The Indonesian Journal of Computer Science
www.ijcs.net
Volume 13, Issue 5, October 2024
https://doi.org/10.33022/ijcs.v13i5.4232

Designing an Effective Job Recommender System based on Embedded Machine Learning Models

Abiola Olaide Ayodele[1], Adedeji Daniel Gbadebo[2]
ayodeleabiola1994@gmail.com[1], agbadebo@wsu.ac.za[2]
[1] Department of Computer and Information Tech, University of Ilorin, Ilorin, Nigeria
[2] Department of Accounting Science, Walter Sisulu University, Mthatha, Eastern Cape, South Africa

Article Information

Received : 2 Jul 2024
Revised : 17 Jul 2024
Accepted : 29 Oct 2024

Keywords

recommender systems, embeddded, machine learning

Abstract

The need to automate employment offers to qualify job searchers has gain attentions. For an automated recommendation systems to be used more frequently, a better user-friendly filtering techniques are required. This paper designs an automated process, referred to as “the job recommender”, which focuses on user-centric design and personalization for recommending and matching applicants with appropriate jobs. We use the bottom-up approach that uses dataset based on filtering algorithms to predict and make recommendations for job seekers. The algorithm helps the recruiters to produce the list of the résumé that best meets the job descriptions. In this context, the random forest (RF) and support vector machines (SVM) are adopted to train the data. They are supplied personalized information (qualifications, result of aptitude test, age, and work experience) reported on the résumés of individual candidates from the pool of submissions, and the system train data to learn the evolution of job selection by candidates based on these machine learning tools. The algorithm used would help the recruiters to produce the list of the résumé that best meets the job descriptions. The algorithms are designed to recommend personalized items tailored to each user's interests. Under the minimum hardware and software requirements, the job recommender system was implemented in streamlit - a python template, for designing the frontend.

A. Motivation

The continuous advancement in computer technology has resulted in a huge amount of user information being generated. The data remains useful for different purposes including to process futuristic needs of customer that will be amounted to increased customer satisfaction as well as providing information on products that are of high quality and demand to ensure companies increase sales and make profits. The data has been serving other purposes, including providing recommendations (Abel et al., 2016; Celma, 2010). For instance, popular online shopping companies, including Flipkart, Amazon and Alibaba, use stored data to recommend products, and social networking websites like Facebook use the gathered information to recommend friends (Diaby et al., 2018; 2014). A machine learning recommendation system leverages on data and algorithms to mimic human learning processes to offering recommendations for e-commerce's goods (Schafer et al., 1999), books (Rhanoui et al., 2022; Priyanka et al., 2015), news (Mitova, 2022) as well as music and movies (Celma, 2010). The tools outperformed traditional job portal systems in image recognition, speech recognition, and natural language processing (Mrzic & Zaimovic, 2020). Like these systems, the process of matching candidates' résumé with job description involves suggesting them for specific positions based on their profiles.

Nowadays, many job openings are advertised online via job portal or social media, leading to internet users to receive jobs alerts that are most times unrelated to their interest. The large volume of applicants presents a significant problem for companies and job recruiters because they must narrow down the pool of résumé to find the best-qualified profile (Buschner et al., 2014). With existing system cost ineffective, there is the need to offer a job recommendation system that optimizes resources by scanning through submitted information on the résumé and fetching out necessary details pertinent to job descriptions. Since machine learning algorithms can get the pattern and in-depth perception of the internet data, it can be used to match various aspects of résumé to the job descriptions provided by companies. The designed system would ensure appropriate matching of job requirements and recommendations, therefore solving the problems of associating job seekers and hiring companies.

Designing a job recommender system is important for several reasons. The system would make it easier and faster for job seekers to identify the appropriate employment prospects. It helps job seekers find openings that they might not have otherwise be found while saving them time and effort in unproductive job hunting for suitable positions. The recommender system can assist businesses to select the best applicants for their open positions based on considered features, including the applicant's certification, experience and scores from skill tests. The recommender design, based on objective criteria than subjective ones like personal traits, would result in more effective recruitment procedures that save time and money as well as reduces likely errors and bias in recruitment process.

