Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Система інтелектуального порівняння профілів для рекомендації робочих вакансій
(тема)

Виконав:
студент 2 курсу, групи
Пільгук Ю.Ю.
(прізвище, ініціали)

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

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник д.т.н., проф. Терзіян В. Я.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Пільгуку Юрію Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Система інтелектуального порівняння профілів для рекомендації робочих вакансій
затверджена наказом університету від 29 березня 20 21 р. № 390Ст
2. Термін подання студентом роботи до екзаменаційної комісії _____ _____ 20 ___ р.
3. Вихідні дані до роботи науково технічні публікації, дані Інтернет, дані відомих наукових проектів, електронні документації, тестові набори даних

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної області та постановка задачі дослідження, розглянути інші теоретичні дослідження для задачі, алгоритми класифікації, опорно-векторні машини, Elasticsearch як інструмент для пошуку та рекомендацій, середовище реалізації, алгоритми, аналіз отриманих результатів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Загальний вигляд експерименту, Рисунок 2 – Розміщення елементів на веб-сторінці, Рисунок 3 – Відповідь API веб-сайту в JSON форматі, Рисунок 4 – Відмінності між стемінгом і лематизацією, Рисунок 5 – Приклад стеммінгу, Рисунок 6 – Хмаринка лічильник слів, Рисунок 7 – Візуалізація алгоритму Random Forest, Рисунок 8 – Візуалізація алгоритму SVM, Рисунок 9 – Процес навчання під контролем, Рисунок 10 – Структура сховища в Elasticsearch, Рисунок 11 – Приклад парсингу веб-сторінки, Рисунок 12 – Кількість записів про посади, Рисунок 13 – Приклад даних з розбиттям по частоті, Рисунок 14 – Приклад інформації яка береться для визначення посади користувача LinkedIn, Рисунок 15 – Матриця плутанини, Рисунок 16 – Розшифровка матриці плутанини, Рисунок 17 – Порівняння моделей на графіку

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
2 Теоретичні дослідження	д.т.н. проф. Терзіян В. Я.		
3 Методологія	д.т.н. проф. Терзіян В. Я.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	29.03.2021	виконано
2	Аналіз завдання та пошук літератури за темою	01.04.2021	виконано
3	Аналіз та ознайомлення з іншими дослідженнями	14.04.2021	виконано
4	Вибір програмних засобів для розробки системи	19.04.2021	виконано
5	Розробка програмного засобу	21.04.2021	виконано
6	Аналіз отриманих результатів	22.04.2021	виконано
7	Оформлювання пояснювальної записки	23.04.2021	виконано
8	Оформлення презентаційних матеріалів	02.05.2021	виконано
9	Представлення на рецензування	14.05.2021	виконано
10	Представлення кваліфікаційної роботи	18.05.2021	

Дата видачі завдання 29 березня 2021 р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 80 с., 13 табл., 17 рис., 2 дод., 55 джерел.

АЛГОРИТМИ, АРХІТЕКТУРА, ВАКАНСІЯ, МАШИННЕ НАВЧАННЯ, МОДЕЛЬ, РЕЗЮМЕ, РЕКОМЕНДАЦІЙНА СИСТЕМА, ELASTICSEARCH, PYTHON

Об’єктом дослідження є можливість застосування машинного навчання для визначення узагальненої посади для вакансії або резюме та його використання в рекомендаційній системі.

Основними методами дослідження є застосування алгоритмів машинного навчання, а саме опорно-векторних машин, random forest, використання парадигми навчання з учителем для моделювання, опираючись на результати попередніх досліджень, а також обробка та аналіз отриманих даних.

В якості рекомендаційної системи пропонується застосування Elasticsearch, як сучасного інструменту для обробки і пошуку даних з його внутрішніми інструментами що дозволяють інтегрувати власні моделі ML і використовувати їх для прогнозування.

Була створена модель та вивчена можливість її застосування для прогнозування узагальненої посади для вакансії, а також перевірено її на резюме кандидатів, для того щоб на основі отриманих прогнозів-передбачень мати змогу рекомендувати, як посади кандидатам, так власне і самих кандидатів які підходять до тої чи іншої посади.

На основі отриманих результатів було зроблено висновок про поліпшення видобутку даних для побудови моделі. Використання більш прогресивних алгоритмів машинного навчання і доцільність використання машинного навчання в рекомендаційних системах в цілому. Також зроблено висновки про перспективність застосування в суміжних сферах, таких як маркетинг.

РЕФЕРАТ

Пояснительная записка: 80 с., 13 табл., 17 рис., 2 доп., 55 источников.

АЛГОРИТМЫ, АРХИТЕКТУРА, ВАКАНСИИ, МАШИННОЕ ОБУЧЕНИЕ, МОДЕЛЬ, РЕЗЮМЕ, РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА, ELASTICSEARCH, PYTHON

Объектом исследования является возможность применения машинного обучения для определения обобщенной должности для вакансии или резюме и его использование в рекомендательной системе.

Основными методами исследования является применение алгоритмов машинного обучения, а именно опорно-векторных машин, random forest, использование парадигмы обучения с учителем для моделирования, опираясь на результаты предыдущих исследований, а также обработка и анализ полученных данных.

В качестве рекомендательной системы предлагается применение Elasticsearch, как современного инструмента для обработки и поиска данных с его внутренними инструментами позволяющими интегрировать собственные модели ML и использовать их для прогнозирования.

Была создана модель и изучена возможность ее применения для прогнозирования обобщенной должности для вакансии, а также проверено ее на резюме кандидатов, для того чтобы на основе полученных прогнозов-предсказаний иметь возможность рекомендовать, как должности кандидатам, так, собственно, и самих кандидатов подходящие к той или другой должности.

На основе полученных результатов был сделан вывод об улучшении добычи данных для построения модели. Использование более прогрессивных алгоритмов машинного обучения и целесообразность использования машинного обучения в рекомендательных системах в целом.

ABSTRACT

Explanatory note: 80 p., 13 tabl., 17 fig., 2 ann., 55 sources

ALGORITHMS, ARCHITECTURE, ELASTICSEARCH, JOBS, MACHINE LEARNING, MODEL, PYTHON, RECOMMENDED SYSTEM, RESUME, SUMMARY

The object of the research is the possibility of using machine learning to determine a generalized position for a vacancy or resume and its use in a recommendation system.

The main research methods are the use of machine learning algorithms, namely support vector machines, random forest, the use of the supervised learning paradigm for modeling, based on the results of previous studies, as well as the processing and analysis of the data obtained.

As a recommendation system, the use of Elasticsearch is proposed, as a modern tool for processing and searching data with its internal tools that allow you to integrate your own ML models and use them for forecasting.

A model was created and the possibility of its application to predict a generalized position for a vacancy was studied, and it was also tested on the resume of candidates in order to be able to recommend both positions to candidates and, in fact, the candidates themselves suitable for that or another position. The accuracy of the resulting model was also analyzed, as well as options for its improvement.

Based on the results obtained, it was concluded that data mining was improved for building a model. The use of more advanced machine learning algorithms and the feasibility of using machine learning in recommender systems in general. Also, conclusions are drawn about the prospects of application in related fields, such as marketing.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

ABT – Analytical Base Table – аналітично базова таблиця;

AI – Artificial Intelligence – штучний інтелект;

ATS – Application Tracking System – система управління кандидатами;

BI – Business Intelligence – бізнес аналітика;

CNN – Convolutional Neural Network – згорткова нейронна мережа;

DDL – Data Definition Language – мова визначення даних;

DNN – Deep Neural Network – глибинна нейронна мережа;

IT – Information Technology – інформаційні технології;

ML – Machine Learning – машинна навчання;

NLP – Natural Language Processing – обробка природної мови;

RF – Random Forest – випадковий ліс;

RNN – Recurrent Neural Network – рекурентна нейронна мережа;

SQL – Structured Query Language – структурована мова запитів;

SVM – Support Vector Machines – опорно векторні машини.

ВСТУП

Пошук роботи – це та проблема, з якою зустрічається кожен. Тому для будь якої платформи, яка займається рекрутингом, дуже важливо рекомендувати саме ті посади, які дійсно відповідають вмінням того чи іншого кандидата. Так само і в зворотному напрямку: для тої чи іншої посади потрібно підібрати того кандидата, який буде максимально відповідати всім вимогам до посади.

Для кваліфікованої людини огляд резюме або опису до вакансії та її розуміння є дуже простим завданням. Для комп'ютера ця задача ускладнюється нескінченною виразністю людської мови, різноманіттям оформлення, опису та структурою резюме кандидатів і описом вакансій. В класичному підході з повнотекстовим пошуком така система не буде все це враховувати і тому може видавати хибні результати. На противагу таким системам можна застосувати сучасні технології машинного навчання та обробки природних мов (NLP).

В роботі досліджується можливість застосування технологій машинного навчання та обробки природних мов для обробки резюме та вакансій, їх подальшої класифікації та застосування в рекомендаційній системі.

За своєю природою резюме та/або вакансії мають досить довільну структуру. Тому важливо отримати якусь структуровану форму при попередній обробці даних.

Головним етапом є класифікація вакансій і самих резюме на основі загального опису за допомогою машинного навчання. Що в свою чергу допомагає визначити приналежність резюме до тої чи іншої вакансії.

Як ключовий елемент класифікації є узагальнена посада/назва посади – це всеосяжний дуже короткий опис форми, який передає всю відповідну інформацію, що стосується роботи. Вона, як правило,

інкапсулює – область, роль та рівень відповідальності будь-якої даної роботи та кваліфікації робітників.

Посади є дуже важливими як усередині організацій, так і для власне самих власників посад. Організації визначають всіх співробітників на основі назв посад. Це стосується таких питань, як заробітна плата, рівень і шкала відповідальності, вибір працівників тощо. Працівники отримують цінність із власних назв посад як засіб самоідентифікації, і це може мати значний вплив на їх залучення та мотивацію.

На додачу до всього ні самі вакансії ні резюме кандидатів ніяк не стандартизовані і нічим не обмеженні. Існують звісно якісь корпоративні стандарти але всі вони будуть різнитися від організації до організації. І відповідно це все ускладнює задачу рекомендацій вакансій та кандидатів на ці вакансії.

Тому саме автоматична класифікація вакансій та резюме, на основі їх опису з застосування методів обробки природних мов NLP та машинного навчання ML, дозволить віднести весь «зоопарк» можливих варіантів і підходів що до опису і структури як резюме так і вакансій до спільних категорій назв посад. Які потім зможуть бути використані для рекомендації.

Також в нас час кожна компанія або уряд збирають величезну кількість даних про своїх робітників і не тільки, як приватну так і публічну інформацію [55]. Тому в якості побічного ефектну це дослідження може бути використано для аналізу і впорядкування великих даних (Big Data), що стосуються робітників, кандидатів та робочих вакансій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Передумови

Посади відіграють дуже важливу роль в організаціях. Невелика кількість слів, що складає назву посади, повинна узагальнювати все, що відповідає певній ролі, і передавати її повне значення. Назви посад, як правило, містять конкретну сферу, навички та рівень відповідальності цієї конкретної функції в організації [1]. Позначаючи людей назвами посад, організація тепер має можливості розподілити людей по відділах суміжних видів діяльності.

Назви посад також є джерелом самоідентифікації окремих працівників. В рамках організації використовуються візитні картки та / або підписи електронною поштою, не більше ніж наше ім’я та назва посади, і ця інформація надає читачеві значну кількість інформації. Поза організацією типова відповідь на запитання *«чим ти займаєшся?»* часто передає свою посаду; а не інші дескриптори, такі як *«одружений, батько двох дітей, люблю шахи, читати та подорожувати ...»*. У нас немає еквівалентного конспекту дуже короткої форми для нашого особистого життя, тому наш професійний має велике значення.

Тому саме посада є елементом що дозволить її визначити або виділити з опису вакансії і відповідно з опису резюме. Що в свою чергу дозволить рекомендувати для користувачів посади а для рекрутів організацій кандидатів.

Але існує таке явище як інфляція назв посад. Одне дослідження показало, що 46% працівників отримали підвищення по службі з новою посадою, хоча обов'язки в основному залишались незмінними. Це використовувався як інструмент під час рецесії в компаніях, щоб мотивувати та утримувати працівників без поліпшення їх фінансових можливостей [2].

З інфляцією посад пов’язане поняття самоназви. Самоназва це коли працівник розробляє власну назву посади, щоб передати цінність, яку, на його думку, він приносить організації. Це ще один недорогий інструмент для підвищення мотивації та залучення працівників, який подається як соціальна перевага, яку ми додаємо до назв посад [3].

Іронічні самоназви посад з метою збільшення цінності самоідентифікації ускладнює класифікацію. Наприклад, коли кандидат на роботу шукає нову роль, звичайною дією при перегляді веб-сайту про оголошення про роботу є введення власного найменування посади або бажаного заголовка роботи у функцію пошуку, отримання списку результатів та прогресу звідти. З прикрашеними або альтернативними назвами вакансій цей процес менш ефективний, оскільки пошук за ключовими словами не дасть усіх назв вакансій.

Традиційно організаційна структура мала жорстку структуру формулювання. Наприклад, конкретні слова позначають порядок стажу, наприклад: молодший, щоб асоціюватися до старшого, віце-президента до президента. Наступним аспектом є приналежність; наприклад, фінанси, IT або маркетинг. Але сучасні назви руйнують цей структурований підхід; де, наприклад, міститься посада «AWS evangelist». Що це за посада із чим її їсти не відомо.

Навіть поза сучасною тенденцією до нетрадиційних назв посад, існує також проблема недостатньої узгодженості назв посад у багатьох організаціях. Організація може мати діючі правила щодо імен присвоєння назв посад, але ці правила будуть відрізнятися від організації до організації. Наприклад, організація може використовувати загальний термін «System Administrator» як загальний заголовок для осіб, які займаються підтримкою інфраструктури. Беручи до уваги, що інша організація може мати іншу назву, що стосуються конкретної посади; наприклад «DevOps Engineer» або інший заголовок, наприклад «Delivery Manager». Всі троє можуть бути сильно взаємозв’язані з точки зору притаманних їм ролей та обов’язків. Але

в той же час всі відрізняються між собою заголовками які є абсолютно не релевантними.

1.2 Проблема

Як правило, класифікація працівника або робочого місця потенційного працівника зазвичай виконується як суб'єктивна вправа на основі візуального зчитування деталей резюме або опису вакансії, самим кандидатом або менеджером по персоналу. Такий підхід може призвести до особистої інтерпретації, деякої невідповідності а можливо навіть упередженості.

В контексті рекомендаційної системи яка працює з сотнями організацій та мільйонами резюме. Розбіжності в інтерпретації, в формі подання та опису між ними ускладнюють цей аналіз, тому що не існує єдиного стандарту який би відповідав тій чи іншій посаді.

