JOB SEARCH AUTOMATION VIA A RECOMMENDER ENGINE AND MACHINE LEARNING

DATA SCIENCE SEMINAR (DATA 5000) – FINAL PROJECT REPORT

Itaf Omar Joudeh
Systems and Computer Engineering
Carleton University
Ottawa, ON, Canada
itaf.joudeh@carleton.ca

Zeyan Wang
SPROTT School of Business
Carleton University
Ottawa, ON, Canada
zeyan.wang@carleton.ca

ABSTRACT

Employment processes can be very frustrating and time consuming. We are aiming to automate job searches in order to improve the efficiency of employment processes. An efficient employment process requires that only qualified candidates apply to jobs. We plan on achieving this by building a website that leverages Application Programming Interfaces (APIs) and machine learning to match job-candidate pairs and recommend jobs to job seekers. In this work, we carried out the data collection, visualization and exploration, cleansing, mining, and analysis phases of a data science project. We acquired a data set of 19,001 job posts from Kaggle, and collected a data set of 32 candidates through a job search questionnaire. We performed both unsupervised and supervised machine learning to arrive at promising results.

KEYWORDS

Employment, Employers, Employees, Jobs, Job Seekers, Job Search Automation, Recommender Engine, Job Posting Boards, Application Programming Interfaces (APIs)

1. INTRODUCTION

1.1. PROBLEM STATEMENT

Employment processes can be very frustrating and time consuming. The average candidate spends months surfing the web for his/her dream job. He/she sends hundreds of applications while receiving very few or no feedback. On the other hand, employers/recruiters receive thousands of applications from qualified as well as unqualified candidates per job post. There are many job posting boards such as LinkedIn, Indeed, CareerBuilder, and Glassdoor that exist today, making employment processes more sophisticated. Employers might post on none or some of these boards, but not on others. This, in turn, forces job seekers to visit more boards to search and apply for jobs.

1.2. MOTIVATION

Having been job seekers at some point in our lives, we know what it is like to be one. We understand the stress, and the patience that come with it. Therefore, we are aiming to simplify and accelerate such processes. Our motivation is to automate job searches in order to increase the efficiency of employment processes for both job seekers and employers. An efficient employment process leads to more employment, less applications, less time, and less effort. This also means that only interesting and matching job posts will be recommended to job seekers, and only qualified candidates will apply to specific jobs.

1.3. OBJECTIVES

Our ultimate goal is to design a website that gathers information from the various job posting boards through Application Programming Interfaces (APIs) and uses machine learning to recommend jobs to job seekers. The five main objectives of this project are:

- data collection, processing, and analysis;
- building a recommender engine that sifts through job posts and candidate profiles to recommend jobs;
- implementing a machine learning algorithm to optimize the results of job recommendations;
- developing a website to enable job seekers to view job posts from multiple job posting boards in one place; and
- validating and verifying the obtained results by testing the website.

Only the first three objectives were accomplished within the given time frame (January – April). However, we will continue to work on the remaining two objectives on our free time in the future. Figure 1 demonstrates an overview of our proposed solution.

The recommender engine takes candidate profiles and job posts, and processes them, accordingly, to find similarities and affinities between job posts and candidate profiles. Lastly, it generates job-candidate pairs, where each pair is associated with a decision variable (*i.e.* response) indicating whether they match or not. The recommender engine then utilizes machine learning to recognize the patterns within job-candidate pairs and their responses. The jobs that match the profile of the candidate will be recommended to him/her. In our future work, we will extend the recommender engine by enabling it to acquire information on candidates' history using actions such as job views and/or applications.

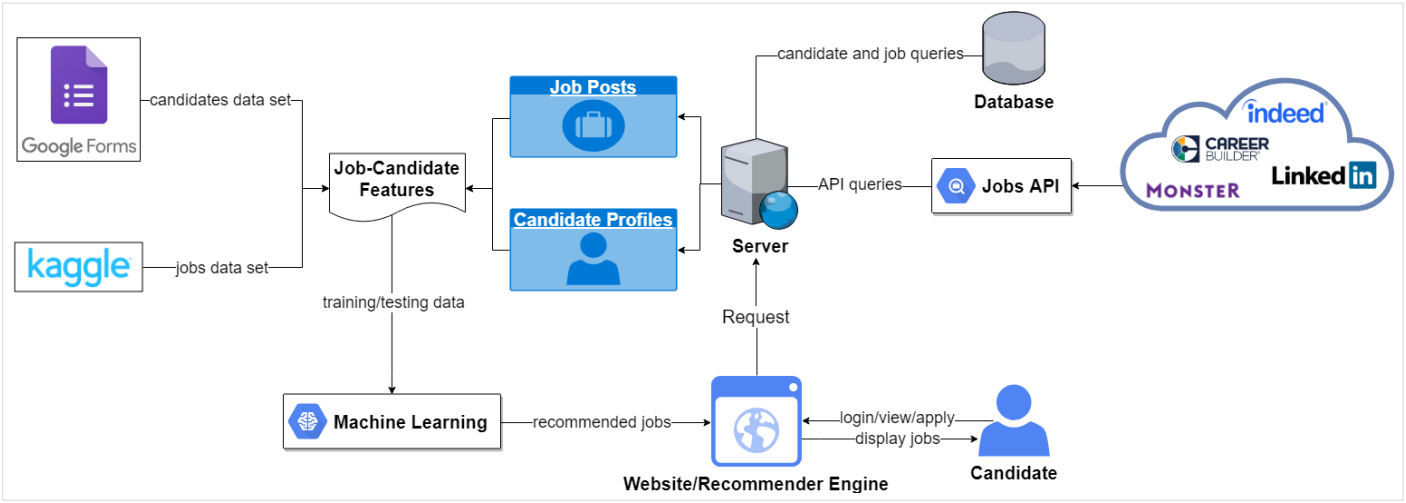


Figure 1: Proposed Solution

In the remainder of this report, we will discuss our methodology, methods, evaluation and validation of results, and the implications of this project. We will use the terms attributes, features, predictors, and variables alternatively in this report.

2. Research Questions

Our work addresses a number of research questions, which include, but are not limited to, the following:

- Can job searches be automated?
- Can jobs and candidates be matched?
- Would it be possible to increase the efficiency of employment processes?
- Can we measure a person's career interests?
- What attributes could represent a candidate's qualifications and career interests?
- What attributes could represent an employer's job offer and requirements?
- What attributes show that a candidate's qualifications fit an employer's needs for a job?
- How can we apply machine learning to solve this problem?
- Are organizations willing to supply information about their employment history and job offers?
- Do job posting boards provide APIs to access job posts on their databases?

At the beginning, we did a quick study on the feasibility of the project by performing a research on the existing job posting boards and what they offer. We investigated Indeed, LinkedIn, Career Builder, Monster, and Glassdoor. We found that Indeed, LinkedIn, Career Builder, and Monster all provide job APIs to access and query job posts from their databases. The structure of the returned query responses varies from one API to the other, but such variations could be worked around. Nonetheless, we found that Indeed is similar to what we would like to accomplish in this project, with the main difference being the job search automation part. Indeed still requires candidates to type keywords and to run the search themselves, which could sometimes be tricky as candidates may not have clear intentions in mind. Our findings convinced us that our objectives are achievable and encouraged us to move forward.

3. Methodology and Methods

One way to approach this problem is by determining if a candidate would be a good overall match for a job. Thus, we need data that tell us whether a candidate would find a job to be a good match and/or whether the employer thinks they are a good match. This can be established by asking the candidates to specify prospective job(s), and by rating their background, experience, and skills based on job requirements. We followed a framework that consists of five stages: data collection, data cleansing, data visualization and exploration, data mining (processing and feature extraction), and finally, data analysis (statistics and machine learning). Figure 2 outlines our methodology.

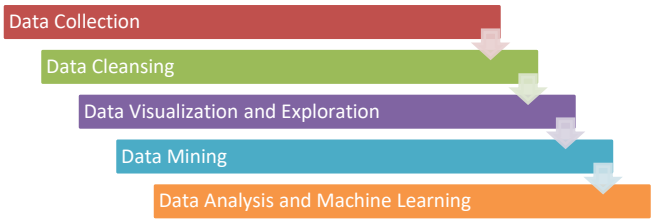


Figure 2: Overview of Methodology

We chose to operate in R because it is a free software environment for statistical computing and graphics. We also thought that this would give us a great opportunity to learn a new programming language. On the downside, R could be slow when dealing with big data sets.

