Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг»

Виконав: студент 2 курсу, групи 2ІСТ-21м
спеціальності 126 – «Інформаційні системи та технології»
_____ Головач В.Р.

Керівник: к.т.н., доц. каф. САІТ
_____Козачко О. М.
«01» _____12_____ 2022 р.

Опонент: д.т.н., проф. каф. КСУ
_____ Ковтун В.В.
«16» _____12_____ 2022 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«05» _____12_____ 2022 р.

м. Вінниця – 2022 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – ІІ-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 126 Інформаційні системи та технології
Освітньо-професійна програма – Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«16» 09 2022 р.

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Головачу Віталію Руслановичу

1. Тема роботи: «Інформаційна технологія прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг»
керівник роботи: Козачко О. М., к.т.н., доц. каф. САІТ,
затверджено наказом закладу вищої освіти від "14" 09 2022 року №203
2. Строк подання студентом роботи «01» 12 2022 року
3. Вихідні дані до роботи:
   - Дані датасету «River Water Quality EDA and Forecasting» платформи Kaggle;
   - Електронна карта Вінницької області.
4. Зміст текстової частини:
   - Прогнозування змін концентрації розчиненого у воді кисню у річках;
   - Вибір оптимальних інформаційних технологій та розвідувальний аналіз даних;
   - Розроблення інформаційної технології та ідентифікація оптимальної моделі для прогнозування;
   - Застосування розробленої інформаційної технології прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг;
   - Економічна частина.
5. Перелік ілюстративного матеріалу:
   - Блок-схема алгоритму інформаційної технології прогнозування;
   - Діаграма важливості ознак;
   - Діаграма перестанови з важливістю ознак;
   - Матриця кореляції;
   - Графік прогнозу валідаційних даних;
   - Графік реальних та спрогнозованих даних за оптимальною моделлю.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
2-3	Мокін В.Б., д.т.н., проф. каф. САІТ	01.10.22р.	30.10.22р.
5	Бурєннікова Н.В., д.е.н., проф. каф. ЕМВМ	15.11.22р.	29.11.22р.

7. Дата видачі завдання «16» 09 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Прогнозування змін концентрації розчиненого у воді кисню у річках	09.2022	
2	Пошук набору даних для дослідження	10.2022	
3	Вибір оптимальних інформаційних технологій та розвідувальний аналіз даних	10.2022	
4	Розроблення інформаційної технології та ідентифікація оптимальної моделі для прогнозування	10.2022	
5	Застосування розробленої інформаційної технології прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг	11.2022	
6	Навчання та перевірка моделей	11.2022	
7	Розробка економічної частини	11.2022	
8	Оформлення матеріалів до захисту МКР	11.2022	

Студент _____ Головач В.Р.

Керівник роботи _____ Козачко О. М.

АНОТАЦІЯ

УДК 004.09

Головач В.Р. Інформаційна технологія прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2022. 117 с.

На укр. мові. Бібліогр.: 21 назв; рис.: 104; табл.: 6.

Магістерська кваліфікаційна робота присвячена моніторингу якості води у річці Південний Буг. В роботі запропоновано технологія, яка аналізує та прогнозує вміст розчиненого у воді кисню у річці Південний Буг. Об’єктом дослідження є процес прогнозування вмісту розчиненого у воді кисню в річці Південний Буг. Галузь застосування – екологічні установи та організації, які займаються аналізом якості водних ресурсів.

Ілюстративна частина – складається з 6 плакатів.

У розділі економічної частини розглянуто питання про доцільність розробки та впровадження інформаційної технології прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг.

Ключові слова: розчинений кисень, інформаційна система, технологія, прогнозування, якість води, кількісний вміст компонентів.

ABSTRACT

Holovach V.R. Information technology for forecasting changes in the concentration of oxygen dissolved in water in the South Bug River. Master's qualification thesis on specialty 126 - information systems and technologies, educational and professional program - information technologies of data and image analysis. Vinnytsia: VNTU, 2022. 117 p.

In Ukrainian speech Bibliography: 21 titles; fig.: 104; table: 6.

In the master's qualification thesis, attention is drawn to the problem of water quality monitoring in South Bug River. Proposed technologies that will help analyze and predict the content of oxygen dissolved in water in river water. Information technology for predicting dissolved oxygen in river water has been implemented. The object of the study is the process of forecasting the content of oxygen dissolved in water in the South Bug River. The field of application is environmental institutions and organizations engaged in the analysis of the quality of water resources.

The illustrative part consists of 6 posters.

In the section of the economic part, the issue of the feasibility of developing and implementing information technology for forecasting changes in the concentration of oxygen dissolved in water in the South Bug River is considered.

Key words: dissolved oxygen, information system, technology, forecasting, water quality, quantitative content of components.

ЗМІСТ

ВСТУП

Актуальність теми. На даний момент стан клімату та його неочікувані зміни можуть впливати на споживання кисню у річкових водах. Це призводить до непередбачуваних змін показників потреби кисню. При цих умовах, вода дуже швидко забруднюється і через природні чинники відбувається зміна споживання кисню води. Цей процес постійно пришвидшується.

При наявній державній системі моніторингу якості цих вод є можливість отримати доступ до накопичених моніторингових даних та спробувати з використанням сучасних інформаційних технологій проаналізувати та спрогнозувати стан речовин у річці Південний Буг. Усі ці процеси призводять до підтримки стану водних ресурсів у нормі.

Метою роботи є підвищення точності прогнозування вмісту розчиненого у воді кисню у річці Південий Буг, за рахунок використання методів машинного навчання.

У роботі планується вирішити наступні завдання:

- дослідити дані моніторингу якості вод у річці Південний Буг та провести аналіз (розвідувальний аналіз) цих даних для розчиненого у воді кисню;
- виконати аналіз даних та прогнозування розчиненого кисню у річковій воді;
- розробити інформаційну технологію для прогнозування розчиненого кисню у річковій воді.

Об’єктом дослідження є процес прогнозування вмісту розчиненого у воді кисню в річці Південний Буг.

Предмет дослідження – інформаційна технологія та методи аналізу та прогнозування значень вмісту розчиненого у воді кисню в річці Південний Буг.

Новизна одержаних результатів. Подальшого розвитку набула інформаційна технологія прогнозування вмісту розчиненого у воді кисню в

річці Південний Буг, яка дозволяє підвищити точність прогнозування за рахунок використання методів машинного навчання і на відміну від існуючих методів прогнозування, враховує аномальні значення показників.

Апробація результатів магістерської кваліфікаційної роботи. Результати роботи доповідались на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблема, перспективи» (Вінниця, 2022-2023 рр.).

Публікації результатів магістерської кваліфікаційної роботи. Опубліковано тези на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблема, перспективи» [1].

1 ПРОГНОЗУВАННЯ ЗМІН КОНЦЕНТРАЦІЇ РОЗЧИНЕНОГО У ВОДІ КИСНЮ У РІЧКАХ

1.1 Загальна характеристики об’єкту дослідження

Вода є одним з найважливіших і життєво необхідних ресурсів на землі. Вода є одним з основних компонентів людини оскільки людина складається на 60% з води. Вода відіграє велику роль в обміні речовин організму. Так і в цілому вона є невід’ємною частиною не тільки нашого життя, а ще й вода має величезне значення у промисловості та сільськогосподарському виробництві. Вона є необхідним для всіх живих організмів на землі, а також є середовищем існування. Але зростання промисловості, розширення території забудови міст, розвиток сільського господарства, значне розширення площ зрошуваних земель та ряд інших чинників з кожним роком все більше і більше загострюють проблему водного ресурсу.

Для екологічної оцінки водойм застосовуються різноманітні показники стану водних екосистем. Одним із них є кисневий режим – чутливий показник рівня забруднення природних вод [2].

Кисень використовується при розкладанні розчинених органічних речовин, відмерлих рослин і тварин. При цьому складні органічні речовини перетворюються на прості (CO2, воду, азот) і знову включаються в колообіг речовин у гідросфері. У воді можуть утворюватися сполуки кисню з іншими хімічними елементами. Більшість таких сполук (оксидів) прямо або опосередковано взаємодіє з водою, утворюючи гідроксиди, які належать до різних класів неорганічних сполук [2].

Тому не дивно, що вміст розчиненого у воді кисню входить в коло основних показників, що визначають поверхневі води як ресурс.

Ні одна система моніторингу якості води не обходиться без визначення концентрації кисню, ні одна екологічна модель не може бути досить повною, якщо в ній не враховується динаміка розчиненого кисню [2].

Прогнозування кисневого режиму водойм, що зазнають змін у гідрологічному режимі під час експлуатації гідротехнічних споруд, є складовою частиною екологічної оцінки. Гідрологічний режим визначає швидкість надходження кисню у воду з атмосфери, а також значно впливає на інтенсивність фотосинтезу та швидкість поглинання кисню донними відкладами. Тому він відіграє за будь-яких інших рівних умов визначальну роль у кисневому режимі водних об'єктів і розглядається в роботі як визначальний фактор для прогнозування кисневого режиму. Питання кількісної оцінки утворення кисню внаслідок фотосинтезу, а також визначення впливу життєдіяльності різних видів водних організмів на кисневий режим водних об'єктів з одним із найскладніших при математичному моделюванні процесів формування якості води, зокрема кисневого режиму. Тому ці питання потребують більш детального вивчення, внаслідок чого з'ясувати можливість і доцільність урахування згаданих факторів при розрахунку самоочисної спроможності (СС) різних водних об'єктів, особливо при розрахунку СС в малозабруднених водоймах, де процеси фотосинтезу можуть відігравати вирішальне значення в оновленні розчиненого у воді кисню [3].

Концентрація розчиненого у воді кисню є одним із найважливіших гідрохімічних параметрів, що визначає інтенсивність окисно-відновних біохімічних процесів, які відбуваються у водоймах. Тому кисневий режим, і вміст кисню у воді та його динаміку часто приймають як інтегральний показник стану водних екосистем і якості води в них. Кисневий режим водойми залежить від температури води. Великий вплив на вміст розчиненого кисню у водоймищі надає величина початкового забруднення. Одним з показників вмісту у воді органічних речовин є величина БСК (біохімічне споживання кисню), яка відображує не кількість органічної речовини, а кількість кисню, яка потрібна для окислення цієї речовини біологічним шляхом. Від початкової БСК значно залежить величина падіння вмісту кисню. Розчинений кисень в очищених стічних водах надає вплив на вміст розчиненого кисню у річковій воді нижче точки скиду стічних вод, а у випадку

поганої аерації річкової води відсутність розчиненого кисню у очищених стічних водах може привести до появи анаеробних умов нижче за течією річки. Отже, скид недостатньо очищених вод, які визначаються за БСК, може бути причиною значного забруднення, якщо у стічній воді відсутній розчинений кисень. Але, з іншої сторони, аерація залежить від гідрогеологічних умов водоймища, які сприяють перемішуванню води. Тому у деяких випадках раціонально вживати заходи до штучного підвищення величини аерації шляхом улаштування на відповідних ділянках водоймища перепадів або інших інженерних споруд, які покращують перемішування та аерацію води. Кисневий режим взимку обумовлений надходженням збідненої киснем води. Найбільш гострий дефіцит кисню виникає в період льодоставу, коли біохімічне та хімічне споживання є досить істотним, фотосинтетична аерація низька або зовсім відсутня, а атмосферна аерація припинена внаслідок льодового покриву. Влітку кисневий режим досить не стабільний.

Причини – низька концентрація у воді, що надходить у водосховище, та переважання процесів деструкції органічної речовини над її продукцією, Значний внесок у зменшення концентрації розчиненого у воді кисню в річковій ділянці водосховища належить витратам його на процеси самоочищення водного середовища в умовах інтенсивного антропогенного забруднення основного річища і придаткової мережі [3].

1.2 Огляд і опис методу прогнозування

Машинне навчання – це клас методів штучного інтелекту, характерною рисою яких є не пряме рішення задачі, а навчання в процесі застосування рішень безлічі подібних завдань. Для побудови таких методів використовуються засоби математичної статистики, чисельних методів, методів оптимізації, теорії ймовірностей, теорії графів, різні техніки роботи з даними в цифровій формі.

Основні задачі машинного навчання є:

– Задачу класифікації;

– Задачу відновлення регресії;

– Задачу кластеризації;

– Задачу ідентифікації;

– Задачу прогнозування.

Задача прогнозування, яка нас і цікавить, історично виникла при дослідженні часових рядів і спробі передбачення їх значень через якийсь проміжок часу [4].

Існують різні методи побудови моделі подібних об’єктів, у тому числі з використанням методі машинного навчання, наприклад:

– Нейронні мережі: обчислення структуруються в термінах взаємозв'язаних груп штучних нейронів, які обробляють інформацію із застосуванням конективістського підходу до обчислень. Сучасні нейронні мережі є нелінійними статистичними інструментами моделювання даних. Їх зазвичай застосовують для моделювання складних взаємозв'язків між входами та виходами, для пошуку закономірностей в даних, або для виявлення статистичної структури в невідомому спільному розподілі ймовірності спостережуваних величин;

– Дерева рішень: використовує як передбачувальну модель дерево рішень, яке відображує спостереження про предмет на висновки про цільове значення предмету;

– Баєсові мережі: ймовірнісна графова модель, яка представляє набір випадкових величин та їхніх умовних незалежностей через спрямований ациклічний граф [5];

Оскільки в даному дослідженні стоїть задача пошуку закономірностей для багатовимірного випадку, то доцільно використовувати дерева рішень

У свою чергу, дерева рішень будуються за такими методами:

– Стекінг. Результати роботи кількох алгоритмів показують останньому алгоритму прийняття рішення;

– Багінг – проста техніка, за допомогою якої можем побудувати

незалежні моделі та комбінувати їх, використовуючи деяку модель усереднення (наприклад, зважене середнє, голосування більшості або нормальне середнє).

Зазвичай беруть випадкову вибірку даних для кожної моделі, так всі моделі трохи відрізняються один від одного. Вибірка будується за моделлю вибору з поверненням. Через те, що дана техніка використовує безліч некорелюючих моделей для побудови підсумкової моделі, це зменшує variance. Прикладом бегінга є модель випадкового лісу (Random Forest, RF);

– Бустинг – це техніка побудови ансамблів, у якій моделі побудовані не незалежно, а послідовно. Як і в багінгу, алгоритм навчають багато разів на окремих вибірках. Але в кожну нову вибірку потрапляють дані з вибірки, на яких попередній алгоритм помилився - алгоритм донавчається.

Ця техніка використовує ідею про те, що наступна модель навчатиметься на помилках попередньої. Вони мають нерівну можливість появи в наступних моделях, і частіше з'являться ті, що дають найбільшу помилку. Провісники можуть бути вибрані з широкого асортименту моделей, наприклад дерева рішень, регресія, класифікатори і т.д. Через те, що провісники навчаються на помилках, здійснених попередніми, потрібно менше часу для того, щоб дістатися реальної відповіді. Але повинні вибирати критерій зупинки з обережністю, інакше це може призвести до перенавчання. Градієнтний бустинг - це приклад бустингу рисунок 1.1;

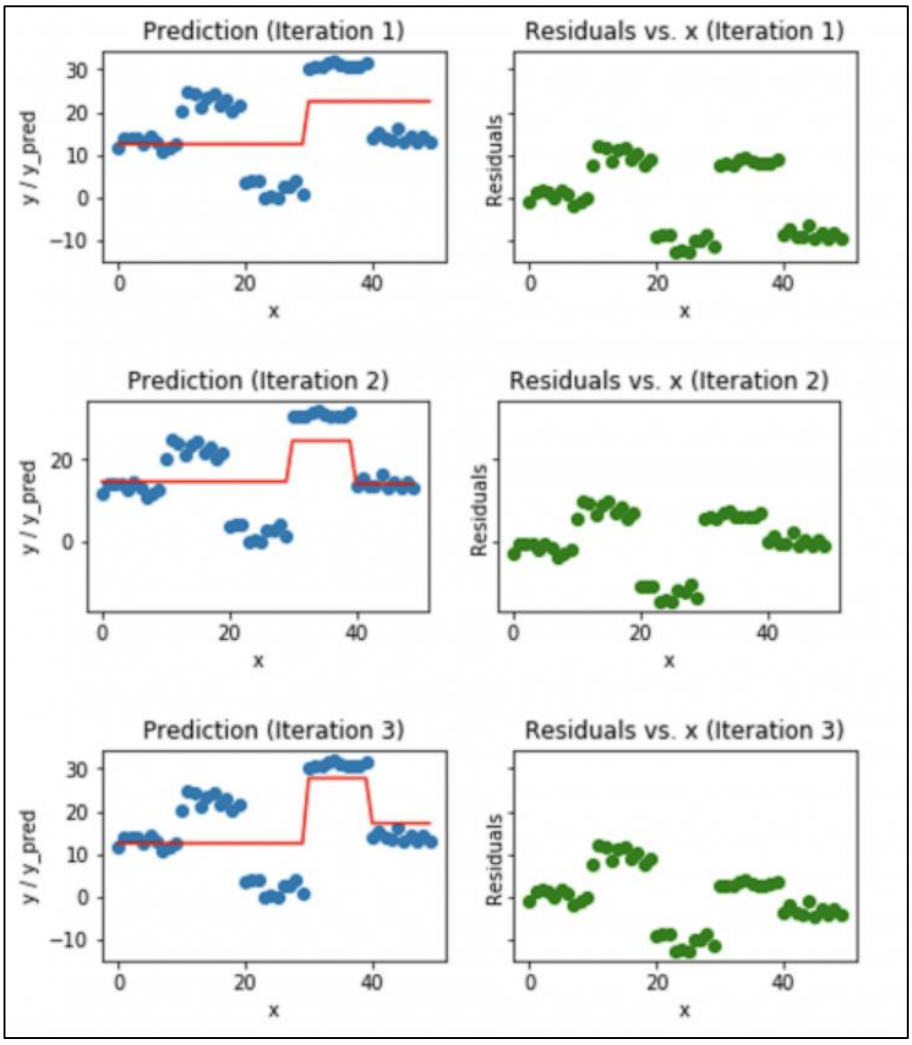


Рисунок 1.1 – Градієнтний бустинг

Найбільш ефективним з них, як відомо, є бустинг. У свою чергу, для автоматизації бустингу існують дві найбільш потужні бібліотеки:

– LightGBM - це швидка, розподілена, високопродуктивна структура підвищення градієнта, заснована на алгоритмі дерева рішень, що використовується для ранжування, класифікації та багатьох інших завдань машинного навчання. Оскільки він заснований на алгоритмах дерева рішень, він розділяє лист дерева з найкращою відповідністю, тоді як інші алгоритми підвищення ділять дерево за глибиною або рівнем, а не листом. Таким чином, при вирощуванні на одному і тому ж листі в Light GBM, листовий алгоритм може зменшити більше втрат, ніж порівняний алгоритм, і, отже, призводить до кращої точності, що рідко може бути досягнуто будь-яким з існуючих алгоритмів підвищення рисунок 1.2;

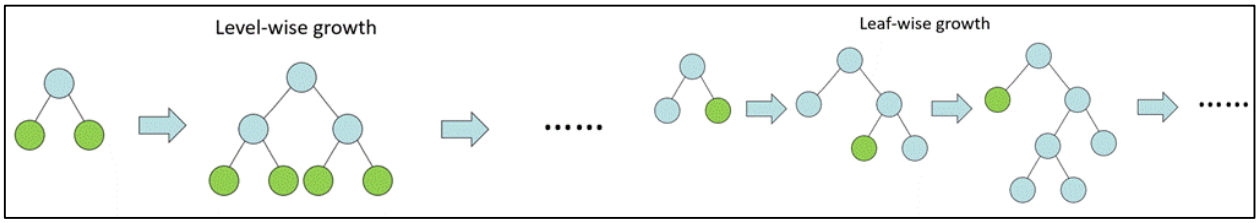


Рисунок 1.2 – Листове зростання для Light GBM

– Sweetviz - це бібліотека Python з відкритим вихідним кодом, яка генерує звіти із зручною візуалізацією для виконання EDA за допомогою всього двох рядків коду. Бібліотека дозволяє швидко створити докладний звіт з усіх характеристик набору даних без особливих зусиль. У можливості Sweetviz також входить цільовий аналіз, порівняння двох датасетів, порівняння двох частин датасета, виділених за певною ознакою, виявлення кореляцій та асоціацій, також Sweetviz дозволяє створювати та зберігати звіт як HTML файл рисунок 1.3;

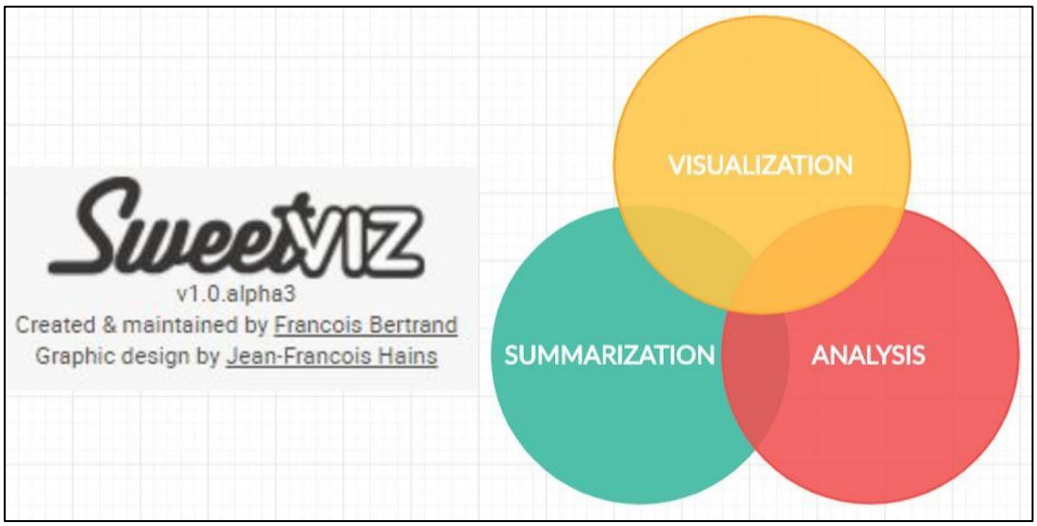


Рисунок 1.3 – Sweetviz

– Autoviz - це бібліотека Python з відкритим вихідним кодом, яка в основному працює над візуалізацією взаємозв’язку даних, вона може знаходити найефективніші функції та створювати творчу візуалізацію лише в одному рядку коду. Autoviz неймовірно швидкий і дуже корисний рисунок 1.4.

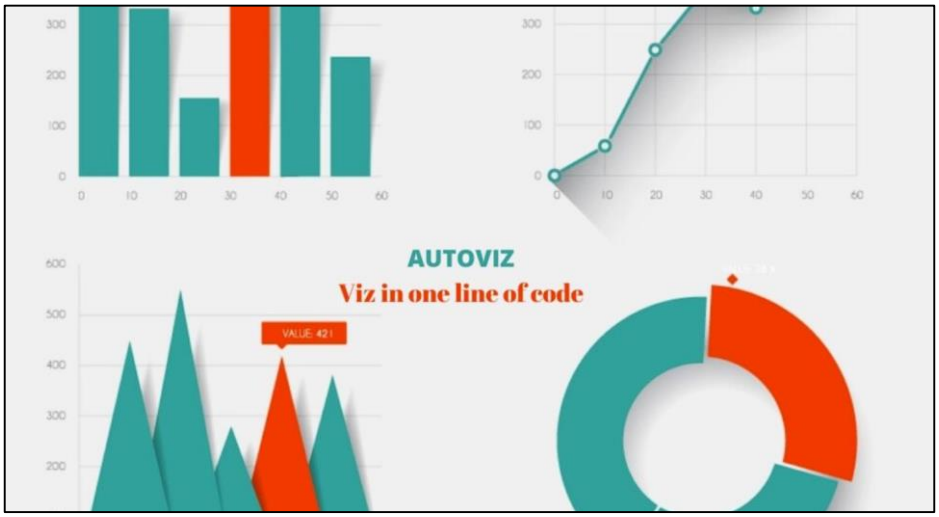


Рисунок 1.4 – Autoviz

Дані бібліотеки допомагають вирішити багато задач, в то числі і задач інформаційних технологій.

1.3 Вибір оптимальних інформаційних технологій

Інформаційні технології даних для прогнозування вмісту розчиненого кисню у річці Південний Буг міститиме в себе побудову діаграми важливості ознак, оскільки визначено, що оптимальними методами бустингу є бібліотеки Sweetviz, Autoviz а також моделі машинного навчаня. Для цього будемо використовувати мову програмування Python.

Python, як і будь-яка інша мова програмування, має свої відмінні риси. Отже, можна виділити наступні:

– Кросплатформеність. Python - це інтерпретована мова, його інтерпретатори існують для багатьох платформ. Тому з запуском його на будь-який ОС не повинно виникнути проблем;

– З Python доступна величезна кількість сервісів, середовищ розробки, і фреймворків. Легко можна знайти необхідний продукт для роботи;

– Можливість під’єднати бібліотеки, написані на C. Це дозволяє підвищити ефективність, поліпшити швидкодію;

– Наявність самих різних джерел інформації про Python. Не важко буде знайти відповідь на питання, які виникають, через те, що існує багато безкоштовної літератури, навчальних відеопосібників, готових початкових кодів та шаблонів для роботи у відкритому доступі;

– Python легко конкурує з іншими мовами програмування, оскільки має безліч переваг. По-перше, це зрозуміла і проста мова програмування. Особливо підходить він для новачків. Можна створити цікаві програми, і при цьому не доведеться сидіти тижнями, вивчаючи складний синтаксис;

– Динамічна типізація – це одна з головних переваг мови Python. Для новачків це можливість спростити написання коду та уникнути безлічі фатальних помилок і багів в роботі. Також в Python немає операторних дужок, з розставленням яких найчастіше виникають складності [6].

За швидкістю виконання програм, коли це стосується великих повномасштабних проектів, Python, звичайно ж, не лідер. Тут мінусом є і автоматичне керування пам'яттю, і повна динамічна типізація. Python значно поступається таким мовам як Java, C, C ++, але і разом з тим з легкістю дає фору JavaScript, Ruby, PHP. Підключення бібліотек, написаних на C і можливість попередньої компіляції коду в байт-код – все це дозволяє поліпшити швидкодію [7].

1.4 Висновки

В даному розділі розглянуто основні характеристики об’єкту дослідження, здійснено аналіз функціонування кисневого режиму і деструкції органічних речовин та проаналізовано основні екологічні проблеми в даному напрямку. Наведено опис методу прогнозування та вибрано оптимальні інформаційні технології для подальшого дослідження, а саме мову програмування Python і моделі прогнозування такі як ARIMA, Facebook Prophet, а також методи машинного навчання.

2 ВИБІР ОПТИМАЛЬНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОЗВІДУВАЛЬНИЙ АНАЛІЗ ДАНИХ

2.1 Підготовка даних

Для того щоб почати, необхідно завантажити пакети маніпуляцій даними, моделі навчання та інженерні особливості. Процес завантаження зображений на рисунку 2.1.

```
# Work with Data - the main Python libraries
import numpy as np
import pandas as pd
import pandas_profiling as pp

# Visualization
import matplotlib.pyplot as plt

# Preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, KFold, ShuffleSplit, GridSearchCV

# Modeling
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb
from xgboost.sklearn import XGBRegressor

# Metrics
from sklearn.metrics import r2_score

import warnings
warnings.simplefilter('ignore')
```

Рисунок 2.1 – Завантаження пакетів даних

Підключаємо дані по вмісту речовин в річці Південний Буг (рис. 2.2).

```
data = pd.read_csv('../input/wq-southern-bug-river-01052021/PB_All_2000_2021.csv', sep=';', hea
der=0)
data
```

Рисунок 2.2 – Підключення даних

На рисунку 2.3 відображена таблиця даних спостереження. Де «id» це номер станції спостереження, «date» дата спостереження і 8 показників

речовин. Нас найбільше цікавить O2 через те, що він є об’єктом дослідження даної роботи.

	id	date	NH4	BSK5	Suspended	O2	NO3	NO2	SO4	PO4	CL
0	1	17.02.2000	0.330	2.77	12.0	12.30	9.50	0.057	154.00	0.454	289.50
1	1	11.05.2000	0.044	3.00	51.6	14.61	17.75	0.034	352.00	0.090	1792.00
2	1	11.09.2000	0.032	2.10	24.5	9.87	13.80	0.173	416.00	0.200	2509.00
3	1	13.12.2000	0.170	2.23	35.6	12.40	17.13	0.099	275.20	0.377	1264.00
4	1	02.03.2001	0.000	3.03	48.8	14.69	10.00	0.065	281.60	0.134	1462.00
...	...	...	...	...	...	...	...	...	...	...	...
2856	22	06.10.2020	0.046	2.69	3.6	8.28	3.80	0.038	160.00	0.726	77.85
2857	22	27.10.2020	0.000	1.52	0.5	11.26	0.56	0.031	147.20	0.634	71.95
2858	22	03.12.2020	0.034	0.29	0.8	11.09	2.58	0.042	209.92	0.484	61.17
2859	22	12.01.2021	0.000	2.10	0.0	14.31	3.94	0.034	121.60	0.424	63.49
2860	22	10.02.2021	0.000	1.78	0.0	14.30	6.30	0.033	134.40	0.582	66.31

Рисунок 2.3 – Фрагмент початкових даних

Формування таблиці показників з додаванням кілометражу та впорядкування їх за зменшенням від витоку до гирла (рис. 2.4).

```
data_about = pd.read_csv('../input/wq-southern-bug-river-01052021/PB_stations.csv', sep=';', he
ader=0, encoding='cp1251')
data_about.sort_values(by=['length'], ascending=False)
```

Рисунок 2.4 – Формування таблиці

Список всіх показників, які знаходяться на річці Південний Буг (рис. 2.5).

	id	length	name_station
20	21	773.0	р. Південний Буг, 773 км, смт. Чорний Острів, Мар'янівське вдсх.
19	20	755.0	р. Південний Буг, 755 км, м. Хмельницький , Хмельницьке вдсх.
18	19	744.0	р. Південний Буг, 744 км, с. Копистин, нижче м.Хмельницький
17	18	711.0	р. Південний Буг, 711 км, смт. Меджибіж, Меджибіжське вдсх.
16	17	692.0	р. Південний Буг, 692 км, с. Щедрове, Щедрівське вдсх.
15	16	652.0	р. Південний Буг, 652 км, м. Хмільник, питний в/з, вище міста
14	15	607.0	р. Південний Буг, 607 км, с. Гущинці, нижче села , питний водозабір м.Калинівка
13	14	582.0	р. Південний Буг, 582 км, м. Вінниця, Сабарівське вдсх, питний в/з міста, вище міста
12	13	569.5	р. Південний Буг, 569,5 км, 500 м нижче скиду ВОКВП ВКГ "Вінницяводоканал" (1,5 км нижче греблі Сабарівського вдсх.)
11	12	537.0	р. Південний Буг, 537 км, смт. Сутиски, Сутиське вдсх., н/б'єф
9	10	413.0	р. Південний Буг, 413 км, с. Маньківка, вище села, питний в/з м.Ладижин
8	9	400.0	р. Південний Буг, 400 км, м. Ладижин, Ладижинське вдсх.
7	8	372.0	р. Південний Буг, 372 км, с. Глибочок, Глибочекське вдсх.
6	7	327.0	р. Південний Буг, 327 км, с. Ставки, кордон Вінницької та Кіровоградської обл.
5	6	316.0	р. Південний Буг, 316 км, м.Гайворон, Гайворонське вдсх.
4	5	237.0	р. Південний Буг, 237 км, питний водозабір смт Побузьке
3	4	206.0	р. Південний Буг, 206 км, м. Первомайськ, Первомайське вдсх.
2	3	153.0	р. Південний Буг, 153 км, с. Олексіївка, питний в/з м. Південно-Українськ
1	2	136.0	р. Південний Буг, 136 км, с. Олександрівка, Олександрівське вдсх.
21	22	97.0	р. Південний Буг, 97 км, м. Вознесенськ, пит.в/з м. Вознесенськ, 2 км до в'їзду у м. Вознесенськ по трасі з м. Миколаїв
10	11	50.0	р. Південний Буг, 50 км, с. Ковалівка, Південно-Бузька ЗС
0	1	0.5	р. Південний Буг, 0,5 км, м. Миколаїв, Бузький лиман, тех. в/з Миколаївської ТЕЦ (ліва частина морського порту)

Рисунок 2.5 – Показники на річці Південний Буг

Дивимося на самі данні бачимо, що порядок постів даних багато міряли часто дані доступні за багато років, але по деяких постам даних не вистачає (рис. 2.6).

