6CCS3PRJ Final Year BSPR

Final Project Report

Author: Khushi Jaiswal

Supervisor: Ievgeniia Kuzminykh

Student ID: 21008516

April 1, 2024

Abstract

As Artificial Intelligence (AI) changes how businesses work, there's a growing need for people who can work in this sector. This report looks into how well university AI courses prepare students for jobs in the real world. We will try to find out where the differences are between what universities teach and what company recruiters want. We will do this by looking closely at what is taught in courses and studying job advertisements from websites like LinkedIn. By using custom data scraping tools to gather information from job advertisements, frequency analysis and classifier-aided sorting techniques this report will show exactly what skills companies are looking for. Similar methodology will be applied on university AI courses to see where they might be missing the mark on what is needed in the job market. By looking at a variety of sources like job adverts, university courses, and other reports available, this report will give a complete picture of the gap between what is being taught and what is actually needed in the AI domain. Main findings will be shown using graphs and pictures to make it easier to understand.

Originality Avowal

I verify that I am the sole author of this report, except where explicitly stated to the contrary. I grant the right to King's College London to make paper and electronic copies of the submitted work for purposes of marking, plagiarism detection and archival, and to upload a copy of the work to Turnitin or another trusted plagiarism detection service. I confirm this report does not exceed 25,000 words.

Khushi Jaiswal

April 1, 2024

Acknowledgements

I am grateful to Dr. Ievgeniia Kuzminykh for her invaluable guidance and support in crafting this report. Her expertise and feedback significantly enhanced its quality. I appreciate her dedication and mentorship.

Contents

Chapter 1

Introduction (Background & Context)

1.1 Context

In today's fast-changing world of artificial intelligence, industries are transforming rapidly due to new technologies. As AI becomes more crucial in businesses, there is a growing need for skilled professionals. This demand has led to the (very recent) introduction of AI courses in universities. However, the skills taught in these courses do not always match industry needs. This study looks into this issue by exploring current AI education, industry views on the skills gap, and existing literature. The goal is to contribute useful insights and address the gap between what universities teach and what the industry needs.

Many previous studies have looked at similar topics to the one discussed here. The current study uses insights from a detailed review of existing literature, which is explained in the literature review section. Reflecting on common patterns in these previous works, has improved the overall quality of the paper. At the same time, areas where more information could be added have been identified to make it easier for readers to understand. This paper is a combination of successful elements from past works, with additional details to make it more accurate.

The outcomes of this study are expected to contribute to the refinement of AI programs taught at universities, aligning them with the AI industry-required standards by finding discrepancies in the current skills of these two groups.

1.2 Aims and Objectives

The aim of the project is to understand the skill and expertise gap between higher education and the labour market. To achieve this aim skills that Artificial Intelligence degrees equip students with will be mapped to the skills that UK AI-based job descriptions require.
More formally, to achieve the above aim, this report is going to undertake the following subtasks:

- Identify skills taught by universities
  - Explore the curriculum offerings of AI courses in universities to understand the modules being taught. Investigate the skills taught by these modules and assess the imbalance in the distribution of these acquired skills among university graduates.
- Identify skills required by employers in the AI sector
  - Develop a web scraping tool designed to extract the full spectrum of skills mentioned in job posting advertisements. Conduct an in-depth analysis of the skills and competencies that employers seek within the AI industry.
- Mapping of skills / Perform the Analysis of data
  - Establish a mapping between the skills instructed in university AI modules and those demanded by job postings. Evaluate the readiness of AI graduates to seamlessly transition into the workforce directly from higher education. Identify and analyse gaps in practical skills, problem-solving abilities, and familiarity with industry tools and practices among these graduates.

1.3 Research Questions

- What skills are required by UK AI industry recruiters?
- What skills are gained by students who have completed an AI degree in the UK?
- What kind of mapping is there between the two datasets; skills acquired by AI degrees and skills AI jobs require?
- Can we identify a skills gap between AI degrees and AI uni courses?

1.4 Report Structure

Up till now, an introduction and aims of this report have been given. This paper will next analyse existing similar literature, some aspects of which will be further enhanced in this report. Next, we will look at what design decisions were made in chronological order along with justification for each design choice made. Next, requirements for the project will be set. After that, this report will give an in-depth description of each of the software components implemented for our research goals. Finally, an evaluation of the findings along with any conclusions drawn from our analysis will be stated. And to finish off, this report talks about legal, social, ethical, and professional considerations along with how future works can extend this current report.

Chapter 2

Literature review

Numerous research papers with comparable objectives have been published in the same domain as this study. A thorough examination of these works has significantly contributed to the development of this paper. Many of these papers offer credible content and methodologies that have proven immensely beneficial to our research. However, it should be acknowledged that various gaps have been identified in the existing literature. This paper aims to fill gaps and offer insights with the current knowledge.

2.1 Similar Papers

A paper that really stood out was a paper published by University of Kansas, USA, titled "An investigation of skill requirements in artificial intelligence and machine learning job advertisements" [26]. The paper is co-authored by Amit Verma and Kamal Lamsal.

Its aims are similar to the aims of this paper and are summarised in a compact manner in the introduction that is provided. The paper is started straight away by referencing well known papers including LinkedIn's 2020 Emerging Job Report and Gartner's study on AI/ML disruption across industries; both of which point towards a growing skills gap in the artificial intelligence (AI) and Machine Learning (ML) field. The paper states that despite high demand, there is a shortage of skilled talent. The authors highlight the importance of training potential recruits early on and adapting university curricula to meet evolving skill requirements.

The methodology employed in this report involves a rigorous process designed to scrape, map and analyse the skill requirements for AI and ML. The source for the job advertisements used by the report was "indeed.com". The report has employed both manual approaches and automated

approaches to collecting the relevant job adverts. Python and its libraries were utilised for web scraping, enabling the extraction of job titles and descriptions. Content analysis techniques were applied to extract relevant information from job descriptions, the report's technique was to focus on keywords that usually are associated with skills required in jobs. The report provides an in-depth description for the readers and talks about the extraction of unigrams, bigrams, and trigrams from the raw data to get a better understanding of the skills required. The software used also collected other relevant statistics i.e. the relative frequency of each skill across job postings, this is a very good approach and enables the identification of skill trends and patterns.

A unique and rather ambitious aspect of this report was the use of a classification framework which was used to categorise skills into distinct categories. The framework consists of AI and ML categories, all collected skills are mapped onto one of the available categories. Additionally, the classification framework was validated by three independent raters to ensure the reliability of the skill categorisation. A high level of agreement among raters was recorded by the alpha coefficient.

Once the data had been collected by a scrapper and categorised by a classifier, the report presented its final results. The analysis revealed a substantial demand for AI and ML professionals, particularly in California, Washington, and New York.

The above mentioned paper is very well written, one of the aspects that really stood out was the use of a classifier. The classifier was able to sort a large amount of data, essentially streamlining the mapping process to a single piece of software. This is a very useful technique and a similar classifier will be created and used in our report. This Kansas paper examines the geographical distribution of AI and ML job opportunities. This process of analysing the key areas with high demand for skilled professionals adds context to the job market trends and emphasises the importance of location in job search strategies. The final positive aspect that really stood out in this paper is the "n-grams" (unigrams, bigrams, and trigrams) that the raw data was collected by. This takes into consideration that skills are not just single words, rather they can be a phrase of two or more words. A similar phrase analysing will also be conducted in our paper.

Despite its positive features, there are a few points that can be criticised and avoided in our study. Firstly, this paper from Kansas often makes decisions without telling the reader about the thought process behind the decision or any alternatives that were also considered. For example, why was indeed picked as the job advert source? Or which kind of classifier was used and why? Unsupported decisions like this make the reader question the reliability of the

paper. Another point of improvement is that although the findings are presented clearly in table form, there are not many visual diagrams that can be seen. Given the task at hand is to analyse data, including a few charts or graphs with the statistics found in them would help not only the authors to find a skills gap but also the readers to better visualise the findings.

Another paper [2] written by both Sam Attwood and Ashley William is studying the skills gap in the cybersecurity industry. Despite a different industry being considered, the approach used should still be applicable for the analysis we are conducting.

The paper outlines the significance of addressing the skills gap, especially in cyber-enabled disciplines like software engineering, and sets out to evaluate the potential of mapping job descriptions to the Cyber Security Body of Knowledge (CyBOK) using TF-IDF representations. TF-IDF stands for "Term Frequency-Inverse Document Frequency." It is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents representations.

The researchers gathered information from a source that provides job listings in the UK related to cybersecurity and software engineering. This data includes details like job titles, descriptions, salary ranges, and locations. They categorised the job listings into different types based on their titles and descriptions.The researchers looked at where the jobs were located and the salaries associated with them. The main part of the study involved connecting the job descriptions to specific areas of knowledge outlined in the CyBOK. They used a method called TF-IDF, which basically measures how important specific words are in a document compared to a larger collection of documents.

They converted the CyBOK into a structured format that a computer can understand and then compared the words in the job descriptions to the words in the CyBOK to see which knowledge areas were most relevant to each job. They then looked at the similarity scores generated by the TF-IDF method to see how closely job descriptions matched up with different knowledge areas in the CyBOK.

The report concludes by stating that a skills gap was found and how students can be better prepared. The analytics discussed throughout the paper are discussed very clearly via diagrams which aid in understanding and also remembering the information stated in the paper. For example this boxplot (figure 2.1) showing the similarity score for job types across the main 5 knowledge categories in the CyBOK.

However, while the use of TF-IDF representations offers a computationally efficient way to

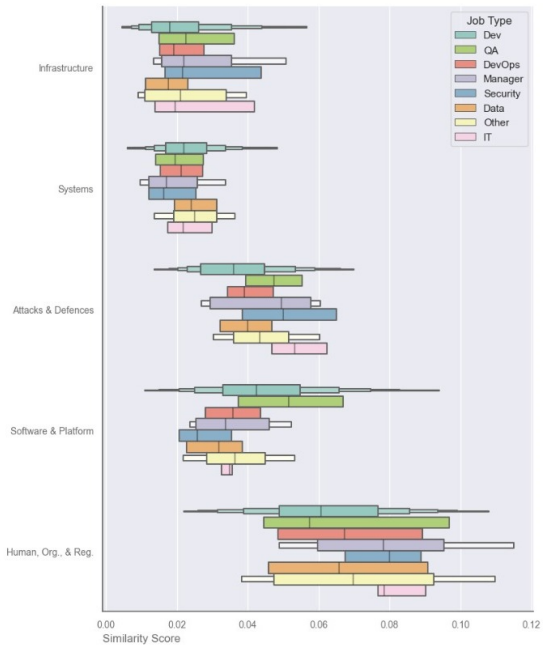

Figure 2.1: Box plot from Cyber Security Paper [2]

map job descriptions to the CyBOK, it oversimplifies the complex relationships between job requirements and knowledge areas. It would be better to use some other technique such as the frequency analysis of the skills collected that was used in the Kansas paper [26] previously discussed.

