

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

**ЗВІТ
з лабораторної роботи №7
з навчальної дисципліни «Data Science Technology»**

Тема:

**РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ПРОГНОЗУВАННЯ ДИНАМІКИ
ЗМІНИ ПОКАЗНИКІВ ЕФЕКТИВНОСТІ ТОРГІВЕЛЬНИХ КОМПАНІЙ (міні
проекти в галузі аналізу даних для завдань електронної комерції)**

Виконав:

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

Перевірив:

Професор кафедри ОТ ФІОТ
Писарчук О.О.

Київ 2023

I. Мета:

дослідити виявити та узагальнити особливості реалізації проектного практикуму в галузі аналізу часових (стохастичних рядів), як характеристика показників ефективності діяльності торгівельних компаній.

II. Завдання:

Розробити програмний скрипт мовою Python, що реалізує функціонал за обраним рівнем складності:

I рівень складності 7 балів

8	Розробити програмний скрипт, що реалізує: 1. Парсинг файлу параметрів: Pr_1.xls; 2. Попередній аналіз даних; 3. Визначення показників ефективності – продаж та прибутку; 4. Визначення математичної моделі даних відповідно до МНК; 5. Визначити динаміку зміни прибутку за регіонами (таблиця, графік, файл).
---	---

III. Результати виконання лабораторної роботи.

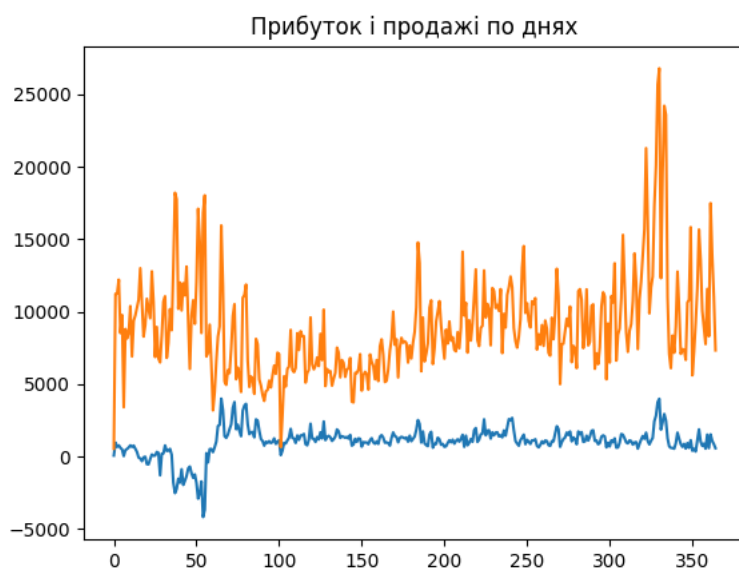
Результати архітектурного проектування та їх опис

Я обрав модульну архітектуру проектування. Файл data_processing.py містить функції для роботи з вхідними даними, які парсяться за допомогою функції parse. У файлі modeling.py знаходяться функція МНК та функція відображення характеристик. Файл main.py являється головним, у якому відбувається запуск всіх функцій.

Опис структури проекту програми

Файли data_processing.py та modeling.py підключаються у головний файл, де відбувається виклик функцій побудови графіків, МНК та інші.

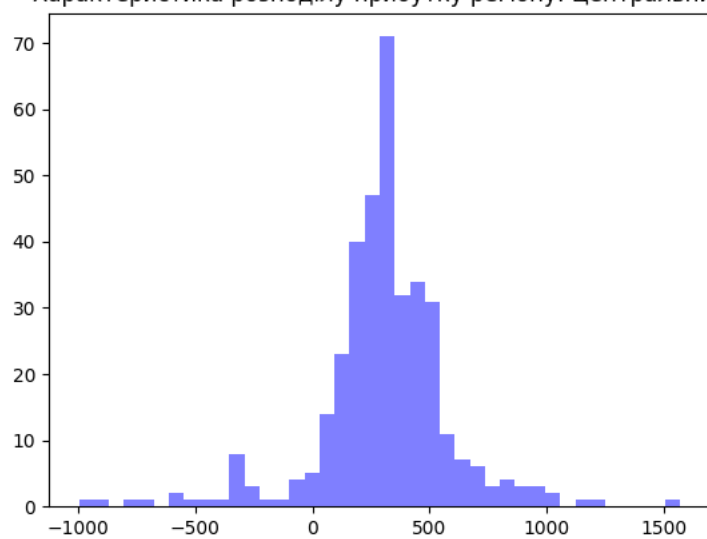
Результати роботи програми відповідно до завдання