В класичному підході з повнотекстовим пошуком існує безліч проблем пов’язаних саме з невірним інтерпретуванням того чи іншого тексту, наприклад для пошуку такої вакансії як *human resource* по ключовим словам може знаходити як вакансії що відносяться до конкретної спеціальності-посади так наприклад і спеціальності наприклад хірургів бо і там і там зустрічається слово *human* але в контексті спеціальності хірург це може бути *human body*.

Проблема, яку намагається вирішити це дослідження, полягає в тому, щоб як найкраще класифікувати деталі роботи-професії за узагальненою назвою посади шляхом застосування методів машинного навчання ML та обробки природних мов NLP, застосовуючи систематичний та алгоритмічний підхід і на основі цього покращити систему рекомендації вакансій та пошуку кандидатів для цих вакансій.

Організаціям важливо, отримати саме тих кандидатів, що відповідають їх вимогам до конкретної вакансії і саме тих кандидатів які

будуть відповідати якомога точніше критеріям що підходять до вакансії, поставленим фінансовим цілям і кваліфікації. Для кандидатів в свою чергу важливо, щоб вони отримували ті рекомендації що до вільних вакансій які в повній мірі відповідають їх вмінням, навичкам та досвіду.

Як один із побічних результатів автоматичного визначення посади, може бути надання кандидатам інформації що до їх рівня і відповідності до тої чи іншої професії а також надання рекомендацій що до можливісті підвищення кваліфікації та підвищення свого потенціалу для подальшого заробітку. Це в свою чергу буде відкидати всі компанії які не відповідають рівню даного кандидати або потенційним можливостям його зростання.

Тому поставлене питання дослідження таке: наскільки передбачення узагальненої назви спеціалізації тобто посади з опису вакансії або резюме можливе завдяки використанню supervised learning?

1.3 Цілі дослідження

Можна виділити наступні цілі для даного дослідження:

– вивчення зв’язку між посадами та можливостями передбачення або узагальнення назви посади, використовуючи детальний опис вакансії та резюме кандидатів;

– побудувати модель та використати supervised learning алгоритми, щоб робити ці прогнози;

– оцінка результатів настройки параметрів обраних алгоритмів та їх можливі альтернативи;

– застосування моделей для рекомендаційної системи.

1.4 Методології дослідження

Робота базується як на первинних, так і на вторинних дослідженнях.

Первинне дослідження проводитиметься у формі експериментів, що проводяться для створення моделей прогнозування на основі навчання з кількісним аналізом результатів цих моделей.

Вторинне дослідження проводитиметься у формі огляду теоретичних досліджень, проведених для того, щоб спостерігати та представити будь-які відповідні дослідження в цій галузі.

1.5 Обмеження

Дані, що використовуються, являються вторинними за своєю суттю, вони зібрані з веб сайту для розміщення вакансій який по своїй суті націлений лише для північно американського ринку. Інформація на веб сайті є публічною і містить лише актуальну інформацію в конкретний момент часу. Тому для накопичення якомога більшої кількості даних вони збиралися протягом року. Для полегшення експерименту дані обмежуються 15 найпоширенішими категоріями посад.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Вступ

В даному розділі розглянуто ряд досліджень, щоб забезпечити теоретичну базу для експерименту та дізнатись, які процеси, технології та інструменти найбільш підходять для вирішення даної проблеми дослідження. Так як спрямованість даного дослідження пов’язана з назвами вакансій та їх описом і відповідно їх певними характеристиками, а саме те що більшість із них являють із себе викладення із застосуванням нейтральної або фактичної мови (тобто перечислення якихось фактів). Тому для вивчення було взято саме такі дослідження, що аналізують подібні типи даних, та мають релевантні висновки з точки зору застосування відповідних алгоритмів, які можуть бути відтворені або використані для цілей цього дослідження. Найбільші цікаві дослідження, з моєї точки зору, наведені нижче.

2.2 Рекомендації людина-робота

Рекрутинг є основною функцією управління людськими ресурсами. І традиційні зусилля виміряти придатність між працівниками та робочими місцями найкраще сформулювали в «Personality-Job Fit» теорії [4], яка виділяє шість типів особистості (реалістичну, слідчу, художню, соціальну, кооперативну та звичайну) [5] і пропонує, те що відповідність між типом особистості та професійним середовище визначає задоволеність роботою та плинністю праці. Також ця теорія отримала широке визнання в наукових колах та промисловому світі, як саме виміряти особистість та придатність між роботою та шукачами роботи є життєво важливою проблемою для кожного рекрутера що з цим стикається. Традиційно профіль особистості людини вимірюється за допомогою добре побудованої анкети, а придатність

визначається рекрутерами без об'єктивної метрики. Очевидно, що цей метод суб'єктивний і часто призводить до упереджень.

Через вибух ринку онлайн-підбору персоналу, аналіз підбору персоналу стає привабливим і привертає до себе все більше уваги [6] від дослідників. Традиційно, як правило, розглядають Person-Job Fit як проблему рекомендації роботи / кандидата. У 2006 р. Малиновський та ін. намагалися знайти хороший збіг між талантами та роботою за допомогою двох різних систем рекомендацій [7] Потім Діабі та ін. представив рекомендаційну систему на основі вмісту, для рекомендування робочих місць користувачам Facebook та LinkedIn [8]. Лу та ін. використовували профілі робочих місць та користувачів та дії, що проводяться користувачами для запропонування гібридної системи рекомендацій [9]. Щоб вирішити проблему, через яку претенденти на роботу не оновлюють профіль користувача своєчасно, Wenxing та інші розширюють профілі користувачів динамічно за допомогою записів про роботу якою вони цікавляться та їх поведінку для кращої рекомендації [10].

Чжан та ін. використав спільну фільтрацію та деяку довідкову інформацію для рекомендації підходящих робочих місць для кандидатів [11].

Нещодавно деякі дослідники намагалися вивчити проблему посади з нових перспектив. Наприклад, Paparrizos та інші використовували історію переходу з роботи на роботу а також дані, пов’язані із працівниками та установами, щоб передбачити наступний перехід на роботу працівника [12]. Чжан та ін. [13] створили узагальнену лінійну змішану модель (GLMix), більш дрібну модель на рівні користувача або елемента, для системи рекомендацій роботи LinkedIn та створили від 20% до 40% більше заявок на роботу. Лі та ін. запропонували підхід до застосування стандартизованих даних сутності для покращення якості пошуку роботи в LinkedIn та для покращення результатів пошуку [14]. У [15], інформація про роботу витягується із соціальної мережі та використовується для побудови

міжкомпаніальної мережі перебору робочих місць, що наочно демонструє потік талантів. Сю та ін. вимірювали популярність навичок роботи шляхом моделювання генерації мережі навичок [6]. Лін та ін. запропонували спільно моделювати як текстову (наприклад, огляди), так і числову інформацію для вивчення прихованих структурних моделей компаній [16]. Шен та ін. намагалися підвищити ефективність набору персоналу шляхом інтелектуального оцінювання співбесід [17].

Незважаючи на те, що вищезазначені дослідження досліджували різні аспекти досліджень посад (person-job fit), небагато з них можуть навести зрозумілі причини, що підтверджують результати роботи / рекомендацій кандидатів, що приносить користь як роботодавцям, так і шукачам роботи.

2.3 Майнінг тексту за допомогою нейронної мережі

Оскільки резюме та оголошення про роботу є текстовими даними, підбір посад можна розглядати як відповідність між текстами. Нещодавно глибока нейронна мережа (DNN) стала однією з найгарячіших методик у цій галузі завдяки своїм хорошим характеристикам.

DNN, що застосовуються при обробці тексту, можна умовно розділити на дві категорії: загорткову нейронну мережу (CNN) та рекурентну нейронну мережу (RNN). CNN націлений на моделювання ієрархічних взаємозв’язків та вилучення локальної семантики. Зусилля щодо застосування CNN у видобутку тексту можуть бути даними [18], де Kalchbrenner та ін. творчо запропонував динамічну згорткову нейронну мережу (DCNN) для моделювання речень. Тоді багато дослідників почали вирішувати проблеми NLP за допомогою CNN. У роботі [19], Kim та ін. продемонстрував, що CNN, навіть просто використовуючи згортковий рівень, також надзвичайно добре виконує багато завдань NLP. На відміну від CNN, RNN добре моделює взаємозв'язки послідовностей та знаходить глобальну семантику [54]. Таким чином, він дуже добре справлявся з

проблемами послідовного маркування при обробці тексту, такими як машинний переклад [20] та розбір контексту [21].

При аналізі тексту, як машинний переклад, так і багатомовна вбудованість слів вивчають вирівняні дані і таким чином, подібні до даної проблеми. Нейронний машинний переклад, метою якого є побудова нейронної мережі для читання речення та виведення правильного перекладу, є новим підходом до машинного перекладу. Більшість цих робіт належать до сімейства Encoder-Decoder. Наприклад, Суцкевер та ін. запропонував використовувати LSTM для відображення речень у реченнях [20]. В дослідженні [22], Чо та співавт. запропонували нову керовану рекурсивну згорткову нейронну мережу для основи трансляції на основі кодера-декодера. Девлін та співавтори запропонували нове формулювання спільної моделі нейронних мереж що дало вагомі емпіричні результати [23]. З іншого боку, багатомовна вбудованість слів спрямована на відображення слів з різних мов у спільний прихований простір. [24], Lauly та співавтори намагалися використовувати автокодер для вивчення подань слів. Подібним чином, Hermann на ін. намагалися привласнити подібні вкладені вирівняні речення для вивчення семантичних уявлень без вирівняних слів [25].

Очевидно, що як нейронний машинний переклад, так і багатомовна вбудованість слів є хорошими рішеннями для вивчення взаємозв’язків між вирівняними даними, які також існують у проблемі посад. Однак більшість із цих методів, такі як [25] потребують узгоджених взаємозв’язків на рівні речення або навіть рівня слів, чого в принципі не спостерігається в проблемі рекомендації посад.

2.4 Інші суміжні дослідження

В ході опитування в галузі веб-майнінгу, зробленого Косалою та Блокліком, стаття визнає той факт, що за допомогою звичайних

інструментів пошуку результати повертаються з низькою точністю. У статті пропонується потенційна основа для веб-майнінгу, включаючи використання NLP та машинного навчання для цілей видобутку веб-контенту для кращих результатів пошуку [26].

Го та інші провели дослідження, побудувавши двигун, який співвідносить резюме посадовим вакансіям [27]. З онлайн оголошень про роботу формується модель роботи на основі посади, галузі навчання, рівня освіти та компетенцій. Подібний процес застосовується до резюме кандидатів. Відстані подібності обчислюються по кожному з чотирьох компонентів моделі роботи. Це дослідження є корисним з точки зору відповідності дослідницькій проблемі в моєму дослідженні, оскільки воно зазначило, як процес узгодження може здійснюватися на основі спеціально розроблених моделей характеристик роботи.

У своєму дослідженні Nigam Lafferty & McCallum вивчають застосування максимальної ентропії (ME) на трьох різних наборах даних, щоб робити багатокласні прогнози [28]. Одним із наборів даних був WebKB; це сукупність даних з інтернет сторінок, взятих з різних університетів, класифікованих на сім типів веб-сторінок. Дані складалися з 4199 сторінок із 23 830 словами. Другим набором даних була ієрархія галузевого сектора; це колекція з 6440 веб-сторінок, класифікованих за галузевим сектором, 71 клас. Цей корпус містив 29 964 слова. Третім набором даних був набір груп новин, колекція з 20 000 статей, розділених на 20 дискусійних груп. Цей корпус містив 57 040 слів. Рівень помилок класифікації варіювався від 7,9% для WekKB, до 15,8% для набору даних Newsgroups та 21,1% для набору даних про галузь промисловості.

У дослідженнях, проведених Cavnar і Trenkle, автори застосували n-грами для класифікації тем [29]. Набір даних складав 778 статей із п’яти груп новин Usenet, а текстові дані містились у семи категоріях поширених запитань, що використовувались як мітки. Результати класифікації варіювали від 30% до 80%. Найнижчий коефіцієнт класифікації – 30% для

предмету AI. Дослідники обґрунтували, що, враховуючи, що текстові дані в FAQ для AI на сьогоднішній день є найбільшою підмножиною часто заданих питань складають 33% всього набору часто заданих питань – ця підмножина була занадто великою і впливала на результати, оскільки вони засновані на частоті n-грамів.

Також було проведено порівняльне дослідження різних типів n-грамів на наборі даних 50 000 оглядів фільмів у базі даних IMDB за чотирма різними алгоритмами; Nave Bayes, Maximum Entropy, Support Vector Machines і Stochastic Gradient Descent [30]. Виконане спостереження полягає в тому, що точність алгоритмів зменшується із збільшенням значення n в n-грамах. Там, де n дорівнює 3 або більше, показники точності зменшуються. Автори відзначають, що слабкою стороною уніграми є те, що речення з «не добре» класифікуються як нейтральне (одне позитивне та одне негативне слово), де насправді «не добре» є цілком негативним настроєм. Тут біграми є більш інформативними, ніж уніграми для виявлення справжніх емоцій. Враховуючи таку поведінку стає зрозумілим що застосування n-грамів буде непридатним для цілей експерименту, враховуючи те, що основна частина тексту в деталях роботи зазвичай є нейтральною без емоційною.

Наступне дослідження досліджувало, як Nave Bayes та SVM як класифікаційні показники двійкових результатів були порівняні між собою [31]. Набори даних включали короткі текстові дані з веб-сайтів для мікроблогів та мікрооглядів та довші текстові дані в формі оглядів фільмів із Usenet та публікацій у блогах. Встановлено, що SVM ефективніше працює з довшим текстом і, отже, збільшує розмірність. Автори застосували SVM лінійного ядра з вартістю = 1 у своїй моделі SVM. Переглядаючи різні алгоритми в ході мого дослідження, SVM виявився найбільш підходящим класифікатором для цього експерименту через підтримку великої довжини тексту. SVM може впоратися з великою розмірністю, пов'язаною з видобутком великого тексту.

У дослідженні, проведеному китайськими дослідниками, класифікація електронної пошти зі спамом була вивчена як англійською, так і китайською мовами, з використанням набору даних «lingsam» близько 3000 електронних листів та набору даних електронної пошти CERENT, з яких було обрано 110 [32]. Підхід полягав у векторизації набору даних, причому слова кожного електронного листа були значенням x, а значення y – двійковим класифікатором, який вказував на наявність спаму чи ні. Порівнювали точність різних алгоритмів за різних станів попередньої обробки тексту. Встановлено, що алгоритм SVM стабільно має вищу точність значення, ніж інші алгоритми, такі як Максимальна Ентропія, та варіанти Nave Bayes. За допомогою порівняння SVM, проведеного для різних типів ядра та параметрів, лінійне ядро збігалося або працювало краще, ніж різні поліноміальні ядра. Крім того, як лінійний, так і поліноміальний ефективніші, ніж радіальне ядро.