3.1. Data Collection

Finding the right data was a huge challenge, given the lack of available data on the topic. Therefore, we proposed a Google Forms survey to collect data about potential candidates [1]. Table 1 illustrates the per section questions of

our survey. The survey asked questions about a candidate's age, gender, education, availability, preferences, and qualifications. Age and gender data were collected for statistical purposes (*i.e.* understanding the diversity in the data). They were not used in our job recommendations as they are irrelevant.

Section	Questions
Personal	• Please specify your age range. • What is your gender?
Education	• What is your highest degree? • What is your major? • If applicable, what is your minor/specialization?
Availability	• What is your current employment status(es)? • If you selected student, when is your anticipated graduation date? • How active are you on your job search? • If you selected actively or casually looking, when are you available to start?
Preferences	• What is your desired job title? • What employment types are you interested in? • What is your preferred work location (city)? (state/ province)? (country)? • What industries are you interested in? • What job functions are you interested in?
Qualifications	• What is your experience level? • Please list your top three hard (*i.e.* technical) skills (*e.g.* programming languages, software tools, project management). • Please list your top three soft (*i.e.* interpersonal) skills (*e.g.* teamwork, communication, leadership).

Table 1: Job Search Questionnaire

Please refer to Appendix A to view chart summaries of candidate responses, directly obtained from Google Forms [1]. As a result of the survey, we collected data about 32 different candidates, 22 of which are females, and 10 are males. Most candidates are unemployed, business or engineering students, between the ages of 16 and 40, at the entry level of their career. Such results introduced a majority bias in our data.

We fetched a number of data sets of online job posts from Kaggle.com. Most of those data sets contain unwell-structured text [2], [3], or contain inadequate data as they are missing crucial information, which should be included in every job post [4]. Some only include small parts of job descriptions, locations, titles, dates, and email/website addresses [4]. One data set, advertised as a data set of jobs data for recommender systems [2], is in French, and this would have made it difficult to manipulate. As can be seen, one drawback of Kaggle is that it could be an unreliable source of data, where anyone can create an account and post on it.

After several attempts, we found a decent data set that contains adequate data about 19,001 online job posts from 2004 to 2015 [5]. This data set comes from the Armenian human resource portal, Career Center, and is scrapped from their Yahoo! mailing group. The job posts within this data set follow a predefined structure; however, in some cases, the job poster did not fill some of the fields of the job posting out. This introduced missing values (*i.e.* NA's) in the data set. The data set is structured in terms of title, location, job description, job responsibilities, required qualifications, salary, application procedures, opening date, application deadline, and company information.

Since the locations of the jobs and our candidates differ, we ignored the location aspect in our data mining and analysis. The job posts are also outdated, while candidates data are very recent. Hence, we assumed that similar jobs will be posted around the same time of the year, and only referred to the months in our data mining and analysis.

3.2. DATA CLEANUP

The authors of the jobs data set cleaned it by removing posts that were not job related and/or had no structure [5]. On the other hand, we chose to clean the candidates data set manually, due to its small size. In more ideal scenarios, where the data set can grow much larger, this needs to be dealt with automatically. We replaced any empty blocks with "NA" and removed any random text. Empty blocks are the result of unrequired questions, whereas random text is the result of candidates' input to free-form, open-ended questions. In our survey, the last two questions about hard and soft skills are required free-form, open-ended questions. We made sure that all candidate responses for such questions follow the same format: three comma-separated skills. Cases where the candidates specified more than three skills, were divided into multiple records to accommodate for more data entries. Entries with the same set of skills ordered differently were reordered to help indicate that they are, in fact, the same. The final candidates data set included 39 records. Figures 3 and 4 show screenshots of the candidates and jobs data frames in R Studio, respectively. The data set of candidates has 20 variables, while the data set of jobs has 25 variables.

	Timestamp	Age.Range	Gender	Degree	Major	Minor.Specialization	Status	Graduation.Date	Activity	Availability	Job.Title	Employment.Types	City	Province.State	Country	Industries	Functions	Experience.Level	Hard.Skills	Soft.Skills
1	2/18/2019 14...	15-25	Female	Bachelor...	Comput...	NA	Graduat...	NA	Actively l...	Immediately	Programm...	Full-Time, Part-Time, ...	Ottawa	Ontario	Canada	Professiona...	Constructio...	Entry	JavaScript, H...	teamwork, ...
2	2/18/2019 21...	15-25	Female	Master o...	Comput...	Data Science	Student	8/30/2019	Casually l...	Immediately	Data Scien...	Full-Time, Part-Time, ...	Ottawa	Ontario	Canada	Professiona...	Science, Tec...	Intermediate	C, MATLAB, r...	willingness...
3	2/18/2019 21...	40-60	Male	Doctor o...	Mathe...	NA	Employed	NA	Not looki...	NA	Data Analyst	Full-Time, Part-Time	Ottawa	Ontario	Canada	Education S...	Education, ...	Professional	mathematics...	teamwork, ...
4	2/18/2019 21...	15-25	Female	Bachelor...	Comput...	NA	Student	12/20/2019	Actively l...	In a year	Business ...	Full-Time, Part-Time	Ottawa	Ontario	Canada	Agriculture,...	HR, Trainin...	Entry	SQL, Java, D...	teamwork, ...
5	2/18/2019 21...	15-25	Female	Bachelor...	Chemic...	Engineering Manage...	Student	8/20/2020	Casually l...	In a year	Chemist	Full-Time, Part-Time, ...	Ottawa	Ontario	Canada	Education S...	Science, Tec...	Entry	HYSYS, MATL...	teamwork, ...
6	2/18/2019 21...	15-25	Male	Master o...	Chemic...	NA	Student	5/31/2020	Casually l...	In a year	Research ...	Full-Time	Ottawa	Ontario	Canada	Professiona...	Science, Tec...	Entry	HYSYS, MATL...	adaptability...
7	2/18/2019 22...	25-40	Female	Bachelor...	Internat...	NA	Employed	NA	Not looki...	NA	Project Ma...	Full-Time, Permanent	Ottawa	Ontario	Canada	Education S...	Education, ...	Professional	project man...	leadership, ...
8	2/18/2019 22...	15-25	Female	Master o...	Busines...	NA	Student	12/20/2019	Actively l...	In 3 months	Coordinator	Full-Time	Ottawa	Ontario	Canada	Professiona...	HR, Trainin...	Internship	Microsoft O...	teamwork, ...
9	2/18/2019 22...	25-40	Male	Master o...	Busines...	NA	Student	02/06/2020	Actively l...	In a month	Business ...	Full-Time, Part-Time, T...	Ottawa	Ontario	Canada	Professiona...	Administrat...	Professional	Java, SFDC, ...	teamwork, ...
10	2/18/2019 22...	25-40	Male	Master o...	Busines...	NA	Student	02/06/2020	Actively l...	In a month	Business ...	Full-Time, Part-Time, T...	Ottawa	Ontario	Canada	Professiona...	Administrat...	Professional	JavaScript, c...	teamwork, ...
11	2/19/2019 8:...	25-40	Female	Master o...	Busines...	Other	Employed	NA	Not looki...	NA	Business ...	Full-Time	Ottawa	Ontario	Canada	Mining, Ma...	Administrat...	Intermediate	auditing, att...	optimism, r...
12	2/19/2019 9:...	25-40	Female	Bachelor...	Electric...	NA	Employed	NA	Casually l...	Immediately	Manager	Full-Time	Ottawa	Ontario	Canada	Professiona...	Manageme...	Intermediate	Coding, net...	Leadership,...
13	2/19/2019 13...	15-25	Female	Master o...	Busines...	Other	Student	09/10/2019	Actively l...	Immediately	HR Consul...	Full-Time	Dublin	California	China	Utilities	Consulting ...	Internship	English, Chi...	Teamwork, ...
14	2/19/2019 13...	15-25	Female	Master o...	Finance	Business	Student	12/20/2019	Actively l...	Immediately	Accountin...	Full-Time	Seattle	Ontario	Canada	Finance an...	Administrat...	Co-op	Language, s...	teamwork, ...
15	2/19/2019 15...	15-25	Female	Master o...	Finance	Economics	Employed	NA	Actively l...	Immediately	Analyst	Full-Time	Abu ...	Ontario	Canada	Retail Trade...	Business / ...	Intermediate	Analytics	Teamwork, ...