Generally, an efficient job recommender system overcomes four challenges of recruitment process such as to: (1) select the most qualified applicants from the group (2) to interpret the candidate résumé (3) confirm that the candidate chosen are capable for the position (4) ensure that the applicant possesses the skills needed for the position. This paper designs an automated process, referred to as "the job recommender", which is aimed at addressing these problems. We use the bottom-up approach that uses dataset based on filtering algorithms to predict and make quality recommendations for job seekers. The system makes it easier to select the best résumé from a sizable pool of candidates and to provide the best method in which candidate résumé would be well-structured based on order of the highest qualification. We use two ensemble - random forest (RF) and the support vector machines (SVM) algorithms - to deliver a high level of accuracy that assist in finding appropriate employment possibilities in line with their résumés/profiles and assist recruiters in locating potential applicants for open positions.

B. Materials

Recommendations System

The effectiveness of recommender systems as a significant tool for reducing the information explosion has been extensively examined by academics and business. The widespread use of the recommendation system in real-time applications was explored in various fields (Mitova, 2022; Rhanoui et al., 2022; Bolandraftar & Imandoust, 2017; Priyanka et al., 2015; Celma, 2010; Mooney et al., 2000). The system is recognized in various domains of knowledge nowadays including product recommendations on e-commerce portals (Schafer et al., 1999), book recommendations (Rhanoui et al., 2022; Priyanka et al., 2015; Mooney et al., 2000), news recommendations (Mitova, 2022), movie and music recommendations (Celma, 2010), and advertisements (García-Sánchez et al., 2021). Wei et al. (2007) shows four types of recommendation techniques, including collaborative filtering, content-based filtering, knowledge-based methods, and hybrid filtering. Bhatia et al. (2019) build a résumé scanner that extracts information from candidate résumé and use BERT sentence classification to complete ranking based on the candidate suitability to the job description. Malinowski et al. (2016) employ the expectation maximization method for the job recommendation that examines applicant's résumé and job description. Al-Otaibi & Ykhlef (2012) implement an employment recommendation service, disclosing the steps that constitute hiring procedures, including how the e-recruitment portal benefits the organization and candidate's characteristics that may influence the selection.

Advisory Systems

The 21st century is a time of data and information. An abundance of data creates numerous benefits, but it also has the drawback of an information explosion. Despite the abundance of jobs available, users cannot easily and quickly select one that matches their area of interest. The recommender system is a simple solution to the information explosion issue. Recommender systems effectively capture hidden information, such as users' personalized job preferences and job attributes based on these data, and filter and display to users a personalized list of job interests from a pool of resources. This is based on historical interaction data between users and vacancies.

There are two different categories of recommendation technologies: collaborative filtering-based and content-based (Yu et al., 2022). While the latter primarily determines the degree of matching from content feature traits, the former focuses on the influence relationship among users. Recent developments in AI have sparked a new wave of machine learning in both theoretical and empirical research (Mrzic & Zaimovic, 2020). Machine learning has increased productivity compared to typical shallow existing job portals. The recommender systems develop user preference models using the interest elements that users' profiles and résumé contain after extracting them from a set of data that at first glance appears to be unimportant. A novel direction in the evolution of recommender systems in recent years is the application of machine learning to job recommender systems. Firms like Spotify, Netflix, Google, Microsoft, and others have done extensive research on how to apply machine learning for job suggestion as a system and have shown positive outcomes in real-world applications.

Collaborative Filtering

Collaborative filtering is a method based on a user's human ratings of an item and identifying patterns across many users who have supplied related ratings to things (Yalcin et al. 2021; Mustafal et al., 2020). The memory-based k-nearest neighbor method of grouping users with related interests is the key procedure performed here. The generation of recommendations will take a long time as the volume of data continually increases (Xu & Liu, 2019; Bobadilla, 2013). Although content-based filtering techniques have an advantage over collaborative filtering, due to the nature of the recruiting process, a job can be rated by the user and a similarity matrix can be created.

