Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации
Ордена Трудового Красного Знамени федеральное государственное
бюджетное образовательное учреждение высшего образования
«Московский технический университет связи и информатики»
Кафедра «МКиИТ»

Лабораторная работа №1(часть 1)

по дисциплине «Data mining»

Москва 2023

Открытый курс по машинному обучению.

 Автор материала: программист-исследователь Mail.ru Group, старший преподаватель Факультета Компьютерных Наук ВШЭ Юрий Кашницкий.

Тема 1. Первичный анализ данных с Pandas

Практическое задание. Анализ данных пассажиров "Титаника"

**Заполните код в клетках (где написано "Ваш код здесь")

In [6]:
```
import numpy as np
import pandas as pd
%matplotlib inline
```

Считаем данные из файла в память в виде объекта Pandas.DataFrame

In [8]:
```
data = pd.read_csv('./titanic_train.csv',
                   index_col='PassengerId')
```

Данные представлены в виде таблицы. Посмотрим на первые 5 строк:

In [9]:
```
data.head(5)
```

Out[9]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Eml
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
4	1	1	Futrelle, Mrs.	female	35.0	1	0	113803	53.1000	C123	

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Eml
			(Lily May Peel)								
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	

In [10]:
```
data.describe()
```

Out[10]:

	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Для примера отберем пассажиров, которые сели в Cherbourg (Embarked=C) и заплатили более 200 у.е. за билет (fare > 200).

Убедитесь, что Вы понимаете, как эта конструкция работает.
Если нет – посмотрите, как вычисляется выражение в квадратных в скобках.

In [11]:
```
data[(data['Embarked'] == 'C') & (data.Fare > 200)].head()
```

Out[11]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Er
119	0	1	Baxter, Mr. Quigg Edmond	male	24.0	0	1	PC 17558	247.5208	B58 B60	
259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	
300	1	1	Baxter, Mrs. James (Helene DeLaudeniere Chaput)	female	50.0	0	1	PC 17558	247.5208	B58 B60	
312	1	1	Ryerson, Miss. Emily Borie	female	18.0	2	2	PC 17608	262.3750	B57 B59 B63 B66	
				male	27.0	0	2	113503	211.5000	C82	

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Er
PassengerId											
			Harry Elkins								

Можно отсортировать этих людей по убыванию платы за билет.

In [12]:
```
data[(data['Embarked'] == 'C') &
     (data['Fare'] > 200)].sort_values(by='Fare',
                                       ascending=False).head()
```

Out[12]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embark
259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	
680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	B51 B53 B55	
738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B101	
312	1	1	Ryerson, Miss. Emily Borie	female	18.0	2	2	PC 17608	262.3750	B57 B59 B63 B66	
743	1	1	Ryerson, Miss. Susan Parker "Suzette"	female	21.0	2	2	PC 17608	262.3750	B57 B59 B63 B66	

Пример создания признака.

In [13]:
```
def age_category(age):
    '''
    < 30 -> 1
    >= 30, <55 -> 2
    >= 55 -> 3
    '''
    if age < 30:
        return 1
    elif age < 55:
        return 2
    else:
        return 3
```

In [14]:
```
age_categories = [age_category(age) for age in data.Age]
```

In [15]:
```
data['Age_category'] = age_categories
```

Другой способ – через `apply`.

In [18]:
```
data['Age_category'] = data['Age'].apply(age_category)
```

1. Сколько мужчин / женщин находилось на борту?

- 412 мужчин и 479 женщин
- 314 мужчин и 577 женщин
- 479 мужчин и 412 женщин
- 577 мужчин и 314 женщин

In [22]:
```
print("На борту было {} мужчин и {} женщин".format(sum(data["Sex"] == "male"), sum(d
```

```
На борту было 577 мужчин и 314 женщин
```

2. Выведите распределение переменной `Pclass` (социально-экономический статус) и это же распределение, только для мужчин / женщин по отдельности. Сколько было мужчин 2-го класса?

- 104
- 108
- 112
- 125

In [23]:
```
pd.crosstab(data["Pclass"], data["Sex"], margins=True)
```

Out[23]:

Sex	female	male	All
Pclass			
1	94	122	216
2	76	108	184
3	144	347	491
All	314	577	891

3. Каковы медиана и стандартное отклонение платежей (`Fare`)? Округлите до 2 десятичных знаков.

- Медиана – 14.45, стандартное отклонение – 49.69
- Медиана – 15.1, стандартное отклонение – 12.15
- Медиана – 13.15, стандартное отклонение – 35.3
- Медиана – 17.43, стандартное отклонение – 39.1

In [28]:
```
print("Медина - {}, стандартное отклонение - {}".format(round(data["Fare"].median(),
```

```
Медина - 14.45, стандартное отклонение - 49.69
```

4. Правда ли, что люди моложе 30 лет выживали чаще, чем люди старше 60 лет? Каковы доли выживших в обеих группах?

