Федеральное государственное бюджетное образовательное учреждение высшего образования
«РЫБИНСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени П. А. СОЛОВЬЁВА»

Институт информационных технологий и систем управления

Кафедра математического и программного обеспечения электронных вычислительных средств

Лабораторная работа №1
"Знакомство с основами загрузки и обработки данных"

по дисциплине
Методы и алгоритмы анализа данных

Выполнили студенты группы ИВМ-22 — Перхуров В. А., Беляев А. Е.

Проверил: — Кулиманов И. Е.

Рыбинск 2022

Задание

Постановка задачи:

Цели:

Получить практические навыки по загрузке и обработке данных

Задачи:

1. Загрузить датасет "MNIST".
2. Вывести изображения нескольких случайных чисел.
3. Загрузить датасет "Titanic".
4. Выполнить ряд заданий по исследованию датасета:
   - Проанализируйте датасет и создайте несколько диаграмм/гистограмм и объясните что они отображают.
   - Заполните записи со значением NaN, объясните как заполняли.
   - Создайте дополнительный признак, объясните для чего он необходим.

1 Описание датасетов

1.1 Описание датасета MNIST

Файл данных содержит изображения нарисованных от руки цифр в оттенках серого от нуля до девяти.

Каждое изображение имеет высоту 28 пикселей и ширину 28 пикселей, всего 784 пикселя. С каждым пикселем связано одно значение пикселя, указывающее яркость или темноту этого пикселя, причем более высокие числа означают темнее. Это значение пикселя представляет собой целое число от 0 до 255 включительно.

Набор данных содержит 785 столбцов. Первый столбец, называемый «метка», представляет собой цифру, нарисованную пользователем. Остальные столбцы содержат значения пикселей связанного изображения.

Каждый столбец пикселей в наборе имеет имя, например pixelx, где x — целое число от 0 до 783 включительно. Чтобы найти этот пиксель на изображении, предположим, что мы разложили x как

$$x = i * 28 + j,$$

где i и j — целые числа от 0 до 27 включительно.

Затем pixelx находится в строке i и столбце j матрицы 28 x 28 (индексируется нулем).

Например, pixel31 указывает пиксель, который находится в четвертом столбце слева и во второй строке сверху.

1.2 Описание датасета Titanic

Набор данных представлен в CSV-файле. Набор содержит признак Survived для каждого пассажира, обозначающий, выжил данный пассажир или нет (0 для умерших, 1 для выживших).

Каждая строчка наборов данных содержит следующие поля:

1. Pclass — класс пассажира (1 — высший, 2 — средний, 3 — низший),
2. Name — имя,
3. Sex — пол,
4. Age — возраст,
5. SibSp — количество братьев, сестер, сводных братьев, сводных сестер, супругов на борту титаника,

6. Parch — количество родителей, детей (в том числе приемных) на борту титаника,
7. Ticket — номер билета,
8. Fare — плата за проезд,
9. Cabin — каюта,
10. Embarked — порт посадки (C — Шербур; Q — Квинстаун; S — Саутгемптон).

В поле Age приводится количество полных лет. Для детей меньше 1 года — дробное. Если возраст не известен точно, то указано примерное значение в формате xx.5.

2 Описание выполнения работы

2.1 Реализация программы для анализа датасета MNIST

2.1.1 Используемые библиотеки

При решении данной задачи были использованы следующие библиотеки:

1. random - генератор случайных чисел,
2. pandas - анализ данных,
3. numpy - математика и работа с массивами,
4. PIL.Image - работа с изображениями.

2.1.2 Ход работы и результат выполнения программы

В ходе выполнения работы была написана консольная программа, которя выводит несколько (задаваемое число) случайных чисел из загруженного датасета.

Пример работы данной программы показан на рисунке 1.

