**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 «Технології Data Science»**

**Тема:**

**РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ПРОГНОЗУВАННЯ ДИНАМІКИ ЗМІНИ**

**ПОКАЗНИКІВ ЕФЕКТИВНОСТІ ТОРГІВЕЛЬНИХ КОМПАНІЙ**

**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:**

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

## **ІІ. Завдання:**

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

**ІІ рівень складності 9 балів.**

Відповідно до технічних умов, табл.2 додатку.



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

* 1. **Синтезована математична модель**

Відповідно до умов задачі синтезовано математичну модель, що реалізує аналіз даних про продажі.

Процес аналізу даних можна поділити на кілька етапів:

1. Збір даних – отримання даних з різних джерел.
2. Очищення та підготовка даних – видалення непотрібних, неповних чи некоректних даних.
3. Статистичний аналіз – розрахунок описової статистики.
4. Візуалізація даних з допомогою графіків та діаграм.
5. Побудова поліноміальної моделі – для прогнозування даних.
6. Оцінка моделі з допомогою RMSE (середньоквадратичне відхилення).
7. Прогнозування майбутніх значень продажів (на півроку).
8. Вивід результатів.

Усі ці кроки допомагають отримати корисну інформацію для прийняття можливих рішень, а також передбачити майбутні значення продажів.

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

На рис. 1 зображена блок-схема до програмного скрипту:

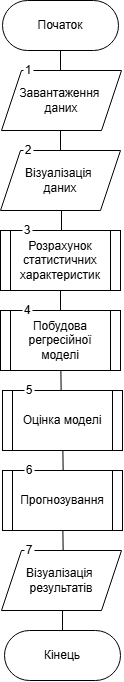


Рисунок 1 – Блок схема алгоритму

Робота алгоритму розпочинається зі зчитування даних із файлу та перевірки їх структури. Ці дії виконуються у блоці 1 блок-схеми алгоритму рис.1.

У блоці 2 відбувається візуалізація даних. Сюди входить візуалізація загального тренду продажів та продажів за кожне півріччя.

У блоці 3 розраховуються статистичні характеристики, а саме загальний обсяг продажів та прибутку та їх опис (середнє, мінімальне та максимальне, квантилі тощо).

У блоці 4 відбувається побудова поліноміальної регресійної моделі та її навчання.

У блоці 5 оцінка навченої моделі з допомогою середньоквадратичної похибки.

У блоці 6 відбувається прогнозування майбутніх значень продажів на наступні півроку.

У блоці 7 візуалізуються результати прогнозування. На цьому виконання алгоритму завершене.

* 1. **Опис структури проекту програми**

Для реалізації розроблених алгоритмів мовою програмування Python з використанням можливостей інтегрованого середовища PyCharm сформовано проєкт.

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

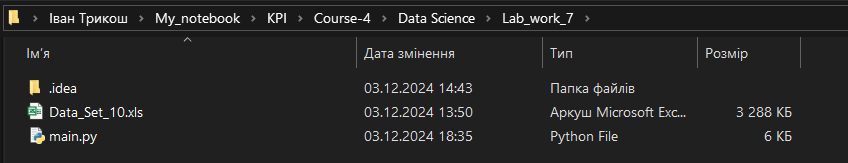


Рисунок 2 – Структура проєкту

Lab\_work\_7 – головний каталог проєкту

main.py – файл програмного коду

Data\_Set\_10.csv – дані про продажі

* 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 3-13 зображено результати роботи алгоритму:

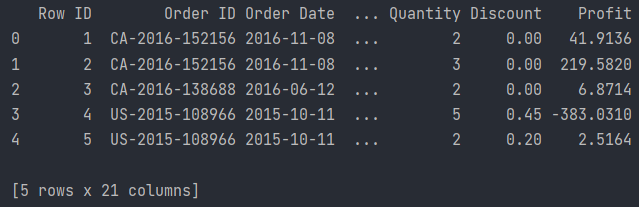


Рисунок 3 – Перші 5 рядків датасету

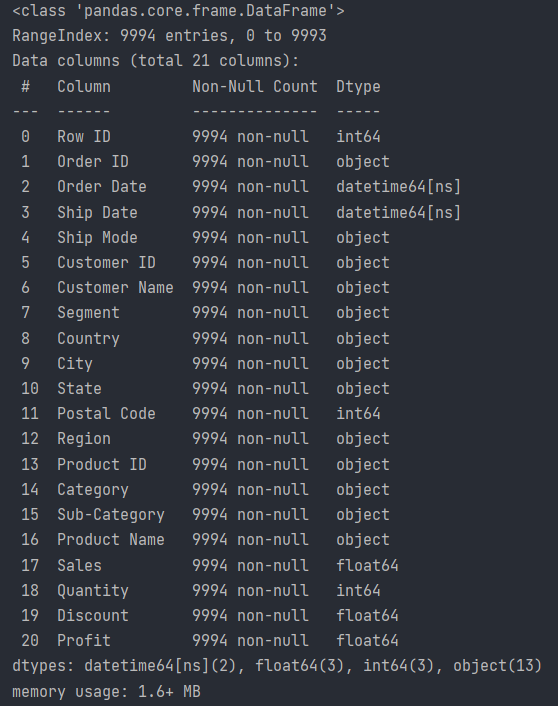


Рисунок 4 – Інформація про датафрейм

Бачимо, що пропущених значень немає, а потрібні нам стовпці Order Date, Sales та Profit уже мають відповідний тип.

