

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Розробка платформи для пошуку роботи з використанням
ШІ»**

Виконав студент групи ІПЗм-24-1
ЛАКИЗЮК Ренат Валентинович

Керівник роботи:
ЄФРЕМОВ Юрій Миколайович

Рецензент:
МОРОЗОВ Андрій Васильович

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Завідувач кафедри інженерії
програмного забезпечення

Тетяна ВАКАЛЮК

«18» вересня 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЛАКИЗІЮК Ренат Валентинович**

Керівник роботи: **ЄФРЕМОВ Юрій Миколайович**

Тема роботи: **«Розробка платформи для пошуку роботи з використанням ШІ»**, затверджена Наказом закладу вищої освіти від **«18» вересня 2025 р., №491/с**

Вихідні дані для роботи: **аналіз та оцінка відповідності резюме та вакансій у платформах для пошуку роботи та впровадження методів машинного навчання у процес оцінки відповідності; способи автоматизації процесу оцінки відповідності між резюме та вакансіями.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Єфремов Ю. М.	22.09.2025р	22.09.2025р
2	Єфремов Ю. М.	22.09.2025р	22.09.2025р
3	Єфремов Ю. М.	22.09.2025р	22.09.2025р

РЕФЕРАТ

Кваліфікаційна робота освітнього ступеня “магістр” присвячена автоматизації процесу оцінки відповідності між резюме і вакансій у платформах для пошуку роботи і впровадження методів машинного навчання у процес оцінки відповідності. Пояснювальна записка складається з вступу, трьох розділів та списку використаної літератури. Пояснювальна записка до кваліфікаційної магістерської роботи містить 57 сторінки, 35 ілюстрацій, 2 таблиці, 4 додатки та 33 посилань на зовнішні джерела.

Метою роботи є розробка та дослідження системи оцінки відповідності резюме та вакансій на основі машинного навчання з порівняльним аналізом традиційних і сучасних підходів для допомоги рекрутерам у підборі кандидатів, пошуку роботи кандидатам.

В роботі процес створення системи оцінки відповідності, яка аналізує текстовий контент резюме і вакансії, і надає оцінку відповідності між ними. Досліджено способи навчання моделей машинного навчання традиційний підхід на основі SVM (Support Vector Machine) з TF-IDF векторизацією і сучасний підхід на основі BERT (Bidirectional Encoder Representations from Transformers) для семантичного аналізу тексту. Проведено порівняльний аналіз методів за метриками точності (Accuracy, Precision, Recall, F1-Score). Також реалізовано покращення у вигляді системи штрафів за невідповідність ролей, нелінійної нормалізації.

КЛЮЧОВІ СЛОВА: МАШИННЕ НАВЧАННЯ, NLP, РЕЗЮМЕ-ВАКАНСІЯ MATCHING, SVM, BERT, СЕМАНТИЧНА СХОЖІСТЬ, TF-IDF, КОСИНУСНА СХОЖІСТЬ.

ABSTRACT

The master's thesis is devoted to the automation of the process of matching resumes and job vacancies on job search platforms and the implementation of machine learning methods in the matching process. The explanatory note consists of an introduction, three chapters, and a list of references. The explanatory note to the master's thesis contains 57 pages, 35 illustrations, 2 tables, 4 appendices, and 33 references to external sources.

The purpose of the work is to develop and study a system for assessing the compatibility of resumes and job vacancies based on machine learning with a comparative analysis of traditional and modern approaches to help recruiters select candidates and candidates find jobs.

The work describes the process of creating a compatibility assessment system that analyzes the text content of resumes and job vacancies and provides an assessment of their compatibility. We investigated methods for training machine learning models: a traditional approach based on SVM (Support Vector Machine) with TF-IDF vectorization and a modern approach based on BERT (Bidirectional Encoder Representations from Transformers) for semantic text analysis. A comparative analysis of the methods was carried out using accuracy metrics (Accuracy, Precision, Recall, F1-Score). Improvements were also implemented in the form of a penalty system for role mismatches and nonlinear normalization.

KEYWORDS: MACHINE LEARNING, NLP, RESUME-JOB MATCHING, SVM, BERT, SEMANTIC SIMILARITY, TF-IDF, COSINE SIMILARITY.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Аналіз проблем оцінки відповідності резюме та вакансій	9
1.2 Аналіз існуючих підходів до оцінки відповідності резюме та вакансій	11
1.3 Постановка задачі дослідження	17
1.4 Вибір засобів реалізації	18
Висновки до розділу 1	20
РОЗДІЛ 2. РОЗРОБКА ТА ПРОЕКТУВАННЯ СИСТЕМИ ОЦІНКИ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ	21
2.1 Розробка системи оцінки відповідності за допомогою машинного навчання	21
2.2 Підготовка даних та навчання моделей	32
2.3 Проектування додатку	34
Висновки до розділу 2	37
РОЗДІЛ 3. ОПИС ПРОГРАМНОГО ПРОДУКТУ	38
3.1 Налаштування до роботи з програмою	38
3.2 Налаштування параметрів оцінки відповідності	47
3.3 Відображення результатів	49
Висновки до розділу 3	52
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТКИ	58

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

BERT – Bidirectional Encoder Representations from Transformers
(Двонаправлені представлення кодувань трансформерів)

ML – Machine Learning (Машинне навчання)

NLP – Natural Language Processing (Обробка природної мови)

SVM – Support Vector Machine (Машина опорних векторів)

TF-IDF – Term Frequency-Inverse Document Frequency (Частота терміна -
обернена частота документа)

ВСТУП

У сучасному світі ринок праці дуже швидко і динамічно змінюється, відповідно пошук роботи чи кандидата теж змінюється і стає дедалі складнішим, а обсяг обробки даних теж збільшується, традиційна обробка і аналіз даних займає не мало сил, часу як у кандидатів, так і у роботодавців. Пошук ускладнюється і для кандидатів, і для роботодавців, а швидкість знайти нову роботу або якісного кандидата іноді буває дуже критичним. Але ми зараз живемо в світі штучного інтелекту (ШІ), який дає нові, чудові можливості для великої кількості сфер. В даному випадку штучний інтелект може спростити, автоматизувати, покращити процес оцінки відповідності між резюме кандидата і вакансією, тобто, ця система автоматичної оцінки відповідності спростить рутинну роботу, наприклад, для рекрутерів, які зможуть зосередитись на резюме кандидатів з більшими оцінками, а кандидати зможуть отримати швидко оцінку своєї відповідності вакансії на яку вони відгукуються.

Актуальність є створення автоматичної оцінки відповідності резюме та вакансії. Це спричинено швидкої обробки великої кількості відгуків на вакансії, а традиційні методи вимагають ручного аналізу резюме, а цей процес може займати багато часу і ресурсів. Таке рішення ШІ може автоматизувати процес оцінки відповідності, яке знижуватиме витрати на підбір персоналу, підвищуючи точність оцінки і забезпечуватиме об'єктивний аналіз.

Мета роботи є розробка і реалізації системи автоматичної оцінки відповідності резюме і вакансії, яка забезпечить оцінку відповідності резюме кандидатів і вакансій на основі аналізу текстових даних, машинного навчання та інших алгоритмів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- дослідити існуючі підходи до автоматичної оцінки відповідності резюме і вакансій;

- дослідити методи обробки природної мови та машинного навчання для аналізу текстової відповідності;
- виконати порівняльний аналіз традиційних (SVM з TF-IDF) та сучасних (BERT) методів машинного навчання;
- створити систему автоматичної оцінки відповідності резюме і вакансії з використанням методів машинного навчання;

Об'єктом дослідження є процес оцінки відповідності резюме і вакансії за допомогою штучного інтелекту.

Предметом дослідження є алгоритми і методи штучного інтелекту для автоматизації процесу оцінки відповідності резюме і вакансії, і їхню реалізацію.

Наукова новизна роботи полягає у використанні методів машинного навчання і обробки природної мови для реалізації оцінки відповідності резюме і вакансії. Особливості такого підходу є порівняльний аналіз традиційних і сучасних методів машинного навчання (SVM з TF-IDF векторизацією і BERT із семантичними ембедінгами) на одних і тих самих даних, що дозволяє об'єктивно оцінити переваги і недоліки кожного із підходів.

Апробація. За темою кваліфікаційної магістерської роботи було опубліковано тези у конференції [1].

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз проблем оцінки відповідності резюме та вакансій

У сучасному світі оцінка відповідності резюме кандидата і вакансії є важливим фактором в наш час. Якщо взяти традиційні вебзастосунки, платформи для пошуку роботи вони працюють в більшості на основі статичних фільтрів, що іноді не завжди дозволяє користувачу налаштувати фільтр під себе, а також роботодавці, рекрутери витрачаються досить велику кількість часу, ресурсів на аналіз і обробку великої кількості резюме кандидатів, які відгукнулись на вакансію, а якщо використовувати традиційні методи оцінки то вони в більшості базуються саме на пошук ключових слів, що може призвести до неточності так як не враховуються синоніми і зміст тексту.

З вище наведеного тексту можна зробити висновок і визначити наступні існуючі проблеми:

- оцінка: більша частина платформ, вебзастосунків оцінюють відповідність на основі ключових слів і ручному аналізу, що може не враховувати такі аспекти як: досвід, навички, зміст;
- висока навантаженість: аналіз і обробка великої кількості відгуків, резюме потребує ресурсів і ручний аналіз, і обробка можуть бути не точними і суб'єктивними;
- час: щоб оцінити відгук з використанням традиційних методів потрібно витрати не мало часу.

Тому в сучасних умовах більшість вебзастосунків, платформ тощо для пошуку роботи необхідні застосування новітніх технологій автоматизації, методи машинного навчання для автоматизації [2], щоб покращити функціонал в тому числі і оцінка відповідності, яка дозволить рекрутерам, роботодавцям фокусуватись на кандидатів з вищими оцінками, а кандидати відразу можуть побачити свої шанси бути запрошеним на співбесіду. З боку аналізу потенційних переваг інтелектуальних алгоритмів, були певні

порівняння традиційних методів оцінки і з використанням автоматизованих методів, результат можемо побачити в таблиці (табл. 1.1).

Таблиця 1.1

Порівняння традиційного і автоматизованого методу в оцінці відповідності резюме і вакансії

Критерії	Традиційні методи	Автоматизовані методи (з AI)
Оцінка відповідності	Роботодавці, рекрутери читають, аналізують резюме вручну при цьому порівнюючи з вимогами вакансії. Через можливу велику кількість даних не просто буде швидко оцінити відповідність.	Використовуючи штучний інтелект, який аналізуватиме резюме, навички, досвід і інші дані, автоматично оцінює відповідність резюме кандидата і вакансії.
Обробка резюме і швидкість	При ручній обробці і аналізу резюме кандидата процес найму кандидата сповільнюється, а обробка і аналіз одного резюме може займати від декількох хвилин до декількох десятків хвилин.	Автоматизовані системи можуть швидко оцінювати, а також робити це паралельно що збільшить швидкість вдвічі, що дозволить обробляти велику кількість резюме за короткий час.
Об'єктивність	Суб'єктивна оцінка може залежати від упереджень людини що перевіряє.	Об'єктивна оцінка на основі алгоритмів машинного навчання, що спричиняє зменшенню впливу упереджень.

1.2 Аналіз існуючих підходів до оцінки відповідності резюме та вакансій

Головною метою даної автоматизації оцінки відповідності резюме і вакансій є те щоб покращити ефективність підбору персоналу і зменшити навантаження при обробці великої кількості відгуків [3], а кандидатам відобразити їх шанси на розгляд вакансії на яку подались. Також дана автоматизація дозволяє швидко обробити досить великі обсяги даних при цьому зменшуючи людського фактори і упереджень, що забезпечить більш об'єктивну оцінку відповідності. На даний час існує декілька платформ, вебзастосунків на території України, де можна виконувати пошук роботи і пошук, підбір кандидатів.

Одна із найпопулярніших і доволі зручних платформ є Work.ua, яка можна сказати є також однією з найбільших платформ для пошуку роботи в Україні. Ця платформа є простою у використанні, можна створювати або завантажувати резюме у вигляді документів різних форматів, має широкий функціонал фільтрів для пошуку вакансій (сфера діяльності, місце роботи, діапазон зарплати, категорія, тип зайнятості та інше). Також дана платформа має велику кількість вакансій різних категорій і спеціальностей, від малого до великого бізнесу, включаючи спеціальності таких сфер як: робітничі, ІТ-сектору, управлінські, військові, медичні, освітні та інших сфер. Містить також історія пошуків, сповіщення про нові вакансії, які користувач шукав в минулому. Щодо роботодавців то платформа їм пропонує базу резюме кандидатів, публікацію вакансій, повідомлення кандидатам. Є веб-версія , а також мобільний застосунок.

Одним з таких головних недоліків можна назвати оцінку відповідності, а саме що вона дуже слабо реалізована. Проблема в тому, що кандидат може максимум отримати тег що він в якомусь топі кандидатів (рис. 1.1) і це дуже рідко спрацьовує, тобто, якщо кандидат навіть відгукнеться на вакансію оцінка відповідності не спрацює і ні кандидат, ні роботодавець не отримують числового показника оцінки, який би дозволив роботодавцям зосередитись на

кандидатів з вищими оцінками, а кандидатам дізнатись їх можливості пройти до етапу співбесіди або отримати зворотній зв'язок. В даній платформі є можливість кандидатам отримувати рекомендовані вакансії, але це працює через історію пошуків, тобто, в список рекомендовані вакансії будуть потрапляти вакансії з назвами схожими на минулі пошукові запити кандидата, іншими словами тут просто підписка на нові вакансії з історії пошуків.

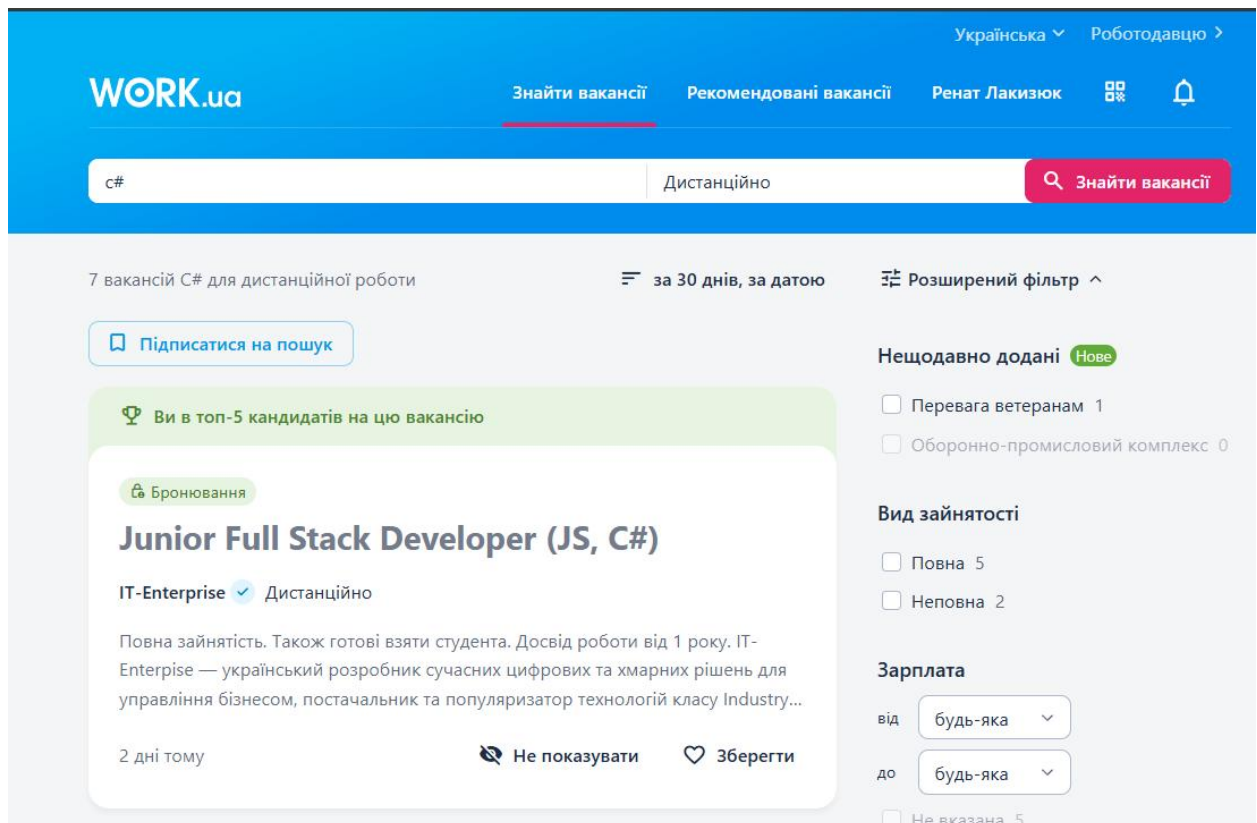


Рис. 1.1. Work.ua з оцінкою на вакансією

Іншою подібною вебзастосунком, додатком є Robota.ua, яка можна сказати з впевненістю є найдавнішою платформою для пошуку роботи в Україні і також звісно входить до числа найбільших платформ з пошуку роботи. Дана платформа пропонує кандидатам детальне створення резюме або іншими словами можна також сказати профіль, тобто, заповнення про себе детальну інформації (особиста інформація, бажана посада, досвід роботи, навички, освіта, курси і тощо), є можливість завантаження резюме файлу, конструктор резюме, спілкування з роботодавцями через вбудований чат, є функціонал обраних вакансій, налаштування розсилок, гнучкий фільтр. Є

великий функціонал і для роботодавців публікація вакансій, доступ до бази резюме, спілкування з кандидатами через вбудований чат, фільтри. До того ж тут теж велика кількість вакансій з різних сфер і спеціальностей і теж дуже багато кандидатів, які в пошуках нової роботи. Також наявний мобільний застосунок, який спрощує користування на мобільних пристроях.

