

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

*На правах
рукопису*

УДК _____

ВИПУСКНА КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

**“ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ НЕЙРОМЕРЕЖНОГО АНАЛІЗУ
РЕЗЮМЕ ДЛЯ ОЦІНКИ ПРОФЕСІЙНОЇ ПРИДАТНОСТІ ЛЮДИНИ”**

Спеціальність 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

МР.ПЗ – 30.00.00.000

Студент

ПЗМ-21 _____ /Альона КОВРІНА/

Науковий керівник

к.т.н., доц. _____ /Тетяна КОВАЛЮК/

Консультант

з питань нормоконтролю

к.т.н., асистент _____ / Анастасія ВЕЧЕРКОВСЬКА /

Допускається до захисту

Завідувач кафедри

д.т.н., проф. _____ /Олексій БИЧКОВ/

Київ – 2022

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра програмних систем і технологій
Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри програмних систем і технологій

_____ (О.С.Бичков)

“ ___ ” _____ 20__ р.

**ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ
СТУДЕНТУ**

Ковріної Альони Євгеніївни

1. Тема випускної кваліфікаційної магістерської роботи «Програмне забезпечення нейромережного аналізу резюме для оцінки професійної придатності людини» та керівник проекту (роботи) Тетяна Володимирівна Ковалюк, к.т.н., доцент затверджені наказом вищого навчального закладу від «__»__ 20__ р. №_____

2. Строк здачі студентом закінченої роботи «__»__ 2022__ р.

3. Вихідні дані до роботи: підручники, навчальні посібники, статті, Інтернет-ресурси.

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)

Аналітична частина:

- обґрунтувати актуальність розробки програмного забезпечення нейромережного аналізу резюме для оцінки професійної придатності людини;
- дослідити та аналізувати існуючі рішення відбору персоналу;

- дослідити підбір персоналу з точки зору математичного моделювання;
- сформувані функціональні вимоги та обрати відповідні технології розробки програмного забезпечення.

Практична частина:

- спроектувати архітектуру систему відповідно до визначених цілей;
- створити функцію парсінгу резюме;
- розробити програмну систему та проаналізувати результати.

5. Консультанти з роботи із зазначенням розділів роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Аналіз предметної області та теоретичних аспектів	Ковалюк Т.В.	28.09.21	15.12.21
Проектування та вибір технологій програмування	Ковалюк Т.В.	15.12.21	18.01.22
Програмна реалізація	Ковалюк Т.В.	18.01.22	06.05.22
Огляд результату виконання поставленої задачі	Ковалюк Т.В.	06.05.22	12.05.22

7. Дата видачі завдання 10 жовтня 2021 р.

Керівник _____ (Ковалюк Т.В.)

Завдання прийняв до виконання _____ (Ковріна А.Є.)

КАЛЕНДАРНИЙ ПЛАН

Номер і назва етапів роботи	Термін виконання етапів роботи	Примітка
1: Дослідження предметної області	28.09.21 – 22.10.21	
2: Аналіз існуючих рішень відбору персоналу	25.10.21 – 30.11.21	
3: Опис функціональних та нефункціональних вимог до системи	30.11.21 – 14.12.21	
4: Модель та підхід до програмування	15.12.21 – 24.12.21	
5: Аналіз та вибір технологій програмування	24.12.21 – 18.01.22	
6: Обробка великих даних	18.01.22 – 06.02.22	
7: Створення нейромережової моделі	06.02.22 – 12.04.22	
9: Тестування системи	25.04.22 – 12.05.22	

Студент – магістр _____ (Ковріна А.Є.)

Керівник роботи _____ (Ковалюк Т.В.)

АНОТАЦІЯ

Випускна кваліфікаційна магістерська робота складається зі вступу, 3 розділів, висновків та списку використаних джерел (20 найменувань). Містить у собі 17 рисунків та 2 додатків. Загальний обсяг роботи становить 80 сторінки.

Темою роботи є “Програмне забезпечення нейромережного аналізу резюме для оцінки професійної придатності людини”.

Об’єктом дослідження є процес підбору резюме за вимогами вакансій.

Предметом дослідження – є методи та засоби застосування нейронних мереж у процесі підбору резюме за вимогами вакансій.

Метою роботи є дослідження можливості застосування нейронних мереж для параметричного підбору резюме, що настроюється, за вимогами вакансій.

Результати роботи: Дослідження можливості застосування штучних нейронних мереж для вирішення завдання підбору резюме за вимогами вакансій. Спроектоване та розроблене програмне забезпечення нейромережного аналізу резюме для оцінки професійної придатності людини.

Ключові слова: нейронні мережі, машинне навчання, резюме, python, аналіз резюме, оцінка професійної придатності людини

ABSTRACT

The final qualifying master's thesis consists of an introduction, 3 chapters, conclusions and a list of sources used (20 items). Contains 17 figures and 2 appendices. The total volume of the work is 80 pages.

The **theme** of the work is " Software for neural network analysis of resumes for candidate professional suitability assessment".

The **object** is the process of selecting resumes for vacancies.

The **subject** is the use of neural networks in the process of selecting resumes for vacancies.

The **goal** of the work is to study the possibility of using neural networks for a customizable parameter-based resume selection for vacancy requirements.

Results of work: Research of possibilities of application of artificial neural networks for the decision of selection of the summary according to requirements of vacancies. Designed and developed neural network resume analysis software for human professional suitability.

Keywords: neural networks, machine learning, resume, python, resume analysis, assessment of human professional suitability.

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1. Порівняльний аналіз процедур відбору та підбору персоналу	12
1.2 Призначення системи	13
1.3 Аналіз існуючих рішень відбору персоналу	14
1.3.1 Аналіз існуючих інтернет-сервісів	17
1.3.1.1. HURMA	18
1.3.1.2 SKIPP	20
1.3.1.3 CleverStaff.....	21
1.3.1.4. Breezy HR	22
Висновок по розділу	24
РОЗДІЛ 2. МОДЕЛІ ТА ПІДХОДИ, ЗАСНОВАНІ НА МАТЕМАТИЧНИХ МЕТОДАХ.....	25
2.1. Математичного моделювання процесу підбору персоналу ...	25
2.2 Модель та підхід до програмування	30
Висновок по розділу	33
РОЗДІЛ 3. ПРОЦЕС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	34
3.1 Постановка завдань проектування	34
3.2 Вибір технологій програмування	36
3.3. Створення нейромережової моделі.....	36
3.3.1 Вимоги до моделі.....	40
3.3.2 Структура моделі.....	40

3.4. Вхідні дані	44
3.4.1. Виявлення та уявлення параметрів	45
3.4.2. Попередня обробка даних.....	46
3.5. Вихідні дані	48
3.6 Наукова новизна проекту.....	48
Висновок по розділу	49
ВИСНОВОК.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТОК А.....	53
ДОДАТОК Б – SOFTWARE ARCHITECTURE DOCUMENT.....	66
Master’s course work Software Architecture Document (SAD)	66
1. Documentation Roadmap	70
1.1. Document Management and Configuration Control Information.....	70
1.2. Purpose and Scope of the SAD	70
1.3. Viewpoint Definitions	74
1.3.1. Masters program student Viewpoint Definition.....	75
1.3.1.1. Abstract	75
1.3.1.2. Stakeholders and Their Concerns Addressed	75
1.3.1.3. Elements, Relations, Properties, and Constraints	75
1.3.1.4. Language(s) to Model/Represent Conforming Views.....	75
2. Architecture Background.....	76
2.1. Problem Background	76
2.1.1. System Overview.....	76

2.1.2. Goals and Context.....	76
2.1.3. Significant Driving Requirements.....	76
2.2. Solution Background.....	76
2.2.1. Architectural Approaches.....	76
2.2.2. Analysis Results.....	76
2.2.3. Requirements Coverage.....	76
3. Referenced Materials.....	77
4. Directory.....	78
4.1. Index.....	78
5. Sample Figures & Tables.....	80

ВСТУП

Застосування штучного інтелекту у багатьох сферах діяльності збільшилося останніми роками. Однак для систематичного впровадження цих підходів необхідне подальше концептуальне та методологічне розуміння. Керівники великих компаній все частіше стикаються з ситуацією, коли попит на висококваліфікованих працівників перевищує пропозицію. Цей факт привів до створення такого напрямку діяльності як «Управління персоналом». У широкому сенсі управління персоналом – це сукупність бізнес-процесів компанії, спрямованих на пошук, залучення, мотивацію і подальший розвиток людського потенціалу.

Найбільш трудомістким та відповідальним процесом у справі управління персоналом, звичайно ж, є пошук та оцінка професіоналізму кандидатів на посади. Це вимагає від менеджерів великого досвіду, а якщо необхідно зібрати великий штат за короткий час, то знадобиться ще й гарна база претендентів. З цієї причини багато компаній віддають перевагу співробітництву з рекрутинговими агентствами, основним видом діяльності яких є підбір персоналу, що дозволяє здійснювати його на високому професійному рівні.

Пошуком кандидатур і проведенням співбесід займаються HR-менеджери, які повинні чудово орієнтуватися в потребах клієнта, а також володіти достатнім професіоналізмом, щоб оцінити потенційного працівника. Вирішення такої задачі вимагає всебічних знань, але незважаючи на професіоналізм рекрутерів, фактор суб'єктивності повністю виключити не вдасться. Необхідно враховувати, що запропонований кадровим агентством працівник, що не пройшов випробувальний термін у компанії-клієнта, буде коштувати агентству не тільки репутації, але і прибутку, оскільки агентство, залежно від конкретних умов договору, що може істотно зменшити суму рахунку, виставленого за послуги.

Отже, пріоритетним завданням рекрутингової компанії є підвищення ефективності підбору резюме і подальшої оцінки кандидатів. Агентства покладаються лише на досвід і професіоналізм працівників HR-відділу але для більш ефективної оцінки кандидатів доцільним є застосування інтелектуальних технологій. Дослідження можливості застосування штучних нейронних мереж для вирішення завдання підбору резюме за вимогами вакансій визначило мету і завдання.

Мета роботи: дослідження можливості застосування нейронних мереж для параметричного підбору резюме, що настроюється, за вимогами вакансій.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- Виконати аналіз предметної області;
- Виконати огляд існуючих рішень;
- Побудувати модель нейронної мережі для підбору резюме згідно до вимог вакансій;
- Провести моделювання, що дозволяє оцінити ефективність підбору резюме з використанням побудованої моделі.

Об'єктом дослідження є процес підбору резюме на вимоги вакансій.

Предметом дослідження є методи та засоби застосування нейронних мереж у процесі підбору резюме за вимогами вакансій.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Порівняльний аналіз процедур відбору та підбору персоналу

Під формулюванням теми цієї кваліфікаційної роботи розуміється аналіз відповідності резюме вимогам вакансії (по суті йдеться про відповідність кандидата вимогам роботодавця). Для вирішення цього завдання важливою стає точність відповідності портрета претендента (резюме) вимогам роботодавця (вакансії). Йдеться вимоги, які пред'являються до виконавця роботи. В одних випадках кандидату для відповідності вимогам роботодавця достатньо кваліфікації, отриманої в курсі підготовки, який можна охарактеризувати певним змістом та тривалістю навчання. В інших випадках потрібна серйозна професійна підготовка зі стандартними вимогами до обсягу спеціальних знань, умінь та навичок.

В умовах ведення бізнесу, що ускладнюються, специфічні вимоги роботодавця до виконавця можуть змінюватися ще до того, як новий кандидат приступить до роботи, і формулюються тільки в загальних рисах. У довгостроковій перспективі важливо не те, якими знаннями чи спеціальними навичками має потенційний працівник, а те, як він проявлятиме себе на роботі.

Розглянемо компанію, в якій можна автоматизувати процес найму, і ця компанія зараз використовує ручну систему для найму співробітників. Оскільки набір співробітників вручну вимагає багато часу, також не варто виключати помилковий найм некомпетентних людей, що призводить до марнування грошей компанії або нерозумної стратегії з погляду економії. Система найму – це процес відбору потенційних кандидатів на вакантну посаду та найму кандидатів, які відповідають вимогам організації; отже, найм є стратегічною функцією відділу кадрів.

Управління людськими ресурсами включає кілька обов'язків, в основному вибір потенційних співробітників, які потрібні компанії, щоб ця

система допомогла компанії вибрати кращого потенційного кандидата, який міг би допомогти в досягненні її цілей. Підбір персоналу – це систематичний процес формування пулу кваліфікованих кандидатів на роботу в організації. Відбір та найм персоналу – життєво важливий процес для компанії. Використання компетентної робочої сили може забезпечити зростання та стабільність. [1]

1.2 Призначення системи

Система управління підбором персоналу — це комплексна система управління всіма процесами підбору персоналу організації. Це полегшує потік інформації компанії у відділах кадрів. Так само, як управління продуктивністю, розрахунок заробітної плати та інші системи, система управління набором персоналу допомагає окреслити процеси найму та ефективно управляти набором персоналу. Особливості, функції та основні переваги системи управління набором персоналу пояснюються нижче:

1. Структурування процесу найму.
2. Більш швидке, неупереджене, точне та надійне оброблення заявок з різних додатків.
3. Скорочення часу на найм та вартості найму.
4. Система допомагає включати та інтегрувати різні посилання, такі як система подання заявок на офіційний сайт компанії, незапитані заявки, аутсорсинг найму, прийняття остаточного рішення в основний процес найму.
5. Підтримування автоматизованої активної бази даних претендентів для більш легкого управління талантами та підвищення ефективності процесів найму.
6. Забезпечення гнучкого, автоматизованого та інтерактивного

інтерфейсу між системою онлайн-заявок, відділом підбору персоналу компанії та здобувачем.

Система управління підбором персоналу – це інноваційний інструмент інформаційної системи, що допомагає економити час та витрати рекрутерів та покращувати процеси найму.

1.3 Аналіз існуючих рішень відбору персоналу

В умовах сучасної ринкової економіки ключем до успіху чи провалу організації є набраний персонал. Кваліфікований персонал здатний покращити ефективність функціонування підприємства, тоді як неправильно підібрана робоча сила здатна звести ефективність діяльності організації до нуля. З цих міркувань стає очевидною ключова позиція етапів підбору та відбору персоналу у загальній системі управління, оскільки підбір персоналу передбачає проведення заходів, які найбільшою мірою впливають на діяльність організації.

В цілому, процес підбору та відбору персоналу можна подати у вигляді схеми, наведеної на рисунку 1.1.

