Automated Model to Rank Candidates for a Job Position based on Data Extracted from LinkedIn Profiles

by

Mouri Hoque Nadia
22341073

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
January 2023

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mouri Hoque Nadia
Student ID: 22341073

Approval

The thesis/project titled "Automated Model to Rank Candidates for a Job Position based on Data Extracted from LinkedIn Profiles" submitted by

1. Mouri Hoque Nadia (22341073)

Of Spring, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 19, 2023.

Examining Committee:

Supervisor:
(Member)

Md.Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md.Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Recruitment process has become very crucial in the vast job market and being able to recruit effectively is a challenge. This is because, there is a scarcity of suitable candidates for any particular job opening. Moreover, the ratio of a suitable candidate to the number of job opening is very low. Therefore, many multinational companies are investing fortunes in their recruitment teams. The only information that the recruiters have during the process of recruitment is the curricular vita, based on which a short interview is scheduled and then the candidates are hired. This does not give the recruiters an insight to their skills and educational background in a single format as different people write CVs in different ways. Also, recently, in most curricular vita the social handles such as Facebook or LinkedIn are provided. Data in these platforms can be taken advantage of to find information which are essential to ensure an efficient and successful recruitment. These data collected can be analyzed to match with the job requirements resulting in a more accurate recruitment process with data driven decision making. The two major entities in this process are the recruiters and the candidates who applied for the job. The challenge is to find a qualified candidate for a particular job that fulfills all the requirements of the job. Therefore, in this paper we have collected a data set of approximately 300 candidates, automatically, from their LinkedIn profiles for a job of a Software Engineer. Then, we have used NER of BERT model to pre-train the dataset – to summarize the text using NLP. Then, we have used the VADER model to carry out sentimental analysis of the text data. After that, we weighted each entities namely: About, Skills, Education Background, Experience and Language. Priority of each attributes were carefully considered by experts at Bangalink Digital Ltd. according to which they are weighted. Then, using XGBoost Machine Learning Algorithm, we have trained the system. Finally, we have used the TOPSIS Algorithm to rank the candidates and have a holistic idea of the quality of the applicants in a descending order of priority.

Keywords: Recruitment, LinkedIn, Ranking Candidates, XGBoost, Topsis, NLP, NER, BERT, VADER, Sentence Summarization, Sentence Scoring, Sentiment Analysis

Dedication

This work is dedicated to my father who would have been proud of me if he was here and my mother who has always given her best effort for the sake of my education.

Acknowledgement

Firstly, all praise to the Great Al-Mighty Allah for whom our thesis have been completed amidst the various difficulties faced especially during and after the pandemic. Secondly, to my advisor - Md.Golam Rabiul Alam sir. He has been very patient with me, believed in me and has guided me in every step to bring out the best outcome possible. I would also like to thank my co-advisor - Md. Moin Mostakim sir for motivating me and encouraging me throughout.

Table of Contents

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CV Curricular Vita

$KNeighborsClassifier$ Python Module

$MultinomialNB$ Python Module

NLP Natural Language Processing

$NLTK$ Natural Language Tool Kit

$OneVsRestClassifier$ Python Module

$XGBoost$ Machine Learning Algorithm

Chapter 1

Introduction

1.1 Motivation

Recommendation systems are very common in our daily lives. Starting from the posts that we see on social media platforms such as on Facebook, Instagram, Twitter, etc., to the music we listen to on Youtube or Spotify. Even shopping sites such as Amazon, Zara, Adidas, etc. and food delivery applications make use of recommendation systems to personalize product recommendation based on the user browsing data.[7] These recommendation systems use user data from purchases history or even the frequency of product viewing history. They track the videos that the users clicked on a certain time frame or save how long users spent at a particular post. These techniques make use of the user data to encourage the engagement of customers and improve customer experience with the help of predictive algorithms. Similar predictive algorithms could be applied to the recruitment process to rank candidates to match the requirements of a job to meet the criteria of a recruiter. For this, the personality trait of the candidates could also to be evaluated using the data obtained. Therefore, sentiment analysis needs to be done on text data to evaluate the Emotional Intelligence of the candidate.It can detect whether the texts are positive, negative or neutral.[10] Emotional Intelligence is a vital part of one's Intelligence as it can determine how a person will handle different scenarios. This is very crucial in a job environment. Different positions require different levels of soft skills. And the expertise in soft skills is matched with the personality trait of a person. Decision making skills is also very important in almost any job role in an organization. Therefore, analyzing the emotional intelligence of a person using the social media data is crucial to accurately match the job description. Moreover, all recruiters do not have the expertise or experience in specialised fields such as technical field like Software Engineering, therefore, a system built to accurately analyze the candidate in order to find the perfect fit can be very effective. For instance, the Human Resource department may not have subject knowledge of a technical position to a very great extent to accurately filter out applicants based on the skills given in their Curricular Vita. However, an automated system can filter candidates more accurately by training a Natural Language Process (NLP) model. Obtaining the data from social media can be done by web scrapping the data. [7] Python has built in libraries, such as selenium, web driver and so on, that are designed to do so. Also, frameworks such as BeautifulSoup have made it even easier to analyze text from social media APIs. Information mined from social media can then be used

and the data can be cleaned by stripping off unnecessary information. Moreover, there are specialized approaches to do this using Artificial Intelligence and Neutral Network Models. Then, simple mathematical models in statistics can be used to evaluate results. Then, a conclusion can be drawn by [1] ranking the candidates in a descending order of quality.

