Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики
Кафедра системного аналізу та інформаційних технологій

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ ОПАДІВ

Пояснювальна записка до магістерської кваліфікаційної роботи

Виконала: студентка 2 курсу, групи 2ІСТ-19м
спеціальності 126 – «Інформаційні системи та технології»
Мельник О. Л.

Керівник: к.т.н., доц. Козачко О. М. _____

Рецензент: к.т.н., доц. Бойко О. Р. _____

Вінниця ВНТУ – 2020 року

Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики
Кафедра системного аналізу та інформаційних технологій

Освітньо-кваліфікаційний рівень магістр
Спеціальність 126 - Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. В. Б. Мокін
"__" _____ 2020 р.

ЗАВДАННЯ
на магістерську кваліфікаційну роботу студентці
Мельник Олені Леонідівні

1. Тема роботи: «Інформаційна технологія аналізу та передбачення опадів», керівник роботи: Козачко О. М., к.т.н., доц. каф.САІТ, затверджені наказом закладу вищої освіти від "__"__2020 року №__
2. Строк подання студенткою роботи_____
3. Вихідні дані до роботи:
   - дані погодних умов м. Вінниця в період з 2010 по 2020 роки Вінницького обласного центру з гідрометеорології.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
   - Обґрунтування доцільності створення інформаційної технології аналізу та передбачення опадів;
   - Розробка інформаційної технології аналізу та передбачення опадів;
   - Реалізація інформаційної технології аналізу та передбачення опадів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
   - Дані після видалення зайвих записів;
   - Діаграма кількості значень нижнього показника хмарності;
   - Діаграма кількості значень показника хмарності;
   - Діаграма важливості атрибутів згідно результатів моделі lgbm;
   - Діаграма важливості атрибутів згідно результатів моделі xgb;
   - Діаграма значень коефіцієнтів важливості атрибутів;
   - Графік залежності опадів від значень температури;
   - Графік залежності опадів від значень вологості.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Руда Л. П., к.е.н., доц. каф. ЕПВМ		

7. Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів МКР	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	09.2020	
2	Огляд існуючих технологій передбачення опадів	09.2020	
3	Розробка інформаційної технології	09.2020	
4	Економічна частина	10.2020	
5	Аналіз головних ознак	10.2020	
6	Реалізація та тестування інформаційної технології	11.2020	
7	Оформлення матеріалів до захисту МКР	11.2020	

Студентка _____ Мельник О. Л.

Керівник роботи _____ Козачко О. М.

Рецензент _____ Бойко О. Р.

РЕФЕРАТ

Магістерська кваліфікаційна робота: 96 стор., 7 табл., 69 рис., 20 джерел.

Об’єкт досліджень – процес передбачення опадів на основі аналізу попередніх метеоданих.

Мета роботи – підвищення точності прогнозування наявності опадів за рахунок використання інформаційних технологій, машинного навчання та аналітичної обробки даних.

Розглянуто історію розвитку передбачення погодних умов, проведено аналіз існуючих технологій аналізу та передбачення опадів. Сформовано датасет для подальшої розробки на основі даних Вінницького обласного центру з гідрометеорології. Проведено розробку інформаційної технології завдяки порівняльному аналізу можливих інструментів та засобів для реалізації. Реалізовано та протестовано технологію аналізу та передбачення. Подальшого розвитку набув метод передбачення опадів Вінницького регіону, який на відмінну від існуючих, визначає інформативні ознаки впливу на основі яких здійснюється прогнозування наявності опадів за рахунок використання алгоритмів машинного навчання, на основі попередньо сформованого набору даних.

Прогнозні припущення про розвиток об’єкта дослідження – розробка інформаційної технології аналізу та передбачення опадів, яка дозволить зчитувати та записувати показники з метеоприладів та приймачів супутників в реальному часі та здійснювати негайну їх обробку та аналіз, збільшення обсягу даних для навчання та підвищення точності роботи моделі.

Галузь застосування – інформаційні технології, спрямовані на аналіз та передбачення погодних умов в локальній місцевості.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ ОПАДІВ, ПРОГНОЗУВАННЯ НАЯВНОСТІ ОПАДІВ, АНАЛІЗ ДАНИХ, МАШИННЕ НАВЧАННЯ.

ABSTRACT

Master's qualification work: 96 pages, 7 tables, 69 pictures, 20 sources.

The object of research – the process of precipitation prediction based on the analysis of previous meteorological data.

The purpose of the work – improving the accuracy of forecasting the presence of precipitation through the using of information technology, machine learning and analytical data processing.

The history of development of weather forecasting was considered, the analysis of existing technologies of analysis and forecasting of precipitations was investigated. The dataset was formed for further development based on the data from the Vinnytsia Regional Center for Hydrometeorology. The development of information technology through comparative analysis of possible tools and means for implementation. The technology of analysis and prediction of precipitation on the basis of a pre-formed dataset was implemented and tested.

Estimated assumptions about the development of the object of study – the development of information technology for analysis and prediction of precipitation, which will read and write indicators from meteorological instruments and satellite receivers in real time and make their immediate processing and analysis, increase the amount of data for training and increase the accuracy of the model.

Scope of application – information technologies aimed at analyzing and predicting weather conditions in the local area.

INFORMATION TECHNOLOGY OF ANALYSIS AND PREDICTION OF PRECIPITATION, PRECIPITATION OF PRECIPITATION, DATA ANALYSIS, MACHINE LEARNING.

ЗМІСТ

ВСТУП

Сьогодні важко уявити світ, в якому ми не вміємо передбачати погодні умови. Кожен другий, а то і перший, щодня перевіряє погоду перед виходом з дому чи поїздкою у відпустку за межі міста, країни, материка. Можливість дізнатися погоду на декілька днів вперед стала повсякденною справою, проте так було не завжди. За часів Арістотеля люди вже почали підкреслювати певні закономірності у змінах погодних умов та спостерігати за ними систематично, щоправда тоді вони грунтувались лише на рівні прикмет. З часом мореплавці вдосконалили передбачення погоди, але це залишалось все тим же мистецтвом, а не наукою, адже тоді ніхто не брав до уваги головний чинник погоди – атмосферний тиск [1].

З розвитком інформаційних технологій зросла і якість передбачення погодних умов. Одним з найпопулярніших напрямів сьогодення є машинне навчання (штучний інтелект).

Для проведення якісного грунтовного аналізу та передбачення погодних умов необхідно мати достовірні дані за попередні дні чи роки, структурувати їх та виділити основні види даних, на які буде покладатись робочий алгоритм.

Наразі такі дані ми можемо отримати з відкритих джерел в мережі Інтернет, наприклад сайти sinoptik.ua, gismeteo.ua, rp5.ua та інших, але найнадійнішим джерелом з даними, підкріпленими думкою фахівців даної галузі, є офіційний сайт Українського Гідрометцентру (meteo.gov.ua).

В основі спостережень та передбачення погоди вище згаданого гідрометцентру лежать супутникові спостереження та показники стаціонарних приладів на території центру (наприклад, опадовимірювач, витяжні термометри, прилади для вимірювання промерзання грунту і тому подібні). Цей метод прогнозування називається синоптичним.

Основним недоліком моніторингу в синоптичних методах є непостійність у часі та просторі, оскільки моніторинг повітря здійснюється під час передачі даних між різними станціями. Ще однією важливою особливістю карти є те, що

станції, з яких можна використовувати дані погодних умов, адже вони знаходяться на відстані не менше 100-150 км одна від одної. Насправді відстань між наземними кліматичними станціями, особливо в гірських районах і в пустельних районах, може становити близько 1000-1500 км, а повітряними – 2000-3000 км. Саме тому можна підставити під сумнів якість збору та обробки даних, оскільки карти погоди збираються протягом 6 годин після заміру показників та карти топографії протягом доби, що призводить до ігнорування можливих змін отриманих значень параметрів та впливу даних змін на передбачення погоди в цілому [1-3].

Отже, розробка інформаційної технології аналізу та передбачення опадів, яка буде забезпечувати швидкий комплексний аналіз даних, буде доцільною з точки зору пошуку оптимального методу прогнозування.

Об'єктом дослідження є процес передбачення опадів на основі аналізу попередніх метеоданих.

Предметом дослідження є інформаційна технологія аналізу і передбачення опадів на реальних даних.

Метою дослідження є підвищення точності прогнозування наявності опадів за рахунок використання інформаційних технологій, машинного навчання та аналітичної обробки даних.

Наукова новизна одержаних результатів. Подальшого розвитку набув метод передбачення опадів Вінницького регіону, який на відміну від існуючих, визначає інформативні ознаки впливу, на основі яких здійснюється прогнозування наявності опадів за рахунок використання алгоритмів машинного навчання.

Практичне значення одержаних результатів можна охарактеризувати наступними пунктами:

– розроблено просту для розуміння та зручну у використанні модель передбачення;

– модель не потребує вибірки даних зі складною структурою, використовується датасет простого формату.

Достовірність теоретичних положень магістерської кваліфікаційної роботи обґрунтовується дотриманням поставленої задачі, детальним ознайомленням з предметною областю та схожими технологіями для коректного розуміння значення та розробки технології, аналізом та поясненням доцільності застосування тих чи інших методів та інструментів реалізації, математично побудованим алгоритмом реалізації та результатами впровадження програмного засобу.

Апробація результатів роботи. Результати роботи були апробовані на XV Міжнародній конференції "Контроль і управління в складних системах" (КУСС-2020, м. Вінниця).

Публікації. За результатами магістерської кваліфікаційної роботи опубліковано: 1 тези на XV Міжнародній конференції "Контроль і управління в складних системах" (КУСС-2020) [4].

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ ОПАДІВ

1.1 Аналіз предметної області

У даній роботі за мету поставлено задачу створення інформаційної технології аналізу та передбачення опадів.

Можна припустити, що прогноз погоди був першим прогнозом, яким зацікавилось людство ще в давні часи аби знати сприятливі умови для робіт у полі (посів, обробка та збір урожаю). Тоді прогнозування опиралось на спостереження та формувались певні закономірності у зміні погодних умов, що зараз становить базу народних прикмет.

Першою офіційною організацією з прогнозування погоди став Метеорологічний департамент у 1854 році, що був заснований Торговою радою Англійського королівського товариства. Необхідність в такій організація полягала у розвитку мореплавства та добі великих географічних відкриттів, адже для вдалої подорожі океаном необхідно було знати не лише шлях, а й погодні умови, з яким доведеться зіштовхнутись у експедиції.

Перший в історії прогноз погоди був опублікований 1 серпня 1860 в «Times», його автором був Роберт Фіцрой [2].

На даному етапі моделювання прогнозу проводилось на основі комплексної роботи зі збору показників про поточний стан атмосфери з 24 метеостанції, організованих по Європі, за допомогою телеграфу на азбуці Морзе. В той же час було розроблено перші прототипи карти циклонів та антициклонів шляхом з’єднання лінією точок з однаковим тиском на карті (рис. 1.1).

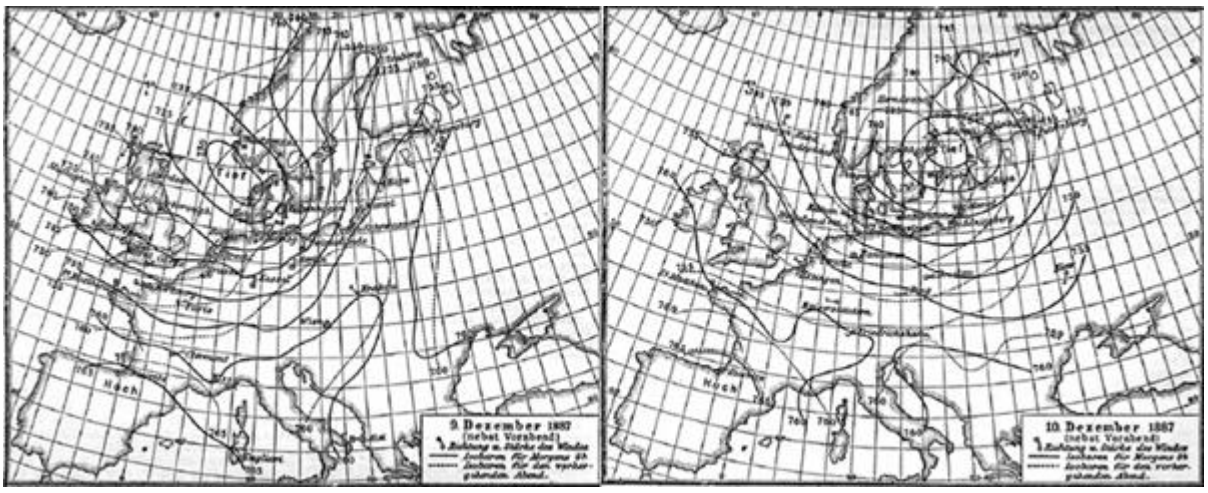


Рисунок 1.1 – Карта циклонів та антициклонів

З розвитком термодинаміки та гідродинаміки у XIX столітті на новий рівень вийшло і прогнозування, в якому почали з’являтись перші математичні підходи розв’язання проблеми.

В 1873 році було опубліковано роботу під назвою «Фізична основа довгострокового прогнозування погоди» («The physical basis of long-range weather forecasting»), де вперше було згадано про застосування математичних методів для опису погодних умов, автором якої був американець К. Еббе.

Точна постановка задачі прогнозування була запропонована норвежцем В. Б'єркнесом в 1904 році у праці «The problem of weather forecasting as a problem in mechanics and physics» («Проблема прогнозування погоди як проблема механіки та фізики») [3]. Саме цей вчений вперше виділив основні параметри впливу на стан атмосфери, серед яких тиск, температура, щільність, вологість і три види швидкості повітряних потоків, на основі яких розробив першу систему рівнянь для рішення задачі прогнозу погоди. Проте саме Л. Річардсон став тим, чиї роботи стали наслідувати наступні покоління метеорологів, адже він вперше застосував рівняння Б'єркнеса для інтегрування системи. Досвід даного дослідження було описано в роботі «Weather Prediction by Numerical Process» («Прогнозування погоди за допомогою числового процесу»), опублікованій 1922 року [2].

Яким чином наразі працює складання прогнозу погоди? Для цього збираються дані про поточний стан атмосфери та проектують зміни, покладаючись на обчислювальну техніку – суперкомп'ютери. Всі зібрані дані, такі як температура, вологість повітря, швидкість вітру, атмосферний тиск і так далі, збираються з усієї цільової території і завантажуються в систему комп'ютера. Світовим лідером в даній галузі наразі є European Centre for medium-range weather forecasts (ECMWF).

Головною проблемою та обмеженням в Україні в даному випадку виступає потужність техніки, від чого напряму залежить швидкість обрахунків та прогнозування. Також не менш важливим є людський фактор, адже для гарантії оновлення та актуальності погоднного прогнозу необхідно вчасно вносити зміни в раніше виміряних даних у разі виникнення потреби. Проте це не єдині чинники впливу, існують також чинники зі сторони атмосфери.

Навіть малопомітні зміни рівня моря можуть мати значний вплив на атмосферу на суші, тому можуть спостерігатися прогалини у спостереженнях. З цієї причини огляд не буде повним. Після формування прогнозу погоди на тиждень він швидше за все зміниться ближче до дати завершення.

Однак сучасні технології розвиваються. Наше розуміння комп'ютерної індустрії та атмосфери постійно змінюється. Як наслідок, очікувано, що прогнози мають більшу точність. За даними британського агентства зараз їх точність прогнозу на чотири дні дорівнює одноденному 30 років тому. Розвиток високотехнологічних методів прогнозування дозволить підвищити точність передбачення найскладнішого показника атмосфери, такого як опади, так як сказати місце, де б пройшов, наприклад, дощ достовірно не можливо. Якщо синоптики обіцяли у вашому місті опади, а їх так і не було, це не означає, що через кілька кілометрів від вас їх також не було, включно в межах вашого міста [5-7].

Не зважаючи на вище приведені причини похибок та помилок у прогнозуванні погоди, комп'ютерні технології розвиваються і наразі дозволяють швидко збирати та обробляти дані спостережень, ускладнюються алгоритми

аналізу даних та побудови передбачень, що дозволяє робити прогноз погодних умов точнішим і деталізованішим (рис. 1.2).

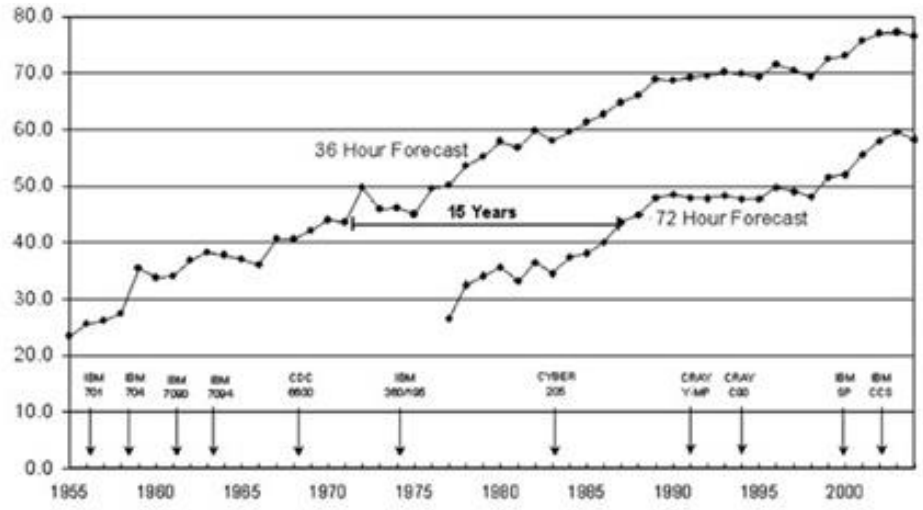


Рисунок 1.2 – Графік відображення прогресу точності прогнозування погодних умов

Основним завданням Українського гідрометцентру є забезпечення цілісної системи спостережень та прогнозування даними, що свідчать про стан атмосфери, навколишнього середовища, а також служать основою для передбачення погодних умов.

Проте ці задачі вимагають сучасного та потужного устаткування на всіх підрозділах системи, що наразі не реалізовано. Тому було вирішено створити технологію аналізу та передбачення опадів, що не вимагає таких затрат, для конкретного міста, а саме – Вінниці.

1.2 Огляд існуючих технологій передбачення опадів

Прогнозування погоди – це застосування науки і технологій для прогнозування стану атмосфери для даного місця і часу [8].

На даний момент галузь передбачення погодних умов розвивається в ногу з інформаційними технологіями. Наразі застосовується кілька технологій

прогнозу погоди, що в свою чергу належать до синоптичного або комп'ютерного прогнозу.

Синоптичний прогноз складається синоптиками на основі синоптичних карт, для цього необхідна висока підготовка та професійний досвід. Головна відмінність від комп'ютерного прогнозу полягає в тому, що кінцеве рішення лежить за людьми.

Комп'ютерний прогноз складається потужними серверами автоматично з використанням моделювання, обрахунку та обробки параметрів на період до 14 днів та з проміжком максимум в 1 годину. На рисунку 1.3 наведено приклад прогнозування на основі чисельної моделі прогнозування погоди.

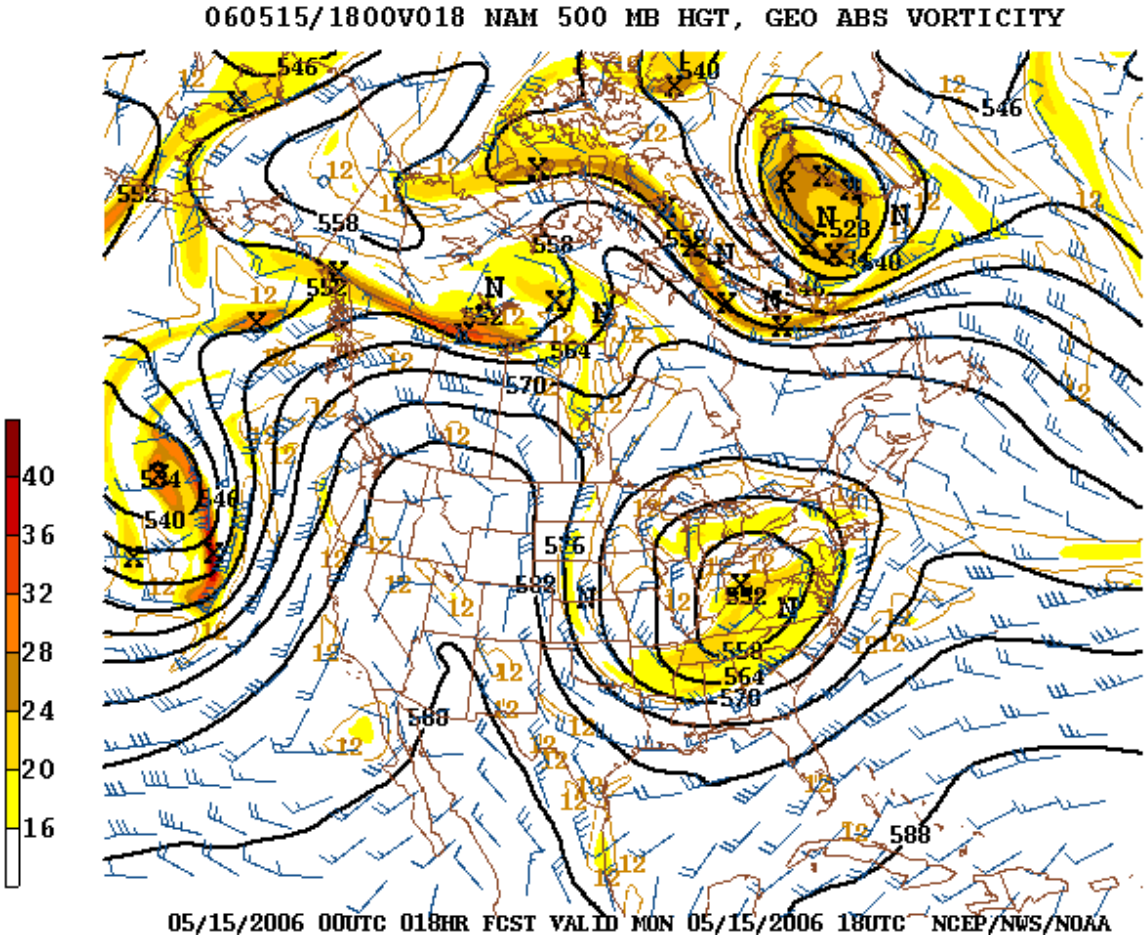


Рисунок 1.3 – Застосування чисельної моделі прогнозування

Якщо судити з точки зору точності методу прогнозування, то враховуючи похибку, синоптики дійшли висновку, що обидва з вище згаданих мають однаку похибку приблизно в 5%, тому жоден не дає 100% точного прогнозу.

Технологія середньострокового прогнозу погоди, що зазвичай застосовується при прогнозі передбачень на наступні 10 днів (рис 1.4). Дана технологія використовує якісні методи та кількісні моделі прогнозування для реалізації. Якісні методи прогнозування – це суб'єктивні думки експертів, що опираються на думки споживачів. Кількісні моделі прогнозування застосовують при прогнозуванні даних у вигляді функцій [9].

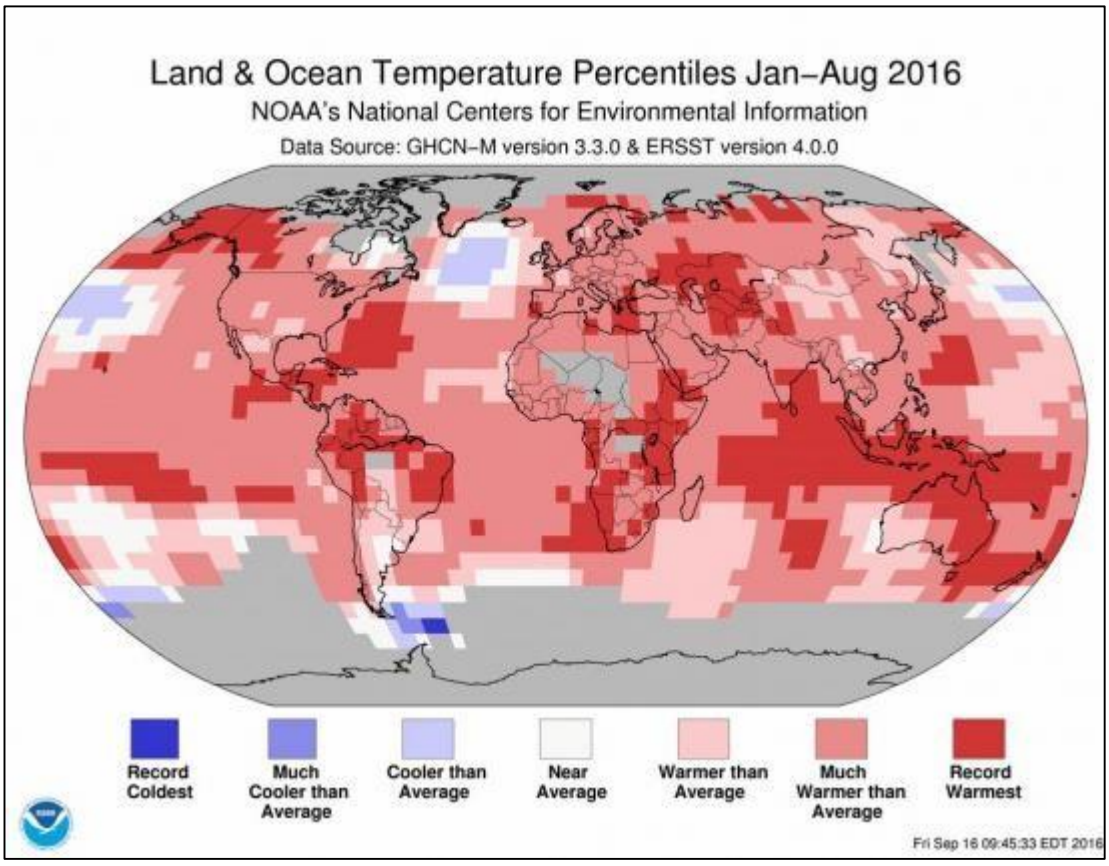


Рисунок 1.4 – Представлення температури верхніх шарів океану та земельних ділянок

Технологія Dual Polarization застосовується на метеорадарах. Поляризація хвилі – це напрям чи орієнтація електричного поля. Перевага технології полягає

в порівнянні потужності та замірах енергії, що радар отримує від горизонтальних та вертикальних імпульсів (рис. 1.5).

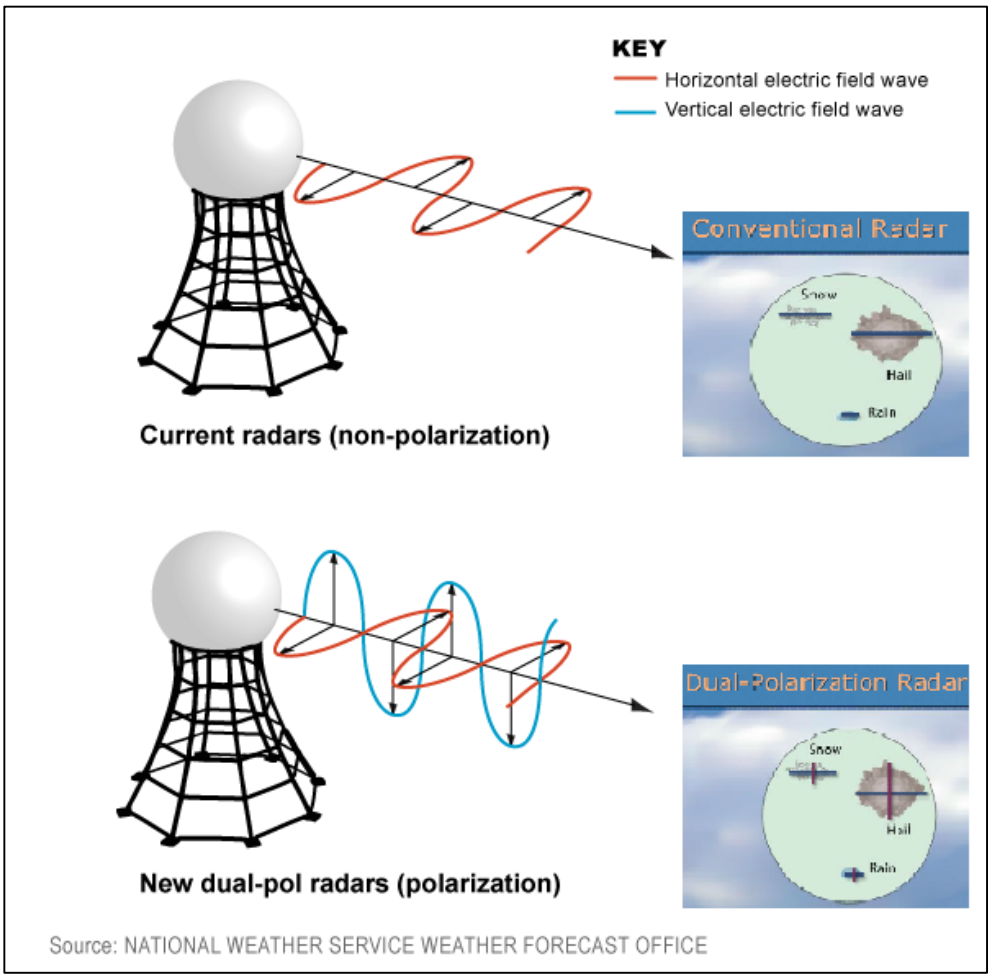


Рисунок 1.5 – Принцип роботи поляризованого радара

Таким чином радар в змозі отримати дані про розмір, форму та склад часток опадів, тоді як радари, реалізовані за допомогою інших технологій, могли виміряти лише енергію в повітрі, що поверталась від радіолокаційного променя у горизонтальному положенні. Як результат у процесі прогнозування стає можливою краща оцінка можливих опадів, їх розподілу по районах, їх класифікації на типи.