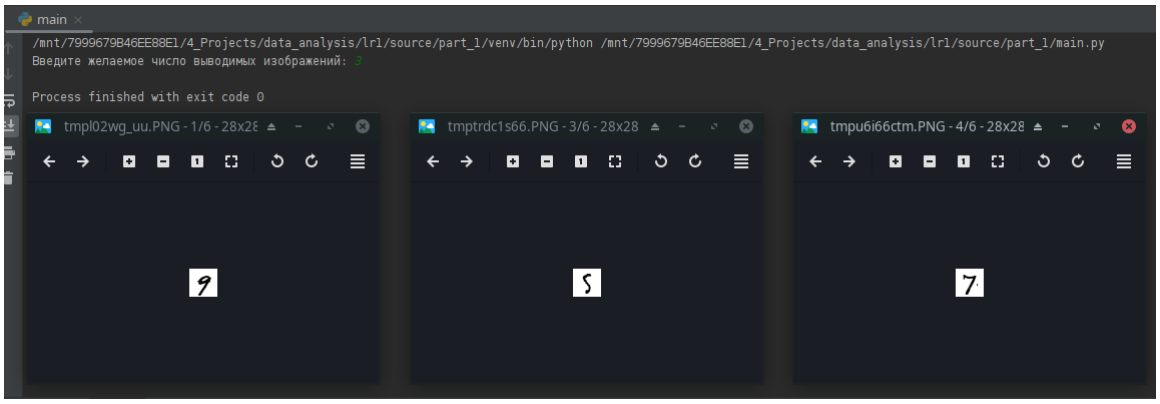

Рис. 1: Результат работы программы для вывода заданного числа изображений

2.1.3 Исходный текст программы

```
import random  # генератор случайных чисел
import pandas as pd  # анализ данных
import numpy as np  # математика и работа с массивами
from PIL import Image  # работа с изображениями

# 1 - Вывести заданное количество случайных чисел
def showImage(df):
```

```
    result_mass = np.zeros((28, 28), dtype=np.uint8)
    index = random.randint(0, 4200)
    for i in range(0, 784, 1):
        result_mass[i//28, i%28] = 255 - df[f"pixel{i}"].values[index]
    Image.fromarray(result_mass).show()

def main():
    # 0 - Читаем датасет
    df = pd.read_csv('./numbers.csv', escapechar='‘', low_memory=False)
    # 1 - Выводим заданное количество случайных чисел
    image_count = int(input("Введите желаемое число "
                            "выводимых изображений: "))
    for i in range(0, image_count, 1):
        showImage(df)

main()
```

2.2 Реализация программы для анализа датасета Titanic

2.2.1 Используемые библиотеки

При решении данной задачи были использованы следующие библиотеки:

1. pandas - анализ данных,
2. numpy - математика и работа с массивами,
3. matplotlib.pyplot - построение графиков.

2.2.2 Ход работы и результат выполнения программы

В ходе выполнения работы была написана программа, которя выводит три графика зависимостей (рисунок 2), построенных на основе данных из заданного датасета, и формирует дополнительный признак на основе имеющихся данных (рисунок 4).

На рисунке 2 показаны три графика (слева направо):

1. Зависимость числа пассажиров взошедших на борт от порта посадки.
2. Зависимость средней стоимости билетов и числа пассажиров от класса.
3. Зависимость выручки от класса

На рисунке 4 показан фрагмент изменённого датасета, где ячейки, содержащие значение NaN, были заполнены строкой "Unknown"с целью идентифкации факта неизвестности значения данного параметра для конкретной записи (пасажира). Замена производилось с помощью метода fillna.

