

Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Мегафакультет трансляционных и информационных технологий

Лабораторная работа №6

Выполнил(и:)

Жаров Александр Павлович

Проверил

Мусаев А.А.

Санкт-Петербург,

2022

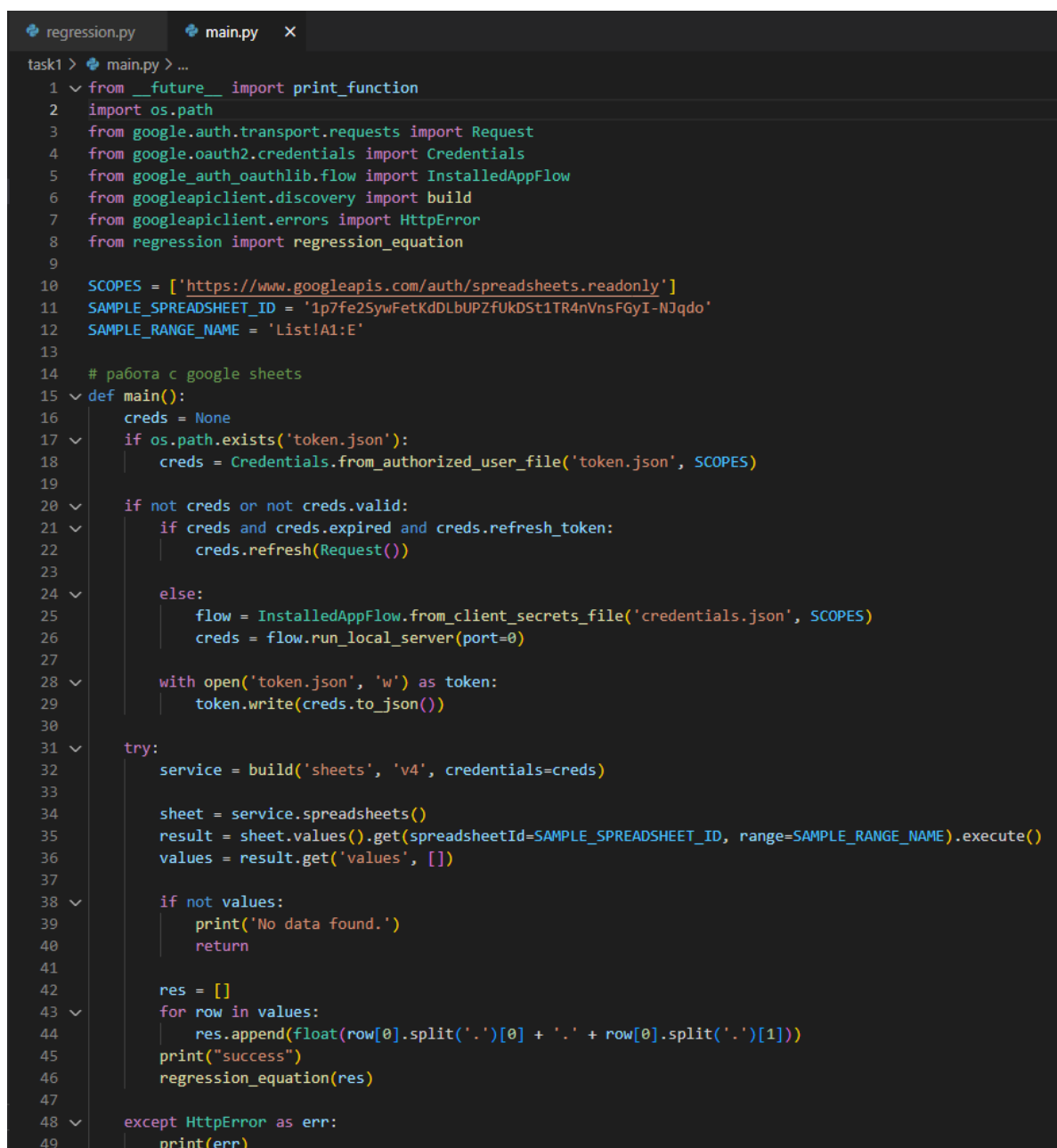
Задание 1

Текст задания

Имеется документ в google-таблицах, содержащий информацию о курсе любой валюты, выбранной студентом. Оценить, завышен или занижен её курс, используя авторегрессионную модель. Допускается использовать любые библиотеки.

