**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и сетевые технологии**

Направление подготовки (специальность) **09.03.03 Прикладная информатика**

Лабораторная работа №6 Вариант №11

Выполнили:

Самаров И.И.

Клейменов В.В.

Проверил: Мусаев А.А.

Санкт-Петербург, 2022

**Задача №1**

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  
import matplotlib.pyplot as plt  
from datetime import datetime  
  
def autoregress(data, dates):  
 b = coeff(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()  
 print("\_\_\_\_\_\_\_\_")  
  
def coeff(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 start():  
 sa = gspread.service\_account(filename="token.json")  
 sheet = sa.open("5task")  
 worksheet = sheet.worksheet("kurs")  
  
 dates = list(map(lambda o: datetime.strptime(o, '%d.%m.%Y'), worksheet.col\_values(1)))  
 y = list(map(lambda o: float(o.replace(',', '.')), worksheet.col\_values(2)))  
 autoregress(y, dates)  
  
 dates = list(map(lambda o: datetime.strptime(o, '%d.%m.%Y'), worksheet.col\_values(4)))  
 y = list(map(lambda o: float(o.replace(',', '.')), worksheet.col\_values(5)))  
 autoregress(y, dates)  
  
  
start()

Код первой задачи

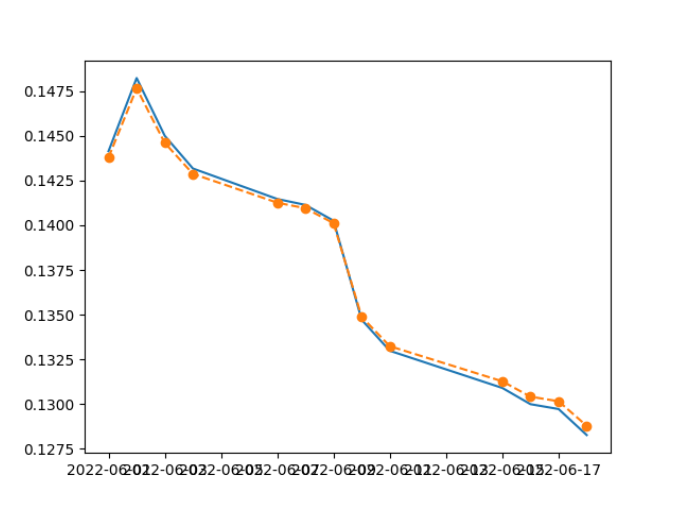
В данной программе реализованы несколько функций:

* start: берутся значение из таблицы
* autoregress: здесь вычисляем коэффициент авторегрессии. Простейшим примером является авторегрессионный процесс первого порядка AR(1)-процесс. Для данного процесса коэффициент авторегрессии совпадает с коэффициентом автокорреляции первого порядка;
* coeff: считается коэффициентом автокорреляции первого порядка.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1 – курс тенге за май и июнь



Рисункок 2 - График за май

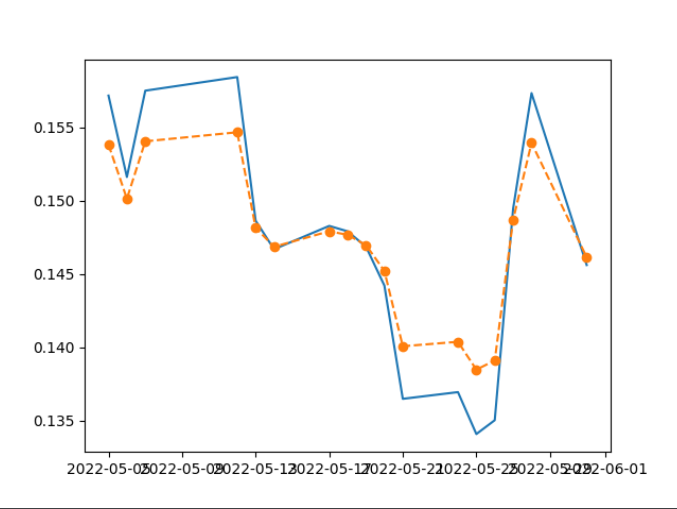


Рисунок 3 – График за июнь

Вывод по задаче.