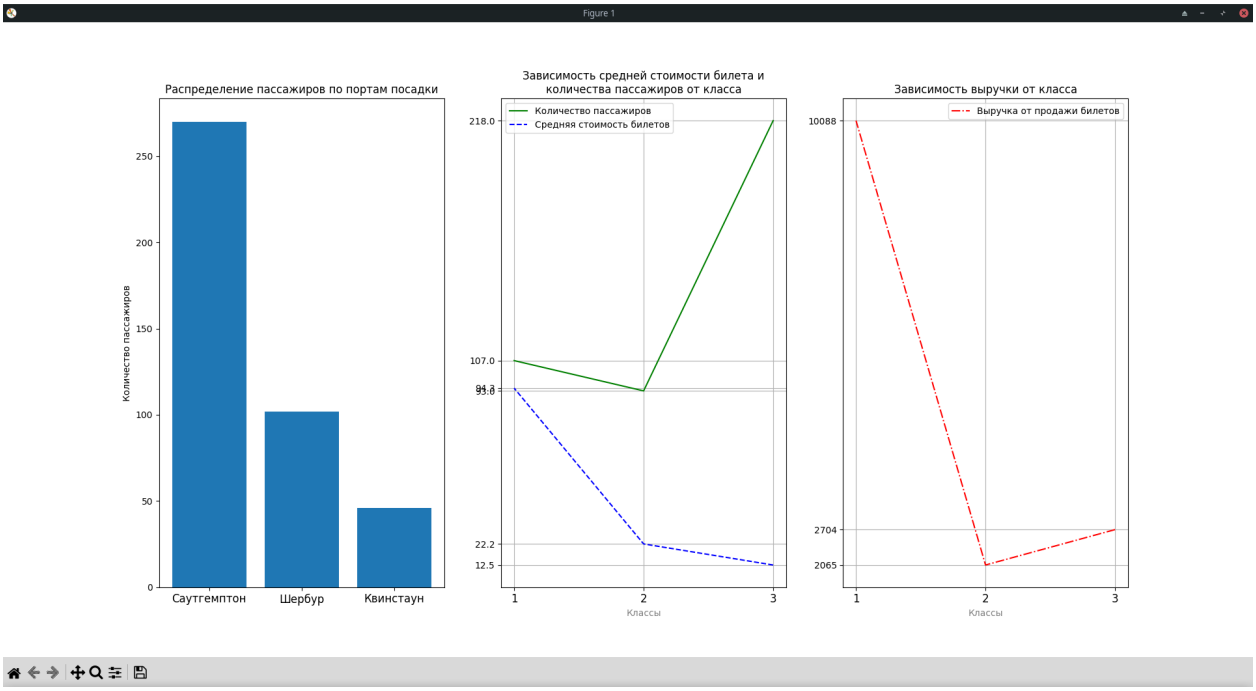

Рис. 2: Диаграммы, построенные при исполнении программы

Так же был добавлен дополнительный признак - буквенный код палубы. Введение нового параметра - буквенного кода палубы, позволит в дальнейшем не производить анализ номера каюты для определения палубы, с целью использования её в дальнейшей обработке.