Somprasertsi & Lalitojwong провели дослідження, щоб відокремити характеристики продукту від думки про продукт, використовуючи частину мовлення промарковану за допомогою Maximum Entropy на наборі даних оглядів цифрових камер з www.amazon.com [33]. Набір даних включає 1250 речень загалом. Слова-іменники та словосполучення іменників вважалися словами, що характеризують продукт, а прикметники – словами думки. Це дозволило здійснити процес диференціації.

У своєму дослідженні Ванг та Меннінг провели аналіз SVM та NB на 8 різних наборах даних різної довжини; від огляду одного речення до оглядів довшої форми та інших наборів текстових даних [34]. Вони дійшли висновку, що біграми покращили ефективність аналізу настроїв. Крім того, вони дійшли висновку, що SVM працював краще, ніж Nave Bayes, з довшим текстом.

Подібне дослідження було проведено Koprinska та співавт., В якому вони використовували два набори електронної пошти по 1099 випадків та 2893 випадки [35]. Автори виявили, що з вибором функцій на основі

отримання інформації, Random Forest перевершив з точки зору точності класифікації та оцінки F1 порівняно з трьома іншими моделями (Decisions tree, SVM та Nave Bayes) при класифікації спаму в обох наборах даних. За ознаками, обраними на основі термінології, Random forest був другим за показником продуктивності на основі тих самих показників.

У своїх дослідженнях Noh та інші проводили експерименти зі стратегії ключових слів у галузі аналізу патентних документів [36]. Огляд існуючих досліджень показав, що кількість значущих ключових слів, як правило, становить близько 30 ключових слів. Слова в патентах є дуже формальними, а не природними парадигмами, подібно до того, як посадова інструкція має більш офіційний характер. Це відображається в інших дослідженнях [37], проведених на корпусі з ACM з 3 606 документів, розбитих на вісім наборів даних. Автори оцінили кількість ключових слів від 5 до 100 за 21 різним алгоритмом та моделями ансамблів із п’ятьма різними стратегіями вибору ознак. Найбільш різке збільшення точності було в діапазоні ключових слів від 5 до 25 ключових слів, коливаючись приблизно від 0,62 до 0,83 точності відповідно. Найкраща точність була досягнута в діапазоні ключових слів 80-85 при значенні точності 0,93. Подібну тенденцію спостерігали і з оцінками F1 з найбільш різким покращенням до 25-30 слів, причому 85 – найкраща оцінка F1. Однак вища точність та оцінка F1 80-85 для 25-30 ключових слів потенційно можуть бути обумовлені вищим рівнем розмірності.

В аналізі показників ефективності завдань з класифікації Соколова та Лапальме стверджують, що для багатокласового експерименту з класифікації, де з ряду класів вибирається один клас – доступні два варіанти [38]. Для загальної оцінки ефективності цими двома варіантами є макро усереднення та мікро усереднення. Макро усереднення – це просте усереднення результатів окремого класу, а мікро - це середньозважена міра. Висновків щодо придатності одного над іншим не зроблено. Дослідження зазначає стосовно класифікації тексту (і в контексті двійкової класифікації),

що тенденція полягає у використанні точності та продуктивності. Обґрунтування полягає в тому, що негативні класи в класифікації тексту не є справжнім негативом з деякими основними властивостями, що робить ці випадки негативними. Негатив є негативним лише тому, що він не позитивний. Заходи попередження та відкликання ігнорують справжні негативи і, отже, їх придатність для класифікації тексту. Застосовуючи класифікацію тексту до даних у моєму дослідженні, точність та відкликання є основою оцінки. Точність також важлива, як це виявлено у багатьох з вищезазначених досліджень.

На закінчення огляд охоплює цілий ряд машинного навчання, а саме алгоритмів навчання з учителем, що використовуються при обробці тексту та оцінці.

Деякі ключові висновки для проекту експерименту:

– support vector machines та random forest – це часто використовувані керовані алгоритми навчання при видобутку тексту;

– n-грами є найбільш корисними для виявлення настроїв, а не є підходящим підходом для цього експерименту, оскільки тон текстових даних опису роботи нейтральний;

– для видобутку тексту у довгій формі, 25-30 слів являли собою оптимальну кількість ключових особливостей для цілей моделювання та забезпечували баланс між точністю та потенційною надмірністю моделі;

– accuracy та recall особливо придатні для оцінки аналізу тексту. Це пов’язано з тим, що при видобуванні тексту негативний (двійковий) результат не є справжнім негативом. Точність та recall – це заходи, орієнтовані на справжні позитивні результати;

– оцінка результатів для багатьох класів може бути перетворена в єдину метрику за допомогою усереднення мікро чи макро. Жоден із цих двох варіантів не вважається більш застосовним, ніж інший.

3 РОЗРОБКА ЕКСПЕРИМЕНТУ ТА МЕТОДОЛОГІЯ

3.1 Вступ

В цьому розділі висвітлено поетапне проведення експерименту. За основу самого експерименту взято дослідження [39], яке покриває більшість потреб для поставлених мною цілей для цієї роботи. Даний експеримент можна умовно розділити на чотири фази. Самі ж умовні фази експерименту зображені на рисунку 3.1.

Розділення на фази:

– для збору даних використовується web-scrapping оголошень про роботу з веб сайту який займається розміщенням вакансій;

– наступним етапом є попередня обробка тексту, який включає перетворення зібраних даних, а саме видалення стоп-слів, спеціальних символів тощо. Потім перетворені дані розділяються на набори даних для навчання та тестування;

– далі іде процес видобутку ознак, в результаті чого формується список ключових слів з використанням частоти термінів як основи для ідентифікації ознак кандидата та вакансії [53];

– наступний крок зосереджений на генерації моделей ознак, отриманих за допомогою алгоритмів навчання з учителем Random Forest (RF) та Support Vector Machines (SVM);.

– крок передбачення або прогнозування являю собою перевірку результатів за допомогою тестових даних, а також частково передбачає перевірку моделі на реальних даних;

– останнім етапом є аналіз отриманих результатів, роботи моделі з тестовими даними, а також частково з випадковими реальними даними для моделі.

Рисунок 3.1 – Загальний вигляд експерименту

3.2 Збір даних

Дані були зібрані за допомогою web-scrapping з веб-сайту *powertofly.com*. Цей веб-сайт займається пошуком талантів по всьому світу, а також розміщує вакансії роботодавців. Цей веб-сайт було обрано, бо він має досить велику кількість оголошень, розділення на категорії по типу роботи, що саме мене і цікавить, а також досить зручне API, яке полегшує scrapping.

Структурно веб сторінка не відрізняється складністю, а основні елементи виділені на рисунку 3.2.

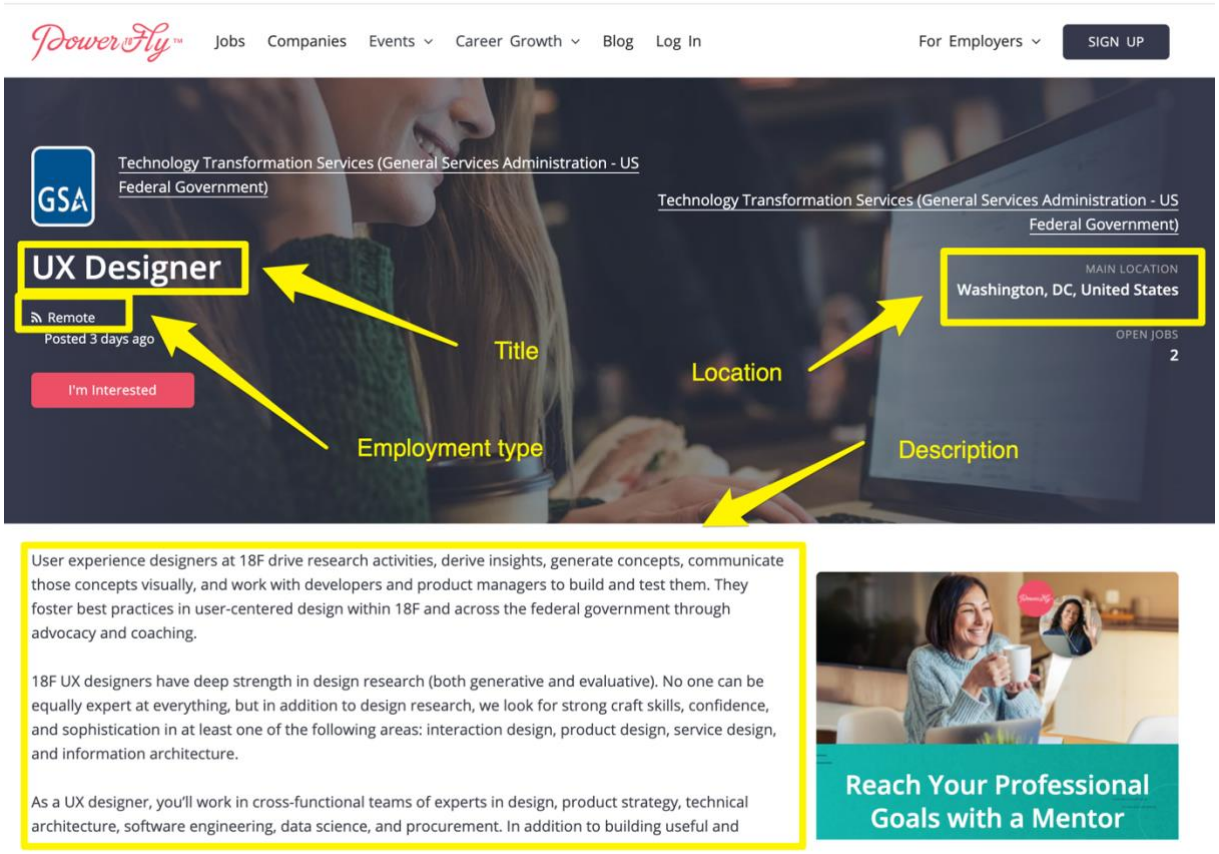

Рисунок 3.2 – Розміщення елементів на веб-сторінці

Так як веб-сайт для рендеру використовує дані з API, що можна побачити на рисунку 3.3 то використовувати інструменти для парсингу не потрібно і можна отримувати необхідні елементи в JSON форматі. Найкращим інструментом для цього буде python бібліотека requests в якій вже є все необхідні інструменти для роботи з JSON [40].

В іншому випадку, наприклад якщо б мова йшла про те що API було захищено за допомогою каптчі або було відсутнє взагалі, то для цього б довелося використовувати парсинг веб-сторінок. Якщо взяти до уваги що я пропоную для розробки мову програмування python, то для таких випадків додатково до python requests бібліотеки чудово б підійшла інша чудова бібліотека, яка називається BeautifulSoup, яка слугує своєрідним інструментом що парсить веб-сторінку і дозволяє працювати з нею звертаючись через зручний інтерфей до HTML елемантів на сторінці, а також вести їх пошук.

Рисунок 3.3 – Відповідь API веб-сайту в JSON форматі

3.3 Попередня обробка тексту

Наступним кроком є попередня обробка тексту. Ця стандартна практика в основному слугує для зменшення розмірності, не впливаючи на ефективність моделі. Вона складається з декількох етапів.

Першим етам в обробці є зведення всіх символи в тексті до малого регістру. Метою тут слугує ціль стандартизувати всі символи, щоб уникнути помилок чутливих до регістру. Деякі функції чутливі до регістру, і способом зменшення ризику помилок є стандартизація всього тексту з малими літерами. Для експерименту і в цілому регістр не має значення для моделі, і тому буде не зайвим застосовувати малі регістри до всіх даних.

Для подальшого процесу потрібно позбутися спеціальних символів. Як і для випадку з регістром цілі цього експерименту не залежать від наявності або відсутності спец символів і не мають особливого значення при прогнозуванні. Тому тут також їх позбуваюся.

Видалення стоп слів це також частина попередньої обробки. Стоп слова – це такі слова як «The», «I», «that» і т.д. Стоп слова не мають жодного прогностичного впливу для моделі, і їх також можна видалити.

Будь-який зайвий пробіл може спричинити неправильне тлумачення слова. У звичайних обставинах слово буде слідувати за єдиним пробілом. Подвійний пробіл може призвести до об’єднання слова із зайвим пробілом. Наприклад, « architect» замість «architect».

Далі потрібно провести стемінг – це процес скорочення слів, одних і тих самих слів але різної форми [49]. Наприклад організувати, організовує, організовуючи, для прикладу дивитися на рисунку 3.5 Рисунок 3.5. Для більшості додатків для видобутку тексту це сімейство варіантів можна розглядати як одне і те ж, корінне слово організувати – репрезентативне для інших. Для свого експерименту я використовую інструментарій NLTK [41, 50, 51, 52]. Також можливо використовувати лематизацію але для мого випадку це не дає ніякого профіту і лише збільшує базу злів. Різницю між стемінгом та лематизацією можна побачити на рисунку 3.4.

Form	Stem	Lemma
Studies	Studi	Study
Studying	Study	Study
beautiful	beauti	beautiful
beautifully	beauti	beautifully

Рисунок 3.4 – Відмінності між стемінгом і лематизацією

Лематизація – це більш комплексний підхід, що базується на визначенні основи слова шляхом лематизації. Першим кроком цього алгоритму є частини мови у реченні, а на другому кроці, до слова застосовуються правила стемінгу відповідно до частини мови. Тобто слова «пальне» та «вітальне» мають проходити через різні ланцюжки правил.