Довгостроковий метеорологічний прогноз (ДМП) спирається на атмосферу як частину єдиної системи, де вона є одним з шарів (рис. 1.6). Дана технологія визначає вплив зовнішніх факторів на атмосферу, які надають

помітний вплив на циркуляцію повітря вцілому, але на поточну погоду конкретно [9].

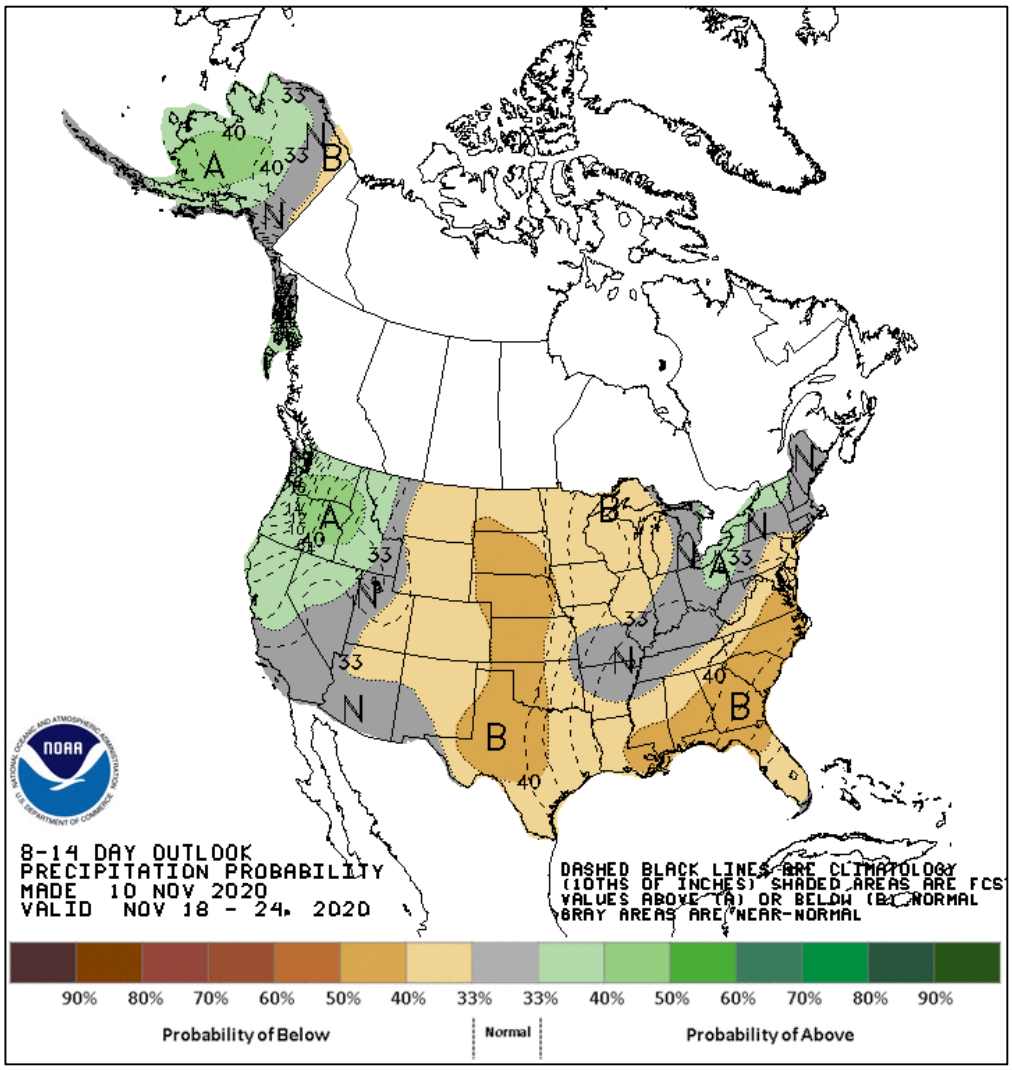


Рисунок 1.6 – Приклад довгострокового прогнозу опадів

Meteum – це нова технологія від компанії Yandex на основі моделі вивчення і прогнозування погоди WRF, реалізована за допомогою напряму машинного навчання [10]. Система розрахунків опирається на сітку двох розмірів (великим і дрібним розширенням) задля точності передбачень та місця прогнозу, з перспективою на збільшення покриття системи (рис. 1.7).

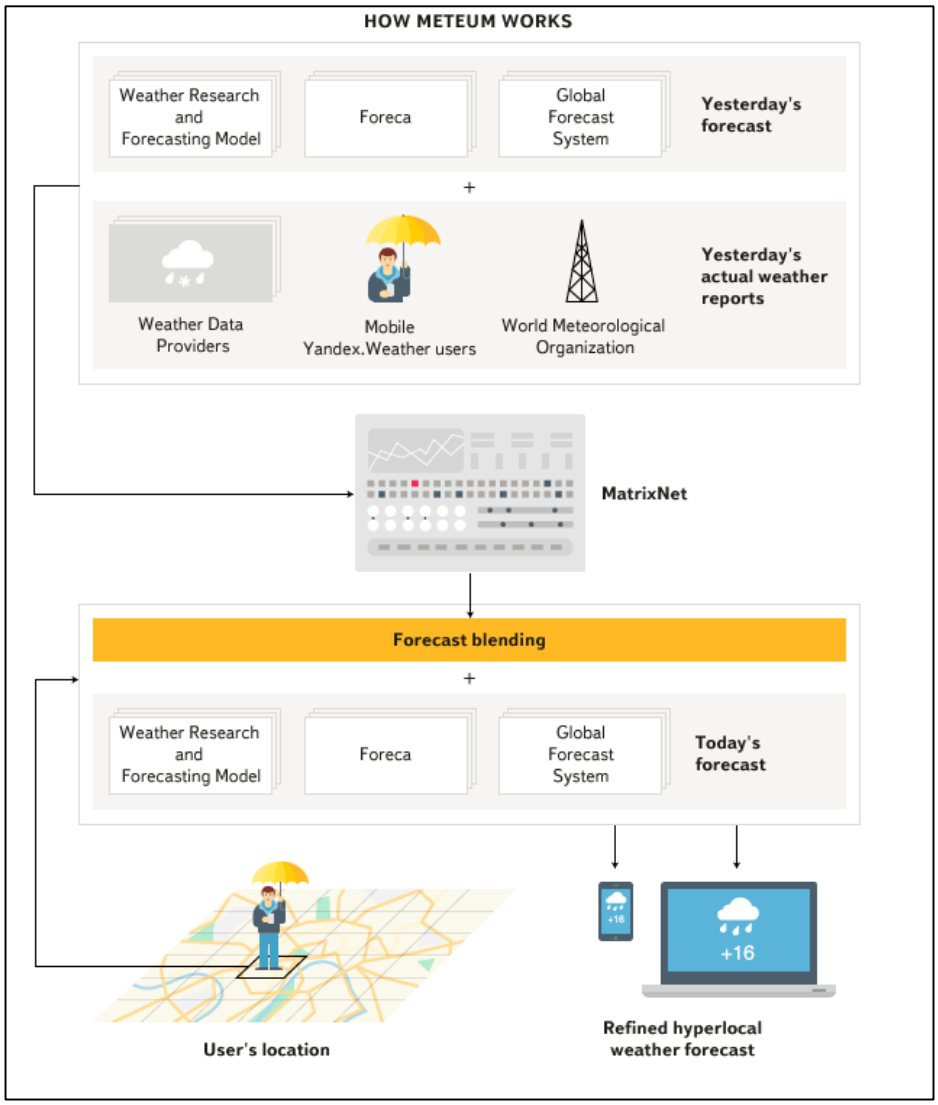


Рисунок 1.7 – Принцип роботи технології Meteum

Досить вдалі спроби створення нової технології прогнозування на звичайних, а не суперком'ютерах, продемонстрували в університетах Майцна і Лугано в Швейцарії на початку 2020 року. Систему побудовано на основі алгоритму SPA (scalable probabilistic approximation – ймовірність апроксимації піддатної масштабуванню). Точність створеної системи за попередніми показниками перевищує показник метеослубж близько на 40%. Обробка даних більшості алгоритмів ML відбувається за принципом так званого «чорного ящика», тобто мало хто розуміє як саме та що відбувається під час аналізу та обрахунку даних, що не стосується SPA, який дозволяє прослідкувати причинозв'язок між вхідними та вихідними даними системи. Саме тому даний алгоритм застосовують не лише в галузі передбачення погодних умов (рис 1.8).

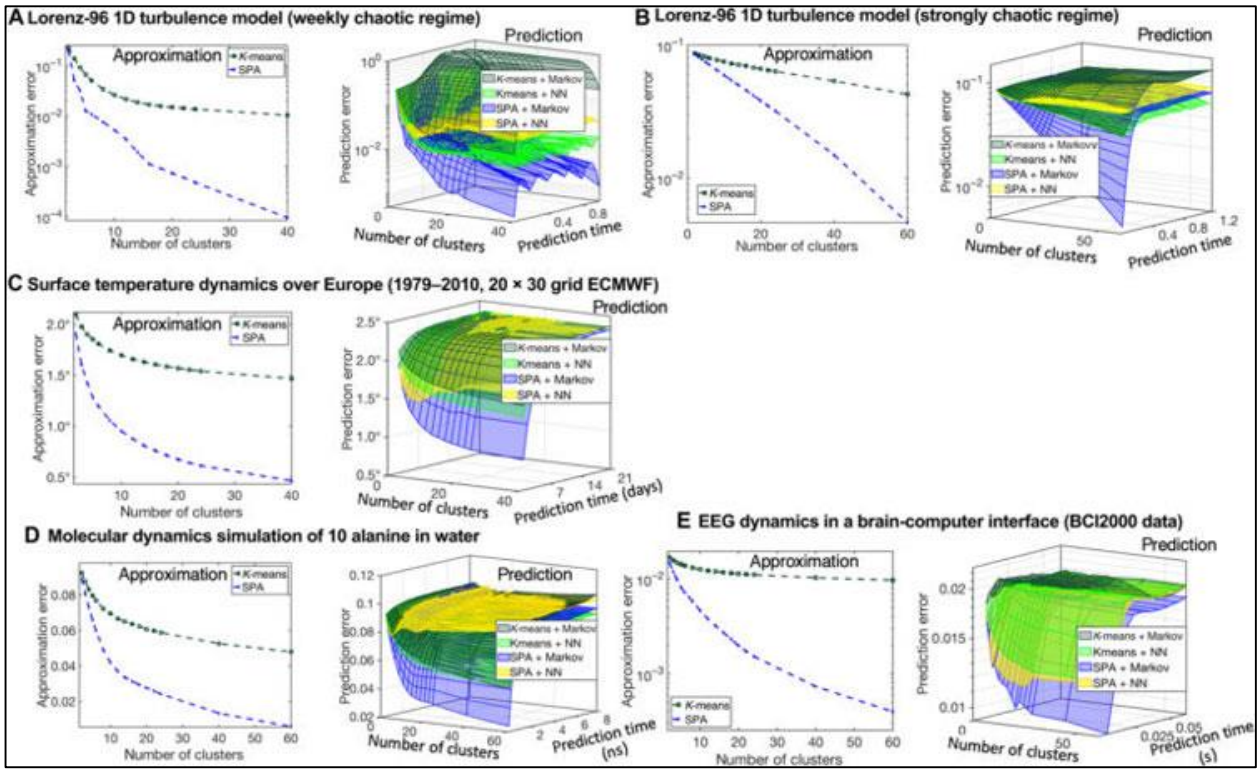


Рисунок 1.8 – Приклади використання алгоритма SPA для різних задач

Не зважаючи на розбіжності в специфіці роботи синоптичних та комп’ютерних методів та технологій передбачення погодних умов, всі з них працюють, опираючись на великі набори даних.

Однією з найпопулярніших платформ, де можна отримати відкриті набори даних, на даний час є система для машинного навчання Kaggle від компанії Google. Розглянемо два найпопулярніших публічних датасетів погодних умов даної платформи для прикладу (рис. 1.9).

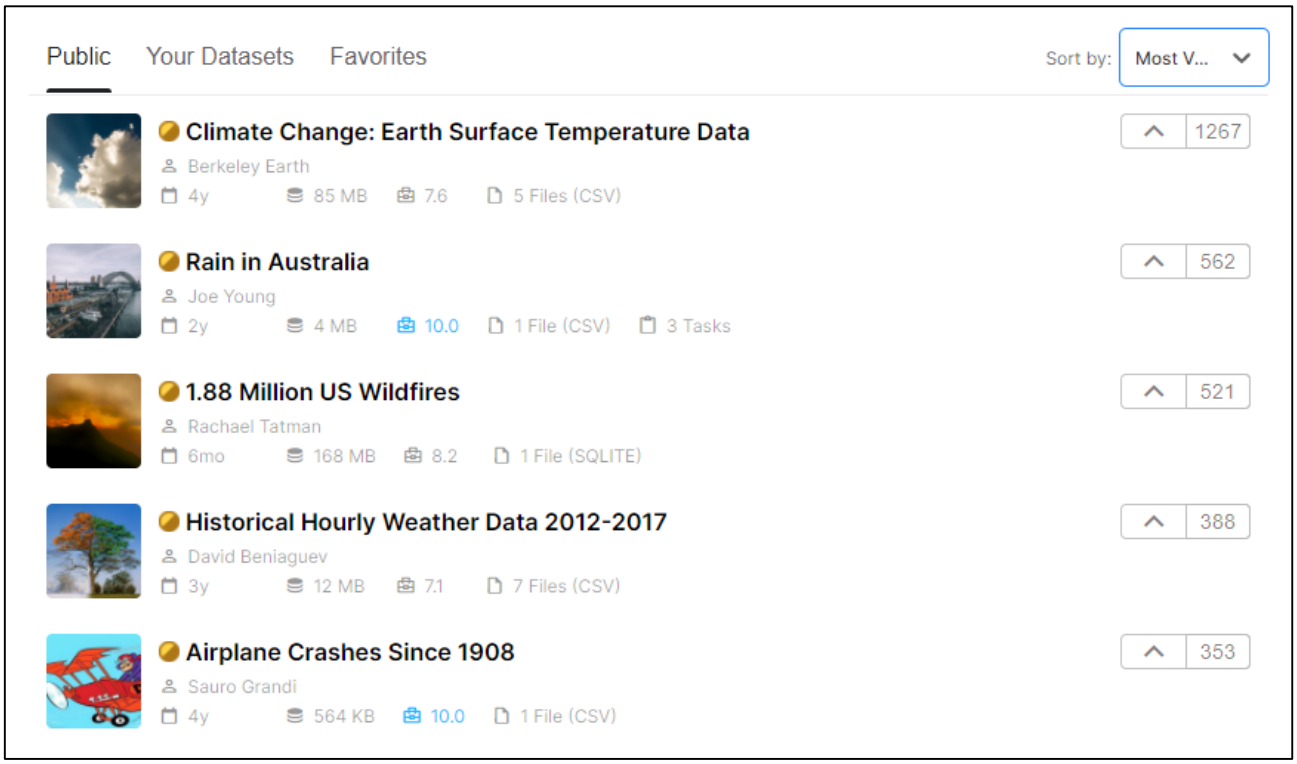


Рисунок 1.9 – Список найпопулярніших публічних погодних датасетів Kaggle

Датасет «Climate Change: Earth Surface Temperature Data» [11] містить світовий набір показників погодних умов, починаючи з 1750 року. Це набір даних, що був перезібраний зі збірки Національної лабораторії Лоуренса Берклі, який містив у собі 1,6 мільярдів звітів про температуру, що були розбиті по 16 архівах, де були показники для середньої температури суші з 1750 року та для максимальної/мінімальної температури суші, а також глобальних температур суші та океану з 1850 року. Датасет складається з наступних файлів:

– Глобальна середня температура землі за країнами (GlobalLandTemperaturesByCountry.csv) (рис. 1.10);

– Глобальна середня температура землі за штатами (GlobalLandTemperaturesByState.csv) (рис. 1.11);

– Глобальна температура землі за основним містом (GlobalLandTemperaturesByMajorCity.csv) (рис. 1.12);

– Глобальна температура землі за містом (GlobalLandTemperaturesByCity.csv) (рис. 1.13).

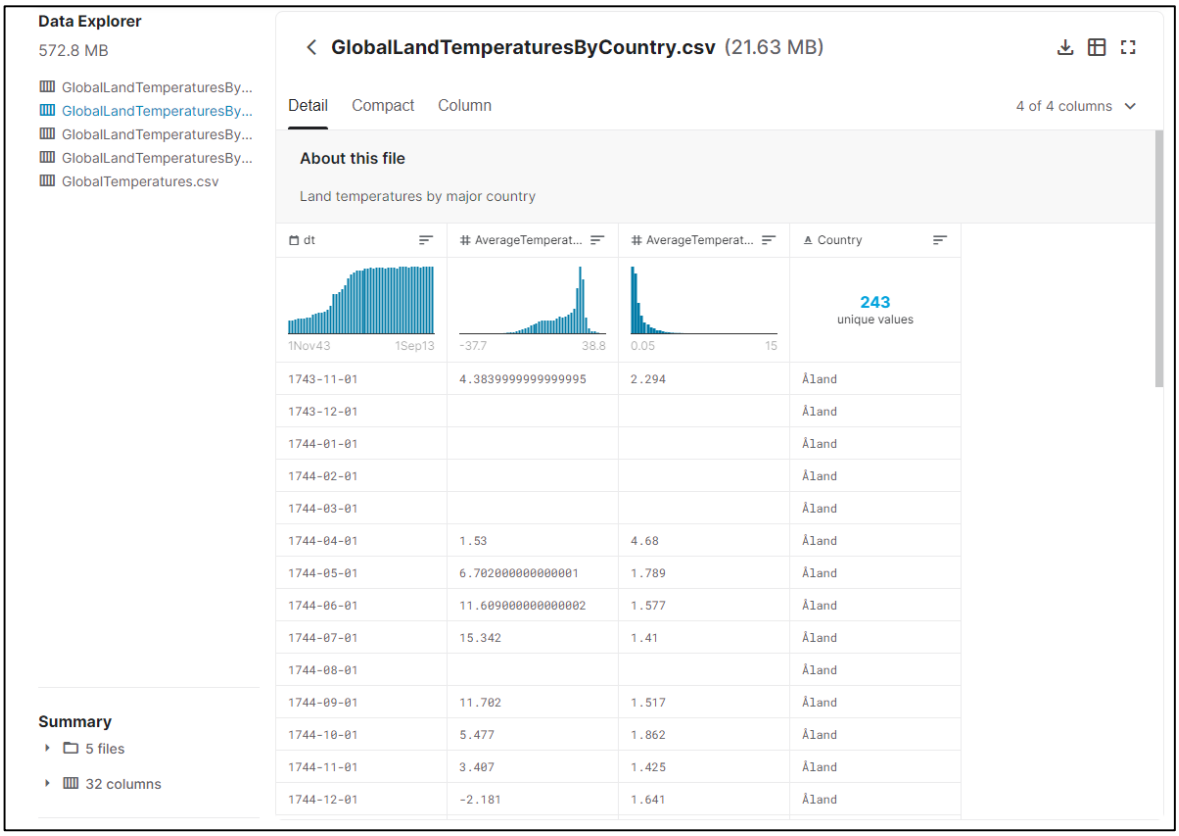


Рисунок 1.10 – Дані глобальної середньої температури землі за країнами

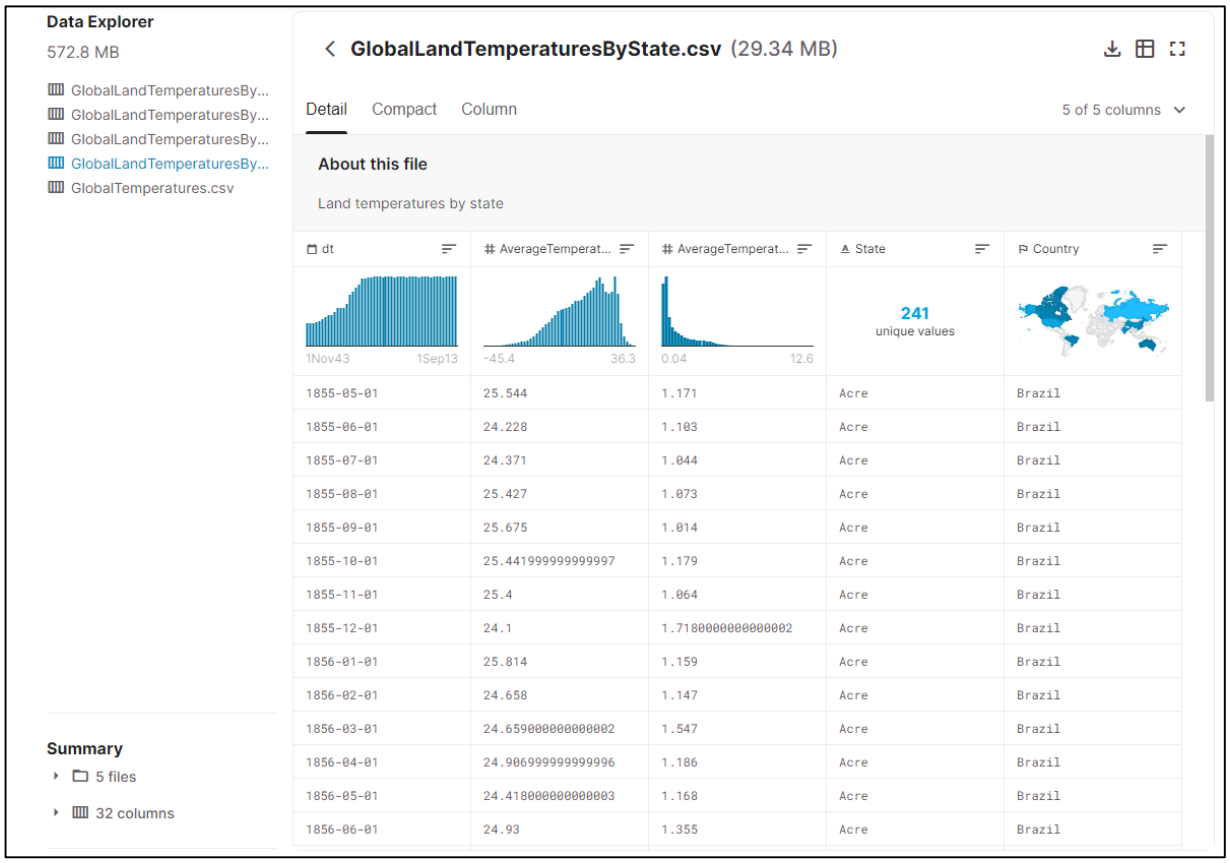


Рисунок 1.11 – Дані глобальної середньої температури землі за штатами

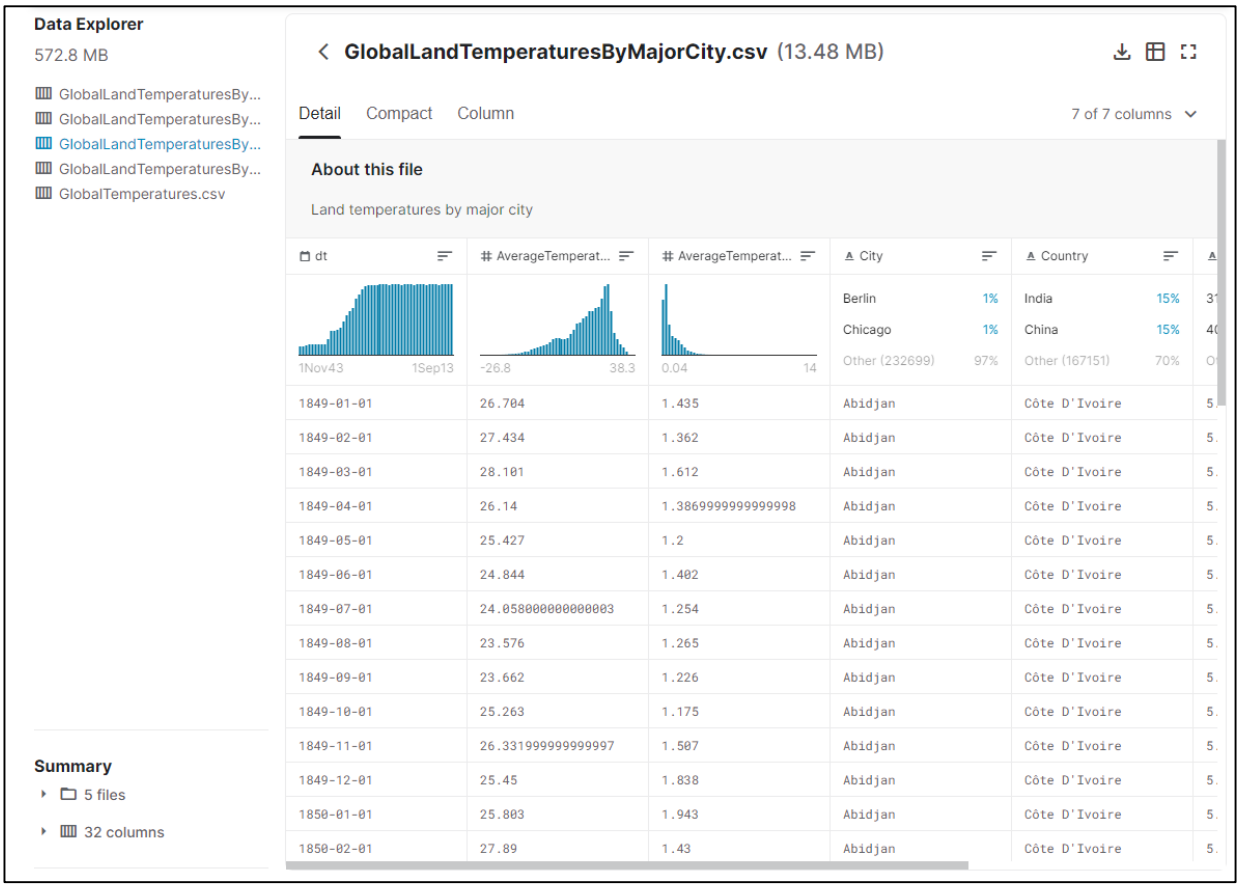


Рисунок 1.12 – Дані глобальної середньої температури землі за основним містом

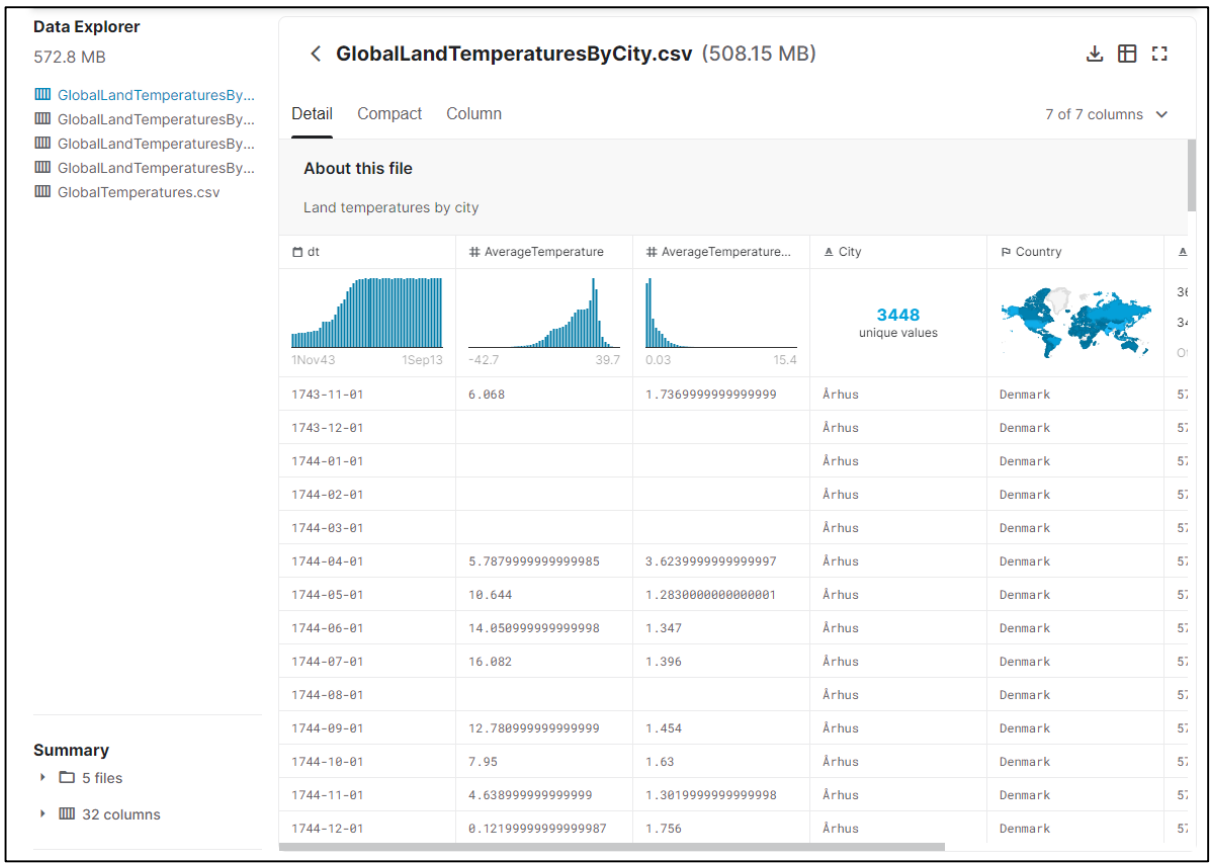


Рисунок 1.13 – Дані глобальної середньої температури землі за містом

Інший популярний відкритий датасет з погодними показниками «Rain in Australia» [12] − це набір даних, що містить близько 10 років щоденних спостережень за погодою на численних австралійських метеостанціях. Цільовим призначенням якого є дати відповідь на питання: «Чи піде дощ наступного дня? Так чи ні».

