МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет радіоелектроніки

Центр післядипломної освіти

Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

_____другий (магістерський)_____
(рівень вищої освіти)

_____Дослідження ефективності застосування сімейств дерев рішень_____

_____для прийняття рішень при відборі кандидатів у команду_____
(тема)

Виконала:

студентка _2_курсу, групи_____ІПЗздм-21-1____

_____Гурова Ю.В._____
(прізвище, ініціали)

Спеціальність __121 – Інженерія програмного___

_____забезпечення_____

Тип програми _____Освітньо-наукова_____

Керівник _____проф. Смеляков С.В._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ _____Дудар З.В._____
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет _____Центр післядипломної освіти_____

Кафедра _____Програмної інженерії_____

Рівень вищої освіти _____другий (магістерський)_____

Спеціальність _____121 – Інженерія програмного забезпечення_____

Тип програми _____освітньо-наукова програма_____

Освітня програма _____Програмна інженерія_____

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«__»_____2023 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентки _____*Гурової Юлії Володимирівни*_____

(прізвище, ім'я, по батькові)

1. Тема роботи: _«Дослідження ефективності застосування сімейств дерев рішень для прийняття рішень при відборі кандидатів у команду»

затверджена наказом по університету від «3» __квітня__2023 р. № _83Стз_

2. Термін подання студенткою роботи до екзаменаційної комісії «12»_травня_2023 р.

3. Вихідні дані до роботи: *методи відбору кандидатів, системи відстеження кандидатів, алгоритми сімейства дерев рішень, OS MacOS, мова програмування Java, Spark Framework, середовище розробки IntelijIDEA.*

4. Перелік питань, що потрібно опрацювати в роботі: *аналіз предметної галузі, мета роботи, аналіз систем відстеження кандидатів, аналіз алгоритмів сімейства дерев рішень, аналіз методів збору вхідних даних, дослідження ефективності алгоритмів машинного навчання при наймі кандидатів на роботу.*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	23.01.2023	Виконано
2	Формування мети роботи	06.02.2023	Виконано
3	Аналіз ATS на основі штучного інтелекту	13.02.2023	Виконано
4	Вибір і аналіз алгоритмів сімейства дерев рішень та методів оцінки їх якості	24.02.2023	Виконано
5	Аналіз джерела даних та методів їх збору	01.03.2023	Виконано
6	Планування і розробка ПЗ	08.03.2023	Виконано
7	Проведення експериментального дослідження і аналіз результатів	23.03.2023	Виконано
8	Підготовка пояснювальної записки	01.04.2023	Виконано
9	Перевірка на плагіат	07.05.2023	Виконано
10	Нормоконтроль	08.05.2023	Виконано
11	Рецензування	09.05.2023	Виконано
12	Підготовка презентації та доповіді	09.05.2023	Виконано
13	Попередній захист	10.05.2023	Виконано
14	Занесення роботи в електронний архів	11.05.2023	Виконано
15	Допуск до захисту у зав. кафедри	12.05.2023	Виконано

Дата видачі завдання «23» січня 2023 р.

Студентка _____ Гурова Ю.В.
(підпис) (прізвище, ініціали)

Керівник роботи _____ проф. Смеляков С.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Кваліфікаційна робота магістра містить: 70 стор., 4 рис., 10 табл., 39 джерел.

ПІДБІР КАНДИДАТІВ, НАЙМ КАНДИДАТІВ, ПРАВИЛА ПРИЙНЯТТЯ РІШЕНЬ, СИСТЕМИ ВІДСТЕЖЕННЯ КАНДИДІТАВ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, МАШИННЕ НАВЧАННЯ, АЛГОРИТМИ МАШИННОГО НАВЧАННЯ, ДЕРЕВА РІШЕНЬ.

Об’єктом дослідження є методи та алгоритми, які використовуються для прийняття рішень при відборі кандидатів для команди та можливість застосування алгоритмів машинного навчання для удосконалення процесу найму.

Метою даного дослідження є підвищення ефективності процесу прийняття рішень при відборі кандидатів з великої кількості даних за допомогою застосування алгоритмів сімейства дерев рішень.

За результатами роботи було проаналізовано існуючі методи відбору кандидатів та системи відстеження кандидатів на основі штучного інтелекту, було проведено експериментальне дослідження ефективності застосування алгоритмів сімейства дерев рішень для прийняття рішень при відборі кандидатів у команду.

Для проведення експерименту було розроблене ПЗ за допомогою мови програмування Java із використанням Spark Framework.

RECRUITMENT CANDIDATES, HIRING CANDIDATES, DECISION RULES, ATS, DATA MINING, MACHINE LEARNING, MACHINE LEARNING ALGORITHMS, DECISION TREES.

The object of the study is the methods and algorithms used to make hiring team candidates and the possibility of using modern machine learning algorithms to improve the hiring process.

The purpose of this study is to improve the efficiency of the decision-making process when selecting candidates from a large amount of data by applying decision tree algorithms.

The work analyzed existing methods of candidate selection and applicant tracking systems based on artificial intelligence and conducted an experimental study of the effectiveness of using decision tree algorithms for making hiring decisions.

To conduct the experiment, software was developed in the Java programming language using the Spark Framework.

Я, Гурова Юлія Володимирівна, студентка групи ІПЗздм-21-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження ефективності застосування сімейств дерев рішень для прийняття рішень при відборі кандидатів у команду», що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлена з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

ПЗ – програмне забезпечення

ATS – система відстеження кандидатів

CNBC – абревіатура назви «Consumer News and Business Channel» – це американський телевізійний канал, що спеціалізується на трансляції новин про бізнес, фінанси та економіку

ML – машинне навчання

NLP – природня обробка мови

GBT – дерева з градієнтним підсиленням

ВСТУП

Найм та утримання працівників є одними з найважливіших елементів успішного функціонування будь-якої компанії, оскільки вони є основою для досягнення бізнес-цілей. Проте процес найму може бути складним та витратним, особливо у випадку великих компаній з великою кількістю кандидатів на роботу.

Застосування алгоритмів машинного навчання може бути корисним інструментом у процесі відбору та оцінки кандидатів на роботу. Це дозволить зменшити витрати та збільшити точність відбору, оскільки машинне навчання може виявляти закономірності в даних та забезпечувати більш об'єктивну оцінку кандидатів на роботу. Наприклад, застосування алгоритмів класифікації дозволить швидко та ефективно відсіяти кандидатів, які не відповідають певним критеріям. Крім того, такі алгоритми можуть забезпечити об'єктивний підхід до відбору кандидатів, оскільки вони не піддаються емоційному впливу, що може вплинути на рішення людини.

Таким чином, дослідження застосування алгоритмів машинного навчання може стати важливим кроком для покращення процесу найму та забезпечення конкурентних переваг в економіці, заснованій на знаннях.

Метою даного дослідження є підвищенні ефективності процесу прийняття рішень при відборі кандидатів з великої кількості даних за допомогою застосування алгоритмів сімейства дерев рішень.

Для досягнення мети дослідження необхідно вирішити такі завдання:

- дослідити існуючі методи прийняття рішень при відборі кандидатів для команди;
- дослідити використання штучного інтелекту у системах відстеження кандидатів;
- проаналізувати збір та обробку даних про кандидатів;
- розглянути та визначити алгоритми сімейства дерев рішень за допомогою яких буде проведене дослідження;
- визначити метрики, що будуть використані для оцінки ефективності застосування обраних алгоритмів;

- розробити ПЗ для проведення експериментальної частини дослідження;
- виміряти ефективність застосування алгоритмів сімейства дерев рішень за допомогою оціночних метрик;
- проаналізувати отримані результати та надати практичні рекомендації стосовно використання кожного з алгоритмів у процесі відбору кандидатів на роботу;
- визначити недоліки використання алгоритмів дерев рішень;
- сформулювати висновки дослідження.

Об’єктом дослідження є методи та алгоритми, які використовуються для прийняття рішень при відборі кандидатів для команди.

Предметом дослідження є ефективність застосування алгоритмів сімейства дерев рішень для прийняття рішень при відбору кандидатів у команду.

Методи дослідження включають аналіз літературних джерел, збір та аналіз даних про процес відбору кандидатів, аналіз систем відстеження кандидатів на основі штучного інтелекту, порівняння ефективності застосування різних алгоритмів сімейства дерев рішень та статистичний аналіз результатів.

Отримані результати дослідження можуть бути використані для поліпшення процесу відбору кандидатів на роботу. На основі побудованої моделі на основі дерев рішень можна розробити програму автоматизованого відбору кандидатів, що дозволить знизити витрати на процес відбору та підвищити ефективність відбору. Крім того, результати можуть бути корисні для менеджерів з управління персоналом при підготовці та аналізі кандидатів на роботу.

Результати дослідження було представлено на VII Міжнародній конференції з комп’ютерної лінгвістики та інтелектуальних систем CoLInS-2023 [1], яка індексується в наукометричних базах CEUR і Scopus.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналітичний огляд

Рекрутинг – це процес найму нового персоналу, який вимагає від рекрутерів розуміння потреб бізнесу та вміння працювати з потенційними кандидатами. Крім того, рекрутинг включає в себе багато етапів, таких як планування вакансії, залучення та пошук кандидатів, відбір кандидатів, проведення співбесіди, перевірку кандидатів та прийняття рішення про працевлаштування та адаптація. В цілому, рекрутинг є складним, але дуже важливим процесом, який може вирішити проблему з нестачею кадрів у компанії та забезпечити ріст бізнесу.

Після залучення потенційних кандидатів за допомогою рекрутингу, наступним кроком є відбір кваліфікованих заявок. Процес відбору заявок є важливою процедурою для будь-якої компанії, оскільки дозволяє знайти кваліфікованих кандидатів. Рекрутер повинен відокремити заявки кандидатів, які не відповідають кваліфікаційним вимогам посади, та обрати тих, які найбільше відповідають вимогам. Для цього можуть застосовуватися різні методи, включаючи оцінку резюме, використання тестів та співбесід з кандидатами. Більш детально про методи відбору кандидатів описано у розділі 1.2. Мета відбору кандидатів полягає у забезпеченні наступного етапу рекрутингу – найму найбільш придатних кандидатів для вакансії.

Підбір правильних кандидатів може вимагати значних зусиль та ресурсів, включаючи час рекрутерів, витрати на рекламу вакансії, огляд резюме та проведення співбесід. Компанії можуть отримувати тисячі резюме на кожну вакансію і наймати спеціальних співробітників для відбору кваліфікованих кандидатів. Наприклад, в 2014 році Google розмістив на своєму сайті 7 000 вакансій, на які компанія отримала 3 мільйони резюме. Згідно іншого дослідження [2], що аналізує дані системи відстеження кандидатів Workable, показник кількості кандидатів на одну вакансію зріс на 45.9% у період з липня 2022 року по лютий 2023 року. Також, послідовно зростає показник середнього значення кількості вакансій на одну компанію, починаючи від грудня 2023. На рисунку 1 наведено

діаграму, що показує тенденцію щодо кількості кандидатів на одну вакансію до кінця березня 2023 року порівняно з середнім показником 2019 року.