Тобто рішення має повний набір. Але тут ситуація подібна до Work.ua з оцінкою відповідності після відгуку не буде виконуватись оцінка відповідності (рис. 1.2), а рекомендовані вакансії побудовані теж на основі історії пошуків, сюди потрапляють нові вакансії в яких заголовок подібний до історії пошуків кандидата або бажана посада в профілі. Автоматизація оцінка відповідності яка б виконувала аналіз резюме і вакансію і в результаті отримували числовий показник немає і це ускладнює обробку відгуків.

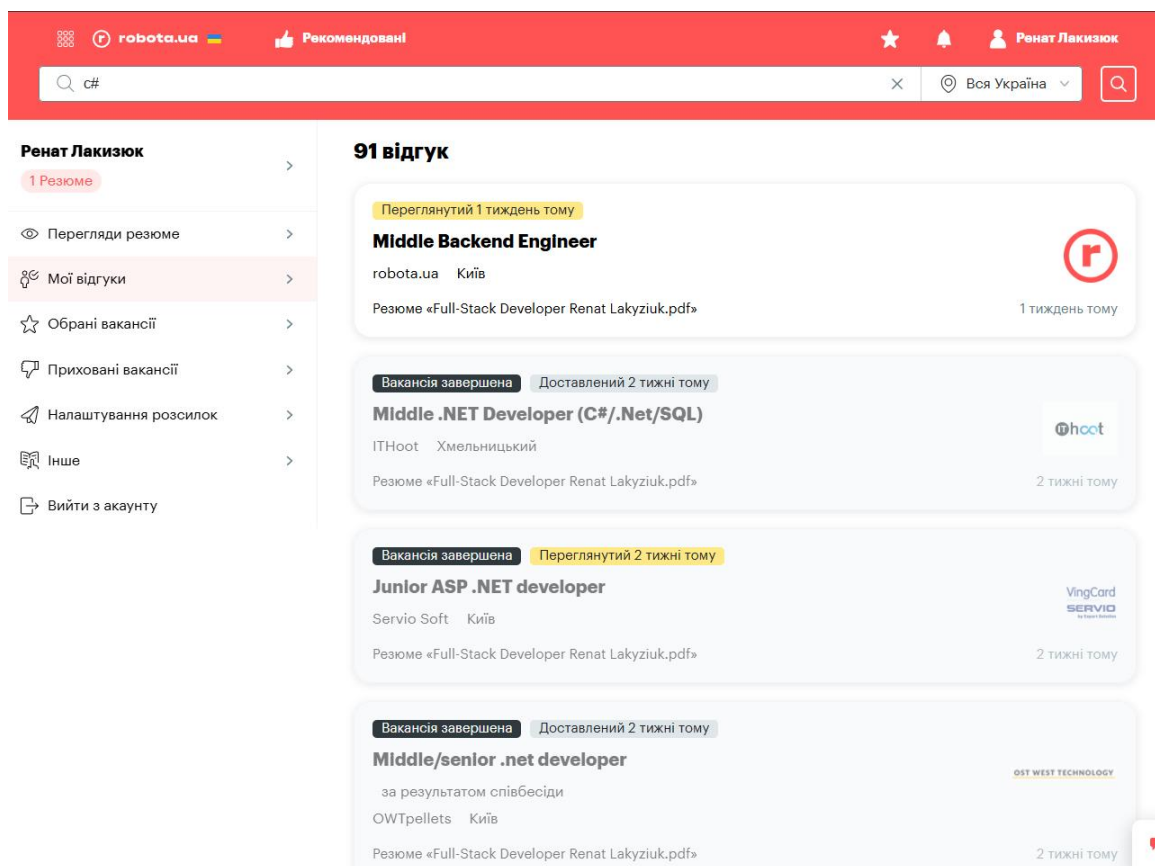


Рис. 1.2. Robota.ua

Третім аналогом є Djinni, ця платформа працює на основі анонімного пошуку, що забезпечує анонімність для тих хто шукає роботу. Це насправді унікальна особливість, яка дуже сильно відрізняє дану платформу від усіх

інших, а також ця особливість дозволяє кандидатам виконувати пошук вакансій не турбуючись про те що їх роботодавець про це дізнається, що є досить актуально для ІТ-спеціалістів, яких велика частина постійно в пошуках нових викликів, можливостей. Ситуація з базою вакансій тут трішки інша тут переважна кількість вакансій стосується ІТ-сфери з інших сфер вакансій тут досить мало і сама платформа фокусується на ІТ-сферу тому не дивно чому тут така ситуація. Тут є система рекомендацій, яка працює на основі зазначених навичок, досвіду, знання мов це все гнучко налаштовується кандидатами в створенні і редагуванні профілю, вбудований чат, детальна аналітика ринку, можливість приховувати профіль від зазначених компаній, бути не доступним для вакансій для певних сфер роботи, дуже гнучкі фільтри (технології, роки досвіду, знання іноземних мов, рівень спеціалізації, типи компаній, тип зайнятості тощо). Все це робить дану платформу дуже популярною серед працівників ІТ-сфери (рис. 1.3), (рис. 1.4).

Щодо недоліків тут відсутня оцінка відповідності, рекомендаційна система працює на основі порівняння зазначених навичок та іншої інформації в профілі користувача і у вакансії. Якщо відгукнутись на вакансію ми не отримаємо оцінку відповідності, але тут працює функціонал який заблоковує можливість кандидату відгукнутись на вакансію, якщо його профіль не відповідає зазначеним вимогам у вакансії.

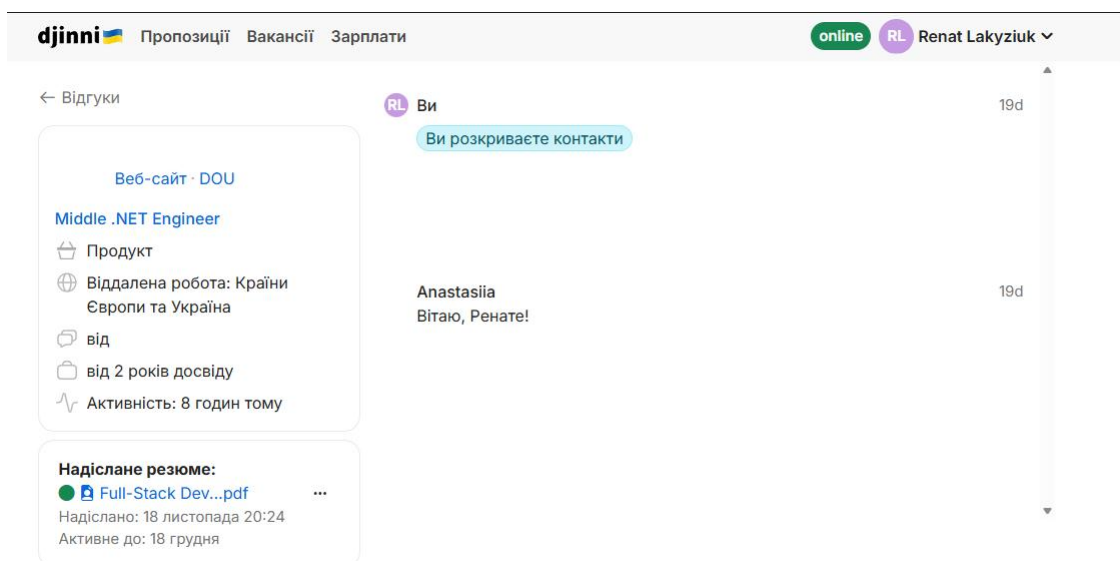


Рис. 1.3. Відгук на вакансію на платформі Djinni.

Пропозиції

Вакансії

Зарплати

online

RL Renat Lakyziuk

Вакансії за моїм профілем 5

Для мене

Всі

Let's Recruit · 171 перегляд · 32 відгуки · 5д

C/C++/C# Developer

Тільки віддалено · Весь світ · Продукт · 2 роки досвіду

MilTech

Загальна інформація: Компанія: NDA. Приватна організація. Існує 2 роки, у складі має декілька великих проєктів, що вже прискорюють нашу перемогу. Нині відкрита вакансія на новий проєкт, де зараз формується команда. Проєкт: NDA, MilTech, Cyber... Більше

Зберегти

Сховати

BVP Software · 913 переглядів · 309 відгуків · 10д

ReactJs Developer

Part-time · Тільки віддалено · Весь світ · 2 роки досвіду

Привіт, Шукаємо ReactJs Developer-а для невеликого one-time проєкту (можуть потім бути ще проєкти). Основне це знати React та вміти працювати з різними API. Сам проєкт це розробка chrome extension (попередній досвід не обов'язковий) Technical... Більше

Зберегти

Сховати

DIZAIR · 878 переглядів · 320 відгуків · 20д

Front-end developer до \$2000

Весь світ · Продукт · 2 роки досвіду · A2 - Елементарний

60 000 – 75 000 грн, В залежності від рівня володіння стеком. Dizair Software IT; 10–50 співробітників Дистанційна робота Повна зайнятість. Досвід роботи від 2 років. Вища освіта. Англійська — середній Middle Frontend Developer (React + Next.js) Ми... Більше

Зберегти

Сховати

Вас найняли на Djinni?

Так

Поки ні

Full-stack Developer

від \$1800 · 2 роки досвіду · A2 - Елементарний · Україна, Житомир

Редагувати профіль

За моїм профілем

Збережені

Сховані

Відгуки та пропозиції

Підписки · На email

У вас поки що немає підписок

Створити підписку

Статистика за 30 днів

2 пропозицій отримано

Ви відкрили контакти на 2 з 2

1 відгуків надіслано

Роботодавці прочитали 1 з 1

2 переглядів профілю

409 місце в рейтингу

за кількістю пропозицій

Більше статистики

Рис. 1.4. Список вакансій в платформі Djinni.co

Якщо проаналізувати все вище зазначене то дані платформи використовують пошук по ключовим словам, метод який базується на основі точного збігу слів між резюме і вакансією [4]. Або метод який аналізує зазначену інформацію кандидатом (досвід роботи, навички тощо) і порівнює їх з вимогами вакансії.

Жодна із цих наведених платформ не використовують автоматизації які побудовані за допомогою методів машинного навчання, таких як трансформерні моделі BERT [5] або традиційних методів NLP [6] з векторизацією тексту. Вони всі просто базуються на простих правилах і збігах, що може обмежувати точність, здатність розуміння синонімів, контекст. Порівняння основних характеристик цих платформ наведено в таблиці (табл. 1.2).

15

Таблиця 1.2

Порівняння платформ для пошуку роботи і оцінки відповідності

Критерії	Work.ua	Robota.ua	Djinni.co
Оцінка відповідності	Пошук ключових слів	Ні	Автоматичні рекомендації на основі навичок
Чисельна оцінка відповідності	Ні	Ні	Ні
Швидкість оцінки	Ручний аналіз	Ручний аналіз	Ручний аналіз, але менша кількість відгуків за рахунок обмеження відгуків якщо дотримано вимог вакансії
Використання ML/NLP	Не виявлено	Не виявлено	Не виявлено

Після аналізу результатів порівняння, можна зробити такий висновок, що всі ці три платформи мають обмежені можливості автоматичної оцінки відповідності резюме і вакансії. Більша частина з них використовує базові фільтри, пошук ключових слів або прості збіги навичок. І це все ще спричиняє проблему з обробкою можливо великої кількості відгуків, резюме і інших даних. В даному випадку відсутність автоматичної оцінки відповідності у вигляді числа ускладнює обробку кандидатів і прийняття рішення. І це підтверджує, що розробка системи оцінки відповідності є актуальною з використанням методів машинного навчання [7], які мають здатність виконувати глибокий семантичний аналіз і надати об'єктивну оцінку.

1.3 Постановка задачі дослідження

Після виконання аналізу всіх наведених переваг і недоліків існуючих платформ для пошуку роботи можна зробити визначення основних аспектів, що можна виправити і покращити в майбутній системі. Основні недоліки раніше згаданих платформ в тому що відсутня оцінка відповідності, яка б аналізувала контент резюме кандидата і контент вакансії [8]. Відсутність такою оцінки відповідності погіршує процес рекомендацій резюме, обробку відгуків, резюме.

При визначенні задачі є дуже важливо врахування багатьох подібних платформ обробляються тексти резюме і вакансії, які можуть повертаються з дуже різних джерел і можуть мати різну структуру, а також формат. Але такі платформи мають обмежену функціональність, яка дозволить виконати оцінку відповідності на основі чітких встановлених ключових слів та статичних фільтрів при цьому не враховуючи семантичний зміст тексту, синоніми.

Знову ж таки аналізуючи переваги і недоліки платформ, можна зробити початкове уявлення про те, як саме має функціонувати система оцінки відповідності. Має бути універсальною, тобто, працювати з різними даними, реалізований аналіз даних з використанням штучного інтелекту для спрощення автоматизації оцінки відповідності між резюме і вакансією [9]. Також має виконуватись глибокий семантичний аналіз текстів, не ігноруючи синоніми, контекст, давати об'єктивну оцінку у вигляді числа, що дозволить швидко і точно оцінювати відповідність, а в подальшому розроблені іншого функціоналу в платформі.

Знаходячись на теперішньому етапі було прийнято рішення розділити створення системи оцінки відповідності на декілька етапів. Перший етап - розробити систему, яка буде вже отримувати дані резюме і вакансії, нормалізувати, очищати текстові дані, щоб в подальшому виконати наступний аналіз. Наступний етап, другий етап це продумати організацію структури обробки даних, а також звісно налаштування параметрів оцінки

відповідності. На фінальному етапі розробити модель машинного навчання для оцінки відповідності резюме і вакансії, яка зможе визначити відповідність і давати результат у вигляді числа.

1.4 Вибір засобів реалізації

Використати можна багато і різних методів машинного навчання і обробки природної мови для оцінки відповідності резюме і вакансії, але важливим моментом є також аргументовано підійти до вибору рішення для кожної проблеми це напряму впливає роботу і результат такої оцінки відповідності. Для цієї задачі добре підійдуть наступні підходи: традиційний метод використовуючи SVM з TF-IDF для векторизації [10] або більш сучасний метод з використанням BERT для семантичного аналізу [11]. Кожний з даних методів має свої позитивні і негативні моменти, тому можна прийняти щоб реалізувати два цих підходи і зробити порівняльний аналіз для визначення кращого підходу, щоб в подальшому обрати оптимальний метод.

SVM алгоритм машинного навчання, який навчається на прикладах і визначає межу між двома класами, а саме підходить і не підходить [12]. Він працює на основі аналізу векторів і визначає на якій стороні межі знаходиться новий випадок. TF-IDF якщо розшифрувати то це Term Frequency-Inverse Document Frequency метод який перетворює текст у вектор, який також враховує частоту слова у тексті і його рідкість слова в цьому тексті [13]. Цей метод дозволяє більш ефективно працювати з текстовими даними. Якщо перейти до переваг то SVM має швидке обчислення, менші вимоги до ресурсів, простіша інтерпретація результатів, але є обмеження розуміння семантики і залежність від точних збігів слів.

BERT або Bidirectional Encoder Representations from Transformers - це нейронна мережа, яка навчена на мільйонах текстів, яка має можливість розуміти семантику мови [14]. Головною різницею між стандартними методами машинного навчання і BERT є те що BERT має зворотні зв'язки і має здатність обробляти не тільки окремі слова, а і цілі послідовності текстів,

також враховуючи з обох сторін контекст [15]. Дякуючи що дана нейронна мережа має здатність розуміти семантичний зміст тексту і контексту слів, це допомагає краще визначити відповідність між резюме кандидата і вакансією, навіть у тих випадках коли можуть використовуватись синоніми і формулювання. З цього можна зробити підсумок що це нам дозволить використати дану нейронну мережу для більш точнішого семантичного аналізу відповідності плюс враховуючи при цьому значення і зв'язки між словами.

Також є дуже важливий аспект перед аналізом даних краще виконувати передобробку тексту це дозволить покращити коректність роботи моделі машинного навчання [16]. Передобробка включає в собі привести текст до нижнього регістру, видалити з тексту спеціальні символи або їх замінити на слова, видалити стоп-слова, також виконати приведення слів до базової форми, наприклад, “developers” приводимо до базової форми “developer” і також фільтрування коротких слів. Всі ці кроки передобробки дозволять нам привести текст резюме і вакансії до стандартизованого нами вигляду це покращить якість векторизації і точність моделі так як не буде зайвих даних, шумів. Якщо не виконувати передобробку або виконувати її погано це спричинить наступні проблеми такі як погані результати тому що слова можуть бути в різних регістрах або формах і після векторизації можуть розрізнятись як різні ознаки.

Після того як виконається передобробка датасети слід поділити на навчальну і тестову вибірку. В більшості випадків розділ відбувається за принципом 80% даних для навчання, а інші частина, тобто, 20% для тестування. Навчальна вибірка даних як можна здогадатись використовується для навчання моделі тоді як тестова вибірка даних використовується щоб перевірити навчену модель і оцінити її точність, ефективність на нових, невидимих даних. Дане розділення дозволить більш об'єктивно оцінити модель і уникнути перенавчання це коли навчена модель запам'ятовує навчальні дані і не може узагальнювати на нових прикладах.

В подальшому щоб це все реалізувати і використовувати в платформі для пошуку роботи, було вирішено реалізувати окрему одиницю мікросервісу на мові Python так як ця мова має велику кількість бібліотек для штучного інтелекту, машинного навчання, обробка природної мови.

Висновки до розділу 1

В даному розділі був проведений аналіз проблематики оцінки відповідності резюме і вакансії, а також вибору технологій для реалізації такої системи з використанням штучного інтелекту.

Також були визначенні основні, загальні проблеми платформ, вебзастосунків для пошуку роботи, які пов'язані з неефективною оцінки відповідності, можливою великою навантаженістю роботодавців, рекрутерів і витратою часу на ручний аналіз. Проаналізовані існуючі підходи до оцінки відповідності традиційні методи машинного навчання (SVM з TF-IDF) і сучасні методи навчання (BERT), а також обмеження оцінки в існуючих платформах.