Датасет оформлено у вигляді одного файлу (рис. 1.14) з наступними показниками: локація метеостанції, де були зняті показники, мінімальна температура повітря (ºС), максимальна температура повітря (ºС), кількість опадів, зафіксованих в даний день (мм), випаровування повітря (мм) в період півночі по 9 годину ранку, кількість сонячних годин протягом доби, напрям найсильнішого пориву вітру протягом доби, швидкість найсильнішого пориву вітру протягом доби (км/год), напрям вітру о 9 годині ранку.

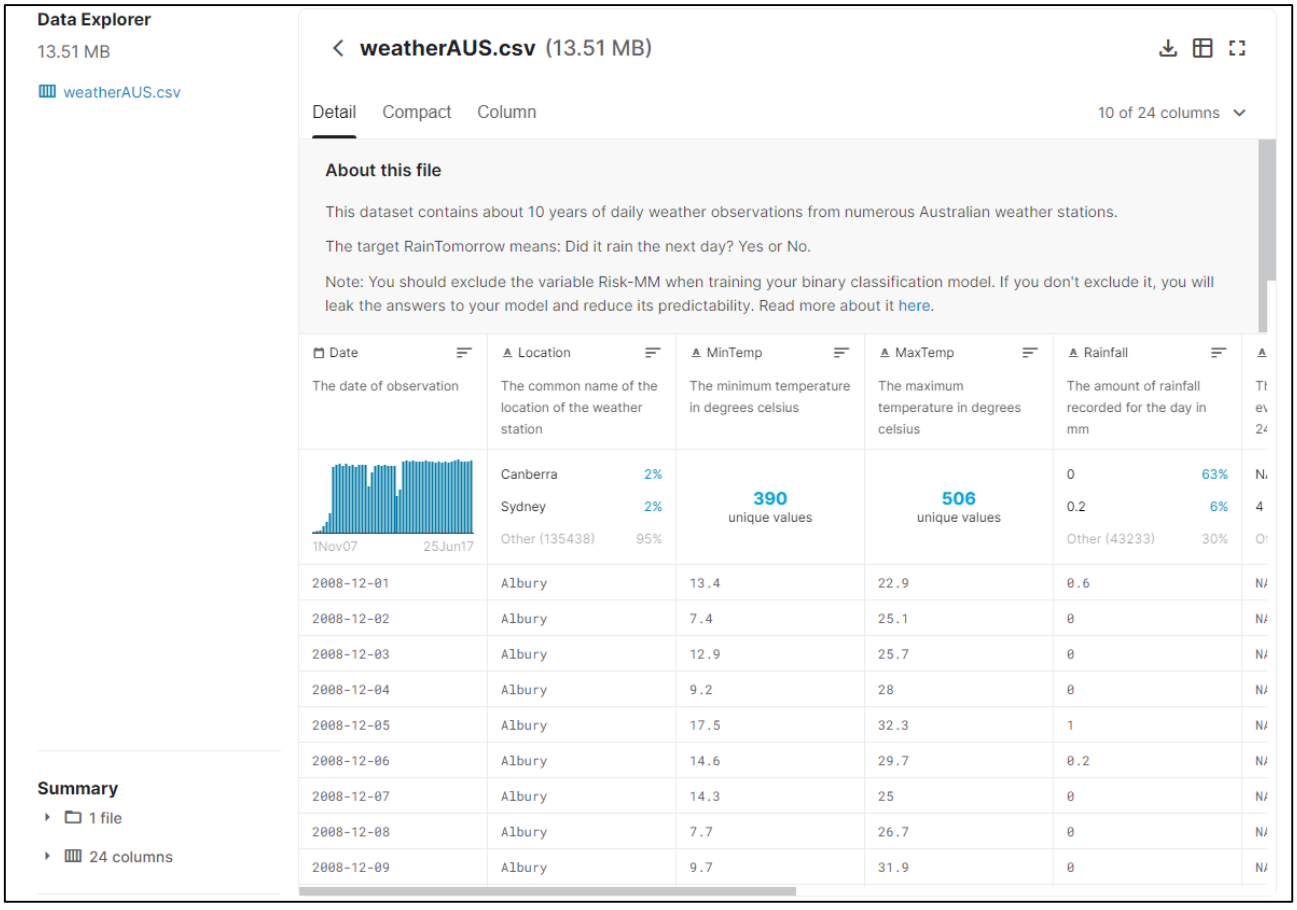


Рисунок 1.14 – Дані файлу датасету «Rain in Australia»

1.3 Теоретичні відомості про аналіз даних

Аналіз даних за допомогою математичних методів став часто використовуватись у дослідженнях різних напрямів діяльності та при реалізації нових технологій. Він набув популярності у сферах природничих, технічних, екологічних, соціальних, економічних та безлічі інших досліджень. Виходячи з цього, здобуття знань і навичок роботи з аналізу даних стали невід'ємною складовою підготовки спеціалістів в галузі інформаційних технологій, кібернетики, системних наук тощо.

Сьогодні значних обертів у застосування набирають нові методи реалізації напряму аналізу даних, такі як data mining (інтелектуальний аналіз даних). Дані методи спрямовані на виявлення неявних закономірностей, що спостерігаються у великих масивах даних, також застосовуються у побудові нейромереж і управлінській справі та на виробництві для контролю і якості товарів.

Сучасні методи аналізу даних зазвичай впроваджують за допомогою комп'ютерних технологій, використовуючи наявне програмне забезпечення, наприклад: MS Excel, MatLAB, MathCAD, Maple, або ж створюючи власні програми на основі побудови алгоритмів. Не зважаючи на те, що сьогодні існує чимало готового програмного забезпечення для реалізації аналізу даних, фахівцям все так же необхідно володіти знаннями з даного напряму, оскільки існує необхідність вибору оптимального методу та параметрів для реалізації аналізу даних.

Мета аналізу даних, як науки, виділити корисну інформацію, зробити по ній висновки та прийняти обґрунтовані рішення. Аналіз даних може реалізовуватись за допомогою різноманітних інструментів – математики, статистики, різних режимів візуалізації. Головна ціль – це зменшення обсягу даних та їх узагальнення [13].

Основною складністю, на яку можна наткнутись в процесі аналізу даних, є об'єм цих самих даних, адже масиви в основному містять сотні чи тисячі записів, а в окремих випадках це число сягає сотень тисяч. Очевидно, що людина

неозброєним оком не в змозі проаналізувати такий обсяг даних та тим паче дійти певної логічної закономірності між ними. Саме тому необхідно застосовувати методи та засоби стиснення даних.

Аналіз даних виконується з кількох послідовних кроків, а саме: збір даних, їх обробка, вивчення, сам аналіз та узагальнення для підбиття підсумків (рис. 1.15).

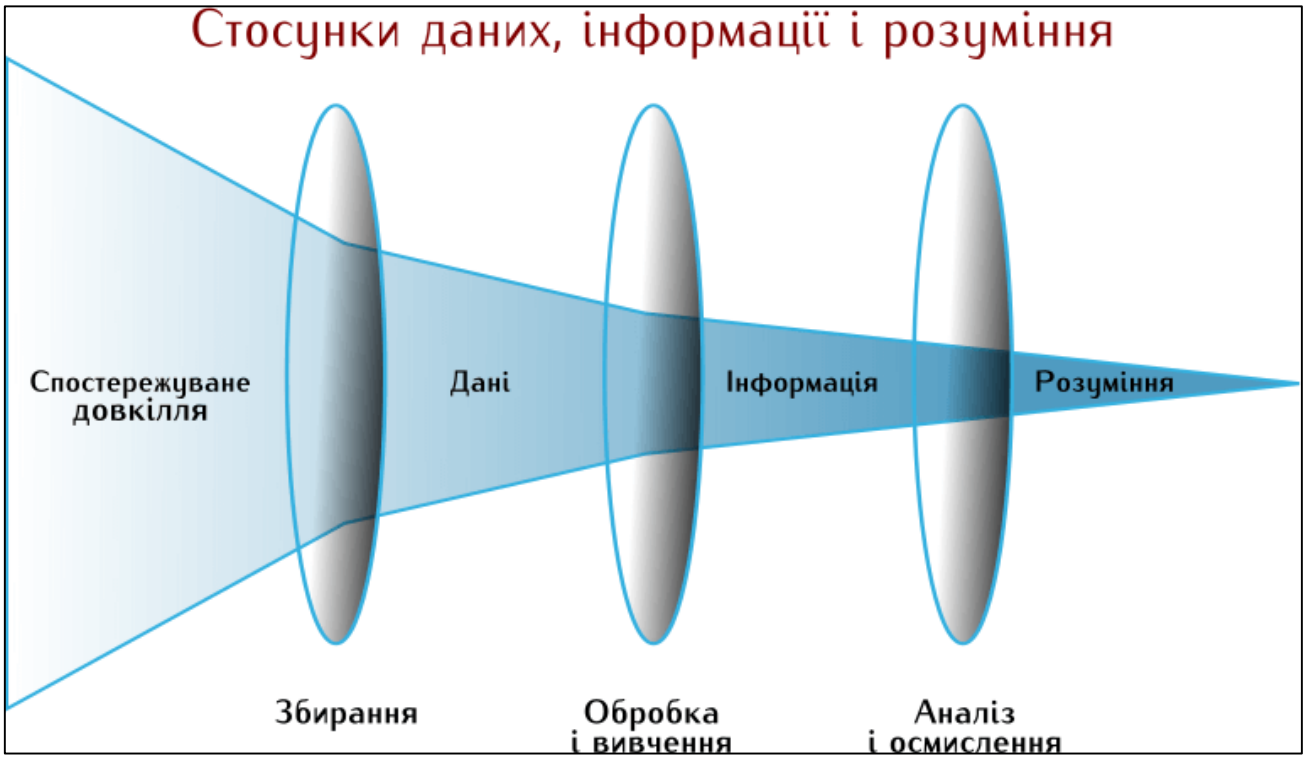


Рисунок 1.15 – Кроки виконання аналізу даних

Статистика — наука збору, представлення, аналізу і розумної інтерпретації даних. Статистика являє собою строгий науковий метод, що дозволяє зрозуміти дані, дійти до їх суті. Окрім узагальнення даних, важливим завданням статистики є формулювання логічних висновків та передбачення залежностей між змінними [14].

Кроки виконання статистичного аналізу зображено на рисунку 1.16.

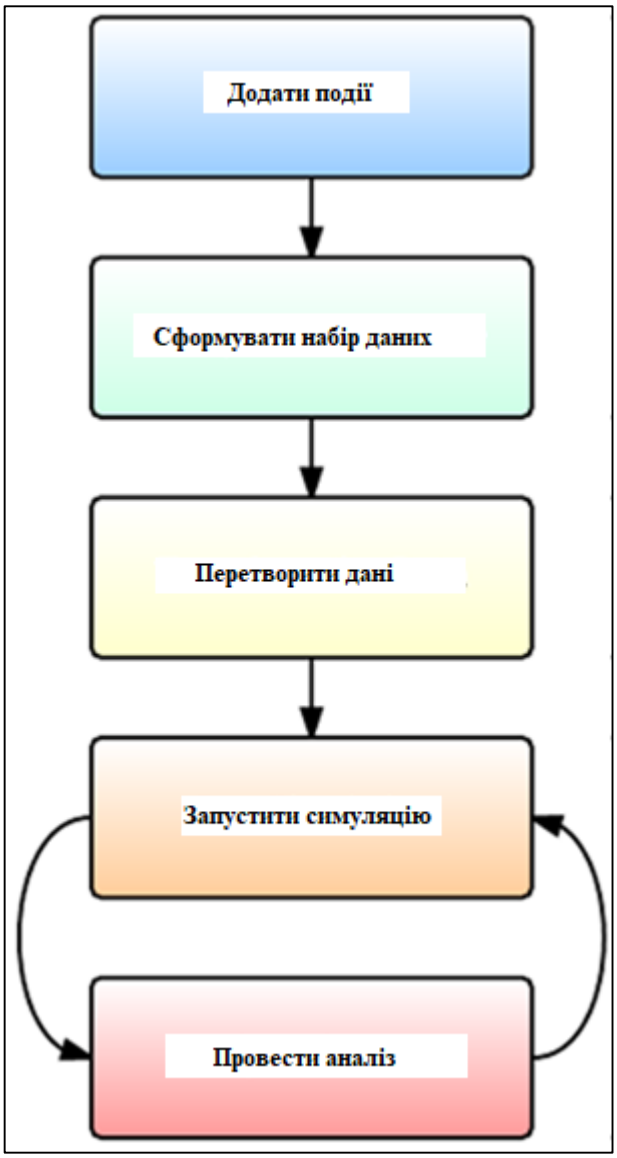


Рисунок 1.16 – Загальний набір кроків статистичного аналізу

1.4 Висновки

В даному розділі було здійснено огляд та аналіз історії розвитку прогнозування погодних умов з часів народних прикмет до заснування перших математично обґрунтованих алгоритмів та створення спеціалізованої обчислювальної машини, прикладів відкритих даних на прикладі датасетів системи Kaggle та галузі аналізу даних. В тому число було сформульовано проблему даної галузі в Україні.

Також було наведено стислий огляд основних технологій як синоптичних, так і комп’ютерних, що застосовуються фахівцями даної галузі, відкритих

наборів погодних даних та принцип роботи аналізу даних. З огляду на вище викладену інформацію, було зроблено висновок, що існуючі методи та технології передбачення спираються на прогнозування повного переліку погодних умов, є часо-, трудо- та ціномісткими, тому доцільно розробити технологію, яка б передбачала наявність опадів в тому чи іншому місці без конкретної їх кількості, адже користувачів порталів прогнозу погоди здебільшого цікавить питання «Будуть сьогодні взагалі якісь опади чи ні?», а вже їх кількість не є надто критичною, тим паче що жодна технологія не зможе дати вам впевненість в кількості опадів саме у вашому поточному місці знаходження через обмеження масштабованість.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ ОПАДІВ

2.1 Огляд вхідних даних

Для оптимального вибору засобів реалізації технології необхідно зробити огляд вхідних даних.

Вхідними даними для виконання магістерської роботи є набір даних зі статистикою щоденних погодних умов протягом останніх 10 років в місті Вінниця, отриманих від Вінницького обласного центру з гідрометеорології. Він ґрунтується на показниках умов, на основі яких можливо визначити ймовірність опадів [4].

Отримані дані представлено у форматі окремих .xls файлів для кожного місяця з січня 2010 по вересень 2020 років. На рисунку 2.1 зображено вигляд такої таблиці як приклад.

ДЕРЖАВНА ГІДРОМЕТЕОРОЛОГІЧНА СЛУЖБА УКРАЇНИ — ТСГ-1

ТАБЛИЦЯ

метеорологічних і агрометеорологічних спостережень

Вінницька державна с/г дослідна станція інституту кормів

Станція: ЦГМ Вінниця; Область (Республіка): Вінницька; Район: Вінницький; Рік: 2010; Місяць: січень; Декада: перша

Число місяця	Температура, °С: повітря середня	повітря максимальна	повітря мінімальна	на грунту (снігу) максим.	на грунту (снігу) мінім.	грунту середня за добу на глибині, см: мінімальна на висоті 2 см	(на мет.майданчику) 5	10	15	20	40	Вологість повітря відносна,%: середня	мінімальна	дефіцит, середній	дефіцит, максимальний	Тривалість сонячного сяйва (годин)	Хмарність, бал: загальна	нижня	Максимальна швидкість вітру, м/с	Опади, мм (сума) за: день	вечір	доба	Тривалість роси (утруднинь), годин	Атмосферні явища (вид, інтенсивність, тривалість)
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	-3.1	-2.4	-4.0	-1.5	-2.4							96	93	0.2	0.4	0.0	10	10	10	8.1	1.9	10.0		□18-6.18°-11.50-18; □18-18; □18-19.22-1.03°; □5.30-6.18, 11.50-15.30; □11-11.50-12.48°-13.09-13.22°-14-18°; □16.04-17.20.
2	-5.5	-2.9	-6.9	-1.6	-7.3							87	80	0.5	0.8	0.0	10	9	12		2.1	2.1		□°18-4.50; □18-18; □°18-18.10, 18.40-1.45; □5-18.
3	-10.2	-5.4	-14.7	-5.7	-16.2							79	74	0.6	0.8	0.0	8	2	10	0.2		0.2		□18-16.20; □18-21.40; □22.30-9.15; □°14.46-18.
4	-8.6	-7.1	-9.6	-4.9	-10.6							85	76	0.5	0.8	0.0	10	9	5	0.4	0.6	1.0		□°18-21.40, 7.20-9.17, 12.35-15.21, 15.48-18; □°21.40-2.35, 4.42-7.20.
5	-8.2	-6.7	-9.8	-6.2	-10.5							83	75	0.5	0.8	0.7	10	9	7	0.0	0.0	0.0		□°18-18.35; □°4.20-5.23, 6.10-7.15; □°5.40-8.25, 10.45-18.
6	-2.8	-0.4	-6.7	-0.7	-6.5							95	92	0.3	0.4	0.0	10	10	17	1.9	0.5	2.4		□°18-18; □20.10-16.48; □3.15-4.38, 5.16-6.24-7.54°-14.25-16.15°; □°16.48-18; □16.48-18.
7	0.5	1.2	-0.4	0.1	-0.7							98	96	0.1	0.3	0.0	10	10	13		2.6	2.6		□18-23.40°-3.20,4.25-5.00-7.44°-7.55,9.57-11.26,12.12-12.36-18°; □°3.20-4.25,7.55-9.57,11.26-12.12;□°18-18.25,3.25-5.42,□18-18.25.
8	-0.4	0.2	-1.2	0.0	-1.1							95	88	0.3	0.7	0.0	10	10	11					□18-18.30°-19.07, 4.38-4.57-16.54°-18; □19.07-19.45°-3.40-4.38°; □21.20-10.22.
9	0.9	2.7	-1.1	0.0	-0.8							100	100	0.0	0.0	0.0	10	10	11	0.1	0.1	0.2		□18-18.15; □18.15-7.05; □°21.10-10.20; □18.15-13.16°-14.24-18°.
10	1.4	2.7	0.8	0.1	-0.1							100	100	0.0	0.0	0.0	10	10	10	0.3	0.7	1.0		□18-18.15°-18; □0.05-6.40, 9.40-18
за декаду	-3.6	2.7	-14.7	0.1	-16.2							92	75	0.3	0.8	0.7	10	9	17	11.0	8.5	19.5		
за місяць																								

Вид спостереження	Вологість грунту, бали	Температура грунту, °С	Температура грунту на глибині вузла кущіння,°С	Температура грунту на глибині вузла (грунтова кірка, мм)	Глибина промерзання грунту, см	Сніговий покрив покриття, бал	Сніговий покрив місцевості, бал	Сніговий покрив висота по	Стійкий сніговий покрив, см
Прилад	Візуально	АМ - 6	КН	АМ - 29А	АМ - 17	Мерзлотомір	Мерзлотомір		
№ ділянки			МП	43	43				
Культура			Оз.пшениця	Оз.пшениця		Оз.пшениця			
Дата	на глибині, см: 0-2	10-12	5	10	мін.(товщина)	сніговлоковерив,см	середня (товщ.)	сніговлоковерив,см	макс.	мінімальна	строкова	сніговлоковерив,см	промерзання, см	відтавання, см	сніговлоковерив,см	промерзання, см	відтавання, см	сніговлоковерив, см			
1																			10	15	
2																			10	21	

Кількість днів за декаду із: градом	хуртовиною (рясою)	туманом	вітром із швидкістю ≥10	вітром із швидкістю ≥15	низовою бурею	відлигою	відносною вологістю повітря ≤30%	опадами ≥1мм	опадами ≥5мм
0	0	4	8	1	0	4	0	6	1

Декадна телеграма

33562 10011 111 90002 2003/ 3115/ 4116/ 50206 60118 7171/ 222 92006 10434 95006 14534 20204 32100 40154

56101 6004/ =

Розрахункова критична температура вимерзання -15.8°

Агрометеорологічний огляд

В першій декаді лютого утримувалася контрастна погода.

Прохолодна погода першої п'ятиденки декади змінилась теплою.

Рисунок 2.1 – Приклад таблиці початкових даних (січень 2010 р.)

Виходячи з попереднього розділу аналізу предметної області, було виявлено, що таблиця містить надлишкові дані, які не застосовуються для прогнозування наявності опадів так як не мають на це суттєвого впливу [16]. Тому такі показники було відсіяно в цілях підвищення продуктивності.

Для подальшої роботи необхідно згрупувати всі дані параметрів, що мають прямий вплив на прогнозування наявності опадів, у єдиний файл.

Таким чином набір даних було створено у вигляді файлу формату .csv та завантажено як приватний датасет на платформу для машинного навчання Kaggle від компанії Google. Вигляд датасету обох форматів зображено на рисунках 2.2-2.3.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	date	avgTemp	maxTemp	minTemp	avgHumidity	minHumidity	avgHumidityDef	maxHumidityDef	cloud	lowCloud	maxWind	precipitation	
2	1/1/2010	-3.1	-2.4	-4	96	93	0.2	0.4	10	10	10	10	
3	1/2/2010	-5.5	-2.9	-6.9	87	80	0.5	0.8	10	9	12	2.1	
4	1/3/2010	-10.2	-5.4	-14.7	79	74	0.6	0.8	8	2	10	0.2	
5	1/4/2010	-8.6	-7.1	-9.6	85	76	0.5	0.8	10	9	5	1	
6	1/5/2010	-8.2	-6.7	-9.8	83	75	0.5	0.8	10	9	7	0	
7	1/6/2010	-2.8	-0.4	-6.7	95	92	0.3	0.4	10	10	17	2.4	
8	1/7/2010	0.5	1.2	-0.4	98	96	0.1	0.3	10	10	13	2.6	
9	1/8/2010	-0.4	0.2	-1.2	95	88	0.3	0.7	10	10	11	0	
10	1/9/2010	0.9	2.7	-1.1	100	100	0	0	10	10	11	0.2	
11	1/10/2010	1.4	2.7	0.8	100	100	0	0	10	10	10	1	
12	1/11/2010	0.4	0.9	-0.2	97	94	0.2	0.4	10	10	9	16.2	
13	1/12/2010	-2.2	-0.1	-4.5	92	89	0.4	0.5	10	10	7	4.5	
14	1/13/2010	-5.5	-4.5	-6	90	85	0.4	0.6	10	10	7	1.3	
15	1/14/2010	-6	-5.1	-6.7	93	91	0.3	0.3	10	10	7	1.7	
16	1/15/2010	-6.4	-5.3	-7.8	91	89	0.3	0.4	10	10	9	3.4	
17	1/16/2010	-9.7	-10.1	-11.6	86	80	0.4	0.5	10	10	9	0.2	
18	1/17/2010	-8	-5.7	-10.6	89	86	0.4	0.5	10	9	11	0	
19	1/18/2010	-11.3	-7.7	-12.9	84	76	0.4	0.6	10	10	11	0.7	
20	1/19/2010	-10.7	-9.7	-12.3	79	70	0.6	0.9	10	10	10	4	
21	1/20/2010	-10.9	-10.4	-12	67	52	0.9	1.3	10	10	11	0.3	
22	1/21/2010	-11.4	-10.4	-12.6	81	75	0.5	0.6	10	10	10	5.3	
23	1/22/2010	-14	-11.8	-16.9	66	49	0.7	0.9	9	6	10	1.6	
24	1/23/2010	-21.3	-16.8	-25.6	67	57	0.4	0.6	1	0	8	0	
25	1/24/2010	-23.1	-18.5	-27.5	76	70	0.2	0.4	1	0	7	0	
26	1/25/2010	-22.2	-16.4	-27	73	64	0.3	0.6	3	0	6	0	
27	1/26/2010	-18.9	-12.6	-23.6	71	50	0.5	1.2	5	0	5	0	
28	1/27/2010	-17.5	-13.8	-20.5	59	50	0.7	1	8	0	4	0	
29	1/28/2010	-13.4	-7.6	-20.2	85	64	0.3	0.5	8	4	11	0.7	
30	1/29/2010	-10.1	-6.8	-12	87	73	0.4	0.8	7	3	8	0	
31	1/30/2010	-9.4	-5.3	-12.8	81	67	0.6	1.3	6	1	10	0	
32	1/31/2010	-1.6	0.6	-7.4	99	93	0.1	0.3	10	10	10	0	
33	2/1/2010	-1.8	0.5	-4.8	88	75	0.6	1.2	10	8	11	6	
34	2/2/2010	-5.5	-2.8	-8.5	82	76	0.7	1.1	10	6	9	0.3	
35	2/3/2010	-8.1	-3.7	-11.5	92	80	0.3	0.7	7	2	6	0	
36	2/4/2010	-9.4	-4.9	-12.5	94	78	0.2	0.9	8	7	6	0	
37	2/5/2010	-8.2	-7.3	-11.4	92	84	0.3	0.5	9	7	7	0	
38	2/6/2010	-9.6	-7.9	-11.4	68	53	0.9	1.3	10	5	11	0.2	

weatherSet

Рисунок 2.2 – Файл .csv з об’єднаними даними отриманих таблиць

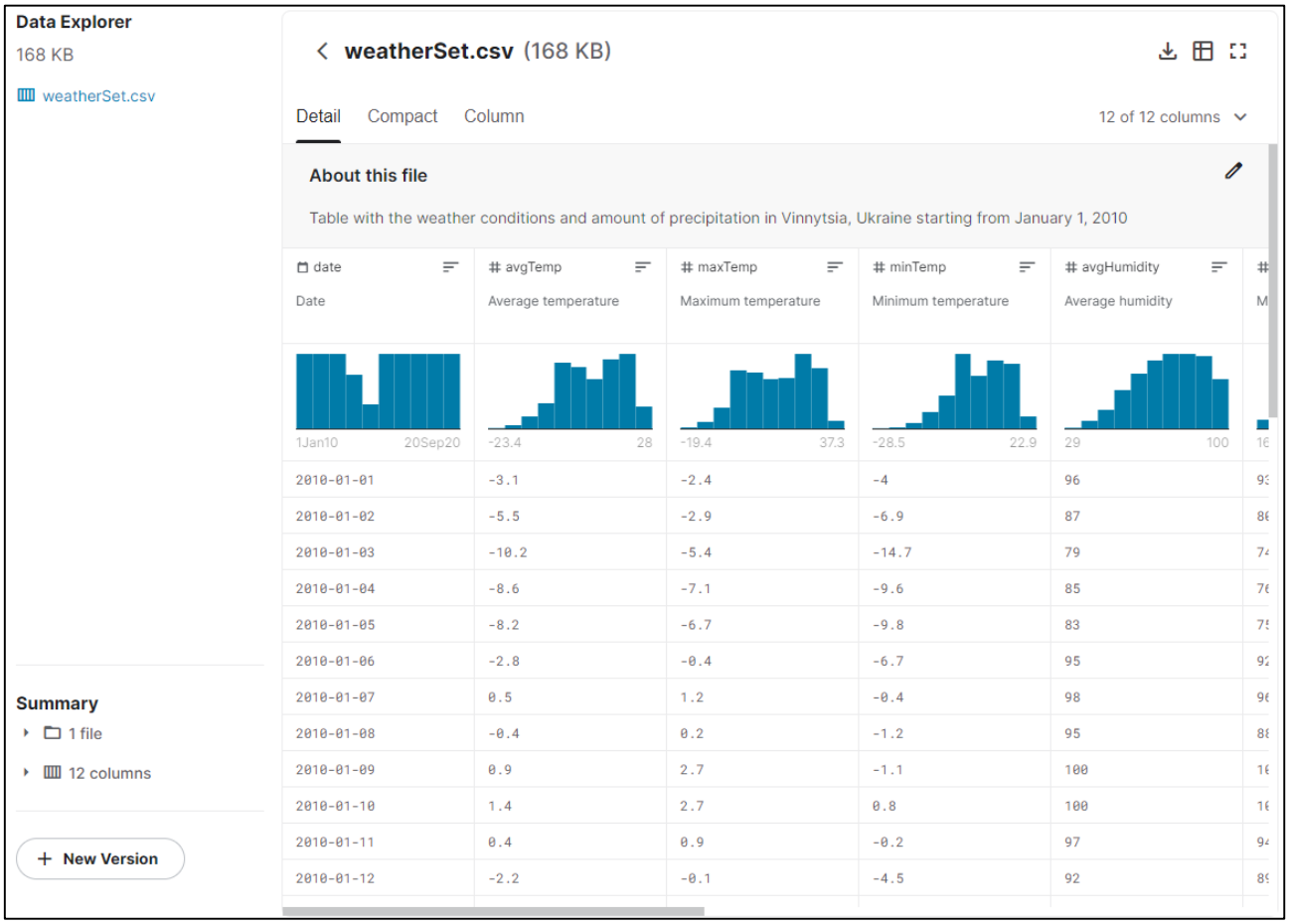


Рисунок 2.3 – Вигляд датасету на платформі Kaggle

Розглянемо атрибути даного набору для розуміння загальної картини алгоритму роботи майбутньої технології аналізу та прогнозування. Набір містить наступні атрибути:

- date – дата спостереження;
- avgTemp – середня температура повітря в градусах Цельсія;
- maxTemp – максимальна температура повітря в градусах Цельсія;
- minTemp – мінімальна температура повітря в градусах Цельсія;
- avgHumidity – середня вологість повітря;
- minHumidity – мінімальна вологість повітря;
- avgHumidityDef – середній дефіцит вологості повітря;
- maxHumidityDef – максимальний дефіцит вологості повітря;
- cloud – оцінка загальної хмарності по шкалі від 0 до 10;

– lowCloud – оцінка хмарності нижнього ярусу по шкалі від 0 до 10;
– maxWind – максимальна швидкість вітру в м/с;
– precipitation – кількість опадів за добу в мм.