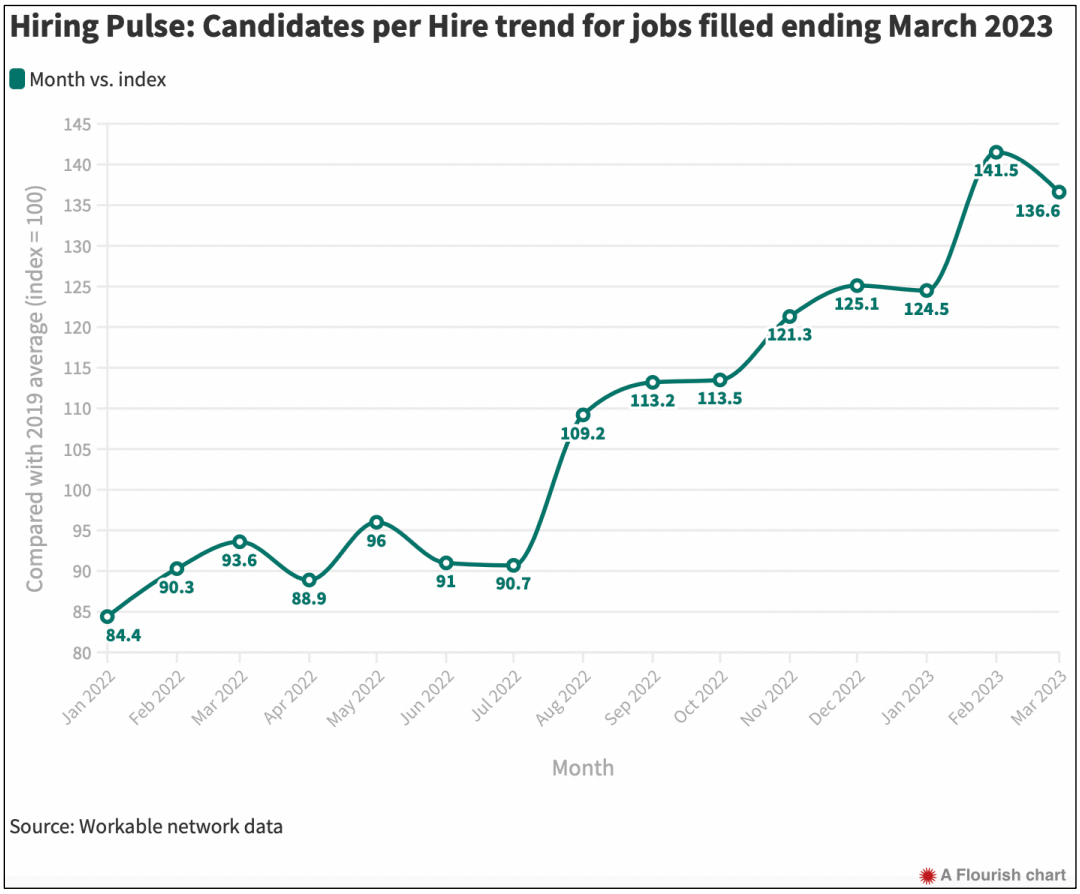

Рисунок 1 – Тенденція щодо кількості кандидатів на одну вакансію [2]

За рахунок збільшення кількості кандидатів, процес відбору стає надто складним і непрозорим, а ще й дуже дорогим. Тому компанії намагаються оптимізувати цей процес, наприклад, застосовуючи системи відстеження кандидатів (розділ 1.3), які допомагають створювати більш ефективний і структурований процес відбору. Системи відстеження кандидатів можуть застосовувати алгоритми машинного навчання у відборі персоналу, що дозволяє значно зменшити час, необхідний для перегляду кожного резюме, та знизити витрати на рекрутинг. Такі алгоритми здатні виявляти ключові слова та фрази, пов'язані з певною вакансією, і відсіювати кандидатів, які не відповідають вимогам.

У разі недостатнього підбору кандидатів компанія може стикнутися з низькою продуктивністю, високими витратами на підготовку та перепідготовку

персоналу, зі зниженням рівня задоволеності клієнтів, втратою позицій на ринку, а також зі збитками у бізнесі. У своєму дослідженні Leadership IQ виявив, що 46% новоприйнятих працівників зазнають невдачі протягом 18 місяців, тоді як лише 19% досягають однозначного успіху. Враховуючи це, важливо вкласти достатні зусилля та ресурси у процес підбору кандидатів для забезпечення успіху компанії.

1.2 Огляд існуючих методів відбору кандидатів

Ефективний відбір кандидатів має вирішальне значення для побудови успішних і продуктивних команд, а з ростом доступності даних і технологій існує широкий спектр методів відбору кандидатів, які можуть допомогти команді з найму створити надійний процес рекрутингу та знайти найкращих кандидатів.

Методи відбору кандидатів – це критерії вибору оптимального кандидата на певну посаду. Вони передбачають аналіз здібностей, освіти, досвіду та особистих якостей людини, щоб визначити, чи зможе вона виконувати ключові завдання та відповідати загальній культурі компанії. Існує кілька категорій методів відбору працівників, зокрема тести або іспити, співбесіди та попередні дослідження. Команди з найму часто використовують їх, щоб визначити найкращих кандидатів з великого списку і побачити, як вони підходять до різних робочих ситуацій. Одночасно можуть використовуватися декілька різних методів задля отримання більш якісного результату у рекрутингу,

Існують наступні методи відбору кандидатів [3]:

- процес сортування резюме – відбір найкращих кандидатів з резюме, зазвичай пошук за ключовими словами, які відповідають опису вакансії;
- вступний скринінг – визначення кандидатів з попередньо складеного списку за допомогою додаткової оцінки кваліфікації та професіоналізму;
- тестове завдання – отримання більш детальної інформації про практичні навички кандидата, необхідні для роботи на посаді за допомогою виконання тестового завдання;

- тест на когнітивні здібності – перевірка когнітивних здібностей, щоб оцінити здатність кандидата обробляти нову інформацію, вирішувати проблеми та встановлювати зв'язки між різними фактами;
- рекомендації кандидата – інформація від попереднього роботодавця або колег, яка може розкрити його поведінку на роботі та загальні здібності;
- особиста співбесіда – проведення співбесіди для звуженої кількості обраних кандидатів;
- оцінка особистості – вимірювання характеристик людини, щоб побачити, чи відповідають вони вимогам конкретної посади або робочому середовищу компанії;
- оцінювання професійних знань – перевірка, чи має кандидат достатньо критично важливих знань для виконання певних посадових обов'язків;
- тест на ситуаційне судження – проведення поведінкового іспиту, що дозволяє краще зрозуміти, як кандидат може реагувати на повсякденні ситуації, зокрема, на конфлікти або події з високим рівнем напруженості. Особливо корисним є для керівних посади, оскільки вони вимагають певних навичок прийняття рішень і міжособистісних здібностей;
- біографічна інформація – розпитування кандидата про його особисту історію, характеристики, хобі та інтереси задля розуміння того, як кандидат може виконувати повсякденні завдання на посаді;
- групова співбесіда – проведення співбесіди з декількома кандидатами одночасно, задля того, щоб зрозуміти як різні кандидати працюють разом, що може допомогти при формування ефективної нової команди;
- стажування або навчання – проведення стажування або навчання для того, щоб навчити кандидата виконувати необхідні завдання на посаді або зрозуміти, чи підходить кандидат для компанії, спостерігаючи за виконанням обов'язків і взаємодією з колегами;
- оцінка доброчесності – процес перевірки, який дозволяє виміряти систему цінностей та етичні стандарти кандидата;

- симуляція роботи – надання кандидату робочого завдання, з яким він може стикатися щодня, щоб оцінити, як він справляється з ним;
- фізичне оцінювання – оцінка фізичних здібностей для професій, які вимагають фізичної праці;
- перевірка біографічних даних – процес перевірки певної інформації про кандидата, щоб перевірити, чи відповідають факти, викладені в резюме (навчальні заклади, сертифікати, тощо);
- внутрішні процеси та рекомендації – розгляд кандидатів, які вже подавали заявки раніше та були перевірені.

Отже, існує безліч різних методів відбору кандидатів, які можуть варіюватися у залежності від конкретної вакансії та організації. Процес відбору кандидатів повинен бути детально продуманий, щоб забезпечити успішне наймання кваліфікованого персоналу. Залежно від вимог конкретної вакансії, можна використовувати традиційні методи, такі як інтерв'ю та тестування, або більш сучасні методи, зокрема інструменти на основі штучного інтелекту.

Неправильне рішення про найм може призвести до високої плинності кадрів, низького морального духу та марної трати ресурсів, тому компанії прагнуть застосовувати найкращі методи відбору кандидатів, щоб зменшити ризики, пов'язані з помилковими рішеннями про найм.

1.3 Опис систем відстеження кандидатів (ATS)

Для знаходження найкращих кандидатів під час процесу подання заявок, компанії можуть використовувати програмне забезпечення, що дозволяє аналізувати дані з резюме кандидатів. Згідно дослідженню Forbes [4], 99% компаній зі списку Fortune 500 використовують систему відстеження кандидатів, щоб прийняти кандидата на посаду. CNBC повідомляє, що понад 75% резюме ніколи не потрапляють до людських очей. Завдяки автоматичному аналізу резюме, ПЗ дозволяє заощадити час і гроші компанії та забезпечити якісний відбір кандидатів. Інструменти для аналізу резюме є частиною більшості систем відстеження кандидатів (ATS) [5]. Розбір резюме допомагає команді з підбору персоналу

шукати ключові слова, додавати конкретні критерії пошуку, щоб знайти найбільш кваліфікованих кандидатів і ефективно зберігати велику кількість резюме.

ATS – це програмний інструмент, який допомагає командам з підбору персоналу відбирати кандидатів під час процесу найму на роботу. Команда Forbes Advisor проаналізувала найкращі варіанти ATS [6] на основі десятків даних і запропонувала найкращі на 2023 рік:

- Freshteam від Freshworks: найкраща в цілому;
- JazzHR: найкраща для стартапів;
- BreezyHR: найкраща безкоштовна система відстеження кандидатів;
- Rippling: найкраща CAP плюс розрахунок заробітної плати;
- Greenhouse: найкраща для середнього бізнесу;
- Zoho Recruit: найкраща за безкоштовну пробну версію;
- BambooHR: найкраща для великих роботодавців;
- Workable: найкраща для компаній, що займаються розробкою програмного забезпечення;
- Bullhorn: найкраща для рекрутингових агентств;
- Recruit CRM: краща ATS плюс CRM;
- Recruitee: віджет найкращих вакансій.

Після подачі заявки на роботу, резюме кандидата проходить через систему відстеження кандидатів (ATS), перш ніж потрапити до команд з підбору персоналу. Ці системи допомагають командам з підбору персоналу зібрати необхідну інформацію про кандидатів та організувати її в одному місці. Основне завдання ATS [5] – впорядкувати та автоматизувати процес підбору персоналу шляхом відстеження та систематизації даних про кандидатів, дозволяючи командам з підбору персоналу зосередитися на розгляді найбільш кваліфікованих кандидатів, які найкраще підходять для роботи. Нижче наведено деякі інші можливості, які може надати ATS:

- розміщення вакансій;
- автоматичний збір та обробка резюме кандидатів;
- формування єдиної бази кандидатів;

– можливість співвіднесення резюме кандидатів відповідно до відкритих вакансій;
– організація та планування співбесід з кандидатами;
– зберігання резюме на основі попередньо визначених критеріїв та історії взаємодії з кандидатами;
– можливості аналізу та створення звітів.

Головна перевага кожної ATS полягає в тому, що вони дозволяють відібрати кількох кандидатів з великої кількості заявок, які надходять на одну вакансію, тим самим оптимізувати та вдосконалити весь процес рекрутингу. За даними Glassdoor (один з найбільших у світі сайтів з пошуку роботи та рекрутингу), у середньому на одну вакансію приходить близько 250 резюме. З цих 250 лише 4-6 кандидатів запрошують на співбесіду і лише 1 кандидат отримує пропозицію на роботу. Тобто, після того, як команда з підбору персоналу відсортовує велику кількість резюме, вона обирає лише декілька кандидатів для співбесіди. Також, ATS може надавати компаніям наступні переваги:

– економія часу;
– фільтрація кандидатів;
– оптимізація процесу підбору працівників;
– покращення якості підбору кандидатів;
– зниження витрат на найм персоналу;
– збереження інформації про кандидатів;
– можливість відстеження кандидатів.

Хоча такі системи мають досить багато переваг, варто розглянути й потенційні недоліки. Ось декілька основних, на які варто звернути увагу:

– відсутність автоматичного підбору кандидатів згідно вакансіям;
– підвищена вартість послуг;
– неточність обробки резюме;
– безпека даних;
– складність у користуванні системою.

Існує дві різні групи систем відстеження кандидатів:

– стандартна ATS (без можливостей штучного інтелекту);
– ATS на основі штучного інтелекту.

У системах на основі штучного інтелекту використовуються прогнозні моделі, засновані на машинному навчанні, які приймають вхідні дані про кандидата та вакансію. Потім система прогнозує, наскільки добре кандидат підходить для вакансії, використовуючи реальні дані про кандидатів і статистичні висновки.

Кожна з цих груп має свої переваги та недоліки, і вибір конкретної системи залежить від потреб компанії та її ресурсів. Незалежно від обраної системи, важливо, щоб вона допомагала забезпечувати ефективний та об'єктивний процес відбору кандидатів, щоб компанії можна було залучати найкращих працівників.

1.4 Формулювання мети та завдання дослідження

У системах відстеження кандидатів на основі штучного інтелекту якість процесу підбору кандидатів залежить від ефективності алгоритмів, що використовуються. Якщо алгоритми не допомагають ефективно фільтрувати кандидатів, то може виникнути проблема з розрізненням кандидатів з різними рівнями кваліфікації. Недостатньо точний алгоритм може привести до того, що рекрутери пропустять важливу інформацію про кандидатів або навіть не звернуть увагу на деякі важливі кваліфікаційні характеристики. Це призводить до найму непідходящих кандидатів або відхилення найкращих кандидатів, як результат – зменшення якості найму у компанії. Тому важливо ретельно перевіряти та вдосконалювати алгоритми у системі відстеження кандидатів, щоб забезпечити максимально точний та ефективний процес підбору кандидатів.

