Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

«Дослідження моделей та методів штучного інтелекту для задачі прогнозування вилову риби»
(тема)

Виконав:здобувач групи СПРМ-22-2

Сєвєрін І.К
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми :освітньо-наукова

Освітня програма Системне проєктування
(повна назва освітньої програми)

Керівник доц. Хряпкін О.В
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ _____
(підпис) (прізвище, ініціали)

2024 р.

Я як студент ХНУРЕ розумію і підтримую політику закладу із академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

10.06.2024 Сєвєрін

(дата, підпис, прізвище студента/-ки)

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Попередній захист проведено 18 червня 2024 р.

Керівник кваліфікаційної роботи доц. Хряпкін О.В

Харківський національний університет радіоелектроніки

Факультет Комп’ютерних наук

Кафедра Системотехніки

Рівень вищої освіти другий(магістерський))

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Освітня програма Комп'ютерні науки
(повна назва)

ЗАТВЕРДЖУЮ:
Зав.кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Сєвєріну Іллі Костянтиновичу
(прізвище, ім’я, по батькові)

1. Тема роботи Дослідження моделей та методів штучного інтелекту для задачі прогнозування вилову риби
затверджена наказом по університету від 09 червня 2024 р. № 684Ст
2. Термін подання студентом роботи до екзаменаційної комісії 19 червня 2024 р.
3. Вихідні дані до роботи: Провести дослідження методів для задачі прогнозування та аналіз впливу вхідних атрибутів на результат.
4. Перелік питань, що потрібно опрацювати в роботі : Аналіз предметної області та формулювання задачі, аналіз існуючих систем, дослідження методів для задачі та аналіз вхідних атрибутів
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп’ютерних ілюстрацій (слайдів) Рисунок 1.1 – Типовий процес класичного прогнозування, Рисунок 1.2 – Процес прогнозування машинного навчання, Рисунок 1.3 – Нейронна мережа для лінійної регресії з чотирма предикторами, Рисунок 1.4 – Нелінійна нейронна мережа, Рисунок 2.1 – Етапи отримання та обробки даних, Рисунок 2.2 – Діаграма розподілу спійманої риби між 9 найбільш популярними видами, Рисунок 2.3 – Розподіл повідомлень про улов видів риб, досліджених у цій роботі, Рисунок 2.4 – Гістограма уловів риби, Рисунок 2.5 – Схема системи захисту від перенавчання, Рисунок 2.6 – Схема багатошарового персептрона, Рисунок 3.1 - Підключення бібліотек та завантаження датасету, Рисунок 3.2 - Пропуски та аномалії у наборі даних, Рисунок 3.3 - Реалізація очищення даних, Рисунок 3.4 - Набір даних після очищення, Рисунок 3.5 - Підключення бібліотек та завантаження даних, Рисунок 3.6 - Реалізація розділення даних, Рисунок 3.7 - Реалізація методу лінійої регресії, Рисунок 3.8 - Реалізація методу випадкового лісу, Рисунок 3.9 - Реалізація методу багатошарового персептрону, Рисунок 3.10 - Реалізація методу регресії на основі опорних векторів, Рисунок 3.11 – Коефіцієнти помилок алгоритмів, Рисунок 3.12 Впровадження методу перетворення у sin/cos,

Рисунок 3.13 – Результат перетворення атрибутів у sin/cos, Рисунок 3.14 – Абсолютна різниця помилок для декількох ітерацій накопичення атрибутів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Хряпкін О.В		

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	01.04.2024	виконано
2	Аналіз предметної області та постановка задачі	21.04.2024	виконано
3	Аналіз методів для вивчення	30.04.2024	виконано
4	Огляд літератури	04.05.2024	виконано
5	Теоретичне лослідження методів для аналізу	10.05.2024	виконано
6	Дослідження вибраних методів та впливовості вхідних атрибутів	25.05.2024	виконано
7	Оформлення пояснювальної записки	15.06.2024	виконано
8	Представлення на рецензування	17.06.2024	виконано
9	Попередній захист	18.06.2024	виконано
10	Захист перед ЕК	20.06.2024	виконано

КАЛЕНДАРНИЙ ПЛАН

Дата видачі завдання _1_ _квітня__2024 р.

Студент_____ Сєвєрін І.К.

(підпис) (прізвище, ініціали)

Керівник роботи _____ доц. Хряпкін О.В.

(підпис) (посада прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 67 с., 25 рис., 2 додаток, 25 джерел.

МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ, ЛІНІЙНА РЕГРЕСІЯ, БАГАТОШАРОВИЙ ПЕРСЕПТРОН, ВИПАДКОВЕ ДЕРЕВО, РЕГРЕСІЯ НА ОСНОВІ ОПОРНИХ ВЕКТОРІВ, SKLEARN

Об'єкт дослідження – Методи прогнозу клювання риби, що допомагають рибалкам оцінювати погодні та астрономічні умови, які впливають на успішність риболовлі. Вплив вхідних даних на результат.

Предмет дослідження – Методи машинного навчання для прогнозування клювання риби на основі погодних та астрономічних умов.

Мета роботи – Дослідження методів машинного навчання для системи прогнозування, яка оцінює активність риби на основі погодних та астрономічних умов. Аналіз вхідних атрибутів до досліджених методів з метою оцінки їх впливу на результат.

Методи дослідження – Порівняння декількох моделей та методів машинного навчання для задачі прогнозування. Порівняння впливу вхідних даних на результат.

В кваліфікаційній роботі проведено порівняння різних методів машинного навчання для задачі прогнозування. Проведено аналіз вхідних даних, та було визначено які з них впливають на результат найбільше. Наведено детальний аналіз і опис найбільш відповідного методу і впливу вхідних данних на результат.

ABSTRACT

Explanatory note: 67 pp., 25 figs., 2 appendix, 25 sources.

MACHINE LEARNING, FORECASTING, LINEAR REGRESSION, MULTILAYER PERCEPTRON, RANDOM FOREST, SUPPORT VECTOR REGRESSION, SKLEARN

Object of study: Methods of predicting fish bites, which help anglers evaluate weather and astronomical conditions affecting fishing success. The impact of input data on the result.

Subject of study: Machine learning methods for predicting fish bites based on weather and astronomical conditions.

Purpose of the work: To investigate machine learning methods for a forecasting system that evaluates fish activity based on weather and astronomical conditions. Analysis of input attributes to the studied methods to assess their impact on the result.

Research methods: Comparison of several models and machine learning methods for the forecasting task. Comparison of the impact of input data on the result.

In the graduation work, a comparison of different machine learning methods for the forecasting task was conducted. An analysis of the input data was performed, and it was determined which of them have the greatest impact on the result. A detailed analysis and description of the most suitable method and the impact of input data on the result are provided.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI (Artificial intelligence) – Штучний інтелект;

ANN (Artificial Neural Networks) – Штучні нейронні мережі;

API (Application programming interface) – Інтерфейс програмування додатків;

ARIMA (Autoregressive integrated moving average) – Авторегресивна інтегрована ковзна середня;

CPUE (Catch per unit effort) – Улов за одиницю зусилля;

GB (Gradient Boosting) – Градієнтний бустинг;

ML (Machine Learning) – Машинне навчання;

RF (Random Forest) – Випадковий ліс;

SVM (Support Vector Machines) – Метод опорних векторів.

ВСТУП

Машинне навчання відкриває нові горизонти в аналізі та використанні даних, пропонуючи передові методи для ефективного вилучення корисної інформації з великих наборів даних. Цей процес полягає в тому, що спеціалізовані алгоритми, навчаючись на основі історичних даних, адаптуються до особливостей цих даних і в подальшому здатні з високою точністю аналізувати та прогнозувати майбутні тенденції та події. Це ключовий аспект у сфері великих даних, де машинне навчання відіграє роль вирішального інструменту для отримання глибоких інсайтів з накопиченої інформації.

Завдяки поєднанню традиційних статистичних методів і сучасних алгоритмічних підходів, системи машинного навчання виявляють зразки, що допомагає бізнесам не тільки виконувати класифікацію даних або їх прогнозування, але й визначати майбутні стратегії на основі об'єктивної інформації. Така можливість зробила машинне навчання невід'ємною частиною аналітичних проектів у багатьох секторах економіки

Особливо актуальним машинне навчання є у сфері наук про дані, де воно служить основою для розробки новітніх аналітичних інструментів. Величезні обсяги даних, які нині генеруються у всіх сферах життя, вимагають високоефективного інструментарію для їх обробки, а машинне навчання дозволяє обробляти ці дані не тільки швидко, але й з необхідною точністю.

Застосування машинного навчання не обмежується якоюсь однією галуззю: воно ефективно використовується в таких різних областях, як фінанси, де допомагає виявляти шахрайство, охорона здоров'я, де сприяє діагностиці захворювань, та ритейл, де оптимізує логістику і керування запасами. Інші приклади включають маркетинг, де технології машинного навчання допомагають у розумінні споживчої поведінки, та кібербезпеку, де

вони здатні ідентифікувати та нейтралізувати потенційні загрози в реальному часі.

Завдяки машинному навчанню, організації отримують інструмент для вирішення складних задач, оптимізації своїх процесів та мінімізації ризиків, що в сучасному швидкозмінному світі є надзвичайно цінним. Відомості, отримані за допомогою алгоритмів машинного навчання, дозволяють компаніям не просто реагувати на поточні виклики, а й адаптуватися до майбутніх змін, забезпечуючи їм конкурентну перевагу.

Таким чином, машинне навчання відіграє вирішальну роль в еволюції сучасних технологій, перетворюючи способи аналізу, обробки та використання даних у невід'ємну складову сучасної інформаційної епохи, що формує майбутнє усіх сфер життєдіяльності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Еволюція методологій прогнозування

Прогнозування відіграє критичну роль у численних галузях діяльності людини, від економіки та науки до управління підприємствами та технологій. Його сутність полягає в аналізі існуючих даних та використанні їх для прогнозування майбутніх подій або результатів. Розвиток методологій прогнозування можна простежити через різні історичні етапи, кожен з яких вніс свої корективи в те, як ми розуміємо та використовуємо статистичні та аналітичні інструменти сьогодні.

Ранні методи прогнозування базувалися переважно на інтуїції та досвіді. Наприклад, у давнину фермери використовували знання про пори року та погодні умови для прогнозування найкращого часу для посіву та збору врожаю. Такі методи були суб'єктивними та часто залежали від місцевих знань та традицій.

З настанням промислової ери, потреба в точніших і обґрунтованих прогнозах почала зростати. Це призвело до розвитку статистичних методів прогнозування. Вчені та математики почали використовувати математичні моделі для обробки даних та виведення прогнозів. Завдяки теорії ймовірностей та статистиці, прогнози стали більш обґрунтованими та повторюваними.

Епоха комп'ютеризації внесла революційні зміни в методології прогнозування. З появою комп'ютерів та збільшенням обчислювальних потужностей вчені отримали змогу обробляти великі масиви даних та застосовувати більш складні алгоритми, такі як штучні нейронні мережі та машинне навчання. Це дозволило не тільки покращити точність прогнозів, але й значно розширити їх застосування.

Сучасні технології біг дата і машинне навчання знову трансформували пейзаж прогнозування, роблячи його ще більш динамічним та інтегрованим.

Методи глибокого навчання та алгоритми на основі штучного інтелекту дозволяють нам аналізувати не тільки кількісні, але й якісні дані, збирати та аналізувати враження користувачів, їхні відгуки та поведінку в реальному часі.

Прогнозування сьогодні є не просто інструментом для підготовки до майбутнього, а ключовим елементом у прийнятті рішень в усіх сферах діяльності. Завдяки постійному розвитку методологій та технологій, ми можемо активно впливати на рішення щодо майбутнього, що робить прогнозування необхідним інструментом у сучасному світі [3].

1.2 Класичні методи прогнозування

Прогнозування є важливою складовою в багатьох галузях, допомагаючи організаціям передбачати майбутні події та ухвалювати зважені рішення. Класичні методи прогнозування базуються на використанні перевірених статистичних моделей та алгоритмів, які дозволяють аналізувати минулі дані та робити прогнози з високою точністю.

Одним з найбільш поширених класичних методів є лінійна регресія. Цей підхід полягає у визначенні лінійної залежності між незалежними змінними (пояснювальними предикторами) та залежною змінною (цільовим показником). Лінійна регресія дозволяє оцінити вплив кожного предиктора на цільовий показник та створити прогнозну модель. Проте, цей метод має обмеження у випадках, коли зв'язок між змінними є нелінійним або коли дані містять складніші структури.

Інший ключовий метод — це авторегресійна інтегрована ковзна середня (ARIMA). Ця модель широко використовується для аналізу часових рядів, тобто даних, зібраних у часовій послідовності. ARIMA об'єднує три компоненти: авторегресію (AR), інтегрування (I) та ковзну середню (MA). Авторегресія враховує залежність поточного значення ряду від попередніх значень, інтегрування використовується для перетворення нестаціонарних

рядів у стаціонарні, а ковзна середня враховує вплив випадкових шумів на прогноз. Завдяки цим компонентам, ARIMA є потужним інструментом для моделювання та прогнозування часових рядів.

Для більш точного прогнозування часто застосовується модель ARIMAX, яка є розширенням ARIMA і включає додаткові пояснювальні змінні. Це дозволяє враховувати зовнішні фактори, що можуть впливати на цільову змінну, підвищуючи точність прогнозів.

Класичні методи прогнозування зазвичай використовуються для аналізу одновимірних або багатовимірних наборів даних з обмеженими та визначеними предикторами. Наприклад, у сфері маркетингу та продажів, де необхідно прогнозувати обсяги продажів продуктів, класичні статистичні методи можуть бути надзвичайно ефективними. Історичні дані про продажі, ціни, сезонні коливання та інші фактори дозволяють створювати точні прогнозні моделі.

Процес класичного прогнозування складається з кількох основних етапів. Спочатку здійснюється збір та підготовка даних, потім дані аналізуються для виявлення основних закономірностей. На основі цього аналізу створюється модель, яка оптимізується для досягнення найкращої відповідності між фактичними та прогнозованими значеннями. Після цього модель використовується для прогнозування майбутніх значень цільової змінної.