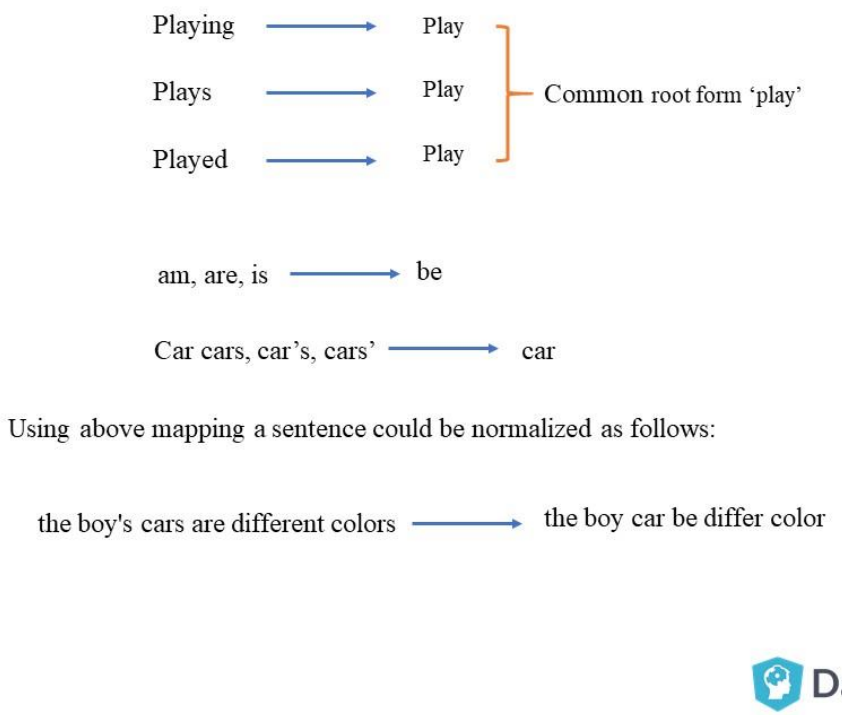

Рисунок 3.5 – Приклад стемінгу (рисунок взято з веб-сайту https://www.datacamp.com/community/tutorials/stemming-lemmatization-python)

Що до набору очікування holdout datasets, то не існує чітких вказівок щодо відповідних рівнів розподілу наборів даних, що спостерігаються. Вони можуть коливатися від 50:50 до 90:10. В даному випадку я буду використовувати 70:30.

Вилучення ознак це процес отримання/вилучення інформативних значень з даних. Частота термінів – загальноприйнятий підхід до вилучення фіч під час аналізу тексту. Це процес вилучення найбільш часто використовуваних термінів у кожному корпусі. Часто вживаний термін може бути інформативною характеристикою та підтримувати процес моделювання. Частота термінів також є відносно простим підходом до виконання порівняно з іншими стратегіями видобутку особливостей, такими як значення важливості, які можуть бути сформовані за допомогою

random forest model. Кінцевою метою є створення Analytics Base Table (ABT), див. таблиця 3.1 для ілюстрації ABT. Це таблиця, що складається з цілі (у цьому експерименті ціллю є числове представлення назви посади), а потім переліку слів, які є ключовими, представляють ціль і полегшують процес прогнозування.

Таблиця 3.1 – Приклад Analytics Base Table

target	**term1**	**term2**	**term3**	**term4**	**term5**	**...**
1	accounting	work	team	accountant	will	..

На рисунку 3.6 зображена візуалізація хмаринки лічильника слів.

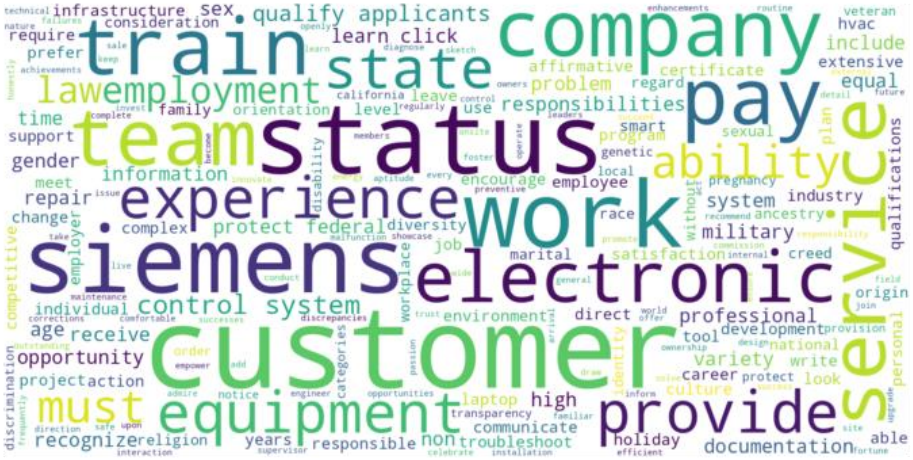

Рисунок 3.6 – Хмаринка лічильник слів

3.4 Алгоритми

3.4.1 Random Forest (випадковий ліс)

Random Forest – це ансамблевий алгоритм, який може виконувати як класифікацію, так і регресію. Ансамбль генерується завантажувальними зразками навчальних даних. Випадковий набір ознак вибирається на етапі розбиття дерева. Прогнози всіх окремих дерев усереднені, щоб отримати

оптимальний прогноз. Цей підхід зменшує вплив сильно прогнозованих змінних і, отже, всіх дерев, які мають високу кореляцію між собою [42].

Випадковий ліс – це керований алгоритм навчання. «Ліс», який він будує, являє собою ансамбль дерев рішень, які зазвичай навчають методом «мішків». Загальна ідея методу мішків полягає в тому, що поєднання моделей навчання підвищує загальний результат.

Простіше кажучи: випадковий ліс будує кілька дерев рішень та об’єднує їх, щоб отримати більш точний та стабільний прогноз.

Однією з великих переваг випадкового лісу є те, що його можна використовувати як для класифікації, так і для регресії, які складають більшість ноксучасних систем машинного навчання. Давайте розглянемо випадковий ліс у класифікації, оскільки класифікація іноді вважається складовою машинного навчання. Візуалізацію роботи алгоритму можна побачити на рисунку 3.7

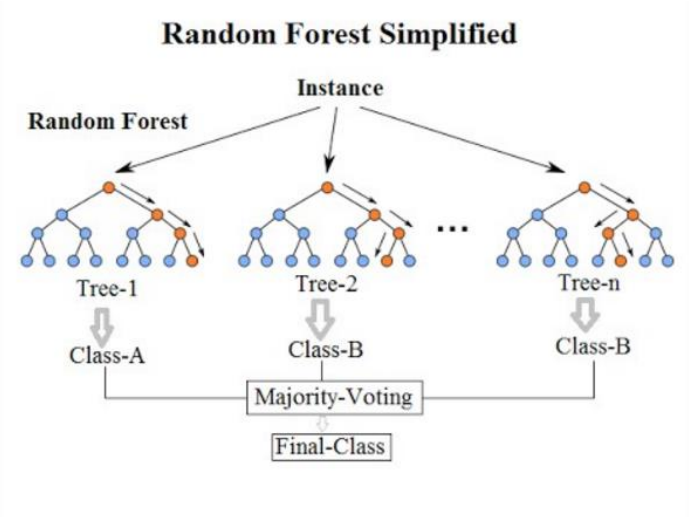

Рисунок 3.7 – Візуалізація алгоритму Random Forest [43]

3.4.2 Support Vector Machines

Support Vector Machines – це алгоритм класифікації та регресії. SVM – це представлення змінних у гіперпросторі. Прогноз робиться на основі визначення межі рішення, яка є гіперплощиною, що відокремлює

один клас від іншого [44]. На Рисунок 3.8 зображена візуалізація для двовимірної ілюстрації SVM.

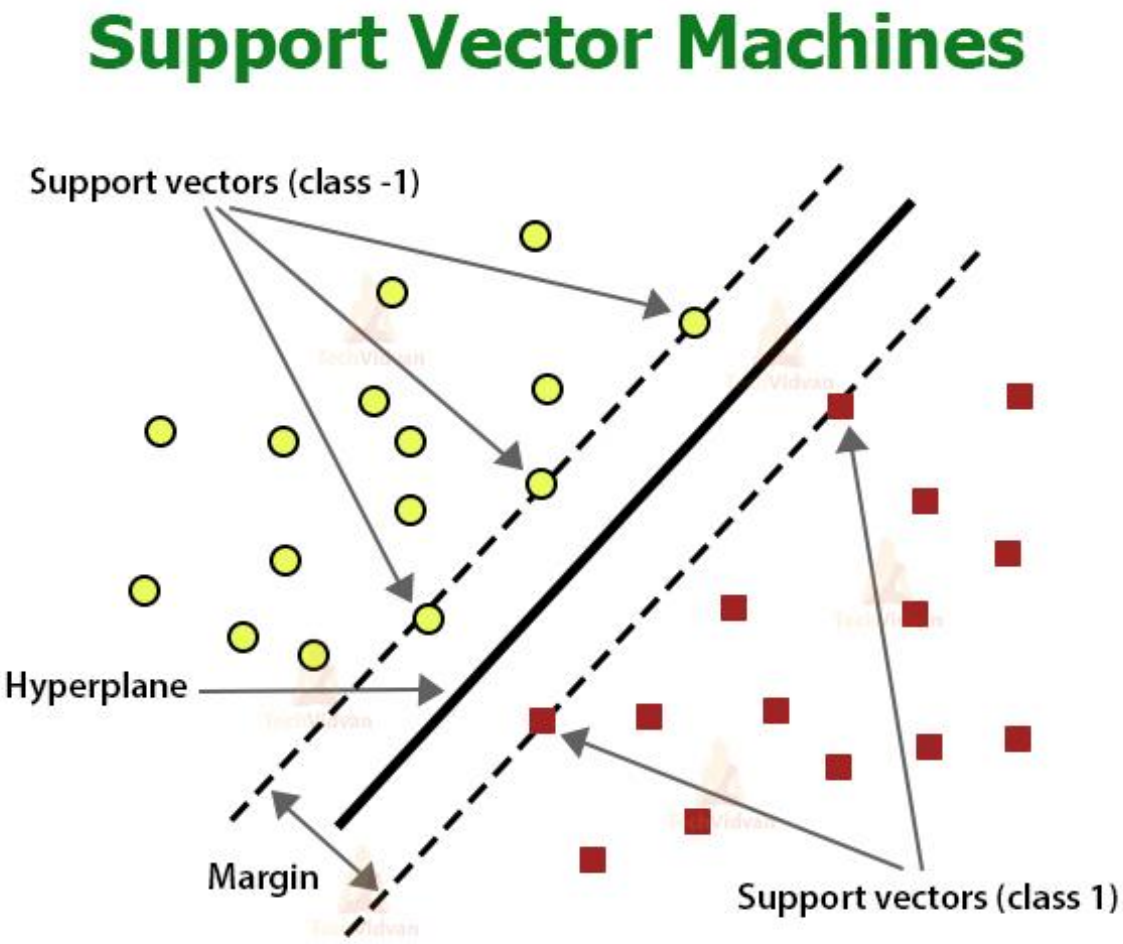

Рисунок 3.8 – Візуалізація алгоритму SVM [45]

3.5 Оцінювання

Порівняння результатів проведено на основі правильності, точності, відкликання та оцінки F відповідно до відповідного дослідження в розділі теоретичних досліджень [38] щодо багатокласової класифікації для класифікації тексту. Визначення правильність (accuracy) зображено в формулі 3.1, визначення точності (precision) зображено в формулі 3.2, Recall та FScore відповідно в формулах 3.3 та 3.4.

Правильність (accuracy)

$$\frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l}, \tag{3.1}$$

Правдиві позитивні результати позначаються як tp_i, правдиві негативні позначаються як tn_i, помилкові негативні позначаються як fn_i, помилкові спрацьовування позначаються як fp_i. l – ідентифікатор класу

Точність (precision)

$$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l}, \tag{3.2}$$

Позитивні спрацювання позначаються tp_i негативні fp_i а l це по класовий ідентифікатор

Recall_M

$$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l}, \tag{3.3}$$

Істинні позитиви позначаються як tp_i, помилкові негативи позначаються як fn_i, l – ідентифікатор класу

Fscore_M

$$\frac{(\beta^2 + 1) Precission_M Recall_M}{\beta^2 Precission_M + Recall_M}. \tag{3.4}$$

Визначення справжнього позитивного, справжнього негативного, хибно позитивного та хибно негативного визначено відповідно до таблиці 3.2.

Таблиця 3.2 – Матриця двійкової сплутаності

Прогнози			
Клас		**Позитивні**	**Негативні**
	Позитивні	TP	FN
	Негативні	FP	TN

Матриця сплутаності допомагає показати прогнозування та відкликання в системі, де відомі значення тестових даних.

4 ВПРОВАДЖЕННЯ

4.1 Система рекомендації

Для початку важливо зрозуміти що основним інструментом що до видачі рекомендацій буде Elasticsearch [46]. Elasticsearch (ES)– пошуковий движок з json rest api, що використовує Lucene і написаний на Java. Опис всіх переваг цього движка є на офіційному сайті. Далі по тексту будемо називати Elasticsearch як ES.

Подібні двигуни використовуються при складному пошуку по базі документів. Наприклад, пошук з урахуванням морфології мови або пошук по geo координатами [47].

В ES вбудований функціонал для машинного навчання, який називається висновок, дозволяє робити прогнози щодо нових даних, використовуючи їх як процесор у конвеєрі передачі, у безперервному перетворенні або як агрегування під час пошуку. Коли нові дані надходять у ваш конвеєр передачі даних або ви виконуєте пошук даних за допомогою агрегування висновків, модель використовується для висновку щодо даних та прогнозування щодо них.

Еластичне контрольоване навчання дозволяє тренувати модель машинного навчання на основі прикладів навчання, які ви надаєте. Потім ви можете використовувати свою модель для прогнозування нових даних. Ця сторінка узагальнює наскрізний робочий процес для навчання, оцінки та розгортання моделі. Він дає огляд на високому рівні кроків, необхідних для виявлення та впровадження рішення за допомогою контрольованого навчання.

Робочий процес навчання під контролем зображений на рисунку 4.1 складається з наступних етапів:

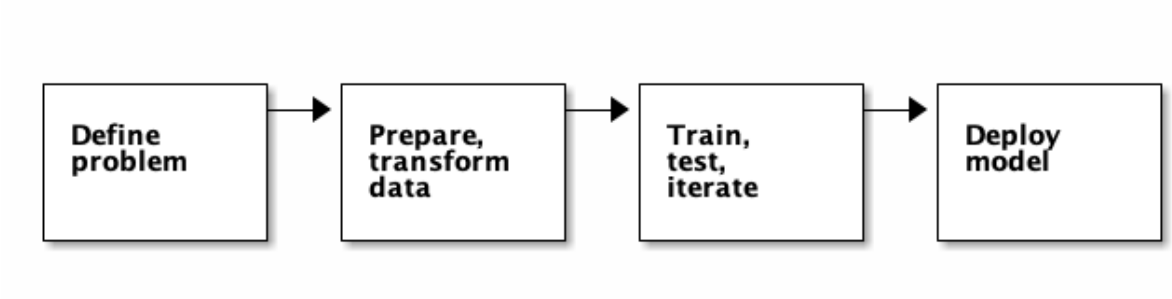

Рисунок 4.1 – Процес навчання під контролем

Це ітеративні етапи, що означає, що після оцінки кожного кроку вам може знадобитися внести корективи, перш ніж рухатися далі.