The papers studied above both have their own advantages and disadvantages but overall provide a robust analysis of the skills required by AI job adverts. A commonality in both of these papers and various other similar papers is web scraping to collect data.

The success of web scraping is further emphasised in a research paper [12] which analyses different data collection methods for online market research in the hospitality domain. This paper compares alternative approaches to data collection in depth.

The paper rightly notes that the internet provides valuable opportunities for collecting and studying consumer choices. The report then goes on to compare web scraping, programming interfaces (APIs), taking surveys and simulating online behaviours in lab settings or surveys. The table included in the paper [12] gives a very good comparison of all the stated methods of data collection. 2.2 The table in figure 2.2 conveys how API access can be challenging, limited, or costly, and simulation studies may lack realism. While outsourcing web scraping to third-party firms is an option, potential pitfalls include lack of discretion in data collection. However web scraping is cost-effective and provides large-scale, real-time data, contrasting with

	Scraped Data	Commercial Web Scraping Service	API	Survey
Cost	Low	Medium	Low/Medium	High
Sample frame	Website users	Website users	Website users	Flexible
Customizability of variables	Medium	Low	Low	High
Ease of frequent collection	Easy	Moderate	Easy	Hard
Data type	Behavioral	Behavioral	Behavioral	Attitudinal
Limitations	Time and programming skills	Data may not be suitable to the researcher's need in terms of variables or content	Limited availability	Time and programming skills

Figure 2.2: Comparison of Common Data Collection Methods [12]

the challenges of conducting surveys.

This next paper [17] by Ms. Charu Sarin, Assistant Professor at Delhi Institute of Advanced Studies Identifies a disparity between the skills perceived as important by students and those valued by employers. Because the skills gap here is being identified in terms of opinions of people, the methodology used is a bit different to the previous reports. Before the main study, a pilot study was conducted with a small sample (35 students) to assess the face validity of the questionnaire. The main study involved collecting primary data from 230 graduate and post-graduate students in the Delhi-NCR region. Convenience sampling was used, which means participants were selected based on their easy availability. In the second part of the study, primary data was collected from 50 human resource (HR) executives in the Delhi region. These HR executives were responsible for recruitment from higher education institutes. Both students and HR executives were asked to rate the importance of skills on a scale ranging from 1 (least important) to 5 (most important). This allowed for quantifying perceptions of skill importance.

Adding on, other papers [8]from the official government website looks extensively at the UK AI Labour market, it correctly points out a skill shortage, difficulties in filling vacancies and the places where most jobs adverts were posted. This study has lots of diagrams so readers can get a better understanding of the statistics it is talking about. This highly visual representation of key finding is also prominent in other similar papers. Such as in this [27] investigation of skills required in artificial intelligence and machine learning jobs by Missouri Western State University and University of Kansas in the USA. The paper's findings stated that although "technical skills like data mining, programming, statistics and big data" are valued in both AI and ML jobs, AI positions tend to be more generic, with an emphasis on communication skills. The importance of soft skills is further illustrated in an article by Cheryl Aasheim [6] where Cherly concludes that "soft skills remain highly valued, in addition to the value placed on emerging hard technological skills".

In conclusion, reading this wide range of papers has given us a very strong understanding of what similar literatures have been doing. Reviewing their methodologies and results have proved beneficial in showing us aspects that will work well in our report and what areas we can improve. For example, the concept of grouping skills into n-grams before doing a frequency analysis shown in a Kansas paper seemed very useful and will be implemented in our paper too. The cybersecurity paper with its diagrams provided brilliant visual aid. This visualisation will also be present in our paper for the ease of the user. Adding on, similar to how the Web scraping for hospitality did an in depth analysis of all the decisions made and the variety of alternatives used will also be reflected upon here. Lastly, the pilot study that was conducted in the delhi paper is a great way to ensure reliability. A similar smaller study will be conducted for this report, mentions of which can be found in the evaluations section. Despite the several good qualities that these papers have, none of them actually map skills from university degrees to the skills that are required by AI recruiters, this is a unique aspect that our paper will have. This mapping will hopefully uncover further disparities, ultimately helping us find a skills gap.

2.2 The Skills Gap

According to LinkedIn's 2023 "Jobs on the Rise" report [24], positions similar to those held by Machine Learning Engineers were recognised as one of the most prominent (top 20) jobs emerging this year. Over the past 6 years, there has been an over 74% increase in demand for these roles, driven by the rapid progress of new technologies and expansion of big data. However, it is quite hard to find candidates with perfect resumes and qualifications for such jobs [15]. Research even within the extensive domain of the UK space industry has revealed that a significant majority, 95% to be exact, of space organisations encounter skill-related challenges. Among them, nearly a quarter express a distinct demand specifically for Artificial intelligence, surpassing the demand for any other technical domain [14] [23].

This same concern is also voiced by Charlie Ackerman, senior Vice President of human resources at Bosch, a leading tech company that uses and offers many AI applications, in stating that rapid advancements in AI have "created a supply demand problem, where the demand for highly technical skilled workers is outpacing the supply"[15].

A research paper published by Oxford university in October 2023 [22] shows just how big

a gap there is between the skill set of job applicants with degree level qualifications and what skills are actually required in AI industries by claiming that there are multiple states in the US where there are "more than 10 job ads per AI professional". The paper, [22], goes further and highlights what a negative impact this large skill gap can have; it asserts that the high demand for AI engineers leads to some candidates with "near to no" experience being hired for senior roles. Which may severely impact the efficiency of the company.

The UK government has already initiated steps to tackle the data skills deficit. In Autumn 2020, DCMS (Distribution Center Management System) and the Office for AI introduced a degree conversion course program focused on data science and AI. This initiative's objective was to produce a minimum of 2,500 graduates within a span of 3 years with the intention of narrowing the skills gap [21] [9]. Data, from April 2020 to March 2023, shows 7,600 students have enrolled on AI and data science postgraduate conversion courses supported by OfS allocated funding [5]. Reports suggest that the programme has had a "substantial positive effect" on the number of postgraduate students in AI in the UK with extra opportunities being provided to women, black students and students with declared disabilities [5]. Despite the success of the program, Alastair Wilson, a segment head of the program, has said that looking into future academic years, universities should explore where further work is needed to continue the shrinking of the present skills gap [5].

Leading on from Wilsons' remarks, conducting a comprehensive analysis of the skill gap between higher education and AI industries will benefit universities, enabling them to offer a broader and more in-depth range of courses to better serve future students. Furthermore, it will help employers to attract a more diverse pool of job applicants with the appropriate qualifications for the positions they are seeking to fill.

Adding on, other papers [19] from the official government website looks extensively at the UK AI Labour market , it correctly points out a skill shortage, difficulties in filling vacancies and the places where most jobs adverts were posted. This study has lots of diagrams so readers can get a better understanding of the statistics it is talking about.

To summarise, many studies investigated the demands of industry and analysed job advert but none made the mapping to programmes in HEI. Adding the mappings between skills provided by university courses and the skills job adverts require will be a novelty of this study. Also, most studies were carried out for the USA there were not many for the UK or for Europe. Many UK universities now offer the full programme in AI (master) which is new and shows the

response to the job market demand. However, we do not know how well these AI programs equip students with skills for a job in the AI sector, this is what we will find out in this study.

Chapter 3

Design

3.1 Introduction

The data analysis done in this report involves various steps and processes, this flowchart provides a visual representation of these processes, making it easier to understand the flow of data. There are two main datasets that are used in this report, as represented by the two parallel columns in the top half of the chart. These columns later merge into a single dataset that is then used to find patterns in the previously acquired data (figure 3.1). First, the skills that

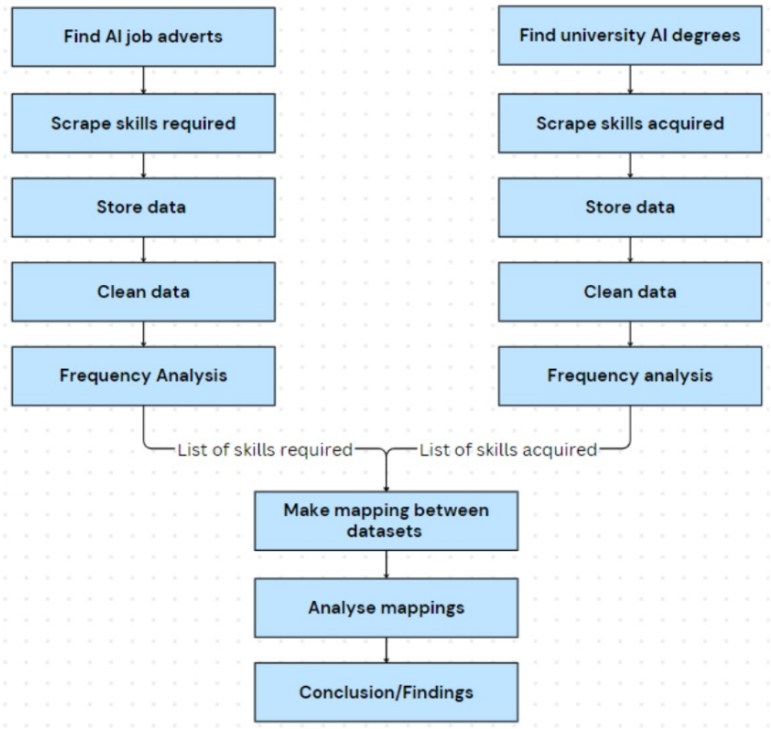

Figure 3.1: Flowchart of Design

employers require for AI related jobs are scraped and stored. But this data also has to be cleaned and the frequency of popular skills should be recorded. This pair of "skill required" and the frequency of the skills together from the first dataset. The second dataset is created in the same way but this set will contain the skills that are acquired by students who have completed AI related degrees at university. We will then perform some sort of mapping between the two datasets to see if any pattern or correlation can be inferred to present in the final report findings.

3.2 Finding Reliable sources

To truly see if there is a skills gap between higher education and the industry, it is a good decision to see what skills are usually:

- Required by employers in the AI industry
- Acquired by university students doing AI related degrees

Making sure that the sources used for this data collection are reliable was very important as the data collected during this stage of the implementation would form the basis of the whole analysis that was to come.

3.2.1 Types of Sources

A data analyst, Luke Barousse, states that all the data that we find on the internet is either clean or dirty and either public or private. Using combinations of these 4 options, there are 4 main data sources that we can collect data from:

- **Private** data can be either Clean or Dirty but this data is private and is not available to us. Popular job data collection sites like linkedIn, indeed, Monster, and Glassdoor do not have an API to access the data. (LinkedIn used to, but it is now deprecated because they did not want anyone to create a linkedIn competitor). Because we do not have access rights, this kind of data will not be useful for this report.
- **Clean and Public** - Clean publicly available data is found on sites like google, data.gov and some github locations. Although there is a lot of data, there is not much in depth information such as what skills are required for a job. This again does not prove to be useful for our aims.

- **Dirty and Public** - Data that is not clean but is public is all the publicly available data on the internet that you can access through web scraping.

