

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки

«Затверджую»
в.о. завідуючого кафедри
комп'ютерних систем та робототехніки
_____ к. ф.-м. н., доцент Максим Хруслов
«___» червня 2025 р.

Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: **«МОДЕЛЬ ЗАСТОСУНКУ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО
ПРОГНОЗУВАННЯ ПОГОДИ НА ОСНОВІ ШІ»**

Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології
Галузь знань 15 – Автоматизація та приладобудування
Освітня програма «Автоматизація та комп'ютерно-інтегровані технології»

Захищено на засіданні
Екзаменаційної комісії № 46
протокол № __ від __.06.2025 р.
Оцінка ____ / ____
Голова Екзаменаційної комісії
_____ **ЧУГАЙ А.М.**

Виконав:
Студент(ка) групи КУ– 41
СЄЧКО Валерія Валеріївна 

Керівник: професор кафедри, доктор
технічних наук
МІРОШНИК Марина Анатоліївна 

Рецензент: професор ЗВО кафедри
Інформаційних технологій
Українського державного
університету залізничного
транспорту, д.т.н., проф.
ДОЦЕНКО С. І. 

Харків – 2025

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 50 сторінок, з яких 30 сторінок основної частини, що містить 5 рисунків, 1 таблиця, 18 найменувань у списку використаних джерел та 4 додатки.

Метою кваліфікаційної роботи є розробка моделі інтелектуального прогнозування погоди з використанням методів машинного навчання на основі мови програмування Python.

Об'єкт дослідження – процес прогнозування метеорологічних показників на основі історичних погодних даних.

Предмет дослідження – математичні моделі машинного навчання та програмні засоби обробки, візуалізації й оцінювання точності прогнозів.

Проблема, що розв'язується у кваліфікаційній роботі, полягає в створенні простої, адаптивної моделі прогнозування температури повітря з використанням регресії хребта (Ridge Regression), яка здатна забезпечити достатній рівень точності та бути масштабованою для подальших досліджень.

Область застосування – побутові, освітні, наукові та виробничі сфери, що потребують доступного та ефективного прогнозування температури повітря. Модель може бути інтегрована до мобільних застосунків, інфраструктури “розумного міста”, аграрного сектору, логістики тощо.

Ключові слова: *прогноз погоди, машинне навчання, Python, Ridge Regression, метеодані, температурний аналіз, MAE, моделювання, Scikit-learn.*

ABSTRACT

The explanatory note to the bachelor's qualification thesis consists of an introduction, three chapters, conclusions, a list of references, and appendices. The total volume of the work is 50 pages, of which 30 pages comprise the main content, including 5 figures, 1 table, 17 references, and 4 appendixes.

The aim of the qualification work is to develop a weather forecasting model using machine learning methods implemented in the Python programming language.

The object of the study is the process of forecasting meteorological indicators based on historical weather data.

The subject of the study includes machine learning models and software tools for processing, visualizing, and evaluating forecast accuracy.

The problem addressed in the thesis is to create a simple and adaptive air temperature forecasting model using Ridge Regression, capable of achieving sufficient prediction accuracy and being scalable for further research.

The field of application includes household, educational, scientific, and industrial domains that require accessible and efficient air temperature forecasting. The model can be integrated into mobile applications, smart city infrastructure, agriculture, logistics, and other sectors.

Keywords: *weather forecasting, machine learning, Python, Ridge Regression, meteorological data, temperature analysis, MAE, modeling, Scikit-learn.*

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО ПРОГНОЗУВАННЯ ПОГОДИ З ВИКОРИСТАННЯМ ІШ	7
1.1. Традиційні та сучасні методи прогнозування погоди.....	7
1.2. Застосування машинного навчання в метеорології.....	10
1.3. Джерела метеоданих та програмні інструменти.....	12
Висновки до розділу 1	14
РОЗДІЛ 2. РОЗРОБКА ТА РЕАЛІЗАЦІЯ МОДЕЛІ ПРОГНОЗУВАННЯ ПОГОДИ З ВИКОРИСТАННЯМ ІШ.....	15
2.1. Постановка задачі	15
2.2. Обмеження та вхідні припущення	16
2.3. Підготовка даних.....	17
2.4. Обґрунтування вибору технічних засобів	20
2.5. Побудова та тренування моделі.....	21
Висновки до розділу 2	24
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ, ПЕРСПЕКТИВИ РОЗВИТКУ МОДЕЛІ	25
3.1. Програмна реалізація моделі прогнозування погоди	25
3.2. Перспективи розвитку моделі та її прикладне значення	28
Висновки до розділу 3	31
ВИСНОВКИ.....	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	34
ДОДАТКИ.....	36
Додаток А.....	36
Додаток В.....	42
Додаток Г	46

ВСТУП

У сучасному світі, де кліматичні зміни набувають глобального масштабу, а погодні умови стають дедалі більш нестабільними та непередбачуваними, потреба в точному та оперативному прогнозуванні погоди виходить за межі лише метеорологічної науки. Надійна інформація про майбутні погодні умови є критично важливою для прийняття рішень у багатьох галузях: сільському господарстві, авіації, морських перевезеннях, будівництві, енергетиці та навіть у повсякденному житті пересічних громадян.

Успішне передбачення екстремальних погодних явищ, таких як зливи, буревії, засухи чи заморозки, дозволяє запобігати людським втратам і значним економічним збиткам. Водночас традиційні чисельні методи прогнозування базуються на складних фізико-математичних моделях атмосфери, які потребують значних обчислювальних ресурсів, часу та високоточних даних. Їх точність часто знижується на довгострокових проміжках і в умовах нестабільного клімату.

З огляду на це, впровадження методів штучного інтелекту, які здатні автоматично виявляти закономірності у великих обсягах історичних метеоданих, відкриває нові перспективи для створення адаптивних, швидкодіючих і точних систем прогнозування.

Актуальність дослідження обумовлена потребою у більш гнучких, ефективних та інтелектуальних інструментах прогнозування, які здатні оперативно реагувати на мінливі умови довкілля. Розробка програмної моделі, яка поєднує сучасні підходи до аналізу даних та прогнозування, є важливим кроком на шляху до підвищення якості погодних сервісів.

Метою роботи є розробка моделі застосунку для інтелектуального прогнозування погоди на основі методів штучного інтелекту, що забезпечить підвищену точність прогнозів, адаптивність до регіональних особливостей і можливість масштабування.

Об'єктом дослідження виступає процес прогнозування погодних умов.

Предмет дослідження – методи та алгоритми штучного інтелекту, що застосовуються для інтелектуального аналізу метеоданих та формування прогнозів погоди.

Методи дослідження: описові статистичні методи, методи попередньої обробки даних (зокрема forward fill), методи машинного навчання (Ridge-регресія), методи оцінки точності моделей (MAE).

Завдання дослідження

1. Виконати аналіз наукової літератури та існуючих підходів до прогнозування погоди за допомогою ІШ.
2. Дослідити доступні відкриті джерела метеоданих і методи їх обробки.
3. Визначити оптимальні алгоритми машинного навчання для реалізації моделі прогнозування погоди.
4. Розробити концептуальну модель застосунку.
5. Провести тестування моделі та оцінити точність прогнозів.
6. Визначити можливості та перспективи подальшого розвитку моделі.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО ПРОГНОЗУВАННЯ ПОГОДИ З ВИКОРИСТАННЯМ ШІ

1.1. Традиційні та сучасні методи прогнозування погоди

Прогнозування погоди – одна з найважливіших задач прикладної метеорології, яка має вирішальне значення для функціонування транспорту, сільського господарства, енергетики та управління надзвичайними ситуаціями. Протягом десятиліть основними методами прогнозування залишались фізико-математичні моделі, які базуються на системі рівнянь гідродинаміки атмосфери, включно з рівняннями Нав'є–Стокса, законами термодинаміки та переносу маси й енергії [13].

Одним із найдавніших методів є синоптичне прогнозування, що базується на аналізі метеорологічних карт та емпіричному досвіді синоптиків. Попри його ефективність для короткострокових прогнозів, цей підхід не дозволяє повноцінно моделювати складну динаміку атмосфери. З середини XX століття домінуючою парадигмою стало чисельне прогнозування погоди (NWP), яке полягає в обчислювальному розв'язанні згаданих рівнянь на сітці, що охоплює всю атмосферу [4].

Сучасні NWP-моделі використовують великі масиви спостережень, зокрема з наземних метеостанцій, супутників і радарів, що дозволяє покращити початкові умови для обчислень. Моделі поділяються на глобальні (наприклад, GFS, ECMWF) і регіональні (WRF, ALARO), які відрізняються просторовою роздільною здатністю і сферами застосування. Для врахування процесів, які не можна змоделювати напрямку (наприклад, конвекції або турбулентності), використовуються фізичні параметризації [13].

Однак традиційні моделі мають низку обмежень. По-перше, вони вимагають надзвичайно потужних обчислювальних ресурсів. По-друге, точність прогнозу значною мірою залежить від якості початкових умов і структури самої моделі. Нарешті, хаотична природа атмосфери обмежує точність довгострокового

прогнозування – зазвичай понад 7–10 днів прогноз стає статистично невизначеним [8].

Ці недоліки стимулювали розвиток альтернативних підходів, зокрема методів штучного інтелекту (ШІ). Машинне навчання дозволяє будувати моделі прогнозування, що ґрунтуються на історичних метеоданих, без явного розв’язання фізичних рівнянь. Основною перевагою ШІ є здатність адаптуватися до нових даних, виявляти приховані залежності між параметрами погоди та забезпечувати високу швидкість обробки [2].

У сучасній метеорології активно досліджуються моделі глибокого навчання – зокрема згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), які демонструють здатність ефективно опрацьовувати великі масиви часово-просторових метеоданих. Такі архітектури добре підходять для задач прогнозування, оскільки дозволяють враховувати як часову послідовність спостережень (через механізми пам’яті у RNN, LSTM), так і просторову взаємодію між регіонами (завдяки локальним фільтрам у CNN).