Рассматривая графики за май, можно установить, что в начале периода(на пике) курс был завышен, относительно прогнозируемого, в конце же наоборот занижен.

Рассматривая графики за июнь, можно установить, что на пиках курс был достаточно сильно завышен, относительно прогнозируемого, а на спадах наоборот, был сильно занижен

В результате данной работы была проведена работа с библиотеками matplotlib[3] и gspread[4].

**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.

from kivy.app import App  
from kivy.uix.button import Button  
from kivy.uix.label import Label  
from kivy.uix.gridlayout import GridLayout  
from kivy.uix.image import AsyncImage  
from kivy.uix.textinput import TextInput  
from kivy.core.window import Window  
from datetime import datetime  
from fpdf import FPDF  
from kivy.uix.dropdown import DropDown  
import gspread  
class Application(App):  
 def build(self):  
 Window.clearcolor = (16 / 255, 71 / 255, 140 / 255, 0);  
 self.grid = GridLayout(cols=2, padding=[30], spacing=10)  
 self.noth1 = Label()  
 self.label = Label(text="Введите данные", font\_size=30)  
 self.speed = TextInput(text = "Введите скорость")  
 dropdown = DropDown()  
  
 for ingrediant in ["g","kg","ton"]:  
 btn = Button(text=ingrediant, size\_hint\_y=None, color="cyan",height = 25)  
 btn.bind(on\_release=lambda btn: dropdown.select(btn.text))  
 dropdown.add\_widget(btn)  
 self.trigger = Button(text="Выберите единицы измерения", background\_color=(255, 207, 0, 200 / 255))  
 self.trigger.bind(on\_release=dropdown.open)  
 dropdown.bind(on\_select=lambda instance, x: setattr(self.trigger, 'text', x))  
  
 dropdown1 = DropDown()  
  
 for ingrediant in ["km\h", "km/s", "m/h","m/s"]:  
 btn = Button(text=ingrediant, size\_hint\_y=None, color="cyan", height=25)  
 btn.bind(on\_release=lambda btn: dropdown1.select(btn.text))  
 dropdown1.add\_widget(btn)  
 self.trigger1 = Button(text="Выберите единицы измерения", background\_color=(255, 207, 0, 200 / 255))  
 self.trigger1.bind(on\_release=dropdown1.open)  
 dropdown1.bind(on\_select=lambda instance, x: setattr(self.trigger1, 'text', x))  
  
  
 self.m = TextInput(text="Введите массу")  
 self.filename = TextInput(text="Введите название файла")  
 self.but = Button(text="Создать PDF", font\_size=30, background\_color=(255, 207, 0, 200 / 255),  
 on\_press=self.go)  
 self.grid.add\_widget(self.label)  
 self.grid.add\_widget(self.noth1)  
 self.grid.add\_widget(self.m)  
 self.grid.add\_widget(self.trigger)  
  
 self.grid.add\_widget(self.speed)  
 self.grid.add\_widget(self.trigger1)  
 self.grid.add\_widget(self.filename)  
 self.grid.add\_widget(self.but)  
 return self.grid  
 def go(self,obj):  
 print(self.trigger.text)  
 google\_table(self.m.text, self.speed.text, self.filename.text,self.trigger.text,self.trigger1.text)  
def create\_pdf(text,filename):  
 pdf = FPDF(orientation='P', unit='mm', format='A4')  
 pdf.add\_page()  
 pdf.set\_font('Arial', 'B', 16)  
 for line in text:  
 if len(line) == 0:  
 pdf.ln()  
 else:  
 pdf.cell(0, 7, ' '.join(line), ln=1)  
 pdf.output(f"{filename}.pdf", 'F')  
def google\_table(m, v, s,me,ve):  
 sa = gspread.service\_account(filename="token.json")  
 sheet = sa.open("5task")  
 worksheet = sheet.worksheet("pdf")  
 worksheet.update('A1', s)  
 worksheet.update('C4', me)  
 worksheet.update('C5', ve)  
 worksheet.update('C3', me+"\*("+ve+")^2")  
 worksheet.update('B1', datetime.today().strftime('%Y-%m-%d'))  
 worksheet.update('B4', m)  
 worksheet.update('B5', v)  
 worksheet.update('B5', v)  
  
 create\_pdf(worksheet.get\_all\_values(), datetime.today().strftime('%Y-%m-%d') + ' - ' + s)  
sa = gspread.service\_account(filename="token.json")  
sheet = sa.open("5task")  
worksheet = sheet.worksheet("pdf")  
if \_\_name\_\_ == "\_\_main\_\_":  
 Application().run()