The data sources that are targeted in this report are dirty(raw data) and public, that is: linkedIn job adverts, university webpages. The sources are talked about in further detail in the two subsequent subsections.

3.2.2 Jobs - Dataset 1

The benefit of using an online job advertisement site is that they usually have a lot of jobs advertised together in the same place, making data collection later on a lot easier. However, there are many different good choices of advertising websites, some examples being LinkedIn, Indeed, Glassdoor, Monster Jobs.

Despite the wide range of options available, this report will make use of LinkedIn. This decision was made because it gives a lot of detailed information about the companies posting job adverts. Each job description tends to be pretty thorough, giving a good sense of what the company is like. Additionally, LinkedIn lists out exactly what they are looking for in a candidate, i.e. the required skills. This will probably be the most important part for this report. Another big factor is that LinkedIn has many filters to help narrow down your search, which makes finding the right job postings a lot easier and more efficient.

3.2.3 Universities - Dataset 2

On the other hand, online universities websites are a highly reliable source for the skills that students gain while at that institution. These sites are very structured, easy to navigate and frequently updated with the latest module changes. On top of this, each university usually only has a single website with all the information in one place, this makes the data collection easier as we do not have to compare among multiple other sources.

The only major decision to make at this stage was which universities to select for the data collection. Here, an unexpected discovery emerged regarding AI courses; there exist many universities that do not offer any AI courses yet. There are 106 (out of 166) universities in the UK that do not provide AI courses at all [20] [18]. Furthermore, In 2017 (i.e. 6 years ago) there were only 26 UK universities offering undergraduate courses in AI [16]. These findings are presented in the pie chart (figure 6.5). We will go deeper into this finding in the evaluation section of this report.

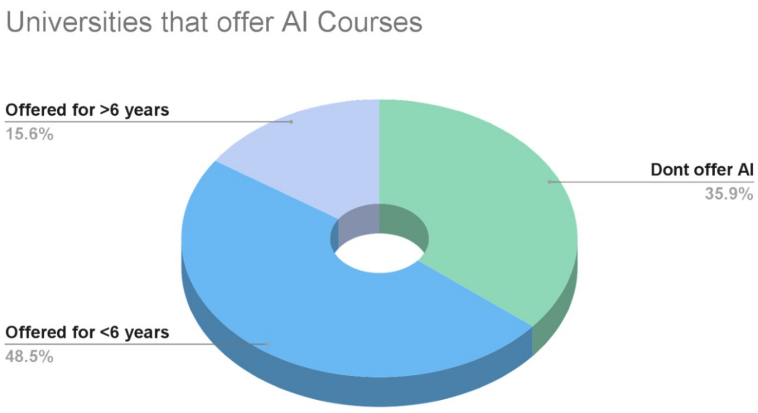

Figure 3.2: Proportion of Universities that offer AI courses

To address this limitation, this report is restricted to mainly focus on universities that do offer AI related courses. Taking into consideration the fact that a skills gap is what we are after, for this report, skills from the most highly ranked universities in the UK according to "university QS ranking" [1] have been collected. The rationale behind this decision being that our analysis can identify a skills gap between the best AI courses that the UK had to offer, then there must be an even larger skills gap posed by university degrees that are not as highly ranked.

3.3 Data Collection

Once the sources are decided, the next step according to the flowchart is to collect data from them. There are a few ways to scrape data from LinkedIn job adverts, some of them are discussed below before coming to our final option of using a third-party tool provider; Clay.

3.3.1 Application programming interface - API (option 1)

API's are very convenient, they usually have all the data present in a sorted, ready to use manner. LinkedIn provides an API for certain purposes, but access to job advertisement data is typically restricted or unavailable. Usually only workers of LinkedIn who have a legitimate business need for LinkedIn job data are able to access such information. So this option is completely infeasible for this report.

3.3.2 Manual Scraping (option 2)

This involves manually copying and pasting data from LinkedIn job adverts into a spreadsheet or database. This is the safest way to collect data without violating LinkedIn's terms of service. However, we will require a very large dataset for this report, for which this manual scraping method is too time-consuming and not scalable. The time spent on performing trivial tasks such as individually scraping each job advert can be better invested in such other parts of the report as data analysis. Adding on, this method is error prone because it is inevitable for humans to make mistakes.

3.3.3 Automated Web Scraping (option 3)

Automated scraping involves using web scraping tools or libraries to extract data from LinkedIn pages. LinkedIn data scraping is legal, but it is still not encouraged by the platform [4].The platform uses algorithms to try and detect unauthorised scraping by monitoring activities that do not seem "human." For example, browsing loads of profiles in a short space of time may set off LinkedIn's automation detection alarm bells. However, there are safe "rate limits" for scraping data on LinkedIn [3]. It is very easy to exceed these rate limits when creating your own web scraper. There is a penalty of scraping in a non-human manner. LinkedIn's official site states "LinkedIn's harsh policies against unauthorised data scraping tools pose a significant risk. Use the wrong tool, and you could find your account flagged or, worse, permanently banned" [25]. In order to scrape from linkedIn in a safe and legal manner it may be better to use a pre-build tool that has already been tested previously.

3.3.4 Third party scraper (chosen option)

A third-party tool provider in the context of data scraping typically refers to a company or service that offers software or tools specifically designed for extracting data from websites like LinkedIn. For the purpose of this report, this method has been considered to be the best way to scrape data. These third-party scrapers can extract large sections of raw data, without causing any legal issues related to unauthorised data scraping (for example exceeding the "rate limit" seen in the previous subsection)

The third party tool that we will use in our project is a chrome extension called Clay. The reason the third party chosen is specifically chrome is because it is widely known so it should not cause any errors and its extensions are very easy to use. Here are some advantages of Clay that make it best for our use:

- Scalability - it can handle large amount of data while not exceeding the rate limit of LinkedIn
- Customization - Clay provides options for customising the scraping process, such as specifying the data fields to extract and setting up filters for the scraping process.
- Compliance - Chrome, being a well reputed browser, prioritises compliance with legal regulations and website terms of service, offering features to help mitigate the risks associated with web scraping.
- Data Management - Clay is able to manage and organise the data scraped into csv. (comma separated value) files, allowing easy data manipulation later on.

3.3.5 Technique

Extensive research has been done on web scraping. Papers [28] [13] [7] often state that web scraping can be divided into 3 stages (as shown in figure 3.3)

- Fetching stage: First, we need to get to the right website to find the information we want. We can do this by using the HTTP protocol.
- Extraction stage: Once we get the webpage, we need to pull out the important information. We use things like regular expressions, HTML parsing libraries, and XPath queries to do this.
- Transformation stage: Now that we have only the important information, we can organise it in a way that makes sense or save it for later. This stored data helps businesses make smarter decisions.

This 3 stage process is implemented by clay and hence will be used for our report. 3.3

3.3.6 Clay (a data scraper)

The web scraper used in this report is a chrome extension called “Clay”. The Clay chrome extension lets you scrape data from any web page directly into clay. The target website for this report is LinkedIn. Figure 3.4 shows the stages of working with clay and also how it all leads to the production of raw data that will later be processed by the python tool seen in the implementation section. In depth analysis of the source code created to process the raw data will be discussed in the Implementation section. For now this code has just very briefly been mentioned in the timeline above for completeness.

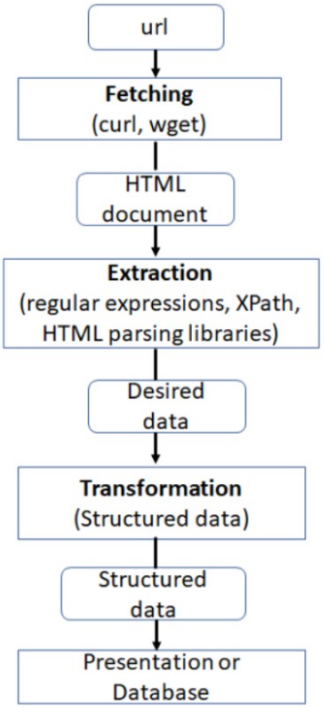

Figure 3.3: 3 stage data extraction process [28]

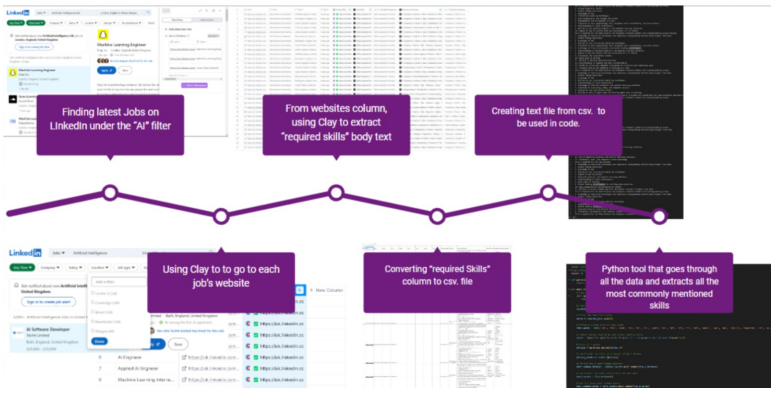

Figure 3.4: Flow chart to show the collection of raw data

3.3.7 Manual Collection

Web scrapers rely heavily on the hierarchical structure of HTML documents to locate and extract relevant data. Each university's website has a different structure, hence different HTML structures. Each data scraper is coded to navigate the format of one website only.

Given these varied layouts of university websites, it was obvious that utilising a single data scraper would not be enough to scrap every university's website. And coding a different web scraper for each university is just impractical. Therefore, manual extraction was deemed to be the most appropriate here. The second set of uncleaned data, comprising the skills taught by approximately 30 leading UK universities' AI degrees, was manually collected to be utilised in this report.

3.4 Data Storage

At this stage, two csv. files are present with either one of job adverts or universities names accompanied with the skills that are required or acquired from each of them respectively.

3.5 Data Processing and Frequency Analysis

3.5.1 Data Processing and Frequency Analysis

The data that we currently have in each csv. file is raw. Raw data is data in its original format straight from its source with minimal alterations. It is really hard to make inferences and find patterns in this kind of data. This raw data was transformed to processed data to make it more easily digestible by human readers. A python processing tool will need to be created for such a transformation. The functionality provided by the Python Natural Language Toolkit (NLTK) is very useful for and can be implemented very easily into our tool. There are various tasks that you can perform with the NLTK but here are some main groups that are usually associated with it:

- Text Processing and Tokenizing - used to break down a text into individual words or sentences, further frequency analysis can then be very easily automate
- Stemming and Lemmatization - This reduces words to their most basic or root form. This then is helpful in tasks like information retrieval and text mining.

- Parsing and Syntax analysis - here sentences are parsed to understand their grammatical structure, this is essential for applications like natural language understanding, question answering, and machine translation.
- Named Entity recognition (NER) - identifies and classifies named entities in text, aiding in information extraction.
- Frequency distribution and probabilities - analysing frequency distribution and probabilities helps understand the distribution of words in a text.
- Sentiment analysis - this determines the sentiments expressed in a piece of text, this can be helpful in understanding public opinions
- Machine Learning integration - facilitates custom classifiers for various natural language processing tasks.