Решение

Начинаем выполнение задание с создание проекта в google workspace, создания таблички в google sheets и подключения всего в коде (Рисунок 1). И выполняем чтение значений из таблички



```
task1 > main.py > ...
1  from __future__ import print_function
2  import os.path
3  from google.auth.transport.requests import Request
4  from google.oauth2.credentials import Credentials
5  from google_auth_oauthlib.flow import InstalledAppFlow
6  from googleapiclient.discovery import build
7  from googleapiclient.errors import HttpError
8  from regression import regression_equation
9
10 SCOPES = ['https://www.googleapis.com/auth/spreadsheets.readonly']
11 SAMPLE_SPREADSHEET_ID = '1p7fe2SywFetKdDLbUPZfUKDSt1TR4nVnsFGyI-NJqdo'
12 SAMPLE_RANGE_NAME = 'List!A1:E'
13
14 # работа с google sheets
15 def main():
16     creds = None
17     if os.path.exists('token.json'):
18         creds = Credentials.from_authorized_user_file('token.json', SCOPES)
19
20     if not creds or not creds.valid:
21         if creds and creds.expired and creds.refresh_token:
22             creds.refresh(Request())
23
24         else:
25             flow = InstalledAppFlow.from_client_secrets_file('credentials.json', SCOPES)
26             creds = flow.run_local_server(port=0)
27
28         with open('token.json', 'w') as token:
29             token.write(creds.to_json())
30
31     try:
32         service = build('sheets', 'v4', credentials=creds)
33
34         sheet = service.spreadsheets()
35         result = sheet.values().get(spreadsheetId=SAMPLE_SPREADSHEET_ID, range=SAMPLE_RANGE_NAME).execute()
36         values = result.get('values', [])
37
38         if not values:
39             print('No data found.')
40             return
41
42         res = []
43         for row in values:
44             res.append(float(row[0].split('.')[0] + '.' + row[0].split('.')[1]))
45             print("success")
46             regression_equation(res)
47
48     except HttpError as err:
49         print(err)
```

Рисунок 1

Затем запускаем эту функцию (рисунок 2)

```
if __name__ == '__main__':  
    main()
```

Рисунок 2.

Перейдем к расчету предсказания методом регрессии (рисунок 3).

```
1  import math  
2  import matplotlib.pyplot as plt  
3  import random  
4  
5  #поиск a  
6  def coef_a(x, y, b):  
7      a = 0  
8      for i in range(len(x)):  
9          a += y[i] - b * x[i]  
10     return a / len(x)  
11  
12 #поиск b  
13 def coef_b(x, y):  
14     avrgX = sum(x) / len(x)  
15     avrgY = sum(y) / len(y)  
16     num = 0  
17     denum = 0  
18  
19     for i in range(len(x)):  
20         num += (y[i] - avrgY) * (x[i] - avrgX)  
21         denum += math.pow((x[i] - avrgX), 2)  
22  
23     return num / denum  
24  
25  
26 #регрессия  
27 def reg(data):  
28     x = data[:-1]  
29     y = data[1:]  
30     b = coef_b(x, y)  
31     a = coef_a(x, y, b)  
32     res = []  
33  
34     for i in data:  
35         rand = random.randint(0, 4)  
36         if rand % 2 == 0:  
37             res.append(a + b * i + random.uniform(0, 0.01 * i))  
38         else:  
39             res.append(a + b * i - random.uniform(0, 0.01 * i))  
40  
41     fig, ax = plt.subplots()  
42     ax.plot(data)  
43     ax.plot(res)  
44     plt.show()
```

Рисунок 3.

Регрессия рассчитывается по следующей формуле $Y_i = \alpha + \beta \cdot Y_{i-1} + \epsilon$. Для начала нам понадобится определить коэффициенты авторегрессии α и β , которые обуславливают скорость изменения и коэффициент зависимости двух соседних значений ряда. Так же для расчета нам понадобится случайное число, которое необходимо для более правдоподобных результатов. В функции `reg` мы производим подсчет предсказанных значений и выводим графики на экран (Рисунок 4).