На рисунку 1.1 зображений типовий процес класичного прогнозування, включаючи етапи збору даних, аналізу, побудови моделі та прогнозування.

Рисунок 1.1 – Типовий процес класичного прогнозування

1.2.1 Основи методу лінійної регресії

Лінійна регресія - це фундаментальний статистичний метод, що використовується для моделювання і прогнозування залежної змінної на основі однієї або кількох незалежних змінних. Цей метод широко застосовується у різних галузях завдяки своїй простоті та ефективності.

Однофакторна лінійна регресія розглядає зв'язок між двома змінними: однією незалежною змінною та однією залежною змінною. Цей підхід використовується, коли є припущення про існування прямого лінійного зв'язку між двома змінними.

Багатофакторна лінійна регресія є розширенням однофакторної і дозволяє враховувати вплив кількох незалежних змінних одночасно. Цей метод є корисним для виявлення і кількісної оцінки впливу різних факторів на залежну змінну.

Лінійні регресійні моделі мають широке застосування у різних галузях, включаючи біологію, поведінкові науки, екологію, соціальні науки та бізнес. Вони використовуються для аналізу і прогнозування даних. Основна перевага лінійної регресії полягає у її простоті і зрозумілості, що дозволяє швидко навчати моделі та отримувати інтерпретовані результати.

Лінійна регресія забезпечує надійний та перевірений підхід до прогнозування майбутніх значень. Завдяки тривалому використанню та ретельному вивченню цього методу, властивості моделей лінійної регресії добре зрозумілі, що робить цей підхід основним інструментом у статистичному аналізі та прогнозуванні.

1.2.2 Авторегресійна інтегрована ковзна середня

Модель авторегресійної інтегрованої ковзної середньої (ARIMA) є потужним інструментом для аналізу та прогнозування часових рядів. Вона широко використовується для виявлення тенденцій і прогнозування майбутніх значень на основі історичних даних.

Модель ARIMA поєднує в собі три ключові компоненти: авторегресію, інтеграцію та ковзне середнє. Кожен з цих компонентів відіграє важливу роль у моделюванні часових рядів.

Авторегресія вказує на те, що поточні значення часового ряду залежать від попередніх значень. Іншими словами, модель використовує лінійну комбінацію попередніх значень для прогнозування майбутніх значень.

Інтеграція використовується для усунення тренду в даних, що робить ряд стаціонарним. Стаціонарність означає, що статистичні властивості ряду залишаються постійними з часом, що є важливою умовою для застосування моделі ARIMA.

Ковзне середнє використовує попередні похибки прогнозування для поліпшення точності моделі. Він враховує відхилення попередніх значень від прогнозованих значень для корекції майбутніх прогнозів.

Модель ARIMA знайшла широке застосування у різних галузях завдяки своїй здатності точно прогнозувати майбутні значення на основі історичних даних. Економіка та фінанси: моделі ARIMA використовуються для прогнозування економічних показників, таких як ВВП, інфляція та курси валют. Вони також допомагають у прогнозуванні цін на акції та інших фінансових інструментів. Бізнес: компанії використовують ARIMA для прогнозування попиту на продукцію, управління запасами та оптимізації виробничих процесів. Точні прогнози дозволяють знизити витрати та підвищити ефективність бізнес-процесів. Кліматичні дослідження: моделі ARIMA застосовуються для прогнозування погодних умов та моніторингу довгострокових кліматичних змін. Вони можуть допомогти у виявленні тенденцій у змінах температури, концентрації парникових газів та інших кліматичних показників.

Однією з головних переваг моделі ARIMA є її здатність враховувати складні патерни в даних часових рядів, що робить її дуже корисною для прогнозування. Проте, модель має також і обмеження. Вона вимагає стаціонарності ряду, що може бути складним завданням для даних з сильними трендами або сезонними компонентами.

1.3 Прогнозування за допомогою методів машинного навчання

Машинне навчання пропонує сучасні та потужні підходи до прогнозування, які можуть значно підвищити точність порівняно з традиційними методами. Основна мета методів машинного навчання для прогнозування залишається такою ж, як і у традиційних методів - максимізувати точність прогнозів при мінімізації функції втрат. Зазвичай функція втрат визначається як сума квадратів похибок у прогнозуванні.

Однією з основних відмінностей між традиційними методами та методами машинного навчання є підхід до мінімізації функції втрат. Традиційні методи зазвичай використовують лінійні моделі, в той час як машинне навчання часто застосовує нелінійні моделі для досягнення цієї мети. Процес прогнозування з використанням машинного навчання зображено на рисунку 1.2.

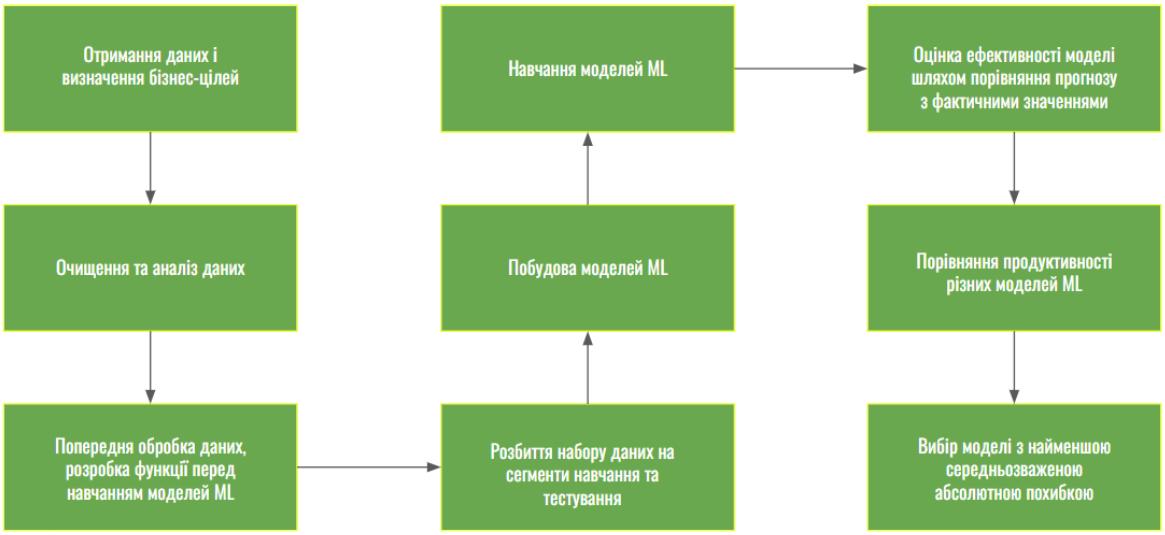

Рисунок 1.2 – Процес прогнозування машинного навчання

У сфері обчислювальних ресурсів вимоги до методів машинного навчання зазвичай є більшими, ніж в статистичних підходах. Хоча, варто визнати, що однією з основних проблем таких методів є складність інтерпретації отриманих моделей. Тим не менш, в сценаріях, де дані належать до великих обсягів та мають велику кількість характеристик, машинне навчання може виявитися пріорітетрнішим за традиційні методами завдяки здатності враховувати складні закономірності та взаємозв'язки.

У великих бізнес-додатках, де розглядаються великі обсяги даних, машинне навчання може забезпечити більш точні прогнози, враховуючи свою здатність моделювати складні взаємозв'язки та нелінійність. Наприклад, в аналізі ризику дефолту за кредитними заявками методи

машинного навчання можуть враховувати тисячі факторів, що впливають на цей ризик, роблячи їх ефективнішими у порівнянні з традиційними методами.

У той час як методи машинного навчання показують високу ефективність у випадках з великими наборами даних і складними взаємозв’язками, традиційні статистичні методи можуть бути більш доцільними для простіших задач, де важливі прозорість і зрозумілість моделі. Традиційні методи можуть забезпечити адекватну точність при значно менших обчислювальних ресурсах і кращій інтерпретованості результатів.

Методи прогнозування машинного навчання є високоефективними в програмах, метою яких є навчання на наборах даних з багатьма характеристиками, де пояснення моделі не настільки критичне. Для випадків з великими наборами даних методи прогнозування на основі машинного навчання часто працюють з більшою точністю, хоча і вимагають більших обчислювальних ресурсів.

1.3.1 Аналіз моделі випадкового лісу

В сучасному аналізі даних випадковий ліс займає одне з провідних місць серед методів машинного навчання. Ця модель використовується для різноманітних завдань, починаючи від класифікації об'єктів до прогнозування числових значень [4].

Відмінністю випадкового лісу є те, що він базується на ансамблі дерев прийняття рішень. Кожне дерево у лісі вирішує певне підзавдання, і кінцевий прогноз формується на основі агрегації прогнозів всіх дерев. Це дозволяє випадковому лісу бути стійким до перенавчання і забезпечує високу точність прогнозів.

У порівнянні з іншими моделями, такими як лінійна регресія чи нейронні мережі, випадковий ліс володіє великою гнучкістю і може

враховувати складні нелинійні залежності в даних. Це робить його відмінним інструментом для прогнозування в різних галузях, від фінансів до медицини.

Також важливою перевагою випадкового лісу є його здатність працювати з великими обсягами даних, включаючи ті, де кількість ознак може перевищувати декілька тисяч. Це робить модель випадкового лісу популярним варіантом для аналізу великих датасетів і прийняття даних масштабних прогнозів [5].

В цілому, випадковий ліс є потужним інструментом для аналізу даних та прогнозування, який знаходить своє застосування в різних галузях та завданнях. Його гнучкість, стійкість до перенавчання і висока точність роблять його незамінним інструментом для аналітиків та дослідників у сучасному світі даних.

1.3.2 Огляд та застосування штучної нейронної мережі

Штучні нейронні мережі - це потужний інструмент у сфері машинного навчання, який моделює навчальні процеси за аналогією зі структурою та функціонуванням біологічного мозку. Їхній високий ступінь гнучкості дозволяє моделювати складні нелінійні зв'язки у даних, але разом з тим, вони можуть вимагати значних обчислювальних ресурсів для навчання та застосування.

Нейронні мережі складаються зі шарів нейронів, кожен з яких приймає вхідні сигнали, обчислює їх ваговану суму та передає результат наступному шару. При цьому впроваджуються нелінійні функції активації, які дозволяють моделі вирішувати складні завдання, які не можна розв'язати за допомогою простих лінійних моделей.

Процес навчання нейронних мереж включає оптимізацію вагових коефіцієнтів за допомогою алгоритмів зворотного поширення помилки. Цей

процес полягає у визначенні таких значень ваг, які мінімізують функцію втрат між прогнозованими та фактичними вихідними значеннями.

Однією з особливостей нейронних мереж є їх здатність до автоматичного виявлення та використання внутрішніх залежностей у даних, що робить їх ефективними в різних завданнях, від класифікації до регресії.

Хоча нейронні мережі можуть бути потужним інструментом для аналізу даних, вони також можуть вимагати значних обчислювальних ресурсів, особливо у випадку великих наборів даних або складних архітектур мережі.

У зв'язку з цим, використання нейронних мереж вимагає ретельного аналізу задачі та ресурсів, доступних для навчання та використання моделі. Однак при правильному застосуванні вони можуть забезпечити високу точність прогнозів та цінні інсайти у дані.

Стандартні прості моделі нейронних мереж не включають прихованих шарів і можуть бути порівняні з лінійними моделями регресії. На рисунку 1.3 представлений приклад нейромережевої версії лінійної регресії з чотирма предикторами.

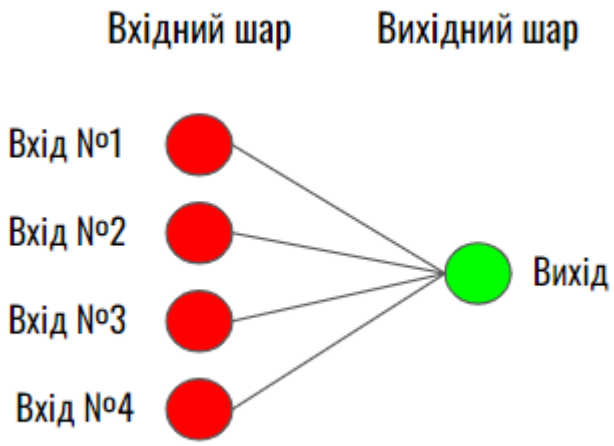

Рисунок 1.3 – Нейронна мережа для лінійної регресії з чотирма предикторами

Коли до нейронної мережі додається проміжний шар із прихованими нейронами, її структура стає нелінійною. Приклад показано на рисунку 1.4.

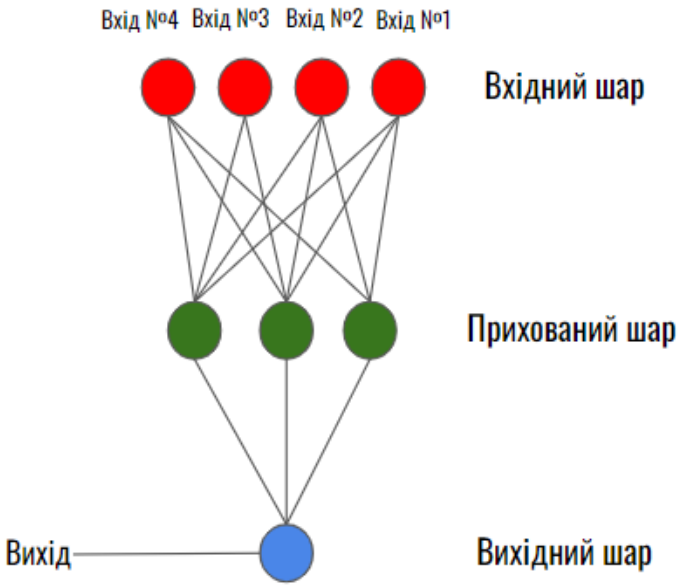

Рисунок 1.4 – Нелінійна нейронна мережа

Цей тип мережі відомий як багаторівнева мережа з прямим зв'язком, де кожен рівень вузлів отримує вхідні дані від попередніх шарів. Виходи вузлів одного шару стають входами для наступного шару. Всі вхідні дані до кожного вузла об'єднуються за допомогою зваженої лінійної комбінації, і після цього отриманий результат піддається не лінійній функції перед виведенням.