1.2 Research Problem

A number of research has been carried out on Twitter data or data from CVs. However, barely any research has been conducted on LinkedIn data to analyze a candidate. Also, LinkedIn is very popularly used both by recruiters and applicants for recruitment recently.
Previous research work has worked with the algorithms and techniques used in this paper. However, the combination of the techniques to solve this particular problem is still rare. A hybrid-model has been proposed in this paper that combines: web scrapping using selenium, NLP techniques such as NER of BERT model, VADER model for Sentiment Analysis, Machine Learning Model: XGBoost Classifier and Topsis Algorithm for ranking. This, particular hybrid-model has barely been worked on combined together to achieve our aim of ranking candidates. Therefore, in this paper we will be working with a unique hybrid model that has intriguing evaluation performance.

1.3 Research Aims and Objectives

The aim is to prepare a model which will automatically extract data from linkedIn and predict and rank candidates in a descending order of quality based on the data obtained. Then we want to summarize data extracted and summarising it using a suitable BERT Model. [10] BERT has a lot of scope in this aspect. Then, using sentimental analysis by using the VADER model. Then, generating scores for different attributes such as About, Job Experience, Education and Skills. Based on these scores, it aims to generate an overall category which will then be ranked using TOPSIS Algorithm.The most suitable methods or models would finally be adapted depending on the comparison of the performance evaluation.
The main Objective of this paper:

1. Deeply understand the setbacks and applications of web scrapping using selenium in python.

2. Understand different techniques of data pre-processing and opting for a technique most suitable.

3. Implementation of text summarizing and sentiment analysis using Spacy in Python.

4. Deeply understand and implement data pre-processing using word tokenize, sent tokenize, text blob, stop words and vectorization using NLTK and using labelEncoder.

5. Generating sentence score to match requirement of the job description and to categorise the data.

6. Using Traditional Machine Learning approach using libraries such as sklearn, naive bayes and their sub modules such as MultinomialNB, OneVsRestClassifier, KNeighborsClassifier.

7. Using TFVectorization to Train the Model.

8. Visualizing the data and results to have an understanding of the outcome of the experiment carried out.

9. Gaining in depth knowledge of XGBoost Classifier Model and implement it on our pre-processed data to carry out supervised learning.

10. Understand TOPSIS Algorithm and implement it in the fully processed and trained dataset to achieve final results.

11. Evaluate the performance of each implementation to see the overall success of the hybrid-model created and opt for the most suitable model in each step.

12. Finally, propose future implementation models and scope of improvements.

Chapter 2

Related Work

According to [7] Twitter API was used to scrape data automatically. The REST API of Python can be used to scrape Twitter data. It can extract data in the form of JASON files. They also discuss crawling techniques that can also be used to extract data from HTML files of the web and then use the data for further manipulation as mentioned in the article. Previously, in their work selenium and web driver was used to automate this process. However, due to LinkedIn's data protection policy the automation cannot be properly done without CAPTCHA interruption generated by the website. Consequently, only a small corpus of up to 300 records can be obtained. Although, it can still be used to mine data, it cannot be used to mine a large dataset. In order to extract large data set we can use a Big Data Tool called Kafka to pipeline the data to stream them in order to obtain them. This is because, a good data model require a large data set to precisely be trained and obtain accurate results. This can be done by using LinkedIn's API called Profile API [2]. This only extracts data from authorized entities, as a result ethical data extraction can be carried out. The automation process can be improved by using undetected chrome driver. Earlier, [1] chrome driver was used, which in practice has a couple of issues as it cannot bypass the CAPTCHA generated by the website. On the other hand, undetected chrome driver can solve this issue. We have bypassed this issue by using time delay in python and manually bypassing the CAPTCHA generated. Then, data pre-processing technique followed earlier [2] was NER (Named Entity Relationship) of BERT (Bidirectional Encoder Representation from Transformers). It has a lot of scope in summarization of texts. According to [2] NER of BERT can be used to obtain key words. Then, they have used Knowledge Graph to rank the candidates in their previous work. As NER model has a lot of future implications for data pre-processing, we have used it to summarize our data. According to [4] XG-Boost Classifer was used to categorize the corpus. Finally, [5] Topsis Algorithm was used to rank the candidates according to a descending order of priority. According to [5] BERT is a retrained/transfer learning language model of Deep Bidirectional Transformers in NLP. It is used to understand language better. It uses the transformer architecture. It is, in simple words, used to understand relationship between sentences better. According to the article, there are many recent uses of this model. Google's AI team has worked on a BERT Model that performs tasks such as question answering, named entity recognition and language understanding. The BERT model shares the same underlying concept as Chat GPT. They both are based on transformer architectures. Although they are fundamentally built on the same con-

cept, they are two different aspects. According to Devlin et al. , BERT works on both the directions jointly. Therefore, it can be finely tuned and one other layer is required for final output. It is a very simple concept yet can give us numerically very accurate results comprising of a GLUE bench mark of up to 80.4 percent. Moreover, it gives a MultiNLI accuracy of 86.7 per cent and SQuAD v1.1 question answering Test F1 score up to 93.2 which is 2 percent better than the performance than that of humans. Moreover, VADER (Valence Aware Dictionary for Sentiment Reasoning), according to Borchers et al. has multiple subdirectories. Therefore, it is capable of understanding order of words and the degree of words such as "slightly", "very" etc. It can also popularly interpret sentiments based on Twitter Corpus. Therefore, this model has been used for the purpose of analyzing the "About" section of the LinkedIn data extracted. [6] LIWC can alternatively be used to VADER as tested out by Borchers et al. Therefore, this can be a future scope of work as well. For this particular paper, we have worked with VADER model. Finally, Topsis (Technique for order of Preference by Similarity to Ideal Solution) is a multi-criteria decision analysis method. It makes a decision based on a pre-defined criteria on particular alternatives. It aims to have the shortest geometric distance from the best alternative and the largest geometric distance from the worst alternative. [7] This has many industrial implications and the use cause fits very well with the aim of the work in this paper. Therefore, we have considered using this Algorithm for the final step of ranking the alternatives/candidates.

