**Міністерство освіти і науки України**

**Національний технічний університет України «КПІ» імені Ігоря Сікорського**

**Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ**

**з лабораторної роботи №1**

**з навчальної дисципліни «Технології Data Science»**

**Тема:**

**ПІДГОТОВКА ТА АНАЛІЗ ДАНИХ ДЛЯ СТАТИСТИЧНОГО НАВЧАННЯ**

**Виконав:**

Студент 4 курсу кафедри ІПІ ФІОТ,

Навчальної групи ІП-11

Трикош І. В.

**Перевірив:**

Професор кафедри ОТ ФІОТ

Писарчук О.О.

**Київ 2024**

## **І. Мета:**

Виявити дослідити та узагальнити особливості застосування методів статистичного навчання для задач визначення статистичних характеристик вхідного потоку даних з використанням спеціалізованих пакетів мови програмування Python.

## **ІІ. Завдання:**

Завдання ІІІ рівня – максимально 9 балів.

1. Провести парсинг самостійно обраного сайту. Вміст даних, що підлягають парсингу – обрати самостійно.
2. Результати парсингу зберегти у файлі. Тип файлу обрати самостійно.
3. Оцінити динаміку тренду реальних даних.
4. Здійснити визначення статистичних характеристик результатів парсингу.
5. Синтезувати та верифікувати модель даних, аналогічних за трендом і статистичними характеристиками реальним даним, які є результатом парсингу.
6. Провести аналіз отриманих результатів.

Дані індексу S&P 500: <https://www.nasdaq.com/market-activity/index/spx/historical?page=1&rows_per_page=1000&timeline=y1>

## **ІІІ. Результати виконання лабораторної роботи.**

* 1. **Синтезована математична модель**

Відповідно до умов задачі синтезовано математичну модель даних, що аналогічна за трендом та статистичними характеристиками до реальних річних показників індексу S&P 500.

Лінія тренду реальних показників індексу є лінійною регресією першого порядку:

де коефіцієнти та знаходяться з допомогою методу найменших квадратів.

Метод найменших квадратів - метод знаходження наближеного розв'язку надлишково-визначеної системи. Мета полягає в налаштуванні параметрів функції моделі, щоб найкраще відповідати набору даних. Для знаходження двох параметрів лінійної регресії метод працює так:

Нехай є пар точок , , де – незалежна змінна, а – залежна змінна. Потрібно знайти модельну функцію . І ця функція повинна якнайкраще відповідати даним. Ця відповідність вимірюється її залишком, який є різницею між спостережуваним значенням та значенням, передбаченим моделлю:

Метод найменших квадратів знаходить оптимальні значення параметрів шляхом мінімізації суми квадратів залишків:

На виході матимемо значення параметрів та .

Для того, щоб синтезувати синтетичні дані, додамо до лінії тренду шум, що є випадковою величиною, розподіленою за нормальним законом розподілу. Щільність закону:

де – математичне сподівання, – дисперсія, а – середнє квадратичне відхилення. Графік цього розподілу наведено на рисунку 1.

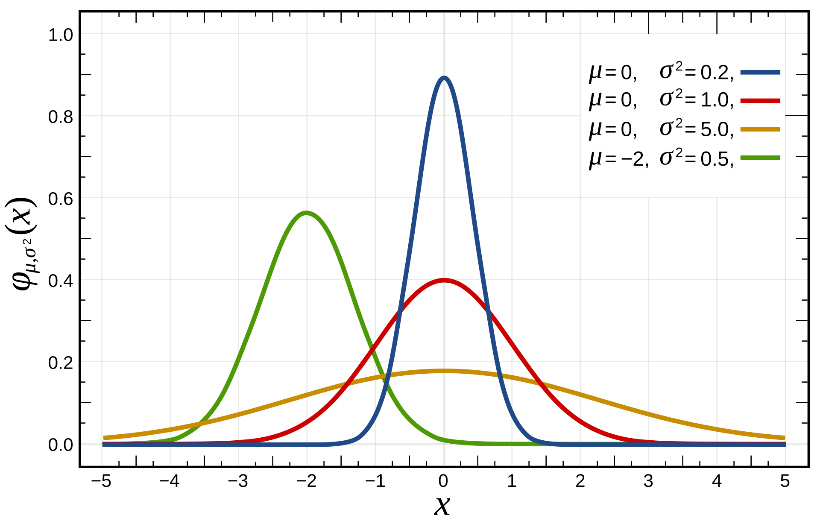


Рисунок 1 – Графік нормального розподілу при різних параметрах

Математичне сподівання знаходиться за формулою:

де – значення випадкової величини, – ймовірність випадкової величини.