1.4 Постановка задачі

У процесі виконання дослідження потрібно аналізувати алгоритми прогнозування, спрямовані на розв'язання завдання прогнозування рибальства, і вибрати найбільш ефективний серед них з врахуванням відповідних характеристик. Окрім цього, слід дослідити вхідні параметри і

емпіричним шляхом визначити ті з них, які найбільше впливають на результативність прогнозу.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- теоретичні дослідження та проектування: провести аналіз методологій прогнозування, включаючи класичні методи (лінійна регресія, авторегресійна інтегрована ковзна середня) та сучасні підходи (випадковий ліс, багатошаровий персептрон, метод опорних векторів).Дослідити дані для аналізу, зокрема, звіти про улов рибалок, погодні дані та інші необхідні атрибути. Підготувати дані для аналізу, включаючи фільтрацію та нормалізацію даних.
- аналіз методів та моделей: описати та реалізувати обрані алгоритми машинного навчання, використовуючи бібліотеку Scikit-learn для Python. Визначити міру успішності прогнозування улову риби та провести порівняльний аналіз точності прогнозування різними методами.
- практичне дослідження: спроектувати систему прогнозування улову риби на основі машинного навчання. Реалізувати програмну частину системи, включаючи завантаження та обробку даних, навчання моделей, прогнозування та оцінку результатів. Підсумувати результати та вдосконалити алгоритм, який показав найкращі результати.
- аналіз важливості вхідних атрибутів: провести аналіз впливу вхідних параметрів на точність прогнозування. Визначити найзначущіші атрибути, які впливають на результативність прогнозів.

Для аналізу прогнозування клювання риби будуть використані наступні сім характеристик: вага, середня температура повітря, швидкість вітру, місячний день, різниця тиску, хмарність і ймовірність опадів. Дані для проведення аналізу будуть зібрані з риболовлі на території України.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ ТА ПРОЕКТУВАННЯ

2.1 Дані для аналізу

Дані для аналізу збираються з різних джерел і об’єднуються в єдине джерело даних, цей процес показано на рисунку 2.1. Варто зауважити, що ці дані в основному є відкритими та доступними, і деякі з них можуть бути створені користувачами. Це означає, що для забезпечення якості та надійності даних необхідна їх фільтрація та нормалізація, яка ретельно описана в наступних підрозділах.

Джерела даних можуть включати в себе різноманітні джерела, такі як відкриті бази даних, наукові дослідження, статистичні звіти тощо. Окрім того, користувачі також можуть сприяти у створенні даних, наприклад, шляхом внесення своїх спостережень або відгуків.

Після збирання даних необхідно виконати їх фільтрацію та нормалізацію для забезпечення їхньої якості та придатності для подальшого аналізу. Ці процеси детально розглянуті в наступних підрозділах.

Кінцеві атрибути, які використовуються для контрольованого машинного навчання, наведені у відповідній таблиці разом з одиницями вимірювання та коментарями предсталені у таблиці 2.1.

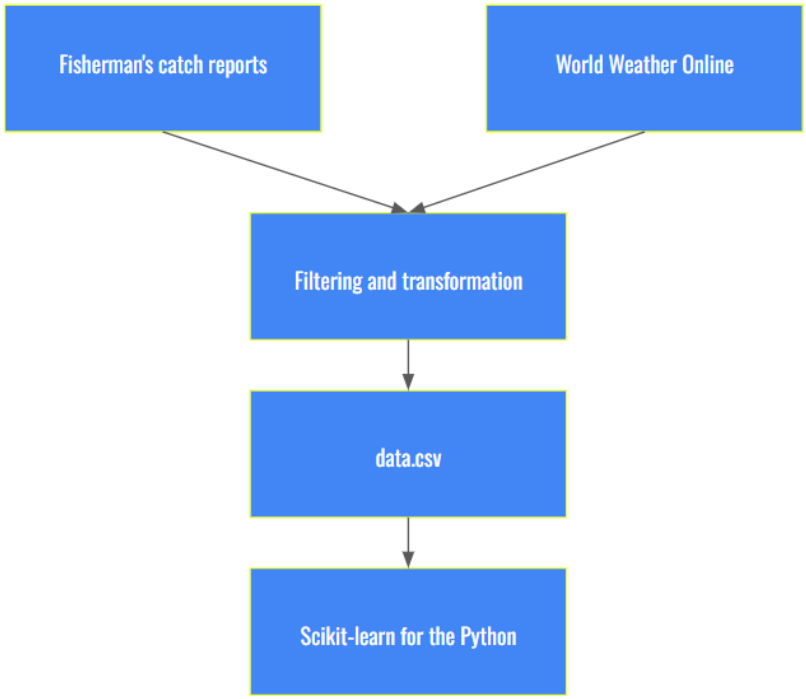

Рисунок 2.1 – Етапи отримання та обробки даних

2.1.1 Звіти про улов рибалок

Процес збору даних про рибальський улов передбачав використання широкого спектру джерел інформації. Крім веб-сайтів та форумів, де рибалки діляться своїм досвідом та результатами виїздів на риболовлю, було звернено увагу і на соціальні мережі, де цінна інформація може знаходитися в коментарях під постами та фотографіями. Наприклад, сторінки в Facebook або Instagram різноманітних риболовних груп та спільнот можуть містити цінні звіти та фотографії про улов.

Під час аналізу звітів було важливо врахувати можливість помилок або неточностей у введених даних. Одним із критеріїв обрання даних для подальшого аналізу була їхня достовірність. Такі атрибути, як дата риболовлі, місце улову, вага риби та її види, відігравали важливу роль у

відборі даних. Звіти з неповною або сумнівною інформацією були виключені з розгляду.

Було також проведено аналіз ваги улову, оскільки іноді деякі звіти могли містити завищені або неправдиві дані. Для забезпечення точності даних проводилася фільтрація та вилучення невірних записів. В результаті було зібрано достатньо об'єктивних та достовірних даних для подальшого аналізу, які узагальнено в таблиці 2.1 разом із відомостями про атрибути, одиниці вимірювання та коментарі.

Таблиця 2.1 – Обрані атрибути для аналізу рибальського улову, з одиницями вимірювання і коментарями

Атрибут	Одиниця виміру	Коментар
Вага	kg	Вага спійманої риби певного виду за риболовлю
Середня температура	°C	Середня температура повітря
Швидкість вітру	m/s	Середня швидкість вітру
Місячний день	day	Місячний день від 0 до 30
Різниця тиску	mm	Різниця тисків
Хмарність	%	Середня хмарість за день
Ймовірність опадів	%	Середня ймовірність опадів за день

Як можна бачити на рисунку 2.2, розподіл улову за видами нерівномірний. Карась звичайний, короп, лящ домінує за кількістю виловів, і з цієї причини був обраний основним видом для дослідження в цьому звіті. Білий амур та плітка були обрані як вторинні види для дослідження. Саме всі ці види риби були обрані для аналізу, тому що вони мають схожі сприятливі для ловлі.

На рисунку 2.2 відображено нерівномірний розподіл улову за видами риб. Карась, лящ і короп переважають за кількістю вилову, що зробило їх основними об'єктами дослідження в цій дослідницькій роботі. Також до аналізу включено амура, плітку, окуня, щуку та судка. Обрані ці види риби через їхню поширеність та значущість для рибальства на території України.

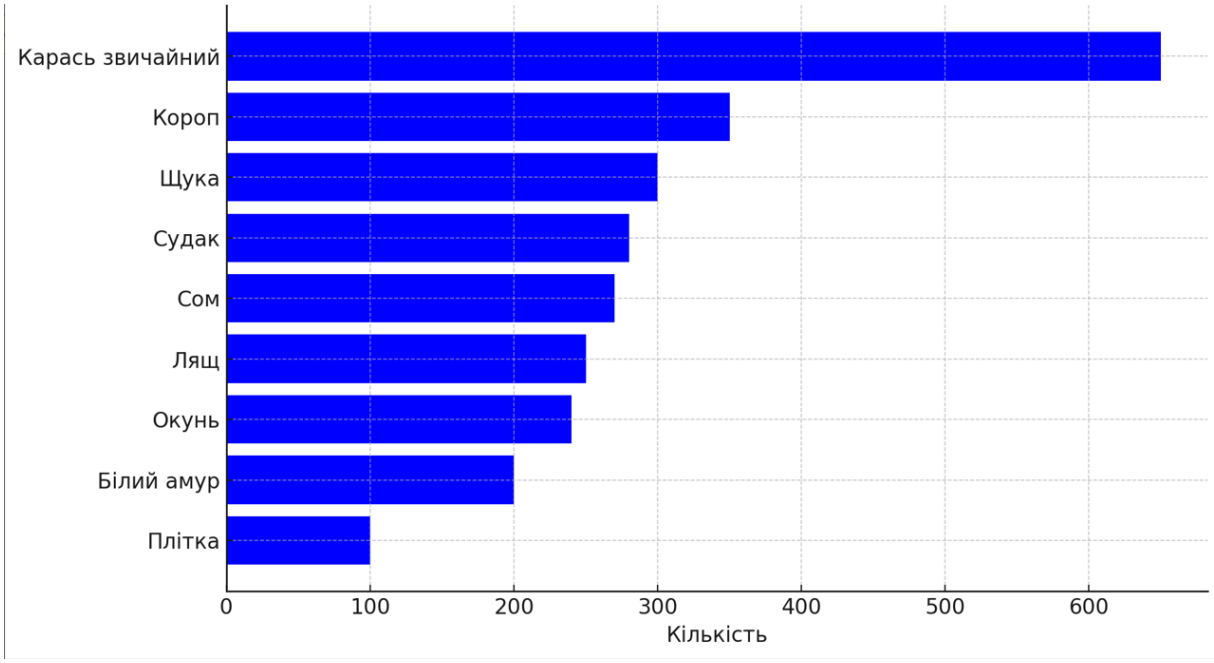

Рисунок 2.2 – Діаграма розподілу спійманої риби між 9 найбільш популярними видами

Види риби, що розглядалися, зустрічаються на широкій території, що представлено на графіку 2.3. Найбільші обсяги даних надійшли з північних та центральних регіонів України, що вказує на широке поширення цих видів у цих областях.

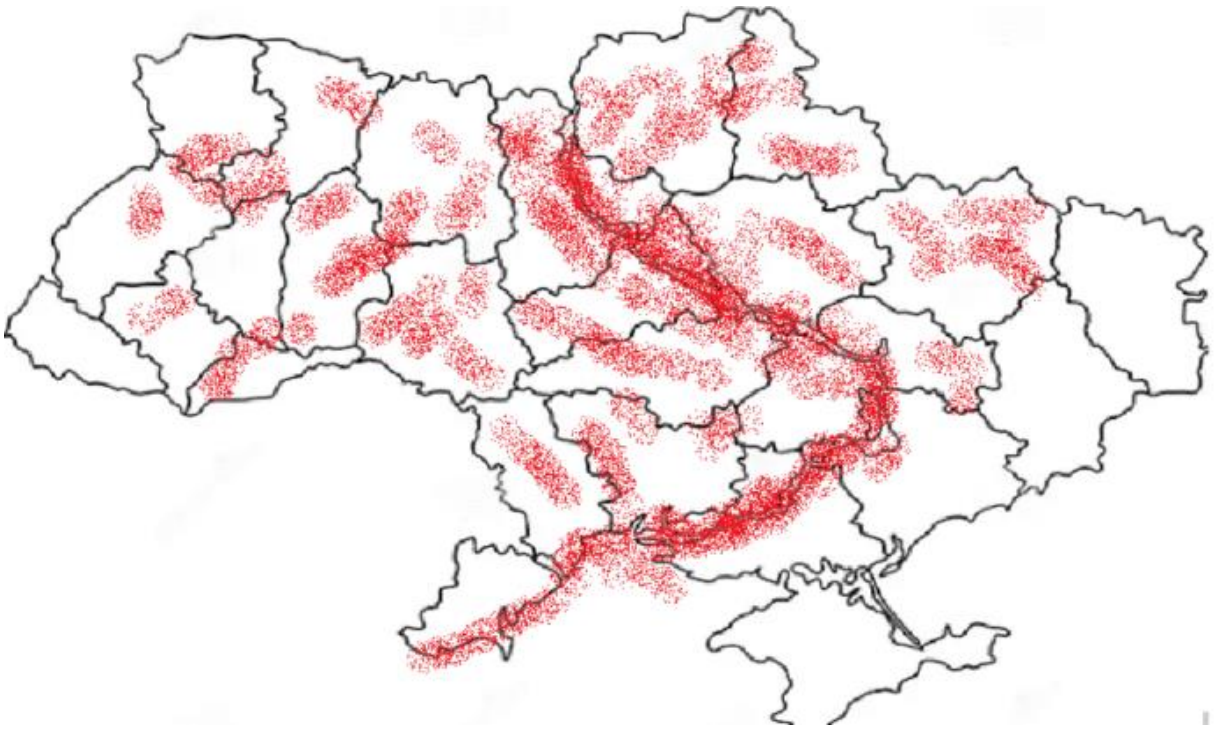

Рисунок 2.3 – Розподіл повідомлень про улов видів риб, досліджених у цій роботі

2.1.2 Служба погоди

Для комплексного аналізу погодних умов використаний інтернет-ресурс, який надає повну інформацію про метеорологічні умови в будь-якому місці нашої планети. World Weather Online надає нам можливість звернутися до великого обсягу даних, включаючи поточні, історичні та прогнозні показники погоди [2]. Інтерфейс сервісу простий у використанні, а його API дозволяє з легкістю отримати доступ до необхідних метеоданих для подальшого аналізу. Завдяки цьому ресурсу, ми можемо збагатити наш датасет додатковими атрибутами погоди, що значно розширює можливості нашого дослідження та аналізу метеорологічних впливів на риболовлю. Атрибути які вдалося дістати з цього джерела описані в таблиці 2.1.