Content-based Filtering

This method generates recommendations by comparing user profiles with keywords and attributes assigned to objects in a database, such as items in an online marketplace. User behaviors, including purchases, ratings, downloads, product searches, and interactions like adding items to a cart or clicking on product links, are used to create user profiles. Imagine suggesting accessories to a customer who recently purchased a smartphone from your website and has previously bought smartphone accessories. Along with keywords related to the smartphone's brand, make, and model, the user profile includes past purchases like phone holders with credit card sleeves. Based on this, the recommender system might suggest similar phone holders for the new phone, including features like an RFID-blocking fabric layer to prevent unauthorized credit card scanning.

Hybrid Filtering

Some authors suggest adopting hybrid filtering techniques (Afoudi et al., 2021; Riyahi & Sohrabi, 2020). hybrid filtering combines several strategies to enhance the effectiveness of recommendations. There are advantages and disadvantages to the recommendation method that was previously described. This strategy is desired to obtain a better recommendation and get over the problems caused by earlier methods. The cold-start problem affects all learning/model-based strategies in one way or another.

C. Methodology

Recommender System

We applied the Porter Stemmer algorithm to scan the résumé. The algorithm first makes each word into its root word for pre-processing the top and stop words. The stop words eliminate any unnecessary terms on the résumé and job description using the TF-IDF vectorizer. The process further contrasts the skills listed in the résumé and job description and determines the résumé scores for each job and compares them with cosine similarity function. The list is then arranged according to their scores in descending, from top to bottom. The system not only displays the job but suggests talents that should be developed for the position. Figure 1 shows the architectural design for the recommender system created. When a candidate logs into the job portal, the Recommendation Composer Module (RCM) receives their meta-data. The RCM then builds a job filter utilizing relaxed parameter values based on the candidate's attributes to retrieve a subset of appropriate jobs. For example, if an applicant has four years of experience, the filter states that the minimum experience is three years.

We follow the standard approach of the Software Development Life Cycle (SDLC). The process ensures the production of high-quality software efficiently and cost-effectively within a short timeframe. We use the waterfall model, which is widely adopted to provide frameworks for progressing through the SDLC's phases. The waterfall paradigm enables segmentation and control to depend on the progression of steps. This strategy is appropriate for small-scale software delivery with well-defined deliverables. Figure 2, depicts the waterfall paradigm