Головною метою даного дослідження є можливість підвищенні ефективності процесу прийняття рішень при відборі кандидатів за допомогою застосування алгоритмів сімейства дерев рішень. Для досягнення мети дослідження необхідно вирішити наступні завдання:

– розглянути процес найму та існуючі методи прийняття рішень при відборі кандидатів для команди;
– розглянути системи відстеження кандидатів;

- проаналізувати використання штучного інтелекту у процесі найму;
- розглянути існуючі алгоритми інтелектуального аналізу даних;
- проаналізувати використання штучного інтелекту у системах відстеження кандидатів;
- визначити підходящі алгоритми машинного навчання для вирішення поставленої задачі;
- визначити та розглянути основні алгоритми сімейства дерев рішень для проведення експериментального дослідження;
- визначити метрики, що будуть використані для оцінки ефективності алгоритмів сімейства дерев рішень;
- проаналізувати попередній збір та обробку даних про кандидатів, визначити найбільш вагомі фактори, що впливають на їх придатність до конкретної вакансії;
- розробити ПЗ для проведення експериментального дослідження;
- провести експериментальне дослідження: побудувати моделі дерев рішень за допомогою обраних алгоритмів та оцінити ефективність їх застосування для вирішення поставленої задачі;
- проаналізувати результати дослідження та визначити найкращий алгоритм для покращення якості процесу прийняття рішень у наймі;
- визначити недоліки використання алгоритмів дерев рішень для вирішення поставленої задачі;
- сформулювати висновки дослідження.

2 РОЗГЛЯД І АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

2.1 Роль машинного навчання у процесі найму

З розвитком технологій машинне навчання стає все більш популярним інструментом у різних галузях. Однією з областей, де машинне навчання особливо досягло успіху, є проблеми класифікації, завдяки своїй здатності вчитися на великих обсягах даних і робити точні прогнози на основі нових даних. У статтях [7, 8] подано всебічний огляд методів машинного навчання та вступ до математичних і статистичних основ машинного навчання. Вони [7, 8] відіграли важливу роль у популяризації використання машинного навчання для задач класифікації та допомогли утвердити машинне навчання як фундаментальний інструмент у галузі науки про дані. У роботі [9] розглядається широке коло питань щодо аналізу та класифікації обмінних сегментів EEG, які відповідають певним корисним сигналам та артефактам. А також вирішення проблем ідентифікації людини, виявлення жертв техногенних катастроф [10], створення сучасних пошукових сервісів [11], систем електронного навчання [12, 13].

Останніми роками використання машинного навчання набуло популярності в процесі найму на роботу [8, 14], оскільки компанії шукають шляхи підвищення ефективності та результативності процесу найму. Ці роботи [8, 14] пропонують моделі для вилучення персональних даних із зовнішніх джерел для спільного аналізу з даними в резюме. Наприклад, багато великих компаній, таких як IBM, Hilton тощо, впровадили алгоритми машинного навчання в процес найму для автоматизації завдань і підвищення ефективності [15-17]. В тому числі і з урахуванням захисту персональних даних, як описано в [17]. З іншого боку, дослідження [18] виявило, що претенденти на роботу загалом мають низький рівень довіри та впевненості в алгоритмічному прийнятті рішень у процесах найму та відбору. Однак дослідження [18] також виявило, що сприйняття претендентів варіюється залежно від їхніх демографічних та особистих характеристик, а такі фактори, як вік, освіта та попередній досвід, впливають на їхнє ставлення до алгоритмічного прийняття рішень.

Алгоритми машинного навчання можуть аналізувати великі обсяги даних [19], виявляти закономірності та робити прогнози, потенційно скорочуючи час і витрати. Включаючи аналіз фотографій [20, 21].

Одне дослідження [22], показало, що алгоритми машинного навчання можуть покращити якість найму на роботу на 25%. У дослідженні використовували набір даних з понад 300 000 претендентів на роботу і порівнювали ефективність алгоритмів машинного навчання з традиційними методами найму. Результати показали, що алгоритми машинного навчання змогли визначити найбільш ефективних кандидатів з більшою точністю, що призвело до підвищення якості найму.

Компанія Pymetrics також розробила платформу для найму на основі машинного навчання, яка використовує когнітивні та емоційні оцінки для оцінки кандидатів на роботу. Платформа використовує алгоритми машинного навчання для аналізу відповідей кандидатів на різні завдання та ігри, які призначені для вимірювання їхніх когнітивних та емоційних рис. Платформа використовується кількома компаніями, включаючи Unilever, щоб допомогти визначити кандидатів, які володіють навичками та рисами, необхідними для певної ролі.

2.2 Аналіз ATS на основі штучного інтелекту

У розділі 1.3 було розглянуто ATS і згадано про існування систем, що використовують алгоритми ML для пошуку і відбору кандидатів, щоб надати рекрутерам значну перевагу в пошуку підходящих кандидатів. Розглянемо деякі з цих систем та проаналізуємо алгоритми, які вони використовують.

До однієї з найкращих ATS на 2023 рік за версію команди Forbes Advisor відноситься система Manatal, яка використовує ML алгоритми для автоматизації процесу рекрутингу. Вони використовують NLP для вилучення інформації про кандидата з резюме та профілів у соціальних мережах, а також такі алгоритми, як логістична регресія, для оцінки придатності кандидата до конкретної роботи.

Логістична регресія – це метод машинного навчання, що використовується для прогнозування категоріальних змінних, тобто для визначення ймовірності

належності об'єкту до однієї з категорій. Логістична регресія є моделлю класифікації, тобто вона прогнозує, до якої категорії належить об'єкт. У логістичній регресії використовуються логістичні функції, які перетворюють числові значення в ймовірності. Наприклад, якщо логістична регресія використовується для класифікації резюме на «підходить» і «не підходить» до вакансії, модель визначить ймовірність того, що кожне резюме належить до кожної з цих категорій. У логістичній регресії використовуються різні методи підгонки моделі до даних, такі як максимальна правдоподібність та метод мінімізації помилок. Цей метод машинного навчання часто використовується в системах відстеження кандидатів для класифікації резюме та оцінки ймовірності успішного проходження кандидатом співбесіди.

Наступною системою нашого аналізу є система SmartRecruiters, яка використовує такі алгоритми, як NLP і машинний зір, для аналізу резюме, профілів у соціальних мережах і відеоінтерв'ю, щоб визначити найкращих кандидатів на вакансію.

Машинний зір – це галузь машинного навчання та штучного інтелекту, яка займається розробкою комп'ютерних систем для розуміння та аналізу зображень та відео. Задачі машинного зору можуть бути різними: від розпізнавання обличчя, автоматичної обробки медичних зображень, до розпізнавання номерів автомобілів на дорозі або визначення розміру та форми предметів у реальному часі. Основними алгоритмами машинного зору є:

- класифікація зображень (classification) – алгоритм , що визначає, до якого класу належить зображення;
- виявлення обличчя (face detection) – алгоритм, який визначає на зображенні обличчя людини;
- виявлення руху (motion detection) – алгоритм, який визначає на зображенні рух об'єктів;
- сегментація зображень (image segmentation) – алгоритм, який дозволяє виділити на зображенні окремі об'єкти та розділити їх на окремі частини;

– оптичний потік (optical flow) – алгоритм, який дозволяє визначити напрямок та швидкість руху об'єктів на зображенні.

Workday – це хмарне програмне забезпечення для управління персоналом, яке включає модуль ATS. Вони використовують алгоритми машинного навчання, такі як дерево рішень і випадковий ліс, для аналізу даних про кандидатів і надання рекомендацій рекрутерам на основі вимог до роботи і профілів кандидатів.

Дерево рішень – це простий підхід, який використовується для розв'язання задач класифікації та регресії. Їх перевагою є легка інтерпретація та розуміння для осіб, які приймають рішення, для порівняння з їх знаннями про предметну область для перевірки та обґрунтування своїх рішень.

Наступною системою є Jobvite, яка використовує ML алгоритми для виявлення закономірностей у резюме та описах вакансій, щоб допомогти рекрутерам знайти найкращих кандидатів. Вони використовують такі алгоритми, як NLP для аналізу текстових даних та алгоритми кластеризації, щоб згрупувати схожі резюме разом.

Останньою системою нашого аналізу є iCIMS, яка використовує алгоритми машинного навчання, щоб допомогти рекрутерам знайти найкращих кандидатів на свої вакансії. Вони використовують такі алгоритми, як предиктивна аналітика, для аналізу даних про кандидатів і надання рекомендацій рекрутерам на основі кваліфікації та навичок кандидатів.

Предиктивна аналітика – це процес використання даних, статистичних алгоритмів та машинного навчання для прогнозування майбутніх результатів. Вона дозволяє аналізувати великі обсяги даних, виявляти кореляції та залежності між різними змінними та прогнозувати майбутні результати на основі цих відносин. Предиктивна аналітика може використовуватися з різними алгоритмами машинного навчання, такими як логістична регресія, дерево рішень, випадковий ліс та нейронні мережі.

Після аналізу цих систем можна стверджувати, що більшість з них використовують ML алгоритми для розв'язання задачі підбору кандидатів на вакансію, використовуючи методи класифікації та кластеризації. Ці алгоритми

дозволяють системам здійснювати аналіз резюме та інших даних про кандидатів з метою прийняття рішення про їхню придатність для конкретної вакансії. Наприклад, застосування алгоритмів класифікації дозволяє розподілити кандидатів на групи в залежності від їхньої придатності для певної посади, тоді як кластеризація дозволяє виділити спільні характеристики серед кандидатів та розподілити їх на групи за цими ознаками.

2.3 Розгляд існуючих алгоритмів інтелектуального аналізу даних

Для обробки та дослідження великих наборів даних використовують інтелектуальний аналіз даних. Метою інтелектуального аналізу даних є пошук прихованих закономірностей або раніше невідомої інформації з великих обсягів даних. Зібрана інформація є корисною для вирішення різних задач. Інтелектуальний аналіз даних може бути застосований у різних галузях, включаючи бізнес, медицину, соціальні науки, науку про дані та інші. Існує багато алгоритмів інтелектуального аналізу даних, кожен з яких призначений для вирішення конкретних задач. Ось деякі з найпоширеніших типів алгоритмів:

- алгоритми нейронної мережі – це процеси обчислення, які забезпечують функціонування нейронної мережі;
- алгоритми класифікації – алгоритми, які використовуються для розподілення об'єктів на певні категорії за їх характеристиками або ознаками;
- алгоритми регресії – алгоритми, які використовуються для передбачення безперервних числових змінних на основі інших атрибутів набору даних;
- алгоритми кластеризації – алгоритми, які використовуються для групування об'єктів у кластери в залежності від схожості між ними;
- алгоритми асоціації – алгоритми, які використовуються для виявлення зв'язків між різними характеристиками або ознаками даних у наборі даних.

Вибір типу алгоритмів інтелектуального аналізу даних залежить від ряду факторів, зокрема:

- типу даних, бо не всі алгоритми підтримують всі типи даних;

– мети аналізу;
– кількості даних, бо для великих обсягів даних можуть бути застосовані алгоритми, які використовують паралельні обчислення;
– вартості аналізу, бо використання деяких алгоритмів може бути дорогим.

В цілому, інтелектуальний аналіз даних дозволяє зрозуміти більше про дані та отримати значиму інформацію, яку можна використовувати для прийняття кращих рішень та досягнення більш ефективних результатів. Перед початком аналізу даних важливо правильно обрати алгоритм, що найкраще підходить для вирішення конкретної задачі. Для цього потрібно провести детальну оцінку потреб та розглянути характеристики різних типів алгоритмів, щоб зрозуміти, який саме може ефективно виконати поставлену задачу.

2.4 Визначення алгоритмів для експериментального дослідження

Вибір алгоритму інтелектуального аналізу для вирішення задач придатності кандидатів на роботу залежить від багатьох факторів, таких як обсяг і якість даних, вимоги до точності класифікації, час, необхідний для обробки даних та ресурси, які доступні для використання.

Одним з можливих типів алгоритмів для вирішення цієї задачі є алгоритм класифікації, який дозволяє розподілити кандидатів на різні класи залежно від їх характеристик та вимог вакансії. В останні роки одним з найпопулярніших алгоритмів для вирішення задач класифікації є алгоритми сімейства дерев рішень, які можуть бути ефективним та досить точними при застосуванні у процесі найму.

У дослідженні [23] порівнюються результати класифікації текстових документів за допомогою різних алгоритмів машинного навчання. Дерево рішень демонструє кращі показники точності. Ці алгоритми набули популярності завдяки своїй високій точності, масштабованості та здатності обробляти великі та складні набори даних, створювати високоточні моделі та надавати легко інтерпретовані та зрозумілі результати. У статтях [24-26] продемонстровано використання алгоритмів дерев рішень у різних галузях та показано їхній потенціал для підвищення точності та масштабованості моделей машинного навчання.