Аналітика фреймів даних дозволяє виконувати різні аналізи ваших даних та коментувати їх результатами. Роблячи це, він надає додаткові уявлення про дані. Виявлення чужого місця визначає незвичні точки даних у наборі даних. Регресія робить прогнози щодо ваших даних після того, як визначає певні взаємозв'язки між вашими точками даних. Класифікація передбачає клас або категорію даної точки даних у наборі даних. Висновок дозволяє постійно використовувати навчені моделі машинного навчання проти вхідних даних.

Процес залишає вихідний індекс незмінним, він створює новий індекс, який містить копію вихідних даних та анотованих даних. Ви можете нарізати та нарізати дані, розширені з результатами, як це зазвичай роблять з будь-яким іншим набором даних. Докладніше читайте, як це працює.

Ви можете оцінити ефективність аналітики фрейму даних, використовуючи API оцінки фрейму даних аналітики щодо розміченого набору даних. Це допомагає зрозуміти розподіл помилок та визначає точки, де модель аналітики фреймів даних працює добре або менш надійно.

Враховуючи всі можливості цього пошукового двигуна і мій особистий досвід вважаю його найдоцільнішим інструментом для використання в підборі рекомендацій як для кандидатів так і для ректрутерів.

Використовуючи одну ключову особливість, а саме можливість скорінгу (scoring) результатів і можливість застосування своєї функції для скорінгу, а також впровадження спеціального поля яке буде заповнюватися натренованою нейронною мережею можна поліпшити результати рекомендацій для користувачів. Сам по собі скорінг імплементує в собі алгоритми TF, IDF, Field Length Normalization і т.д.

Так як в своїй основі це все таки пошуковий двигун, тому доцільно використовувати структуру яка б дозволяли вносити користувачеві певні уточнення наприклад як бажаний діапазон зарплати і так інше повний список полів представлений на рисунку 4.2.

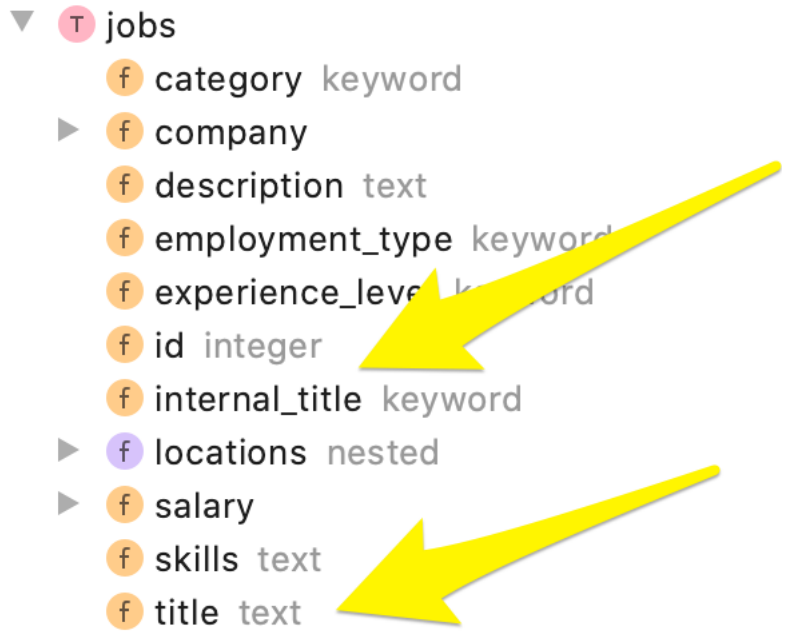

Рисунок 4.2 – Структура сховища в Elasticsearch

Хочу звернути увагу що в якості ключового поля для машинного навчання є поле «internal_title» саме воно відповідає за рекомендації а також має переваги при урахуванні скорінгу і впливає на кінцевий результат.

4.2 Джерело даних

Досліджуючи оголошення про роботу з вищезазначеного веб-сайту в режимі розробника та переглядаючи мережевий трафік можна з легкістю виділити API запити які відповідають за дані про роботу.

Важливою особливістю цих даних є категорія (це така собі узагальнена посада) – саме вона буде ключовим елементом для експерименту з моделювання, а також опис роботи – в якій міститься текст, що використовується для навчання та тестування моделей. Так як самі дані представляють собою добре сформований JSON то це не потребує якихось додаткових маніпуляцій для скорінгу, все цілком робиться за допомогою бібліотеки для python requests без будь яких додаткових маніпуляцій що показано на рисунку 4.3

Самі деталі роботи за виключенням category являють собою текст досить довільної форми та ще містить html елементи тому важливим етам є попередня обробка для тексту.

```
response = requests.request('GET',
'https://powertofly.com/api/v1/jobs/', headers={
    'Content-Type': 'application/json',
```

Рисунок 4.3 – Приклад парсингу веб-сторінки

На кожен запит видається максимум сто результатів. Однак змінюючи URL-адресу і використовуючи pagination та додавши до URL номер сторінки (page=2 і т.д.) можна отримати всі публічно доступні дані. Мене цікавили лише 2 основні це опис посади та категорія за допомогою якою я і збираюся класифікувати посади.

4.3 Попередня обробка тексту

Для основного тексту деталей роботи виконувались наступні операції:

– переведення в нижній регістр;

– видалення спец символів включно з HTML тегами;

– видалено спец слова;

– видалено лишні пробіли;

– стемінгу.

В таблиці 4.1 наведений приклад тексту до попередньої обробки. Як можна побачити це в таблиці, дані являють собою в основному HTML текст, а також в тексті мається дуже багато інформації яка не має ніякого безпосереднього впливу для визначення узагальненої посади і лише розширює корпус даних.

Таблиця 4.1 – Дані до попередньої обробки тексту

Назва посади	**Опис**
Marketing	The Company \n Spot...
Quality Assurance	At R...
Marketing	Facebook's mission is to give people the power...
Finance	Gemeinsam machen wir den Unterschied</div...
Sales	Inside Sales Manager Not even ...
Finance	Microsoft is on a mission to empower ...
Design	Position Description: Off the court,...
Product Management	Facebook's mission is to give people the power...
Product Management	Product Management Consultant <p...
Human Resources	Facebook's mission is to give people the power...
Sales	The Role: Credit Product Specialist<...
Marketing	 The Role: <sp...
Civil Engineering	Description:Provide support for the Fabricatio...
Marketing	Date Posted: 2021-03-29-07:00 Country:...
Quality Assurance	At Bristol Myers Squibb, we are inspi...

4.3.1 Нижній регістр

Так як основним засобом для моделювання я обрав Python, а він по своїй природі чутливий до регістру символів, і щоб уникнути будь яких можливих помилок потрібно все стандартизувати. Таким стандартом будуть літери нижнього регістру. Ця трансформація не впливає на моделі прогнозування, оскільки тут не має потреби ідентифікувати конкретні граматичні конструкції, такі як власні іменники та початок речень, які пишуться з великої літери. Мета полягає у визначенні ключових слів, які мають значення для прогнозування посади, і тому правила граматики не мають значення.

4.3.2 Видалення пунктуації

Подібно до нижнього регістру вище, пунктуація не має особливого значення для цього моделювання. Розділові знаки, такі як крапки, коми тощо, не є особливо інформативними та не впливають назву посади.

4.3.3 Спеціальні символи

Як і для нижнього регістру, спеціальні символи не приносять користі моделюванню, і їх видалення зменшує розмірність основного тексту. Наприклад: «rock» та «rock,»

Ці два слова будуть оцінені як різні слова, однак для цілей цього експерименту з моделювання слово «rock» є найпомітнішим бітом інформації, і кома не приносить значення. Видаливши розділові знаки, кількість розмірів (унікальних слів) зменшилася на один, що сприяє ефективності обробки, і модель буде покращена за рахунок зменшення розмірності.

4.3.4 Стоп-слова

Як згадувалось у розділі 3.3, такі слова, як «the», «is», «that» є стоп-словами. Стоп-слова не мають значення для цілей цієї моделі на приклад: «the project» проти «project»

У цьому прикладі слово «the» не вносить жодного додаткового значення у визначенні посади, важливе слово – «project». Існує також допоміжна вигода, яка призводить до зменшення розмірів, як зазначено вище, зі спеціальними символами.

4.3.5 Пробіли

Відповідно до розділу 3.3, вилучення спеціальних символів та пунктуації найкращою практикою є заміна символу пробілом, а не нульовим рядком.

Наприклад в таблиці 4.2 наведено один з прикладів для заміни пробілів.

Таблиця 4.1 – Приклад заміни пробілів

Оригінал	**Заміна без пробілів**	**Заміна з пробілами**
Accounting/Finance	AccountingFinance	Accounting Finance
completed Technology	completed Technology	completed Technology

У першому прикладі пунктуації, що замінюється нульовим рядком, результатом є нове об’єднане слово. Це мало важить для моделі та не сприяє зменшенню розмірності, оскільки було створено нове унікальне слово. Однак, надаючи заміну одним пробілом, ми отримуємо два різних слова. Ймовірно обидва слова, вже існують або існуватимуть, тому розмірність буде зменшена. Зворотний бік цього – другий приклад, коли заміна на

нульовий рядок дає сприятливий результат двох різних слів, які, швидше за все, зменшать розмірність. Заміна пробілом призводить до подвійного пробілу між двома словами. Рішення полягає в запуску видалення пробілів після запуску заміни на пробіли.

4.3.6 Стемінг

Як уже згадувалося в розділі 3.3, стемінг – це процес скорочення слів шляхом усічення різних форм одного і того ж слова. Візьмемо до прикладу ось це слово organise, organises, organising. В таблиці 4.3 показані дані після попередньої обробки

Таблиця 4.2 – Дані після попередньої обробки

Назва посади	**Опис**
Finance	silicon engineer solutions team ses look exper...
Finance	role us c voice customer voc team part insight...
Product Management	please note visa sponsorship available positio...
Civil Engineering	facebook mission give people power build commu...
Product Management	project management tech refresh incentive lead...
Sales	mission zocdoc tech company begin better healt...
Finance	silicon engineer solutions team ses look exper...
Finance	role us c voice customer voc team part insight...

Продовження таблиці 4.3

Назва посади	**Опис**
Product Management	please note visa sponsorship available positio...
Civil Engineering	facebook mission give people power build commu...
Product Management	project management tech refresh incentive lead...
Sales	mission zocdoc tech company begin better healt...
Data	people data report manager facebook manage tea...
Data	freddie mac important work build better house ...
Civil Engineering	facebook mission give people power build commu...
Civil Engineering	electrical product team electronic systems pro...
DevOps	solution specialist applications development s...
Design	lead way get back american express know right ...
Finance	customer success microsoft aspire help custome...
Sales	call club vision mission value support pillars...

4.3.7 Розбиття даних

Після завантаження і обробки даних в системі налічується трохи більше 86000 записів. Сам набір даних обмежений 15 найпоширенішими назвами посад. Таке обмеження частково впливає на Random Forest, який не підтримував більше 32 прогнозів. На рисунку 4.4 зображена гістограма, що ілюструє кількість посад і в наборі даних.

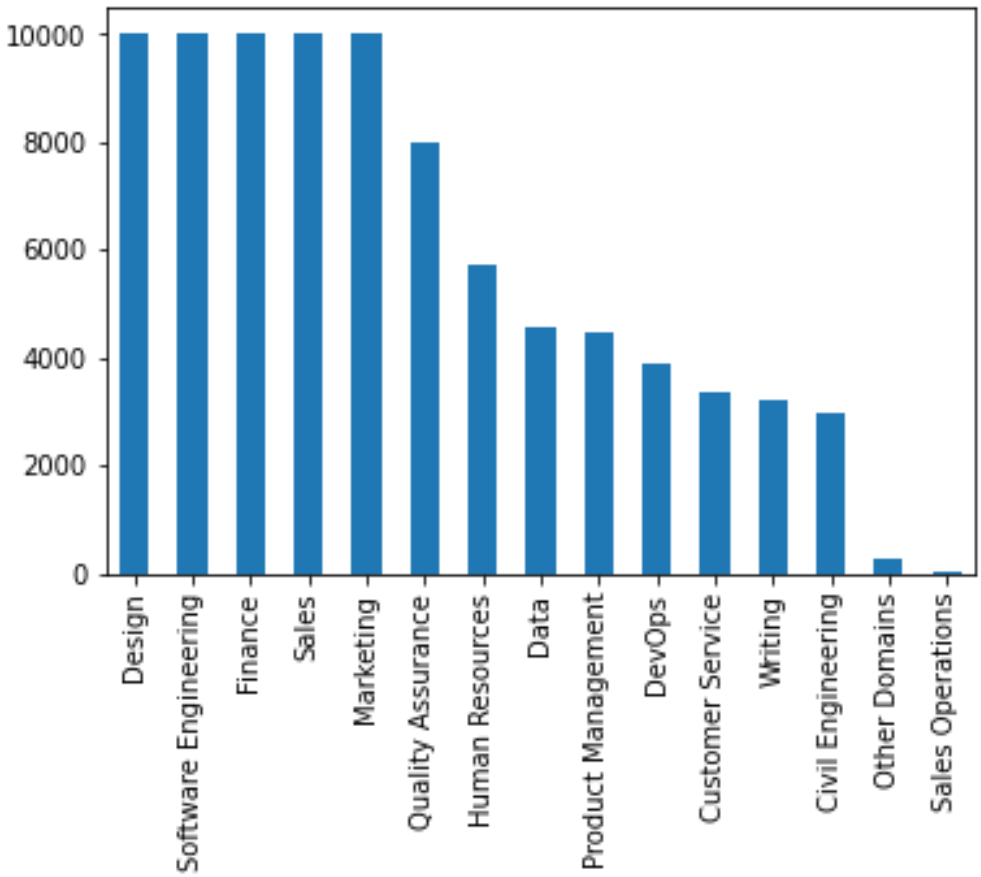

Рисунок 4.4 – Кількість записів про посади

Також, більш детально, кількісне розподілення між посадами можна бачити в таблиці 4.4

Таблиця 4.3 – Кількісне розподілення посад

Заголовок посади	Кількість записів
Sales Operations	37
Other Domains	275
Civil Engineering	2963
Writing	3190
Customer Service	3336
DevOps	3890
Product Management	4475
Data	4560
Human Resources	5733
Quality Assurance	7978
Design	9999
Finance	9999
Marketing	9999
Sales	9999
Software Engineering	9999

Набір даних конвертується в два набори для тестування та тренування. Розбиття встановлено на рівні 70/30, а розбивка стратифікована для кожної посади. Крім цього визначається фактор. Фактор – це спеціальний тип даних, який розпізнає унікальні значення у списку та зберігає їх у фоновому режимі. В таблиці 4.5 показано приклад розбиття даних.