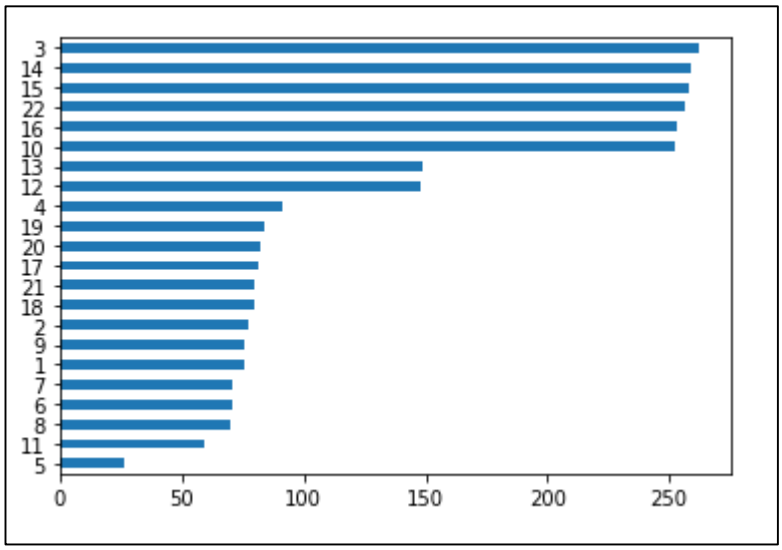


Рисунок 2.6 – Кількість даних по постах

Для початку перетворимо дані в формат «date» та створимо ознаку «year» (рис. 2.7).

```
# Determination the year of observations
data['ds'] = pd.to_datetime(data['date'])
data['year'] = data['ds'].dt.year
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2861 entries, 0 to 2860
Data columns (total 13 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   id         2861 non-null   int64
 1   date       2861 non-null   object
 2   NH4        2858 non-null   float64
 3   BSK5       2860 non-null   float64
 4   Suspended  2845 non-null   float64
 5   O2         2858 non-null   float64
 6   NO3        2860 non-null   float64
 7   NO2        2858 non-null   float64
 8   SO4        2812 non-null   float64
 9   PO4        2833 non-null   float64
 10  CL         2812 non-null   float64
 11  ds         2861 non-null   datetime64[ns]
 12  year       2861 non-null   int64
dtypes: datetime64[ns](1), float64(9), int64(2), object(1)
memory usage: 290.7+ KB
```

Рисунок 2.7 – Перетворення даних

Далі витягуємо номер поста і рік, групуємо за номером поста та впорядковуємо за зменшенням (рис. 2.8).

```
data[['id', 'year']].groupby(by=['id']).min().sort_values(by=['year'], ascending=False)
```

Рисунок 2.8 – Групування за номером поста та впорядкування за зменшенням

Тобто видно, що на 5 посту дані починають лише з 2019 року на 13 посту дані починаються з 2006 року, а решта з 2000 (рис. 2.9).

	year
id	
5	2019
13	2006
1	2000
21	2000
20	2000
19	2000
18	2000
17	2000
16	2000
15	2000
14	2000
12	2000
2	2000
11	2000
10	2000
9	2000
8	2000
7	2000
6	2000
4	2000
3	2000
22	2000

Рисунок 2.9 – Кількість даних по роках

Тепер переглянемо кінцеві дати даних все так само як і з минулим тільки з «max» (рис. 2.10).

```
data[['id', 'year']].groupby(by=['id']).max().sort_values(by=['year'], ascending=False)
```

Рисунок 2.10 – Групування за номером поста та впорядкування за збільшення

Можна побачити, що пости 3,5,10,14,15,16 і 22 мають свіжі дані по 2021 рік, а решта лише по 2018 рік (рис. 2.11).

id	year
22	2021
14	2021
3	2021
5	2021
10	2021
16	2021
15	2021
21	2018
20	2018
19	2018
18	2018
17	2018
1	2018
13	2018
2	2018
11	2018
9	2018
8	2018
7	2018
6	2018
4	2018
12	2018

Рисунок 2.11 – Кількість даних по роках

Хоча, якщо обмежитися 2018 роком, то можна взяти всі станції.

Отже, зі списку доцільним є відібрати 3 послідовних пости 14,15,16 подивимося на них в таблиці «data_about» (рис. 2.12).

```
stations_good = [14, 15, 16]
data_about[data_about['id'].isin(stations_good)]
```

	id	length	name_station
13	14	582.0	р. Південний Буг, 582 км, м. Вінниця, Сабарівське вдсх, питний в/з міста, вище міста
14	15	607.0	р. Південний Буг, 607 км, с. Гущинці, нижче села , питний водозабір м.Калинівка
15	16	652.0	р. Південний Буг, 652 км, м. Хмільник, питний в/з, вище міста

Рисунок 2.12 – Відібрані послідовно розташовані пости

Це послідовно розташовані пости біля питних водозаборів зокрема Вінниця, Гущинці, Хмільник. Отже, доцільно зробити Вінницький пост цільовим тобто тим який було б цікаво прогнозувати за даними попередніх постів. Але особливості взаємодії речовин у воді показують, що більш доцільно моделювати не один показник по його ж значенню вище по течії, а також інші показники, що можуть на це вплинути. Тобто для того, щоб моделювати O2 буде доцільним взяти до нього ще NH4 і BSK5 вони знижують рівень кисню в воді відповідно і впливають на процеси окислення легко окиснюваних речовин (рис. 2.13).

```
target_data_name = 'O2'
#feature_target_all = ['NH4', 'BSK5', 'NO3', 'NO2', 'SO4', 'PO4', 'CL']
feature_target_all = ['BSK5', 'NH4']
feature_data_all = feature_target_all + [target_data_name]
feature_data_all
```

Рисунок 2.13 – Вибір речовин прогнозування

Тепер витягуємо ці три показники. Тут є така особливість, що є сенс видалити ті рядки де дані не повні, щоб залишися лише повні рядки тому як так простіше моделювати результат (рис. 2.14).

```
df_indicator = data[['id', 'ds'] + feature_data_all]
df_indicator = df_indicator[df_indicator['id'].isin(stations_good)].dropna().reset_index(drop=True)
df_indicator
```

	id	ds	BSK5	NH4	O2
0	14	2000-10-01	1.9	1.30	15.8
1	14	2000-01-02	2.5	2.20	10.6
2	14	2000-01-03	4.1	0.68	10.3
3	14	2000-04-04	3.4	0.81	8.7
4	14	2000-05-16	3.0	0.27	6.5
...	...	...	...	...	...
763	16	2020-08-12	5.6	0.17	6.8
764	16	2021-01-28	5.4	0.26	6.5
765	16	2021-02-16	6.8	1.64	8.0
766	16	2021-03-16	4.4	2.29	7.7
767	16	2021-06-04	6.1	0.16	10.8

Рисунок 2.14 – Показники по трьох даним

Ось бачимо 768 рядків, але це по різних постах, а нам потрібно, що у нас окремо виділений «таргет» яким є 16 пост по O2. Тому таблиці потрібно перекомпілювати.

Для подальшого компілювання таблиці потрібно використати команду «pivot_table» з бібліотеки Pandas. Індекс обробимо дату, номер колонок це будуть номери постів, а значення будуть ті сам значення, що і в таблиці (рис. 2.15).

```
cols = []
for station in stations_good:
    for feature in feature_data_all:
        cols.append(str(station) + "_" + feature)
cols
```

```
['14_BSK5',
 '14_NH4',
 '14_O2',
 '15_BSK5',
 '15_NH4',
 '15_O2',
 '16_BSK5',
 '16_NH4',
 '16_O2']
```

Рисунок 2.15 – Синтезування даних прогнозування

Для подальшого компілювання таблиці потрібно використати команду «pivot_table» з бібліотеки Pandas. Індекс обробимо дату, номер колонок це будуть номери постів, а значення будуть ті сам значення, що і в таблиці (рис. 2.14). При цьому для того, що підписи стовпців більш зручними й не багаторівневими спеціальним циклом синтезуємо їх назви роблячи парами номер стовпця через знак «_» і показників. Маємо ось таку табличку (рис. 2.16).

```
df = pd.pivot_table(df_indicator, index=["ds"], columns=["id"], values=feature_data_all).dropna
()
df.columns = cols
df
```

	14_BSK5	14_NH4	14_O2	15_BSK5	15_NH4	15_O2	16_BSK5	16_NH4	16_O2
ds									
2000-01-02	2.5	2.7	3.1	2.20	2.20	2.40	10.6	11.9	10.3
2000-01-03	4.1	4.4	6.0	0.68	0.87	0.54	10.3	10.3	11.9
2000-01-08	2.9	3.4	9.3	0.37	0.25	0.14	5.8	10.0	12.3
2000-04-04	3.4	3.1	3.4	0.81	1.22	0.51	8.7	8.7	8.1
2000-04-07	4.4	5.3	7.6	0.10	0.07	0.14	6.4	6.1	7.4
...	...	...	...	...	...	...	...	...	...
2020-12-08	6.2	6.8	6.0	0.57	0.28	0.20	5.2	6.8	8.3
2021-01-28	6.4	5.6	5.4	0.21	0.27	0.26	7.1	6.2	6.5
2021-02-16	11.6	6.2	6.8	2.52	1.92	1.64	6.5	6.2	8.0
2021-03-16	6.2	4.2	4.4	1.03	1.38	2.29	9.2	8.7	7.7
2021-06-04	5.6	6.2	6.1	0.43	0.17	0.16	12.3	13.9	10.8

239 rows × 9 columns

Рисунок 2.16 – Таблиця прогнозування

Відповідно останній стовпчик і є той самий «таргет» який буду прогнозувати. Це вміст O2 на 16 пості за даними інших показників, що є у цьому створі.

На графіку видно, що в цілому показники однорідні, але можна побачити, що у 2010-2012 роках є деякі аномальні дані, які дуже різко відрізняються від всіх інших, що демонструється на рисунку 2.17.

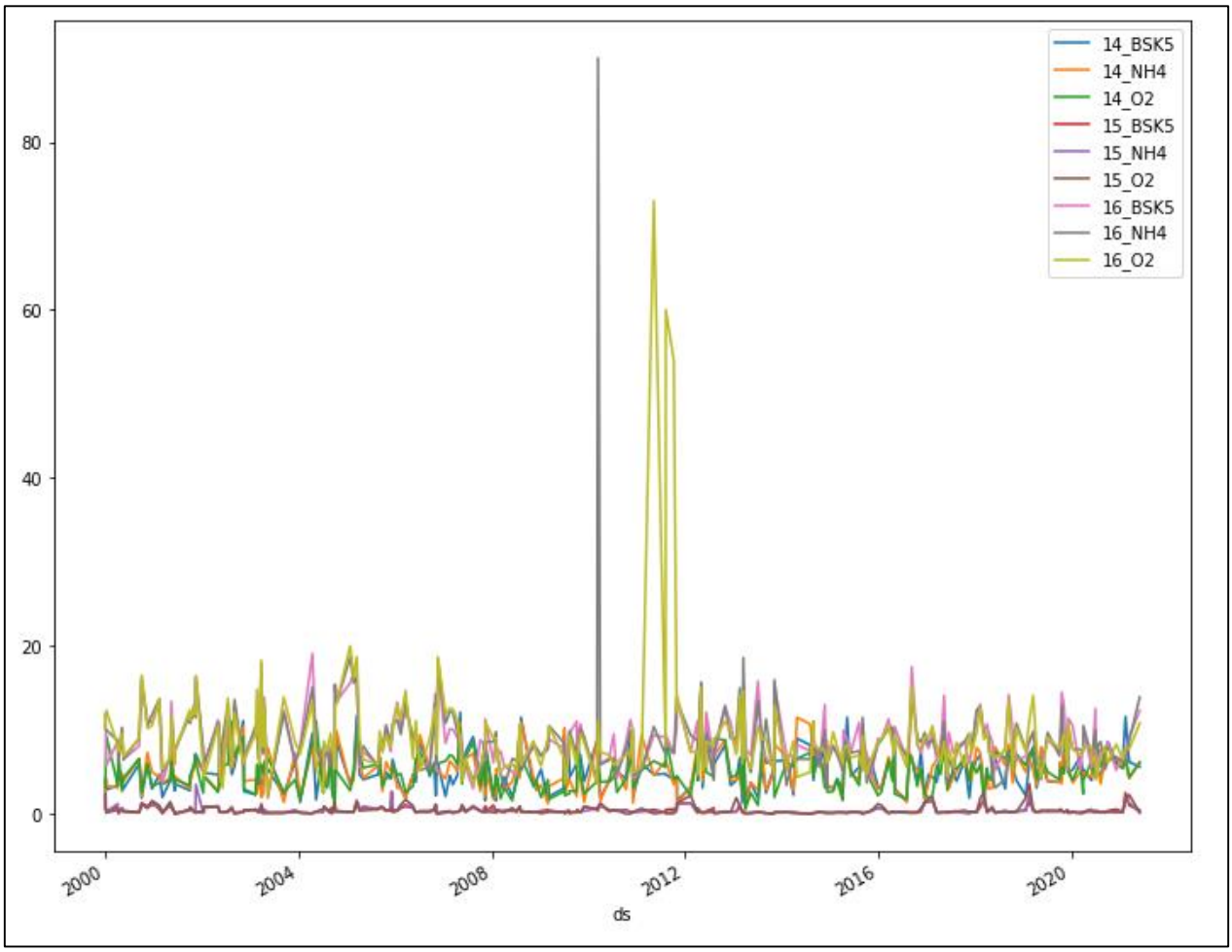


Рисунок 2.17 – Графік показників за роками

Тепер повторимо це з урахуванням не двох а чотирьох показників, виділимо їх. Далі необхідно завантажити пакети маніпуляцій даними, моделі навчання та інженерні особливості. Процес завантаження зображений на рисунку 2.18, 2.19.

```
# Import libraries
import random
import os
import numpy as np
import pandas as pd
import requests
import pandas_datareader as web

# Date
import datetime as dt
from datetime import date, timedelta, datetime

# EDA
import matplotlib.pyplot as plt
from matplotlib.pylab import rcParams
import plotly.express as px
import plotly.graph_objects as go
from plotly.offline import init_notebook_mode
init_notebook_mode(connected=True)

# FE
from tsfresh import extract_features, select_features, extra
ct_relevant_features
from tsfresh.utilities.dataframe_functions import impute
from sklearn.inspection import permutation_importance
import eli5
from eli5.sklearn import PermutationImportance
import shap
```

Рисунок 2.18 – Завантаження пакетів даних

```
# Time Series - EDA and Modelling
import statsmodels.api as sm
from statsmodels.graphics.tsaplots import plot_acf, plot_pac
f
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima_model import ARIMA

# Metrics
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error, mean_absolut
e_percentage_error

# Modeling and preprocessing
from sklearn.preprocessing import StandardScaler, MinMaxScal
er
from sklearn.model_selection import train_test_split, GridSe
archCV
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR, LinearSVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import BaggingRegressor, AdaBoostRegre
ssor
from sklearn.neural_network import MLPRegressor
from prophet import Prophet
import xgboost as xgb
from xgboost import XGBRegressor
import lightgbm as lgb
from lightgbm import LGBMRegressor

import warnings
warnings.filterwarnings("ignore")
```

Рисунок 2.19 – Завантаження пакетів даних (продовження)

Додаємо врахування аномальних даних в річці Південний Буг (рис. 2.20).

```
is_anomalies = True
```

Рисунок 2.20 – Додавання аномальних даних

Вказуємо, які типи моделей будуть використовуватись у виконанні поставленої задачі, у нашому випадку це - Facebook Prophet, ARIMA & AutoArima, дерева рішень, нейронні мережі і т.д. рисунок 2.21. Якщо буде потрібно вимкнути одну з моделей, достатньо змінити значення “True” на “False” напроти бажаної моделі. Також додавання автоматичної побудова моделі Arima для часових рядів рисунок 2.22.

```
is_Prophet = True
is_ARIMA = True
is_other_ML = True
```

Рисунок 2.21 – Типи моделей які будуть використовуватися

```
if is_ARIMA:
    !pip install pmdarima
    import pmdarima as pm
```

Рисунок 2.22 – Побудова моделі ARIMA для часових рядів

Імпортуємо дані про якість води у річці Південний Буг, назви індикаторів, інформацію про станції, а також отримуємо дані з заданих показниками датами (рис 2.23, 2.24).

```
def get_water_data(target_indicator_name : str,
                   id_target_station : int,
                   date_start : str = "2000-01-02",
                   feature_indicator_names : list = [], # list of str
                   id_feature_station : list = [],      # list of int
                   date_end : str = "2021-06-04"):
    all_indicator_names = feature_indicator_names + [target_indicator_name]
    print('Selected indicator names:', all_indicator_names)


    pd.set_option('max_colwidth',200)
    all_id_stations = id_feature_station + [id_target_station]
    data_about = pd.read_csv('../input/wq-southern-bug-river-01052021/PB_stations.csv', s
ep=';', header=0, encoding='cp1251')
    print('All stations:')
    display(data_about.sort_values(by=['length'], ascending=False))
    print('\nSelected stations:')
    display(data_about[data_about['id'].isin(all_id_stations)])


    data = pd.read_csv('../input/wq-southern-bug-river-01052021/PB_All_2000_2021.csv', se
p=';', header=0)
    data['ds'] = pd.to_datetime(data['date'])


    df_indicator = data[['id', 'ds'] + all_indicator_names]
    df_indicator = df_indicator[df_indicator['id'].isin(all_id_stations)].dropna().reset_
index(drop=True)
```

Рисунок 2.23 – Імпорт даних

```
    cols = []
    for station in all_id_stations:
        for feature in all_indicator_names:
            cols.append(str(station) + "_" + feature)

    df = pd.pivot_table(df_indicator, index=["ds"], columns=["id"], values=all_indicator_
names).dropna()
    df.columns = cols
    df = df.reset_index(drop=False)


    new_target_name = str(id_target_station) + "_" + target_indicator_name


    df = df[(df['ds']>=date_start) & (df['ds']<=date_end)].reset_index(drop=True)

    return df, all_indicator_names, all_id_stations, new_target_name
```

Рисунок 2.24 – Імпорт даних (продовження)

Нумерації станцій – проти течії річки, за принципом: 21 – вихід річки, 1 – гирло. Список всіх показників, які знаходяться на річці Південний Буг (рис. 2.25).

	id	length	name_station
20	21	773.0	р. Південний Буг, 773 км, смт. Чорний Острів, Мар'янівське вдсх.
19	20	755.0	р. Південний Буг, 755 км, м. Хмельницький , Хмельницьке вдсх.
18	19	744.0	р. Південний Буг, 744 км, с. Копистин, нижче м.Хмельницький
17	18	711.0	р. Південний Буг, 711 км, смт. Меджибіж, Меджибіжське вдсх.
16	17	692.0	р. Південний Буг, 692 км, с. Щедрове, Щедрівське вдсх.
15	16	652.0	р. Південний Буг, 652 км, м. Хмільник, питний в/з, вище міста
14	15	607.0	р. Південний Буг, 607 км, с. Гущинці, нижче села , питний водозабір м.Калинівка
13	14	582.0	р. Південний Буг, 582 км, м. Вінниця, Сабарівське вдсх, питний в/з міста, вище міста
12	13	569.5	р. Південний Буг, 569,5 км, 500 м нижче скиду ВОКВП ВКГ "Вінницяводоканал" (1,5 км нижче греблі Сабарівського вдсх.)
11	12	537.0	р. Південний Буг, 537 км, смт. Сутиски, Сутиське вдсх., н/б'єф
9	10	413.0	р. Південний Буг, 413 км, с. Маньківка, вище села, питний в/з м.Ладижин
8	9	400.0	р. Південний Буг, 400 км, м. Ладижин, Ладижинське вдсх.
7	8	372.0	р. Південний Буг, 372 км, с. Глибочок, Глибочекське вдсх.
6	7	327.0	р. Південний Буг, 327 км, с. Ставки, кордон Вінницької та Кіровоградської обл.
5	6	316.0	р. Південний Буг, 316 км, м.Гайворон, Гайворонське вдсх.
4	5	237.0	р. Південний Буг, 237 км, питний водозабір смт Побузьке
3	4	206.0	р. Південний Буг, 206 км, м. Первомайськ, Первомайське вдсх.
2	3	153.0	р. Південний Буг, 153 км, с. Олексіївка, питний в/з м. Південно-Український
1	2	136.0	р. Південний Буг, 136 км, с. Олександрівка, Олександрівське вдсх.
21	22	97.0	р. Південний Буг, 97 км, м. Вознесенськ, пит.в/з м. Вознесенськ, 2 км до в'їзду у м. Вознесенськ по трасі з м. Миколаїв
10	11	50.0	р. Південний Буг, 50 км, с. Ковалівка, Південно-Бузька ЗС
0	1	0.5	р. Південний Буг, 0,5 км, м. Миколаїв, Бузький лиман, тех. в/з Миколаївської ТЕЦ (ліва частина морського порту)

Рисунок 2.25 – Показники на річці Південний Буг

Зі списку доцільним є відібрати 3 послідовних пости 14 15 16 подивимося на них в таблиці «data_about» (рис. 2.26). Це послідовно розташовані пости біля питних водозаборів зокрема Вінниця, Гущинці, Хмільник. Отже, доцільно зробити Вінницький пост цільовим тобто тим який було б цікаво прогнозувати за даними попередніх постів (рис. 2.27). Але особливості взаємодії речовин у воді показують, що більш доцільно моделювати не один показник по його ж значенню вище по течії, а також інші показники, що можуть на це вплинути. У цьому випадку будемо використовувати 4 речовини, щоб моделювати O2: NH4, BSK5, NO3 і NO2 (рис. 2.28).

	id	length	name_station
13	14	582.0	р. Південний Буг, 582 км, м. Вінниця, Сабарівське вдсх, питний в/з міста, вище міста
14	15	607.0	р. Південний Буг, 607 км, с. Гущинці, нижче села , питний водозабір м.Калинівка
15	16	652.0	р. Південний Буг, 652 км, м. Хмільник, питний в/з, вище міста

Рисунок 2.26 – Відібрані послідовно розташовані пости

```
id_target_station = 14
id_feature_station = [15, 16]
```

Рисунок 2.27 – Задання постів дослідження

```
target_indicator_name = 'O2'
feature_indicator_names = ['NO3', 'NO2', 'BSK5', 'NH4']
```

Рисунок 2.28 – Вибір речовин прогнозування

Для подальшого компілювання таблиці потрібно використати команду «pivot_table» з бібліотеки Pandas. Індекс робимо дату, номер колонок - будуть номери постів, а значеннями будуть ті сам значення, що і в таблиці. При цьому для того, що підписи стовпців більш зручними й не багаторівневими спеціальним циклом синтезуєм їх назви роблячи парами номер стовпця через знак «_» і показників. Маємо ось таку таблицю (рис. 2.29).

	ds	15_NO3	15_NO2	15_BSK5	15_NH4	15_O2	16_NO3	16_NO2	16_BSK5	16_NH4	16_O2	14_NO3	14_NO2	14_BSK5	14_NH4	14_O2
0	2000-01-02	2.5	2.7	3.1	2.20	2.20	2.40	0.270	0.130	0.130	8.80	8.40	7.70	10.6	11.9	10.3
1	2000-01-03	4.1	4.4	6.0	0.68	0.87	0.54	0.090	0.170	0.270	8.80	9.10	8.80	10.3	10.3	11.9
2	2000-01-08	2.9	3.4	9.3	0.37	0.25	0.14	0.240	0.150	0.160	1.50	7.00	0.90	5.8	10.0	12.3
3	2000-04-04	3.4	3.1	3.4	0.81	1.22	0.51	0.090	0.140	0.090	4.60	4.90	3.50	8.7	8.7	8.1
4	2000-04-07	4.4	5.3	7.6	0.10	0.07	0.14	0.170	0.320	0.360	2.30	2.10	1.70	6.4	6.1	7.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
234	2020-12-08	6.2	6.8	6.0	0.57	0.28	0.20	0.061	0.036	0.046	0.68	1.03	0.71	5.2	6.8	8.3
235	2021-01-28	6.4	5.6	5.4	0.21	0.27	0.26	0.065	0.070	0.080	2.95	3.27	2.01	7.1	6.2	6.5
236	2021-02-16	11.6	6.2	6.8	2.52	1.92	1.64	0.408	0.174	0.068	13.79	7.95	7.71	6.5	6.2	8.0
237	2021-03-16	6.2	4.2	4.4	1.03	1.38	2.29	0.122	0.134	6.790	10.90	8.56	7.70	9.2	8.7	7.7
238	2021-06-04	5.6	6.2	6.1	0.43	0.17	0.16	0.129	0.179	0.164	6.57	8.07	6.38	12.3	13.9	10.8

Рисунок 2.29 – Таблиця прогнозування

Відповідно останній стовпчик і є той самий «таргет» який хочем прогнозувати. Це вміст O2 на 14 пості за даними інших показників, що є у цьому створі.

Використаємо стандартні бібліотеки, стандартні методи є бібліотека TSFRESH – яка дозволяє для заданого часового ряду автоматично згенерувати до 1000 ознак, там багато варіантів наприклад: за кожну годину, час, день, тиждень і тому подібне. Недолік цієї бібліотеки в тому що, якщо ваш ряд не містить певної «сезонності» велика кількість таких комбінацій дасть «NAN» або константу як відомо для машинного навчання важливі ознаки, що змінюють значення, якщо вона однакова константа для всіх варіантів значень то вона – неінформативна.

Тому використовуємо функцію «extracted_features» видобуває всі такі ознаки, видаляє «NAN» значення і потім видаляє ті у котрих середньоквадратичне відхилення або дисперсія рівна 0, тобто, які не є змінними - а константою рисунок 2.30

```python
def get_tsfresh_features(data):
    # Get statistic features using library TSFRESH
    # Thanks to https://www.kaggle.com/code/vbmokin/btc-growth-forecasting-with-advanced-fe-for-ohlc

    # Extract features
    extracted_features = extract_features(data, column_id="ds", column_sort="ds")

    # Drop features with NaN
    extracted_features_clean = extracted_features.dropna(axis=1, how='all').reset_index(drop=True)

    # Drop features with constants
    cols_std_zero  = []
    for col in extracted_features_clean.columns:
        if extracted_features_clean[col].std()==0:
            cols_std_zero.append(col)
    extracted_features_clean = extracted_features_clean.drop(columns = cols_std_zero)

    extracted_features_clean['ds'] = data['ds']   # For the merging

    return extracted_features_clean
```

```python
%%time
# FE with TSFRESH
extracted_features_clean = get_tsfresh_features(df[['ds', target_name]])
extracted_features_clean
```

Рисунок 2.30 – Бібліотеки TSFRESH

Запуск цієї функції дає нам 24 ознаки, вони тут підписані і можна прочитати про кожну окремо рисунок 2.31.

	14_O2__sum_values	14_O2__abs_energy	14_O2__median	14_O2__mean	14_O2__root_mean_square	14_O2__maximum	14_O2__absolute_maximum	14_O2__min
0	10.3	106.09	10.3	10.3	10.3	10.3	10.3	10.3
1	11.9	141.61	11.9	11.9	11.9	11.9	11.9	11.9
2	12.3	151.29	12.3	12.3	12.3	12.3	12.3	12.3
3	8.1	65.61	8.1	8.1	8.1	8.1	8.1	8.1
4	7.4	54.76	7.4	7.4	7.4	7.4	7.4	7.4
...	...	...	...	...	...	...	...	...
234	8.3	68.89	8.3	8.3	8.3	8.3	8.3	8.3
235	6.5	42.25	6.5	6.5	6.5	6.5	6.5	6.5
236	8.0	64.00	8.0	8.0	8.0	8.0	8.0	8.0
237	7.7	59.29	7.7	7.7	7.7	7.7	7.7	7.7
238	10.8	116.64	10.8	10.8	10.8	10.8	10.8	10.8

239 rows × 24 columns

Рисунок 2.31 – Ознаки по кожній речовині

Далі можемо бачити статистику по цим ознаках, які також можна розглянути окремо рисунок 2.32. а на рисунку 2.33 відображений список ознак, які отримали при виконанні аналізу.