Однією з переваг використання алгоритмів дерев рішень у процесі найму на роботу є те, що вони можуть обробляти як категоріальні, так і безперервні дані. Це дозволяє включати в прогнозну модель широкий спектр типів даних, що підвищує точність прогнозів. Алгоритми дерев рішень також здатні обробляти відсутні дані, що є поширеною проблемою у великих наборах даних.

Однак є певні обмеження у використанні алгоритмів дерев рішень при наймі на роботу. Одне з них полягає в тому, що вони схильні до перенавчання, що може статися, коли алгоритм занадто складний і занадто точно відповідає навчальним даним. Це може призвести до поганого узагальнення та зниження точності нових даних. У статті [27] представлено запропоновані стратегії відсікання.

Незважаючи на ці обмеження, використання алгоритмів на основі дерев у процесі найму дає надію на підвищення точності прогнозування та зменшення упередженості в процесі найму. Оскільки технологія продовжує розвиватися, важливо вивчити потенційні переваги використання цих алгоритмів, а також забезпечити їх етичне та відповідальне застосування.

Розглянемо алгоритми сімейства дерев рішень для розв'язання задачі класифікації у наступному розділі.

3 ОГЛЯД СІМЕЙСТВА ДЕРЕВ РІШЕНЬ

3.1 Опис структури сімейства дерев рішень

Одним з багатьох рішень для вирішення проблем найму є використання алгоритмів сімейства дерев рішень [28], оскільки вони добре підходять для задач класифікації, що є типовим випадком використання в процесі найму. Сімейство дерев рішень – це колекція алгоритмів машинного навчання, що використовують деревоподібну структуру для розв'язання задач класифікації та регресії.

Серед найпоширеніших алгоритмів сімейства дерев рішень [28], що можуть бути використані для вирішення задач з найму, можна виділити такі:

– дерево рішень (Decision Tree);
– випадковий ліс (Random Forest);
– градієнтні дерева з підсиленням (Gradient Boosted Trees).

Ці алгоритми є популярними в машинному навчанні та зазвичай використовуються для задач класифікації та регресії. Різниця між ними полягає у тому, як вони створюють та оптимізують дерево рішень. Кожен алгоритм має свої особливості та переваги, що дозволяє обрати найбільш підходящий для вирішення конкретної задачі.

Загальна структура дерев рішень складається з наступних елементів:

– кореневий вузол – це початковий вузол, який ділиться на наступні вузли;
– вузли – це підрозділи дерева, які відображають різні характеристики або ознаки даних, від кожного вузла зазвичай відходять два або більше вузлів;
– гілки – це зв'язки між вузлами, які відображають відношення між ознаками даних та рішеннями, які необхідно прийняти;
– листові вузли – це кінцеві вузли дерева, які містять відповіді на питання або прогнози.

Структура дерева може бути багатошаровою, тобто містити кілька рівнів вузлів та гілок. Кожен шар дерева відображає нові ознаки даних та рішення, що необхідно прийняти на основі попередніх рішень.

Загальна структура випадкового лісу та градієнтних дерев також базується на структурі дерева, але вони містять декілька моделей з різними параметрами та характеристиками, що дозволяє отримувати більш точні та стійкі прогнози.

Алгоритми сімейства дерев рішень дозволяють особам, які приймають рішення, оцінювати кожного кандидата за різними критеріями. Кожне дерево рішень може мати різні критерії, такі як навички, досвід, доступність і стиль спілкування, які порівнюються для визначення найкращого кандидата.

3.2 Розгляд основних алгоритмів побудови дерев рішень

3.2.1 Дерево рішень

Алгоритм дерево рішень [29] є одним з алгоритмів машинного навчання для побудови класифікаційної моделі. Дерево рішень використовується для створення моделі, що дозволяє класифікувати або робити передбачення щодо конкретного об'єкту на основі декількох ознак. Він базується на створенні дерева, що допомагає зробити прогноз на основі порівняння вхідних даних з навчальною вибіркою. Кожен вузол у дереві представляє ознаку або атрибут, а гілка – можливе значення або результат. Алгоритм навчається на даних і створює найкращу структуру дерева для класифікації нових точок даних на основі раніше переглянутих даних.

Процес створення дерева рішень починається з кореневого вузла, який представляє найбільш значущу ознаку. Алгоритм вибирає ознаку, яка дає найбільший інформаційний приріст або індекс Джині – міру домішок у даних, і розбиває дані на підмножини на основі значень цієї ознаки. Цей процес повторюється для кожної підмножини до тих пір, поки не буде досягнуто критерію зупинки, наприклад, заданої глибини або дані не стануть занадто малими.

При використанні дерева рішень для вирішення проблем класифікації при наймі працівників, алгоритм навчається на основі набору мічених даних, що представляють попередні успішні та неуспішні найми. Потім алгоритм використовує ці дані для створення дерева рішень, яке може передбачити ймовірність успіху нових кандидатів на основі набору ознак.

Однією з переваг дерева рішень є зрозумілість. Кожен вузол у дереві можна легко інтерпретувати як рішення, засноване на певній ознаці, що дозволяє зрозуміти, як алгоритм дійшов до такої класифікації.

3.2.2 Випадковий ліс

Алгоритм випадкового лісу [30] є популярним розширенням алгоритму дерева рішень для вирішення проблем класифікації. Випадковий ліс – це ансамбль дерев рішень, де кожне дерево навчається на різній підмножині даних і різній підмножині ознак. Остаточне класифікаційне рішення приймається шляхом об'єднання результатів усіх окремих дерев.

Алгоритм випадкового лісу має кілька переваг порівняно з деревом рішень:

– зменшує ймовірність перенавчання за рахунок навчання декількох дерев на різних підмножинах даних. Це означає, що алгоритм може охопити ширший спектр закономірностей у даних, що призводить до кращої продуктивності узагальнення на нових даних;
– випадкові ліси можуть обробляти великі набори даних високої розмірності та зашумлені дані. Випадково обираючи підмножини ознак для кожного дерева, алгоритм може ефективно зменшити кількість ознак, що призводить до швидшого навчання та підвищення точності;
– випадкові ліси можна легко розпаралелити, що дозволяє ефективно обробляти великі набори даних.

Процес створення випадкового лісу передбачає навчання декількох дерев рішень, кожне з яких має випадково вибрану підмножину навчальних даних та ознак. Під час навчання кожне дерево вирощується незалежно, без обрізки або передчасної зупинки. Остаточне рішення про класифікацію приймається шляхом об'єднання результатів усіх окремих дерев, як правило, більшістю голосів.

Однією з проблем випадкових лісів є пошук оптимальної кількості дерев для використання в ансамблі. Додавання більшої кількості дерев до лісу може підвищити точність, але також збільшує обчислювальні витрати і може призвести

до надмірної підгонки. Для знаходження оптимальної кількості дерев для конкретної задачі можна використовувати перехресну перевірку та інші методи.

Отже, алгоритм випадкового лісу є потужним розширенням алгоритму дерева рішень для вирішення задач класифікації. Він має кілька переваг над окремими деревами рішень, включаючи зменшення перенавчання, підвищення точності на даних високої розмірності та ефективну паралельну обробку.

3.2.3 Градієнтні дерева з підсиленням

Дерева з градієнтним підсиленням [31] – це алгоритм машинного навчання, який використовується для побудови ансамблю дерев рішень для розв'язання задач класифікації та регресії. Цей алгоритм особливо ефективний у роботі зі складними наборами даних, які містять нелінійні зв'язки між ознаками та результатами. У порівнянні зі звичайними деревами рішень, градієнтні дерева здатні досягати більш високої точності прогнозування.

Метою цього алгоритму є створення моделі, яка мінімізує загальну похибку шляхом послідовного об'єднання прогнозів декількох дерев рішень. Алгоритм починається з навчання одного дерева рішень на всьому наборі даних. Потім обчислюється похибка між прогнозованим і фактичним результатами, і на залишковій похибці навчається друге дерево рішень. Процес повторюється для попередньо визначеної кількості ітерацій, причому кожне нове дерево навчається на залишковій похибці попереднього дерева.

У GBT внесок кожного дерева в остаточний прогноз зважується відповідно до його точності, причому більш точні дерева отримують більшу вагу. Це гарантує, що фінальна модель надає більшу вагу найточнішим деревам, покращуючи її загальну продуктивність.

Однією з ключових переваг GBT є автоматична обробка відсутніх даних та категоріальних ознак без необхідності попередньої обробки даних. Це робить їх особливо корисними для роботи зі складними реальними наборами даних, де попередня обробка даних може зайняти багато часу і призвести до помилок. Однак, GBT можуть бути схильні до надмірної підгонки, особливо коли кількість дерев в

ансамблі велика. Для запобігання перенавчання та покращення узагальнюючих характеристик моделі можуть бути використані різні методи регулювання.

Підсумовуючи, можна сказати, що дерева з градієнтним підсиленням мають низку переваг над іншими алгоритмами машинного навчання, зокрема здатність працювати з відсутніми даними та категоріальними ознаками, а також високу точність на складних наборах даних. Однак важливо забезпечити регулювання моделі, щоб запобігти надмірному перенавчанню та забезпечити її ефективність узагальнення на нових даних.

3.3 Огляд методів оцінки якості дерев рішень

Для оцінки важливості ознак та їх впливу на вихідні класи в процесі побудови моделей класифікації в машинному навчанні використовуються індекс Джині, ентропія та інформаційний приріст. Розрахунок цих показників допомагає визначити, які ознаки найбільше впливають на результат класифікації, та вибрати оптимальний набір ознак для побудови моделі. Індекс Джині та ентропія є метриками, які використовуються для вимірювання неоднорідності даних у вибірці та визначення впливу ознак на розділення даних на класи. Чим менше значення цих метрик, тим більш ефективно розділяються дані на класи. Інформаційний приріст використовується для оцінки важливості ознак в класифікаційних задачах, вимірюючи різницю між початковою та після-роздільною ентропією. Чим більший приріст інформації, тим важливіша є ознака.

Оцінка ефективності моделі дерева рішень передбачає вимірювання якості моделі за допомогою оціночних метрик [32]. Для оцінки ефективності моделі можна використовувати декілька оціночних метрик, включаючи точність, точність передбачення, повноту та F1-оцінку.

Точність, точність передбачення та повнота обчислюються безпосередньо з результатів матриці помилок і є відсотковими показниками, що базуються на абсолютному числі, яке відображається у матриці помилок.

Матриця помилок – це таблиця n на n, яка використовується для опису ефективності моделі класифікації. Кожен рядок матриці помилок представляє

фактичний клас, а кожен стовпець – передбачуваний клас. У нашій класифікації є лише дві категорії результатів – так або ні (1 або 0). Матриця помилок – це комбінація нашого прогнозу (1 або 0) і фактичного значення (1 або 0). Діаграма матриці помилок класифікаційної моделі показана на рисунку 2.

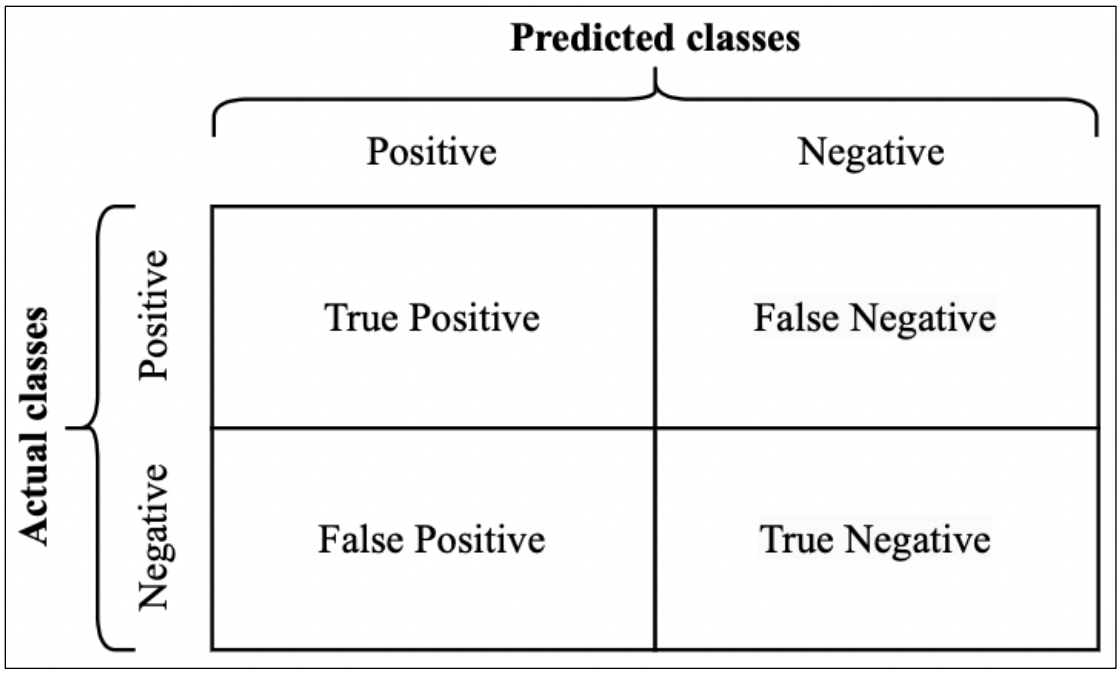

Рисунок 2 – Матриця помилок для бінарної класифікації

Для створення діаграми матриці помилок було використано інструмент draw.io [33]. Побудовану діаграму можна знайти за наступним посиланням [34].