We will later see the implementation of some of these NLTK features in the implementation section of this report.

3.5.2 Frequency Analysis

The tool to clean the data that we just mentioned in the previous section will also give each skill an associated frequency depending on how frequently that skill appeared in the raw data. This frequency acts as a weighting, giving more weight to frequently occurring skills and less to others. This weighting system will be very useful for the evaluation of the collected data later on. See the implementation section for a full breakdown of the tool implemented.

3.6 Mapping

Once both the datasets have had frequency analysis, some sort of mapping will be needed to be performed between them in order to evaluate them. This can be seen as the merged section of the flow chart 3.1.

Another tool has been put together to aid in the mapping process, and you will find a detailed breakdown of its features in the implementation section. But stepping back a bit, the main idea is to organise the skills from both sets into 12 big categories related to AI. Doing this by hand is not really practical, so we will need to write some code to automate the sorting process. This way, we can make sure our analysis of AI skills across the datasets is both accurate and efficient.

Recall: one of the features of the NLTK is machine learning integration which facilitates custom classifiers for various natural language processing tasks. A machine learning classifier will be used for the categorisation of the skills from each dataset into one of the twelve pre-defined groups.

Alternative Categorisation methods: There is a diverse range of methods available for categorisation via machine learning, each suited to different data and problem domains. Because we have pre-defined the number and names of categories that our scraped skills will be sorted into, supervised learning algorithms will be the best choice. Supervised learning algorithms are trained on labelled data (categories), allowing them to make accurate predictions or classifications on unseen data. With proper implementation and a large enough training set, supervised learning algorithms can process large volumes of data efficiently, making them suitable for real-time applications and big data analytics. Below are some of the supervised learning models that have been experimented with for this report before coming to the final choice of using a Naive Bayes classifier.

3.6.1 Support Vector Machines - SVMs (Option 1)

Support Vector Machines (SVMs) are powerful supervised learning models which can be used for both classification and regression tasks. They are particularly effective at solving binary classification problems but can be extended to handle multi-class classification as well.

The key idea behind SVMs is to find a hyperplane in a higher dimensional space that separates the classes in the feature space. This hyperplane is chosen to maximise the margin, which is the distance between the hyperplane and the nearest data points from each class. These data points are also called support vectors (hence the name support vector machines). SVMs use a kernel trick to transform input data into a higher-dimensional space, simplifying the separation of the classes by finding a hyperplane. This process computes the dot products in the higher dimensional space without explicitly transforming the data, making SVMs computationally efficient, particularly with high-dimensional data. However, we know that the data that we want to categorise is not of a high dimensional space so this advantage of SVMs would go to waste in our implementation.

However, as the number of classes increases, the complexity of the classification problem also grows, making SVMs less practical for sorting tasks involving a large number of categories. Hence, SVMs may not be the most suitable choice for sorting skills into 12 categories.

3.6.2 Decision Trees (Option 2)

Decision trees can also be used for both classification and regression tasks. It works by recursively partitioning the input space into subsets based on the values of input features. Each internal node of the tree represents a feature along with a decision rule, and each leaf node represents a class label or a regression output. The process of constructing a decision tree involves selecting the best features at each node to split data, typically based on some criteria like the gini impurity or information gain.

The main advantage of decision trees is their interpretability, the tree structure is easy to visualise and understand. However, decision trees are prone to overfitting, especially when the tree becomes too deep, this can lead to poor generalisation performance on unseen data. Finding the optimal tree structure can be quite challenging. Given the large number of different categories that we have, a decision tree may not be the best choice as it will not be able to capture the relationship between skills and categories, leading to suboptimal performance. Moreover, decision trees are prone to overfitting if the dataset is not sufficiently large or diverse. Alternative approaches such as Random Forests, that combine multiple decision trees, may be more appropriate, but these are much more complex and leading to lack of interpretability, understanding the decision process becomes very hard very quickly here.

3.6.3 Naive Bayes (Option 3)

The Naive Bayes model is a probabilistic machine learning algorithm based on Bayes' theorem, which describes the probability of an event occurring given prior knowledge of conditions that might be related to the event. Despite its simplicity, Naive Bayes is powerful and commonly used for classification tasks, including text classification. Text classification is exactly what we are trying to do in this report. The "naive" part refers to the independence assumption, where we assume that the presence of a particular feature in a class is independent of the presence of other features in the class. Although this assumption does not always hold true in real-world scenarios, naive bayes can still perform well.

The algorithm works by calculating the posterior probability of each class given the input features and then selecting the class with the highest probability as the predicted class.

One of the main advantages of Naive Bayes is its simplicity and efficiency. It requires a small amount of training data to estimate the parameters and is computationally inexpensive, making it suitable for large datasets and real-time applications. Because of its simple model, it is less prone to overfitting compared to more complex models.

The simplicity and efficiency of Naive Mayes make it well-suited for handling large datasets with numerous features, such as the long list of skills that we have. Furthermore, the algorithm can handle multi-class classification tasks effectively, making it suitable for categorising skills into the 12 distinct categories. Finally, Naive Bayes tends to perform well even with limited training data, this makes it even more favourable for our project because it means that the training set that we give it can be smaller than the ones required by other supervised learning algorithms.

3.6.4 Advantages of Naive Bayes

These are some of the main advantages of Naive Bayes that make it favourable for our use.

- Simplicity and Efficiency - Naive Bayes classifiers are straightforward and easy to implement. They have minimal parameters to tune, making them computationally efficient, especially for large datasets. Both of our datasets are very large, each containing 2000 different skills with their corresponding frequency analysis.

- Handling High-Dimensional Data - Naive Bayes performs well even with high-dimensional data, such as text documents. It can handle a large number of features efficiently without suffering from a large number of dimensionality, making it suitable for tasks with a large number of input variables. The input variables that we will give the classifier will be phrases of a varied number of words in them.

- Scalability: Due to their simplicity and efficiency, Naive Bayes classifiers scale well with increasing dataset sizes. They can handle large datasets and real-time applications with minimal computational resources

- Less Sensitivity to Irrelevant Features: Naive Bayes classifiers are robust to irrelevant features in the data. Since they assume feature independence, irrelevant features are less likely to affect the classification performance significantly. This property makes Naive Bayes particularly useful for noisy or incomplete training data.

- Effective with Small Data: Naive Bayes classifiers require a relatively small amount of training data to estimate the parameters accurately. This makes them suitable for tasks with limited labelled data, where other more complex algorithms might overfit.

3.6.5 SciKit-learn

The python library that we will use to implement the naive bayes classifier is sciKit-learn. Scikit-learn is an open-source machine learning library that provides simple and efficient tools for data analysis. It is built on NumPy, SciPy, and Matplotlib and offers a wide range of supervised and unsupervised learning algorithms for classification, regression, clustering, dimensionality reduction, etc. Scikit-learn is designed to be user-friendly, with a clean and consistent API, making it very easy to use.

3.7 Mapping Results

By this time, we have two datasets, each having all the skills present in them sorted into 1 of 12 different categories. The categories from each dataset can now easily be mapped onto each other and compared to see if any patterns can be found. The findings of this mapping will be provided in the evaluation section of this report

Chapter 4

Specification

A large quantity of data will be required for the data analysis that this report aims to do. As discussed previously, it is impractical to manually collect such large quantities of data. For this project, a web scraping tool needs to be created to initially scrape job adverts, gathering skills sought by employers in prospective employees. It will also browse university websites to compile skills taught in AI courses at the degree level. These processes result in two datasets that are later mapped and utilised for analysis. Additionally, another software component is required for the mapping between these two datasets.

Establishing precise specifications early on for the tools implemented will ensure that the resulting output meets a satisfactory standard and effectively serves its intended function. Functional, non-functional and other relevant requirements for this data scraper are stated below.

4.1 Functional Requirements

- **Data Extraction** - The tool should be able to extract relevant information from LinkedIn job postings i.e. job titles, company names, location and skills required.
- **Skill Identification** - It should be capable of identifying and extracting specific skills mentioned in the job postings. At the same time it should ignore other irrelevant words in the job description.
- **Scalability** - The tool should be scalable to handle a large volume of job postings without a significant decrease in performance.

- **Data Storage** - The extracted data, including job titles and skills, should be stored in a structured format for further analysis (e.g. csv. file).
- **Automation** - The tool should automate the process of navigating through job postings and extracting information, reducing the need for manual intervention.
- **Error Handling** - The tool should have a mechanism to handle errors gracefully, e.g. cases where a job posting has information in an unexpected format or stored data has a specific field missing.

4.2 Non-Functional Requirements

- **Accuracy** - The tool must precisely extract information from job postings, minimising errors in categorization and ensuring an accurate representation of skills.
- **Reliability** - The tool should consistently perform under diverse conditions, remaining resilient to changes in web page structure and recovering gracefully from errors to ensure data integrity.
- **Speed** - The tool should operate swiftly with minimal response time, facilitating efficient extraction of skills data from a large volume of job postings.

4.3 Other Requirements

I used the browser extension "Clay" to initially extract data from Linked-In. Clay is a good choice because:

- **Flexible** - Clay can be used for data extraction on many different job posting sites
- **Ease of use** - Clays intuitive user interface makes it very easy to users to understand and extract data.
- **end product format** - Clay creates a csv. file of extracted data, from which it is easy to manipulate and filter useful data.

A CSV. File is a good choice for data collection because:

- **Lightweight** - CSV files are lightweight compared to more complex data storage formats. They don't include formatting or formulas, which can be advantageous when the goal is to store and share raw data.
- **Simplicity** - CSV files have a simple and straightforward format, making them easy to create, read, and edit by both humans and machines.
- **Compatibility with Data Analysis Tools** - Many data analysis tools, statistical software, and programming libraries (e.g., Pandas in Python) have robust support for reading and writing CSV files, making it convenient for data scientists and analysts.

The web scraping tool will be coded in python. Reasons for this:

- **Abundance of Libraries and Frameworks** - Python has many ecosystem of libraries and frameworks specifically designed for web scraping that simplify the process, providing pre-built functions for common tasks e.g. BeautifulSoup and Scrapy.
- **Cross-Platform Compatibility** - Python can run on different operating systems without modification. Hence making it advantageous for developers working on various platforms.
- **Built-in Features** - Python comes with many built-in features that facilitate web scraping e.g. regular expressions for pattern matching.
- **Strong Data Analysis and Visualization Capabilities** - Python has powerful data analysis libraries like Pandas and visualisation libraries like Matplotlib and Seaborn. They make it convenient to analyse and visualise data extracted during the web scraping process.

Chapter 5

Implementation

It is evident from the design section of the report that two main tools were implemented for this project. First, a data processing tool to clean the raw data that clay collected. The cleaned data for each dataset would have all the most frequently occurring skills and their respective frequencies. This section of this report is split into two main parts; one for the implementation of the data scraper to clean the two datasets and one for the implementation of the machine learning model to generate a mapping between the two datasets. We will first look at the implementation of the data scraper.