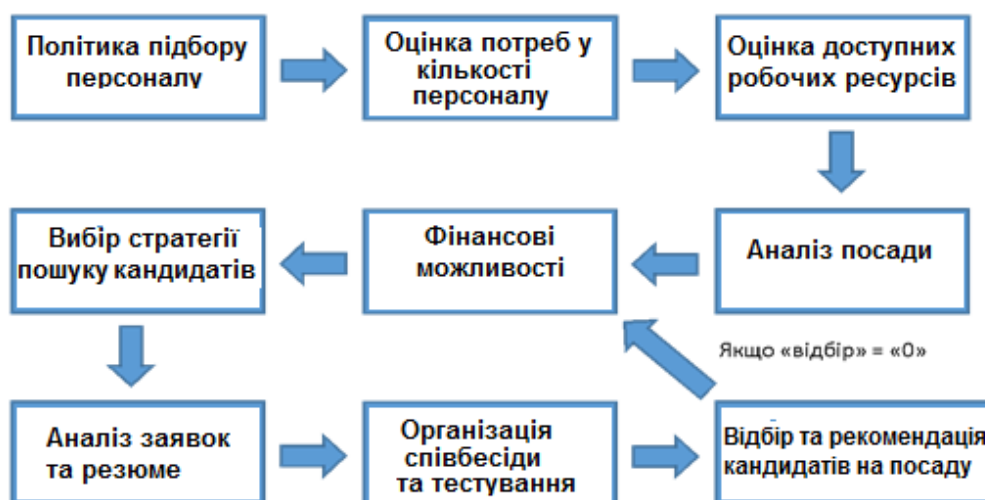


Рисунок 1.1. Процес підбору та відбору персоналу

Процес на рис. 1.1 є циклічним: у разі, якщо не вдалося відібрати жодного кандидата на останньому етапі, слід або збільшити фінансове забезпечення, або знизити вимоги до кандидатів.[1]

Підбір та відбір персоналу є пріоритетним питанням у рамках здійснення процесу з управління персоналом не тільки через вплив на ефективність діяльності організації, але й з точки зору конкурентної переваги організації. Пошук та відбір персоналу завжди грали ключову роль у процесі управління персоналом як провідну частину організаційного функціонування підприємства. У сучасному світі ці процеси стають більш значущими, оскільки в умовах дедалі більшої конкуренції організації все частіше розглядають своїх працівників як джерело конкурентної переваги.

Методи підбору персоналу можна класифікувати за кількома критеріями. Якщо класифікувати методи підбору персоналу за цільовою аудиторією, їх склад буде виглядати так:

- рекрутинг;
- хедхантінг;
- прелімінаринг;
- ексклюзивний пошук.

Рекрутинг – процес пошуку персоналу, спрямований на підбір та найм працівників нижчої та середньої ланки, так званий масовий або лінійний пошук. Цей процес включає відбір, тобто оцінку кандидатів. Рекрутинг полягає у складанні грамотного опису вакансії та його розміщенні на сайті компанії або на спеціалізованих порталах з працевлаштування, а також надсилання до кадрових агенцій. Цей метод орієнтований на кандидатів, які самостійно займаються пошуком роботи.

Даний процес, що проводиться кадровими агентствами, які мають велику базу резюме, а також статистику працевлаштування рекомендованих

працівників, дозволяє моделювати підбір резюме за вимогами вакансій, спираючись на інформацію, зазначену в резюме і вакансіях, відповідно.

Ексклюзивний пошук – процес пошуку персоналу, спрямований на підбір працівників вищої управлінської ланки, а також рідкісних фахівців. Найчастіше цей метод використовують, якщо в організації стоїть мета знайти людей, здатних надати ключове вплив на ефективність функціонування підприємства.

Хедхантінг (з англ. «Полювання за головами») – підбір керівників шляхом їх переманювання з існуючого місця роботи. Спосіб підбору персоналу, що ґрунтується на "полюванні" за співробітником, який володіє бажаним набором якостей і навичок, зайнятим в іншій організації і не ставить за мету зміну робочого місця. Головною відмінністю від попередніх методів є те, що залучити такого співробітника на посаду є складним завданням, оскільки для успішного переманювання його необхідно «мотивувати» та запропонувати вигідніші умови, ніж на місці існуючої роботи. Практично повна відсутність необхідної статистичної інформації ускладнює процес моделювання, спрямованого на вирішення задачі відповідності резюме та вакансії.

Прелімінаринг – метод підбору, що ґрунтується на залученні молодих спеціалістів (студентів та випускників вузів) за допомогою організації місць для проходження виробничої практики, стажувань. Зазначений метод, у разі успішного застосування, має на увазі проходження спеціалістом усіх наступних етапів просування, починаючи з профорієнтації та адаптації молодого спеціаліста та закінчуючи роботою на посаді керівника вищої ланки управління.

Підсумовуючи розгляд методів підбору персоналу слід підкреслити, що ефективність використання того чи іншого методу пов'язана з можливістю формалізації дій та процедур рекрутингового процесу за допомогою

математичного моделювання. До методу, найбільш вдало сумісного з математичним моделюванням, можна віднести рекрутинг, оскільки кадрові агенції мають можливість ретельно досліджувати кандидатів, а також мають широку базу даних про незайняту робочу силу і характеризують свою діяльність відкритістю і прозорістю. [1]

1.3.1 Аналіз існуючих інтернет-сервісів

На даний момент активно здійснюються спроби автоматизувати найбільш рутинні та затратні за часом трудові функції HR-менеджерів. Насамперед до них відносяться документаційне забезпечення процесу управління персоналом та рекрутинг – діяльність з підбору та відбору персоналу. У практиці HR-менеджменті давно використовуються різні інформаційні продукти по типу «ІС», що дозволяють автоматизувати виконання функцій кадрового обліку, нарахування та виплату заробітної плати, управління трудовими ресурсами.

Досягнення сучасних технологій штучного інтелекту дозволяють розширити сферу застосування ІТ-технологій у практиці управління персоналом. Велика робота проводиться над їх застосуванням в рекрутингу. Зацікавленість бізнесу у розвитку цього напрямку підтверджує зростаючий у світі обсяг інвестицій у нові HR-технології, який у 2015 році, за даними Delite, налічував \$2,4 млрд., що на 60% більше, ніж роком раніше. Найбільші вкладення йдуть на автоматизовані системи управління персоналом, системи управління талантами, послуги для корпоративного навчання.

Аналіз літератури та професійних сервісів показує, що на сьогоднішній день у рекрутингу використовуються такі технології, як Big Data, алгоритм Sabber, HR-боти та інше. [2]

1.3.1.1. HURMA

Система автоматизації HR та рекрутингу – це комплексне рішення для автоматизації щоденних завдань HR-менеджера, рекрутера і кожного співробітника в компанії.

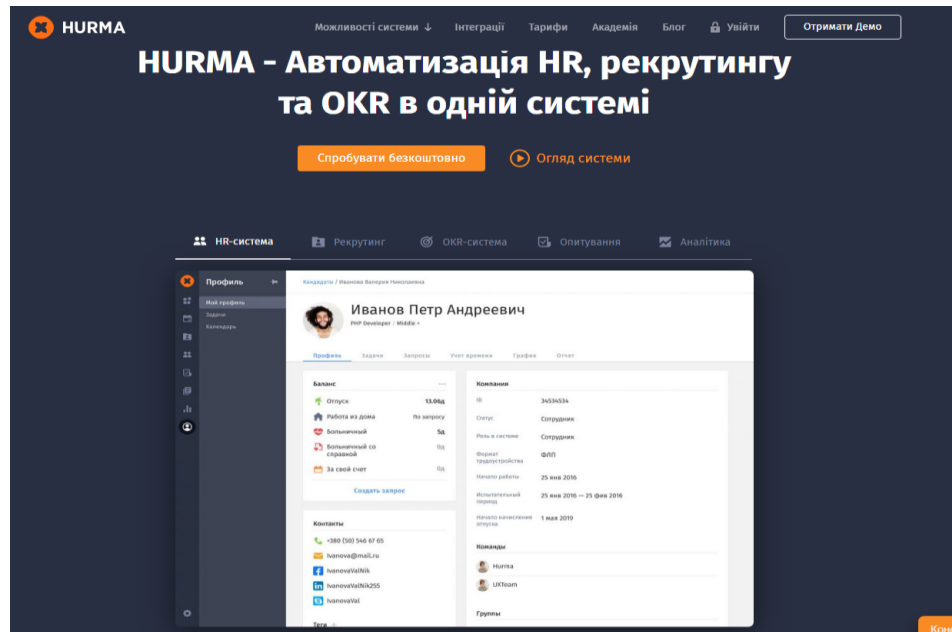


Рисунок 1.2. Система автоматизації HR та рекрутингу HURMA

Автоматизуючи свої процеси, користувачі отримують масу звільненого часу, який можна направити на роботу з лояльністю співробітників, їх мотивацію, формування внутрішнього і зовнішнього HR-бренду компанії та інші завдання, що вимагають безпосередньої уваги. HRM-система – це не заміна HR-менеджера, вона істотно полегшує його життя.

Автоматизуючи ведення бази кандидатів, всі події, які призначаються для кандидатів, відгуки та розміщення вакансій, рекрутер може приділити більше часу особистому спілкуванню з кандидатом, спілкуванню з командою для більш точного формування вимог і аналізу рекрутингових каналів.

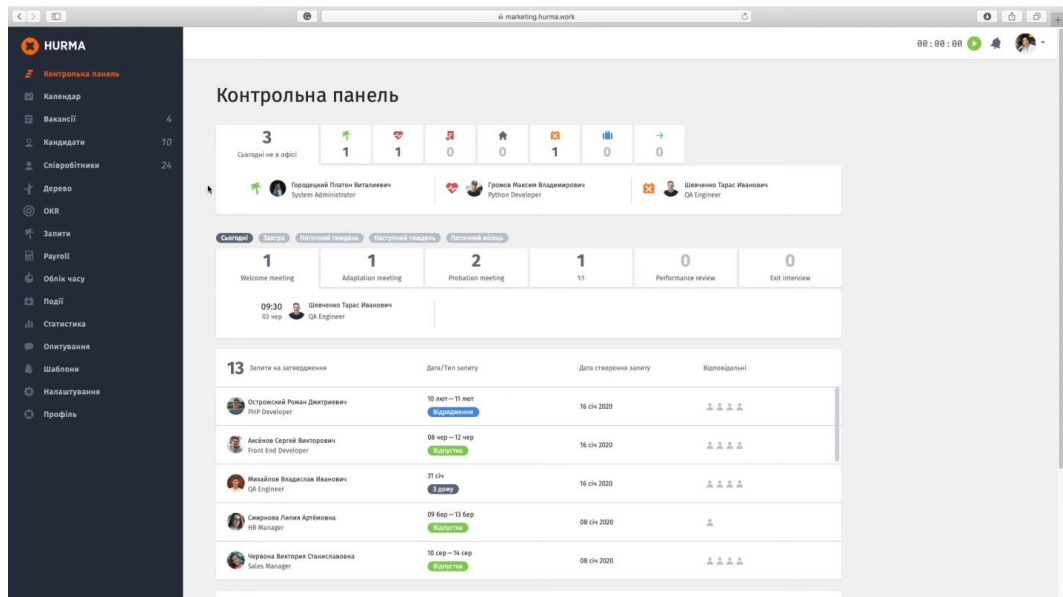


Рисунок 1.3. Контрольная панель системы автоматизации HR та рекрутингу HURMA

Співробітник, в свою чергу, отримує просту та зрозумілу схему нарахування відпусток і лікарняних, можливість прямо з месенджера зробити запит на відпустку або роботу з дому і відсутність потреби смикати HR-а з будь-якого питання.

Основні можливості Hurma System

- HR-дашборд;
- Онбординг та адаптація;
- База співробітників;
- Події та нагадування;
- Опитування та Performance review;
- Календар компанії;
- Відстеження настрою співробітників;

- Чат-бот та багато іншого;
- HR-аналітика;
- Сценарії: автоматизація онбордингу та офбордингу;
- Автоматизація запитів працівників: відсутності, переробки, відрядження та інші події;
- Облік переробок, робочого часу, розрахунки зарплат з допомогою Payroll.

1.3.1.2 SKIPP

Skipp — human cloud платформа для автоматизованого найму ІТ-фахівців високого рівня. Сервіс вирішує одну з проблем будь-якої компанії, якій потрібно зробити той чи інший ІТ-продукт — пошук та наймання команди розробників. Це рішення допомагає клієнтам миттєво нарощувати ресурси розробки для трансформації та розвитку бізнесу. Skipp також допомагає і самим ІТ-фахівцям – їм не доводиться самостійно шукати замовників, на платформі підбираються проекти, які не лише приносять дохід, а й стають гарним доповненням до портфолію.

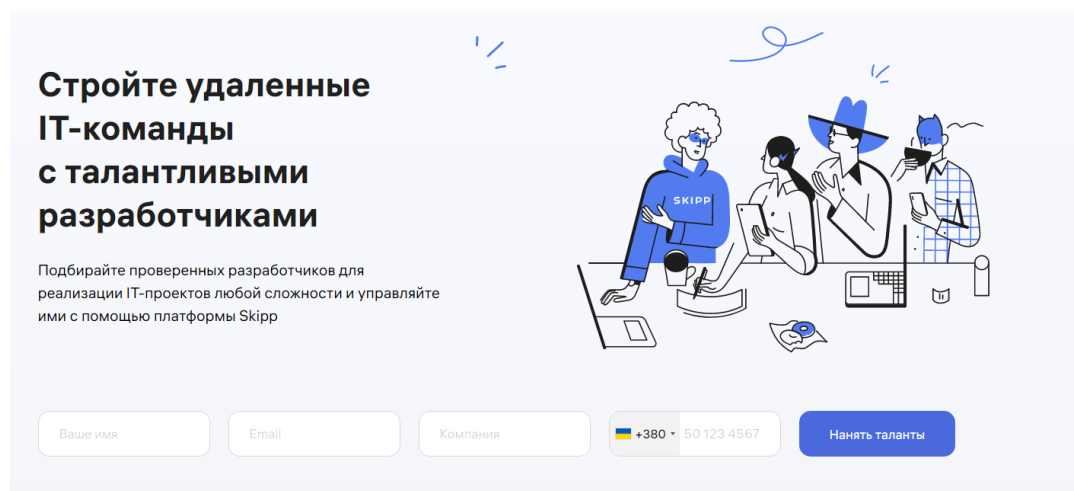


Рисунок 1.4. Хмарна платформа Skipp

Кому підійде:

- розробникам, котрі хочуть працювати на фрілансі, але бути захищеними від ризиків;
- тим, хто планує напрацьовувати портфоліо та розвивати навички;
- компаніям, які чітко бачать результат у вигляді продукту, але не знають, як до нього дійти. [20]

1.3.1.3 CleverStaff

CleverStaff — це єдина система професійного підбору персоналу. Що включає система?

Управління базою даних: Технологія парсингу резюме, Інтеграція з поштою та завантаження резюме з неї, Масове завантаження даних. Дані сервісу start pack.ru. Інтеграція з LinkedIn та сайтами пошуку роботи.

ATS: Етапи роботи, що настроюються. Шаблони листів, що настроюються, Інтеграція з календарями Google і Outlook, Керування ролями і доступом.

Автоматизація рекрутингу: Пропозиції та рейтинги кандидатів, Швидкі звіти з вакансій, Вирва рекрутингу.

Розширені функції: Інтеграції, налаштування та API, HR-модуль, Збереження резервних копій бази, Моментальні звіти та доступ для клієнтів.

Турбота про користувачів: Зручний інтерфейс (UX), Моментальні звіти та доступ для клієнтів, Оновлення раз на 1-2 тижні, Імпорт даних до 3 днів.

Основні можливості CleverStaff:

- розміщення вакансій;
- оцінки;
- управління кандидатами;
- інструменти спільної роботи;
- управління контактами / CRM;
- планування співбесід;
- пошук резюме;
- інтеграція соціальних мереж;
- інтеграція пошти;
- налаштування брендингу.

1.3.1.4. Breezy HR

Breezy HR — це онлайн-сервіс для пошуку співробітників, що дозволяє рекрутерам швидко знаходити кандидатів та працювати з ними, працюючи з LinkedIn та AngelList, спільно працюючи з командою та майбутніми працівниками в режимі реального часу. Плюс є інтеграція з дошками оголошень типу Indeed, Dribbble, Monster, StackOverflow, Glassdoor та ін.