2.2 Огляд методу вирішення поставленої задачі

Машинне навчання – це напрям у галузі комп’ютерних наук, що спеціалізується на методах навчання штучних комп’ютерних систем на основі даних, що поміщають в систему [17]. На основі машинного навчання наразі побудовано набагато більше систем, ніж ми здогадуємось, адже ми користуємося ними щодня, від веб-пошуку до фільтрації дописів на різних сайтах та в соцмережах.

Виділення ознак генерується шляхом перетворення вихідних "необроблених" даних (наприклад, значення пікселів у зображення) у послідовне зображення (вектор), з якого система дослідження (класифікатор) може переглядати та виділяти певні образ . Такі методи машинного навчання обмежені можливістю обробляти природні дані в їх первісному вигляді [17]. Існуючі типи класифікаторів систематизовано за різними критеріями і їх короткі характеристики наведено рисунку 2.4.

Критерій	Тип	Короткий опис
Використання або невикористання навчальних даних	Класифікація з учителем	За підходами до класифікації з учителем вхідні дані поділяють, використовуючи набір зразків як навчальні дані
	Класифікація без учителя	Підходи до класифікації без учителя, відомі як кластеризація, не беруть до уваги мітки навчальних даних для класифікації вхідних даних
Урахування або неврахування будь-якого припущення про розподіл вихідних даних	Напівавтоматичне навчання	Навчання в напівавтоматичних підходах відбувається з використанням даних як з мітками, так і без них
	Параметричні класифікатори	Параметричні класифікатори ґрунтуються на припущенні, що функція щільності ймовірності для кожного класу відома
	Непараметричні класифікатори	Непараметричні класифікатори не обмежуються жодними припущеннями про розподіл вхідних даних
Розгляд одного класифікатора або ансамблю	Один класифікатор	Використовується єдиний класифікатор для призначення мітки для об'єкта
	Ансамбль класифікаторів	Під час визначення мітки для об'єкта враховуються результати кількох (ансамблю) класифікаторів
Використання або невикористання технології жорсткого поділу, де кожен об'єкт належить лише одному кластеру	Жорсткий класифікатор	Технології жорсткої класифікації не враховують подальші зміни різних класів
	М'який (нечіткий) класифікатор	Нечіткі класифікатори моделюють поступові граничні зміни, забезпечуючи оцінку ступеня подібності всіх класів
Видача класифікатором розподілу ймовірності належності до всіх класів	Імовірнісний класифікатор	Класифікатор здатен для заданого зразка оцінити розподіл імовірності на множині класів
	Неймовірнісний класифікатор	Підхід визначає лише найбільш придатний клас для вхідного образу

Рисунок 2.4 – Різновидності підходів до класифікації залежно від критеріїв

Найбільш узагальнений поділ методів машинного навчання визначає категорії навчання з учителем або без, supervised learning та unsupervised learning відповідно.

Методи supervised learning класифікують та розбивають вхідні дані у класи. Щоб навчити класифікатор даного типу, необхідно робити маркування різних видів даних у вибірці. В свою чергу методи unsupervised learning не проводять кроків класифікації, вони не потребують маркованих даних для навчання, працюють за принципом віднаходження закономірностей щодо змін у вхідних даних і створюють відповідні кластери (кожному кластеру відповідає певна закономірність змін).

Оскільки завданням роботи є аналіз методів класифікації, то надалі розглядатимемо лише методи навчання з учителем. Класифікатори бувають параметричними – працюють на основі припущень, наприклад метод максимальної правдоподібності (maximum likelihood), та непараметричні – є більш популярними у використанні завдяки можливості визначати належність вхідних даних до того чи іншого класу та ймовірності цієї належності, не спираючись на припущення, наприклад, логістична регресія (logistic regression).

Дерево класифікації методів машинного навчання зображено на рисунку 2.5.

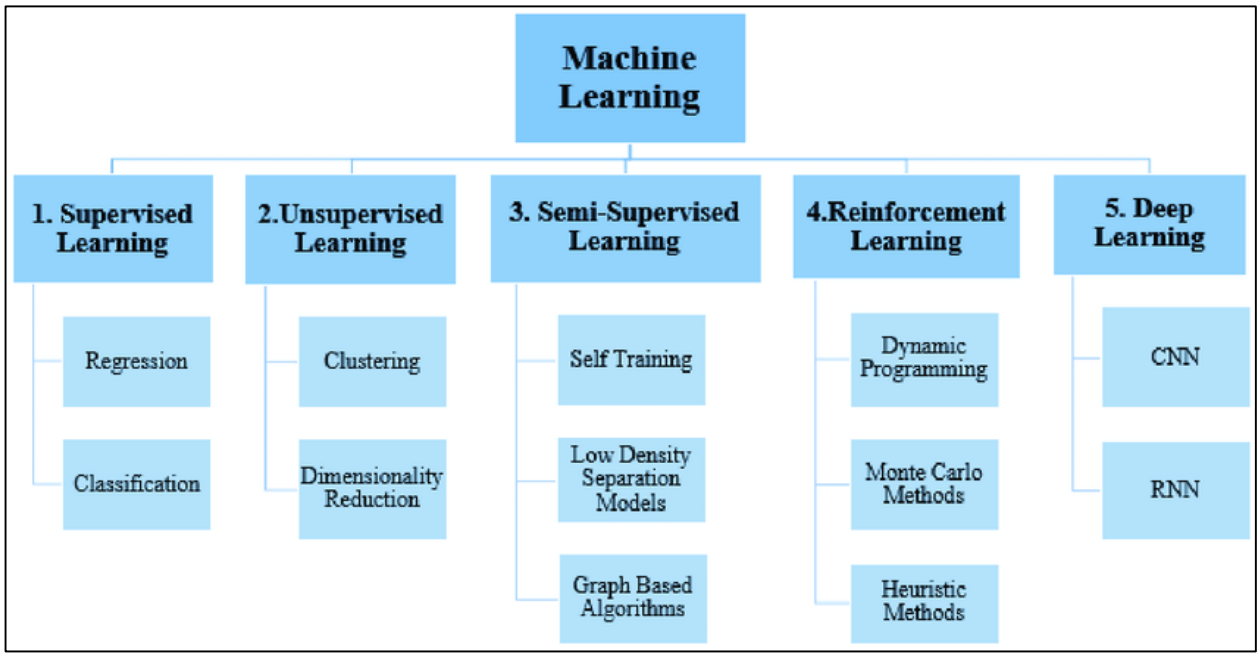


Рисунок 2.5 – Види машинного навчання

2.3 Вибір мови програмування

Спираючись на рейтинг найпопулярніших мов програмування сервісу GitHub, що використовуються в ноутбуках системи, було вирішено провести аналіз для вибору найоптимальнішого шляху реалізації технології передбачення опадів. Рейтинг складався на основі кількості репозиторіїв з використанням ML, створених на вказаному сервісі. Отже, серед лідерів Python та C++. Проведемо короткий огляд для наочного порівняння.

Висока популярність у використанні в сфері ML дісталась C++ полягає у створенні компанією Google бібліотеки TensorFlow , що зосереджена під розробку нейромереж. C++ надають перевагу при необхідності використовувати навчальну модель в мобільних додатках або роботах. Крім того, популярність C++ обумовлена розвитком розподіленої високопродуктивної платформи для градієнтного бустінга Microsoft LightGBM (підвищує швидкість і ефективність навчання ML-моделі) і бібліотеки Turi Create (спрощує розробку користувача моделей машинного навчання для початківців розробників).

Найбільш вживаною мовою програмування серед розробників Data science залишається Python, популярність якій переважно додають великий набір інструментів та методів реалізації ML, а також масштабована база бібліотек, заточених під даний напрям. Одними з кращих бібліотек даної мови програмування стали Scikit-Learn, в якій зосереджено чималу кількість алгоритмів, та ChatterBot, що застосовується для обробки мови і навчається за допомогою діалогів.

Так як Python здобув високий рівень довіри у даній сфері та має високу продуктивність, розглянемо його переваги щодо реалізації технології передбачення, аби пересвідчитись у правильності вибору мови програмування в його користь (рис. 2.6).

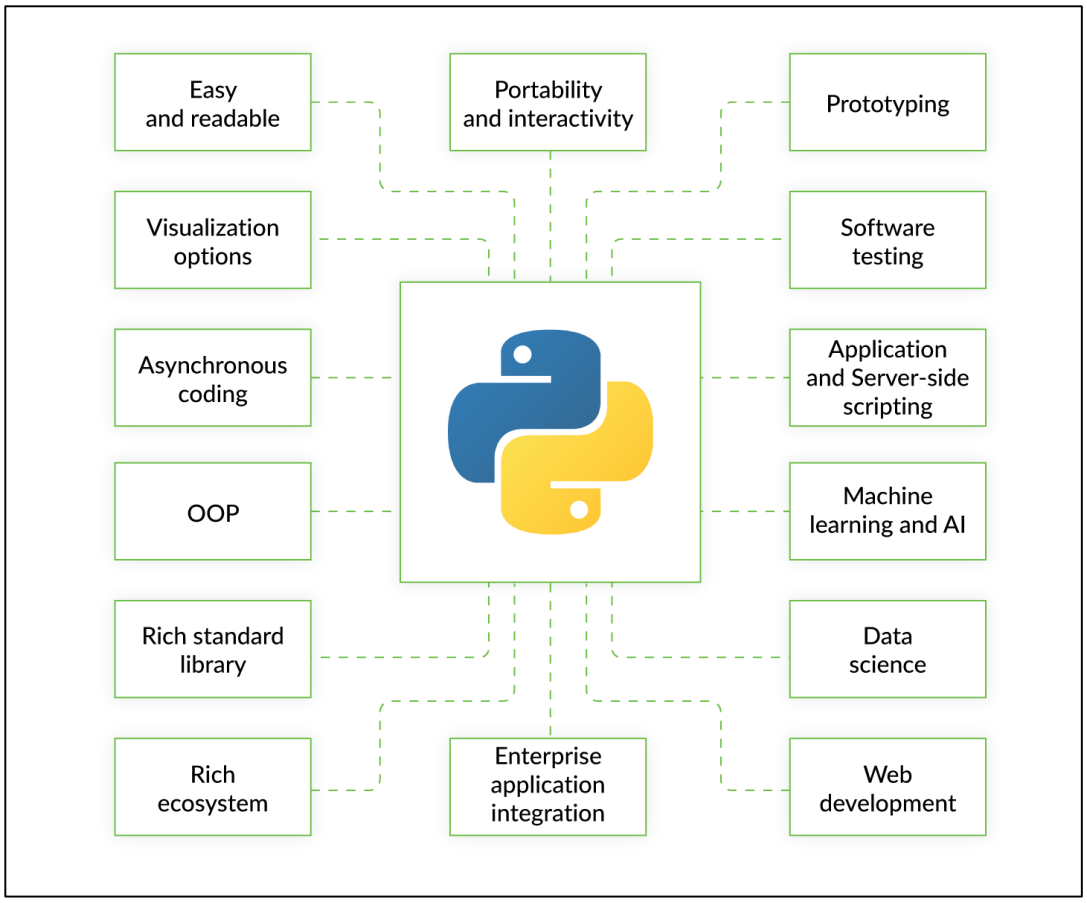


Рисунок 2.6 – Основні переваги використання мови Python

Однією з головних переваг мови Python при виборі для початківців є зрозумілість та простота вивчення. Python високорівневий та зручний у роботі, лаконічний та зручний в читанні готового коду, що спрощує розробку методів ML так як самі алгоритми досить складні для розуміння. Також Python має досить потужну базу якісної документації та підтримку. Гнучкість цієї мови програмування дає можливість вибору між об'єктно-орієнтованими підходами та скриптами, він допомагає об'єднати різноманітні типи даних.

Зважаючи на переваги використання Python в галузі машинного навчання, було вирішено реалізувати технологію аналізу та передбачення опадів за допомогою саме цієї мови програмування.

Також Python став найбільш затребуваною мовою програмування серед вакансій для Data Science спеціалістів (рис. 2.7).

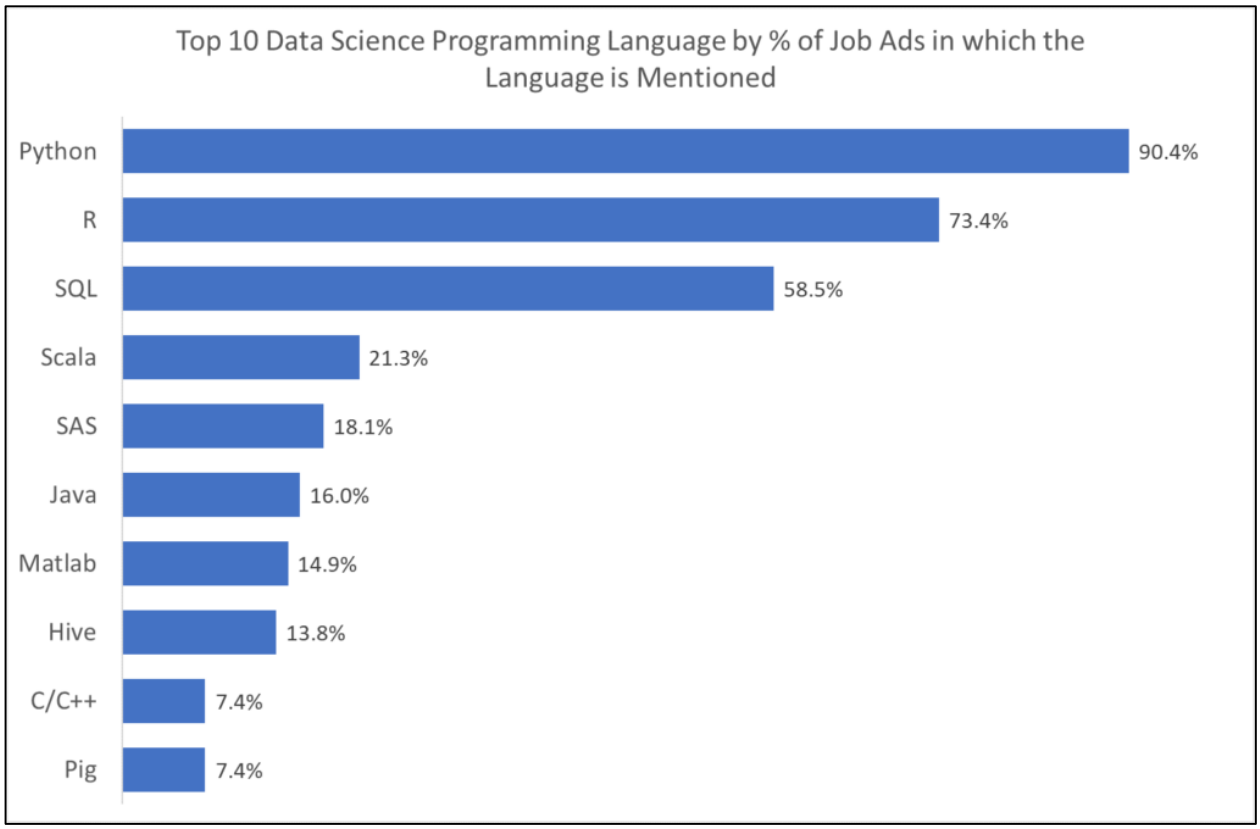


Рисунок 2.7 – Рейтинг мов програмування вакансій Data Science серед роботодавців

Коротко представимо основну інформацію про Python для загального ознайомлення.

Python - це універсальна сучасна мова програмування високого рівня, до переваг якої відносять високу продуктивність програмних рішень і структурований код, що добре читається [18].

Один з найпоширеніших у використанні інтерпретаторів CPython написано на мові C. Наразі Python розвивається великими кроками, кожних кілька років виходять оновлення, підтримує усі найпопулярніші операційні системи, як на кишенькових, так і на стаціонарних комп'ютерах. Застарілі системи автоматично виключаються з ядра системи, для того щоб працювати з ними необхідно буде відмовитись від переваг нових версій Python.

Основною фішкою щодо форматування тексту коду для кращого розуміння та візуального сприйняття є розбиття його на блоки за допогомою відступів, використовуючи клавіші «Space» і «Tab». Синтаксис мови довзволяє

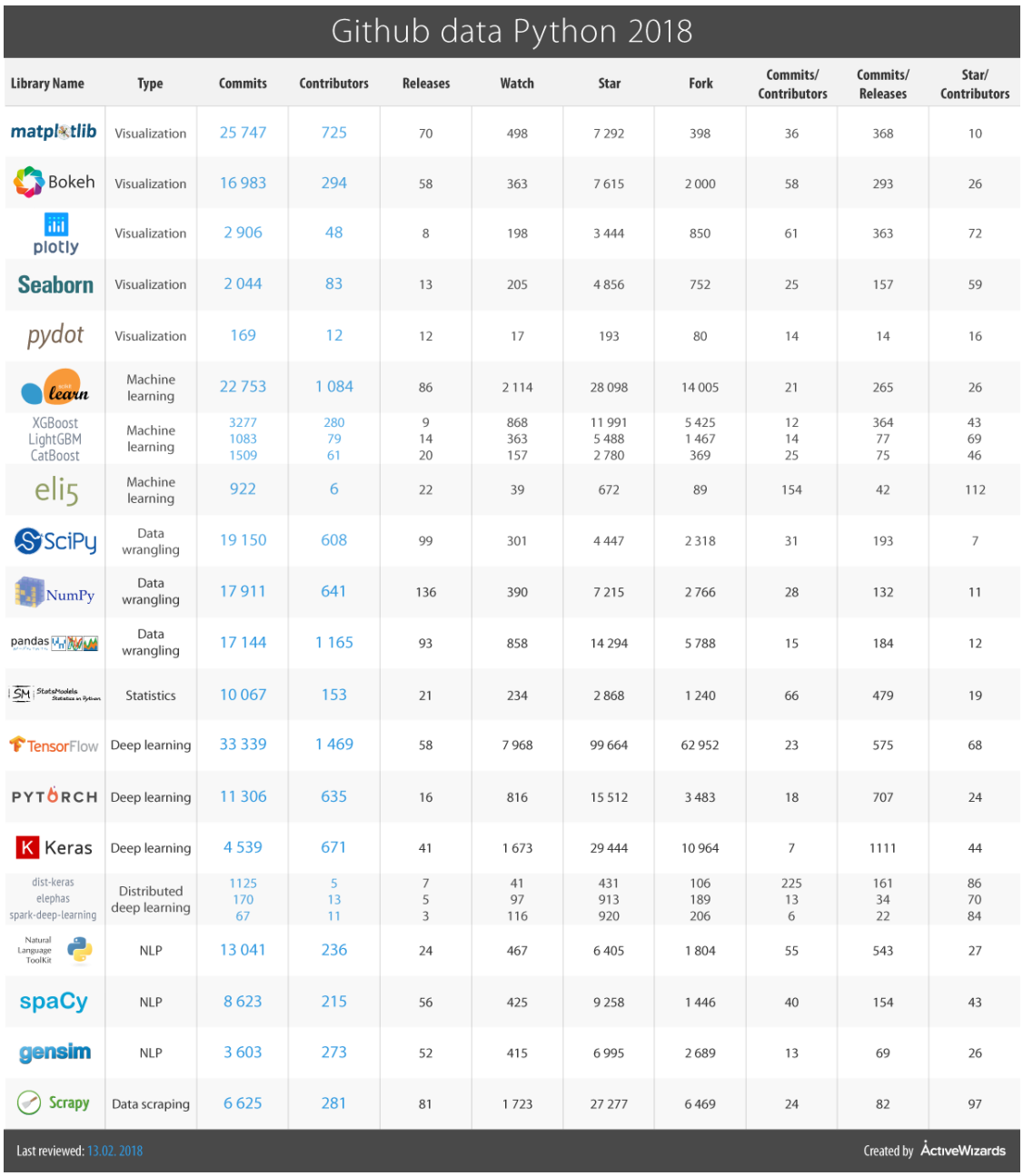
досить помітно скорочувати кількість рядків коду в об’ємних програмах за рахунок відсутності використання різного виду дужок для позначення початку/кінця блоку.

Python популярна та широко застосовувана мова у багатьох сферах людською діяльності та комп’ютерних досліджень. Вона допомагає обслуговувати хмарні сервіси та сховища даних.

2.4 Огляд основних бібліотек для машинного навчання

Основою на якій стоїть вся галузь машинного навчання є математика. В бібліотеках та класифікаторах здебільшого використовуються теорія ймовірностей, математична оптимізація та статистика. Саме тому розробляти нові методи та алгоритми у сфері машинного навчання можуть вчені, що не мають великих здібностей та знань в програмуванні.

На рисунку 2.8 зображено найпопулярніші бібліотеки за версією сервісу Github 2018 року.



Github data Python 2018

Library Name	Type	Commits	Contributors	Releases	Watch	Star	Fork	Commits/ Contributors	Commits/ Releases	Star/ Contributors
matplotlib	Visualization	25 747	725	70	498	7 292	398	36	368	10
Bokeh	Visualization	16 983	294	58	363	7 615	2 000	58	293	26
plotly	Visualization	2 906	48	8	198	3 444	850	61	363	72
Seaborn	Visualization	2 044	83	13	205	4 856	752	25	157	59
pydot	Visualization	169	12	12	17	193	80	14	14	16
learn	Machine learning	22 753	1 084	86	2 114	28 098	14 005	21	265	26
XGBoost LightGBM CatBoost	Machine learning	3277 1083 1509	280 79 61	9 14 20	868 363 157	11 991 5 488 2 780	5 425 1 467 369	12 14 25	364 77 75	43 69 46
eli5	Machine learning	922	6	22	39	672	89	154	42	112
SciPy	Data wrangling	19 150	608	99	301	4 447	2 318	31	193	7
NumPy	Data wrangling	17 911	641	136	390	7 215	2 766	28	132	11
pandas	Data wrangling	17 144	1 165	93	858	14 294	5 788	15	184	12
StatsModels	Statistics	10 067	153	21	234	2 868	1 240	66	479	19
TensorFlow	Deep learning	33 339	1 469	58	7 968	99 664	62 952	23	575	68
PYTORCH	Deep learning	11 306	635	16	816	15 512	3 483	18	707	24
Keras	Deep learning	4 539	671	41	1 673	29 444	10 964	7	1111	44
dist-keras elephas spark-deep-learning	Distributed deep learning	1125 170 67	5 13 11	7 5 3	41 97 116	431 913 920	106 189 206	225 13 6	161 34 22	86 70 84
Natural Language ToolKit	NLP	13 041	236	24	467	6 405	1 804	55	543	27
spaCy	NLP	8 623	215	56	425	9 258	1 446	40	154	43
gensim	NLP	3 603	273	52	415	6 995	2 689	13	69	26
Scrapy	Data scraping	6 625	281	81	1 723	27 277	6 469	24	82	97

Last reviewed: 13.02. 2018 Created by ActiveWizards

Рисунок 2.8 – Найпопулярніші бібліотеки Python для ML

Розглянемо кілька вагомих бібліотек різних типів, що найчастіше використовуються у Machine Learning [19].

NumPy – одна з найбільш застосовуваних бібліотек Python для роботи з великим обсягом даних (матриці або масиви), що працює на основі великого набору математичних функцій. Бібліотека підходить для виконання базових прикладних обчислень, таких як випадкові числа, регресія, ряди Фур'є або мінімізація. Саме тому являється основним інструментом для виконання етапу

аналізу даних. NumPy закладена в основу багатьох інших бібліотек як основний інструмент обробки та аналізу даних.

Scikit-learn – бібліотека створена на основі двох інших: NumPy та SciPy. Зручна у використанні та зрозуміла бібліотека, що допоможе перетворити дані або вибрати функцію всього за допомогою кількох рядків коду. Містить широкий вибір алгоритмів, як контрольованих, так і неконтрольованих, для роботи з класифікацією, кластеризацією та регресією. Великим плюсом є підтримка майже всіх існуючих алгоритмів даною бібліотекою.

Pandas є однією з кращих високорівневих бібліотек для роботи зі структурою даних, дозволяє згрупувати, відфільтрувати, скомбінувати дані таким чином, щоб вони були максимально зрозумілими для сприйняття, а відповідно використання та аналізу. Однією з переваг бібіліотеки є змога працювати з часовими рядами та легко маніпулювати їх вмістом. Легко працює з даними оформленими в різноманітних форматах, таких як: CSV-, Excel-, JSON-файли або базами даних SQL.

Бібліотекою для побудови графіків, якою користується що не кожен, хто працює з обробкою даних, є стандартна бібліотека Python Matplotlib. Вона вимагає більш об’ємного формулювання для створення графіків чи фігур, які гарно виглядають та максимально точно передають дані, над якими проводився аналіз, адже бібліотека є досить низькорівневою. Проте, не зважаючи на такий здавалося б недолік, водночас вона досить гнучка завдяки відсутності великої кількості чітко встановлених функції та обмежень, що дозволяє реалізувати майже всі можливі графіки. Matplotlib дає широкий вибір у типі графіків, за її допомогою можна побудувати гістограму, різноманітні діаграми та реалізувати графік в площині недекартових координат. Підтримує графічний інтерфейс користувача всіх операційних систем та дозволяє експортувати графіки у різноформатні файли (SVG, JPG, PNG, GIF, PDF).

LightGBM – фреймворк для реалізації градієнтного бустінга, в основі якого лежить алгоритм дерева рішень. Головним принципом роботи є розподіл листя дерева по найкращій відповідності, в той час як інші алгоритми працюють по

методу поділу в глибину або рівню дерева. Завдяки такому виду розподілу зменшуються втрати і це приводить до більш точного результату, ніж будь-яким іншим алгоритмом. Головними переваги, що впливають на вибір даної бібліотеки для реалізації методів ML, є швидкість навчання моделі та її висока ефективність, низька затратність пам’яті, висока точність та сумісність з великим обсягом даних.

Seaborn – це високорівнева бібліотека візуалізації графіків на базі іншої бібліотеки Matplotlib. З її допомогою можна швидше та простіше згенерувати графіки, побудовані низькорівневою бібліотекою, адже вона містить більш зрозумілі вбудовані налаштування для оформлення графіків. Також Seaborn дозволяє скоротити кількість коду для реалізації більш складних типів візуалізації на відміну від Matplotlib. Узагальнивши, можна дійти висновку що дана бібліотека є деяким API на базі іншої.

Для зручної роботи з датасетом будуть використовуватись всі вище наведені бібліотеки, виходячи зі структури даних та можливостей вище описаних фреймворків це дозволить підняти продуктивність майбутньої моделі.

2.5 Висновки

В даному розділі було проведено огляд вхідних даних, їх аналіз та розроблено об’єднаний файл для подальшого використання, на основі проведеного групування всіх вагомих параметрів, для подальшої роботи дані було завантажено в платформу Kaggle, також було наведено список атрибутів датасету для подальшої розробки методики реалізації інформаційної технології.

Наведено короткий опис сфери машинного навчання для більш глибокого розуміння галузі, в якій буде проводитись реалізація нової технології передбачення.

На основі огляду мови програмування та бібліотек ML було розроблено методику реалізації технології аналізу та передбачення опадів за допомогою інструментів, які найбільше підходять для рішення поставленої задачі.

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ ОПАДІВ

3.1 Аналіз та конструювання головних ознак

Спочатку необхідно підключити до нашого коду дві основні бібліотеки, які будуть в подальшому неодноразово використовуватися: numpy та pandas. Їх опис було проведено у підрозділі 2.4. Код для підключення бібліотек наведено на рисунку 3.1.

```
import numpy as np
import pandas as pd
```

Рисунок 3.1 – Підключення бібліотеки numpy та pandas

Для роботи з даними погодних умов, описаними в підрозділі 2.1, необхідно завантажити їх до нашого робочого файлу. Зробити це можна саме за допомогою бібліотеки pandas. Даний код зображено на рисунку 3.1.

```
train_data = pd.read_csv('../input/weather-2010/weatherSet.csv')
test_data = pd.read_csv('../input/weather-2010/weatherSet.csv')
```

Рисунок 3.2 – Завантаження даних

Після завантаження ми отримали об’єкт pandas, записаний у нашу зміну, яка містить двовимірну таблицю з нашими даними та дозволяє переглядати, модифікувати і працювати з ними за допомогою широкого набору методів. Для відображення записів таблиці додаймо код, зображений на рисунку 3.3. Результат його роботи зображено на рисунку 3.4.

```
train_data.head()
```

Рисунок 3.3 – Код для відображення перших п’яти рядків таблиці

	date	avgTemp	maxTemp	minTemp	avgHumidity	minHumidity	avgHumidityDef	maxHumidityDef	cloud	lowCloud	maxWind	precipitation
0	2010-01-01	-3.1	-2.4	-4.0	96.0	93	0.2	0.4	10.0	10.0	10	10.0
1	2010-01-02	-5.5	-2.9	-6.9	87.0	80	0.5	0.8	10.0	9.0	12	2.1
2	2010-01-03	-10.2	-5.4	-14.7	79.0	74	0.6	0.8	8.0	2.0	10	0.2
3	2010-01-04	-8.6	-7.1	-9.6	85.0	76	0.5	0.8	10.0	9.0	5	1.0
4	2010-01-05	-8.2	-6.7	-9.8	83.0	75	0.5	0.8	10.0	9.0	7	0.0

Рисунок 3.4 – Елементи таблиці об’єкту pandas

Для отримання базових аналітичних даних потрібно побудувати звіт по об’єкту pandas. Це можна зробити за допомогою бібліотеки pandas_profiling. Вона приймає як аргумент об’єкт pandas та генерує HTML-звіт з базовим статистичним аналізом даних. Код для підключення бібліотеки та її виклику зображено на рисунку 3.5.

```
import pandas_profiling as pp
pp.ProfileReport(train_data)
```

Рисунок 3.5 – Використання бібліотеки pandas_profiling

Згенерований звіт наведено на рисунку 3.6.