У межах проєкту WeatherBench було створено еталонний набір даних для оцінки моделей прогнозування погоди на основі машинного навчання, що включає глобальні атмосферні змінні з високою роздільністю. Це дозволило стандартизувати порівняння результатів та забезпечити спільну платформу для розробників моделей нового покоління.

Експерименти, проведені в рамках WeatherBench, показали, що деякі моделі глибокого навчання здатні досягати рівня точності, порівнянного або навіть вищого, ніж традиційні чисельні методи прогнозування погоди (NWP – Numerical Weather Prediction). При цьому моделі ШІ забезпечують значно нижче споживання обчислювальних ресурсів і швидший час обробки, що робить їх привабливими для оперативного прогнозування, зокрема в умовах обмежених ресурсів або мобільного використання.

Такий підхід відкриває перспективи побудови нових гібридних систем, які поєднують фізичні моделі атмосфери з навчанням на даних, що дозволяє

одночасно враховувати фізику процесів і гнучкість адаптивного моделювання [14].

Ще одним поширеним прикладом використання штучного інтелекту в метеорології є застосування регресійних моделей, зокрема методів з регуляризацією, таких як Ridge-регресія. Цей підхід особливо ефективний у задачах, де вхідні змінні демонструють високу мультиколінеарність, тобто сильні лінійні зв'язки між собою, що може негативно впливати на стабільність і точність класичних моделей лінійної регресії. Регуляризація в Ridge-регресії додає до функції помилки додатковий член, який штрафує великі вагові коефіцієнти, зменшуючи їхній вплив та запобігаючи перенавчанню. Це дає змогу створювати стійкі моделі, здатні забезпечувати прийнятну точність навіть за умов обмеженої кількості даних або високої варіативності атмосферних показників [11].

Завдяки своїй простоті, прозорості результатів та невисоким обчислювальним витратам, регресійні підходи залишаються популярними для побудови локальних моделей прогнозування, орієнтованих, наприклад, на температуру повітря, вологість чи кількість опадів. Зокрема, моделі такого типу зручно використовувати в рамках автоматизованих систем прогнозування для окремих населених пунктів або регіонів з обмеженою кількістю доступних метеостанцій. При правильному налаштуванні Ridge-регресія може досягати точності прогнозу в межах похибки до 3°C , що є задовільним рівнем для прикладного використання [11].

Однак, з огляду на складність сучасних кліматичних процесів, зростаючий обсяг даних та потребу в багатовимірному прогнозуванні, у науковій спільноті спостерігається поступовий перехід до більш складних і гнучких підходів. Зокрема, все більше дослідників визнають потенціал гібридних моделей, які поєднують фізично обґрунтовані чисельні методи (Numerical Weather Prediction – NWP) із методами машинного навчання. Така інтеграція дозволяє об'єднати детерміновану природу фізичних моделей, що базуються на рівняннях Нав'є –

Стокса та термодинамічних законах, із адаптивною здатністю ІІІ до виявлення прихованих закономірностей у даних [18].

Гібридні підходи відкривають можливість використання машинного навчання як постпроцесора для корекції результатів NWP, як окремого модуля для уточнення короткострокових прогнозів або як інструмента для побудови ймовірнісних сценаріїв. Наприклад, сучасні експерименти в межах проєкту *WeatherBench* засвідчили, що алгоритми глибокого навчання – зокрема згорткові нейронні мережі – можуть досягати точності, порівнянної або навіть вищої за класичні NWP-моделі, при цьому споживаючи суттєво менше обчислювальних ресурсів [14; 5].

Це відкриває шлях до широкого застосування таких моделей у системах, де обчислювальні потужності обмежені або потрібно забезпечити прогноз у реальному часі. Таким чином, гібридні моделі є не лише компромісом між точністю та продуктивністю, а й потенційною новою парадигмою в метеорологічному прогнозуванні, що може забезпечити високоякісні прогнози за умов динамічної зміни клімату та високої невизначеності.

Отже, прогнозування погоди еволюціонує від виключно фізико-математичних моделей до гнучкіших та швидкодіючих інтелектуальних систем, здатних не лише покращити точність прогнозів, а й зробити їх доступнішими для широкого кола користувачів.

1.2. Застосування машинного навчання в метеорології

Упродовж останнього десятиліття машинне навчання (ML) стало потужним інструментом у метеорології, відкриваючи нові можливості для прогнозування погоди. На відміну від традиційних фізико-математичних моделей, ML-алгоритми здатні виявляти складні нелінійні залежності між атмосферними параметрами, використовуючи великі обсяги історичних та реального часу даних. Це дозволяє створювати моделі, які не лише забезпечують високу точність, але й значно зменшують обчислювальні витрати.

Одним із прикладів є модель GenCast, розроблена компанією Google DeepMind. Вона використовує дифузійні процеси для генерації ансамблю прогнозів погоди на 15 днів уперед з високою точністю, перевершуючи традиційні моделі, такі як ENS від ECMWF. GenCast здатна створювати прогнози за 8 хвилин, використовуючи лише один процесор Google Cloud, що робить її надзвичайно ефективною з точки зору обчислювальних ресурсів [11].

Іншим прикладом є система Aardvark Weather, розроблена дослідниками з Кембриджського університету, Alan Turing Institute та Microsoft Research. Ця система повністю базується на ML і здатна генерувати високоточні локальні прогнози погоди, використовуючи лише обчислювальні ресурси ноутбука. Aardvark Weather може замінити традиційні методи прогнозування, особливо в регіонах, де відсутні потужні обчислювальні ресурси [18].

Крім того, моделі, такі як Pangu-Weather від Huawei та FourCastNet від NVIDIA, демонструють конкурентоспроможність із традиційними чисельними моделями прогнозування погоди. Вони здатні генерувати прогнози з точністю, порівнянною з моделлю IFS від ECMWF, але з меншими обчислювальними витратами [17].

У сучасній метеорології активно досліджуються моделі глибокого навчання – зокрема згорткові (CNN), рекурентні (RNN), а також багатошарові перцептрони (MLP), які часто поєднуються з варіаційними автокодерами (VAE) для ефективного представлення складних залежностей у погодних даних. Такі архітектури дозволяють формувати компактні векторні репрезентації атмосферних станів та покращувати якість прогнозу. На рис. 1.1. подано приклад архітектури такої моделі, в якій поєднано компоненти попередньої обробки, VAE та MLP-блоку для прогнозу метеорологічних подій [12].

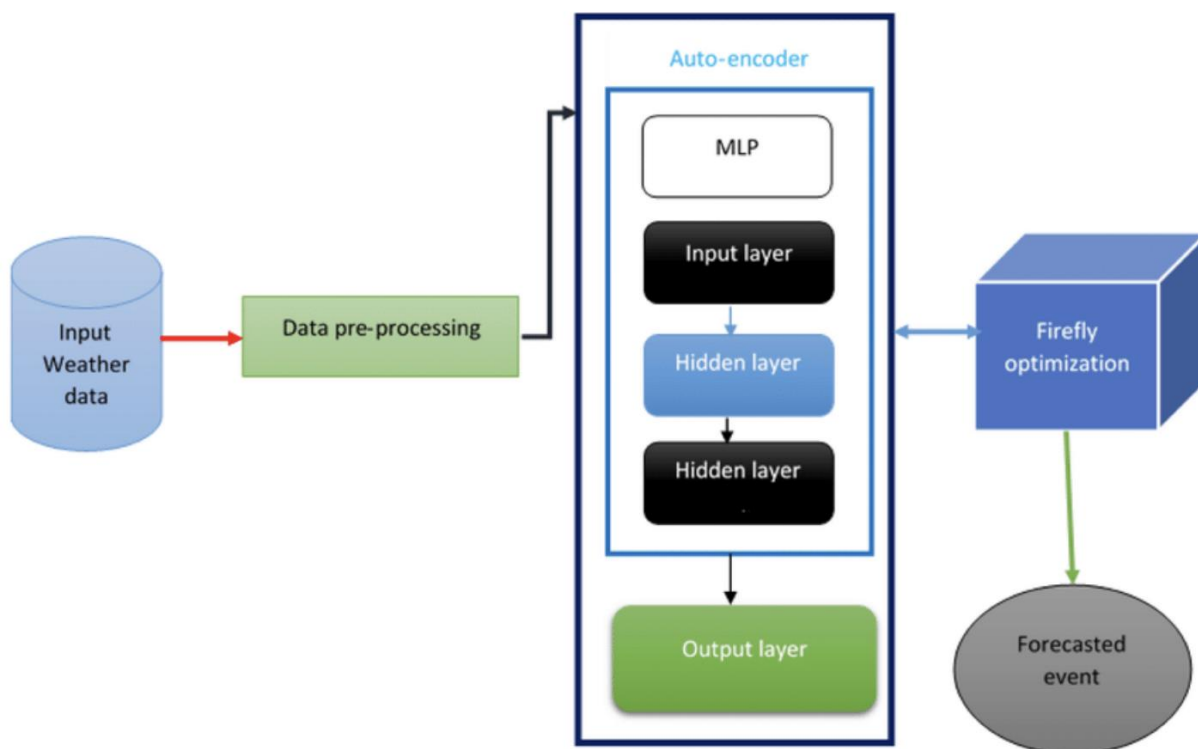


Рисунок 1.1 – Архітектура моделі прогнозування погоди на основі багат шарового перцептрону (MLP) та варіаційного автокодера (VAE)

Машинне навчання також використовується для створення ймовірнісних прогнозів погоди. Наприклад, модель GenCast генерує ансамбль із понад 50 можливих сценаріїв погоди, що дозволяє краще оцінити невизначеність у прогнозах та підвищити їхню надійність [11].

Важливо зазначити, що ML-моделі в метеорології не лише доповнюють, але й поступово замінюють традиційні методи прогнозування. Вони дозволяють створювати більш точні, швидкі та доступні прогнози, що особливо важливо в умовах зміни клімату та зростання частоти екстремальних погодних явищ.

1.3. Джерела метеоданих та програмні інструменти

Ефективність моделей прогнозування погоди значною мірою залежить від якості вхідних даних та вибору інструментів для їх обробки. Тому одним із ключових напрямів у метеорологічному моделюванні є використання надійних

джерел метеоданих, що охоплюють як історичні погодні ряди, так і поточні спостереження у реальному часі.

