Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

**Лабораторная работа №6**

**Вариант №5**

Выполнил:

Олейникова П.Л.

Проверил:

Мусаев А.А.

Санкт-Петербург,

2022

**Задача №1**

Гитхаб:

https://github.com/OleinikovaPolina/university/tree/main/Python-Programming/Lab-6.

Задание.

Имеется документ в google-таблицах, содержащий информацию о курсе любой валюты, выбранной студентом. Оценить, завышен или занижен её курс, используя авторегрессионную модель [1]. Допускается использовать любые библиотеки.

Решение.

import gspread  
from oauth2client.service\_account import ServiceAccountCredentials  
import matplotlib.pyplot as plt  
from datetime import datetime  
  
  
def regression\_equation(data, dates):  
 b = search\_b(data)  
 a = sum(data) / len(data) \* (1 - b)  
 forecast = []  
 for i in data:  
 forecast.append(a + b \* i)  
   
 fig, ax = plt.subplots()  
 ax.plot(dates, data)  
 ax.plot(dates, forecast, 'o--')  
 plt.show()  
  
  
def search\_b(data):  
 y1 = sum(data[1:]) / len(data[1:])  
 y2 = sum(data[:-1]) / len(data[:-1])  
 numerator = 0  
 denominator1 = 0  
 denominator2 = 0  
 for i in range(1, len(data)):  
 numerator += (data[i] - y1) \* (data[i - 1] - y2)  
 denominator1 += (data[i] - y1) \*\* 2  
 denominator2 += (data[i - 1] - y2) \*\* 2  
 return numerator / (denominator1 \* denominator2) \*\* 0.5  
  
  
def google\_table():  
 # https://docs.google.com/spreadsheets/d/1ucR-ZxJ8MmW26QvipwZN0o2-OrQOUhkKcpLS0vvHoM0/edit?usp=sharing  
 scope = [  
 'https://www.googleapis.com/auth/spreadsheets',  
 'https://www.googleapis.com/auth/drive'  
 ]  
 credentials = ServiceAccountCredentials.from\_json\_keyfile\_name('python-341913-2d5e4a16dd7e.json', scope)  
 gc = gspread.authorize(credentials)  
 wks = gc.open("Python").worksheet('Лист2')  
  
 dates = list(map(lambda o: datetime.strptime(o, '%d.%m.%Y'), wks.col\_values(1)))  
 y = list(map(lambda o: float(o.replace(',', '.')), wks.col\_values(2)))  
 regression\_equation(y, dates)  
  
 dates = list(map(lambda o: datetime.strptime(o, '%d.%m.%Y'), wks.col\_values(4)))  
 y = list(map(lambda o: float(o.replace(',', '.')), wks.col\_values(5)))  
 regression\_equation(y, dates)  
  
  
google\_table()

Код для 1 задачи

В данной программе реализованы несколько функций:

* google\_table: с помощью библиотеки gspread были взяты данные из google таблицы (даты и курс), здесь же вызывается следующая функция;
* get\_regression\_equation: здесь получаются коэффициент авторегрессии. Простейшим примером является авторегрессионный процесс первого порядка AR(1)-процесс. Для данного процесса коэффициент авторегрессии совпадает с коэффициентом автокорреляции первого порядка;
* search\_b: считается коэффициентом автокорреляции первого порядка.

Ниже приведены примеры для работы программы (Рисунок 1, 2, 3).

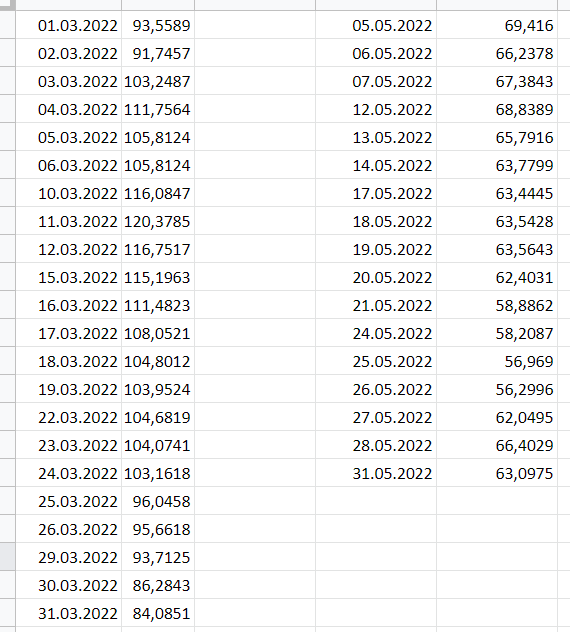


Рисунок 1 – исходные данные для программы

В первой и второй колонке приведен курс доллара за март, в четвертой и пятой – за май.