Основні завдання дослідження були сформовані, які в собі включають аналіз існуючих платформ, дослідження методів штучного інтелекту, створення алгоритмів оцінки відповідності, реалізації і тестування системи, а також оцінку результатів.

Базуючись на аналізі сучасних технологій було аргументовано вибір методів машинного навчання для оцінки відповідності. Було вибрано два підходи: традиційний підхід на SVM + TF-IDF і BERT для більш точного семантичного аналізу. Щоб працювати зі штучним інтелектом вибір пав на scikit-learn для реалізації SVM моделі, а transformers щоб працювати і реалізувати BERT модель, це дозволить впровадити оцінку відповідності базуючись на різних підходів машинного навчання, після реалізації провести порівняльний аналіз їх ефективності щоб в подальшому визначити більш ефективний метод.

РОЗДІЛ 2. РОЗРОБКА ТА ПРОЕКТУВАННЯ СИСТЕМИ ОЦІНКИ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

2.1 Розробка системи оцінки відповідності за допомогою машинного навчання

Для реалізації системи оцінки відповідності резюме і вакансії додаток має використовувати методи машинного навчання і обробки природної мови з мовою програмування Python. Python має чудову екосистему для роботи з текстовими даними, машинним навчанням, штучним інтелектом і це робить його досить гарним варіантом для рішення даної задачі.

Щоб реалізувати систему оцінки відповідності був зроблений вибір на наступні моделі і бібліотеки:

`scikit-learn` - модуль Python для машинного навчання, який побудований на базі `SciPy`. Даний інструмент надає досить широкий вибір алгоритмів класифікації, включаючи SVM, а також інструменти для векторизації текстів, в тому числі TF-IDF векторизатор, який дає змогу перетворити текстові дані в числові вектори щоб в подальшому обробити алгоритмами машинного навчання;

`pandas` - бібліотека для маніпуляції та аналізу даних, надає структури даних і операції для маніпулювання числовими таблицями і часовими рядками, яка також дозволяє зручно працювати з датасетами резюме і вакансії, виконуючи такі операції як: об'єднання даних, фільтрацію і підготовку даних для навчання моделей;

`numpy` - додає підтримку великих багатовимірних масивів, матриць, великою кількістю математичних функцій для операцій над масивами і матрицями. Використовується для роботи, обчислень з векторами і матрицями при роботі з TF-IDF векторами і ембедінгами;

`nltk` - бібліотека для обробки природної мови, яка має інструменти для токенізації, видалення стоп-слів, лематизації та стемінгу. Також дозволяє виконати передобробку текстів резюме і вакансії, приводячи їх до нашого стандартизованого вигляду перед тим як виконати векторизацію;

transformers - бібліотека від Hugging Face для роботи з трансформерними моделями, одна з них яка нам потрібна BERT, а також надає інструменти для використання сучасних методів семантичного аналізу.

torch - використовується для глибокого навчання, а саме використовується для роботи з BERT моделлю, надає чудовий набір інструментів для роботи з ембедінгами і нейронними мережами;

scipy - використовується для наукових обчислень, також надає інструменти для роботи з розрідженими матрицями.

Google Colab - хмарне середовище де виконується код Python з підтримкою GPU, також дозволяє ефективно обробляти велику кількість даних і навчати моделі машинного навчання.

В якості бази для оцінки відповідності обрано було два підходи: SVM традиційний метод з векторизацією за допомогою TF-IDF і сучасним методом BERT для семантичного аналізу. Класичний підхід з використанням SVM і TF-IDF до обробки текстових даних має переваги у таких аспектах швидкість навчання навіть на великій кількості даних, менша потреба в потужності ресурсів, але даний підхід має певні обмеження в розумінні багатозначності слів і залежностей від точних збігів термінів, що іноді може спричиняти до пропуску релевантних пар при використанні різних синонімів для опису подібних навичок. Щодо BERT то ця модель може демонструвати більш вищу якість аналізу, можливість фокусуватись на важливих аспектах тексту і зв'язки між словами. Недоліками BERT є більші вимоги до обчислювальних потужностей і часу на відміну від класичних методами, що буває іноді критичним коли обробляється велика кількість даних у реальному часу.

Такі підходи можуть аналізувати різні формати резюме і вакансій при цьому враховуючи семантичний контекст і розробляти більш універсальний програмний продукт, який також зможе бути корисним у різних сферах ринку праці.

Щоб реалізувати оцінку відповідності резюме і вакансії було обрано таких два підходи: класичний метод на базі SVM з TF-IDF векторизацією і сучасний метод на базі BERT для семантичного аналізу.

Далі річ піде про підхід на основі SVM з TF-IDF векторизацією. Для того щоб краще зрозуміти роботу системи оцінки відповідності використовуючи SVM з TF-IDF векторизацією для початку давайте ознайомимося з основними компонентами та етапи обробки. Давайте розглянемо (рис. 2.1) де зображена загальна структура процесу оцінки відповідності з використанням SVM.

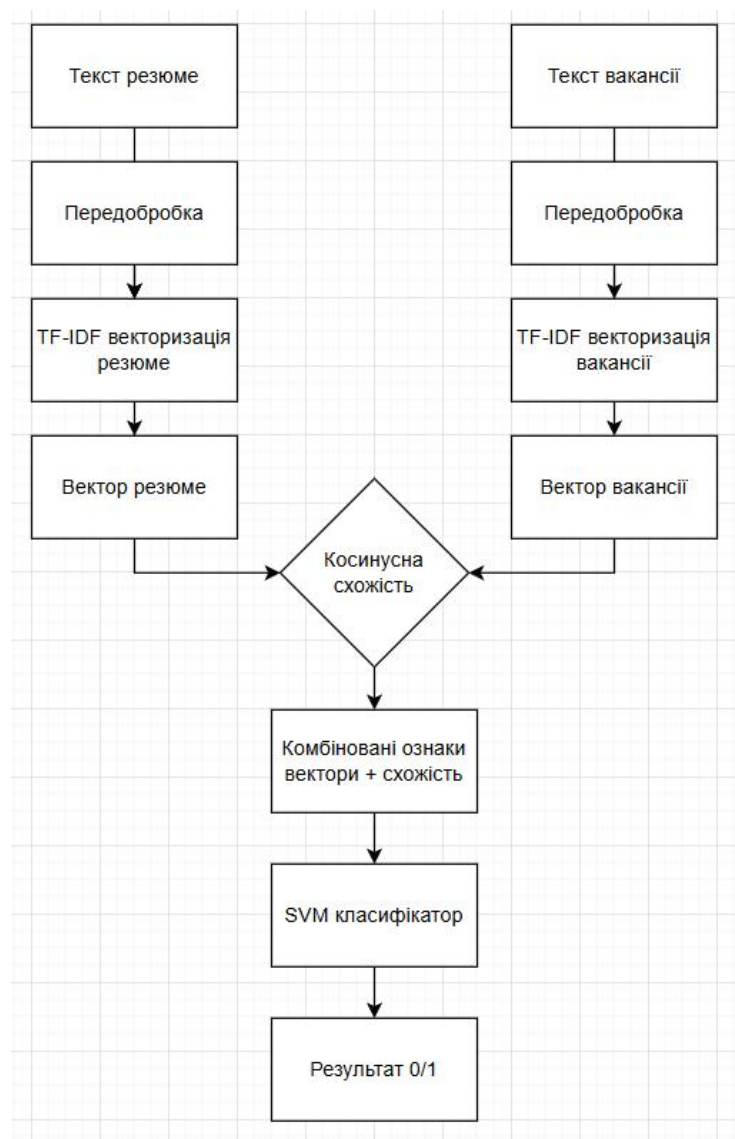


Рис. 2.1. Загальна структура процесу оцінки відповідності на основі SVM.

Після розгляду основних елементів структури, які зображені на (рис. 2.1), можна побачити що перший етап це передобробка тексту резюме і

вакансії, яка включає кілька важливих етапів щоб привести текст до стандартизованого вигляду. На (рис. 2.2) зображений схематичний процес передобробки тексту.

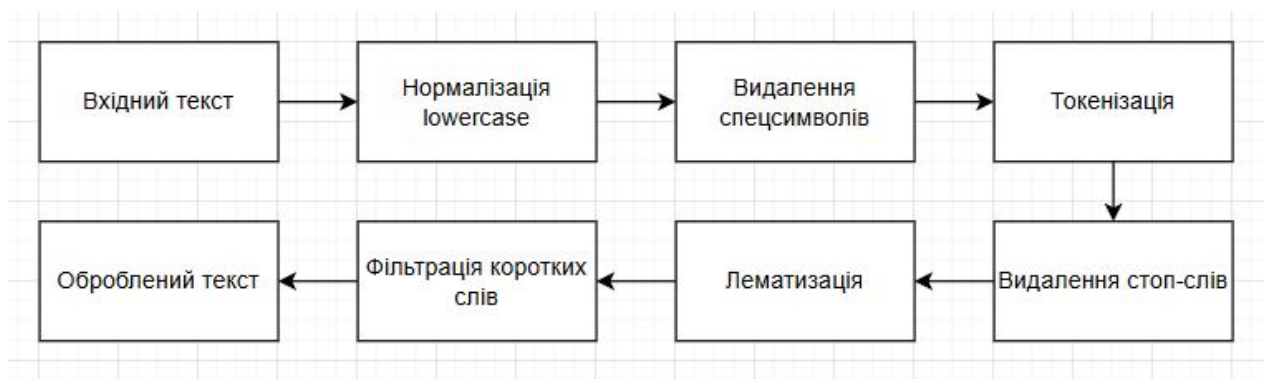


Рис. 2.2. Етапи передобробки тексту

На першому етапі передобробки відбувається текстова нормалізація, мається на увазі приведення тексту до нижнього регістру, видаленні спецсимволів (тут залишаються тільки цифри і літери) і видалення зайвих пробілів. Все це дозволяє уніфікувати тексти і зменшити їх варіативність написання слів. Вже на другому етапі виконується токенизація і фільтрація, яка включає в собі розбиття тексту на окремі слова (токени), видалення стоп-слів (stopwords) або ще їх можна назвати загальновживані слова, які несуть смислове навантаження, наприклад, “the”, “a”, “is”. Лематизація виконує приведення слів до базової форми, тобто слова “developers”, “running” після цього процесу лематизації стають “developer”, “run”, а також виконується фільтрація коротких слів в даному випадку менші 2 символів. Всі ці етапи передобробки дають змогу привести текст резюме і вакансії до стандартизованого вигляду, що дасть можливість покращити якість векторизації і точність моделі.

Після цієї передобробки відбувається векторизація тексту за допомогою методу TF-IDF (Term Frequency Inverse Document Frequency) метод перетворення тексту в числові вектори, де TF представляє частоту слова в документі, а IDF обернена частота документа, тобто, показує рідкість слова в корпусі. Далі наведена формула TF-IDF:

$$\mathbf{TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D)}$$

де:

- $\mathbf{TF(t, d)}$ - частота слова = (скільки разів слова t зустрічається в документі d) / (загальна кількість слів у документі d)
- $\mathbf{IDF(t, D)}$ - обернена частота слова у корпусі = $\log((\text{загальна кількість документів у корпусі } D) / (\text{кількість документів, що містять слова } t))$

Тут є нове слова корпус і воно має на увазі що корпус це весь набір документів разом іншими слова це всі резюме і всі вакансії у навчальному датасеті.

Далі наведена детальна формула TF-IDF з урахуванням логарифмічної нормалізації:

$$\mathbf{TF - IDF(t, d, D) = (\frac{f(t, d)}{\sum f(t', d)}) \times \log(\frac{N}{df(t)})}$$

де:

- $\mathbf{f(t, d)}$ - скільки разів слово t зустрічається в документі d ;
- $\mathbf{\sum f(t', d)}$ - сума всіх слів у документі d (загальна кількість слів);
- $\mathbf{N}$ - загальна кількість документів у корпусі D (всі резюме та вакансії разом);
- $\mathbf{df(t)}$ - скільки документів містять слово t .

Особливостями реалізації є створення окремих векторизаторів для резюме і вакансій, що зможу врахувати особливості кожного типу тексту. Параметри векторизації можуть мати максимум 3000 найважливіших ознак ($\text{max_features}=3000$), врахування біграмів ($\text{ngram_range}=(1,2)$) для послідовності 1 чи 2 слів, мінімальну частоту слова в тексті/документі 2 ($\text{min_df}=2$) і максимальну частоту 80% тексту/документів ($\text{max_df}=0.8$). Для кращого розуміння можна навести приклад, слова “Javascript” буде мати більшу вагу, якщо воно частіше зустрічається в резюме кандидата, але рідше в інших резюме.

Далі після векторизація відбувається обчислення косинусної схожості між векторами резюме і вакансії. Якщо говорити просто то косинусна схожість - це міра схожості між двома векторами, яка вимірює кут між ними у багатовимірному просторі. Наступна формула косинусної схожості:

$$\cos(\theta) = \frac{\mathbf{A} \times \mathbf{B}}{\|\mathbf{A}\| \times \|\mathbf{B}\|}$$

де:

- $\mathbf{A}$ - вектор резюме
- $\mathbf{B}$ - вектор вакансії
- $\mathbf{A} \cdot \mathbf{B}$ - скалярний добуток векторів $\mathbf{A}$ та $\mathbf{B}$
- $\|\mathbf{A}\|$ - норма (довжина) вектора $\mathbf{A}$
- $\|\mathbf{B}\|$ - норма (довжина) вектора $\mathbf{B}$
- $\cos()$ - косинус
- θ - кут між векторами $\mathbf{A}$ та $\mathbf{B}$

В результаті косинусна схожість повертає нам значення в діапазоні від -1 до 1, у випадку якщо ми отримали 1 це означає про повну схожість (вектори направлені в одному напрямку), 0 говорить про ортогональність це коли вектори перпендикулярні, а -1 повідомляє про повну протилежність, тобто, вектори направлені в протилежні напрямки. У випадку текстових даних значення в більшості випадків буде в діапазоні від 0 до 1 тому що TF-IDF вектори мають невід'ємні значення.

Для оптимізації порогу здійснюється перебір порогів від значення 0.1 до значення 0.9 з кроками 0.016, після цього відбувається вибір порогу з найкращим F1-score на навчальній вибірці. Ці дії дозволяють знайти оптимальний баланс між точністю і повнотою класифікації.

Для того щоб виконати класифікацію за допомогою SVM використовується комбінація деяких ознак: конкатенація векторів резюме і вакансії з додаванням косинусної схожості як допоміжної ознаки. Наступна наведена формула комбінованих ознак для SVM.

$X_{combined} = [X_{resume}, X_{job}, \text{cosine_similarity}(X_{resume}, X_{job})]$

де:

- X_{resume} - TF-IDF вектор резюме (розмірність 3000)
- X_{job} - TF-IDF вектор вакансії (розмірність 3000)
- $\text{cosine_similarity}(X_{resume}, X_{job})$ - косинусна схожість між векторами (скаляр)
- $X_{combined}$ - комбінований вектор ознак (розмірність 6001)

SVM є алгоритмом машинного навчання, який здатний знаходити оптимальну гіперплощину для розділення класів у просторі ознак. Далі наведена формула функції прийняття рішення SVM:

$$f(x) = \sum (a_i \times y_i \times K(x_i, X)) + b$$

де:

- x - вхідний вектор ознак
- x_i - опорні вектори (support vectors)
- y_i - мітки класів опорних векторів (-1 або +1)
- a_i - коефіцієнти Лагранжа для опорних векторів
- $K(x_i, x)$ - функція ядра (kernel function)
- b - зміщення (bias)

Для лінійного ядра $K(x_i, x) = x_i \cdot x$ (скалярний добуток), для RBF ядра $K(x_i, x) = \exp(-\gamma \times \|x_i - x\|^2)$, де γ - параметр ядра.

Гіперпараметри підбираються за допомогою GridSearchCV, де перебираються параметри $C[0.1, 1, 10]$ для регуляризації, $\text{kernel}["linear", "rbf"]$ для типу ядра та $\text{gamma}["scale", "auto"]$ для параметра RBF ядра. Такий параметр як C перевіряє баланс між максимізацією відступу між класами і мінімізацією помилок класифікацій, якщо брати більші значення C то вони призводять до жорсткішої гіперплощини, а менше до м'якішої.

Навчання відбувається на поділенні даних 80% навчальна, а 20% тестова вибірка, прогнозування відбувається як бінарна класифікація 0 (не

підходить), а 1 (підходить) і застосовується `decision_function` для того щоб нормалізувати до шкали 0-100.

Щодо переваг даного підходу є швидкість обчислень, менші вимоги до ресурсів, а недоліками є залежність від точних збігів, нездатність враховувати синоніми, розуміння змісту.

Щодо підходу на основі BERT для семантичного аналізу, для створення більш точно аналізу відповідності резюме і вакансії був обраний один із сучасних методів на основі BERT [17]. BERT - це велика нейронна мережа, навчена на великій кількості текстів, яка здатна розуміти семантику мови. Також існують і інші варіанти трансформених моделей, такі як RoBERTa [18], DistilBERT [19], TinyBERT [20], ALBERT [21], DeBERTa [22]. Відмінність від стандартних методів машинного навчання, BERT має зворотні зв'язки і здатна не лише обробляти окремі слова, а також цілі послідовності тексту з врахуванням контексту з двох сторін. На (рис. 2.3) продемонстрована загальна структура процесу оцінки відповідності за допомогою BERT.