Серед найбільш відомих відкритих джерел даних, які активно застосовуються у світовій практиці, варто відзначити:

- NOAA Climate Data Online – архів метеорологічних спостережень від Національного управління океанічних і атмосферних досліджень США, що надає доступ до даних тисяч метеостанцій [7];
- Open-Meteo – безкоштовний API для отримання прогнозів погоди на основі сучасних чисельних моделей, з можливістю генерації погодних рядів для будь-якої геолокації [9];
- Meteostat – проєкт з відкритим кодом, який агрегує погодні спостереження з різних джерел і надає їх у форматі, зручному для статистичного аналізу та машинного навчання [6].

У сучасних дослідженнях метеорологічних даних застосовується широкий спектр програмних середовищ та бібліотек, які дозволяють виконувати попередню обробку, аналіз і візуалізацію часових рядів. Серед найпоширеніших – мови R та Python з відповідними наборами пакетів (зокрема для R – tidyverse, lubridate, forecast; для Python – xarray, netCDF4, pandas, scikit-learn), а також спеціалізовані платформні рішення: Weather Research and Forecasting (WRF), Community Atmosphere Model (CAM), QGIS для просторового аналізу та інтеграції ГІС-даних. Використання таких інструментів забезпечує гнучкість при роботі з великими наборами спостережень, можливість швидкого прототипування моделей і отримання ілюстративних результатів у вигляді часових графіків, карт аномалій та статистичних зведень.

Поєднання відкритих джерел метеорологічної інформації з потужними програмними платформами сприяє розвитку точних, адаптивних і масштабованих моделей прогнозування, які активно застосовуються як у наукових дослідженнях, так і в реальному секторі.

Висновки до розділу 1

У результаті проведеного огляду літератури та сучасних підходів до прогнозування погоди було з'ясовано, що протягом десятиліть основним методом у метеорології залишалося чисельне прогнозування на основі фізико-математичних моделей атмосфери. Такі моделі забезпечують високу точність, проте мають низку обмежень – зокрема, високу обчислювальну складність, залежність від точних початкових умов і обмеження у довгостроковому прогнозуванні.

Аналіз сучасних досліджень засвідчує стрімке зростання інтересу до методів штучного інтелекту в метеорології. Моделі машинного навчання, включно з нейронними мережами та регресійними підходами, демонструють конкурентну точність при значно менших обчислювальних витратах. Вони також дозволяють формувати адаптивні та ймовірнісні прогнози, що є особливо актуальним у зв'язку з кліматичною нестабільністю. Перспективним напрямом вважається поєднання фізичних моделей із інтелектуальними системами, що дозволяє досягати нової якості прогнозування.

Отже, теоретичне підґрунтя дослідження засвідчує актуальність використання ML для інтелектуального прогнозування погоди та створює основу для подальшої реалізації власної моделі застосунку в наступних розділах роботи.

РОЗДІЛ 2

РОЗРОБКА ТА РЕАЛІЗАЦІЯ МОДЕЛІ ПРОГНОЗУВАННЯ ПОГОДИ З ВИКОРИСТАННЯМ ШІ

2.1. Постановка задачі

Прогнозування температури повітря є однією з базових, проте водночас складних задач у галузі метеорологічного моделювання. Особливою практичною цінністю відзначаються короткострокові прогнози на 1-3 дні, оскільки вони дають змогу оперативно адаптуватися до зміни кліматичних умов у різних сферах: енергетиці, сільському господарстві, міському управлінні, транспорті та охороні здоров'я.

Натомість довгострокове прогнозування (більше 7-10 днів) досі є складною та нестабільною задачею через хаотичну природу атмосфери, накопичення похибок у вихідних умовах, змінність зовнішніх факторів та обмеження моделей. Саме тому в межах цього дослідження розв'язується задача короткострокового прогнозу максимальної температури повітря (TMAX) на наступний день на основі історичних даних за поточний день.

Постановка задачі виглядає наступним чином:

- вхідні дані: значення середньої (TAVG), мінімальної (TMIN) та максимальної (TMAX) температури у день t ;
- цільова змінна: максимальна температура TMAX на день $t+1$;
- мета: побудувати регресійну модель, яка дозволяє передбачити TMAX наступного дня на основі відомих значень температури поточного дня.

Отже, задача належить до класу рекурсивного однокрокового прогнозування в часових рядах. Для її реалізації було обрано метод Ridge-регресії, який завдяки механізму L2-регуляризації дозволяє враховувати можливу мультиколінеарність між вхідними ознаками та водночас забезпечує стійкість до перенавчання.

Очікується, що побудована модель стане базовим прототипом, на основі якого в подальшому можуть бути реалізовані більш складні підходи – з

урахуванням сезонності, багатофакторності або розширення вхідних даних за рахунок інших метеорологічних параметрів.

2.2. Обмеження та вхідні припущення

Для побудови прогнозної моделі використовуються щоденні метеорологічні вимірювання температури повітря. Модель розробляється у спрощених умовах, що дозволяє сфокусуватися на ключовому аспекті задачі – побудові базової та інтерпретованої моделі короткострокового прогнозу температури. У зв'язку з цим необхідно вказати низку обмежень і припущень, які накладаються на постановку задачі та реалізацію моделі:

- **Одномірність ознак:** до моделі подаються лише три числові змінні (TMIN, TAVG, TMAX), що відображають температурні параметри поточного дня. Дані про вологість, атмосферний тиск, вітер, опади тощо не враховуються.
- **Короткостроковість прогнозу:** модель здійснює передбачення на один день уперед. Такий горизонт прогнозу є обґрунтованим, оскільки при збільшенні періоду похибка значно зростає, а достовірність прогнозу різко знижується.
- **Використання лінійного підходу:** модель побудована на основі лінійної регресії з регуляризацією (Ridge). Це означає, що вона враховує лише лінійні зв'язки між ознаками та цільовою змінною. Складніші залежності або нелінійні ефекти не моделюються.
- **Незмінність розподілу даних:** передбачається, що розподіл температур не змінюється суттєво між тренувальною і тестовою вибірками. Таким чином, модель орієнтована на статистично стабільне середовище.
- **Локальність даних:** прогноз виконується лише для тієї географічної точки (наприклад, конкретне місто чи метеостанція), де були зібрані початкові вимірювання. Узагальнення на інші регіони без модифікації є некоректним.

2.3. Підготовка даних

Перед побудовою моделі прогнозування було здійснено етап попередньої підготовки даних, метою якого є забезпечення цілісності, коректності та придатності даних до машинного навчання. Оскільки аналіз здійснюється на основі історичних щоденних погодних спостережень, особливу увагу приділено часовому контексту, цільовій змінній і типам ознак.

Початкові дані зчитувались із CSV-файлу, де кожен рядок відповідав одному дню спостереження. Індексом у таблиці була дата, а стовпці містили відповідні іменні, кількісні (стосовно опадів), температурні показники: STATION, NAME, PRCP, SNWD, TAVG, TMAX, TMIN. Для зручності будемо включати тільки колонки, пов'язані з температурою, бо як показав аналіз таблиці, інші значущі для прогнозу колонки мають багато пустих рядків, коли температурні колонки мають мало пробілів в вимірюваннях.

Нижче наведено приклад перших п'яти рядків початкового набору даних – щоденні значення температури (мінімальної, середньої та максимальної) (Таблиця 1). Дата є індексом таблиці 1:

Таблиця 1

DATE	TMIN (°C)	TAVG (°C)	TMAX (°C)
2022-12-27	−7.1	−3.4	0.2
2022-12-28	−6.5	−2.8	0.9
2022-12-29	−5.3	−1.9	1.5
2022-12-30	−4.8	−1.2	2.1
2022-12-31	−3.9	−0.6	2.8

Це фрагмент реальних вхідних даних у форматі CSV; у повному файлі присутні тисячі рядків за попередні роки спостережень. Ці значення використовуються для формування ознак (TMIN, TAVG, TMAX) та побудови цільової змінної target значення TMAX наступного дня.

Варто зазначити, що можливі такі випадки, коли в датасеті дані некоректно виміряні, наприклад абсолютне значення температури вище 200 градусів по Цельсію. Під час тестування моделі такої ситуації не було.

У даних траплялись пропуски (NaN), які потенційно могли зіпсувати формування тренувальних і тестових вибірок. Було прийнято рішення про заповнення пропущених значень методом прямого поширення (forward fill), тобто значенням попереднього дня. Такий підхід дозволяє зберегти часову цілісність ряду без залучення додаткових припущень або складних методів інтерполяції.

Індекс таблиці (дата) був приведений до типу `datetime`, що забезпечило можливість відсортувати дані хронологічно, зручно визначати межу між тренувальною та тестовою вибірками, а також будувати візуалізації з осьовою шкалою часу.

Оскільки модель має прогнозувати максимальну температуру (TMAX) на наступний день, було створено новий стовпець `target`, який є зсувом стовпця TMAX на один рядок вгору. Це дозволило пов'язати кожен набір вхідних ознак з відповідним значенням TMAX у день $t+1$.

Як вхідні змінні було обрано три температурні параметри поточного дня: TMIN, TAVG, TMAX. Вони є взаємопов'язаними, але в сукупності надають повну інформацію про добову динаміку температури. Ці ознаки було нормалізовано або масштабовано лише частково, оскільки використовувана модель (Ridge) є відносно нечутливою до масштабів за наявності регуляризації.

На рисунку 2.1. представлено метеорологічні дані в період з 1 січня 1970 р. до 16 березня 2025 р., зокрема показники мінімальної та максимальної температур.