	14_O2__sum_values	14_O2__abs_energy	14_O2__median	14_O2__mean	14_O2__root_mean_square	14_O2__maximum	14_O2__absolute_maximum	14_O2__m
count	239.000000	239.000000	239.000000	239.000000	239.000000	239.000000	239.000000	239.0000
mean	9.273305	132.093397	9.273305	9.273305	9.273305	9.273305	9.273305	9.273305
std	6.803888	450.523919	6.803888	6.803888	6.803888	6.803888	6.803888	6.803888
min	2.200000	4.840000	2.200000	2.200000	2.200000	2.200000	2.200000	2.200000
25%	6.520000	42.510800	6.520000	6.520000	6.520000	6.520000	6.520000	6.520000
50%	8.000000	64.000000	8.000000	8.000000	8.000000	8.000000	8.000000	8.000000
75%	10.350000	107.125000	10.350000	10.350000	10.350000	10.350000	10.350000	10.35000
max	73.000000	5329.000000	73.000000	73.000000	73.000000	73.000000	73.000000	73.00000

Рисунок 2.32 – Статистика за ознаками

```
['14_O2__sum_values',
 '14_O2__abs_energy',
 '14_O2__median',
 '14_O2__mean',
 '14_O2__root_mean_square',
 '14_O2__maximum',
 '14_O2__absolute_maximum',
 '14_O2__minimum',
 '14_O2__benford_correlation',
 '14_O2__quantile__q_0.1',
 '14_O2__quantile__q_0.2',
 '14_O2__quantile__q_0.3',
 '14_O2__quantile__q_0.4',
 '14_O2__quantile__q_0.6',
 '14_O2__quantile__q_0.7',
 '14_O2__quantile__q_0.8',
 '14_O2__quantile__q_0.9',
 '14_O2__cwt_coefficients__coeff_0__w_2__widths_(2, 5, 10, 20)',
 '14_O2__cwt_coefficients__coeff_0__w_5__widths_(2, 5, 10, 20)',
 '14_O2__cwt_coefficients__coeff_0__w_10__widths_(2, 5, 10, 20)',
 '14_O2__cwt_coefficients__coeff_0__w_20__widths_(2, 5, 10, 20)',
 '14_O2__fft_coefficient__attr_"real"__coeff_0',
 '14_O2__fft_coefficient__attr_"abs"__coeff_0',
 'ds']
```

Рисунок 2.33 – Перелік отриманих ознак

2.2 Розвідувальний аналіз

Для того, щоб провести розвідувальний аналіз потрібно перетворити дані або як варіант представити їх в зручній формі, коректній формі. Результатами розвідувального аналізу зазвичай зображаються як схеми, графіки, діаграми, таблиці тощо. Методами за допомогою яких відбувається аналіз називають:

- Кластерний аналіз;
- Аналіз часових рядів;
- Аналіз відповідностей;
- Дерева класифікацій;
- Факторний аналіз;
- Покрокова лінійна та нелінійна регресія;
- Аналіз дискримінантних функцій;
- Логлінійний аналіз;
- Канонічні кореляції;
- Багатовимірне шкалювання.

Зазвичай результатами розвідувального аналізу не користуються за для вироблення управлінських рішень. Їхнє застосування — допомога в розробці найкращої стратегії для поглибленого вивчення, аналізу, висування гіпотез, чи інших математичних методів [8].

Застосовуємо «ProfileReport» за для побудови різноманітних графіків, гістограм по кожному показнику. Які можна проаналізувати і вивчити (рис. 2.34).

```
%%time
if is_EDA_with_Pandas_Profiling:
    profile = df.profile_report(title='Pandas Profiling Report for dataset')
    profile.to_file(output_file="profile.html")
    display(profile)
```

```
Summarize dataset: 100% 254/254 [00:45<00:00, 3.86it/s, Completed]
Generate report structure: 100% 1/1 [00:10<00:00, 10.89s/it]
Render HTML: 100% 1/1 [00:05<00:00, 5.45s/it]
Export report to file: 100% 1/1 [00:00<00:00, 18.04it/s]
```

Рисунок 2.34 – Застосування «ProfileReport»

Після виконання цієї команди, ми можемо побачити гістограми даних по кожному показнику за допомогою яких виконувався аналіз і буде виконане подальше прогнозування (рис. 2.35-2.40).

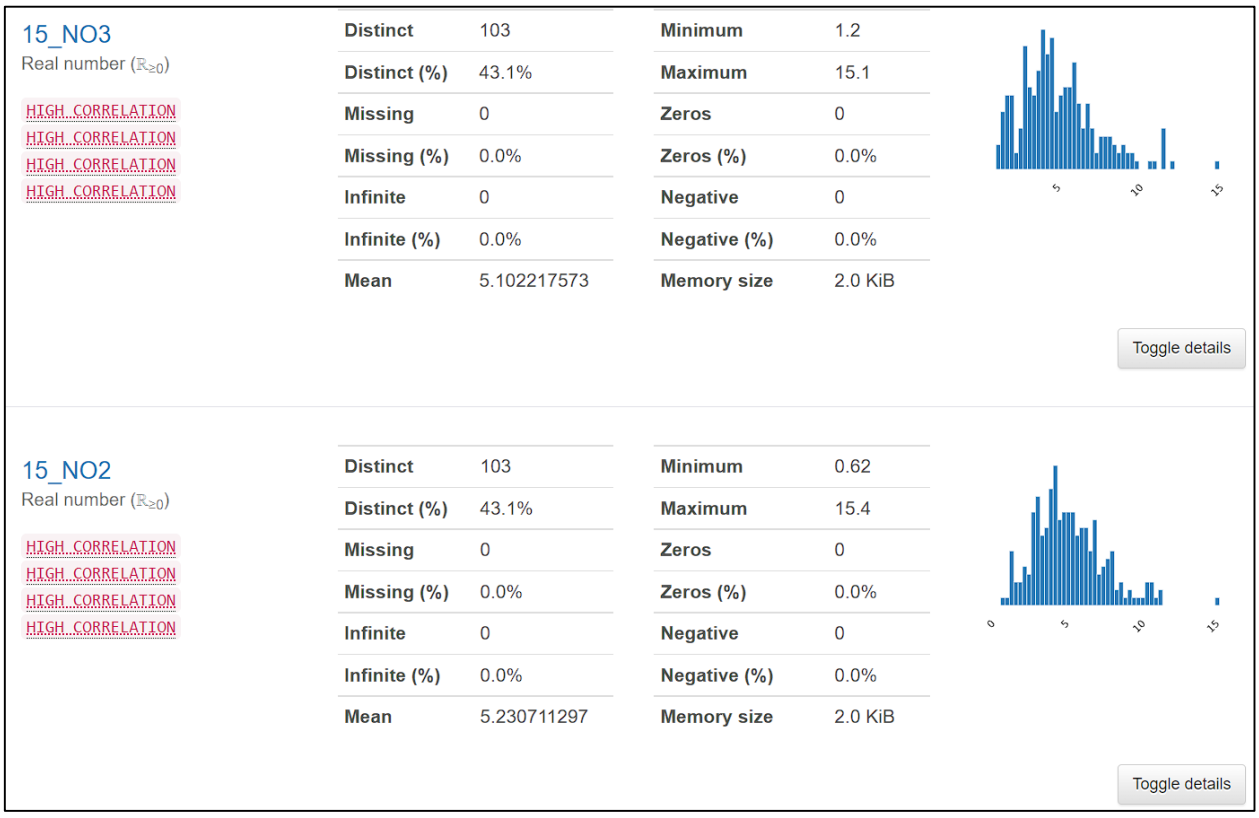
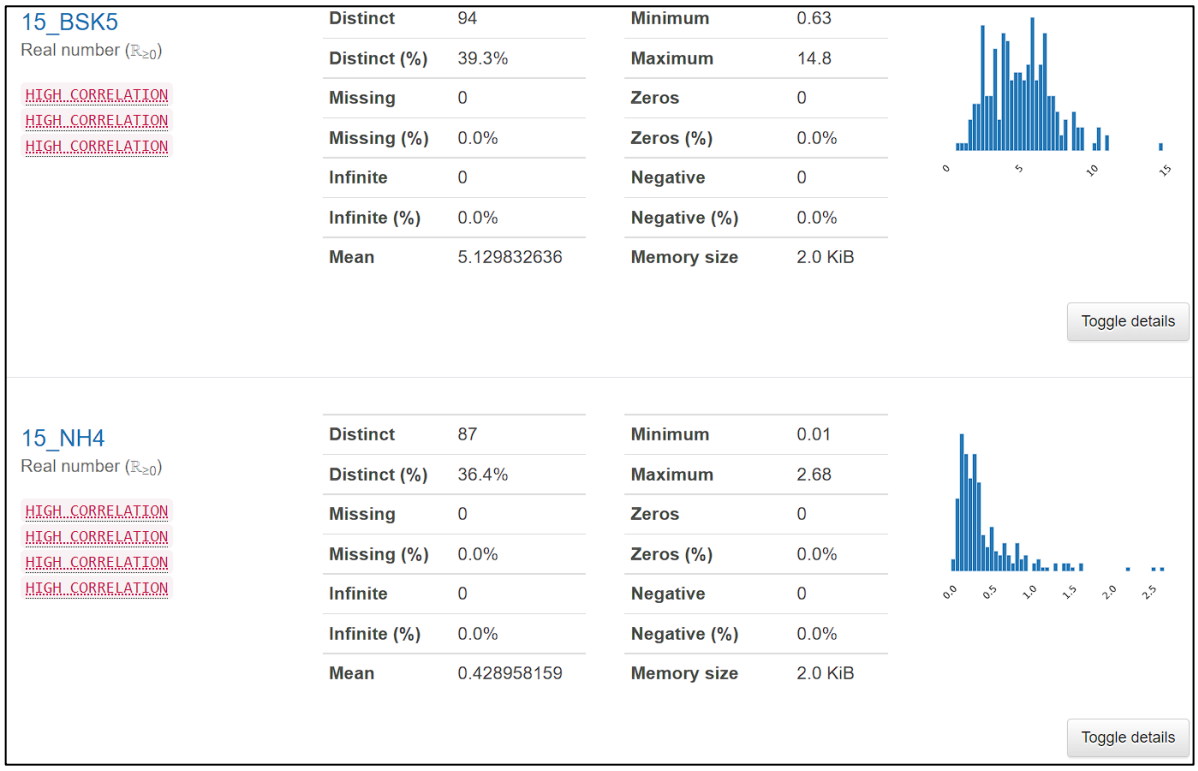


Рисунок 2.35 – Гістограми показників



15_BSK5
Real number (ℝ≥0)

HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION

Distinct	94	Minimum	0.63
Distinct (%)	39.3%	Maximum	14.8
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	5.129832636	Memory size	2.0 KiB

Toggle details

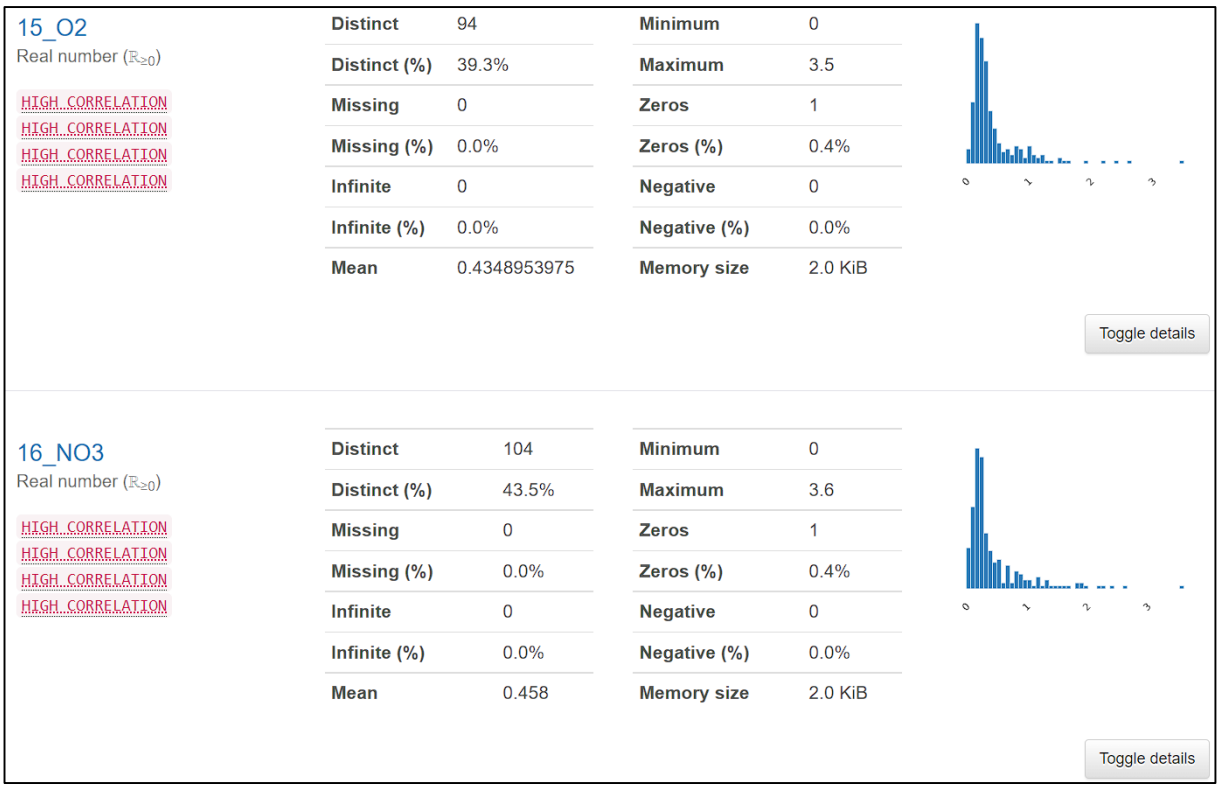
15_NH4
Real number (ℝ≥0)

HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION

Distinct	87	Minimum	0.01
Distinct (%)	36.4%	Maximum	2.68
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	0.428958159	Memory size	2.0 KiB

Toggle details

Рисунок 2.36 – Гістограми показників (продовження)



15_O2
Real number (ℝ≥0)

HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION

Distinct	94	Minimum	0
Distinct (%)	39.3%	Maximum	3.5
Missing	0	Zeros	1
Missing (%)	0.0%	Zeros (%)	0.4%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	0.4348953975	Memory size	2.0 KiB

Toggle details

16_NO3
Real number (ℝ≥0)

HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION

Distinct	104	Minimum	0
Distinct (%)	43.5%	Maximum	3.6
Missing	0	Zeros	1
Missing (%)	0.0%	Zeros (%)	0.4%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	0.458	Memory size	2.0 KiB

Toggle details

Рисунок 2.37 – Гістограми показників (продовження)

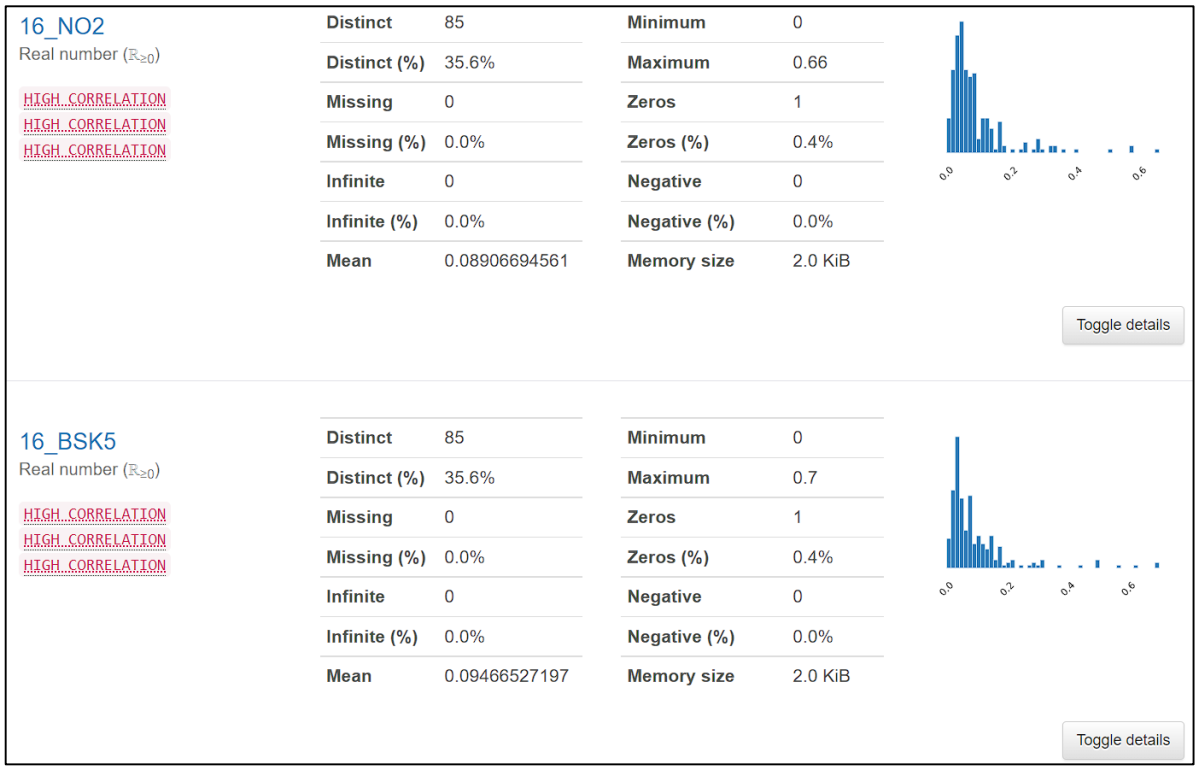


Рисунок 2.38 – Гістограми показників (продовження)

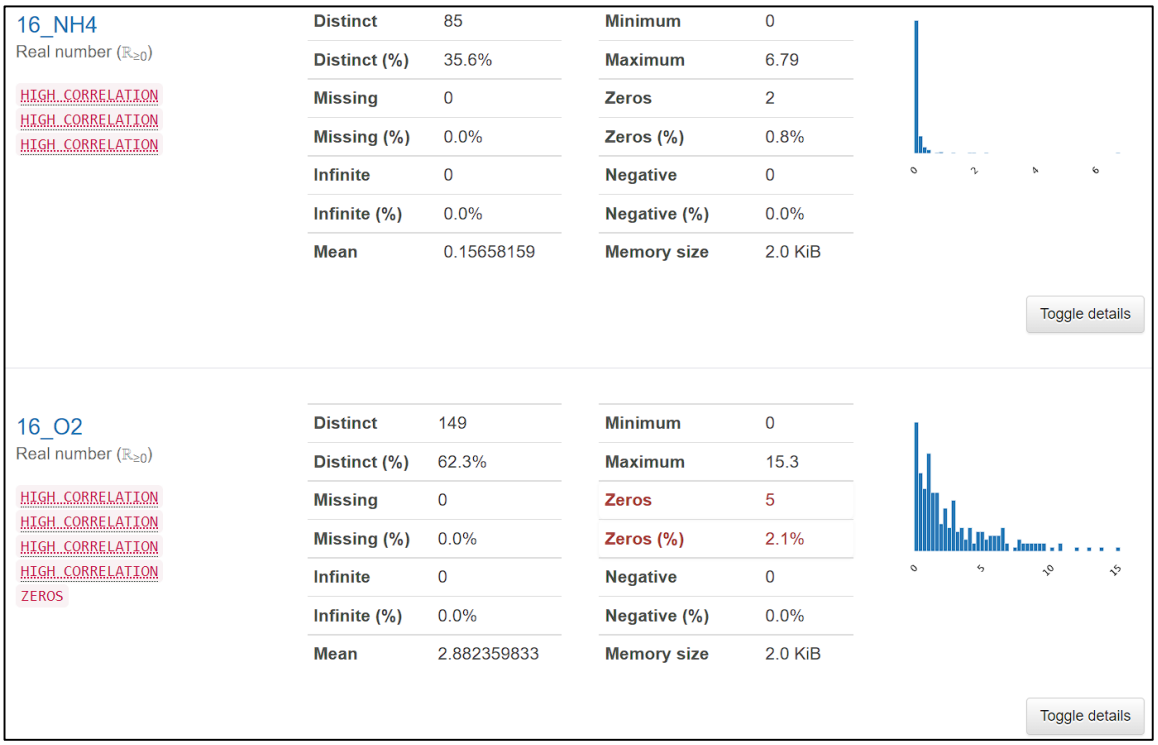


Рисунок 2.39 – Гістограми показників (продовження)

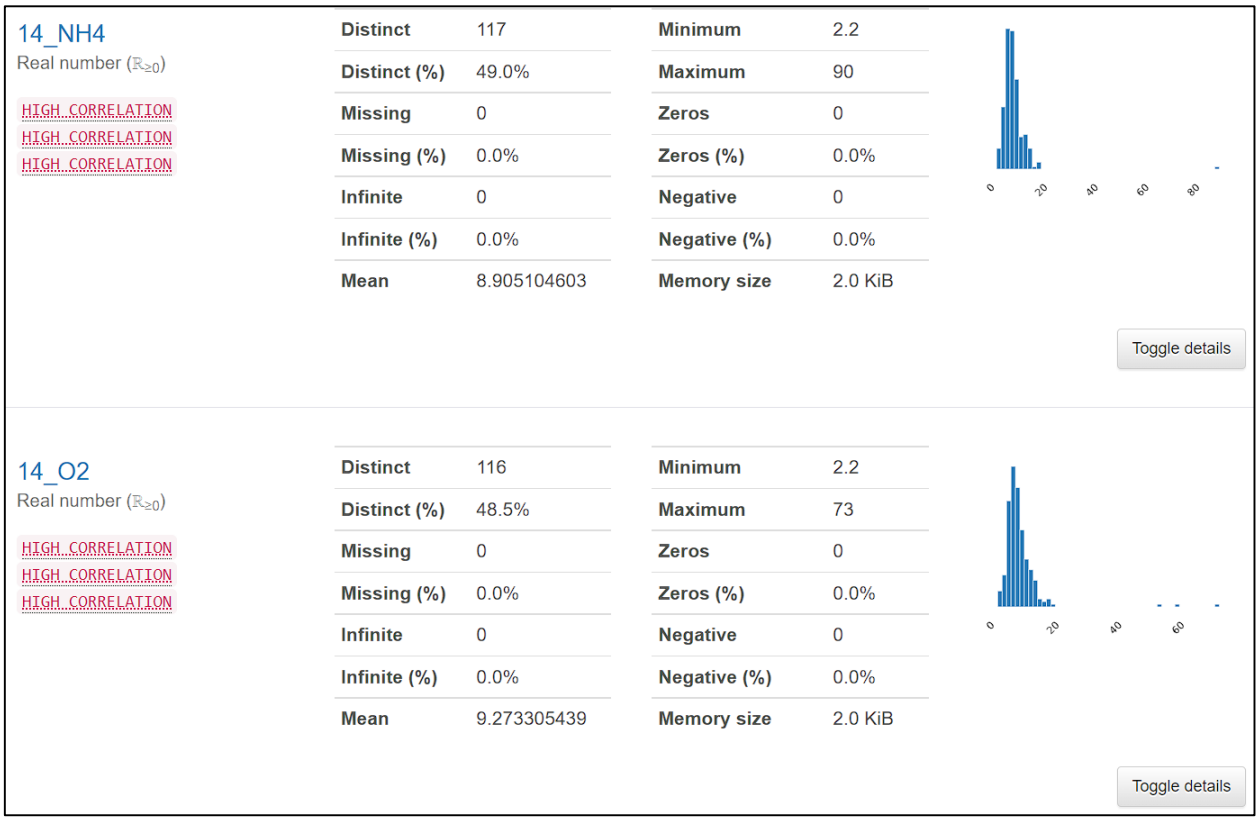


Рисунок 2.40 – Гістограми показників (продовження)

Однією з особливостей з особливостей «Pandas Profiling Report» є вкладка «Interactions», в якій можна побачити зображення кореляції між показниками (рис. 2.41).

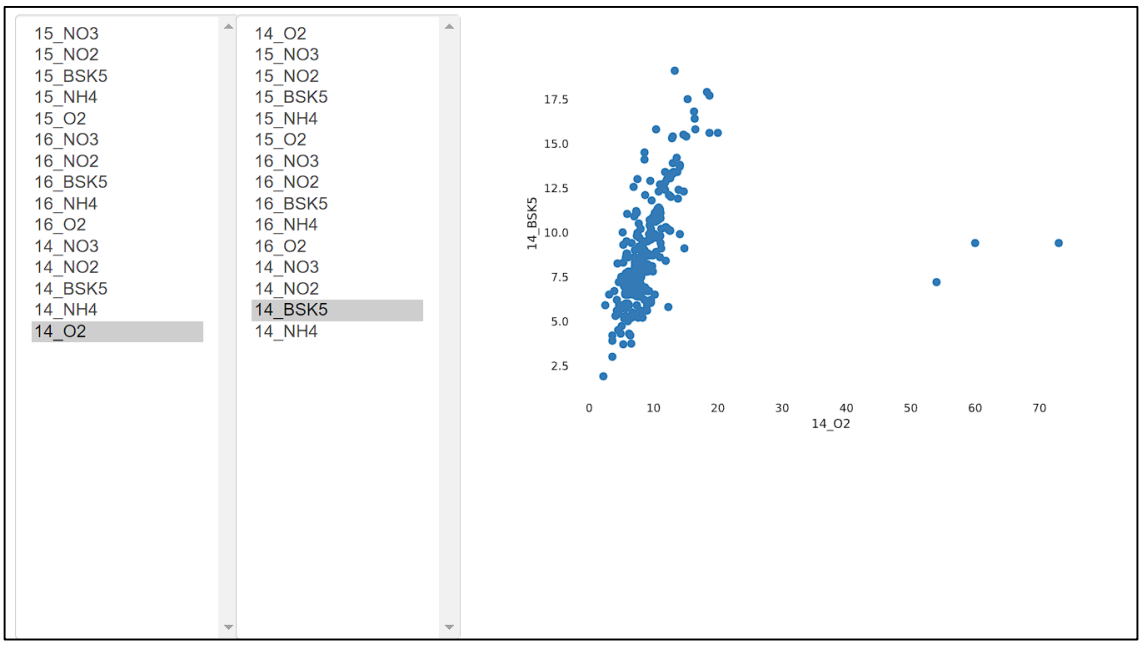


Рисунок 2.41 – Кореляція між 14_O2 і 14_BSK5

Тобто можна оцінити, які аномалії наскільки воно близьке до лінійної регресії як показано на рисунку 2.41. В такий спосіб можна зробити попереднє вивчення співвідношень між даними показників [9].

Коефіцієнт кореляції Пірсона (r) є мірою лінійної кореляції між двома змінними. Це значення лежить між -1 і +1, -1 вказує на загальну негативну лінійну кореляцію, 0 вказує на відсутність лінійної кореляції та 1 вказує на загальну позитивну лінійну кореляцію [10]. Крім того, r інваріантний при окремих змінах розташування та масштабу двох змінних, маючи на увазі, що для лінійної функції кут до осі x не впливає на r (рис. 2.42).

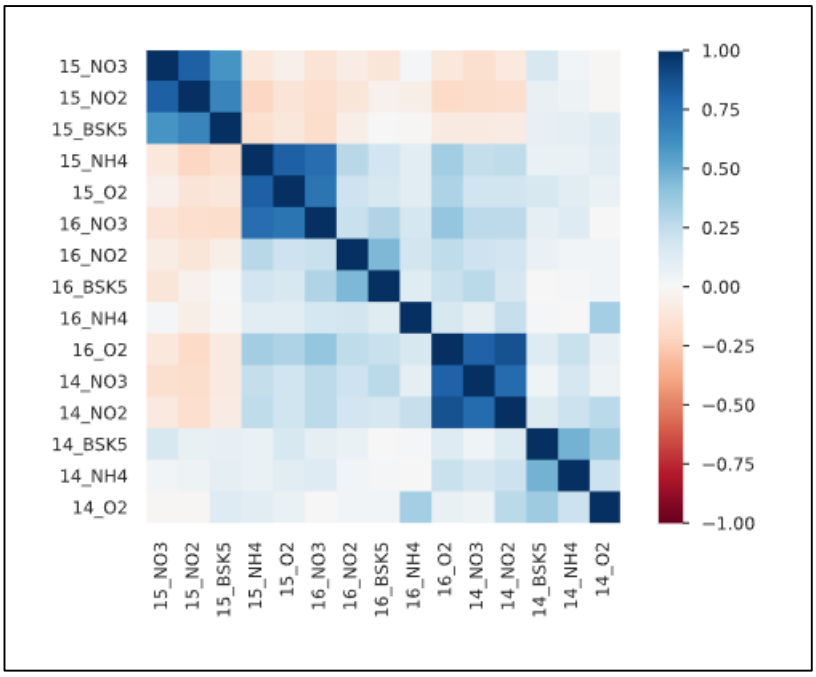


Рисунок 2.42 – Матриця кореляції Пірсона

Коефіцієнт рангової кореляції Спірмена (ρ) є мірою монотонної кореляції між двома змінними, і тому є кращим у ловлі нелінійних монотонних кореляцій, ніж p Пірсона. Це значення лежить між -1 і +1, -1, що вказує на загальну негативну монотонну кореляцію, 0 вказує на відсутність монотонної кореляції і 1 вказує на загальну позитивну монотонну кореляцію (рис. 2.43).

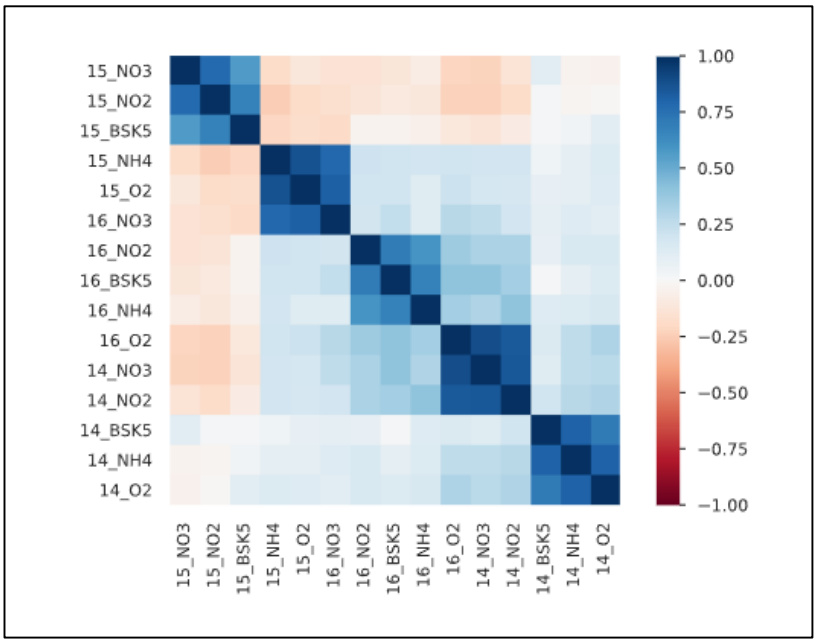


Рисунок 2.43 – Матриця кореляції Спірмена

Подібно до коефіцієнта рангової кореляції Спірмена, коефіцієнт кореляції рангу Кендалла (τ) вимірює порядковий зв'язок між двома змінними. Це значення лежить між -1 і +1, -1 вказує на загальну негативну кореляцію, 0 вказує на відсутність кореляції та 1 вказує на загальну позитивну кореляцію (рис. 2.44).