5.1 Data Processing Tool

There are many ways to analyse the scraped data but the technique implemented here is to find which skills occurred more than once in the raw data and record their frequencies. This process is conducted on both the datasets before performing the mapping between the two.

5.1.1 Clay

Clay first sends a HTTP request to the website's server to retrieve the HTML contents of the page. The received HTML content from the websites is parsed, this involves breaking the HTML text into individual components such as tags, attributes and text content. In our case, the clay extension breaks the received page into separate jobs and each job is broken into smaller sections like the job title, job description, the skills required, the date it was added to the website etc. The user can then choose which attributes they want displayed in the final output that clay provides.

For this report, the filters on linkedIn have been first set to only display AI related vacancies in the UK that had been put up in the last three months. Next, clay's data scraping preferences are altered to make sure that it only collected each job advert's title, and the skills required by that job. After having collected data from just over 150 adverts, the scraping process was stopped and clay produced a csv. file. The data in the file however, was not clean, and will need to be processed to provide useful insights.

5.1.2 Input

The input to this data processing tool is a long list of skills that have been collected from either job adverts or university webpages. The job dataset has skills from 158 AI related job adverts. The university dataset has skills from 30 different universities AI courses. Either way, both the datasets currently do not have any structure at all. They both contain many repetitions of skills, bullet points, numerical values, dashes, commas, other punctuation marks, etc. In essence, whatever format the data was present in the scraped website has been completely transferred to our dataset. Our goal is to clean this up and present that data is a clearer, easier to digest way.

In the file structure for the implementation, both the datasets are stored in separate text files. These text files are then passed onto the main code through the filePath parameter.

5.1.3 N-grams

Below (figure 5.1) is the code written for text processing . This code performs text processing and analysis tasks that are usually associated with the NLTK; cleaning text, removing stop words, and generating n-grams (n-grams are sequences of n consecutive words).

The python code defines two main functions; `generate_ngrams` and `most_frequent_phrases_and_words_from_file`. The large method names, though not required, provide a very intuitive way of explaining exactly what the method does. Due to the large names, these two functions may be referred to as function 1 and function 2 in this subsection of the report. The `generate_ngrams` function takes the list of words and an integer, 'n', as input and returns a continuous sequence of n words from the initially provided text. List comprehension is a simple way of creating lists in python and is often used in the context of natural language processing. This first function makes use of list comprehension and the zip function to generate these n-grams by iterating over the list of words. This function is a helper function that is called by the longer, second function that we have.

```python
def generate_ngrams(words, n):
    return zip(*[words[i:] for i in range(n)])

def most_frequent_phrases_and_words_from_file(file_path, n=2, top_k_phrases=5):
    # Read the content of the file
    with open(file_path, 'r', encoding='utf-8') as file:
        text = file.read()

    # Remove non-alphanumeric characters and convert to lowercase
    cleaned_text = re.sub(r'[^a-zA-Z0-9\s]', '', text.lower())

    # Split the text into words
    words = cleaned_text.split()

    # Define a custom list of stop words
    stop_words = set(['for', 'the', 'and', 'in', 'to', 'of', 'with', 'on', 'at', 'by', 'is', 'are', 'were', 'was', 'am',
                      'skills', 'required', 'or', 'a', 'experience', 'such', 'as', 'from', 'but', 'like', 'some', 'these',
                      'into', 'be', 'etc', 'eg', 'an'])

    # Remove custom stop words and purely numeric words
    words = [word for word in words if word not in stop_words and not word.isnumeric()]

    # Generate n-grams
    phrases = generate_ngrams(words, n)

    # Use Counter to count occurrences of each phrase
    phrase_counts = Counter(phrases)

    # Get the top_k most common phrases
    most_common_phrases = phrase_counts.most_common(top_k_phrases)


    return most_common_phrases
```

Figure 5.1: Text processing tool

5.1.4 Most common phrases

The second function reads text from a file specified by its `file_path` parameter. This file contains raw data that was previously collected containing the list of skills that are required by employers. This second function first cleans the text by removing non-alphanumeric characters, converting it to lowercase, and splitting it into individual words.

Next, common "stop words" are removed from all the words. Stop words are words that only make sense when said with a larger string of words but alone they are not useful on skills analysis. For example, some common stop words are "etc.", "e.g.", "such as", there are many more. Then this method also removes any purely numeric words from the list of words. Up till now this creates a long list of words in lowercase all together showing the skills required by AI employers.

The methods explained in this section will be helpful for the frequency analysis that will need to be done on the raw data next.

5.1.5 Frequency Analysis

This long list of words has uniform structure and can be passed on to the first method explained above. This method finds n-grams from this list. The counter class is used to count the

occurrences of each n-gram skill. Adding a relevant frequency of the skills occurrence gives a weighting to the skills and helps in giving us a better understanding as to which skills are most commonly required by employers. The second function returns the most common n-grams, along with their frequencies, as a list of tuples. For better understanding - the tuple: [((‘programming’), 67), ((‘machine’, ’learning’,), 82)] shows that there are 67 occurrences of the word “programming” and there are 82 occurrence of the 2-worded phrase “machine learning” in the input text.

An investigation like ours should only require phrases containing 1, 2, 3 or 4 words to have their frequency analysed. Longer phrases do not usually occur or only occur a very small number of times to be considered significant. But to ensure a good coverage, even 5-worded phrases are included in our analysis.

5.1.6 Output

This skill phrase frequency analysis is conducted on both datasets, collecting the top 400 most commonly occurring phrases for each of the 5 different lengths of phrases, giving us a total of 1000 skills in each dataset. This result is currently just being outputted to the terminal. The format printed to the terminal (figure 5.2) is what we will use in the next part of the implementation. This is the result that we wanted from this part of the implementation but to

Figure 5.2: The processed data being printed out in the terminal

aid readability, the `list_of_tuples_to_excel` function is added to the bottom of this tool. As its name suggests, this function puts each element of the tuple in an excel sheet (figure 5.3) named ”output” which can be found in the source code provided with this report. Note: The

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Word (length 1)	Count		Phrase (length 2)	Count		Phrase (length 3)	Count		Phrase (length 4)	Count		Phrase (length 5)	Count
2	data	330		machine, learning	200		python, coding, knowledge	33		python, coding, knowledge, sql	33		data, python, coding, knowledge, sql	28
3	learning	256		data, science	76		coding, knowledge, sql	33		analytical, techniques, manipulating, deriving	28		data, science, analytics, graduate, scheme	27
4	machine	205		computer, science	76		machine, learning, solutions	30		data, python, coding, knowledge	28		science, analytics, graduate, scheme, leading	27
5	science	160		ability, work	44		data, science, analytics	28		data, science, analytics, graduate	27		analytics, graduate, scheme, leading, banking	27
6	knowledge	156		data, scientist	38		analytical, techniques, manipulating	28		science, analytics, graduate, scheme	27		graduate, scheme, leading, banking, group	27
7	ability	151		python, coding	35		techniques, manipulating, deriving	28		analytics, graduate, scheme, leading	27		advanced, analytics, machine, learning, solutions	24
8	python	116		knowledge, sql	34		data, python, coding	28		graduate, scheme, leading, banking	27		analytical, techniques, manipulating, deriving, insight	24
9	strong	111		coding, knowledge	33		science, analytics, graduate	27		scheme, leading, banking, group	27		techniques, manipulating, deriving, insight, data	24
10	computer	109		learning, solutions	30		analytics, graduate, scheme	27		manipulating, deriving, insight, data	26		manipulating, deriving, insight, data, python	24
11	ai	106		manipulating, deriving	30		graduate, scheme, leading	27		advanced, analytics, machine, learning	25		deriving, insight, data, python, coding	24
12	job	105		deep, learning	30		scheme, leading, banking	27		analytics, machine, learning, solutions	24		insight, data, python, coding, knowledge	24
13	analytics	101		statistical, modeling	29		leading, banking, group	27		techniques, manipulating, deriving, insight	24		python, coding, knowledge, sql, statistical	22
14	techniques	84		data, engineering	29		analytics, machine, learning	26		deriving, insight, data, python	24		knowledge, analytical, techniques, manipulating, deriving	21
15	analytical	77		advanced, analytics	29		manipulating, deriving, insight	26		insight, data, python, coding	24		coding, knowledge, sql, statistical, modeling	16
16	understanding	77		science, analytics	28		deriving, insight, data	26		coding, knowledge, sql, statistical	22		knowledge, sql, statistical, modeling, techniques	16
17	engineering	76		data, analytics	28		advanced, analytics, machine	25		knowledge, analytical, techniques, manipulating	21		data, engineering, data, analytics, visualization	13
18	include	72		cloud, environments	28		insight, data, python	24		knowledge, sql, statistical, modeling	16		engineering, data, analytics, visualization, data	12
19	work	70		analytical, techniques	28		knowledge, sql, statistical	22		sql, statistical, modeling, techniques	16		data, analytics, visualization, data, science	12
20	research	69		techniques, manipulating	28		statistical, modeling, techniques	22		data, engineering, data, analytics	14		scheme, leading, banking, group, knowledge	12

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Word (length 1)	Count		Phrase (length 2)	Count		Phrase (length 3)	Count		Phrase (length 4)	Count		Phrase (length 5)	
2	learning	82		artificial, intelligence	55		natural, language, processing	15		natural, language, processing, nlp	6		introduction, natural, language, processing, nlp	2
3	systems	65		machine, learning	48		learning, artificial, intelligence	7		artificial, intelligence, machine, learning	4		natural, language, processing, computer, vision	2
4	data	62		software, engineering	21		language, processing, nlp	6		machine, learning, artificial, intelligence	3		machine, learning, knowledge, representation, reasoning	2
5	computer	60		computer, science	21		knowledge, representation, reasoning	5		machine, learning, deep, learning	3		learning, knowledge, representation, reasoning, ethics	2
6	intelligence	59		natural, language	18		learning, deep, learning	4		introduction, natural, language, processing	3		knowledge, representation, reasoning, ethics, artificial	2
7	artificial	56		language, processing	17		foundations, artificial, intelligence	4		data, mining, text, analytics	3		representation, reasoning, ethics, artificial, intelligence	2
8	machine	52		computer, vision	17		logic, computer, science	4		data, science, machine, learning	3		reasoning, ethics, artificial, intelligence, deep	2
9	programming	46		deep, learning	14		artificial, intelligence, machine	4		neural, networks, deep, learning	2		ethics, artificial, intelligence, deep, learning	2
10	ai	42		data, science	13		intelligence, machine, learning	4		software, engineering, machine, learning	2		artificial, intelligence, deep, learning, data	2
11	software	41		data, mining	11		learning, natural, language	4		software, engineering, group, project	2		intelligence, deep, learning, data, mining	2
12	project	38		knowledge, representation	9		ai, machine, learning	3		artificial, intelligence, computer, systems	2		deep, learning, data, mining, text	2
13	science	38		humancomputer, interaction	8		learning, knowledge, representation	3		computer, science, software, engineering	2		learning, data, mining, text, analytics	2
14	computing	35		learning, artificial	8		machine, learning, artificial	3		artificial, intelligence, sustainable, development	2		data, mining, text, analytics, robotics	2
15	engineering	33		neural, networks	6		artificial, intelligence, data	3		data, mining, machine, learning	2		stage, computing, placement, preparation, software	2
16	introduction	29		processing, nlp	6		introduction, artificial, intelligence	3		machine, learning, research, methods	2		computing, placement, preparation, software, engineering	2
17	robotics	27		reinforcement, learning	6		systems, machine, learning	3		programming, logic, computer, science	2		machine, learning, pathway, supervised, unsupervised	1
18	language	26		learning, data	6		machine, learning, deep	3		software, engineering, computer, systems	2		learning, pathway, supervised, unsupervised, learning	1
19	processing	26		cyber, security	6		advanced, software, engineering	3		operating, systems, networks, internet	2		pathway, supervised, unsupervised, learning, classification	1
20	vision	24		big, data	6		introduction, natural, language	3		research, methods, computer, science	2		supervised, unsupervised, learning, classification, regression	1

Figure 5.3: The processed data in the output excel

format printed in the terminal is what will be used in the next (mapping) stage. The output is only shown in an excel sheet for illustration purposes.