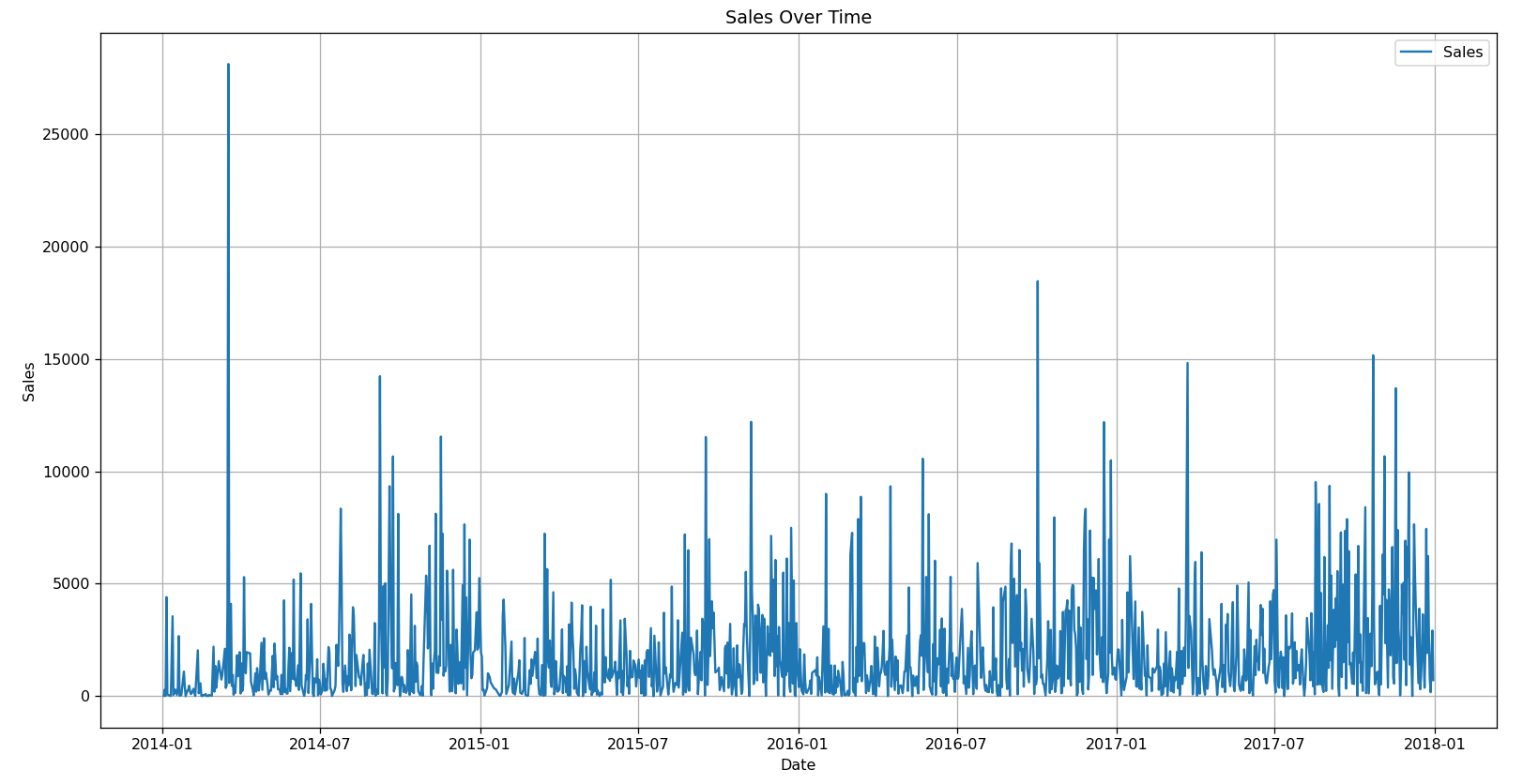


Рисунок 5 – Зміна продажів з часом

Бачимо, що продажі розподілені нерівномірно.

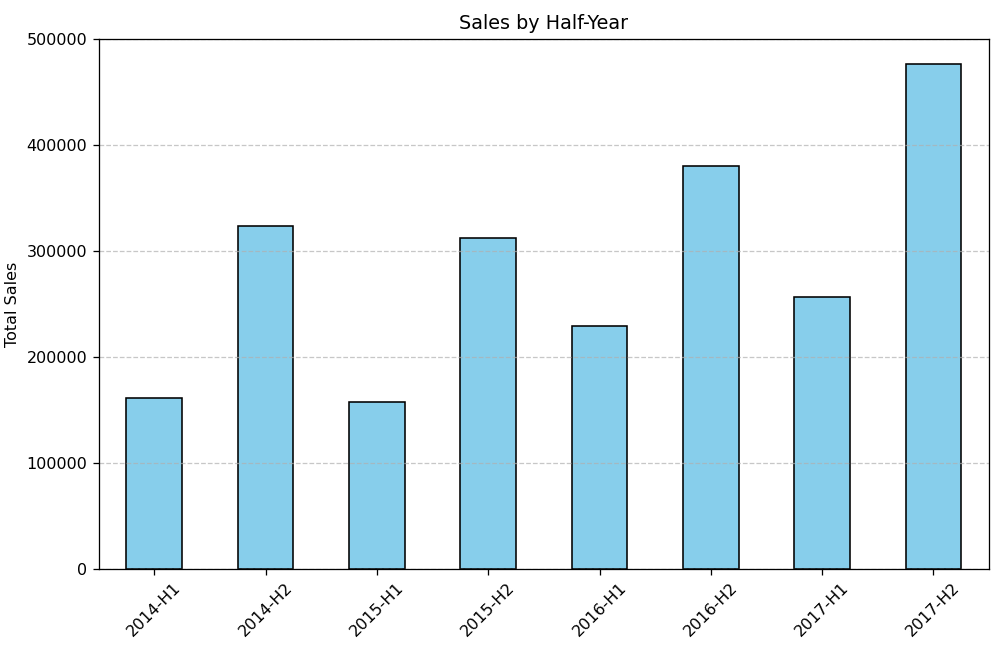


Рисунок 6 – Продажі за кожне півріччя

Тут помічаємо, що кількість продажів за друге півріччя порівняно більше, ніж за перше, причому сама кількість потроху росте.



Рисунок 7 – Загальні продажі та прибуток за весь час

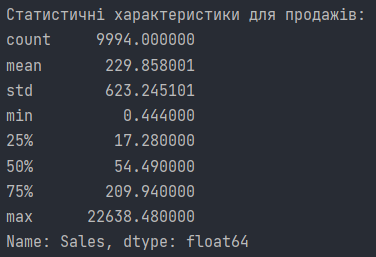


Рисунок 8 – Статистичні характеристики для продажів

Бачимо, що ця характеристика дуже нерівномірна. Більше ніж 75% значень є меншими за середнє, а різниця між максимальним та мінімальним велика, як і стандартне відхилення.

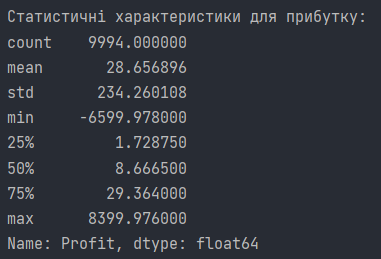


Рисунок 9 – Статистичні характеристики для прибутку

Тут ситуація дещо краща, проте все одно є великі коливання між значеннями. Тепер побудуємо поліноміальну регресійну модель 9-го порядку і навчимо її на даних про продаж.

