Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Лабораторна робота №2

Прикладні задачі машинного навчання

Тема: Часові ряди і прості лінійна регресія

Виконав
студент групи ІП-11:
Панченко С. В.

Перевірив:
Нестерук А. О

Київ 2023

ЗМІСТ

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Мета роботи – дослідити лінійну регресію на прикладі прогнозування січневих температур у Нью-Йорку за рокам з використанням Python.

2 ЗАВДАННЯ

1. Завантажити метеорологічні дані в 1895-2022 роках з CSV-файлу в
2. Бібліотеку Seaborn використати для графічного представлення даних DataFrame у вигляді регресійної прямої, що представляє графік зміни обраних показників за період 1895-2018 років
3. Спрогнозувати дані на 2019, 2020, 2021 та 2022 рік
4. Оцінити за формулою, якими могли б бути показники до 1895 року
5. Скористатися функцією regplot бібліотеки Seaborn для виведення всіх точок даних
6. Виконати масштабування осі у
7. Порівняти отриманий прогноз для 2019, 2020, 2021 та за 2022 роки з даними на NOAA «Climate at a Glance»: https://www.ncdc.noaa.gov/cag/ і зробити висновок
8. Зробити звіт про роботу

3 ВИКОНАННЯ

3.1 Завантажити метеорологічні дані в 1895-2022 роках з CSV-файлу в DataFrame. Після цього дані відформатувати для використання

Зчитаємо даніз CSV-файлу, уикористовуючи метод read_csv.

In [27]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
pd.options.display.max_rows = 10
pd.options.display.max_columns = 10
nyc = pd.read_csv('data/ave_hi_nyc_jan_1895-2018.csv')
nyc
```

Out[27]:

	Date	Value	Anomaly
0	189501	34.2	-3.2
1	189601	34.7	-2.7
2	189701	35.5	-1.9
3	189801	39.6	2.2
4	189901	36.4	-1.0
...	...	...	...
119	201401	35.5	-1.9
120	201501	36.1	-1.3
121	201601	40.8	3.4
122	201701	42.8	5.4
123	201801	38.7	1.3

124 rows × 3 columns

Рисунок 3.1 - Завантаження датасету

Відформатуємо датафрейм, а саме: переназвемо стовпці та застосуємо цілочисельне ділення, поділивши значеннння років на 100.

In [28]:
```
nyc.columns = ['Date', 'Temperature', 'Anomaly']
nyc.Date = nyc.Date.floordiv(100)
nyc
```

Out[28]:

	Date	Temperature	Anomaly
0	1895	34.2	-3.2
1	1896	34.7	-2.7
2	1897	35.5	-1.9
3	1898	39.6	2.2
4	1899	36.4	-1.0
...	...	...	...
119	2014	35.5	-1.9
120	2015	36.1	-1.3
121	2016	40.8	3.4
122	2017	42.8	5.4
123	2018	38.7	1.3

124 rows × 3 columns

Рисунок 3.2 - Форматований датафрейм

Налаштужмо точність виведення чисел.

```
In [29]: pd.options.display.precision = 2
         pd.options.display.precision
```

Рисунок 3.3 - Налаштування точності виведення

Знайдемо основні статистичні показники.

```
In [30]: nyc.Temperature.describe()

Out[30]: count    124.00
         mean      37.60
         std        4.54
         min       26.10
         25%       34.58
         50%       37.60
         75%       40.60
         max       47.60
         Name: Temperature, dtype: float64
```

Рисунок 3.4 - Основні статистичні показники

3.2 Зображення лінійної регресії для 1895 по 2018.

Імпортуємо модуль stats з пакету scipy та за допомогою функції linregress знайдемо лінійну регресію, передавши в аргумети дати та температури.

```
In [31]: from scipy import stats
         linear_regression = stats.linregress(x=nyc.Date, y=nyc.Temperature)
```

Рисунок 3.5 - Розрахунок лінійної регресії

Розрахувавши регресію, дізнаємося про коефіцієнт нахилу.

```
In [32]: linear_regression.slope

Out[32]: 0.014771361132966163
```

Рисунок 3.6 - Коефіцієнт нахилу

Дізнаємося про точку перетину прямої лінії

```
In [33]: linear_regression.intercept

Out[33]: 8.694993233674289
```

Рисунок 3.7 - Точка перетину прямої лінії

Створимо функцію lin_predict, яка буде видавати спрогнозовані значення для лінійної регресії.