Figure 1

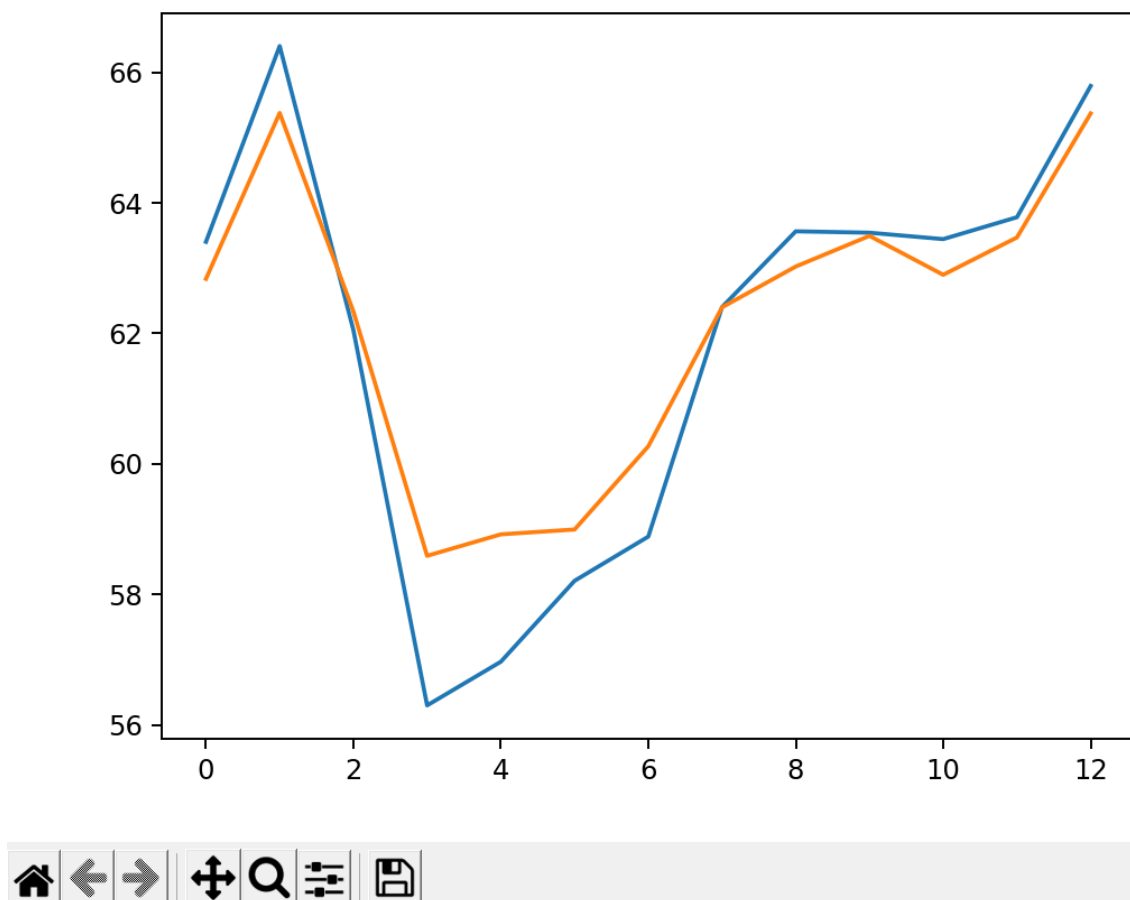


Рисунок 4

Задание 2

Текст задания

Имеется оформленный шаблон в таблицах google, содержащий описание и пустое поле для некоторых переменных (x , y , z , ...) и функции ($f(x, y, z, \dots)$). Пример: Пользователь задает созданном интерфейсе название PDF-файла и переменные, после чего происходит расчет функций и все данные сохраняются в заранее подготовленной форме в гугл-таблицах. Аналогичная таблица сохраняется в виде PDF-файла, название которого содержит дату создания файла (ГГГГ-ММ-ДД) и название, заданное в интерфейсе (пример: «2022-02-02 – кинетическая энергия первый расчет», «2022-02-03 – кинетическая энергия второй расчет»)

