

Передбачення погоди

В роботі необхідно скористатись датасетом

- <https://www.kaggle.com/code/alessandromajumba/iea-model-world-weather-repositor> у або
- https://www.meteoblue.com/uk/weather/archive/export?daterange=2022-09-04%20-%202024-03-19&locations%5B%5D=basel_switzerland_2661604&domain=ERA5T&min=2023-01-01&max=2024-03-19¶ms%5B%5D=¶ms%5B%5D=temp2m¶ms%5B%5D=¶ms%5B%5D=¶ms%5B%5D=¶ms%5B%5D=¶ms%5B%5D=¶ms%5B%5D=¶ms%5B%5D=&utc_offset=%2B00%3A00&timeResolution=hourly&temperatureunit=CELSIUS&velocityunit=KILOMETER_PER_HOUR&energyunit=watts&lengthunit=metric°ree_day_type=10%3B30&gddBase=10&gddLimit=30 - дані по Базелю, Швейцарія

Це дані, отримані з датчиків та записані в БД для обробки результатів.

Задача залежить від кратності варіанту. Той, чий номер в списку ділиться на відповідний номер варіанту без остачі - бере собі відповідне передбачення.

В таблиці є дані з 29.08.23 по 06.03.24.

- 1) Передбачення погоди (температури) на наступний день. День потрібно вибрати один з існуючих, щоб можна було зрозуміти, чи вірне ваше передбачення. Підходить 2гий датасет.
- 2) Передбачення погоди на наступний рік. Рік теж треба вибрати один з існуючих. Температуру писати на день: 5-11 градусів цього дня
- 3) Передбачення погоди на 20 років вперед. (теж по 2гому датасету)
- 4) Аналітика даних по вітру з візуалізацією мінімум в 3х проекціях (в які періоди він сильний, коли слабкий, графік його сили, коли він впливає на температуру, як температура відчувається (холодніше, ніж насправді на 2 градуси, чи тепліше, та подібне). перший датасет.
- 5) Аналітика по якості повітря - візуалізація (коли можна алергикам виходити, коли берегтись, як змінюється в залежності від сезону, чи є зв'язок з іншими параметрами погоди). перший датасет.

Вибір техніки передбачення лежить на вас - алгоритми або машинне навчання.

За використання додаткових технологій аналогічно до минулої лаби будуть додаткові бали. Технології мають бути відмінними від минулої лаби, щоб бали були підвищені. Якщо в минулий раз було використано Монго, то тепер виберіть іншу БД.

Рекомендовані технології, фреймворки на додаткові бали

- 1) Flyway або Liquibase або інші інструменти міграції БД. Вибрати можна тут <https://www.ropstam.com/best-database-schema-migration-tools/> За цим посиланням взагалі про міграцію даних: <https://www.astera.com/type/blog/data-migration-tools/> Як скористатись цим тулом:

1. Створити необхідну структуру БД.
2. завантажити дані з файлу в (усе ж реляційну) БД
3. додати чейндж сети на додавання таблиць чи можливо колонок для вирішення вашої задачі.
- 2) Тести - інструмент, який дає можливість впевнитись, що ваша програма робоча. Тести обов'язкові, як завжди. В цій програмі хотілось би, щоб не залежно від вибраної техніки реалізації ви б долучили до програми Sanity Checks - тести "здорового глузду" вашої програми: <https://www.geeksforgeeks.org/sanity-testing/>. Вони можуть включати перевірку Health Check - що кожен модуль програми працює, зв'язок з базою працює, МЛ модуль працює, інтерфейс реагує, і т.п. Ось описання Health Check: <https://alhardy.github.io/app-metrics-docs/getting-started/health-checks/index.html> Якщо у вас алгоритм обрахування нових значень, то протестований має бути алгоритм. Якщо використовується МЛ - необхідна оцінка якості показників, це теж вважається свого роду тестом.
- 3) Можна зробити модульність програми - застосувати модульну чи мікросервісну архітектуру, якщо модулів декілька: окремо МЛ частина, окремо запуск програми, окремо БД та окремо ЮІ,...
<https://javarush.com/ua/quests/lectures/ua.questservlets.level14.lecture05>

Взагалі про передбачення погоди:

<https://www.yachtingmonthly.com/sailing-skills/how-to-build-your-own-weather-forcecast-73104>

Поради щодо передбачень

Для передбачення сумою функцій

Тим хто хоче зробити передбачення наближаючи температурні коливання графіками функцій, зокрема, синусами, але не виходило. Пропоную вашій увазі мою версію наближення функціями, але спрощену, адже не використовувала реальні дані, але будувала тільки алгоритм наближення. Важливо розуміти, що любі коливання низького порядку (за день та декадні) слід накладати на коливання більш високого порядку (річні + глобальне потепління, яке задає вектор коливань). Під накладанням слід мати на увазі суму графіків. Для обчислення передбачень важливо їх нормувати ділячи на кількість понять, які ви об'єднуєте: (Денні + річні + декадні)/3.