Chapter 3

Methodology

3.1 Proposed Model

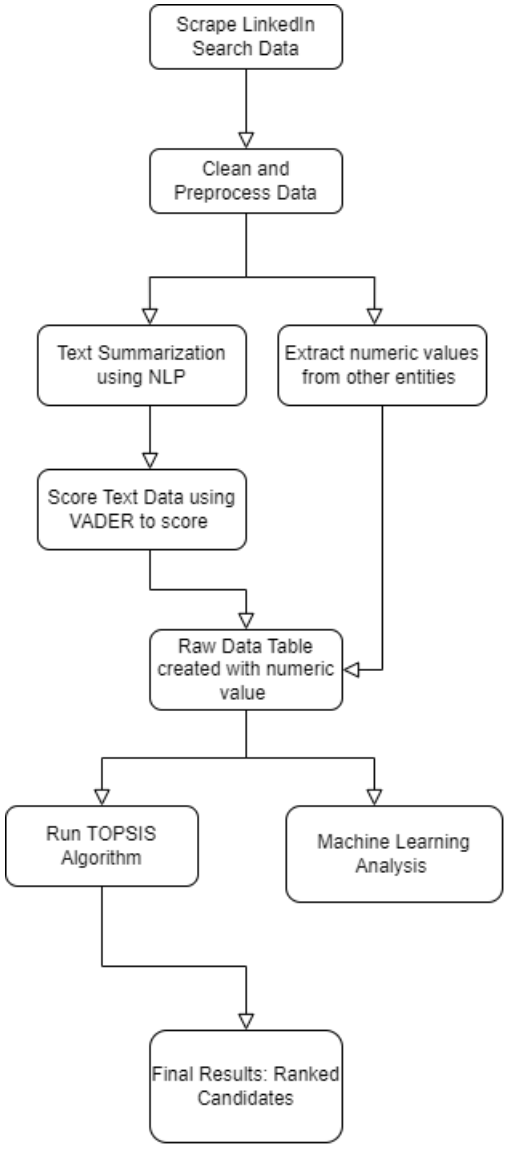

Fig. 3.1 Top level overview of the proposed model

3.2 Data Collection

Dataset of around 300 alternatives were obtained by web scraping using selenium. The data obtained was in CSV format. CSV format enhanced readability and ease of editing. This was a fully automated process carried out using the python library: [9]Selenium. Sub modules such as web driver, BeautifulSoup, Tag, Selector, Desired-Capabilities etc. A fully automated system was built that scrapped close to 300 data set from LinkedIn. Since the data was extracted against the data extraction policy of LinkedIn, we have used a unique ID to replace the name of the individuals whose data was obtained in order to maintain data privacy and confidentiality. Also, the data obtained was publicly available and was strictly used for research purpose only. This corpus will not, in any circumstance, be shared with/sold to any third party organization.
This model automatically, searched the position "Software Engineer" through the linkedIn search bar. For demostration purpose, search results usig my own profile examples are shown in this paper:

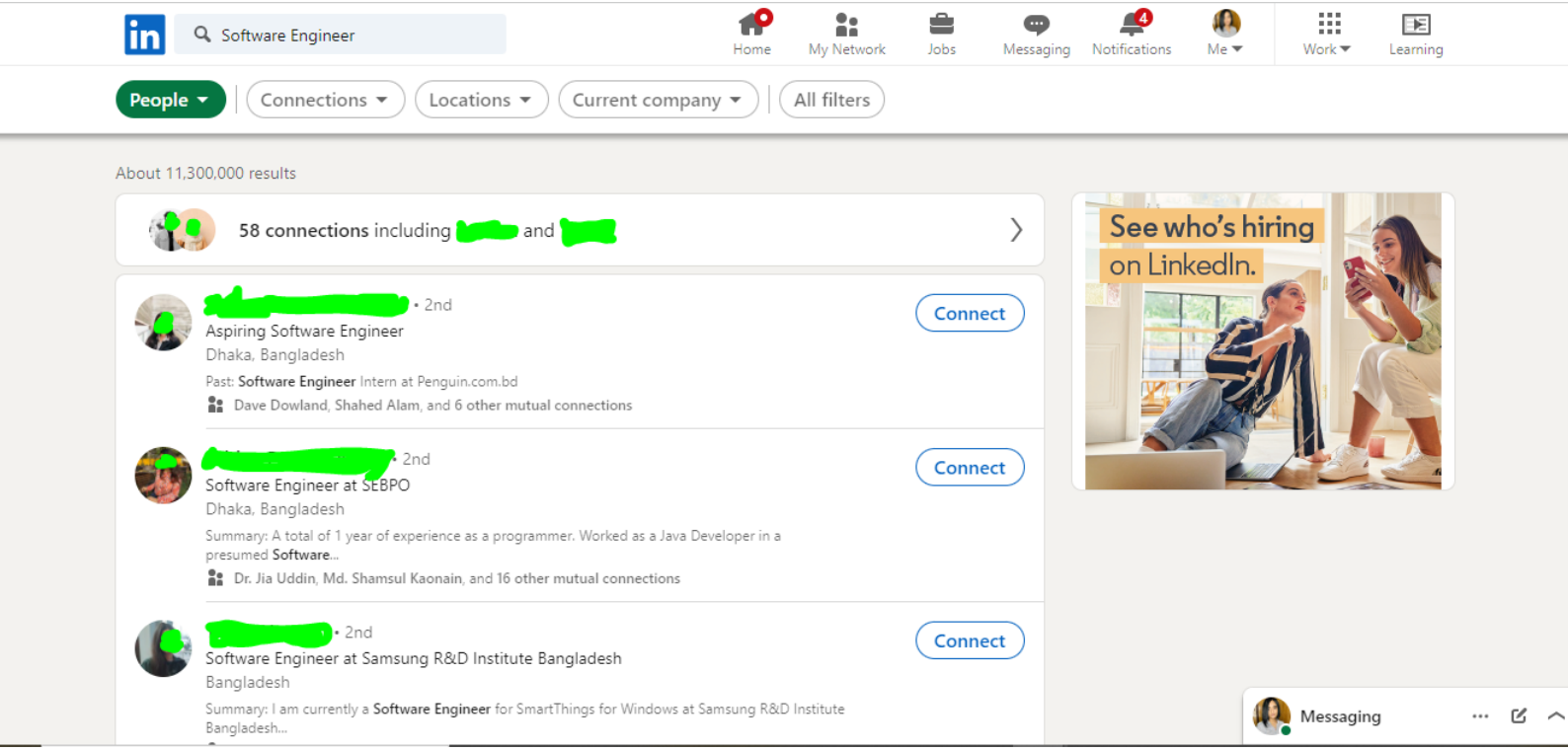

Fig. 3.2.1: Search Results

The program then saved each profile link in an array and then visited them individually in a loop.
After visiting each profile, the program saved attributes that are divided into each sections of one's LinkedIn profile and saved them in a data frame. For instance, the attributes "About", "Education", "Experience", "Skills", and "languages" were taken from the sections of their profiles as shown below:

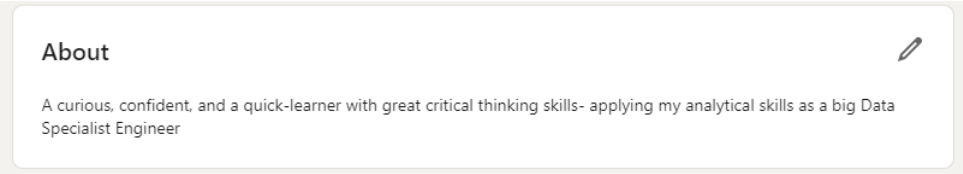

Fig. 3.2.2: "About" section

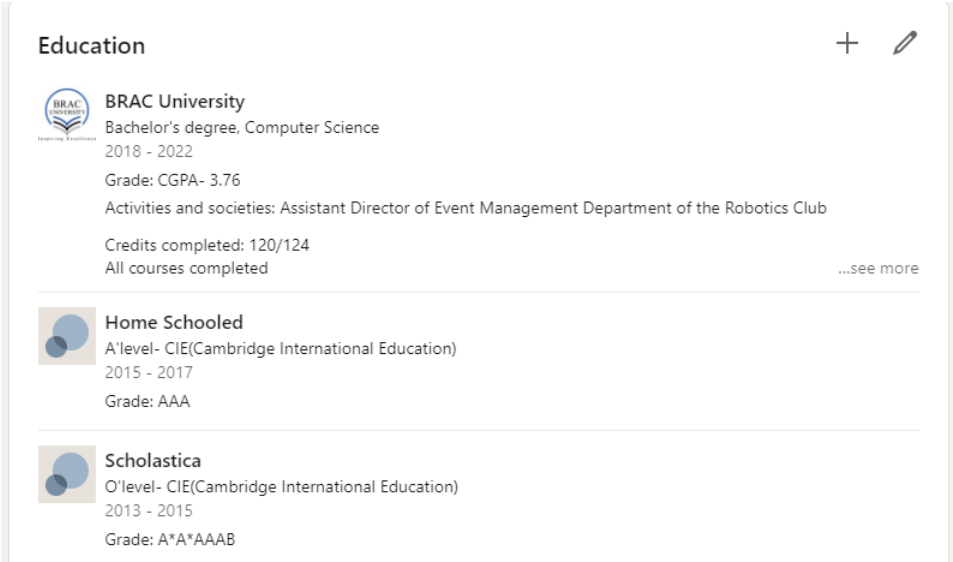

Fig. 3.2.3: "Education" section

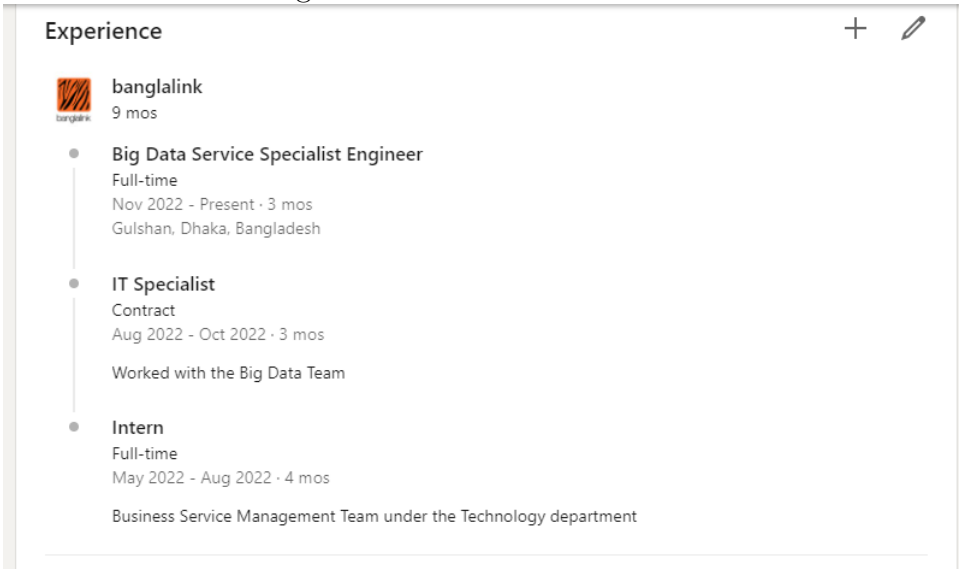

Fig. 3.2.4: "Experience" section

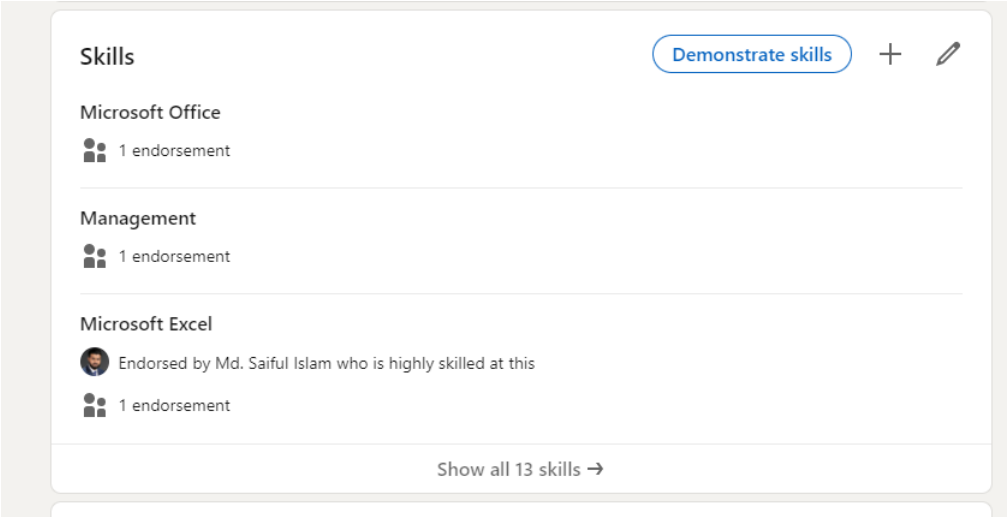

Fig. 3.2.5: "Skills" section

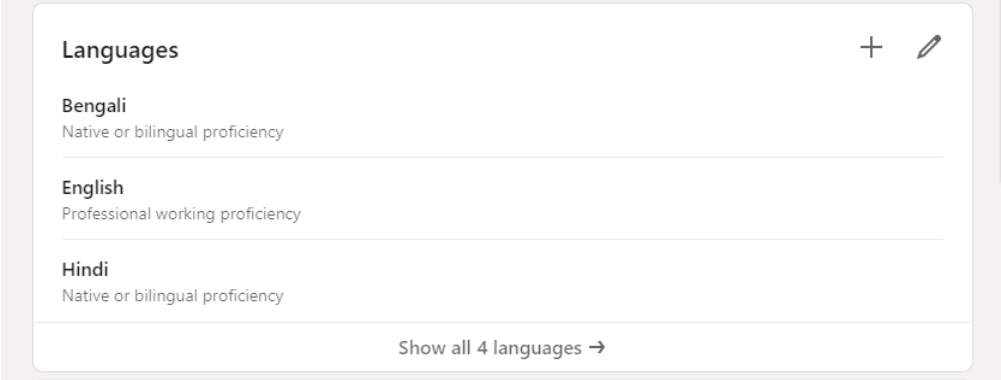

Fig. 3.2.6: "languages" section

3.3 Data Pre-processing

3.3.1 Preparing and cleaning data

The raw data obtained was first prepared to protect individual privacy by generating a unique ID. The unique ID was generated using a hash function. Then, the absolute value of the hash function was used as a hash function generates both positive and negative values. After that, regular expression was used to clean the raw data obtained. Online platform RegX was used to generate a generalized regular expression suitable for the dataset to be cleaned at its primal stage.The raw data obtained was first prepared to protect individual privacy by generating a unique ID. The unique ID was generated using a hash function. Then, the absolute value of the hash function was used as a hash function generates both positive and negative values. After that, regular expression was used to clean the raw data obtained. Online platform RegX was used to generate a generalized regular expression suitable for the dataset to be cleaned at its primal stage.

3.3.2 Feature Engineering

The texts obtained in attributes namely "Job Title", "About", "Experience", "Education", and "Skills" were concatenated together to form a single attribute for a

holistic analysis. This is because we have a comparatively small corpus. Concatenated data will result in larger corpus for each alternatives. Moreover, analyzing all the corpus at once for each alternative will enhance the accuracy of the results as compared to analysis carried out separately for small attributes of a single alternative. The texts obtained in attributes namely "Job Title", "About", "Experience", "Education", and "Skills" were concatenated together to form a single attribute for a holistic analysis. This is because we have a comparatively small corpus. Concatenated data will result in larger corpus for each alternatives. Moreover, analyzing all the corpus at once for each alternative will enhance the accuracy of the results as compared to analysis carried out separately for small attributes of a single alternative. [4]

3.3.3 Data Cleaning

The resulting data is further cleaned to prepare for scoring. NER from BERT Model is not used to clean the data using the Spacy library of Python. Spacy uses the NER Model. Alternatively, for faster time complexity for cleaning, NER can also be disabled while still taking advantage of the Spacy Library from Python for cleaning purpose. Finally, the data is fully cleaned and ready to be analyzed through manipulation. We will be using the NER model for pre training purpose in the later steps.The resulting data is further cleaned to prepare for scoring. NER from BERT Model is not used to clean the data using the Spacy library of Python. Spacy uses the NER Model. Alternatively, for faster time complexity for cleaning, NER can also be disabled while still taking advantage of the Spacy Library from Python for cleaning purpose. Finally, the data is fully cleaned and ready to be analyzed through manipulation. We will be using the NER model for pre training purpose in the later steps.