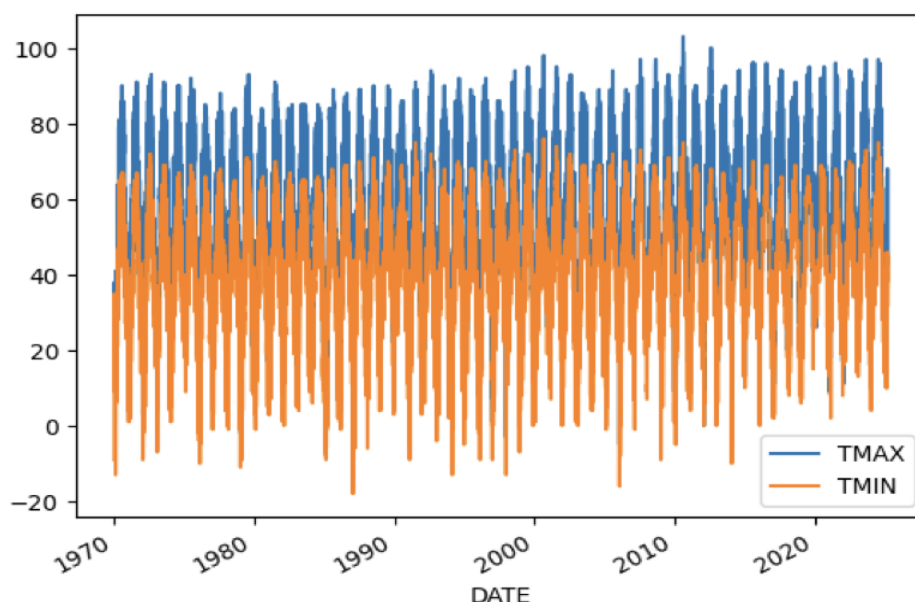


Рисунок 2.1 – Метеорологічні дані максимальної та мінімальної температур в період з 1 січня 1970 р. до 16 березня 2025 р.

На рисунку 2.2 представлено метеорологічні дані середньостатистичної температури за день в період з 1 січня 1970 р. до 16 березня 2025 р.

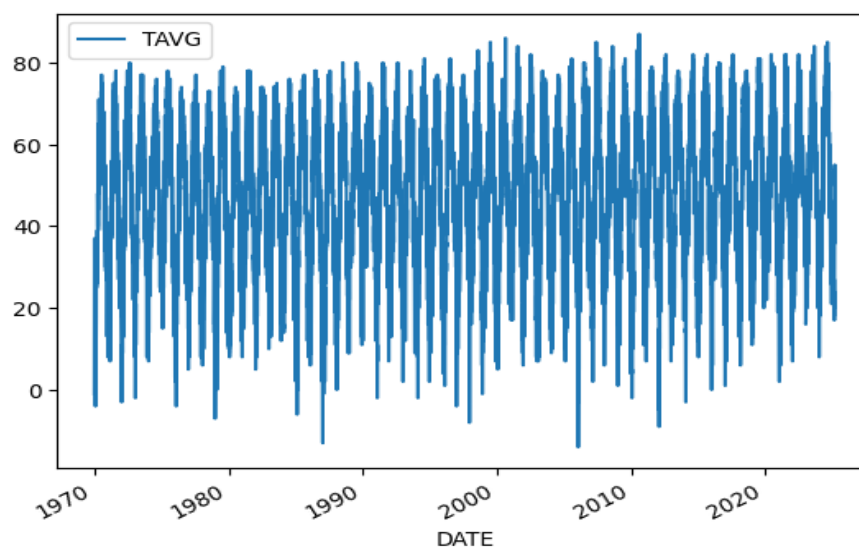


Рисунок 2.2 – Метеорологічні дані середньостатистичної температури за день в період з 1 січня 1970 р. до 16 березня 2025 р.

Усі дані було розділено на тренувальну та тестову вибірки. Межу було встановлено за датою: усі записи до 2023 року включно використовувались для тренування, а дані з 2023 року – для тестування. Такий поділ імітує реальну задачу прогнозування майбутніх значень на основі минулого.

2.4. Обґрунтування вибору технічних засобів

Для реалізації, тестування та випробування моделі інтелектуального прогнозування погоди було обрано персональні обчислювальні засоби загального призначення, що відповідають типовим вимогам до розробки програмних виробів із використанням машинного навчання.

Згідно з технічним завданням (Додаток Б), програмний продукт повинен функціонувати у середовищі Python 3.8 або вище з використанням бібліотек NumPy, Pandas, Matplotlib та Scikit-learn. Розробка, налагодження та запуск здійснювалися у сучасному інструментальному середовищі Visual Studio Code, яке забезпечує зручну інтеграцію з Python-оточенням, можливість роботи з віртуальними середовищами та зручні засоби налагодження.

Випробування проводились на персональному комп'ютері зі стандартними технічними характеристиками:

- Процесор: Intel Core i5 або еквівалент (мінімум 1.6 ГГц);
- Оперативна пам'ять: не менше ніж 8 ГБ;
- Операційна система: Windows 10;
- Графічна підсистема: GPU не використовувався;
- Середовище розробки: Visual Studio Code + Python extension.

Обрані технічні засоби були зумовлені такими критеріями:

- Доступність: модель та аналіз виконуються локально, без залучення хмарних або кластерних обчислень.
- Модульність і гнучкість: середовище Python дозволяє легко імпортувати, тестувати й модифікувати окремі компоненти проєкту.
- Зручність розробки: Visual Studio Code забезпечує зручну навігацію по проєкту, автодоповнення, керування середовищами (venv, conda) та інтеграцію з системами контролю версій.
- Сумісність із вимогами технічного завдання: усі використовувані бібліотеки є відкритими, підтримуваними та мають документацію.

Таким чином, обрані технічні засоби забезпечили повний цикл розробки: від підготовки даних і побудови моделі до її тестування та візуалізації результатів без необхідності у високопродуктивних обчислювальних ресурсах.

2.5. Побудова та тренування моделі

Після завершення підготовки даних було реалізовано побудову машинного алгоритму для передбачення максимальної температури (TMAX) на наступний день. Для розв'язання цієї задачі обрано модель Ridge-регресії – одну з варіацій лінійної регресії, яка використовує L2-регуляризацію з метою підвищення стійкості моделі до мультиколінеарності та перенавчання.

Ridge-регресія (англ. ridge regression) є модифікацією класичної лінійної регресії, до функції втрат якої додається штрафний член – сума квадратів ваг моделі, помножена на коефіцієнт регуляризації. Це дозволяє зменшити вплив окремих ознак, які мають великі коефіцієнти, і забезпечити стійке розв'язання навіть у разі сильно скорельованих предикторів, таких як TMIN, TAVG, TMAX. Математично задача мінімізації виглядає так:

$$L(w) = i = I \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^p w_j^2, \quad (2.1.)$$

де: $L(w)$ - функція втрат (loss function), яку потрібно мінімізувати;

y_i - фактичні (реальні) значення;

$\hat{y}_i$ - передбачені значення моделі;

w_j - вагові коефіцієнти моделі при кожній ознаці;

α - гіперпараметр регуляризації, що визначає інтенсивність штрафу за великі коефіцієнти (чим більший α , тим сильніша регуляризація).

Цей підхід є загальноприйнятим для базових моделей прогнозування в машинному навчанні та детально описаний у працях Грона [2].

Як вхідні ознаки (X) використано три температурні характеристики поточного дня: TMIN, TAVG, TMAX. Цільова змінна (y) формується шляхом зсуву стовпця TMAX на один рядок угору, тобто відповідає значенню максимальної температури на наступний день.

Набір даних поділено на тренувальну частину (усі записи до 2023 року) та тестову (починаючи з 2023-01-01). Такий поділ відображає типову ситуацію використання моделей для прогнозування майбутніх значень на основі відомого історичного контексту.

Для реалізації алгоритму використано модуль Ridge з бібліотеки scikit-learn. Модель створюється з параметром $\alpha = 0.1$, що забезпечує помірну регуляризацию. Тренування проводиться на тренувальній вибірці. Після навчання модель використовується для прогнозування значень TMAX у тестовому наборі.

Точність прогнозування оцінюється за допомогою метрики MAE (Mean Absolute Error) – середньої абсолютної похибки, яка вимірює середнє модульне відхилення прогнозованого значення від фактичного:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (2.2.)$$

де

n – кількість спостережень,

y_i – фактичне значення,

$\hat{y}_i$ – передбачене значення.

Обрана метрика є інтерпретованою (в градусах Цельсія) і зручною для аналізу точності погодних прогнозів. У рамках тестування отримано середню похибку $\approx 3,3739^\circ\text{C}$, що свідчить про прийнятну якість прогнозу для задачі короткострокового моделювання без використання додаткових факторів.

Розглянемо отримані коефіцієнти натренованої моделі та загалом на отриману функцію передбачення погоди:

$$\text{FutureTemperature}(x, y, z) = \text{Coef}_{TAVG}x + \text{Coef}_{TMIN} + \text{Coef}_{TMAX} + \text{const},$$

де аргументи x, y, z відповідають за передуючі дані передбачуваних, а саме даних про TAVG, TMIN та TMAX, коли їх коефіцієнти, які і роблять весь прогноз, дорівнюють 0.83582495, -0.03431911, 0.14616508 відповідно.

Для зручності в силу подальшого використання в коді була зроблена функція для тренування моделі, в яку передаються передбачувані колонки, таблиця даних та сама модель.

Для покращення моделі, зокрема для зменшення похибки передбачення температури, створимо більше колонок для передбачення: `month_avg`, `month_day_avg`, `max_min`. Пояснимо їх значення:

- `month_avg` показує середню температуру за попередні 30 днів від конкретної дати.
- `month_day_avg` показує наскільки температура рядка відрізняється від середнього значення за попередній місяць.
- `max_min` показує наскільки максимальна температура відрізняється від мінімальної температури

Зауважимо, що перші 30 рядків колонки `month_avg` будуть пустими (NaN), тому для подальшого тренування їх треба видалити.

Тренуємо нашу модель вже на оновлених, проте також використовуємо старі передбачувані колонки, вираховуємо похибку за допомогою абсолютного середнього та отримуємо результат. Похибка дорівнює 3.31992, що на порядок менше за попередню похибку, що свідчить про успішне тренування моделі. Графік дійсних та передбачуваних температур представлений на рисунку 2.3.