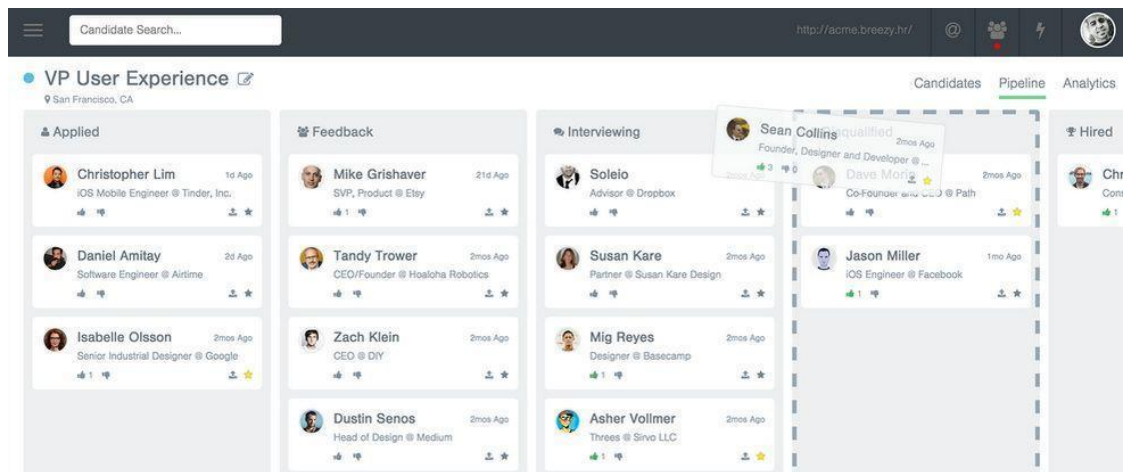


Рисунок 1.5. Онлайн-сервіс пошуку співробітників Breezy HR

Можливості Breezy HR:

- простий набір та відстеження кандидатів;
- оптимізований механізм під мобільні пристрої;
- безкоштовна та преміум-інтеграція з робітними сайтами;
- фірмові сторінки;
- настроювані питання та вимоги до кандидатів;
- код для вставки на сайт;
- розбір резюме;
- відео-співбесіда;
- необмежену кількість користувачів;
- співробітництва у реальному часі з підтримкою згадок;
- інтеграція календаря Google;
- планування інтерв'ю;

- нагадування кандидатам;
- підтримка зовнішніх рекрутерів;
- інструмент із офіційною інтеграцією slack;
- рекрутингова аналітика. [20].

Висновок по розділу

У цьому розділі проведено порівняльний аналіз процедур набору, відбору та підбору персоналу, представлено обґрунтування використання терміну «підбір» стосовно формулювання теми роботи. Надалі вважається, що аналіз відповідності резюме вимогам вакансії та аналіз відповідності кандидата вимогам роботодавця тотожно рівні.

Проведено аналіз ефективності сучасних методів підбору персоналу, за результатом якого зроблено висновок, що методом, що найбільш підходить для моделювання, є рекрутинг. Даний метод надає можливість моделювати процес підбору резюме за вимогами вакансій, спираючись на інформацію, вказану безпосередньо у резюме та вакансії. При ексклюзивному пошуку або хедхантинг практично повністю відсутня необхідна статистична інформація, що значно ускладнює процес моделювання.

РОЗДІЛ 2. МОДЕЛІ ТА ПІДХОДИ, ЗАСНОВАНІ НА МАТЕМАТИЧНИХ МЕТОДАХ

2.1. Математичного моделювання процесу підбору персоналу

Методи пошуку кандидатів досить добре розроблені, створено велику кількість теорій, що описують процес залучення працівників. Однак найскладнішим, витратним за часом і неоднозначним за результатом залишається етап, на якому приймається рішення про найм співробітника. Проведення співбесід та перегляд резюме HR-менеджерами завжди пов'язані з людською суб'єктивністю. Крім того, не так просто знайти менеджера з персоналу, який володіє достатнім професійним досвідом, утримати його, мотивувати тощо.[5]

Як уже було сказано, найпридатнішим для моделювання є процес прийняття рішення про найм кандидата на посаду, або, якщо йдеться про рекрутингове агентство, рекомендації його клієнту. Математичне моделювання прийняття рішень є розділом дисципліни під назвою «Дослідження операцій», у якій для обґрунтування рішень у різних галузях людської діяльності застосовуються кількісні методи. Під операцією розуміється будь-яка система дій, об'єднаних єдиним задумом і вкладених у досягнення тієї чи іншої мети. Визначення конкретних параметрів операції, у цьому контексті, буде прийняттям рішення.

Метою дослідження операцій є кількісне обґрунтування рішень відповідно до деякого показника ефективності. Важливо підкреслити, що прийняття рішень є прерогативою управлінця, відповідальної особи, тому, як і було зазначено в попередньому розділі використання кількісних методів у процесі добору персоналу може скласти систему підтримки прийняття рішення.

Далі розглядатимемо процес підбору персоналу з позиції HR менеджера кадрового агентства, перед яким стоїть завдання відбору із загальної маси кандидатів тих, кого агентство згодом рекомендує фірмі-клієнту.

Варто зазначити, що на послуги підбору персоналу надається своєрідна гарантія: у випадку, якщо запропонований працівник не проходить випробувального терміну, сума рахунка, що виставляється, може бути істотно зменшена. У деяких випадках ця сума може становити до 50% гонорару рекрутингової агенції, або якщо протягом випробувального терміну претендент і роботодавець захочуть розлучитися, агентство робить рівноцінну заміну. У будь-якому разі незадоволеність клієнта підбором буде коштувати агентству не тільки репутації, тому внутрішня звітність кадрової компанії завжди містить інформацію про тих кандидатів, які не пройшли випробувальний термін, а отже, їхнє резюме можна легко витягти з бази.

Перший підхід до застосування кількісних методів у справі підбору персоналу, запропонований у дослідженні, ґрунтується на оцінці ймовірності проходження випробувального терміну працівником. Далі, знаючи можливість проходження і встановивши якийсь поріг, менеджер може приймати рішення про рекомендації працівника клієнту, ґрунтуючись на результатах, передбачених моделлю.

Подібний підхід практикується у кредитному скорингу, де передбачається наявність статистичної залежності між соціальними даними позичальника та його сумлінністю. Аналогічно цьому у дослідженні робиться припущення про наявність статистичної залежності між інформацією, що вказується працівником у резюме, та фактом проходження випробувального терміну. Як фактори у скорингових моделях можуть виступати такі дані, як вік, стать, сімейний стан, стаж роботи (загальний та стаж роботи на останньому робочому місці), наявність дітей, освіта, дохід, термін проживання за поточним місцем проживання та інші соціально-демографічні дані. У справі

кредитного скорингу нині широко використовують такі методи статистики, як регресія, дискримінантний аналіз, дерева класифікації, методи дослідження операцій, штучні нейронні мережі.

Прагматичний підхід банківської системи до вирішення завдання пояснює розмаїття застосовуваних методів: практично керівники банків намагаються використовувати те, що приносить швидкий результат, часто нехтуючи поясненням причин тих чи інших статистичних закономірностей. Статистику, за якою оцінюються моделі, становлять анкетні дані позичальників та їх кредитна історія за певний період. Кожній кредитній історії надається відповідна мітка залежно від того, чи є історія «поганою» чи «хорошою».

До категорії «поганих» потрапляють кредитні історії з фактом затримки платежів на тривалий термін. У деяких випадках вказати точно приналежність кредитної історії до однієї з груп важко, тоді їх поміщають у третю групу – середні. Моделі, які можуть передбачати приналежність клієнта до однієї з кількох (більше двох) груп, називаються моделями впорядкованого вибору. Якщо використовуються моделі бінарного вибору, то групу «середні» просто виключають із вибірки. Також важливим показником навчальної вибірки є співвідношення у ній числа «хороших» та «поганих» клієнтів. Для деяких методів, таких як регресія, можливе використання рівної кількості «поганих» та «хороших» клієнтів у навчальній вибірці, у той час як використання байєсівських мереж або дерев класифікації передбачає реальне співвідношення «поганих» та «хороших» клієнтів.[5-6]

Тепер можна спробувати перекласти поняття зі скорингу на проблему відбору персоналу. Як анкет виступатимуть резюме працівників, а розподіл у відповідну групу «поганих» або «хороших» здійснюватиметься за фактом проходження випробувального терміну, що навіть спрощує завдання, оскільки не доведеться виносити рішення про віднесення працівника до тієї чи іншої

категорії самостійно щодо кожного працівника. як у випадку з оцінкою кредитної історії.

Як було зазначено вище, методи регресійного аналізу дозволяють використовувати навчальну вибірку, що містить рівну кількість спостережень з обох груп, тому практична реалізація першого підходу до моделювання відбору персоналу буде заснована на використанні бінарної регресії. В економетриці моделі бінарного вибору є різновидом про моделі дискретного вибору, або, як їх ще називають, моделей якісного вибору. Такі моделі передбачають вибір між двома або більше дискретними альтернативами, на відміну від стандартних економетричних моделей, що оперують переважно безперервними змінними.

Математична постановка задачі підбору персоналу в цьому випадку виглядає наступним чином: нехай X є вектор факторів, що відображає інформацію, представлену в резюме працівника, який впливає на залежну змінну Y , яка, у свою чергу, приймає лише два значення: 1 – якщо працівник пройшов випробувальний термін і 0 – інакше.

Тоді ймовірність того, що $Y = 1$, необхідно висловити як функцію від факторів за формулою (2.1):

$$P(Y = 1|\bar{X}) = F(X) \quad (2.1)$$

Як правило, як функція F використовуються деякі інтегральні функції розподілів, а як аргумент береться виважена сума значень факторів, де вагові коефіцієнти є параметрами моделі

Другий підхід до моделювання рекрутингу передбачає розгляд дій HR менеджера як завдання класифікації. Завдання класифікації є формалізовані завдання, у яких безліч об'єктів розділено певним чином класи. З одного боку, є безліч об'єктів, котрим відомі їх класи. З іншого боку, є безліч об'єктів, належність до класів яких невідома. Завдання полягає в тому, щоб побудувати

алгоритм, який здійснюватиме класифікацію довільного об'єкта з другої множини. Такі завдання, як правило, вирішуються за допомогою штучних нейронних мереж, про що йтиметься далі.

Якщо розглядати завдання рекрутера як задачу класифікації, то цілком очевидно, що класів буде всього два – «хороший» кандидат, у якому можна бути впевненим, що він пройде випробувальний термін та принесе користь організації, та «поганий» кандидат, який відповідно випробувальний термін не мине.

Математична постановка завдання у такому вигляді виглядатиме так: кожному кандидату, з яким працює рекрутингове агентство, поставимо у відповідність ознаковий опис, побудований на інформації, витягнутій з його резюме. Багато кандидатів поділено на два класи: ті працівники, що пройшли випробувальний термін у компанії клієнта, і ті, хто його не пройшов. Отже, є безліч описів кандидатів X і безліч найменувань класів, що складається, відповідно, з двох елементів – 0, якщо працівник не пройшов Y випробувальний термін у компанії-клієнта, і 1, якщо працівник пройшов випробувальний термін.

Значення цільової залежності $F: X \rightarrow Y$ відомі на об'єктах навчальної вибірки $\{(x_1, y_1), \dots, (x_n, y_n)\}$ (як уже було сказано вище, будь-яка рекрутингова компанія має інформацію про проходження та проживання). Потрібно побудувати алгоритм $k: X \rightarrow Y$, здатний здійснювати класифікацію довільного кандидата.

Тепер необхідно виділити фактори, які будуть включені до відповідної моделі. Як було з'ясовано у першому розділі, дуже багато чого про кандидата можна почерпнути з інформації, зазначеної в резюме. Однією з труднощів, з якими стикаються економетристи при оцінці параметрів моделей, є пропуск даних у статистиці. Вирішити цю проблему можна такими способами:

- видалити записи із пропусками даних;
- замінити відсутні значення на моду (даний спосіб добре працює у разі, якщо відсутніх даних мало; спочатку потрібно обчислити розподіл параметра, після чого замінити відсутні значення на його моду);
- відсутні дані можна змодельовати, наприклад, як у попередньому випадку, обчисливши за наявними даними розподіл параметра та провівши випадковий експеримент;
- також можна замінити відсутні рядки лінійною комбінацією інших рядків.

Втім, якщо йдеться про резюме, що надсилаються кадровому агентству, то дана проблема не повинна мати широкого поширення, тому що всі резюме заповнюються типовим чином. До того ж, у разі оцінки резюме, навіть відсутні дані можуть щось сказати про кандидата. Наприклад, відсутність у резюме запитуваної заробітної плати може говорити про те, що людина нездатна оцінити свою власну працю або просто недбало підійшла до процесу складання резюме.[7]

2.2 Модель та підхід до програмування

Модель бінарного вибору – модель залежності бінарної змінної, що застосовується в економетриці (що приймає всього два значення — 0 і 1) від сукупності факторів. Існують обмеження, пов'язані з використанням звичайної лінійної регресії, оскільки значення, що отримується при прогнозі має потрапляти у відрізок $[0; 1]$, а лінійна регресія допускає і негативні значення, і значення вище 1. Ця проблема вирішується з використанням інтегральних функцій розподілу. Найчастіше використовуються функції нормального розподілу (пробіт), логістичного розподілу (логіт) та розподілу Гомпертца

(гомпіт). Від вибору функції розподілу безпосередньо залежить відповідність прогнозів, отриманих з допомогою моделі, реальним даним.

Припускаючи, що залежна змінна Y , яка визначає можливість або неможливість прийняти на роботу здобувача (або, у випадку з кадровим агентством – рекомендувати його фірмі-клієнту), приймає лише два значення: $\{0, 1\}$, ймовірність того, що вона прийме відповідне значення можна виразити як функції деяких факторів (2.2) та (2.3):

$$Prob(Y = 1|x) = F(x^T \beta), \quad (2.2)$$

$$Prob(Y = 0|x) = 1 - F(x^T \beta). \quad (2.3)$$

Набір параметрів β відображає вплив зміни кожного фактора на ймовірність. Завдання полягає в тому, щоб підібрати адекватну функцію у правій частині рівняння.

Одна з можливостей – використовувати звичайну лінійну регресію (2.4):

$$F(x, \beta) = x^T \beta. \quad (2.4)$$

Оскільки $E[y|x] = 0[1 - F(x, \beta)] + 1[F(x, \beta)] = F(x, \beta)$, можна побудувати регресійну модель (2.5):

$$y = x^T \beta + s \quad (2.5)$$

Лінійна ймовірнісна модель має ряд недоліків, головний з них – те, що результат, отриманий за допомогою моделі може виходити за межі відрізка $[0; 1]$.