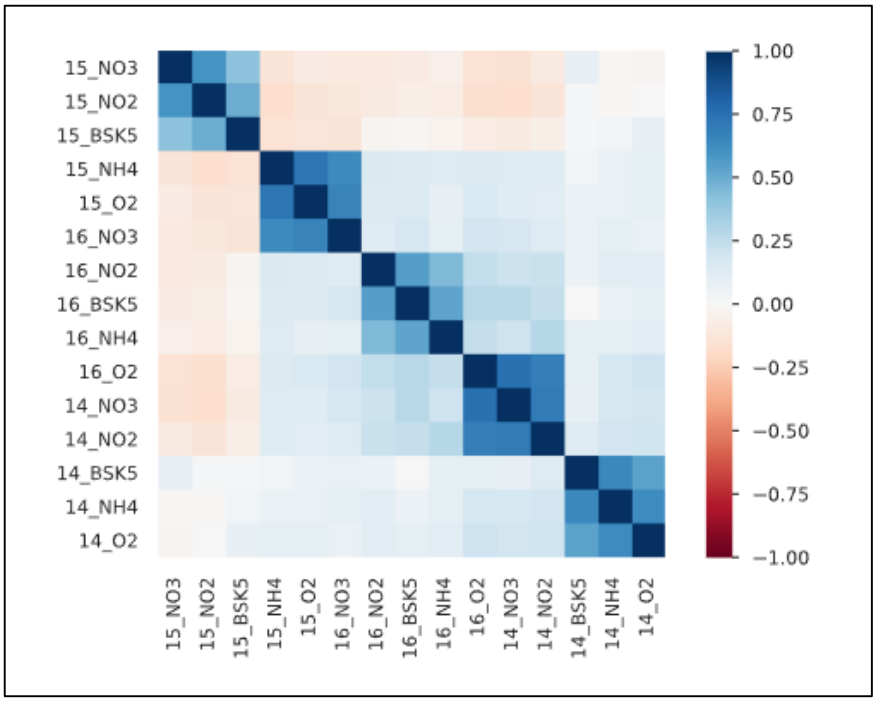


Рисунок 2.44 – Матриця кореляції Кендалла

Phik (φk) - це новий і практичний коефіцієнт кореляції, який працює послідовно між категоріальними, порядковими та інтервальними змінними, фіксує нелінійну залежність і повертається до коефіцієнта кореляції Пірсона у разі двовимірного нормального розподілу вхідних даних (рис. 2.45).

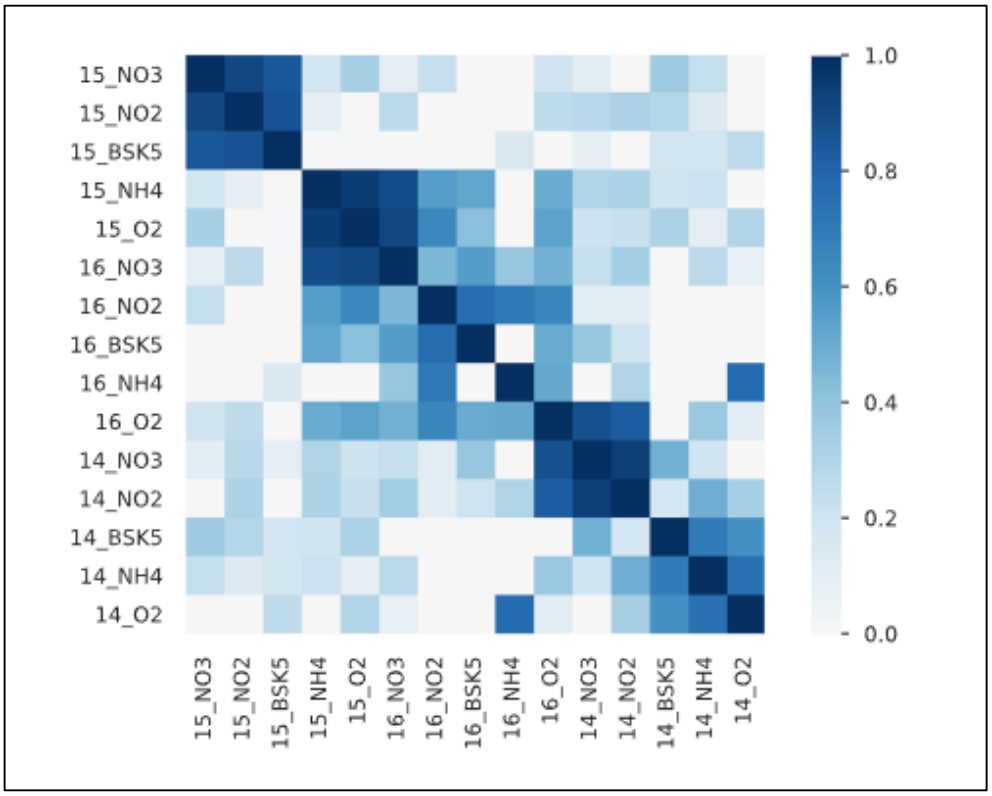


Рисунок 2.45 – Матриця кореляції Phik

Як ми можем бачити на матриці Пірсона, Спірмена, Кендалла кореляції даних вимірюється за шкалою від -1 до 1. Аналізуючи кореляцію ми можемо бачити, що фактор корелює на 80% із цільовою змінною O2 у 14 створі з більшістю речовин у цьому ж створі.[10-15].

Надалі ми бачимо діаграму пропущених даних, але оскільки ми відфільтрували їх вище то всі стовпці мають значення у комірці. Що продемонстровано на рисунку 2.46.

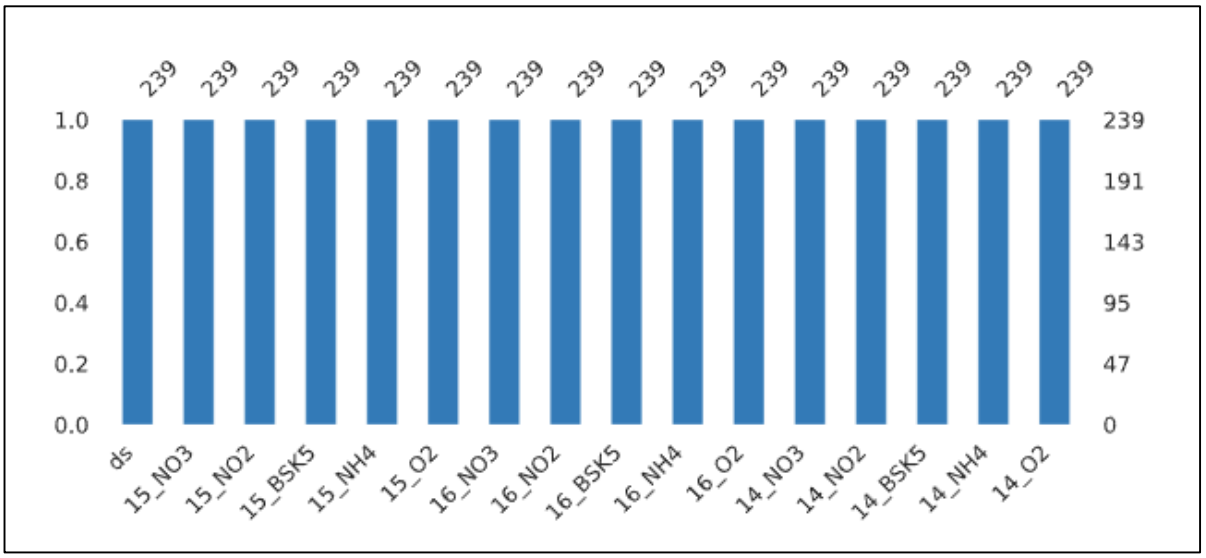


Рисунок 2.46 – Діаграма кількості пропущених даних

Виходячи з графіку на рисунку 2.17 ми вже знаємо, що у нас є аномальні дані, які погано впливають на кінцевий результат прогнозування, тому потрібно відтворити новий фільтр даних, щоб зменшити розкид між ними (рис. 2.47).

```
print(len(df))
for col in df.columns.tolist():
    df = df[df[col] <= float(df.quantile([.96])[col])]
df = df.reset_index(drop=True)
print(len(df))
df.describe()
```

Рисунок 2.47 – Фільтр даних

Тепер на побудованій таблиці після фільтрації даних видно, що розкид між даними зменшився і вони стали схожі на більш однорідні (рис. 2.48).

	14_BSK5	14_NH4	14_O2	15_BSK5	15_NH4	15_O2	16_BSK5	16_NH4	16_O2
count	163.000000	163.000000	163.000000	163.000000	163.000000	163.000000	163.000000	163.000000	163.000000
mean	4.711963	4.863067	4.754663	0.338368	0.323141	0.317669	7.995767	7.618650	7.693129
std	1.784020	1.824135	1.725002	0.224533	0.210732	0.232399	2.279181	2.197337	2.109158
min	1.300000	0.620000	0.630000	0.070000	0.000000	0.000000	1.900000	2.200000	2.200000
25%	3.450000	3.600000	3.350000	0.180000	0.180000	0.160000	6.400000	6.200000	6.200000
50%	4.500000	4.600000	4.700000	0.280000	0.260000	0.230000	7.800000	7.500000	7.600000
75%	5.950000	6.200000	6.200000	0.405000	0.410000	0.390000	9.600000	9.150000	9.100000
max	9.800000	8.800000	8.600000	1.080000	0.990000	1.100000	14.500000	12.900000	12.600000

Рисунок 2.48 – Таблиці відфільтрованих даних

Для того, щоб до кінця переконатися побудуємо графік (рис. 2.49).

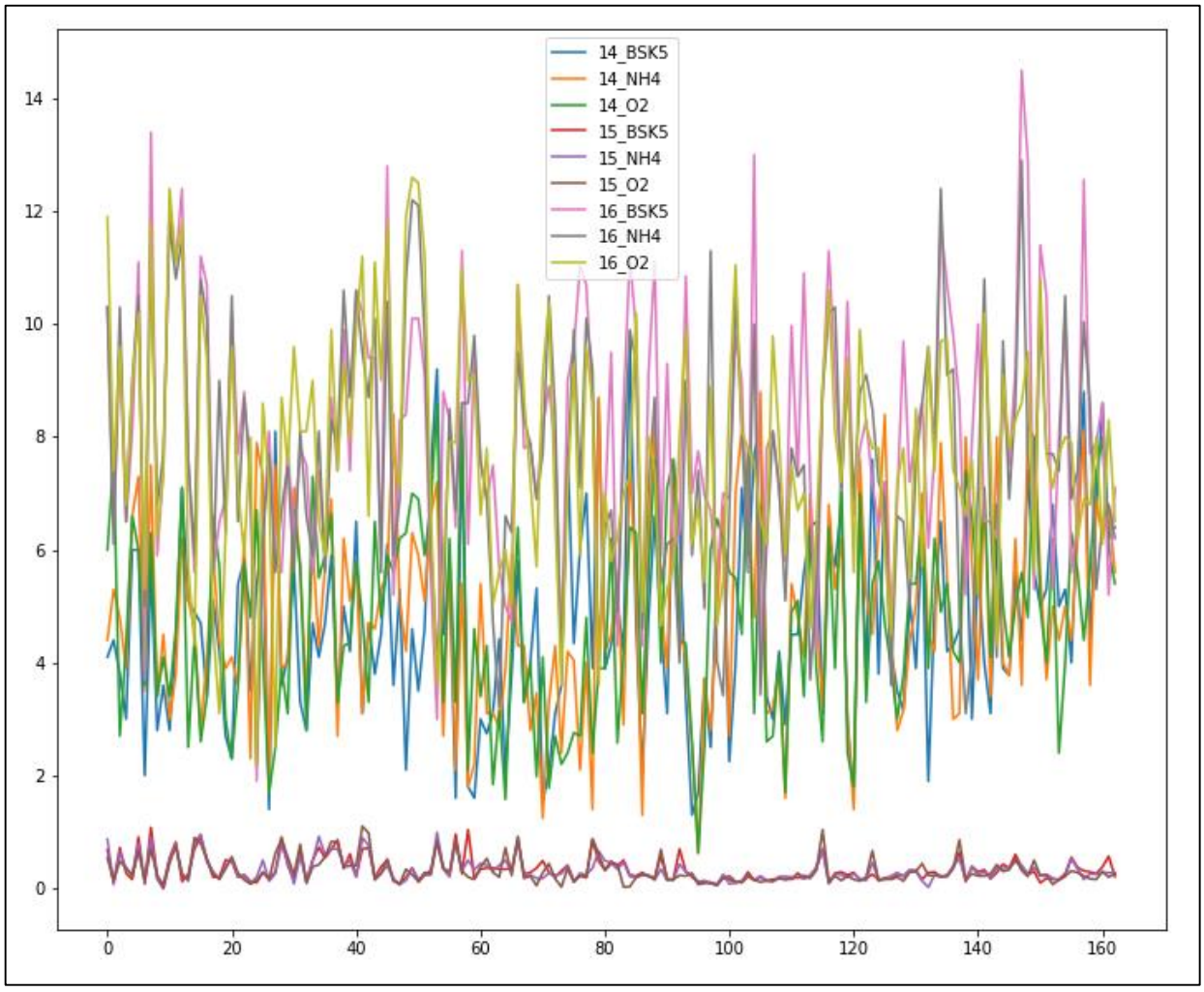


Рисунок 2.49 – Графік показників

Видно, що дані однорідні, тепер маємо змогу розпочати подальшу розробку прогнозування «таргету» для цього виділяєм його, щоб він був окремо (рис. 2.50).

```
target_name = '16_' + target_data_name
target_data = df.pop(target_name)
target_data
```

```
0      11.9
1       7.4
2       9.6
3       6.8
4       9.0
       ...
158     6.8
159     8.0
160     6.1
161     8.3
162     6.5
Name: 16_O2, Length: 163, dtype: float64
```

Рисунок 2.50 – Виділений «таргет»

Зараз розіб'ємо наші дані на тренувальні й тестові за допомогою команди «train_test_split», тестовий обсяг буде 40% (рис. 2.51).

```
train, test, target, target_test = train_test_split(df, target_data, test_size=0.4, random_state=0)
print(train.shape, test.shape)
```

```
(97, 8) (66, 8)
```

Рисунок 2.52 – Розподіл даних

Далі можна побачити статистику і кількість даних, які пішли до тренувальної та тестової дати показано на рисунках 2.53, 2.54.

```
train.describe([.05, .5, .96])
```

	14_BSK5	14_NH4	14_O2	15_BSK5	15_NH4	15_O2	16_BSK5	16_NH4
count	97.000000	97.000000	97.000000	97.000000	97.000000	97.000000	97.000000	97.000000
mean	4.941649	4.910412	4.797423	0.350268	0.341485	0.325711	7.943814	7.608763
std	1.797095	1.803313	1.768582	0.231415	0.226575	0.257026	2.308790	2.100462
min	1.520000	0.620000	0.630000	0.070000	0.000000	0.000000	3.000000	3.420000
5%	2.040000	1.760000	2.076000	0.100000	0.088400	0.066000	4.280000	4.086000
50%	4.700000	4.700000	4.800000	0.280000	0.260000	0.230000	7.800000	7.500000
96%	8.273600	8.149600	7.600000	0.868000	0.874800	0.883200	11.476000	10.912000
max	9.800000	8.800000	8.600000	1.040000	0.990000	1.100000	14.500000	12.900000

Рисунок 2.53 – Статистика тренувальних даних

```
test.describe()
```

	14_BSK5	14_NH4	14_O2	15_BSK5	15_NH4	15_O2	16_BSK5	16_NH4
count	66.000000	66.000000	66.000000	66.000000	66.000000	66.000000	66.000000	66.000000
mean	4.374394	4.793485	4.691818	0.320879	0.296182	0.305848	8.072121	7.633182
std	1.722818	1.865999	1.670272	0.214556	0.183353	0.191834	2.250334	2.348857
min	1.300000	1.300000	1.700000	0.091000	0.022000	0.025000	1.900000	2.200000
25%	3.100000	3.140000	3.325000	0.170000	0.172500	0.165000	6.500000	6.100000
50%	4.200000	4.550000	4.650000	0.255000	0.235000	0.240000	7.685000	7.630000
75%	5.675000	6.200000	5.990000	0.367500	0.385000	0.380000	9.575000	9.575000
max	8.600000	8.400000	8.000000	1.080000	0.900000	0.920000	13.400000	12.400000

Рисунок 2.54 – Статистика тестових даних

За для поліпшення роботи з даними потрібно їх стандартизувати у двох вибірках train і test використовуючи команду «StandardScaler» (рис. 2.55).

```
# Standartization data
scaler = StandardScaler()
train = pd.DataFrame(scaler.fit_transform(train), columns = train.columns)

# Display training data
train
```

Рисунок 2.55 – Стандартизація test і train

У свою чергу, тренувальні дані ще раз розділимо, виділимо в них валідаційні, це та частина «датасету», які необхідна для вибору найкращої моделі, щоб її потому застосувати для тестових даних [16-20]. Тут валідаційні дані вибираються обсягом 20% (рис. 2.56).

```
train_all = train.copy()
target_all = target.copy()
train, valid, target_train, target_valid = train_test_split(train_all, target_all, test_size=0.
2, random_state=0)
```

Рисунок 2.56 – Вибірка валідаційних даних

Виконуємо трьох разову кросс-валідацію для тренувальної дати, що відображено на рисунку 2.57.

```
cv_train = ShuffleSplit(n_splits=5, test_size=0.4, random_state=0)
```

Рисунок 2.57 – Валідація train

Задаємо часові інтервали для датасетів, перший - інтервал буде з 2016 по 2018, другий - з 2019 до 2022. Рисунок 2.58.

```
final_data = 2021
years_num_period = 2
year1_start = 2016
year1_end = year1_start + years_num_period
year2_start = year1_start + years_num_period + 1
year2_end = year2_start + years_num_period if (year2_start + years_num_period) < final_d
ata else final_data
print(f"Interval 1 - from {year1_start} to {year1_end}, Interval 2 - from {year2_start}
to {year2_end}")
```

Рисунок 2.58 – Часові інтервали

Тепер у нас є 2 датасети за заданими часовими інтервалами, які відображені на рисунках 2.59, 2.60. На наступному етапі будемо їх порівнювати.

	14_NO3	14_NO2	14_BSK5	14_NH4	14_O2
0	8.00	7.60	9.00	8.3	8.6
1	7.60	7.60	8.80	8.8	8.9
2	0.16	1.10	11.30	10.2	10.6
3	0.47	0.75	9.10	10.3	8.1
4	0.20	1.91	7.05	7.1	7.2
5	3.00	2.50	10.40	9.1	9.4
6	0.55	0.63	6.60	6.9	5.6
7	0.46	0.60	7.80	8.8	9.9
8	0.30	0.12	17.50	14.1	15.3
9	1.30	1.61	8.30	9.1	8.2
10	4.03	5.80	7.20	8.5	7.8
11	6.80	5.80	8.80	9.7	9.2

Рисунок 2.59 – Датасет №1

	14_NO3	14_NO2	14_BSK5	14_NH4	14_O2
0	2.80	3.33	6.8	7.7	7.50
1	0.53	3.00	8.6	8.6	11.00
2	3.10	3.50	9.9	9.6	14.10
3	0.00	0.00	6.5	6.5	7.70
4	3.18	4.56	6.2	4.1	4.30
5	0.27	0.28	8.8	9.7	9.10
6	0.28	0.40	7.5	6.9	7.80
7	5.70	1.40	9.2	8.6	8.30
8	0.15	0.81	14.5	12.9	8.60
9	0.00	0.00	12.9	7.7	9.50
10	4.37	3.16	5.3	6.3	5.65
11	1.50	2.80	11.4	10.2	10.80

Рисунок 2.60 – Датасет №2

Візуалізуємо два датасети і можем бачити, що в обох інтервалах по заданих роках дані знаходяться фактично на одному рівні тобто однорідні рисунок 2.61

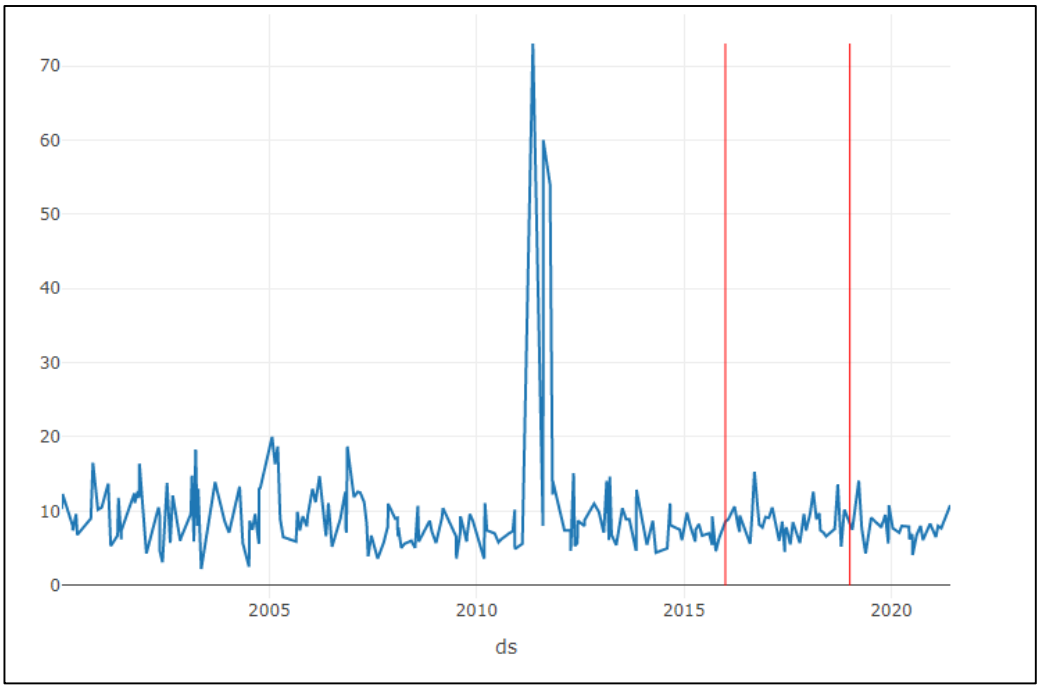


Рисунок 2.61 – Візуалізація дати першого і другого періоду

2.3 Порівняння за допомогою SweetViz

Перший інструментарій - це бібліотека SweetViz інсталюємо та підтягуємо його рисунок 2.62. Команда «from IPython.display import IFrame» дозволяє виводити графіки прямо в результаті запуску.

```
!pip install sweetviz
import sweetviz as sv
from IPython.display import IFrame
```

Рисунок 2.62 – Інсталяція SweetViz

Команда «analyze(df1)». Перший із цих етапів виводиться одразу і на екран і в html, потім можна в розділі «Data» окремо розглянути рисунок 2.63.

```
df1_report = sv.analyze(df1)
df1_report.show_html('Analyst1.html')
IFrame(src = 'Analyst1.html',width=1000,height=600)
```

Рисунок 2.63 – Візуалізація датафрейму «df1»

І так, що вона показує, ряд показників, по кожній ознаці виводить цілий ряд статистики MAX, Q1/Q3 - це квантиль, медіану, середнє і ще ряд інших показників.

Так воно виводить, спочатку по одному періоду «df1», а потім по іншому «df2» і теж зберігає в «Analist2.html». Можна окремо вивчити і подивитись на кожен з них рисунок 2.64, 2.65.

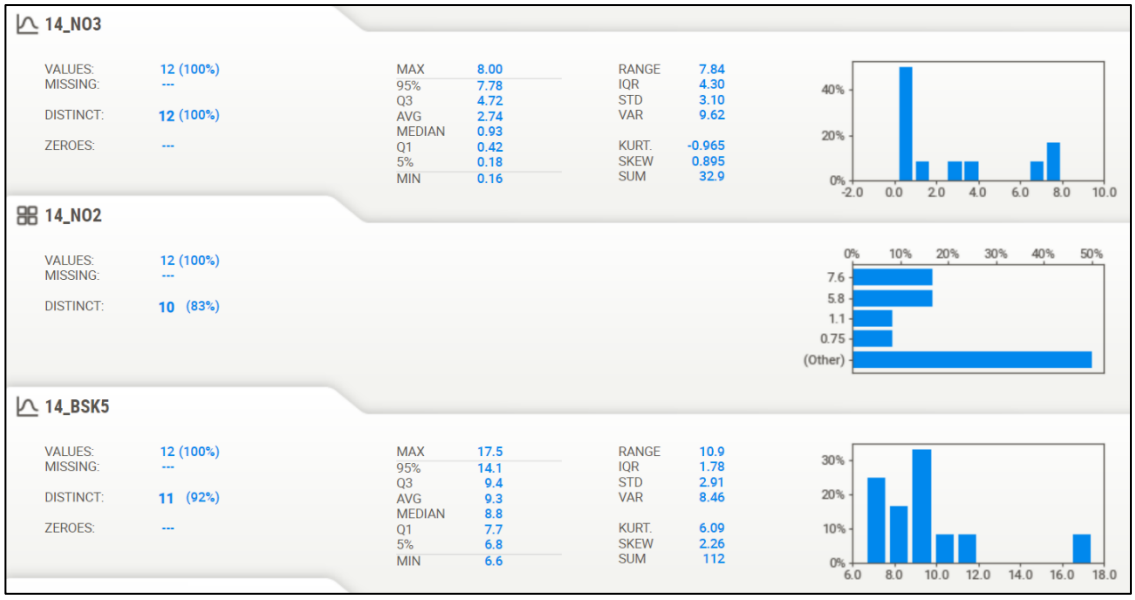


Рисунок 2.64 – Статистика «df1»

Рисунок 2.65 – Статистика «df1» (продовження)

Також візуалізуємо дані датасету «df2», щоб у подальшому порівняти їх з даним «df1» рисунок 2.66, 2.67.

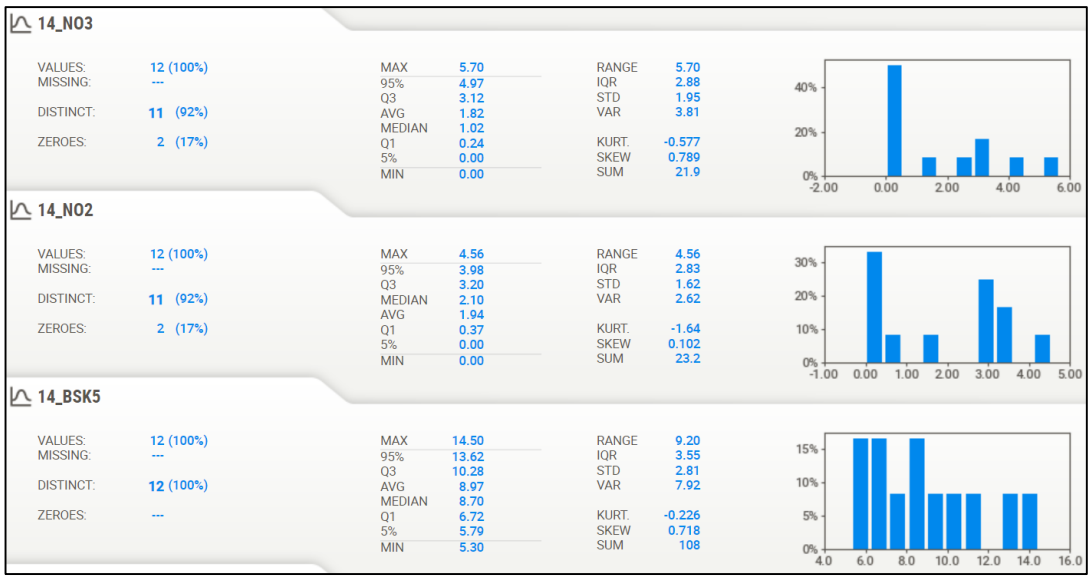


Рисунок 2.66 – Статистика «df2»

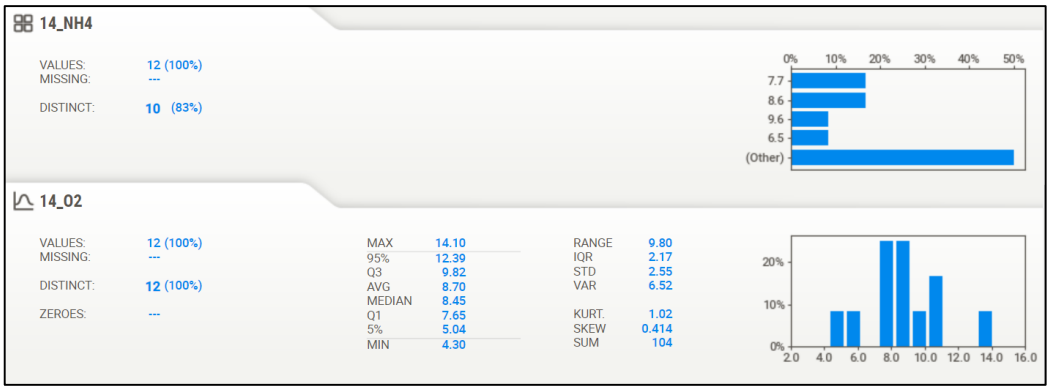


Рисунок 2.67 – Статистика «df2»

Але найбільш цікавить наявність в цій бібліотеці інструменту «compare(df1,df2)», воно порівнює «df2» з «df1», тобто роки з 2016 по 2018 із роками 2019 по 2021 і от в такому вигляді виводить як показано на рисунку 2.68.

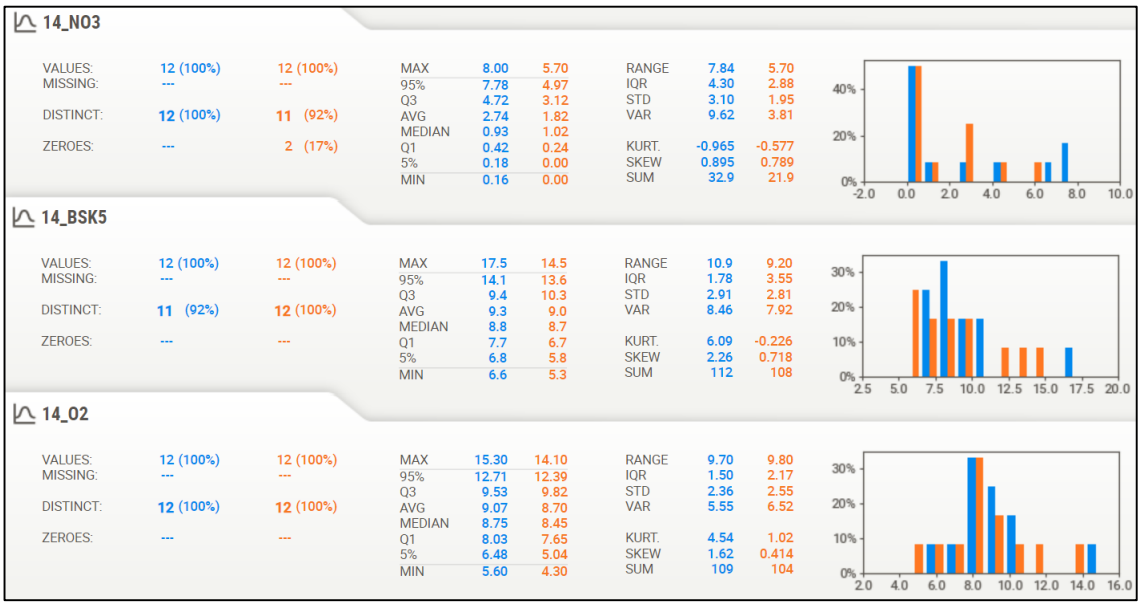


Рисунок 2.68 – Порівняння «df1» і «df2»

Можемо бачити на побудованих гістограмах, що дані по показника 14_O2 і 14_BSK5 співставні за порівнюваними роками.