В данном задании мы решили реализовать интерфейс при помощи модуля kivy[2]

Создали класс Application(родительским является kivy.app.App)

Создали GridLayout, в который добавили поля с текстом и кнопку создания pdf

При помощи модуля fpdf было реализовано создание pdf файла, а данные в соответствии задании брались из гугл таблицы, которая предварительно заполнялись в программе.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4 – гугл таблица для задания

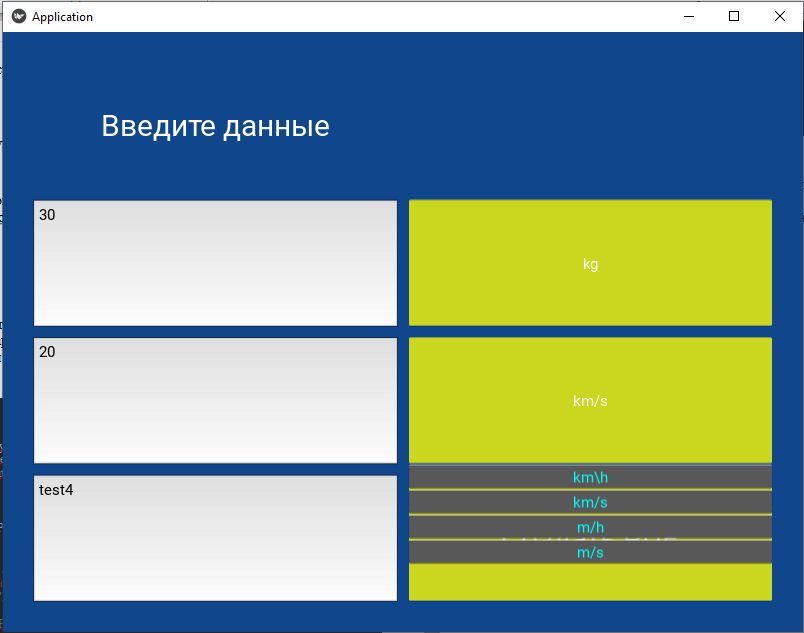


Рисунок 5 – интерфейс программы

Вывод: было крайне интересно реализовать создание pdf файла при помощи модуля FPDF[5], еще и написав для этого интерфейс, а данный брать из гугл таблицы. Можно сказать, это уже работа неплохого обьема, сочетающая разные модули и выполняющая практически полезную задачу.

**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 kivy.app import App  
from kivy.uix.button import Button  
from kivy.uix.label import Label  
from kivy.uix.gridlayout import GridLayout  
from kivy.uix.image import AsyncImage  
from kivy.uix.textinput import TextInput  
from kivy.core.window import Window  
from datetime import datetime  
from fpdf import FPDF  
import gspread,os,glob,webbrowser,PyPDF2  
class Application(App):  
 def build(self):  
 Window.clearcolor = (16 / 255, 71 / 255, 140 / 255, 0);  
 self.grid = GridLayout(cols=1, padding=[30], spacing=10)  
 self.grid.add\_widget(Label(text = "Выберете файл для открытия",font\_size = 30))  
 global a,i,buttons  
 os.chdir('C:/Users/SnusP/PycharmProjects/prog2.1/Lab-6')  
 for i in glob.glob("\*.pdf"):  
 a.append(i)  
 print(a)  
 for i in a:  
 global c  
 self.add\_button(i)  
 return self.grid  
 def go(self,a):  
 global flag  
 if flag == True:  
 return webbrowser.open(a)  
 def add\_button(self,i):  
 global flag  
 flag = False  
 c = Button(text=f"{str(i.split(' - ')[1:])}", font\_size=30, background\_color=(255, 207, 0, 200 / 255),  
 on\_press=lambda x: self.go(i))  
  
 flag = True  
 print(a)  
 self.grid.add\_widget(c)  
 def openpdf(self,a):  
 return webbrowser.open\_new(a)  
buttons = []  
a = []  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 Application().run()