5.2 Naive Bayes Machine Learning Model

Now the data from both datasets has been processed. The next tool is used for the mapping between these two datasets. The mapping will allow us to find any patterns or discrepancies between the two sets indicating a skills gap or the absence of it.

There are many ways of achieving a good mapping, this report uses the method of categorisation. All the skills collected can be broadly sorted into some categories. The gathered skills can be categorised with the number and titles of these categories being flexible. I, as the author of this report, have chosen 12 categories namely; programming & software development, machine learning, data science and analytics, engineering, business and management, cloud and technologies, soft skills (communication & collaboration), maths and statistics, research and development, industry specific knowledge, tools and technologies, ethics. These categories provide a good coverage over all the skills that have been collected so far. The software of the machine to do the sorting for us is described in the subsequent subsections.

5.2.1 Input

The input here is the output produced from the previous processing tool implemented. I.e. a list of tuples, each tuple having a skill/skill phrase and how many times it occurred in the input raw data i.e. [((‘programming’), 67), ((‘machine’, ’learning’,), 82), ...]. First we remove the commas from the first elements of each tuple in the list so that the muli-worded phrases look more like real skills now. This format is preferred by our naive bayes model. The result is stored in the inputArray (figure 5.4) list.

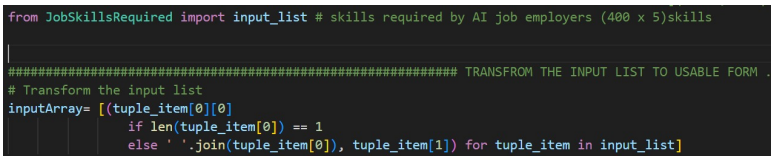

Figure 5.4: converting the data to a using-able format

5.2.2 Training set

Supervised machine learning models need a training set to train themselves before they are ready to sort unseen data into the predefined categories. Our training data is stored in the data variable. This data serves as the foundational training data, providing the necessary input for the naive bayes’ learning process. Inside this set, many skills that are commonly associated with each of the 12 (above mentioned) categories are stored in an array. The skills are stored in such a way that each has exactly one label. These skill-label pairs are what our classifier will learn from.

5.2.3 Dataframe

The next section seen in this file (figure 5.5) is where the naive bayes classifier is actually implemented. First a pandas dataframe is created from the data that we have provided. A pandas DataFrame is a two-dimensional, size-mutable tabular data structure with labelled axes (rows and columns). Each row represents a separate observation or data point. Each column represents a different variable or feature. Each row and column has an associated label, known as an index, which allows for easy referencing of data. The data frames stored in the df variable.

```
############################################################ TRAINING THE MACHINE ...
df = pd.DataFrame(data)

# Split the data into training and testing sets
train_data, test_data = train_test_split(df, test_size=0.1, random_state=12) # use either

# Create a pipeline with a CountVectorizer and a Multinomial Naive Bayes classifier
model = make_pipeline(CountVectorizer(), MultinomialNB())

# Train the model
model.fit(train_data['Skill'], train_data['Category'])

# Make predictions on the test set
predictions = model.predict(test_data['Skill'])

# Evaluate the model
print(classification_report(test_data['Category'], predictions))
```

Figure 5.5: converting the data to a using-able format

5.2.4 Splitting testing vs. training data

Not all of the data provided in the training data is used to train the model, some is also used for testing the model to give accuracy statistics on the model developed later on. The `test_train_split` function of the scikit-learn package previously discussed in this report does exactly that. In our implementation, 90% of the training data is used to train the model and the remaining 10% is used as the testing set. The `random_state` parameter is just set for reproducibility and making sure consistent results are obtained each time.

5.2.5 Pipelining

After splitting the data into training and testing sets, the next step in the implementation involves setting up a pipeline to streamline the process of text classification. This pipeline combines two essential components: the CountVectorizer and the Multinomial Naive Bayes classifier.

The CountVectorizer is the first thing taken into consideration in the pipeline. It transforms the text data into a numerical format that the machine learning model can understand. I.e. The text is converted into a matrix where each row represents a document and each column represents a unique word in the collection of documents. The values in the matrix represent frequencies of occurrence of each word in each document.

The Multinomial Naive Bayes classifier is the other input to the pipeline. This just sets the classifier to be used to the naive bayes classifier. This classifier will use Bayes' Theorem to classify data into our predefined categories. As previously discussed in this report, this classifier is known for its effectiveness in text classification tasks, particularly when dealing with datasets containing word counts.

5.2.6 Training the Model

Once we have set up the pipeline for text classification, the next crucial step is model training. This process involves teaching the model how to understand the relationship between the input skills and the corresponding categories in the training data. The "fit" method from the Scikit learn python library starts the training process. During the training process, the model adjusts its internal parameters based on the training data to optimise its predictive ability. These adjustments aim to minimise errors and maximise the model's accuracy in categorising unseen data.

5.2.7 Model testing

The next section of code is dedicated for testing the model that we have created. The `classification_report` method generates a classification report (figure 5.6) for the model created. The 10% of training data that was initially split by the `test_train_split` function is used for this testing phase. By printing the classification report, we can gain insights into how well the trained model performs for each of the 12 predefined categories. Here is a simple

	precision	recall	f1-score	support
Business and Management	1.00	0.50	0.67	2
Cloud and Technologies	0.67	0.33	0.44	6
Data Science and Analytics	1.00	0.75	0.86	4
Engineering	1.00	0.80	0.89	5
Industry Specific Knowledge	0.83	1.00	0.91	5
Machine Learning	0.64	1.00	0.78	7
Maths and Statistics	1.00	0.86	0.92	7
Programming & Software Development	0.67	0.67	0.67	12
Research and Development	0.62	1.00	0.77	5
Soft Skills (Communication and Collaboration)	1.00	0.89	0.94	9
Tools and Technologies	1.00	1.00	1.00	3
ethics	1.00	1.00	1.00	6
accuracy			0.82	71
macro avg	0.87	0.82	0.82	71
weighted avg	0.84	0.82	0.81	71

Figure 5.6: Classification report for the Naive Bayes model

explanation of each of the terms in the report:

- Precision - measures the proportion of correctly predicted instances of a particular category (class) out of all instances that were predicted to belong to that category. I.e. true positives.
- Recall - measures the proportion of correctly predicted instances of a particular category

out of all instances that actually belong to that category.

- F1-score - is the harmonic mean of precision and recall hence.
- Support - refers to the number of actual occurrences of each class in the testing data.

Our model, with an average precision of 87% and an average f1-score of 82%, can be considered highly reliable. We can count on this model to correctly classify skills 87% of the time.

5.2.8 Output

The final section of our implementation will actually use the model to perform the categorisation on the two datasets that we initially had. The code initialises counters for each category, setting them to zero. We then iterate over each tuple in the inputArray, which contains pairs of skills and their respective frequencies. For each skill, the model predicts its category using the trained text classification pipeline. Based on the prediction, the frequency of the skill is added to its designated category. Finally, all 12 categories and their counts are printed out for each of the two data sets. (figure 5.7) The results obtained from the two datasets will now be analysed to

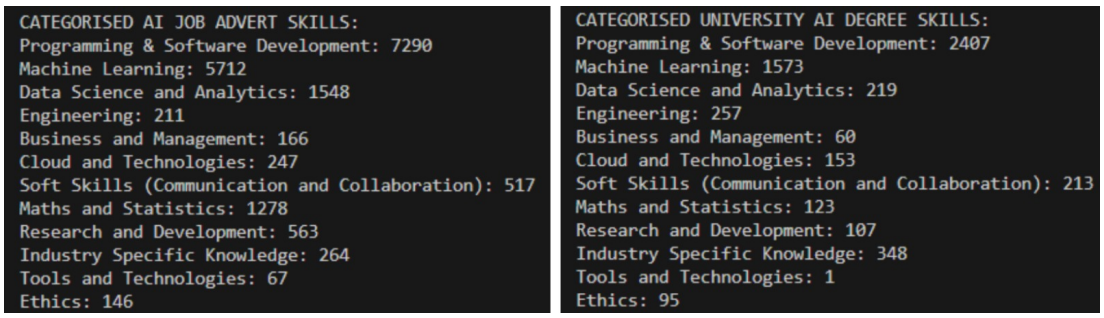

Figure 5.7: The final results obtained by the classification

see if any patterns or skills gap can be found.

Chapter 6

Evaluation

6.1 Skills gap evaluation

Here we will discuss the final results that have been gained from all the data we have collected, cleaned, sorted and mapped from the two datasets. Hence, seeing if a skills gap is actually present.

These are the 12 categories that were selected. Each has some most common skills that are related to it. (figure 6.1) The categories may be referred to by their abbreviations mentioned in this table later on in this report.