2.4 Порівняння за допомогою AutoViz

Другий інструментарій – це бібліотека AutoViz тут так само, інсталюємо і підтягуємо дані рисунок 2.69.

```
!pip install autoviz

df1['class'] = 0
df2['class'] = 1
data = pd.concat([df1, df2], axis=0)
data
```

Рисунок 2.67 – Інсталяція AutoViz

Аналізувати доцільно, коли існує показник, по якому можно дивитися як один період відрізняється від іншого, у нас - O2.

Можливі різні способи роботи, але воно працює таким чином: передаєш йому один датасет, в якому якимось чином по певній ознаці відрізняються ті періоди, що ви хочете порівняти, це може бути 4 періоди і 6, воно найбільш ефективне коли періодів побільше .

В даному випадку ми просто поєднуємо той, що «df1» і «df2». Далі бачимо об'єднаний датасет на 24 значення рисунок 2.70.

	14_NO3	14_NO2	14_BSK5	14_NH4	14_O2	class
0	8.00	7.60	9.00	8.3	8.60	0
1	7.60	7.60	8.80	8.8	8.90	0
2	0.16	1.10	11.30	10.2	10.60	0
3	0.47	0.75	9.10	10.3	8.10	0
4	0.20	1.91	7.05	7.1	7.20	0
5	3.00	2.50	10.40	9.1	9.40	0
6	0.55	0.63	6.60	6.9	5.60	0
7	0.46	0.60	7.80	8.8	9.90	0
8	0.30	0.12	17.50	14.1	15.30	0
9	1.30	1.61	8.30	9.1	8.20	0
10	4.03	5.80	7.20	8.5	7.80	0
11	6.80	5.80	8.80	9.7	9.20	0
0	2.80	3.33	6.80	7.7	7.50	1
1	0.53	3.00	8.60	8.6	11.00	1
2	3.10	3.50	9.90	9.6	14.10	1
3	0.00	0.00	6.50	6.5	7.70	1
4	3.18	4.56	6.20	4.1	4.30	1
5	0.27	0.28	8.80	9.7	9.10	1
6	0.28	0.40	7.50	6.9	7.80	1
7	5.70	1.40	9.20	8.6	8.30	1
8	0.15	0.81	14.50	12.9	8.60	1
9	0.00	0.00	12.90	7.7	9.50	1
10	4.37	3.16	5.30	6.3	5.65	1
11	1.50	2.80	11.40	10.2	10.80	1

Рисунок 2.70 – Об’єднані «df1» і «df2»

Підключаємо інструмент «AutoViz_Class» зазвичай він виводить дані з файлу, так як він запрограмований на те, щоб виводити дані з файлу який він зчитує, тому один із способів – це зберегти дані у файл, а потім зчитати але є

простіший, вхідні дані пусто, формат дата, «таргет» по-якому розрізняти класи «class» далі все автоматично. Також важлива ось ця команда «%matplotlib inline», якщо її не зробити, графіки не виведуться рисунок 2.71.

```
from autoviz.AutoViz_Class import AutoViz_Class
%matplotlib inline
AV = AutoViz_Class()
AV.AutoViz(filename='',
           dfte=data,
           depVar='class',
           verbose=1,
           lowess=True,
           max_rows_analyzed=data.shape[0],
           max_cols_analyzed=data.shape[1])
```

Рисунок 2.71 – Підключення інструменту «AutoViz_Class»

Далі вже на побудованих графіках можливо побачити порівняння по ось цих класах по кожній ознаці рисунок 2.72, 2.73.

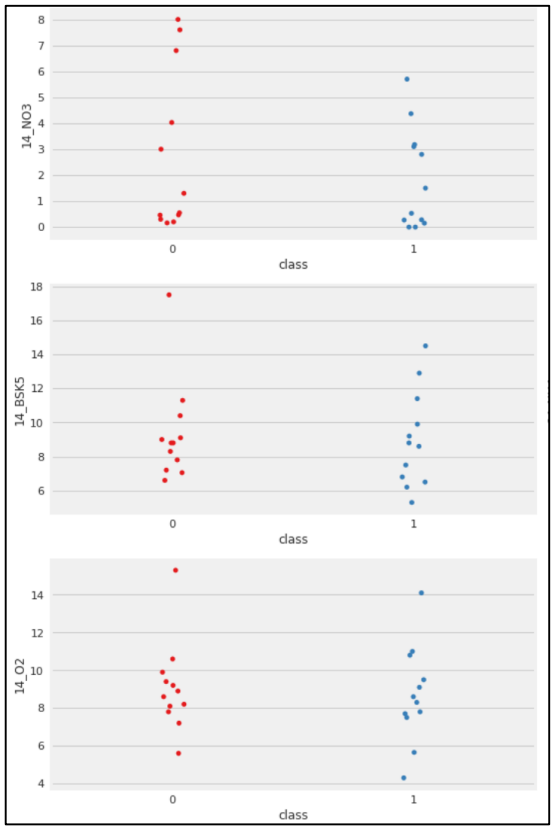


Рисунок 2.72 – Порівняння по кожній ознаці

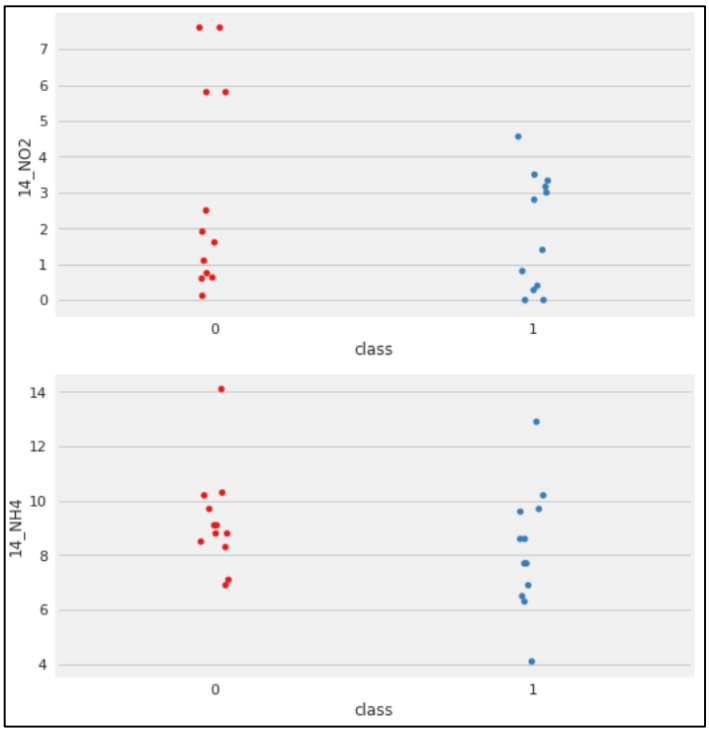


Рисунок 2.73 – Порівняння по кожній ознаці (продовження)

На рисунку 2.74 відображено розкид значень NO2, NO3, NH4, BSK5 і O2 видно, що значень доволі таки мало і важко відстежити будь яку залежність.

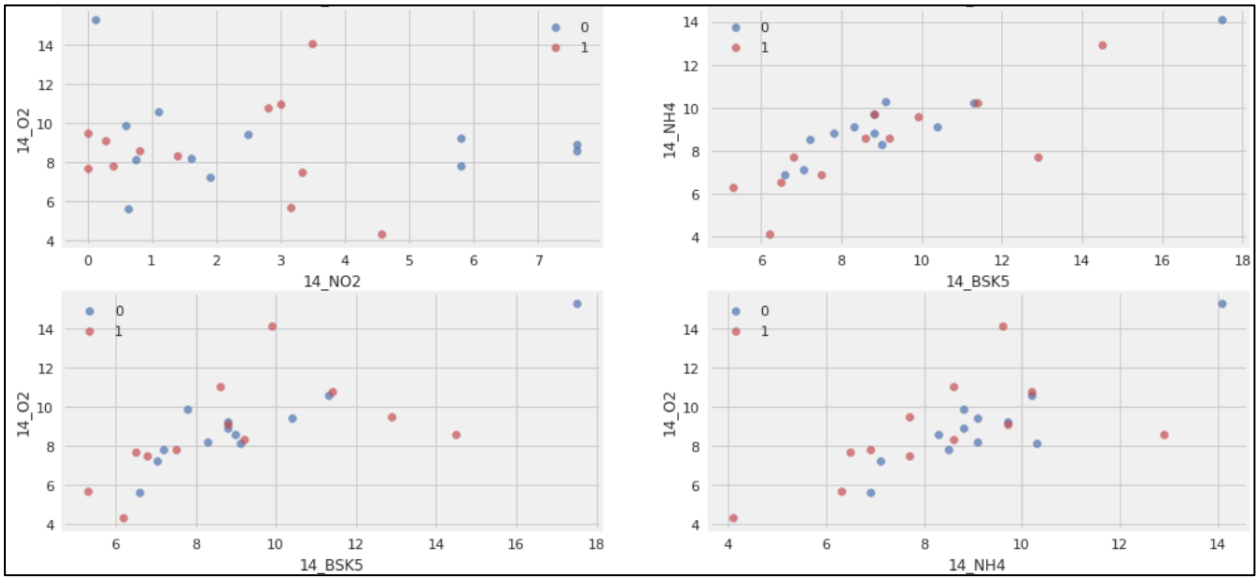


Рисунок 2.74 – Розкид значень показників

Гістограми, так само по цих періодах, можна взяти не по 2 роки, а по 4 або 6. На побудованих гістограмах бачимо, що дані за останніми датами близько схожі рисунок 2.75, 2.76.

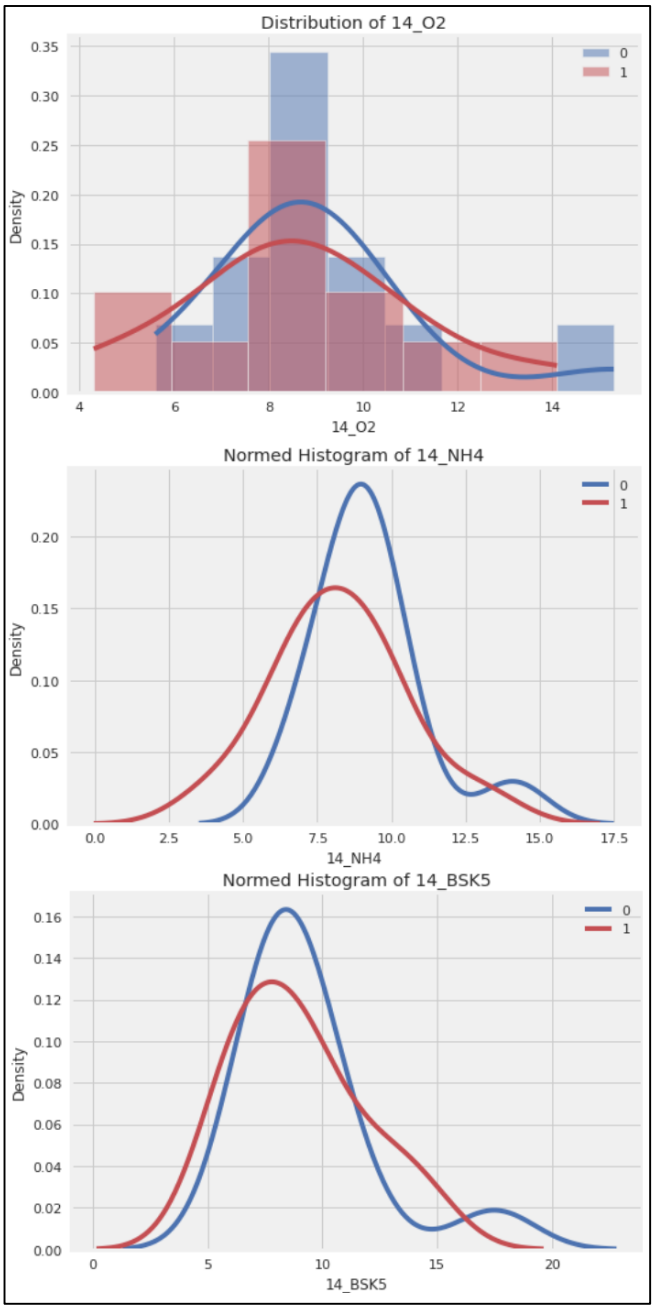


Рисунок 2.75 – Гістограми кореляції за роками

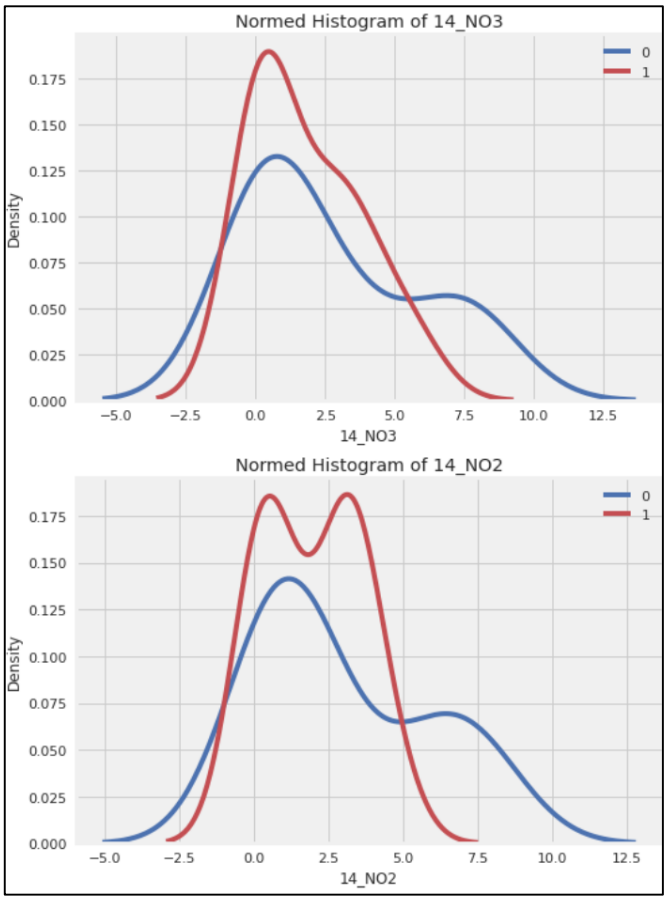


Рисунок 2.76 – Гістограми кореляції за роками (продовження)

Тут показано стільки в «таргеті» значень, але ми відразу робили їх однакової довжини то вони відповідно однакові рисунок 2.77.

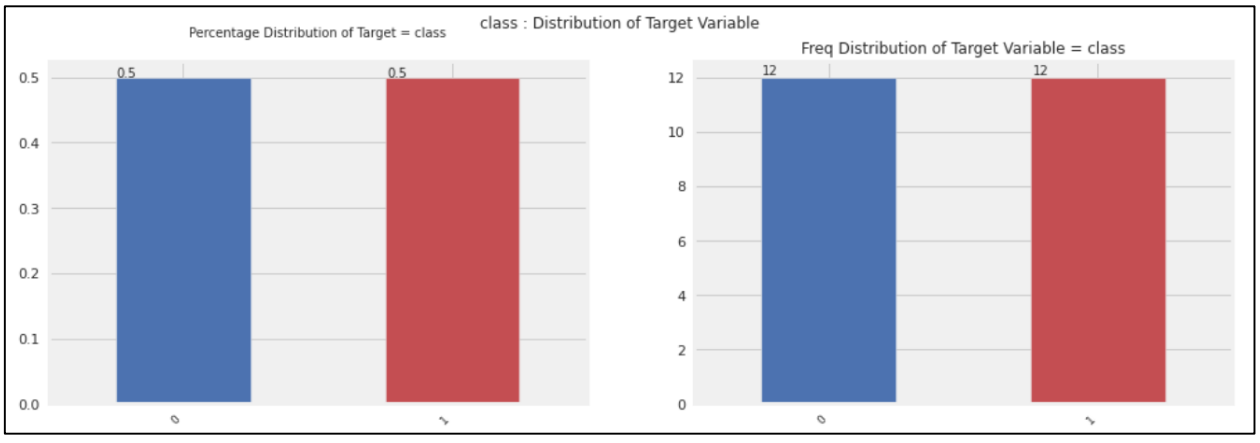


Рисунок 2.77– Кількість значень «таргету»

А ось тут цікаво це - «Box Plots» розкид значень, так звані «вусаті» діаграми, їх краї це аномальні значення, а основна частина між 1 і 3 квантилем

показуються таким боксом, і можна бачити яким чином ці розкидки між собою співставні за останні і передостанні два роки рисунок 2.78-2.80.

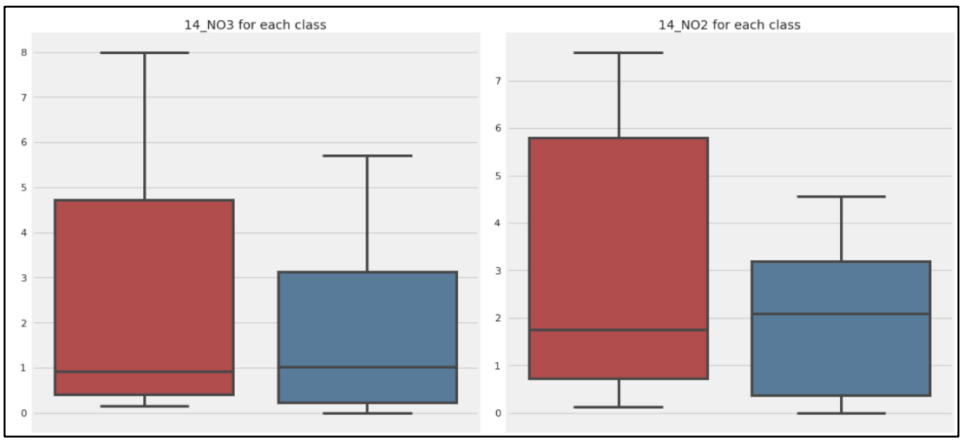


Рисунок 2.78 – Діаграми «Box Plots»

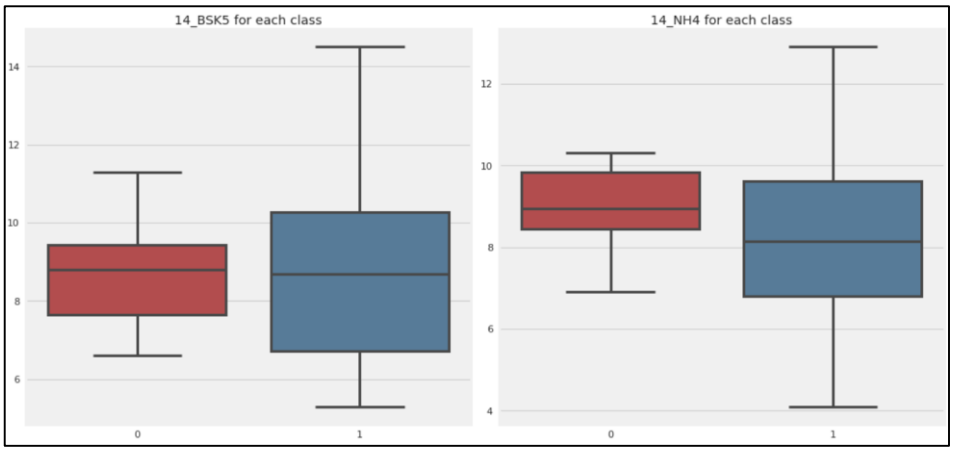


Рисунок 2.79 – Діаграми «Box Plots» (продовження)

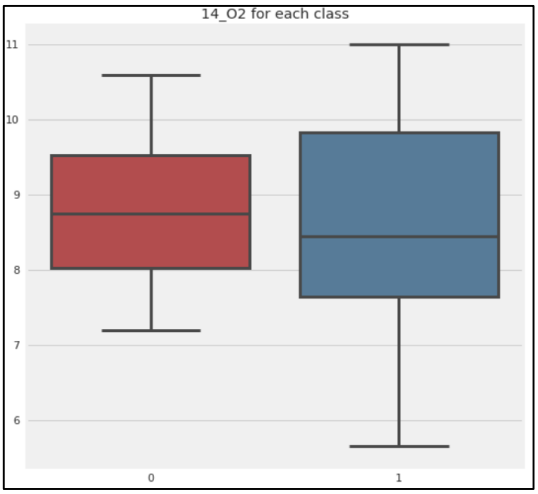


Рисунок 2.80 – Діаграми «Box Plots» (продовження)

Коефіцієнт кореляції тепловий, тобто чим більший коефіцієнт кореляції тим більше воно «розжарене» і ближче до жовтого тим більша кореляції між показниками, чим кореляція менше тим більш воно чорне рисунок 2.81.

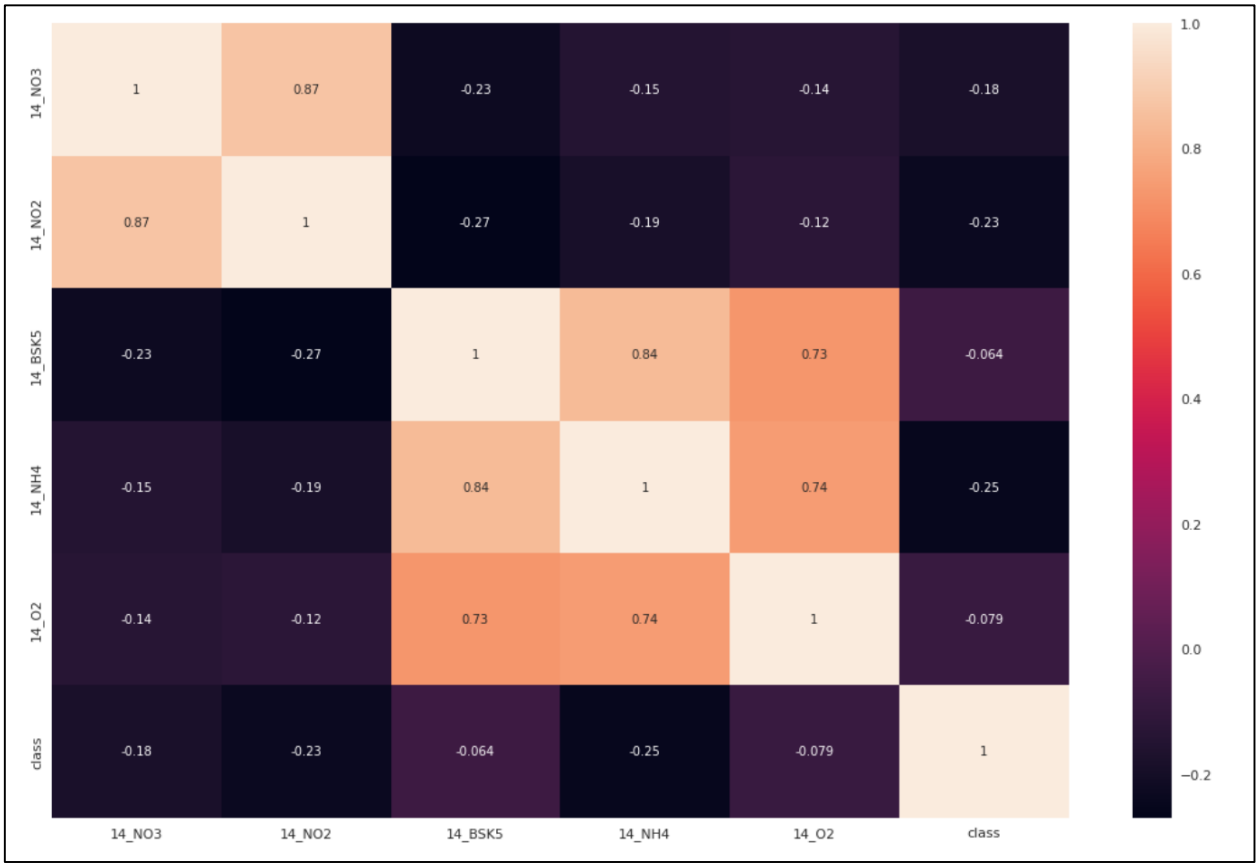


Рисунок 2.81 – Матриця кореляції

Тут очікувані ці 3 параметри, які доволі таки щільно корелюють один між одним, а саме: BSK5, NH4 і O2. На жаль AutoViz виводить не так багато інформації, воно само вирішує, що може бути цікавим і виводить, якщо класів більше може й регресію побудувати, у вигляді скрипки додати графіки.

Такий приклад автоматичний бібліотек, які одразу дозволяють побудувати ряд графіків, які можна використати у подальшому прогнозуванні.

Згідно з вище вказаних підготовлених даних можна сказати, що у нас є все для побудови подальшої машинної-моделі. В котрій вже можливо буде підбивати підсумки прогнозування O2 вмісту в річці Південний Буг.

2.5 Висновки

У даному розділі здійснено розвідувальний аналіз початкових даних задачі прогнозування, що вирішується в роботі на основі методів статистичного та кореляційного аналізу за допомогою бібліотек AutoViz і SweetViz. В наступному розділі результати цих аналізів будуть використані для побудови моделей машинного навчання для прогнозування O2 вмісту в річці Південний Буг.

3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ТА ІДЕНТИФІКАЦІЯ ОПТИМАЛЬНОЇ МОДЕЛІ ПРОГНОЗУВАННЯ

3.1 Розробка алгоритму функціонування інформаційної технології прогнозування

Прогнозування вмісту розчиненого у воді кисню в річці Південний Буг пропонується здійснювати за алгоритмом, який складається з таких кроків:

1. проаналізувати початкові часові ряди на наявність детрендованих кореляцій;
2. сформувати цільові ознаки;
3. стандартизувати ознаки за методом «sklearn.preprocessing.StandardScaler»;
4. сформувати навчальну, валідаційну та тестову вибірку для різних моделей;
5. навчати моделі за допомогою навчальної вибірки. В процесі навчання використати валідаційну вибірку для уникнення ефекту перенавчання;
6. обрати оптимальну модель за заданою метрикою на тестовій вибірці, яка не приймала участь у навчанні;
7. донавчити оптимальні моделі з попереднього кроку на розширеному наборі даних: навчальна та валідаційна вибірки. Здійснити прогнозування на основі тестовоі вибірки;
8. визначити точність прогнозування;
9. проаналізувати важливість ознак, що найбільше впливають на результат прогнозування.

На рис. 3.1 наведено блок-схема алгоритму функціонування інформаційної технології прогнозування вмісту розчиненого у воді кисню в річці Південний Буг.

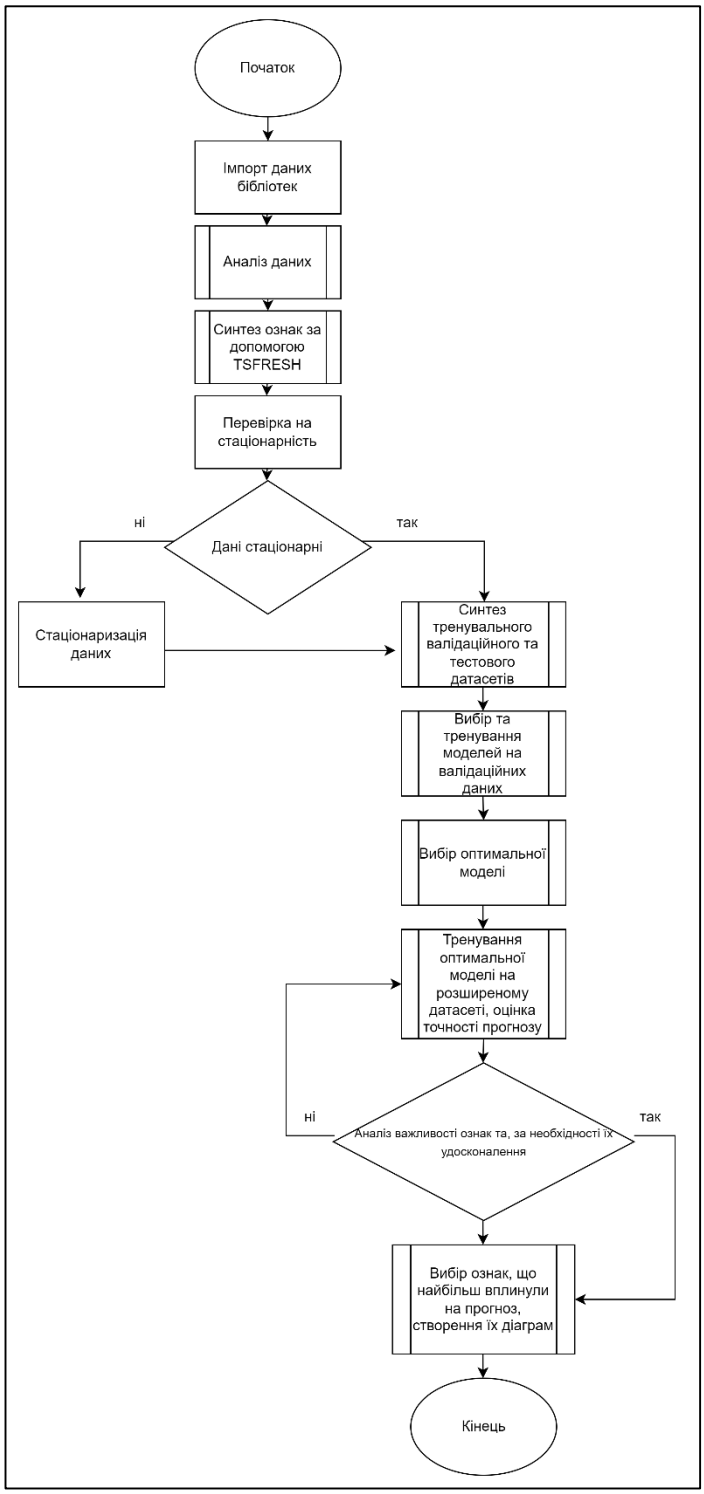


Рисунок 3.1 – Блок-схема алгоритму функціонування інформаційної технології прогнозування вмісту розчиненого у воді кисню в річці Південний Буг

3.2 Прогнозування методом Facebook Prophet

Модель FB Prophet ефективна лише для значення самої послідовності, а не для її різниці, оскільки вона занадто зашумлена. Для підвищення точності прогнозу рекомендується використовувати FB Prophet для моделювання 2021 року. На рисунку 3.2 можна бачити розмір побудованого набору даних в якому: 10 – валідаційні, 10 – тестові, і 219 – тренувальні.

```
Origin dataset has 239 rows and 2 features
Get training dataset with 219 rows
Get validation dataset with 10 rows
Get test dataset with 10 rows
```

Рисунок 3.2 – Розмір побудованих тренувального, валідаційного та тестового наборів даних

Бібліотека Facebook Prophet була запущена компанією Facebook як API для виконання прогнозування, пов'язаного з даними часових рядів. Бібліотека настільки потужна, що здатна обробляти стаціонарність даних, а також компоненти, пов'язані з сезонністю. Під стаціонарністю мається на увазі, що середнє, дисперсія і коваріація в даних повинні бути постійними, якщо розділимо дані на сегменти за часом, а сезонність означає той же тип тенденції, якому слідують дані, якщо розділити їх на часові інтервали [17].