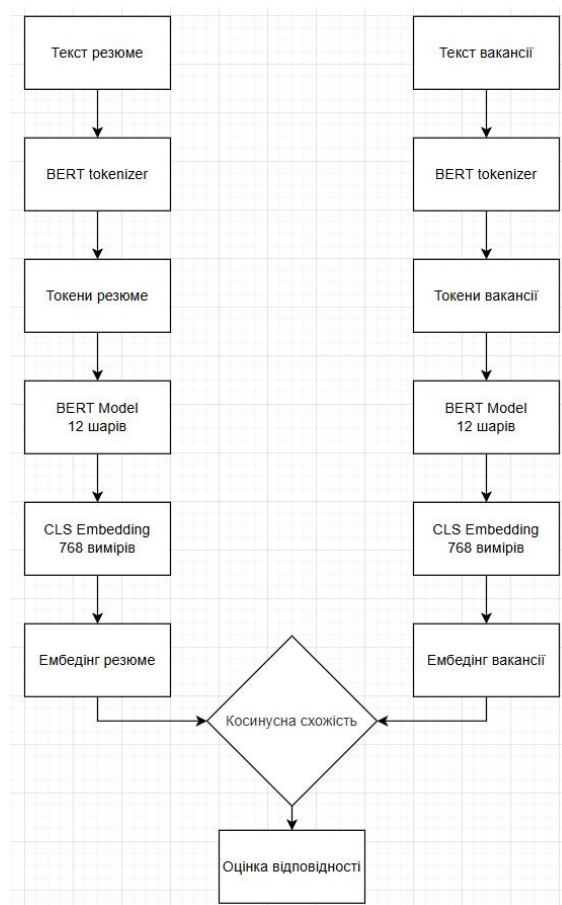


Рис. 2.3. Загальна структура процесу оцінки відповідності на основі BERT.

Особливість моделі є двонаправлене кодування щоб розуміти контекст обидві сторони. Процес отримання ембедінгів має кілька етапів. Спочатку виконується токенизація тексту резюме або вакансії за допомогою BERT tokenizer з максимальною довжиною в 512 токенів, якщо текст за довгий він обрізається, якщо короткий то доповнюється спеціальними токенами. На зображенні (рис. 2.4) зображено такий процес токеназації BERT. Додаються спеціальні токени [CLS] на початку, щоб виконати класифікації і кодування загального змісту тексту [SEP] як роздільник меж реченнями тексту. І є ще токен [PAD], який може використовувати, додаватись у випадках коли текст короткий і потрібно доповнити додатковими спеціальними токенами до максимальної довжини 512 токенів.



Рис. 2.4. Процес токенизації BERT.

Далі текст, який пройшов процес токенизації в подальшому пропускається через BERT модель [23], яка складається з 12 шарів трансформерів або ще може мати назву transformers layers. Кожний із цих шарів має механізм так званої уваги attention machanism і пов'язані шари feed-forward network. На наступному зображенні (рис. 2.5) відображено спрощену таку структуру одного такого шару BERT.

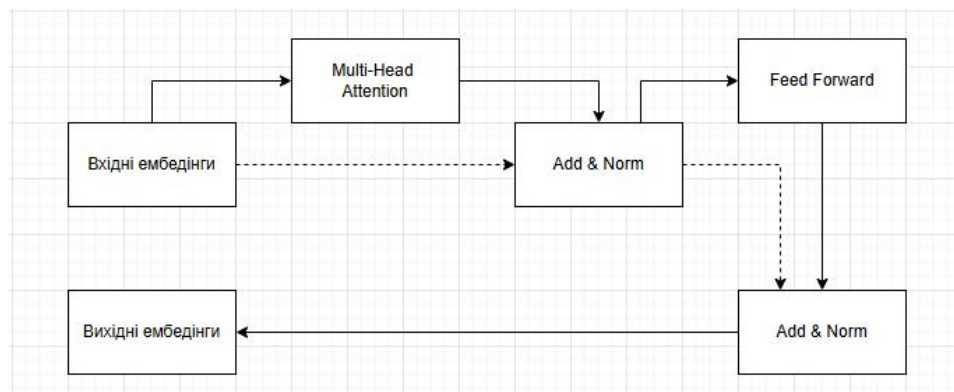


Рис. 2.5. Структура одного шару BERT трансформера.

Такий механізм як механізм уваги або ще можуть його називати *attention mechanisms* дає змогу моделі робити фокус на важливих частинах тексту і робити зв'язки між віддаленими словами. Формула механізму уваги:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q} \times \mathbf{K}^T}{\sqrt{d_k}}\right) \times \mathbf{V}$$

де:

- $\mathbf{Q}$ (Query) - матриця запитів
- $\mathbf{K}$ (Key) - матриця ключів
- $\mathbf{V}$ (Value) - матриця значень
- d_k - розмірність ключів (для BERT-base $d_k = 64$)
- softmax - функція нормалізації

BERT застосовує Multi-Head Attention, де увага робиться на обчисленні паралельно в кількох так званих “головах” для BERT-base це 12 голів, це дає можливість моделі враховувати різні типи залежностей між словами. Після проходження всіх цих 12 шарів BERT, беруться ембедінги з [CLS] токена це перший токен. В результаті 768 вимірний вектор, який кодує семантичний зміст тексту. Формула отримання ембедінгу з CLS токена зображена:

$$\mathbf{E} = \text{BERT}([\text{CLS}] \text{ token}_1 \text{ token}_2 \dots \text{ token}_n [\text{SEP}])[0]$$

де:

- $\mathbf{E}$ - 768-вимірний ембедінг тексту
- $\text{BERT}(\dots)$ - вихід BERT моделі (послідовність ембедінгів для кожного токена)
- 0 в квадратних дужках - індекс CLS токена (перший токен)

Для кращого розуміння наведемо приклад, таке слова як “база” в контексті “база даних” і “база товарів” буде мати різні векторні представлення в наслідок контекстно-залежним ембедінгам BERT, може постати питання чому так відбувається, а відповідь насправді не складна, просто BERT враховує контекст навколо слова, а не тільки саме слово як

може бути в SVM. Після того як отримали ембедінги резюме і вакансії відбувається обчислення семантичної схожості, в цьому нам знову допоможе функція косинусної схожості, яка здатна обчислювати схожість між ембедінгами [24 - 26]. Дана формула косинусної схожості для методу BERT ембедінгів:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{E_resume} \times \mathbf{E_job}}{\|\mathbf{E_resume}\| \times \|\mathbf{E_job}\|}$$

де:

- $\mathbf{E_resume}$ - 768-вимірний ембедінг резюме
- $\mathbf{E_job}$ - 768-вимірний ембедінг вакансії
- $\mathbf{E_resume} \cdot \mathbf{E_job}$ - скалярний добуток ембедінгів
- $\|\mathbf{E_resume}\|$ - норма (довжина) ембедінгу резюме
- $\|\mathbf{E_job}\|$ - норма (довжина) ембедінгу вакансії

Це все нам показує, наскільки семантично можуть бути близькі резюме і вакансія, також враховуючи синоніми і контекст, семантичний зв'язок між словами і загальний зміст, а не лише точні збіги слів. Для прикладу “Backend Developer” і “Backend Engineer” будуть мати семантичну схожість, даже якщо ці слова різні, але BERT має розуміння, що “Developer” і “Engineer” можна сказати синоніми в контексті програмування. Далі після того як обчислилась семантична схожість далі вступає в дії формула нелінійної нормалізації:

$$\text{normalized_score} = \text{similarity}^{2.5}$$

де:

- similarity - косинусна схожість між ембедінгами (діапазон 0-1)
- normalized_score - нормалізована оцінка (діапазон 0-1)

Ця дія робить оцінку більш строгою так як схожість потрібна для високих балів, що може покращити розрізнення між середніми оцінками і високими. Ну наприклад, якщо в результаті в нас схожість вийшла 0.9, то після того як ми виконаємо нормалізацією, ми отримаємо значення приблизно 77 балів замість того щоб отримати 90. Далі може відбуватись

додаткова перевірка на невідповідність ролей на основі аналізу тексту. Ця перевірка працює наступним чином вона шукає ключові слова ролей у перших 150 символах, де міститься в більшості інформація про посади, далі якщо знайдено посади, роль, наприклад, “manager”, а у вакансії написано, що шукають “developer”, то в такому випадку застосовується штраф приблизно в 70% до нормалізованої оцінки і це значно може знизити фінальну оцінку, в інших випадках якщо невідповідні ролі може застосовуватись штраф приблизно в 30%-50%. Наступна формула застосування штрафу за невідповідності ролей:

$$\text{final_score} = \text{normal_score} \times (1 - \text{penalty})$$

де:

- `normalized_score` - нормалізована оцінка після степеневої функції
- `penalty` - штраф за невідповідність ролей (0.7 для критичних невідповідностей, 0.3-0.5 для інших)
- `final_score` - фінальна оцінка відповідності

Перевагами цього підходу можна сказати є краще розуміння семантики це відбувається за рахунок врахування синонімів і контексту, вища точність на складних випадках [27, 28], а щодо недоліків то це вищі вимоги до ресурсів, менш повільні обчислення і потреба в GPU для того щоб ефективніше працювати.

2.2 Підготовка даних та навчання моделей

Для того щоб реалізувати таку оцінку відповідності резюме і вакансії потрібно використати певний набір даних, які будуть містити описи резюме кандидатів і вакансій у вигляді тексту і інформацію про їх відповідність. Під час виконання була проблема в доступі до реальних даних від компаній через конфіденційність, для вирішення цієї проблеми було прийняте рішення використати відкриті набори даних. Платформа Kaggle дає таку можливість отримати відкриті набори даних, саме тут були взяті наступні датасети

Resume Dataset [29] і LinkedIn Job Postings [30], дані набори даних мають приклад даних резюме і вакансій, які підійдуть для задачі оцінки відповідності.

Щодо першого датасету це Resume Dataset. Це csv файл, який має наступну структуру колонку “Category” і колонку “Resume”, в цього датасету є великий плюс він має різну довжину і структуру текстів, що краще відображає реальні умови.

Наступний датасет, тобто, другий датасет це LinkedIn Job Postings, теж форма файла csv, щодо структури він має наступні колонки “job_id”, “company_name”, “title”, “description”, “max_salary”, “pay_period”, “location”, “company_id”, “views”, “med_salary”, “min_salary”, “formatted_work_type”, “applies”, “original_listed_time”, “remote_allowed”, “job_posting_url”, “application_url”, “application_type”, “expiry”, “closed_time”, “formatted_experience_level”, “skills_desc”, “listed_time”, “posting_domain”, “sponsored”, “work_type”, “currency”, “compensation_type”, “normalized_salary”, “zip_code”, “fips”. Цей датасет має велику кількість і різну інформацію про вакансії, але з даного датасету нам потрібно буде тільки колонка “description”.

Для того щоб тренувати модель нам потрібний єдиний набір даних і щоб це отримати були виконані деякі дії. Спочатку було об’єднання даних, з датасетів були створені пари “резюме - вакансії” з мітками відповідності, мітка відповідності для кожної пари створювалась у вигляді бінарного значення 0 або 1. В результаті після виконання цих дій було отримано близько 5000 пар резюме-вакансія і вони були розділені на навчальну і тестову вибірку за принципом 80% навчальних даних і 20% тестових. Для оцінки якості моделей були використанні метрики і бенчмарки [31, 32]. Також можна зазначити що набори даних мають репрезентативність тому що охоплюються різні професії, сфери діяльності, кваліфікації, але можуть траплятись так званий шум це помилки, нестандартні аббревіатури, що можуть потребувати додаткової обробки для покращення аналізу.

Але такі текстові дані не дуже підходять тому що вони можуть бути різними і краще працювати із стандартизованим виглядом тому передобробка тексту є ключовим етапом. Перший етапом є текстова нормалізація, а саме приведення тексту до нижнього регістру, видалення спеціальних символів залишаєм тільки літери і числа, а також видаляєм зайві пробіли. Другим етапом є токенізація і фільтрація це розбиття тексту на окремі слова або токени, видалення стоп-слів або їх ще називають загальновживані слова, які не несуть смислового навантаження, лематизація для приведення слів до базової форми і фільтрація коротких слів. Все це дозволяє нам привести тексти резюме і вакансії до єдиного вигляду, який покращує такі аспекти як подальшу якість обробки.

2.3 Проектування додатку

Додаток з використанням оцінки відповідності резюме та вакансій повинен бути зручним у використанні для користувачів, забезпечувати автоматичну оцінку відповідності за допомогою методів машинного навчання, надаючи об'єктивні та швидкі результати для оптимізації процесу рекрутингу.

Користувачі даної системи, залежно від їхньої ролі, мають доступ до різних функціональних можливостей. Дана система забезпечує аутентифікацію користувачів, що мається на увазі вхід в систему і їхнє відповідне встановлення ролей, а також авторизацію, яка надає користувачам певний доступ до певного функціоналу в залежності від ролі самого користувача. У кандидатів є можливість створювати, оновлювати і видаляти свої резюме, у тому числа також завантажувати резюме у вигляді файлу. Також у кандидатів є можливість відгукуватись на вакансії, після чого виконається оцінювання відповідності резюме і вакансії. Щодо роботодавців то вони можуть створювати, редагувати і закривати вакансії, вказуючі вимоги вакансії, її опис і інші деталі. Усі користувачі мають можливість

переглядати результат оцінки відповідності у вигляді числового показника у діапазоні від 0 до 100.

Визначимо дві ролі: Кандидат (Employee) та Роботодавець (Employer). Їх функціональність зображена на (рис. 2.6).

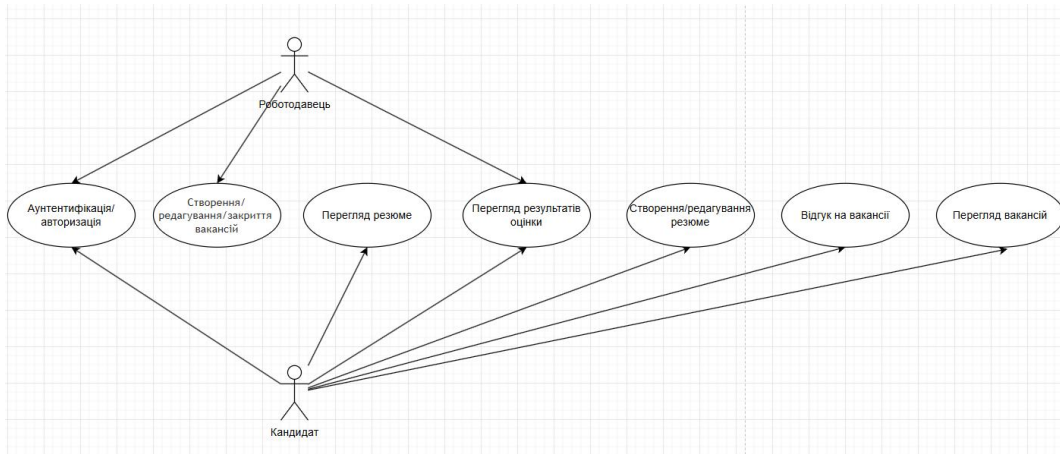


Рис. 2.6. Доступна функціональність в залежності від ролі користувача.

Визначені наступні функціональні вимоги, що система має працювати без збоїв і неочікуваних помилок; оскільки будь-яка запит, зміна інформарції є запитом до серверної частини, користувач має бути онлайн і мати стабільне інтернет-з'єднання; система має автоматично виконувати оцінювання відповідності при відгукуванні кандидата на вакансію і в результаті надати користувачу оцінку у вигляді числового показника.

Щодо нефункціональних вимог то анімації інтерфейсу не мають перевищувати 1 секунди, а час відгуку для звичайних запитів на сервер не має перевищувати 2,5 секунди, для складних запитів - 5 секунд. Інтерфейс має бути зрозуміло інтуїтивно і відповідати сучасним вимогам UI/UX дизайну. Що стосується оперативності, система має забезпечити одночасну обробку значної кількості запитів на оцінку відповідності. Вимоги до масштабності передбачається більша підтримки для збільшення навантаження без втрати продуктивності, а також при цьому збереження коректної роботи всього попереднього функціоналу після оновлення системи до новітньої версії.

Система реалізована за принципами Clean Architecture з чітким розділенням відповідальності. Структура проекту має доменний шар (бізнес-

логіка) з доменними сутностями, інтерфейсами та винятками, шар застосунку (Use Cases) з DTO для API та бізнес-логікою, інфраструктурний шар з HTTP клієнтами та ML моделями (SVM, BERT), шар представлення (API) з маршрутами та конфігурацією. Основними компонентами є PreprocessingService для передобробки тексту, MatchingAlgorithm для реалізації SVM алгоритму з TF-IDF векторизацією, BertMatchingAlgorithm для реалізації BERT алгоритму з семантичним аналізом, CalculateMatchUseCase для бізнес-логіки обчислення відповідності і MlModelService для роботи з ML моделями. Також система надає REST API інтерфейс для інтеграції з іншими сервісами платформи. Основні endpoints включають обчислення відповідності за ResumeId та VacancyId з отриманням даних із зовнішніх API, обчислення відповідності за текстом без потреби зовнішніх API викликів, пакетне обчислення для одного резюме та кількох вакансій, обчислення для одного резюме проти кількох вакансій та обчислення для однієї вакансії проти кількох резюме.

Для простішого опису роботи додатку створімо діаграму розгортання. На ній відобразимо браузер користувача, де буде розгорнута клієнтська частина, для сервера створімо окремий елемент де буде розташований сам сервер та створена бізнес-логіка, також для збереження інформації реалізуємо елемент бази даних де буде розгорнута PostgreSQL, зображено на (рис. 2.7).

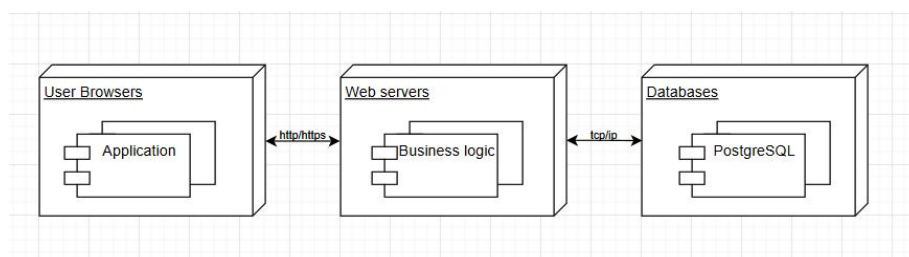


Рис. 2.7. Діаграма розгортання.

