*Государственное образовательное учреждение высшего профессионального образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

**«Лабораторная работа №2»**

«Технологии машинного обучения»

**ИСПОЛНИТЕЛЬ:**

Студен группы РТ5-61

Курьянов А.И.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю.Е.

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2020 г.

**Задание:**

**В этом задании вы должны использовать Pandas, чтобы ответить на несколько вопросов о наборе данных.**

**Подключение библиотек и загрузка датасета:**

In[1]:

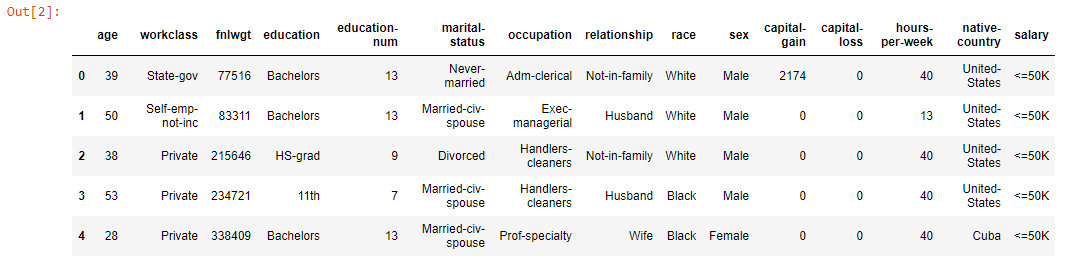
import numpy as np

import pandas as pd

In[2]:

data = pd.read\_csv('data/adult.data.csv')

data.head()



**1. Сколько мужчин и женщин (характеристика sex) представлено в этом наборе данных?**

In[3]:

data['sex'].value\_counts()



**2. Каков средний возраст (характеристика age) женщин?**

In[4]:

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



**3. Каков процент граждан Германии (характеристика native-country)?**

In[5]:

float((data['native-country'] == 'Germany').sum()) / data.shape[0]



**4-5. Каково среднее и стандартное отклонение возраста для тех, кто зарабатывает более 50 тысяч в год (характеристика salary) и тех, кто зарабатывает менее 50 тысяч в год?**

In[6]:

ages1 = data.loc[data['salary'] == '>50K', 'age']

ages2 = data.loc[data['salary'] == '<=50K', 'age']

print("Средний возраст богатых: {0} +- {1} лет, бедных - {2} +- {3} лет.".format(

round(ages1.mean()), round(ages1.std(), 1),

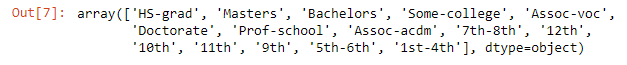
round(ages2.mean()), round(ages2.std(), 1)))



**6. Правда ли, что люди, которые зарабатывают более 50 тысяч, имеют хотя бы среднее образование? (образование - бакалавр, проф-школа, доцент, ассистент, магистр или доктор)**

In[7]:

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



**7. Отображение статистики возраста для каждой расы (характеристика race) и каждого пола (характеристика sex). Используйте groupby() и describe(). Найти максимальный возраст мужчин американо-индийско-эскимосской расы.**

In[8]:

for (race, sex), sub\_df in data.groupby(['race', 'sex']):

print("Race: {0}, sex: {1}".format(race, sex))

print(sub\_df['age'].describe())

Out[8]:

Race: Amer-Indian-Eskimo, sex: Female

count 119.000000

mean 37.117647

std 13.114991

min 17.000000

25% 27.000000

50% 36.000000

75% 46.000000

max 80.000000

Name: age, dtype: float64

Race: Amer-Indian-Eskimo, sex: Male

count 192.000000

mean 37.208333

std 12.049563

min 17.000000

25% 28.000000

50% 35.000000

75% 45.000000

max 82.000000

Name: age, dtype: float64

Race: Asian-Pac-Islander, sex: Female

count 346.000000

mean 35.089595

std 12.300845

min 17.000000

25% 25.000000

50% 33.000000

75% 43.750000

max 75.000000

Name: age, dtype: float64

Race: Asian-Pac-Islander, sex: Male

count 693.000000

mean 39.073593

std 12.883944

min 18.000000

25% 29.000000

50% 37.000000

75% 46.000000

max 90.000000

Name: age, dtype: float64

Race: Black, sex: Female

count 1555.000000

mean 37.854019

std 12.637197

min 17.000000

25% 28.000000

50% 37.000000

75% 46.000000

max 90.000000

Name: age, dtype: float64

Race: Black, sex: Male

count 1569.000000

mean 37.682600

std 12.882612

min 17.000000

25% 27.000000

50% 36.000000

75% 46.000000

max 90.000000

Name: age, dtype: float64

Race: Other, sex: Female

count 109.000000

mean 31.678899

std 11.631599

min 17.000000

25% 23.000000

50% 29.000000

75% 39.000000

max 74.000000

Name: age, dtype: float64

Race: Other, sex: Male

count 162.000000

mean 34.654321

std 11.355531

min 17.000000

25% 26.000000

50% 32.000000

75% 42.000000

max 77.000000

Name: age, dtype: float64

Race: White, sex: Female

count 8642.000000

mean 36.811618

std 14.329093

min 17.000000

25% 25.000000

50% 35.000000

75% 46.000000

max 90.000000

Name: age, dtype: float64

Race: White, sex: Male

count 19174.000000

mean 39.652498

std 13.436029

min 17.000000

25% 29.000000

50% 38.000000

75% 49.000000

max 90.000000

Name: age, dtype: float64

**8. Среди кого больше доля тех, кто много зарабатывает (> 50 тыс.): Замужние или одинокие мужчины (характеристика marital-status)? Считается, что в браке находятся те, кто имеет семейное положение, начиная с женатых (Married-civ-spouse, Married-spouse-absent or Married-AF-spouse), остальные считаются холостяками.**

In[9]:

data.loc[(data['sex'] == 'Male') &

(data['marital-status'].isin(['Never-married',

'Separated',

'Divorced',

'Widowed'])), 'salary'].value\_counts()



In[10]:

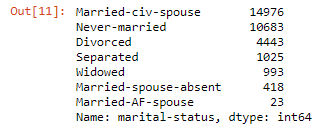
data.loc[(data['sex'] == 'Male') &

(data['marital-status'].str.startswith('Married')), 'salary'].value\_counts()



In[11]:

data['marital-status'].value\_counts()



**9. Какое максимальное количество часов человек работает в неделю (характеристика hours-per-week)? Сколько человек работает такое количество часов и каков процент тех, кто много зарабатывает среди них?**

In[12]:

max\_load = data['hours-per-week'].max()

print("Max time - {0} hours./week.".format(max\_load))

num\_workaholics = data[data['hours-per-week'] == max\_load].shape[0]

print("Total number of such hard workers {0}".format(num\_workaholics))

rich\_share = float(data[(data['hours-per-week'] == max\_load)

& (data['salary'] == '>50K')].shape[0]) / num\_workaholics

print("Percentage of rich among them {0}%".format(int(100 \* rich\_share)))

Out[12]:



**10. Подсчитайте среднее время работы (hours-per-week) тех, кто зарабатывает мало и много (salary) для каждой страны (native-country).**

In[13]:

for (country, salary), sub\_df in data.groupby(['native-country', 'salary']):

print(country, salary, round(sub\_df['hours-per-week'].mean(), 2))

Out[13]:

? <=50K 40.16

? >50K 45.55

Cambodia <=50K 41.42

Cambodia >50K 40.0

Canada <=50K 37.91

Canada >50K 45.64

China <=50K 37.38

China >50K 38.9

Columbia <=50K 38.68

Columbia >50K 50.0

Cuba <=50K 37.99

Cuba >50K 42.44

Dominican-Republic <=50K 42.34

Dominican-Republic >50K 47.0

Ecuador <=50K 38.04

Ecuador >50K 48.75

El-Salvador <=50K 36.03

El-Salvador >50K 45.0

England <=50K 40.48

England >50K 44.53

France <=50K 41.06

France >50K 50.75

Germany <=50K 39.14

Germany >50K 44.98

Greece <=50K 41.81

Greece >50K 50.62

Guatemala <=50K 39.36

Guatemala >50K 36.67

Haiti <=50K 36.33

Haiti >50K 42.75

Holand-Netherlands <=50K 40.0

Honduras <=50K 34.33

Honduras >50K 60.0

Hong <=50K 39.14

Hong >50K 45.0

Hungary <=50K 31.3

Hungary >50K 50.0

India <=50K 38.23

India >50K 46.48

Iran <=50K 41.44

Iran >50K 47.5

Ireland <=50K 40.95

Ireland >50K 48.0

Italy <=50K 39.62

Italy >50K 45.4

Jamaica <=50K 38.24

Jamaica >50K 41.1

Japan <=50K 41.0

Japan >50K 47.96

Laos <=50K 40.38

Laos >50K 40.0

Mexico <=50K 40.0

Mexico >50K 46.58

Nicaragua <=50K 36.09

Nicaragua >50K 37.5

Outlying-US(Guam-USVI-etc) <=50K 41.86

Peru <=50K 35.07

Peru >50K 40.0

Philippines <=50K 38.07

Philippines >50K 43.03

Poland <=50K 38.17

Poland >50K 39.0

Portugal <=50K 41.94

Portugal >50K 41.5

Puerto-Rico <=50K 38.47

Puerto-Rico >50K 39.42

Scotland <=50K 39.44

Scotland >50K 46.67

South <=50K 40.16

South >50K 51.44

Taiwan <=50K 33.77

Taiwan >50K 46.8

Thailand <=50K 42.87

Thailand >50K 58.33

Trinadad&Tobago <=50K 37.06

Trinadad&Tobago >50K 40.0

United-States <=50K 38.8

United-States >50K 45.51

Vietnam <=50K 37.19

Vietnam >50K 39.2

Yugoslavia <=50K 41.6

Yugoslavia >50K 49.5