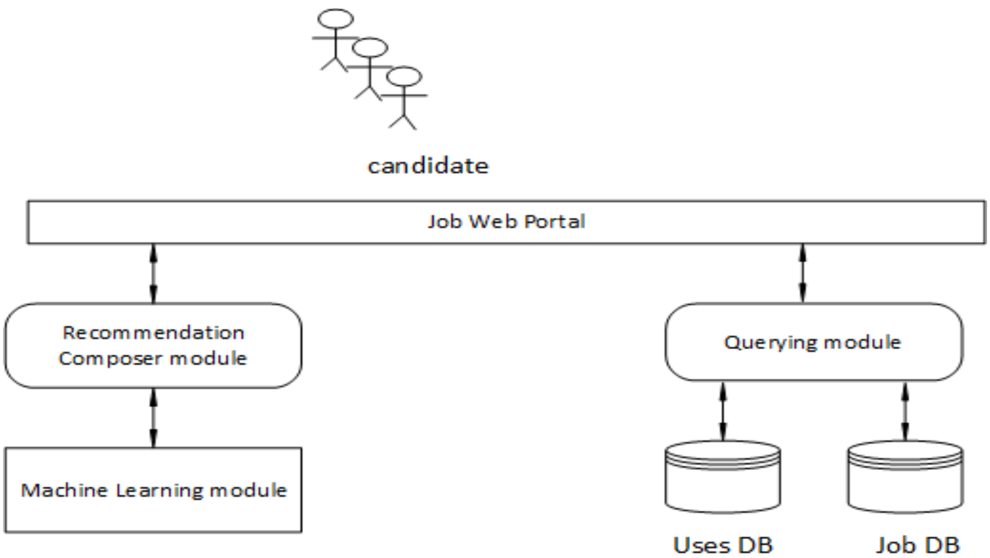

Figure 1. Architectural design
Source: Authors

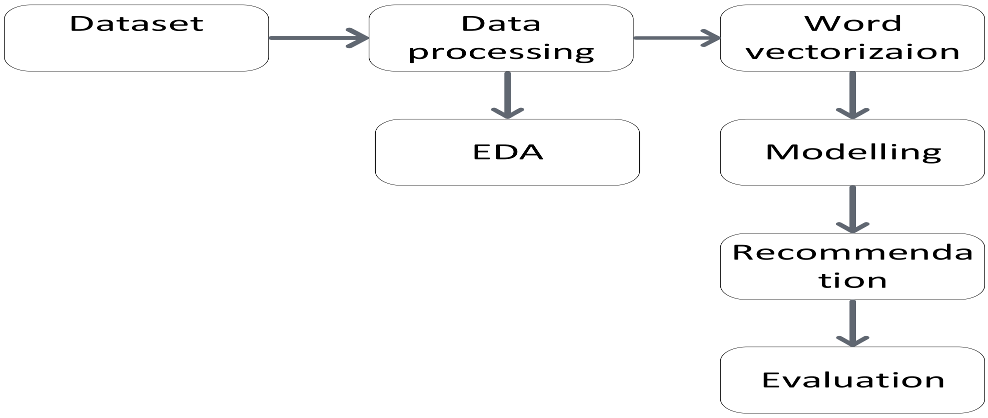

Figure 2. Pipeline for modelling data
Source: Authors

Functional and Non-functional Requirements

The functional requirements are directives on what the system must carry out to satisfy the user's expectations. The features that the user notices can be thought of as functional requirements, and the considered design they include (1) to predict if the employee is qualifying (2) to train and test data (3) to visualize the data and (4) to find d a position with the same requirements. The non-functional requirements (NFRs) are the collection of criteria that outline the capabilities, objectives, and constraints of the system while striving to enhance functionality. They delineate the system's effectiveness, encompassing aspects such as speed, security, reliability, precision, and data integrity.

We use case diagrams (see Figure 3) – diagrammatic representation of system behaviors – to show the requirements for the complete or specific component of the system. The job filter is sent to the querying module by the recommendation composer module. The results collected from the databases using the job filter are then presented to the RCM via the querying module. The querying module then builds queries in accordance with the filters given to it, retrieve pertinent entries from the databases, and presents the output in raw and vectorized versions.

For the various vectorized computations and models, the vectorized format can be utilized directly as an input. The raw format can be used to provide suggestions that are readable by humans. Sub-recommendations are generated by several means using the RCM, after which it creates the final list of positions for the candidates to view. Although the behaviour of the recommending system is typically regarded as saturated, user preference in the system is unchanged in the job domain as the user rarely switches his desire to work on new skill sets or change his employment domain. Because job adverts expire after a role is filled, the suggested job would not be in the system. When it comes to content-based recommendations, there are two methods: (1) to examine the explicit content properties, and (2) to create the user and item profiles from content rated by users.