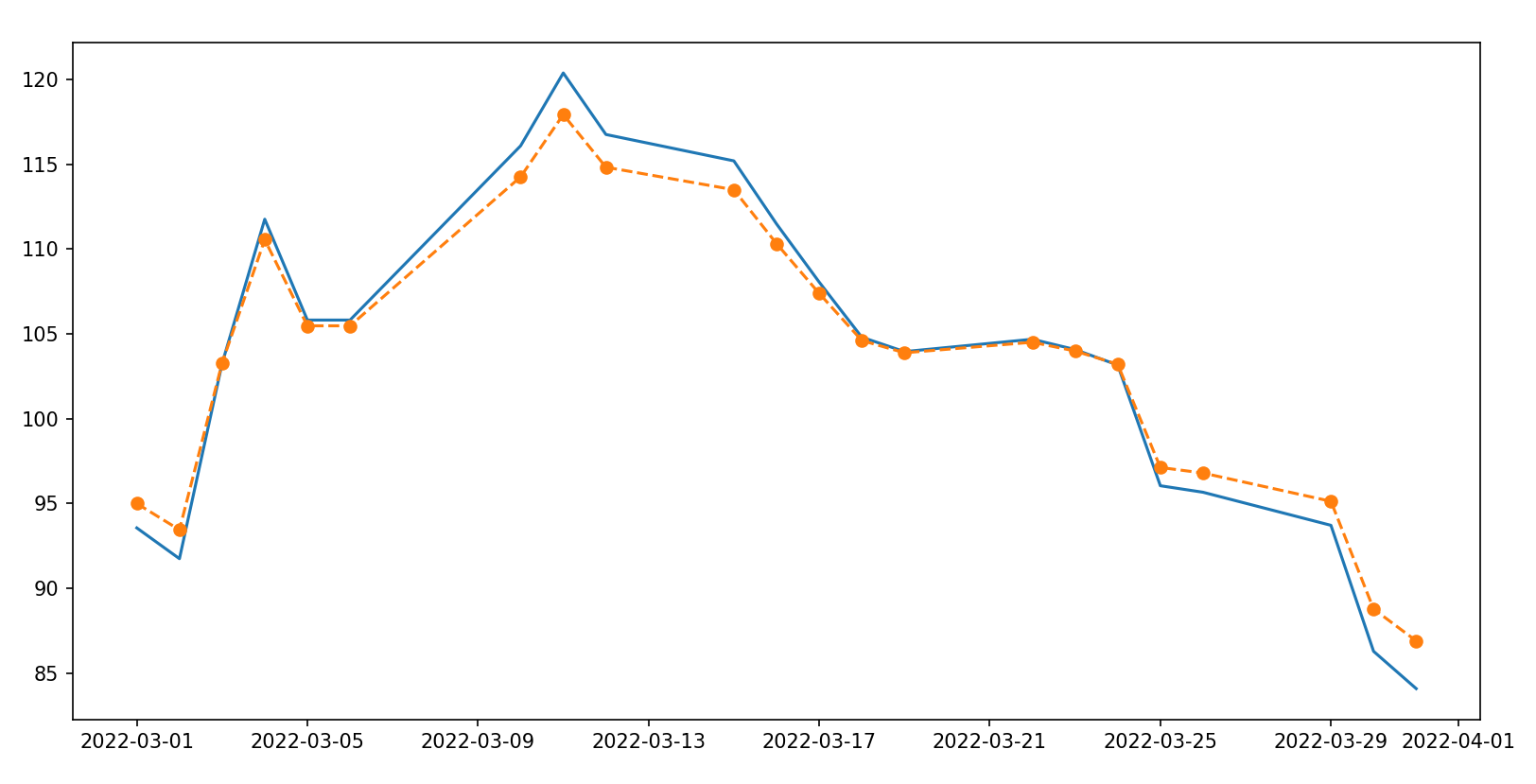


Рисунок 2 – Результат работы программы для марта

Синим цветом показаны исходные данные, рыжим – спрогнозированные. В данном случае, когда курс достиг пика, он был завышен.