Figure 3: Candidates Data Frame

	job_id	jobpost	date	Title	Company	AnnouncementCode	Term	Eligibility	Audience	StartDate	Duration	Location	JobDescription	JobRequirment	RequiredQual	Salary	ApplicationP	OpeningDate	Deadline	Notes	AboutC	Attach	Year	Month	IT
1	0	AMERIA I...	05-Jan...	Chief ...	AMERIA Inv...	NA	NA	NA	NA	NA	NA	Yerevan, A...	AMERIA Investme...	- Supervises fina...	To perform this ...	NA	To apply for thi...	NA	26-Jan-04	NA	NA	NA	2004	1	FALSE
2	1	Internati...	07-Jan...	Full-ti...	Internation...	NA	NA	NA	NA	NA	3 months	IREX Arme...	NA	NA	- Bachelor's De...	NA	Please submit ...	NA	12-Jan-04	NA	The Intern...	NA	2004	1	FALSE
3	2	Caucasus...	07-Jan...	Coun...	Caucasus E...	NA	NA	NA	NA	NA	Renewabl...	Yerevan, A...	Public outreach a...	- Working with t...	- Degree in envi...	NA	Please send re...	NA	20 January...	NA	The Cauca...	NA	2004	1	FALSE
4	3	Manoff G...	07-Jan...	BCC S...	Manoff Gr...	NA	NA	NA	NA	NA	NA	Manila, Ph...	The LEAD (Local E...	- Identify gaps in...	- Advanced deg...	NA	Please send co...	NA	23 January...	NA	NA	NA	2004	1	FALSE
5	4	Yerevan B...	10-Jan...	Softw...	Yerevan Bra...	NA	NA	NA	NA	NA	NA	Yerevan, A...	NA	- Rendering tech...	- University deg...	NA	Successful can...	NA	20 January...	NA	NA	NA	2004	1	TRUE
6	5	Boutique...	10-Jan...	Sales...	Boutique "...	NA	NA	NA	NA	NA	NA	Yerevan, A...	Saleswoman will ...	NA	- Candidates sh...	NA	For further inf...	NA	01-Feb-04	NA	NA	NA	2004	1	FALSE
7	6	OSI Assis...	11-Jan...	Chief ...	OSI Assista...	NA	NA	NA	NA	NA	NA	Yerevan, A...	The Armenian Br...	NA	- University deg...	NA	For submissio...	NA	16 January...	NA	NA	NA	2004	1	FALSE
8	7	Internati...	13-Jan...	Non-...	Internation...	NA	NA	NA	NA	NA	6 months	IREX Arme...	NA	NA	NA	NA	To apply, pleas...	NA	16-Jan-04	NA	The Intern...	NA	2004	1	FALSE
9	8	Yerevan B...	13-Jan...	Assist...	Yerevan Bra...	NA	NA	NA	NA	NA	NA	Yerevan, A...	NA	- Coordinating t...	- University deg...	NA	Successful can...	NA	27 January...	NA	NA	NA	2004	1	FALSE
10	9	American...	13-Jan...	Progr...	American E...	NA	NA	NA	NA	NA	NA	NA	The incumbent a...	NA	NOTE: All applic...	NA	Interested can...	NA	26 January...	NA	NA	NA	2004	1	FALSE
11	10	Internati...	13-Jan...	Short-...	Internation...	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	For more infor...	NA	Applicatio...	NA	NA	NA	2004	1	FALSE
12	11	Internati...	13-Jan...	Non-...	Internation...	NA	NA	NA	NA	NA	6 months	IREX Arme...	NA	NA	NA	NA	To apply, pleas...	NA	16-Jan-04	NA	The Intern...	NA	2004	1	FALSE
13	12	Institute ...	13-Jan...	Chief ...	Institute fo...	NA	NA	NA	NA	NA	5 year POS...	Tashkent, ...	ISC seeks an exp...	NA	- Masters degre...	NA	Interested app...	NA	08-Feb-04	NA	Water Us...	NA	2004	1	FALSE
14	13	Food Sec...	14-Jan...	Com...	Food Secur...	NA	NA	NA	NA	NA	NA	Ijevan tow...	Food Security Re...	- Assist the Tavus...	- Higher Educat...	NA	Interested pers...	NA	Open until...	NA	NA	NA	2004	1	FALSE
15	14	Teleplus L...	14-Jan...	Gener...	Teleplus LLC	NA	NA	NA	NA	NA	NA	Yerevan, A...	NA	- Manage and co...	- Degree in Busi...	NA	If you believe t...	NA	Open	NA	NA	NA	2004	1	FALSE

Figure 4: Jobs Data Frame

3.3. DATA VISUALIZATION AND EXPLORATION

An R Notebook was designed to visualize and explore both jobs and candidates data. Although we will not cover all of the generated plots in this report, we will highlight a few of our observations. To view and run the designed notebook, please refer to the dataVisualizationAndExploration.Rmd module available on our GitHub repository at https://github.com/Itaf/JobSearchAutomation.

While visualizing the IT aspect of jobs data, we observed that the number of IT jobs is constantly increasing as we progress in time (see Figure 5). This is worth-mentioning for job seekers wanting to improve their skillset, as well as the young generations, who are still thinking about their upper education (*i.e.* postsecondary degrees). Also, this proves that the tech industry is on the rise, and that there is a need for employees in the IT sector.

Figure 5: IT Jobs

We encountered an issue with the inputs regarding the locations. The city, state/province, and country fields in our survey were separated and independent from one another, which allowed the candidates to enter wrong information. For example, as illustrated in Figure 6, a candidate selected Dublin to be in China, but it is in Ireland. We carry some of this responsibility for not enforcing the input style. However, our capabilities were limited to what Google Forms offer. This problem should be fixed when we build our website, where we will have more control over the data fields.

Another challenging issue that we noticed, is the fact that there are several manners in which one can describe something. In other words, different words can have similar meanings. This is where Natural Language Processing (NLP) algorithms can come into play. Due to the lack of our knowledge at the time of project execution, as well as time constraints, we did not perform NLP.

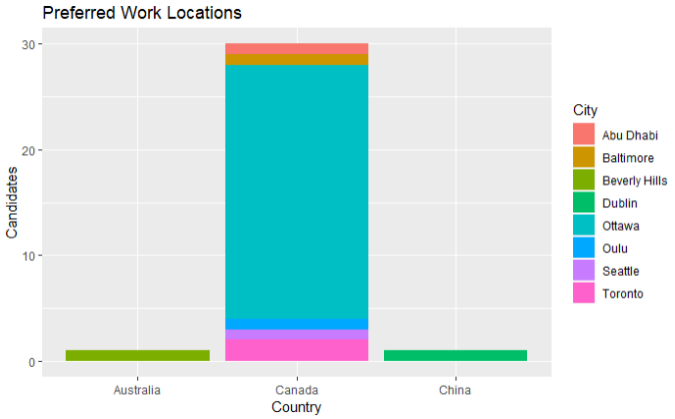


Figure 6: Wrong Inputs for Preferred Work Locations

3.4. DATA MINING

In terms of data mining, we have written an R script to process the candidates and jobs data sets. More specifically, the script:

- converts all text to lower case using space and/or comma separators;
- splits sentences and paragraphs into words;
- compares the words in each category of a candidate's data with the words in each job post;
- counts the number of word matches in a category; and
- finds the percent match between job-candidate pairs per category using the following formula:

$$\%\ Match\ (category) = \frac{\sum_1^n Word\ Matches}{n}$$

where n is the number of words in a category (*i.e.* degree, major, minor/specialization, hard skills, and soft skills). For example, let's say a candidate's major is computer systems engineering. For each of the words in the candidate's major, the script will look for matches in a job post using R's `grepl()` function. If it finds a match, the word match is set to 1; otherwise, it is set to 0. Finally, it will calculate the percent match by dividing the sum of word matches by 3. Thus, if the words "computer" and "engineering" were found, but the word "systems" was not, the percent match would be:

$$\%\ Match\ (major) = \frac{1+0+1}{3} = \frac{2}{3} = 0.66666666666$$

This was done differently for the employment status, activity, availability, and experience level categories. Candidates were rated on their need for a job based on their current employment status, where graduates and unemployed candidates were assigned the highest score of 1 and retired candidates were assigned the lowest score of 0. A student was assigned a score of 0.5, while a laid-off person was assigned a score of 0.75. An employed person was assigned a score of 0.25. The logic behind this is simple. If a person retires, he/she are less likely to work afterwards. If a person is a new graduate, he/she probably has tuition loans, such as loans to Ontario's Student Assistance Program (OSAP), to pay back or simply needs to gain experiences. If a person is unemployed, then he/she must work in order to afford living expenses. If a person is a student, he/she might need to work to be able to pay tuition fees as well as living expenses, but they are not in need as their parents or guardians would most likely help. Laid-off candidates would need to replace their job sooner or later; yet, they would have saved some money from their previous job. Employed candidates do not really need to find a job, but they might be interested in finding a better job. In cases where a candidate has more than one employment status, the average score over all statuses was calculated.