```
In [34]: def lin_predict(lin_regression, argument):
             return np.round(lin_regression.slope * argument + lin_regression.intercept, 2)
```

Рисунок 3.8 - Функція lin_predict

Імпортуємо бібліотеку seaborn та застосуємо функцію regplot для відображення лінійної регресії. Передамо в неї в якості аргумента дату, значення температуру та параметр за замовчуванням scatter=False, щоб відобразити лише пряму.

```
In [43]: import seaborn as sns
         years = np.array(range(1895, 2018))
         predict = lin_predict(linear_regression, years)
         fig, axes = plt.subplots(2)
         sns.lineplot(x=years, y=predict, ax=axes[0])
         sns.regplot(x=nyc.Date, y=nyc.Temperature, ax=axes[1], scatter=False)

Out[43]: <AxesSubplot: xlabel='Date', ylabel='Temperature'>
```

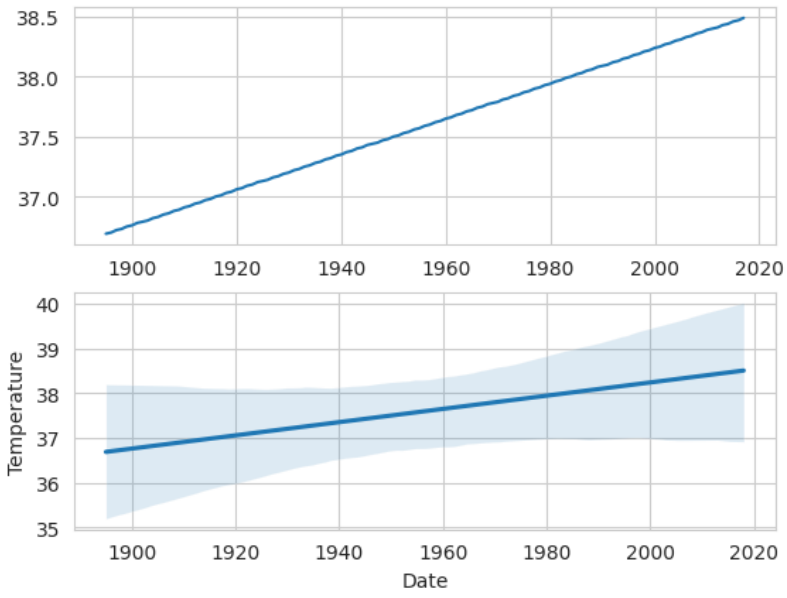

Рисунок 3.9 - Лінійна регресія для 1895 по 2018

3.3 Спрогнозувати дані на 2019, 2020, 2021 та 2022 рік

Спрогнозуємо дані для наступних років. Тобто підставимо роки у формулу лінійної регресії.

```
In [36]: years = np.array([2019, 2020, 2021, 2022])
         predict = lin_predict(linear_regression, years)
         df_predict = pd.DataFrame(dict(years=years, temperature=predict))
         df_predict
```

Out[36]:

	years	temperature
0	2019	38.52
1	2020	38.53
2	2021	38.55
3	2022	38.56

Рисунок 3.10 - Прогнозовані температури за роками

3.4 Оцінити за формулою, якими могли б бути показники до 1895 року

Обчислимо показники до 1895 року.

```
In [37]: years = np.array(range(1885, 1896))
         predict = lin_predict(linear_regression, years)
         df_predict = pd.DataFrame(dict(years=years, temperature=predict))
         df_predict
```

Out[37]:

	years	temperature
0	1885	36.54
1	1886	36.55
2	1887	36.57
3	1888	36.58
4	1889	36.60
...	...	...
6	1891	36.63
7	1892	36.64
8	1893	36.66
9	1894	36.67
10	1895	36.69

11 rows × 2 columns

Рисунок 3.11 - Показники температур з 1885 по 1895 роки включно

Як можна побачити, температура поступово зростає, і з періоду 1885 по 2023 роки спостерігається збільшення на 2 градуси.

3.5 Скористатися функцією regplot бібліотеки Seaborn для виведення всіх точок даних

Встановлюємо стиль відображення, побудуємо графік роки-температури. Побачимо, що дані доволі розкидані.