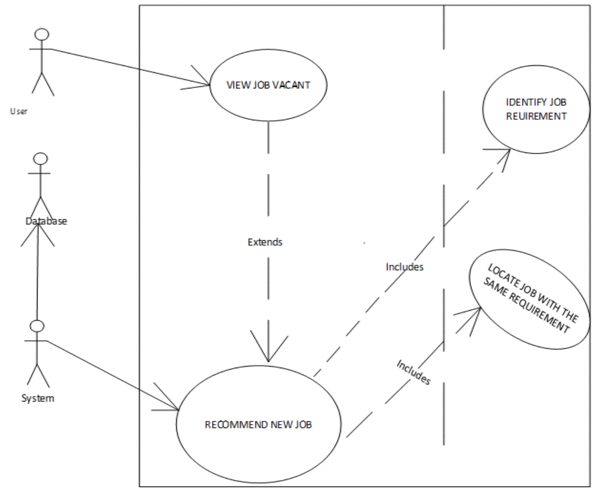

Figure 3. Use Case Diagram

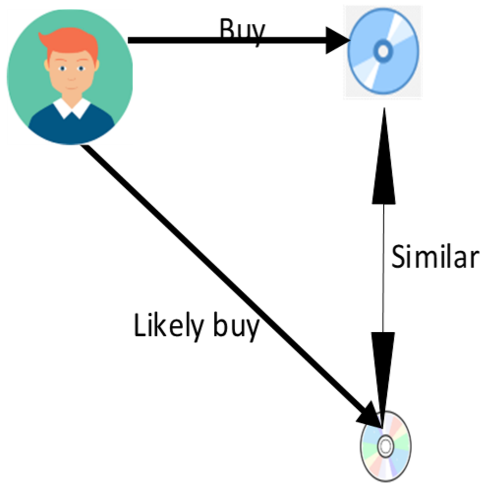

Figure 4. Content-Based Recsys

Data and Methods

The data, downloaded from Kaggle and the online portal(s), includes three columns - id, category, and résumé - are present in the data, which is in csv format. The 'id' is the résumé's sequential number, 'category' is the industry the job belongs to, and résumé is the candidate's full résumé. We first organized and transformed the data into format that is informative and usable for machine learning algorithms.

Since finding the résumés of the right candidates from the pool of submissions is the target, we train data to learn the evolution of job selection by candidates. Many studies have defined ways for integrating procedures to increase predictive accuracy. The techniques build an ensemble of hypotheses and aggregate the predictions of the models. The precision obtained by this set of hypotheses is superior to the precision of each individual component. The accuracy of the algorithms utilized, the quality and validity of the data used, and the relevancy of the job are aspects that affect how effective job recommender systems are.

We use the random forest and support vector machine to examine the accuracy of the system. The methods are machine learning approaches to forecast

https://doi.org/10.33022/ijcs.v13i5.4232

and categorize résumé into several categories that correlate to the various candidate specialties. The system makes it easier to select the best résumé from a sizable pool of candidates and to provide the best method in which candidate résumé would be well-structured based on order of the highest qualification. The approaches appear promising in the context of automatic job suggestions based on their performance for variable predictions. They are utilized to pinpoint the precise characteristics that users choose when they are drawn to a job offer. Their performance are scrutinized using evaluation metrics, including but not limited to accuracy, precision, recall, and F1-score.

Random Forest

The random forest (RF) algorithm is a robust tool for the construction of multiple decision trees during the training phase, each based on a random subset of the data. The algorithm aggregates the outputs of all trees, employing either a voting mechanism for classification or averaging for regression tasks. The flexibility and ability of RF to handle both classification and regression problems have continuously fueled its adoption, thus making it a cornerstone for the evaluation of an accurate job recommendation system. The random forest (regression) predictor is defined as:

$$\hat{f}_{\mathrm{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b) \tag{1}$$

B are number of trees {T(x;Θ_b)}_1^B grown, Θ_b characterizes the b th random forest tree in terms of split variables, cutpoints at each node, and terminal-node values.

Support Vector Machine

Support Vector Machine (SVM) is an algorithm used for classification (text or image), regression (linear or nonlinear), gene analysis, handwriting identification, spam detection, face detection, and anomaly (or outlier) detections. Because SVM can manage nonlinear, spatial and high-dimensional data, they are efficient and adaptable in multiple relationship applications including modeling recommendation scenarios. SVR achieves this flexibility with kernel functions, which map the input data into higher-dimensional spaces where linear relationships are more apparent. The problem finds the hyperplane that minimises training errors through slack variables as defined by equation (2):

$$\min_{\omega,b,\xi} 0.5\omega^T\omega + C\sum_{i=1}^{m} \xi_i \tag{2}$$

const: $y_i(\langle\omega, x_i\rangle + b) \geq 1 - \xi_i \ (i = 1, \dots, m)$.