Таблиця 4.4 – Приклад розбиття даних

Посада	Тест	Тренування
Sales Operations	11	26
Other Domains	83	193
Civil Engineering	889	2074
Writing	957	2233
Customer Service	1001	2335
DevOps	1167	2723

Продовження таблиці 4.5

Посада	**Тест**	**Тренування**
Product Management	1343	3133
Data	1368	3192
Human Resources	1720	4013
Quality Assurance	2393	5585
Design	3000	6999
Finance	3000	6999
Marketing	3000	6999
Sales	3000	6999
Software Engineering	3000	6999

4.3.8 Вилучення особливостей

У розділі 3.3 ознаки виділено на основі частоти термінів, тобто визначення найпоширеніших термінів. Але тут є одна особливість що ці терміни потрібно визначати не на цілому корпусі даних а окремо для кожної з посад. На рисунку 4.5 наведено приклад деяких результатів з частими термінами та кількістю випадків.

Частота термінів або інвертована частота документу – це статистичний показник, що використовується для оцінки важливості слів у контексті в контексті документу, що є частиною колекції документів чи корпусу. Вага (значимість) слова пропорційна кількості вживань цього слова у документі, і обернено пропорційна частоті вживання слова у інших документах колекції.

Частота визначається наступною формулою:

$$TF = \frac{n_i}{\sum_k n_k}, \tag{4.1}$$

де n_i є число входжень слова в документ, а в знаменнику – загальна кількість слів в документі.

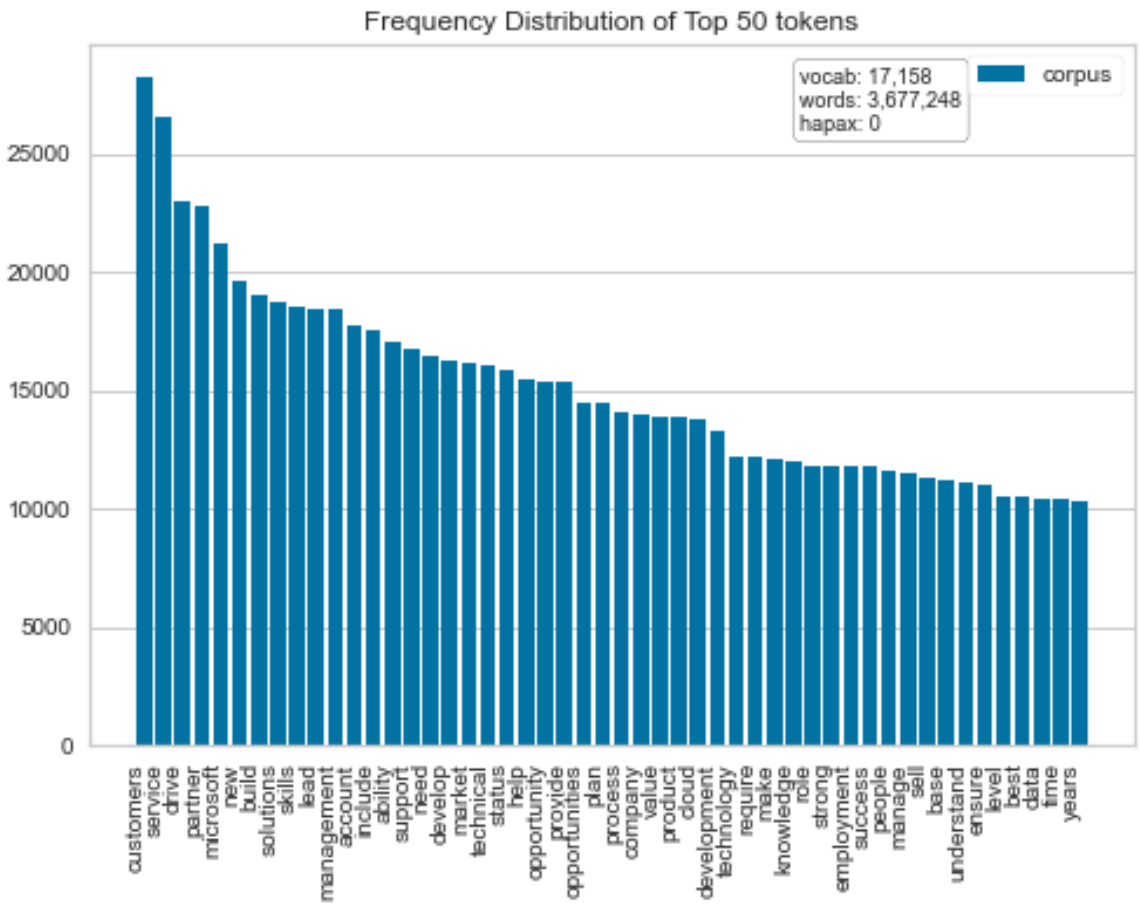

Рисунок 4.5 – Приклад даних з розбиттям по частоті

До цього з даними потрібно провести додаткові перетворення, по-перше, кількість випадків відкидається, оскільки сам термін є тим, що потрібно для моделювання, а не частоти. Список транспонується, щоб дотримуватися стилю ABT, як це обговорювалося в розділі 3.3.

Використовуючи попередній приклад в таблиці 4.6, запис ABT матиме такий вигляд:

Таблиця 4.5 – Базова аналітична таблиця

target	**term1**	**term2**	**term3**	**term4**	**term5**
1	microsoft	dell	will	work	nice

Після створення повної аналітичної таблиці було помічено, що в кожному випадку з’являється слова які не маю ніякого впливу на можливе передбачення: «microsoft, dell, will, work, nice». Ці слова були виключені, оскільки вони не мали жодного впливу на процес моделювання, оскільки були загальними для всіх і, отже, не диференціювались. Тобто більшість цих слів має відношення до опису не самої посади вакансії а відношення до опису компанії яка розміщує цю вакансію. Це на мою думку створює певні проблеми оскільки деякі компанії наприклад як Microsoft можуть мати забагато однотипних вакансій і враховуючи те що таких вакансій може бути більшість то відповіде це може дуже сильно вплинути на результати.

Окрім того, що не є інформативним, загальний для всіх примірників термін підриває алгоритм SVM. SVM працює на основі ідентифікації розділяючої гіперплощини для категоризації елементів. У сценарії із загальним виміром для всіх елементів ця розділова площина не може бути отримана.

4.4 Моделювання

4.4.1 Random Forest

Першою генерованою моделлю є Random Forest. Використовуючи базовий алгоритм без налаштування параметрів, результати можна оцінити в таблиці 4.7

Таблиця 4.6 – Результати роботи алгоритму Random Forest

Метод Оцінки	**Результат**
Правильність (Accuracy)	56.21%
Точність (Precision)	54.56%
FscoreM	49.25%
RecallM	49.03%

У розділі 3.5 було вивчено основні способи оцінювання. А саме визначення фокусується на правильності, точності, $Recall_м$ та $Fscore_м$

Правильність (accuracy)

$$\frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l}, \tag{4.2}$$

Точність (precision)

$$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l}, \tag{4.3}$$

$Recall_м$

$$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l}, \tag{4.4}$$

$Fscore_м$

$$\frac{(\beta^2 + 1)Precission_M Recall_M}{\beta^2 Precission_M + Recall_M}. \tag{4.5}$$

4.4.2 Random Forest тюнінг (тонке налаштування)

Random Forest має деякі налаштування для функції, *class_weight*, *criterion*, *max_depth*, *max_features*, *max_leaf_nodes*. Але для роботи використовувалися базові значення за замовчуванням, що насправді і є оптимальним значення.

4.5 Support Vector Machines

Маєр та інші [48] використовували алгоритму SVM для створення моделей для результатів трьох моделей з використанням таких параметрів: SVM з лінійним ядром, cost value = 1 SVM з поліноміальним ядром, degree value =1, cost =1 SVM з радиальния ядром, cost = 1. Результати наведені в таблиці 4.8

Таблиця 4.7 Результати роботи алгоритму SVM

Метод Оцінки	**Лінійне Ядро**	**Поліноміальне Ядро**	**Радіальне Ядро**
Правильність (Accuracy)	76.18%	11.2%	71.99%
Точність (Precision)	67.28%	6.6%	61.1%
FScore	68.25%	1.2%	62.66%
Recall	69.68%	0.74%	60.21%

Після запуску численних варіантів алгоритмів SVM з альтернативними значеннями вартості та ступеня, параметр збільшувався з кроком 1. Лінійний SVM підтримував свою ефективність, тоді як поліноміальний та радіальний або збігалися, або зменшували перевагу при налаштуванні параметрів. Висновок про те, що лінійний показник був найкращим або абсолютно найкращим, відповідає літературі, розглянутій вище. На цьому етапі поліноміальна та радіальна моделі були вилучені з подальшого аналізу.

Дивлячись на модель, ми отримуємо результати, як показано в таблиці 4.9. Існує незначна зміна в точності порівняно з Random Forest. Міра

F покращилась, а точність знизилась. Модель лінійного ядра SVM на даних дала поліпшення для всіх метрик.

Таблиця 4.8 Результати роботи алгоритму Random Forest та SVM

Метод Оцінки	**SVM Лінійне Ядро**	**Random Forest**
Правильність (Accuracy)	76.18%	55.7%
Точність (Precision)	67.28%	53.76%
FscoreM	68.25%	49.25%
RecallM	69.68%	49.03%

Для перевірки роботи натренованої моделі використаємо випадковий профайл з веб-сайту LinkedIn. Використаємо інформацію яку користувач залишив про себе в описі свого профілю як зображено на рисунку 4.6. В самому профілі ми можемо побачити що користувач ідентифікує себе як «Civil Engineer», а в розділі про себе зазначає наступну інформацію «Experienced Licensed Engineer with a demonstrated history of working in the utilities industry. Skilled in AutoCAD, Foundation Design, Revit, Structural Engineering, and Structural Analysis. Strong engineering professional with a Master's degree focused in Structural Engineering from New Jersey Institute of Technology.» Код для виконання перевірки дивитися в лістингу 4.1. В результати виконання перевірки прогнозування отримуємо результат:

```
Prediction-> ['Civil Engineering']
```

Такий результат підтверджує що модель може працювати не тільки з вакансіями але й з профілями користувачів та їх резюме.

Лістинг 4.1 – Прогнозування посади

```
from common.text_processing import prepare_data
ask_result =
Tfidf_vect.transform([prepare_data('''Experienced Licensed
```

```
Engineer with a demonstrated history of working in the
utilities industry. Skilled in AutoCAD, Foundation Design,
Revit, Structural Engineering, and Structural Analysis.
Strong engineering professional with a Master's degree
focused in Structural Engineering from New Jersey Institute
of Technology.''')])
prediction = SVM.predict(ask_result)
print("Prediction->", Encoder.inverse_transform(prediction))
```

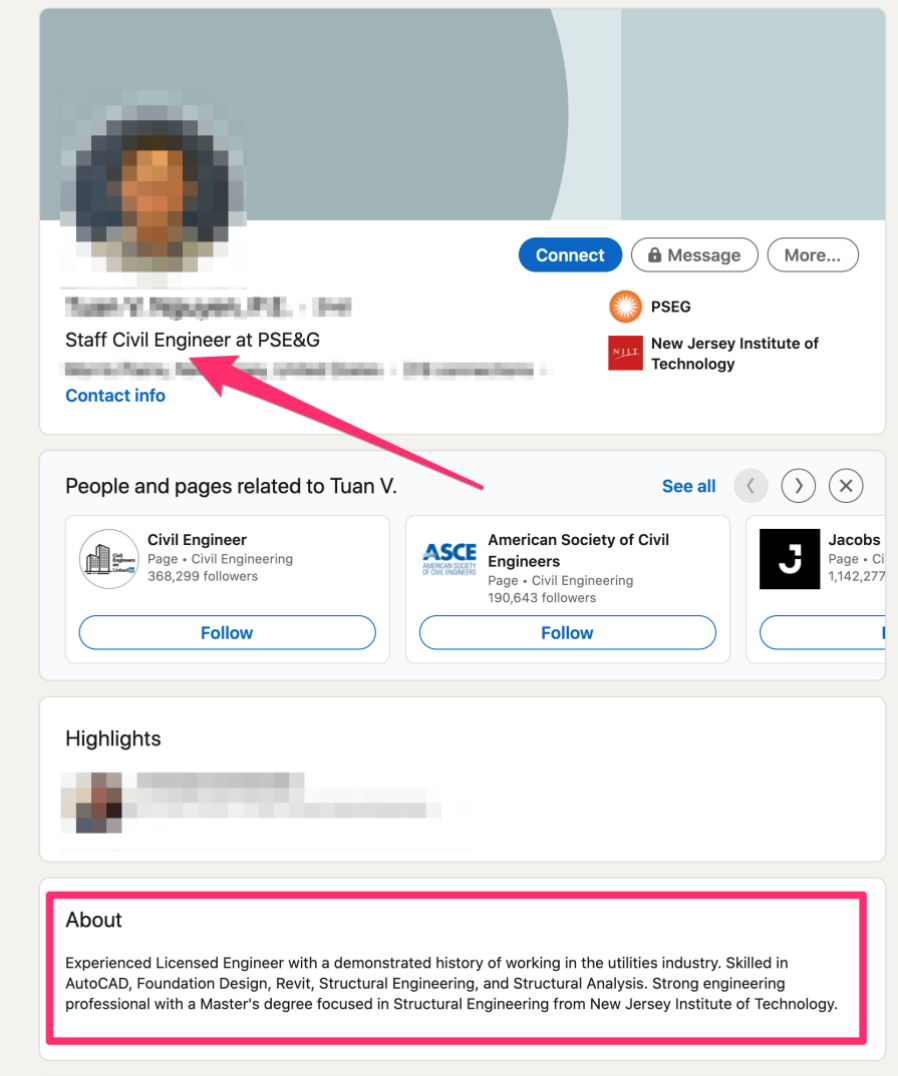

Рисунок 4.6 – Приклад інформації яка береться для визначення посади користувача LinkedIn

Одним і з цікавих моментів дослідження є те, що трапляються вакансії, які мають дуже сплутану назву та опис. Для прикладу на веб-сайті «powertofly.com» можна знайти назву вакансії «Project Manager – Civil Engineer», яка також має ще і дуже сплутаний опис, схожий по деяким із вимог на «software engineer» за деякими вимогами на «design» і ні за якими критеріями не схожий на «project manager». Так як я не являюся спеціалістом з вибору і підбору професій то мені особисто складно зробити якийсь висновок до якої узагальнюючої посади можна віднести вакансію як на прикладі, тому цікаво таку ситуацію буде вирішувати моя натренована модель. Результати прогнозування для такого експерименту наведено в таблиці 4.10

Таблиця 4.9 – Результати перевірки моделі на складній вакансії

Project Manager – Civil Engineer	
Тип експерименту	**Результат прогнозування**
Лише заголовок	Product management
Лише вимоги до посади (qualification, education, skills)	Civil Engineering
Лише опис до посади	Software Engineering
Лише вміння (skills)	Software Engineering

Результати представлені в таблиці 4.10 ще раз доводять що визначення посади це не проста справа навіть для тренованої людини але ще і те що всі особливості в формулюванні дуже різноманітні і різняться від компанії до компанії. Беручи до уваги таку особливість при подальшій класифікації на виробництві потрібно робити прогнози для всього опису вакансії і аж ніяк не її частини.

