Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)

О Т Ч Е Т

по лабораторной работе

по дисциплине: Технологии машинного обучения

на тему: Изучение библиотек обработки данных

Студент ИУ5-62Б А.Д. Карягин
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **Ю.Е. Гапанюк**
(Подпись, дата) (И.О.Фамилия)

2020 г.

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

In [2]:
```
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')
```

In [3]:
```
data = pd.read_csv('data/adult.data.csv') data.head()
```

Out[3]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	se
0	39	State-gov	77516	Bachelors	13	Never- married	Adm clerica	Not-in-family	White	Ma
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Ma
2	38	Private	215646	HS-grad	9	Divorced	Handlers- c	Not-in-family	White	Ma
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers- cleaners	Husband	Black	Ma
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Fema

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

In [4]:
```
data['sex'].value_counts() # data.groupby('sex').count()
```

Out[4]:
```
Male      21790
Female    10771
Name: sex, dtype: int64
```

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

In [5]:
```
data.groupby(['sex'])['age'].mean()
```

Out[5]:
```
sex
Female    36.858230
Male      39.433547
Name: age, dtype: float64
```

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

In [55]:
```
print(round((data['native-country'] == 'Germany').sum() / data.shape[0]
* 100, 2), "%")
```

```
0.42 %
```

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?

In [56]:
```
ages1 = data.loc[data['salary'] == '>50K', 'age'] ages2 = data.loc[data['salary'] == '<=50K', 'age']
print("The average age of the rich: {0} +- {1} years, poor - {2} +- {3} years.".format(
round(ages1.mean()), round(ages1.std(), 1),
round(ages2.mean()), round(ages2.std(), 1)))
```

```
The average age of the rich: 44.0 +- 10.5 years, poor - 37.0 +- 14.0 yea
rs.
```

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)

In [8]:
```
data.loc[data['salary'] == '>50K', 'education'].unique()
```

Out[8]:
```
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)
```

No, it isn't true

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.

In [57]:
```
data.groupby(['race', 'sex'])['age'].describe() # the maximum age of men of Amer-Indian-Eskimo race is 82
```

Out[57]:

race	sex	count	mean	std	min	25%	50%	75%	max
Amer-Indian-Eskimo	**Female**	119.0	37.117647	13.114991	17.0	27.0	36.0	46.00	80.0
	Male	192.0	37.208333	12.049563	17.0	28.0	35.0	45.00	82.0
Asian-Pac-Islander	**Female**	346.0	35.089595	12.300845	17.0	25.0	33.0	43.75	75.0
	Male	693.0	39.073593	12.883944	18.0	29.0	37.0	46.00	90.0
Black	**Female**	1555.0	37.854019	12.637197	17.0	28.0	37.0	46.00	90.0
	Male	1569.0	37.682600	12.882612	17.0	27.0	36.0	46.00	90.0
Other	**Female**	109.0	31.678899	11.631599	17.0	23.0	29.0	39.00	74.0
	Male	162.0	34.654321	11.355531	17.0	26.0	32.0	42.00	77.0
White	**Female**	8642.0	36.811618	14.329093	17.0	25.0	35.0	46.00	90.0
	Male	19174.0	39.652498	13.436029	17.0	29.0	38.0	49.00	90.0

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.

In [58]:
```
data.loc[(data['sex'] == 'Male') & (~data['marital-status'].str.startswi th('Married')), 'salary'].value_counts()
```

Out[58]:
```
<=50K    7552
>50K      697
Name: salary, dtype: int64
```

In [59]:
```
data.loc[(data['sex'] == 'Male') & (data['marital-status'].str.startswit h('Married')), 'salary'].value_counts()
```

Out[59]:
```
<=50K    7576
>50K     5965
Name: salary, dtype: int64
```

married > singe men (earn >50K)

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?

In [60]:
```
max_num = data['hours-per-week'].max()
quantity = data.loc[data['hours-per-week'] == max_num, 'age'].count() per=data[(data['hours-per-week'] == ma
)].shape[0]/quantity*100
print('maximum number of hours a person works per week^ ', max_num) print('people work such a number of ho
print('the percentage of those who earn a lot (>50K): ', round(per, 2), "%")
```

```
maximum number of hours a person works per week^  99
people work such a number of hours:  85
the percentage of those who earn a lot (>50K):  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?

In [61]:
```
pd.options.display.max_rows = 999
data.groupby(['native-country', 'salary'])['hours-per-week'].mean()
```

Out[61]:
```
native-country              salary
?                           <=50K     40.164760
                            >50K      45.547945
Cambodia                    <=50K     41.416667
                            >50K      40.000000
Canada                      <=50K     37.914634
                            >50K      45.641026
China                       <=50K     37.381818
                            >50K      38.900000
Columbia                    <=50K     38.684211
                            >50K      50.000000
Cuba                        <=50K     37.985714
                            >50K      42.440000
Dominican-Republic          <=50K     42.338235
                            >50K      47.000000
Ecuador                     <=50K     38.041667
                            >50K      48.750000
```

El-Salvador	<=50K	36.030928
	>50K	45.000000
England	<=50K	40.483333
	>50K	44.533333
France	<=50K	41.058824
	>50K	50.750000
Germany	<=50K	39.139785
	>50K	44.977273
Greece	<=50K	41.809524
	>50K	50.625000
Guatemala	<=50K	39.360656
	>50K	36.666667
Haiti	<=50K	36.325000
	>50K	42.750000
Holand-Netherlands	<=50K	40.000000
Honduras	<=50K	34.333333
	>50K	60.000000
Hong	<=50K	39.142857
	>50K	45.000000
Hungary	<=50K	31.300000
	>50K	50.000000
India	<=50K	38.233333
	>50K	46.475000
Iran	<=50K	41.440000
	>50K	47.500000
Ireland	<=50K	40.947368
	>50K	48.000000
Italy	<=50K	39.625000
	>50K	45.400000
Jamaica	<=50K	38.239437
	>50K	41.100000
Japan	<=50K	41.000000
	>50K	47.958333
Laos	<=50K	40.375000
	>50K	40.000000
Mexico	<=50K	40.003279
	>50K	46.575758
Nicaragua	<=50K	36.093750
	>50K	37.500000
Outlying-US(Guam-USVI-etc)	<=50K	41.857143
Peru	<=50K	35.068966
	>50K	40.000000
Philippines	<=50K	38.065693
	>50K	43.032787
Poland	<=50K	38.166667
	>50K	39.000000
Portugal	<=50K	41.939394
	>50K	41.500000
Puerto-Rico	<=50K	38.470588
	>50K	39.416667
Scotland	<=50K	39.444444
	>50K	46.666667
South	<=50K	40.156250
	>50K	51.437500
Taiwan	<=50K	33.774194
	>50K	46.800000
Thailand	<=50K	42.866667
	>50K	58.333333
Trinadad&Tobago	<=50K	37.058824
	>50K	40.000000
United-States	<=50K	38.799127
	>50K	45.505369
Vietnam	<=50K	37.193548

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

Japan <=50K 41.000000 >50K 47.958333