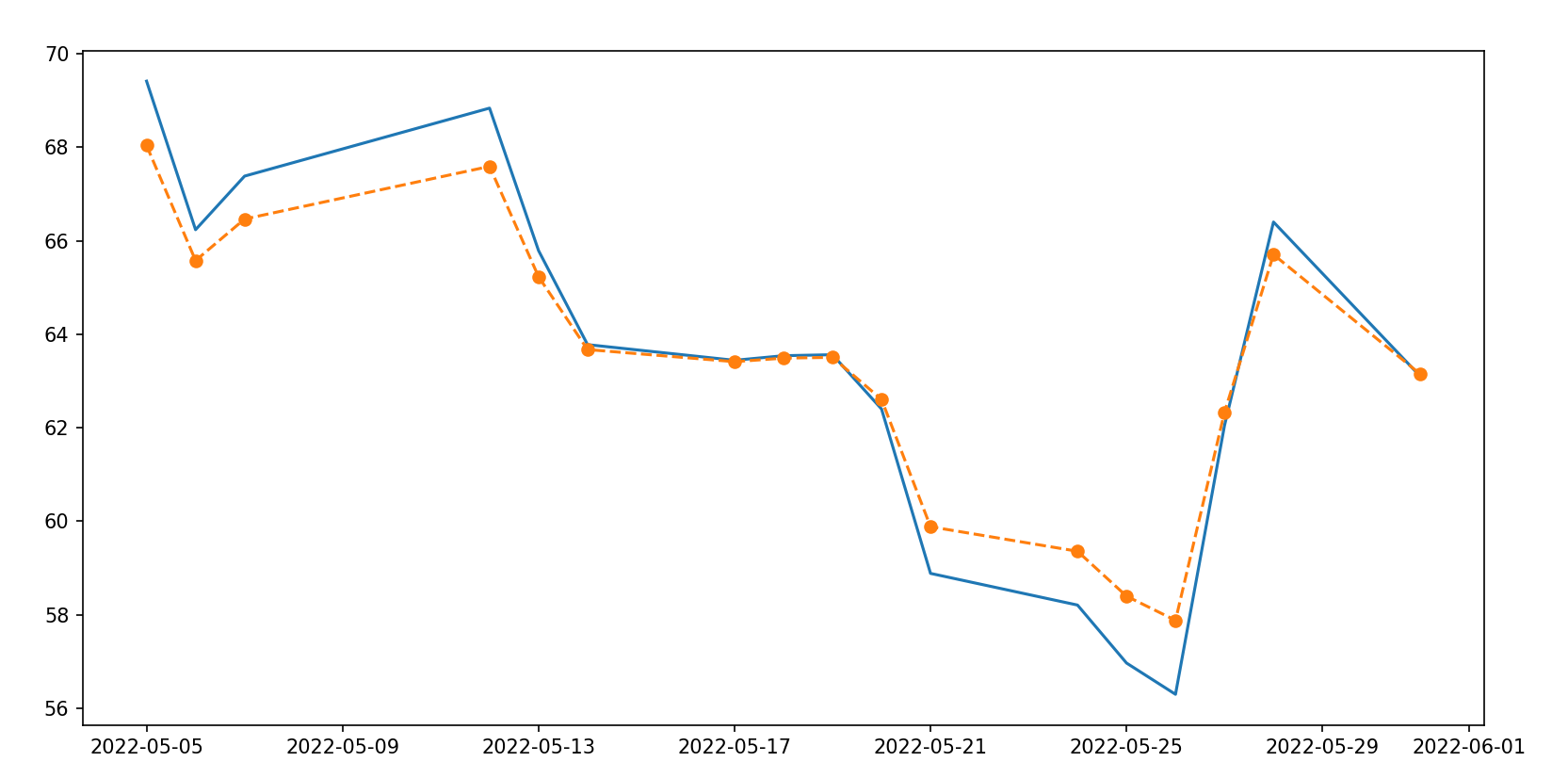


Рисунок 3 – Результат работы программы для мая

В данном случае, в начале месяца курс был завышен, перед концом - занижен.

Вывод по задаче. В результате данной работы была создана программа, позволяющая по данным из google таблицы о курсе валюты наглядно увидеть завышен ли он или занижен. Также была проведена работа с библиотеками matplotlib [2], gspread [3] и другими.