Також для більшого розуміння і кругозору мовних моделей і їх можливостей було звернено до джерел [33, 34], які описують розвиток мовних моделей і їх здатність до узагальнення знань.

Висновки до розділу 2

В даному 2 розділі була розроблена і спроектована оцінка відповідності резюме і вакансії за допомогою машинного навчання з використанням мовою програмування Python. Були обрані і реалізовані два підходи: традиційний метод з SVM з TF-IDF векторизацією і BERT для семантичного аналізу.

Підхід SVM з TF-IDF має кілька переваг це швидкість в обчисленні, менші вимоги до ресурсів, але натомість має обмеження в розумінні семантики і залежність від точних збігів. Тоді як підхід BERT має краще розуміння семантики, але має більші потреби до ресурсів, часу.

Відбулась підготовка даних з використанням датасетів з платформи Kaggle, також було створено приблизно 5000 пар резюме-вакансія з мітками відповідності у вигляді бінарного числа 0 чи 1, розділення даних на навчальний набір і тестувальний із співвідношенням 80% на 20%. Навчання моделей відбувалось в середовищі Google Colab з підтримкою GPU.

Було спроектовано систему за принципом Clean Architecture з REST API. Визначенні дві ролі кандидат і роботодавець (employee and employer).

Порівняльний аналіз ефективності методів показав, що модель BERT має кращі результати через можливість семантичного розуміння тексту, але потребує більше ресурсів, а SVM модель теж показала гарні результати при менших вимогах ресурсів і швидшому обчисленні, але є обмеження в розумінні семантики.

РОЗДІЛ 3. ОПИС ПРОГРАМНОГО ПРОДУКТУ

3.1 Налаштування до роботи з програмою

Після того як користувач відкрив ресурс в браузері він бачить стартову сторінку з коротким описом продукту, переваги оцінки відповідності резюме і вакансії з допомогою використання машинного навчання, а також кнопки “Sign In”, “Get Started”, “Upload Resume”, “Find Candidates” всі ці кнопки перенаправлять користувача на сторінку аутентифікації/авторизації або на дашборд, а ще на головній сторінці знаходиться опис. Все це можна подивитись на (рис. 3.1).

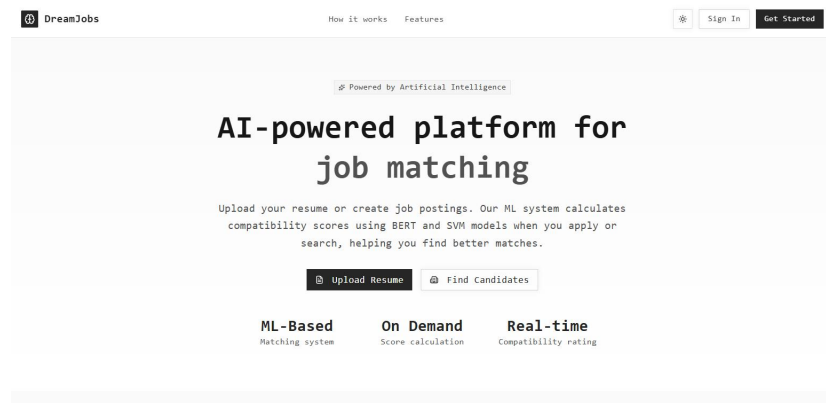


Рис. 3.1. Головна сторінка.

Після того як користувача буде перенаправлено на сторінку аутентифікації він побачить форму з табами логіну користувача або реєстрація кандидата або роботодавця. Також на цій сторінці є дві вкладки/таби “Логін” і “Реєстрація”, що дозволяє користувачам перемикатись між цими вкладками. Форма аутентифікації зображена на (рис. 3.2).

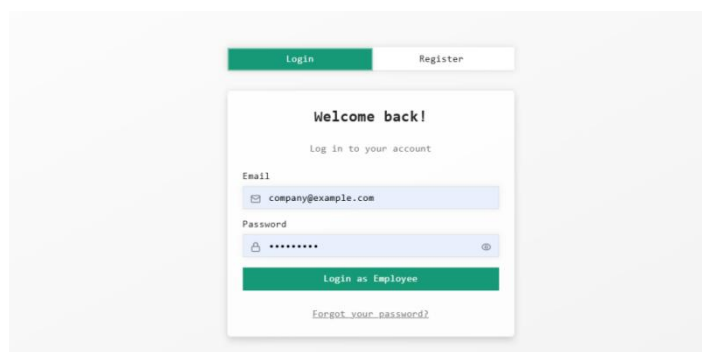
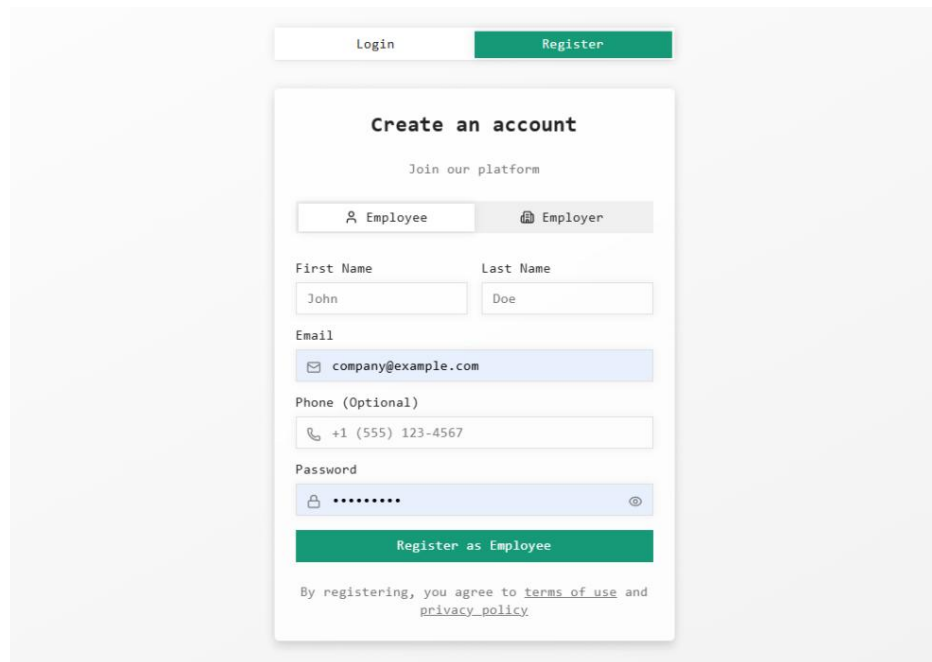


Рис. 3.2. Форма логіну.

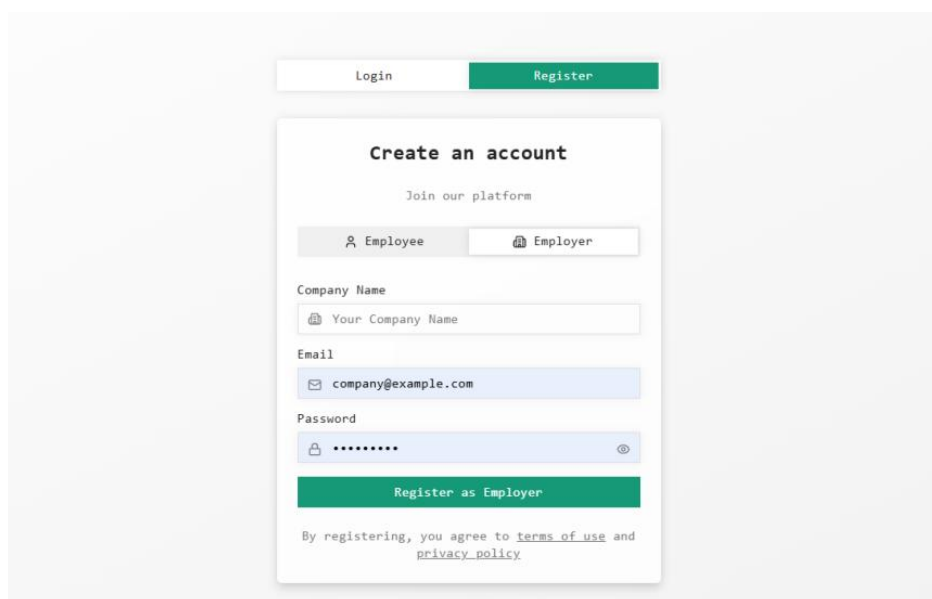
Якщо користувач обире вкладку реєстрацію відобразиться форма реєстрації користувача. У вкладці реєстрації є ще дві вкладки кандидат (Employee) і роботодавець (Employer), де користувач може зареєструватись з ролями кандидата або роботодавця. Форма для реєстрації користувача з роллю кандидат зображена на (рис. 3.3).



The screenshot shows a web interface for creating an account. At the top, there are two buttons: 'Login' and 'Register'. Below them is a 'Create an account' section with the subtitle 'Join our platform'. There are two tabs: 'Employee' (selected) and 'Employer'. The form fields include: 'First Name' (with the value 'John'), 'Last Name' (with the value 'Doe'), 'Email' (with the value 'company@example.com'), 'Phone (Optional)' (with the value '+1 (555) 123-4567'), and 'Password' (masked with dots). A green button labeled 'Register as Employee' is at the bottom. Below the button, there is a line of text: 'By registering, you agree to [terms of use](#) and [privacy policy](#)'.

Рис. 3.3. Форма реєстрації користувача з ролею кандидат.

Поля введення даних у формах для реєстрації кандидата і роботодавця звісно відрізняються. Ці поля і форму в цілому для реєстрації користувача в ролі роботодавця можна переглянути на зображенні (рис. 3.4).



The screenshot shows the same 'Create an account' web interface, but with the 'Employer' tab selected. The form fields are: 'Company Name' (with the placeholder 'Your Company Name'), 'Email' (with the value 'company@example.com'), and 'Password' (masked with dots). A green button labeled 'Register as Employer' is at the bottom. Below the button, there is a line of text: 'By registering, you agree to [terms of use](#) and [privacy policy](#)'.

Рис. 3.4. Форма реєстрації користувача з ролею роботодавець.

Далі опис буде йти про реєстрацію. Отже, якщо користувач зареєструвався під роллю кандидат, після успішної його реєстрації користувача буде перенаправлено на сторінку дашборду. Зареєструємо користувача з такою інформацією на (рис. 3.5).

The screenshot shows a registration form titled "Create an account" with the subtitle "Join our platform". At the top, there are "Login" and "Register" buttons. Below the title, there are two tabs: "Employee" (selected) and "Employer". The form fields include: "First Name" (filled with "Renat"), "Last Name" (filled with "Lakyziuk"), "Email" (filled with "renat@example.com"), "Phone (Optional)" (filled with "+380121212123"), and "Password" (masked with dots). A green "Register as Employee" button is at the bottom. Below the button, a note states: "By registering, you agree to [terms of use](#) and [privacy policy](#)".

Рис. 3.5. Реєстрація кандидата.

Після цього потрібляє на сторінку дашборд зображена на (рис. 3.6). Тут кандидат може побачити зліва меню, яке буде його перенаправляти на інші сторінки, а з іншого боку кандидату тут відображається список опублікованих вакансій, його відгуки і збереженні вакансії у відповідних вкладках на даній сторінці.

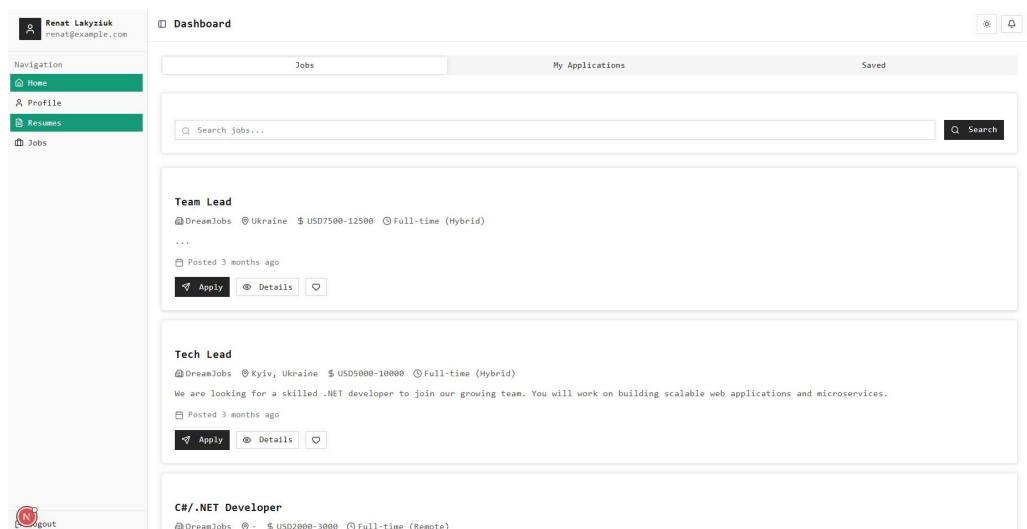
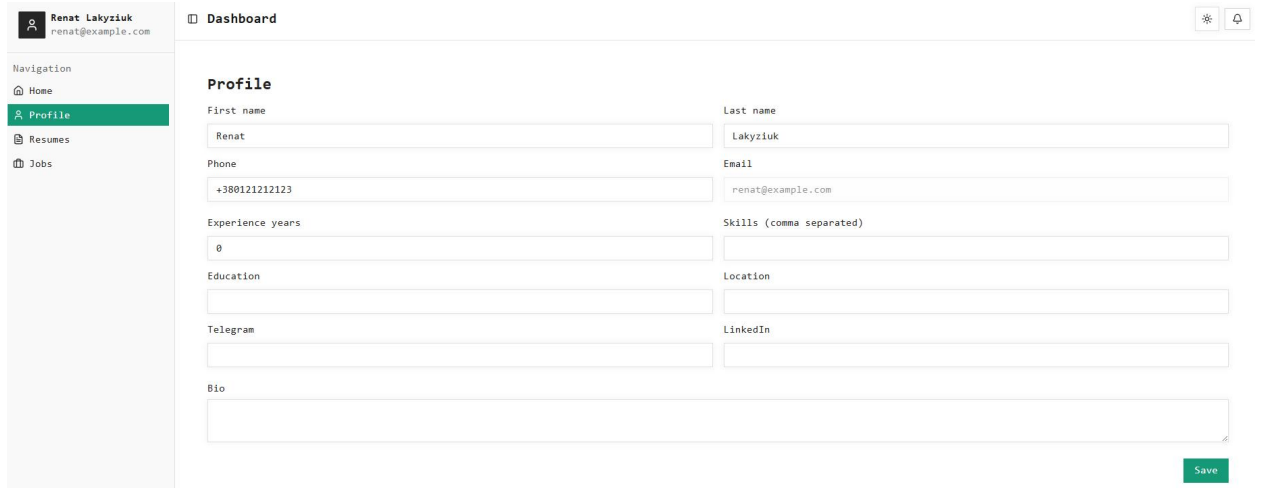


Рис. 3.6. Сторінка дашборд кандидата.

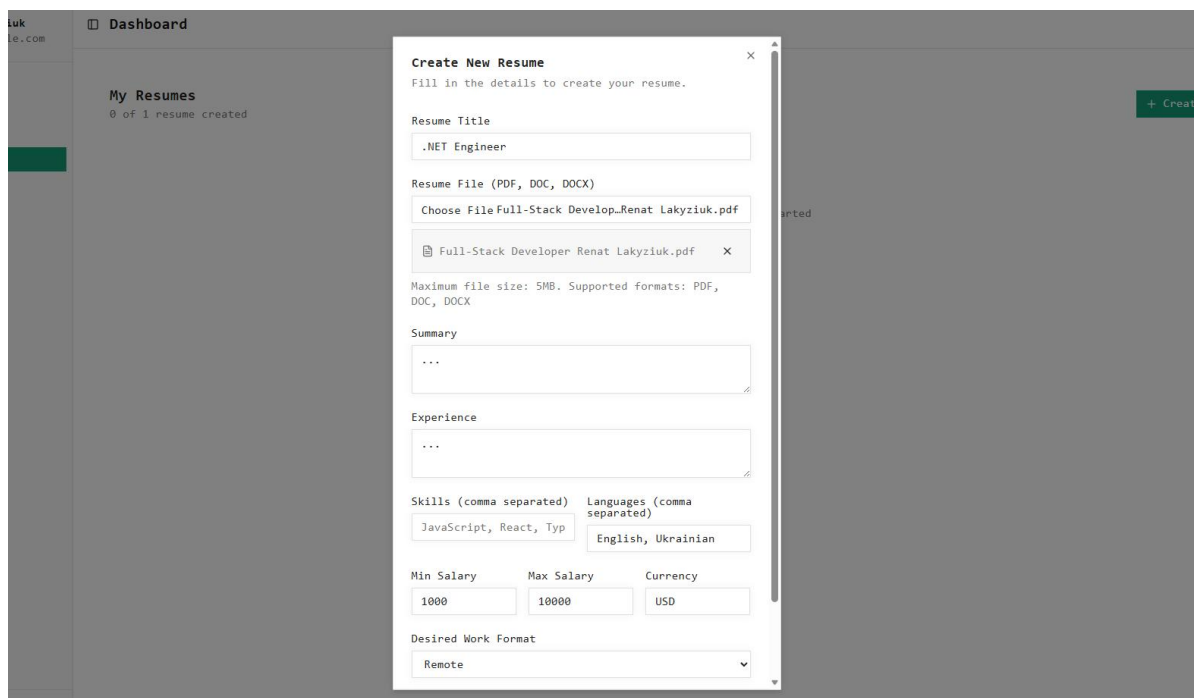
Далі користувач може перейти на сторінку профілю і додати про себе інформацію таку як ім'я, прізвище, номер телефону, електронну пошту, досвід, навички і іншу інформацію, яка буде використана для відображення інформації роботодавцю. Сторінка профілю зображена на (рис. 3.7).