Результати виконання для Центрального регіону:

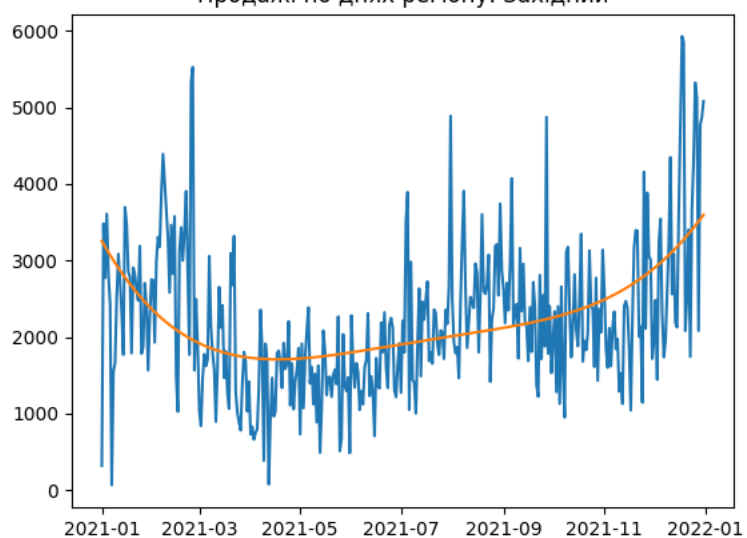


Характеристика розподілу прибутку регіону: Центральний



Результати виконання для Західного регіону:

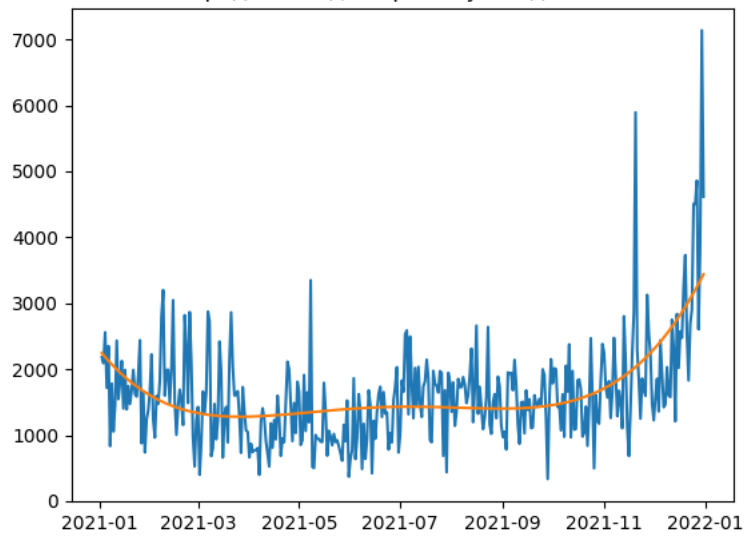
Продажі по днях регіону: Західний



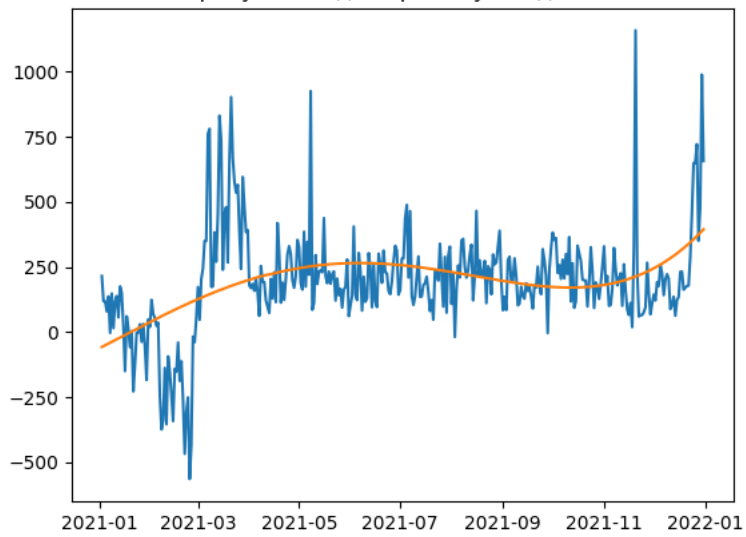


Результати виконання для Південного регіону:

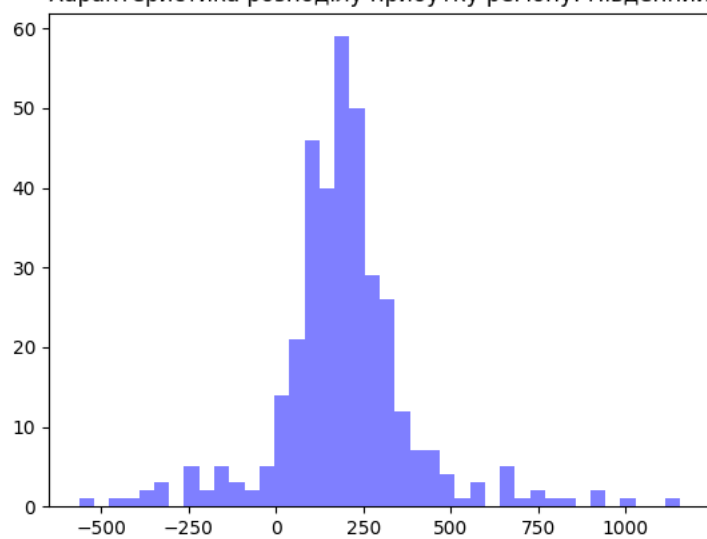
Продажі по днях регіону: Південний



Прибуток по днях регіону: Південний

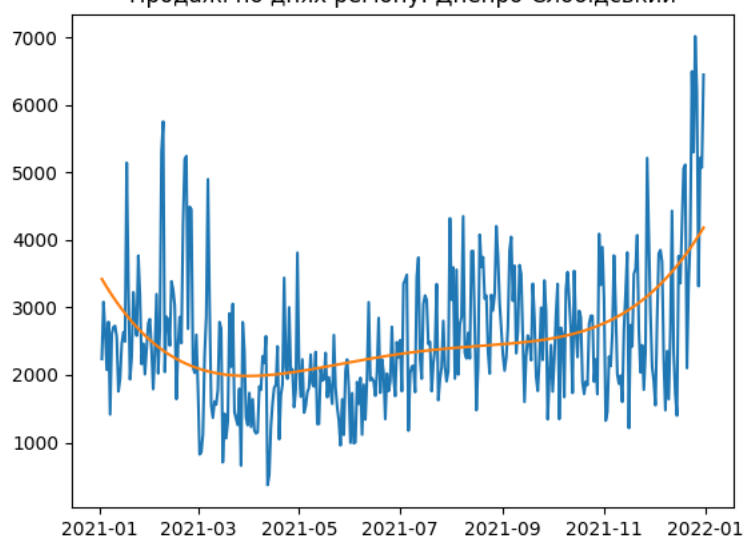


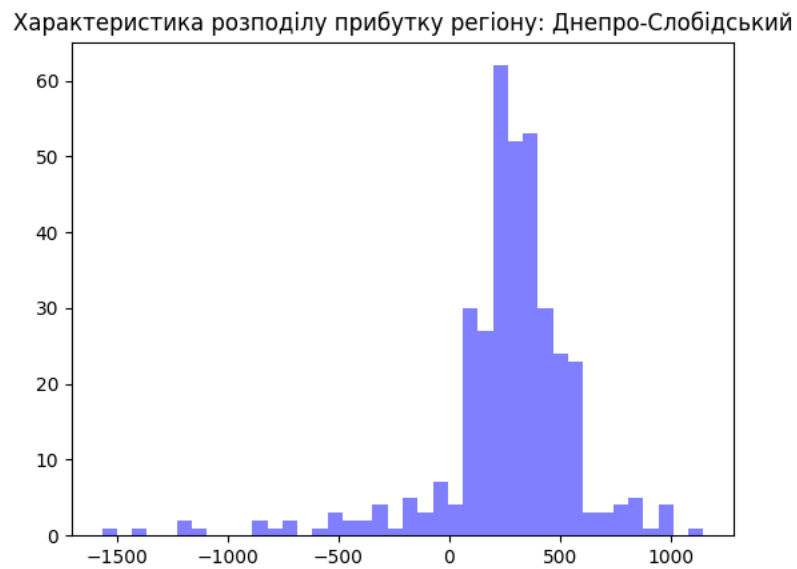
Характеристика розподілу прибутку регіону: Південний