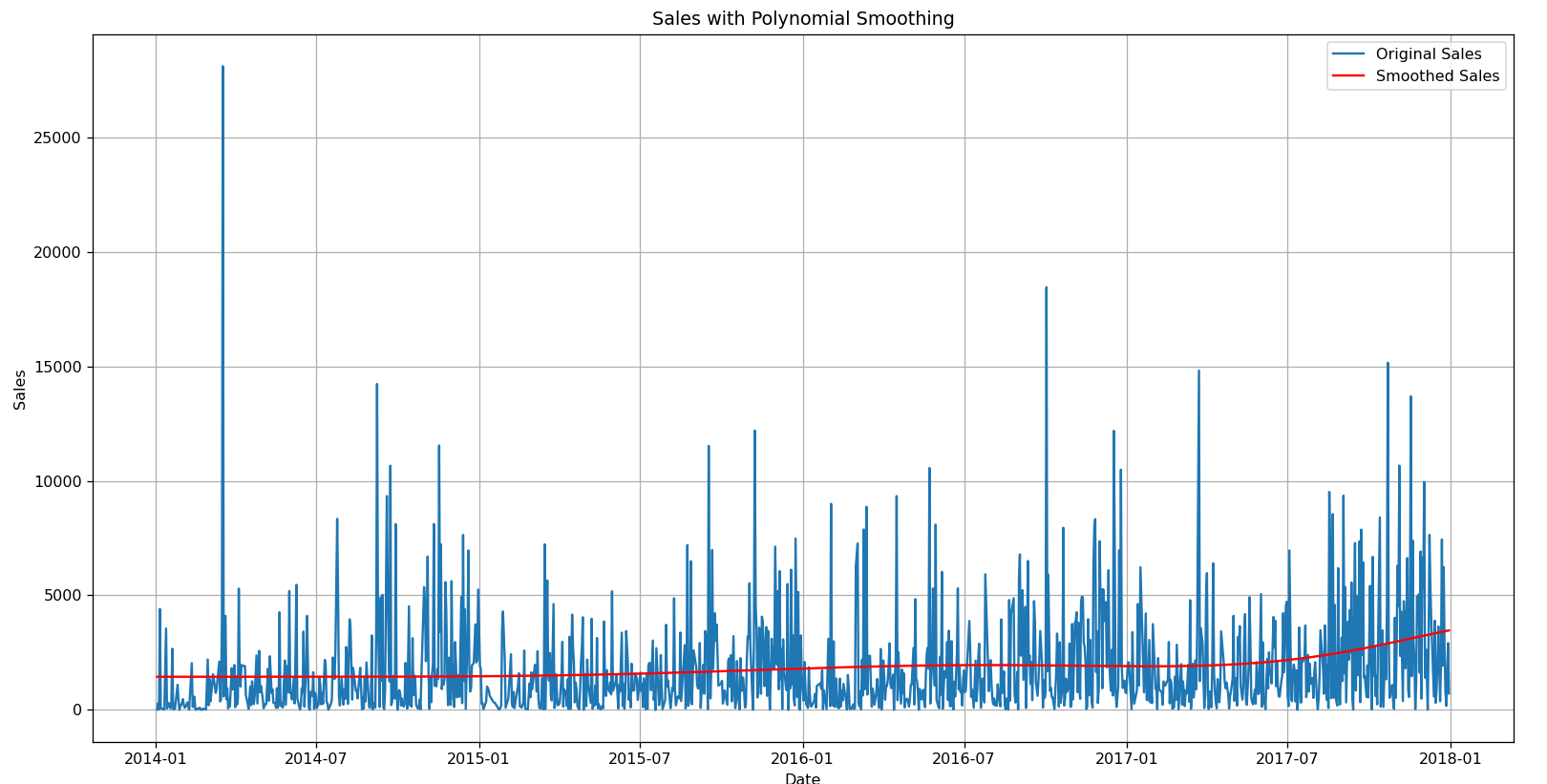


Рисунок 10 – Порівняння поліноміальної моделі з реальними значеннями

Фактично, поліноміальна модель згладила (усереднила) наші дані. Тепер перевіримо, наскільки велика похибка у навчанні.



Рисунок 11 – Значення середньоквадратичної похибки

Бачимо, що значення доволі велике, але це і не дивно, адже регресія не дуже підходить для аналізу часових рядів.