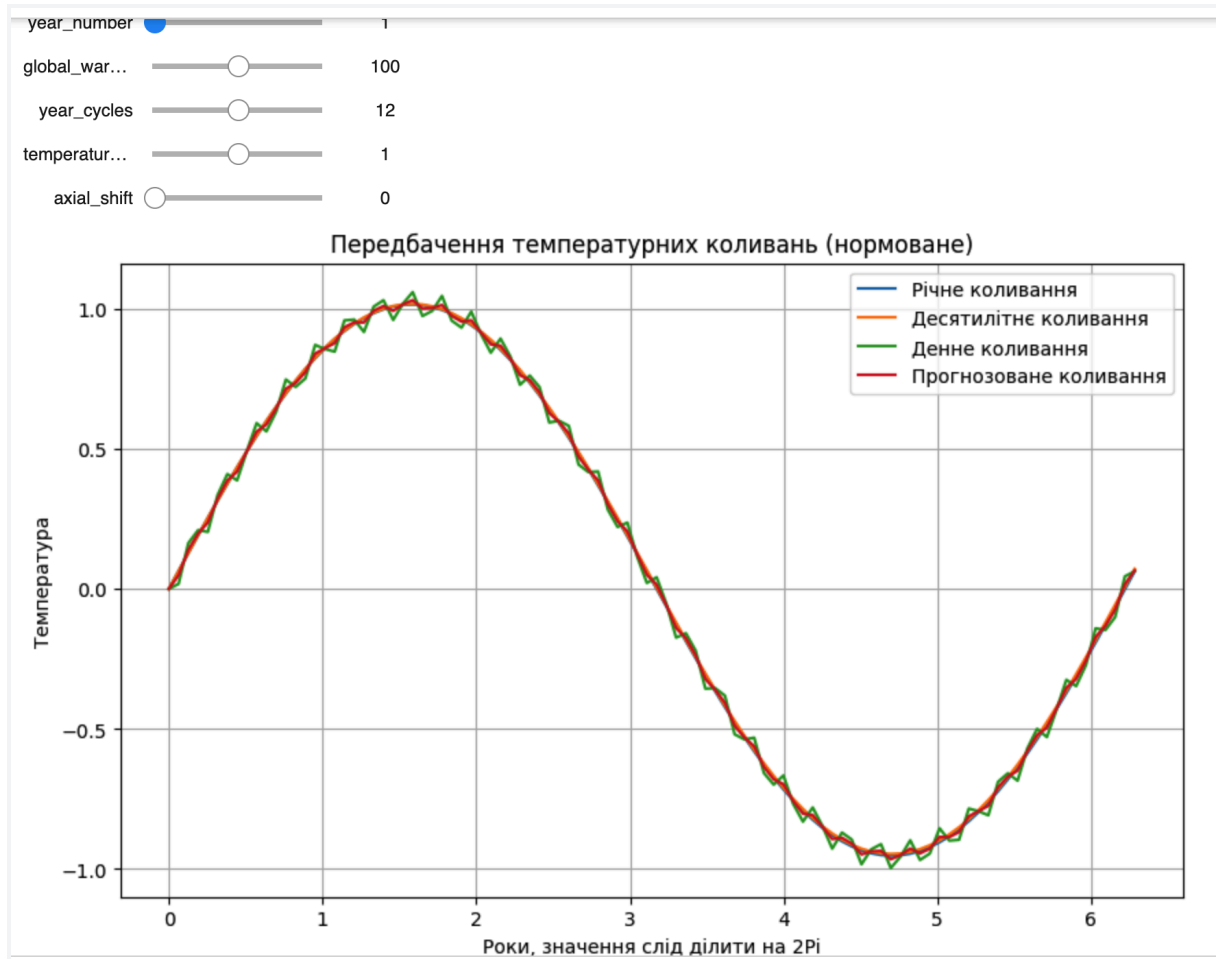
Для передбачення регресією

Якщо ви користуєтесь регресією, то майте на увазі, що всі показники температури за роки їй не можна давати, адже її робота - провести лінію на оптимально-середній відстані від усіх точок, де спостерігатиметься сумарне найбільше скупчення. Стосовно температури, це близько 8-11 градусів. Але ж ці показники нерелевантні. Лінійну регресію слід застосовувати