**Задача №2**

Задание.

Имеется оформленный шаблон в таблицах google, содержащий описание и пустое поле для некоторых переменных (x, y, z, …) и функции (f(x, y, z, …)). 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.

import gspread  
from oauth2client.service\_account import ServiceAccountCredentials  
from dash import Dash, dcc, html  
from dash.dependencies import Input, Output, State  
from datetime import datetime  
from fpdf import FPDF  
  
  
def google\_table(m, v, s):  
 # https://docs.google.com/spreadsheets/d/1ucR-ZxJ8MmW26QvipwZN0o2-OrQOUhkKcpLS0vvHoM0/edit?usp=sharing  
 scope = [  
 'https://www.googleapis.com/auth/spreadsheets',  
 'https://www.googleapis.com/auth/drive'  
 ]  
 credentials = ServiceAccountCredentials.from\_json\_keyfile\_name('python-341913-2d5e4a16dd7e.json', scope)  
 gc = gspread.authorize(credentials)  
 wks = gc.open("Python").sheet1  
 wks.update('A1', s)  
 wks.update('B1', datetime.today().strftime('%Y-%m-%d'))  
 wks.update('B4', m)  
 wks.update('B5', v)  
  
 create\_pdf(wks.get\_all\_values(), datetime.today().strftime('%Y-%m-%d') + ' - ' + s)  
  
  
def create\_pdf(text, filename):  
 font\_size\_pt = 14  
 font\_size\_mm = font\_size\_pt \* 0.5  
  
 pdf = FPDF(orientation='P', unit='mm', format='A4')  
 pdf.set\_auto\_page\_break(True, margin=14)  
 pdf.add\_page()  
 pdf.set\_font(family='Times', size=font\_size\_pt)  
  
 for line in text:  
 if len(line) == 0:  
 pdf.ln()  
 else:  
 pdf.cell(0, font\_size\_mm, ' '.join(line), ln=1)  
  
 pdf.output('pdfs/' + filename + '.pdf', 'F')  
  
  
app = Dash(\_\_name\_\_)  
  
app.layout = html.Div([  
 html.P('Расчет кинетической энергии\n', style={'margin-bottom': 0}),  
 dcc.Input(  
 id='input\_n',  
 placeholder="масса",  
 type='number',  
 style={'width': 115},  
 min=1  
 ),  
 dcc.Input(  
 id='input\_m',  
 placeholder="скорость",  
 type='number',  
 style={'width': 115},  
 min=1  
 ),  
 html.Br(),  
 dcc.Input(  
 id='input\_s',  
 placeholder="название файла",  
 style={'width': 230},  
 ),  
 html.Br(),  
 html.Button('Рассчитать', id='get\_res', n\_clicks=0),  
 html.Div(id='res'),  
])  
  
  
@app.callback(  
 Output('res', 'children'),  
 Input('get\_res', 'n\_clicks'),  
 State('input\_n', 'value'),  
 State('input\_m', 'value'),  
 State('input\_s', 'value')  
)  
def update\_output(\_, input\_n, input\_m, input\_s):  
 if input\_n and input\_m and input\_s:  
 try:  
 google\_table(input\_n, input\_m, input\_s)  
 return 'Файл сохранен'  
 except ValueError:  
 return 'Ошибка'  
 return 'Введите значения'  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run\_server(debug=True)

Код для 2 задачи

В данной программе реализовано:

* построение интерфейса с помощью библиотеки dash [4] с полями ввода данных и вывода;
* функция update\_output, вызывающая основную и сообщающая пользователю сообщение;
* функция google\_table, записывающая переданные ей значения в таблицу и вызывающая функцию создания pdf документа в которую передает все данные из таблицы;
* функция create\_pdf, записывающая всю информацию в документ и сохраняющая его.

Ниже приведены примеры для работы программы (Рисунок 4, 5, 6).

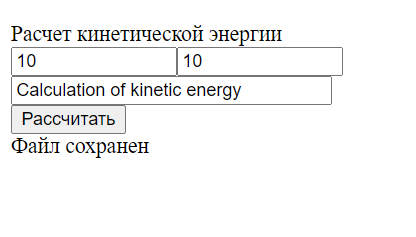


Рисунок 4 – Интерфейс программы

Вводим значения, нажимаем на кнопку, получаем ответ через какое-то время. Название файла необходимо вводить на английском.