В данной программе мы также решили воспользоваться библиотекой kivy, создание интерфейса по анологии с третьей задачей, кол-во кнопок зависит от кол-ва файлов в директории с расширением .pdf, а по нажатию на кнопку выполняется метод openpdf, который открывает ваш пдф файл

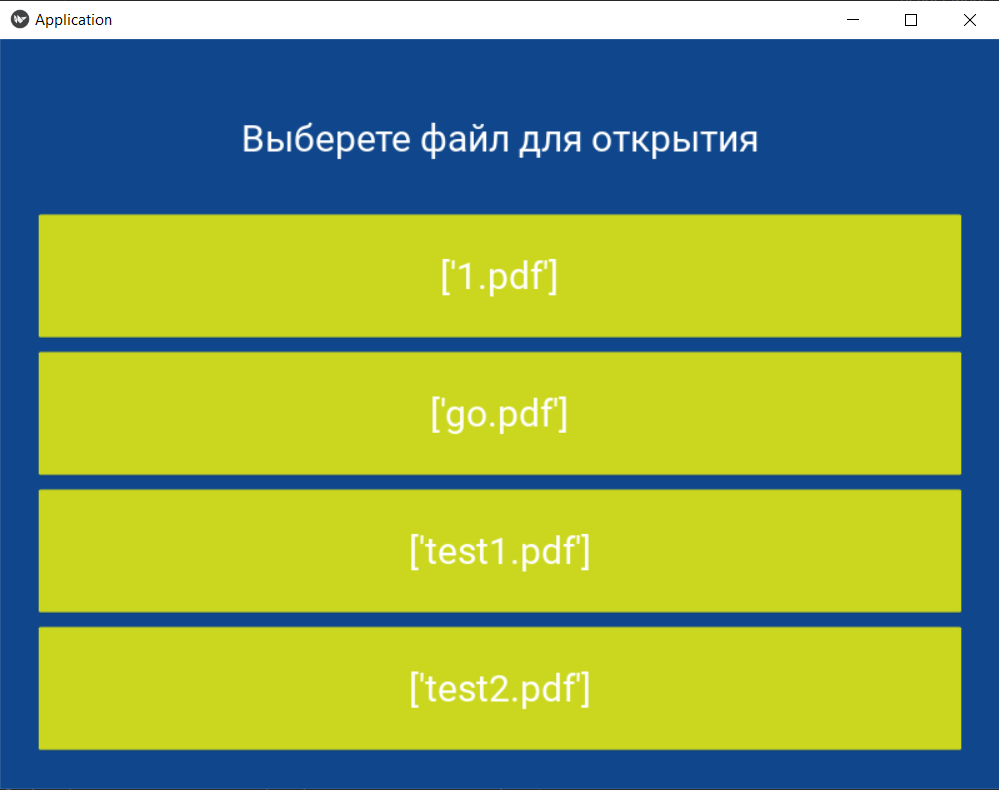


Рисунок 6 – интерфейс программы

**Вывод:** эта задача для нас была не менее интересна, чем вторая , возникли некие трудности связанные с открытием файлов, в том числе, связанные с четким форматом названий(и выводом их в интерфейс определенным образом). Но тем не менее, задача была выполнена. Было интересно поработать с разработкой интерфейсов в ходе этой лабораторной.

**СПИСОК ЛИТЕРАТУРЫ**

1. Wikipedia. Авторегрессионная модель [Электронный ресурс] – https://ru.wikipedia.org/wiki/Авторегрессионная\_модель (16.06.2022);
2. Kivy - Open source Python library for rapid development of applications [Электронный ресурс] – <https://kivy.org/> (16.06.2022).
3. Matplotlib: Visualization with Python. [Электронный ресурс] – <https://matplotlib.org/> (16.06.2022).
4. Gspread. [Электронный ресурс] – <https://docs.gspread.org/en/latest/> (16.06.2022);
5. PyFPDF. FPDF for Python. [Электронный ресурс] – <https://pyfpdf.readthedocs.io/en/latest/> (16.06.2022).