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

ДИСЦИПЛИНА Технологии машинного обучения

Отчёт по лабораторной работе № 2

«Изучение библиотек обработки данных»

Вариант 9

Выполнил:

Студент группы ИУ5-63 _____ **Королев С.В.**

(Подпись, дата) (Фамилия И.О.)

Проверил:

_____ **Гапанюк Ю.Е.**

(Подпись, дата) (Фамилия И.О.)

Москва, 2020

mlcourse.ai (https://mlcourse.ai) - Open Machine Learning Course

Author: Yury Kashnitsky (https://www.linkedin.com/in/festline/). Translated and edited by Sergey Isaev (https://www.linkedin.com/in/isvforall/), Artem Trunov (https://www.linkedin.com/in/datamove/), Anastasia Manokhina (https://www.linkedin.com/in/anastasiamanokhina/), and Yuanyuan Pao (https://www.linkedin.com/in/yuanyuanpao/).

Assignment #1 (demo)

Исследовательский анализ данных с Pandas

То же назначение, что и Kaggle Kernel (https://www.kaggle.com/kashnitsky/a1-demo-pandas-and-uci-adult-dataset) + solution (https://www.kaggle.com/kashnitsky/a1-demo-pandas-and-uci-adult-dataset-solution).

В этом задании вы должны использовать Pandas, чтобы ответить на несколько вопросов о Adult (https://archive.ics.uci.edu/ml/datasets/Adult) наборе данных. (Вам не нужно скачивать данные - они уже есть в хранилище). Выберите ответы в web-form (https://docs.google.com/forms/d/1uY7Mpl2trKx6FLWZte0uVh3ULV4Cm_tDud0VDFGCOKg).

Уникальные значения всех функций (для получения дополнительной информации, пожалуйста, смотрите ссылки выше):

- `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	sex	capital-gain	capital-loss
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0

1. Сколько мужчин и женщин (половая особенность) представлено в этом наборе данных?

In [15]:
```
data['sex'].value_counts()
```

Out[15]:
```
Male      21790
Female    10771
Name: sex, dtype: int64
```

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

In [30]:
```
round(float(data.loc[data['sex']=='Female', ['age']].mean()))
```

Out[30]: 37

3. Каков процент граждан Германии (функция *родной страны*)?

In [45]:
```
print(float(data.loc[data['native-country']=='Germany', ['native-country']].count()/dat
a['native-country'].count()*100), '%')
```

```
0.42074874850281013 %
```

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

In [57]:
```
print('Среднее отклонение')
print(round(data.groupby(['salary'])['age'].mad()))
print('Стандартное отклонение')
print(round(data.groupby(['salary'])['age'].std()))
```

```
Среднее отклонение
salary
<=50K    11.0
>50K      8.0
Name: age, dtype: float64
Стандартное отклонение
salary
<=50K    14.0
>50K     11.0
Name: age, dtype: float64
```

6. Правда ли, что люди, которые зарабатывают более 50 тысяч, имеют хотя бы среднее образование? (*Образование - Бакалавр, Проф-школа, Assoc-acdm, Assoc-voc, Masters* или *Докторская* функция)

In [135]:
```
flag = True
for i in data.loc[data['salary'] == '>50K', 'education'].unique():
    for j in ['1st-4th','5th-6th','7th-8th','9th','10th','11th','HS-grad','Preschool']:
        if i == j :
            flag = False
            break
    if flag == False :
        break

if flag == True:
    print ('Правда')
else :
    print ('Неправда')
```

```
Неправда
```

7. Отображение статистики по возрасту для каждой расы (функция *раса*) и каждого пола (функция *пол*). Используйте *groupby()* и *describe()*. Найти максимальный возраст мужчин *амер-индейцев-эскимосов* расы.

Статистика по возрасту для каждой расы и каждого пола

In [142]:
```
data.groupby(['race', 'sex'])['age'].describe()
```

Out[142]:

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

Максимальный возраст мужчин амер-индейцев-эскимосов расы

In [151]:
```
int(data.loc[data['race'] == 'Amer-Indian-Eskimo'].loc[data['sex'] == 'Male', ['age']].
max())
```

Out[151]: 82

8. Среди кого больше доля тех, кто зарабатывает много (> 50 тыс.): Замужние или одинокие мужчины (*семейное положение*)? Считается замужем тех, кто имеет *семейное положение*, начиная с *Женат* (Женат-гражданский супруг, Женат-супруг отсутствует или Женат-супруг / супруга), остальные считаются холостяками.

In [8]:
```
ManMarPoor = 0
ManMarRich = 0
for stat in ['Married-civ-spouse','Married-spouse-absent','Married-AF-spouse']:
    a, b = data.loc[data['sex'] == 'Male'].loc[data['marital-status'] == stat]['salary
'].value_counts()
    ManMarPoor += a
    ManMarRich += b
ManPoor, ManRich = data.loc[data['sex'] == 'Male']['salary'].value_counts()

MarProc = ManMarRich / (ManMarPoor + ManMarRich)
AloneProc = (ManRich - ManMarRich) / ((ManPoor + ManRich) - (ManMarPoor + ManMarRich))

if MarProc > AloneProc :
    print ('Среди замужних мужчин больше доля тех, кто зарабатывает много')
else :
    print ('Среди одиноких мужчин больше доля тех, кто зарабатывает много')
# print(MarProc, AloneProc)
```

```
Среди замужних мужчин больше доля тех, кто зарабатывает много
0.4405139945351156 0.08449509031397745
```

9. Какое максимальное количество часов работает человек в неделю (функция *hours-per-week*)? Сколько человек работает такое количество часов, и каков процент тех, кто зарабатывает много (> 50 тыс.) Среди них?

Максимальное количество часов, которое работает человек в неделю

In [9]:
```
int(data['hours-per-week'].max())
```

Out[9]: 99

Количество человек, которые работает максимальное число часов в неделю

In [26]:
```
allPeople = int(data.loc[data['hours-per-week'] == data['hours-per-week'].max(), ['sala
ry']].count())
print (allPeople)
```

```
85
```

Процент от предыдушего результата тех, кто зарабатывает много

In [28]:
```
richPeople = int(data.loc[data['hours-per-week'] == data['hours-per-week'].max()].loc[d
ata['salary'] == '>50K', ['age']].count())
print (richPeople/allPeople*100, '%')
```

```
29.411764705882355 %
```

10. Посчитайте среднее время работы (*часов в неделю*) для тех, кто мало и много зарабатывает (*зарплата*) для каждой страны (*родная страна*). Что это будет для Японии?

In [33]:
```
pd.set_option('display.max_rows', None) # Вывод всей таблицы, без сворачивания
data.groupby(['native-country', 'salary'])['hours-per-week'].mean()
```

Out[33]:

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

In [37]:
```
data.loc[data['native-country'] == 'Japan'].groupby(['native-country', 'salary'])['hours-per-week'].mean()
```

Out[37]:
```
native-country  salary
Japan           <=50K     41.000000
                >50K      47.958333
Name: hours-per-week, dtype: float64
```