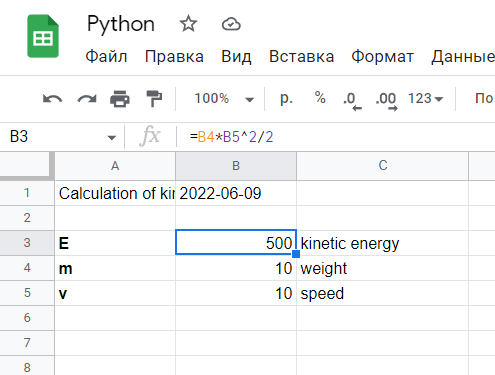


Рисунок 5 – Google таблица с введенными значениями

В данном случае происходит расчет кинетической энергии через массу и скорость.

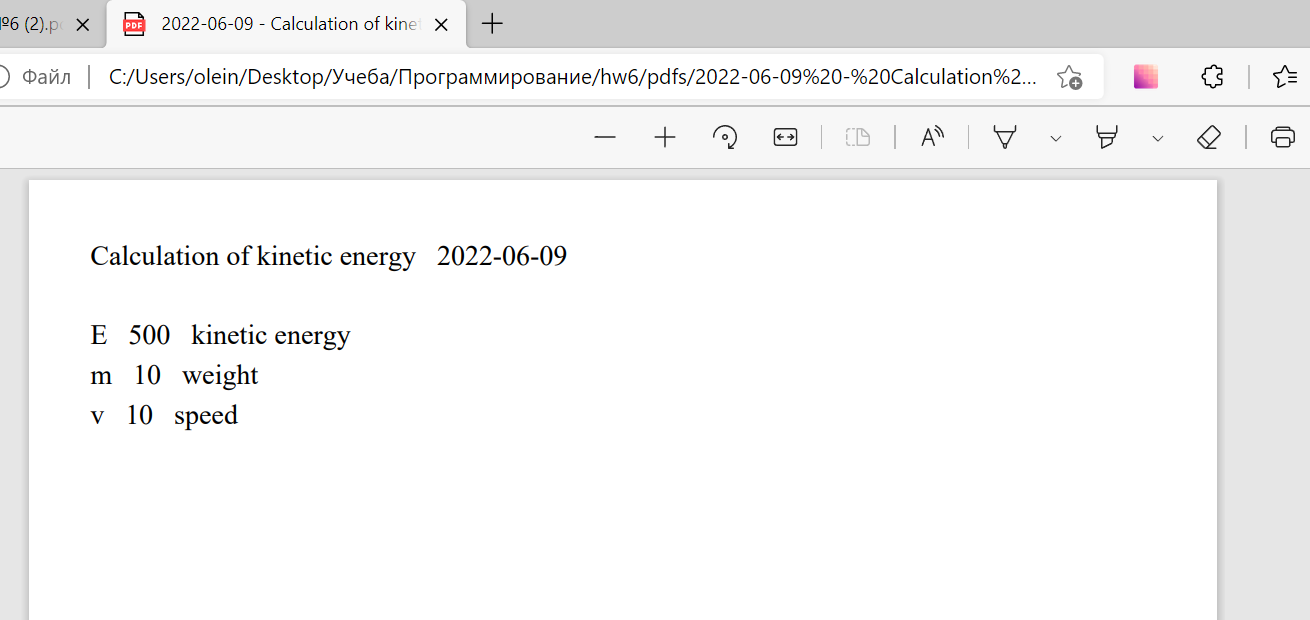


Рисунок 5 – сохраненный pdf документ

В документе находится вся информация из таблицы.

Вывод по задаче. В результате данной работы был создан интерфейс позволяющий пользователю удобно ввести данные и получить pdf файл с результатами расчетов, приведенных в google таблице. Также была проведена работа с библиотеками gspread[3], dash[4], fpdf[5].

**Задача №3**

Задание.