Решение

Сначала по аналогии с первым заданием подключим гугл табличку (Рисунок 5) с шаблоном (Рисунок 6)

```
4 from google.oauth2.credentials import Credentials
5 from google_auth_oauthlib.flow import InstalledAppFlow
6 from googleapiclient.discovery import build
7 from googleapiclient.errors import HttpError
8 from interface import main_screen
9
10 SCOPES = ['https://www.googleapis.com/auth/spreadsheets.readonly']
11 SAMPLE_SPREADSHEET_ID = '111gVm1cFct2tOPYUxcWTsd0kXhdzdZSnVH0L8oAtX6M'
12 SAMPLE_RANGE_NAME = 'List!A1:E'
13
14 # работа с google sheets
15 def main():
16     creds = None
17
18     if os.path.exists('token.json'):
19         creds = Credentials.from_authorized_user_file('token.json', SCOPES)
20
21     if not creds or not creds.valid:
22         if creds and creds.expired and creds.refresh_token:
23             creds.refresh(Request())
24         else:
25             flow = InstalledAppFlow.from_client_secrets_file(
26                 'credentials.json', SCOPES)
27             creds = flow.run_local_server(port=0)
28
29     with open('token.json', 'w') as token:
30         token.write(creds.to_json())
31
32     try:
33         service = build('sheets', 'v4', credentials=creds)
34
35         sheet = service.spreadsheets()
36         result = sheet.values().get(spreadsheetId=SAMPLE_SPREADSHEET_ID, range=SAMPLE_RANGE_NAME).execute()
37         values = result.get('values', [])
38
39         if not values:
40             print('No data found.')
41             return
42
43         res = []
44         for row in values:
45             res.append(row[0])
46         main_screen(res)
47     except HttpError as err:
48         print(err)
49
50 if __name__ == '__main__':
51     main()
```

Рисунок 5

E		Кинематическая	Кинематическая	0
m		Масса		
v		Скорость		2022-06-22
				<input type="text"/>

Рисунок 6

Затем с помощью Tkinter создаем интерфейс для ввода значений (рисунок 7), необходимые поля берем из таблички (рисунок 8).

```

1  from tkinter import *
2  from fpdf import FPDF
3  from datapy import main as send_to_sheet
4  import datetime
5
6  copy_data = []
7  button_data = []
8
9  def build():
10     res = []
11     for i in range(len(button_data)):
12         if i != len(button_data) - 1:
13             res.append(button_data[i].get())
14     send_to_sheet(res)
15     create_file(copy_data, res, button_data[-1].get())
16
17 #верстка основного экрана
18 def main_screen(data):
19     global copy_data
20     copy_data = data
21     window = Tk()
22     window.geometry("800x600")
23     window.title("Задание 2")
24
25     for i in data:
26         string = Frame(window)
27         string.pack()
28
29         label = Label(string, text=i, padx=5, pady=5)
30         label.pack(side=LEFT)
31
32         entry = Entry(string, width=15)
33         entry.pack(side=LEFT)
34         button_data.append(entry)
35
36     string_file_name = Frame(window)
37     string_file_name.pack()
38
39     label = Label(string_file_name, text="Название pdf", padx=5, pady=5)
40     label.pack(side=LEFT)
41
42     entry = Entry(string_file_name, width=15)
43     entry.pack(side=LEFT)
44     button_data.append(entry)
45
46     string_last = Frame(window)
47     string_last.pack()
48     Button(string_last, text="Посчитать", command=build, padx=5, pady=5).pack(side=LEFT)
49

```

Рисунок 7

Задание 2

E

m

v

Название pdf

Рисунок 8

Далее при нажатии на кнопку “Посчитать” выполняется функция build в которой значения записываются в табличку и вызывается функция для создания файла (рисунок 9).

```
#создание таблицы с pdf
def create_file(data, res, name):
    table = []
    for i in range(len(data)):
        table.append([str(data[i]), str(res[i])])

    pdf = FPDF()
    pdf.set_font("Arial", size=12)
    pdf.add_page()

    col_width = pdf.w / 4.5
    row_height = pdf.font_size
    for row in table:
        for item in row:
            pdf.cell(col_width, row_height,
                    txt=item, border=1)
        pdf.ln(row_height)

    pdf.output(str(datetime.datetime.now()).split()[0] + " " + name + ".pdf")
```

Рисунок 9

Файл pdf создается при помощи библиотеки fpdf и сохраняется с заданным названием (Рисунок 10).