Продовжуючи вивчати отриману модель SVM, подував матрицю плутанини щоб побачити розбіжності між передбачуваними та фактичними посадами див., рисунок 4.7.

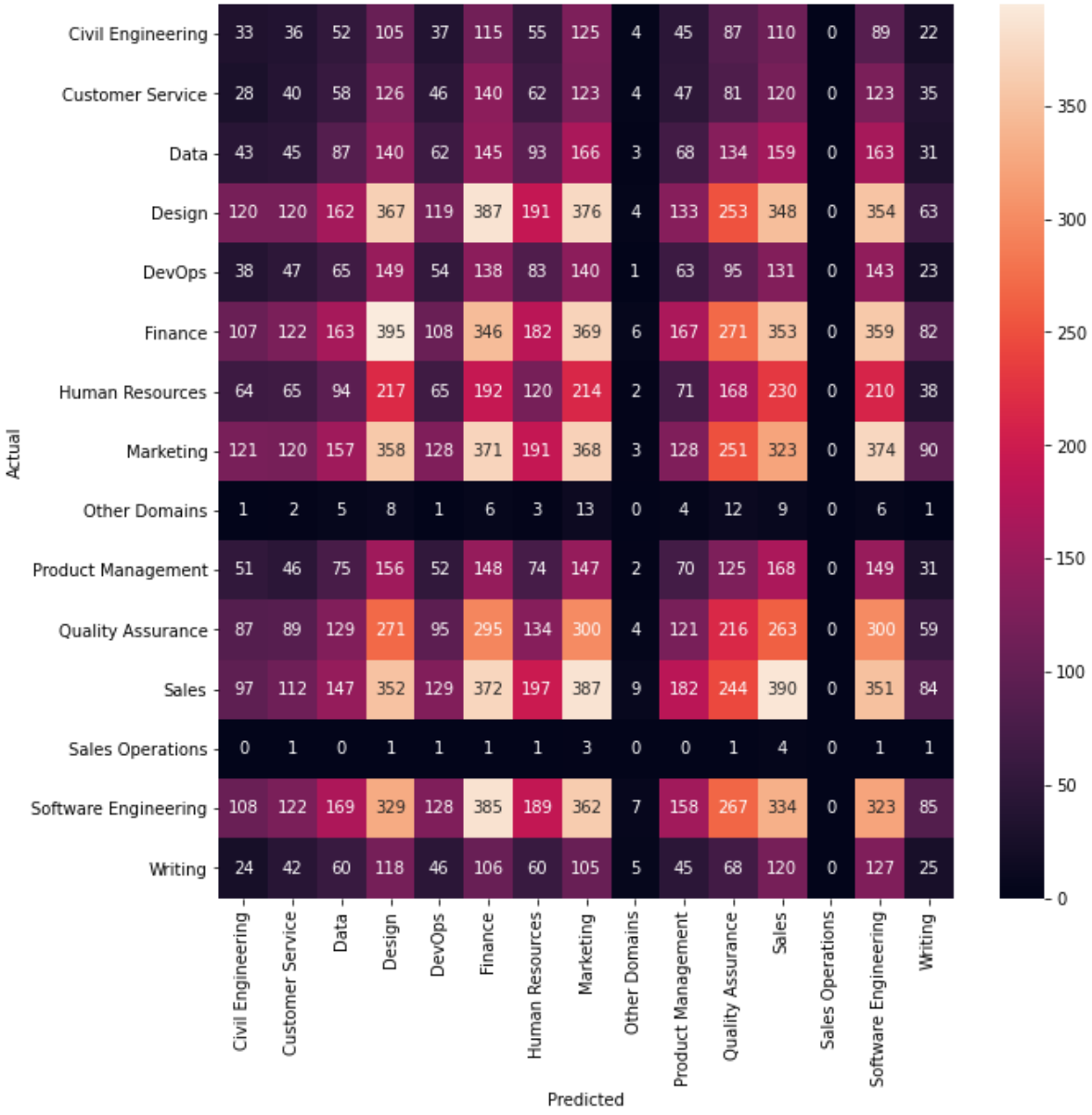

Рисунок 4.7 – Матриця плутанини

Переважна більшість передбачень потрапляє на середину (передбачувана мітка = фактична мітка), де потрібно, щоб вони були там. Однак ми бачимо що існує ціла низка помилкових класифікацій, і може бути цікаво подивитися, чим вони спричинені.

Далі пробую розшифрувати матрицю плутанини за допомогою того ж python, сам код представлено на лістингу 4.2

Лістинг 4.2 – Python код для розшифровки матриці плутанини.

```
from IPython.display import display
dataset['category_id'] = dataset['title'].factorize()[0]
category_id_df = dataset[['title',
'category_id']].drop_duplicates().sort_values('category_id')
category_to_id = dict(category_id_df.values)
id_to_category = dict(category_id_df[['category_id',
'title']].values)

for predicted in category_id_df.category_id:
    for actual in category_id_df.category_id:
        if predicted != actual and conf_mat[actual, predicted] >=
10:
          print("'{}' predicted as '{}' : {}
examples.".format(id_to_category[actual],
id_to_category[predicted], conf_mat[actual, predicted]))
          display(dataset.loc[indices_test[(Test_Y ==
(np.where(Encoder.classes_==id_to_category[actual])[0][0])) &
(predictions_SVM ==
(np.where(Encoder.classes_==id_to_category[predicted])[0][0]))]][
['title', 'description']])
          print('')
```

Проаналізувавши розшифровку більш детально можна зрозуміти що більшість вакансій тісно пов’язаних з людським ресурсом найчастіше викликають похибку прогнозування. На мою думу це пов’язано з тим що більшість із них мають багато спільного в описі між посадами що ускладнює ідентифікацію вакансії. Тобто щоб спробувати уникнути таких проблем потрібно більше детально проаналізувати слова які є спільними між вакансіями та їх позбутися. Для прикладу якщо звернути увагу на вакансії «Software Engineering» то можна побачити що більшість таких вакансій належить компанії Microsoft, і в опис до вакансій додається опис компанії де розповідається яка вона хороша і як вони піклуються про людей. З іншого

боку якщо поглянути на «Writing» то одним із ключових вимог до кандидатів є знання Microsoft Word що і може спонукати видачу хибних результатів.

Результат розшифровки матриці плутанини можна побачити на рисунку 4.8

```
'Software Engineering' predicted as 'Design' : 93 examples.
```

	title	description
16856	Software Engineering	position description believe solutions enginee...
85632	Software Engineering	time inspire future healthcare together siemen...
53510	Software Engineering	position contingent upon contract fund raytheo...
19987	Software Engineering	department candidate part enterprise risk solu...
61821	Software Engineering	description engineer technology e manager sele...
...	...	...
58996	Software Engineering	microsoft dynamics 365 suite easy learn easy u...

Рисунок 4.8 – Розшифровка матриці плутанини

Також аналіз показує що існує дублювання таких термінів, як «управління», «проект», «проекти», і це являється проблемою що розмежувати мітки, щоб створити граничну гіперплощину, достатню для прогнозування. Вивчаючи особливості інших фіч, зустрічаються багато термінів таких як «інженер» або наприклад «людина». Тобто ці терміни є загальними для більшості вакансій інженер-програміст, інженер-будівельник, інженер-тестувальник. А «людина» в описі до вакансій за звичай характеризує відношення компанії до робітників але аж ні як професії пов’язані з підбором персоналу або медициною «Human Resources» та «Human Body». Тому можливо одною із стратегій для поліпшення результатів може бути додавання доменної інформації із посиланням на часті терміни, не включені в оригінальний набір.

5 ОЦІНКА ТА АНАЛІЗ

Як показують результати таблиці 4.7, точність Random Forest становить 55.7%. Така оцінка обумовлена застосування процедур попередньої обробки тексту з метою видалення непотрібних слів. Зокрема, показник FScore становить 49.03%, Precision – 53.76% і Recall – 49.25%. Використовуючи той самий набір даних, але без застосування стемінгу, алгоритм Random Forest має точність 53.8%. Коефіцієнт Fscore становить 48.7%, Precision – 52.3% і Recall – 48.5%.

Результати при застосуванні алгоритму Support Vector Machine на базовому наборі даних мають коефіцієнт точності 76.18% відповідно до таблиці 4.8. FScore становить 69.68%, Precision - 67.28% і Recall 68.25%. Ці показники послідовно повторювались у 3 різних типах ядер (лінійних, поліноміальних, радіальних).

Загальна ефективність узагальнена в таблиці 5.1 нижче, а гістограма на рисунку 5.1.

Таблиця 5.1 – Загальний підсумок моделі

Модель	**Accuracy**	**Precision**	**Recall**	**Fscore**
SVM stemmed	76.18%	67.28%	68.25%	69.68%
SVM non stemmed	72.6%	65.9%	66.1%	68.5%
Random Forest stemmed	55.7%	53.76%	49.25%	49.03%
Random Forest non stemmed	53.8%	52.3%	48.5%	48.7%

Для таблиці 5.1, а також рисунка 5.1 спостерігається, що SVM, як правило, перевершує Random Forest.

Ефективність SVM була досягнута шляхом вибіркового розгляду прогнозів за класами та цільових термінів, які сприяли б формуванню межі гіперплощини.

При дослідженні багато класової класифікації тексту різноманітних корпусів помилки класифікації становлять від 8% до 29% для різних моделей [28].

Найефективнішою моделлю у цьому дослідженні була лінійна модель SVM.

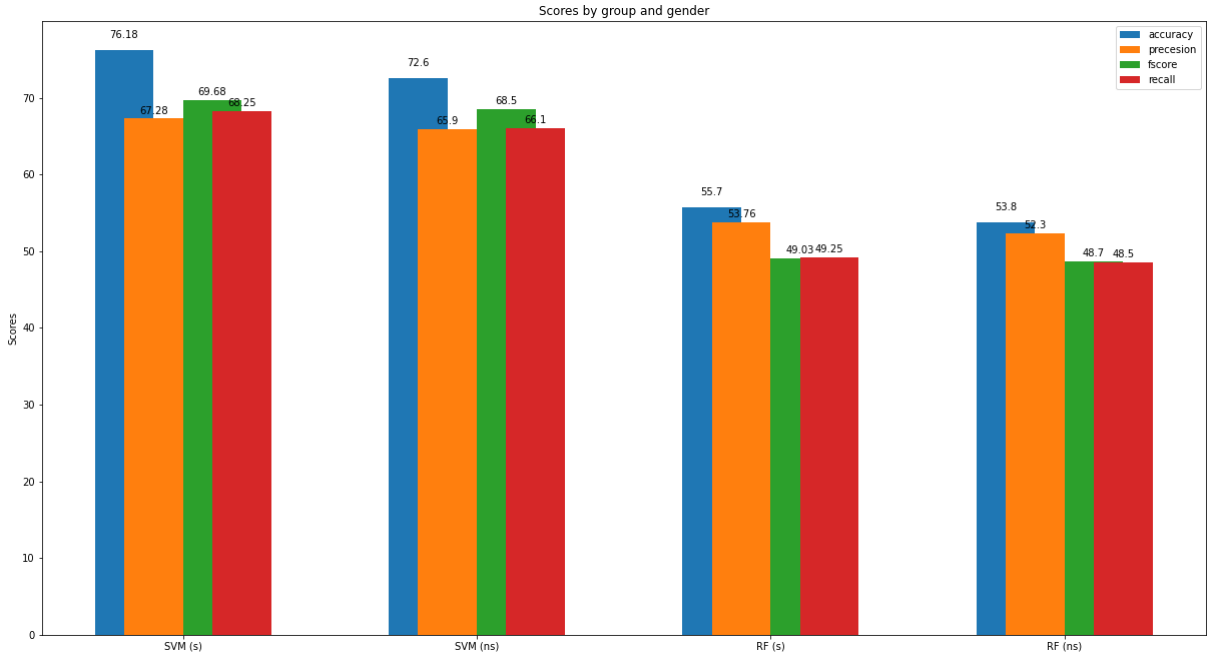

Рисунок 5.1 – Порівняння моделей на графіку

Це демонструє, що модель може бути виготовлена із рівнем точності близько 76%, що ілюструє ступінь взаємозв'язку між назвами посадами.

ВИСНОВКИ

Метою цього дослідницького дослідження було вивчити ступінь передбачуваності узагальненої посади за посадовою інструкцією та резюме кандидати за допомогою контрольованого машинного навчання і дослідити можливість використання моделей в рекомендаційній системі. Цілями, описаними в розділі 1.3, були:

– вивчення зв’язку між посадами та можливостями передбачення або узагальнення назви посади, використовуючи детальний опис вакансії та резюме кандидатів;

– побудувати модель та використати алгоритми навчання з учителем, щоб робити ці прогнози;

– оцінка результатів настройки параметрів обраних алгоритмів та їх можливі альтернативи.

Застосування моделей для рекомендаційної системи

Як зазначено на самому початку, процес класифікації робочих місць, як правило, є суб'єктивним, залежить не тільки від менеджерів по підбору персоналу, кандидатів, їхньої ідентифікації як особистості та бажання виразити себе на робочому поприщі, але також і від конкретної компанії, її внутрішніх політик та стандартів. Враховуючи ту особливість що більшість ATS є автоматизованими системами на рівні з рекомендаційними система, це потребую відповідного підходу до автоматизації класифікації посад і відповідно до цієї класифікації надання необхідних рекомендацій. Що в свою чергу ускладнюється різноманіттям усього вище сказаного. Це є дуже актуальним для компаній які займаються агрегацією вакансій та кандидатів для підбору та пошуку як вакансій так і кандидатів відповідно.

В цілому визначення посади має велике значення для всіх зацікавлених сторін так як це безпосередньо впливає на кваліфікацію, заробітну плату та навіть мотивацію людини, а ще це допомагає визначити

де саме людина перебуває в організаційній структурі і яка її відповідальність.

В цій роботі представлена проблема та можливість у використанні машинного навчання для впровадження систематичного підходу, заснованого на алгоритмізації, що застосовується до реальних даних про роботу і слугує вирішення проблеми класифікації посад.