Дисперсія знаходиться за формулою:

де – значення випадкової величини, – ймовірність випадкової величини і – математичне сподівання.

Середнє квадратичне відхилення знаходиться за формулою:

де – дисперсія.

* 1. **Результати архітектурного проектування та їх опис**

На рис. 2 зображена блок-схема до моделі до програмного скрипту:



Рисунок 2 – Блок схема алгоритму

Робота алгоритму розпочинається з ініціалізації параметрів програми. Сюди входить створення веб драйвера та визначення URL сайту, звідки будуть парситися дані. Ці дії виконуються у блоці 1 блок-схеми алгоритму рис.2.

У блоці 2 виконується знаходження на сайті таблиці з річними даними про S&P 500 з допомогою веб драйвера.

У блоці 3 витягуються дані з таблиці. Спочатку витягуються заголовки стовпців, а потім уже самі значення індексу.

У блоці 4 створюється датафрейм з отриманих даних. Також тут дані приводяться до потрібного типу, а непотрібні стовпці видаляються.

У блоці 5 відбувається збереження очищеного датафрейму у файл.

У блоці 6 виконується створення лінії тренду. Коефіцієнти цієї лінії знаходяться методом найменших квадратів.

У блоці 7 до лінії тренду додається шум, розподілений за нормальним розподілом. На основі цього створюються синтетичні дані.

У блоці 8 відбувається відображення реальних та синтетичних даних і лінії тренду.

У блоці 9 відбувається вивід статистичних характеристик реальних та синтетичних даних. На цьому виконання алгоритму завершене.

* 1. **Опис структури проекту програми**

Для реалізації розробленого алгоритму мовою програмування Python з використанням можливостей інтегрованого середовища PyCharm сформовано проєкт.

Проєкт базується на лінійній бізнес-логіці функціонального програмування та має таку структуру.

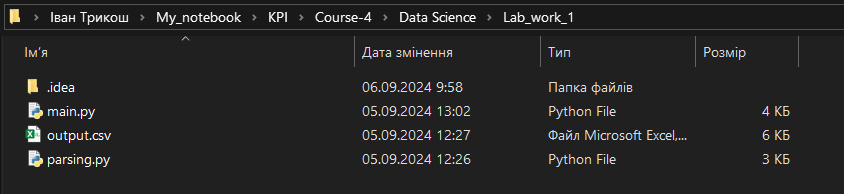


Рисунок 3 – Структура проєкту

Lab\_work\_1 – головний каталог проєкту

main.py – файл програмного коду для синтезу моделі даних

parsing.py – файл, що виконує парсинг сайту та очищення даних

output.csv – результат роботи скрипта у файлі parsing.py

* 1. **Результати роботи програми відповідно до завдання (допускається у формі скриншотів)**

Результатом роботи програми є сукупність послідовності графічних вікон, що реалізують умови завдання лабораторної роботи.

1. На рисунках 4-5 зображено результат парсингу сайту біржі Nasdaq:

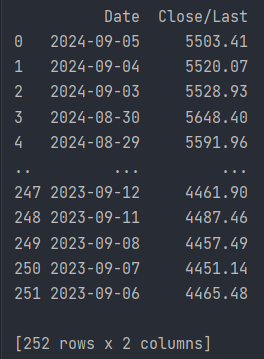


Рисунок 4 – Отримані дані із сайту

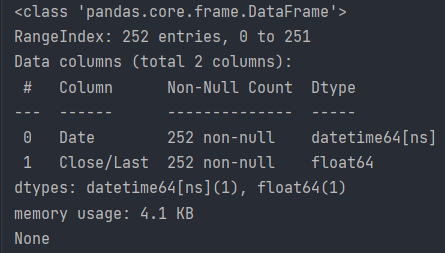


Рисунок 5 – Інформація про отриманий датафрейм

1. На рисунку 6 зображено реальний часовий ряд та його лінію тренду:

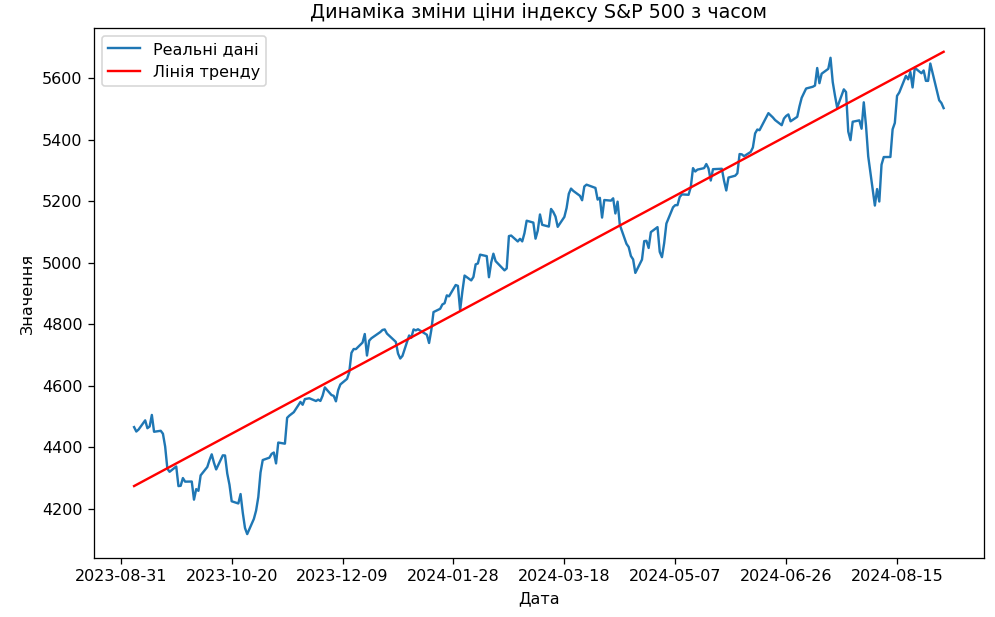


Рисунок 6 – Дані про значення індексу S&P 500 за рік та лінія тренду

Бачимо, що лінія тренду зростаюча, бо індекс росте.

1. На рисунку 7 зображено реальний часовий ряд, його лінію тренду та синтезовані дані:

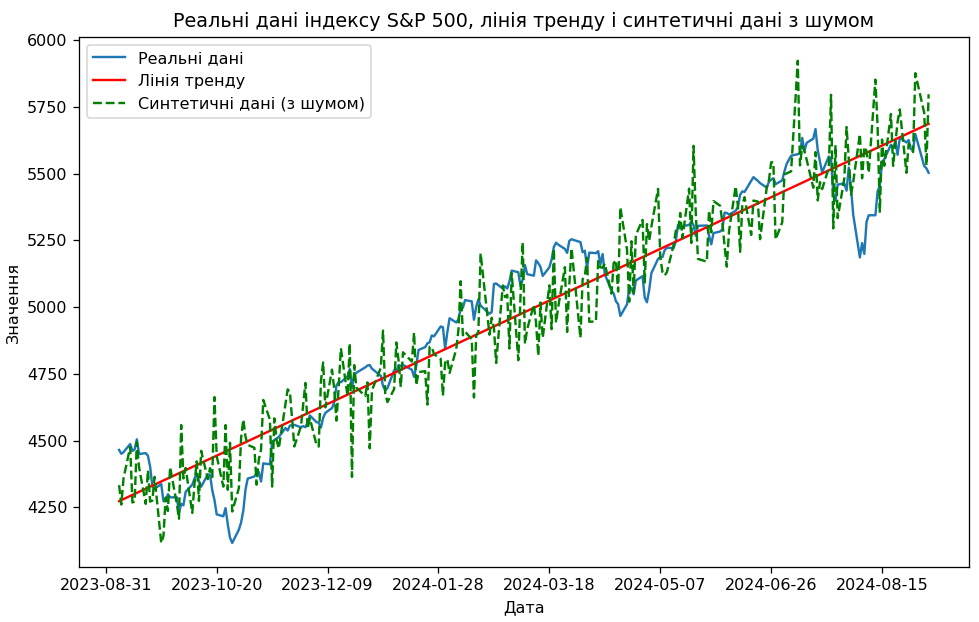


Рисунок 7 – Порівняння реальних та синтетичних даних

Синтетичні дані мають дещо більші коливання у ціні.

1. На рисунку 8 зображено статистику для реальних даних:

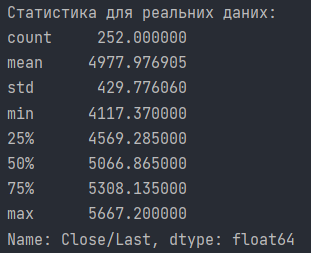


Рисунок 8 – Статистика реальних даних

1. На рисунку 9 зображено статистику для синтетичних даних:

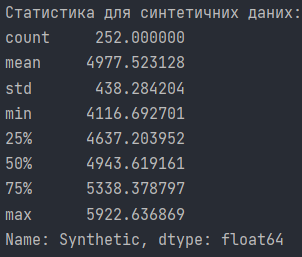


Рисунок 9 – Статистика синтетичних даних

Бачимо, що основні параметри, а саме математичне сподівання (mean) та середнє квадратичне відхилення (sdt) майже однакові, тому дані синтезовано успішно.

* 1. **Програмний код, що забезпечує отримання результату (допускається у формі скриншотів)**

Програмний код послідовно реалізує алгоритм на рис.1 та спрямований на отримання результатів, поданих на рис.4-9.

Для спрощення програмного коду і раціоналізації обчислень застосовано функціональні механізми створення підпрограм.

При цьому використано можливості Python бібліотек: numpy, selenium, pandas, seaborn, matplotlib.

Контекстні коментарі пояснюють сутність окремих скриптів наведеного коду програми.

parsing.py

*"""  
Отримання даних про річну зміну ціни індексу S&P 500 з біржі NASDAQ за допомогою парсингу  
"""  
  
from* selenium *import* webdriver  
*import* pandas *as* pd  
*from* selenium.webdriver.common.by *import* By  
  
*# Ініціалізація веб драйвера відповідно до браузера*driver = webdriver.Chrome()  
  
*# Отримання сторінки за URL*url = "https://www.nasdaq.com/market-activity/index/spx/historical?page=1&rows\_per\_page=1000&timeline=y1"  
driver.get(url)  
  
*# Очікування, поки сторінка завантажиться*driver.implicitly\_wait(10)  
  
*# Сторінка нестатична, тому таблицю з даними треба отримати за допомогою скрипта*shadow\_host = driver.find\_element(By.CSS\_SELECTOR, "nsdq-table")  
shadow\_root = driver.execute\_script("return arguments[0].shadowRoot", shadow\_host)  
  
*# Знаходження відповідної таблиці*table = shadow\_root.find\_element(By.CSS\_SELECTOR, "div.simple-table-template.table")  
  
*# Витягування заголовків стовпців таблиці*headers = [header.text *for* header *in* table.find\_elements(By.CSS\_SELECTOR, "div.table-header-cell")]  
*# Витягування рядків таблиці та значень у комірках*rows = []  
*for* row *in* table.find\_elements(By.CSS\_SELECTOR, "div.table-row"):  
 cells = [cell.text *for* cell *in* row.find\_elements(By.CSS\_SELECTOR, "div.table-cell")]  
 rows.append(cells)  
  