New Open... titanic_modded.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Deck
2	0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	Unknown	Q	Unknown
3	1	893	1	3	Wilkes, Mrs. Jam	female	47.0	1	0	363272	7.0	Unknown	S	Unknown
4	2	894	0	2	Myles, Mr. Thom	male	62.0	0	0	240276	9.6875	Unknown	Q	Unknown
5	3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	Unknown	S	Unknown
6	4	896	1	3	Hirvonen, Mrs. A	female	22.0	1	1	3101298	12.2875	Unknown	S	Unknown
7	5	897	0	3	Svensson, Mr. Jo	male	14.0	0	0	7538	9.225	Unknown	S	Unknown
8	6	898	1	3	Connolly, Miss. K	female	30.0	0	0	330972	7.6292	Unknown	Q	Unknown
9	7	899	0	2	Caldwell, Mr. Alb	male	26.0	1	1	248738	29.0	Unknown	S	Unknown
10	8	900	1	3	Abrahim, Mrs. Jo	female	18.0	0	0	2657	7.2292	Unknown	C	Unknown
11	9	901	0	3	Davies, Mr. John	male	21.0	2	0	A/4 48871	24.15	Unknown	S	Unknown
12	10	902	0	3	Ilieff, Mr. Ylio	male	Unknown	0	0	349220	7.8958	Unknown	S	Unknown
13	11	903	0	1	Jones, Mr. Charl	male	46.0	0	0	694	26.0	Unknown	S	Unknown
14	12	904	1	1	Snyder, Mrs. Joh	female	23.0	1	0	21228	82.2667	B45	S	B
15	13	905	0	2	Howard, Mr. Ben	male	63.0	1	0	24065	26.0	Unknown	S	Unknown
16	14	906	1	1	Chaffee, Mrs. He	female	47.0	1	0	W.E.P. 5734	61.175	E31	S	E
17	15	907	1	2	del Carlo, Mrs. S	female	24.0	1	0	SC/PARIS 2167	27.7208	Unknown	C	Unknown
18	16	908	0	2	Keane, Mr. Danie	male	35.0	0	0	233734	12.35	Unknown	Q	Unknown
19	17	909	0	3	Assaf, Mr. Gerios	male	21.0	0	0	2692	7.225	Unknown	C	Unknown
20	18	910	1	3	Ilmakangas, Miss	female	27.0	1	0	STON/O2. 31012	7.925	Unknown	S	Unknown
21	19	911	1	3	Assaf Khalil, Mrs	female	45.0	0	0	2696	7.225	Unknown	C	Unknown
22	20	912	0	1	Rothschild, Mr. M	male	55.0	1	0	PC 17603	59.4	Unknown	C	Unknown
23	21	913	0	3	Olsen, Master. A	male	9.0	0	1	C 17368	3.1708	Unknown	S	Unknown
24	22	914	1	1	Flegenheim, Mrs	female	Unknown	0	0	PC 17598	31.6833	Unknown	S	Unknown
25	23	915	0	1	Williams, Mr. Ric	male	21.0	0	1	PC 17597	61.3792	Unknown	C	Unknown
26	24	916	1	1	Ryerson, Mrs. Ar	female	48.0	1	3	PC 17608	262.375	B57 B59 B63 B6	C	B
27	25	917	0	3	Robins, Mr. Alexa	male	50.0	1	0	A/5. 3337	14.5	Unknown	S	Unknown