Повнота визначається як відношення кількості істинних позитивних результатів до загальної кількості фактичних позитивних результатів.

$$Recall\ (R) = \frac{TP}{TP + FN},$$

де TP – кількість позитивних прикладів, правильно класифікованих; FN – кількість позитивних прикладів, які були помилково класифіковані моделлю як негативні.

Високий показник повноти вказує на те, що модель ефективно ідентифікує позитивні приклади, тоді як низький показник повноти, що модель може пропустити деякі позитивні приклади. Повнота особливо важлива у випадках, коли

пропуск позитивного прикладу може мати серйозні наслідки. Однак високий показник може бути досягнутий за рахунок зниження точності передбачення, оскільки модель може також ідентифікувати деякі негативні приклади як позитивні.

Точність передбачення визначається як відношення кількості істинно позитивних результатів до загальної кількості істинно позитивних результатів.

$$Precission\ (P) = \frac{TP}{TP + FP},$$

де TP – кількість позитивних прикладів, правильно класифікованих; FP – кількість негативних прикладів, які були помилково класифіковані моделлю як позитивні.

Високий показник точності передбачення вказує на те, що модель ефективно ідентифікує лише ті приклади, які є позитивними, тоді як низький показник точності передбачення означає, що модель робить велику кількість хибнопозитивних прогнозів. Точність передбачення особливо важлива у випадках, коли помилкова ідентифікація негативного прикладу як позитивного може мати серйозні наслідки, наприклад, при виявленні шахрайства або фільтрації спаму. Однак висока точність передбачення може бути досягнута за рахунок нижчого показника повноти, оскільки модель може пропустити деякі справді позитивні приклади. Тому оптимальний баланс між точністю передбачення та повнотою буде залежати від конкретних потреб.

Точність визначається як відношення кількості правильних прогнозів до загальної кількості прогнозів і являє собою частку прикладів, які модель правильно класифікувала.

$$Accuracy\ (A) = \frac{TP + TN}{TP + TN + FP + FN},$$

де TP – істинно-позитивні; TN – істинно-негативні; FP – хибно-позитивні; FN – хибно-негативні.

Високий показник точності означає, що модель ефективно прогнозує як позитивні, так і негативні приклади, тоді як низький показник точності означає, що модель робить велику кількість неправильних прогнозів. Точність є корисною метрикою, коли класи в наборі даних збалансовані, тобто кількість позитивних і негативних прикладів приблизно однакова. Однак, коли класи незбалансовані, точність може вводити в оману, і інші метрики, такі як точністю передбачення та повнота, можуть бути більш доречними.

F1-оцінка – це середнє гармонійне значення точності передбачення та повноти, і використовується для оцінки загальної ефективності моделі в правильному визначенні позитивних прикладів при мінімізації кількості помилкових результатів та хибних результатів.

$$F1\ score = 2\frac{R * P}{R + P},$$

де R – значення пригадування; P – значення точності.

Високий показник F1-оцінки вказує на те, що модель ефективно ідентифікує позитивні приклади, мінімізуючи помилкові та хибнонегативні результати.

F1-оцінка особливо корисна, коли класи в наборі даних незбалансовані, тобто один клас має значно більше екземплярів, ніж інший. У цьому випадку високий показник точності може ввести в оману, оскільки модель може просто передбачити клас більшості. З іншого боку, F1-оцінка враховує як точність передбачення, так і повноту, і дає більш збалансовану оцінку роботи моделі.

Також, необхідно оцінити час навчання моделі, яка важлива з кількох причин: розподіл ресурсів, оптимізація продуктивності, оптимізація витрат.

Таким чином, оцінка ефективності побудованої моделі передбачає поєднання кількісних та якісних методів. Важливо використовувати кілька метрик і методів, щоб отримати повне розуміння ефективності моделі.

4 ВИЗНАЧЕННЯ ДЖЕРЕЛА ДАНИХ ТА МЕТОДІВ ЇХ ЗБОРУ

Збір та підготовка даних є важливим кроком у прогнозуванні придатності кандидатів у процесі найму на роботу. Це допомагає забезпечити точні прогнози, уникнути упередженості та приймати більш обґрунтовані рішення найму, що призводить до кращих результатів як для компанії, так і для кандидатів.

Щоб вирішити проблему класифікації при наймі за допомогою алгоритмів машинного навчання, необхідно зібрати відповідні дані про кандидатів на роботу і належним чином їх позначити. Нижче наведено кілька прикладів даних, які можна використовувати для вирішення цієї проблеми:

- базова особиста інформація про кандидатів на роботу (ім'я, вік, стать і контактні дані);
- дані про освіту (ступінь, спеціальність та навчальний заклад);
- дані про попередній досвід роботи (компанії, в яких вони працювали, їхні посади та посадові обов'язки);
- дані про навички (мова програмування, інструменти та сертифікати);
- дані про поведінку кандидатів на співбесідах (їхні відповіді на конкретні запитання або загальне враження);
- дані зі стандартизованих оцінок, які оцінюють конкретні навички або риси, що мають відношення до роботи.
- рекомендації, які можуть включати відгуки та рекомендації від попередніх роботодавців або колег, які можуть дати уявлення про трудову етику кандидата, його комунікативні навички та інші важливі фактори.

Одним з найпоширеніших способів отримання вищезгаданих даних є збір інформації з резюме за допомогою технології «Resume Parsing» [35].

Резюме – це документ, який містить інформацію про професійний досвід, навички, освіту, досягнення та інші важливі дані, що характеризують кандидата на роботу. Зазвичай, резюме є першим етапом у відборі кандидатів на вакансію, і його використання дозволяє підприємству швидко ознайомитися з кандидатом та визначити його придатність для вакантної посади. Резюме може бути складеним як

у вільній формі, так і за допомогою спеціальних сервісів для його автоматизованого створення. Важливо, щоб резюме було чітким, зрозумілим та містило лише необхідну та достовірну інформацію.

Розбір резюме (Resume Parsing) – це процес автоматичного виділення та структурування різних даних резюме, такі як освіта, досвід роботи, контактні дані тощо, для того, щоб використовувати ці дані для подальшої обробки, наприклад, в системах відстеження кандидатів (ATS). Цей процес може бути побудовано за допомогою алгоритмів машинного навчання, що допомагає скоротити час, необхідний для обробки великої кількості резюме, та зменшити ймовірність помилок, пов'язаних з ручним введенням даних. Але, необхідно врахувати деякі недоліки та обмеження, що можуть виникнути при автоматичному аналізі резюме:

– відсутність повної інформації;
– низька точність, що може призвести до неправильного аналізу кандидатів;
– мова резюме може відрізняються від тих, на яких програма була навчена;
– при використанні нестандартного формату резюме можуть виникнути проблеми з правильним аналізом документу.

На рисунку 3 схематично зображено процес розбору резюме.

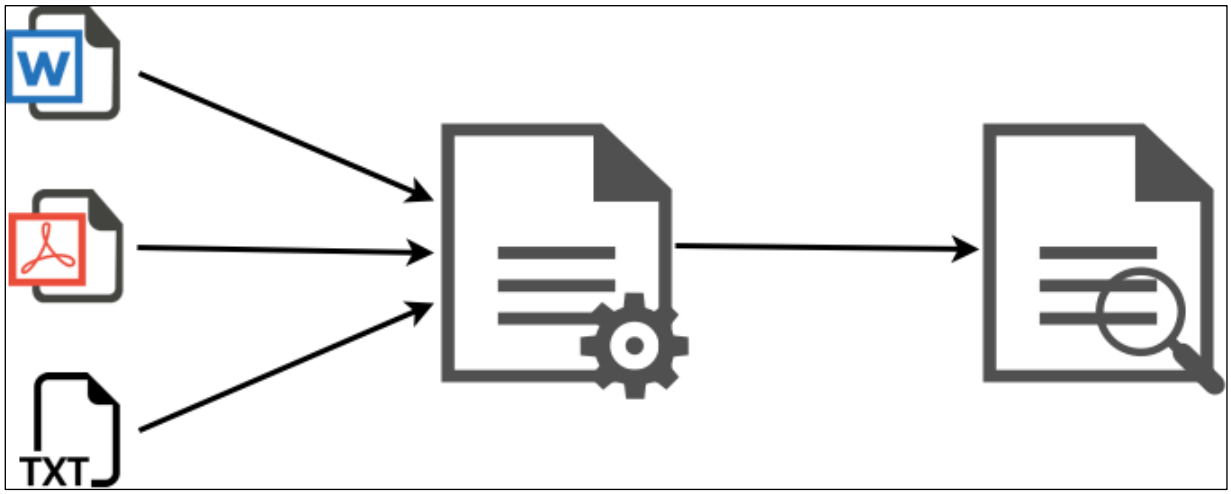

Рисунок 3 – Процес розбору резюме (рисунок виконаний самостійно)

Схема процесу складається з декількох етапів. Спочатку відбувається отримання і завантаження резюме кандидата в систему. Формат резюме може мати

різний вигляд, але найпоширеніші формати це DOCS-документ, PDF-файл або TXT-файл. Після цього текст резюме піддається обробці та аналізу. За допомогою ML та алгоритмів NLP система визначає ключові слова та фрази, пов'язані з освітою, досвідом роботи, навичками кандидата тощо. Після того, як інформація з резюме була екстрагована, вона зберігається у базі даних. Далі ця інформація може бути використана для більш детального аналізу потенційних кандидатів.

Синтаксичний аналіз резюме є ключовим етапом при аналізі резюме. Для синтаксичного аналізу використовуються методи та техніки обробки природної мови (NLP). Однією з ключових задач NLP є розуміння та генерація тексту. Для досягнення цих задач використовуються методи машинного навчання, статистичні методи та правила лінгвістики.

Існує велика кількість методів, що дозволяють розібрати та проаналізувати резюме. Нижче наведено три основні методи для синтаксичного аналізу резюме:

– статистичний синтаксичний аналіз – застосування числових моделей для аналізу структури резюме;
– розбір резюме за ключовими словами – сканування резюме на предмет наявності спеціальних і заздалегідь визначених фраз і слів, які відповідають оригінальному опису вакансії;
– граматичний синтаксичний аналіз – використовує граматичні правила під час сканування резюме, щоб забезпечити контекст для фраз і слів.

Після того, як ці дані будуть зібрані, попередньо оброблені та очищені, їх можна використовувати для побудови прогностичної моделі, яка допоможе визначити кандидатів, які, найімовірніше, добре підходять для конкретної вакансії.

5 МЕТОДИКА ПРОВЕДЕННЯ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ

5.1 Визначення вхідних даних для експериментальної частини

Для проведення експериментального дослідження було обрано набір даних [36], що включає різні атрибути 614 кандидатів. Можливі значення та детальний опис усіх атрибутів набору даних наведено у таблиці 1.

Таблиця 1 – Опис атрибутів набору даних з можливими значеннями

Атрибут	Опис	Можливе значення
Serial_no	Ідентифікаційний номер особи	1 – 614
Gender	Ідентичність, стать	Male/Female/Others
Python_exp	Досвід роботи з мовою програмування Python	Yes/No/Undefined
Experience_Years	Досвід роботи	0 – 3
Education	Освіта	Graduated/Not Graduated
Internship	Досвід стажування	Yes/No/Undefined
Score	Оцінка	0 – 700
Salary*10E4	Заробітна плата	150 – 81k
Offer_History	Наявність пропозицій	Yes/No/Undefined
Location	Місцезнаходження	L1/L2/Others
Recruitment_Status	Рішення про прийняття на роботу	Y/N

Після аналізування атрибутів набору даних можна стверджувати, що не всі з них корисні для проведення експерименту, тому їх можна виключити на етапі підготовки даних. Атрибути, такі як порядковий номер, оцінка, заробітна плата та місцезнаходження, не будуть мати вплив на прогноз.

Атрибутом результату є результат рішення про прийняття на роботу (статус рекрутингу). Кількісні характеристики обраного набору даних за результуючим атрибутом наведено у таблиці 2.