*# Створення датафрейму та очищення та перетворення даних до потрібного типу*df = pd.DataFrame(rows, columns=headers)  
df['Date'] = pd.to\_datetime(df['Date'], format='%m/%d/%Y')  
df['Close/Last'] = df['Close/Last'].replace(',', '', regex=*True*).astype(float)  
df = df[['Date', 'Close/Last']]  
print(df)  
print(df.info())  
df.to\_csv("output.csv")  
  
*# Завершення роботи драйвера*driver.quit()

main.py

*"""  
Оцінка динаміки тренду ціни протягом рік індексу S&P 500 на біржі NASDAQ.  
Створення синтетичних даних за цей період на основі лінії тренду реальних даних та шуму за нормальним розподілом.  
"""  
  
import* matplotlib.pyplot *as* plt  
*import* numpy *as* np  
*import* pandas *as* pd  
*import* seaborn *as* sns  
  
*# Зчитування даних у датафрейм та початкове перетворення*df = pd.read\_csv('output.csv')  
df['Date'] = pd.to\_datetime(df['Date'])  
df = df.sort\_values(by='Date', ascending=*True*).reset\_index(drop=*True*)  
  
*# Переведемо дату в кількість днів для лінійної регресії*df['Days'] = (df['Date'] - df['Date'].min()).dt.days  
  
*# Обчислення коефіцієнтів лінійної регресії y = α + β \* x*coefficients = np.polyfit(df['Days'], df['Close/Last'], 1)  
beta = coefficients[0]  
alpha = coefficients[1]  
  
*# Обчислення значень для лінії тренду*df['Trend'] = beta \* df['Days'] + alpha  
  
*# Побудова графіка реальних даних та лінії тренду*plt.figure(figsize=(10, 6))  
sns.lineplot(x='Date', y='Close/Last', data=df, label='Реальні дані')  
sns.lineplot(x='Date', y='Trend', data=df, label='Лінія тренду', color='red')  
plt.gca().xaxis.set\_major\_locator(plt.MaxNLocator(nbins=10))  
plt.title('Динаміка зміни ціни індексу S&P 500 з часом')  
plt.xlabel('Дата')  
plt.ylabel('Значення')  
plt.show()  
  
*# Додаємо шум (нормальний розподіл з середнім 0 і стандартним відхиленням, близьким до реального)*np.random.seed(42)  
std\_dev = np.std(df['Close/Last'] - df['Trend'])  
noise = np.random.normal(loc=0, scale=std\_dev, size=len(df))  
  
*# Створюємо синтетичні дані, додаючи шум до тренду*df['Synthetic'] = df['Trend'] + noise  
  
*# Побудова графіка реальних, синтетичних даних і лінії тренду*plt.figure(figsize=(10, 6))  
sns.lineplot(x='Date', y='Close/Last', data=df, label='Реальні дані')  
sns.lineplot(x='Date', y='Trend', data=df, label='Лінія тренду', color='red')  
sns.lineplot(x='Date', y='Synthetic', data=df, label='Синтетичні дані (з шумом)', linestyle='--', color='green')  
plt.gca().xaxis.set\_major\_locator(plt.MaxNLocator(nbins=10))  
plt.title('Реальні дані індексу S&P 500, лінія тренду і синтетичні дані з шумом')  
plt.xlabel('Дата')  
plt.ylabel('Значення')  
plt.legend()  
plt.show()  
  
*# Описова статистика для реальних даних*real\_statistics = df['Close/Last'].describe()  
print("Статистика для реальних даних:")  
print(real\_statistics)  
  
*# Описова статистика для синтетичних даних*synthetic\_statistics = df['Synthetic'].describe()  
print("\nСтатистика для синтетичних даних:")  
print(synthetic\_statistics)

## **IV. Висновки.**

У ході виконання лабораторної роботи було проведено дослідження особливостей застосування методів статистичного навчання для задач визначення статистичних характеристик вхідного потоку даних з використанням спеціалізованих пакетів мови програмування Python та синтезовано модель даних. З допомогою selenium реалізовано парсинг біржі Nasdaq для отримання даних про зміну за рік індексу S&P 500. З допомогою numpy було побудовано лінію тренду і обчислено нормальні похибки для отримання синтетичних даних.

Виконав: студент Трикош І. В.