Тому, природно припустити такі умови, (2.6) та (2.7):

$$Prob(x) = 1 \quad (2.6)$$

$$Prob(x) = 0 \quad (2.7)$$

Для будь-якої інтегральної функції розподілу можна вважати, що вона

задовольнятиме таким умовам. Широкої популярності в економетриці набула так звана пробіт-модель, у якій використовується функція (2.8) стандартного нормального розподілу:

$$Prob(Y = y|x) = \int_{-\infty}^{x^T \beta} \phi(t) dt = \Phi(x^T \beta) \quad (2.8)$$

Також популярна логіт-модель, що використовує функцію (2.9) логістичного розподілу:

$$Prob(Y = y|x) = \frac{\exp(x^T \beta)}{1 + \exp(x^T \beta)} = \Lambda(x^T \beta). \quad (2.9)$$

І гомпіт-модель, що використовує функцію розподілу Гомпертца (2.10), так само має complementary log log model, або просто cloglog

$$Prob(Y = y|x) = 1 - \exp[-\exp(x^T \beta)]. \quad (2.10)$$

Оцінка параметрів β здійснюється методом максимальної ймовірності. Кожне спостереження є схемою Бернуллі, тому функція правдоподібності (2.11) з'являється у вигляді:

$$Prob(Y_1 = y_1, \dots, Y_n = y_n | \mathbf{X}) = \prod_{y_i=0} [1 - F(x_i^T \beta)] \prod_{y_i=1} F(x_i^T \beta). \quad (2.11)$$

Функцію правдоподібності n для спостережень (2.12) можна переписати у вигляді:

$$L(\beta | data) = \prod_{i=1}^n [F(x_i^T \beta)]^{y_i} [1 - F(x_i^T \beta)]^{1-y_i}. \quad (2.12)$$

Тепер запишемо рівняння правдоподібності:

$$\frac{d \ln L}{d \beta} = \sum_{i=1}^n \left[\frac{y_i f_i}{F_i} + (1 - y_i) \frac{-f_i}{(1 - F_i)} \right] x_i = 0. \quad (2.13)$$

Оскільки дані рівняння є нелінійними, вони вирішуються за допомогою чисельних методів, таких як багатовимірна інтерпретація методу Ньютона (14):

$$\beta^{j+1} = \beta^j - H^{-1}(\beta^j) \text{grad}L(\beta^j) \quad (2.14)$$

де H - матриця Гессе.

Особливість застосування бінарної регресії для оцінки працівника полягає в необхідності дати кількісну інтерпретацію якісним змінним: освіта, стать, навички та ін Досвід роботи може включати в себе так само і оцінку самих організацій, в яких працював кандидат, оцінку посад та інше.

Висновок по розділу

В цьому розділі розглянуто математичного моделювання процесу підбору персоналу.

Перший підхід до застосування кількісних методів у справі підбору персоналу, запропонований у дослідженні, ґрунтується на оцінці ймовірності проходження випробувального терміну працівником. Здійснена математична постановка задачі підбору персоналу.

Другий підхід до моделювання рекрутингу передбачає розгляд дій HR менеджера як завдання класифікації. Подана математична постановка завдання рекрутера як задача класифікації.

Розглянута особливість застосування бінарної регресії для оцінки працівника.

РОЗДІЛ 3. ПРОЦЕС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Постановка завдань проектування

Перед тим, як сформулювати вимоги до моделі, розглянемо, як за допомогою інформаційного простору відбувається взаємодія працівника та працедавця. Використання сучасного інформаційного простору у пошуку роботи та, відповідно, пошуку персоналу представлено на рис. 3.1.



Рисунок 3.1 Використання інформаційного простору для пошуку роботи та персоналу

Зауважимо, що працівником у цій схемі може бути, як кандидат (здобувач), який займається пошуком роботи вперше, так і людина, яка має певний стаж і працює нині.

Спираючись на цю схему, визначимо, як саме відбувається взаємодія працівника та роботодавця. Працівник створює резюме і розміщує його на спеціалізованих сайтах, виконує пошук вакансій по галузях (професійних областях) на позиції, що цікавлять. Роботодавець розміщує інформацію про

фірму, створює вакансію на спеціалізованих сайтах, виконує пошук резюме по галузях або звертається до кадрових (рекрутингових) агентств.

Взаємодія на рівні пошуку та аналізу здобувачем вакансії і, відповідно, пошуку та аналізу роботодавцем або HR-менеджером резюме накладає певні функціональні вимоги з обох сторін на роботу нейромережевої моделі.

Для претендента важливо знати:

- повну інформацію про фірму (організацію);
- інформацію про посади та вимоги, які пред'являються роботодавцем;
- умови роботи (зарплата, графік роботи, тип зайнятості).

Для роботодавця або HR-менеджера, у свою чергу, важливо знати якомога повніший портрет здобувача (стаття, вік, освіта, досвід роботи, очікувана зарплата, регіон проживання тощо).

Побудуємо схему, яка відобразатиме процес прийняття рішень. Відобразимо варіант застосування штучної нейронної мережі у цьому процесі та покажемо його пунктиром (рис. 3.2).

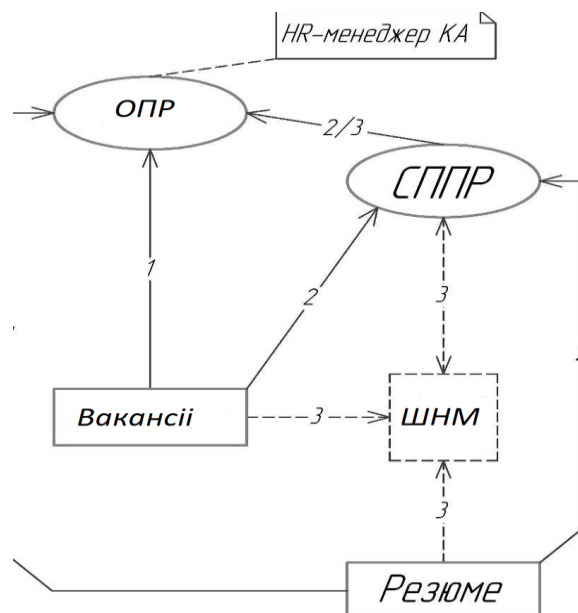


Рисунок 3.2. Схема прийняття рішення

- ОПР – особа, яка приймає рішення;
- СППР – система підтримки прийняття рішень;
- ШНМ – штучна нейронна мережа.

Розглядаючи цю схему, відзначимо таке. Остаточне рішення може бути прийнято декількома способами: лише особою, яка приймає рішення, за допомогою СППР, а також за допомогою СППР або аналітичної платформи, що включає алгоритм ШНМ (або побудованої на ньому).

3.2 Вибір технологій програмування

Для розробки програмного модулю було обрано мову програмування PYTHON. У програмі було розроблено 3 моделі з різними варіаціями. Розглянемо програмне рішення, яке було створене на основі цих моделей. Повний код програми можна подивитися у Додатку А.

3.3. Створення нейромережової моделі

Аналіз даних з інтернет-платформи по пошуку роботи дозволив сформулювати таблицю (базу даних), комірки якої містять значення параметрів резюме та вакансій (вимог вакансій). Фрагменти даної таблиці представлені у Додатку А.

У цій таблиці більше 1000 рядків, тобто вибірка складається з 1000+ резюме та вакансій. По кожному рядку, за підтримки спеціаліста кадрової служби, було винесено два рішення: перше – лише з аналізу базових параметрів, друге – сумарне. Стовпці у таблиці – параметри вакансій та резюме. У поданих фрагментах таблиці можна побачити значення всіх параметрів за першими 35 рядками.

Під час виконання роботи було створено моделі нейромережева модель, та моделі для порівняння:

- Модель лінійної регресії;
- Модель дерева рішень.

Розглянемо рішення для кожної з них та зробимо порівняння середніх помилок моделей. Порівняльна оцінка моделей дозволяє визначити середні помилки моделей.

Отримуємо наступний графік:

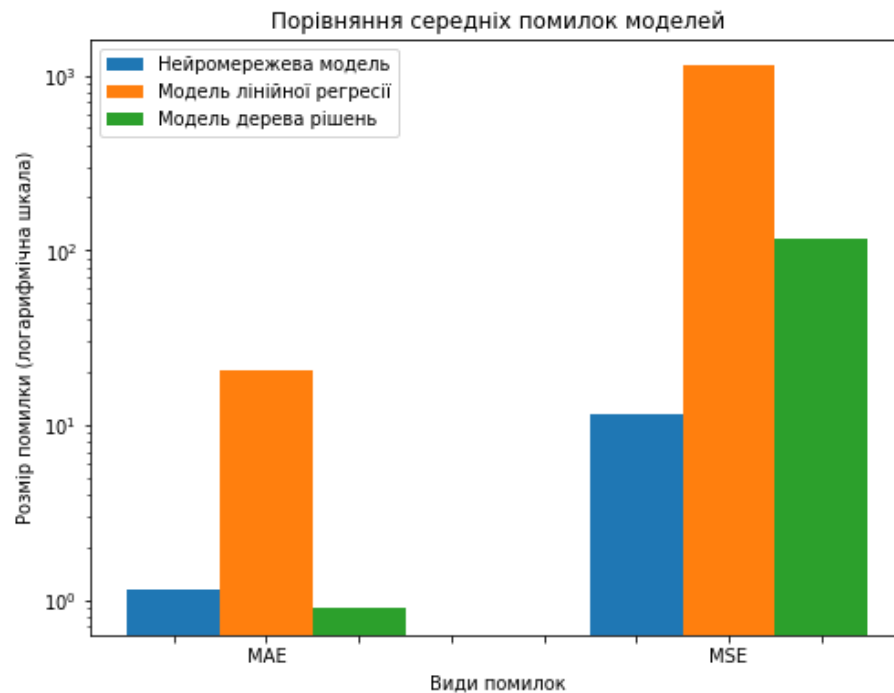


Рисунок 3.3 Порівняльний графік помилок моделей

На порівняльному графіку можна побачити, що характеристики якості інформаційної придатності моделей прогнозування описують, наскільки достовірно, обрана як генератор прогнозу, модель описує ретроспективу досліджуваного явища. Чим точніше побудована модель пояснювала минуле, тим більша ймовірність того, що вона вдало передбачатиме майбутнє.

Надійність моделей прогнозування оцінюється шляхом порівняння фактичних та передбачених значень. Ця різниця дозволяє перевірити, чи застосовна до конкретних даних розглянута модель і припущення, на яких

вона заснована. Основними оціночними характеристиками якості прогнозу моделі є нижченаведені показники.

Варто зазначити, що у нашому випадку найбільш якісною інформаційна придатність виявлено у моделі дерева рішень. Ця модель дуже проста і одночасно найбільш розповсюджена модель машинного навчання.

Наступна за надійністю є нейромережева модель, яка для всіх рядів з лінійним трендом є найбільш точною у порівнянні з моделлю регресії.

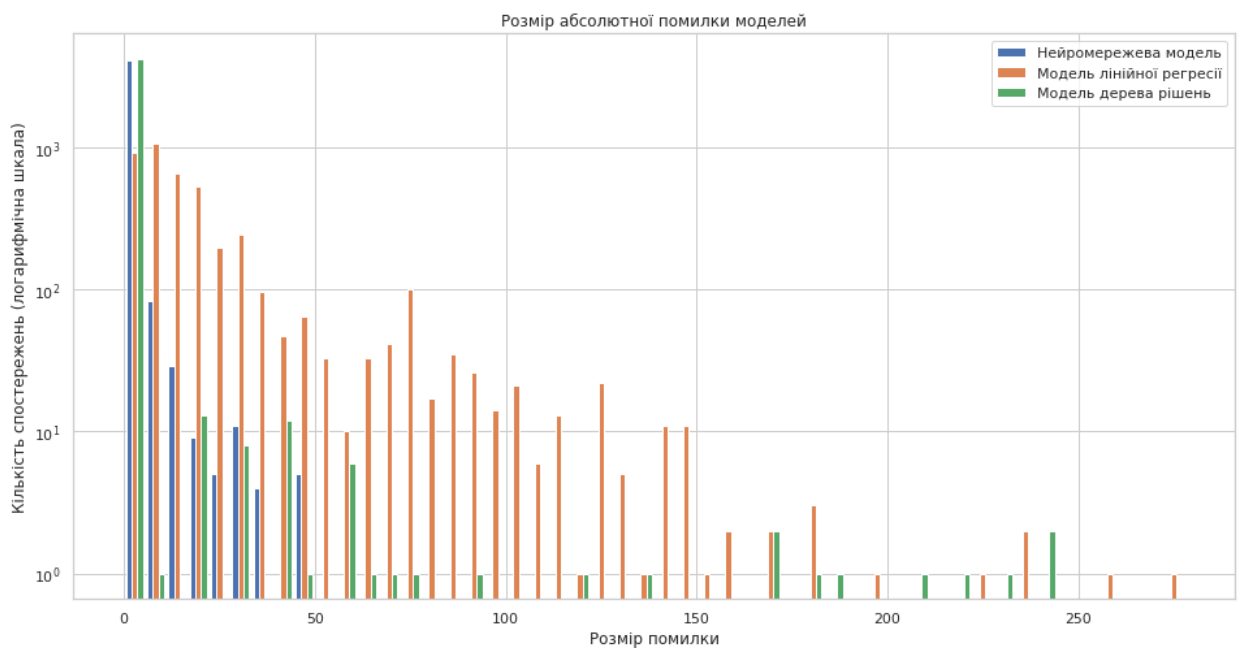


Рисунок 3.4 Порівняльний графік помилок моделей

Результати для обчислення абсолютної помилки моделей такий самий як і у першому випадку.

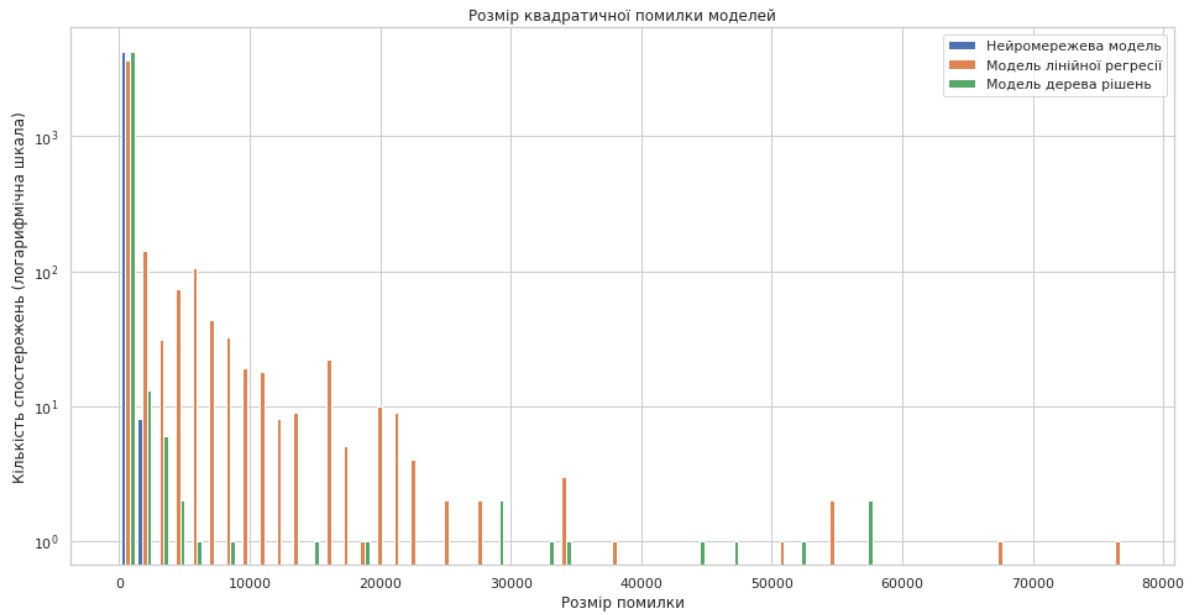


Рисунок 3.5 Порівняльний графік помилок моделей

У цьому випадку найкращий результат у нейромережевої моделі.