Результати виконання для Днепро-Слобідського регіону:

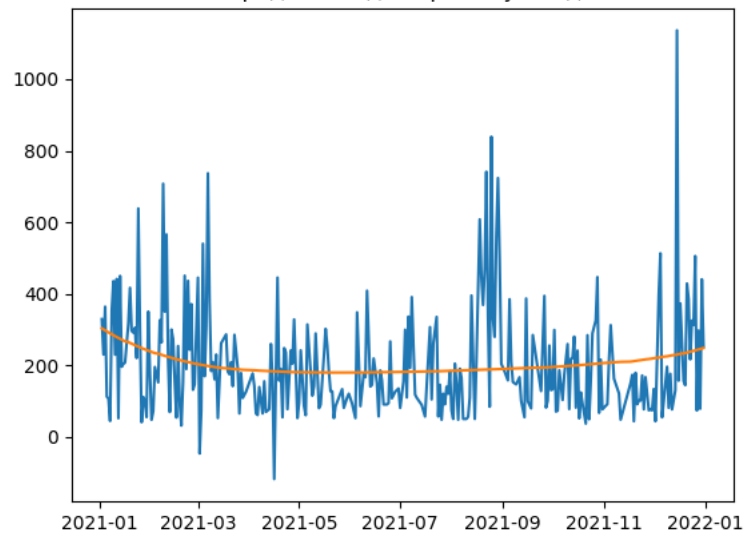
Продажі по днях регіону: Днепро-Слобідський



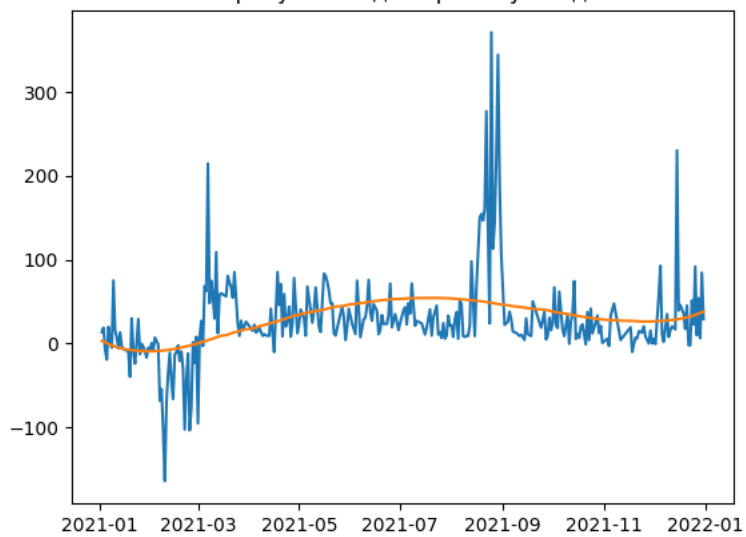


Результати виконання для Східного регіону:

Продажі по днях регіону: Схід



Прибуток по днях регіону: Схід





Програмний код, що забезпечує отримання результату

main.py

```
from data_processing import parse, visualize, data_sum_sales,
data_sum_profit, region_clustering
from modeling import MNK, Stat_characteristics

if __name__ == '__main__':
    file_path = 'Pr_1.xls'
    data, magaz_dict = parse(file_path)

    sales = data_sum_sales(data)
    profit = data_sum_profit(data)

    visualize(profit.values(), sales.values(), text="Прибуток і продажі по
днях")

    cluster = region_clustering(data, magaz_dict)

    for region in cluster:
        sales_data = cluster[region]['sales']
        dates = sales_data.keys()
        MNK_sales = MNK(list(sales_data.values()))
        visualize(sales_data.values(), MNK_sales, text=f'Продажі по днях
регіону: {region}', keys=dates)

        profit_data = cluster[region]['profit']
        dates = profit_data.keys()
        MNK_profit = MNK(list(profit_data.values()), True)
        visualize(profit_data.values(), MNK_profit, text=f'Прибуток по днях
регіону: {region}', keys=dates)
        Stat_characteristics(list(profit_data.values()), f'Характеристика
розподілу прибутку регіону: {region}')
```