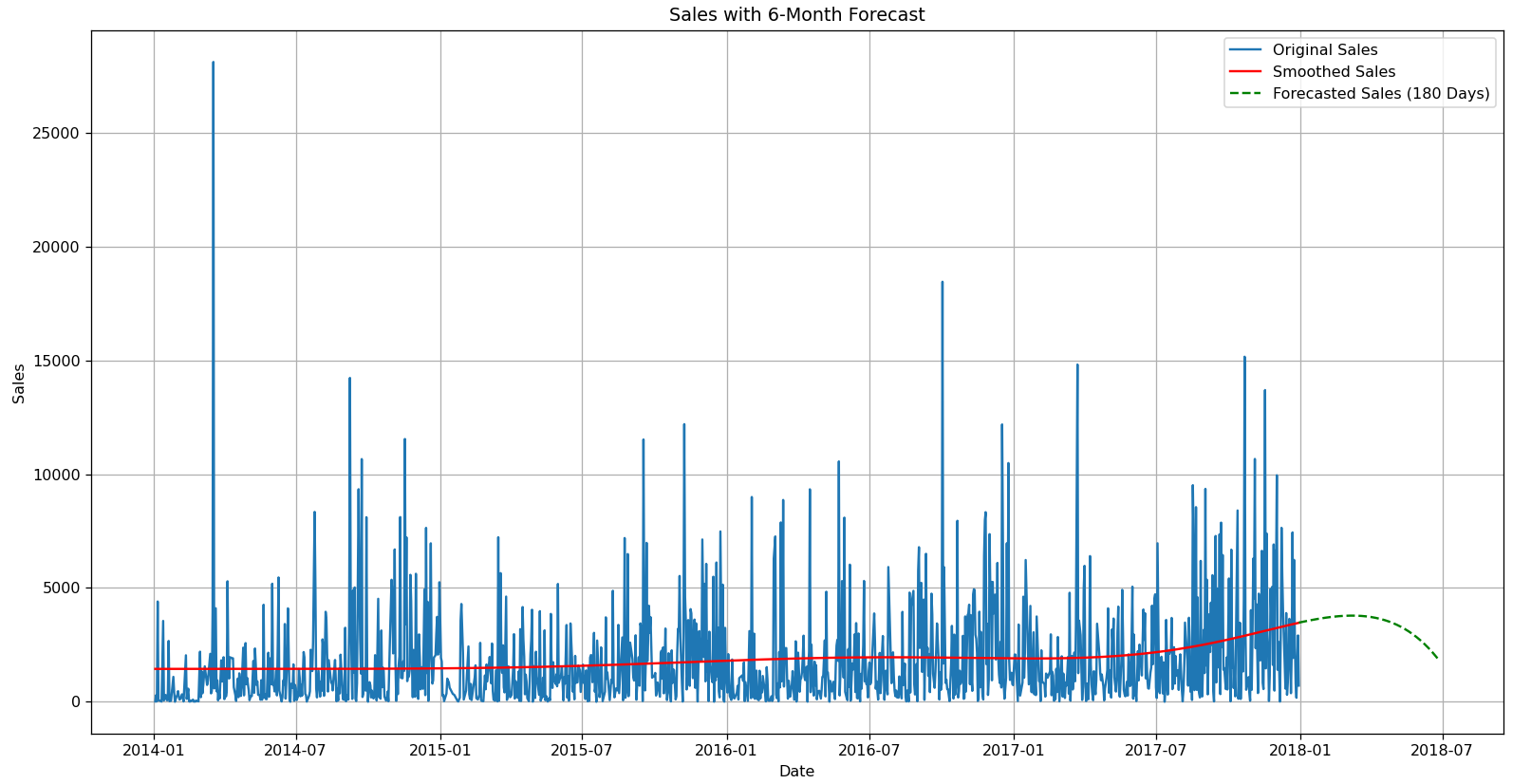


Рисунок 12 – Прогноз по продажах на наступні півроку

По графіку спостерігаємо, що продажі до кінця першого кварталу потроху ростимуть, а потім почнуть падати аж до кінця півріччя.