тільки коли точки будуть на одній прямій. Наприклад, температура знижується з літа до зими і піднімається з зими до літа. Можна побудувати 2 регресії між цими проміжками на кожен рік. Або ж при аналізі даних з 1940 на 1 день, взяти дані за цей 1 день за всі роки і на їх основі побудувати регресію.

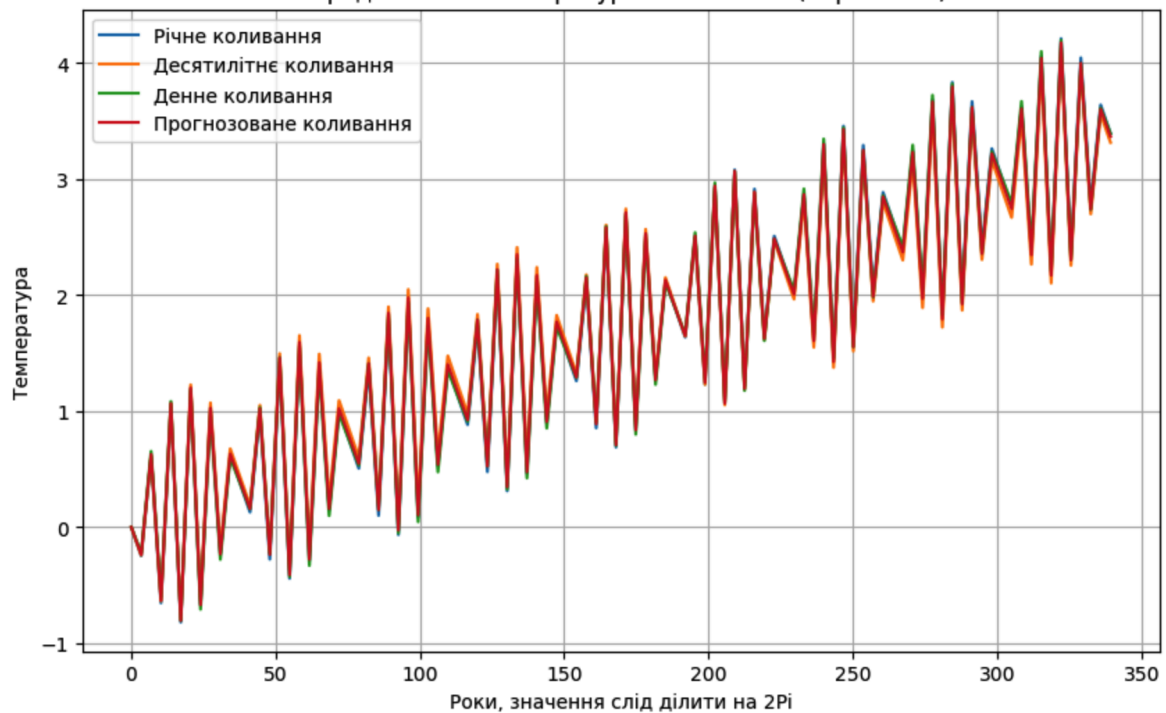
Також можна застосовувати розкладення Фур'є, машинне навчання.

Пам'ятайте, немає поганих методів та алгоритмів, є їх некоректне застосування та помилки при розрахунках.