6.2 Results Produced

These are the results from the final sorting (figure 6.2). We initially considered many more job adverts than universities, this discrepancy can naturally be seen in the final frequencies. For each category, the job advert dataset has a higher frequency. When these exact frequencies are plotted on a bar graph, useful analysis can not take place because the frequencies are not being compared fairly. I.e. we collected skills required by around 150 job adverts and we collected skills acquired from around 30 universities, so the count of job advert skills is bound to always be higher. (figure 6.3)

6.2.1 Proportions

Instead we will be comparing the proportions that each category accounts for in a given dataset. We will use a pie chart (figure 6.4) to compare and visual each dataset for this report. (abbre-

Category	Common Skills Found	Abbreviation
Programming & Software Development	Languages Algorithms Version control	PROG
Machine learning	Supervised & unsupervised learning Neural networks Deep learning architectures	ML
Data science and Analytics	Data cleaning & processing Predictive modelling Visualisation	DATA
Engineering	Agile System Design Distributed Computing	ENG
Business and Management	Project Management presentation Cost-Benefit Analysis	BIZ
Cloud and Technologies	AWS Docker Big Data	CLOUD
Soft Skills (communication & collaboration)	Communication Collaboration Problem-Solving	SOFT
Maths and Statistics	Probability Linear Algebra Optimization	MATHS
Research and Development	Literature Review Prototyping Innovation	R&D
Industry Specific knowledge	Domain Expertise Industry Challenges Relevant Datasets	IND
Tools and Technologies	Jupyter Notebooks TensorFlow Pandas	TOOLS
Ethics	Fairness Transparency Bias Detection	ETH

Figure 6.1: The 12 categories, their examples and abbreviations

Category	Freq in Uni skills	Freq in Job skills
Programming & Software Development	2407	7290
Machine learning	1573	5712
Data science and Analytics	219	1548
Engineering	257	211
Business and Management	60	166
Cloud and Technologies	153	247
Soft Skills (communication & collaboration)	213	517
Maths and Statistics	123	1278
Research and Development	123	563
Industry Specific knowledge	348	264
Tools and Technologies	1	67
Ethics	95	146

Figure 6.2: The final results

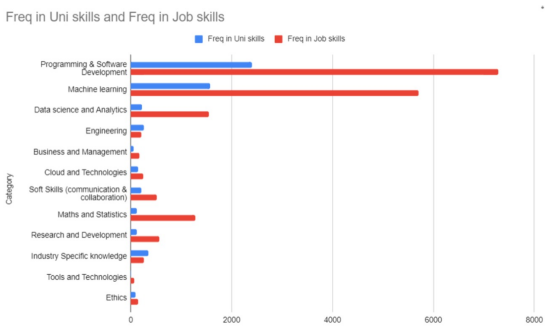

Figure 6.3: Bar chart of the final results

viations from above table are used in the chart). Some key findings:

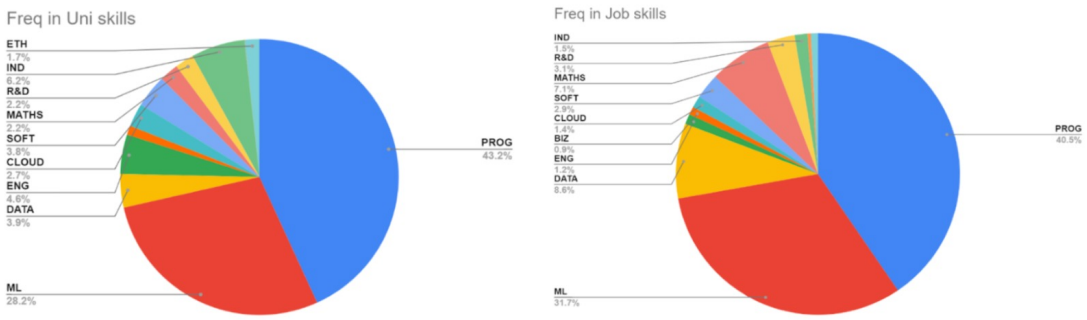

Figure 6.4: Pie chart of the final results

- Strong technical skills dominance - Both in university degrees and AI jobs, the dominant categories are ”Programming & Software Development” and ”Machine Learning.” This highlights the continued and significant demand for strong technical skills in the field of AI.
- Increased demand for data science in jobs - ”Data Science and Analytics” shows a substantial increase in importance in AI jobs compared to university degrees. This indicates the growing significance of data science skills in practical AI applications.
- Increased demand for data science in jobs - ”Data Science and Analytics” shows a substantial increase in importance in AI jobs compared to university degrees. This indicates the growing significance of data science skills in practical AI applications.
- Business and management emphasis in jobs - although business and management skills form a relatively low proportion of both datasets, they are more sought for in AI jobs.
- amounts of soft skills - Soft skills are roughly equally emphasised in both skill sets.
- Maths and statistics significance in jobs - a high level of quant based skills are demanded by job adverts, but the ratio of these being taught at university is much smaller, which is not as expected.
- Industry knowledge is more prominent at universities - surprisingly, while AI jobs still recognize the importance of domain-specific knowledge this category is more prominent in university degrees. We can infer that industoes usually prefer their employees to be well rounded rather than have specific domain knowledge.
- Ethics Recognition in both - both sets recognize the importance of Ethics, reflecting the growing awareness of ethical considerations in AI development and deployment.

6.2.2 The skills gap

In conclusion, technical skills are taught and also sought in both our dataset, no skills gap can be seen here. Similarly with ethics, which is also given equal importance in jobs and universities. However, a surprising finding is that jobs require students to have a strong analytical background, which sometimes students graduating from universities do not have. Another important finding is that recruiters want students to have a wide range of general knowledge and strong communication & collaboration skills rather than domain specific knowledge. Overall we have identified a skills gap, where recruiters do not expect specific domain knowledge, they instead are looking for more well rounded students with strong analytical skills and who can communicate well in the workplace.

6.2.3 Anomalies

Keep in mind that we have used a relatively small amount of data to obtain our results. On top of this, the ML model that was used has 87% precision, so some anomalies are inevitable. One obvious anomaly is the fact that only 1 skill was identified from the universities dataset that falls into the "tools and technologies" category.

6.3 University data set

6.3.1 Findings

The report uncovered a surprising finding when collecting the university dataset which was briefly mentioned before. There are many universities in the UK that do not offer any AI courses yet. Subsequent analysis further supported this fact. There are 106 (out of 166) universities in the UK that do not provide AI courses at all [20] [18]. Furthermore, In 2017 (i.e. 6 years ago) there were only 26 UK universities offering undergraduate courses in AI [16]. (figure 6.5.) Concluding from the given visual, 36% of universities do not even provide AI courses, so they are definitely contributing to the skill gap. Around 50% of universities do offer AI courses but have only been doing so for less than 6 years, so we can not be sure if they have good enough teaching standards as compared to the other courses that they offer. This fact may actually deter students from taking this course. Lastly, only around 16% of universities in the UK have been offering AI courses for more than 5 years.

Universities that will start to offer AI courses in the near future will obviously not be as experienced in this field as the current universities. So those AI courses will further widen our

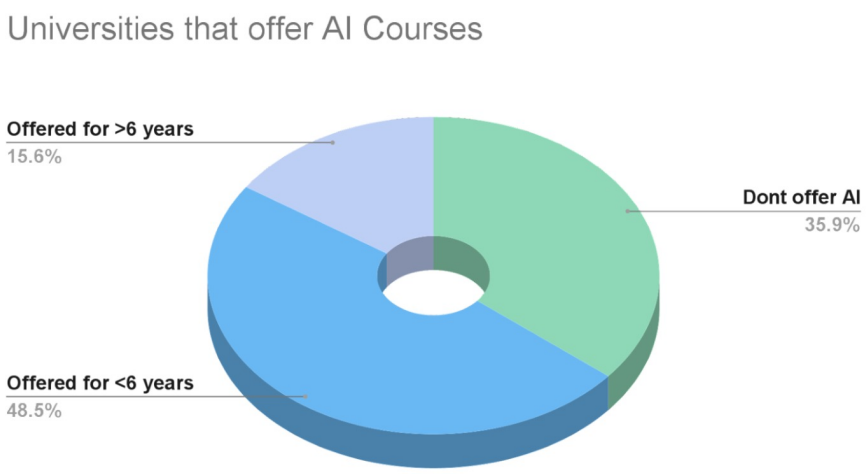

Figure 6.5: Proportion of Universities that offer AI courses

previously found skills gap.

6.3.2 Comparing with initial hypothesis

Before the start of this study, a smaller sample of universities was taken. This smaller set of skills were processed and sorted manually to get a better understanding of the structure and format of the data. This kind of tester set also allows us to identify patterns, inconsistencies, and potential challenges that may arise when dealing with larger datasets. The results found can be seen in figure 6.6. Coincidentally, our final results are actually very similar to this initial,

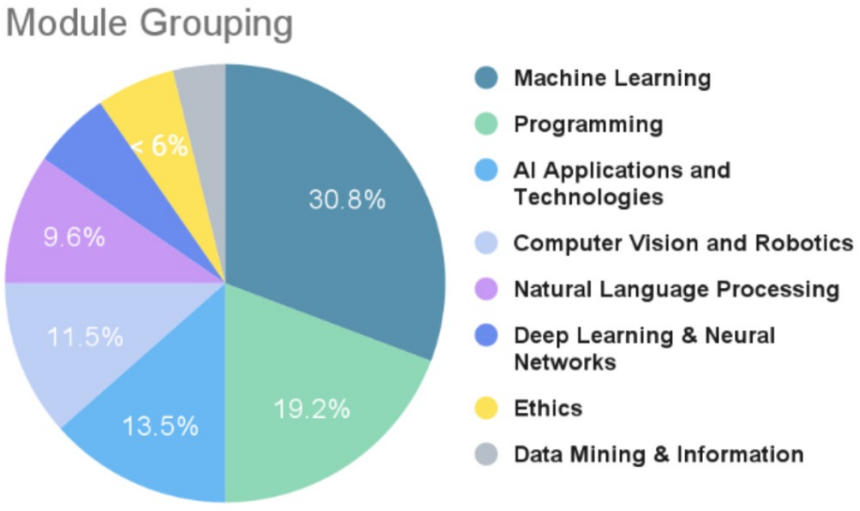

Figure 6.6: initial classification done with a smaller sample

smaller investigation that was taken. Meaning that the differences we noticed between what we expected and what we found in the smaller sample also showed up when we looked at a bigger

batch of data. This makes us feel pretty good about our results and the way we analysed the data. It also reminds us why it is so important to really dig into the data before jumping into the main study. By making sure our ideas hold up with both small and big sets of data, we feel a lot more sure about what we have discovered.

6.4 Job adverts dataset

6.4.1 Geographical location

Because of the increasing number of work from home jobs and people accepting longer travel times, the geographical location of universities and companies can often be overlooked, but it is still worth analysing. This diagram shows that the majority of the AI related degrees are taught in London. A report also mentions that UCL, Imperial and King's have comparatively larger AI and machine learning research groups, leading to the London cluster becoming much larger than the other ones on the diagram (figure 6.7) [11] [10]. This diagram (figure 6.7) coincides

Figure 6.7: Geographical Locations of most AI job vacancies 6.7

perfectly with the jobs dataset that we collected. Most of the job postings that were scrapped for this report were also for London based jobs and only a few were from other places around the UK. This indicates that the skills gap is not due to any geographical factors, rather it is the result of universities incorrectly catering to industry requirements i.e. the jobs are already

being posted where the most AI degrees are awarded anyway.

6.5 Limitations identified

We must recognise that no report is perfect in all aspects, this holds for our current report as well. Below we will discuss alternative designs and methods of evaluation that our report used.