The candidates were also rated depending on whether they are actively, casually, or not searching for a job. Active candidates were given a score of 1, casually active candidates were given a score of 0.5, and non-active candidates were given a score of 0. Lastly, the date of availability to start at a new job position was determined by comparing the candidates' input with the current date. If the candidate selected immediately available, he/she was assigned an availability match rate of 1. If the month a candidate is available to start is the same as the month specified in a job post, the rate was set to 0.75 because they might need a prior notice. If the month a candidate is available to start is before the month specified in the job post, the rate was set to 0.50 because they might not be available anymore. If the month a candidate is available to start is after the month specified in a job post, the rate was set to 0.25 because they are not available yet. The eligibility of candidates for a job position was evaluated through their experience level. If their experience level matches the required level, they were given a score of 1; otherwise, they were given a score of 0.5 as they might still be eligible.

Our final data set consists of nine predictors, and 741,039 records of job-candidate pairs (39 candidates × 19,001 jobs). The job-candidate samples were then labelled with "No" or "Yes" according to the preferences of the candidates (*i.e.* desired job title, employment types, industries, and job functions). If the overall match rate over all preference categories (score) is greater than or equal to 25%, then the job and the candidate probably match. If the score is less than 25%, then they probably do not match. A "No" represents a decision to not recommend the job to the candidate, whereas a "Yes" represents a decision to recommend the job to the candidate. Figure 7 shows a screenshot of the data frame of job-candidate pairs in R Studio. The columns represent the nine predictors we described earlier and the response labels, and the rows represent samples of job-candidate pairs. To view and run the script, please refer to the dataMining.R module on our GitHub repository at https://github.com/Itaf/JobSearchAutomation.

Initially, we fell into the pitfall of defining labels based on the average match rate across all categories. We were also using the data regarding candidate preferences to extract features (*i.e.* calculated percent matches for desired job title, employment types, industries, and job functions). Thus, we were defining the labels from inside the data. This produced perfect classification accuracies and raised the concern of mislabelling the data.

	Degree.Match	Major.Match	Minor.Match	Need	Activity	Availability	Eligibility	Hard.Skills.Match	Soft.Skills.Match	Response
1	0.0000000	0.00	0.0	0.500	0.0	0.00	0.5	0.0000000	0.6666667	No
2	1.0000000	0.40	0.0	0.500	0.0	0.00	0.5	0.0000000	1.0000000	No
3	0.0000000	0.60	0.0	0.375	0.0	0.00	0.5	0.0000000	0.0000000	No
4	0.6666667	0.50	0.0	0.250	0.5	0.25	0.5	0.0000000	0.3333333	Yes
5	0.6666667	0.50	0.0	0.250	0.5	0.25	0.5	0.0000000	0.3333333	Yes
6	0.6666667	0.50	0.0	0.250	0.5	0.25	0.5	0.0000000	0.6666667	Yes
7	0.6666667	0.50	0.0	0.250	0.5	0.25	0.5	0.0000000	0.3333333	Yes
8	0.6666667	0.50	0.0	0.250	0.5	0.25	0.5	0.0000000	0.3333333	Yes
9	0.6666667	0.50	0.0	0.250	0.5	0.25	0.5	0.0000000	0.6666667	Yes
10	0.6666667	0.50	0.0	0.250	0.5	0.25	0.5	0.0000000	0.3333333	Yes
11	0.5000000	0.00	0.0	0.250	0.0	0.00	0.5	0.0000000	0.3333333	No
12	0.4000000	0.50	1.0	0.750	1.0	0.25	0.5	0.0000000	0.3333333	Yes
13	0.0000000	1.00	0.0	0.375	0.5	0.25	0.5	0.0000000	1.0000000	Yes
14	0.0000000	0.00	0.0	0.375	0.5	0.25	0.5	0.0000000	0.6666667	No
15	1.0000000	0.50	0.0	0.500	1.0	0.25	0.5	0.3333333	0.6666667	No

Figure 7: Data Frame of Job-Candidate Pairs

3.5. DATA ANALYSIS

In the world of data science, the "No Free Lunch" theorem applies. The theorem states that there is no one model that works best for every problem. Hence, we tried many different machine learning approaches in order to analyze and classify the data.

We attempted *k*-means clustering to generate groupings of candidates and jobs. As it is an unsupervised learning model, the data does not need to be labelled. Since most of the data in the data frames of candidates and jobs consist of text variables, we needed to transform them into numerical values. To do so, we **1)** converted them to lowercase, **2)** factorized them as categorical variables using R's `as.factor()` function, and **3)** coerced them to become interpretable as numbers using R's `as.numeric()` function. Dimensionality reduction could be very beneficial for saving resources such as memory and processing power. We chose to reduce the dimensionality of the data for the sole purpose of visualization, as 20 or 25 variables could become ugly to visualize in pairs. One method to reduce the dimensionality of a problem is Principal Components Analysis (PCA). The variability in the data can usually be explained by fewer derived variables, expressed as linear combinations of the original variables. To determine the principal components of the original variables, we used R's `princomp()` function. According to the performed PCA, it appeared that over 95% of the variance in both the data set of candidates as well as jobs can be explained with the principal components of eight variables. In preparation for the clustering analysis, we also considered scaling the data so that all variables are represented on the same scale. This was done using R's `scale()` function. One disadvantage of *k*-means clustering is that it could become very unpredictable as the produced clusters can sometimes be hard to interpret and understand.

Furthermore, we trained a range of supervised learning models on the data frame of job-candidate pairs to predict whether a candidate is a good match for a job or not, and in turn, recommend jobs to qualified candidates. In preparation for supervised machine learning, we split the data into training and testing sets. A training set of 80% of the data was sampled with replacement, whereas the remaining 20% of the data were held out for testing. This method is called holdout testing, and it is the simplest type of cross validation. Holdout testing is usually preferred when working with large data sets such as ours. One disadvantage of this method is that it requires the learner to be fully trained before testing it, making it tough to prevent overfitting without benchmarking the training performance as opposed to the testing performance. Another disadvantage of this technique is that it heavily depends on the way the data was split to start with. Consequently, the evaluation is greatly affected by the data points that end up in the training set.

The attempted supervised learners include linear regression, classification and probabilistic decision trees, naïve Bayes, neural networks, and ensemble models. Table 2 summarizes the leveraged supervised machine learning models.

Model	R Function(s)	Parameters
Linear Regression	`glm()` `predict()`	family = binomial prediction type = "response"
Decision Tree	`rpart()` `predict()`	prediction types = "class", "prob"
Naïve Bayes	`naïveBayes()` `predict()`	prediction type = "class"
Neural Network	`nnet()` `predict()`	size = 5 prediction type = "class"
Ensemble	multiple	Combinations of linear models, decision trees, naïve Bayes models, and neural networks (see Table 3)

Table 2: Supervised Machine Learning

We implemented four different ensemble models, which consist of combinations of the above supervised learning models. An advantage of ensemble modeling is that it is a way to run more powerful models as it relies on the predictions from multiple classifiers to achieve more accurate predictions. However, the performance of any ensemble model often constitutes the average of the performances of the involved models, no more or less. Table 3 summarizes our ensemble models.

Ensemble	Involved Models
Ensemble of different classifiers on ***training data***	linear regression, classification tree, probabilistic tree, naïve Bayes, neural network
Ensemble of different classifiers on four ***subsets of training data***	linear regression, classification tree, naïve Bayes, neural network
Ensemble of differing ensembles on four ***subsets of training data***	four ensembles of a linear regression model, classification tree, probabilistic tree, naïve Bayes model, neural network
Ensemble of similar ensembles on four ***subsets of training data***	ensemble of five linear regression models, ensemble of five classification trees, ensemble of five naïve Bayes models, ensemble of five neural networks

Table 3: Ensemble Models

The simplest ensemble model ensembles the predictions of the linear regression model, classification and probabilistic trees, naïve Bayes model, and the neural network to classify the data based on the mostly predicted class among the five of them. The other three models use subsets of the training data rather than training on the whole set. The training data set was partitioned into four subsets. Each subset contained all samples of the minority class ("Yes") and an equal proportion of the majority class ("No"). Supervised learners were then separately trained on those subsets of the training data. Afterwards, the results obtained by the different classifiers or ensembles were again combined by predicting the mostly predicted class among them. One advantage of this method is that it protects against class imbalance, while exploiting as much of the data as possible.