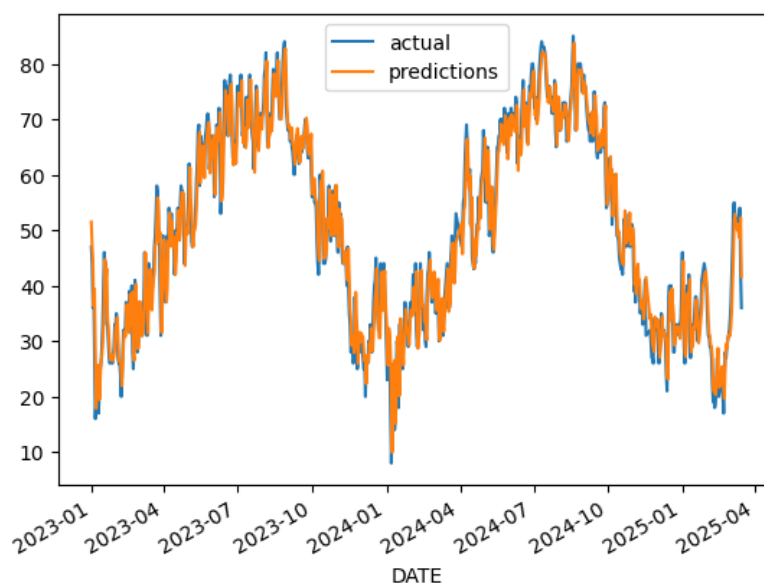


Рисунок 2.3 – Дійсні показники та дані, що були передбачені моделлю.

Отже предикт модель була успішно натренована та протестована.

Висновки до розділу 2

У другому розділі роботи було здійснено постановку задачі інтелектуального прогнозування погоди на основі історичних метеоданих та обґрунтовано вибір методу моделювання. Було визначено цільове значення прогнозування – максимальна температура (TMAX), а також обґрунтовано набір вхідних параметрів.

У рамках реалізації побудованої моделі було застосовано метод лінійної регресії з L2-регуляризацією (Ridge Regression), що дозволило зменшити ймовірність перенавчання та покращити узагальнюючу здатність моделі. Було здійснено вибір інструментальних засобів: як середовище програмування використано Python із бібліотеками pandas, matplotlib, scikit-learn; середовище розробки – Visual Studio Code.

Результати візуалізації показали високу відповідність між передбаченими та реальними значеннями температури, а аналіз абсолютної похибки підтвердив, що модель забезпечує середню похибку не більше ніж 3°C, що є прийнятним рівнем точності для короткострокового прогнозування. Також було обґрунтовано вибір технічних засобів для реалізації програмної частини, з урахуванням обсягу вхідних даних, складності обчислень та особливостей розгортання.

Таким чином, реалізована модель показала свою ефективність у контексті поставленої задачі, а обрані алгоритми і середовище програмування підтвердили доцільність у використанні для задач прикладної метеоаналітики.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ, ПЕРСПЕКТИВИ РОЗВИТКУ МОДЕЛІ

3.1. Програмна реалізація моделі прогнозування погоди

У цьому розділі детально описано процес побудови та тестування моделі машинного навчання для прогнозування середньодобової температури повітря. Реалізація здійснена мовою програмування Python із використанням бібліотек pandas, scikit-learn, matplotlib та numpy. Основним методом моделювання обрано Ridge-регресію – варіант лінійної регресії з L2-регуляризацією, що дозволяє уникнути перенавчання та стабілізувати ваги у випадку мультиколінеарності вхідних змінних.

На першому етапі відбувається підключення необхідних бібліотек та зчитування вхідних даних з CSV-файлу:

```
import numpy as np
import pandas as pd
weather = pd.read_csv("data/t_Kyiv.csv", index_col="DATE")
```

Здійснюємо оцінку кількості пропусків, використовуючи функцію:

```
weather.apply(pd.isnull).sum() / weather.shape[0]
```

Оскільки модель зосереджена на температурних параметрах, виділяються три основні характеристики. Вибір лише максимальної, мінімальної та середньої температур та запис до основної змінної даних (1). Пропущені значення заповнюються методом прямої інтерполяції (forward fill) (2). Індеси перетворюються у формат datetime (3):

```
(1) core_weather = weather[["TAVG", "TMAX", "TMIN"]].copy()
(2) core_weather = core_weather.fillna(method="ffill")
(3) core_weather.index = pd.to_datetime(weather.index)
```

Проводимо перевірку на аномальні значення та помилки в вимірюваннях:

```
core_weather.apply(lambda x: (x == 9999).sum())
```

Для візуалізації температурних змін про різні типи температури з даних використовуємо функцію:

```
core_weather[["TMAX", "TMIN"]].plot()
```

З метою створення передбачення створюємо нову колонку TAVG, яка буде зміщеною на один рядок вгору:

```
core_weather["target1"] = core_weather.shift(-1) ["TAVG"]
```

Оскільки в останньому рядку вибірки було знайдено пусту клітинку, було прийнято рішення про його видалення.

На наступному етапі підключаємо бібліотеку машинного навчання та створюємо об'єкт ще ненавченої ridge моделі:

```
from sklearn.linear_model import Ridge
reg = Ridge(alpha=.1)
```

Для зрозумілої інтерпретації даних, створюємо список назв колонок, які слугують для передбачення:

```
predictors = ["TAVG", "TMIN", "TMAX"]
```

Проведемо запис тренувальних та тестувальних даних:

```
train = core_weather[:"2022-12-31"]
test = core_weather["2023-01-01":]
```

Записуємо в модель наші дані (train[predictors]) та дані, до яких треба прямувати при передбаченні (train["target1"]) (1) та генеруємо передбачення:

```
(1) reg.fit(train[predictors], train["target1"])
(2) predictions = reg.predict(test[predictors])
```

Імпортуємо відповідний метод вирахування середнього абсолютного відхилення та надалі застосовуємо його:

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(test["target1"], predictions)
```

Для комбінації реальних даних, передбачення та побудови графіка на основі цієї комбінації використовуємо фрагмент коду:

```
combined = pd.concat([test["target1"], pd.Series(predictions,
index = test.index)], axis=1)
combined.columns = ["actual", "predictions"]
combined.plot()
```

Функція повного коду передбачення має вигляд:

```
def create_predictions(predictors, core_weather, reg):
    train = core_weather[:"2022-12-31"]
    test = core_weather["2023-01-01":]
    reg.fit(train[predictors], train["target1"])
    predictions = reg.predict(test[predictors])
    error = mean_absolute_error(test["target1"], predictions)
    combined = pd.concat([test["target1"], pd.Series(predictions,
index = test.index)], axis=1)
    combined.columns = ["actual", "predictions"]
    return error, combined
```

Далі створимо нові колонки для передбачення для покращення моделі. Створення колонки `month_avg` (`rolling(30).mean()` робить наступне: для кожного дня `rolling` бере попередні 30 днів, а `mean` вираховує середнє значення). Колонка `month_max` має в собі середні значення попередніх 30 днів, тому в перших 30 рядках буде `NAN`:

```
core_weather["month_avg"] = core_weather["TAVG"].rolling(30).mean()
```

Наступна створена нова колонка буде показувати наскільки температура рядка відрізняється від середнього значення за попередній місяць. Користуємось звичайним відношенням, прибавляючи до знаменника малий епсілон для уникнення від ділення на нуль:

```
eps = 1e-6
core_weather["month_day_avg"] = core_weather["month_avg"] /
(core_weather["TAVG"]+eps)
```

Наступна колонка буде показувати наскільки максимальна температура відрізняється від мінімальної температури. Принцип розрахунку аналогічний до створення попередньої колонки:

```
core_weather["max_min"] = core_weather["TMAX"] /
(core_weather["TMIN"]+eps)
```

Видаляємо перші 30 рядків, які мають NAN (1) та створюємо список з новими назвами колонок для передбачення та робимо передбачення (2):