Виведемо тепер діаграму продажів по півріччям разом із прогнозом (останній стовпчик):

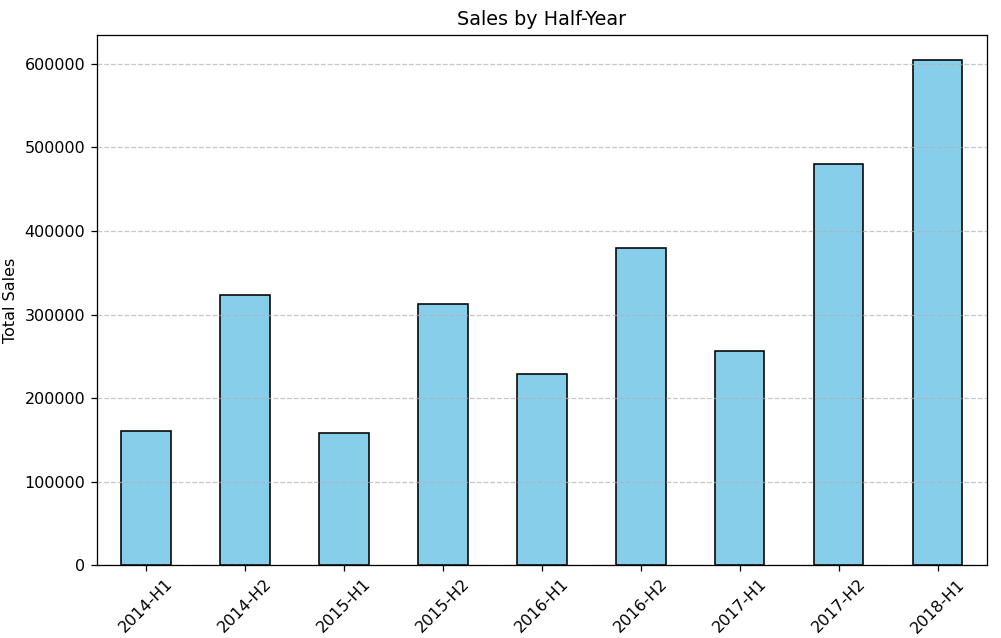


Рисунок 13 – Прогноз по продажах на наступні півроку

Бачимо, що прогнозовані продажі значно зростуть, у порівнянні з другим півріччям 2017, але це дещо не відповідає загальному тренду, адже у першому півріччі продажі зазвичай низькі.

* 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.3-13.

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

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

Контекстні коментарі пояснюють сутність окремих скриптів наведеного коду програми.

main.py

*import* pandas *as* pd  
*import* matplotlib.pyplot *as* plt  
*import* numpy *as* np  
*from* sklearn.linear\_model *import* LinearRegression  
*from* sklearn.preprocessing *import* PolynomialFeatures  
*from* sklearn.metrics *import* mean\_squared\_error  
  
*# Завантаження даних*file\_name = 'Data\_Set\_10.xls'  
data = pd.read\_excel(file\_name, sheet\_name='Orders')  
  
*# Перевірка структури даних*print(data.head())  
data.info()  
  
*# Візуалізація даних  
# Загальний тренд продажів*data\_grouped = data.groupby('Order Date')['Sales'].sum()  
plt.figure(figsize=(10, 6))  
plt.plot(data\_grouped.index, data\_grouped.values, label='Sales')  
plt.title('Sales Over Time')  
plt.xlabel('Date')  
plt.ylabel('Sales')  
plt.legend()  
plt.grid(*True*)  
plt.show()  
  
*# Продажі за кожне півріччя  
# Створення колонки для півріччя*data['Half-Year'] = data['Order Date'].dt.year.astype(str) + '-' + data['Order Date'].dt.month.apply(*lambda* x: 'H1' *if* x <= 6 *else* 'H2')  
  
*# Групування даних за півріччями та обчислення сумарних продажів*half\_year\_sales = data.groupby('Half-Year')['Sales'].sum().sort\_index()  
  
*# Побудова стовпчикової діаграми*plt.figure(figsize=(10, 6))  
half\_year\_sales.plot(kind='bar', color='skyblue', edgecolor='black')  
plt.title('Sales by Half-Year')  
plt.xlabel('Half-Year')  
plt.ylabel('Total Sales')  
plt.xticks(rotation=45)  
plt.grid(axis='y', linestyle='--', alpha=0.7)  
plt.show()  
  
  
*# Розрахунок продажів*total\_sales = data['Sales'].sum()  
print(f'Загальний обсяг продажів: {total\_sales}')  
  
*# Розрахунок прибутку*total\_profit = data['Profit'].sum()  
print(f'Загальний прибуток: {total\_profit}')  
  