Таблиця 2 – Кількісні характеристики набору даних

Показник	Значення
Загальна кількість прикладів	614
Кількість позитивних прикладів	422 (69%)
Кількість негативних прикладів	192 (31%)

З таблиці 2 видно, що кількість позитивних прикладів перевищує кількість негативних прикладів. Набір даних, використаний для експериментальної частини нашого дослідження, можна знайти за наступним посиланням [36].

5.2 Підготовка до проведення експериментів

Основним завданням експериментальної частини дослідження є навчання моделей за допомогою обраних алгоритмів сімейства дерев рішень та аналіз ефективності їх застосування для вирішення задачі придатності кандидатів на роботу.

Для проведення експерименту було розроблено програмне забезпечення, використовуючи мову програмування Java з використанням Spark Framework [37].

Мова програмування Java була обрана через її надійність, незалежність від платформи та потужну підтримку розподілених обчислень, що робить її придатною для великомасштабної обробки даних за допомогою Spark Framowork.

Spark Framework – інструмент, призначений для ефективної паралельної та розподіленої обробки великих обсягів даних. Spark Framowork можна використовувати для обробки даних, що зберігаються на багатьох різних комп'ютерах у мережі, що дозволяє їй горизонтально масштабуватися і обробляти надзвичайно великі набори даних. Він має кілька вбудованих бібліотек для таких завдань, як машинне навчання, обробка графіків і потокове передавання даних, що

робить його універсальним інструментом для широкого спектру завдань з обробки даних. Spark має зручний API на декількох мовах програмування, включаючи Scala, Python і Java [38], що дозволяє розробникам легко використовувати та інтегрувати його в існуючі конвеєри обробки даних. Він також добре інтегрується з іншими інструментами для роботи з великими даними, забезпечуючи комплексну та інтегровану платформу для роботи з великими даними.

Процес навчання моделі класифікації з використанням алгоритмів дерев рішень включав декілька етапів. На рисунку 4 наведено експериментальний робочий процес для етапу навчання, що забезпечує загальний огляд процесу.

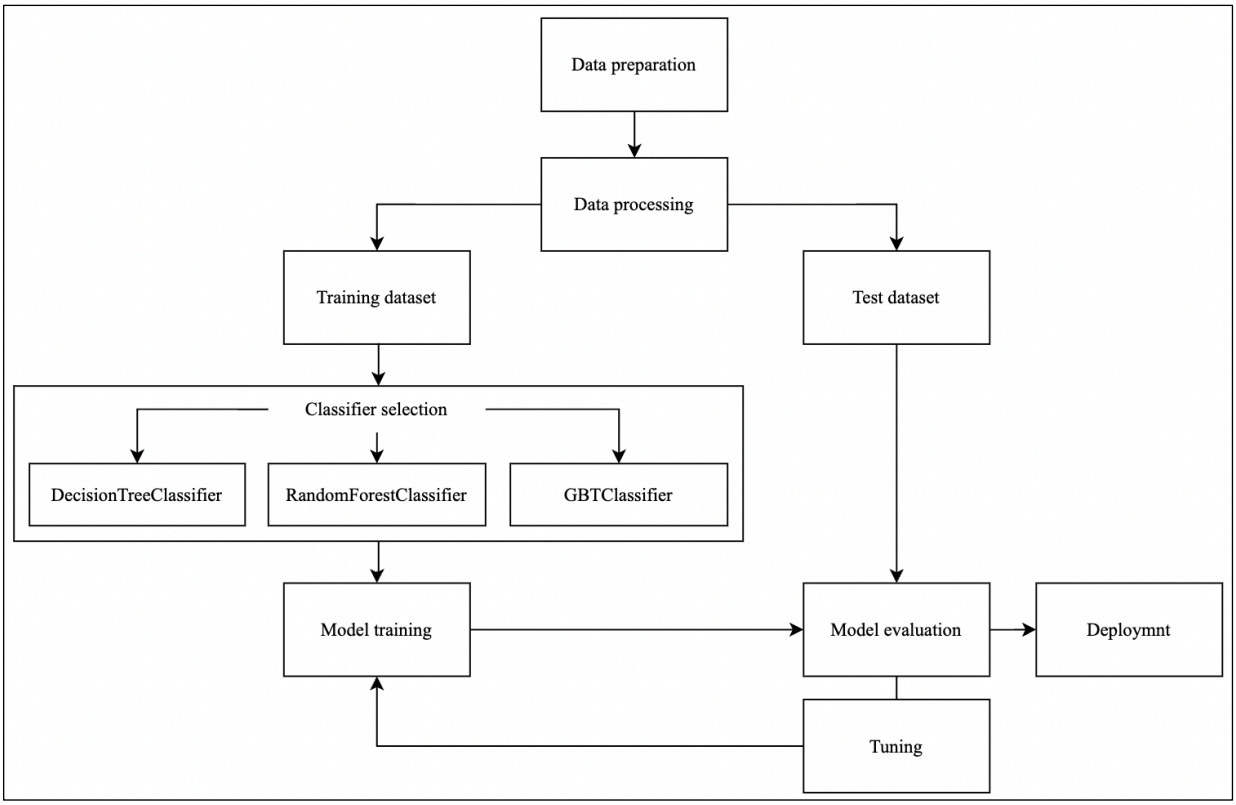

Рисунок 4 – Процес проведення експерименту

Діаграма побудована за допомогою інструменту draw.io [33] і доступна за посиланням [39].

Першим кроком є підготовка даних для моделі шляхом очищення та обробки даних. Потім імпортуються необхідні бібліотеки Spark і завантажуються дані у Spark DataFrame. Далі дані попередньо обробляються за допомогою API

машинного навчання Spark для масштабування, нормалізації та кодування категоріальних змінних.

Наступним кроком є етап нормалізації даних для того, щоб забезпечити точні та узгоджені результати. Нормалізація даних – це процес приведення всіх значень атрибутів до певного бажаного діапазону, щоб гарантувати, що кожна функція робить однаковий внесок в аналіз. Алгоритми сімейства дерев рішень чутливі до відмінностей у шкалах і розподілах характеристик даних, що може призвести до некоректної поведінки алгоритму і недостовірних результатів. Нормалізуючи дані, можна пом'якшити ці проблеми і підвищити точність та ефективність алгоритмів сімейства дерев рішень.

У таблиці 3 надана інформація про ненормовані атрибути вхідного набору даних, яка включає їх початкові значення та відповідні нормалізовані значення.

Таблиця 3 – Ненормалізовані атрибути з початковими та нормалізованими значеннями

Атрибут	Початкове значення	Нормалізоване значення
Gender	Male/Female/Others	1/0/2
Python_exp	Yes/No/Undefined	1/0/2
Education	Graduated/Not Graduated	1/0
Internship	Yes/No/Undefined	1/0/2
Offer_History	Yes/No/Undefined	1/0/2
Recruitment_Status	Y/N	1/0

Перед початком навчання моделі за допомогою одного з алгоритмів дерев рішень необхідно розділити набір даних на два окремі набори – навчальний та тестовий. Навчальний набір даних використовується для навчання моделі класифікації, тоді як тестовий набір даних використовується для перевірки ефективності та точності прогнозування. Для експерименту набір даних було поділено на навчальний та тестовий набори у пропорції 80% і 20% відповідно.

Такий підхід дозволяє зменшити ризик перенавчання моделі та забезпечити її роботу на нових, раніше не бачених даних.

Кількісні характеристики наборів даних детально описано у таблиці 4.

Таблиця 4 – Кількісні характеристики набору даних

Показник	Навчальні дані	Дані тестування	Загалом
Відсоткове співвідношення (%)	80	20	100
Загальна кількість прикладів	498	116	614
Кількість позитивних прикладів	343	79	422
Кількість негативних прикладів	155	37	192

Розділивши дані на навчальний та тестовий набори, модель можна навчати на навчальному наборі даних. Модель використовує дані з навчального набору даних, щоб навчитися робити прогнози. Атрибутом результату набору даних було обрано статус рекрутингу (результат рішення про прийняття на роботу), який визначається двома групами – «так» або «ні».

Після підготовки даних обираються алгоритми сімейства дерев рішень, такі як дерево рішень, випадковий ліс або дерева з градієнтним підсиленням. Потім обрана модель навчається на попередньо оброблених даних за допомогою бібліотеки Spark's MLlib, де вказуються параметри моделі та алгоритм навчання.

Після того, як модель навчено, її можна використовувати для прогнозування на тестовому наборі даних. Прогнози можна порівняти з фактичними значеннями на тестовому наборі даних, щоб оцінити ефективність моделі. Ефективність моделі оцінюється за допомогою таких метрик, як точність, точність передбачення, повнота та F1-оцінка.

Після аналізу оціночних показників модель може бути доопрацьована для покращення її продуктивності. Цей процес може включати коригування параметрів моделі, вибір або вилучення функцій, а також вивчення різних алгоритмів для пошуку найбільш ефективних.

Нижче описано кілька вхідних параметрів, які були використані для навчання моделі класифікації за допомогою алгоритмів дерева рішень:

– MaxDepth, що визначає максимальну глибину дерева (використовується в усіх алгоритмах дерева рішень, за замовчуванням 5);
– MaxBins, що визначає максимальну кількість бінів, які використовуються для дискретизації неперервних ознак (використовується в усіх алгоритмах дерева рішень, за замовчуванням 5);
– Impurity – міра домішок, що використовується для побудови дерева (значення за замовчуванням «gini» для дерев рішень та випадкових лісів, і «variance» для дерев з градієнтним підсиленням);
– NumTrees – використовується для випадкових лісів і контролює кількість дерев у лісі (за замовчуванням 20);
– FeatureSubsetStrategy – використовується для випадкових лісів і контролює кількість ознак, які слід враховувати при пошуку найкращого розбиття у кожному вузлі (за замовчуванням «auto»);
– MaxIter – використовується для градієнтних дерев і контролює максимальну кількість ітерацій у моделі (за замовчуванням 20);
– StepSize – використовується для дерев з градієнтним підсиленням і контролює розмір кроку для кожної ітерації моделі (за замовчуванням 0.1).

Важливо зазначити, що ефективність кожного алгоритму сильно залежить від конкретних змінних. Тому важливо налаштовувати ці параметри для оптимізації роботи моделі.

5.3 Навчання класифікаційних моделей

На основі обраних алгоритмів сімейства дерев рішень було розроблено програмне забезпечення для навчання моделі за обраним алгоритмом та отримання результатів експериментальної частини дослідження. Для навчання класифікаційних моделей було обрано три різні алгоритми дерев рішень: дерево рішень, випадковий ліс та градієнтні дерева з підсиленням. Поточна версія (3.3.1) бібліотеки Spark's MLlib надає два типи мір домішок для класифікаторів дерева

рішень та випадкового лісу – індекс Джині та ентропія. Тому, виходячи з цих умов, для експерименту було навчено наступні 5 моделей з відповідними параметрами:

- модель дерева рішень (impurity: Gini index, max depth: 5, max bins: 32, min instances per node: 1, min weight fraction per node: 0, min info gain: 0, max memory in MB: 256);
- модель дерева рішень (impurity: entropy, max depth: 5, max bins: 32, min instances per node: 1, min weight fraction per node: 0, min info gain: 0, max memory in MB: 256);
- модель випадкового лісу (impurity: Gini index, max depth: 5, max bins: 32, number of trees: 20, min instances per node: 1, min weight fraction per node: 0, min info gain: 0);
- модель випадкового лісу (impurity: entropy, max depth: 5, max bins: 32, number of trees: 20, min instances per node: 1, min weight fraction per node: 0, min info gain: 0);
- модель дерев з градієнтним підсиленням (max depth: 5, max bins: 32, min instances per node: 1, min weight fraction per node: 0, min info gain: 0, max memory in MB: 256, max iteration: 20, loss type: logistic, step size: 0.1, subsampling rate: 1).

Також, для кожної класифікаційної моделі було виміряно час, необхідний для її навчання. Отримані результати наведені у таблиці 5.

Таблиця 5 – Результати часу навчання моделі

Модель	Час навчання, мс
Дерево рішень (індекс Джині)	1846
Дерево рішень (ентропія)	803
Випадковий ліс (індекс Джині)	964
Випадковий ліс (ентропія)	789
Дерева, що ростуть на градієнті	1728

Як показано у таблиці 5, час навчання, необхідний для кожного алгоритму, суттєво відрізняється, причому найшвидшим є алгоритм випадкового лісу, а дерева рішень та дерева з градієнтним підсиленням потребують більше часу.