Where ξ_i>0,ω are coefficients of the hyperplane, b is the bias term, ξ_i are slack variables and y_i (ω,x_i) is the mapping of input (x_i) into a higher-dimensional space.

D. System Analysis: Requirement, Implementation and Documentation

4.1. System Requirement

The minimum- hardware and software requirements for designing the job recommender system are highlighted. The hardware requirements are a minimum of 1GB of random-access memory; minimum 250GB of hard disk; a minimum Intel Pentium processor 1.5ghz; a monitor, keyboard, CPU/laptop; a mouse; and an internet connection. The software requirements are a working operating system (Windows 7/8/10, MacOS); A web browser (Chrome/Firefox) and a text editor (Jupyter Notebook/Vs Code).

4.2. System Implementation

The job recommender system was implemented in streamlit - a pPython template, for designing the frontend. For the system to relate properly, the admin page was created where the admin can see the list of qualified candidates for the job with the database server for fetching and storing the data. The system implementation is sub divided into admin interface, detail interface and predict interface. Figure 5 shows the admin sign-up/login, which presents a page showing the drop-down for admin to sign up/log in.

Figure 6 depicts the prediction page, where the company chooses the type of prediction (RV or SVM) to perform in deciding the choice of candidates. The page displays where the values or criteria required for the job are fit into the trained model for prediction. If the values or the criteria best match with the trained data, the machine will predict Yes, otherwise the prediction will be No. Figure 7 gives the admin page, which displays a table populated with the list of the accepted candidates. For the designed software, the admin must create an account from the admin page and login in with the details, before he can access the accepted candidates list. If the login detail is not correct, the error message will pop up. Finally, the visualization page for RF (Figure 8) and SVM (Figure 8) are depicted. The visualization page with the frequency in category (Class 0 denoted No, Class 1 denoted Yes). If the candidate qualifies for the job, the graph will be on class 1, otherwise the graph will move to class 2.

The two algorithms support vector and random forest classification proofed efficacy toward the accurate prediction. It was found out that the two models predicting outcome are the same and this makes them to be recommend for the future works. Although Random Forest behaves slow in ranking the most qualified candidates among the pools of candidates with the same qualifications but have little differences, while the Support Vector Machine proved to be faster. The future of machine learning-based job recommender systems lies in their ability to cater to the diverse needs and preferences of individual job seekers. Job recommender systems can considerably increase the effectiveness and efficiency of the hiring process. Previous attempts at designing the job recommender systems via implementation of machine learning tools have no unique matching for individual job preferences, making online advertises to suggest unrelated jobs to individuals. This problem is caused due to poor design of algorithms and inadequate focus on personalization.

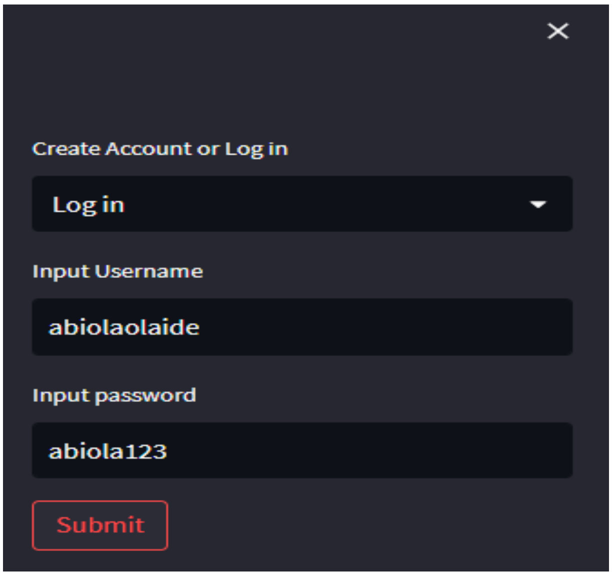

Figure 5. Admin log in and sign-up page.