*# Статистичні характеристики вибірки*print('Статистичні характеристики для продажів:')  
print(data['Sales'].describe())  
print('Статистичні характеристики для прибутку:')  
print(data['Profit'].describe())  
  
  
*# Поліноміальна регресія 9-го порядку для згладжування  
# Беремо дані по датах і продажах*sales\_data = data\_grouped.reset\_index()  
sales\_data['Days'] = (sales\_data['Order Date'] - sales\_data['Order Date'].min()).dt.days  
X = sales\_data[['Days']]  
y = sales\_data['Sales']  
  
*# Регресійна модель*poly = PolynomialFeatures(degree=9)  
X\_poly = poly.fit\_transform(X)  
poly\_model = LinearRegression()  
poly\_model.fit(X\_poly, y)  
  
*# Додавання згладжених продажів до даних*sales\_data['Sales Smoothed'] = poly\_model.predict(X\_poly)  
  
*# Візуалізація згладжування*plt.figure(figsize=(10, 6))  
plt.plot(sales\_data['Order Date'], sales\_data['Sales'], label='Original Sales')  
plt.plot(sales\_data['Order Date'], sales\_data['Sales Smoothed'], label='Smoothed Sales', color='red')  
plt.title('Sales with Polynomial Smoothing')  
plt.xlabel('Date')  
plt.ylabel('Sales')  
plt.legend()  
plt.grid(*True*)  
plt.show()  
  
  
*# Обчислення RMSE для оцінки моделі*rmse = mean\_squared\_error(sales\_data['Sales'], sales\_data['Sales Smoothed'], squared=*False*)  
print('RMSE: ', rmse)  
  
  
*# Прогнозування на пів року*future\_days = np.arange(sales\_data['Days'].max() + 1, sales\_data['Days'].max() + 181).reshape(-1, 1)  
future\_days\_poly = poly.transform(future\_days)  
future\_sales = poly\_model.predict(future\_days\_poly)  
  
*# Візуалізація прогнозу*plt.figure(figsize=(10, 6))  
plt.plot(sales\_data['Order Date'], sales\_data['Sales'], label='Original Sales')  
plt.plot(sales\_data['Order Date'], sales\_data['Sales Smoothed'], label='Smoothed Sales', color='red')  
future\_dates = pd.date\_range(sales\_data['Order Date'].max(), periods=180, freq='D')  
plt.plot(future\_dates, future\_sales, label='Forecasted Sales (180 Days)', linestyle='--', color='green')  
plt.title('Sales with 6-Month Forecast')  
plt.xlabel('Date')  
plt.ylabel('Sales')  
plt.legend()  
plt.grid(*True*)  
plt.show()  
  
  
*# Продажі за кожне півріччя + прогноз*future\_dates = [data['Order Date'].max() + pd.DateOffset(days=i) *for* i *in* range(1, 181)]  
forecast\_df = pd.DataFrame({  
 'Order Date': future\_dates,  
 'Sales': future\_sales  
})  
data\_ext = pd.concat([data, forecast\_df], ignore\_index=*True*)  
  
*# Створення колонки для півріччя*data\_ext['Half-Year'] = data\_ext['Order Date'].dt.year.astype(str) + '-' + data\_ext['Order Date'].dt.month.apply(*lambda* x: 'H1' *if* x <= 6 *else* 'H2')  
  
*# Групування даних за півріччями та обчислення сумарних продажів*half\_year\_sales = data\_ext.groupby('Half-Year')['Sales'].sum().sort\_index()  
  
*# Побудова стовпчикової діаграми*plt.figure(figsize=(10, 6))  
half\_year\_sales.plot(kind='bar', color='skyblue', edgecolor='black')  
plt.title('Sales by Half-Year')  
plt.xlabel('Half-Year')  
plt.ylabel('Total Sales')  
plt.xticks(rotation=45)  
plt.grid(axis='y', linestyle='--', alpha=0.7)  
plt.show()

## **IV. Висновки.**

У ході виконання лабораторної роботи було досліджено, виявлено та узагальнено особливості реалізації проектного практикуму в галузі аналізу часових (стохастичних рядів), як характеристика показників ефективності діяльності торгівельних компаній. Було виконано аналіз продажів компанії, побудовано та навчено регресійну модель, і зроблено прогноз продажів на наступні пів року.

Виконав: студент Трикош І. В.