The screenshot shows a user profile page within a dashboard. The user is Renat Lakzyiuk, with email renat@example.com. The page has a sidebar with navigation links: Home, Profile (active), Resumes, and Jobs. The main content area is titled 'Profile' and contains a form with the following fields: First name (Renat), Last name (Lakzyiuk), Phone (+380121212123), Email (renat@example.com), Experience years (0), Skills (comma separated), Education, Location, Telegram, LinkedIn, and Bio. A 'Save' button is located at the bottom right of the form.

Рис. 3.7. Сторінка профілю кандидати.

Також є сторінка резюме де користувач може створити резюме і завантажити файл, форма створення резюме і завантаження файлу зображено на (рис. 3.8). Резюме кандидата бере пряму участь у оцінюванні відповідності так як моделі передається контент резюме і вакансії, які в подальшому будуть опрацьовані.



The screenshot shows a 'Create New Resume' modal form. The form is titled 'Create New Resume' and has a close button (X). It contains the following fields: Resume Title (.NET Engineer), Resume File (PDF, DOC, DOCX) (Choose File Full-Stack Develop...Renat Lakzyiuk.pdf), Summary (...), Experience (...), Skills (comma separated) (JavaScript, React, Typ), Languages (comma separated) (English, Ukrainian), Min Salary (1000), Max Salary (10000), Currency (USD), and Desired Work Format (Remote). A '+ Create' button is visible in the top right corner of the dashboard background.

Рис. 3.8. Створення резюме і завантаження файлу.

Наступним кроком ми переглянемо і відгукнемось на вакансію. Після відгуку на вакансію буде запущено процес оцінювання відповідності між резюме і вакансією. В цьому процесі участь бере опис вакансії і контент файлу резюме, яке ми завантажили. На (рис. 3.9) зображено перегляд вакансії.

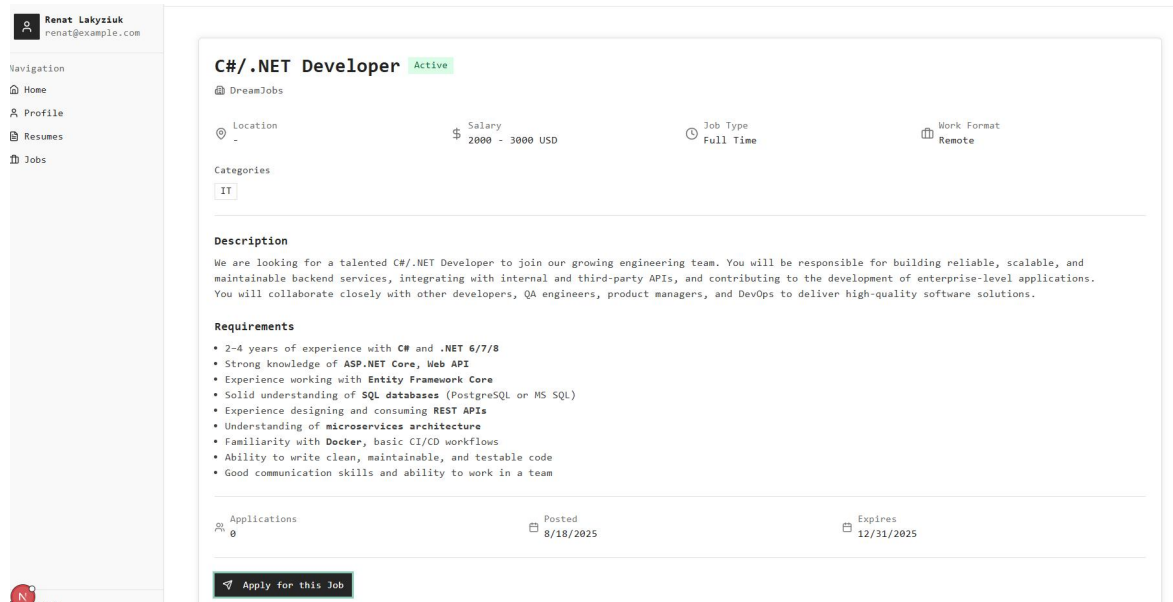


Рис. 3.9. Перегляд резюме.

На наступному зображенні (рис. 3.10) відображено процес відгуку, де є перелік резюме з яких користувач обирає, яке використати для відгуку на вакансію. Система поки обмежує кількість резюме кандидата і зараз кандидат не може створити більше одного резюме.

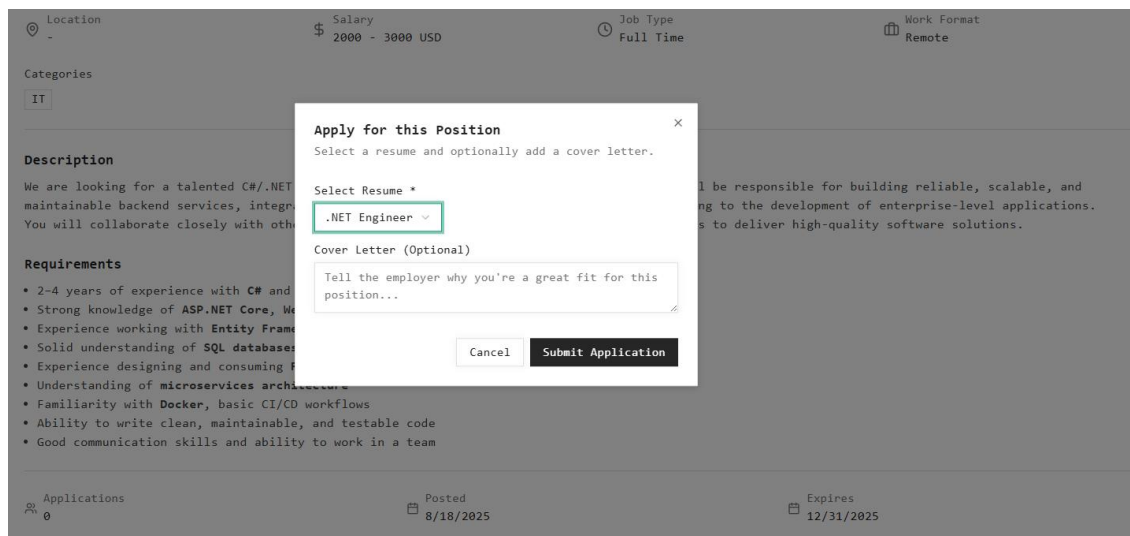


Рис. 3.10. Відгукуємось на вакансію.

Після того як ми зробили відгук на вакансію частина сторінки зміниться і ми побачимо такий користувацький інтерфейс, який зображений

на (рис. 3.11), це зображення відображає процес обробки оцінювання оцінки відповідності між резюме кандидата і вакансією на яку відгукнувся кандидат.

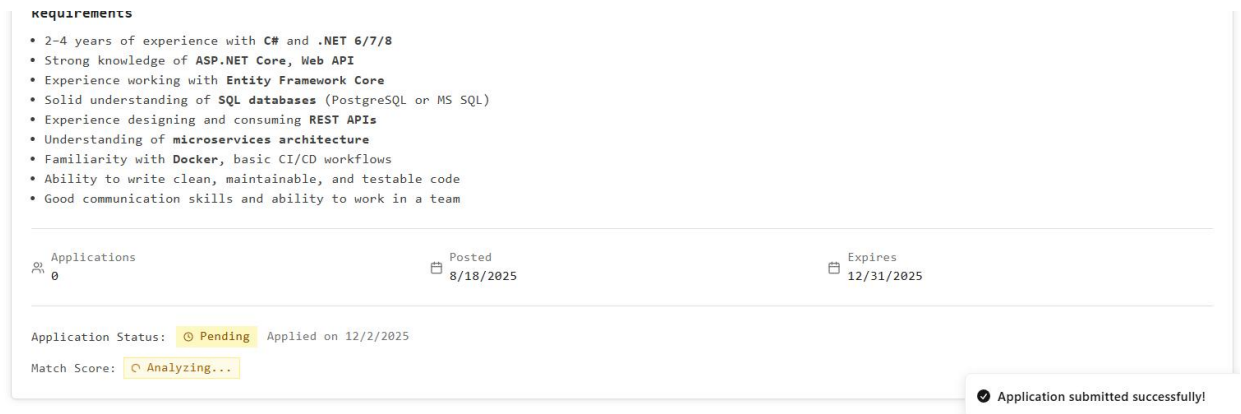


Рис. 3.11. Запуск процесу аналізу оцінки відповідності.

Через певний час аналіз оцінки відповідності буде завершено, в більшості випадків це займає часу менше хвилина. Після того як відбудеться завершення аналізу ми побачимо результат як на (рис. 3.12). На цьому зображенні відображений результат оцінювання і як його бачить результат користувач в кінцевому випадку.

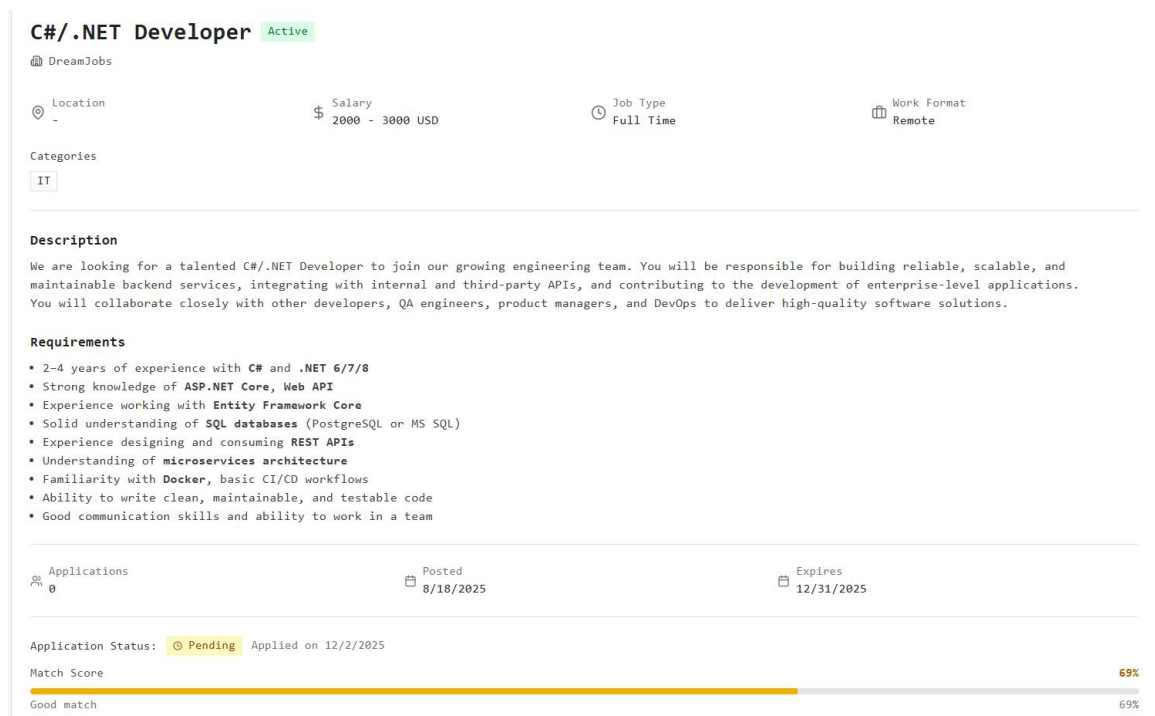
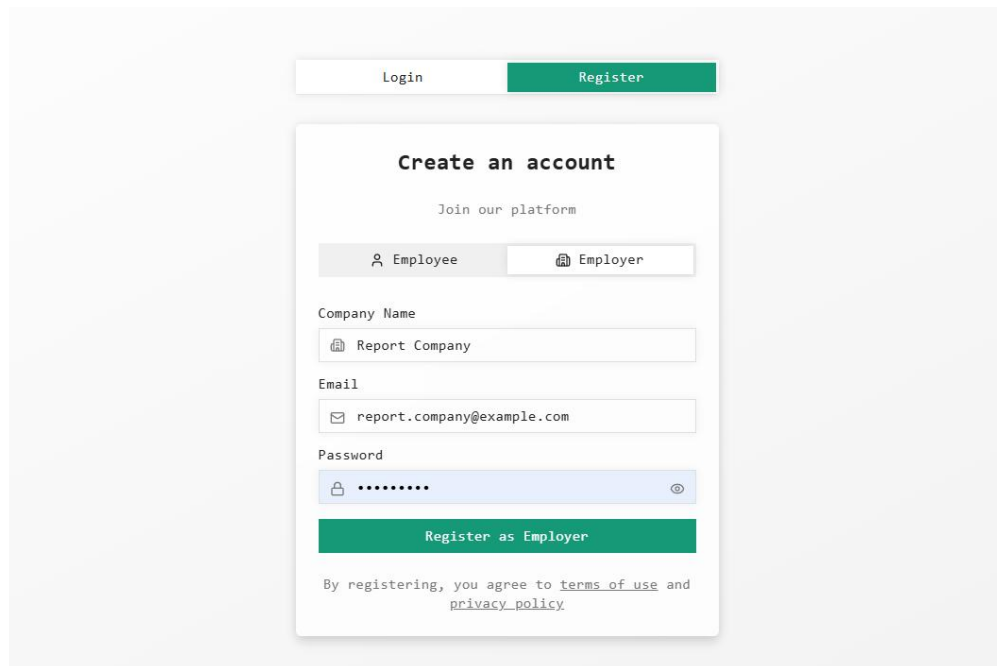


Рис. 3.12. Результат аналізу оцінки відповідності.

Далі можна перейти до огляду роботодавців. Створимо роботодавця з такими даними як на (рис. 3.13). Для того щоб зареєструвати роботодавця

потрібно у форму ввести такі дані назву компанії, електронна пошта і пароль цього достатньо для реєстрації роботодавця.



The screenshot shows a web form titled "Create an account" with the subtitle "Join our platform". At the top, there are two buttons: "Login" and "Register". Below the title, there are two tabs: "Employee" and "Employer", with "Employer" being the active tab. The form contains the following fields: "Company Name" with a sub-label "Report Company" and a button icon; "Email" with the example address "report.company@example.com"; and "Password" with a masked input field. A green button labeled "Register as Employer" is positioned below the password field. At the bottom, a text line states: "By registering, you agree to [terms of use](#) and [privacy policy](#)".

Рис. 3.13. Створення роботодавця.

Після того як ми створили успішно роботодавця, нас буде перенаправлено на сторінку дашборду роботодавця на (рис. 3.14). Після реєстрації буде відображатись тільки меню зліва, все інше буде пустим. На цій сторінці є дві вкладки в одній буде список вакансій роботодавця, а на іншій список кандидатів, які відгукнулись на вакансії роботодавця.

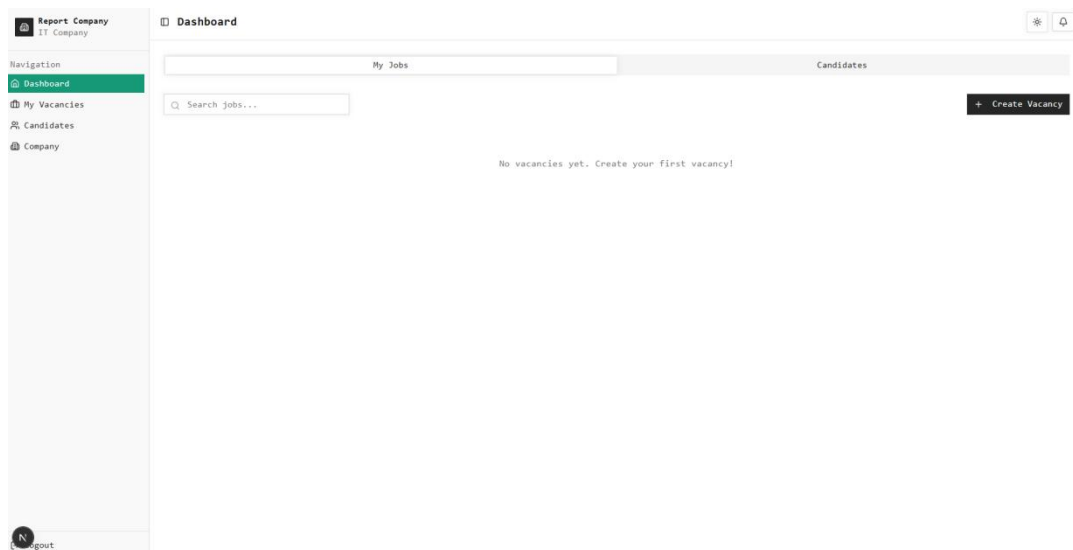


Рис. 3.14. Дашборд роботодавця.

Далі створимо нашу першу вакансію, форма створення вакансії частково зображена на (рис. 3.15). У формі для створення резюме потрібно

вказати наступну інформацію заголовок або посаду, опис вакансії, вимоги, заплатну вилку (не обов'язково), місце, тип і формат роботи.

The screenshot shows the 'Create New Vacancy' form in the 'Report Company' dashboard. The form is titled 'Create New Vacancy' and contains several input fields and a rich text editor. The 'Title' field is filled with 'Project Manager'. The 'Description' and 'Requirements' fields are empty. The 'Min Salary' field is filled with '1000', the 'Max Salary' field is filled with '2000', and the 'Currency' field is filled with 'USD'. The 'Location' field is filled with 'Ukraine'. The 'Job Type' field is filled with 'Full Time' and the 'Work Format' field is filled with 'Remote'. The form also includes a 'Logout' button in the bottom left corner.

Рис. 3.15. Форма створення вакансії.

Після створення вакансії вона буде відображатись на сторінці “Дашборд” або “Мої вакансії” (рис. 3.16). Також якщо роботодавець створить інші свої вакансії він їх тут теж зможе побачити.

The screenshot shows the 'Dashboard' page in the 'Report Company' dashboard. The page has a navigation sidebar on the left with options: 'Dashboard', 'My Vacancies', 'Candidates', and 'Company'. The main content area is titled 'Dashboard' and contains a search bar labeled 'Search Jobs...'. To the right of the search bar is a '+ Create Vacancy' button. Below the search bar is a table of jobs. The first job listed is 'Project Manager' with status 'Active', location 'Ukraine', salary '\$ 1000 - 2000 USD', and type 'Full Time Remote'. It has 0 applications and was posted on 12/2/2025. Action buttons for this job are 'View', 'Edit', 'Deactivate', and 'Delete'.