E	134
m	23
v	5

Рисунок 10

Задание 3

Текст задания

Создать интерфейс, который предлагает выбрать и открыть ранее созданные PDF файлы (директория задается заранее в программе). Пользователь видит название файла без даты (пример: «кинетическая энергия первый расчет», «кинетическая энергия второй расчет»).

Решение

Сперва создаем интерфейс выбора директории (Рисунок 12) и туда же добавляем найденные файлы pdf с кнопкой “Открыть”. Для этого сразу же создадим удобный класс Button_Class (Рисунок 11)

```
1  from tkinter import *
2  from functions import find
3  from functions import open
4  directory = None
5  window = None
6
7  #задаем каждой кнопке имя файла
8  class Button_Class:
9      def __init__(self, string, name):
10         self.name = name
11         self.button = Button(string, text="Открыть", command= self.open_file, padx=5, pady=5).pack(side=LEFT)
12
13     def open_file(self):
14         open(self.name)
15
16 def build():
17     name = directory.get()
18     window.destroy()
19     main(find(name))
20
21 def main(array):
22     global directory
23     global window
24     window = Tk()
25     window.geometry("800x600")
26     window.title("Найти pdf")
27
28     str1 = Frame(window)
29     str2 = Frame(window)
30
31     str1.pack()
32     str2.pack()
33
34     Label(str1, text="Введите другую директорию ", padx=5, pady=5).pack(side=LEFT)
35
36     for i in array:
37         string = Frame(window)
38         string.pack()
39         Label(string, text=i, padx=5, pady=5).pack(side=LEFT)
40         Button_Class(string, i)
41
42     directory = Entry(str1, width=15)
43     directory.pack(side=LEFT)
44
45     Button(str2, text="Изменить", command=build, padx=5, pady=5).pack(side=LEFT)
46
47     window.mainloop()
48
49 main([])
```

Рисунок 11

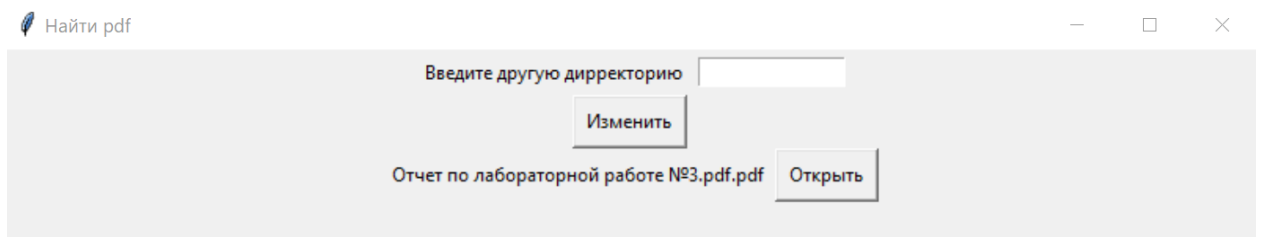


Рисунок 12

Далее прописываем функции для поиска файлов pdf и открытия файлов (Рисунок 13).

```
1  import os
2  import webbrowser
3
4  #поиск pdf в указанной директории
5  def find(directory):
6      res = []
7      if directory:
8          os.chdir(directory)
9
10         for f in os.listdir():
11             if f.endswith(".pdf"):
12                 res.append(os.path.join(f))
13
14         return res
15
16     #открытие pdf
17     def open(name):
18         webbrowser.open_new(name)
19
20
```

Рисунок 13

Для открытия файлов будем использовать библиотеку webbrowser, а для поиска файлов в директории библиотеку os. Каждый раз когда мы меняем директорию, главный экран перезапускается уже с найденными в pdf из новой директории.

Вывод

Были освоены новые библиотеки для работы с файлами, закреплены навыки работы с google sheets и разработана авторегрессивная модель для предсказания значений курса валюты.

Ссылка на гит

1. <https://github.com/SashaZharov/programming-lab6>