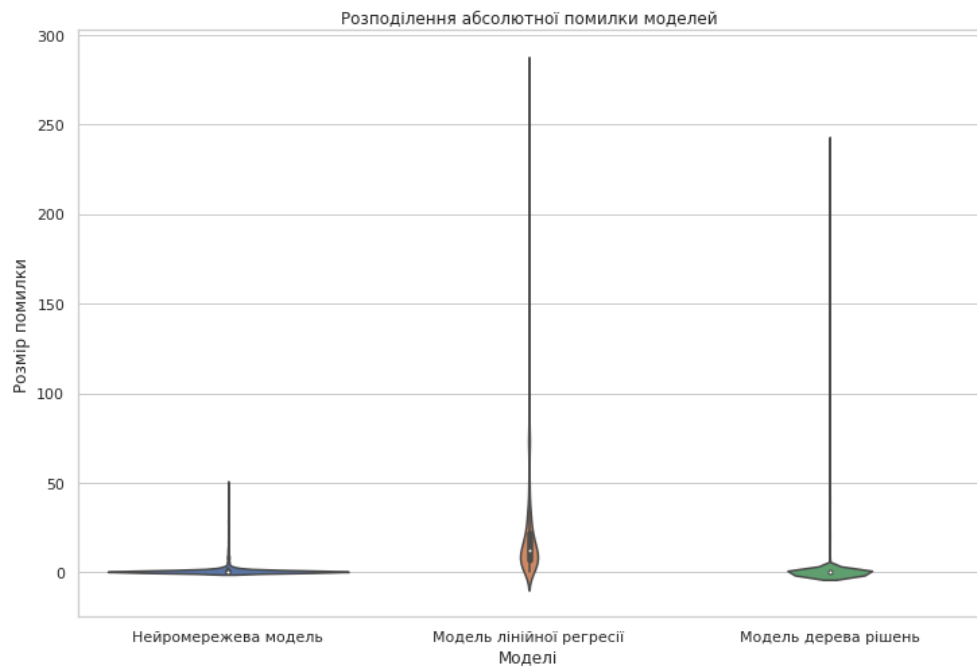


Рисунок 3.6 Порівняльний графік помилок моделей

При розподіленні абсолютної помилки моделей найкращий результат у нейромережевої моделі.

3.3.1 Вимоги до моделі

Перейдемо безпосередньо до вимог. Сформулюємо вимоги до функціональності моделі:

- модель має бути адаптована під сегмент ринку праці, орієнтований на підбір кадрів нижчої та середньої ланки (рекрутинг);
- модель повинна мати здатність до навчання, тобто. має відтворювати та узагальнювати досвід роботодавців або HR-менеджерів;
- параметри вакансій та резюме в даній моделі мають бути поділені на базові (БП) та додаткові (ДП);
- необхідне визначення відповідності резюме вимогам вакансій за базовими параметрами (БП);
- виходом для нейромережевої моделі має бути скалярна величина $\mu \in [0,1]$ загальної оцінки відповідності вимогам вакансії. [14,16]

3.3.2 Структура моделі

Далі безпосередньо охарактеризуємо структуру моделі класифікації, що відносить резюме кандидатів на основі їх ознак (значень параметрів) до одного з двох класів: 1 – резюме відповідає вимогам вакансії, 0 – резюме не відповідає вимогам вакансії. Враховуючи вимоги до моделі, описані в підрозділі 3.1, було вирішено використовувати дві багатошарові нейронні мережі, перша з яких вирішуватиме задачу відповідності резюме вимогам вакансій за базовими параметрами, друга – виносити сумарну оцінку відповідності. Уявімо схему використання цих нейронних мереж – на рис. 3.3.

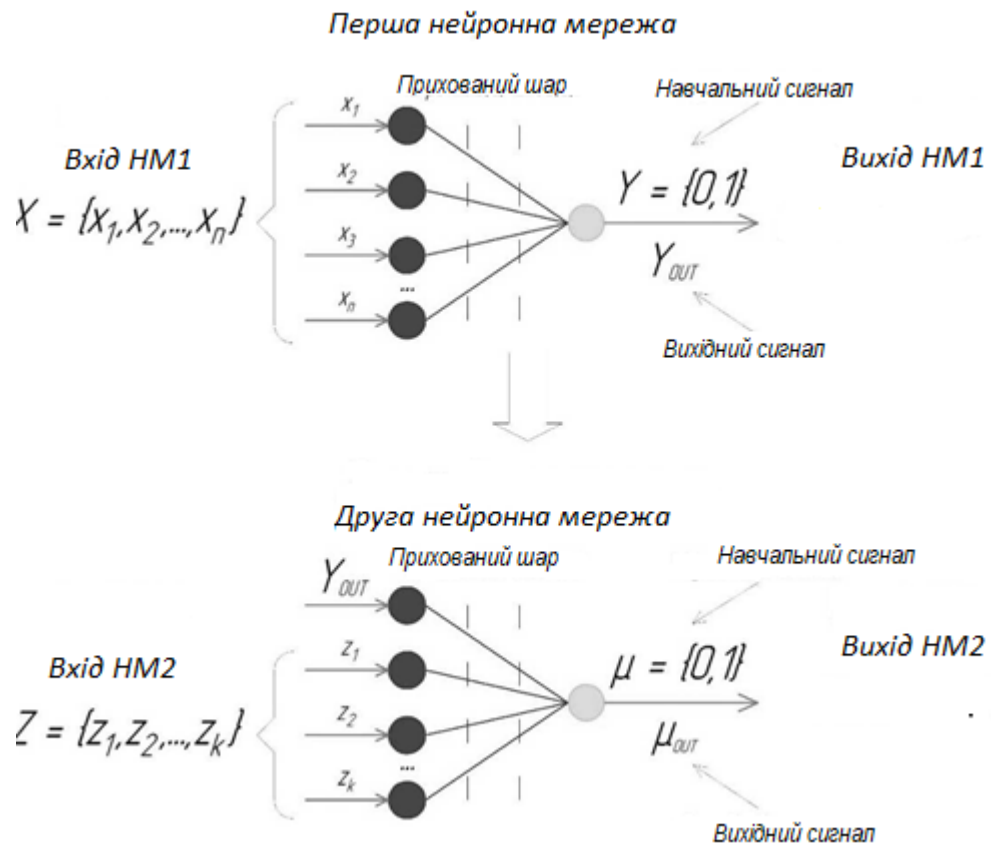


Рисунок 3.7. Схема використання нейронних мереж

$X = \{x_1, x_2, \dots, x_n\}$ – вхідні сигнали – значення базових параметрів резюме ($X = X_r$) і вакансій ($X = X_v$), $Z = \{z_1, z_2, \dots, z_k\}$ – значення додаткових параметрів резюме ($Z = Z_r$) і вакансій ($Z = Z_v$).

$Y = \{0, 1\}$ – навчальний (еталонний) сигнал на виході першої НМ, Y_{out} – вихідний сигнал, що формується на виході першої НМ. $\mu = \{0, 1\}$ – навчальний (еталонний) сигнал на виході другої НМ, μ_{out} – вихідний сигнал, що формується на виході другої НМ.

Для розуміння функціонування моделі розробимо загальну схему нейромережевого механізму. Розроблена схема представлена на рис. 3.4 і 3.5:

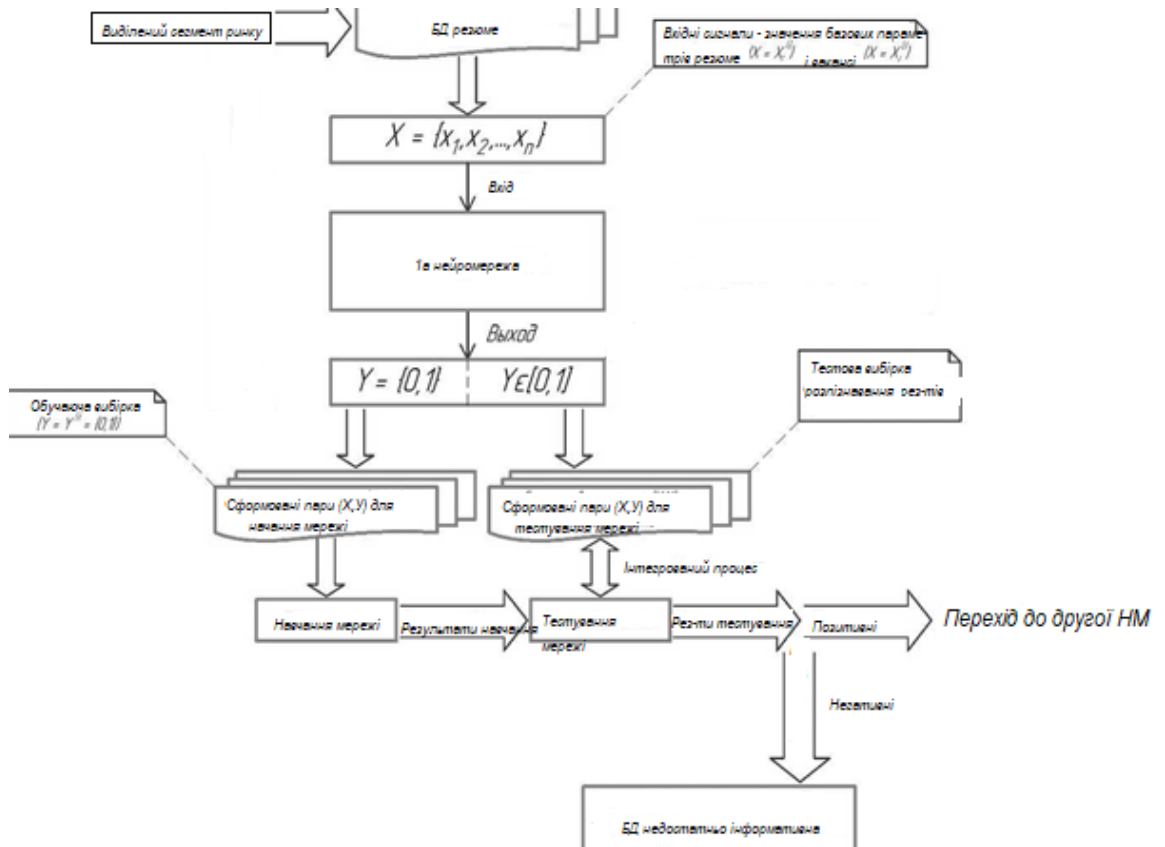


Рисунок 3.8 Схема нейромережевого механізму

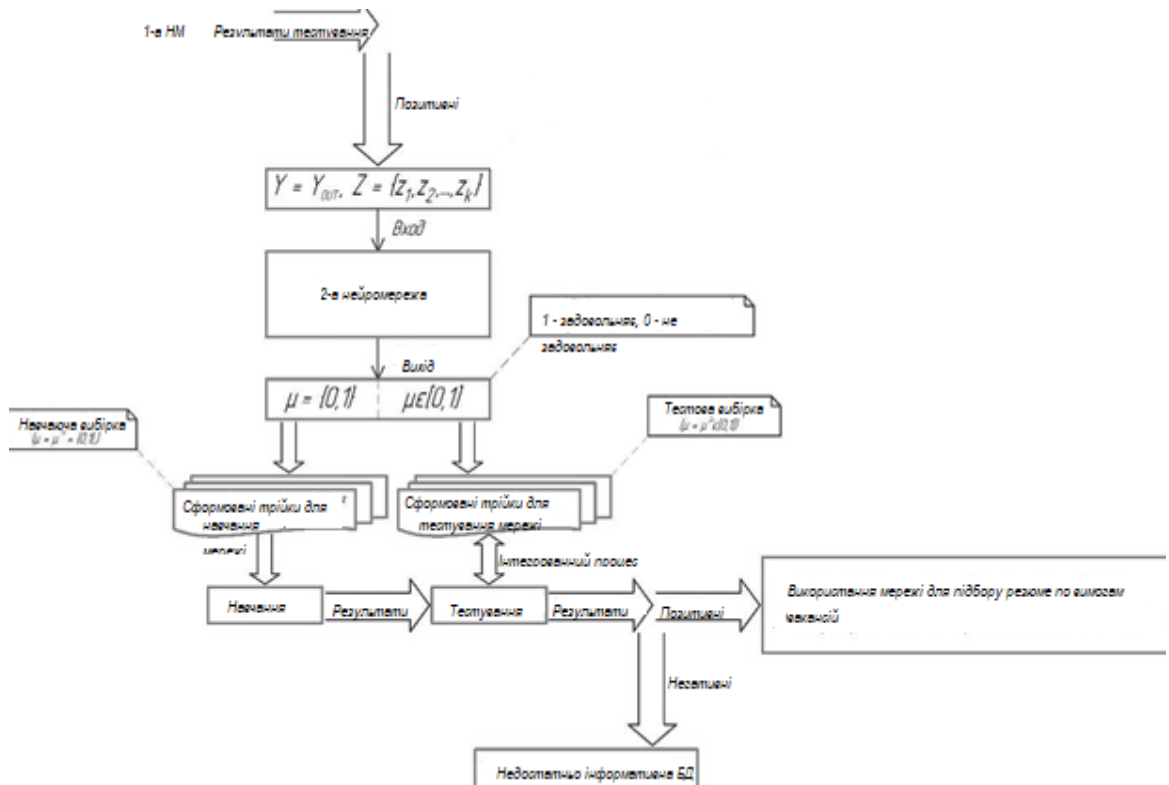


Рисунок 3.9 Схема нейромережевого механізму

Алгоритм роботи НМ можна зрозуміти за блок-схемою, побудованою нижче. Блок-схема застосовна до кожної із двох нейронних мереж. Розроблена блок-схема представлена на рис. 3.6.