Рис. 3: Результат внесения изменений в исходный датасет

2.2.3 Исходный текст программы

```
import pandas as pd # анализ данных
```

```
import numpy as np # математика и работа с массивами
from matplotlib import pyplot as plt # построение графиков

# Получить списки ключей и значений из отсортированного
# по убыванию словаря
def getKeysAndValues(dataset, with_sort):
    dict_dataset = dict(dataset)
    keys = []
    values = []
    if with_sort:
        dict_dataset = sorted(dict_dataset.items(),
                              key=lambda x: x[1],
                              reverse=True)
        for k, v in dict_dataset:
            keys.append(k)
            values.append(v)
    else:
        for k, v in dict_dataset.items():
            keys.append(k)
            values.append(v)
    return (keys, values)

# Отрисовать первую диаграмму.
# Зависимость числа пассажиров зашедших на борт от порта посадки.
# Зависимость средней стоимости билетов и числа пассажиров от класса.
# Зависимость выручки от класса.
def showDiagrams(main_df):
    # подсчёт числа пассажиров, взошедших на борт в городах
    boarding = pd.value_counts(main_df['Embarked'].values, sort=True)
    for i in range(0, len(boarding.index.values), 1):
        if boarding.index.values[i] == 'S':
            boarding.index.values[i] = 'Саутгемптон'
        else:
            if boarding.index.values[i] == 'C':
                boarding.index.values[i] = 'Шербур'
            else:
                boarding.index.values[i] = 'Квинстаун'

    boarding_keys, boarding_values = getKeysAndValues(boarding, True)
    top_boarding = len(boarding_keys)

    plt.subplot(131)
    plt.title('Распределение пассажиров по портам посадки')
    plt.bar(np.arange(top_boarding), boarding_values)
```

```
plt.xticks(np.arange(top_boarding),
           boarding_keys,
           rotation=0,
           fontsize=12)
plt.ylabel('Количество пассажиров')

# подсчёт числа пассажиров по классам
classes = pd.value_counts(main_df['Pclass'].values, sort=False)
classes_keys, classes_values = getKeysAndValues(classes, False)

# подсчёт средней стоимости билетов и общей выручки
average_fare = {}
revenue = {}
for i in classes_keys:
    dfs = main_df[['Fare', 'Pclass']].loc[main_df['Pclass'] == i]
    average_fare[i] = dfs['Fare'].mean(axis=0)
    revenue[i] = dfs['Fare'].sum()

average_fare_keys, average_fare_values =
    getKeysAndValues(average_fare, False)
revenue_keys, revenue_values = getKeysAndValues(revenue, False)

plt.subplot(132)
plt.grid(True)
plt.title('Зависимость средней стоимости билета и\n'
          'количества пассажиров от класса')
plt.xticks(classes_keys, rotation=0, fontsize=12)
plt.yticks(classes_values+average_fare_values)
plt.xlabel('Классы', color='gray')
plt.plot(classes_keys, classes_values, 'g',
         average_fare_keys, average_fare_values, 'b--')
plt.legend(['Количество пассажиров','Средняя стоимость билетов'],
           loc=2)

plt.subplot(133)
plt.grid(True)
plt.title('Зависимость выручки от класса')
plt.xticks(classes_keys, rotation=0, fontsize=12)
plt.yticks(revenue_values)
plt.xlabel('Классы', color='gray')
plt.plot(revenue_keys, revenue_values, 'r-.')
plt.legend(['Выручка от продажи билетов'], loc=1)
plt.show()
```

```
    return

# Заполнить поля значением Unknown.
# Ячейки, содержащие значение NaN, были заполнены строкой "Unknown"
# с целью идентифкации факта неизвестности
# значения данного параметра для конкретной записи (пасажира).
# Замена производилось с помощью метода fillna.
def fillNanFields(main_df):
    main_df = main_df.fillna('Unknown')
    return main_df

# Создать дополнительный признак (буквенного кода палубы).
# Введение нового параметра - буквенного кода палубы, позволит в
# дальнейшем не производить анализ номера кающы
# для определения палубы, с целью использования её в дальнейшей
# обработке.
def createAnAdditionalAttribute(main_df):
    main_df = main_df.assign(Deck=main_df.Cabin)
    for i in range(len(main_df.Deck)):
        if main_df.Deck[i] != 'Unknown':
            main_df.loc[i, 'Deck'] = main_df.loc[i, 'Deck'][:1]

    return main_df

# Основная функция
def main():
    # 0 - Читаем датасет
    df = pd.read_csv('./titanic.csv', escapechar='‘', low_memory=False)
    # 1 - Рисуем диаграммы зависимостей
    showDiagrams(df)
    # 3 - Заполняем поля значением Unknown
    df = fillNanFields(df)
    # 4 - Создаём дополнительный признак
    df = createAnAdditionalAttribute(df)
    # 5 - Сохраняем изменнеия в новый файл
    df.to_csv('./titanic_modded.csv')

# Запускаем программу
main()
```

3 Вывод.

В результате выполнения лабораторной работы были получены навыки по загрузке и начальной обработке данных на примере двух датасетов.

В ходе выполнения работы были написаны две программы (по одной на каждый датасет). Обе выполняют чтение датасетов из фалов в формате ".csv"и реализуют заданный функционал.

Первая (для датасета "MNIST") формирует заданное число случайно выбранных изображений и выводит их на экран.

Вторая (для датасета "Titanic") выполняет две операции:

1. Анализ данных из датасета и вывод их в виде графиков.

2. Изменение и добавление новых данных в имеющийся датасет и сохранение его в отдельный файл.