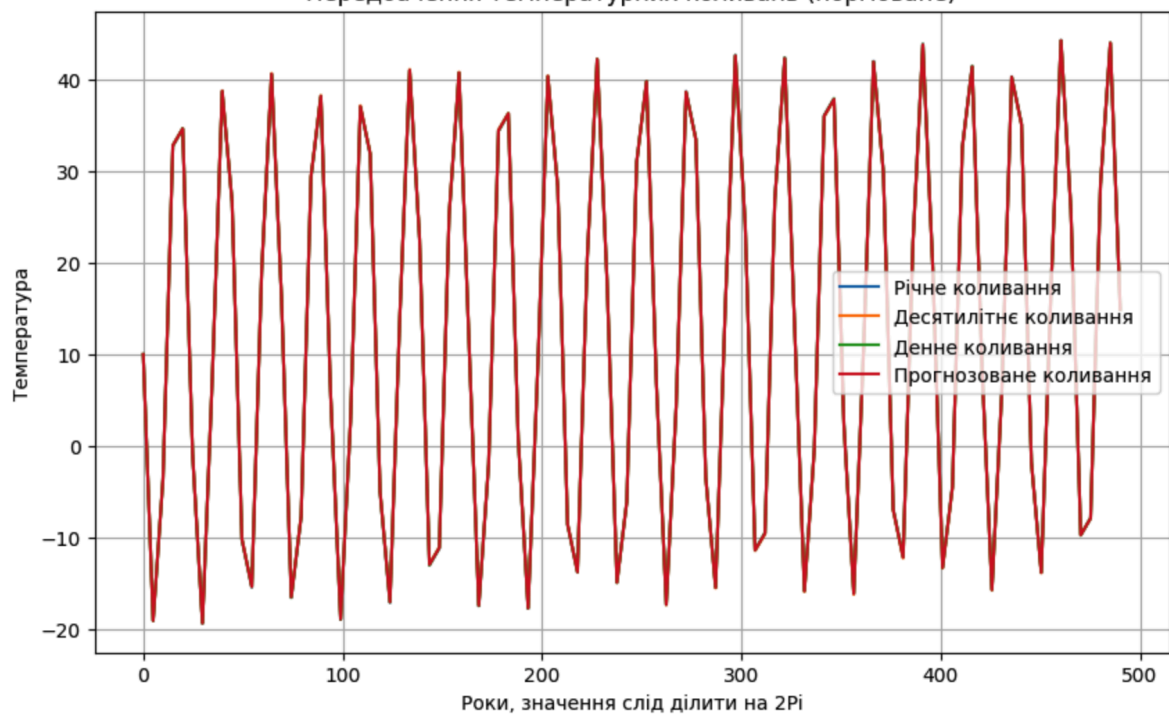
year_number 54
global_war... 100
year_cycles 10

Передбачення температурних коливань (нормоване)



year_number 78
global_war... 100
year_cycles 12

Передбачення температурних коливань (нормоване)



Код

```
%pip install -q ipywidgets

import numpy as np
import matplotlib.pyplot as plt

from ipywidgets import interactive

# Кількість років

def predict_temperature(year_number=30, global_warming_rate = 100,
year_cycles = 12, temperature_change = 1, axial_shift = 0):
    x = np.linspace(0, year_number*2*np.pi, 100)

    # Функція глобального потепління
    global_warming_func = x/global_warming_rate

    # Функція річного коливання - sin(x), адже вона враховує одне літо
та одну зиму, починаючись з весни.
    # temperature_change - амплітуда температур поділена на 2. Якщо =
30, то змінюється від -30 до 30.
    # axial_shift - наскільки зсунута амплітуда температур
temperature_change * 2 відносно нуля. Якщо =5, то -25 до 35.
    year_temp_change = temperature_change * np.sin(x) +
global_warming_func + axial_shift

    # Функція зміни більш екстремальними роками та більш помірними кожні
10 років
    decage_temp_change = np.sin(x/year_cycles/2/np.pi)/10 +
year_temp_change

    # Функція денного коливання температури. В ідеалі, має бути стільки
коливань, скільки днів в році.
    # Ділення на 20 дає зменшення амплітуди, тобто денні коливання не
перевищують 5% сезонного коливання.
    # Важливо накласти цю функцію на рік, адже денні коливання
відбуваються в межах річних.
    day_temp_change = (np.sin(365*x)/20 + year_temp_change)

    # Результируючий графік передбачення погодних коливань. Ділення на
два, адже ми складаємо 2 поняття - денне та річне,
    # але амплітуда має залишитись
```

```
temp_prediction_func = (year_temp_change + decage_temp_change +
day_temp_change)/3

# Побудова графіків
plt.figure(figsize=(10, 6))
plt.plot(x, year_temp_change, label='Річне коливання')
plt.plot(x, decage_temp_change, label='Десятилітнє коливання')
plt.plot(x, day_temp_change, label='Денне коливання')
plt.plot(x, temp_prediction_func, label='Прогнозоване коливання')
plt.xlabel('Роки, значення слід ділити на 2Pi')
plt.ylabel('Температура')
plt.title('Предбачення температурних коливань (нормоване)')
plt.legend()
plt.grid(True)
plt.show()

w=interactive(predict_temperature,year_number=(1,100))
w
```