

```
[4]: data.loc[data['sex'] == 'Female', 'age'].mean()
```

```
[4]: 36.85823043357163
```

3. What is the percentage of German citizens (*native-country* feature)?

```
[5]: data['native-country'].value_counts(normalize=True).loc['Germany'] * 100
```

```
[5]: 0.42074874850281013
```

4-5. What are the mean and standard deviation of age for those who earn more than 50K per year (*salary* feature) and those who earn less than 50K per year?

```
[6]: poor = data.loc[data['salary'] == '<=50K']
     rich = data.loc[data['salary'] == '>50K']
     print(rich['age'].mean(), rich['age'].std(), poor['age'].mean(),
      ↪poor['age'].std())
```

```
44.24984058155847 10.51902771985177 36.78373786407767 14.020088490824813
```

6. Is it true that people who earn more than 50K have at least high school education? (*education – Bachelors, Prof-school, Assoc-acdm, Assoc-voc, Masters* or *Doctorate* feature)

```
[7]: rich['education'].unique()
```

```
[7]: array(['HS-grad', 'Masters', 'Bachelors', 'Some-college', 'Assoc-voc',
            'Doctorate', 'Prof-school', 'Assoc-acdm', '7th-8th', '12th',
            '10th', '11th', '9th', '5th-6th', '1st-4th'], dtype=object)
```

7. **Display age statistics for each race (*race* feature) and each gender (*sex* feature). Use *groupby()* and *describe()*. Find the maximum age of men of *Amer-Indian-Eskimo* race.**

```
[8]: grouped = data.groupby(by=['race', 'sex'])
     grouped['age'].describe()
```

```
[8]:                                  count       mean        std   min   25%  ↪
 ↪50%  \
race               sex
Amer-Indian-Eskimo Female   119.0  37.117647  13.114991  17.0  27.0 ↪
 ↪36.0
                   Male     192.0  37.208333  12.049563  17.0  28.0 ↪
 ↪35.0
Asian-Pac-Islander Female   346.0  35.089595  12.300845  17.0  25.0 ↪
 ↪33.0
                   Male     693.0  39.073593  12.883944  18.0  29.0 ↪
 ↪37.0
Black              Female  1555.0  37.854019  12.637197  17.0  28.0 ↪
 ↪37.0
                   Male    1569.0  37.682600  12.882612  17.0  27.0 ↪
 ↪36.0
Other              Female   109.0  31.678899  11.631599  17.0  23.0 ↪
 ↪29.0
                   Male     162.0  34.654321  11.355531  17.0  26.0 ↪
 ↪32.0
White              Female  8642.0  36.811618  14.329093  17.0  25.0 ↪
 ↪35.0
                   Male   19174.0  39.652498  13.436029  17.0  29.0 ↪
 ↪38.0

                             75%   max
race               sex
Amer-Indian-Eskimo Female  46.00  80.0
                   Male    45.00  82.0
Asian-Pac-Islander Female  43.75  75.0
                   Male    46.00  90.0
Black              Female  46.00  90.0
                   Male    46.00  90.0
Other              Female  39.00  74.0
                   Male    42.00  77.0
White              Female  46.00  90.0
                   Male    49.00  90.0
```

```
[9]: print("Max age among Amer-Indian-Eskimo men: {}.".format(grouped.
      ↪get_group(('Amer-Indian-Eskimo', 'Male'))['age'].max()))
```

```
Max age among Amer-Indian-Eskimo men: 82.
```

8. Among whom is the proportion of those who earn a lot (>50K) greater: married or single men (*marital-status* feature)? Consider as married those who have a *marital-status* starting with *Married* (Married-civ-spouse, Married-spouse-absent or Married-AF-spouse), the rest are considered bachelors.

[10]:
```
men = data.loc[data['sex'] == 'Male']
married_indices = men['marital-status'].str.startswith('Married')
married_men = men.loc[married_indices]
single_men = men.loc[~married_indices]
rich_married_proportion = married_men.loc[married_men['salary'] ==
  ↪'>50K'].shape[0] / married_men.shape[0]
rich_single_proportion = single_men.loc[single_men['salary'] == '>50K'].
  ↪shape[0] / single_men.shape[0]
print("Percentage of rich men among married men: {:.2%}.".
  ↪format(rich_married_proportion))
print("Percentage of rich men among single men: {:.2%}.".
  ↪format(rich_single_proportion))
```

```
Percentage of rich men among married men: 44.05%.
Percentage of rich men among single men: 8.45%.
```

9. What is the maximum number of hours a person works per week (*hours-per-week* feature)? How many people work such a number of hours, and what is the percentage of those who earn a lot (>50K) among them?

[11]:
```
max_hpw = data['hours-per-week'].max()
max_hpw_workers = data.loc[data['hours-per-week'] == max_hpw]
print('Max hours per week: {}.'.format(max_hpw_workers.shape[0]))
rich_max_hpw_workers = max_hpw_workers.loc[data['salary'] == '>50K']
print('Percentage of rich people among "workaholics": {:.2%}.'.
  ↪format(rich_max_hpw_workers.shape[0] / max_hpw_workers.shape[0]))
```

```
Max hours per week: 85.
Percentage of rich people among "workaholics": 29.41%.
```

10. Count the average time of work (*hours-per-week*) for those who earn a little and a lot (*salary*) for each country (*native-country*). What will these be for Japan?

[12]:
```
country_salary_groups = data.groupby(by=['native-country', 'salary'])
country_salary_groups['hours-per-week'].mean()
```

[12]:
```
native-country  salary
?               <=50K     40.164760
                >50K      45.547945
Cambodia        <=50K     41.416667
                >50K      40.000000
Canada          <=50K     37.914634
                            …
United-States   >50K      45.505369
Vietnam         <=50K     37.193548
```

```
                 >50K      39.200000
Yugoslavia       <=50K     41.600000
                 >50K      49.500000
Name: hours-per-week, Length: 82, dtype: float64
```

[13]:
```
print('Average hours per week among rich Japanese: {}.'.
 ↪format(country_salary_groups.get_group(('Japan',
 ↪'>50K'))['hours-per-week'].mean()))
print('Average hours per week among poor Japanese: {}.'.
 ↪format(country_salary_groups.get_group(('Japan',
 ↪'<=50K'))['hours-per-week'].mean()))
```

```
Average hours per week among rich Japanese: 47.958333333333336.
Average hours per week among poor Japanese: 41.0.
```