Основним джерелом даних для цього дослідження був веб-сайт «powertofly.com». Дані з даного веб-сайту збиралися шляхом парсингу за допомого додаткових бібліотек для мови програмування python. Для самих даних, відповідно, були застосовані стандартні методи обробки тексту. Попередньо перед обробкою, щоб видалені такі елементи, як пунктуація, пробіли, спеціальні символи, перетворені словосполучення, щоб зменшити розмірність.

Одним із важливих моментів під час обробки даних було помічено наявність паразитного тексту, який не має ніякого відношення до самої посади але досить часто зустрічається і впливає на частоту потрапляння певних термінів в даних. Одним із таких паразитних текстів є опис компанії. З одної сторони це ніби і не проблема, але наприклад якщо йде мова про якусь велику компанію в якої може бути тисячі вакансій Software Engineer, а для інших компаній посада Software Engineer зустрічається рідко або і взагалі відсутня то тут цілковита перевага по самим даним на боці великої компанії і відповідно в частотному розподілі буде наявність слів з опису компанії. Тому важливим моментом підготовки даних для навчання моделей є вилучення такого паразитного тексту або ж відповідне розподілення в рівності вакансій між компаніями.

Сама модель була створена з використанням частоти термінів для ідентифікації ключових слів, які потенційно можуть бути інформативними для узагальнення посади. Прогнози були зроблені за допомогою алгоритмів SVM та Random Forest на моделі.

Альтернативні прогнози були зроблені за допомогою налаштування параметрів для алгоритмів. Для алгоритму SVM зміненими параметрами були типи ядра (Лінійний, Поліноміальний) та пов'язані з ними параметри. Для Random Forest функція генерування оптимальної моделі шляхом зміни розмірів вибірки також не дала кращої моделі порівняно з базовою версією за замовчуванням.

Найефективнішою моделлю була лінійна модель SVM з вибірково взятими зі списку найбільш часто використовуваними термінами та цільовими показниками, передбачених для класу прогнозів.

Огляд інших досліджень проведений у цій роботі виявився досить плідним і цікавим в плані подальшого розвитку рекомендаційних систем. Одним цікавим моментів такого навчання є досліджень прогнозування твоєї наступної посади базуючись на твоєму попередньому досвіді. Як на мене це може надати додаткові переваги для таких рекомендаційних платформ, наприклад коли буде прогнозовано твою можливу наступну посаду і надано рекомендацій що до її досягнення і наприклад буде порекомендовано якісь навчальні курси. Тобто велику кількість вже наявних досліджень дає можливість для подальшого перспективного розвитку рекомендаційних платформ

Ще мною було розглянуто ряд відповідних досліджень, які досліджували застосування алгоритмів машинного навчання до інших Інтернет-даних (таких як електронна пошта зі спамом та аналіз оглядів фільмів). Що власне і спонукало мене на використання до прикладу алгоритмів машинного навчання SVM та Random Forest.

Зі свого дослідження я можу зробити висновок про те, що SVM є більш ефективним. В моєму дослідженні була отримана модель даних із достатньою силою прогнозування 76% показника точності на моделі, що базується на SVM, і застосована методологія може служити орієнтиром для автоматизованих прогнозів.

Моє та інші дослідження в цьому напрямку показують перспективність застосування ML для класифікації посад, їх узагальнення і в цілому можуть повністю автоматизувати цей процес.

Зважаючи на всі проблеми я б хотів приділити більше уваги проблемі дата майнингу, а саме отриманню більш конкретних термінів які відносяться до посади. Ще б було цікаво отримати table structure і вже потім з цих табличних даних отримати короткий опис посади або резюме користувача і вже потім до нього застосувати ML алгоритми. Відповідно потрібно застосовувати більш досконалі методи обробки тексту для NLP, та застосовувати більш складні методи машинного навчання наприклад такі як CNN.

Для своєї роботи я бачу перспективу застосування не тільки в рекомендаційній системі, а також наприклад для маркетингу для таргетування курсів навчання, підвищення кваліфікації тощо.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Cable, D., Grant, A., & Berg, J., "What's in a Job Title? Business Strategy Review, 24(4), 74–74," 2013. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=92727356&site=ehost-live doi: 10.1111/ j.1467-8616.2013.01007.x.

2. F. Giancola, «Inflated Job Titles: When and How HR Bends the Rules. Employee Benefit Plan Review , 68 (11), 30–32,» May 2014. [Онлайновий]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=95847456&site=ehost-live.

3. Grant, A. M., Berg, J. M., & Cable, D. M., «Job Titles as Identity Badges: How Self-Reflective Titles Can Reduce Emotional Exhaus- tion. Academy of Management Journal , 57 (4), 1201–1225,» August 2014. [Онлайновий]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=97342265&site=ehost-live doi: 10.5465/amj.2012.0338.

4. S. P. Robbins, Organizational behavior, Delhi: Pearson Education India, 2001.

5. J. L. Holland, Making vocational choices: A theory of careers., Upper Saddle River, NJ: Prentice Hall, 1973.

6. T. Xu, H. Zhu, C. Zhu, P. Li та H. Xiong, Measuring the Popularity of Job Skills in Recruitment Market: A Multi-Criteria Approach., 2018.

7. J. Malinowski, T. Keim, O. Wendt та T. Weitzel, «Matching people and jobs: A bilateral recommendation approach,» в *39th Annual Hawaii International Conference on System Sciences*, 2006.

8. M. Diaby, E. Viennet та T. Launay, «Toward the next generation of recruitment tools: an online social network-based job recommender system,» в *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2013.

9. Y. Lu, S. E. Helou та D. Gillet, «A recommender system for job seeking and recruiting website,» в *22nd International Conference on World Wide Web*, 2013.

10. W. Hong, S. Zheng та H. Wang, «Dynamic user profile-based job recommender system. In Computer Science & Education (ICCSE),» в *2013 8th International Conference*, 2013.

11. Y. Zhang, C. Yang та Z. Niu, «A Research of Job Recommendation System Based on Collaborative Filtering,» в *Seventh International Symposium on Computational Intelligence and Design*, 2015.

12. I. Paparrizos, B. B. Cambazoglu та A. Gionis, «Machine learned job recommendation,» в *Fifth ACM Conference on Recommender Systems*, 2011.

13. X. Zhang, Y. Zhou, Y. Ma, B.-C. Chen, L. Zhang та D. Agarwal, «GLMix: Generalized Linear Mixed Models For Large-Scale Response Prediction,» в *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

14. J. Li, D. Arya, V. Ha-Thuc та S. Sinha, «How to Get Them a Dream Job?: Entity-Aware Features for Personalized Job Search Ranking,» в *22th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016.

15. Y. Cheng, Y. Xie, Z. Chen, A. Agrawal, A. Choudhary та S. Guo, «Jobminer: A real-time system for mining job-related patterns from social media,» в *19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013.

16. H. Lin, H. Zhu, Y. Zuo, C. Zhu, J. Wu та H. Xiong, «Collaborative Company Profiling: Insights from an Employee’s Perspective,» в *AAAI*, 2017.

17. D. Shen, H. Zhu, C. Zhu, T. Xu, C. Ma та H. Xiong, A Joint Learning Approach to Intelligent Job Interview Assessment., 2018.

18. N. Kalchbrenner, E. Grefenstette, P. Blunsom, D. Kartsaklis, N. Kalchbrenner, M. Sadrzadeh, N. Kalchbrenner, P. Blunsom, N. Kalchbrenner та

P. Blunsom, «A Convolutional Neural Network for Modelling Sentences,» в *52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

19. Y. Kim, «Convolutional neural networks for sentence classification,» в *2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

20. I. Sutskever, O. Vinyals та Q. V. Le, Sequence to sequence learning with neural networks, 2014.

21. O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever та G. Hinton, Grammar as a foreign language, 2015.

22. K. Cho, B. v. Merriënboer, D. Bahdanau та Y. Bengio, «Syntax, Semantics and Structure in Statistical Translation,» в *On the Properties of Neural Machine Translation: Encoder–Decoder Approaches,* 2014.

23. J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz та J. Makhoul, «4. Fast and Robust Neural Network Joint Models for Statistical Machine Translation,» в *52th Annual Meeting of the Association for Computational Linguistics*, 2014.

24. S. Lauly, A. Boulanger та H. Larochelle, «Learning multilingual word representations using a bag-of-words autoencoder,» *arXiv,* 2014.

25. K. M. Hermann та P. Blunsom, «Multilingual Distributed Representations without Word Alignment,» в *ICLR,* 2014.

26. Kosala, R., & Blockeel, H., «Web Mining Research: A Survey. SIGKDD Explor. Newsl., 2(1), 1–15,» June 2000. [Онлайновий]. Available: http://doi.acm.org/10 .1145/360402.360406 doi: 10.1145/360402.360406 .

27. S. A. F. &. H. T. Guo, «RsuMatcher A personalized rsum-job matching system. Expert Systems with Applications , 60, 169–182,» October 2016. [Онлайновий]. Available: http://www.sciencedirect.com/science/article/pii/S0957417416301798 doi: 10.1016/j.eswa.2016.04.013 .

28. Nigam, K., Lafferty, J., & McCallum, A., Using maximum entropy for text classification, т. I, 1999, p. 61–67.

29. Cavnar, W. B., & Trenkle, J. M., N-gram-based text categorization. Ann Arbor MI, 1994, pp. 48113(2), 161–175.

30. Tripathy, A., Agrawal, A., & Rath, S. K., «Classification of sentiment reviews using n-gram machine learning approach. Expert Systems with Applications, 57, 117–126,» September 2016. [Онлайновий]. Available: http://www.sciencedirect.com/ science/article/pii/S095741741630118X doi: 10.1016/j.eswa.2016.03.028.

31. Bermingham, A., & Smeaton, A. F., Classifying Sentiment in Microblogs: Is Brevity an Advantage? In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, New York, NY, USA: (pp. 1833–1836)ACM. Retrieved 2017-02-15, from http://doi.acm.org/10.1145/1871437 .1871741 doi: 10.1145/1871437.1871741, 2010.

32. Xiu-li, P., Yu-qiang, F., & Wei, J., «A Chinese Anti-Spam Filter Approach Based on Support Vector Machine.,» 2007.

33. Sompras, G., & Lalitrojwong, P., «Extracting product features and opinions from product reviews using dependency analysis. In (pp. 2358– 2362). IEEE,» August 2010. [Онлайновий]. Available: http://ieeexplore.ieee.org/document/ 5569865/ doi: 10.1109/FSKD.2010.5569865.

34. Wang, S., & Manning, C. D., «Baselines and Bigrams: Simple, Good Sentiment and Topic Classification.,» 2012. [Онлайновий]. Available: http://dl.acm.org/citation.cfm?id=2390665.2390688.

35. Koprinska, I., Poon, J., Clark, J., & Chan, J., «Learning to classify e-mail. Information Sciences, 177(10), 2167–2187,» May 2007. [Онлайновий]. Available: http://www .sciencedirect.com/science/article/pii/S0020025506003707 doi: 10.1016/ j.ins.2006.12.005.

36. Noh, H., Jo, Y., & Lee, S., «Keyword selection and processing strategy for applying text mining to patent analysis - ScienceDirect,» [Онлайновий]. Available: http://0-www.sciencedirect.com.ditlib.dit.ie/science/article/pii/ S0957417415000652?via%3Dihub.

37. Onan, A., Korukolu, S., & Bulut, H., «Ensemble of keyword extraction methods and classifiers in text classification. Expert Systems with Applications, 57, 232–247,» September 2016. [Онлайновий]. Available: http://www.sciencedirect.com/ science/article/pii/S0957417416301464 doi: 10.1016/j.eswa.2016.03.045.

38. Sokolova, M., & Lapalme, G., «A systematic analysis of performance measures for classification tasks. Information Processing & Management , 45 (4), 427–437,» July 2009. [Онлайновий]. Available: http://www.sciencedirect.com/science/ article/pii/S0306457309000259 doi: 10.1016/j.ipm.2009.03.002 .

39. J. Lynch, An Analysis of Predicting Job Titles Using Job Descriptions, Masters Dissertation, Technological University Dublin, 2017.

40. «Requests: HTTP for Humans,» [Онлайновий]. Available: https://docs.python-requests.org/en/master/.

41. «Natural Language Toolkit,» [Онлайновий]. Available: https://www.nltk.org/.

42. L. Breiman, "Random forests. MACHINE LEARNING," pp. 45 (1), 5–32, October 2001.

43. «Random Forest,» [Онлайновий]. Available: https://en.wikipedia.org/wiki/Random_forest.

44. Cortes, C., & Vapnik, V., "Support Vector Networks. MACHINE LEARNING, 20(3), 273–297," 1995. [Online]. Available: https://link.springer.com/article/10.1023/ A:1022627411411.

45. «SVM,» [Онлайновий]. Available: https://techvidvan.com/tutorials/svm-in-r/.

46. «Elasticsearch,» [Онлайновий]. Available: https://www.elastic.co/.

47. mkuzmin, «Основы Elasticsearch,» [Онлайновий]. Available: https://habr.com/ru/post/280488/.

48. Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., «Misc Functions of the Department of Statistics, Probability Theory Group,»

February 2017. [Онлайновий]. Available: https://cran.r-project.org/web/packages/e1071/ index.html .

49. M. Bouchet-Valat, "SnowballC: Snowball stemmers based on the C libstemmer UTF-8 library.," August 2014. [Online]. Available: https://cran.r-project .org/web/packages/SnowballC/index.html.

50. V. Breni, "Search online: Evidence from acquisition of information on on- line job boards and resume banks. Journal of Economic Psychology, 42, 112–125," June 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S0167487014000051 10.1016/j.joep.2014.02.003.

51. M. Porter., «An algorithm for suffix stripping. Program , 40 (3), 211–218,» July 2006. [Онлайновий]. Available: http://0-www.emeraldinsight.com.ditlib.dit.ie/ doi/full/10.1108/00330330610681286 doi: 10.1108/00330330610681286.

52. Wickham, H., & RStudio., «Easily Harvest (Scrape) Web Pages,» June 2016. [Онлайновий]. Available: https://cran.r-project.org/web/packages/rvest/ index.html.

53. Q. Chuan, Z. Hengshu, X. Tong, Z. Chen, L. Jiang, E. Chen та H. Xiong, Enhancing Person-Job Fit for Talent Recruitment: An Ability-aware Neural Network Approach., 2018.

54. H. Xu, Z. Yu, J. Yang, H. Xiong та H. Zhu, «Talent Circle Detection in Job Transition Networks,» в *22th ACM SIGKDD international conference on Knowledge discovery and data mining.*, 2016.

55. Ermolayev, V., Akerkar, R., Terziyan, V., & Cochez M. (2013). Towards Evolving Knowledge Ecosystems for Big Data Understanding. In: R. Akerkar (Ed.),Big Data Computing (pp. 3-56). Boca Raton, FL: Taylor & Francis. doi:10.1201/b16014-3