Рис. 3.16. Створена вакансія на сторінці дашборд.

Роботодавець може переглядати кандидатів і їх резюме. На (рис. 3.17) зображений список з кандидатів, які відгукнулись на вакансії роботодавця. Тут роботодавець може побачити таку основну інформацію про кандидатів ім'я, посада, мови, якими володіє користувач, заголовок резюме, кнопка на посилання резюме, дату відгуку на вакансію, кнопка для зберігання користувача у списку збережених.

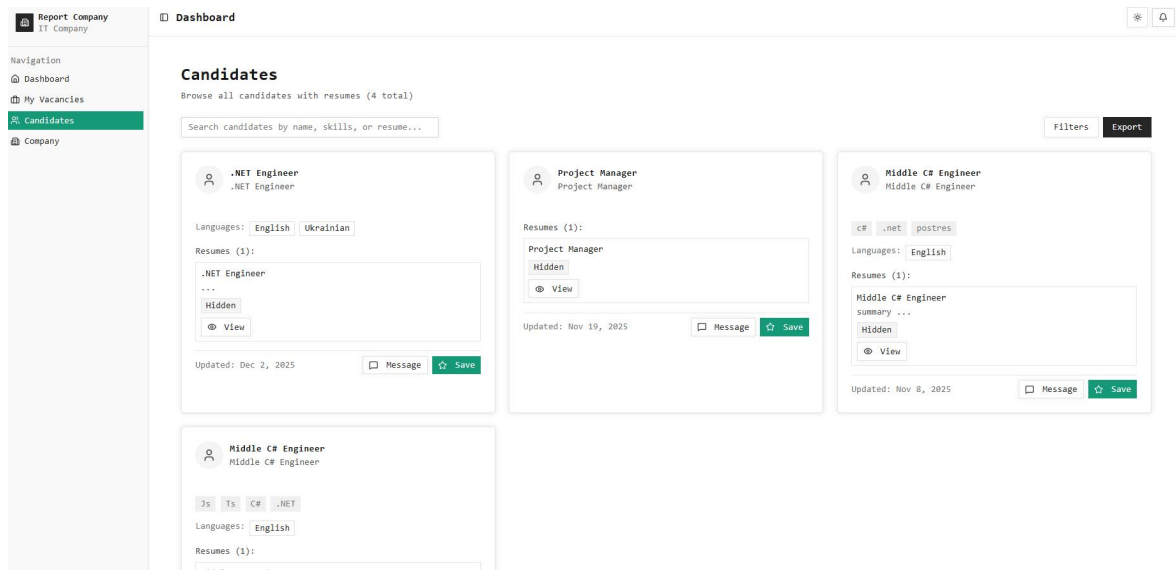


Рис. 3.17. Список кандидатів.

На наступному зображенні (рис. 3.18) зображено детальний перегляд певного кандидата і його резюме. Тут відображена вся інформація, яка буде корисна роботодавцю: назва посади, короткий опис, досвід, зарплатні очікування, формат роботи і звісно завантажене резюме у вигляді файлу кандидата, яке на цій же сторінці можна переглянути.

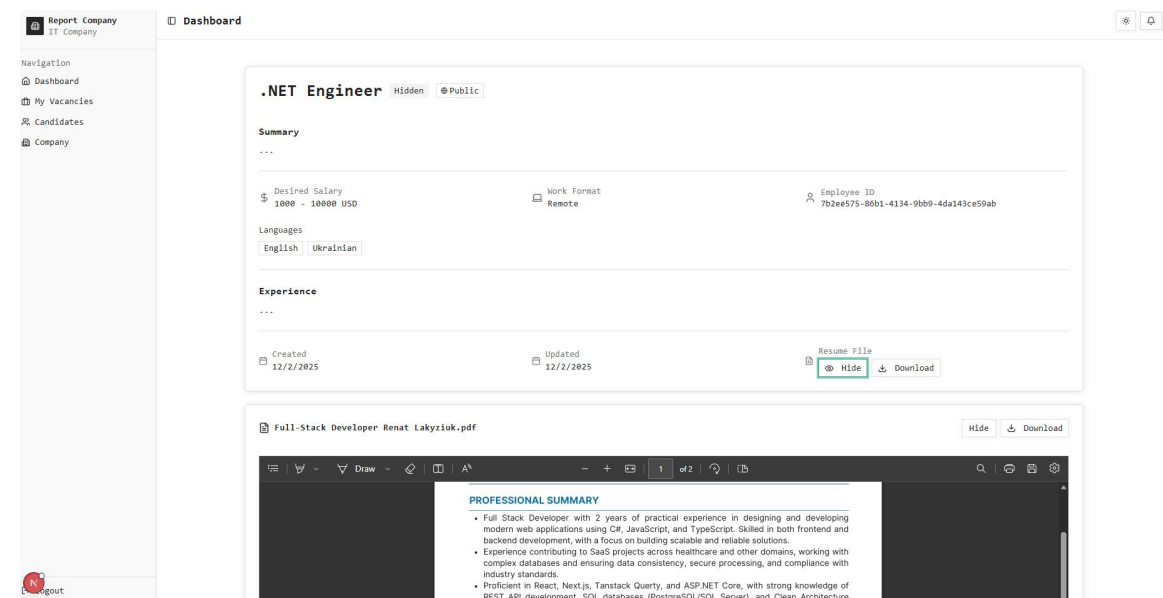


Рис. 3.18. Перегляд кандидата і його резюме.

На сторінці дашборд роботодавця є вкладка “Candidates”, де буде відображатись список кандидатів, які відгукнулись на вакансії роботодавця, на (рис. 3.19) це можна переглянути, а також тут відображається оцінка

відповідності, яка була запущена коли кандидат відгукнувся на дану вакансію і в результаті ця оцінка відображається і кандидату і роботодавцю для подальшої обробки.

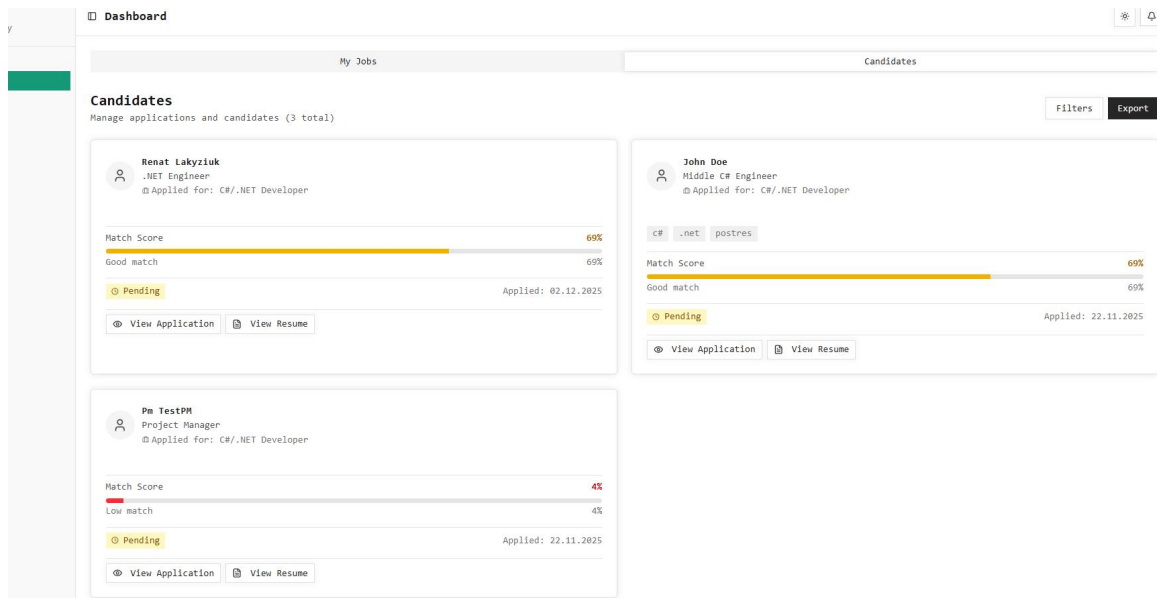


Рис. 3.19. Список кандидатів які відгукнулись на вакансії роботодавця.

3.2 Налаштування параметрів оцінки відповідності

Перед тим як використовувати оцінку відповідності резюме і вакансії потрібно налаштувати деякі параметри обчислення, які можуть змінювати поведінку систему, а також впливають на точність оцінювання. Різні типи оцінок можуть мати спільні властивості, тому більш такі загальні параметри згруповані в конфігурацію, щоб уникнути дублювання інформації і забезпечити узгодження між різними можливими компонентами системи. Це дозволить більш гнучко керувати параметрами і адаптовувати під певні вимоги, без прямого внесення змін до програмного коду. Налаштування параметрів відбувається через конфігураційний файл `.env`, який розміщений в кореневій директорії сервісу аналітики, даний файл має змінні середовища і визначає поведінку системи під час обчислення оцінки відповідності. Даний форма файлу, а саме формат `.env` є стандартним підходом у сучасній розробці для зберігання параметрів конфігурації, кожний рядок має пару типу “ключ-значення”, де ключ є назвою змінної середовища, а значення є відповідними

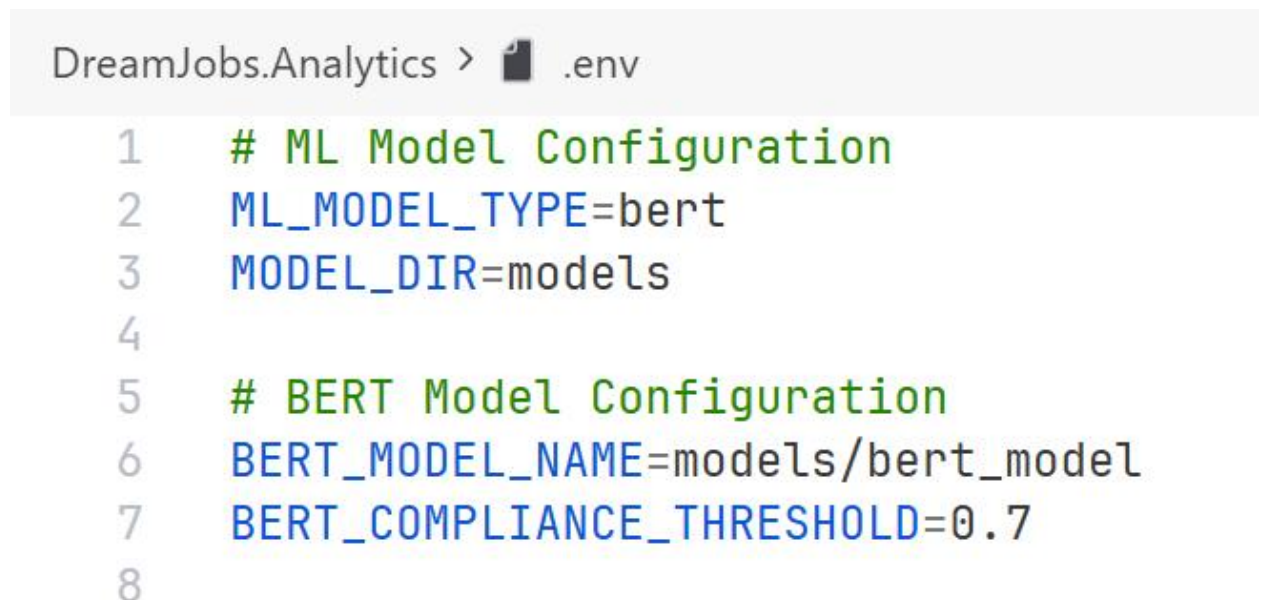
параметром налаштування. Використання такого файлу нам дозволяє певну поведінку змінювати параметри без редагування коду і дає нам більшу гнучкість налаштування для різних середовищ, наприклад, тестування, виробництво, розробка. Система має підтримку налаштування типу моделі машинного навчання, яка буде використовуватись для того щоб обчислити оцінку відповідності. В конфігураційному файлі є змінна `ML_MODEL_TYPE`, яка визначатиме, яку модель використовувати для оцінювання відповідності між резюме і вакансією. Ця змінна приймає одне значення з двох: “svm” в цьому випадку буде використаний метод опорних векторів з TF-IDF векторизацією, або інше значення “bert”, яке вказує на використання сучасної трансформерної моделі BERT із семантичними ембедінгами. За замовчуванням було вирішено призначати модель BERT іншими слова Bidirectional Encoder Representations from Transformers так як ця модель може забезпечити і показує кращу точність порівняно з моделлю SVM, але SVM або Support Vector Machine модель може бути корисна у випадках, коли більш важливішим аспектом є швидкість обробки або коли ресурси є обмеженими.

Для BERT моделі є додаткові параметри налаштування, які дають змогу більш гнучко налаштовувати процес оцінки відповідності. Основним параметром є `BERT_COMPLIANCE_THRESHOLD`, який відповідає за поріг відповідності або який визначає мінімальне значення семантичної схожості, для того, щоб пару резюме-вакансія рахувати відповідною. Значення цього параметра є в діапазоні від 0 до 1, де 0 означає відсутність схожості між контентом резюме і вакансії, коли 1 означає ідентичність переданих текстів. Якщо встановлювати значення ближче до 1, то оцінка буде більш строгою і вимагає більшу ступінь схожості. У випадку, коли встановлювати менше значення порогу оцінки то оцінка буде більш м’якою і це дозволить легше і більше виявляти відповідність при частковому збігу навичок, вимог і інше. Найбільш оптимальне значення цієї змінної є в діапазоні від 0.3 до 0.7 в залежності від вимог до точності оцінювання. Окрім цього система також

підтримує налаштування назви і шляху до навченої моделі BERT через змінну BERT_MODEL_NAME.

Також можна вказувати шлях в якій знаходяться навчені вже моделі через змінну MODEL_DIR, по замовчуванню значення встановлено models, це та директорія в якій система буде шукати локально модель.

Всі ці налаштування дозволять гнучко змінювати параметри без необхідності змінювати код, вони дозволять гнучко змінити модель, вказати шлях, директорії в якій знаходяться моделі, достатньо лише змінити параметри і перезапустити сервіс. Всі змінні env які стосуються для налаштування моделей у файлі .env наведені на (рис. 3.20):



```
DreamJobs.Analytics > .env
1  # ML Model Configuration
2  ML_MODEL_TYPE=bert
3  MODEL_DIR=models
4
5  # BERT Model Configuration
6  BERT_MODEL_NAME=models/bert_model
7  BERT_COMPLIANCE_THRESHOLD=0.7
8
```

Рис. 3.20. Налаштування моделі у файлі .env.

3.3 Відображення результатів

Коли виконалось обчислення оцінки відповідності між резюме та вакансією результат оцінки відображається користувачу у зручному та зрозумілому вигляді. Якщо оцінка відповідності ще не завершила обчислення, користувач бачить індикатор процесу обчислення, що повідомляє його про те, що ще аналізується оцінка відповідність, вже після завершення обчислення відображається числова оцінка відповідності у діапазоні від 0 до 100, а також візуальний індикатор, який допомагає швидко оцінити рівень відповідності.

Кандидат який відгукнувся на вакансію і процес обчислення оцінки відповідності пари резюме-вакансії ще відбувається, тоді користувач бачить такий інтерфейс зображений на (рис. 3.21).

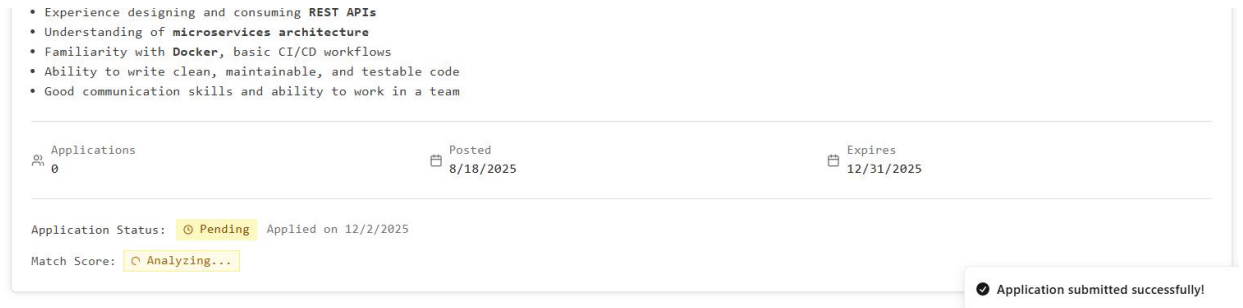


Рис. 3.21. Вакансія з оцінкою відповідності, що обчислюється

Якщо оцінка відповідності обчислена, система відображає її у вигляді прогресу з кольоровим індикатором, який залежить від значення оцінки. Для високих оцінок (70-100 балів) використовується зелений колір, що вказує на відмінну відповідність. Для середніх оцінок (50-70 балів) використовується жовтий колір, що вказує на хорошу відповідність. Для низьких оцінок (0-50 балів) використовується червоний колір, що вказує на низьку відповідність. Поряд з візуальним індикатором відображається числове значення у відсотках і текстове описування рівня відповідності: "Excellent match" для високих оцінок, "Good match" для середніх та "Low match" для низьких. Вакансія із середньою оцінкою відповідності зображена на (рис. 3.22).