The next few sections show and discuss the results of the data analysis phase. To view and experiment with our data analysis approaches and results, please refer to the dataAnalysis.Rmd module on our GitHub repository at https://github.com/Itaf/JobSearchAutomation.

4. Evaluation and Validation

As aforementioned, we used k-means clustering to find similar jobs, and similar candidates. One would probably get different results each time they run k-means clustering for a specific value of k. So, we ran replicates to see what could happen. Since we did not know ahead of time how many candidate clusters and how many job clusters we will be using, we constructed clusters for various values of k. We ran 30 replicates for each number of clusters from 2 to 15. That is 420 calls to the clustering algorithm, R's `kmeans()` function, in total. For each clustering scheme, we tracked both the Davies-Bouldin (DB) index and the Within Sums of Squares (WithinSS) and kept them in memory. The DB index is measure that is used to determine the optimal number of clusters for the clustering schemes in the data. For a given data set, the optimal number of clusters is obtained by maximizing the DB index. The WithinSS is a measure of how similar observations are within each cluster, and how different they are from observations in other clusters. For a given data set, the optimal number of clusters is obtained when there is a sudden drop in the slope of the WithinSS curve such that adding more clusters won't cause as big of a decrease anymore.

For the data set of candidates, the optimal number of clusters according to the DB index and the WithinSS curve is 2. To see what would happen if we increase the number of clusters, we also plotted the results of k-means clustering with $k = 3$ and $k = 4$. Figures 8 through 12 demonstrate the results of our clustering analysis on the data set of candidates. The clusters in Figures 8 through 10 are represented in terms of two discriminant coordinates to distinguish between classes. The two discriminant coordinates are formed by making linear combinations of two or more variables from the data set of candidates. The clustering schemes of $k = 2$ and $k = 3$ look very good, with clear boundaries between the clusters. At $k = 4$, the clusters started to collide a little bit. These results could be easily explained by looking at how many different age groups, genders, employment statuses, or activity levels exist in our candidates data. If we look back at the data, we can find a maximum of two or three notable entries in each of these categories.

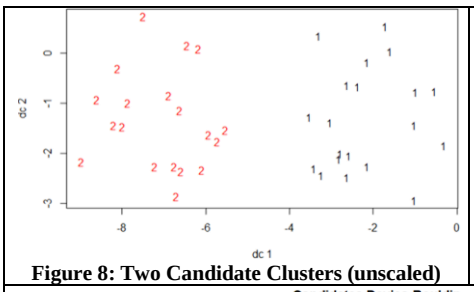


Figure 8: Two Candidate Clusters (unscaled)

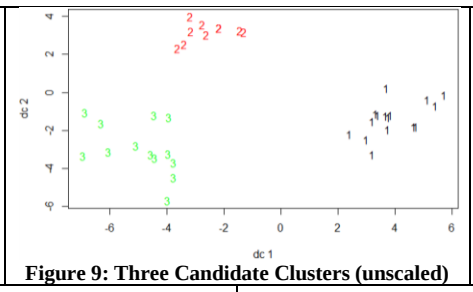


Figure 9: Three Candidate Clusters (unscaled)

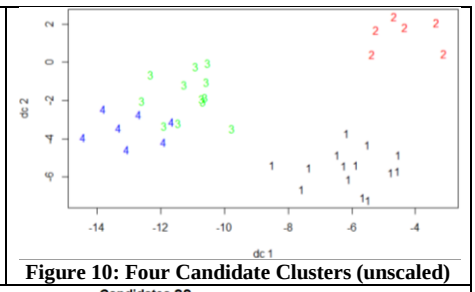


Figure 10: Four Candidate Clusters (unscaled)

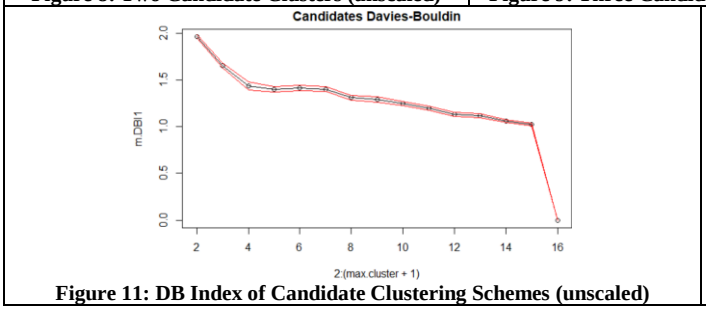


Figure 11: DB Index of Candidate Clustering Schemes (unscaled)

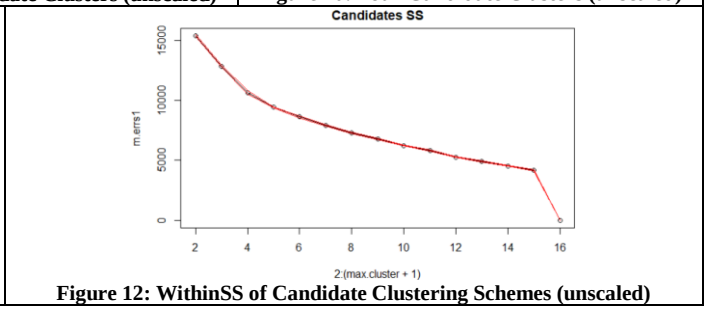


Figure 12: WithinSS of Candidate Clustering Schemes (unscaled)

For the data set of jobs, the optimal number of clusters according to the DB index and the WithinSS curve is 4. To see what would happen if we reduce/increase the number of clusters, we also plotted the results of k-means clustering with $k = 3$ and $k = 5$. Figures 13 through 17 demonstrate the results of our clustering analysis on the data set of jobs. The clusters in Figures 13 through 15 are represented in terms of two discriminant coordinates to distinguish between classes. The two discriminant coordinates are made by forming linear combinations of two or more variables from the data set of jobs. All three clustering schemes of $k = 3$, $k = 4$, and $k = 5$, do not seem to provide clear boundaries between clusters. The clusters are close to each other and hard to distinguish from one another. This indicates that the algorithm was not able to find k means that are far enough. This was also the case for all values of k up to 15.

Also, we tried to scale the data to see how the results would change. Scaling the data influenced the DB index and the WithinSS curve for both candidates and jobs data, but the resulting clustering schemes were similar in fashion. After scaling, the optimal number of candidate clusters became 3, while the optimal number of job clusters became 2.

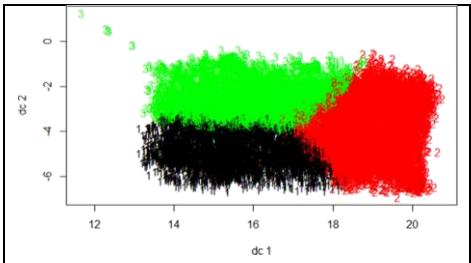


Figure 13: Three Job Clusters (unscaled)

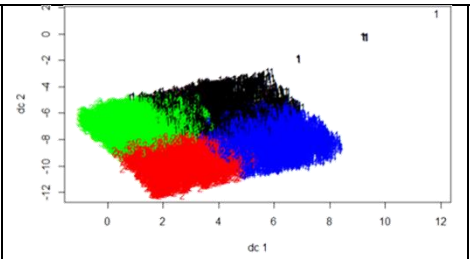


Figure 14: Four Job Clusters (unscaled)

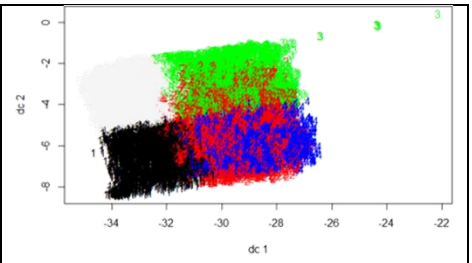


Figure 15: Five Job Clusters (unscaled)