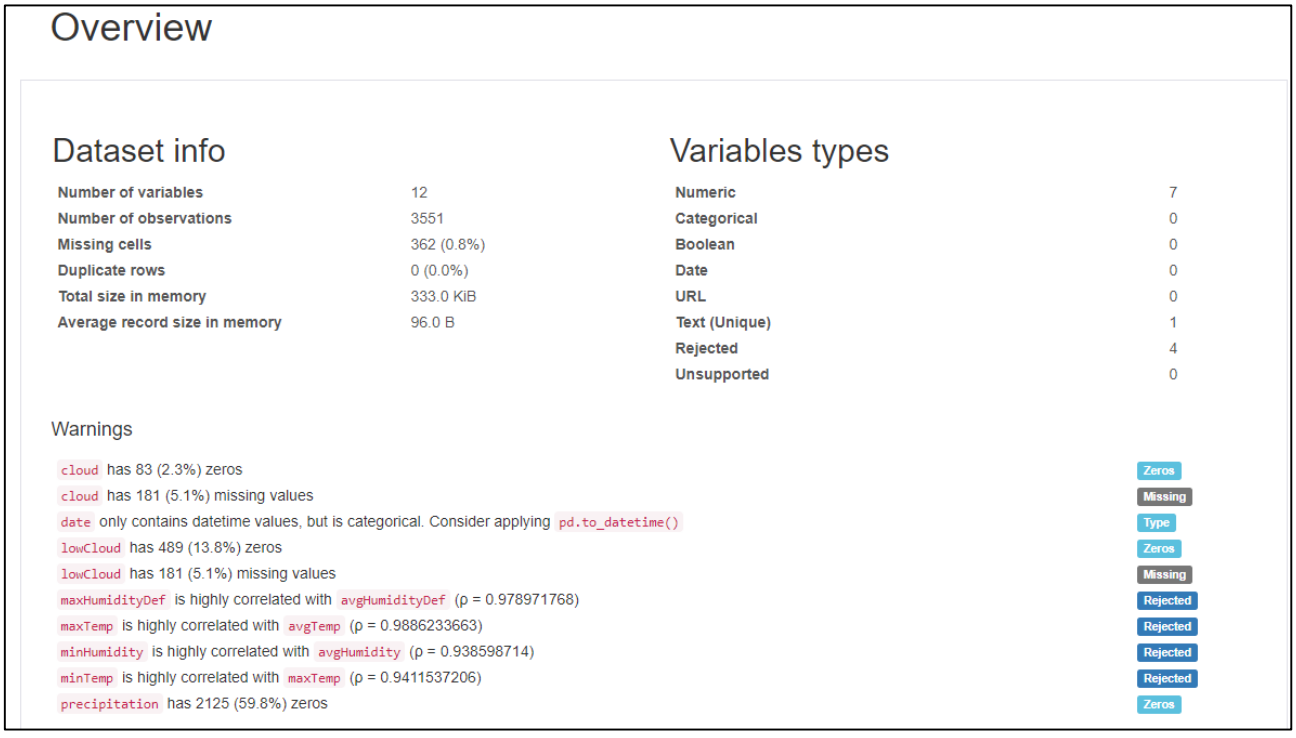


Рисунок 3.6 – Звіт для набору даних погодних умов

Як видно з звіту, у наборі даних відсутня незначна кількість значень для декількох атрибутів. Крім того, деякі з атрибутів мають високу кореляцію з іншими. Цю інформацію потрібно використати при подальшому аналізі. Для детальнішого розуміння взаємозв'язків всередині даного набору даних, порахуємо попарну кореляцію усіх стовпців таблиці, використавши відповідний метод об'єкту pandas та візуалізуємо це допомогою бібліотеки seaborn. Код наведено на рисунку 3.7, а результат його роботи на рисунку 3.8.

```
corr = train_data.corr()
fig = plt.figure(figsize=(15,10))
sns.heatmap(corr)
```

Рисунок 3.7 – Знаходження та відображення кореляції для набору даних

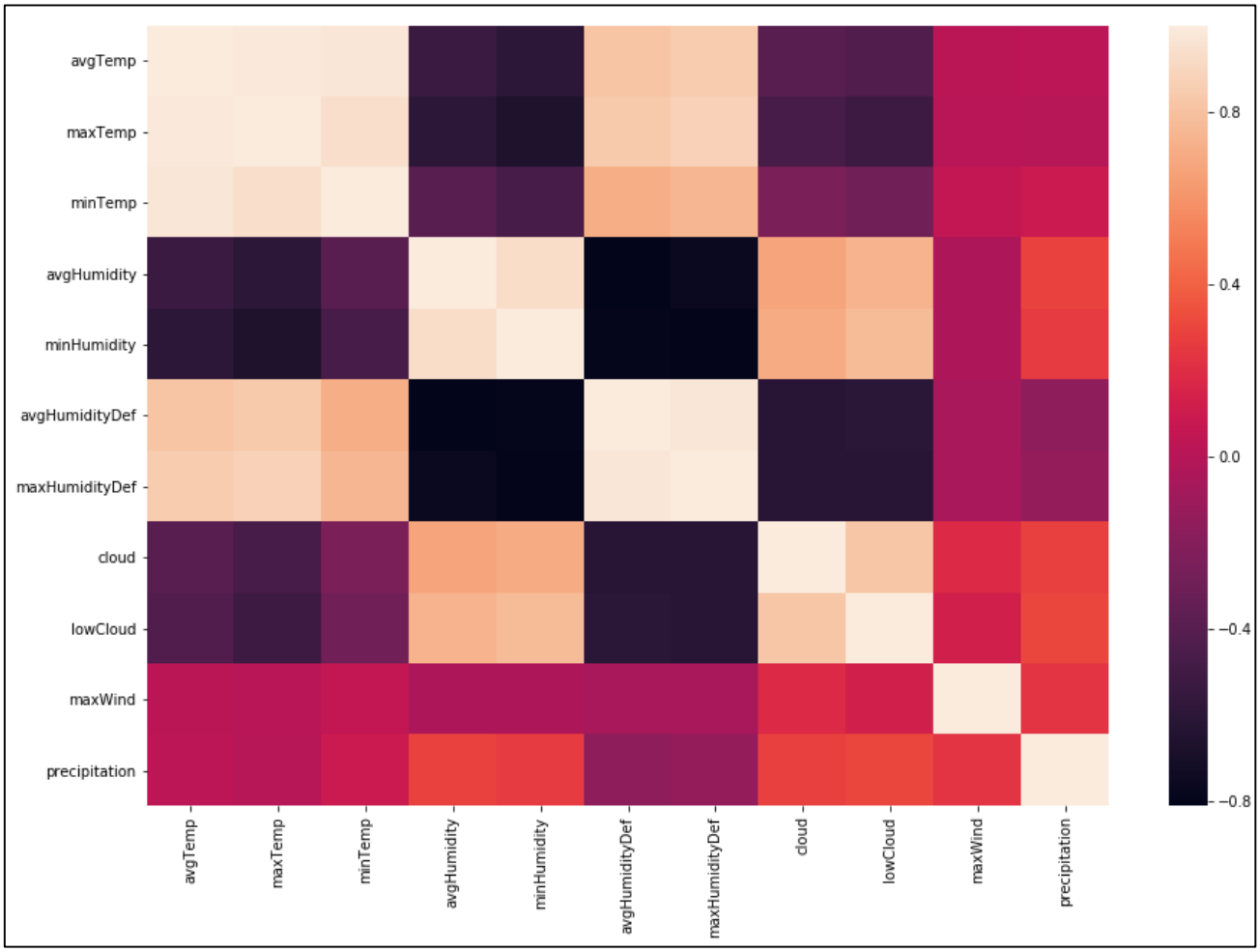


Рисунок 3.8 – Візуалізація обрахованої попарної кореляції стовпців

Наступним кроком буде більш детальний аналіз окремих атрибутів. Спочатку побудуємо графік залежності наявності опадів від значення мінімальної температури. Для цього спочатку необхідно для всіх ненульових значень опадів встановити 1. Це дозволить чітко відобразити для яких значень опади були, а для яких ні. Реалізація даної логіки зображено на рисунку 3.9.

```
plt.figure(figsize=(8,8))
sns.FacetGrid(analysis_data, hue="precipitation", size=8).map(sns.kdeplot, "minTemp").add_legend()
plt.ioff()
plt.show()
```

Рисунок 3.9 – Відображення залежності опадів від значень температури

Результат виконання даного коду зображено на рисунку 3.10.

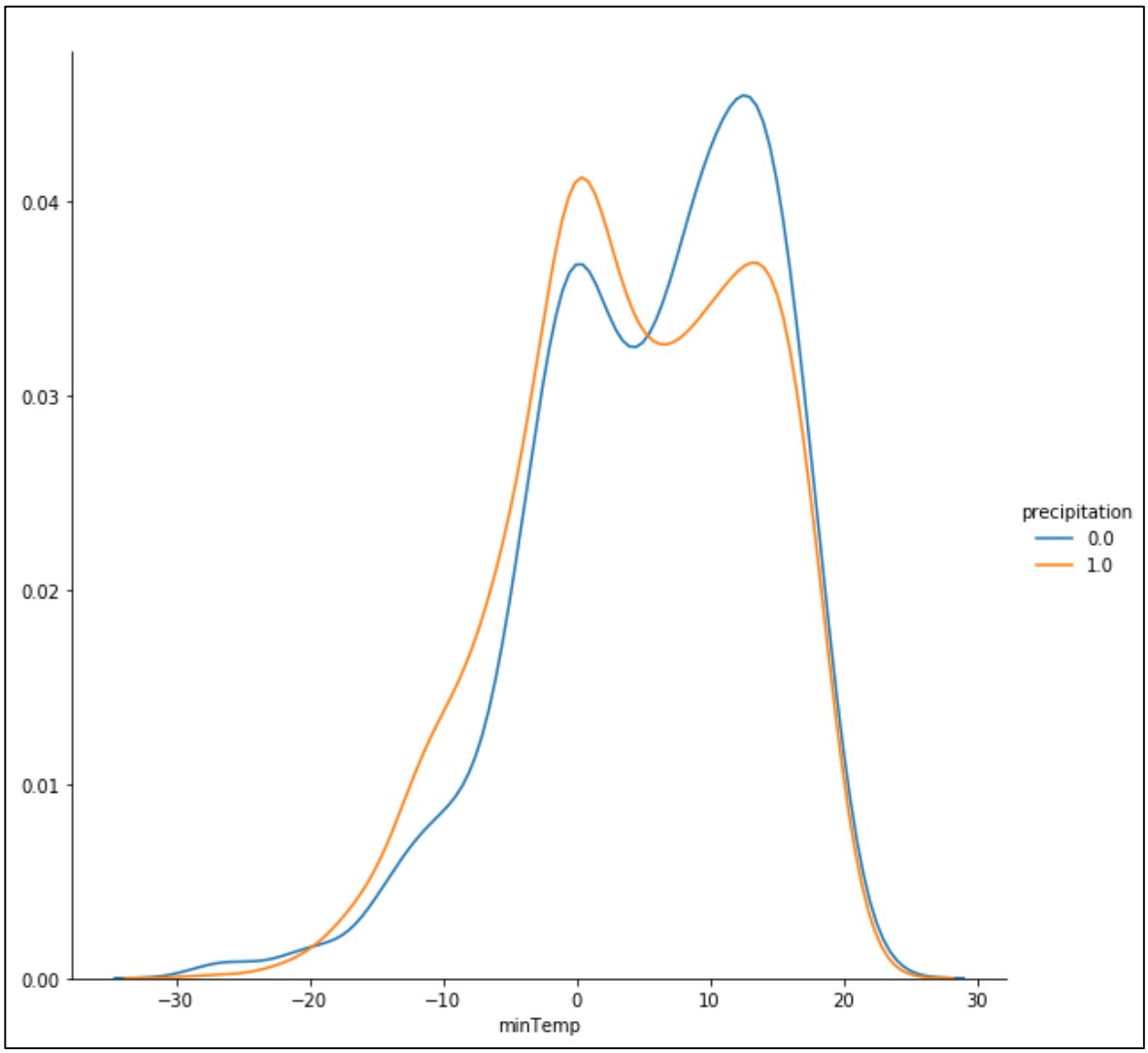


Рисунок 3. 10 – Графік залежності опадів від значень температури

З графіку видно, що в цілому спостерігається більш менш однаковий розподіл наявності або відсутності опадів від мінімальної температури. Зробимо подібне також для значень мінімальної вологості. Код для відображення та результат його роботи зображено на рисунках 3.11-3.12.

```
plt.figure(figsize=(8,8))
sns.FacetGrid(analysis_data, hue="precipitation", size=8).map(sns.kdeplot, "minHumidity").add_legend()
plt.ioff()
plt.show()
```

Рисунок 3.11 – Відображення залежності опадів від значень вологості

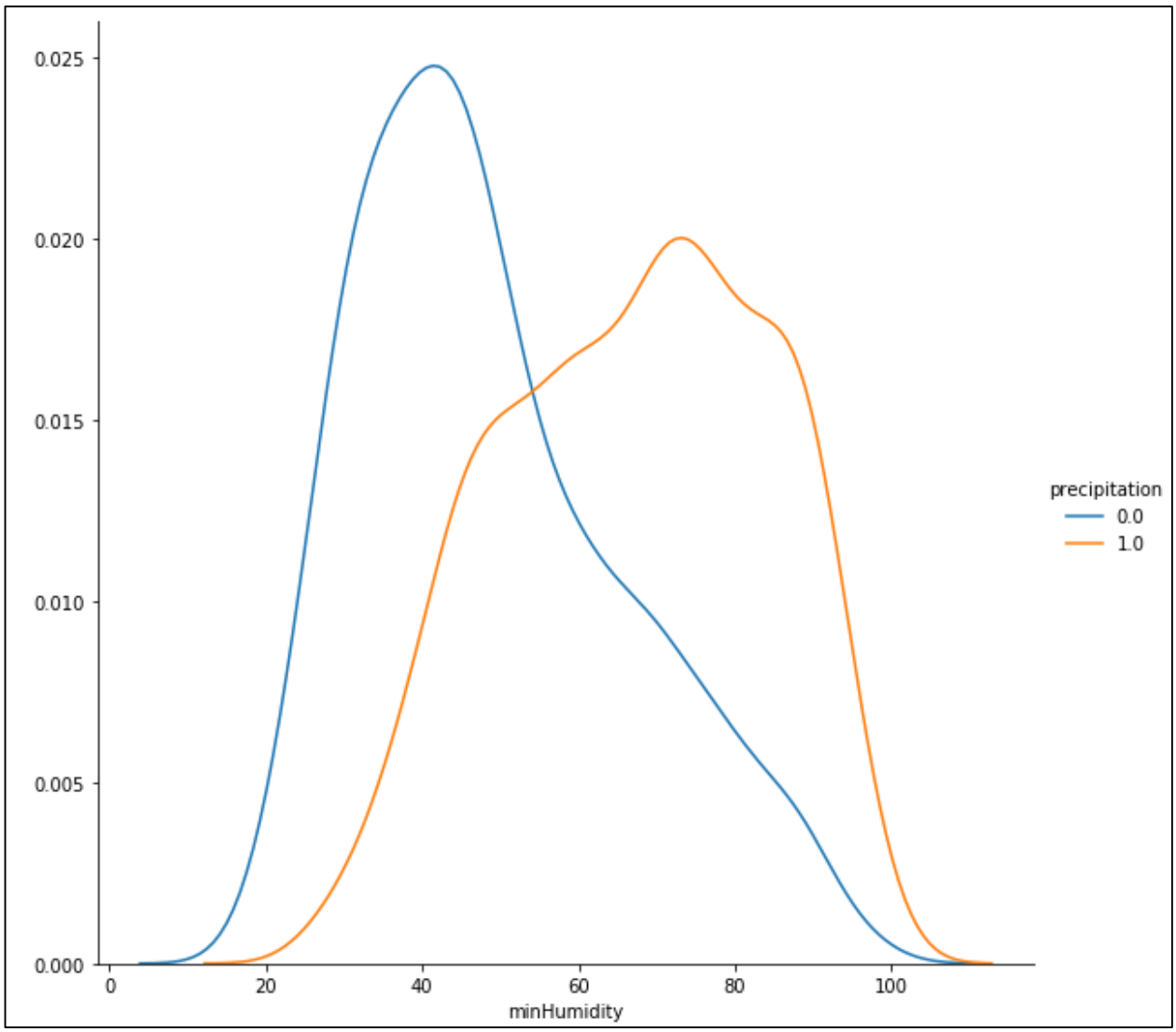


Рисунок 3.12 – Графік залежності опадів від значень вологості

Тут вже можна помітити, що для меншого значення показника мінімальної вологості (менше 60 відсотків) спостерігається відсутність опадів. У той же час для значень більшість присутніх опадів спостерігається для значення вологості у діапазоні від 55 до 85 відсотків. Тепер побудуємо діаграму розподілу опадів по відношенню до значень показника нижньої хмарності. Даний код наведено на рисунку 3.13.

```
plt.figure(figsize=(8,8))
sns.FacetGrid(train_data, hue="precipitation", size=8).map(sns.countplot, "lowCloud").add_legend()
plt.ioff()
plt.show()
```

Рисунок 3.13 – Відображення діаграми кількості значень нижнього показника хмарності

Діаграма, побудовано даним кодом, зображена на рисунку 3.14.

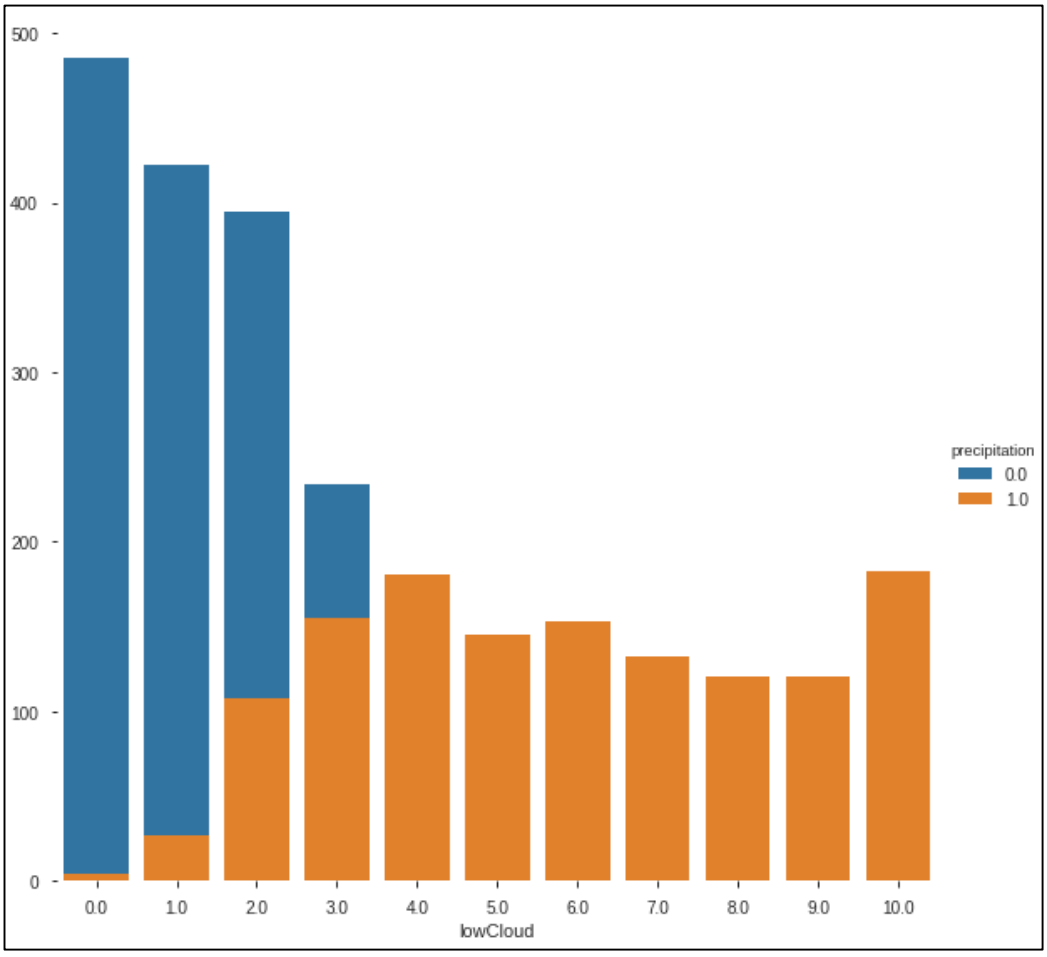


Рисунок 3.14 – Діаграма кількості значень нижнього показника хмарності

Помітно, що для усіх значень, починаючи з 4, спостерігається відсутність опадів. Подібну діаграму побудуємо також для показника хмарності. Код та результат його роботи зображені на рисунках 3.15-3.16 відповідно.

```
plt.figure(figsize=(8,8))
sns.FacetGrid(train_data, hue="precipitation", size=8).map(sns.countplot, "cloud").add_legend()
plt.ioff()
plt.show()
```

Рисунок 3.15 – Відображення діаграми кількості значень хмарності

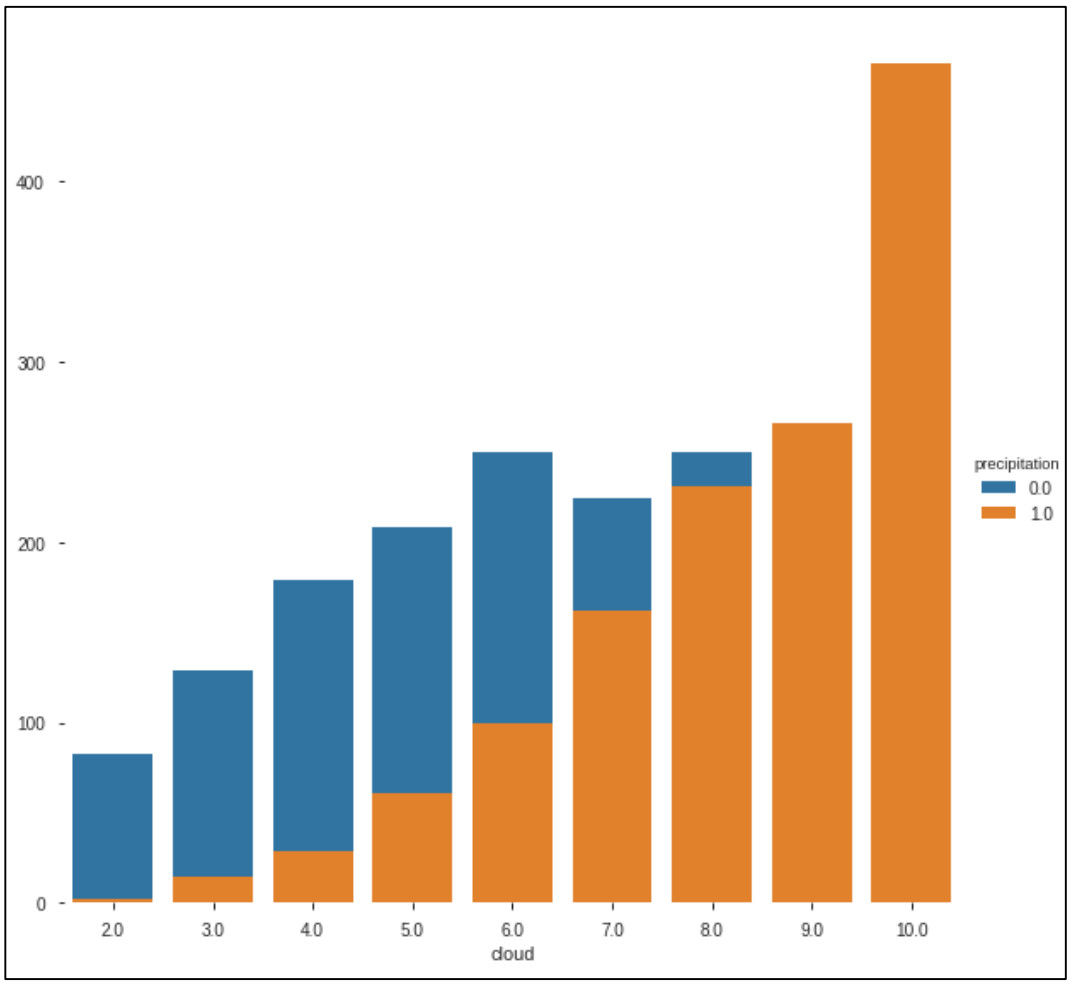


Рисунок 3.16 – Діаграма кількості значень показника хмарності

З цієї діаграми видно, що кількість опадів стрімко збільшується починаючи з значення 5 даного показника.

Враховуючи весь проведений аналіз, можна зробити висновок, що нам необхідно видалити атрибут date, так як він не несе ніякого впливу на показники опадів, а також нам потрібно позбутися від пропущених даних для показників cloud та lowCloud. Для цього додаймо наступний код, який зображений на рисунку 3.17, опис модифікованого об’єкту наведено на рисунку 3.18.

```
cols_to_remove = ['date']
train_data.drop(cols_to_remove, axis=1, inplace=True)
test_data.drop(cols_to_remove, axis=1, inplace=True)

train_data = train_data.dropna(how='any')
test_data = test_data.dropna(how='any')
```

Рисунок 3.17 – Видалення зайвих даних

	avgTemp	maxTemp	minTemp	avgHumidity	minHumidity	avgHumidityDef	maxHumidityDef	cloud	lowCloud	maxWind	precipitation
count	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000
mean	9.476460	14.177935	5.181923	73.131869	56.161128	4.984671	9.704332	6.463205	3.858457	9.873294	0.394659
std	9.963356	11.441242	8.876597	14.257363	19.347193	4.603779	8.564911	2.855579	3.123722	3.304965	0.488850
min	-23.400000	-19.400000	-28.500000	29.000000	16.000000	0.000000	0.000000	0.000000	0.000000	3.000000	0.000000
25%	1.600000	4.500000	-0.800000	63.000000	41.000000	1.100000	2.100000	4.000000	1.000000	7.000000	0.000000
50%	10.400000	15.400000	6.000000	74.000000	53.000000	3.400000	7.400000	7.000000	3.000000	9.000000	0.000000
75%	18.100000	24.100000	12.500000	85.000000	72.000000	7.900000	15.500000	9.000000	6.000000	11.000000	1.000000
max	28.000000	37.300000	22.900000	100.000000	100.000000	23.800000	47.700000	10.000000	10.000000	28.000000	1.000000

Рисунок 3.18 – Дані після видалення зайвих записів

В результаті, як видно на рисунку 3.18. ми втратили 181 запис, що не є досить критичним. Тепер необхідно підготувати ці дані для подальшої роботи з ними та можливості тренувати і тестувати моделі. Для цього додаймо код, який зображений на рисунку 3.19.

```
y = train_data['precipitation']
del train_data['precipitation']

X = train_data;

X_train, X_valid, y_train, y_valid = train_test_split(X, y, train_size=0.8, test_size=0.2, random_state=0)
```

Рисунок 3.19 – Розподіл даних на тренувальні та тестувальні

Наступним кроком є визначення найвагоміших атрибутів нашого набору даних. Спочатку знайдемо показники важливості кожного атрибуту використавши модель lgbm, натренуємо її нашими даними та використаємо відповідний метод моделі для отримання переліку атрибутів та коефіцієнтів їх важливості. Для цього додаймо код, зображений на рисунку 3.20.

```
train_set = lgbm.Dataset(X_train, y_train, silent=False)
valid_set = lgbm.Dataset(X_valid, y_valid, silent=False)

params = {
        'boosting_type':'gbdt', 'objective': 'regression', 'num_leaves': 31, 'learning_rate': 0.05,
        'max_depth': -1, 'subsample': 0.8, 'bagging_fraction' : 1, 'max_bin' : 5000 , 'bagging_freq': 20,
        'colsample_bytree': 0.6, 'metric': 'rmse', 'min_split_gain': 0.5, 'min_child_weight': 1,
        'min_child_samples': 10, 'scale_pos_weight':1, 'zero_as_missing': True, 'seed':0,
    }

modelL = lgbm.train(params, train_set = train_set, num_boost_round=1000,
                    early_stopping_rounds=50,verbose_eval=10, valid_sets=valid_set)
```

Рисунок 3.20 – Створення моделі lgbm

Та побудуємо на основі цих значень діаграму, використавши код, який зображений на рисунку 3.21. Сама діаграма зображена на рисунку 3.22.

```
fig =  plt.figure(figsize = (15,15))
axes = fig.add_subplot(111)
lgbm.plot_importance(modelL,ax = axes,height = 0.5)
plt.show();plt.close()
```

Рисунок 3.21 – Побудова діаграма важливості атрибутів набору даних

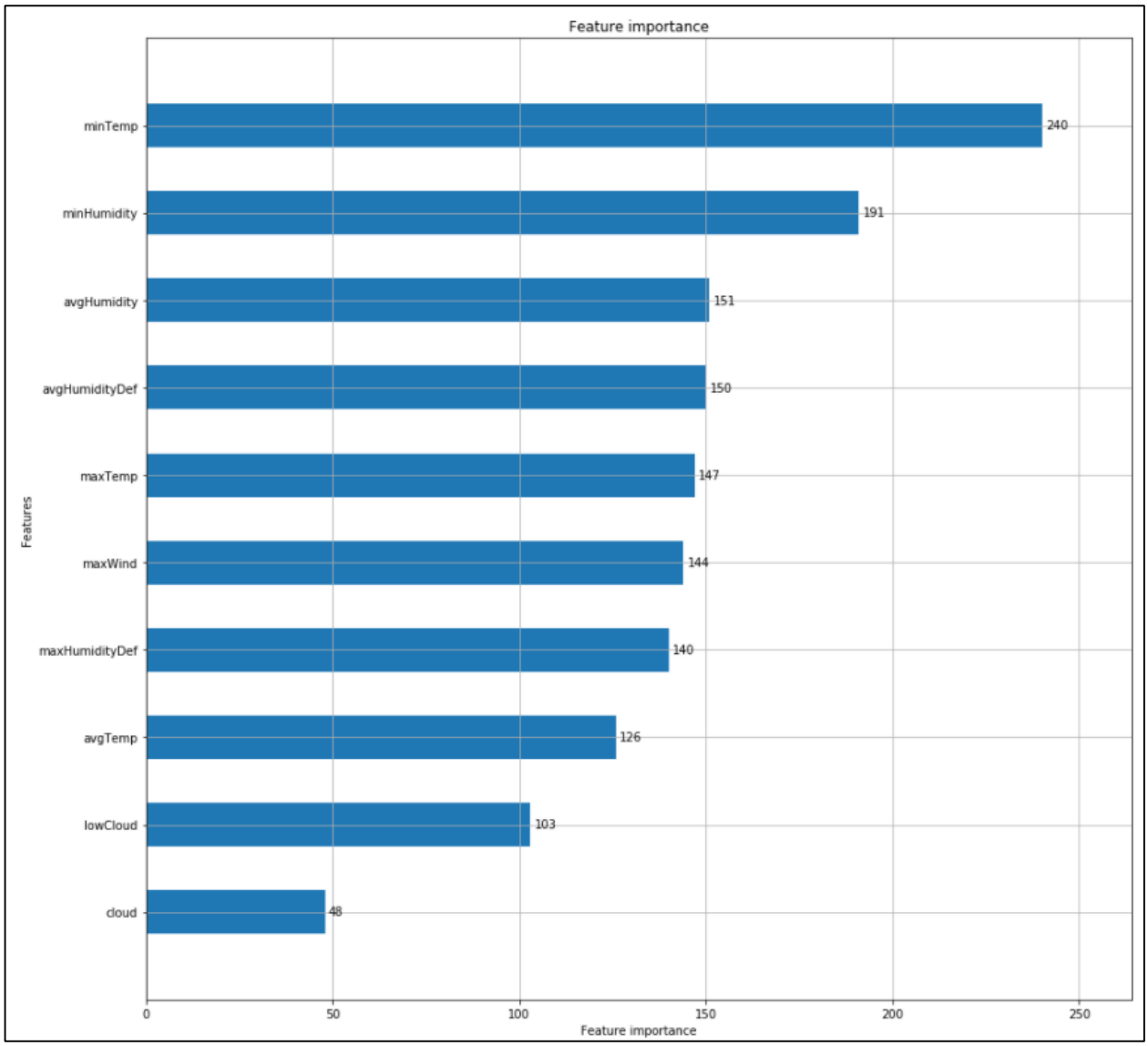


Рисунок 3.22 – Діаграма важливості атрибутів згідно результатів моделі lgbm

Тепер побудуємо спробуємо визначити коефіцієнти важливості атрибутів, використавши бібліотеку xgb. Алгоритм виконання повністю збігається з тим, який застосовувався для бібліотеки lgbm. Код для створення моделі xgb наведено на рисунку 3.23

```
data_tr  = xgb.DMatrix(X_train, label=y_train)
data_cv  = xgb.DMatrix(X_valid   , label=y_valid)
data_train = xgb.DMatrix(X)
data_test  = xgb.DMatrix(y)
evallist = [(data_tr, 'train'), (data_cv, 'valid')]

parms = {
    'max_depth':8, 'eta':0.3, 'subsample':0.8, 'lambda ':4,
    'colsample_bytree ':0.9, 'colsample_bylevel':1, 'min_child_weight': 10
}
modelx = xgb.train(parms, data_tr, num_boost_round=200, evals = evallist,
                   early_stopping_rounds=30, maximize=False,
                   verbose_eval=10)
```

Рисунок 3.23 – Створення моделі xgb

Діаграма значень коефіцієнтів зображена на рисунку 3.24.