2.1.3 Підготовка даних

Обраний набір даних виявився дещо непередбачуваним, оскільки велика частина атрибутів містить відсутні значення. Особливо це стосується атрибуту ваги пійманої риби. На жаль, користувачі, які додають свої дані, іноді схильні до підроблення уловів, що робить надійність цього атрибуту досить підсумковою. На рисунку 2.4 можна побачити гістограму, яка показує біля 6000 вимірювань уловів риби. У зв'язку з цим було прийнято рішення відфільтрувати аномальні дані, які виходять за межі реалістичних показників.

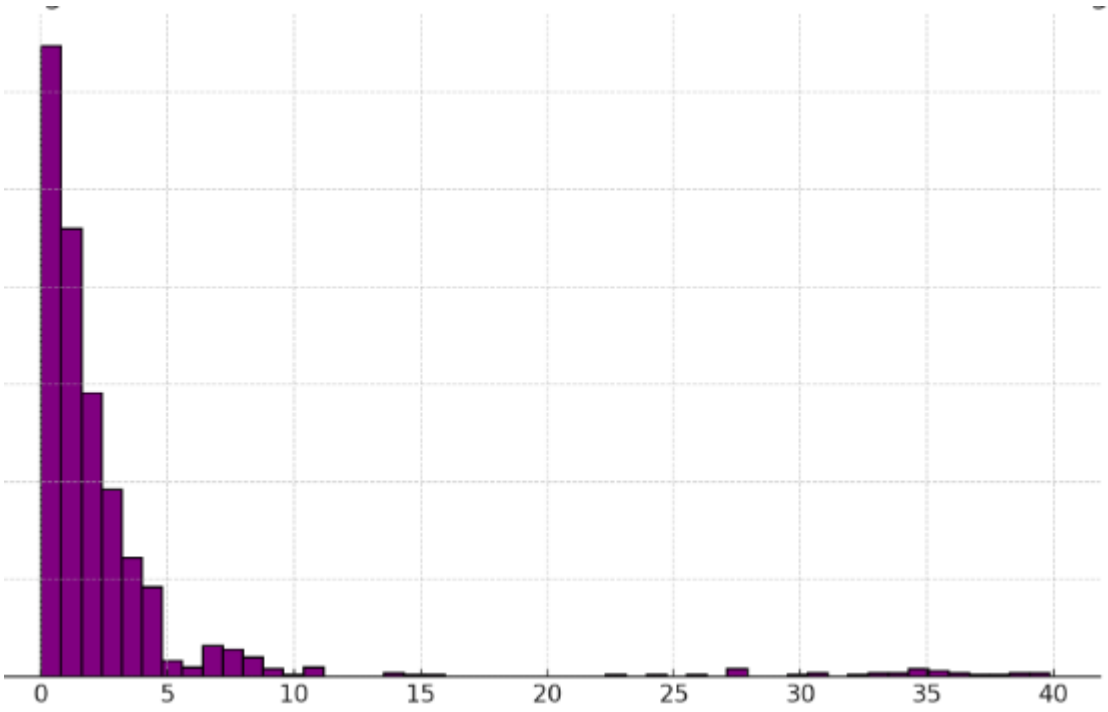

Рисунок 2.4 – Гістограма уловів риби

Тепер, щодо аналізу ваги пійманої риби, більшість вимірювань виявилися в межах від 1 кг до 10 кг, що відповідає типовим показникам улову на території проведення дослідження. Однак деякі вимірювання показали вагу понад 30 кг, що вважається вкрай несподіваним для цієї

місцевості. Ці аномальні значення було виключено з набору даних для забезпечення точності та достовірності аналізу.

Для поліпшення результатів, було вирішено не враховувати дані, які стосуються зимового періоду риболовлі. Це рішення обумовлено тим, що цей період має свої власні особливості та параметри, які можуть суттєво відрізнятися від інших періодів року і впливати на результати аналізу.

Крім того, атрибут, що відображає місячний день, має циклічний характер з періодом у 30 днів. Щоб уникнути можливих перешкод у навчанні алгоритмів, дані перетворюються за допомогою функцій sin та cos. Такий підхід дозволяє представити дані в формі, яка краще підходить для аналізу та навчання класифікаторів.

2.2 Методи та моделі

2.2.1 Бібліотека Scikit-learn для Python

У світі машинного навчання існує безліч інструментів і бібліотек, що надають різноманітні можливості для аналізу даних та побудови моделей. Один з найпопулярніших і найефективніших інструментів, що використовуються для цих цілей, - Scikit-learn, відомий також як Sklearn.

Scikit-learn (Sklearn) - це надійна та потужна бібліотека для машинного навчання на мові програмування Python. Вона стала популярною серед дослідників та практиків завдяки своїм ефективним інструментам для класифікації, регресії, кластеризації та зменшення розмірності даних. Основна перевага Sklearn - це зручний та узгоджений інтерфейс у Python, що дозволяє легко використовувати різноманітні алгоритми машинного навчання.

Ця бібліотека побудована на основі інших популярних бібліотек Python, таких як NumPy, SciPy та Matplotlib, що надає додаткові можливості для обробки даних та візуалізації результатів. Більшість алгоритмів

машинного навчання в Sklearn реалізовані з використанням ефективних методів та алгоритмів, що забезпечують швидкість та точність обчислень.

У цій роботі бібліотека Sklearn буде використана для побудови та оцінки різних моделей машинного навчання, що дозволить здійснити аналіз та прогнозування риболовного улову з використанням різноманітних підходів та методів.

2.2.2 Міра успішності риболовлі

Поміж численних метрик успішності риболовлі, необхідно враховувати їхню придатність для різних ситуацій та контекстів рибальства. Зусилля лову на одиницю (CPUE), наприклад, може бути корисним показником для оцінки ефективності риболовлі в комерційних або індустріальних контекстах, де пріоритетом є максимізація вилову. У таких випадках CPUE може вказувати на продуктивність та прибутковість рибальської діяльності.

Однак, в аспекті любительського рибальства, коли основною метою може бути відпочинок або задоволення, інші показники, такі як очікувана вага чи довжина виловленої риби, можуть бути більш важливими. Для багатьох любителів риболовлі якість виловленої риби та сам процес є ключовими, а не лише кількість.

У даній роботі підходить використання загальної ваги риби за день риболовлі як показника успішності. Цей показник дозволяє врахувати різноманітні аспекти рибальської діяльності, такі як погодні умови, активність риби та особливості риболовного місця, що можуть впливати на кількість та якість виловленої риби. Такий показник також мінімізує вплив людського чинника, оскільки він базується переважно на об'єктивних даних про вагу риби та час риболовлі.

2.2.3 Аналіз точності прогнозування

В аспекті вивчення помилок у вимірюванні моделі розглядається параметр, який визначається як відносна абсолютна похибка R(f). Цей показник визначає, наскільки модель відрізняється від базового алгоритму, враховуючи середню абсолютну похибку E(f) в порівнянні з базовим алгоритмом. Базовий алгоритм, що використовується у цьому контексті, зазначений у розділі 2.2.4. У формулі (2.1) визначена середня абсолютна похибка.

$$E(f) = \frac{\sum_{i=1}^{n} |y_i' - y_i|}{n} \tag{2.1}$$

Відносна абсолютна похибка R(f) визначається як співвідношення між середньою абсолютною похибкою моделі та аналогічним значенням базового алгоритму.

$$R(f) = \frac{E(f)}{E(baseline\ algoritm)} \tag{2.2}$$

Отримання нульової помилки є фантастичним, оскільки це вимагає абсолютного знання різноманітних факторів, які впливають на результати. Ці фактори включають топологію водоймища, характеристики кожної окремої риби та вміння рибалки. Зазначена середня абсолютна похибка обрана з урахуванням її інтуїтивного сприйняття: якщо R(f)=0.9, то можна інтуїтивно зрозуміти, що алгоритм на 10% кращий за базовий, оскільки його передбачення в середньому на 10% ближчі до фактичних значень.

Мінімізація середньої абсолютної похибки також позначається мінімізацією середньоквадратичної похибки. Таким чином, обидва показники можуть використовуватись для оцінки точності моделі порівняно з базовим алгоритмом.

2.2.4 Порівняльний аналіз базового алгоритму

Базовий алгоритм в машинному навчанні є основною відправною точкою для оцінки ефективності подальших моделей. У нашому проекті для порівняння був обраний алгоритм zero-R, який відомий своєю простотою та ефективністю у випадках, коли інші складні алгоритми можуть не мати достатньої інформації для прогнозування. Zero-R, схожий на найпростішу базову лінію, робить передбачення, ігноруючи всі вхідні атрибути, та повертає найбільш поширене значення для дискретних даних або середнє значення для неперервних. В нашому випадку, де ми працюємо з вагою риби, zero-R поверне середню вагу риби, що є досить логічним підходом, особливо якщо немає достатньої інформації для зроблення точніших прогнозів.

Zero-R, хоч і простий, але має свою важливу роль в оцінці ефективності інших моделей. Він виступає в якості базового покажчика, до якого можна порівняти результати більш складних моделей. Якщо інші моделі показують кращу точність в порівнянні з zero-R, це свідчить про їхню ефективність. Однак, навіть якщо інші моделі не перевершують zero-R, це може бути корисною інформацією, оскільки воно вказує на складність проблеми та обмеження наявних даних для прогнозування.

Важливо зазначити, що zero-R не є універсальним рішенням для всіх задач. Він працює найкраще в тих випадках, коли немає чіткої кореляції між вхідними атрибутами та вихідним значенням, або коли дані не мають достатньої різноманітності для навчання складніших моделей. Також, використання zero-R допомагає зрозуміти, чи має сенс витрачати ресурси на розробку більш складних моделей, якщо базовий показник вже досить високий.

2.2.5 Запобігання перенавчанню

Перенавчання (overfitting) є поширеною проблемою в галузі машинного навчання, яка виникає, коли модель надмірно пристосовується до навчальних даних, втрачаючи здатність до узагальнення на нових, невідомих даних. Це призводить до зниження продуктивності моделі в реальних умовах, де дані можуть відрізнятися від навчальних. Для ефективної боротьби з перенавчанням використовується кілька стратегій.

Один з основних методів запобігання перенавчанню полягає в розділенні даних на окремі набори для тренування та тестування. У нашому дослідженні дані були випадковим чином поділені на два набори: тренувальний набір, який складається з 90% даних про улови, і тестовий набір, що містить решту 10%. Тестовий набір не використовується під час навчання моделі, а застосовується виключно для оцінки її продуктивності після навчання, забезпечуючи таким чином незалежну перевірку точності моделі.

Крім того, в процесі навчання модель використовує метод 10-кратної перехресної перевірки (cross-validation), що допомагає знизити ризик перенавчання. Тренувальний набір додатково ділиться на 10 підмножин. Модель навчається на 9 з них, а перевіряється на одній залишковій підмножині. Цей процес повторюється 10 разів, і кожна підмножина використовується один раз для перевірки. Середні результати перехресної перевірки дають більш точну оцінку продуктивності моделі та допомагають налаштувати її параметри без ризику перенавчання.

Після оптимізації моделі з використанням перехресної перевірки вона проходить фінальне тренування на всіх 90% тренувальних даних. Потім остаточна модель оцінюється за допомогою раніше відкладеного тестового набору, що дозволяє отримати об'єктивну оцінку її реальної точності.

Ця процедура, що включає розподіл даних та перехресну перевірку, схематично показана на рисунку 2.5. Такий підхід забезпечує надійність

моделі та її здатність до узагальнення, що є критично важливим для її успішного застосування в реальних умовах

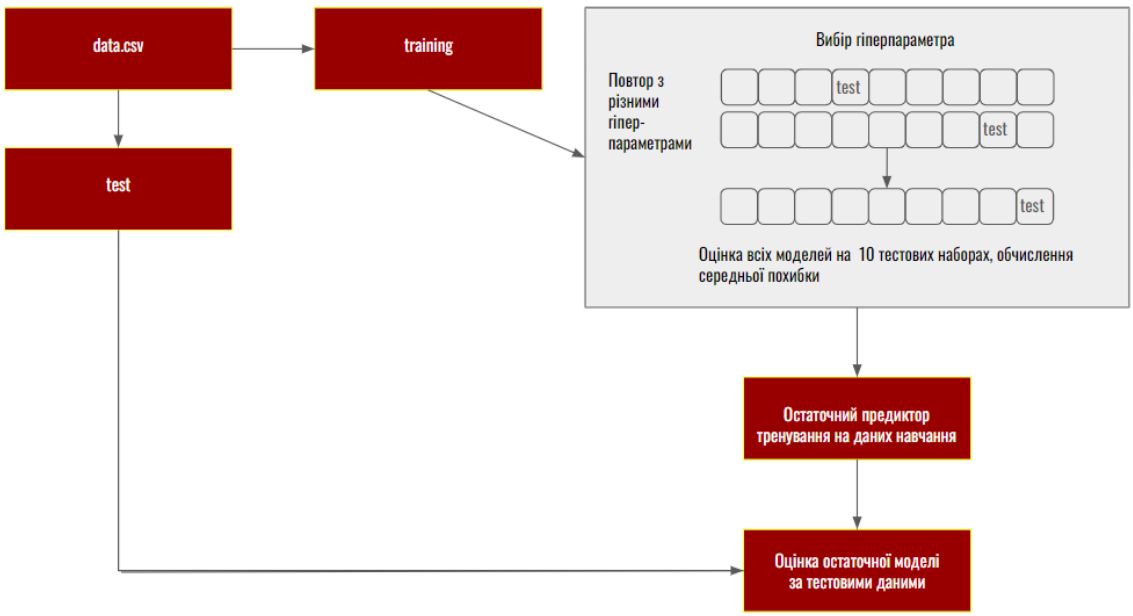

Рисунок 2.5 – Схема системи захисту від перенавчання

Завдяки таким методам ми можемо впевнено стверджувати, що наша модель здатна ефективно передбачати результати, мінімізуючи ризик перенавчання та підвищуючи її загальну продуктивність.