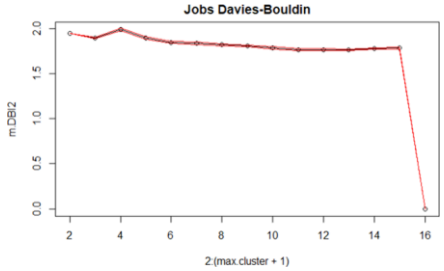


Figure 16: DB Index of Job Clustering Schemes (unscaled)

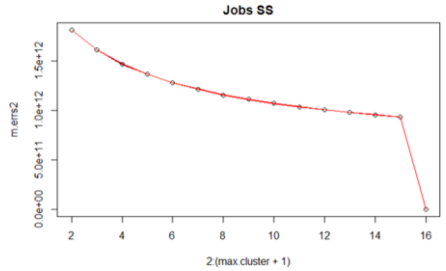


Figure 17: WithinSS of Job Clustering Schemes (unscaled)

The original data set of job-candidate pairs was divided into an 80% sample for training and 20% for testing. We trained a variety of models using our training set. We tested the models by predicting the responses for the testing set. Since the testing set includes new data points, such evaluation reflects the actual performance of the classifiers. We validated and evaluated the performance of the different supervised learning models by plotting confusion matrices and computing five performance metrics: accuracy, sensitivity, specificity, precision, and F1 score. Accuracy is a measure of correctness achieved in the overall prediction (*i.e.* out of all predictions, how many are predicted correctly). Sensitivity/recall is a measure of how many observations of the "Yes" class are predicted correctly. Specificity is a measure of how many observations of the "No" class are predicted correctly. Precision is a measure of correctness achieved in "Yes" predictions (*i.e.* out of the observations predicted as "Yes", how many are actually labelled "Yes"). The F1 score is a measure of effectiveness of classification in terms of the weighted average of sensitivity and precision. In our case, the "No" class is the negative scenario, and the "Yes" class is the positive scenario.

Initially, we only ran the binomial linear regression model, classification tree, probabilistic decision tree, naïve Bayes model, and neural network on the training set. However, the obtained confusion matrices suggested that there is a class imbalance in the data as the "No" class predominated. Table 4 shows a sample confusion matrix with evident class imbalance. This confusion matrix displays the predictions of a trained binomial linear regression model for the testing set.

		PREDICTED	
		No	Yes
ACTUAL	No	116626	498
	Yes	30927	425

Table 4: Sample Confusion Matrix showing Class Imbalance

To get a better understanding of the class imbalance, we generated a plot summarizing the responses within the training set (see Figure 18). The imbalance was, in fact, due to having more "No" cases than "Yes" cases, which is what we would expect given the nature of our data. First, the candidate-to-job ratio is 39: 19001. Secondly, a person has one (or few) areas of expertise, while jobs come from a wide-range of sectors.

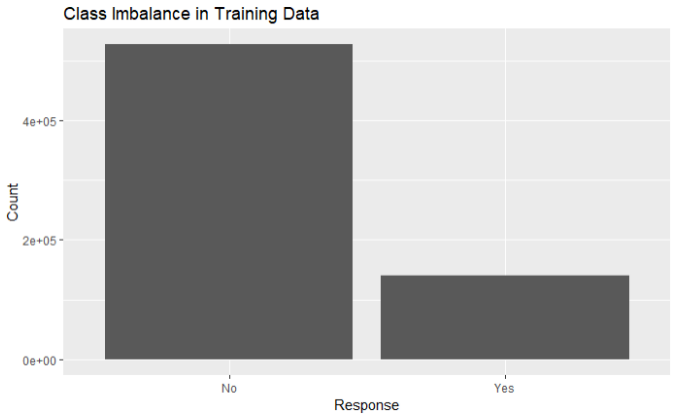


Figure 18: Responses of the Imbalanced Training Set

The training set was then balanced using the Random Over-Sampling Examples (ROSE) algorithm [6]. ROSE utilizes a smoothed-bootstrap approach to create a sample of synthetic data from the features space of minority and majority class observations. The new samples are drawn from a conditional kernel density estimate of the two classes. The data generated by oversampling techniques usually have repeated observations, whereas the data generated using undersampling possibly lose important information from the original data. On the other hand, the data generated using ROSE provide better estimates of the original data. Figure 19 summarizes the responses of the produced training set.

Another way to deal with class imbalance is by dividing the training set into subsets, where each subset contains an equal amount of "No" and "Yes" observations, as described in the previous section.

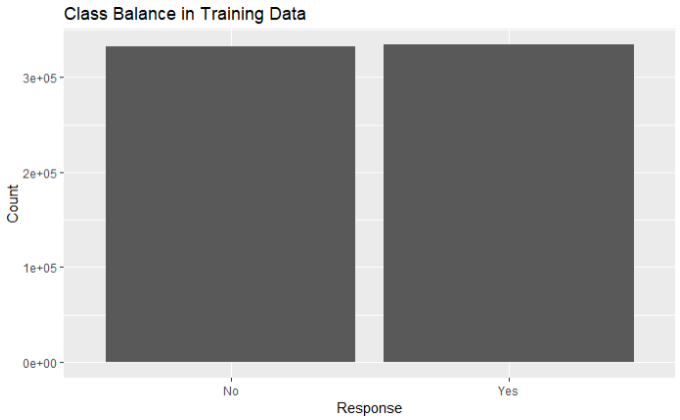


Figure 19: Responses of the Balanced Training Set

Recall that we attempted a total of eight supervised learning models. Figures 20 through 24 summarize the achieved performances by the eight models with and without class imbalance. In addition, the results include the performances of the models involved in ensemble models. Although we will only present the results achieved by an 80-20% split in the data, we actually tried to split the data into different amounts to see how the size of the training set would affect the achieved performances. In particular, the tested data splits include a 90-10%, 80-20%, 75-25%, 70-30%, and a 60-40% split. For a full list and/or to visualize some of our results, please refer to resultsClean.xlsx and resultsEvaluationAndValidation.Rmd on our GitHub repository at https://github.com/Itaf/JobSearchAutomation.

According to the accuracy metric, the learners achieved better performances with the existing class imbalance. However, accuracy is not a good measure for performance as it only indicates the ratio of correct predictions. If "No" is the majority class and the learner predicts all samples as a "No", then it would still get a high accuracy, because the actual label of most of the data is "No". Hence, accuracy does not reflect problems such as class imbalance. This also explains why the sensitivity for imbalanced classification is low, and the specificity is high.

With class imbalance, the highest achieved accuracy is 80% (classification tree, neural network, and the simplest type of ensemble models), sensitivity is 26% (probabilistic tree), specificity is 100% (classification tree, linear regression, and simplest ensemble model), precision is 81% (classification tree, and simplest ensemble model), and the highest achieved F1 score is 26% (probabilistic tree). After balancing the classes, the highest achieved accuracy is 70% (naïve Bayes, ensemble of naïve Bayes models, and simplest ensemble model), sensitivity is 67% (classification tree), specificity is 76% (naïve Bayes, and ensemble of naïve Bayes models), precision is 36% (simplest ensemble model), and the highest achieved F1 score is 45% (neural network, and ensemble of neural networks). In general, there is no one learner that always performs better than the others. There is a trade-off between the five metrics. If one metric is high, that is not necessarily the case for the other metrics.

