Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря Сікорського
Кафедра обчислювальної техніки ФІОТ

ЗВІТ
з лабораторної роботи №2
з навчальної дисципліни «Data Science Technology»

Тема:

СТАТИСТИЧНЕ НАВЧАННЯ З ПОЛІНОМІАЛЬНОЮ РЕГРЕСІЄЮ

Виконав:
Студент 3 курсу кафедри ОТ ФІОТ,
Навчальної групи ІМ-13
Тавлуй Д. О.

Перевірив:
Професор кафедри ОТ ФІОТ
Писарчук О.О.

Київ 2023

І. Мета:

Виявити дослідити та узагальнити особливості реалізації процесів статистичного навчання із застосуванням методів обробки Big Data масивів та калмановської рекурентної фільтрації з використанням можливостей мови програмування Python.

ІІ. Завдання:

Лабораторія провідної ІТ-компанії реалізує масштабний проект розробки універсальної платформи з обробки Big Data масиву статистичних даних поточного спостереження для виявлення закономірностей і прогнозування розвитку контрольованого процесу. Платформа передбачає розташування back-end компоненти на власному хмарному сервері з наданням повноважень користувачам заздалегідь адаптованого front-end функціоналу універсальної платформи. Замовниками ресурсів платформи є: державні та комерційні компанії валютного трейдінгу для прогнозування динаміки зміни курсу валют та ціни інших товарів; метеорологічні служби для прогнозування параметрів метеоумов; департаменти охорони здоров'я для прогнозування зміни показників епідеміологічних ситуацій тощо.

Завдання ІІІ рівня – максимально 9 балів.

Реалізувати групу вимог 1 та 2.

Група вимог_1:

1. Отримання вхідних даних із властивостями, заданими в Лр_1;
2. Модель вхідних даних із аномальними вимірами;
3. Очищення вхідних даних від аномальних вимірів. Спосіб виявлення аномалій та очищення обрати самостійно;
4. Визначення показників якості та оптимізація моделі (вибір моделі залежно від значення показника якості). Показник якості та спосіб оптимізації обрати самостійно.
5. Статистичне навчання поліноміальної моделі за методом найменших квадратів (МНК – LSM) – поліноміальна регресія для вхідних даних, отриманих в п.1,2. Спосіб реалізації МНК обрати самостійно;
6. Прогнозування (екстраполяцію) параметрів досліджуваного процесу за «навченою» у п.5 моделлю на 0,5 інтервалу спостереження (об'єму вибірки);
7. Провести аналіз отриманих результатів та верифікацію розробленого скрипта.

Група Вимог_2:

1. Отримання вхідних даних із властивостями, заданими в Лр_1;
2. Модель вхідних даних із аномальними вимірами;
3. Очищення вхідних даних від аномальних вимірів. Спосіб виявлення аномалій та очищення обрати самостійно;
4. Визначення показників якості та оптимізація моделі Показник якості та спосіб оптимізації обрати самостійно.
5. Залежно від результатів п.4 реалізувати рекурентне згладжування alfa-beta, або alfa-beta-gamma фільтром сформованих в п.1, 2 вхідних даних. Прийняти заходи подолання явища «розбіжності» фільта.
6. Провести аналіз отриманих результатів та верифікацію розробленого скрипта.

ІІІ. Результати виконання лабораторної роботи.

3.1. Синтезована математична модель

Пошук даних
Як і в першій лабораторній роботі, я обрав ціну 31.10348 г золота у гривнях за останні 4 місяця: https://index.minfin.com.ua/ua/exchange/nbu/bullion/xau/

Так як вартість 31.10348 г золота приблизно дорівнює 70.000 гривень, подальші результати будуть досить великими, але від того навіть цікавіше.

Визначення характеристик

Тренд
Я використав метод найменших квадратів для оцінки динаміки тренду:
Y(t) = 71364.48453635788 + -106.3126543202884 * t + 1.4803376418667489 * t^2

З цього випливає, що
a = 1.4803376418667489,
b = -106.3126543202884,
c = 71364.48453635788

Отже, враховуючи кількість k реальних вимірювань, можемо зробити модель тренду, яка буде працювати для будь-якої кількості n випадкових вимірювань:

Y (k, n, x) = c + (b / (n/k)) * x + (a / ((n/k)^2)) * x^2, де
n – кількість вимірювань
k - кількість реальних вимірювань
x – виміряне значення

3.2. Результати архітектурного проектування та їх опис

Я обрав модульну архітектуру проектування, так як програма стає більш зрозумілою, код багаторазовим для інших задач.