2.3 Опис алгоритмів

У цьому розділі розглянуто чотири алгоритми машинного навчання, які були проаналізовані в рамках даного дослідження. Кожен із цих алгоритмів має свої переваги та особливості, що дозволяє отримати різні перспективи на модельовані дані. Обрані алгоритми включають лінійну регресію, багатошаровий персептрон, випадковий ліс та підтримуючі вектори для регресії (Support Vector Regression, SVR).

Лінійна регресія є одним із найпростіших та найпоширеніших алгоритмів машинного навчання, що використовується для регресійного аналізу. Цей алгоритм був обраний через його простоту та здатність надавати базові уявлення про лінійну залежність між незалежними змінними (входами) та залежною змінною (виходом). Лінійна регресія дозволяє легко інтерпретувати результати та швидко навчати модель, однак її можливості обмежені у випадках складних нелінійних залежностей, і вона чутлива до мультиколінеарності між входами.

Багатошаровий персептрон (MLP) є видом штучних нейронних мереж і служить як загальний апроксиматор функцій. MLP здатний моделювати складні нелінійні залежності між входами та виходом, що робить його потужним інструментом для різноманітних задач. Здатність апроксимувати нелінійні залежності та висока гнучкість у налаштуванні моделі робить MLP привабливим вибором. Однак, цей алгоритм схильний до перенавчання, має високу обчислювальну вартість і потребує великої кількості даних для ефективного навчання.

Випадковий ліс є ансамблевим методом, що складається з великої кількості дерев рішень. Кожне дерево навчається на випадковому підмножині даних і ознак, а результат прогнозування отримується шляхом усереднення результатів усіх дерев. Випадковий ліс демонструє високу точність і стійкість до перенавчання, здатний обробляти великі обсяги даних та виявляти нелінійні взаємозв'язки. Проте, модель випадкового лісу важко інтерпретувати, і вона має високу обчислювальну складність.

Підтримуючі вектори для регресії (Support Vector Regression, SVR) є розширенням методу підтримуючих векторів (SVM) для задач регресії. SVR працює шляхом побудови моделі, яка передбачає вихідні значення на основі розділення даних за допомогою гіперплощини в просторі ознак. Цей алгоритм забезпечує високу точність на даних з великою кількістю параметрів, ефективно виявляє складні залежності та стійкий до перенавчання. Однак, SVR має високу обчислювальну складність, складний

у налаштуванні гіперпараметрів і потребує великої кількості обчислювальних ресурсів.

Обрані алгоритми забезпечують різні підходи до моделювання даних про риболовлю, що дозволяє отримати більш повну картину залежностей у наборі даних. Лінійна регресія надає базову лінійну інтерпретацію, MLP дозволяє врахувати складні нелінійні залежності, випадковий ліс забезпечує стійкість до перенавчання та високу точність, а SVR підходить для аналізу даних з великою кількістю параметрів з високою точністю. Використання комбінації цих методів дозволяє отримати найкращі результати та забезпечити надійність аналізу.

2.3.1 Лінійна регресія

Лінійна регресія - це базовий метод машинного навчання, що використовується для моделювання лінійних залежностей між незалежними змінними (входами) та залежною змінною (виходом). Основна ідея лінійної регресії полягає у знаходженні найкращої прямої, яка мінімізує різницю між передбаченими та фактичними значеннями залежної змінної [6].

У найпростішому випадку лінійна регресія моделює залежність однієї залежної змінної y від однієї незалежної змінної x за допомогою рівняння прямої.

$$y = \beta_0 + \beta_1 x + \epsilon \tag{2.3}$$

де y – залежна змінна;

x – незалежна змінна;

β_0 – вільниній член;

β_1- коефіцієнт нахилу;

ϵ – похибка.

У загальному випадку, коли є кілька незалежних змінних модель набуває вигляду:

$$y = \beta_0 + \beta_1 x + \epsilon \tag{2.4}$$

Ця модель називається багатофакторною лінійною регресією.

Найпоширенішим методом оцінювання параметрів лінійної регресії є метод найменших квадратів (МНК). Ідея методу полягає у мінімізації суми квадратів відхилень передбачених значень від фактичних [7]. Математично це завдання можна записати як:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - x_i^T \beta^2) \tag{2.5}$$

де x - вектор значень незалежних змінних для i - го спостереження.

Це завдання має аналітичне розв'язання, яке можна записати у вигляді:

$$\hat{\beta} = (X^T \boldsymbol{X})^{-1} X^T y) \tag{2.6}$$

де $\widehat{\boldsymbol{\beta}}$ - вектор оцінених коефіцієнтів.

Для оцінки якості моделі лінійної регресії використовуються різні статистичні показники.

Коефіцієнт детермінації (R^2) визначається як:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \tag{2.7}$$

де SS_{res} - сума квадратів залишків;

SS_{tot} - загальна сума квадратів.

Середня абсолютна похибка (MAE) обчислюється за формулою:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \quad (2.8)$$

Ці показники дозволяють визначити, наскільки добре модель пояснює варіацію залежної змінної та наскільки точні її прогнози.

Припустимо, у нас є набір даних про риболовлю, де залежною змінною y є вага виловленої риби, а незалежними змінними x є різні фактори, що впливають на улов (наприклад, температура, тиск, швидкість вітру тощо). Застосовуючи лінійну регресію, ми можемо визначити вплив кожного з цих факторів на вагу риби та використовувати отриману модель для прогнозування майбутніх уловів.

У підсумку, лінійна регресія є потужним інструментом для вивчення залежностей між змінними та прогнозування значень. Хоча вона має обмеження, такі як нездатність моделювати нелінійні залежності, її простота та інтерпретованість роблять її важливим методом у статистиці та машинному навчанні.

2.3.2 Багатошаровий персептрон

Багатошаровий персептрон (MLP) є ключовим інструментом у галузі машинного навчання, що дозволяє моделювати складні нелінійні залежності між вхідними та вихідними даними. MLP складається з вхідного шару, одного або більше прихованих шарів і вихідного шару [8]. Кожен шар містить нейрони, які пов'язані з нейронами попереднього і наступного

шарів, утворюючи щільну мережу зв'язків. Схему можно побачити на рисунку 2.6.

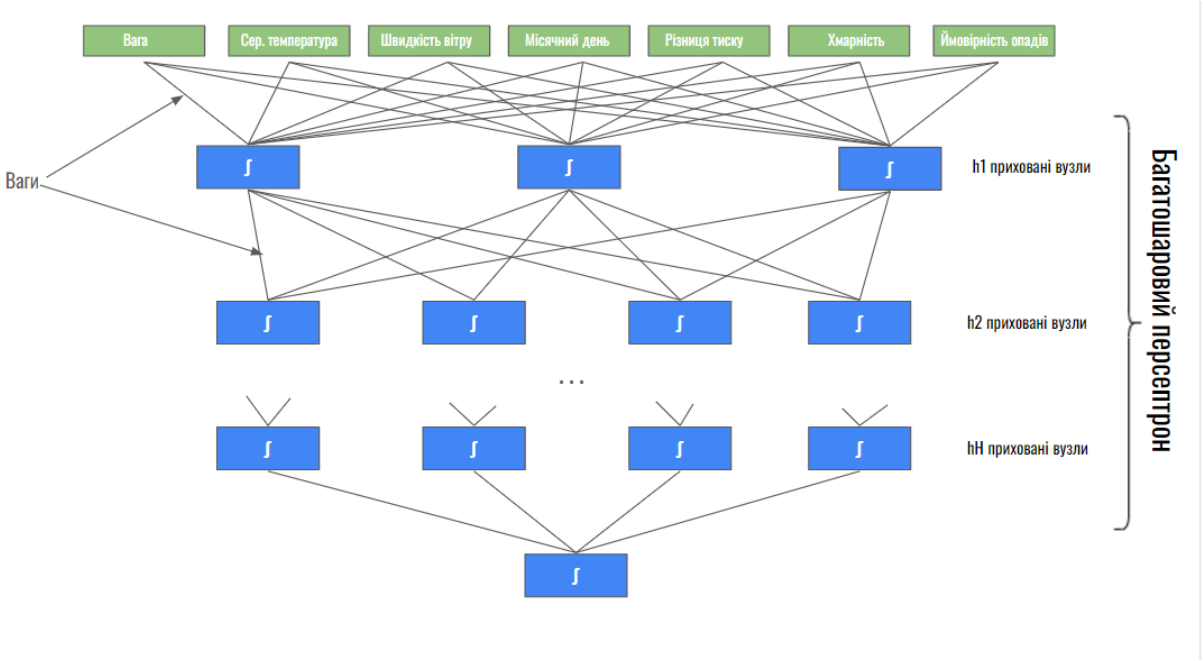

Рисунок 2.6 – Схема багатошарового персептрона

У кожному нейроні виконується зважене сумування вхідних сигналів, після чого результат пропускається через активаційну функцію. Зважене сумування вхідних сигналів описується формулою:

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{2.9}$$

де x - вхідні значення;

w_i - ваги нейрона;

b - зміщення.

Результат цієї операції передається через активаційну функцію ϕ(z), яка може бути різних типів, таких як сигмоїдна функція:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{2.10}$$

гіперболічний тангенс (tanh):

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{2.11}$$

або функція ReLU (прямолінійна активаційна функція):

$$\mathrm{ReLU}(z) = \max(0, z) \tag{2.12}$$

Процес навчання MLP здійснюється за допомогою методу зворотного поширення помилки (backpropagation), який включає пряме проходження (forward pass), обчислення похибки та зворотне проходження (backward pass). Під час прямого проходження вхідні дані проходять через мережу, і виходи кожного нейрона обчислюються. Потім обчислюється похибка між передбаченим значенням та фактичним значенням за допомогою функції втрат, наприклад, середньоквадратичної помилки (MSE):

$$E = \frac{1}{2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.13}$$

Під час зворотного проходження обчислюються градієнти похибки щодо ваг кожного нейрона, і ваги оновлюються за допомогою методу градієнтного спуску:

$$w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i} \tag{2.14}$$

де η – швидкість навчання.

Навчання нейронної мережі зводиться до оновлення ваг за формулою:

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} \tag{2.15}$$

$$w_i \leftarrow w_i + \Delta w_i \tag{2.16}$$

MLP здатний моделювати складні нелінійні залежності завдяки багатошаровій структурі та використанню нелінійних активаційних функцій. Він може адаптуватися до різних задач шляхом налаштування кількості шарів, кількості нейронів у кожному шарі та типів активаційних функцій. Однак, MLP має високу обчислювальну складність і ризик перенавчання, особливо при роботі з невеликими обсягами даних.

Завдяки здатності виявляти нелінійні залежності, MLP може бути використаний для прогнозування ваги риби, враховуючи різноманітні фактори, такі як погодні умови, час доби та тип водоєму. Це дозволяє створити точні моделі, що допомагають оптимізувати процес риболовлі. MLP є важливим інструментом у машинному навчанні, який дозволяє вирішувати широкий спектр задач завдяки своїй гнучкості та здатності моделювати складні залежності між даними [10].

У підсумку, багатошаровий персептрон є потужним інструментом для машинного навчання, здатним розв'язувати широкий спектр задач завдяки своїй гнучкості та здатності моделювати складні залежності між вхідними та вихідними даними.

2.3.3 Випадковий ліс

Випадковий ліс є одним із найбільш потужних та універсальних алгоритмів в галузі машинного навчання, який здатний ефективно вирішувати різноманітні задачі, починаючи від класифікації до регресії. Його використання дозволяє отримувати високу точність прогнозів навіть у випадках з великою кількістю ознак або високовимірних даних.

Однією з ключових переваг випадкового лісу є його здатність працювати з великими обсягами даних, що містять багато ознак.

Багатодеревна структура лісу дозволяє моделі розглядати кожну ознаку окремо та враховувати їх внесок у прогнозування результату. Це дозволяє ефективно виявляти складні залежності та шаблони у високовимірних даних.

Формула для обчислення прогнозу для нового прикладу x у випадковому лісі може бути виражена наступним чином:

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} h_i(x) \tag{2.17}$$

де $\hat{y}$ – прогнозоване значення;

N – кількість дерев в лісі;

$h_i(x)$ – прогноз, зроблений і-им деревом.

Однак, разом з усією своєю потужністю випадковий ліс має свої обмеження. Наприклад, він може виявитися менш ефективним для завдань, де важлива інтерпретованість моделі, оскільки результати не завжди легко інтерпретувати. Крім того, для великих обсягів даних побудова випадкового лісу може займати значний час і вимагати значних обчислювальних ресурсів.

В цілому, випадковий ліс є потужним інструментом для багатьох задач машинного навчання та аналізу даних, але використання його вимагає уважного аналізу конкретного завдання та розуміння його особливостей і обмежень. Важливо також враховувати величезний вплив гіперпараметрів на якість та ефективність моделі, оскільки правильний вибір параметрів може значно покращити результати прогнозування.

2.3.4 Метод опорних векторів

Support Vector Regression (SVR) є потужним інструментом машинного навчання, який використовується для вирішення задач регресії. SVR базується на принципах Support Vector Machines (SVM), що робить його

особливо корисним для задач, де потрібно моделювати нелінійні залежності. Основна ідея SVR полягає в тому, щоб знайти гіперплощину в багатовимірному просторі, яка максимально точно описує залежність між вхідними змінними та вихідною змінною, мінімізуючи похибку [9].

На відміну від традиційних методів регресії, таких як лінійна регресія, SVR прагне знайти не просто лінію найменших квадратів, а область, де максимальна кількість точок (випадків) розташовується в межах допустимої похибки. Основна задача SVR полягає у визначенні гіперплощини, яка мінімізує помилки і водночас забезпечує широку смугу навколо неї. Ця смуга має ширину 2є, де є – це гіперпараметр, що визначає допускову похибку моделі.