5.4 Опис методу оцінки ефективності

Для оцінки ефективності класифікаційних моделей, побудованих за допомогою алгоритмів сімейства дерев рішень, використали метрики якості, що описані у розділі 3.3, а саме: точність (accuracy), точність передбачення (precision), повноту (recall) та F1-оцінку (F1-score). Також, важливим показником є час навчання моделі, який слід враховувати при виборі алгоритму для аналізу даних, оскільки він може суттєво вплинути на ефективність та швидкість аналізу. Для зальної оцінки якості моделі використали суму цих оціночних показників.

Після навчання класифікаційних моделей було проведено серію експериментів для кожного з алгоритмів з випадковим генеруванням декількох навчальних і тестових наборів даних та розраховано мінімальні, середні та максимальні значення метрик на тестовому наборі даних.

Таким чином, оцінка ефективності побудованої класифікаційної моделі поєднує кількісні та якісні методи. Важливо використовувати кілька метрик і методів, щоб отримати повне розуміння ефективності моделі та визначити сфери для вдосконалення.

6 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ

Розглянемо результати експерименту. Розбиття набору даних та оцінка точності класифікаційної моделі були виконані 20 разів для кожного з алгоритмів, щоб отримати статистично достовірні результати. Нижче наведено мінімальне, середнє та максимальне значення оціночних показників для кожної побудованої класифікаційної моделі.

Результати експериментів, отримані при тестуванні моделі дерева рішень на тестовому наборі даних, наведено у таблицях 6 і 7. Таблиця 6 показує точність, точність передбачення, повноту та F1-оцінку моделі дерева рішень, навченої за допомогою критерію індексу Джині.

Таблиця 6 – Оціночні показники моделі дерева рішень (індекс Джині)

Метрика	Мінімальне	Середнє	Максимальне
Час навчання, мс	1408	1846	2003
Точність, %	0.71	0.7379	0.768
Точність передбачення, %	0.861	0.89	0.91
Повнота, %.	0.146	0.2352	0.243
F1-оцінка, %	0.6	0.6817	0.72

Як показано у таблиці 6, час навчання моделі становить 1,846 секунд. Модель досягла загальної точності 73,8%, що означає, що вона змогла правильно передбачити мітки класів для 73,8% прикладів у тестовому наборі даних. Крім того, модель досягла точності передбачення 89% і повноти 23,5%, що означає, що вона змогла правильно передбачити 89% позитивних прикладів і уникнути помилкових результатів. Крім того, наша модель досягла F1-оцінки у 68,2%, що свідчить про те, що вона досить хороша як за точністю передбачення, так і за повнотою.

Загалом, модель дерева рішень досягла достатньої точності та F1-оцінки, з високою точністю передбачення, але відносно низьким показником повноти. Це свідчить про те, що модель добре ідентифікує позитивні випадки, але, можливо, пропустила значну кількість позитивних випадків у наборі даних.

У таблиці 7 наведено результати оціночних метрик для моделі дерева рішень, навченої за допомогою критерію ентропії.

Таблиця 7 – Оціночні показники моделі дерева рішень (ентропія)

Метрика	Мінімальне	Середнє	Максимальне
Час навчання, мс	728	803	944
Точність, %	0.689	0.7475	0.782
Точність передбачення, %	1	1	1
Повнота, %.	0.21	0.2353	0.252
F1-оцінка, %	0.667	0.6895	0.691

Як показано у таблиці 7, час навчання моделі становить 0,803 секунди. Модель досягла загальної точності 74,75%, з оцінками точності передбачення та повноти 100% та 23,5% відповідно, а F1-оцінка склала 68,95%.

Загалом, результати показують, що модель є відносно точною, але має проблеми з ідентифікацією позитивних випадків. Високий показник точності передбачення вказує на те, що модель добре ідентифікує справді позитивні результати, але низький показник повноти те, що вона може пропускати значну кількість реальних позитивних випадків.

За результатами оцінки ефективності класифікаційної моделі дерева рішень з використанням індексу Джині та критерію ентропії, можна зробити висновок, що обидва критерії дають приблизно однакові результати кожної з метрик. Однак, дещо кращий результат було отримано при використанні критерію ентропії.

Загалом, експеримент з використанням алгоритму дерева рішень був успішним у створенні моделі, яка забезпечує досить непогану точність класифікації і швидкість навчання.

Результати експериментів, отримані при тестуванні класифікаційної моделі випадкового лісу на тестовому наборі даних, наведено у таблицях 8 і 9. У таблиці 8 наведено результати оціночних метрик для моделі випадкового лісу, навченої за критерієм індексу Джині.

Таблиця 8 – Оціночні показники моделі випадкового лісу (індекс Джині)

Метрика	Мінімальне	Середнє	Максимальне
Час навчання, мс	768	789	846
Точність, %	0.724	0.7766	0.78
Точність передбачення, %	1	1	1
Повнота, %.	0.29	0.3235	0.342
F1-оцінка, %	0.712	0.7355	0.745

Як показано у таблиці 8, час навчання моделі становить 0,789 секунди. Модель досягла загальної точності 77,7%, з оцінками точності передбачення та повноти 100% та 32,4% відповідно, а F1-оцінка склала 73,6%.

Наведені результати свідчать про те, що модель працює досить добре, але їй важко точно ідентифікувати позитивні випадки. Хоча показник точності передбачення є високим, що означає, що модель добре прогнозує справді позитивні результати, низький показник повноти свідчить про те, що модель пропускає значну кількість реальних позитивних випадків.

У таблиці 9 наведено результати оціночних метрик для моделі випадкового лісу, навченої за критерієм ентропії.

Таблиця 9 – Оціночні показники моделі випадкового лісу (ентропія)

Метрика	Мінімальне	Середнє	Максимальне
Час навчання, мс	889	964	1001
Точність, %	0.767	0.7864	0.8
Точність передбачення, %	1	1	1
Повнота, %.	0.32	0.3529	0.36
F1-оцінка, %	0.722	0.75	0.768

Як показано у таблиці 9, час навчання моделі становить 0,964 секунди. Модель досягла загальної точності 78,6%, з оцінками точності передбачення та повноти 100% і 35,3% відповідно, а F1-оцінка склала 75%.

Загалом, результати показують, що модель випадкового лісу є відносно точною, але все ще має проблеми з ідентифікацією позитивних випадків. Високий показник точності передбачення вказує на те, що модель добре ідентифікує справжні позитивні результати, але відносно низький показник повноти те, що вона все ще може пропускати значну кількість реальних позитивних випадків.

Результати оцінки ефективності моделі випадкового лісу з використанням індексу Джині та критерію ентропії також дали дуже схожі результати, з дещо кращими результатами для критерію ентропії. Загалом, експеримент з використанням алгоритму випадкового лісу також виявився успішним у створенні моделі, яка забезпечує досить непогану точність класифікації і швидкість навчання.

Таблиця 10 показує результати оціночних метрик для навченої моделі дерев з градієнтним підсиленням.

Таблиця 10 – Оціночні показники моделі дерев з градієнтним підсиленням

Метрика	Мінімальне	Середнє	Максимальне
Час навчання, мс	1244	1728	2104
Точність, %	0.689	0.7087	0.72
Точність передбачення, %	0.722	0.75	0.768
Повнота, %.	0.1721	0.1764	0.1789
F1-оцінка, %	0.625	0.6417	0.68

Як показано в таблиці 10, час навчання моделі становить 1,728 секунди. Модель досягла загальної точності 70,9%, з оцінками точності передбачення та повноти 75% та 17,6%, відповідно, та F1-оцінкою 64,2%.

Таким чином, модель досягла помірного рівня точності, але вона має труднощі з ідентифікацією позитивних випадків, про що свідчить низький показник повноти. Оцінка точності передбачення є кращою, що свідчить про те, що модель добре ідентифікує справді позитивні приклади.

Загалом, експеримент з використанням алгоритму градієнтних дерев з підсиленням також був успішним у створенні моделі.

7 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

У таблицях 6-10 наведено результати експериментальної частини дослідження, і показано, що використання алгоритмів сімейства дерев рішень продемонструвало багатообіцяючі рівні ефективності з точки зору точності, точності передбачення та повноти. Точність алгоритмів була високою, що свідчить про те, що моделі змогли правильно класифікувати придатність великої кількості кандидатів на вакансію.

Перш ніж аналізувати ефективність алгоритмів сімейства дерев рішень, ми оцінили вплив домішок для нашого завдання класифікації. Індекс Джині та критерій ентропії – це два поширені показники, які використовуються для оцінки домішок у деревах рішень. Порівняння цих двох мір показало, що вони мають незначний вплив на ефективність наших деревовидних моделей, причому критерій ентропії дещо випереджає індекс Джині. Ці результати свідчать про те, що будь-яка з цих мір може бути ефективно використана для вирішення задач класифікації.

На основі порівняння моделей класифікації, алгоритм випадкового лісу є найефективнішим з точки зору досягнення вищої загальної точності, точності передбачення, повноти та F1-оцінки, перевершуючи за всіма цими показниками як дерево рішень, так і алгоритм градієнтних дерев з підсиленням. Із загальною точністю 78,6% модель випадкового лісу досягла найвищого показника точності серед трьох моделей. Показники точності передбачення та повноти для моделі випадкового лісу також були вищими, ніж у двох інших алгоритмів: показник точності передбачення склав 100%, а показник повноти – 35,3%. Для порівняння, алгоритм дерева рішень досягнув 100% точності та 23,5% повноти, в той час як градієнтні дерева з підсиленням досягли 75% точності та 17,6% повноти.

Однак час навчання, необхідний для кожного алгоритму, суттєво відрізняється: дерево рішень є найшвидшим, а випадковий ліс і градієнтні дерева з підсиленням потребують більше часу. В той час як алгоритм дерева рішень є найшвидшим і займає лише 0,803 секунди для навчання моделі, алгоритми випадкового лісу та градієнтних дерев займають 0,964 секунди та 1,728 секунди відповідно.

Проаналізувавши результати, можна сказати, що вибір алгоритму залежить від конкретних потреб. Якщо час навчання є критичним фактором, а висока ефективність не настільки важлива, алгоритм дерева рішень може бути підходящим варіантом. Однак, якщо потрібен найвищий рівень точності передбачення та повноти, кращим вибором буде алгоритм випадкового лісу.

Важливо зазначити, що алгоритм градієнтних дерев з підсиленням також може бути хорошою альтернативою алгоритму випадкового лісу. Наприклад, якщо час навчання не такий критичний, але є потреба в пріоритеті точності передбачення над повнотою, то градієнтні дерева з підсиленням можуть бути кращим вибором, ніж випадкові ліси.

Хоча результати цих експериментів показують, що алгоритм випадкового лісу є найефективнішим серед трьох алгоритмів у процесі навчання, варто також зазначити, що розмір і якість набору даних, який використовувався в експериментах, могли вплинути на ефективність результатів алгоритмів. Різницю в порівнянні результатів, отриманих в трьох експериментах, можна вважати незначною, тому використання будь-якого з розглянутих алгоритмів було успішним при створенні моделі.

Результат експериментальної частини дослідження надав цінну інформацію про те, наскільки ефективними можуть бути алгоритми дерев рішень для допомоги в пошуку та відборі правильних кандидатів на вакансію. Найкраще використання моделей залежить від компромісу між часом навчання, точністю, точністю передбачення і повнотою, а також від конкретних вимог. Алгоритм дерева рішень може бути підходящим варіантом для додатків, де час навчання є критично важливим, тоді як алгоритму випадкового лісу слід надавати перевагу для додатків, де потрібен найвищий рівень точності та повноти. Алгоритм градієнтних дерев з підсиленням може бути хорошою альтернативою алгоритму випадкового лісу в деяких ситуаціях, залежно від конкретних потреб.

8 ОГЛЯД МОЖЛИВОСТЕЙ ПОДАЛЬШОГО ДОСЛІДЖЕННЯ

Проведення цього дослідження створює основу для майбутніх досліджень у сфері рекрутингу, що можуть бути корисними для покращення процесу найму кандидатів на роботу. Оскільки алгоритми сімейства дерев рішень продовжують розвиватися та стають все більш досконалими, вони можуть бути використані для вирішення різних проблем у сфері рекрутингу, таких як забезпечення рівної можливості для всіх кандидатів та прогнозування їхньої продуктивності на роботі. Для досягнення цих цілей необхідні подальші дослідження в області ефективності алгоритмів сімейства дерев рішень в різних контекстах та за різних умов.

Хоча алгоритми сімейства дерев рішень показали непогані результати у експериментальній частині дослідження, проте використання цих алгоритмів має кілька недоліків, що викликають потребу в подальших дослідженнях для вдосконалення їх ефективності та розуміння кращого способу їх застосування у процесі найму на роботу. Нижче наведено деякі недоліки:

– недостатня точність при роботі зі складними даними або великими наборами даних;
– дерева рішень не завжди враховують взаємозв'язки між різними атрибутами і можуть недооцінювати їх вплив на придатність кандидатів;
– побудова моделей на основі алгоритмів сімейства дерев рішень може вимагати значних витрат часу та ресурсів, особливо при роботі з великими наборами даних;
– для досягнення оптимальної ефективності використання алгоритмів сімейства дерев рішень необхідна експертна підготовка даних;
– інтерпретація результатів може бути складною задачею, що може ускладнювати прийняття рішень на практиці.