3.3.4 Text Scoring: AFINN vs VADER Model

Now, the data is fully cleaned and ready to be processed. We have used python scripts to calculate "Years of experience" from "Experience", "Years of Education" from "Education" and "No. of Skills" from "Skills" attributes respectively. This is because require an empirical value in order to carry out further analysis. However, the attributes "About" cannot be randomly given a numeric value. Therefore, we have carried out text scoring techniques in order to generate a numeric value that is relevant to the corpus. We have scored the textual data using two models namely: [5]AFINN and VADER to carry out a sentimental analysis on the "About" attribute. This will not only give a text based score but also give an insight on the emotional intelligence of the candidate. AFINN generates scored between -1 and 1 after analyzing the texts. On the other hand, VADER evaluates the sentiments of the texts and comes to a conclusion whether it is a positive, negative or neutral emotion and at the same time generated a compound score. We have used VADER as it is more precise and detects the degree of sentiments and also can identify emoticons. Finally, the compound score can then be used for further analysis.Now, the data is fully cleaned and ready to be processed. We have used python scripts to calculate "Years of experience" from "Experience", "Years of Education" from "Education" and "No. of Skills" from "Skills" attributes respectively. This is because require

an empirical value in order to carry out further analysis. However, the attributes "About" cannot be randomly given a numeric value. Therefore, we have carried out text scoring techniques in order to generate a numeric value that is relevant to the corpus. We have scored the textual data using two models namely: AFINN and VADER to carry out a sentimental analysis on the "About" attribute. This will not only give a text based score but also give an insight on the emotional intelligence of the candidate. AFINN generates scored between -1 and 1 after analyzing the texts. On the other hand, VADER evaluates the sentiments of the texts and comes to a conclusion whether it is a positive, negative or neutral emotion and at the same time generated a compound score. We have used [3]VADER as it is more precise and detects the degree of sentiments and also can identify emoticons. Finally, the compound score can then be used for further analysis.

3.4 Data Summarization and pre-training

3.4.1 Preparing text for the NLP Model

In order to prepare the corpus/text data for sentiment analysis it has gone through a number of stages. The NLTK (Natural Language Tool Kit) was used to prepare the data for text summarization. First, word tokenization was used to split texts from white space or punctuations. As each words were put into a list after undergoing the split function, it was then used to calculate the word frequency count. In turn, the frequency count was helpful in formation of readable sentences which was done using the sentence tokenization technique. sent token from python library was used to make tokens. Later, text blobs were created from the tokens for the NLP model to be able to differentiate between various parts of speech and understand the sentences fully.

3.4.2 Text Summarization

Before, running the proposed algorithms we pre-trained the model using TFVectorization. TFVectorization processes the independent text attributes. We fed the model with the summarized texts and the corresponding categories/ grades for each of these text summary as a whole. Spacy from python was used to do so. The underlying algorithm used by Spacy is the [8]NER from the BERT model for NLP. This is how the texts have been fully prepared before running the proposed algorithms.

3.5 TOPSIS Algorithm

Finally, [6]TOPSIS Algorithm has been carried out on the derived numeric data. First the scores has been converted into a numpy array using python and the alternatives were numeric in appearance but string in type. The weights: 0.2, 0.3, 0.1, 0.2, 0.2 for the attributes: 'Total Experience Years', 'Skills counts', 'Education Score', 'about score', 'total score' were carefully assigned as suggested by the line manager of Big Data Services, IT Department of Banglalink Digital Ltd.

The following are the steps followed by a TOPSIS Algorithm:

1. Create an evaluation Matrix with M alternatives and N criteria:

$$(\mathrm{a}_{ij})_{MXN} \tag{3.1}$$

2. Normalize the evaluation Matrix.

$$\mathrm{a}_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^{M}(a_{ij})^2}} \tag{3.2}$$

3. Calculate weighted normalized decision matrix using(Weight given by experts):

$$\mathrm{x}_{ij} = a_{ij} * w_j \tag{3.3}$$

4. Best (shortest geometric distance) and worst(Longest geometric distance) alternative calculated:

$$\mathrm{x}_j^b = \max_{i=1}^{M} x_{ij} \tag{3.4}$$

$$\mathrm{x}_j^w = \min_{i=1}^{M} x_{ij} \tag{3.5}$$

5. Calculate Euclidean distance between target alternatives and best/worst alternatives:

$$\mathrm{d}_i^b = \sqrt{\sum_{j=1}^{N}(x_{ij} - x_j^b)^2} \tag{3.6}$$

$$\mathrm{d}_i^w = \sqrt{\sum_{j=1}^{N}(x_{ij} - x_j^w)^2} \tag{3.7}$$

6. Calculate TOPSIS Score: similarities to the worst alternatives:

$$\mathrm{s}_i = \frac{d_i^w}{d_i^w + d_i^b} \tag{3.8}$$

7. Rank in descending order to obtain the ranking.

Chapter 4

Implementation

4.1 Machine Learning Analysis

For our experiments we have worked closely with supervised learning models as this is a relatively new work with very specific requirements. We worked with XGBoost Classifier Algorithm along-side of TOPSIS algorithm as they go hand in hand and can very easily implemented using few lines of python code. In order to carry out machine learning analysis we have two types of data set. First, the training data which is the independent variable. Secondly, the test or target data also known as the dependent

4.1.1 Training Data (Independent Variable: Whole Text)

Target data is a variable is the data that we categorized. This is because it gives us a goal/target/label that we want each summarized blob of text to represent. In This manner, the machine can be able to learn what we want it to understand.

4.1.2 Label Encoder used to convert to binary

Finally we have used the LabelEncoder from python library to convert our test and train dataset into machine readable format. This entire flow of work was consequently used to feed the XGBoost Classifier which was fed into the TOPSIS algorithm using very simple and few lines of python code.