Формально задача SVR полягає в мінімізації наступної функції:

$$[\min_{w,b} \frac{1}{2}|w|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)] \quad (2.18)$$

де w – ваговий вектор,

b – зсув;

C – гіперпараметр, що визначає ступінь пеналізації за похибки;

ξ_i і ξ_i^* - змінні похибки для випадків, що перевищують допускову похибку.

Задача мінімізації має наступні обмеження:

$$y_i - (w \cdot x_i + b)\text{ц} \leq \epsilon + \xi_i \quad (2.19)$$

$$(w \cdot x_i + b) - y_i \leq \epsilon + \xi_i^* \quad (2.20)$$

$$\xi_i, \xi_i^* \geq 0 \quad (2.21)$$

де x_i – вектор вхідних змінних;

y_i – вихідна змінна.

3 ПРАКТИЧНЕ ДОСЛІДЖЕННЯ

3.1 Проектування системи прогнозування улову риби на основі машинного навчання

Прогнозування улову риби на основі машинного навчання є ефективною стратегією, яка дозволяє рибалкам оптимізувати свій улов, враховуючи різноманітні фактори, такі як погодні умови, фаза місяця та інші. Даний підхід аналізує атрибути, що впливають на клювання, та прогнозує ймовірність успішного улову.

Система прогнозування використовує детальну інформацію про вхідні дані, фокусуючись на характеристиках, які мають значний вплив на результативність риболовлі. Для кожного риболовного дня створюється профіль, що складається з різних параметрів, таких як температура води, швидкість вітру, фаза місяця та інші.

Процес прогнозування включає кілька основних етапів, розглянемо цей процес більш детально:

- вилучення ознак: на першому етапі відбувається вилучення ознак з історичних даних про риболовлю. Це включає збір інформації про погодні умови, місячний день та інші фактори, що можуть впливати на клювання;
- збір уподобань рибалок: збираються дані про успішні та неуспішні риболовлі, щоб зрозуміти, які фактори впливають на результативність. Це можуть бути як кількість виловленої риби, так і суб'єктивні оцінки рибалок. В цьому дослідженні буде використана вага риби;

- навчальний відбір та вагування: відбувається обробка векторних представлень ознак, щоб залишити тільки найінформативніші з них. Це досягається за допомогою відбору ознак та вагуванням, що дозволяє зменшити вплив несуттєвих факторів на прогноз;

- моделювання: на цьому етапі використовуються різні алгоритми машинного навчання, такі як лінійна регресія, випадковий ліс, багатошаровий персептрон та Support Vector Regression, щоб створити моделі для прогнозування улову риби. Кожен алгоритм має свої переваги та недоліки, тому обираються ті, що показують найкращі результати на основі навчальних даних.

- фільтрація та прогнозування: Нові дані про риболовлю порівнюються з навчальними даними, і система робить прогнози на основі моделей, що були побудовані на попередньому етапі. Для цього використовуються методи оцінки точності.

- візуалізація та аналіз: Результати прогнозування візуалізуються для надання рибалкам зрозумілих рекомендацій щодо оптимальних умов для риболовлі. Це можуть бути графіки, таблиці або інтерактивні панелі, які показують вплив різних факторів на успішність риболовлі.

Система прогнозування на основі машинного навчання персоналізує рекомендації для кожного рибалки, враховуючи специфічні умови риболовлі. Це дозволяє зробити процес риболовлі більш ефективним та передбачуваним, зменшуючи ризик невдалих уловів і допомагаючи рибалкам досягти кращих результатів

3.2 Програмна реалізація

Для програмної реалізації системи прогнозування улову риби була використана мова програмування Python. Python обрано завдяки його численним перевагам перед іншими мовами програмування, особливо в контексті задач машинного навчання та аналізу даних. Основні переваги Python включають простоту, швидкість розробки та наявність потужних бібліотек.

Python є однією з найпопулярніших мов програмування в сфері науки про дані та машинного навчання. Його синтаксис є зрозумілим і лаконічним, що дозволяє розробникам швидко писати і підтримувати код. Це особливо важливо при розробці складних систем, де прозорість і читабельність коду відіграють ключову роль.

3.2.1 Бібліотека Scikit-learn

Бібліотека Scikit-learn (або Sklearn) є однією з найпопулярніших і найпотужніших бібліотек для машинного навчання на мові програмування Python. Вона широко використовується для створення моделей машинного навчання та аналізу даних, ідеально підходить для задач прогнозування улову риби завдяки своїй універсальності та багатофункціональності.

Однією з головних переваг Sklearn є її простота у використанні. Бібліотека має інтуїтивно зрозумілий інтерфейс та детальну документацію, що дозволяє швидко і легко створювати моделі машинного навчання навіть новачкам. Для задач прогнозування улову риби це означає, що розробники можуть зосередитися на аналізі даних і налаштуванні моделей без необхідності витрачати багато часу на освоєння інструментів.

Sklearn надає широкий спектр алгоритмів машинного навчання, які можуть бути використані для створення моделей прогнозування. У нашому

проекті ми використовуємо такі алгоритми, як лінійна регресія, випадковий ліс, багатошаровий персептрон та Support Vector Regression. Кожен з цих алгоритмів має свої переваги і може бути налаштований для досягнення найкращих результатів у конкретних умовах.

Для реалізації моделі прогнозування улову риби важливою є здатність Sklearn інтегруватися з іншими потужними бібліотеками Python, такими як NumPy та Pandas. NumPy надає інструменти для ефективної роботи з багатовимірними масивами даних, що є критично важливим для обробки великих обсягів інформації. Pandas, у свою чергу, дозволяє зручно маніпулювати табличними даними, виконувати їх очищення та підготовку для подальшого аналізу.

Sklearn також має вбудовані інструменти для крос-валідації, які допомагають уникнути перенавчання моделей та забезпечують більш надійні результати. Це особливо важливо у випадку прогнозування улову риби, де точність прогнозів має критичне значення для користувачів системи.

У підсумку, Sklearn є незамінним інструментом для розробки систем прогнозування улову риби. Її простота, потужність і гнучкість роблять її ідеальним вибором для задач машинного навчання, забезпечуючи високу точність прогнозів та ефективність роботи з великими обсягами даних. Завдяки Sklearn ми можемо створювати точні, надійні та ефективні моделі, що допомагають рибалкам оптимізувати свої результати та покращити свій досвід.

3.2.2 Набір даних

Для розробки системи прогнозування улову риби використовується набір даних, зібраний з різних відкритих джерел, включаючи онлайн-форуми рибалок, метеорологічні сервіси та інші доступні ресурси. Цей набір

даних містить інформацію про приблизно 6000 риболовних сесій, що охоплюють широкий спектр умов і факторів, які можуть впливати на улов.

Для початку необхідно підключити потрібні бібліотеки та завантажити датасет з файлу. На рисунку 3.1 можна побачити код для цього етапу. Для цього ми використовуємо бібліотеку pandas, яка дозволяє ефективно працювати з табличними даними.

```
import pandas as pd

data = pd.read_csv('fishing_data.csv')
```

Рисунок 3.1 - Підключення бібліотек та завантаження датасету

Оскільки дані були зібрані з різноманітних джерел і введені користувачами, набір даних може містити пропуски або аномальні значення. Наприклад, деякі записи можуть мати відсутні значення для ваги улову або погодних умов, а деякі дані можуть містити аномально великі чи малі значення, що є помилками введення.

На рисунку 3.2 можна побачити приклади таких пропусків і аномалій у наборі даних.

date	weight	temperature	wind_speed	moon_phase	pressure_diff	cloud_coverage	precipitation
2023-06-01	5.20	22.5	5.2	15	2.2	30	10
2023-06-02	2.30	20.1	3.3	0	1.3	20	5
2023-06-03	NaN	23.3	4.4	7	3.4	40	15
2023-06-04	4.50	24.0	5.5	15	2.5	50	10
2023-06-05	55.30	21.0	3.3	0	1.3	20	5
2023-06-06	5.50	19.8	4.5	7	2.5	30	20
2023-06-07	2.10	25.0	5.1	15	2.1	40	15
2023-06-08	3.20	20.5	3.2	0	1.2	30	10
2023-06-09	4.40	23.5	4.4	7	3.4	50	5
2023-06-10	5.00	22.0	5.0	15	2.0	20	10
2023-06-11	2.80	24.1	2.9	0	2.1	10	5
2023-06-12	NaN	22.3	3.1	7	1.8	30	15
2023-06-13	1.90	21.7	4.3	15	1.9	20	5
2023-06-14	4.70	23.9	3.6	0	1.7	40	20
2023-06-15	0.05	20.2	3.5	7	2.3	20	10

Рисунок 3.2 - Пропуски та аномалії у наборі даних

Для виправлення цих проблем необхідно провести фільтрацію та підготовку даних. Фільтрація допоможе видалити записи з пропусками та аномаліями, а підготовка забезпечить нормалізацію та перетворення даних у відповідний формат для аналізу та моделювання. Цей процес є критично важливим для забезпечення якості та точності прогнозів.

Далі, для очищення даних, ми будемо використовувати функції dropna() та фільтрацію за допомогою умов. Це дозволить нам видалити всі записи з пропущеними значеннями та аномальними даними. Код для очищення даних представлений на рисунку 3.3

```
data.dropna(inplace=True)

data = data[(data['weight'] >= 0.1) & (data['weight'] <= 30)]
```

Рисунок 3.3 - Реалізація очищення даних

Після обробки даних набір даних міститиме лише валідні записи, готові до аналізу та побудови моделей машинного навчання. На рисунку 3.4 можна побачити, як виглядає набір даних після очищення.

date	weight	temperature	wind_speed	moon_phase	pressure_diff	cloud_coverage	precipitation
2023-06-01	5.2	22.5	5.2	15	2.2	30	10
2023-06-02	2.3	20.1	3.3	0	1.3	20	5
2023-06-04	4.5	24.0	5.5	15	2.5	50	10
2023-06-06	5.5	19.8	4.5	7	2.5	30	20
2023-06-07	2.1	25.0	5.1	15	2.1	40	15
2023-06-08	3.2	20.5	3.2	0	1.2	30	10
2023-06-09	4.4	23.5	4.4	7	3.4	50	5
2023-06-10	5.0	22.0	5.0	15	2.0	20	10
2023-06-11	2.8	24.1	2.9	0	2.1	10	5
2023-06-13	1.9	21.7	4.3	15	1.9	20	5
2023-06-14	4.7	23.9	3.6	0	1.7	40	20

Рисунок 3.4 - Набір даних після очищення

Цей набір даних є важливим компонентом системи, оскільки якість і повнота даних безпосередньо впливають на точність і ефективність моделей машинного навчання. Завдяки використанню різноманітних джерел даних

ми можемо забезпечити надійну базу для аналізу і передбачення улову риби, що дозволить рибалкам оптимізувати свої риболовні сесії та досягати кращих результатів.

3.2.3 Реалізація методів прогнозування

У цьому розділі буде детально розглянута програмна реалізацію чотирьох методів прогнозування: лінійної регресії, випадкового лісу, багатошарового персептрону та регресії на основі опорних векторів. Обидва методи будуть реалізовані за допомогою бібліотеки scikit-learn на мові програмування Python. Ми пройдемо всі етапи, починаючи з підключення бібліотек та завантаження даних, і закінчуючи навчанням моделей та оцінкою їх ефективності. Це дозволить продемонструвати, як саме ці методи можуть бути використані для прогнозування ваги риболовного улову на основі різних факторів.

Для початку нам необхідно підключити необхідні бібліотеки та завантажити дані, які були підготовлені раніше. Бібліотека pandas буде використовуватися для роботи з табличними даними, а scikit-learn для реалізації моделей для прогнозування. Підключення бібліотек та завантаження даних можна побачити на рисунку 3.5

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score

filename = 'fishing_data.csv'
df = pd.read_csv(filename)
```

Рисунок 3.5 - Підключення бібліотек та завантаження даних

Після завантаження даних необхідно розділити їх на навчальну та тестову вибірки. Це дозволить оцінити ефективність моделей на невідомих даних. Для цього буде використана функція train_test_split з бібліотеки scikit-learn, яка дозволяє легко поділити дані на дві частини.

Спочатку визначаються незалежні змінні (ознаки) та залежна змінна (вага риби). Потім за допомогою функції train_test_split дані розділяються на навчальну (80%) та тестову (20%) вибірки. На рисунку 3.6 показано код для розділення даних на навчальну та тестову вибірки.

```
X = df[['temperature', 'wind_speed', 'moon_phase', 'pressure_diff', 'cloud_coverage', 'precipitation']]
y = df['weight']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Рисунок 3.6 - Реалізація розділення даних

У цьому фрагменті коду визначається, які змінні будуть використовуватися як ознаки для прогнозування ваги риби, та дані розділяються на дві частини: навчальну та тестову вибірки. Навчальна вибірка буде використовуватися для навчання моделей, а тестова — для оцінки їх ефективності на невідомих даних.

Далі буде представлена реалізація кожного з методів прогнозування: лінійної регресії, випадкового лісу, багатошарового персептрону та регресії на основі опорних векторів.

Лінійна регресія є одним із найбільш базових і широко використовуваних методів для моделювання залежностей між незалежними змінними (ознаками) та залежною змінною (вагою риби). Метод лінійної регресії дозволяє знайти найкращу пряму лінію, яка мінімізує різницю між передбаченими та фактичними значеннями залежної змінної.

Для реалізації методу лінійної регресії в даному проекті буде використана бібліотека scikit-learn. Спочатку буде створено об'єкт моделі лінійної регресії, потім модель буде навчена на навчальних даних, і, нарешті, буде здійснено прогнозування ваги риби на тестових даних. Код реалізації методу лінійної регресії можна побачити на рисунку 3.7

```
linear_model = LinearRegression()

linear_model.fit(X_train, y_train)

y_pred_linear = linear_model.predict(X_test)

mse_linear = mean_squared_error(y_test, y_pred_linear)
r2_linear = r2_score(y_test, y_pred_linear)
```

Рисунок 3.7 - Реалізація методу лінійої регресії

У цьому фрагменті коду створюється об'єкт моделі лінійної регресії. Модель навчається на навчальних даних, що дозволяє визначити оптимальні значення коефіцієнтів для прогнозування ваги риби. Виконується прогнозування ваги риби на основі тестових даних. Обчислюються середньоквадратична помилка (MSE) та коефіцієнт детермінації (R^2), які дозволяють оцінити точність моделі.