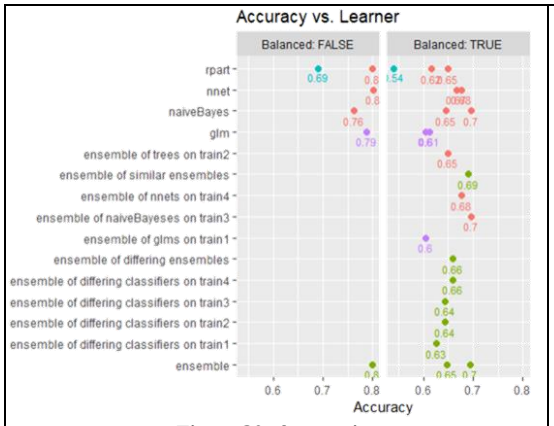


Figure 20: Accuracies

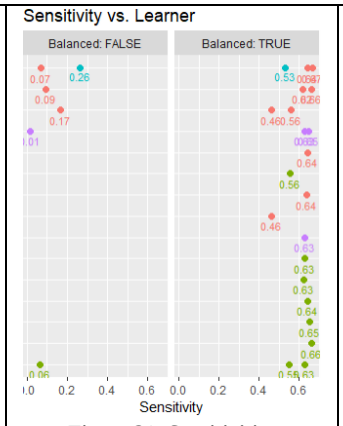


Figure 21: Sensitivities

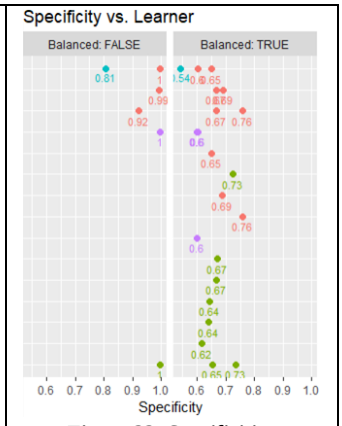


Figure 22: Specificities

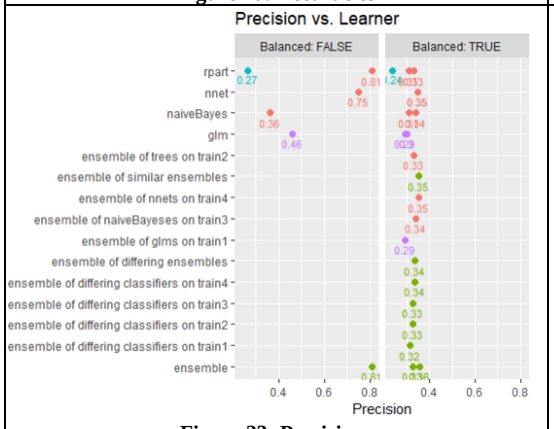


Figure 23: Precisions

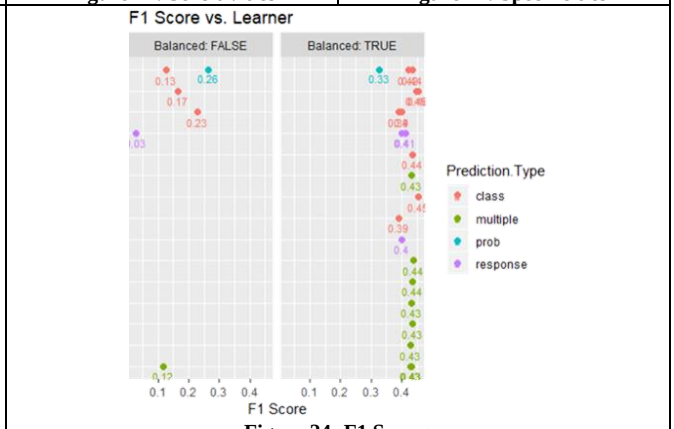


Figure 24: F1 Scores

5. Limitations, Challenges, Pitfalls

To reiterate, some of the limitations, challenges, and pitfalls present in this project are:

- Data Inadequacy – there is a lack of available data about job offers and career interests
- Majority Bias – most candidates are young female students from the school of business or engineering
- User Inputs – one cannot trust the inputs of the users as they can contain random text of any form and shape
- Natural Language – different words can have similar meanings due to diverse user inputs
- Mixed Data – the data have multiple variable types: numbers, dates, text
- Locations and Dates – the candidates are from Canada, while the jobs data set has outdated jobs from Armenia
- Data Misinterpretation – we, at some point, used the data on candidates' career interests to extract features and labelled the observations depending on the average of all features
- Class Imbalance – the data is associated with a class imbalance, where the "No" class prevails the "Yes" class
- Job APIs – available job APIs require permissions or partnership keys (*i.e.* access tokens), and the format of job posts differs depending on the used API

6. Implications

Our results imply that there is a potential to automate job searches, but there is still a long way to go. Before moving to the production stage and starting to build a website, one would want to ensure customer and client satisfaction. We spent quite a time on our machine learning component in hopes of improving the classification/prediction performance of our recommender engine. However, achieving accurate job recommendations would require massive amounts of data to cover as many scenarios as possible. Although the data set of job-candidate pairs that we used is big, it is not large enough to accommodate all possible proficiencies and expertise. It is, nevertheless, associated with a majority bias towards a specific group.

7. Business Plan

Our goal is to build a recommender engine which could automate the job search process and accelerate employment processes for both employees and employers. With the data set of candidates collected through the Google Forms survey, and of jobs collected from Kaggle, machine learning has been performed through several algorithms in R. This is the core part of the project, as it determines the job recommendations to the users (candidates). We intend to make accurate recommendations in order to accomplish what we promised in our slogan: "*more employment, less applications, less time, and less effort*." The product would then be the recommender engine, and our service is going to be the website which would be built using R's shiny package. The product would only be launched after all objectives are met.

The target customers would be job seekers. Since it is a website and the job postings are world-wide, it would have little or no geographic limitation. The prioritized alternatives would be university/college students who are in Canada. Based on the Strengths-Weaknesses-Opportunities-Threats (SWOT) analysis presented in Table 5, the positioning is abundant and efficient so as to differentiate ourselves with the boards which already exist. To reach out to our alternatives, the initial cycle of promotion would be done through social media. We would promote the website on LinkedIn, Facebook, Twitter, Instagram, and so on. At the early stages, the marketing of our product would rely on mouth-to-mouth suggestions and recommendations.

STRENGTHS	WEAKNESSES
• Combines the job postings from several boards • Leverages machine learning to match the jobs with the users • Free for the users	• Dataset is not large enough to achieve a high classification/ prediction performance • Insufficient financial foundation
OPPORTUNITIES	**THREATS**
• Increasing needs in market • Experience enhancement for the users • Scaling-up plan for future development • Collaboration with existing job posting boards	• Strong and experienced competitors • Competitor connections

Table 5: SWOT Analysis

The purpose is not to earn money but to simplify the job search for our users. Until now, we do not have any financial support. If the project goes well and we establish a relatively large amount of customer base, there might be some investors and companies who are interested in job search automations and recommender engines. Therefore, we would want to make connections with the companies who would like to post jobs exclusively on our website and receive candidate applications from our website. A scaling-up strategy would be planned in the future according to further developments in the project. Advertising might be considered, while we will not charge any fee to the job seekers to keep our initial goal.

8. Conclusion

This project is a pilot project for job search automation using recommender engines and machine learning. Many machine learning algorithms were tested. *k*-means clustering was used to create clusters of candidates and jobs, but it failed to do so for the jobs data set. The evaluation of the supervised learning results suggested that all algorithms should be perceived equally as there is a trade-off between the measured performance metrics. In conclusion, there is a great potential for job search automation; yet, there is still a lot to be done.

REFERENCES

[1] Itaf Omar Joudeh and Zeyan Wang. Job Search Questionnaire. Retrieved February 18, 2019 from https://forms.gle/FnpeDJULyJTcefr9A.

[2] Tondji LIonel. 2018. Jobs Data for recommender systems. (January 2018). Retrieved February 23, 2019 from https://www.kaggle.com/tondji/jobs-data-for-recommender-systems.

[3] PromptCloud. 2017. US jobs on Monster.com. (September 2017). Retrieved February 25, 2019 from https://www.kaggle.com/PromptCloudHQ/us-jobs-on-monstercom.

[4] PromptCloud. 2018. Canadian jobs on Eluta.ca. (June 2018). Retrieved March 3, 2019 from https://www.kaggle.com/PromptCloudHQ/canadian-jobs-on-elutaca.

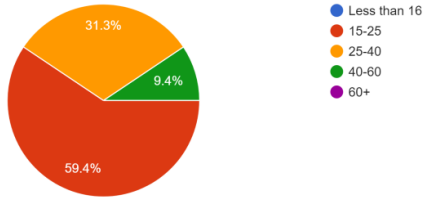
[5] Mad Hab. 2017. Online Job Postings. (April 2017). Retrieved March 5, 2019 from https://www.kaggle.com/madhab/jobposts.

[6] Nicola Lunardon, Giovanna Menardi, and Nicola Torelli. *R Package ROSE: Random Over-Sampling Examples (version 0.0-3)*. Università di Trieste and Università di Padova, Italy, 2013. Retrieved March 24, 2019 from http://cran.r-project.org/web/packages/ROSE/index.html.

Appendix A: Summary of Responses to the Job Search Questionnaire

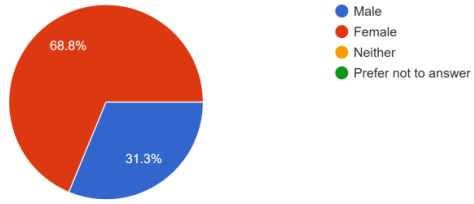
Please specify your age range.