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.

from dash import Dash, dcc, html  
from dash.dependencies import Input, Output, State  
import os  
import webbrowser  
  
directory = 'C:/Users/olein/Desktop/Учеба/Программирование/hw6/pdfs/'  
  
def get\_files():  
 dir\_name = 'pdfs'  
 files = os.listdir(dir\_name)  
 options = []  
 for file in files:  
 options.append({'label': " - ".join(file.split(' - ')[1:]), 'value': file})  
 return options  
  
  
app = Dash(\_\_name\_\_)  
  
app.layout = html.Div([  
 html.P('Файлы\n'),  
 dcc.Dropdown(get\_files(), '', id='files'),  
 html.Button('Открыть', id='get\_res', n\_clicks=0),  
 html.Div(id='res'),  
])  
  
  
@app.callback(  
 Output('res', 'children'),  
 Input('get\_res', 'n\_clicks'),  
 State('files', 'value')  
)  
def update\_output(\_, input\_n):  
 if input\_n:  
 webbrowser.open\_new(directory + input\_n)  
 return ''  
 return 'Выберите файл'  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run\_server(debug=True)

Код для 3 задачи

В данной программе реализовано:

* построение интерфейса с помощью библиотеки dash [4] с полями ввода данных и вывода, кнопкой;
* функция получения имен файлов, удаляющая даты;
* функция открытия в браузере выбранного файла или вывода сообщения об ошибке.

Ниже приведены примеры для работы программы (Рисунок 7).

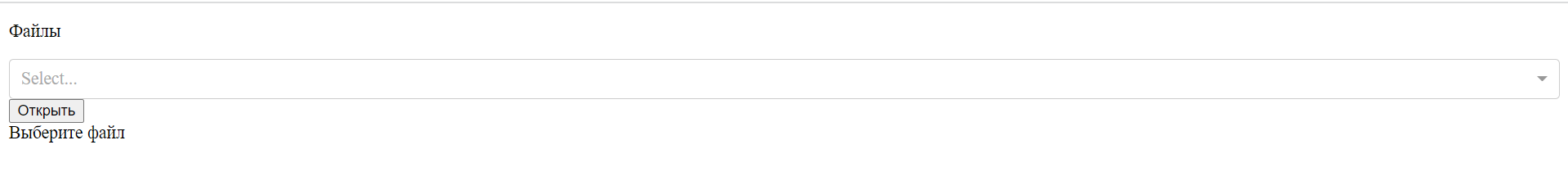
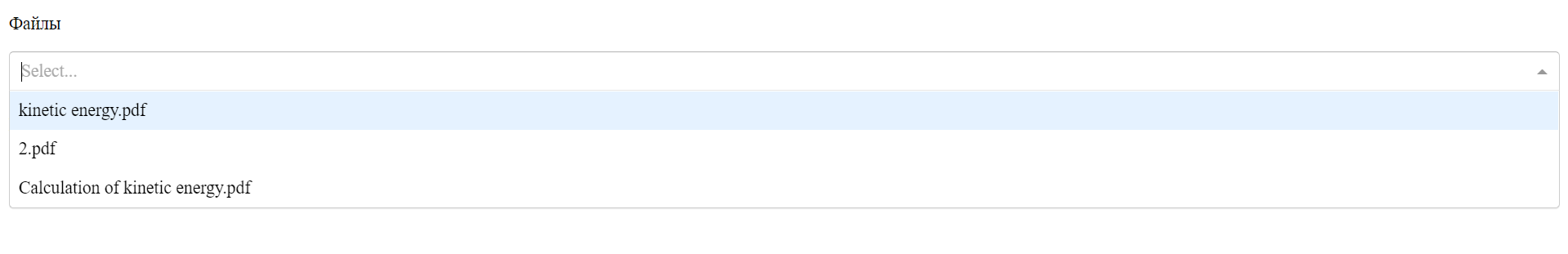
 

Рисунок 1 –интерфейс программы

Вывод по задаче. В результате данной работы был создан интерфейс для выбора и открытия созданных файлов с помощью библиотек dash [4] и других. В данной задаче важно, чтобы имена фалов были определенного стандарта.

**СПИСОК ЛИТЕРАТУРЫ**

1. Wikipedia. Авторегрессионная модель [Электронный ресурс] – https://ru.wikipedia.org/wiki/Авторегрессионная\_модель (09.06.2022);
2. Matplotlib: Visualization with Python. [Электронный ресурс] – <https://matplotlib.org/> (09.06.2022);
3. Gspread. [Электронный ресурс] – <https://docs.gspread.org/en/latest/> (21.02.2022);
4. Plotly. Dash Python User Guide. [Электронный ресурс] – <https://dash.plotly.com/> (09.06.2022);
5. PyFPDF. FPDF for Python. [Электронный ресурс] – <https://pyfpdf.readthedocs.io/en/latest/> (09.06.2022).