data_processing.py

```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
```

```

def parse(file):
    df = pd.read_excel(file)

    data = np.transpose(df.iloc[:, 0:6].values)
    district_dict = dict(zip(df.iloc[:, 9].dropna().values,
df['Period'].dropna().values))

    return data, district_dict

def visualize(*args, text, keys=0):
    plt.clf()
    for arg in args:
        plt.plot(keys, arg) if keys else plt.plot(arg)
    plt.title(text)
    plt.show()

def data_sum_sales(data):
    result = {}
    for i in range(len(data[3])):
        current_key = data[1][i]
        sales = data[3][i] * data[5][i]

        if current_key in result:
            result[current_key] += sales
        else:
            result[current_key] = sales
    return result

def data_sum_profit(data):
    result = {}
    for i in range(len(data[3])):
        current_key = data[1][i]
        profit = (data[5][i] - data[4][i]) * data[3][i]
        if current_key in result:
            result[current_key] += profit
        else:
            result[current_key] = profit
    return result

def calculate_sales_profit(row):
    date = row[1]
    sales = [date, row[3] * row[5]]
    profit = [date, (row[5] - row[4]) * row[3]]
    return sales, profit

def region_clustering(data, region_dict):
    result = {}

    for i in range(len(data[3])):
        magaz_code = data[0][i]
        region = region_dict.get(magaz_code, "")

        sales, profit = calculate_sales_profit(data[:, i])

```

```

        if region not in result:
            result[region] = {'sales': [], 'profit': []}

        result[region]['sales'].append(sales)
        result[region]['profit'].append(profit)

    for region in result:
        region_date_clustering(result[region])

    return result

def cluster_dates(data, key):
    clustered_dates = {}
    for entry in data[key]:
        if entry[0] not in clustered_dates:
            clustered_dates[entry[0]] = entry[1]
        else:
            clustered_dates[entry[0]] += entry[1]
    sorted_dates = dict(sorted(clustered_dates.items(), key=lambda x:
pd.to_datetime(x[0])))
    data[key] = sorted_dates

def region_date_clustering(region):
    cluster_dates(region, 'sales')
    cluster_dates(region, 'profit')

```

modeling.py

```

import numpy as np
from matplotlib import pyplot as plt
import math as mt

def MNK (Y_coord, show_model = False):
    iter = len(Y_coord)
    Yin = np.zeros((iter, 1))
    F = np.ones((iter, 5))

    for i in range(iter):
        Yin[i, 0] = Y_coord[i]
        F[i, 1] = float(i)
        F[i, 2] = float(i * i)
        F[i, 3] = float(i * i * i)
        F[i, 4] = float(i * i * i * i)

    FT=F.T
    FFT = FT.dot(F)
    FFTI=np.linalg.inv(FFT)
    FFTIFT=FFTI.dot(FT)
    C=FFTIFT.dot(Yin)
    Yout=F.dot(C)
    if(show_model):
        print('Модель прибыли:')
        print('y(t) = ', C[0, 0], ' + ', C[1, 0], ' * t', ' + ', C[2, 0], ' *
t^2', ' + ', C[3, 0], ' * t^3', ' + ', C[4, 0], ' * t^4')
    return Yout

def Stat_characteristics (SL, Text):

```

```

Yout = MNK(SL)
iter = len(Yout)
SL0 = np.zeros((iter ))
for i in range(iter):
    SL0[i] = SL[i] - Yout[i, 0]
mS = np.median(SL0)
dS = np.var(SL0)
scvS = mt.sqrt(dS)
print(f'----- {Text} -----')
print(f'Мат. очікування = {mS}')
print(f'Дисперсія = {dS}')
print(f'Середньоквадратичне відхилення = {scvS}')
print('-----')
plt.title(Text)
plt.hist(SL, bins=40, facecolor="blue", alpha=0.5)
plt.show()
return

```

Висновки

Під час виконання лабораторної роботи був створений програмний модуль, який застосовує метод найменших квадратів (МНК) для дослідження та передбачення динаміки ефективності торговельних компаній у різних регіонах. Застосовуючи цей модуль, можна аналізувати та візуалізувати зміни в обсягах продажів та рівня прибутку. Додатково був проведений статистичний аналіз розподілу прибутку між різними регіонами.

Отримані в ході дослідження результати дозволяють зробити висновки щодо ефективності діяльності торговельних компаній у визначених регіонах. Ця інформація становить ключовий компонент для управлінських рішень та стратегічного планування, допомагаючи приймати обґрунтовані рішення щодо подальшого розвитку та оптимізації діяльності компаній у конкретних географічних областях.