Chapter 5

Results and Discussion

5.1 Performance Evaluation

The accuracy of XGBClassifier Classifier on training set was 98 per cent and the accuracy of XGBClassifier Classifier on test set was 87 per cent. The results obtained was excellent considering the setback of having a small data set. The table below illustrates the overall results of the model used:

	precision	recall	f1-score	support
0	0.93	0.93	0.93	27
1	0.75	0.82	0.78	11
2	0.00	0.00	0.00	1
accuracy			0.87	39
macro avg	0.56	0.58	0.57	39
weighted avg	0.85	0.87	0.86	39

Fig. 5.1.1: Performance evaluation matrix

The performance of the model was further evaluated by constructing a confusion matrix based on the train and test set. The confusion matrix evaluates the accuracy, specificity, sensitivity, precision and recall of the model. The x axis was fed with the test data set and the y axis was fed with the train dataset. The resulting matrix below illustrates the outcome:

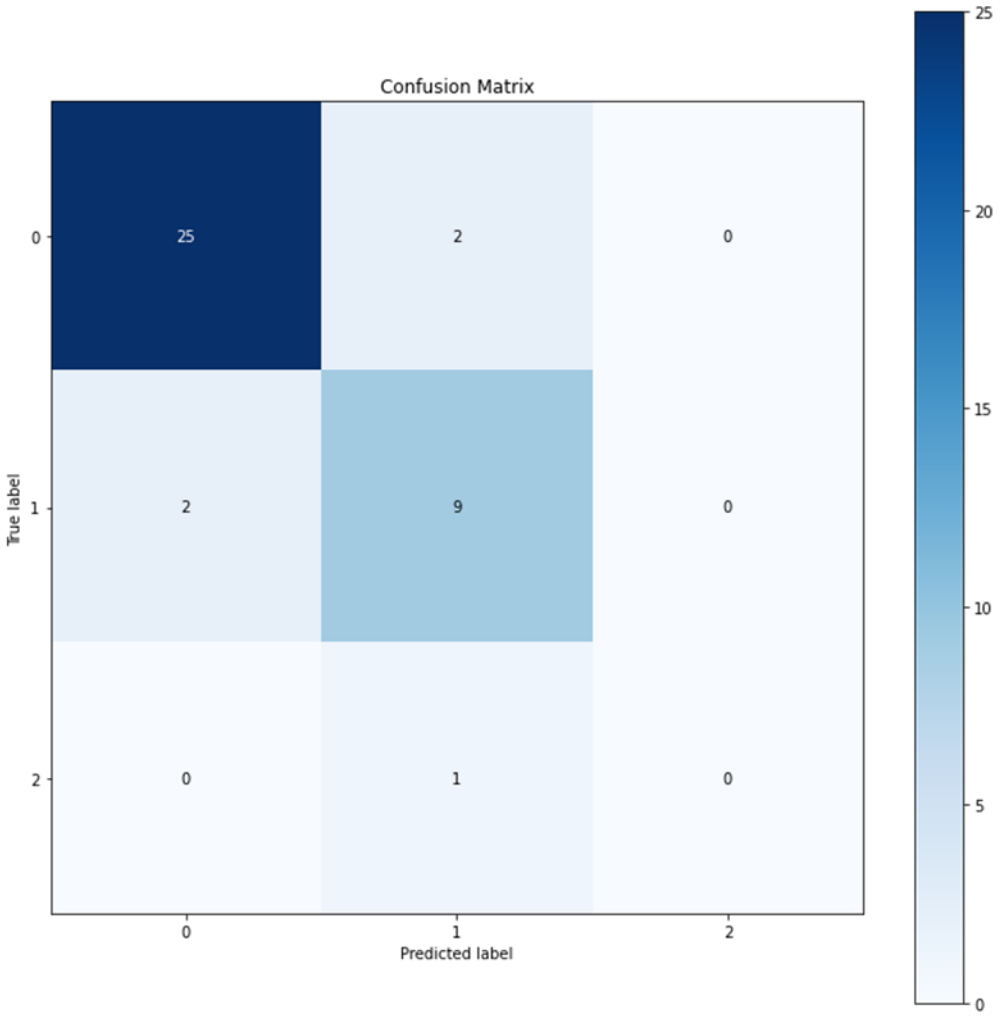

Fig. 5.1.2: Performance evaluation matrix

5.2 Results

At first we began with our raw data set that we obtained after pre-processing the data:

	Total_Experience_Years	Skills_counts	Education_Score	about_score	total_score
(0,)	10.0	62.0	4.0	0.9973	76.9973
(1,)	3.0	45.0	3.0	0.7650	51.7650
(2,)	8.0	46.0	3.0	0.9700	57.9700
(3,)	10.0	48.0	3.0	0.0000	61.0000
(4,)	13.0	11.0	3.0	0.9915	27.9915
...	...	...	...	...	...
(187,)	13.0	43.0	3.0	0.8807	59.8807
(188,)	10.0	44.0	3.0	0.9812	57.9812
(189,)	5.0	35.0	3.0	0.8689	43.8689
(190,)	9.0	51.0	3.0	0.9873	63.9873
(191,)	5.0	21.0	3.0	0.9638	29.9638

192 rows × 5 columns

Fig. 5.2.1: Numeric Representation of Raw Dataset

After successfully running the whole algorithm, the following were the outcomes of each step of the algorithm:

1. Using Eq.(3.2) normalizing the ratings and the outcome is as follows:

	X_{0}	X_{1}	X_{2}	X_{3}	X_{4}
(0,)	0.091574	0.099736	0.089510	0.088654	0.100336
(1,)	0.027472	0.072389	0.067132	0.068004	0.067455
(2,)	0.073259	0.073998	0.067132	0.086227	0.075541
(3,)	0.091574	0.077215	0.067132	0.000000	0.079490
(4,)	0.119046	0.017695	0.067132	0.088138	0.036476
...	...	...	...	...	...
(187,)	0.119046	0.069172	0.067132	0.078289	0.078031
(188,)	0.091574	0.070781	0.067132	0.087223	0.075556
(189,)	0.045787	0.056303	0.067132	0.077240	0.057166
(190,)	0.082416	0.082041	0.067132	0.087765	0.083382
(191,)	0.045787	0.033782	0.067132	0.085676	0.039046