6.5.1 Clay

Although the chrome extension, Clay, is an efficient and easy to use tool that served its purpose for our analysis very well, it has a drawback that only unveiled itself after a lot of implementation. Clay only works well for LinkedIn, when it is used to extract data from other job advertising websites, it often leads to erroneous results. Due to this drawback, this report has only considered jobs advertised on LinkedIn, and not other equally prominent websites such as glassdoor or indeed. So, if there was any bias, discrimination or randomness present in the adverts posted by LinkedIn, these will heavily impact our current findings. This research would be more credible and accurate if we had used more scrapers on a wider range of courses.

6.5.2 Classifier

The classification report shows that the precision level of the classifier is 87%. Although this is pretty high, it can still be improved. The classifier implemented in this report is a naive bayes classifier. But another classifier like a random forest could have provided us with more precision. The reason random forests were only considered, not used was because of the high complexity levels and the large amount of time that would be taken for implementation. Random forests usually give very accurate results because they use a bunch of decision trees to make predictions. These trees work together to avoid making mistakes, especially when dealing with lots of different factors, making random forests great for handling complex data.

Chapter 7

Legal, Social, Ethical and Professional Issues

7.1 Legal Issues

It is crucial to recognize that using tools to scrape data from LinkedIn may violate LinkedIn's terms of service. Regarding data scraping from platforms like LinkedIn, this report ensures compliance with relevant data privacy laws, including the General Data Protection Regulation (GDPR). Prior to scraping data, thorough examination of the terms of service of these platforms was conducted to prevent any violations; there is a limit of the number of jobs adverts that can be viewed on LinkedIn, at no point in this report was that limit exceeded. Additionally, the data collected from LinkedIn is anonymized to protect individual privacy.

7.2 Social Issues

All the software tools used in this report are only processing or manipulating already anonymous data. So, the project's focus of addressing the skills gap in AI education and industry needs will not in any way promote discrimination and bias towards a certain group or practice.

7.3 Ethical Issues

Measures have been taken to uphold ethical standards throughout the research process. No surveys were taken in the duration of this project, so there is no personal information that can

be disclosed. Even the large dataset that was used to train the naive bayes model does not contain any personally identifiable data.

7.4 Professional Issues

We must make sure that the results of this research are professionally accurate. The classification report showing the high level of precision is a good indication of the correctness of our findings. The accuracy of the result can alternatively be peer reviewed to make sure no professionally incorrect claims are made. Collaboration with stakeholders, including universities, industry experts, and policymakers, can enhance the relevance and impact of the study.

Chapter 8

Conclusion and Future Work

8.0.1 Conclusion

This project began by looking at current reports that have tried to identify a skills gap between AI jobs and universities AI degrees. We then explored all the sources available and picked the best for both of our datasets. Data from these sources was scrapped using clay and then the software we created was used to process this raw data. We then created and used a naive bayes machine learning model to sort the skills into categories to map the datasets onto each other.

We gained many small amounts of other information along the way, for example there is a large proportion of universities that do not currently teach AI modules. Also, the geographical location of AI related jobs and the places some AI students graduate from perfectly coincide with each other. Overall, the final mapping has been able to successfully identify a skills gap; employers do not expect specific domain knowledge in applicants, they instead are looking for more well rounded students with strong analytical skills and who can communicate well in the workplace. This is valuable information that can be used by universities that do already teach AI related degrees.

8.1 Future Works

There are several ways to extend our work in the future:

- Larger datasets - Future works could also more simply use the software provided here but use a larger data set. This will also give a more accurate and robust mapping. Anomalies also usually have less of an effect on larger datasets. Furthermore, larger datasets to get a broader view and also verify the results of this current report.

- Alternative classifiers - Experimenting with different classifiers may give us a different, more accurate mapping, leading to more accurate analysis. For example random forest or SVMs (which have been discussed in this report) would be good options to experiment with.

- Survey and Interviews: - future works can conduct surveys and interviews with employers, AI professionals, and university faculty to gather qualitative insights into the skills gap and the effectiveness of current AI degree programs in meeting industry needs. Qualitative data can complement quantitative analysis and provide deeper insights to the findings.

- Consider the perspectives of other stakeholders, such as AI students and recent graduates, in future analysis. Understanding their experiences and perceptions of the skills gap could provide additional insights into areas for improvement in AI education and training.

References

[1] Qs world university rankings for computer science and information systems 2023 — top universities. https://www.topuniversities.com/university-subject-rankings/computer-science-information-systems. (Accessed on 03/31/2024).

[2] Sam Attwood and Ashley Williams. Exploring the uk cyber skills gap through a mapping of active job listings to the cyber security body of knowledge (cybok). In *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, pages 273–278, 2023.

[3] Bardeen.ai automations. How to do linkedin data scraping — magical. https://www.getmagical.com/blog/linkedin-data-scraping#:~:text=The%20platform%20uses%20algorithms%20to,for%20scraping%20data%20on%20LinkedIn. (Accessed on 03/31/2024).

[4] Bardeen.ai automations. How to scrape linkedin data in 2024. https://www.bardeen.ai/posts/linkedin-scraping#:~:text=So%2C%20currently%2C%20LinkedIn%20data%20scraping,not%20encouraged%20by%20the%20platform. (Accessed on 03/31/2024).

[5] UK Department for Education (DfE. New analysis shows over 7,600 students have enrolled on ai and data science courses to tackle digital skills gaps - office for students. https://www.officeforstudents.org.uk/news-blog-and-events/press-and-media/new-analysis-shows-over-7-600-students-have-enrolled-on-ai-and-data-science-courses-to-tackl (Accessed on 03/31/2024).

[6] Adrian Gardiner, Cheryl Aasheim, Paige Rutner, and Susan Williams. Skill requirements in big data: A content analysis of job advertisements. *Journal of Computer Information Systems*, 58(4):374–384, 2018.

[7] Daniel Glez-Peña, Anália Lourenço, Hugo López-Fernández, Miguel Reboiro-Jato, and Florentino Fdez-Riverola. Web scraping technologies in an api world. *Briefings in bioinformatics*, 15(5):788–797, 2014.

[8] UK Gov. 9 key findings from understanding the uk ai labour market: 2020 report - gov.uk. `https://www.gov.uk/government/publications/understanding-the-uk-ai-labour-market-2020/9-key-findings-from-understanding-the-uk-ai-labour-market-2020-report`. (Accessed on 03/31/2024).

[9] Nickky At Gradulance. Postgraduate conversion courses in data science and artificial intelligence - office for students. `https://www.officeforstudents.org.uk/advice-and-guidance/skills-and-employment/postgraduate-conversion-courses-in-data-science-and-artificial-intelligence/`. (Accessed on 12/04/2023).

[10] Professor Dame Wendy Hall and Jérôme Pesenti. Clusters. `https://imactivate.com/clusters/?options=true&datagroup=Artificial%20Intelligence&location=null`. (Accessed on 04/01/2024).

[11] Professor Dame Wendy Hall and Jérôme Pesenti. Growing the artificial intelligence industry in the uk. 2017.

[12] Saram Han and Christopher K Anderson. Web scraping for hospitality research: Overview, opportunities, and implications. *Cornell Hospitality Quarterly*, 62(1):89–104, 2021.

[13] Moaiad Ahmad Khder. Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing & Its Applications*, 13(3), 2021.

[14] Phoebe Li, Robin Williams, Stephen Gilbert, Stuart Anderson, et al. Regulating artificial intelligence and machine learning-enabled medical devices in europe and the united kingdom. *Law, Technology and Humans*, 5(2):94–113, 2023.

[15] Jihad S Obeid, Matthew Davis, Matthew Turner, Stephane M Meystre, Paul M Heider, Edward C O'Bryan, and Leslie A Lenert. An artificial intelligence approach to covid-19 infection risk assessment in virtual visits: A case report. *Journal of the American Medical Informatics Association*, 27(8):1321–1325, 2020.

[16] Government of UK. Ai sector deal - gov.uk. https://www.gov.uk/government/publications/artificial-intelligence-sector-deal/ai-sector-deal. (Accessed on 03/31/2024).

[17] Charu Sarin. Analyzing skill gap between higher education and employability. *Research Journal of Humanities and Social Sciences*, 10(3):941–948, 2019.

[18] UK Gov HE sector. 113 institutions offering artificial intelligence (ai) courses in the uk. https://www.hotcoursesabroad.com/study/training-degrees/uk/artificial-intelligence-ai-courses/loc/210/cgory/cb.31-4/sin/ct/programs.html#:~:text=106%20Universities%20in%20the%20UK,Intelligence%20(Ai)%20degrees%20and%20courses&text=Are%20you%20looking%20for%20Artificial,online%20or%20distance%20learning%20options. (Accessed on 03/31/2024).

[19] UK Gov HE sector. 9 key findings from understanding the uk ai labour market: 2020 report - gov.uk. https://www.gov.uk/government/publications/understanding-the-uk-ai-labour-market-2020/9-key-findings-from-understanding-the-uk-ai-labour-market-2020-report. (Accessed on 02/08/2024).

[20] UK Gov HE sector. How can you study in the uk? https://www.timeshighereducation.com/student/advice/everything-you-need-know-about-studying-uk. (Accessed on 03/31/2024).

[21] UK HE Gov sector. Quantifying the uk data skills gap - full report - gov.uk. https://www.gov.uk/government/publications/quantifying-the-uk-data-skills-gap/quantifying-the-uk-data-skills-gap-full-report#:~:text=The%20government%20is%20already%20taking,2%2C500%20graduates%20over%203%20years. (Accessed on 12/04/2023).

[22] Oxfored Research Team. Expert comment: Ai demand is booming for the right skills and for the technology 'glue-guys' — university of oxford. https://www.ox.ac.uk/news/2023-10-09-expert-comment-ai-demand-booming-right-skills-and-technology-glue-guys. (Accessed on 12/04/2023).

[23] Basil P Tucker and Hank C Alewine. Solutions looking for problems? how humanities, arts, and social sciences can inform the space sector. *Space Policy*, page 101595, 2023.

[24] LinkedIn News UK. (4) linkedin jobs on the rise 2023: The 25 uk roles that are growing in demand — linkedin. https://www.linkedin.com/pulse/linkedin-jobs-rise-2023-25-uk-roles-growing-demand-linkedin-news-uk/. (Accessed on 12/04/2023).

[25] LinkedIn Team UK. (5) avoid linkedin bans: The only safe way to scrape profiles in 2023 — linkedin. https://www.linkedin.com/pulse/avoid-linkedin-bans-only-safe-way-scrape-profiles-2023-jordan-yusko-9x8cf/. (Accessed on 03/31/2024).

[26] Amit Verma, Kamal Lamsal, and Payal Verma. An investigation of skill requirements in artificial intelligence and machine learning job advertisements. *Industry and Higher Education*, 36(1):63–73, 2022.

[27] Amit Verma, Kamal Lamsal, and Payal Verma. An investigation of skill requirements in artificial intelligence and machine learning job advertisements. *Industry and Higher Education*, 36(1):63–73, 2022.

[28] Bo Zhao. Web scraping. *Encyclopedia of big data*, 1, 2017.

includeAppendices/SourceCode