Задаємо такі моделі:

– Стандартні параметри для річної та тижневої сезонності налаштовано так, щоб вони були відсутні, щоб можна вручну встановлювати їх окремо та мати контрольований вибір найкращих параметрів: yearly_seasonality=False, weekly_seasonality=False;

– Без щоденної сезонності (можлива присутність даних, але за щодобовими даними ідентифікувати неможливо): daily_seasonality = False;

– Точки зміни орієнтації слід шукати в усьому діапазоні даних (за стандартом шукають тільки 80% даних): changepoint_range = 1;

– Ряд дуже нестабільний, тому в першому наближенні необхідно встановити велике значення навколо точки зміни, щоб можливо розсіяти значення, тобто низька регуляризація: changepoint_prior_scale = 0,5;

– Аномальні дати ідентифіковані: holidays = holidays_df. Використаємо сезонність із власними параметрами;

– Форма ряду показує, що потрібно брати мультиплікативний, тобто коли всі складові перемножуються, оскільки він явно є нелінійним: seasonality_mode = 'multiplicative'.

Використовуємо різні варіанти пошуку параметрів моделі. Результати ідентифікації оптимальних параметрів і структури моделі наведені на рисунку 3.3.

```
model.add_seasonality(name='seasonality', period=period_days,
                      fourier_order=fourier_order_seasonality,
                      mode = 'multiplicative', prior_scale = 0.5)
```

Рисунок 3.3 – Засвідчувальні оптимальні параметри та структура моделі прогнозування

На рисунку 3.4 показано приклад того, як модель прогнозує дані перевірки

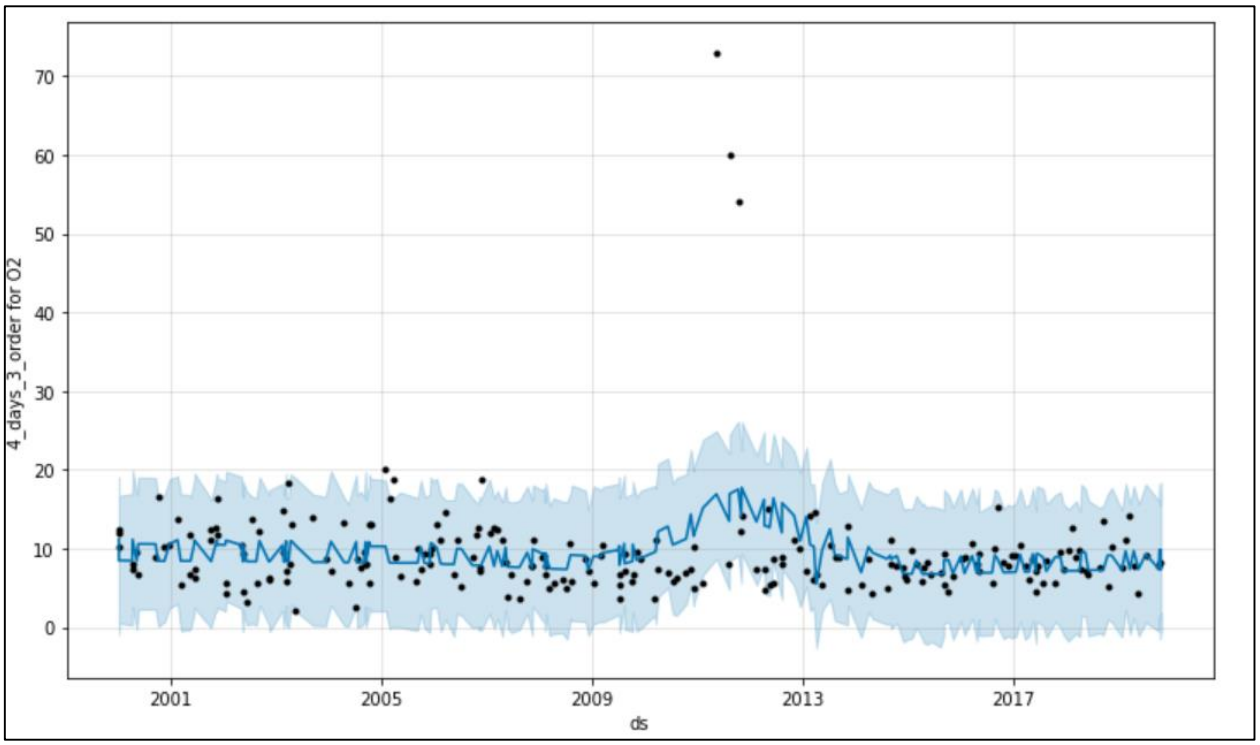


Рисунок 3.4 – Результат прогнозування курсу на основі Facebook Prophet

В загальному модель прогнозу з використанням технології Facebook Prophet показав себе візуально не погано, але в деяких місцях є не суттєві похибки. Якщо брати в порівняння інші моделі, то звичайно знайшлися моделі краще. Всі прогнози з побудованої моделі зберігаються в таблиці результатів і аналізуються разом з результатами інших моделей – про це мова піде нижче.

3.3 Прогнозування з використанням ARIMA

Побудуємо модель для прогнозування ознаки «O2» розчиненого у воді кисню як часовий ряду за допомогою моделі ARIMA.

ARIMA відноситься до класу моделей, які пояснюють даний часовий ряд на основі його минулих значень – тобто власних лагів і помилок прогнозу. Рівняння може бути використано для прогнозування майбутніх значень. Будь-який "несезонний" часовий ряд, який демонструє закономірності і не є випадковим білим шумом, може бути змодельований за допомогою моделі ARIMA [18].

Таким чином, ARIMA, скорочено від AutoRegressive Integrated Moving Average, – це алгоритм прогнозування, заснований на ідеї, що інформація про минулі значення часового ряду може бути використана для прогнозування майбутніх значень. Моделі ARIMA задаються трьома параметрами порядку: (*p, d, q*), де, p-порядок AR-терміна; d-кількість диферентів, необхідне для того, щоб зробити часовий ряд стаціонарним; q-порядок члена MA.

Авторегресія AR (*p*) - регресійна модель, яка використовує залежний зв'язок між поточним спостереженням і спостереженнями за попередній період. Компонент авторегресії(AR (*p*)) відноситься до використання минулих значень у рівнянні регресії для часового ряду.

Інтеграція I (*d*) - використовує диференціювання спостережень (віднімання спостереження з спостереження на попередньому часовому кроці) для того, щоб зробити часовий ряд стаціонарним. Диференціювання передбачає віднімання поточних значень ряду з його попередніх значень d раз.

Ковзне середнє MA (*q*) - модель, що використовує залежність між спостереженням і залишковою помилкою від моделі ковзного середнього, застосованої до запізнілих спостережень. Компонент ковзного середнього зображує помилку моделі як комбінацію попередніх членів помилки. Порядок q являє собою кількість членів, які повинні бути включені в модель.

Для початку спробуємо оцінити значення параметрів *p, d, q*. де вибирається різниця, що стаціонарізує ряд з використанням критерію Діка–Фуллєра:

– Якщо сам ряд стаціонарний: $d = 0$;

– якщо перша різниця ряду нерухома: $d = 1$;

– якщо стаціонарною буде перша різниця ряду: $d = 2$ і т. д.

Другий підхід полягає в тому, щоб якомога швидше візуально оцінити, значення яких параметрів *d* автокореляції та часткової автокореляції потрапляють в «коридор» p-value = 0,05 (на графіку зазвичай представлений темно-синьою областю навколо осі абсцис) (рис. 3.5, 3.6).

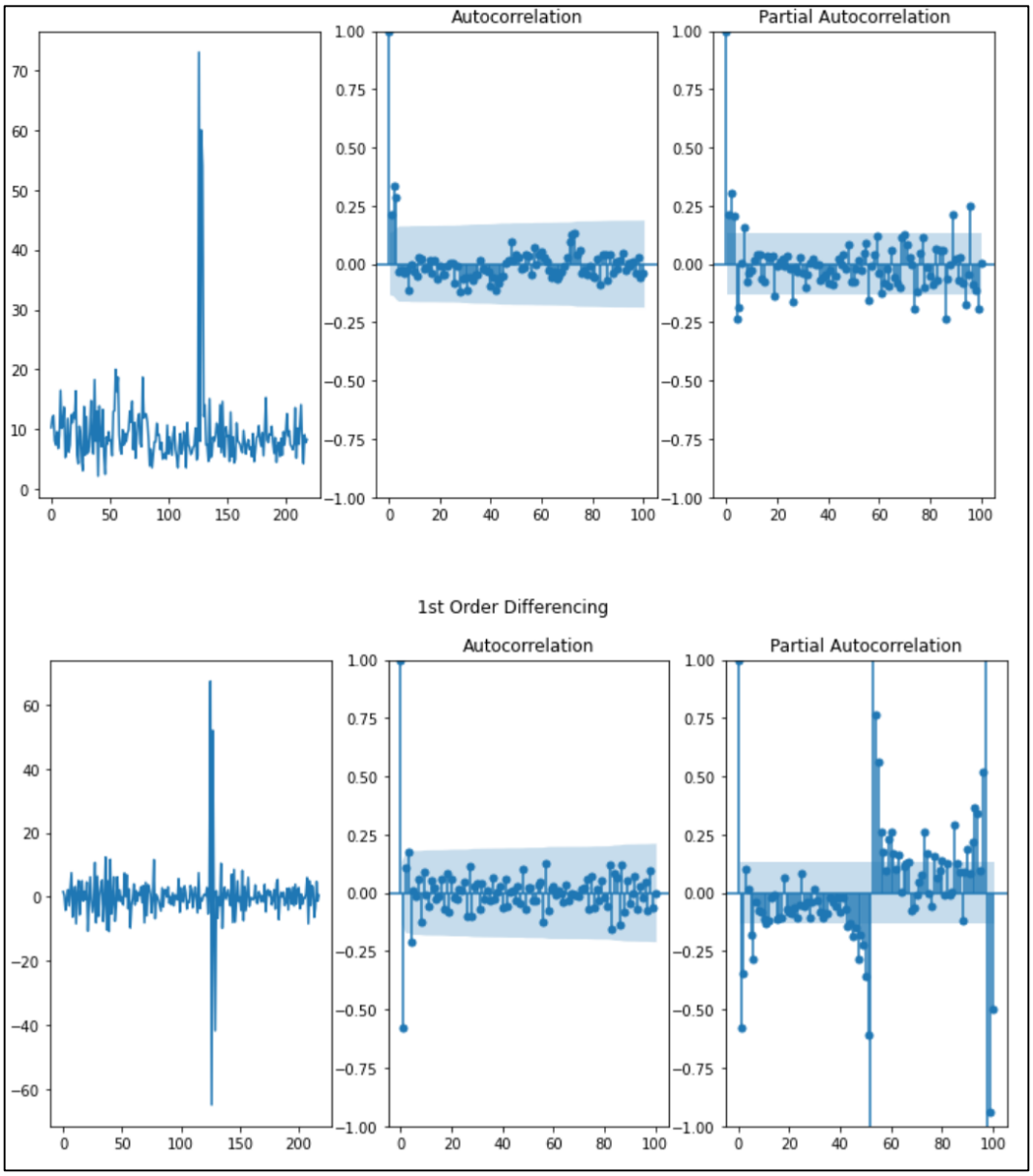


Рисунок 3.5 – Автокореляційні та часткові автокореляційні функції для ряду даних O2

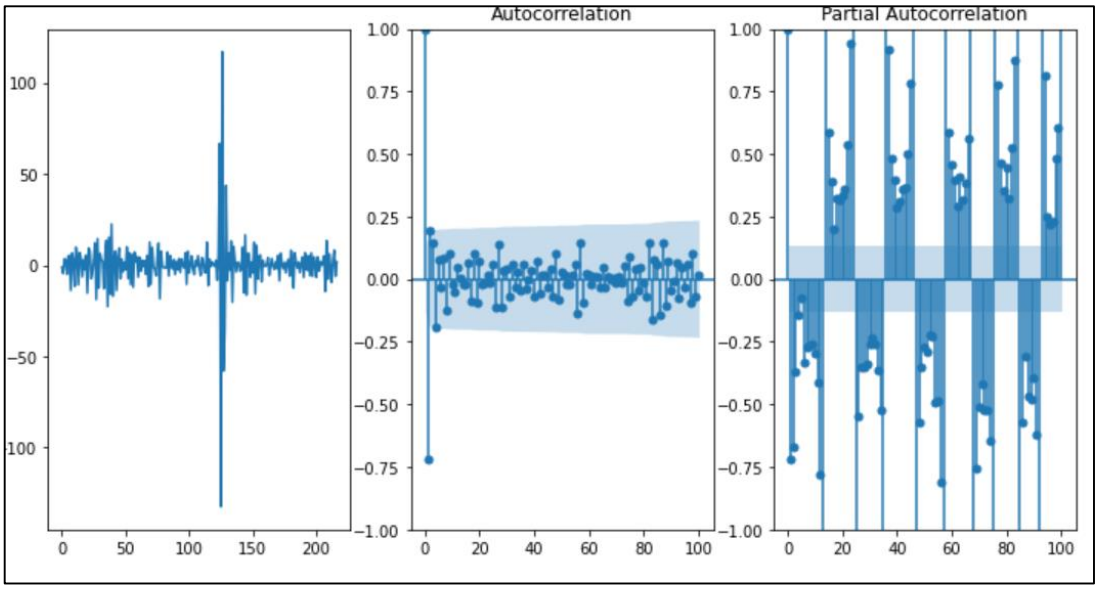


Рисунок 3.6 – Автокореляційні та часткові автокореляційні функції для ряду даних O2 (продовження)

Знайдемо ще інші параметри - авторегресію і ковзне середнє. Для цього обчислюємо часткову автокореляційну функцію (PACF) і дивимося, при якому зміщенні («lag») її значення потрапляє в область значущості p-value = 0,05, враховуючи як її верхню, так і нижню межі – значення дорівнює одним із таких порядку *p* зсуву моделі авторегресії (рис. 3.7).

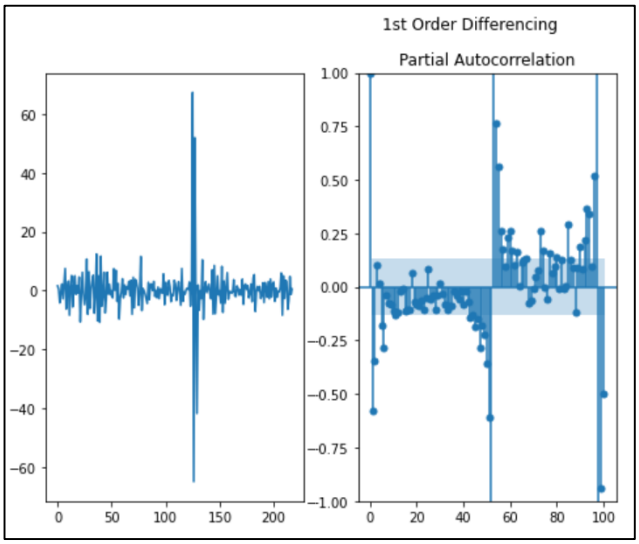


Рисунок 3.7 – Часткова автокореляційна функція першої різниці

Як можна побачити з рисунка 3.6, $p = 5$.

Обчислимо автокореляційну функцію (ACF), щоб побачити, для якого зсуву («lag») її значення зменшується до значущої області p-value = 0,05, враховуючи її верхню та нижню межі – значення цього зсуву дорівнює А рухається середня модель порядку величини q (рис. 3.8).

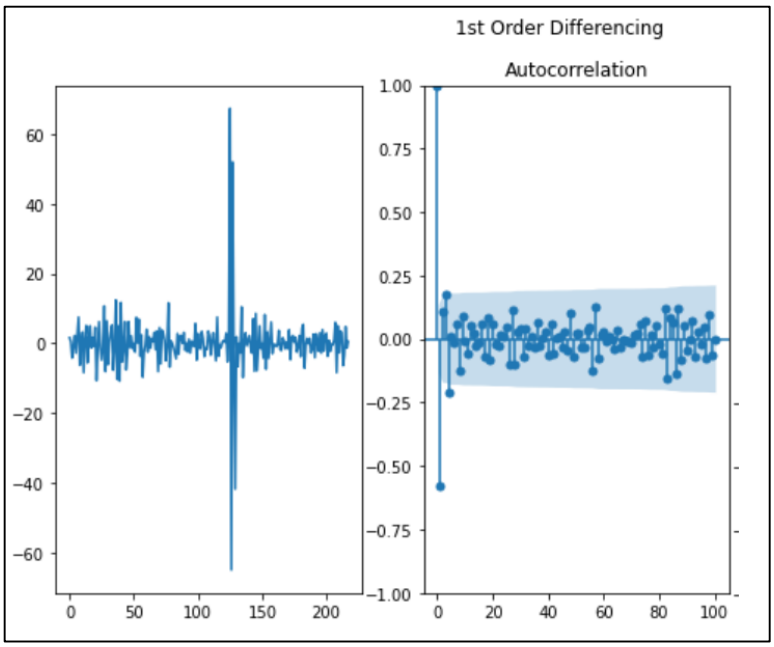


Рисунок 3.8 – Автокореляційна функція першої різниці

По результату видно з рисунку 2.12, що $q = 0$. Отже, аналіз показує, що оптимальною моделью має бути:

$$\text{SARIMAX}(p, q, d) = \text{SARIMAX}(5, 0, 0).$$

На рисунках 3.9, 3.10 результат моделей валідаційних даних .

SARIMAX Results

Dep. Variable:	y	No. Observations:	219
Model:	SARIMAX(5, 0, 0)	Log Likelihood	-709.192
Date:	Wed, 05 Oct 2022	AIC	1432.383
Time:	14:08:05	BIC	1456.107
Sample:	0	HQIC	1441.965
	- 219		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	6.0059	1.155	5.200	0.000	3.742	8.270
ar.L1	0.0912	0.073	1.244	0.214	-0.053	0.235
ar.L2	0.3734	0.023	16.121	0.000	0.328	0.419
ar.L3	0.2809	0.060	4.695	0.000	0.164	0.398
ar.L4	-0.2024	0.056	-3.590	0.000	-0.313	-0.092
ar.L5	-0.1791	0.055	-3.286	0.001	-0.286	-0.072
sigma2	37.9171	1.243	30.504	0.000	35.481	40.353

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	28841.59
Prob(Q):	0.99	Prob(JB):	0.00
Heteroskedasticity (H):	0.51	Skew:	5.78
Prob(H) (two-sided):	0.00	Kurtosis:	58.02

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Рисунок 3.9 – Результат прогнозування за моделлю ARIMA

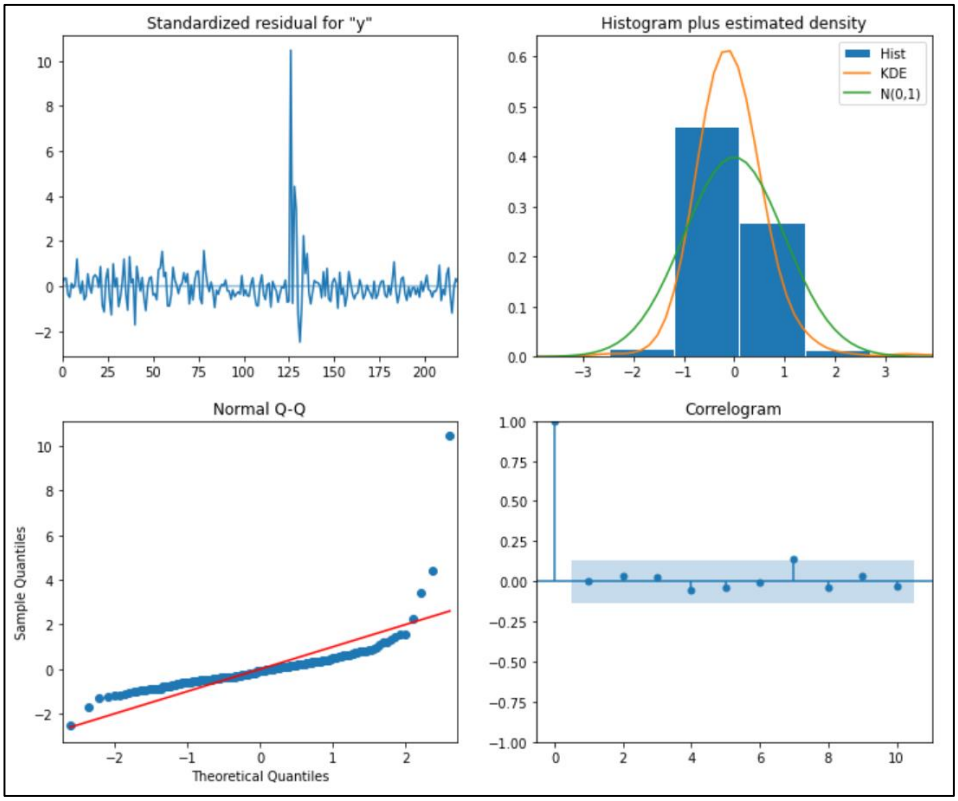


Рисунок 3.10 – Результат похибки прогнозу валідаційних даних моделлі ARIMA

Як бачимо, результати виглядають прийнятними похибки схожі на нормальний закон, розподіл близький до лінійного і похибки більш-менш відповідають вимогам, апарат ARIMA не дозволяє побудувати достатню модель і потрібно використовувати інші методи.

3.4 Прогнозування за допомогою методів машинного навчання

Побудуємо мнржину моделей, використовуючи методи машинного навчання. Використовуємо ознаки, вибрані в розділі 1. Нормуємо їх методом «sklearn.preprocessing.StandardScaler» (рис. 3.11).

	name	model	param_grid
0	Linear Regression	LinearRegression()	{'fit_intercept': [True, False]}
1	KNeighbors Regressor	KNeighborsRegressor()	{'n_neighbors': [3, 5, 10, 20, 30], 'leaf_size': [10, 20, 30]}
2	Support Vector Machines	SVR()	{'kernel': ['linear', 'poly', 'rbf', 'sigmoid'], 'C': [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0], 'tol': [0.001, 0.0001]}
3	Linear SVR	LinearSVR()	{'C': [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0]}
4	Random Forest Regressor	RandomForestRegressor()	{'n_estimators': [40, 50, 60, 80], 'min_samples_split': [30, 40, 50, 60], 'min_samples_leaf': [10, 12, 15, 20, 50], 'max_features': ['auto'], 'max_depth': [3, 4, 5, 6]}
5	Bagging Regressor	BaggingRegressor()	{'max_features': [0.05], 'n_estimators': [3, 4, 5, 6], 'warm_start': [False]}
6	XGB Regressor	XGBRegressor(base_score=None, booster=None, callbacks=None,\n colsample_bylevel=None, colsample_bynode=None,\n colsample_bytree=None, early_stopping_rounds=None,\n ...	{'n_estimators': [50, 70, 90], 'learning_rate': [0.01, 0.05, 0.1, 0.2], 'max_depth': [3, 4, 5]}
7	MLP Regressor	MLPRegressor()	{'hidden_layer_sizes': [2, 3, 4], 'solver': ['lbfgs', 'sgd'], 'learning_rate': ['adaptive'], 'learning_rate_init': [0.001, 0.01], 'max_iter': [1000]}

Рисунок 3.11 – Моделі машинного навчання та їх основні параметри

Лінійнй регресійний (Linear Regresion) аналіз використовується для прогнозування значення змінної на основі значення іншої змінної. Змінна, значення якої бажаєте передбачити, називається залежною змінною. Змінна, яку використовують для прогнозування значення іншої змінної, називається незалежною змінною [18].

Ця форма аналізу оцінює коефіцієнти лінійного рівняння, що включає одну або кілька незалежних змінних, які найкращим чином передбачають значення залежної змінної. Лінійна регресія відповідає прямій лінії або поверхні, яка мінімізує розбіжності між передбаченими і фактичними вихідними значеннями. Існують прості калькулятори лінійної регресії, які

використовують метод "найменших квадратів" для визначення найкращої лінії для набору парних даних. Потім оцінюється значення X (залежна змінна) за Y (незалежна змінна).

Модель k-nearest neighbors regressor (KNN) – це простий, легко реалізований алгоритм контрольованого машинного навчання, який можна використовувати для вирішення завдань класифікації та регресії.

Random forest regressor – це алгоритм контрольованого навчання, який використовує метод ансамблевого навчання для класифікації та регресії. Випадковий ліс – це метод багінгу, а не метод бустингу. Дерева у випадковому лісі працюють паралельно, що означає відсутність взаємодії між ними при побудові дерев.

Лінійний SVR, заснований на методі опорних векторів, являє собою алгоритм для вирішення задач регресії. Алгоритм Linear SVR застосовує метод лінійного ядра і добре працює з великими наборами даних.

Bagging Regressor – це ансамблевий алгоритм машинного навчання, який об'єднує прогнози багатьох дерев рішень. Він також простий у реалізації, оскільки має кілька ключових гіперпараметрів і розумну евристику для налаштування цих гіперпараметрів [17]. Bagging добре працює в цілому і служить основою для цілої області алгоритмів ансамблю дерев рішень, таких як популярні алгоритми випадкового лісу, а також менш відомі алгоритми ансамблю Pasting, Random Subspaces і Random Patches.

Під час використання методу GridSearchCV були обрані параметри кожної моделі рисунок 3.12

```
Tuning model 'Linear Regression'
Best parameters: {'fit_intercept': True}

Tuning model 'KNeighbors Regressor'
Best parameters: {'leaf_size': 10, 'n_neighbors': 30}

Tuning model 'Support Vector Machines'
Best parameters: {'C': 1.0, 'kernel': 'rbf', 'tol': 0.001}

Tuning model 'Linear SVR'
Best parameters: {'C': 15.0}

Tuning model 'Random Forest Regressor'
Best parameters: {'max_depth': 5, 'max_features': 'auto', 'min_samples_leaf': 50, 'min_samples_split': 50, 'n_estimator
s': 40}

Tuning model 'Bagging Regressor'
Best parameters: {'max_features': 0.05, 'n_estimators': 4, 'warm_start': False}

Tuning model 'XGB Regressor'
Best parameters: {'learning_rate': 0.05, 'max_depth': 3, 'n_estimators': 50}

Tuning model 'MLP Regressor'
Best parameters: {'hidden_layer_sizes': 2, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'max_iter': 1000, 's
olver': 'sgd'}
```

Рисунок 3.12 – Оптимальні параметри ідентифікованих моделей, як побудовані для багатьох ознак та для валідаційних даних

3.5 Вибір оптимальної моделі

Після обчислення всіх моделей на валідаційні дані перевірки відбувається значення прогнозу і порівняння передбачення з відомим цільовим значенням у наборі даних перевірки за 3 показниками: r2_score,rmse, mape. На рис. 3.13 зображено результат оптимальної моделі у випадку прогнозування.

	name_model	type_data	r2_score	rmse	mape
0	Prophet_4_days_3_order	valid	-0.12888	1.507844	17.717992
5	Prophet_7_days_12_order	valid	-1.481851	2.235735	21.158751
6	Prophet_14_days_3_order	valid	-0.731924	1.867656	21.473867
1	Prophet_4_days_12_order	valid	-1.045838	2.029869	21.510995
4	Prophet_7_days_3_order	valid	-1.842827	2.392807	23.109124
2	Prophet_5_days_3_order	valid	-1.265159	2.135904	23.212625
3	Prophet_5_days_12_order	valid	-1.492111	2.240351	25.092032
7	Prophet_14_days_12_order	valid	-3.407019	2.979236	27.075351
8	ARIMA_auto	valid	-1.426119	2.21049	27.394824
12	Linear SVR	valid	-1247.676434	50.148435	604.156941
11	Support Vector Machines	valid	-1307.444256	51.334583	612.168311
15	XGB Regressor	valid	-1318.414275	51.549329	619.184893
14	Bagging Regressor	valid	-1374.32323	52.630175	633.168605
10	KNeighbors Regressor	valid	-1774.788039	59.803691	714.319461
16	MLP Regressor	valid	-1732.419541	59.085957	717.277372
13	Random Forest Regressor	valid	-1827.017491	60.676791	718.38066
9	Linear Regression	valid	-1778.548918	59.866986	723.893312

Рисунок 3.13 –Моделі машинного навчання

З графіку 3.13 можна визначити найкращу похибку моделей серед всіх трьох метрик. Оптимальна модель задовільна лише за метрикою MAPE. Можна зробити висновок, що під час прогнозування оптимальної моделі на «Prophet_4_days_3_order» – оптимальна модель, у якої похибки за метриками MAPE має доволі не погану точність. Оптимальна модель RMSE має задовільний результат і відносну похибку.

3.6 Висновки

У даному розділі роботи проаналізовано та спрогнозовано вміст розчиненого у воді кисню за допомогою методів ARIMA та Facebook Prophet, а також методами машинного навчання. Результати дослідження показали, що оптимальна модель прогнозування розчиненого у воді кисню є «Facebook Prophet», в якої похибка за метрикою RMSE дорівнює 1.5 мг/л, а за метрикою MAPE - 17.71%.

4 ЗАСТОСУВАННЯ РОЗРОБЛЕНОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ ЗМІН КОНЦЕНТРАЦІЇ РОЗЧИНЕНОГО У ВОДІ КИСНЮ У РІЧЦІ ПІВДЕННИЙ БУГ

4.1 Прогнозування тестових даних на практиці

Модель яка була обрана для кожного показника, була повторно ідентифікована на розширеному наборі даних «train+valid», після чого були перевірені дані, що прогнозуються та порівнюються з реальними даними (рис. 4.1-4.4).