- 22.7% среди молодых и 40.6% среди старых
- 40.6% среди молодых и 22.7% среди старых
- 35.3% среди молодых и 27.4% среди старых
- 27.4% среди молодых и 35.3% среди старых

In [31]:
```
yung = data[data["Age"] < 30]["Survived"]
old = data[data["Age"] > 60]["Survived"]

yung = round(100 * yung.mean(), 1)
old = round(100 * old.mean(), 1)

print("{}% среди молодых и {}% среди старых".format(yung, old))
```

```
40.6% среди молодых и 22.7% среди старых
```

5. Правда ли, что женщины выживали чаще мужчин? Каковы доли выживших в обеих группах?

- 30.2% среди мужчин и 46.2% среди женщин
- 35.7% среди мужчин и 74.2% среди женщин
- 21.1% среди мужчин и 46.2% среди женщин
- 18.9% среди мужчин и 74.2% среди женщин

In [34]:
```
female = data[data["Sex"] == "female"]["Survived"]
male = data[data["Sex"] == "male"]["Survived"]

female = round(100 * female.mean(), 1)
male = round(100 * male.mean(), 1)

print("{}% среди мужчин и {}% среди женщин".format(male, female))
```

```
18.9% среди мужчин и 74.2% среди женщин
```

6. Найдите самое популярное имя среди пассажиров Титаника мужского пола?

- Charles
- Thomas
- William
- John

In [37]:
```
name = data[data['Sex'] == 'male']['Name'].apply(lambda x: x.split(',')[1].split()[1
name.value_counts().head(1)
```

Out[37]:
```
William    35
Name: Name, dtype: int64
```

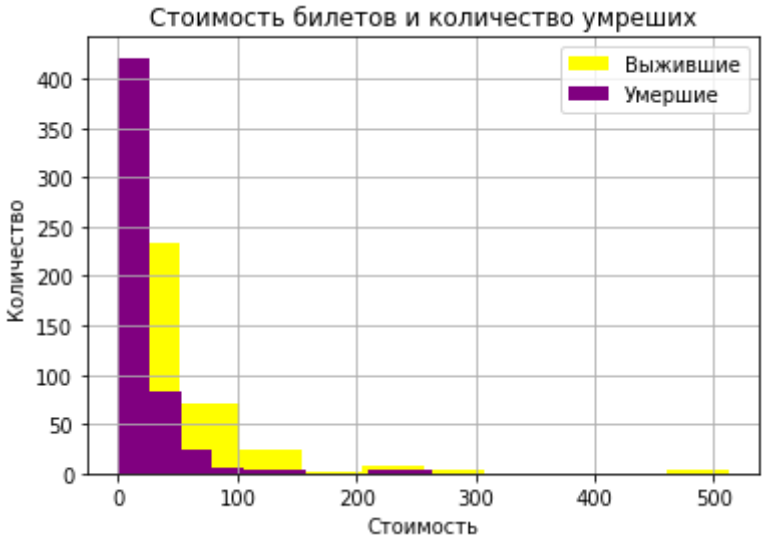
7. Сравните графически распределение стоимости билетов и возраста у спасенных и у погибших. Средний возраст погибших выше, верно?

- Да
- Нет

In [44]:
```
import matplotlib.pyplot as plt

#графическое представление
```

```
data[data["Survived"] == 1]["Fare"].hist(color="yellow", label= "Выжившие")
data[data["Survived"] == 0]["Fare"].hist(color="purple", label= "Умершие")
plt.title("Стоимость билетов и количество умреших")
plt.xlabel("Стоимость")
plt.ylabel("Количество")
plt.legend();
```



In [45]:

```
data[data["Survived"] == 1]["Age"].hist(color="yellow", label= "Выжившие")
data[data["Survived"] == 0]["Age"].hist(color="purple", label= "Умершие")
plt.title("Количество умреших по возрасту")
plt.xlabel("Возраст")
plt.ylabel("Количество")
plt.legend();
```

In [46]:

```
#Средний возраст погибших и выживших
data.groupby('Survived')['Age'].mean()
```

Out[46]:

```
Survived
0    30.626179
1    28.343690
Name: Age, dtype: float64
```

8. Как отличается средний возраст мужчин / женщин в зависимости от класса обслуживания? Выберите верные утверждения:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

- В среднем мужчины 1-го класса старше 40 лет

- В среднем женщины 1-го класса старше 40 лет
- Мужчины всех классов в среднем старше женщин того же класса
- В среднем люди в 1 классе старше, чем во 2-ом, а те старше представителей 3-го класса

In [48]:
```
pd.crosstab(data['Pclass'], data['Sex'], values=data['Age'], aggfunc=np.mean)
```

Out[48]:

Sex	female	male
Pclass		
1	34.611765	41.281386
2	28.722973	30.740707
3	21.750000	26.507589