192 rows × 5 columns

Fig. 5.2.2: Normalised Value of Data

2. Then, using Eq.(3.3) calculating the weighted normalized ratings produced the following table:

	X_{0}	X_{1}	X_{2}	X_{3}	X_{4}
(0,)	0.018315	0.029921	0.008951	0.017731	0.020067
(1,)	0.005494	0.021717	0.006713	0.013601	0.013491
(2,)	0.014652	0.022199	0.006713	0.017245	0.015108
(3,)	0.018315	0.023165	0.006713	0.000000	0.015898
(4,)	0.023809	0.005309	0.006713	0.017628	0.007295
...	...	...	...	...	...
(187,)	0.023809	0.020752	0.006713	0.015658	0.015606
(188,)	0.018315	0.021234	0.006713	0.017445	0.015111
(189,)	0.009157	0.016891	0.006713	0.015448	0.011433
(190,)	0.016483	0.024612	0.006713	0.017553	0.016676
(191,)	0.009157	0.010134	0.006713	0.017135	0.007809

192 rows × 5 columns

Fig. 5.2.3: Table with Weighted Normalised Ratings:

3. Identifying PIS(A*) and NIS(A-) using Eq.(3.4) and Eq.(3.5):

	X_{0}	X_{1}	X_{2}	X_{3}	X_{4}
A^*	0.042124	0.030403	0.013426	0.017749	0.021095
A^-	0.000000	0.001448	0.006713	0.000000	0.004105

Fig. 5.2.4: Table containing (A*) and (A-)

4. Calculating Separation Measures and Similarities to PIS using Eq.(3.6):

	S^*	S^-	C^*
(0,)	0.024253	0.041477	0.631022
(1,)	0.039208	0.026723	0.405314
(2,)	0.030053	0.032616	0.520445
(3,)	0.031723	0.030759	0.492284
(4,)	0.034651	0.030045	0.464400
...	...	...	...
(187,)	0.022542	0.036290	0.616837
(188,)	0.027054	0.033947	0.556497
(189,)	0.037591	0.024793	0.397424
(190,)	0.027488	0.035699	0.564973
(191,)	0.041468	0.021602	0.342511

192 rows × 3 columns

Fig. 5.2.5: Euclidean Distance

5. Ranking candidates/alternatives using Erq.(3.8):

	C^*	S^*	S^-
1	[16]	[174]	[20]
2	[166]	[30]	[159]
3	[42]	[135]	[38]
4	[161]	[28]	[139]
5	[177]	[12]	[164]
...	...	...	...
188	[91]	[90]	[89]
189	[118]	[59]	[100]
190	[146]	[41]	[126]
191	[50]	[148]	[56]
192	[180]	[17]	[187]

192 rows × 3 columns

Fig. 5.2.6: TOPSIS Score

6. The final Ranking in descending order of priority identified using a serial number. The below shows the top 5 results for readability purpose:

['16']
['166']
['42']
['161']
['177']

Chapter 6

Conclusion and Future Scope

Recruitment processes are time consuming, laborious and expensive. A successful system built to automate the process can make the work fast, effective and efficient. It will reduce human labor and human error. Companies will benefit from it in a number of ways. They will be able to invest in recruits who are suitable and more resourceful for their designation. There are numerous future scope of this work. First of all, data extraction can be improved by using LinkedIn's Profile API followed by the usage of Kafka to pipeline the data. This will result in a huge number of dataset which will enhance the accuracy of the model. Then, the data can be stored in a Hadoop File System and Hive Database can be used to query the unstructured data into a structured data as per the requirement. This platform is suggested as we are talking about huge amount of data for better precision of the model. Moreover, VADER could be replaced with LIWC model for sentiment analysis in the future as it is a much faster and more accurate algorithm. Finally, the overall outcome of the results can be illustrated using data visualization tools such as Tableau for better User Interface and experience.

Bibliography

[1] R. Kessler, N. Béchet, J.-M. Torres-Moreno, M. Roche, and M. El-Bèze, "Job offer management: How improve the ranking of candidates," in *International Symposium on Methodologies for Intelligent Systems*, Springer, 2009, pp. 431–441.

[2] P. Lops, M. De Gemmis, G. Semeraro, F. Narducci, and C. Musto, "Leveraging the linkedin social network data for extracting content-based user profiles," in *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 293–296.

[3] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the international AAAI conference on web and social media*, vol. 8, 2014, pp. 216–225.

[4] M. Fang, "Learning to rank candidates for job offers using field relevance models," *Master's thesis*, 2015.

[5] F. Koto and M. Adriani, "A comparative study on twitter sentiment analysis: Which features are good?" In *Natural Language Processing and Information Systems: 20th International Conference on Applications of Natural Language to Information Systems, NLDB 2015, Passau, Germany, June 17-19, 2015, Proceedings 20*, Springer, 2015, pp. 453–457.

[6] S. Nădăban, S. Dzitac, and I. Dzitac, "Fuzzy topsis: A general view," *Procedia computer science*, vol. 91, pp. 823–831, 2016.

[7] E. R. Kaburuan, A. S. L. Lindawati, M. R. Putra, D. N. Utama, *et al.*, "A model configuration of social media text mining for projecting the online-commerce transaction (case: Twitter tweets scraping)," in *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, IEEE, vol. 7, 2019, pp. 1–4.

[8] Z. Liu, F. Jiang, Y. Hu, C. Shi, and P. Fung, "Ner-bert: A pre-trained model for low-resource entity tagging," *arXiv preprint arXiv:2112.00405*, 2021.

[9] S. Nyamathulla, P. Ratnababu, N. S. Shaik, *et al.*, "A review on selenium web driver with python," *Annals of the Romanian Society for Cell Biology*, pp. 16 760–16 768, 2021.

[10] Y. Wang, Y. Allouache, and C. Joubert, "Analysing cv corpus for finding suitable candidates using knowledge graph and bert," in *DBKDA 2021, The Thirteenth International Conference on Advances in Databases, Knowledge, and Data Applications*, 2021.