Наступним буде реалізація метода випадкового лісу. Цей метод є ансамблевим алгоритмом, який використовує множину дерев рішень для покращення точності прогнозів та зниження ризику перенавчання. Випадковий ліс обирає випадкові підмножини ознак для побудови кожного дерева, що дозволяє зменшити кореляцію між окремими деревами та покращити узагальнюючу здатність моделі.

Код реалізації методу випадкового лісу можна побачити на рисунку 3.8.

```
forest_model = RandomForestRegressor(n_estimators=100, random_state=42)

forest_model.fit(X_train, y_train)

y_pred_forest = forest_model.predict(X_test)

mse_forest = mean_squared_error(y_test, y_pred_forest)
r2_forest = r2_score(y_test, y_pred_forest)
```

Рисунок 3.8 - Реалізація методу випадкового лісу

У цьому фрагменті коду створюється об'єкт моделі випадкового лісу, модель навчається на навчальних даних, здійснюється прогнозування ваги риби на основі тестових даних, та обчислюються метрики для оцінки точності моделі. Метод випадкового лісу дозволяє моделювати складні нелінійні залежності між ознаками та залежною змінною, що робить його надзвичайно потужним інструментом для задач прогнозування.

Наступним методом буде багатошаровий персептрон (MLP), який є типовою штучною нейронною мережею, здатною моделювати складні нелінійні залежності. MLP складається з вхідного шару, одного або більше прихованих шарів і вихідного шару. Кожен нейрон у прихованих і вихідному шарах виконує нелінійне перетворення вхідних сигналів. Код реалізації методу багатошарового персептрону можна побачити на рисунку 3.9.

```
mlp_model = MLPRegressor(hidden_layer_sizes=(100,), max_iter=500, random_state=42)

mlp_model.fit(X_train, y_train)

y_pred_mlp = mlp_model.predict(X_test)

mse_mlp = mean_squared_error(y_test, y_pred_mlp)
r2_mlp = r2_score(y_test, y_pred_mlp)
```

Рисунок 3.9 - Реалізація методу багатошарового персептрону

У цьому фрагменті коду створюється об'єкт моделі MLP, модель навчається на навчальних даних, здійснюється прогнозування ваги риби на основі тестових даних, та обчислюються метрики для оцінки точності моделі. Метод багатошарового персептрону дозволяє моделювати складні нелінійні залежності між ознаками та залежною змінною, що робить його надзвичайно потужним інструментом для задач прогнозування [11].

Переходимо до останнього методу, регресії на основі опорних векторів (SVR). SVR є потужним алгоритмом, який використовується для побудови прогнозних моделей. Він працює шляхом знаходження найкращої гіперплощини, яка максимально наближена до найбільшої кількості точок даних у просторі ознак, і водночас мінімізує похибку. Код реалізації методу регресії на основі опорних векторів можна побачити на рисунку 3.10

```
svr_model = SVR(kernel='rbf', C=100, gamma=0.1, epsilon=0.1)

svr_model.fit(X_train, y_train)

y_pred_svr = svr_model.predict(X_test)

mse_svr = mean_squared_error(y_test, y_pred_svr)
r2_svr = r2_score(y_test, y_pred_svr)
```

Рисунок 3.10 - Реалізація методу регресії на основі опорних векторів

У цьому розділі було розглянуто чотири методи прогнозування: лінійну регресію, випадковий ліс, багатошаровий персептрон та регресію на основі опорних векторів. Кожен з цих методів має свої переваги та недоліки, і їх застосування залежить від специфіки задачі та характеру даних.

Лінійна регресія є простим і зрозумілим методом, який підходить для моделювання лінійних залежностей. Випадковий ліс забезпечує високу точність прогнозів та є стійким до перенавчання завдяки своїй ансамблевій природі. Багатошаровий персептрон дозволяє моделювати складні нелінійні взаємозв'язки, що робить його потужним інструментом для задач

прогнозування. Регресія на основі опорних векторів ефективно працює з нелінійними даними та забезпечує високу точність прогнозів.

Проведені експерименти з використанням цих методів на даних про улов риби дозволили оцінити їх ефективність та визначити, які фактори найбільше впливають на результативність прогнозування. У наступному розділі будуть детально розглянуті результати порівняння цих методів та зроблені висновки щодо їх застосування у задачі прогнозування ваги улову риби.

3.3 Підсумок результатів

У цьому розділі буде проаналізовано результати роботи чотирьох алгоритмів: лінійної регресії, випадкового лісу, багатошарового персептрону та регресії на основі опорних векторів (SVR). Усі моделі оцінювалися за допомогою 10-кратної перехресної перевірки. Порівняння коефіцієнтів помилок для алгоритмів представлено на рисунку 3.9, де показано їх відносну ефективність у порівнянні з базовою лінією. Коефіцієнт нижче 1.0 свідчить про те, що алгоритм перевершує базовий рівень.

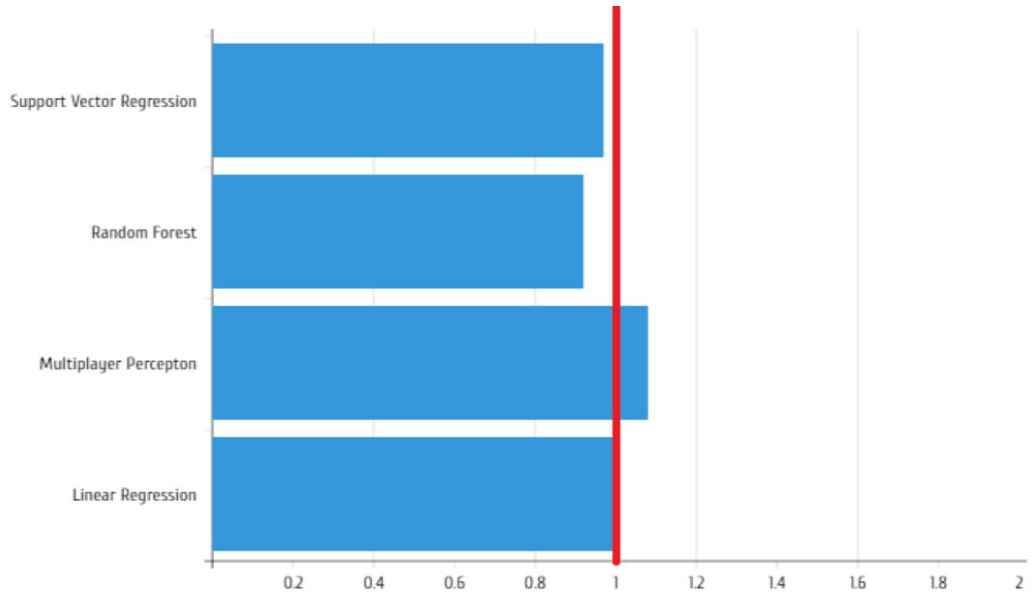

Рисунок 3.11 – Коефіцієнти помилок алгоритмів

Лінійна регресія показала результати, аналогічні базовому алгоритму, з коефіцієнтом помилки, що дорівнює 1.0, підтверджуючи, що для простих лінійних залежностей вона є ефективним інструментом. Випадковий ліс виявився найбільш ефективним серед усіх розглянутих методів, зменшивши похибку до 0.92, що робить його найкращим вибором для прогнозування ваги риби. Це підтверджує високу стійкість та здатність випадкового лісу ефективно працювати з комплексними наборами даних.

Багатошаровий персептрон, хоча й мав деякі труднощі з налаштуванням, показав дещо гірші результати у порівнянні з базовим алгоритмом, з коефіцієнтом помилки 1.08. Важливо зазначити, що налаштування гіперпараметрів і перетворення періодичних атрибутів у синус/косинус, описане в розділі 2.1.3, дещо покращило результати, але загальний коефіцієнт помилки залишився вище 1.0.

Регресія на основі опорних векторів (SVR) також показала значні результати, зменшивши середньоквадратичну помилку до 0.97. Використання ядра RBF дозволило моделі ефективно працювати з нелінійними даними, забезпечуючи високу точність прогнозів.

Ці результати підтверджують, що для задачі прогнозування ваги улову риби найбільш ефективними є методи випадкового лісу та регресії на основі опорних векторів. Випадковий ліс, зокрема, демонструє найкращі результати завдяки своїй здатності ефективно працювати з багатовимірними та складними наборами даних.

3.3.1 Вдосконалення алгоритму випадкового лісу

Випадковий ліс виявився найефективнішим серед усіх розглянутих алгоритмів у задачі прогнозування ваги улову риби. Однак, завжди існує можливість покращення результатів моделі шляхом оптимізації гіперпараметрів та застосування додаткових методів передобробки даних.

Одним із методів, які було випробувано для покращення результатів, було перетворення періодичних атрибутів у sin/cos. Це дозволяє моделі краще враховувати циклічну природу таких атрибутів, як місячний день. Періодичні атрибути можуть спричиняти труднощі для моделей, оскільки традиційні числові представлення не враховують їх циклічну природу. Перетворення таких атрибутів у значення синуса та косинуса дозволяє моделі краще враховувати цю циклічність.

Для вдосконалення моделі випадкового лісу, періодичні атрибути, такі як місячний день, були перетворені у sin/cos, після чого модель була навчена знову. Впровадження цього методу можна побачити у наступному фрагменті коду на рисунку 3.12

```
X_train['sin_moon_day'] = np.sin(2 * np.pi * X_train['moon_day'] / 30)
X_train['cos_moon_day'] = np.cos(2 * np.pi * X_train['moon_day'] / 30)
X_test['sin_moon_day'] = np.sin(2 * np.pi * X_test['moon_day'] / 30)
X_test['cos_moon_day'] = np.cos(2 * np.pi * X_test['moon_day'] / 30)

X_train = X_train.drop('moon_day', axis=1)
X_test = X_test.drop('moon_day', axis=1)

forest_model = RandomForestRegressor(n_estimators=100, random_state=42)
forest_model.fit(X_train, y_train)
y_pred_forest = forest_model.predict(X_test)

mse_forest = mean_squared_error(y_test, y_pred_forest)
r2_forest = r2_score(y_test, y_pred_forest)
```

Рисунок 3.12 Впровадження методу перетворення у sin/cos

Результати після застосування перетворення sin/cos для випадкового лісу показані на рисунку 3.13 Хоча результати демонструють деяке покращення, вони недостатньо значні, щоб довести, що це перетворення суттєво покращує ефективність моделі. Коефіцієнт детермінації (R^2) та середньоквадратична помилка (MSE) залишаються близькими до попередніх значень.

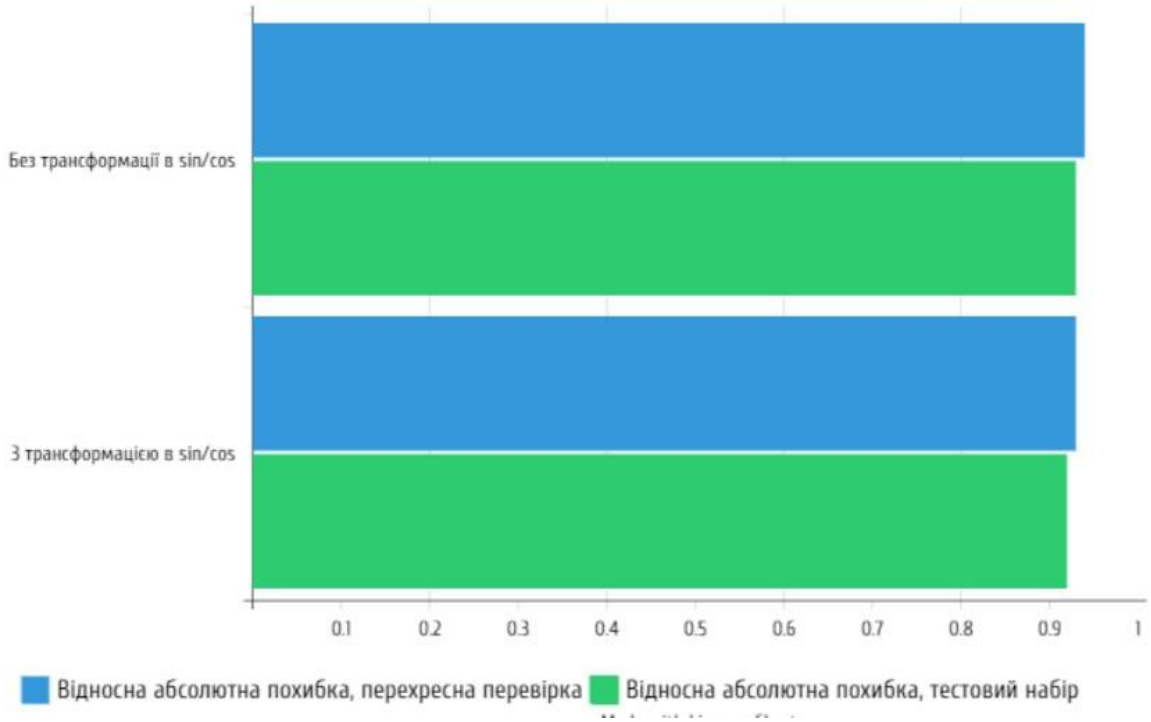

Рисунок 3.13 – Результат перетворення атрибутів у sin/cos

Таким чином, перетворення періодичних атрибутів у sin/cos може мати певний позитивний ефект на результати моделі, але в даному випадку ефект був мінімальним. Проте, цей метод є перспективним і може бути корисним для інших наборів даних або інших моделей. Додаткові експерименти з іншими гіперпараметрами та методами передобробки можуть також допомогти у вдосконаленні моделі.