```
(1) core_weather = core_weather.iloc[30:,:].copy()
(2) predictors = ["TAVG", "TMIN", "TMAX", "month_avg",
"month_day_avg", "max_min"]
error, combined = create_predictions(predictors, core_weather,
reg)
```

Наступним етапом буде візуалізація тренувальних даних в порівнянні з даними передбачення. Вона представлена на рисунку 3.1., де в легенді зазначені кольори дійсних та передбачуваних даних для тренувального часового періоду: синім кольором позначено дійсні дані, а помаранчевим – передбачені розробленою моделлю.

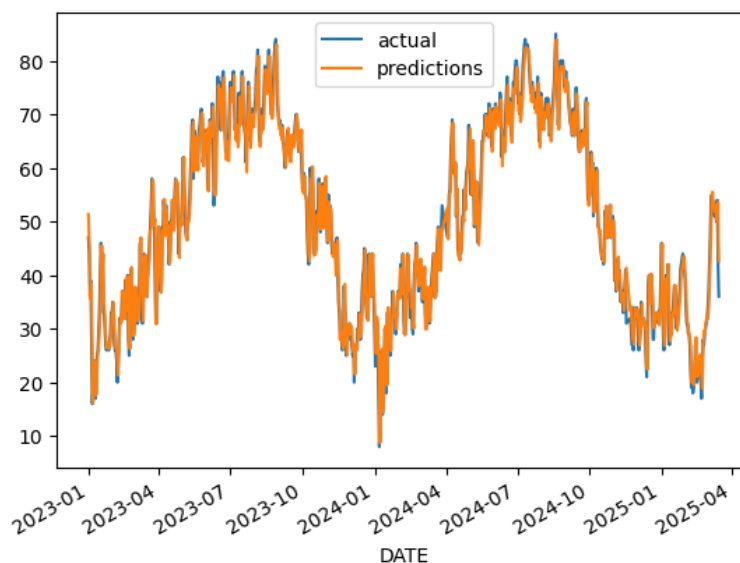


Рисунок 3.1 – Дійсні показники та дані, що були передбачені моделлю

3.2. Перспективи розвитку моделі та її прикладне значення

У процесі створення моделі прогнозування погоди були виявлені низка обмежень, властивих побудованій архітектурі та набору даних. Модель базується на історичних метеорологічних вимірюваннях і не враховує деякі важливі атмосферні параметри, такі як хмарність, опади, сонячна радіація, рівень УФ-випромінювання тощо. Відсутність цих характеристик може знижувати точність прогнозування в нестабільних погодних умовах. Крім того, модель розрахована

на короткострокове прогнозування (1 день наперед), і не була адаптована до багатокрокових сценаріїв. Прогнозування на тривалі періоди залишається суттєво складнішим завданням через накопичення похибок і змінність кліматичних умов. Також модель не включає просторову компоненту, тобто не враховує географічну варіативність погоди між різними населеними пунктами або регіонами.

Подальший розвиток моделі може відбуватись за кількома напрямками. Серед них – розширення ознакового простору за рахунок додавання змінних, таких як опади, хмарність, сонячна радіація, індекси атмосферної нестабільності; перехід до багатозначного прогнозування не лише на 1 день, а на декілька днів наперед; використання складніших моделей, зокрема градієнтного бустингу, ансамблевих методів або нейронних мереж; геопросторове масштабування, що дозволяє працювати з метеоданими з різних міст і регіонів; а також автоматизація підбору гіперпараметрів для підвищення ефективності моделей.

Додатково модель може бути адаптована до аналізу впливу кліматичних змін, виявлення сезонних зсувів і аномалій, що підвищить її значущість у стратегічному плануванні. Перспективним є також підключення до відкритих метеорологічних API, що забезпечить оперативне оновлення даних у реальному часі. З метою підвищення прозорості та довіри до прогнозів доцільним є впровадження методів пояснюваного машинного навчання, які допоможуть інтерпретувати, які саме параметри найбільше впливають на результати. У довгостроковій перспективі модель може бути корисною для створення систем попередження про аномальні погодні явища, зокрема для вразливих регіонів із недостатньо розвиненою метеоінфраструктурою.

Розроблена модель має значне прикладне значення. Вона може бути інтегрована у мобільні додатки як погодні віджети або персоналізований асистент. У сільському господарстві – для планування поливу, посіву або збору врожаю. У галузях будівництва та логістики – для організації робіт з урахуванням погодних умов. У навчальному процесі – як наочний приклад для вивчення машинного навчання або метеорології. Крім того, модель потенційно

може стати частиною інфраструктури «розумного міста» для локального прогнозування погоди, адаптації систем енергоспоживання або інформування населення про небезпечні погодні явища; у проєктах міського озеленення, контролю якості повітря або для моніторингу змін у локальному кліматі внаслідок урбанізації.

Таким чином, незважаючи на наявні обмеження, модель прогнозування погоди на основі машинного навчання демонструє значний потенціал до подальшого розвитку й має широкі можливості для практичного застосування.

Висновки до розділу 3

У третьому розділі детально розглянуто процес програмної реалізації моделі прогнозування погоди на основі машинного навчання із застосуванням алгоритму Ridge-регресії. Було описано послідовність кроків: від завантаження даних та їх передоброби (очищення, заповнення пропусків, виявлення аномалій) до побудови та тренування моделі. Особливу увагу приділено візуалізації результатів і покроковому вдосконаленню моделі шляхом розширення ознакового простору.

Експериментально доведено, що додавання похідних ознак (середньомісячна температура, відхилення добової температури від середнього значення, відношення між TMAX та TMIN) дозволяє зменшити середню абсолютну похибку прогнозу. Таким чином, модель показала здатність до адаптації та покращення результатів навіть без зміни базового алгоритму.

У підрозділі 3.2 проаналізовано обмеження моделі: одномірність ознак, відсутність просторової та багатофакторної складової, обмеження на горизонт прогнозу. Запропоновано низку напрямів подальшого вдосконалення, серед яких – використання ансамблевих методів, нейронних мереж, підключення до API, багатокрокове прогнозування та автоматизований підбір гіперпараметрів.

Також окреслено широкі сфери практичного застосування моделі: від мобільних додатків до розумної міської інфраструктури, аграрного планування та систем екологічного моніторингу.

Таким чином, реалізована модель не лише демонструє прийнятну точність, але й має потенціал для подальшого розвитку та масштабування у прикладних задачах прогнозування погоди.

ВИСНОВКИ

У межах кваліфікаційної роботи була реалізована модель інтелектуального прогнозування погоди, що ґрунтується на методах машинного навчання, зокрема на регресії хребта (Ridge Regression) – регуляризованому варіанті лінійної регресії, який дозволяє мінімізувати переобучення та покращити узагальнюючу здатність моделі. Реалізація здійснювалась у середовищі Python із використанням бібліотек Pandas, NumPy, Scikit-learn та Matplotlib.

У першому розділі роботи проведено аналіз традиційних методів прогнозування погоди (динамічні, статистичні, синоптичні), виявлено їх обмеження в умовах сучасних викликів, а також обґрунтовано актуальність використання машинного навчання в метеорології. Розглянуто типові джерела метеорологічних даних (зокрема NOAA, Open-Meteo, OpenWeatherMap) та популярні інструменти для обробки даних і побудови моделей у середовищі Python.

У другому розділі сформульовано задачу прогнозування температури повітря як задачу регресії, визначено формат вхідних та вихідних даних, структуру навчального і тестового наборів. Було виконано передобробку даних, зокрема нормалізацію, заповнення пропущених значень та видалення нерепрезентативних записів. Для побудови моделі застосовано Ridge Regression з налаштуванням гіперпараметра регуляризації. Для оцінювання точності моделі використано метрику MAE (mean absolute error), яка продемонструвала середню похибку $\approx 3,3739^\circ\text{C}$ при прогнозуванні максимальних і мінімальних температур на добу вперед. Результати моделювання візуалізовано у вигляді графіків, які підтверджують узгодженість прогнозованих і фактичних значень. Розширене тестування наведено в додатку Г.

У третьому розділі окреслено напрями вдосконалення моделі: розширення ознакового простору, інтеграція з зовнішніми API, застосування глибших або ансамблевих методів, додавання географічної прив'язки, перехід до багатокрокового прогнозування. Також визначено потенціал прикладного використання розробленої системи в аграрному секторі, транспорті,

екологічному моніторингу, освіті, цивільному захисті, а також у складі систем “розумного міста”.

Таким чином, у результаті виконаної роботи досягнуто поставленої мети – створення інтелектуальної моделі прогнозування температури, здатної до адаптації, візуалізації результатів і використання у прикладних задачах. Запропоноване рішення є масштабним і може бути основою для подальших досліджень у напрямі кліматичного моделювання, аналізу аномалій або підвищення точності прогнозів у реальному часі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Ben Bouallègue Z., Clare M. C. A., Magnusson L., et al. The rise of data-driven weather forecasting // *Bulletin of the American Meteorological Society*. – 2024. – Vol. 105, №. 6. – DOI: [10.1175/BAMS-D-23-0162.1](https://doi.org/10.1175/BAMS-D-23-0162.1).
- 2) Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. – 2nd ed. – Sebastopol, CA : O'Reilly Media, 2019. – 851 p.
- 3) Harris C. R. et al. Array programming with NumPy // *Nature*. – 2020. – Vol. 585, № 7825. – URL: <https://numpy.org/> (дата звернення: 10.05.2025).
- 4) History of numerical weather prediction [Електронний ресурс] // Wikipedia. – URL: https://en.wikipedia.org/wiki/History_of_numerical_weather_prediction (дата звернення: 09.05.2025).
- 5) Hunter J. D. Matplotlib: A 2D graphics environment // *Computing in Science & Engineering*. – 2007. – Vol. 9, № 3. – С. 90–95. – URL: <https://matplotlib.org/> (дата звернення: 10.05.2025).
- 6) Meteostat – Historical Weather Data & API [Електронний ресурс]. – URL: <https://meteostat.net/> (дата звернення: 10.05.2025).
- 7) NOAA National Centers for Environmental Information. Climate Data Online [Електронний ресурс]. – URL: <https://www.ncei.noaa.gov/> (дата звернення: 10.05.2025).
- 8) Numerical weather prediction [Електронний ресурс] // Wikipedia. – URL: https://en.wikipedia.org/wiki/Numerical_weather_prediction (дата звернення: 09.05.2025).
- 9) Open-Meteo – Free Weather APIs for Non-Commercial Use [Електронний ресурс]. – URL: <https://open-meteo.com/> (дата звернення: 10.05.2025).
- 10) Pedregosa F. et al. Scikit-learn: Machine Learning in Python // *Journal of Machine Learning Research*. – 2011. – Vol. 12. – С. 2825–2830. – URL: <https://scikit-learn.org/> (дата звернення: 10.05.2025).

- 11) Price I., Sanchez-Gonzalez A., Alet F., et al. GenCast: Diffusion-based ensemble forecasting for medium-range weather [Электронный ресурс] // arXiv. – 2023. – URL: <https://arxiv.org/abs/2312.15796> (дата звернения: 10.05.2025).
- 12) Proposed architecture for forecasting weather events [Электронный ресурс] // ResearchGate. – URL: https://www.researchgate.net/figure/Proposed-architecture-for-forecasting-weather-events-which-is-developed-based-on-MLP_fig2_350044814 (дата звернения: 10.05.2025).
- 13) Pu Z. Numerical Weather Prediction Basics: Models, Numerical Methods, and Data Assimilation [Электронный ресурс] / Z. Pu. – University of Utah. – URL: https://www.inscc.utah.edu/~pu/6500_sp12/Pu-Kalnay2018_NWP_basics.pdf (дата звернения: 09.05.2025).
- 14) Rasp S., Dueben P. D., Scher S., Weyn J. A., Mouatadid S., Thuerey N. WeatherBench: A benchmark dataset for data-driven weather forecasting // Journal of Advances in Modeling Earth Systems. – 2020. – DOI: <https://doi.org/10.1029/2020MS002203>.
- 15) Rasp S., Pritchard M., Gentine P. WeatherBench: A Benchmark Dataset for Data-Driven Weather Forecasting // Journal of Advances in Modeling Earth Systems. – 2020. – Vol. 12, Issue 11. – DOI: 10.1029/2020MS002203.
- 16) The pandas development team. pandas-dev/pandas: Pandas 1.3.0 [Электронный ресурс]. – Zenodo, 2021. – URL: <https://pandas.pydata.org/> (дата звернения: 10.05.2025).
- 17) The rise of machine learning in weather forecasting [Электронный ресурс] // ECMWF. – 2023. – URL: <https://www.ecmwf.int/en/about/media-centre/science-blog/2023/rise-machine-learning-weather-forecasting> (дата звернения: 10.05.2025).
- 18) Vaughan A., Markou S., Tebbutt W., et al. Aardvark weather: end-to-end data-driven weather forecasting [Электронный ресурс] // arXiv. – 2024. – URL: <https://arxiv.org/abs/2404.00411> (дата звернения: 10.05.2025).

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Бакалавр**
Галузь знань: 15 – Автоматизація та приладобудування
Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології
Освітня програма «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри комп'ютерних
систем та робототехніки

к. ф.-м. н., доц. ХРУСЛОВ М. М.

«02» жовтня 2024 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Сечко Валерії Валеріївни

(прізвище, ім'я, по батькові студента)

1. Тема роботи Розробка моделі застосунку для інтелектуального прогнозування погоди на основі ШІ

керівник роботи Мірошник Марина Анатоліївна, професор, доктор технічних наук,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 16 квітня 2025 року №4101-5/962