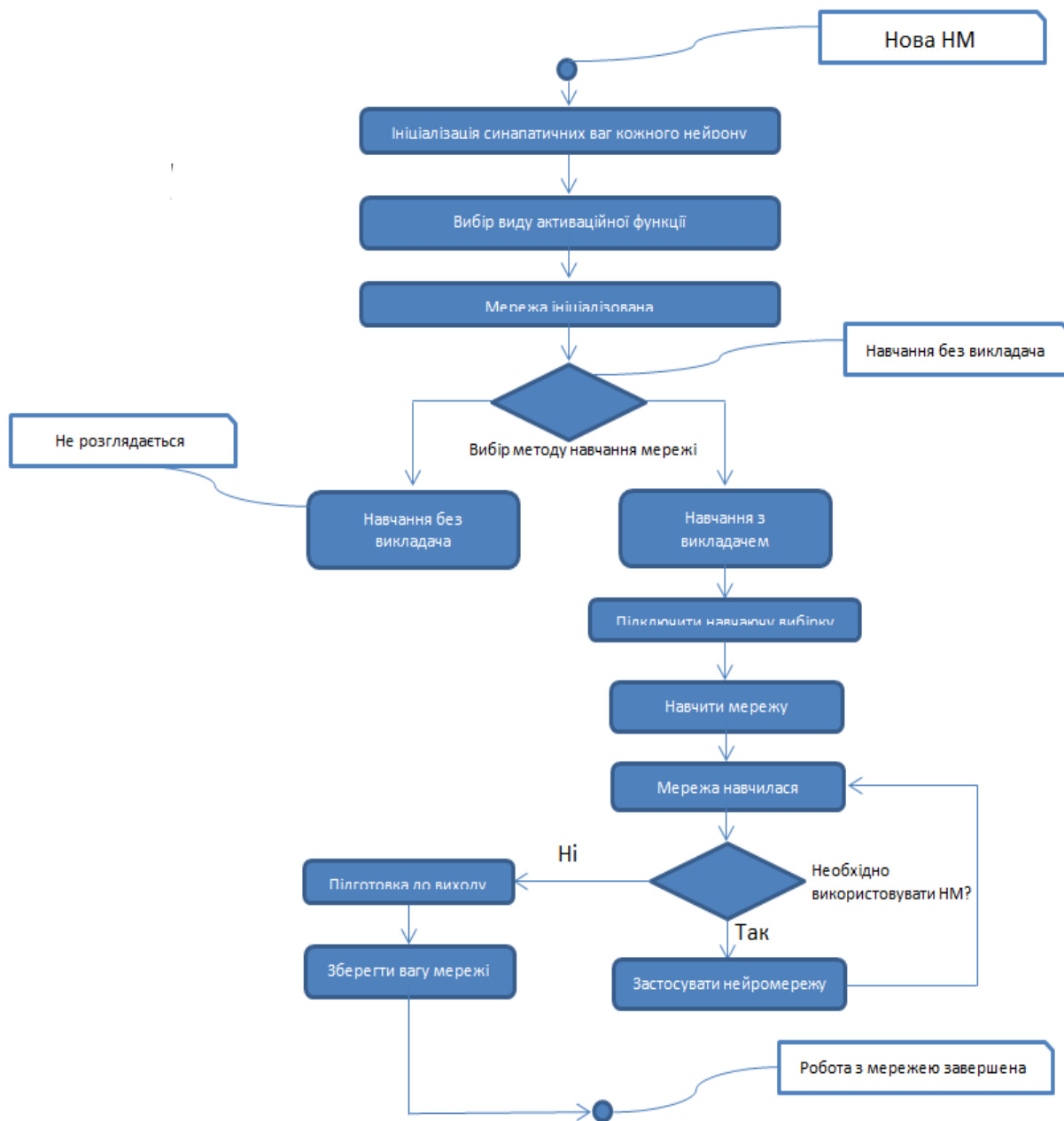


Рисунок 3.10. Схема роботи НМ

Як бачимо, алгоритм навчання складається з трьох рівнів: підбору структури, початкової активаційної функції та вагових коефіцієнтів.

3.4. Вхідні дані

Виявлені параметри шляхом розпізнавання основної інформації у резюме та вакансіях (парсингу) представлені на малюнку 3.8. Праворуч від параметра вказано його приналежність: резюме, вакансії чи резюме та вакансії.

382;27-04-2022;Разработчик сайтов;~25 років;Луцьк;Повна зайнятість	неповна зайнятість
383;27-04-2022;Front-end developer;17 000 грн;21 рік;Київ;Повна зайнятість	дистанційна робота.; ; НАУ Електроніка і телекомунікації
384;27-04-2022;Junior програмист;10 000 грн;20 років;Запоріжжя;Повна зайнятість	неповна зайнятість
385;27-04-2022;Front-end developer;~24 роки;Харків;Повна зайнятість	неповна зайнятість
386;27-04-2022;Python-програмист;20 000 грн;21 рік;Запоріжжя;Дистанційна робота.;Київ; СФУ ИМИФИ	Красноярск Незакінчена вища
387;27-04-2022;Javascript developer (front-end);~35 років;Харків;Повна зайнятість.; ; Kharkiv National University of Radio Electronics Computer Engineering and Management	Харків Вища
388;27-04-2022;Junior Java Developer;~21 рік;Київ;Повна зайнятість	неповна зайнятість
389;27-04-2022;Java-разработчик;~20 років;Київ;Неповна зайнятість	дистанційна робота.; ; Национальный технический университе
390;27-04-2022;Junior We Developer (.Net Core	Blazor);11 000 грн;20 років;Київ;Повна зайнятість
391;27-04-2022;Програмист;~26 років;Кам'янське (Дніпродзержинськ);Повна зайнятість	неповна зайнятість
392;27-04-2022;Android-разработчик;~27 років;Київ;Повна зайнятість	неповна зайнятість
393;27-04-2022;Junior/Middle С# Developer;~37 років;Черкаси;Повна зайнятість	неповна зайнятість
394;27-04-2022;Junior Python/C++ developer (Intern)	System;8 000 грн;20 років;Київ;Неповна зайнятість
395;27-04-2022;Trainee Front-End Developer;~32 роки;Київ;Повна зайнятість.; ; ХНУ ім.В. Н. Каразіна Геолого-Географічний	Харків Вища
396;27-04-2022;Системный администратор	IT-специалист
397;27-04-2022;HTML-верстальщик	Frontend Developer;15 000 грн;26 років;Львів;Дистанційна робо
398;27-04-2022;Frontend developer;25 000 грн;19 років;Київ;Повна зайнятість	неповна зайнятість
399;27-04-2022;Програміст	фахівець з інформаційних технологій
300;27-04-2022;Front-end Developer;15 000 грн;23 роки;Житомир;Повна зайнятість	неповна зайнятість
301;27-04-2022;Salesforce developer;~27 років;Кривий Ріг;Неповна зайнятість	дистанційна робота.; ; Криворожский Экономический Институ
302;27-04-2022;Javascript developer;~20 років;Харків;Повна зайнятість.; ; Национальный технический университет "Харковский политехнический институт" Автоматизация и компью	Харків Вища
303;27-04-2022;Front-end програміст;~25 років;Одеса;Повна зайнятість	неповна зайнятість
304;27-04-2022;Javascript-програміст;10 000 грн;21 рік;Київ;Неповна зайнятість	дистанційна робота.; ; КПІ ім. Ігоря Сікорського ФІОТ
305;27-04-2022;Web developer;~23 роки;Ужгород;Повна зайнятість	неповна зайнятість

Рисунок 3.8 Вхідні дані

Далі виконаємо структурування виявлених параметрів з метою їхнього розподілу за так званими модулями. Ця процедура необхідна для кращого розуміння структур резюме і вакансії і, як наслідок, для аналізу інформації, що подається на нейронну мережу. Наприклад, для резюме таких модулів, як бачимо, було виділено 7. Зазначимо, що назва даним модулям було присвоєно самостійно, але деякі з них можуть частково або повністю повторювати окремі пункти резюме, складені на вищезгаданих інтернет-платформах.

Структурування параметрів резюме показано нижче:

1. Освіта
2. Додаткова освіта
3. Навички MySQL
4. Навички .NET
5. Досвід
6. Знання англійської мови

7. Знання інших мов (окрім англійської, української та російської)

Наступним етапом роботи буде визначення можливих значень, які може набувати той чи інший параметр. Зрозуміло, що це стосується не всіх, а певної сукупності представлених параметрів. Тобто до тієї сукупності, для якої необхідно позначити будь-які метрики. Після того, як значення параметрів стануть відомими, буде проведено класифікацію всіх параметрів за певними типами даних.

Слід зазначити, що з різних підходів до складання резюме і вакансій, і навіть різнорідності самих вакансій і резюме, параметрам «Обов'язки» і «Ключові навички» надалі може бути присвоєно лише чисельні значення.

3.4.1. Виявлення та уявлення параметрів

Проведемо класифікацію параметрів за типами даних, з прийнятих ними значень, визначених етапом раніше. Дані за своєю природою різноманітні. Умовно виділимо такі типи даних: якісні (категоріальні та ординальні), кількісні та булеві. Розподілимо параметри за виділеними типами:

1. Якісні дані

- ПІБ
- Місто
- Посада
- Професійна галузь

1.1. Категоріальні

- Стать
- Сімейний стан

1.2. Ординальні (порядкові)

- Освіта (рівень)

- Досвід роботи у сфері
- Тип зайнятості
- Знання англійської мови

2. Кількісні дані

- Вік (у роках)
- Обов'язки (кількість)
- Ключові навички (кількість)
- Зарплата

3. Булеві

- Телефон
- Електронна пошта
- Адреса
- Додаткова освіта (наявність)
- Додаткові мови

Зазначимо, параметри, що приймають значення «вказано, не вказано», «вказано, не вказано», «є, ні», віднесені нами до булевого типу даних, тепер матимуть одне значення – «0 або 1».

При вирішенні завдання підбору резюме за вимогами вакансій вирішено використовувати нейронну мережу, що самонавчається, мовою програмування Python.

3.4.2. Попередня обробка даних

Наступним етапом буде попередня обробка значень параметрів, що складається з кодування (для значень, що мають рядковий тип) та нормалізації.

Опишемо процедуру нормалізації числових даних. Для приведення значень діапазону $[a, b]$ використовується наступна формула:

$$z^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \cdot (b - a) + a \quad (3.1)$$

де x – поточне значення, x_{\min} , x_{\max} – мінімальні та максимальні значення поля.

Для рядкового типу даних у зазначеній платформі є нормалізатори «Унікальні значення» та «Битова маска». Опишемо їх докладніше. Нормалізатор «Унікальні значення» використовується, якщо значення поля можна впорядкувати, наприклад, у порядку: А, В, С, тоді будуть присвоєні наступні значення: А – 0, В – 1, С – 2. Потім ці значення наводяться до необхідного діапазону, як це було описано вище для числових даних.

Нормалізатор «Битова маска» передбачає 2 способи кодування:

- кодування «Комбінація бітів»;
- кодування «Позиція біта».

Використовуючи кодування «Комбінація бітів», кількість необхідних параметрів входів визначається за такою формулою:

$$n = \log_2(m) \quad (3.2)$$

де m – кількість значень параметра, які потрібно закодувати;
 n – кількість входів (кількість чисел у масці).

Використовуючи спосіб «Позиція біта», кодування відбувається шляхом заміни маску з m чисел. Перше унікальне значення має код, де крайнім правим числом є, інші – а. Кожне наступне отримує новий код, шляхом зміщення числа одну позицію. Останнє унікальне значення має код, де число знаходиться в крайній лівій позиції.[17,18]

3.5. Вихідні дані

Значення залежності $X \rightarrow Y$ відомі з навчальної вибірки. Завдання полягає в тому, щоб побудувати алгоритм $k: X \rightarrow Y$, який здійснював би підбір довільного резюме шляхом віднесення його до одного з двох класів: 1 – резюме відповідає вимогам вакансії, 0 – резюме не відповідає вимогам вакансії.

Нагадаємо, що $Y = \{0, 1\}$ – навчальний (еталонний) сигнал на виході першої НС, Y_{OUT} – вихідний сигнал, що формується на виході першої НМ. Відповідно, $\mu = \{0, 1\}$ – навчальний (еталонний) сигнал на виході другої НМ, μ_{OUT} – вихідний сигнал, що формується на виході другої НМ.

Числові значення вихідного сигналу легко інтерпретовані при розв'язанні даної задачі і дозволяють співвідносити отримане на виході мережі значення з еталонним, звертаючи увагу на помилку, що було б неможливо у разі пред'явлення вихідному сигналу «якісних» значень, тобто значень, що мають рядковий тип.[19]

3.6 Наукова новизна проекту

Найбільш вагомими результатами, що становлять наукову новизну є такі:

- удосконалено процес відбору кандидатів на конкретну вакансію;
- орієнтація на інноваційну спрямованість і потенційну спроможність підтримати інноваційний розвиток підбору персоналу;
- скорочено терміни пошуку ідеальних кандидатів на будь-яку посаду;
- висунуто пропозиції, реалізація яких надалі дозволить враховувати переваги конкретних роботодавців, виражені у вакансіях на будь-які позиції, або шляхом аналізу їх вимог експертами предметної галузі

Запропонований у роботі підхід до побудови нейромережевої моделі дозволяє враховувати всі можливі значення параметрів, які у конкретної

вибірці. Таким чином, модель застосовна для різних професійних областей, що, у свою чергу, є істотною перевагою для рекрутингових агенцій, що надають послуги організаціям різного профілю.

Наукова новизна одержаних результатів полягає у тому, що робота є одним з перших системним дослідженням процесів найму працівників та розвитку технологій, практичних рекомендацій щодо підвищення ефективності відбору кандидатів на українській мові.

–розроблено концепцію розвитку процесу найму;

–використано методичний підхід до дослідження процесу пошуку кандидатів на ту чи іншу посаду;

–досліджено сутність і структуру нейронних мереж для пошуку вакансій та резюме;

–визначено стратегічні напрями розвитку програмного рішення.

Метою є вибір найкращої альтернативи, яка відповідає пошуковому образу конкретного запиту, та подальше ранжування списку альтернатив від найкращого до найгіршого.

Висновок по розділу

У даному розділі були описані етапи по побудові моделі, а саме, сформульовані вимоги до моделі, розроблена структура, що передбачає використання двох багатосарових НМ, виявлено та представлено вхідні параметри, наведено опис їхньої попередньої обробки. Визначено та охарактеризовано вихідні параметри.

ВИСНОВОК

У рамках кваліфікаційної роботи виконано дослідження можливості застосування штучних нейронних мереж для параметричного підбору резюме, що настроюється, за вимогами вакансій.

У розділі 1 проведено порівняльний аналіз процедур відбору та підбору персоналу. Зроблено висновок, що аналіз відповідності резюме вимогам вакансії та аналіз відповідності кандидата вимогам роботодавця тотожно рівні.

Проведено аналіз ефективності сучасних методів підбору персоналу, за результатом якого зроблено висновок, що методом, що найбільш підходить для моделювання, є рекрутинг. Даний метод надає можливість моделювати процес підбору резюме за вимогами вакансій, спираючись на інформацію, вказану безпосередньо у резюме та вакансії. При ексклюзивному пошуку або хедхантинг практично повністю відсутня необхідна статистична інформація, що значно ускладнює процес моделювання.

Також було проведено огляд існуючих рішень, що використовуються у системі керування персоналом. Для кращого розуміння завдань, у яких має місце їх застосування, розглянуті рішення були розбиті на групи: проаналізовано інтернет-сервіси, програмні продукти, а також моделі та підходи, засновані на математичних методах.

У другому розділі було розглянуто моделі та підходи, засновані на математичних методах. Було розглянуто підбір персоналу з точки зору математичного моделювання. Третій розділ було присвячено побудові моделей та програмному рішенню завдання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аллін, О.М., Кадри ефективного бізнесу. Підбір та мотивація персоналу/О.М. Аллін, Н.І. Сальникова – М.: Генезіс, 2005. – 248 с.
2. Баскіна, Т., Техніки успішного рекрутменту/Т. Баскіна. – 2-ге вид., Перероб. та дод. – М.: Альпіна Паблішер, 2014. – 288 с.
3. Бодянський, Є. В., Штучні нейронні мережі: архітектури, навчання, застосування: монографія / О. В. Бодянський, О.Г. Руденко. – Харків: Телетех, 2004. – 369 с.
4. Ларіонов, Г.В., Інноваційне формування тенденцій управління людськими ресурсами: монографія / Г. В. Ларіонов. – М: Видавничо-торгівельна корпорація «Дашков і К °», 2014. – 160 с.
5. Риков, А. С., Системний аналіз: моделі та методи прийняття рішень та пошукової оптимізації / А. С. Риков. – М.: МІСіС, 2009. – 608 с.
6. Cohen, J., Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences / P. Cohen, S. G. West, L. S. Aiken. – 3rd edition. – London: Lawrence Erlbaum Associates Publishers, 2003. – 736 p.
7. Galushkin, A. I., Neural Networks Theory / A. I. Galushkin. – New York: Springer-Verlag Berlin Heidelberg, 2007. – 396 p.
8. Hoffmann, J., Generalized linear models: An applied approach / J. Hoffmann. – Boston, MA: Allyn & Bacon, 2003. – 216 p.
9. Hosmer, D. W., Applied logistic regression / David W. Hosmer, Jr., Stanley Lemeshow, R. X. Sturdivant. – 3rd edition. – New York: John Wiley & Sons INC, 2013. – 528 p.
10. Izenman, A. J., Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning Springer / A.J. Izenman. – New York: Springer-Verlag, 2008. – 760 p.
11. Kohonen, T., Self-Organizing Maps / T. Kohonen. – 3rd edition. – New York: Springer-Verlag Berlin Heidelberg, 2001. – 501 p.