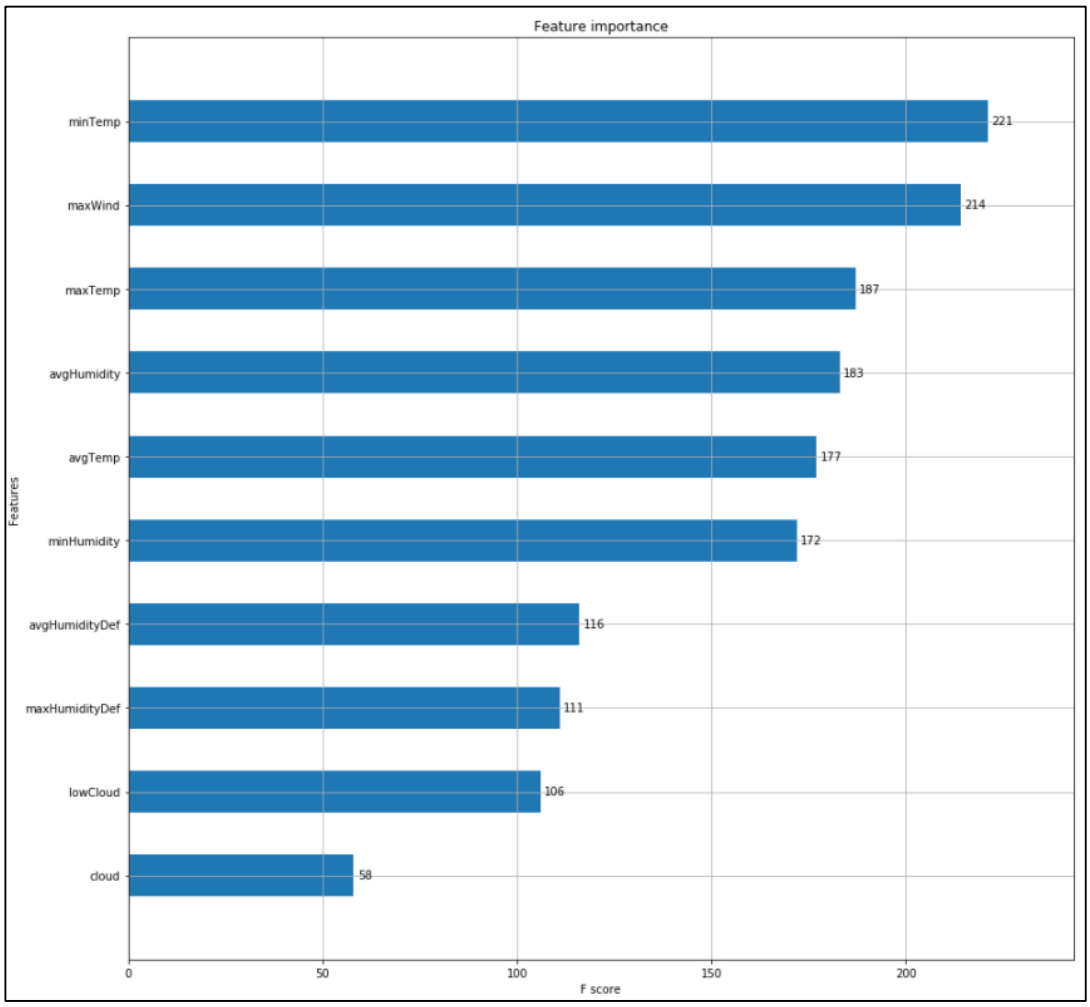


Рисунок 3.24 – Діаграма важливості атрибутів згідно результатів моделі xgb

Тепер, проведемо подібні розрахунки за допомогою моделі LinearRegression. Код та результат його виконання зображені на рисунках 3.25-3.26.

```
linreg = LinearRegression()
linreg.fit(X, y)
coeff_linreg = pd.DataFrame(X.columns)
coeff_linreg.columns = ['feature']
coeff_linreg["score_linreg"] = pd.Series(linreg.coef_)
coeff_linreg.sort_values(by='score_linreg', ascending=False)
```

Рисунок 3.25 – Знаходження коефіцієнтів важливості за допомогою LinearRegression

	feature	score_linreg
1	maxTemp	0.393227
8	lowCloud	0.268171
9	maxWind	0.244437
2	minTemp	0.074363
3	avgHumidity	0.041354
4	minHumidity	0.035451
6	maxHumidityDef	0.027409
7	cloud	0.005716
5	avgHumidityDef	-0.147902
0	avgTemp	-0.355524

Рисунок 3.26 – Значення коефіцієнтів важливості за допомогою LinearRegression

І наостанок, виконаємо ці самі дії з моделлю LogisticRegression. Код та результат його виконання зображені на рисунках 3.27-3.28.

```
Scaler_train = preprocessing.MinMaxScaler()
train = pd.DataFrame(
    Scaler_train.fit_transform(X),
    columns=X.columns,
    index=X.index
)

logreg = LogisticRegression()
logreg.fit(train, y)
coeff_logreg = pd.DataFrame(train.columns)
coeff_logreg.columns = ['feature']
coeff_logreg["score_logreg"] = pd.Series(logreg.coef_[0])
coeff_logreg.sort_values(by='score_logreg', ascending=False)
coeff_logreg["score_logreg"] = coeff_logreg["score_logreg"].abs()
feature_score = pd.merge(feature_score, coeff_logreg, on='feature')
```

Рисунок 3.27 – Знаходження коефіцієнтів важливості за допомогою LogisticRegression

	feature	score_logreg
0	avgTemp	0.105288
1	maxTemp	0.939418
2	minTemp	1.203480
3	avgHumidity	3.807568
4	minHumidity	0.653257
5	avgHumidityDef	2.174296
6	maxHumidityDef	1.381805
7	cloud	2.618418
8	lowCloud	1.314107
9	maxWind	3.915504

Рисунок 3.28 – Значення коефіцієнтів важливості за допомогою LogisticRegression

Всі отримані дані згрупуємо в один набір, порахуємо середні та загальні значення для кожного атрибуту. Для цього реалізуємо код, зображений на рисунку 3.29.

```
feature_score = pd.merge(feature_score, coeff_linreg, on='feature')
feature_score = feature_score.fillna(0)
feature_score = feature_score.set_index('feature')

feature_score = pd.DataFrame(
    preprocessing.MinMaxScaler().fit_transform(feature_score),
    columns=feature_score.columns,
    index=feature_score.index
)
feature_score['mean'] = feature_score.mean(axis=1)
feature_score.sort_values('mean', ascending=False).plot(kind='bar', figsize=(20, 10))
feature_score['total'] = 0.48 * feature_score['score_lgb'] + 0.48 * feature_score['score_xgb']
                       + 0.04 * feature_score['score_linreg']
feature_score.sort_values('total', ascending=False).plot(kind='bar', figsize=(20, 10))
```

Рисунок 3.29 – Групування коефіцієнтів важливості атрибутів

Таблицю результатів обрахунків наведено на рисунку 3.30.

feature	score_lgb	score_xgb	score_linreg	score_logreg	mean	total
minTemp	1.000000	1.000000	0.538865	0.288223	0.706772	0.974035
maxWind	0.641509	0.578231	0.676980	1.000000	0.724180	0.622245
maxTemp	0.377358	0.884354	0.538450	0.218919	0.504770	0.617574
avgHumidityDef	0.415094	0.517007	0.463164	0.543016	0.484570	0.468331
maxHumidityDef	0.226415	0.653061	0.330047	0.335025	0.386137	0.435500
minHumidity	0.179245	0.673469	0.000000	0.143816	0.249133	0.413618
avgHumidity	0.301887	0.476190	0.219066	0.971672	0.492204	0.404818
avgTemp	0.000000	0.741497	1.000000	0.000000	0.435374	0.365918
lowCloud	0.264151	0.210884	0.868651	0.317257	0.415236	0.246221
cloud	0.160377	0.000000	0.735391	0.659577	0.388836	0.104122

Рисунок 3.30 – Результуючі значення коефіцієнтів важливості атрибутів

Додатково представимо ці результати у вигляді діаграми. Дана діаграма зображена на рисунку 3.31.

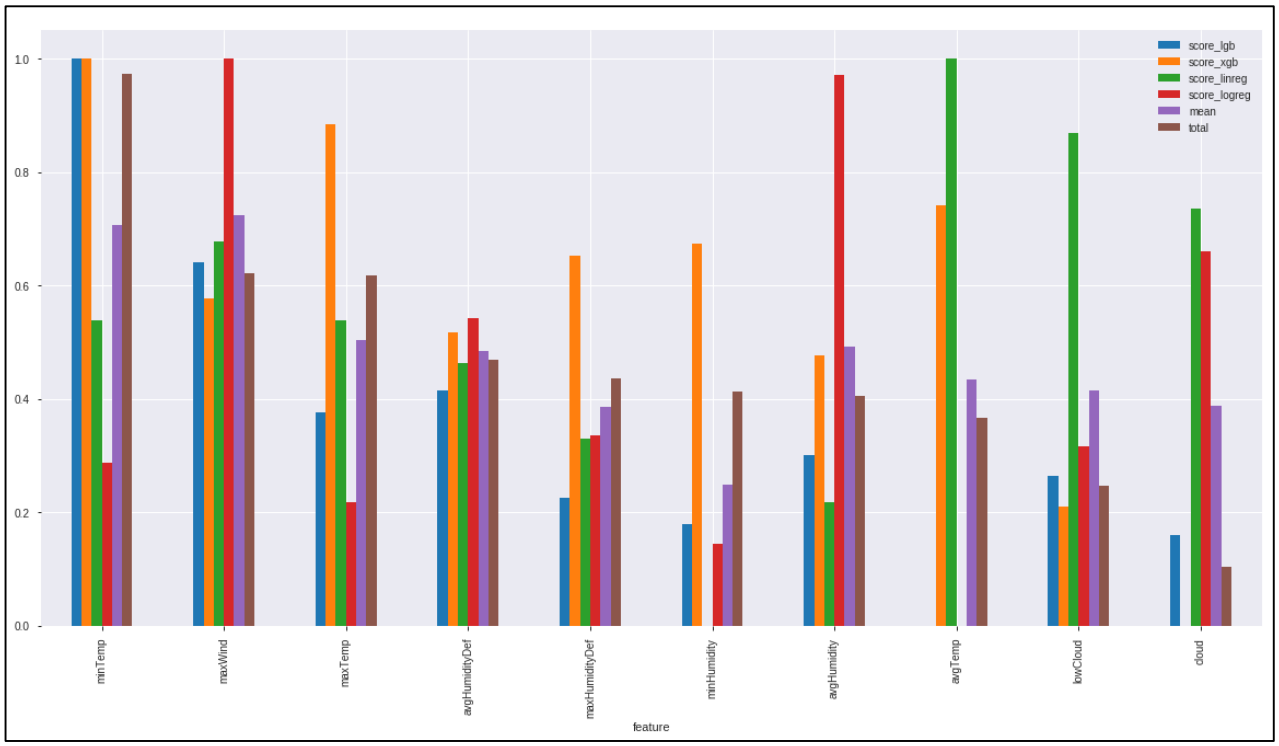


Рисунок 3.31 – Діаграма значень коефіцієнтів важливості атрибутів

Враховуючи проведений аналіз даних та обраховані ваги кожного атрибуту, було вирішено використовувати для подальшого тренування моделей атрибути зазначені на рисунку 3.32.

```
feature_columns = ['avgTemp', 'maxHumidityDef', 'maxWind', 'avgHumidity', 'cloud']
X = X[feature_columns];

X.head()
```

Рисунок 3.32 – Перелік найвагоміших атрибутів

Приклад обраних даних, наведено на рисунку 3.33.

	avgTemp	maxHumidityDef	maxWind	avgHumidity	cloud
0	-3.1	0.4	10	96.0	10.0
1	-5.5	0.8	12	87.0	10.0
2	-10.2	0.8	10	79.0	8.0
3	-8.6	0.8	5	85.0	10.0
4	-8.2	0.8	7	83.0	10.0

Рисунок 3.33 – Приклад найвагоміших даних

3.2 Вибір оптимальної моделі передбачення

У попередньому розділі було обрано найвагоміші дані і тепер необхідно використати їх для вибору моделі, яка даватиме найкращий показник точності передбачення. Для цього розглянемо п’ять моделей [19] – натренуємо їх нашим тренувальним набором даних та знайдемо оцінку точності для кожної з них.

Першою моделлю є GradientBoostingClassifier. Код для створення моделі, її тренування та визначення точності наведено на рисунку 3.34.

```
gbc = GradientBoostingClassifier()
gbc.fit(X_train, y_train)
pred = gbc.predict(X_valid)

result = accuracy_score(y_valid, pred) * 100
results.append(result)
```

Рисунок 3.34 – Застосування моделі GradientBoostingClassifier

Результат точності для даного класифікатора становить 78.1%, що є досить гарним показником. Наступним перевіримо роботу моделі AdaBoostClassifier. Аналогічний код для перевірки точності наведено на рисунку 3.35.

```
abc = AdaBoostClassifier()
abc.fit(X_train, y_train)
pred = abc.predict(X_valid)

result = accuracy_score(y_valid, pred) * 100
results.append(result)
```

Рисунок 3.35 – Застосування моделі AdaBoostClassifier

Класифікатор демонструє точність в 78.4%, що трішки перевершує попередній. Тепер спробуємо застосувати модель RidgeClassifier. Код для перевірки наведено на рисунку 3.36

```
ridge = RidgeClassifier()
ridge.fit(X_train, y_train)
pred = ridge.predict(X_valid)

result = accuracy_score(y_valid, pred) * 100
results.append(result)
```

Рисунок 3.36 – Застосування моделі RidgeClassifier

Точність передбачення становить 79.5%, що поки є найкращим результатом. Тепер визначимо точність передбачення для моделі BaggingClassifier. Даний код зображено на рисунку 3.37.

```
bagging = BaggingClassifier()
bagging.fit(X_train, y_train)
pred = bagging.predict(X_valid)

result = accuracy_score(y_valid, pred) * 100
results.append(result)
print(result)
```

Рисунок 3.37 – Застосування моделі BaggingClassifier

Модель демонструє точність 75.6%, що є найменшим серед усіх попередніх. Наостанок, спробуємо подібний алгоритм для моделі MLPClassifier. Код для перевірки зображено на рисунку 3.38.

```
mlp = MLPClassifier()
mlp.fit(X_train, y_train)
pred = mlp.predict(X_valid)

result = accuracy_score(y_valid, pred) * 100
results.append(result)
print(result)
```

Рисунок 3.37 – Застосування моделі MLPClassifier

Показник точності дорівнює 76.2%. Тепер, для візуалізації усіх отриманих результатів побудуємо діаграму, використавши код наведений на рисунку 3.38. Результат його роботи зображено на рисунку 3.39.

```
x = np.arange(5)

fig, ax = plt.subplots()
fig.set_size_inches(18.5, 10.5)
plt.bar(x, results)
ax.set_ylim(bottom=75)
plt.xticks(x, ('GradientBoosting', 'AdaBoost', 'Ridge', 'Bagging', 'MLP'))
plt.show()
```

Рисунок 3.38 – Створення графіку для порівняння результатів точності

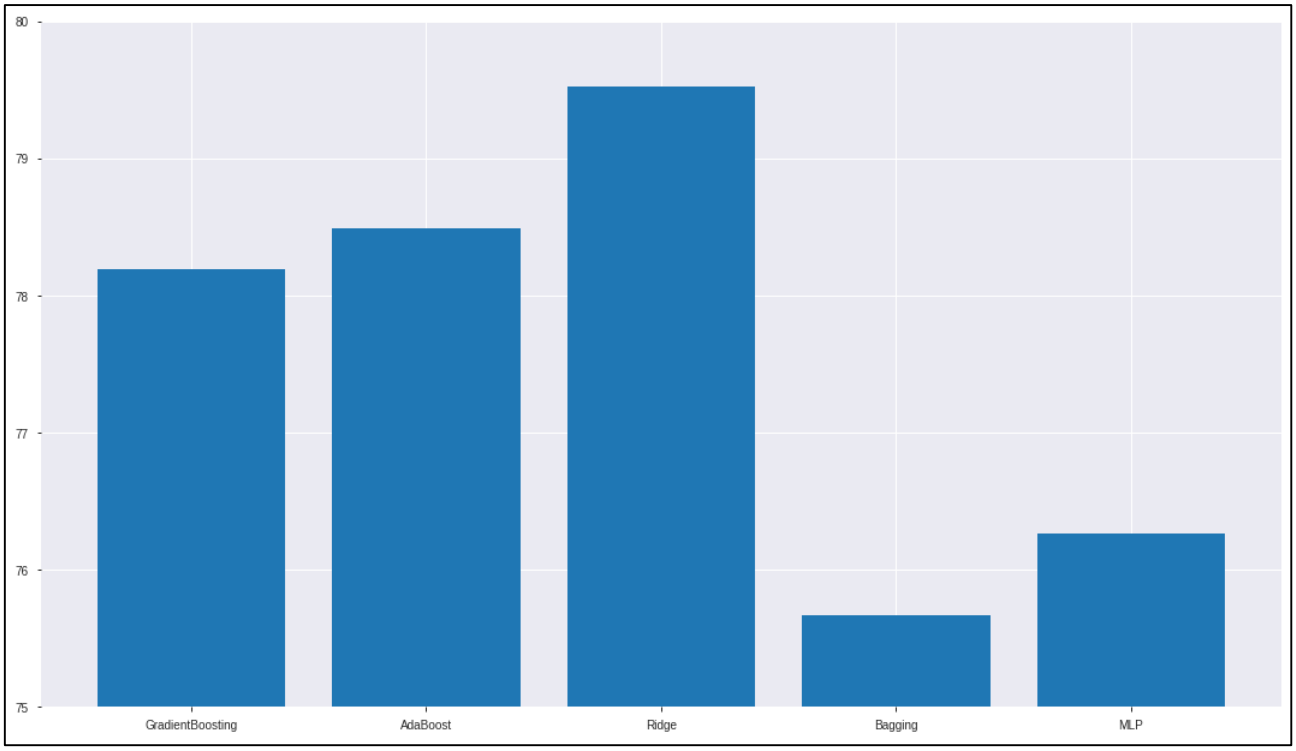


Рисунок 3.39 – Порівняння результатів точності

З графіку порівняння видно, що найкраще себе показала модель RidgeClassifier. Саме вона і буде використовуватися для створення нашої інформаційної технології передбачення опадів.

3.3 Тестування розробленої моделі

Протестуємо розроблену інформаційну технологію передбачення, використавши обрану в попередньому розділі модель RidgeClassifier. Для перевірки створимо об’єкт передбачення і натренуємо його нашими даними, як показано на рисунку 3.40.

```
clss = RidgeClassifier()
clss.fit(X, y)
```

Рисунок 3.40 – Тренування моделі

Знайдемо точність передбачення для цієї моделі, використавши код з рисунку 3.41.

```
test_X = test_data[feature_columns]
test_y = test_data['precipitation']
clss.score(test_X, test_y)
```

Рисунок 3.41 – Перевірка точності передбачення

На даних для тестування модель показує точність у 78.9%, що можна вважати задовільним. Намалюємо діаграму для попарного порівняння реальних даних з передбаченими. Для цього спочатку виберемо певну кількість записів з набору тренувальних даних, як це показано на рисунку 3.42 та зобразимо діаграму за допомогою коду з рисунку 3.43.

```
plot_data = test_data.sample(120)

condition = plot_data[feature_columns]
target = plot_data['precipitation']

pred = clss.predict(condition)
```

Рисунок 3.42 – Вибір випадкових даних для попарного порівняння результатів передбачення

```
colors = np.empty(len(pred), dtype=object)

target_array = np.array(target)

for i in range(len(pred)):
    colors[i] = 'blue' if pred[i] == target_array[i] else 'red'

x1 = np.arange(len(pred))
x2 = np.arange(len(pred))

fig, (ax1, ax2) = plt.subplots(2, 1)
fig.set_size_inches(22, 10.5)
fig.suptitle('Target vs Prediction')

ax1.set_ylabel('Target')
ax1.step(x1, target, where="post")
ax1.scatter(x1, target , marker='o',c = colors)


ax2.set_ylabel('Prediction')
ax2.step(x2, pred, where="post")
ax2.scatter(x2, pred , marker='o',c = colors)

plt.show()
```

Рисунок 3.43 – Зображення попарного порівняння результатів передбачення з реальними даними

Побудована діаграма зображена на рисунку 3.44.

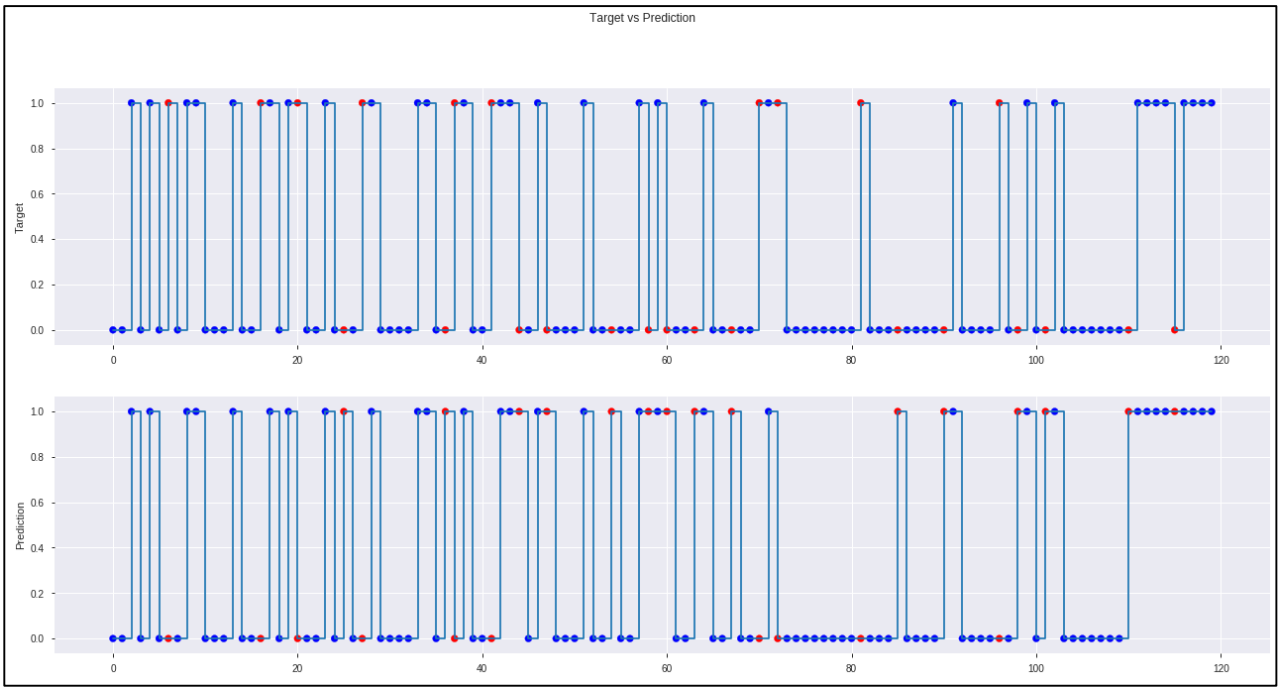


Рисунок 3.44 – Графік попарного порівняння результатів передбачення з реальними даними

Як видно з графіку, інформаційна технологія демонструє досить малу кількість хибних передбачень.

3.4 Висновки

В даному розділі було проведено аналіз атрибутів датасету, що використовується для реалізації технології прогнозування, та сконструйовано перелік головних ознак з найбільшим впливом на результат передбачення опадів.

Серед п’яти моделей (GradientBoosting, AdaBoost, Ridge, Bagging, MLP) обрано найточнішу модель прогнозування та наведено графік їх порівнянь.

Використовуючи обрані найвагоміші атрибути та найточнішу модель передбачення, було проведено тестування та отримано результат точності, який склав 78.9%.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, а саме, інформаційної технології аналізу та передбачення опадів, створеної в результаті науково-технічної діяльності.

Проведення технологічного аудиту здійснюється за допомогою експертного методу. Для цього залучено 2-х незалежних експертів.

Оцінювання комерційного потенціалу розробки здійснюється за 12 критеріями, наведеними в таблиці 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
1	2	3	4	5	6
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

1	2	3	4	5	6
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовую	Потрібні дорогі матеріали	Потрібні досяжні та дешеві	Всі матеріали для реалізації ідеї відомі

Продовження таблиці 4.1

1	2	3	4	5	6
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу розробки зведемо в таблицю 4.2, рівень потенціалу визначимо по даних таблиці 4.3.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Експерти	
	1.	2
	Бали, виставлені експертами:	
1	4	4
2	2	4
3	3	2
4	4	4
5	4	4
6	3	2
7	3	4
8	4	3
9	2	2
10	4	3

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Згідно результатів оцінювання комерційного потенціалу розробки, що наведено в таблиці 4.2, середньоарифметична сума балів, розрахована на основі висновків експертів складає 37,5. Використовуючи дані таблиці 4.3 можна зробити висновок, що рівень потенціалу розробки – вище середнього.

4.2 Прогнозування витрат на виконання роботи

Прогнозування витрат на розробку інформаційної технології аналізу та передбачення опадів, передбачає: розрахунок витрат, які безпосередньо стосуються виконавців даної роботи; розрахунок загальних витрат на виконання роботи та прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи, можна здійснити за такими статтями та формулами:

1. Основна заробітна плата кожного із розробників (дослідників) $З_о$, якщо вони працюють в наукових установах бюджетної сфери:

$$З_о = \frac{М}{Т_р} \cdot t \text{ [грн]}, \quad (4.1)$$

де М – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн. У 2020 році величини окладів (разом з встановленими доплатами і надбавками) рекомендується брати в межах (5000…15000) грн. за місяць;

$Т_р$ – число робочих днів в місяці, приймемо $Т_р = 22$ дні;

t – число робочих днів роботи розробника (дослідника).

Згідно формули 4.1 розрахуємо основну заробітну плату для кожного з розробників. Результати розрахунку внесемо до таблиці 4.4.