2. Строк подання студентом роботи 30 травня 2025 року

3. Перелік питань, які потрібно розробити)

- Аналіз сучасних підходів до прогнозування погоди з використанням штучного інтелекту та машинного навчання та огляд моделей, що вже існують.
- Розробка концептуальної моделі застосунку для прогнозування погоди на основі методів глибокого навчання для аналізу метеорологічних даних.
- Тренування та тестування моделі прогнозування погоди, оцінка її ефективності.
- Розробка інтерфейсу користувача для взаємодії із застосунком прогнозування погоди та тестування застосунку, оцінка його продуктивності та коректності прогнозів.
- Визначення перспектив розвитку застосунку та можливі шляхи покращення точності прогнозування.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Затвердження теми роботи	28.12.2024-24.01.2025
2	Огляд наукової літератури у сфері прогнозування на основі машинного навчання та штучного інтелекту	01.02.2025-15.03.2025
3	Аналіз існуючих методів прогнозування погоди за допомогою ШІ	17.03.2025-28.03.2025
4	Визначення вимог до моделі прогнозування та вибір алгоритмів машинного навчання	31.03.2025-04.04.2025
5	Розробка концептуальної моделі застосунку	07.04.2025-14.04.2025
6	Тренування та тестування моделі прогнозування погоди	14.04.2025-18.04.2025
7	Розробка програмної архітектури та інтерфейсу застосунку	21.04.2025-30.04.2025
8	Тестування роботи застосунку, аналіз точності прогнозів	01.05.2025-07.05.2025
9	Визначення обмежень та перспектив подальшого розвитку	08.05.2025-13.05.2025
10	Підготовка і оформлення звітних матеріалів та додатків кваліфікаційної роботи. Оформлення списку літератури	14.05.2025-21.05.2025
11	Оформлення пояснювальної записки кваліфікаційної роботи відповідно вимогам до звітів про НДР та оформлення презентації	29.05.2025
12	Презентація та захист кваліфікаційної роботи	30.05.2025

5. Дата видачі завдання 02 жовтня 2024 року.

Студент

Сечко В. В.

ініціали, прізвище

підпис

Керівник роботи

Мірошник М. А.

ініціали, прізвище

підпис

Затверджую

«_____» _____ 2024 р.

**Технічне завдання
на розробку програмного виробу
«Модель застосунку для інтелектуального прогнозування погоди на
основі ШІ»**

Назва розділу	Назва та зміст підрозділу
1. Вступ	1.1. Назва програмного виробу “Модель застосунку для інтелектуального прогнозування погоди на основі ШІ” 1.2. Галузь застосування: прогнозування погоди, гідрометеорологічні дослідження
2. Підстава для розробки	2.1. Навчальний план за спеціальністю 151 “Автоматизація та комп’ютерно-інтегровані технології” 2.2. Завдання на кваліфікаційну роботу - Розробка моделі застосунку для інтелектуального прогнозування погоди на основі ШІ
3. Призначення розробки	3.1. Мета розробки програмного виробу: створення моделі прогнозування погоди за допомогою методу аналізу даних шляхом виведення їх в окрему модель машинного навчання 3.2. Призначення програмного виробу: отримання, обробка та аналіз метеорологічних даних з подальшим формуванням погодного прогнозу. 3.3. Вихідні дані для розробки: набори історичних метеорологічних даних з відкритих джерел NOAA, методичні матеріали щодо проєктування інтелектуальних систем, бібліотеки numpy, pandas, scikit-learn
4. Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: модель повинна здійснювати обробку вхідних метеорологічних даних, виконувати прогнозування погодних показників із заданою точністю та забезпечувати можливість тестування результатів для подальшого аналізу.

	4.2. Вимоги до надійності модель має забезпечувати стабільне функціонування при багаторазовому запуску на різних наборах даних, бути стійкою до окремих пропущених або пошкоджених вхідних значень, а також мати контрольні метрики для оцінки її ефективності (похибка не перевищує 3,5) 4.3. Вимоги до умов експлуатації: модель повинна запускатися в середовищі, яке підтримує відповідне програмне забезпечення Python із бібліотеками numpy, pandas, sklearn на персональному комп'ютері або сервері з доступом до джерел метеорологічних даних. 4.4. Вимоги до складу параметрів технічних засобів: сумісність із середовищами Python (версії 3.8 і вище), підтримка стандартних форматів вхідних/вихідних даних (CSV, JSON), можливість інтеграції з зовнішніми системами через API або скрипти для обміну даними. 4.5. Вимоги до інформаційної та програмної сумісності: модель повинна бути сумісною із середовищем Python версії 3.8 або вище, підтримувати обробку вхідних та вихідних даних у форматах CSV та JSON, а також забезпечувати можливість інтеграції з зовнішніми сервісами через API або скрипти для обміну даними. 4.6. Вимоги до маркіровки та упаковки: фізична упаковка не потрібна. 4.7. Вимоги до транспортування та зберігання: модель та супровідні файли мають зберігатися в цифровому вигляді у репозиторіях з контролем версій на Git, хмарних сховищах або локальних серверах. Зберігання повинно гарантувати цілісність даних і можливість резервного копіювання. 4.8. Спеціальні вимоги: відсутні
5. Вимоги до програмної документації.	Програмою документацією до виробу “Модель застосунку для інтелектуального прогнозування погоди на основі ШІ” вважати: 1) Справжнє Технічне завдання на розробку програмного виробу;

	2) Програму та методику випробувань розробленого програмного виробу; 3) Опис програмного виробу (не потрібно). 4) Текст програми (не потрібно).
6. Техніко-економічні показники	У цьому розділі можуть бути: 1) орієнтовна оцінка ефективності розробленого виробу; 2) терміни та можливі витрати коштів на розробку програмного виробу; 3) порівняння економічних показників розробленого програмного виробу з вітчизняними та зарубіжними зразками чи аналогами; 4) інші подібні питання.
7. Стадії та етапи розробки	Аналітична стадія (жовтень 2024 – березень 2025): - Затвердження теми роботи (02.10.2024-05.11.2025) - Огляд наукової літератури з прогнозування погоди за допомогою ШІ (10.11.2025-15.02.2025) - Аналіз існуючих методів прогнозування (17.02.2025 – 28.03.2025) Проектна стадія (березень – квітень 2025): - Визначення вимог до моделі та вибір алгоритмів машинного навчання (31.03.2025 – 04.04.2025) - Розробка концептуальної моделі (07.04.2025 – 14.04.2025) Реалізаційна стадія (квітень – травень 2025): - Програмування (кодування) моделі, її тренування та тестування (14.04.2025 – 18.04.2025) - Тестування роботи моделі, аналіз точності прогнозів (01.05.2025 – 07.05.2025) Завершальна стадія (травень 2025): - Визначення обмежень та перспектив подальшого розвитку (08.05.2025 – 13.05.2025) - Підготовка та оформлення звітних матеріалів, додатків і списку літератури (14.05.2025 – 21.05.2025)

	- Оформлення пояснювальної записки та презентації (29.05.2025) - Презентація та захист кваліфікаційної роботи (30.05.2025)
8. Порядок контролю та приймання	У цьому розділі мають бути зазначені загальні вимоги до приймання розробленого програмного виробу, наприклад: 1) Перевірку ходу розроблення заданих проектних документів керівнику курсової роботи здійснювати 2 рази на місяць. 2) Розроблені проектні документи подати в електронному вигляді та на паперових носіях в одному примірнику.

Виконавець
студентка групи КУ-41
Сечко В.В.



Замовник
доктор техн. наук, професор
Мірошник М.А.



**Програма та методика випробувань програмного виробу
«Модель застосунку для інтелектуального прогнозування погоди на
основі ШІ»**

1. Об'єкт випробувань

1.1 Найменування програмного виробу: “Модель застосунку для інтелектуального прогнозування погоди на основі ШІ”.

1.2 Область застосування: прогнозування погоди, гідрометеорологічні дослідження.

2. Мета випробувань

Підтвердження відповідності функціональних, надійнісних та якісних характеристик розробленої моделі прогнозування погоди вимогам, зазначеним у Технічному завданні, шляхом оцінки точності прогнозів, стійкості до варіацій вхідних даних і здатності моделі до відтворюваності результатів.

3. Загальні положення

3.1 Підстави щодо випробувань

Підставою для проведення випробувань є: 1) Наказ ХНУ № 4101-5/962 від 16.04.2025 про затвердження тем кваліфікаційних робіт бакалаврів; 2) Навчальний план дисципліни».

3.2 Місце та тривалість випробувань

Наприклад, "Приймальні випробування проводяться на базі комп'ютерного класу кафедри".

Підрозділ 3.3. «Обсяг випробувань»

Приймальні випробування програмного виробу проводяться в обсязі відповідно до цієї Програми та методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться за участю Замовника, Виконавця та членів комісії, призначених для прийому курсової роботи.

4. Вимоги до програмного виробу

Функціональні вимоги: програма повинна забезпечувати завантаження вхідного датасету (CSV, Excel або ін.), здійснюється масштабування, нормалізація або перетворення даних перед навчанням.

Програма повинна створювати та навчати модель лінійної регресії з регуляризацією типу Ridge (Ridge з бібліотеки scikit-learn) на основі історичних метеоданих, що містять середню, мінімальну та максимальну температуру (TAVG, TMIN, TMAX), для прогнозування цільової змінної target1.

Модель забезпечує можливість зміни: моделі (через заміну об'єкта reg — Ridge можна налаштовувати або замінити іншим типом регресії); ознакової частини (змінна predictors, яка задає набір вхідних параметрів моделі); навчального періоду (train = core_weather[:"2022-12-31"]); гіперпараметра регуляризації Ridge-моделі (наприклад, alpha, якщо явно задано: Ridge(alpha=...)).

5. Вимоги до програмної документації

Склад програмної документації, що подається на випробування включає:

- 1) Технічне завдання на розробку програмного виробу (представлено в Додатку 2 до курсової роботи).
- 2) Програма та методика випробувань розробленого програмного виробу (представлена в Додатку 3 до курсової роботи).

6. Засоби та порядок випробувань

6.1 Засоби випробувань

Випробування моделі прогнозування погоди проводяться на персональному комп'ютері з процесором не менше 1.6 ГГц, оперативною пам'яттю від 8 ГБ, бажано з GPU, у середовищі Python 3.8+ із використанням бібліотек Pandas, NumPy та вбудованих засобів візуалізації. Застосовуються тестові скрипти та інструменти аналізу точності прогнозів, модель запускається у вигляді Python-скриптів.

6.2 Порядок проведення випробувань

Випробування проводять у два етапи:

1-й етап - Ознайомчий

1-й підетап: Перевірка комплектності програмної документації

- Перевірка комплектності представленої програмної документації здійснюється за критерієм відповідності її складу Технічному завданню.
- Перевірка комплектності технічних та програмних засобів здійснюється за критерієм відповідності їх складу Технічному завданню.

2-й підетап: Перевірка якості документації

- Якість програмної документації відповідає вимогам ДЕСТ 19.301-79 ЄСПД.
- Вся документація (опис, інструкція користувача, технічне завдання, вихідний код) представлена у повному обсязі, структурована та відповідає стандартам.

2-й етап – Власне випробування програмного виробу

Тест 1 – Перевірка працездатності моделі

Методика проведення:

- Запуск програми здійснюється через середовище Visual Studio Code.
- Вибирається файл train.ipynb.
- Послідовно виконуються всі клітинки коду.
- Перевіряється відсутність помилок виконання.
- Очікуваний результат – виведення графіків, метрик або результатів обчислень.

Висновок: в результаті виконання Тесту 1 модель вважається працездатною, оскільки виконання всіх клітинок завершилось успішно без помилок.

Тест 2 – Перевірка виконання функціональних вимог

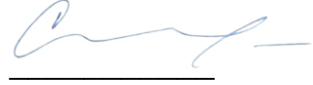
Методика проведення:

- Перевірка, чи здійснюється завантаження та попередня обробка даних.
- Перевірка, чи здійснюється побудова та навчання моделі.
- Перевірка, чи виводяться результати обробки/тестування.
- Перевірка на правильність реалізації функцій: вивід графіків, точності, втрат, класифікаційних метрик.

Висновок: в результаті виконання Тесту 2 функціональність моделі відповідає заявленим вимогам технічного завдання.

Загальний висновок за результатами випробувань: програмний виріб вважається таким, що пройшов випробування в цілому, оскільки успішно виконані всі Тести 1 та 2.

Виконавець: студентка групи КУ-41Сєчко В.В.



Фрагмент лістингу коду з поясненнями