12. Haykin, S., *Neural Networks and Learning Machines* / S. Haykin. – 3rd edition. – London: Pearson Education Inc., 2009. – 906 p
13. Азарнова, Т. В., Застосування нейромереж у процесі добору персоналу/Т.В. Азарнова, І. Н. Тернових // Вісник ВДУ. – 2009. – № 2. – С. 76-80.
14. Алгоритм навчання RProp – математичний апарат [Електронний ресурс]. – Режим доступу: HTTPS
15. Галушкін А. І., *Нейронні мережі: основи теорії* / А. І. Галушкін. – М.: Горяча лінія – Телеком, 2012. – 496 з
16. Бодянський, Є. В., *Штучні нейронні мережі: архітектури, навчання, застосування: монографія* / О. В. Бодянський, О. Г. Руденко. – Харків: Телетех, 2004. – 369 с.
17. Георгіца І.В., Кузнєцова А.В., Уявлення знань системи підбору персоналу з урахуванням нечіткої логіки // Сучасні проблеми науку й освіти. – 2015. – № 1-1.
18. Карелін В. П., *Моделі та методи представлення знань та вироблення рішень в інтелектуальних інформаційних системах з нечіткою логікою* // Вісник ТІУіЕ. – 2014. – №1 (19). – С. 75-83.
19. Мухамедієва Д. Т., *Моделі та алгоритми нечіткого виведення в завданнях інтелектуального аналізу даних* // Проблеми обчислювальної та прикладної математики. – 2016. – №3 (5). – С. 55-62.
20. Платформа для підбору сервісів – <https://startpack.ru/> [Електронний ресурс]. – Режим доступу: HTTPS

ДОДАТОК А

Парсинг інформації про освіту

```
def get_education_vector(row):
    vector = [0, 0]
    if row['education'].strip() == '-' and row['additional_info'].strip() == '-':
        return vector
    else:
        string = row['education'] + ' ' + row['additional_info']
        try:
            re.search('Вища|[Мм]агістр|Магістр|[Бб]акалавр|[Мм]астер|[Вв]achelor',
string).group(0)
                vector[0] = 1
                return vector
            except:
                pass
        try:
            re.search('[Нн]езакінчена вища', string).group(0)
            vector[1] = 1
            return vector
        except:
            return vector
# Парсинг інформації про додаткову освіту
def get_additional_edu_vector(row):
    vector = [0]
    if row['additional_education'].strip() == '-' and row['additional_info'].strip()
== '-':
        return vector
    else:
        string = row['additional_education'] + ' ' + row['additional_info']
        try:
            re.search('Вища|Master|Bachelor', string).group(0)
            vector[0] = 1
            return vector
        except:
            pass
```

```

try:
    re.search('курс|сертификат|сертифікат|course|certificate',
string).group(0)
    vector[0] = 1
    return vector
except:
    return vector
# Парсинг інформації про навички MySQL
def get_mysql_vector(row):
    vector = [0, 0, 0]
    if row['skills'].strip() == '-' and row['additional_info'].strip() == '-':
        return vector
    else:
        string = row['skills'] + ' ' + row['additional_info']
        string = string.lower()
        try:
            re.search('my\s*sql', string).group(0)
            vector[0] = 1
            return vector
        except:
            pass
        try:
            re.search('sql', string).group(0)
            vector[1] = 1
            return vector
        except:
            pass
        try:
            re.search('db|database|баз[аи] даних|баз[аы] данных|субд',
string).group(0)
            vector[2] = 1
            return vector
        except:
            return vector
# Парсинг інформації про навички .net
def get_dotnet_vector(row):
    vector = [0, 0]

```

```

if row['skills'].strip() == '-' and row['additional_info'].strip() == '-':
    return vector
else:
    string = row['skills'] + ' ' + row['additional_info']
    string = string.lower()
    try:
        re.search('\.s*net', string).group(0)
        vector[0] = 1
        return vector
    except:
        pass
try:
    re.search('c#|f#|visual basic|vb|visual c++', string).group(0)
    vector[1] = 1
    return vector
except:
    return vector
# Парсинг інформації про мови
def get_lang_from_lang(row):
    english_vector = [0, 0, 0]
    others_vector = [0, 0, 0]
    string = row['languages']
    lang_list = re.findall('[А-Я]\w+\s—\s\w+\s*', string)
    other_langs = 0
    for item in lang_list:
        if 'Українська' in item or 'Російська' in item:
            continue
        elif 'Англійська' in item:
            if 'вільно' in item or 'просунутий' in item:
                english_vector[0] = 1
            elif 'середній' in item or 'вище' in item:
                english_vector[1] = 1
            elif 'початковий' in item:
                english_vector[2] = 1
        else:
            other_langs += 1
    if other_langs >= 3:

```

```

    others_vector[2] = 1
elif other_langs == 2:
    others_vector[1] = 1
elif other_langs == 1:
    others_vector[0] = 1
return english_vector + others_vector
def get_lang_from_additional(row):
    english_vector = [0, 0, 0]
    others_vector = [0, 0, 0]
    lang_list = ['polish|польсь*к', 'spanish|іспаньск|испанск',
                 'german|н[іе]мець*к', 'italian|італійськ|итальянск',
                 'slovak|словаць*к', 'slovenian|словенсь*к', 'czech|чеськ|чешск']
    string = row['additional_info'].lower()
    if 'languages' in string or 'знання мов' in string or 'знание языков' in
string:
        try:
            re.search('english', string).group(0)
            english_vector[1] = 1
        except:
            pass
        try:
            re.search('англійськ|ангийск', string).group(0)
            english_vector[2] = 1
        except:
            pass
        other_langs = 0
        for lang in lang_list:
            try:
                re.search('англійськ|ангийск', string).group(0)
                other_langs += 1
            except:
                continue
        if other_langs == 0:
            pass
        elif other_langs == 1:
            others_vector[0] = 1
        elif other_langs == 2:

```

```

        others_vector[1] = 1
    elif other_langs >= 3:
        others_vector[2] = 1
    return english_vector + others_vector
def get_languages_vector(row):
    english_vector = [0, 0, 0]
    others_vector = [0, 0, 0]
    if row['languages'].strip() == '-' and row['additional_info'].strip() == '-':
        return english_vector + others_vector
    elif row['languages'].strip() != '-':
        vector = get_lang_from_lang(row)
        return vector
    else:
        vector = get_lang_from_additional(row)
        return vector
# Функція рахує кількість місяців
def calc_months_number(exp):
    numbers = re.findall('\d{1,2}\s', exp)
    if len(numbers) == 2:
        months = int(numbers[0].strip())*12 + int(numbers[1].strip())
    elif 'місяц' in exp:
        months = int(numbers[0].strip())
    else:
        months = int(numbers[0].strip())*12
    return months
def get_exp_from_eng_add(string):
    months = 0
    years_list = re.findall('\d{1,2}?.?year', string)
    if len(years_list) > 0:
        for item in years_list:
            months += int(re.search('\d+', item).group(0))*12
    months_list = re.findall('\d{1,2}?.?month', string)
    if len(months_list) > 0:
        for item in months_list:
            months += int(re.search('\d+', item).group(0))
def get_exp_from_uaru_add(string):
    months = 0

```

```

years_list = re.findall('\d{1,2}.?год|\d{1,2}.?лет)', string)
if len(years_list) == 0:
    years_list = re.findall('\d{1,2}.?рік|\d{1,2}.?років)', string)
if len(years_list) > 0:
    for item in years_list:
        months += int(re.search('\d+', item).group(0))*12
months_list = re.findall('\d{1,2}.?М[іе]сяц', string)
if len(months_list) > 0:
    for item in months_list:
        months += int(re.search('\d+', item).group(0))
# Парсинг інформацію про досвід
def get_experience_vector(row):
    experience_period = 0
    vector = [0, 0, 0, 0, 0]
    if row['experience'].strip() == '-' and row['additional_info'].strip() == '-':
        return vector
    elif row['experience'].strip() != '-':
        string = row['experience'].lower()
        exp_list = re.findall('\((?:\d{1,2}\s\w+\s*)+\)', string)
        for exp in exp_list:
            experience_period += calc_months_number(exp)
    else:
        string = row['additional_info'].lower()
        try:
            exp_string = re.search('work(?:ing)* experience.*|employment
experience.*')
            experience_period = get_exp_from_eng_add(exp_string)
        except:
            try:
                exp_string = re.search('опыт работы|досвід роботи.*')
                experience_period = get_exp_from_uaru_add(exp_string)
            except:
                pass
    if experience_period < 12:
        pass
    elif experience_period < 24:
        vector[0] = 1

```

```

elif experience_period < 36:
    vector[1] = 1
elif experience_period < 48:
    vector[2] = 1
elif experience_period < 72:
    vector[3] = 1
elif experience_period >= 72:
    vector[4] = 1
return vector
# Функція парсингу таблиці
def parse_table(df):
    data = []
    functions = [get_education_vector, get_additional_edu_vector,
get_mysql_vector,
get_dotnet_vector, get_languages_vector, get_experience_vector]
    for i in range(len(df)):
        row = df.iloc[i, :]
        result_vector = []
        for func in functions:
            result_vector.extend(func(row))
        data.append(result_vector)
    data = np.array(data)
    print(data.shape)
    return data
Оцінка моделей
# Імпорт метрик оцінки
from sklearn.metrics import mean_squared_error, mean_absolute_error
# Функція підрахунку кількості спостережень з помилками вище деяких
порогових значень
def get_count_in_error_range(errors):
    count = []
    error_0_1 = np.where((errors >= 0) & (errors < 1))
    count += [len(errors[error_0_1])]
    error_1_2 = np.where((errors >= 1) & (errors < 2))
    count += [len(errors[error_1_2])]
    error_2_3 = np.where((errors >= 2) & (errors < 3))
    count += [len(errors[error_2_3])]

```

```

error_3_5 = np.where((errors >= 3) & (errors < 5))
count += [len(errors[error_3_5])]
error_5_10 = np.where((errors >= 5) & (errors < 10))
count += [len(errors[error_5_10])]
error_10_20 = np.where((errors >= 10) & (errors < 20))
count += [len(errors[error_10_20])]
error_20_plus = np.where(errors >= 20)
count += [len(errors[error_20_plus])]
return count

```

Оцінка нейромережевої моделі

Функція завантаження моделі з файлу

```
from keras.models import load_model
```

Завантаження моделі

```
model = load_model('/content/model_neuro.h5')
```

Отримаємо передбачення моделі на тестовій виборці

```
prediction = model.predict(x_test)
```

```
prediction = prediction.flatten()
```

Середня абсолютна та квадратична помилки

```
abs_error = abs(prediction - y_test)
```

```
sqr_error = abs_error**2
```

```
MAE_neuro = sum(abs_error) / len(abs_error)
```

```
MSE_neuro = sum(sqr_error) / len(sqr_error)
```

```
print('Середня абсолютна помилка: ', MAE_neuro)
```

```
print('Середня квадратична помилка: ', MSE_neuro)
```

Виведемо графік абсолютної та квадратичної помилок в тестовій

виборці

```
plt.figure(figsize=(16, 8))
```

```
plt.bar(range(len(abs_error)), abs_error,
```

```
label='Абсолютна помилка нейромережевої моделі в тестовій
```

виборці')

```
plt.xlabel('Спостереження')
```

```
plt.ylabel('Значення помилки')
```

```
plt.legend()
```

```
plt.show()
```

Підрахуємо кількість спостережень з помилками в деяких діапазонах

```
count = get_count_in_error_range(abs_error)
```

```

# Виведемо графік розподілу спостережень в деяких діапазонах
абсолютної помилки
error_thresholds = ['0 - 1', '1 - 2', '2 - 3', '3 - 5', '5 - 10', '10 - 20', '20+']
plt.figure(figsize=(16, 8))
plt.bar(error_thresholds, count,
        label=('Кількість спостережень, в різних діапазонах\n'
              'абсолютної помилки нейромережевої моделі'))
plt.xlabel('Діапазони значень помилки')
plt.ylabel('Кількість спостережень')
plt.legend()
plt.show()
# Виведемо 10 випадкових прикладів з бази
start_idx = random.randint(0, len(y_test)-10)
print('Реальні значення\n', y_test[start_idx:start_idx+10])
print('Прогнозовані нейромережею значення\n',
      np.array([round(x) for x in prediction[start_idx:start_idx+10]]))
Оцінка моделі лінійної регресії
# Перебачення моделі
prediction_lin_reg = linear_regression_model.predict(x_test)
abs_error_lin_reg = abs(prediction_lin_reg - y_test)
sqr_error_lin_reg = abs_error_lin_reg**2
MAE_linreg = sum(abs_error_lin_reg) / len(abs_error_lin_reg)
MSE_linreg = sum(sqr_error_lin_reg) / len(sqr_error_lin_reg)
print('Середня абсолютна помилка моделі лінійної регресії: ',
MAE_linreg)
print('Середня квадратична помилка моделі лінійної регресії: ',
MSE_linreg)
plt.figure(figsize=(16, 8))
plt.bar(range(len(abs_error_lin_reg)), abs_error_lin_reg,
        label='Абсолютна помилка моделі лінійної регресії в тестовій
вибірці')
plt.xlabel('Спостереження')
plt.ylabel('Значення помилки')
plt.legend()
plt.show()
count = get_count_in_error_range(abs_error_lin_reg)
error_thresholds = ['0 - 1', '1 - 2', '2 - 3', '3 - 5', '5 - 10', '10 - 20', '20+']

```

```

plt.figure(figsize=(16, 8))
plt.bar(error_thresholds, count,
        label=('Кількість спостережень, в різних діапазонах\n'
              'абсолютної помилки моделі лінійної регресії'))
plt.xlabel('Діапазони значень помилки')
plt.ylabel('Кількість спостережень')
plt.legend()
plt.show()
start_idx = random.randint(0, len(y_test)-10)
print('Реальні значення\n', y_test[start_idx:start_idx+10])
print('Прогнозовані моделлю лінійної регресії значення\n',
      np.array([round(x) for x in prediction_lin_reg[start_idx:start_idx+10])))

```

Оцінка моделі дерева рішень
Порівняльна оцінка моделей

```

fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot()
x = np.arange(2)
y1 = [MAE_neuro, MSE_neuro]
y2 = [MAE_linreg, MSE_linreg]
y3 = [MAE_tree, MSE_tree]
w = 0.2
ax.bar(x, y1, label='Нейромережева модель', width=w)
ax.bar(x + w, y2, label='Модель лінійної регресії', width=w)
ax.bar(x + 2*w, y3, label='Модель дерева рішень', width=w)
ax.set_xticklabels(['', 'MAE', '', '', 'MSE'])
plt.xlabel('Види помилок')
plt.ylabel('Розмір помилки (логарифмічна шкала)')
plt.title('Порівняння середніх помилок моделей')
plt.yscale('log')
plt.legend()
plt.show()
fig = plt.figure(figsize=(16, 8))
plt.hist([abs_error, abs_error_lin_reg, abs_error_tree], bins=50,
        label=['Нейромережева модель', 'Модель лінійної регресії', 'Модель
дерева рішень'])
plt.xlabel('Розмір помилки')
plt.ylabel('Кількість спостережень (логарифмічна шкала)')