Note: This shows a drop-down for admin to sign up/log in.

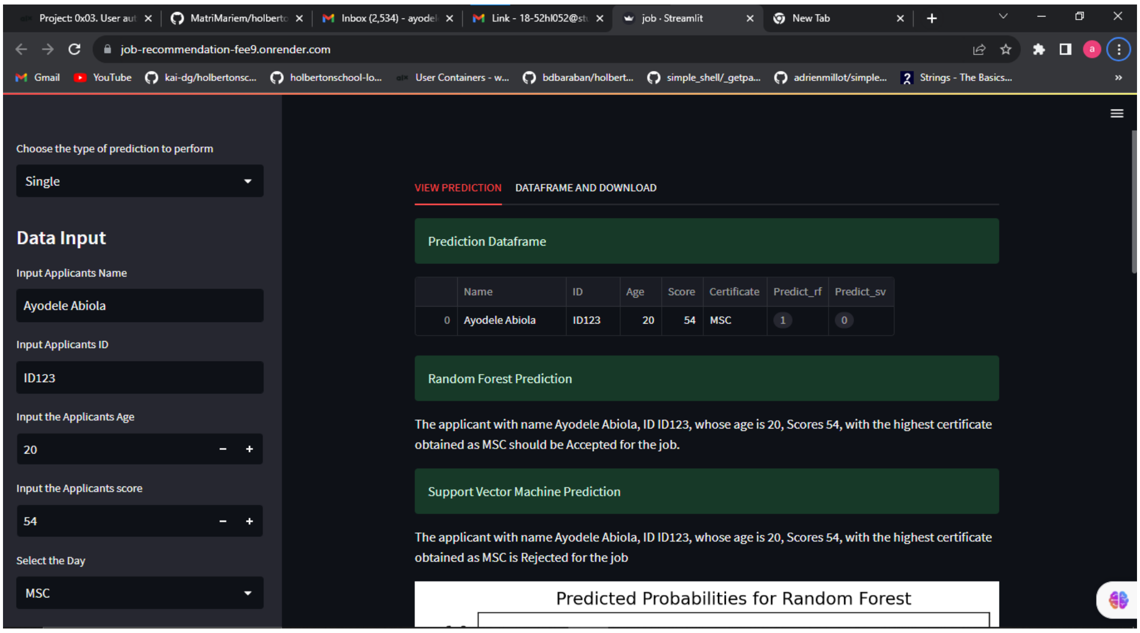

Figure 6. View prediction page.

Note: This page displays where the values or criteria required for the job are fit into the trained model for prediction. If the values or the criteria are best match with the trained data, the machine will predict Yes, otherwise the prediction will be No.