Таблиця 4.4 – Результати розрахунку витрат на виконання даної роботи

Найменування посади виконавця	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на оплату праці, грн.
1. Інженер-програміст	15000	681,82	11	7500,02
2. Науковий керівник	9500	431,80	5	2299,98
Всього				$З_о = 9800$

2. Додаткова заробітна плата $З_д$ всіх розробників розраховується як (10…12)% від суми основної заробітної плати всіх розробників, тобто:

$$З_д = (0{,}1 \dots 0{,}12) \cdot (З_о + З_р) \text{ [грн]}, \quad (4.2)$$

$$З_д = 0{,}1 \cdot (9800) = 980 \text{ (грн)}.$$

3. Нарахування на заробітну плату $Н_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$Н_{зп} = (З_о + З_д) \cdot \frac{\beta}{100} \text{ [грн]}, \quad (4.3)$$

де $З_о$ – основна заробітна плата розробників, грн;

$З_д$ – додаткова заробітна плата всіх розробників та робітників, грн;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, %. Ставка єдиного внеску на загальнообов'язкове державне соціальне страхування для бюджетних організацій у 2020 році встановлена у розмірі 22%.

$$Н_{зп} = (9800 + 980) \cdot \frac{22}{100} = 2371{,}6 \text{ (грн)}.$$

4. Амортизація обладнання, комп'ютерів та приміщень А, які використовувались під час (чи для) виконання даного етапу роботи. Результати розрахунків внесемо у таблицю 4.5.

У спрощеному вигляді амортизаційні відрахування А розраховуються за формулою:

$$А = \frac{Ц * На}{100} \cdot \frac{Т}{12} \text{ [грн]}, \quad (4.4)$$

де Ц – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн;

$Н_а$ – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $Н_а$ = (10...25)%;

Т – термін, використання обладнання, приміщень тощо, місяці.

Таблиця 4.5 – Результати розрахунків амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
1. Комп'ютери	70000	20	1,5	1750
2. Приміщення	128000	10	1	1050
Всього				A = 2800

5. Витрати на послуги, що були використані під час виконання наукової розробки.

Підрахунки витрат на послуги, використані під час наукової діяльності, наведено в таблиці 4.6.

Таблиця 4.6 – Послуги, що використовуються при виготовленні програми

Найменування послуг	Кількість, шт.	Ціна за одиницю, грн.	Сума, грн.
1. Послуга , доступ до мережі Internet шт.	3	200	600
2. Послуга середовище розробки PyCharm, шт.	2	500	1000
3. Програмне забезпечення для роботи з Python	1	500	500
Всього			2100 грн.

6. Витрати на комплектуючі, що були використані для розробки

Підрахунок витрат на комплектуючі для розробки наведено в таблиці 4.7.

Таблиця 4.7 – Витрати на комплектуючі, що були використані для розробки

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Диск	шт.	10	1	10
Блокнот	шт.	35	1	35
Диск	шт.	10	1	10
Блокнот	шт.	35	1	35
Ручка	шт.	20	1	20
Папір	уп.	185	1	180
Всього з урахуванням транспортних витрат:				245

7. Витрати на силову електроенергію $В_е$ розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п \text{ [грн]}, \quad (4.5)$$

де В – вартість 1 кВт-год. електроенергії, в 2020р. В ≈ 2,5 грн./кВт;

П – установлена потужність обладнання, кВт; П = 1,5 кВт;

Ф – фактична кількість годин роботи обладнання, годин. Приймемо, що Ф = 29 годин;

$К_п$ – коефіцієнт використання потужності, $К_п < 1$, приймемо $К_п = 0,9$;

Тоді витрати на електроенергію складуть:

$$В_е = 2,5 \cdot 1,5 \cdot 29 \cdot 0,9 = 100 \text{ (грн)}.$$

8. Інші витрати $В_{ін}$ можна прийняти як (100…300)% від суми основної заробітної плати фахівців, які виконували дану роботу, тобто:

$$В_{ін} = (0{,}1..3) \cdot (З_о + З_р)\ [грн], \quad (4.6)$$

$$В_{ін} = 2 \cdot (9800 + 980) = 21560\ (грн).$$

9. Сума всіх попередніх статей витрат дає витрати на виконання даної частини етапу роботи – В.

$$В = 9800 + 980 + 2371{,}6 + 2100 + 245 + 2800 + 100 + 21560 =$$
$$= 39711{,}6\ (грн).$$

Розрахунок загальних витрат на виконання даної роботи. Загальна вартість всієї наукової роботи визначається за формулою:

$$В_{заг} = \frac{В}{\alpha}\ [грн], \quad (4.7)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях. Для нашого випадку приймемо, що α = 0,5.

Тоді:

$$В_{заг} = \frac{39711{,}6}{0{,}5} = 79432{,}2\ (грн).$$

Прогнозування загальних витрат на виконання та впровадження результатів виконаної наукової роботи виконується за формулою:

$$ЗВ = \frac{В_{заг}}{\beta}, \quad (4.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи. Якщо розробка знаходиться:

- на стадії науково-дослідних робіт, то $\beta \approx 0{,}1$;

- на стадії технічного проектування, то $\beta \approx 0{,}2$;

- на стадії розробки конструкторської документації, то $\beta \approx 0{,}3$;

- на стадії розробки технологій, то $\beta \approx 0{,}4$;

- на стадії розробки дослідного зразка, то $\beta \approx 0{,}5$;

- на стадії розробки промислового зразка, $\beta \approx 0{,}7$;

- на стадії впровадження, то $\beta \approx 0{,}9$.

$$ЗВ = \frac{79432{,}2}{0{,}5} = 158864{,}4 \text{ (грн)}.$$

Тобто прогнозовані витрати на виконання та можливе впровадження результатів даної роботи можуть скласти приблизно 159 тис. грн.

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі необхідно кількісно спрогнозувати, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи.

При цьому потрібно виходити з того, що в умовах ринку узагальнюючим позитивним результатом, що його отримує підприємство від впровадження результатів розробки, є збільшення чистого прибутку підприємства.

Саме зростання чистого прибутку забезпечить підприємству надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Зростання чистого прибутку ми можемо оцінити у теперішній вартості грошей.

Розрахуємо збільшення чистого прибутку підприємства $\Delta П_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, за формулою:

$$\Delta П_i = \sum_1^n (\Delta Ц_o \cdot N + Ц_o \cdot \Delta N) \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\upsilon}{100}\right), \quad (4.9)$$

де $\Delta Ц_o$ – покращення основного оціночного показника від впровадження результатів розробки у даному році.

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки.

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$Ц_o$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

λ – коефіцієнт, який враховує сплату податку на додану вартість. У 2020 році ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0{,}8333$.

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $\rho = 0{,}2...0{,}3$;

υ – ставка податку на прибуток. У 2020 році υ = 18%.

Припустимо, що в результаті впровадження результатів наукової розробки покращується якість певного продукту, що дозволяє підвищити ціну його реалізації на 10 тис. грн. Кількість користувачів також збільшиться: протягом першого року – на 10, протягом другого року – ще на 40, протягом третього року – ще на 50.

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 500 користувачів, а її ціна – 50 тис. грн.

Потрібно спрогнозувати збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Припустимо, що наша розробка буде користуватися підвищеним попитом на ринку протягом 3-років після впровадження.

Збільшення чистого прибутку підприємства $\Delta П_i$ протягом першого року складе:

$$\Delta П_1 = (10000 \cdot 500 + 60000 \cdot 10) \cdot 0{,}8333 \cdot 0{,}2 \cdot \left(1 - \frac{18}{100}\right) = 765303 \text{ (грн)}.$$

Збільшення чистого прибутку $\Delta П_2$ від можливої реалізації даної розробки протягом другого року складе:

$$\Delta П_2 = \left(10000 \cdot 500 + 60000 \cdot (10 + 40)\right) \cdot 0{,}8333 \cdot 0{,}2 \cdot \left(1 - \frac{18}{100}\right) = \\ = 1093289 \text{ (грн)}.$$

Збільшення чистого прибутку $\Delta П_3$ від можливої реалізації даної розробки протягом третього року складе:

$$\Delta П_3 = \left(10000 \cdot 500 + 60000 \cdot (10 + 40 + 50)\right) \cdot 0{,}8333 \cdot 0{,}2 \cdot \left(1 - \frac{18}{100}\right) = \\ = 1503273 \text{ (грн)}.$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Розрахований у п. 4.3 комерційний ефект від можливого впровадження розробок ще не означає, що ця розробка реально буде впроваджена. Якщо збільшення прогнозованого прибутку від впровадження результатів наукової розробки є вигідним для підприємства, то це ще не означає, що інвестор погодиться фінансувати дану розробку. Інвестор погодиться вкладати кошти у реалізацію даної наукової розробки тільки за певних умов.

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

1-й крок. Розраховуємо теперішню вартість інвестицій PV, що вкладаються в наукову розробку. Такою вартістю ми можемо вважати прогнозовану величину загальних витрат ЗВ на виконання та впровадження результатів НДДКР, розраховану нами раніше за формулою (4.10), тобто будемо вважати, що ЗВ = PV = 381650.грн.

2-й крок. Розраховуємо очікуване збільшення прибутку $\Delta П_i$, що його отримає підприємство від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку також було розраховане нами раніше за формулою (4.11).

3-й крок. Для спрощення подальших розрахунків будують вісь часу, на яку наносять всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів. Характеристику руху платежів (інвестицій та додаткових прибутків) наведено на рис. 4.1.

Рисунок 4.1 - Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4-й крок. Розраховуємо абсолютну ефективність вкладених інвестицій $Е_{абс}$. Для цього використаємо формулу:

$$Е_{абс} = (ПП – PV), \qquad (4.10)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій PV = ЗВ, грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_{1}^{т} \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

Якщо $Е_{абс} < 0$, то результат від проведення наукових досліджень та їх впровадження буде збитковим і вкладати кошти в проведення цих досліджень ніхто не буде.

Якщо $Е_{абс} > 0$, то результат від проведення наукових досліджень та їх впровадження принесе прибуток, але це також ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даного проекту.

$$ПП = \frac{765303}{(1+0{,}1)^2} + \frac{1093289}{(1+0{,}1)^3} + \frac{1503273}{(1+0{,}1)^4} = 2480642 \text{ (грн).}$$

Тоді, абсолютна ефективність вкладених інвестицій складе:

$$Е_{абс} = 2480642 - 158864{,}4 = 2321777{,}6 \text{ (грн).}$$

Оскільки $Е_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

5-й крок. Розраховуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього використаємо формулу:

$$E_в = \sqrt[Т_ж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.12)$$

де $E_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;

PV –теперішня вартість інвестицій PV = ЗВ, грн;

$Т_ж$ – життєвий цикл наукової розробки, роки.

$$E_в = \sqrt[4]{1 + \frac{2321777,6}{158864,4}} - 1 = 0,94 \approx 94\%.$$

Далі, розрахована величина $E_в$ порівнюється з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$\tau = d + f, \quad (4.13)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2020 році в Україні d = (0,14...0,2);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,1), але може бути і значно більше.

Якщо величина $E_в > \tau_{мін}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки. В іншому випадку фінансування наукової розробки здійснюватися не буде.

$$\tau_{мін} = 0,2 + 0,1 = 0,3,$$

$$\tau_{мін} = 30\%.$$

Оскільки $E_в = 59\% > \tau_{мін} = 30\%$, то потенційний інвестор буде зацікавлений у фінансуванні даної наукової розробки.

6-й крок. Розраховуємо термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ можна розрахувати за формулою 4.14:

$$T_{ок} = \frac{1}{E_в}, \quad (4.14)$$

$$T_{ок} = \frac{1}{0{,}94} = 1{,}07 \text{ (років)}.$$

Оскільки $T_{ок} < 3…5$-ти років, то фінансування даної наукової розробки в принципі є доцільним.

4.5 Висновки

Оцінено комерційний потенціал розробки, рівень якого є вище середнього, що свідчить про доцільність ймовірного впровадження системи та отримання прибутку від її використання. Розраховано витрати на виконання наукової роботи та впровадження її результатів. Сума всіх статей витрат складає 39711,6 грн, до яких входять витрати на: основну заробітну плату розробників – 9800 грн; додаткову ЗП – 980 грн; нарахування на ЗП – 2371,6 грн; амортизація обладнання – 2800 грн; витрати на силову електроенергію – 100 грн;. Загальна вартість всієї наукової роботи складає 79432,2 грн. Розраховано комерційні ефекти від реалізації результатів розробки. Позитивні результати від провадження розробки очікуються протягом трьох років після впровадження. Збільшення чистого прибутку протягом 1-го року складає – 765 303. грн, протягом 2-го року – 1 093 289 грн, протягом 3 – го року – 1 503 273 грн. Розраховано ефективність вкладених інвестицій та період їх окупності. Абсолютна ефективність вкладених інвестицій $E_{абс} > 0$, свідчить про те, що впровадження наукових досліджень принесе прибуток, а розрахована відносна (щорічна) ефективність вкладених в

наукову розробку інвестицій $E_в$ порівнюється з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, так як $E_в > \tau_{мін}$, то потенційний інвестор буде зацікавлений у фінансуванні даної наукової розробки. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок} = 1,07$ років, звідси $T_{ок} < 3 .. 5$ років, що свідчить про доцільність фінансування даної наукової розробки.

ВИСНОВКИ

В роботі запропонована модель прогнозування наявності опадів, яка на відміну від існуючих визначає інформативні ознаки впливу, на основі яких здійснюється прогнозування наявності опадів за рахунок використання алгоритмів машинного навчання.

В розділі 1 було здійснено огляд та аналіз історії розвитку прогнозування прогодних умов з часів народних прикмет до заснування перших математично обґрунтованих алгоритмів та створення спеціалізованої обчислювальної машини, прикладів відкритих даних на прикладі датасетів системи Kaggle та галузі аналізу даних. В тому число було сформульовано проблему даної галузі в Україні. Також було наведено стислий огляд основних технологій як синоптичних, так і комп'ютерних, що застосовуються фахівцями даної галузі, відкритих наборів погодних даних та принцип роботи аналізу даних.

В розділі 2 було проведено огляд вхідних даних, їх аналіз та розроблено об'єднаний файл для подальшого використання, на основі проведеного групування всіх вагомих параметрів, для подальшої роботи дані було завантажено в платформу Kaggle, також було наведено список атрибутів датасету для подальшої розробки методики реалізації інформаційної технології. Наведено короткий опис сфери машинного навчання для більш глибокого розуміння галузі, в якій буде проводитись реалізація нової технології передбачення. На основі огляду мови програмування та бібліотек ML було розроблено методику реалізації технології аналізу та передбачення опадів за допомогою інструментів, які найбільше підходять для рішення поставленої задачі.

В розділі 3 було проведено аналіз атрибутів датасету, що використовується для реалізації технології прогнозування, та сконструйовано перелік головних ознак з найбільшим впливом на результат передбачення опадів. Серед п'яти моделей (GradientBoosting, AdaBoost, Ridge, Bagging, MLP) обрано найточнішу модель прогнозування та наведено графік їх порівнянь. Використовуючи обрані

найвагоміші атрибути та найточнішу модель передбачення, було проведено тестування та отримано результат точності, який склав 78.9%.

В розділі 4 було оцінено комерційний потенціал розробки, рівень якого є вище середнього, що свідчить про доцільність ймовірного впровадження системи та отримання прибутку від її використання. Розраховано витрати на виконання наукової роботи та впровадження її результатів. Розраховано комерційні ефекти від реалізації результатів розробки. Розраховано ефективність вкладених інвестицій та період їх окупності.

За результатами магістерської кваліфікаційної роботи опубліковано тези доповіді на XV Міжнародній конференції "Контроль і управління в складних системах" (КУСС-2020). Отже, поставлені завдання магістерської кваліфікаційної роботи були виконані в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Синоптичний аналіз і прогноз [Електронний ресурс]. – Режим доступу: https://geography.lnu.edu.ua/wp-content/uploads/2017/09/Prak9.pdf.

2. История прогноза погоды [Електронний ресурс]. – Режим доступу: https://habr.com/ru/post/179687/.

3. Как машины предсказывают нам погоду [Електронний ресурс]. – Режим доступу: https://vas3k.ru/blog/how_to_weather/.

4. Інформаційна технологія аналізу та передбачення опадів / О. Л. Мельник, М. В. Дратований, О. М. Козачко // Матеріали XV міжнародної конференції "Контроль і управління в складних системах (КУСС-2020)", м. Вінниця, 8-10 жовтня 2020 р.

5. Як роблять прогнози погоди і чому вони іноді не збуваються? прогноз [Електронний ресурс]. – Режим доступу: https://www.bbc.com/ukrainian/features-51545290.

6. Прогнози погоди і клімату та притаманні їм обмеження [Електронний ресурс]. – Режим доступу: http://prima.franko.lviv.ua/faculty/geology/phis_geo/fourman/E-books-FVV/Interactive%20books/Meteorology/Weather%20Forecasting/Weather%20Ukraine/Meteo-forecasting/Analyze-forecast%20of%20limits%20climate.htm.

7. Прогнозування погоди [Електронний ресурс]. – Режим доступу: http://prima.franko.lviv.ua/faculty/geology/phis_geo/fourman/E-books-FVV/Interactive%20books/Meteorology/ForecaseM.htm.

8. Прогноз погоды [Електронний ресурс]. – Режим доступу: https://ru.qaz.wiki/wiki/Weather_forecasting.

9. О технологиях долгосрочных прогнозов прогноз [Електронний ресурс]. – Режим доступу: http://seakc.meteoinfo.ru/training.

10. День сурка. Как новые технологии изменили прогноз погоды [Електронний ресурс]. – Режим доступу: https://club.esetnod32.ru/articles/analitika/den-surka/.

11. Climate Change: Earth Surface Temperature Data (Exploring global temperatures since 1750) [Електронний ресурс]. – Режим доступу:

https://www.kaggle.com/berkeleyearth/climate-change-earth-surface-temperature-data/.

12. Rain in Australia (Predict rain tomorrow in Australia) [Електронний ресурс]. – Режим доступу: https://www.kaggle.com/jsphyg/weather-dataset-rattle-package/.

13. Методи аналізу даних [Електронний ресурс]. – Режим доступу: https://books.google.com.ua/books?id=FwniCAAAQBAJ&pg=PA5&lpg=PA5&dq=#v=onepage&q&f=false/.

14. Основи статистики та аналізу даних - Машинне навчання [Електронний ресурс]. – Режим доступу: https://socialdata.org.ua/manual4/.

15. Условия, влияющие на распределение атмосферных осадков [Електронний ресурс]. – Режим доступу: MCXAhttps://www.activestudy.info/usloviya-vliyayushhie-na-raspredelenie-atmosfernyx-osadkov/.

16. Огляд методів машинного навчання для класифікації великих обсягів супутникових даних [Електронний ресурс]. – Режим доступу: http://journal.iasa.kpi.ua/article/download/114466/123508/.

17. Python – краткий обзор языка и его назначения [Електронний ресурс]. – Режим доступу: https://techrocks.ru/2019/01/21/about-python-briefly/.

18. Библиотеки Python, необходимые для машинного обучения [Електронний ресурс]. – Режим доступу: https://techrocks.ru/2018/10/05/python-libraries-for-machine-learning/.

19. Classification in Python with Scikit-Learn and Pandas [Електронний ресурс]. – Режим доступу: https://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/.

20. Методичні вказівки до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 –«Інформаційні системи та технології» денної форми навчання / Уклад. В. Б. Мокін, С. О. Жуков, А. Р. Ящолт, О. М. Козачко, Л. М. Скорина. – Електронне видання. – Вінниця : ВНТУ, 2018. – 48 с.

Додаток А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. В.Б. Мокін
(підпис)
"__" _____ 2020

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПЕРЕДБАЧЕННЯ ОПАДІВ»

08–53.МКР.004.02.00.ТЗ

Керівник магістерської кваліфікаційної роботи
к.т.н., доц.
_____ О. М. Козачко
(підпис)
"__" _____ 2020 р.

Розробила студентка гр. 2ІСТ-19м
_____ О. Л. Мельник
(підпис)
"__" _____ 2020 р.

Вінниця 2020

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № __ по ВНТУ від «__» _____ 2020 р., та індивідуальне завдання на МКР, затверджене протоколом № __ засідання кафедри САІТ від «__» _____ 2020 р.

2. Джерела розробки:

– Синоптичний аналіз і прогноз [Електронний ресурс]. – Режим доступу: https://geography.lnu.edu.ua/wp-content/uploads/2017/09/Prak9.pdf. – Назва з екрана;

– Основи статистики та аналізу даних - Машинне навчання [Електронний ресурс]. – Режим доступу: https://socialdata.org.ua/manual4/. – Назва з екрана;

– Інформаційна технологія аналізу та передбачення опадів / О. Л. Мельник, М. В. Дратований, О. М. Козачко // Матеріали XV міжнародної конференції "Контроль і управління в складних системах (КУСС-2020)", м. Вінниця, 8-10 жовтня 2020 р.

3. Мета і призначення роботи

Підвищення точності прогнозування наявності опадів за рахунок використання інформаційних технологій, машинного навчання та аналітичної обробки даних.

4. Вихідні дані для проведення робіт:

– дані погодних умов м. Вінниця в період з 2010 по 2020 роки Вінницького обласного центру з гідрометеорології.

5. Методи дослідження:

– методи машинного навчання;
– регресійні моделі прогнозування.

6. Етапи роботи і терміни їх виконання

a) Аналіз предметної області ..... __. – __
b) Огляд існуючих технологій аналізу та передбачення опадів __ – __
c) Розробка технології аналізу та передбачення опадів..... __ – __
d) Реалізація технології аналізу та передбачення опадів. ..... __ – __

7. Очікувані результати та порядок реалізації

Отримання програмного забезпечення інформаційної технології аналізу та передбачення опадів.

8. Вимоги до розробленої документації

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 – «Інформаційні системи та технології» денної форми навчання».

9. Порядок приймання роботи

Публічний захист «___» _____ 2020 р.
Початок розробки «___» _____ 2020 р.
Граничні терміни виконання МКР «___» _____2020 р.

Розробила студентка групи 2ІСТ-19м _____ Мельник О. Л.

Додаток Б

Лістинг програми

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
import eli5

import lightgbm as lgbm
import xgboost as xgb

from sklearn import preprocessing
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LinearRegression, RidgeCV
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.ensemble import GradientBoostingRegressor

train_data = pd.read_csv('../input/weather-2010/weatherSet.csv')
test_data = pd.read_csv('../input/weather-2010/weatherSet.csv')

train_data.info()

train_data.head()

cols_to_remove = ['date']
train_data.drop(cols_to_remove, axis=1, inplace=True)

test_data.drop(cols_to_remove, axis=1, inplace=True)

train_data = train_data.dropna(how='any')
test_data = test_data.dropna(how='any')

numerics = ['int8', 'int16', 'int32', 'int64', 'float16', 'float32', 'float64']
categorical_columns = []
features = train_data.columns.values.tolist()
for col in features:
    if train_data[col].dtype in numerics: continue
    categorical_columns.append(col)
indexer = {}
for col in categorical_columns:
    if train_data[col].dtype in numerics: continue
    _, indexer[col] = pd.factorize(train_data[col])

for col in categorical_columns:
    if train_data[col].dtype in numerics: continue
    train_data[col] = indexer[col].get_indexer(train_data[col])

train_data.head()
train_data.info()

corr = train_data.corr()
fig = plt.figure(figsize=(15,10))
sns.heatmap(corr)
```

```
corr.sort_values(by=["precipitation"],ascending=False).iloc[0].sort_values(ascendin
g=False)

analysis_data = train_data.copy()
analysis_data['precipitation'] = analysis_data['precipitation'].where(analysis_data
['precipitation'] == 0, 1)

analysis_data = train_data.copy()
analysis_data['precipitation'] = analysis_data['precipitation'].where(analysis_data
['precipitation'] == 0, 1)

plt.figure(figsize=(8,8))
sns.FacetGrid(analysis_data, hue="precipitation", size=8).map(sns.kdeplot, "minHumi
dity").add_legend()
plt.ioff()
plt.show()

plt.figure(figsize=(8,8))
sns.countplot(data=train_data,x='lowCloud')

plt.figure(figsize=(8,8))
sns.countplot(data=train_data,x='cloud')

y = train_data['precipitation']
del train_data['precipitation']

X = train_data;

# data split
X_train, X_valid, y_train, y_valid = train_test_split(X, y, train_size=0.8, test_si
ze=0.2, random_state=0)

train_set = lgbm.Dataset(X_train, y_train, silent=False)
valid_set = lgbm.Dataset(X_valid, y_valid, silent=False)

params = {
        'boosting_type':'gbdt',
        'objective': 'regression',
        'num_leaves': 31,
        'learning_rate': 0.05,
        'max_depth': -1,
        'subsample': 0.8,
        'bagging_fraction' : 1,
        'max_bin' : 5000 ,
        'bagging_freq': 20,
        'colsample_bytree': 0.6,
        'metric': 'rmse',
        'min_split_gain': 0.5,
        'min_child_weight': 1,
        'min_child_samples': 10,
        'scale_pos_weight':1,
        'zero_as_missing': True,
        'seed':0,
    }

modelL = lgbm.train(params, train_set = train_set, num_boost_round=1000,
                    early_stopping_rounds=50,verbose_eval=10, valid_sets=valid_set)

fig =  plt.figure(figsize = (15,15))
```

```
axes = fig.add_subplot(111)
lgbm.plot_importance(modelL,ax = axes,height = 0.5)
plt.show();plt.close()

feature_score = pd.DataFrame(X.columns, columns = ['feature'])
feature_score['score_lgb'] = modelL.feature_importance()

data_tr  = xgb.DMatrix(X_train, label=y_train)
data_cv  = xgb.DMatrix(X_valid   , label=y_valid)
data_train = xgb.DMatrix(X)
data_test  = xgb.DMatrix(y)
evallist = [(data_tr, 'train'), (data_cv, 'valid')]

parms = {'max_depth':8, #maximum depth of a tree
         'eta'      :0.3,
         'subsample':0.8,#SGD will use this percentage of data
         'lambda '  :4, #L2 regularization term,>1 more conservative
         'colsample_bytree ':0.9,
         'colsample_bylevel':1,
         'min_child_weight': 10}
modelx = xgb.train(parms, data_tr, num_boost_round=200, evals = evallist,
                 early_stopping_rounds=30, maximize=False,
                 verbose_eval=10)

fig =  plt.figure(figsize = (15,15))
axes = fig.add_subplot(111)
xgb.plot_importance(modelx,ax = axes,height = 0.5)
plt.show();plt.close()

feature_score['score_xgb'] = feature_score['feature'].map(modelx.get_score(importan
ce_type='weight'))

# Linear Regression

linreg = LinearRegression()
linreg.fit(X, y)
coeff_linreg = pd.DataFrame(X.columns)
coeff_linreg.columns = ['feature']
coeff_linreg["score_linreg"] = pd.Series(linreg.coef_)
coeff_linreg.sort_values(by='score_linreg', ascending=False)

eli5.show_weights(linreg)

coeff_linreg["score_linreg"] = coeff_linreg["score_linreg"].abs()

feature_score = pd.merge(feature_score, coeff_linreg, on='feature')
feature_score = feature_score.fillna(0)
feature_score = feature_score.set_index('feature')
feature_score

feature_score = pd.DataFrame(
    preprocessing.MinMaxScaler().fit_transform(feature_score),
    columns=feature_score.columns,
    index=feature_score.index
)

# Create mean column
feature_score['mean'] = feature_score.mean(axis=1)
```

```
# Plot the feature importances
feature_score.sort_values('mean', ascending=False).plot(kind='bar', figsize=(20, 10
))

feature_score['total'] = 0.48 * feature_score['score_lgb'] + 0.48 * feature_score['
score_xgb'] + 0.04 * feature_score['score_linreg']

# Plot the feature importances
feature_score.sort_values('total', ascending=False).plot(kind='bar', figsize=(20, 1
0))

feature_score.sort_values('total', ascending=False)

feature_columns = ['minHumidity', 'maxWind', 'minTemp', 'lowCloud',  'cloud']
X = X[feature_columns];

X.head()