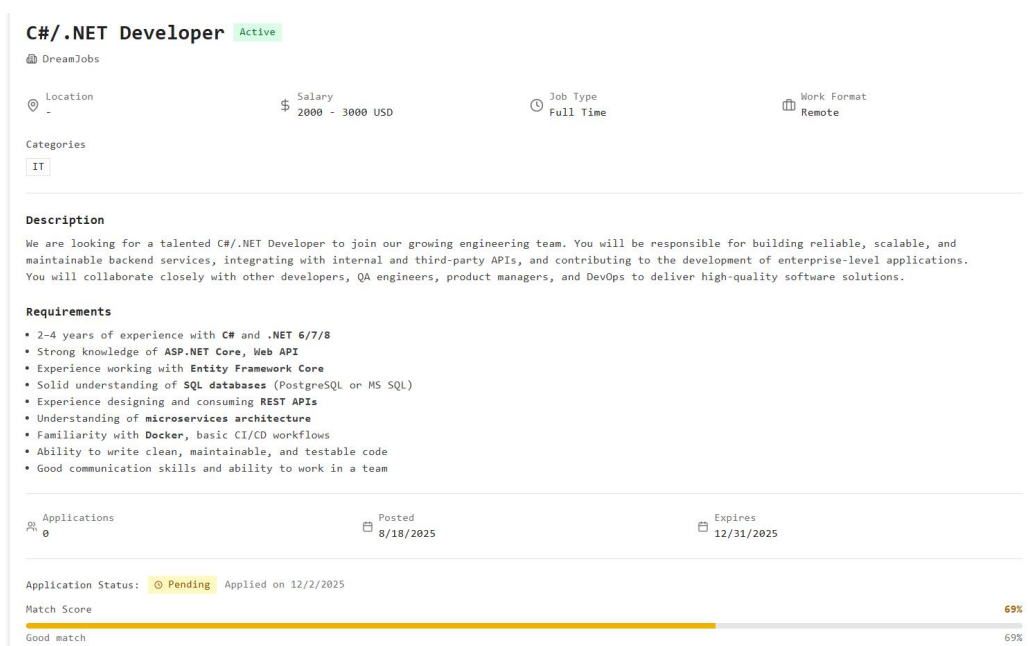


Рис. 3.22. Вакансія з високою оцінкою відповідності

Роботодавець може переглянути відгук на вакансію від кандидата і в нього теж буде відображатись ця оцінка відповідності. Інтерфейс відображення оцінки такий же, але інша частина буде відрізнятись так як для роботодавця буде доступний перегляд інформації про кандидата, його резюме, перегляд своєї вакансії на яку відгукнувся кандидат. Для роботодавця інтерфейс з відображення оцінки відповідності буде виглядати наступним чином зображено на (рис. 3.23).

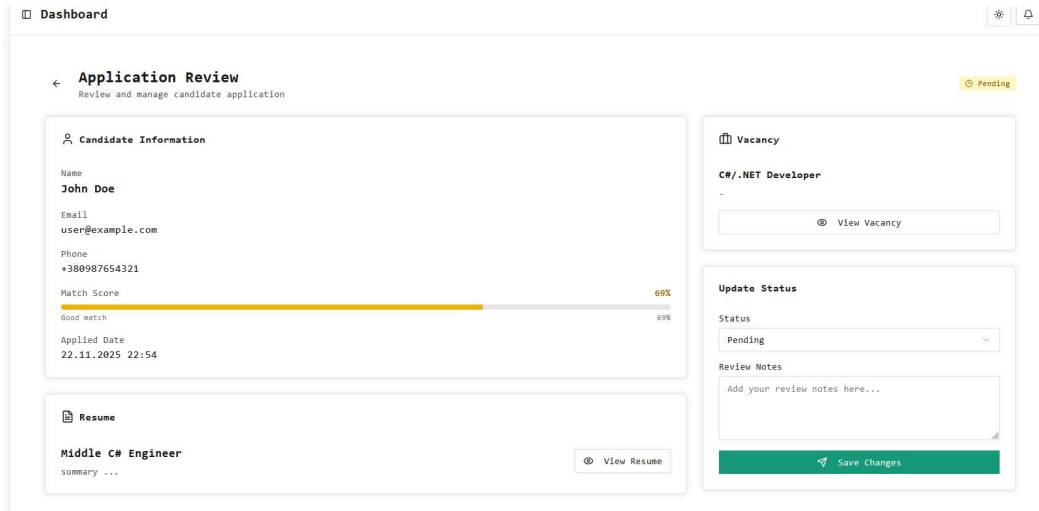


Рис. 3.23. Відображення оцінки для роботодавця.

На наступному зображенні буде відображено декілька оцінок, включно з різними. Приклади різних рівнів оцінки відповідності зображені на (рис. 3.24).

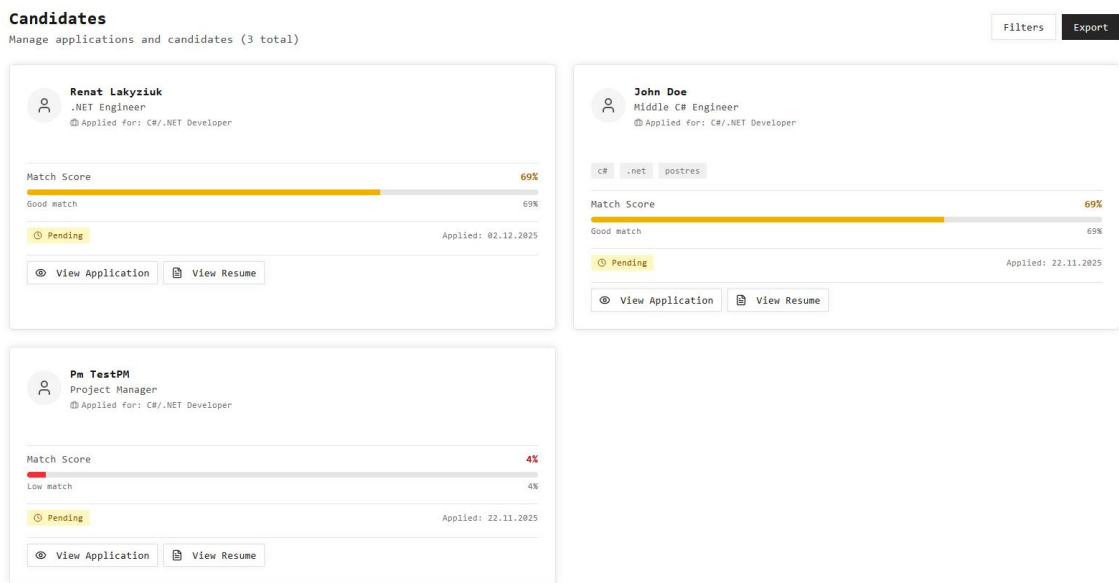


Рис. 3.24. Приклади різних рівнів оцінки відповідності

Висновки до розділу 3

В цьому розділі було описано методи як користувач може взаємодіяти з продуктом. Описано деякі варіанти налаштування параметрів оцінки відповідності через конфігураційний файл, а це дає змогу змінити модель машинного навчання і налаштувати пороги відповідності. Також був розглянутий інтерфейс, включно з інтерфейсом для різних ролей з яким буде працювати користувач. Були описані відображення результату оцінки відповідності для користувачів з різними ролями, результат відображається у вигляді візуальних індикаторів, числових значень і це все забезпечує швидкому сприйняттю результату користувачу.

ВИСНОВКИ

Під час виконання роботи було розроблено програмний продукт для оцінки відповідності резюме та вакансій, який може вирішувати певну проблему для користувача. Даний програмний продукт має призначення для управління процесом оцінки відповідності між резюме кандидатів і вакансіями роботодавців, а також включає такий механізм як автоматична оцінка відповідності на основі машинного навчання. Також було підготовлено технічну специфікацію та обрані відповідні інструменти для реалізації цієї програми.

У цій роботі описується процес підготовки даних, створення і навчання моделей машинного навчання. Для навчання моделей було розроблено систему підготовки даних на основі аналізу текстових даних резюме та вакансій, також детально описано процес вибору конкретної архітектури моделей та реалізацію алгоритмів попередньої обробки даних. Велику роль відіграють дві моделі машинного навчання: традиційний метод SVM (Support Vector Machine), який працює з локальними ознаками тексту через TF-IDF векторизацію і сучасний метод BERT (Bidirectional Encoder Representations from Transformers), який використовує архітектуру трансформерів з механізмом уваги (attention mechanism) щоб розуміння семантику тексту. Подальший порівняльний аналіз цих підходів дозволив визначити переваги і недоліки кожного методу в контексті задачі оцінки відповідності.

Після реалізації програми було проведено тестування, яке спрямоване на перевірку якості і коректної роботи продукту, а отримані результати відповідають встановленим вимогам до програми і відображають ефективність запропонованих підходів для оцінки відповідності резюме та вакансій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лакизюк Р. В. Алгоритми рекомендацій вакансій кандидатам та рекомендацій кандидатів роботодавцям // Комп'ютерні технології: інновації, проблеми, рішення : тези доп. VII Всеукр. наук.-техн. конф., м. Житомир, 02–03 груд. 2024 р. Житомир, 2024. С. 51–52. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2025/01/51-1.pdf>.
2. What is NLP (natural language processing). IBM. URL: <https://www.ibm.com/topics/natural-language-processing>.
3. Popular Machine Learning Algorithms: CART, Random Forest, GBM, XGBoost, LightGBM and CatBoost. Medium. URL: <https://medium.com/@arcyalcin/popular-machine-learning-algorithms-cart-random-forest-gbm-xgboost-lightgbm-and-catboost-2c22018cadedf>.
4. Kowsari K., Meimandi K. J., Heidarysafa M., Mendu S., Barnes L., Brown D. Text Classification Algorithms: A Survey // Information. 2019. Vol. 10, Iss. 4. Article 150. URL: <https://www.mdpi.com/2078-2489/10/4/150>.
5. Word Embedding Techniques: Word2Vec and TF-IDF Explained. Towards Data Science. URL: <https://towardsdatascience.com/word-embedding-techniques-word2vec-and-tf-idf-explained-c5d02e34d08>.
6. Qaiser S., Ali R. Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents // International Journal of Computer Applications. 2018. Vol. 181, Iss. 1. Pp. 25–29. URL: <https://www.ijcaonline.org/archives/volume181/number1/qaiser-2018-ijca-917395.pdf>.
7. Wolf T., Debut L., Sanh V. et al. Transformers: State-of-the-Art Natural Language Processing // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2020. URL: <https://aclanthology.org/2020.emnlp-demos.6/>.
8. Tunstall L., von Werra L., Wolf T. Natural Language Processing with Transformers: Building Language Applications with Hugging Face. Sebastopol : O'Reilly Media, 2022. 479 p.

9. González-Carvajal S., Garrido-Merchán E. C. Comparing BERT against Traditional Machine Learning Text Classification // Journal of Computational and Cognitive Engineering. 2023. Vol. 2, Iss. 4. Pp. 352–356. URL: <https://arxiv.org/abs/2005.13012>.
10. BERT Explained: State of the art language model for NLP. Towards Data Science. URL: <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>.
11. Devlin J., Chang M. W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019. Vol. 1. URL: <https://aclanthology.org/N19-1423/>.
12. Devlin J., Chang M. W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019. URL: <https://arxiv.org/abs/1810.04805>.
13. Bhatia P., Celikkaya B., Khalil M. End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT. URL: <https://arxiv.org/abs/1910.03089>.
14. Maheshwary S., Misra H. Matching Resumes to Jobs via Deep Siamese Network // Companion Proceedings of the 2018 World Wide Web Conference. 2018. Pp. 87–88. URL: <https://dl.acm.org/doi/10.1145/3184558.3186942>.
15. Natural Language Processing using Tfidf, Word2Vec and Bert. Medium. URL: <https://medium.com/@js2441995/natural-language-processing-using-tfidf-word2vec-and-bert-825cc2c663c3>.
16. Shahmirzadi O., Lugowski A., Younge K. Text Similarity in Vector Space Models: A Comparative Study // International Conference on Machine Learning and Applications. 2018. Pp. 659–666. URL: <https://arxiv.org/abs/1810.00664>.
17. Rogers A., Kovaleva O., Rumshisky A. A Primer in BERTology: What We Know About How BERT Works // Transactions of the Association for Computational Linguistics. 2020. Vol. 8. URL: <https://arxiv.org/abs/2002.12327>.

18. Liu Y., Ott M., Goyal N. et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. URL: <https://arxiv.org/abs/1907.11692>.
19. Sanh V., Debut L., Chaumond J., Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2020. URL: <https://arxiv.org/abs/1910.01108>.
20. Jiao X., Yin Y., Shang L. et al. TinyBERT: Distilling BERT for Natural Language Understanding // Findings of the Association for Computational Linguistics (EMNLP 2020). 2020. Pp. 4163–4174. URL: <https://arxiv.org/abs/1909.10351>.
21. Lan Z., Chen M., Goodman S., Gimpel K., Sharma P., Soricut R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations // Proceedings of the 8th International Conference on Learning Representations. 2020. URL: <https://arxiv.org/abs/1909.11942>.
22. He P., Liu X., Gao J., Chen W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention // International Conference on Learning Representations. 2021. URL: <https://arxiv.org/abs/2006.03654>.
23. Sun C., Qiu X., Xu Y., Huang X. How to Fine-Tune BERT for Text Classification // China National Conference on Chinese Computational Linguistics. 2020. URL: <https://arxiv.org/abs/1905.05583>.
24. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. 2019. URL: <https://arxiv.org/abs/1908.10084>.
25. Gao T., Yao X., Chen D. SimCSE: Simple Contrastive Learning of Sentence Embeddings // Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2022. Pp. 6894–6910. URL: <https://arxiv.org/abs/2104.08821>.
26. Wang B., Kuo C. C. J. SBERT-WK: A Sentence Embedding Method by Dissecting BERT-based Word Models // IEEE/ACM Transactions on Audio,

Speech, and Language Processing. 2020. Vol. 28. Pp. 2146–2157. URL: <https://arxiv.org/abs/2002.06652>.

27. Li Q., Peng H., Li J. et al. A Survey on Text Classification: From Traditional to Deep Learning // ACM Transactions on Intelligent Systems and Technology. 2022. Vol. 13, Iss. 2. Pp. 1–41. URL: <https://arxiv.org/abs/2008.00364>.

28. Minaee S., Kalchbrenner N., Cambria E., Nikzad N., Chenaghlu M., Gao J. Deep Learning Based Text Classification: A Comprehensive Review // ACM Computing Surveys. 2021. Vol. 54, Iss. 3. Pp. 1–40. URL: <https://arxiv.org/abs/2004.03705>.

29. Duttakiit G. Resume Dataset. Kaggle. URL: <https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset>.

30. Arshkon. LinkedIn Job Postings Dataset. Kaggle. URL: <https://www.kaggle.com/datasets/arshkon/linkedin-job-postings>.

31. Muennighoff N., Tazi N., Magne L., Reimers N. MTEB: Massive Text Embedding Benchmark // Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics. 2023. Pp. 2014–2037. URL: <https://arxiv.org/abs/2210.07316>.

32. Thakur N., Reimers N., Rücklé A., Srivastava A., Gurevych I. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. URL: <https://arxiv.org/abs/2104.08663>.

33. Radford A., Wu J., Child R., Luan D., Amodei D., Sutskever I. Language Models are Unsupervised Multitask Learners. OpenAI. 2019. URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

34. Brown T. B., Mann B., Ryder N. et al. Language Models are Few-Shot Learners // Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020. URL: <https://arxiv.org/abs/2005.14165>.

ДОДАТКИ

Код передоброби тексту

```
import re

import pandas as pd

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

stop_words = set(stopwords.words('english'))

lemmatizer = WordNetLemmatizer()

def preprocess_text(text: str) -> str:

    if pd.isna(text) or not text:

        return ""

    text = text.lower()

    text = re.sub(r'^[a-zA-Z0-9\s]', ' ', text)

    text = re.sub(r'\s+', ' ', text).strip()

    tokens = text.split()

    tokens = [

        lemmatizer.lemmatize(word)

        for word in tokens

        if word not in stop_words and len(word) > 2

    ]

    return ' '.join(tokens)
```

Код тренування SVM моделі

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.svm import SVC
from sklearn.metrics.pairwise import cosine_similarity
from scipy.sparse import hstack, csr_matrix

# TF-IDF векторизація
resume_vectorizer = TfidfVectorizer(max_features=3000, ngram_range=(1, 2))
job_vectorizer = TfidfVectorizer(max_features=3000, ngram_range=(1, 2))

X_resumes = resume_vectorizer.fit_transform(resumes_processed)
X_jobs = job_vectorizer.fit_transform(jobs_processed)

# Косинусна схожість
similarities = [cosine_similarity(X_resumes[i], X_jobs[i])[0, 0]
                 for i in range(len(resumes_processed))]

# Комбінація ознак
X_final = hstack([X_resumes, X_jobs, csr_matrix(np.array(similarities).reshape(-1, 1))])

# Підбір гіперпараметрів
param_grid = {'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf'], 'gamma': ['scale', 'auto']}
grid_search = GridSearchCV(SVC(random_state=42), param_grid, cv=3, scoring='f1')
grid_search.fit(X_train, y_train)
```

Код отримання BERT ембедінгів

```
import torch

from transformers import BertTokenizer, BertModel

tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertModel.from_pretrained('bert-base-uncased')

def get_bert_embeddings(text):
    inputs = tokenizer(text, return_tensors='pt', max_length=512, truncation=True, padding=True)

    with torch.no_grad():
        outputs = model(**inputs)

    cls_embedding = outputs.last_hidden_state[:, 0, :]

    return cls_embedding.squeeze(0).cpu().numpy()
```

Код оцінки відповідності з нелінійною нормалізацією

```
from sklearn.metrics.pairwise import cosine_similarity

def calculate_match_score(resume_embedding, vacancy_embedding, role_penalty):

    # Косинусна схожість

    similarity = cosine_similarity([resume_embedding] [vacancy_embedding])[0, 0]

    # Нелінійна нормалізація (степенева функція)

    power_factor = 2.5

    normalized_similarity = similarity ** power_factor

    base_score = normalized_similarity * 100

    # Застосування штрафу за невідповідність ролей

    adjusted_score = base_score * (1.0 - role_penalty)

    # Обмеження при низькій схожості

    if similarity < 0.5:

        adjusted_score = min(adjusted_score, 30.0)

    return round(adjusted_score, 2)
```