Усі виклики функцій відбуваються у файлі **main.py**
Парсинг сайту з реальними даними відбувається у файлі **data_parsing.py**
Усі математичні дії та побудова графіку відбувається у файлі **math_functions.py**
Alfa-beta фільтр та згладжування відбувається у файлі **data_manipulating.py**

3.3. Опис структури проекту програми

Усі файли підключаються у головний файл, у якому відбуваються всі виклики функцій, побудова графіків та вивід у консоль.

3.4. Результати роботи програми відповідно до завдання

Реальні дані:

Характеристики:

y(t) = 71364.48453635788 + -106.3126543202884 * t + 1.4803376418667489 * t^2

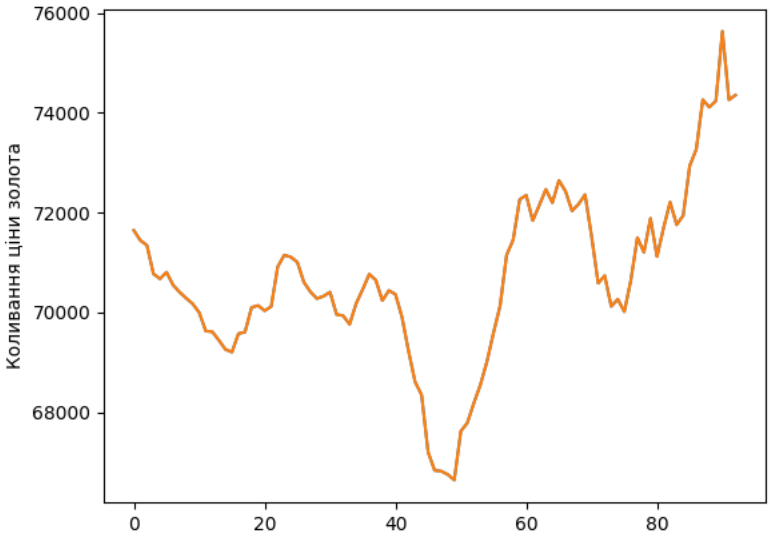
---------- Коливання ціни золота ----------