32 responses



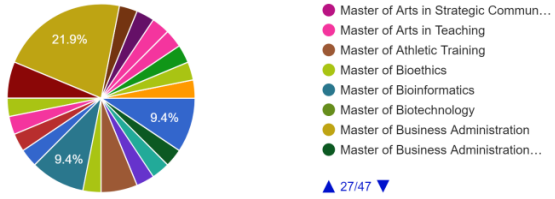
What is your gender?

32 responses



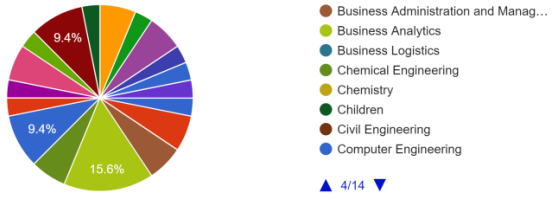
What is your highest degree?

32 responses



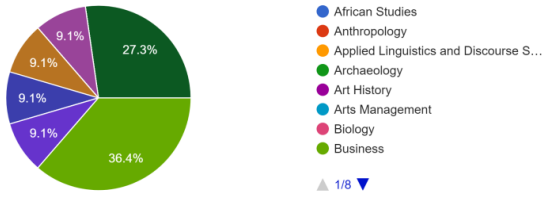
What is your major?

32 responses



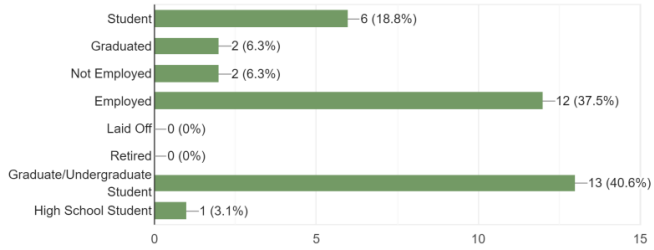
If applicable, what is your minor/specialization?

11 responses



What is your current employment status(es)?

32 responses

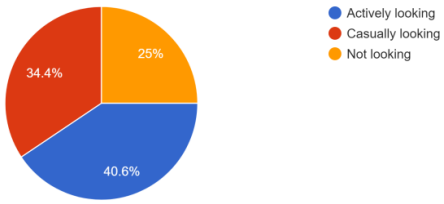


If you selected student, when is your anticipated graduation date?

20 responses

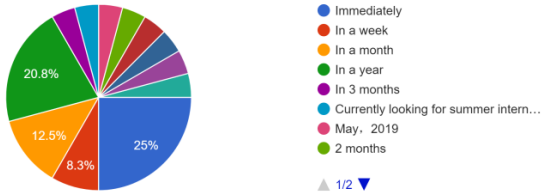
How active are you on your job search?

32 responses



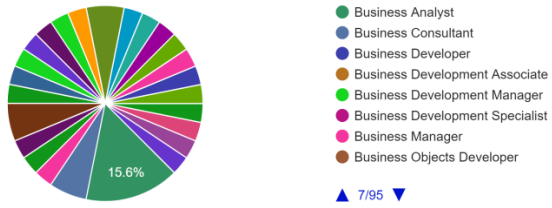
If you selected actively or casually looking, when are you available to start?

24 responses



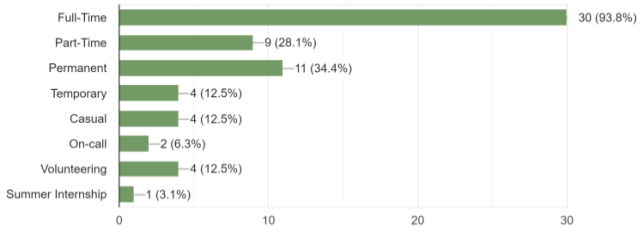
What is your desired job title?

32 responses



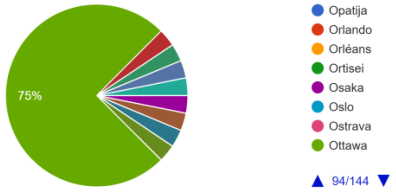
What employment types are you interested in?

32 responses



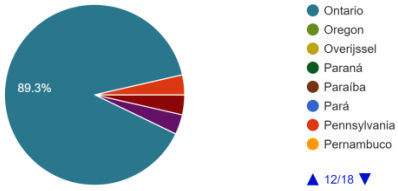
Where is your preferred work location (city)?

32 responses



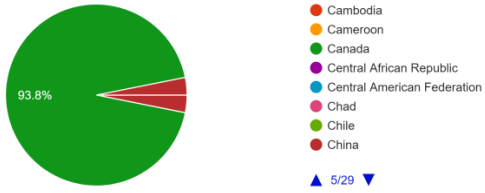
Where is your preferred work location (state/province)?

28 responses



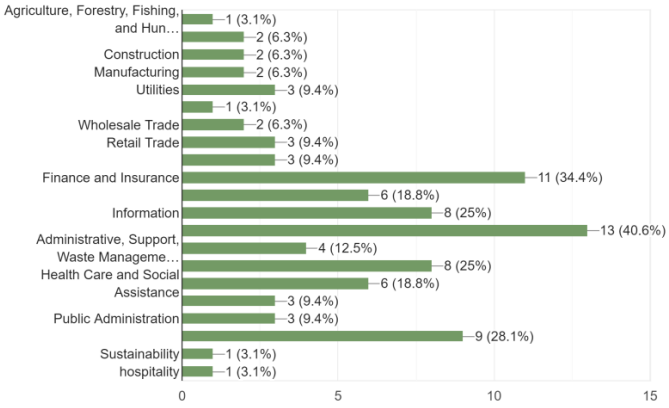
Where is your preferred work location (country)?

32 responses



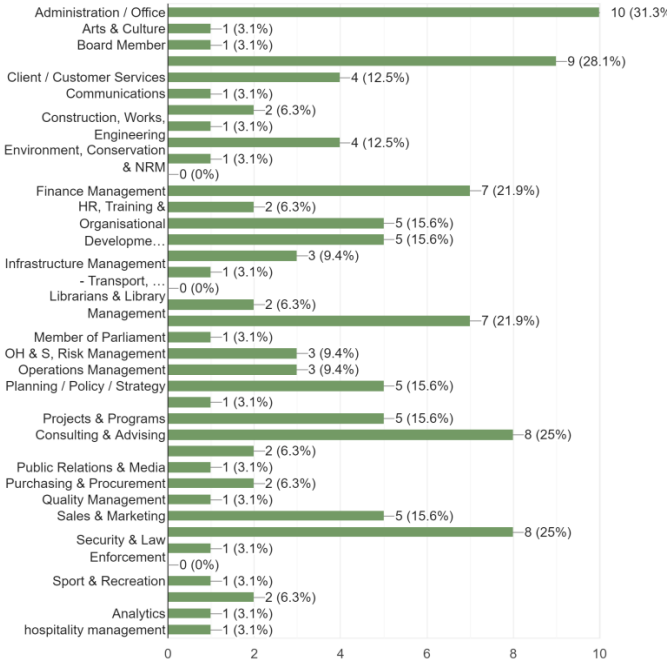
What industries are you interested in?

32 responses



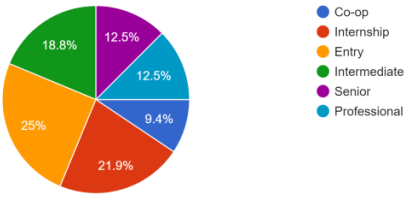
What job functions are you interested in?

32 responses



What is your experience level?

32 responses



Please list your top three hard (i.e. technical) skills (e.g. programming languages, software tools, project management).

32 responses

JavaScript, HTML, MySQL
C, MATLAB, research
mathematics, statistics, data analysis
SQL, Java, DBMS
HYSYS, MATLAB, Microsoft Office
HYSYS, MATLAB, Microsoft Office
project management, partnership development, designing new initiatives
Microsoft Office, translation, project management
Java/JavaScript, SFDC, development/consulting/administration
auditing, attention to detail, management
Coding, network configuration, data analysis
English Chinese ps

Please list your top three soft (i.e. interpersonal) skills (e.g. teamwork, communication, leadership).

32 responses

teamwork, time management, communication
willingness to learn, communication, leadership
teamwork, leadership, writing
teamwork, communication, leadership
teamwork, leadership, communication
adaptability, creativity, communication
leadership, mentorship, supportive
teamwork, communication, time management
teamwork, communication, global sense
optimism, responsibility,
Leadership, management, teamwork
Teamwork communication leadership