3.4 Аналіз важливості вхідних атрибутів

Для оцінки важливості вхідних атрибутів було проведено експеримент з навчанням алгоритму випадкового лісу на окремих атрибутах. Метою цього аналізу було визначити, наскільки кожен окремий атрибут сприяє точності прогнозування успіху риболовлі.

Жоден окремий атрибут не виявився достатньо інформативним для точного прогнозування результату. Усі атрибути, коли використовувалися окремо, показали результати, еквівалентні або гірші за базовий рівень. Це

свідчить про те, що для точного прогнозування успіху риболовлі необхідно враховувати комбінацію декількох атрибутів.

Для детальнішого аналізу було проведено дослідження з накопиченням атрибутів, де кожен новий атрибут додавався до моделі поетапно. Результати цих експериментів представлені на рисунку 3.15

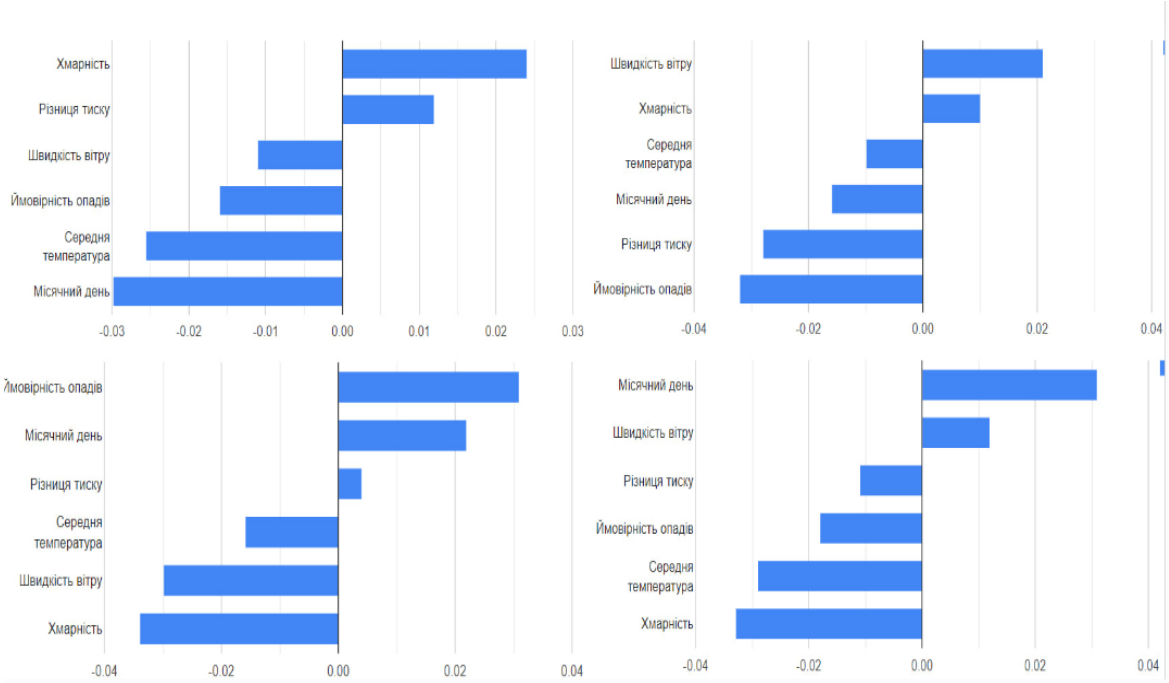

Рисунок 3.14 – Абсолютна різниця помилок для декількох ітерацій накопичення атрибутів

На рисунку кожен графік містить чотири стовпці, які представляють різницю між відносною абсолютною похибкою алгоритму випадкового лісу, натренованого на певній кількості атрибутів, та базовим алгоритмом. Мітки на осі x позначають останній доданий атрибут. Перший стовпець представляє різницю помилок для випадкового лісу, натренованого на першому атрибуті, другий стовпець – для випадкового лісу, натренованого на двох атрибутах, і так далі.

З аналізу видно, що для суттєвого зменшення похибки необхідно використовувати кілька атрибутів одночасно. Найбільший вплив на точність прогнозування мають комбінації атрибутів, що включають

температуру, швидкість вітру та місячний день. Це вказує на важливість комплексного підходу до аналізу даних для досягнення максимальної точності прогнозування.

Таким чином, цей аналіз підтверджує, що для точного прогнозування ваги улову риби необхідно враховувати взаємодію декількох важливих атрибутів, а не покладатися на окремі параметри. Це підкреслює важливість комплексного підходу до збору та обробки даних для побудови ефективних моделей прогнозування.

3.4.1 Аналіз результатів

Результати цього дослідження підтверджують, що алгоритми машинного навчання, зокрема випадковий ліс, можуть ефективно прогнозувати очікувану вагу вилову у любительському риболовлі. Випадковий ліс демонструє гарні результати між різними видами риб, що є дуже цікавим, оскільки різні види займають різні географічні регіони та біологічні ніші. Це особливо корисно для рибалок-любителів, які зазвичай покладаються на неперевірені методи, такі як місячні календарі. Той факт, що лінійна регресія не змогла перевершити базову лінію, вказує на нелінійний зв’язок між вхідними атрибутами та цільовою змінною.

Метод регресії на основі опорних векторів (SVR) показав результати гірші, ніж випадковий ліс, але кращі, ніж багатошаровий персептрон. Це свідчить про те, що SVR краще справляється з нелінійними зв’язками у даних, проте все ж не може зрівнятися з випадковим лісом за точністю прогнозів. SVR може бути більш стабільним до деяких видів шуму у даних порівняно з багатошаровим персептроном, але менш ефективним у використанні всіх можливостей різноманітних вхідних атрибутів.

Багатошаровий персептрон схильний до перенавчання, що може бути спричинене кількома факторами. Ймовірно, кількість даних є занадто малою, щоб модель могла точно наблизитися до цільової змінної. Також багатошаровий персептрон може бути більш чутливим до помилкових

даних або неправильно маркованих атрибутів, що завжди є проблемою, враховуючи, що дані частково створюються користувачами та переважно походять від сторонніх джерел. Хоча було перевірено декілька комбінацій гіперпараметрів, не можна виключати можливість існування такої комбінації, яка дала б значно кращий результат.

Випадковий ліс показує хорошу роботу для кількох видів, використовуючи різні підмножини вхідних атрибутів. Завдяки тому, що вихід випадкового лісу є середнім результатом кількох незалежних моделей, він має природний захист від перенавчання та впливу викидів. Зниження помилки при навчанні на повному навчальному наборі, порівняно з 90% набору, як у випадку перехресної перевірки, ще більше підтверджує гіпотезу про те, що алгоритм дійсно вивчає справжні характеристики даних і використовує їх для зменшення похибки.

Дослідження важливості атрибутів для прогнозування показало, що найбільш значущими атрибутами є швидкість вітру, різниця тиску та температура. Ці атрибути мають найбільший вплив на точність прогнозу ваги вилову, оскільки вони безпосередньо впливають на поведінку риби та її активність. Випадковий ліс виявився ефективним у використанні цих атрибутів для підвищення точності прогнозів.

ВИСНОВКИ

У даній магістерській кваліфікаційній роботі було проведено комплексне дослідження та аналіз моделей і методів штучного інтелекту для задачі прогнозування вилову риби. Було розглянуто та порівняно кілька алгоритмів машинного навчання, зокрема лінійну регресію, випадковий ліс, багатошаровий персептрон та метод опорних векторів (SVR).

У теоретичній частині роботи виконано детальний огляд класичних та сучасних методів прогнозування. Було досліджено еволюцію методологій прогнозування, описано основні принципи та математичні моделі, які лежать в основі кожного з розглянутих алгоритмів.

У ході практичного дослідження було зібрано та підготовлено набір даних для аналізу, включаючи звіти про улов рибалок та погодні умови. Після попередньої обробки даних проведено навчання моделей на основі бібліотеки Scikit-learn для Python, що дозволило реалізувати та оцінити ефективність кожного з алгоритмів.

Особлива увага була приділена запобіганню перенавчанню моделей та забезпеченню їх узагальнюваності. Було застосовано методи крос-валідації та оптимізації параметрів моделей, що дозволило підвищити точність прогнозів.

Зважаючи на велику кількість факторів, які можуть вплинути на успіх любительської риболовлі, результати експерименту виявилися задовільними, оскільки похибка була зменшена на 13% при використанні семи вхідних атрибутів. Зменшення похибки на 100% означало б досконале знання очікуваної ваги риби, тому зменшення на 13% є значним досягненням, особливо враховуючи, що результати були узагальнені для кількох видів риб.

Випадковий ліс виявився найефективнішим алгоритмом, що можна пояснити його стійкістю до перенавчання, особливо в умовах наявності помилкових навчальних даних. Цей алгоритм зміг покращити базову лінію

на 13%, що означає, що середнє оцінене значення ваги риби на 13% ближче до реальної ваги порівняно з базовою моделлю. Це свідчить про те, що використання випадкового лісу може суттєво підвищити точність прогнозів, покращуючи шанси на успішну риболовлю.

Метод регресії на основі опорних векторів (SVR) показав результати гірші, ніж випадковий ліс, але кращі, ніж багатошаровий персептрон. Це свідчить про те, що SVR краще справляється з нелінійними зв'язками у даних, проте він все ж не може зрівнятися з випадковим лісом за точністю прогнозів. Багатошаровий персептрон, у свою чергу, виявився схильним до перенавчання, що може бути спричинене малою кількістю даних та наявністю помилкових атрибутів або неправильно маркованих даних.

Дослідження важливості атрибутів для прогнозування показало, що найзначущими атрибутами є швидкість вітру, різниця тиску та температура. Ці атрибути мають найбільший вплив на точність прогнозу ваги вилову, оскільки вони безпосередньо впливають на поведінку риби та її активність. Випадковий ліс виявився ефективним у використанні цих атрибутів для підвищення точності прогнозів.

Загалом, результати цього дослідження можуть бути дуже корисними для рибалок-любителів, які зазвичай покладаються на неперевірені методи, такі як місячні календарі. Використання алгоритмів машинного навчання, зокрема випадкового лісу, може суттєво підвищити точність прогнозів та ефективність риболовлі, надаючи користувачам більш надійні інструменти для планування своїх риболовецьких заходів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Портал рибалок. URL: http://www.fgids.com/about/ (дата звернення 15.05.2024)
2. World Weather Online. URL: https://www.worldweatheronline.com/aboutus.aspx (дата звернення 17.05.2024)
3. The evolution of forecasting techniques. Genpact. URL: https://www.genpact.com/insight/technical-paper/the-evolution-of-forecasting-techniques-traditional-versus-machine-learning-methods (дата звернення: 16.05.2024).
4. Genuer, R., Poggi, J.-M. and Tuleau-Malot, C. (2010). Variable selection using random forests. Pattern Recognition Letters, 31 (14), pp.2225–2236. DOI: http://dx.doi.org/10.1016/j.patrec.2010.03.014 (дата звернення: 19.05.2024).
5. Genuer, R., Poggi, J.-M. and Tuleau-Malot, C. (2015). VSURF: An R Package for Variable Selection Using Random Forests. The R Journal, 7 (2), pp.19–33. DOI: https://doi:10.32614/RJ-2015-018 (дата звернення: 19.05.2024).
6. Fox J. Applied Regression Analysis and Generalized Linear Models, 2nd ed. Sage Publications, 2008.
7. Gabadinho A., Ritschard G., Muller N.S., Studer M. Analyzing and Visualizing State Sequences in R with TraMineR. // Journal of Statistical Software. 2011. V.40 (4). p. 1-37. URL: http://www.jstatsoft.org/v40/i04/.
8. Everitt B.S., Howell D.C. Encyclopedia of Statistics in Behavioral Science. Chichester: Wiley & Sons Ltd., 2005. 2132 p.
9. Cristianini N., Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, 2000.

10. Burnham K.P., Anderson D.R. Model selection and multimodel inference: a practical information-theoretic approach. N.Y.: Springer-Verlag, 2002. 496 p.
11. Faraway J.J. Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models. Chapman, Hall/CRC, 2006. 345 p.
12. Breiman L. Random forests // Machine Learning. 2001. V. 45 (1). P. 5-32.
13. Vapnik V.N. The Nature of Statistical Learning Theory. Springer, 1995
14. Kuhn M. Predictive Modeling with R and the caret Package. 2013. URL: https://www.r-project.org/ (дата звернення: 16.05.2024).
15. McArdle B.H., Anderson M.J. Fitting multivariate models to community data: a comment on distance-based redundancy analysis // Ecology. 2001. V. 82, № 1. P. 290-297.
16. Mount J., Zumel N. Exploring Data Science. Manning Publications Co., 2016. 184 p.
17. Kuhn M. Variable Selection Using The caret Package. 2012. URL: http://citeseerx.ist.psu.edu (дата звернення: 16.05.2024).
18. MacQueen J. Some methods for classification and analysis of multivariate observations. // The Fifth Berkeley Symposium on Mathematical Statistics and Probability. eds L.M. Le Cam, J. Neyman. Berkeley, CA: University of California Press. 1967. P. 281–297.
19. McArdle B.H., Anderson M.J. Fitting multivariate models to community data: a comment on distance-based redundancy analysis // Ecology. 2001. V. 82, № 1. P. 290-297.
20. Zaki M., Meira W. Data Mining and Analysis. Fundamental Concepts and Algorithms. Cambridge University Press, 2014. URL: http://www.dataminingbook.info/
21. Zhao Y. R and Data Mining: Examples and case studies. Academic Press, 2012. 256 p.

22. Zhao Y., Zhang C., Cao L. Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction. Information Science Reference. 2009.
23. Hand D. Classifier Technology and the Illusion of Progress // Statistical Science. 2006. V. 21. P. 1-14.
24. Hunt E.B., Marin J., Stone P.J. Experiments in Induction. New York: Academic Press, 1966.
25. Kursa M., Rudnicki W. Feature Selection with the Boruta Package // Journal of Statistical Software. 2010. V.36 (11). P. 2-12.