Мат. очікуання = 166.40384544247354

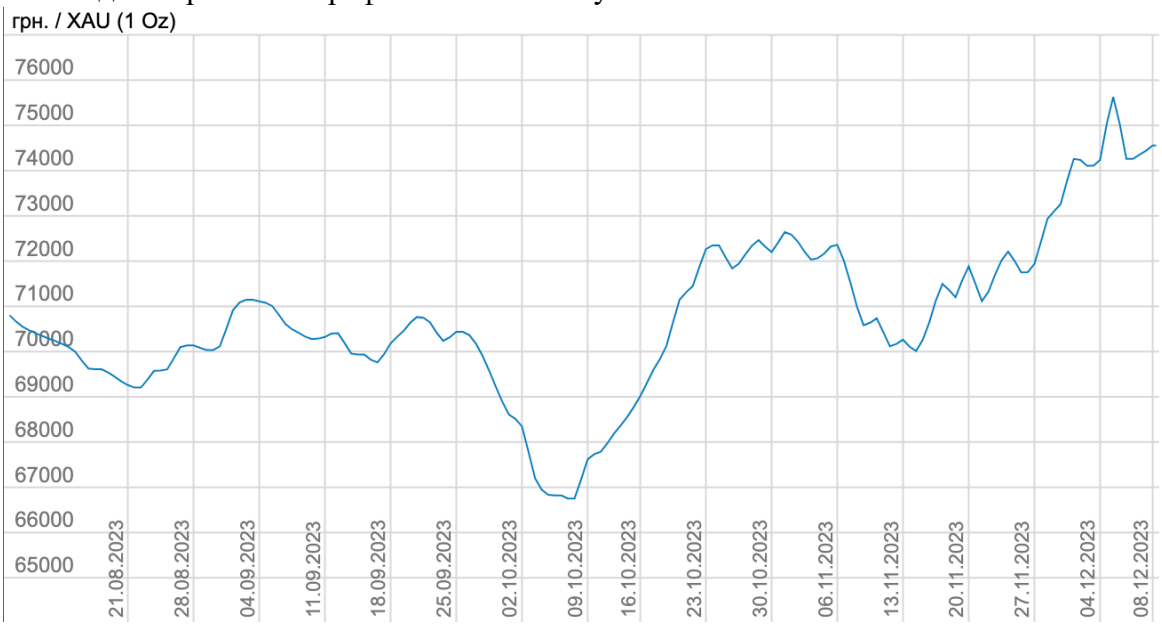
Дисперсія = 1413136.9731218165

Середньоквадратичне відхилення = 1188.7543788023734

Графік:



Також для порівняння графік з самого сайту



Можна побачити, що графіки досить схожі, відмінність лише в тому, що згенерований трохи приплюснутий.

Очищені дані від аномальних вимірів:

Характеристики:

y(t) = 71415.61317884605 + -104.43272737339676 * t + 1.415905811268203 * t^2

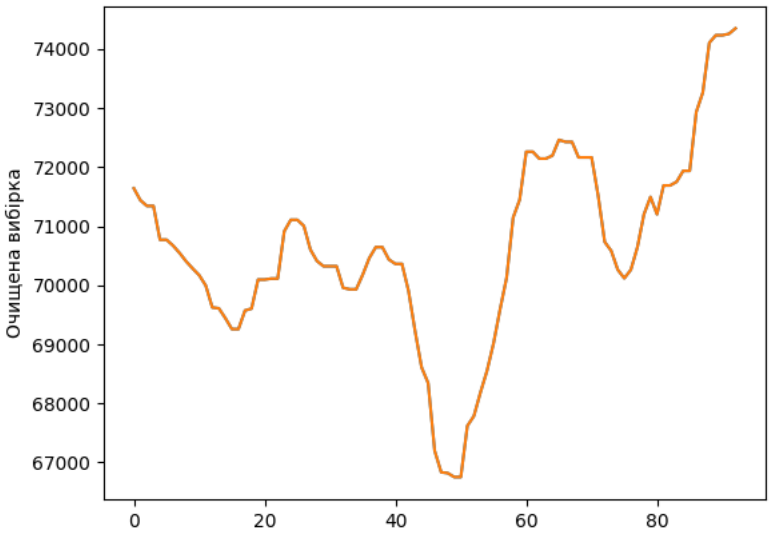
---------- Очищена вибірка ----------

Мат. очікування = 131.46865265567612

Дисперсія = 1354025.074588962

Середньоквадратичне відхилення = 1163.6258310079584

Графік:



Очищена вибірка з моделлю:

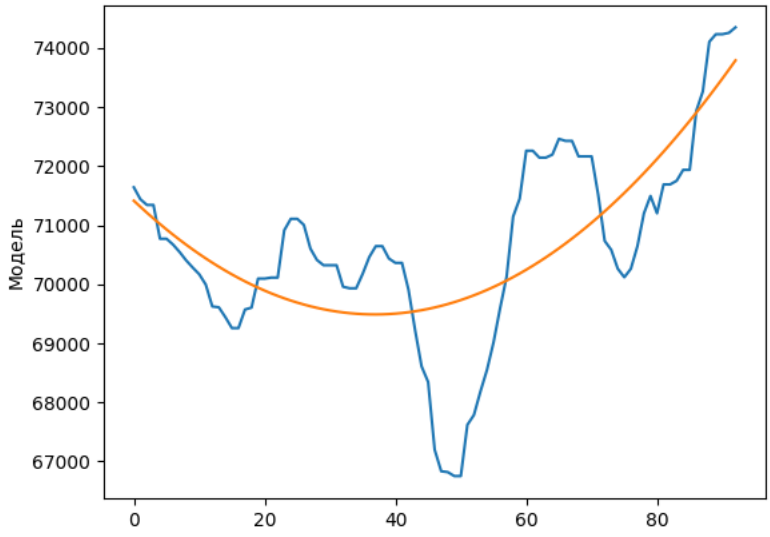
Характеристики:

---------- Модель ----------

кількість елементів вибірки= 93

Коефіцієнт детермінації (ймовірність апроксимації)= 0.49240799875455943

Графік:



МНК прогнозування:

Характеристики:

---------- Прогнозована ----------

Кількість елементів вибірки =140

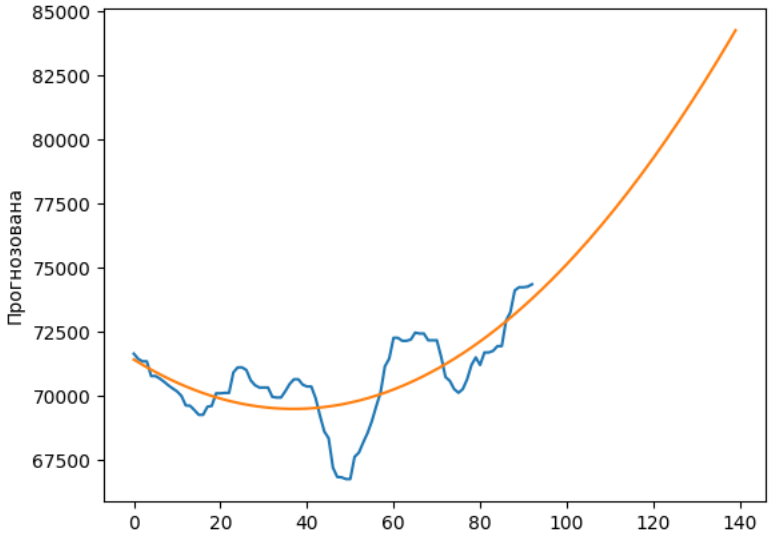
Мат. очікування = 1.1641532182693481e-10

Дисперсія =4.555114145710135e-20

Середньоквадратичне відхилення = 2.1342713383518353e-10

Довірчий інтервал прогнозованих значень =1.0031075290253627e-08

Графік:



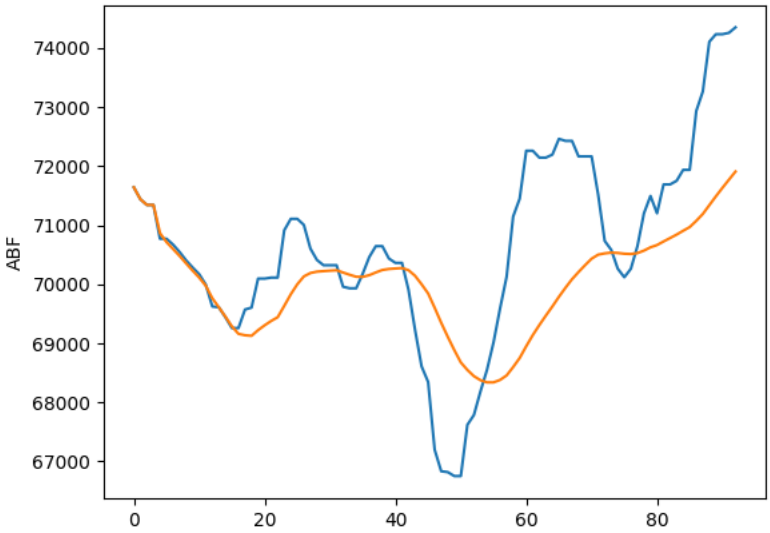
Alfa-beta фільтр:

---------- ABF ----------

кількість елементів вибірки= 93

Коефіцієнт детермінації (ймовірність апроксимації)= 0.27234315552599375

Графік:



Як можна побачити, коефіцієнт детермінації досить малий, що нажаль є погано

3.5. Програмний код, що забезпечує отримання результату

main.py

```
import math as mt
from data_parsing import parsing_site, file_parsing
from math_functions import Plot_AV, print_MNK_characteristics,
print_predict_MNK_characteristics, MNK_Extrapol, r2_score
from data_manipulating import ABF, Sliding_Window_AV_Detect_sliding_wind


if __name__ == '__main__':
    n = 10000
    n_Wind = 3
    extrapolation_koef = 0.5

    print('Обрано: Парсинг табличних даних
https://index.minfin.com.ua/ua/exchange/nbu/bullion/xau/')
    url = "https://index.minfin.com.ua/ua/exchange/nbu/bullion/xau/"
    parsing_site(url)

    data_array = file_parsing(url, 'hrn-to-gold.xlsx', 'Курс (грн.)')
    print_MNK_characteristics(data_array, 'Коливання ціни золота')
    Plot_AV(data_array, data_array, 'Коливання ціни золота')

    smoothed = Sliding_Window_AV_Detect_sliding_wind(data_array, n_Wind)
    Plot_AV(smoothed, smoothed, 'Очищена вибірка')

    info = print_MNK_characteristics(smoothed, 'Очищена вибірка')
    Plot_AV(info, smoothed, 'Модель')
    r2_score(smoothed, info, 'Модель')

    koef = mt.ceil(len(smoothed)*extrapolation_koef)
    predicted = MNK_Extrapol(smoothed, koef)
    Plot_AV(predicted, smoothed, 'Прогнозована')
    print_predict_MNK_characteristics(koef, predicted, 'Прогнозована')

    abf = ABF(smoothed)
    r2_score(smoothed, abf, 'ABF')
    Plot_AV(abf, smoothed, 'ABF')
```

data_parsing.py

```
import numpy as np
import pandas as pd
import re
import requests


def parsing_site(url):
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36'}
    html_source = requests.get(url, headers=headers).text
    html_source = re.sub(r'<.*?>', lambda g: g.group(0).upper(),
html_source)
    dataframe = pd.read_html(html_source)[0]

    dataframe.to_excel("hrn-to-gold.xlsx")
    return dataframe


def file_parsing(url, file_name, data_name):
    dataframe = pd.read_excel(file_name)
    dataframe = dataframe.iloc[:-1]
    values = dataframe[data_name].values / 10000
```

```
    S_real = np.zeros(len(values))

    for i, value in enumerate(values):
        S_real[i] = value

    print(f'Джерело даних: {url}')
    return S_real
```

math_functions.py

```
import numpy as np
import matplotlib.pyplot as plt
import math as mt


def Plot_AV (S0_L, SV_L, Text):
    plt.plot(SV_L)
    plt.plot(S0_L)
    plt.ylabel(Text)
    plt.show()
    return


def MNK (S0):
    iter = len(S0)
    Yout = np.zeros((iter, 1))
    F = np.ones((iter, 3))
    for i in range(iter):
        Yout[i, 0] = float(S0[i])
        F[i, 1] = float(i)
        F[i, 2] = float(i * i)
    FT = F.T
    FFT = FT.dot(F)
    FFTI = np.linalg.inv(FFT)
    FFTIFT = FFTI.dot(FT)
    C = FFTIFT.dot(Yout)
    y_output = F.dot(C)
    print('Регресійна модель:')
    print(f'y(t) =  {C[0,0]}  + {C[1,0]}  * t  +  {C[2,0]}  * t^2')
    return y_output


def MNK_Extrapol (S0, koef):
    iter = len(S0)
    Yout_Extrapol = np.zeros((iter+koef, 1))
    Yin = np.zeros((iter, 1))
    F = np.ones((iter, 3))
    for i in range(iter):
        Yin[i, 0] = float(S0[i])
        F[i, 1] = float(i)
        F[i, 2] = float(i * i)
    FT=F.T
    FFT = FT.dot(F)
    FFTI=np.linalg.inv(FFT)
    FFTIFT=FFTI.dot(FT)
    C=FFTIFT.dot(Yin)
    print('Регресійна модель:')
    print(f'y(t) =  {C[0,0]}  + {C[1,0]}  * t  +  {C[2,0]}  * t^2')
    for i in range(iter+koef):
        Yout_Extrapol[i, 0] = C[0, 0]+C[1, 0]*i+(C[2, 0]*i*i)
    return Yout_Extrapol
```

```python
def print_MNK_characteristics(data, title):
    num_iterations = len(data)
    y_output = MNK(data)
    result_data = np.zeros((num_iterations))
    for i in range(num_iterations):
        result_data[i] = data[i] - y_output[i, 0]

    median = np.median(result_data)
    var = np.var(result_data)
    meanSquare = mt.sqrt(var)

    print(f'------------ {title} -------------')
    print(f'Мат. очікування = {median}')
    print(f'Дисперсія = {var}')
    print(f'Середньоквадратичне відхилення = {meanSquare}')

    return y_output


def print_predict_MNK_characteristics(koef, data, title):
    num_iterations = len(data)
    y_output = MNK(data)
    result_data = np.zeros((num_iterations))
    for i in range(num_iterations):
        result_data[i] = data[i] - y_output[i, 0]

    median = np.median(result_data)
    var = np.var(result_data)
    meanSquare = mt.sqrt(var)

    scvS_extrapol = meanSquare * koef
    print(f'------------ {title} -------------')
    print(f'Кількість елементів вибірки ={len(data)}')
    print(f'Мат. очікування = {median}')
    print(f'Дисперсія ={var}')
    print(f'Середньоквадратичне відхилення = {meanSquare}')
    print(f'Довірчий інтервал прогнозованих значень ={scvS_extrapol}')


def r2_score(SL, Yout, Text):
    iter = len(Yout)
    numerator = 0
    denominator_1 = 0
    for i in range(iter):
        numerator = numerator + (SL[i] - Yout[i, 0]) ** 2
        denominator_1 = denominator_1 + SL[i]
    denominator_2 = 0
    for i in range(iter):
        denominator_2 = denominator_2 + (SL[i] - (denominator_1 / iter)) **
2
    R2_score_our = 1 - (numerator / denominator_2)
    print('------------', Text, '-------------')
    print('кількість елементів вибірки=', iter)
    print('Коефіцієнт детермінації (ймовірність апроксимації)=',
R2_score_our)

    return R2_score_our
```