```
import numpy as np
import pandas as pd

weather = pd.read_csv("data/t_Kyiv.csv", index_col="DATE")

weather

# відсоток пустих вимірювань в рядках
weather.apply(pd.isnull).sum()/weather.shape[0]

# з всіх вимірювань вибираємо лише максимальну, мінімальну та
середню температури
core_weather = weather[["TAVG", "TMAX", "TMIN"]].copy()

core_weather

# пусті вимірювання
core_weather[pd.isnull(core_weather["TAVG"])]

# тут використовується форвард філ, тобто дублювання даних з
попереднього рядка
core_weather = core_weather.fillna(method="ffill")

core_weather.apply(pd.isnull).sum()/weather.shape[0]

# Далі будемо перевіряти чи всі дані мають правильний тип даних
для машинного навчання

core_weather.index
# тип індекса - object, що означає що це просто рядок, але для ML
треба це все перетворити в тип datetime

core_weather.index = pd.to_datetime(weather.index)

core_weather.index

# перевіряємо на помилки в вимірюваннях
core_weather.apply(lambda x: (x == 9999).sum())

#як ми бачимо в датасеті немає помилок в вимірюваннях

#З цього блоку коду починається аналіз даних
```

```

core_weather[["TMAX", "TMIN"]].plot()

core_weather[["TAVG"]].plot()

#З наступного блоку коду почнеться тренування моделі

# Спершу треба зрозуміти що ми хочемо передбачити. В даному
випадку ми будемо старатись передбачити
# максимальну температуру наступного дня використовуючи історичні
дані.

# Створимо для цього нову "таргетну" колонку,
# Ця колонка - це буде дублікат колонки TMAX, сдвинутий вгору

core_weather["target1"] = core_weather.shift(-1)["TAVG"]

core_weather

# як ми бачимо в колонці target1 кожний рядок певного дня вказує
на завтрашню погоду,
# тобто рядок дати першого січня 1970 вказує на погоду другого
січня 1970. І ці завтрашні температури
# в рядках є цілями, таргетами.
#
# Можна помітити, що в останньому рядку Nan, тому його можна
видалити

core_weather = core_weather.iloc[:-1, :].copy()
core_weather

# Нарешті можна приступити до машинного навчання

from sklearn.linear_model import Ridge

reg = Ridge(alpha=.1) # альфа - це коефіцієнт, який контролює
наскільки великий коефіцієнт регресивної моделі аналізує,
# чим більше Penalty, тим більше запобігає
overfitting моделі (див. в інтернеті)

predictors = ["TAVG", "TMIN", "TMAX"] # Які колонки слугують для
передбачення

train = core_weather[:"2022-12-31"] # на цьому будемо тренувати
тестову модель
test = core_weather["2023-01-01":] # а на цьому будемо перевіряти

```

передбачення

```
# Тут ми "засовуємо" в модель передбачення наші дані
(train[predictors])
# та до чого треба прямувати при передбаченні (train["target1"])
reg.fit(train[predictors], train["target1"])

predictions = reg.predict(test[predictors]) # Генеруємо
передбачення

from sklearn.metrics import mean_absolute_error
mean_absolute_error(test["target1"], predictions) # вираховує
середню абсолютну похибку

# як ми бачимо середня похибка поміж передбачень сягає приблизно
3.3 градуси,
# що є +- нормально, (((але можна покращити точність))).

# Скомбінуймо реальні дані з передбаченнями

combined = pd.concat([test["target1"], pd.Series(predictions,
index = test.index)], axis=1)
combined.columns = ["actual", "predictions"]

combined

combined.plot()

reg.coef_
# це коефіцієнти моделі (формули), в яку підставляються у вчорашні
дані для прогнозування сьогодні.
# тобто це приблизно виглядає як future_t = 0.41*t_avg-
0.06*t_min+0.064*t_max+const

# Зробимо функцію для передбачень

def create_predictions(predictors, core_weather, reg):
    train = core_weather[:"2022-12-31"] # на цьому будемо
    тренувати тестову модель
    test = core_weather["2023-01-01":] # а на цьому будемо
    перевіряти передбачення
    reg.fit(train[predictors], train["target1"])
    predictions = reg.predict(test[predictors])
    error = mean_absolute_error(test["target1"], predictions)
    combined = pd.concat([test["target1"], pd.Series(predictions,
index = test.index)], axis=1)
```



```

combined.columns = ["actual", "predictions"]
return error, combined

# далі ми створимо більше колонок для передбачення

# rolling(30).mean() робить наступне: для кожного дня rolling бере
попередні 30 днів
# а mean вираховує середнє значення. Колонка month_max має в собі
середні значення
# попередніх 30 днів, тому в перших 30и рядках буде NAN
core_weather["month_avg"] =
core_weather["TAVG"].rolling(30).mean()

eps = 1e-6

# наступна нова колона буде показувати наскільки температура рядка
відрізняється
# від середнього значення за попередній місяць.
core_weather["month_day_avg"] = core_weather["month_avg"] /
(core_weather["TAVG"]+eps)

# наступна нова колона буде показувати наскільки максимальна
температура відрізняється
# від мінімальної температури
core_weather["max_min"] = core_weather["TMAX"] /
(core_weather["TMIN"]+eps)

# видаляємо перші 30 рядків, які мають NAN
core_weather = core_weather.iloc[30:,:].copy()

core_weather

predictors = ["TAVG", "TMIN", "TMAX", "month_avg",
"month_day_avg", "max_min"] #, "month_day_avg", "max_min"
error, combined = create_predictions(predictors, core_weather,
reg)

error

combined.plot()

# Спочатку ми групуємо дані по місяцях
.groupby(core_weather.index.month),
# для кожного дня з місяця .apply(lambda x: x.expanding(1).mean())
бере
# кожен попередній рядок даного рядка та розраховує середнє

```

```

значення
core_weather["month_avg_t"] =
core_weather["TAVG"].groupby(core_weather.index.month).apply(lambda
a x: x.expanding(1).mean())

# Спочатку ми групуємо дані по роках
.groupby(core_weather.index.year),
# для кожного дня з місяця .apply(lambda x: x.expanding(1).mean())
бере
# кожен попередній рядок даного рядка та розраховує середнє
значення
core_weather["day_of_year_avg"] =
core_weather["TAVG"].groupby(core_weather.index.year).apply(lambda
x: x.expanding(1).mean())

core_weather

predictors = ["TAVG", "TMIN", "TMAX", "month_avg",
"month_day_avg", "max_min", "month_avg_t", "day_of_year_avg"] #,
"month_day_avg", "max_min"
error, combined = create_predictions(predictors, core_weather,
reg)

```