# data split for train
X_train, X_valid, y_train, y_valid = train_test_split(X, y, train_size=0.8, test_si
ze=0.2, random_state=0)

linear = LinearRegression()
linear.fit(X_train, y_train)

results = []
result = linear.score(X_valid, y_valid)
results.append(result)

print(result)

gradient_boost = GradientBoostingRegressor()
gradient_boost.fit(X_train, y_train)

result = gradient_boost.score(X_valid, y_valid)
results.append(result)

print(result)

ridge_cv = RidgeCV()
ridge_cv.fit(X_train, y_train)

result = ridge_cv.score(X_valid, y_valid)
results.append(result)

print(result)

x = np.arange(3)

fig, ax = plt.subplots()
plt.bar(x, results)
ax.set_ylim(bottom=0)
plt.xticks(x, ('Linear', 'GradientBoosting', 'RidgeCV'))
plt.show()

clss = GradientBoostingRegressor()
clss.fit(X, y)

data = test_data.iloc[20]
```

```
condition = data[feature_columns]
result = data['precipitation']

condition

pred = clss.predict([condition])

print(pred)
print(result
```

Додаток В

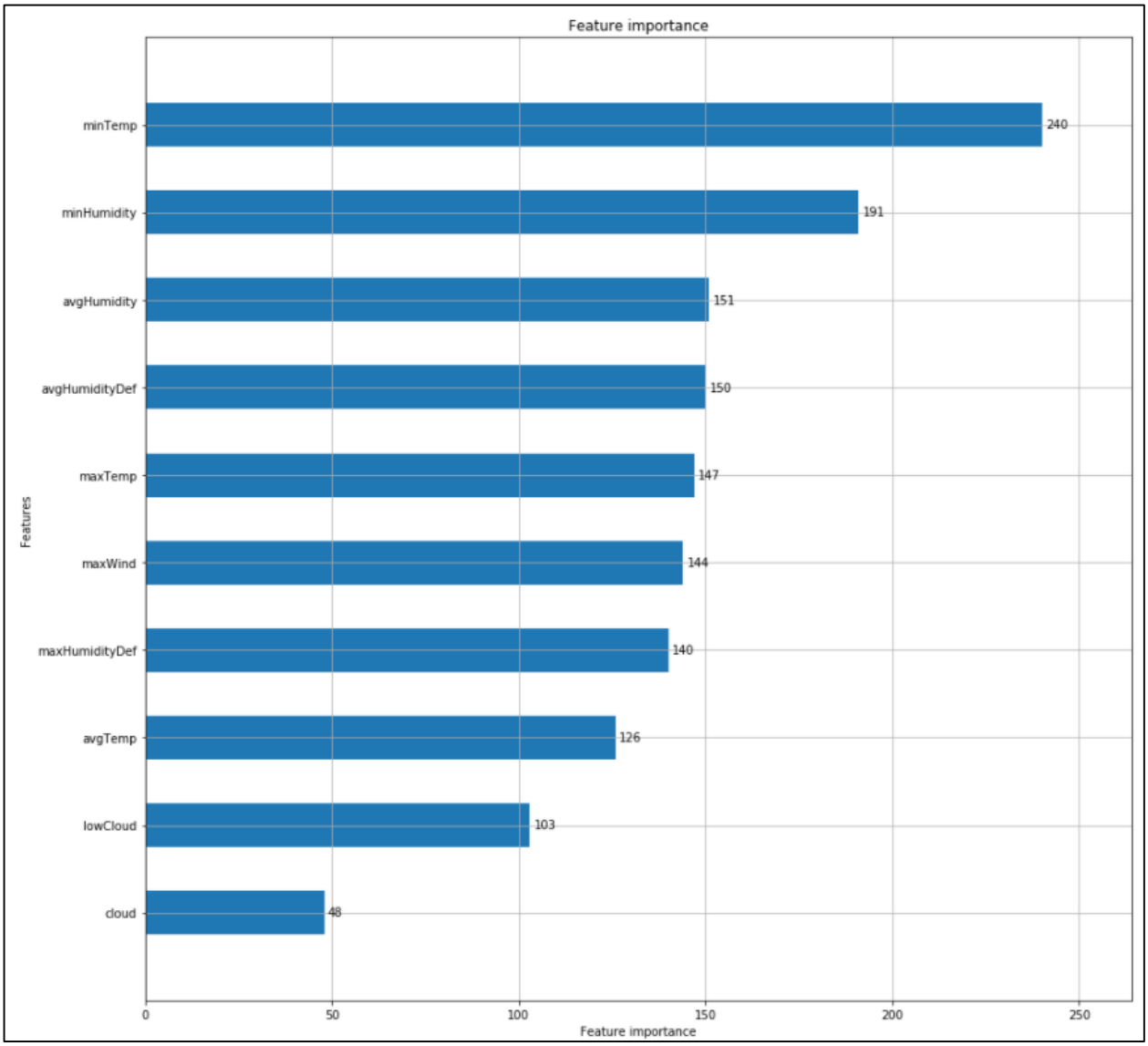
Графічна частина

Дані після видалення зайвих записів

	avgTemp	maxTemp	minTemp	avgHumidity	minHumidity	avgHumidityDef	maxHumidityDef	cloud	lowCloud	maxWind	precipitation
count	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000	3370.000000
mean	9.476460	14.177935	5.181923	73.131869	56.161128	4.984671	9.704332	6.463205	3.858457	9.873294	0.394659
std	9.963356	11.441242	8.876597	14.257363	19.347193	4.603779	8.564911	2.855579	3.123722	3.304965	0.488850
min	-23.400000	-19.400000	-28.500000	29.000000	16.000000	0.000000	0.000000	0.000000	0.000000	3.000000	0.000000
25%	1.600000	4.500000	-0.800000	63.000000	41.000000	1.100000	2.100000	4.000000	1.000000	7.000000	0.000000
50%	10.400000	15.400000	6.000000	74.000000	53.000000	3.400000	7.400000	7.000000	3.000000	9.000000	0.000000
75%	18.100000	24.100000	12.500000	85.000000	72.000000	7.900000	15.500000	9.000000	6.000000	11.000000	1.000000
max	28.000000	37.300000	22.900000	100.000000	100.000000	23.800000	47.700000	10.000000	10.000000	28.000000	1.000000

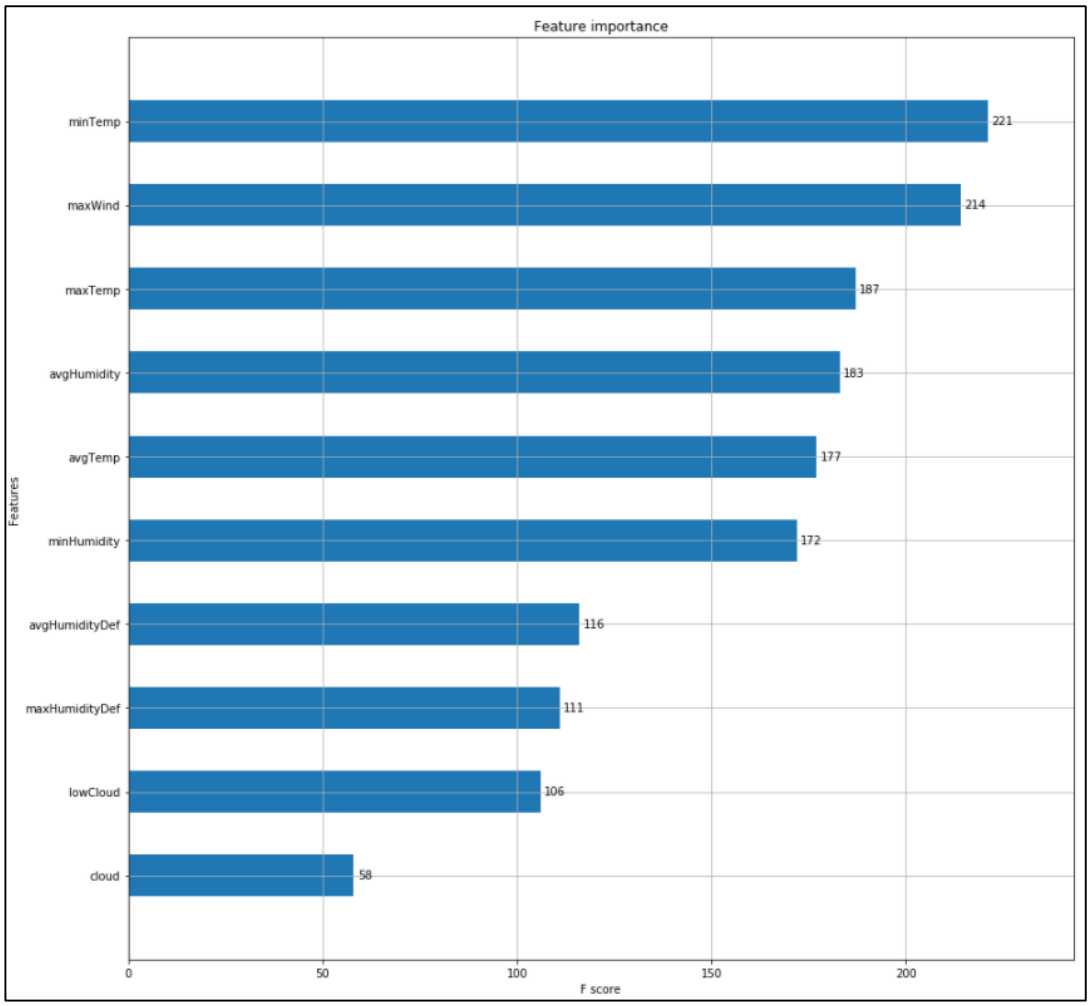
					08-53.МКР.004.02.000.ПЛ	
					Інформаційна технологія аналізу та передбачення опадів	Літ. / Маса / Масштаб: 1 : 1
Змн.	Арк.	№ докум.	Підпис	Дата		Арк. 1 / Аркушів 8
Розроб.		Мельник О. Л.			Дані після видалення зайвих записів	2ІСТ-19м
Перевір.		Козачко О. М.				
Т. Контр.						
Реценз.		Бойко О. Р.				
Н. Контр.		Жуков С. О.				
Затверд.		Мокін В. Б.				

Діаграма важливості атрибутів згідно результатів моделі lgbm



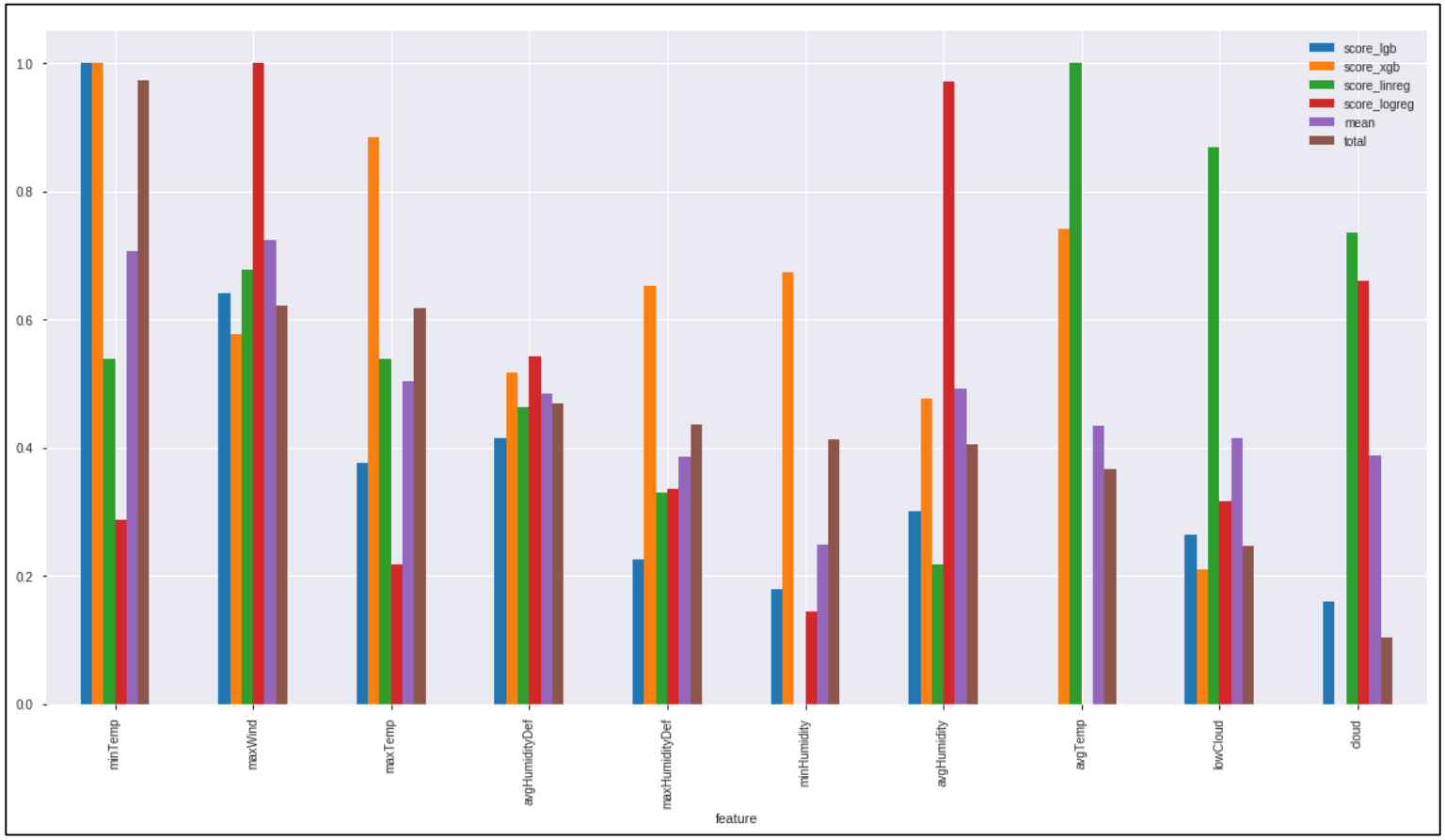
					08-53.МКР.004.02.000.ПЛ	
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна технологія аналізу та передбачення опадів	Літ. / Маса / Масштаб: 1 : 1
Розроб.		Мельник О. Л.				
Перевір.		Козачко О. М.				Арк. 2 / Аркушів 8
Т. Контр.						
Реценз.		Бойко О. Р.			Діаграма важливості атрибутів згідно результатів моделі lgbm	2ІСТ-19м
Н. Контр.		Жуков С. О.				
Затверд.		Мокін В. Б.				

Діаграма важливості атрибутів згідно результатів моделі xgb



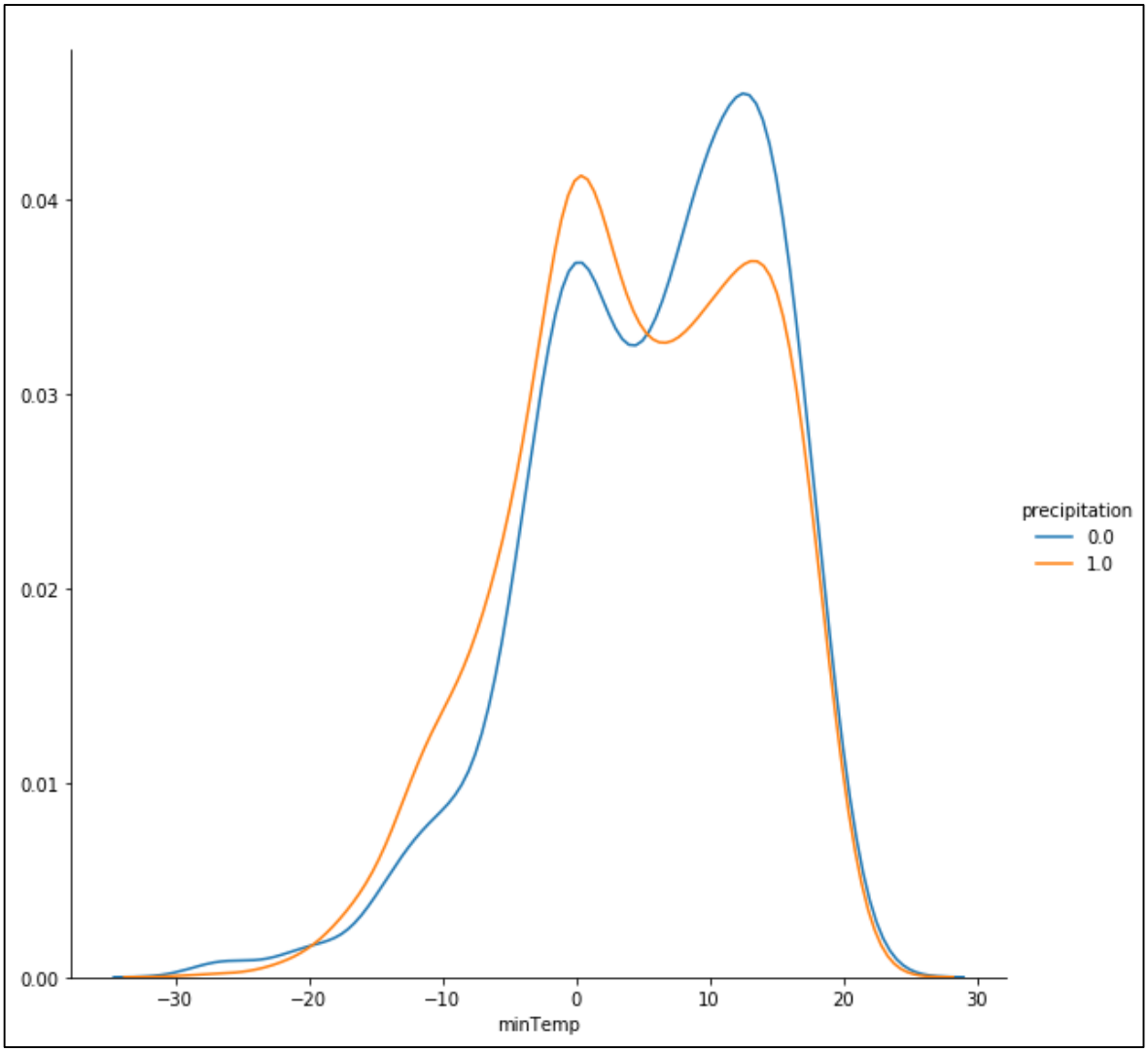
					08-53.МКР.004.02.000.ПЛ	
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна технологія аналізу та передбачення опадів	Літ. / Маса / Масштаб
Розроб.		Мельник О. Л.				1 : 1
Перевір.		Козачко О. М.				
Т. Контр.						Арк. 3 / Аркушів 8
Реценз.		Бойко О. Р.			Діаграма важливості атрибутів згідно результатів моделі xgb	2ІСТ-19м
Н. Контр.		Жуков С. О.				
Затверд.		Мокін В. Б.				

Діаграма значень коефіцієнтів важливості атрибутів



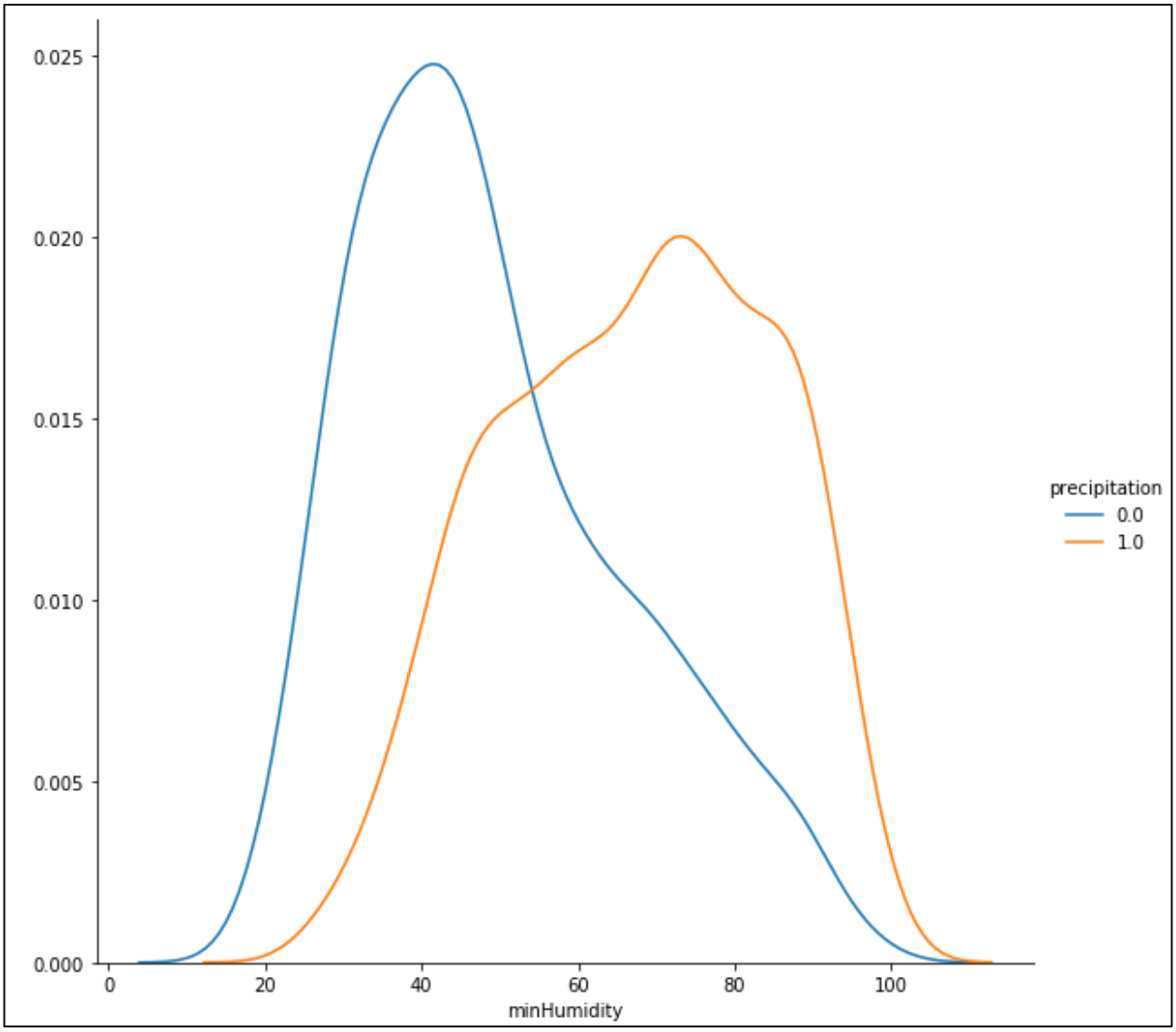
					08-53.МКР.004.02.000.ПЛ		
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна технологія аналізу та передбачення опадів	Літ.	Маса / Масштаб
Розроб.		Мельник О. Л.					1 : 1
Перевір.		Козачко О. М.					
Т. Контр.						Арк. 4	Аркушів 8
Реценз.		Бойко О. Р.			Діаграма значень коефіцієнтів важливості атрибутів	2ІСТ-19м	
Н. Контр.		Жуков С. О.					
Затверд.		Мокін В. Б.					

Графік залежності опадів від значень температури



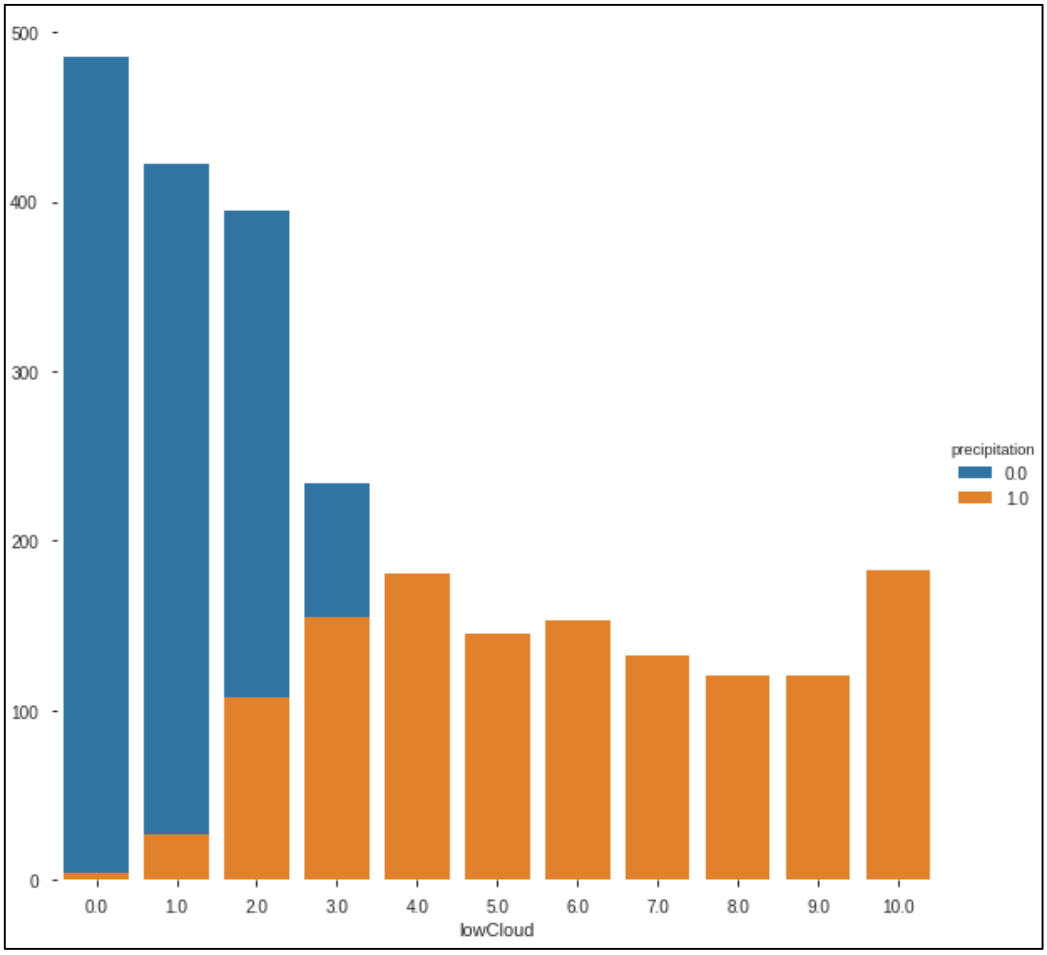
					08-53.МКР.004.02.000.ПЛ		
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна технологія аналізу та передбачення опадів	Літ.	Маса / Масштаб
Розроб.		Мельник О. Л.					1 : 1
Перевір.		Козачко О. М.				Арк. 5	Аркушів 8
Т. Контр.							
Реценз.		Бойко О. Р.			Графік залежності опадів від значень температури	2ІСТ-19м	
Н. Контр.		Жуков С. О.					
Затверд.		Мокін В. Б.					

Графік залежності опадів від значень вологості



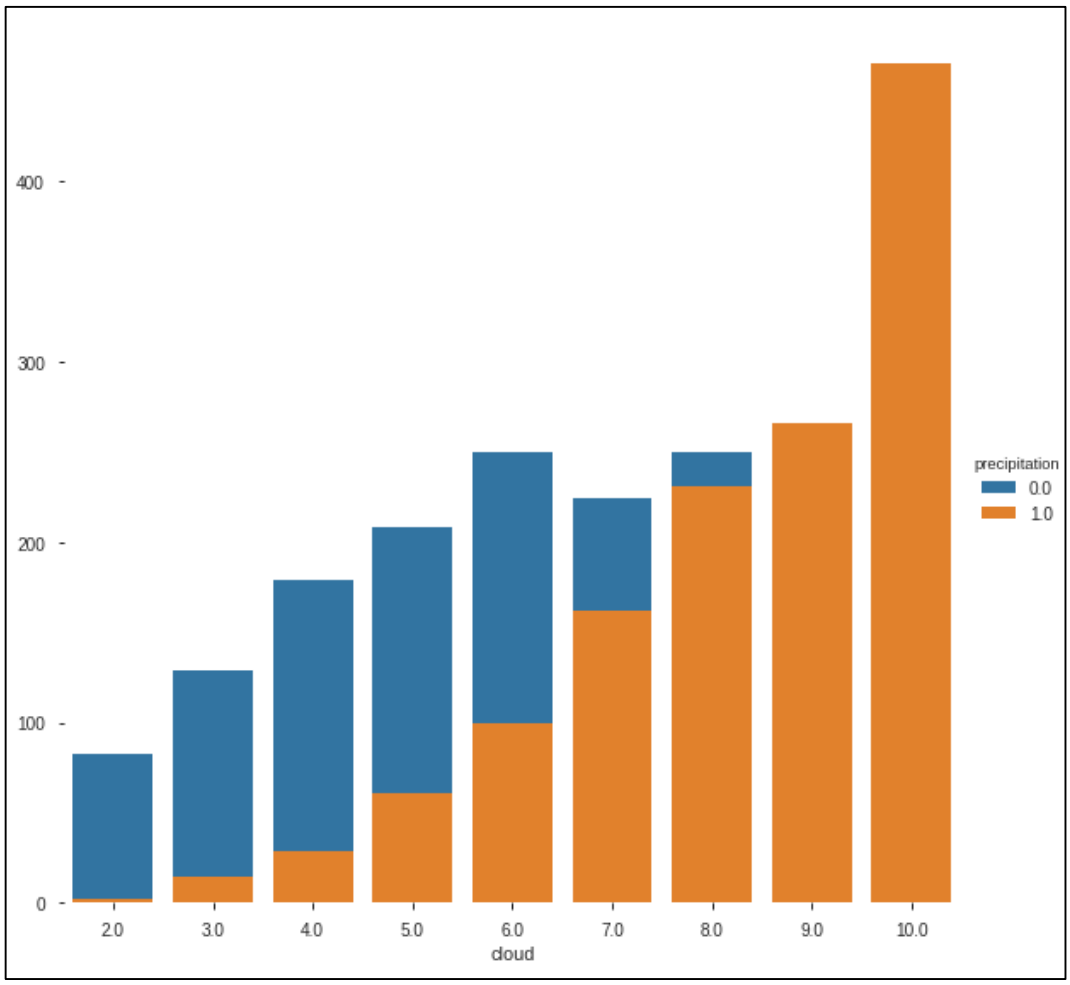
					08-53.МКР.004.02.000.ПЛ		
					Інформаційна технологія аналізу та передбачення опадів	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата				1 : 1
Розроб.		Мельник О. Л.				Арк. 6	Аркушів 8	
Перевір.		Козачко О. М.						
Т. Контр.								
Реценз.		Бойко О. Р.			Графік залежності опадів від значень вологості	2ІСТ-19м		
Н. Контр.		Жуков С. О.						
Затверд.		Мокін В. Б.						

Діаграма кількості значень нижнього показника хмарності



					08-53.МКР.004.02.000.ПЛ		
					Інформаційна технологія аналізу та передбачення опадів	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата				1 : 1
Розроб.		Мельник О. Л.				Арк. 7	Аркушів 8	
Перевір.		Козачко О. М.						
Т. Контр.					Діаграма кількості значень нижнього показника хмарності	2ІСТ-19м		
Реценз.		Бойко О. Р.						
Н. Контр.		Жуков С. О.						
Затверд.		Мокін В. Б.						

Діаграма кількості значень показника хмарності



					08-53.МКР.004.02.000.ПЛ		
					Інформаційна технологія аналізу та передбачення опадів	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата				1 : 1
Розроб.		Мельник О. Л.				Арк. 8	Аркушів 8	
Перевір.		Козачко О. М.						
Т. Контр.					Діаграма кількості значень показника хмарності	2ІСТ-19м		
Реценз.		Бойко О. Р.						
Н. Контр.		Жуков С. О.						
Затверд.		Мокін В. Б.						