```

```
plt.title('Розмір абсолютної помилки моделей')
plt.yscale('log')
plt.legend()
plt.show()
fig = plt.figure(figsize=(16, 8))
plt.hist([sqr_error, sqr_error_lin_reg, sqr_error_tree], bins=60,
         label=['Нейромережева модель', 'Модель лінійної регресії', 'Модель
дерева рішень'])
plt.xlabel('Розмір помилки')
plt.ylabel('Кількість спостережень (логарифмічна шкала)')
plt.title('Розмір квадратичної помилки моделей')
plt.yscale('log')
plt.legend()
plt.show()
from pandas._libs.tslibs import timestamps
import seaborn
models = (['Нейромережева модель']*len(abs_error) +
         ['Модель лінійної регресії']*len(abs_error) +
         ['Модель дерева рішень']*len(abs_error))
errors = list(abs_error) + list(abs_error_lin_reg) + list(abs_error_tree)
data = pd.DataFrame({'Розмір помилки': errors, 'Моделі': models})
seaborn.set(style='whitegrid')
plt.figure(figsize=(12, 8))
seaborn.violinplot(x='Моделі', y='Розмір помилки', data=data)
plt.ylabel('Розмір помилки')
plt.title('Розподілення абсолютної помилки моделей')
```

ДОДАТОК Б – SOFTWARE ARCHITECTURE DOCUMENT**Kyiv national university name of Taras Shevchenko****Master's course work Software Architecture Document
(SAD)****CONTENT OWNER: Kovrina Alona****DOCUMENT NUMBER:**

- 1
- ●
- ●
- ●
- ●
- ●

RELEASE/REVISION:

- 0.1
-
-
-
-
-

RELEASE/REVISION DATE:

- 23.04.2022

Table of Contents

1	Documentation Roadmap	3
1.1	Document Management and Configuration Control Information	3
1.2	Purpose and Scope of the SAD	4
1.3	Viewpoint Definitions	5
1.3.1	Master program student Viewpoint Definition	6
1.3.1.1	Abstract	7
1.3.1.2	Stakeholders and Their Concerns Addressed	7
1.3.1.3	Elements, Relations, Properties, and Constraints	7
1.3.1.4	Language(s) to Model/Represent Conforming Views	7
2	Architecture Background	8
2.1	Problem Background	8
2.1.1	System Overview	8
2.1.2	Goals and Context	8
2.1.3	Significant Driving Requirements	8
2.2	Solution Background	8
2.2.1	Architectural Approaches	9
2.2.2	Analysis Results	9
2.2.3	Requirements Coverage	9
3	Referenced Materials	10
4	Directory	11

4.1	Index	11
4.2	Glossary	11
4.3	Acronym List	12
5	Figures & Tables	13
	List of Figures	

Figure 1:	Dataflow diagram	13
-----------	------------------	----

List of Tables

Table 1 : :	Stakeholders and Relevant Viewpoints	6	
Table 2 : :	Logistic Regression v/s Decision Tree Classification	1	3

1. Documentation Roadmap

The Documentation Roadmap should be the first place a new reader of the SAD begins. But for new and returning readers, it is intended to describe how the SAD is organized so that a reader with specific interests who does not wish to read the SAD cover-to-cover can find desired information quickly and directly.

Sub-sections of Section 1 include the following.

- Section 1.1 (“Document Management and Configuration Control Information”) explains revision history. This tells you if you’re looking at the correct version of the SAD.
- Section 1.2 (“Purpose and Scope of the SAD”) explains the purpose and scope of the SAD, and indicates what information is and is not included. This tells you if the information you’re seeking is likely to be in this document.
- Section 1.3 (“How the SAD Is Organized”) explains the information that is found in each section of the SAD. This tells you what section(s) in this SAD are most likely to contain the information you seek.
- Section 1.4 (“Stakeholder Representation”) explains the stakeholders for which the SAD has been particularly aimed. This tells you how you might use the SAD to do your job.
- Section 1.5 (“Viewpoint Definitions”) explains the *viewpoints* (as defined by IEEE Standard 1471-2000) used in this SAD. For each viewpoint defined in Section 1.5, there is a corresponding view defined in Section 3 (“Views”). This tells you how the architectural information has been partitioned, and what views are most likely to contain the information you seek.
- Section 1.6 (“How a View is Documented”) explains the standard organization used to document architectural views in this SAD. This tells you what section within a view you should read in order to find the information you seek.

1.1. Document Management and Configuration Control Information

- Revision Number: 0.1
- Revision Release Date: 19.04.2022
- Purpose of Revision: Initial project release
- Scope of Revision: Whole project

1.2. Purpose and Scope of the SAD

This SAD specifies the software architecture. All information regarding the software architecture may be found in this document, although much information is incorporated by reference to other documents.

What is software architecture? The software architecture for a system¹ is the structure or structures of that system, which comprise software elements, the externally-visible properties of those elements, and the relationships among them [Bass 2003]. "Externally visible" properties refers to those assumptions other elements can make of an element, such as its provided services, performance characteristics, fault

¹ Here, a system may refer to a system of systems.

handling, shared resource usage, and so on. This definition provides the basic litmus test for what information is included in this SAD, and what information is relegated to downstream documentation.

Elements and relationships. The software architecture first and foremost embodies information about how the elements relate to each other. This means that architecture specifically omits certain information about elements that does not pertain to their interaction. Thus, a software architecture is an *abstraction* of a system that suppresses details of elements that do not affect how they use, are used by, relate to, or interact with other elements. Elements interact with each other by means of interfaces that partition details about an element into public and private parts. Software architecture is concerned with the public side of this division, and that will be documented in this SAD accordingly. On the other hand, private details of elements—details having to do solely with internal implementation—are not architectural and will not be documented in a SAD.

Multiple structures. The definition of software architecture makes it clear that systems can and do comprise more than one structure and that no one structure holds the irrefutable claim to being the architecture. The neurologist, the orthopedist, the hematologist, and the dermatologist all take a different perspective on the structure of a human body. Ophthalmologists, cardiologists, and podiatrists concentrate on subsystems. And the kinesiologist and psychiatrist are concerned with different aspects of the entire arrangement's behavior. Although these perspectives are pictured differently and have very different properties, all are inherently related; together they describe the architecture of the human body. So it is with software. Modern systems are more than complex enough to make it difficult to grasp them all at once. Instead, we restrict our attention at any one moment to one (or a small number) of the software system's structures. To communicate meaningfully about an architecture, we must make clear which structure or structures we are discussing at the moment—which *view* we are taking of the architecture.

Thus, this SAD follows the principle that documenting a software architecture is a

matter of documenting the relevant views and then documenting information that applies to more than one view.

For example, all non-trivial software systems are partitioned into implementation units; these units are given specific responsibilities, and are the basis of work assignments for programming teams. This kind of element will comprise programs and data that software in other implementation units can call or access, and programs and data that are private. In large projects, the elements will almost certainly be subdivided for assignment to sub-teams. This is one kind of structure often used to describe a system. It is a very static structure, in that it focuses on the way the system's functionality is divided up and assigned to implementation teams.

Other structures are much more focused on the way the elements interact with each other at runtime to carry out the system's function. Suppose the system is to be built as a set of parallel processes. The set of processes that will exist at runtime, the programs in the various implementation units described previously that are strung together sequentially to form each process, and the synchronization relations among the processes form another kind of structure often used to describe a system.

None of these structures alone is *the* architecture, although they all convey architectural information. The architecture consists of these structures as well as many others. This example shows that since architecture can comprise more than one kind of structure, there is more than one kind of element (e.g., implementation unit and processes), more than one kind of interaction among elements (e.g., subdivision and synchronization), and even more than one context (e.g., development time versus runtime). By intention, the definition does not specify what the architectural elements and relationships are. Is a software element an object? A process? A library? A database? A commercial product? It can be any of these things and more.

These structures will be represented in the views of the software architecture that are provided in Section 3.

Behavior. Although software architecture tends to focus on structural information, *behavior of each element is part of the software architecture* insofar as that behavior can be observed or discerned from the point of view of another element. This behavior is what allows elements to interact with each other, which is clearly part of the software architecture and will be documented in the SAD as such. Behavior is documented in the element catalog of each view.

1.3. Viewpoint Definitions

The SAD employs a stakeholder-focused, multiple view approach to architecture documentation, as required by ANSI/IEEE 1471-2000, the recommended best practice for documenting the architecture of software-intensive systems [IEEE 1471].

As described in Section 1.2, a software architecture comprises more than one software structure, each of which provides an engineering handle on different system qualities. A *view* is the specification of one or more of these structures, and documenting a software architecture, then, is a matter of documenting the relevant views and then documenting information that applies to more than one view [Clements 2002].

ANSI/IEEE 1471-2000 provides guidance for choosing the best set of views to document, by bringing stakeholder interests to bear. It prescribes defining a set of viewpoints to satisfy the stakeholder community. A viewpoint identifies the set of concerns to be addressed, and identifies the modeling techniques, evaluation techniques, consistency checking techniques, etc., used by any conforming view. A view, then, is a viewpoint applied to a system. It is a representation of a set of software elements, their properties, and the relationships among them that conform to a defining viewpoint. Together, the chosen set of views show the entire architecture and all of its relevant properties. A SAD contains the viewpoints, relevant views, and information that applies to more than one view to give a holistic description of the system.

The remainder of Section 1.5 defines the viewpoints used in this SAD. The following

table summarizes the stakeholders in this project and the viewpoints that have been included to address their concerns.

Table 1: Stakeholders and Relevant Viewpoints

Stakeholder	Viewpoint(s) that apply to that class of stakeholder's concerns
Kovrina Alona	Masters program student

1.3.1. Masters program student Viewpoint Definition

1.3.1.1. Abstract

This viewpoint describes the view of a student in masters program who's grade is dependant on the result of the project

1.3.1.2. Stakeholders and Their Concerns Addressed

Main concern of a student is to get their project up to university's standard and to get best possible grade

1.3.1.3. Elements, Relations, Properties, and Constraints

The software consists of 1 main script that incorporates multiple modules for resume scoring and analysis, numpy and pyplot for visual representation, and fpdf for report creation

1.3.1.4. Language(s) to Model/Represent Conforming Views

Software was made on python using C++ external modules

2. Architecture Background

2.1. Problem Background

The project has text processing and attempt at subjective scoring which can prove problematic for machine learning

2.1.1. System Overview

This software takes a structured text resume and scores it based on input parameters

2.1.2. Goals and Context

The goal is to create a software prototype which can be used for resume scoring, taking the workload of from the HR

2.1.3. Significant Driving Requirements

- Implementation of multiple different modules
- Single script does all (AIO)
- Data aggregation

2.2. Solution Background

2.2.1. Architectural Approaches

The AIO architecture was chosen due to relatively small amount of data that needs to be processed and the analytical nature of the project

2.2.2. Analysis Results

After development in testing this architecture proved to be sufficient for the set task

2.2.3. Requirements Coverage

- Python interpreter
- C++ development kit
- PDF reader

3. Referenced Materials

Barbacci 2003	<p>Barbacci, M.; Ellison, R.; Lattanze, A.; Stafford, J.; Weinstock, C.; & Wood, W. <i>Quality Attribute Workshops (QAWs)</i>, Third Edition (CMU/SEI-2003-TR-016). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003.</p> <p><http://www.sei.cmu.edu/publications/documents/03.reports/03tr016.html>.</p>
Bass 2003	<p>Bass, Clements, Kazman, <i>Software Architecture in Practice</i>, second edition, Addison Wesley Longman, 2003.</p>
Clements 2001	<p>Clements, Kazman, Klein, <i>Evaluating Software Architectures: Methods and Case Studies</i>, Addison Wesley Longman, 2001.</p>
Clements 2002	<p>Clements, Bachmann, Bass, Garlan, Ivers, Little, Nord, Stafford, <i>Documenting Software Architectures: Views and Beyond</i>, Addison Wesley Longman, 2002.</p>
IEEE 1471	<p>ANSI/IEEE-1471-2000, <i>IEEE Recommended Practice for Architectural Description of Software-Intensive Systems</i>, 21 September 2000.</p>

<p>Signal processing methods for music transcription, chapter 5. Springer Science & Business Media</p>	<p>Klapura, A., and Deivi, M. (Ed.). (2007).</p>
--	--

4. Directory

4.1. Index

4.2. Glossary

Term	Definition
<p>software architecture</p>	<p>The structure or structures of that system, which comprise software elements, the externally visible properties of those elements, and the relationships among them [Bass 2003]. "Externally visible" properties refer to those assumptions other elements can make of an element, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on.</p>

view	A representation of a whole system from the perspective of a related set of concerns [IEEE 1471]. A representation of a particular type of software architectural elements that occur in a system, their properties, and the relations among them. A view conforms to a defining viewpoint.
view packet	The smallest package of architectural documentation that could usefully be given to a stakeholder. The documentation of a view is composed of one or more view packets.
viewpoint	A specification of the conventions for constructing and using a view; a pattern or template from which to develop individual views by establishing the purposes and audience for a view, and the techniques for its creation and analysis [IEEE 1471]. Identifies the set of concerns to be addressed, and identifies the modeling techniques, evaluation techniques, consistency checking techniques, etc., used by any conforming view.

4.3. Acronym List

API	Application Programming Interface; Application Program Interface; Application Programmer Interface
ATAM	Architecture Tradeoff Analysis Method
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CORBA	Common object request broker architecture

COTS	Commercial-Off-The-Shelf
EPIC	Evolutionary Process for Integrating COTS-Based Systems
IEEE	Institute of Electrical and Electronics Engineers
KPA	Key Process Area
OO	Object Oriented
ORB	Object Request Broker
OS	Operating System
QAW	Quality Attribute Workshop
RUP	Rational Unified Process
SAD	Software Architecture Document
SDE	Software Development Environment
SEE	Software Engineering Environment
SEI	Software Engineering Institute Systems Engineering & Integration Software End Item
SEPG	Software Engineering Process Group
SLOC	Source Lines of Code
SW-CMM	Capability Maturity Model for Software
CMMI-SW	Capability Maturity Model Integrated - includes Software Engineering
UML	Unified Modeling Language

5. Sample Figures & Tables

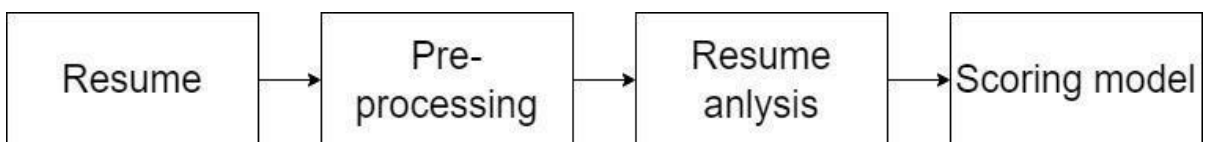


Figure 1: Dataflow diagram

Table 2: Logistic Regression v/s Decision Tree Classification

Criteria	Logistic Regression	Decision Tree Classification
Interpretability	Less interpretable	More interpretable
Decision Boundaries	Linear and single decision boundary	Bisects the space into smaller spaces
Ease of Decision Making	A decision threshold has to be set	Automatically handles decision making
Overfitting	Not prone to overfitting	Prone to overfitting

Robustness to noise	Robust to noise	Majorly affected by noise
Scalability	Requires a large enough training set	Can be trained on a small training set