```
In [38]: sns.set_style('whitegrid')
         axes = sns.regplot(x=nyc.Date, y=nyc.Temperature)
```

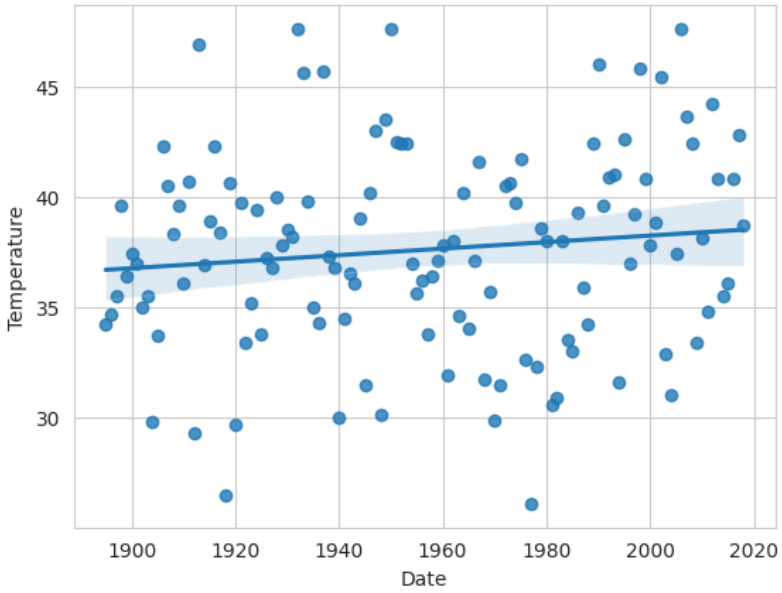

Рисунок 3.12 - Графік лінійної регресії роки-температури

3.6 Виконати масштабування осі y

За допомогою методу set_ylim вкажемо межі від 10 до 70 градусів.

```
In [39]: axes = sns.regplot(x=nyc.Date, y=nyc.Temperature)
         axes.set_ylim(10, 70)

Out[39]: (10.0, 70.0)
```

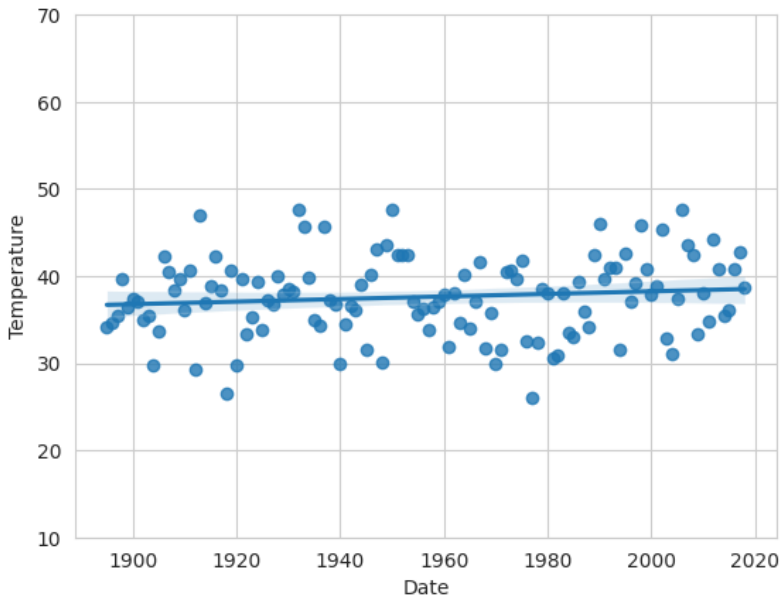

Рисунок 3.13 - Масштабований графік від 10 до 70 градусів

3.7 Порівняти отриманий прогноз для 2019, 2020, 2021 та за 2022 роки з даними на NOAA «Climate at a Glance»: https://www.ncdc.noaa.gov/cag/ і зробити висновок

Подивимося на сайті дані за період 2019-2022 років. Побачимо, що фактичні дані сильно відрізняються від того, що спрогнозувала лінійна регресія. Можна зробити висновок, що треба давати їй іще якісь дані для кращого прогнозування, наприклад: кількість опадів, кількість CO2 тощо.

```
In [40]: from PIL import Image
         im = Image.open('data/Screenshot from 2023-03-15 20-01-39.png')
         im
```

Out[40]:

New York, New York Average Temperature

January-December

Year	Average Temperature	Rank	Anomaly *1901-2000 Mean: 53.6°F*
2022	56.3°F	112	2.7°F
2021	56.9°F	121	3.3°F
2020	57.3°F	125	3.7°F
2019	55.6°F	105	2.0°F
2018	55.9°F	108	2.3°F

Рисунок 3.14 - Справжні зафіксовані дані середніх температур

4 ВИСНОВОК

Під час виконання цієї лабораторної роботи здобув базові навички використання пакету scipy мови Python, досліджуючи середні температури в січня у Нью-Йорку з 1895 до 2022 років, обчисливши лінійну регресію та зробивши прогноз. У результаті спрогнозовані дані не збігалися з фактичними даними. Отже, точність низька та потрібно врахувати додаткові параметри.