Optimal model by metrics "rmse" is "Prophet_4_days_3_order" with type "Prophet" parameters [4, 3]

	name_model	r2_score	rmse	mape	params
0	Prophet_4_days_3_order	-0.12888	1.507844	17.717992	[4, 3]

Рисунок 4.1 – Похибки і оптимальні параметри моделі Facebook Prophet, побудованої для тестових даних

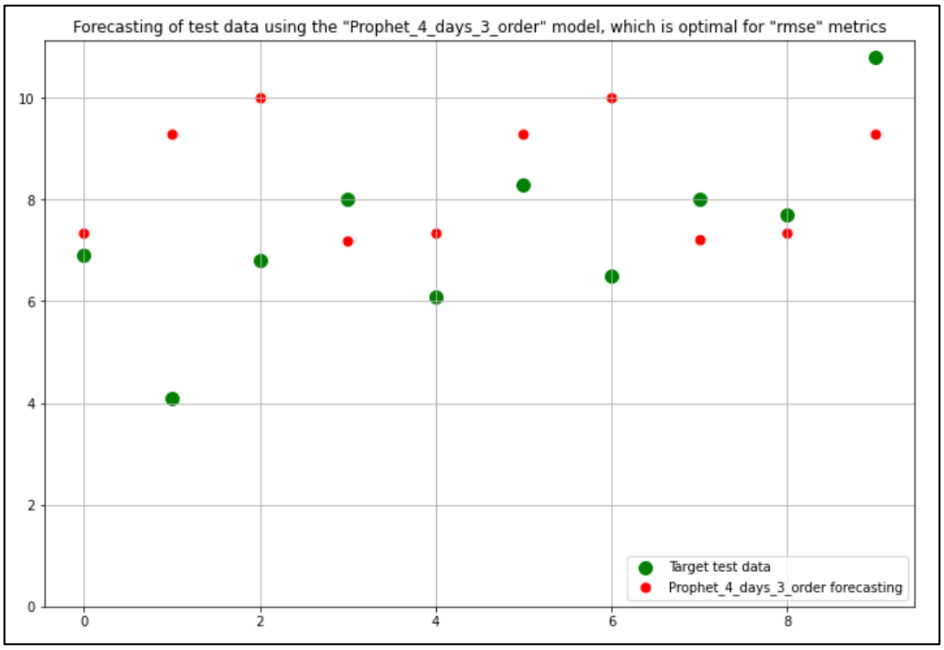


Рисунок 4.2 – Реальні тестові дані розчиненого у воді кисню та їх прогноз за моделлю Facebook Prophet, за метрикою RMSE

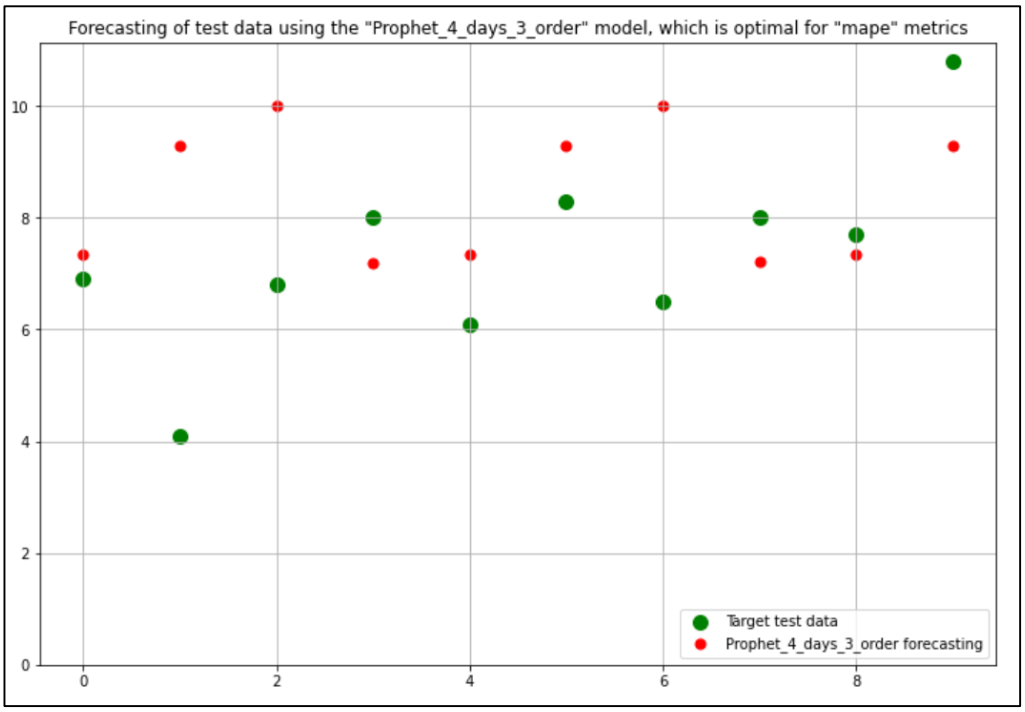


Рисунок 4.3 – Реальні тестові дані розчиненого у воді кисню та їх прогноз за моделлю Facebook Prophet, за метрикою MAPE

4.2 Аналіз важливості ознак

Зазвичай, після вирішення завдання машинного навчання з багатьма ознаками та функціями, аналіз важливості цих ознак приймається в першому наближенні. Тоді, слід повертатися до етапу «Feature engineering» у синтезі та вилученні функцій і вносити корективи, наприклад: вилучати малоцінні та непотрібні ознаки та робити більш цінні та нові ознаки для прогнозування. Після отримання задовільних результатів (принаймні за деякими показниками) необхідно описати та аналізувати данні. Також потрібно вказувати значення ознаки в остаточному рішенні, описаному в цьому підрозділі 4.1.

Також можна оцінити важливість та цінність ознак обраної найкращої моделі кількома способами.

Бібліотека SHAP – це інструмент візуалізації, який можна використовувати для того, щоб зробити модель машинного навчання більш зрозумілою, візуалізуючи її результати. Він може використовуватися для

пояснення прогнозів будь-якої моделі шляхом обчислення внеску кожної ознаки в передбачення [17].

Використовуючи SHAP, можем зробити результати більш зрозумілими для користувачів, які не дуже добре розбираються в результатах моделей машинного навчання. Завдяки цій здатності SHAP також можем використовувати його для візуалізації даних.

Sklearn – це бібліотека, яку використовують багато фахівців з машинного навчання по всьому світу для швидкого побудови циклів машинного навчання. Вона надає дуже простий у використанні інтерфейс і має реалізацію багатьох алгоритмів машинного навчання [17]. Незважаючи на те, що sklearn надає простий у використанні інтерфейс і реалізацію всіх алгоритмів, не можем дізнатися точну продуктивність нашої моделі тільки на основі таких показників, як accuracy, r^2, roc curve, precision-recall curve і т.д. Нам потрібно щось більш надійне, що може дати нам розуміння того, чому модель працює певним чином для конкретного прикладу.

Потрібен спосіб зрозуміти, який внесок вносять різні ознаки в передбачення, щоб краще зрозуміти, чи добре узагальнена наша модель і чи надійна вона. У python є бібліотека eli5, яка може допомогти нам краще зрозуміти навчені моделі, чому вони роблять конкретний прогноз на конкретному прикладі [17]. Це допоможе краще зрозуміти нашу модель, щоб дізнатися про важливі особливості, а також про надійність моделі.

Рисунки 4.4-4.7 показують графіки важливості ознак («Feature Importance») для кращих моделей цього підрозділу, побудовані різними способами. Діаграма ознак для найкращих моделей машинного навчання, створених за допомогою бібліотеки SHAP для моделювання розчиненого у воді кисню O2 на рисунку 3.9.

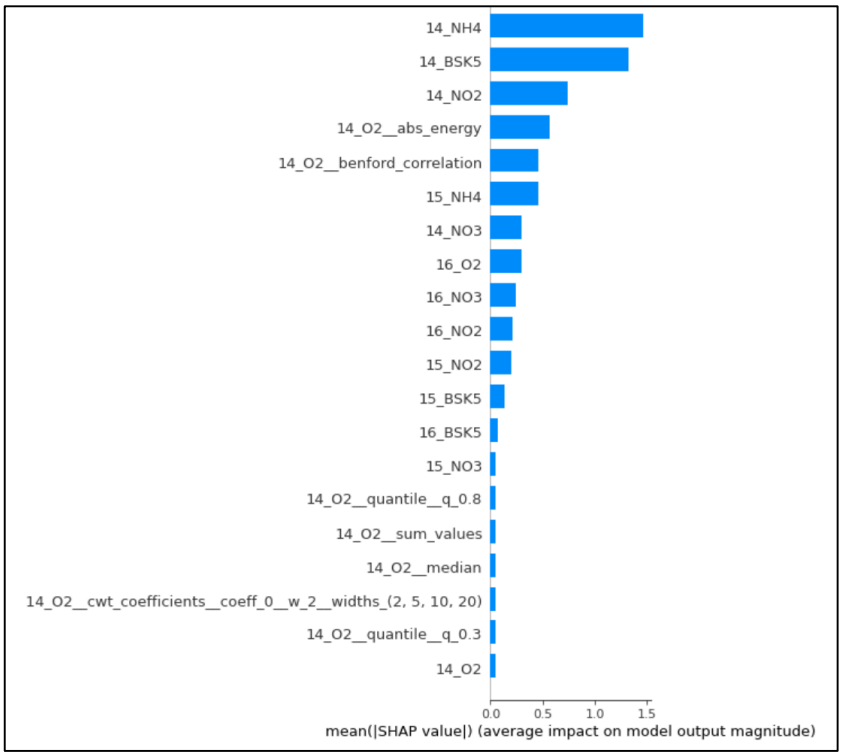


Рисунок 4.4 – Діаграма ознак для найкращих моделей машинного навчання

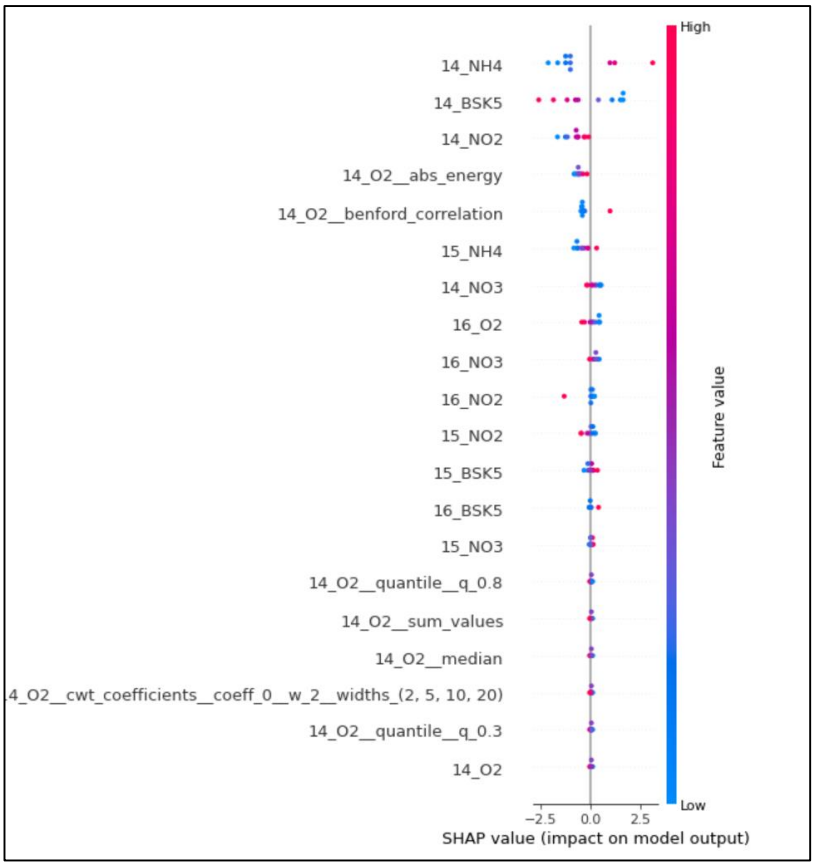


Рисунок 4.5 – Детальна діаграма важливості ознак, створена за допомогою бібліотеки SHAP для моделювання розчиненого у воді кисню

Діаграма «Force plot», яка враховує важливість ознак і створена за допомогою бібліотеки SHAP для моделювання курсу розчиненого у воді кисню зображена на рисунку 4.6.

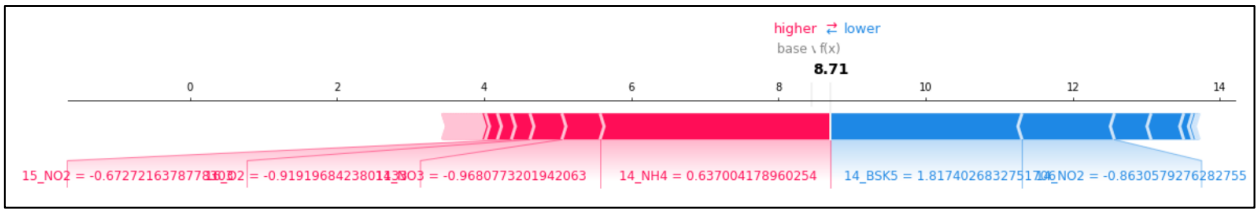


Рисунок 4.6 – Діаграма «Force plot»

Діаграма перестановки з врахуванням важливості ознак і побудована за допомогою модуля inspection та бібліотеки sklearn зображена на рисунку 4.7.

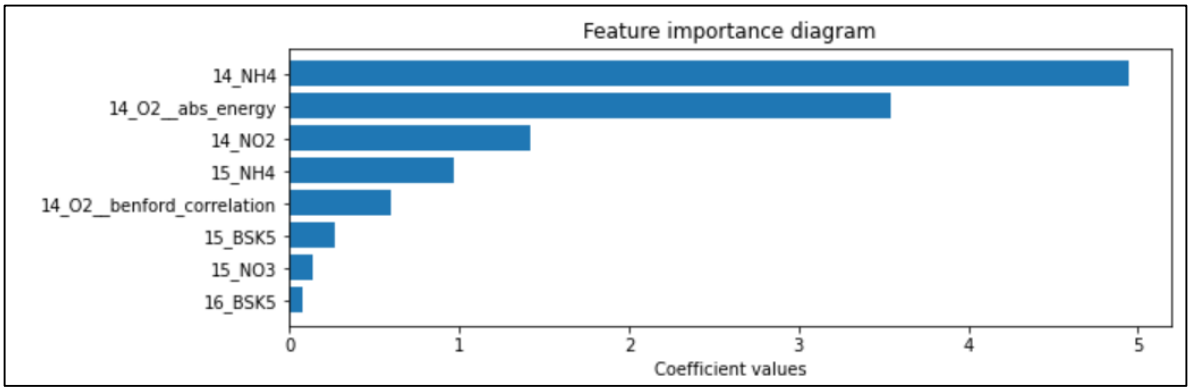
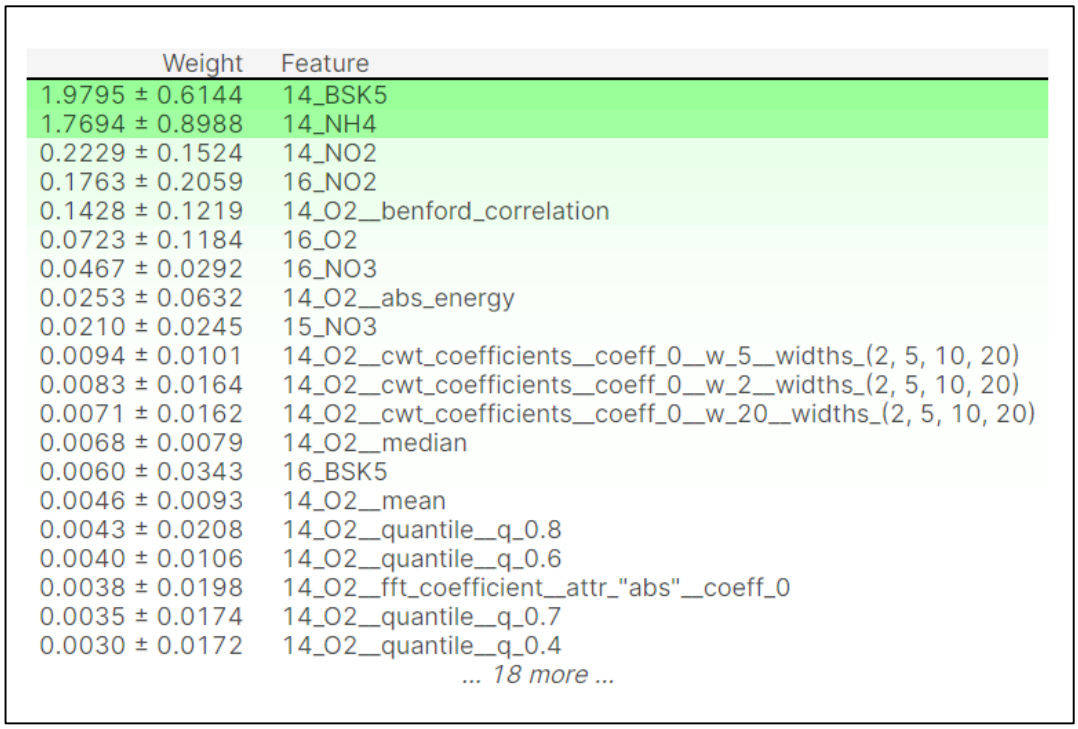


Рисунок 4.7 – Діаграма перестановки з важливістю ознак

Діаграма перестановки із врахуванням важливості ознак, побудовані за допомогою бібліотеки ELI5 зображено на рисунку 4.8.



Weight	Feature
1.9795 ± 0.6144	14_BSK5
1.7694 ± 0.8988	14_NH4
0.2229 ± 0.1524	14_NO2
0.1763 ± 0.2059	16_NO2
0.1428 ± 0.1219	14_O2__benford_correlation
0.0723 ± 0.1184	16_O2
0.0467 ± 0.0292	16_NO3
0.0253 ± 0.0632	14_O2__abs_energy
0.0210 ± 0.0245	15_NO3
0.0094 ± 0.0101	14_O2__cwt_coefficients__coeff_0__w_5__widths_(2, 5, 10, 20)
0.0083 ± 0.0164	14_O2__cwt_coefficients__coeff_0__w_2__widths_(2, 5, 10, 20)
0.0071 ± 0.0162	14_O2__cwt_coefficients__coeff_0__w_20__widths_(2, 5, 10, 20)
0.0068 ± 0.0079	14_O2__median
0.0060 ± 0.0343	16_BSK5
0.0046 ± 0.0093	14_O2__mean
0.0043 ± 0.0208	14_O2__quantile__q_0.8
0.0040 ± 0.0106	14_O2__quantile__q_0.6
0.0038 ± 0.0198	14_O2__fft_coefficient__attr_"abs"__coeff_0
0.0035 ± 0.0174	14_O2__quantile__q_0.7
0.0030 ± 0.0172	14_O2__quantile__q_0.4
	... 18 more ...

Рисунок 4.8 – Діаграма перестановки з важливістю ознак

Якщо підводити підсумки можна побачити по діаграмам, що ознаками з найбільшою важливістю є BSK5 та NH4 на посту №14, а найменш цінними – O2 і NO3 на посту №16.

4.3 Висновки

У даному розділі розроблено інформаційну технологію аналізу та прогнозування розчиненого у воді кисню в річці Південний Буг. На основі тестових даних здійснено прогнозування за допомогою моделлю Facebook Prophet та проведено аналіз важливості ознак, у результаті якого виявлено, що найважливішими ознаками є BSK5 та NH4 на посту №14, а найменш цінними – O2 і NO3 на посту №16.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нової інформаційної технології прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг. Метою дослідження є підвищення точності прогнозування вмісту розчиненого у воді кисню, за рахунок використанням методів машинного навчання. Подібні технології вже існують, але – для англомовних текстів та інших задач. Зокрема, не має адаптацій до задачі прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг. Особливістю технології є високий показник точності автоматичної класифікації. За аналоги були взяті датасети платформи Kaggle, які аналізують та прогнозують дані вмісту речовин. Орієнтовна вартість такої системи, для комерційного сегменту, складає близько 20000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п’ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.[21]

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	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 роботоздат-ність продук-ту в реаль-них умовах

Продовження табл. 5.1

Ринкові переваги					
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
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
Ринкові перспективи					
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
Практик на 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 із 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 використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 5.1

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

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Козачко О.М.	Ящолт А.Р.	Варчук І.В.
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	2
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	4	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	2	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	2	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	4	3	4
Сума	43	40	42
Середньоарифметична сума балів	(43+40+42) / 3 = 41,67		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, 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

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок підвищення точності прогнозування вмісту розчиненого у воді кисню, за рахунок використанням методів машинного навчання. Подібні технології вже існують, але для англомовних текстів та інших задач. Зокрема, не має адаптацій до задачі прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг. Особливістю технології є високий показник точності автоматичної класифікації.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$З_о = \frac{M}{T_р} \cdot t, \tag{5.1}$$

де М – місячний посадовий оклад конкретного розробника (дослідника), грн.;

$T_р$ – число робочих днів за місяць, 21 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.5.

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	31000	1476,19	29	42809,524
Програміст	29000	1380,95	29	40047,619
Всього				82857,14

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

5.2.2 Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 10 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 10\ \% / 100\ \% \qquad (5.2)$$

$З_д = (82857,14 \cdot 10\ \% / 100\ \%) = 8285,71$ (грн.)

5.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22\ \% / 100\% \qquad (5.3)$$

$Н_з = (82857,14 + 8285,71) \cdot 22\ \% / 100\ \% = 20051,43$ (грн.)

5.2.5. Оскільки для розроблювального пристрою не потрібно витрачати матеріали та комплектуючи, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{Тв} \cdot \frac{t_{вик}}{12} \text{ [грн.].} \quad (5.4)$$

де Ц – балансова вартість обладнання, грн.;

Т – термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 20000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,38 міс.

$$A_{обл} = \frac{20000}{2} \times \frac{1{,}38}{12} = 1150{,}79 \ грн.$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 5.5

Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних активів є безкоштовною, то $В_{нем.ак.} = 0$ грн.

Таблиця 5.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортиза-ційні відрахуван-ня, грн.
Комп'ютер та комп'ютерна периферія (Apple MacBook Pro 13 Early 2015)	20000	2	1,38	1150,794
Офісне обладнання (меблі)	20000	4	1,38	575,397
Приміщення	650000	20	1,38	3740,079
Всього				5466,27

5.2.6 Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п, \qquad (5.5)$$

де В – вартість 1 кВт-години електроенергії для 1 класу підприємства, В = 6,2 грн./кВт;

П – встановлена потужність обладнання, кВт. П = 0,3 кВт;

Ф – фактична кількість годин роботи обладнання, годин.

$К_п$ – коефіцієнт використання потужності, $К_п$ = 0,9.

$В_е = 0,9 \cdot 0,3 \cdot 8 \cdot 29 \cdot 6,2 = 388,368$ (грн.)

5.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені

безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (5.6)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

$I_в$ = 82857,14 * 50% / 100% = 41428,57 (грн.)

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100…150% від суми основної заробітної плати дослідників:

$$Н_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (5.7)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$Н_{нзв}$ = 82857,14 * 100 % / 100 % = 82857 (грн.)

5.2.9 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$В_{заг}$ = 82857,14+8285,71+20051,43+5466,27+388,37+41428,57+
+82857 = 241334,64 грн.

5.2.11 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (5.8)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η=0,2; розробки конструкторської документації, то η=0,3; розробки технологій, то η=0,4; розробки дослідного зразка, то η=0,5; розробки промислового зразка, то η=0,7; впровадження, то η=0,9. Оберемо η = 0,5, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

ЗВ = 241334,64 / 0,5 = 482669 грн.

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого

прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб’єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов’язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об’єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta П_i = (\pm \Delta Ц_0 \cdot N + Ц_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (5.9)$$

де $\pm\Delta Ц_о$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$Ц_о$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $Ц_о = Ц_б \pm \Delta Ц_о$;

Цб – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт λ = 0,8333.

р – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році ϑ =18%.

Припустимо, що при прогнозованій ціні 9800 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 1000 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 800 шт., протягом другого року – на 500 шт., протягом третього року на 300 шт.

До моменту впровадження результатів наукової розробки реалізації продукту не було:

$\Delta П_1$ = (0*1000 + (9800 + 1000)*800)* 0,8333* 0,38) * (1 - 0,18) = 2035786,585 грн.

$\Delta П_2$ = (0*1000 + (9800 + 1000)*(800+500)* 0,8333* 0,38) * (1 - 0,18) = 3645719,854 грн.

$\Delta П_3$ = (0*1000 + (9800 + 1000)*(800+500+300)* 0,8333* 0,38) * (1 - 0,18) = 4487039,821 грн.

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 10168546,26 грн.

5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (5.10)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

Т – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ = 0,05…0,15;

t – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$ПП = (2035786,585/(1+0,1)^1)+(3645719,854/(1+0,1)^2)+(4487039,821/$$
$$/(1+0,1)^3) = 1850715,08 + 3012991,615 + 3371179,429 = 8234886,122 \text{ грн.}$$

Далі розраховують величину початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{інв} * ЗВ, \quad (5.11)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{інв}$=2...5, але може бути і більшим;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 482669 = 965338,55 \text{ грн.}$$

Тоді абсолютний економічний ефект $Е_{абс}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$Е_{абс} = ПП – PV, \quad (5.12)$$

$$Е_{абс} = 8234886,122 – 965338,55 = 7269547,57 \text{ грн.}$$

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої но-рми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього використаємо формулу:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{5.13}$$

$T_ж$ – життєвий цикл наукової розробки, роки.

$$E_в = \sqrt[3]{(1 + 7269547{,}57/965338{,}55} - 1 = 1{,}043$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \tag{5.14}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d = (0,09…0,14);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,5).

$$\tau_{\min} = 0{,}14 + 0{,}05 = 0{,}19.$$

Так як $E_в > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_в}, \tag{5.15}$$

$T_{ок}$ = 1 / 1,043 = 0,96 р.

Оскільки $T_{ок}$ < 3-х років, а саме термін окупності рівний 0,96 роки, то фінансування даної наукової розробки є доцільним.

5.4 Висновки

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 482669 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,96 роки.

ВИСНОВКИ

В процесі виконання роботи досліджено та доведено актуальність теми прогнозування розчиненого у воді кисню O2.

Розглянуто основні характеристики об’єкту дослідження, здійснено аналіз функціонування кисневого режиму і деструкції органічних речовин та проаналізовано основні екологічні проблеми в даному напрямку. Наведено опис методу прогнозування та вибрано оптимальні інформаційні технології для подальшого дослідження, а саме платформу Python і моделі прогнозування такі як ARIMA, Facebook Prophet а також методи машинного навчання.

Здійснено розвідувальний аналіз початкових даних задачі прогнозування, що вирішується в роботі на основі методів статистичного та кореляційного аналізу за допомогою бібліотек Autoviz і Sweetviz. Результати цих аналізів використані для побудови моделей машинного навчання для прогнозування O2 вмісту в річці Південний Буг.

Проаналізовано та спрогнозовано вміст розчиненого у воді кисню за допомогою використання ARIMA та Facebook Prophet, також спрогнозовано за допомогою ряду методів машинного навчання. Результати дослідження показали, що оптимальна модель прогнозування розчиненого у воді кисню є «Facebook Prophet», в якої похибка за метрикою RMSE виявилась 1.5 мг/л. Також результат має малу помилку MAPE 17.71%.

Розроблено інформаційну технологію аналізу та прогнозування розчиненого у воді кисню O2 в річці Південний Буг. На основі тестових даних здійснено прогнозування за допомогою моделлю Facebook Prophet та проведено аналіз важливості ознак, у результаті якого виявлено, що найважливішими ознаками є BSK5 та NH4 на посту №14, а найменш цінними – O2 і NO3 на посту №16.

Отже всі поставлені завдання та мета роботи були досягнуті і виконані у повному обсязі, підвищена точності прогнозування вмісту розчиненого у воді кисню, за рахунок використання методів машинного навчання.

За результатами даної роботи була зроблена доповідь на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблема, перспективи» (Вінниця, 2022-2023 рр.), на тему «Інформаційна технологія прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг». Опубліковано тези на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблема, перспективи».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Козачко О.М., Головач В.Р., Інформаційна технологія прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг. *Всеукраїнська науково-практична інтернет конференція*: «Молодь в науці: дослідження, проблеми, перспективи», м. Вінниця, 2022-2023 рр., URL: https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/view/16827

2. В. Б. Мокін, Є. М. Крижановський . Геоінформаційні системи в екології: Електронний навчальний посібник під ред. Крижановського Є. М.Вінниця : ВНТУ, 2014. 192 с

3. Мокін В. Б. Новий підхід щодо розробки автоматизованої геоінформаційної системи для комплексного екологічного контролю стану забруднення довкілля / В. Б. Мокін, М. П. Боцула, Є. М. Крижановський, А. Р. Ящолт, Вісник Національного технічного університету України «Київський політехнічний інститут», Серія «Хімічна інженерія, екологія та ресурсозбереження». Київ : НТУУ «КПІ». 2013. № 1 (11). С. 97-102.

4. Мокін В. Б., Ящолт А. Р. Новий підхід до розробки інформаційних моделей автоматизованих систем екоінспекційного контролю. *X Міжнародна науково-технічна конференція «Системний аналіз та інформаційні технології»*: збірник матеріалів, Киів, 2008. С. 385.

5. Мокін В. Б. Дзюняк Д. Ю., Бондалєтов К. О., Горячев Г. В., Боцян В. В. Практична реалізація мобільної аналітичної комп'ютерної системи моніторингу стану атмосферного повітря з підсистемою веб-аналізу та виведення даних на геопортал. *V-й Всеукраїнський з'їзд екологів з міжнародною участю (Екологія/Ecology-2015)*: Збірник наукових праць, Вінниця: ТОВ «Нілан-ЛТД», 2015. С. 106.

6. Технологія оцінювання параметрів викидів зі стаціонарних джерел за даними оперативного моніторингу / В. Мокін, Д. Дзюняк : *Матеріали III Міжнародної науково-практичної конференції "Summer*

InfoCom Advanced Solutions 2016", м. Київ, 1-2 грудня 2016 р., Київ : ТОВ «Інженеринг», 2016. №3 С. 66-68.

7. Blau, D. M. A Time Series Analysis of Self Employment in the United States: *Journal of Political Economy*, 2015, 445-467.

8. Бібліотека Pandas URL: https://uk.wikipedia.org/wiki/Pandas

9. DataSciense URL: https://coursehunter.net/course/data-science-uroven1-instrumenty-i-tehnologii

10. 20 популярних моделей URL: https://www.kaggle.com/vbmokin/heart-disease-comparison-of-20-models

11. Метод наївний баєсів класифікатор URL: https://uk.wikipedia.org/wiki/Naivnyy_Baesiv_klasyficator

12. Firebase Realtime Database URL: https://firebase.google.com/docs/database/.

13. Modeling and Forecasting Electricity Loads and Prices: *A Statistical Approach / Rafał Weron*. Wrocław University of Technology, 2006, John Wiley & Sons, Inc.. С. 81-83.