Тому дослідження підкреслює потенціал для проведення подальших досліджень у цій галузі з метою подальшого вдосконалення цих алгоритмів та їх ефективного застосування у практичних задачах.

ВИСНОВКИ

У результаті виконання роботи було досліджено можливість підвищення ефективності процесу прийняття рішень при відборі кандидатів з великої кількості даних за допомогою застосування машинного навчання, а саме алгоритмів сімейства дерев рішень. Для цього були виконані наступні завдання:

- проаналізовано існуючі методи прийняття рішень при відборі кандидатів для команди та системи відстеження кандидатів на основі штучного інтелекту;
- проаналізовано та визначено алгоритми сімейства дерев рішень для проведення експериментального дослідження;
- визначено метрики для оцінки ефективності алгоритмів сімейства дерев рішень;
- проаналізовано попередній збір та обробку даних про кандидатів;
- розроблено ПЗ для проведення експериментального дослідження;
- виміряно ефективність застосування кожного з обраних алгоритмів за допомогою оціночних метрик та проаналізовано отримані результати;
- визначено недоліки використання алгоритмів сімейства дерев рішень.

Це дослідження визначило різні методи та алгоритми, які можуть бути використані для прийнятті рішень при наймі кандидатів у команду. Було проведено кілька експериментів з використанням різних алгоритмів машинного навчання, таких як дерево рішень, випадковий ліси та градієнтні дерева з підсиленням. Експерименти проводилися на наборі даних, який був розділений на навчальний (80%) і тестовий (20%). Навчальний набір даних використовувався для навчання моделі, а тестовий набір даних використовувався для оцінки та порівняння моделей, які були навчені за наведеними вище алгоритмами, між собою. Ефективність моделей оцінювалася за допомогою декількох метрик, таких як час обробки, точність, точність передбачення, повнота та F1-оцінка.

Результати експериментів показали, що індекс Джині та критерій ентропії мали незначний вплив на ефективність наших класифікаційних моделей, причому ентропія дещо випереджала Джині. Це свідчить про те, що обидва критерії можуть

бути ефективно використані для нашої задачі класифікації. Порівняння класифікаційних моделей показало, що алгоритм випадкового лісу досягнув найвищої загальної точності, точності передбачення, повноти та F1-оцінки серед трьох протестованих алгоритмів. Однак час навчання, необхідний для кожного алгоритму, суттєво відрізняється, причому дерево рішень є найшвидшим, а випадковий лісу та дерева з градієнтним підсиленням потребують більше часу.

Хоча алгоритми сімейства дерев рішень показали непогані результати у експериментальному дослідження, проте використання цих алгоритмів має кілька недоліків. Для вирішення цих недоліків необхідні подальші дослідження в області вдосконалення ефективності алгоритмів сімейства дерев рішень у різних контекстах та за різних умов.

Крім того, важливо зазначити, що алгоритми машинного навчання не повинні використовуватися для прийняття остаточного рішення про найм самостійно, а скоріше як один із методів у процесі найму. Людське судження та досвід завжди повинні бути частиною процесу прийняття рішень, оскільки алгоритми можуть надавати прогнози лише на основі даних, на яких вони були навчені, і можуть бути не в змозі повністю врахувати такі фактори, як культурна придатність або м'які навички.

Результати дослідження було представлено на VII Міжнародній конференції з комп’ютерної лінгвістики та інтелектуальних систем CoLInS-2023 [1], яка індексується в наукометричних базах CEUR і Scopus.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kirill Smelyakov, Yuliia Hurova, Serhii Osiievskyi. "Analysis of the Effectiveness of Using Machine Learning Algorithms to Make Hiring Decisions", 7th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2023), April 20–21, 2023. – CEUR-WS, 2023, ISSN 1613-0073. – Volume 3387, PP. 77-92.

2. Your Hiring Pulse report for April 2023. URL: https://resources.workable.com/stories-and-insights/hiring-pulse (дата звернення: 25.01.2023).

3. 17 Effective Employee Selection Methods To Consider. URL: https://www.indeed.com/career-advice/career-development/employee-selection-methods (дата звернення: 29.01.2023).

4. Want To Be Noticed By Recruiters? Try This Resume Strategy To Get Through The Applicant Tracking System. URL: https://www.forbes.com/sites/robinryan/2021/02/09/resume-keywords-and-the-applicant-tracking-system-atswhat-you-need-to-know/?sh=2f26edab4bcc (дата звернення: 01.02.2023).

5. What Is an Applicant Tracking System? Definition and Tips. URL: https://in.indeed.com/career-advice/career-development/what-is-an-applicant-tracking-system (дата звернення: 01.02.2023).

6. 11 Best Applicant Tracking Systems Of 2023. URL: https://www.forbes.com/advisor/business/best-applicant-tracking-systems/ (дата звернення: 02.02.2023).

7. L. Ruff et al., "A Unifying Review of Deep and Shallow Anomaly Detection," in Proceedings of the IEEE, vol. 109, no. 5, pp. 756-795, May 2021, doi: 10.1109/JPROC.2021.3052449.

8. M. V. Todescato, J. Hilger and G. Dal Bianco, "A New Strategy to Seed Selection for the High Recall Task," in IEEE Latin America Transactions, vol. 19, no. 12, pp. 2105-2112, Dec. 2021, doi: 10.1109/TLA.2021.9480153.

9. M. M. Eduardo Vasconcellos et al., "Siamese Convolutional Neural Network for Heartbeat Classification Using Limited 12-Lead ECG Datasets," in IEEE Access, vol. 11, pp. 5365-5376, 2023, doi: 10.1109/ACCESS.2023.3236189.

10. G. Krivoulya, I. Ilina, V. Tokariev and V. Shcherbak, "Mathematical Model for Finding Probability of Detecting Victims of Man-Made Disasters Using Distributed Computer System with Reconfigurable Structure and Programmable Logic," 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2020, pp. 573-576, doi: 10.1109/PICST51311.2020.9467976.

11. Sharonova, N., Kyrychenko, I., Gruzdo, I., Tereshchenko, G. "Generalized Semantic Analysis Algorithm of Natural Language Texts for Various Functional Style Types", in CEUR Workshop Proceedings, 2022, 3171, pp. 16-26.

12. Sharonova, N., Kyrychenko, I., Tereshchenko, G., "Application of big data methods in E-learning systems", 2021 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), 2021. – CEUR-WS, 2021, ISSN 16130073. - Volume 2870, PP. 1302-1311.

13. J. I. Sales and F. de Sousa Ramos, "Learning in a Hiring Logic and Optimal Contracts," in IEEE Access, vol. 9, pp. 154540-154552, 2021, doi: 10.1109/ACCESS.2021.3128039.

14. V. S. Pendyala, N. Atrey, T. Aggarwal and S. Goyal, "Enhanced Algorithmic Job Matching based on a Comprehensive Candidate Profile using NLP and Machine Learning," 2022 IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService), Newark, CA, USA, 2022, pp. 183-184, doi: 10.1109/BigDataService55688.2022.00040.

15. V. Rathor, N. Kaur and R. Rautela, "Employee Hiring using Machine Learning," 2022 International Conference on Cyber Resilience (ICCR), Dubai, United Arab Emirates, 2022, pp. 1- 3, doi: 10.1109/ICCR56254.2022.9995882.

16. T. Chakraborti, A. Patra and J. A. Noble, "Contrastive Fairness in Machine Learning," in IEEE Letters of the Computer Society, vol. 3, no. 2, pp. 38-41, 1 July-Dec. 2020, doi: 10.1109/LOCS.2020.3007845.

17. J. Lu, H. Liu, Z. Zhang, J. Wang, S. K. Goudos and S. Wan, "Toward Fairness-Aware Time- Sensitive Asynchronous Federated Learning for Critical Energy Infrastructure," in IEEE Transactions on Industrial Informatics, vol. 18, no. 5, pp. 3462-3472, May 2022, doi: 10.1109/TII.2021.3117861.

18. A. Vos, K. Poels and A. Jacobs, "Exploring the Impact of Algorithmic Decision-Making on Recruitment and Selection Processes: The Perceptions of Job Applicants," in IEEE Transactions on Human-Machine Systems, vol. 51, no. 1, pp. 42-50, Feb. 2021, doi: 10.1109/THMS.2020.3039475.

19. K. Smelyakov, A. Chupryna, D. Sandrkin and M. Kolisnyk, "Search by Image Engine for Big Data Warehouse," 2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 2020, pp. 1-4, doi: 10.1109/eStream50540.2020.9108782.

20. K. Smelyakov, A. Chupryna, O. Bohomolov and I. Ruban, "The Neural Network Technologies Effectiveness for Face Detection," 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP), 2020, pp. 201-205, doi: 10.1109/DSMP47368.2020.9204049.

21. K. Smelyakov, A. Chupryna, O. Bohomolov and N. Hunko, "The Neural Network Models Effectiveness for Face Detection and Face Recognition," 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 2021, pp. 1-7, doi: 10.1109/eStream53087.2021.9431476.

22. M. J. Barber, H. Kim, "Can machine learning improve the quality of hires? Evidence from a randomized controlled trial," National Bureau of Economic Research, Paper 27547, Jul. 2020.

23. G. N. Awaludin et al., "Comparison of Decision Tree C4.5 Algorithm with K-Nearest Neighbor (KNN) Algorithm in Hadith Classification," 2020 6th International Conference on Computing Engineering and Design (ICCED), Sukabumi, Indonesia, 2020, pp. 1-6, doi: 10.1109/ICCED51276.2020.9415796.

24. R. Mukherjee and A. De, "Development of an Ensemble Decision Tree-Based Power System Dynamic Security State Predictor," in IEEE Systems Journal, vol. 14, no. 3, pp. 3836-3843, Sept. 2020, doi: 10.1109/JSYST.2020.2978504.

25. C. -H. Hsu, "Optimal Decision Tree for Cycle Time Prediction and Allowance Determination," in IEEE Access, vol. 9, pp. 41334-41343, 2021, doi: 10.1109/ACCESS.2021.3065391.

26. X. Wang and F. Liu, "Data-Driven Relay Selection for Physical-Layer Security: A Decision Tree Approach," in IEEE Access, vol. 8, pp. 12105-12116, 2020, doi: 10.1109/ACCESS.2020.2965963.

27. I. Yildirim and M. Celik, "An Efficient Tree-Based Algorithm for Mining High Average-Utility Itemset," in IEEE Access, vol. 7, pp. 144245-144263, 2019, doi: 10.1109/ACCESS.2019.2945840.

28. The Guide to Decision Tree-based Algorithms in Machine Learning (Including Real Examples). URL: https://omdena.com/blog/decision-tree-based-algorithms (дата звернення: 26.02.2023).

29. Decision Tree Classification Algorithm. URL: https://www.javatpoint.com/machine-learning- decision-tree-classification-algorithm (дата звернення: 26.02.2023).

30. Random Forest Algorithm. URL: https://www.javatpoint.com/machine-learning-random-forest- algorithm (дата звернення: 26.02.2023).

31. Gradient Boosting Algorithm: A Complete Guide for Beginners. URL: https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/ (дата звернення: 27.02.2023).

32. How to Evaluate Classification Models. URL: https://www.edlitera.com/blog/posts/evaluating-classification-models (дата звернення: 28.02.2023).

33. Draw.io. [Online]. Available. URL: https://app.diagrams.net (дата звернення: 01.03.2023).

34. The confusion matrix diagram of the binary classification model. URL: https://drive.google.com/file/d/1IR81h_KrxTtRZkwX-zIl4dDABsaJp0IP/view?usp=sharing.

35. What Is Resume Parsing? (With Benefits and Tips). URL: https://www.indeed.com/career-advice/resumes-cover-letters/resume-parsing (дата звернення: 02.03.2023).

36. Dataset: Recruitment data. URL: https://www.kaggle.com/datasets/rafunlearnhub/recruitment-data.

37. Apache Spark Benefits: Reasons Why Enterprises are Moving To this Data Engineering Tool. URL: https://www.ksolves.com/blog/big-data/spark/apache-spark-benefits-reasons-why-enterprises-are-moving-to-this-data-engineering-tool (дата звернення: 08.03.2023).

38. Spark documentation. URL: https://spark.apache.org/docs/latest/api/java (дата звернення: 10.03.2023).

39. The experimental workflow. URL: https://drive.google.com/file/d/1PCEkscE0ZaMJXwyYZh_adWr3FrciEtaw/view?usp=sharing.