data_manipulating.py

```
import numpy as np
import math as mt


def ABF (S0):
    iter = len(S0)
    Yin = np.zeros((iter, 1))
    YoutAB = np.zeros((iter, 1))
    T0=1
    for i in range(iter):
        Yin[i, 0] = float(S0[i])

    Yspeed_retro=(Yin[1, 0]-Yin[0, 0])/T0
    Yextra=Yin[0, 0]+Yspeed_retro
    alfa=2*(2*1-1)/(1*(1+1))
    beta=(6/1)*(1+1)
    YoutAB[0, 0]=Yin[0, 0]+alfa*(Yin[0, 0])

    for i in range(1, iter):
        YoutAB[i,0]=Yextra+alfa*(Yin[i, 0]- Yextra)
        Yspeed=Yspeed_retro+(beta/T0)*(Yin[i, 0]- Yextra)
        Yspeed_retro = Yspeed
        Yextra = YoutAB[i,0] + Yspeed_retro
        alfa = (2 * (2 * i - 1)) / (i * (i + 1))
        beta = 6 /(i* (i + 1))
    YoutAB[0] = S0[0]
    return YoutAB

def Sliding_Window_AV_Detect_sliding_wind (S0, n_Wind):
    iter = len(S0)
    j_Wind=mt.ceil(iter-n_Wind)+1
    S0_Wind=np.zeros((n_Wind))
    Midi = np.zeros(( iter))
    for j in range(j_Wind):
        for i in range(n_Wind):
            l=(j+i)
            S0_Wind[i] = S0[l]
        Midi[l] = np.median(S0_Wind)
    S0_Midi = np.zeros((iter))
    for j in range(iter):
        S0_Midi[j] = Midi[j]
    for j in range(n_Wind):
        S0_Midi[j] = S0[j]
    return S0_Midi
```

Висновки

У цій лабораторній роботі я використав частину програми з першої роботи, а також додав нові можливості з умови цієї роботи. Я дізнався про прогнозування тренду за допомогою MNK та alpha-beta фільтрацією, і застосував на практиці.
В результаті тренд було спрогнозовано дуже точно, було взято коефіцієнт екстраполяції 0.5 (спрогнозовано 50% від вибірки додатково), при якому довірчий інтервал прогнозованих значень був досить малий, що свідчить про гарний прогноз. Проте alfa-beta фільтрація відпрацювала нажаль погано.