14. Кофанов В.І., Огняник М.С. Нормативно–методичне забезпечення визначення якості води при оцінці впливу на навколишнє середовище. *Наукові праці Інституту геологічних наук НАН України*. Вінниця, 2008. URL: http://dspace.nbuv.gov.ua/bitstream/handle/123456789/5595/02-Kofanov.pdf?sequence=1

15. Garlapati, Anusha, et al. "Stock Price Prediction Using Facebook Prophet and Arima Models."*2021 6th International Conference for Convergence in Technology (I2CT)*". IEEE, 2021 С. 17.

16. Метод оцінювання параметрів стаціонарного джерела викиду на основі моделі Гауса за даними оперативного моніторингу зони розсіювання / В. Б. Мокін, Д. Ю. Дзюняк. Вінниця : Математичне моделювання в економіці, 2016. 27-35 с.

17. Kadiyala, Akhil, and Ashok Kumar. Applications of python to evaluate the performance of bagging methods/ *International Scientific and Technical Journal "Information Technologies and Computer Engineering"*. 2008. №. 1(11). С. 37–43.

18. Feng D. C. et al. Interpretable XGBoost-SHAP machine-learning model for shear strength prediction of squat RC walls : *Journal of Structural Engineering*. 2021. №. 11. С. 73.

19. Datastart. Scikit Learn URL: https://datastart.ru/blog/read/chto-takoe-scikit-learn-gayd-po-populyarnoy-bibliotekepython-dlya-nachinayuschih

20. ELI5 URL: https://eli5.readthedocs.io/en/latest/index.html

21. Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт : уклад. Вінниця : ВНТУ, 2021. 42 с.

Додаток А

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«_19_» _____09_____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
«Інформаційна технологія прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг»

Керівник: к.т.н., доц. каф. САІТ
_____ Козачко О. М.
«_19_» _____09_____ 2022 р.

Розробив: студент гр. 2ІСТ-21м
_____ Головач В. Р .
«_19_» _____09_____ 2022 р.

Вінниця 2022

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № 203 по ВНТУ від «14» _09_ 2022 р., та індивідуальне завдання на МКР, затверджене протоколом № 3 засідання кафедри САІТ від «14» ____09___ 2022 р.

2. Джерела розробки:

– Датасети на платформі Kaggle - WQ SB river : EDA and Forecasting URL: https://www.kaggle.com/nikaapril/wq-sb-river-eda-and-forecasting-new

– Дані про моніторинг якості річкової води у річці Південний Буг URL: http://monitoring.davr.gov.ua/EcoWaterMon/GDKMap/Index

3. Мета і призначення роботи:

Розробити інформаційну технологію для аналізу та прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг. яка б давала можливість робити прогнози змін концентрації розчиненого у воді кисню враховуючи дані вимірювань із створів, що розташовані вище по течії.

4. Вихідні дані для проведення робіт:

– Дані датасету «River Water Quality EDA and Forecasting» платформи Kaggle;

– Електронна карта Вінницької області.

5. Методи дослідження:

Машинне навчання, лінійна регресія, Random Forest Regressor, LightGBM, Facebook Prophet, ARIMA, AutoViz, SweetViz, регресивний аналіз, метод лінійної регресії.

6. Етапи роботи і терміни їх виконання:

1. Аналіз предметної області…….....20.09 – 30.09
2. Розробка інформаційної технології……..……...01.10 – 20.10
3. Реалізація інформаційної технології…….……..21.10 – 05.11
4. Економічна частина……..…….……..……..06.11 – 19.11
5. Оформлення пояснювальної записки……..…….20.11 – 30.11

7. Очікувані результати та порядок реалізації:

Визначити найкраще місце розташування для встановлення нових постів моніторингу розчиненого у воді кисню у місті Вінниця.

8. Вимоги до розробленої документації «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……«20» __12_ 2022 р.
Початок розробки…….....« 10 » __09__ 2022 р.
Граничні терміни виконання МКР…….…….....« 30 » __12__ 2022 р.

Розробив студент групи 2ІСТ-21м _____ Головач В.Р.

Додаток Б

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія прогнозування змін концентрації розчиненого у воді кисню у річці Південний Буг»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ
Науковий керівник: Козачко О. М. к.т.н., доц. каф. САІТ

Показники звіту подібності Unicheck

Оригінальність	90,55 %
Схожість	9,45 %

Аналіз звіту подібності (відмітити потрібне):

- ✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Опис прийнятого рішення:

Робота допускається до захисту

Особа, відповідальна за перевірку Жуков С. О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи Головач В.Р.

Керівник роботи Козачко О. М.

Додаток В

Лістинг програмни

```
import random
import os
import numpy as np
import pandas as pd
import datetime as dt
from datetime import date, timedelta, datetime
import matplotlib.pyplot as plt
from matplotlib.pylab import rcParams
import plotly.express as px
import plotly.graph_objects as go
from plotly.offline import init_notebook_mode
init_notebook_mode(connected=True)
import pandas_profiling as pp
import statsmodels.api as sm
from statsmodels.tsa.stattools import adfuller
import sweetviz as sv
from IPython.display import IFrame
import warnings
warnings.filterwarnings("ignore")
is_EDA_with_Pandas_Profiling = True
is_anomalies = True
def fix_all_seeds(seed):
    np.random.seed(seed)
    random.seed(seed)
    os.environ['PYTHONHASHSEED'] = str(seed)
```

```
random_state = 42
fix_all_seeds(random_state)

target_indicator_name = 'O2'
feature_indicator_names = ['NO3', 'NO2', 'BSK5', 'NH4']

id_target_station = 14

id_feature_station = [15, 16]



def get_water_data(target_indicator_name : str,
                   id_target_station : int,
                   date_start : str = "2000-01-02",
                   feature_indicator_names : list = [],
                   id_feature_station : list = [],
                   date_end : str = "2021-06-04"):

    all_indicator_names = feature_indicator_names + [target_indicator_name]
    print('Selected indicator names:', all_indicator_names)

    pd.set_option('max_colwidth',200)
    all_id_stations = id_feature_station + [id_target_station]
    data_about = pd.read_csv('../input/wq-southern-bug-river-
01052021/PB_stations.csv', sep=';', header=0, encoding='cp1251')
    print('All stations:')
    display(data_about.sort_values(by=['length'], ascending=False))
    print('\nSelected stations:')
    display(data_about[data_about['id'].isin(all_id_stations)])
```

```
    data = pd.read_csv('../input/wq-southern-bug-river-
01052021/PB_All_2000_2021.csv', sep=';', header=0)
    data['ds'] = pd.to_datetime(data['date'])


    df_indicator = data[['id', 'ds'] + all_indicator_names]
    df_indicator =
df_indicator[df_indicator['id'].isin(all_id_stations)].dropna().reset_index(drop=Tru
e)



    cols = []
    for station in all_id_stations:
        for feature in all_indicator_names:
            cols.append(str(station) + "_" + feature)


    df = pd.pivot_table(df_indicator, index=["ds"], columns=["id"],
values=all_indicator_names).dropna()
    df.columns = cols
    df = df.reset_index(drop=False)


    new_target_name = str(id_target_station) + "_" + target_indicator_name


    df = df[(df['ds']>=date_start) & (df['ds']<=date_end)].reset_index(drop=True)


    return df, all_indicator_names, all_id_stations, new_target_name


df, all_indicator_names, all_id_stations, target_name =
get_water_data(target_indicator_name,
```

```
                                                id_target_station,
                                                feature_indicator_names =
feature_indicator_names,

id_feature_station=id_feature_station)
print(f'\nData for processing (target name - "{target_name}"):')
df

df.describe()

def check_stationarity(series):

    result = adfuller(series.values)

    print('ADF Statistic: %f' % result[0])
    print('p-value: %f' % result[1])
    print('Critical Values:')
    for key, value in result[4].items():
        print('\t%s: %.3f' % (key, value))

    if (result[1] <= 0.05) & (result[4]['5%'] > result[0]):
        print("\u001b[32mStationary\u001b[0m")
    else:
        print("\x1b[31mNon-stationary\x1b[0m")

check_stationarity(df[target_name])

check_stationarity(df[target_name].diff().dropna())
```

```
check_stationarity(df[target_name].diff().diff().dropna())

if is_EDA_with_Pandas_Profiling:
    profile = df.profile_report(title='Pandas Profiling Report for dataset')
    profile.to_file(output_file="profile.html")
    display(profile)

df.reset_index(drop=False)



df = df.reset_index(drop=False)
df

if is_anomalies:
    fig = px.line(df, x="ds", y=target_name,
                  title=f"Investigation of dates of anomalous changes in the target rate",
                  log_y=False,template='gridon',width=800, height=600)
    fig.show()

df

if is_anomalies:
    anomalous_dates = ['2014-02-09', '2013-04-04']
    holidays_df = pd.DataFrame(columns = ['ds', 'lower_window', 'upper_window',
'prior_scale'])
    holidays_df['ds'] = anomalous_dates
    holidays_df['holiday'] = 'anomalous_dates'
    holidays_df['lower_window'] = 0
    holidays_df['upper_window'] = 0
```

```
    holidays_df['prior_scale'] = 10
    display(holidays_df)



def plot_with_anomalies(df, cols_y_list, cols_y_list_name, dates_x,
anomalous_dates, log_y=False):


    fig = px.line(df, x=dates_x, y=cols_y_list[0],
title=cols_y_list_name[cols_y_list[0]], log_y=log_y, template='gridon',width=800,
height=600)
    y_max = df[cols_y_list[0]].max()
    for i in range(len(cols_y_list)-1):
        fig.add_trace(go.Scatter(x=df[dates_x], y=df[cols_y_list[i+1]], mode='lines',
name=cols_y_list_name[cols_y_list[i+1]]))
        max_i = df[cols_y_list[i+1]].max()
        y_max = max_i if max_i > y_max else y_max


    y_min = min(df[cols_y_list[0]].min(),0)
    for i in range(len(anomalous_dates)):
        anomal_date = anomalous_dates[i]
        #print(anomal_date, y_min, y_max)
        fig.add_shape(dict(type="line", x0=anomal_date, y0=y_min,
x1=anomal_date, y1=y_max, line=dict(color="red", width=1)))
    fig.show()


if is_anomalies:
    plot_with_anomalies(df, [target_name],
                        {target_name : f"Anomalous dates for {target_name}"},
                        'ds', anomalous_dates, False)
```

```
final_data = 2021

years_num_period = 2

year1_start = 2016

year1_end = year1_start + years_num_period

year2_start = year1_start + years_num_period + 1

year2_end = year2_start + years_num_period if (year2_start + years_num_period) < final_data else final_data

print(f"Interval 1 - from {year1_start} to {year1_end}, Interval 2 - from {year2_start} to {year2_end}")


station_EDA = 14

all_indicator_names


df['year'] = df.ds.dt.year

df


def get_dataset(df, indicators_list, station, year_start, year_end):



    cols = ['14' + "_" + indicator for indicator in indicators_list]

    return df[(df['year'] >= year_start) & (df['year'] <= year_start)][cols].reset_index(drop=True)



df1 = get_dataset(df, all_indicator_names, station_EDA, year1_start, year1_end)

print('Dataset1:')

display(df1)


df2 = get_dataset(df, all_indicator_names, station_EDA, year2_start, year2_end)

print('\nDataset2:')
```

```
display(df2)


def get_date_for_year(year_int):
    # Get date as str for year as int and 01.01
    return str(year_int) + "-01-01"


plot_with_anomalies(df, [target_name],
                    {target_name : f"The starting dates of the two datasets for {target_indicator_name}"},
                    'ds', [get_date_for_year(year1_start), get_date_for_year(year2_start)], False)



df1_report = sv.analyze(df1)
df1_report.show_html('Analyst1.html')
IFrame(src = 'Analyst1.html',width=1000,height=600)

df2_report = sv.analyze(df2)
df2_report.show_html('Analyst2.html')
IFrame(src = 'Analyst2.html',width=1000,height=600)



comparison_report = sv.compare(df1,df2)
comparison_report.show_html('Comparison.html')
IFrame(src = 'Comparison.html',width=1000,height=600)



comparison_report = sv.compare(df1,df2)
```

```
comparison_report.show_html('Comparison_main.html')
IFrame(src = 'Comparison_main.html',width=1000,height=600)



df1['class'] = 0
df2['class'] = 1
data = pd.concat([df1, df2], axis=0)
data

from autoviz.AutoViz_Class import AutoViz_Class
%matplotlib inline
AV = AutoViz_Class()
AV.AutoViz(filename='',
        dfte=data,
        depVar='class',
        verbose=1,
        lowess=True,
        max_rows_analyzed=data.shape[0],
        max_cols_analyzed=data.shape[1])
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ ЗМІН КОНЦЕНТРАЦІЇ РОЗЧИНЕНОГО У ВОДІ КИСНЮ У РІЧЦІ ПІВДЕННИЙ БУГ

Виконав: студент гр. 2ІСТ-21м

_____ Головач В.Р.

«_01_» _____12_____ 2022 р.

Керівник: к.т.н., доц. каф. САІТ

_____ Козачко О. М.

«_02_» _____12_____ 2022 р.

Нормоконтроль: к.т.н., доцент

_____ Жуков С. О.

«_02_» _____12_____ 2022 р.

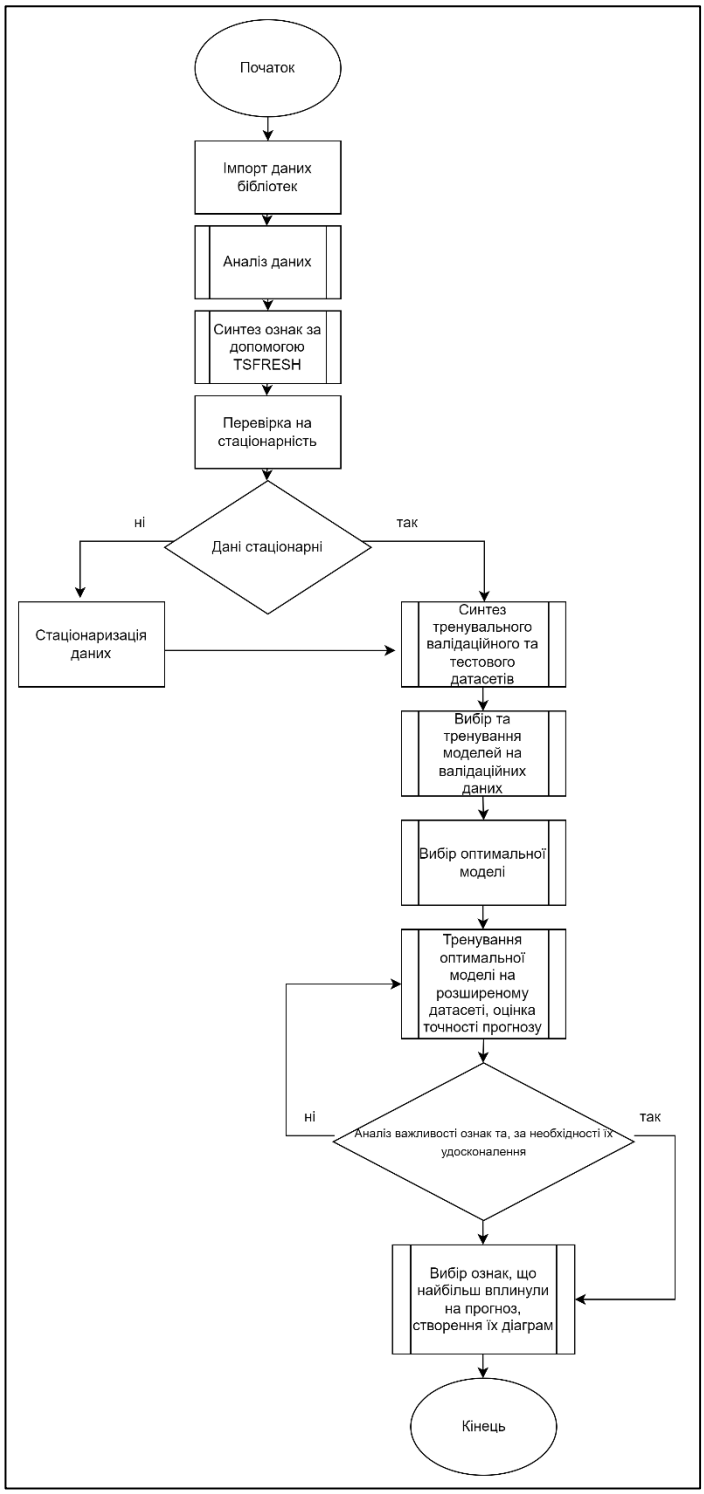


Рисунок Г.1 – Блок-схема алгоритму функціонування інформаційної технології прогнозування вмісту розчиненого у воді кисню в річці Південний Буг

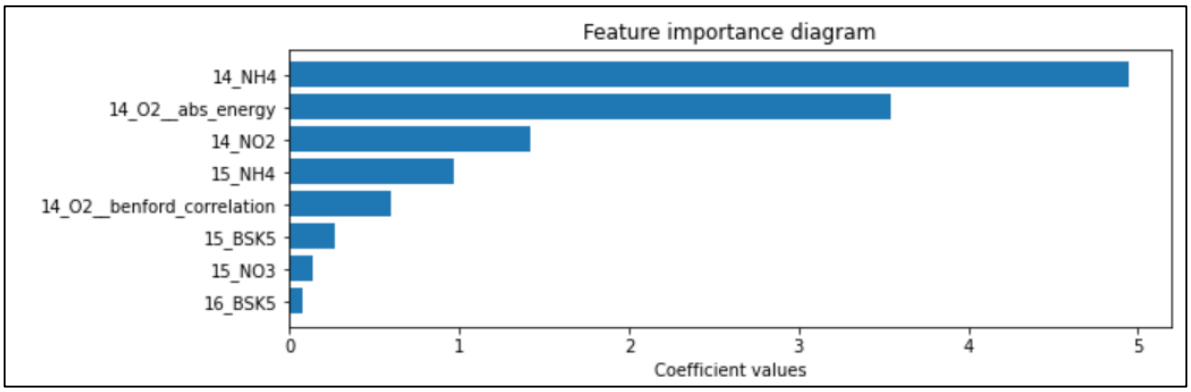
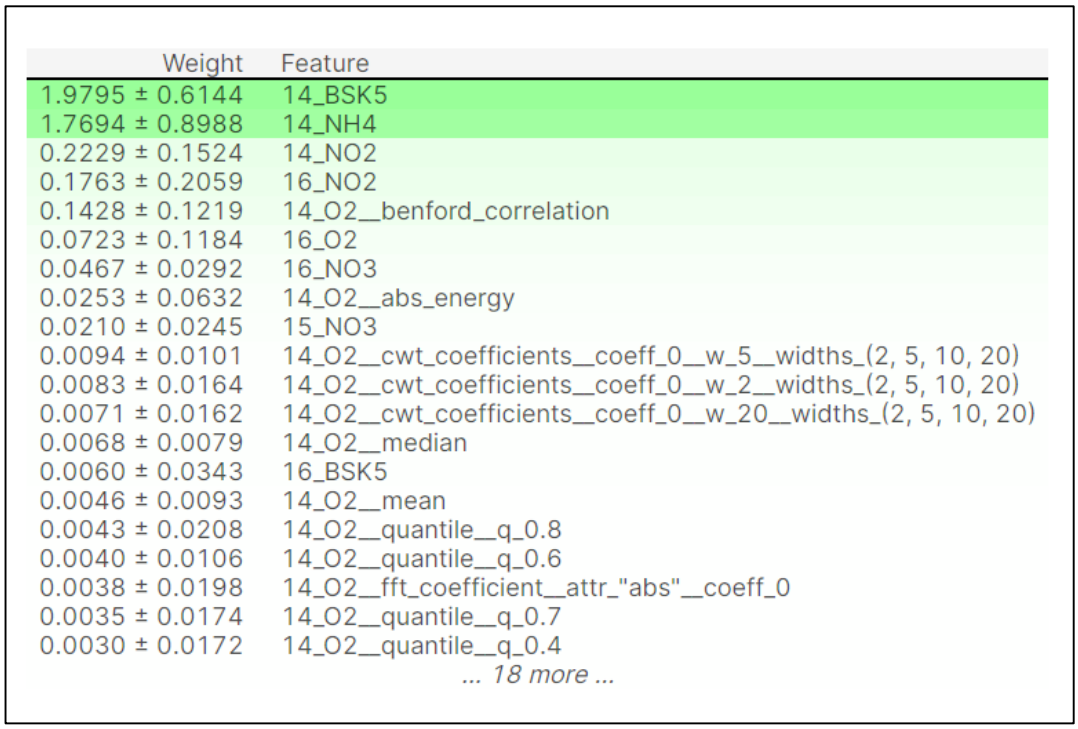


Рисунок Г.2 – Діаграма важливості ознак



Weight	Feature
1.9795 ± 0.6144	14_BSK5
1.7694 ± 0.8988	14_NH4
0.2229 ± 0.1524	14_NO2
0.1763 ± 0.2059	16_NO2
0.1428 ± 0.1219	14_O2__benford_correlation
0.0723 ± 0.1184	16_O2
0.0467 ± 0.0292	16_NO3
0.0253 ± 0.0632	14_O2__abs_energy
0.0210 ± 0.0245	15_NO3
0.0094 ± 0.0101	14_O2__cwt_coefficients__coeff_0__w_5__widths_(2, 5, 10, 20)
0.0083 ± 0.0164	14_O2__cwt_coefficients__coeff_0__w_2__widths_(2, 5, 10, 20)
0.0071 ± 0.0162	14_O2__cwt_coefficients__coeff_0__w_20__widths_(2, 5, 10, 20)
0.0068 ± 0.0079	14_O2__median
0.0060 ± 0.0343	16_BSK5
0.0046 ± 0.0093	14_O2__mean
0.0043 ± 0.0208	14_O2__quantile__q_0.8
0.0040 ± 0.0106	14_O2__quantile__q_0.6
0.0038 ± 0.0198	14_O2__fft_coefficient__attr_"abs"__coeff_0
0.0035 ± 0.0174	14_O2__quantile__q_0.7
0.0030 ± 0.0172	14_O2__quantile__q_0.4
	... 18 more ...

Рисунок Г.3 – Діаграма перестановки з важливістю ознак

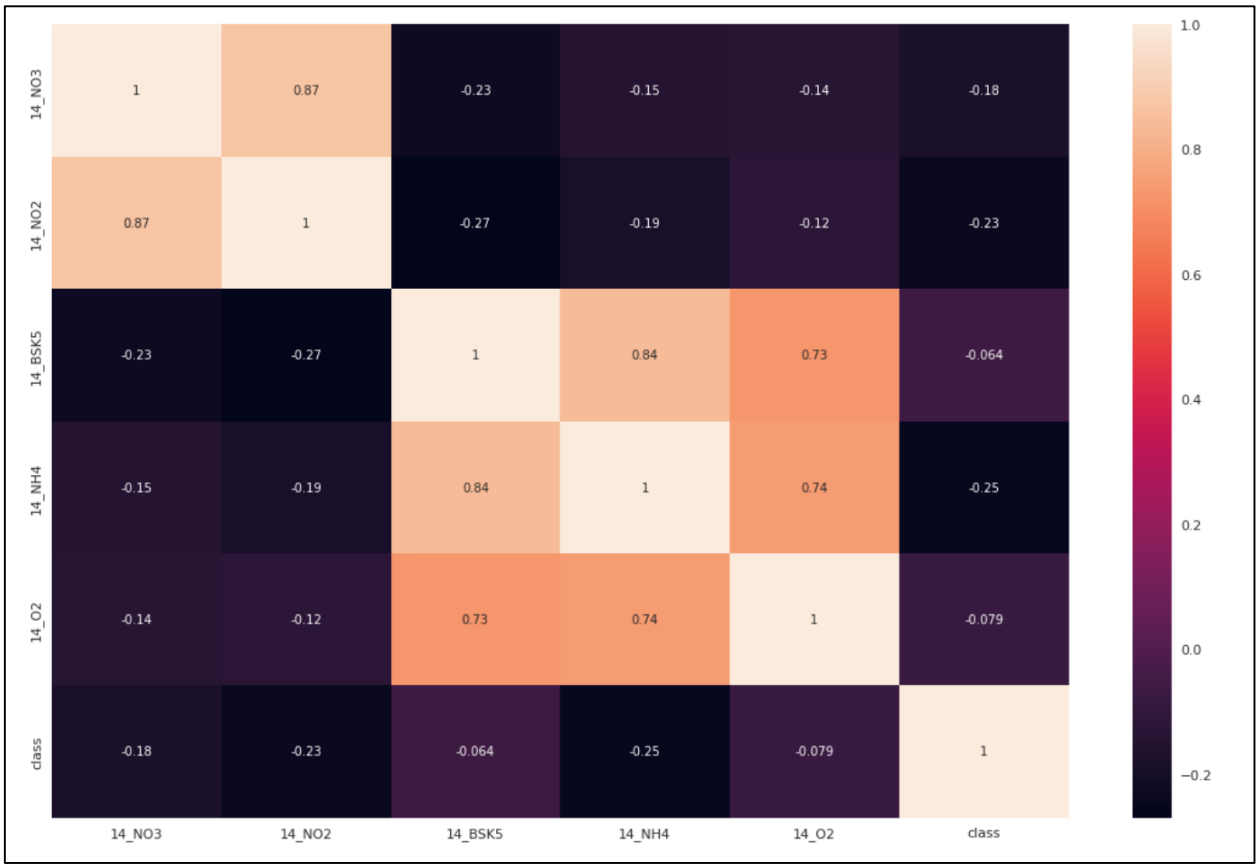


Рисунок Г.4 – Матриця кореляції

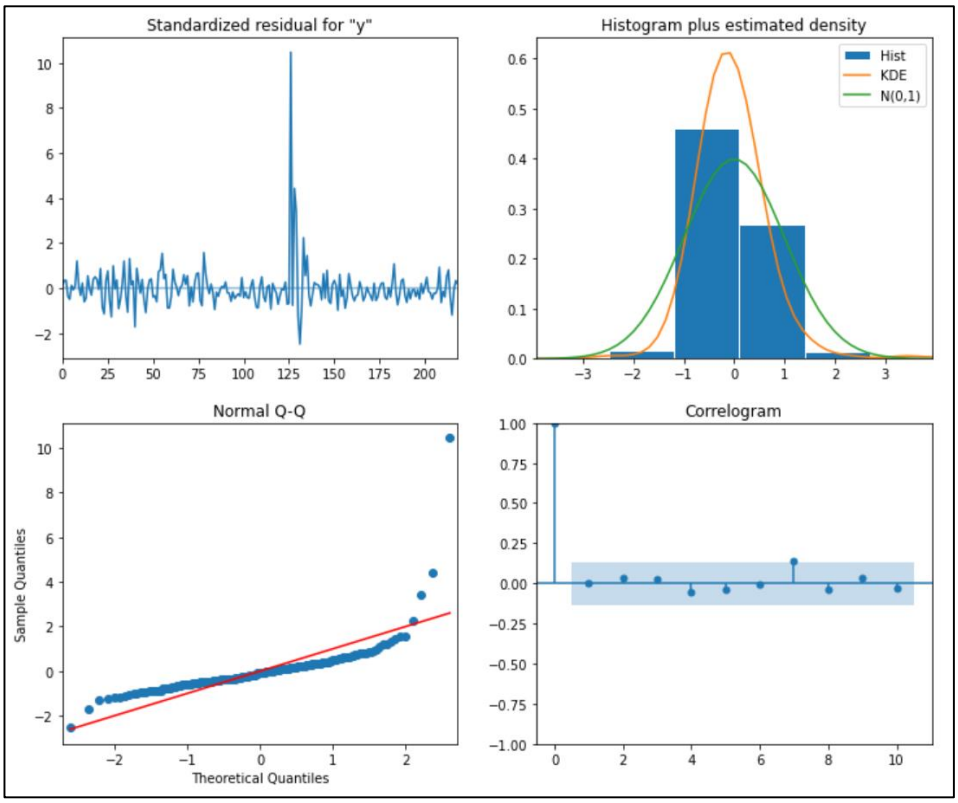


Рисунок Г.5 – Графік прогнозу валідаційних даних;

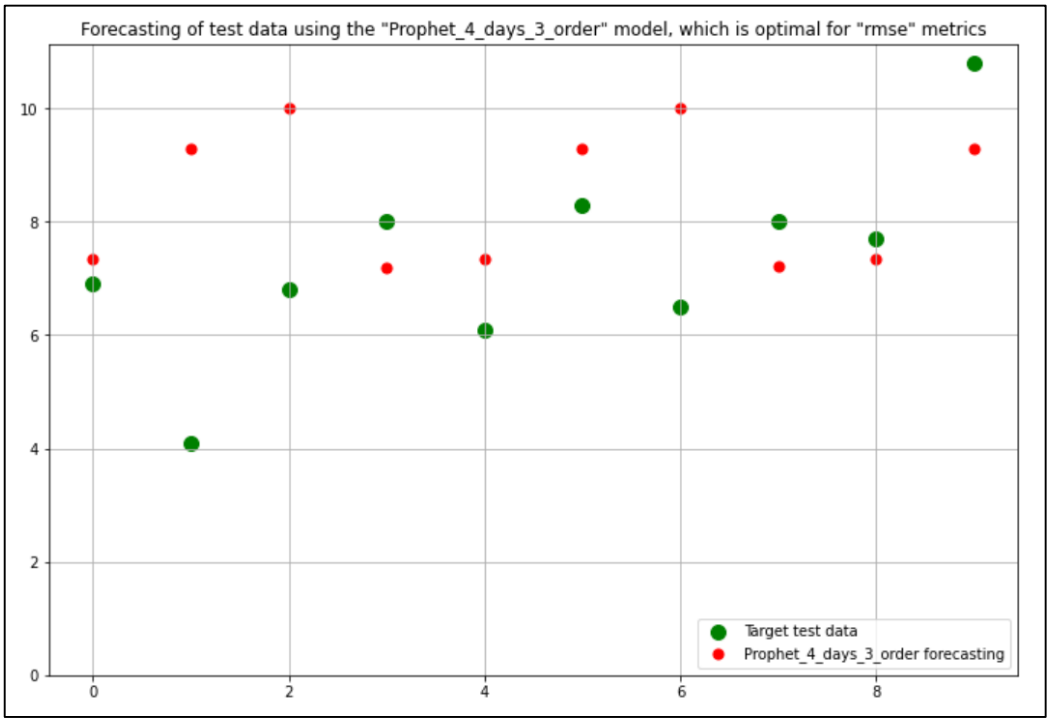


Рисунок Г.6 – Реальні тестові дані розчиненого у воді кисню та їх прогноз за моделлю Facebook Prophet, з параметрами оптимальними за метриками RMSE та MAPE