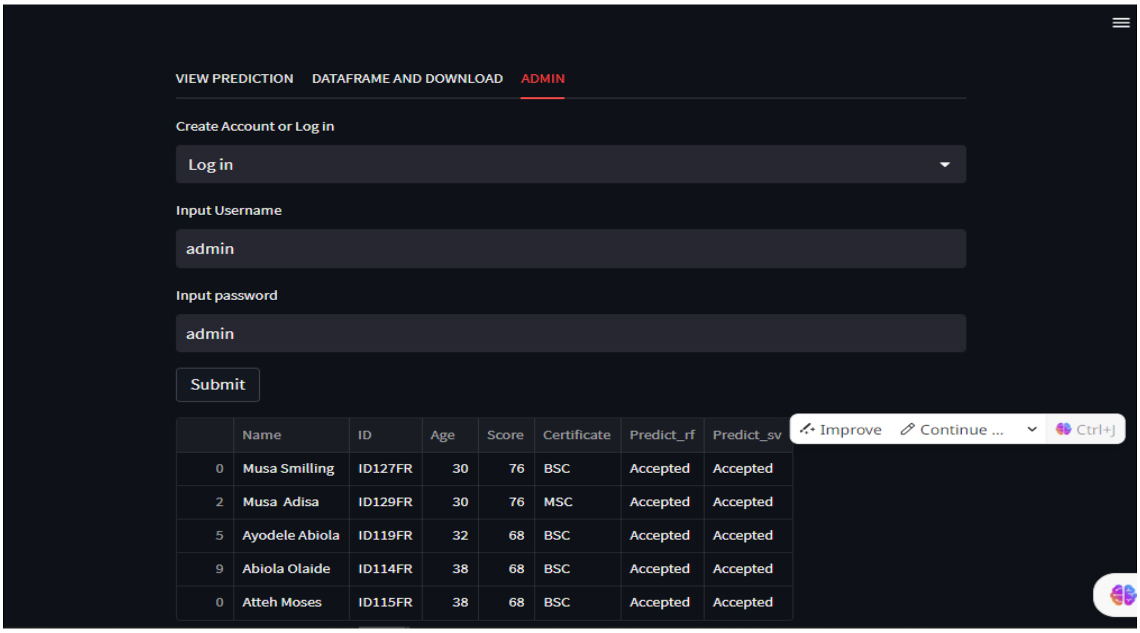

Figure 7. Admin page

Note: The page with a table populated with the list of the accepted candidates. The admin must create an account, from the admin page and login in with the details, before he can be able to access the accepted candidates list. If the login detail is not correct, the error message will pop up.

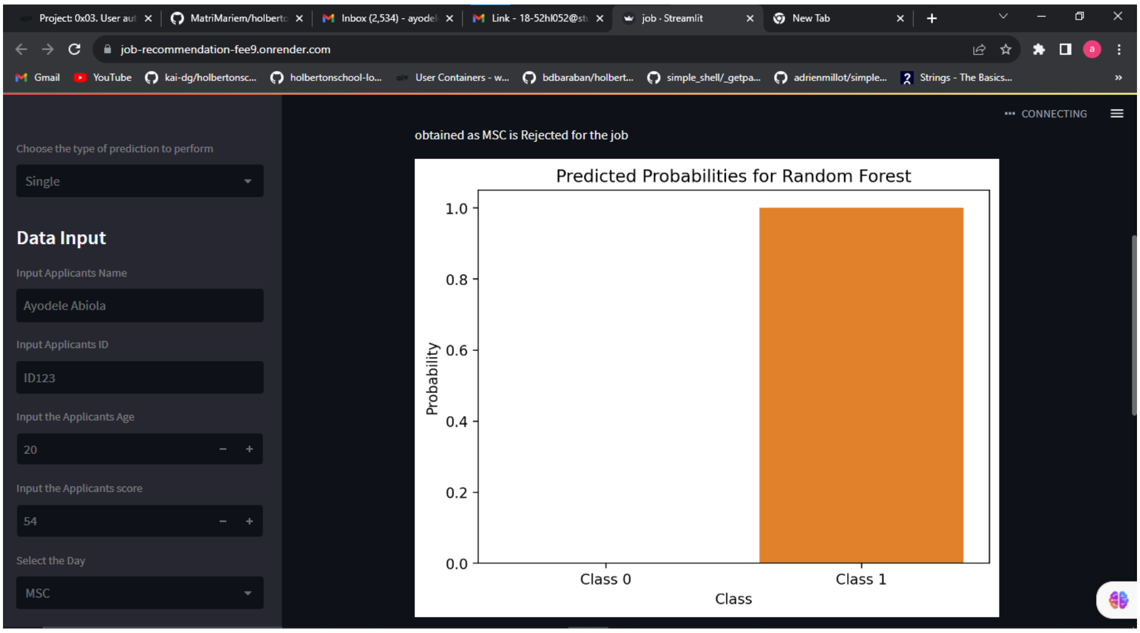

Figure 8. Visualization page for RF

Note: The visualization page with the frequency in category (Class 0 denoted No, Class 1 denoted Yes). If the candidate qualifies for the job, the graph will be on class 1, otherwise the graph will move to class 2.

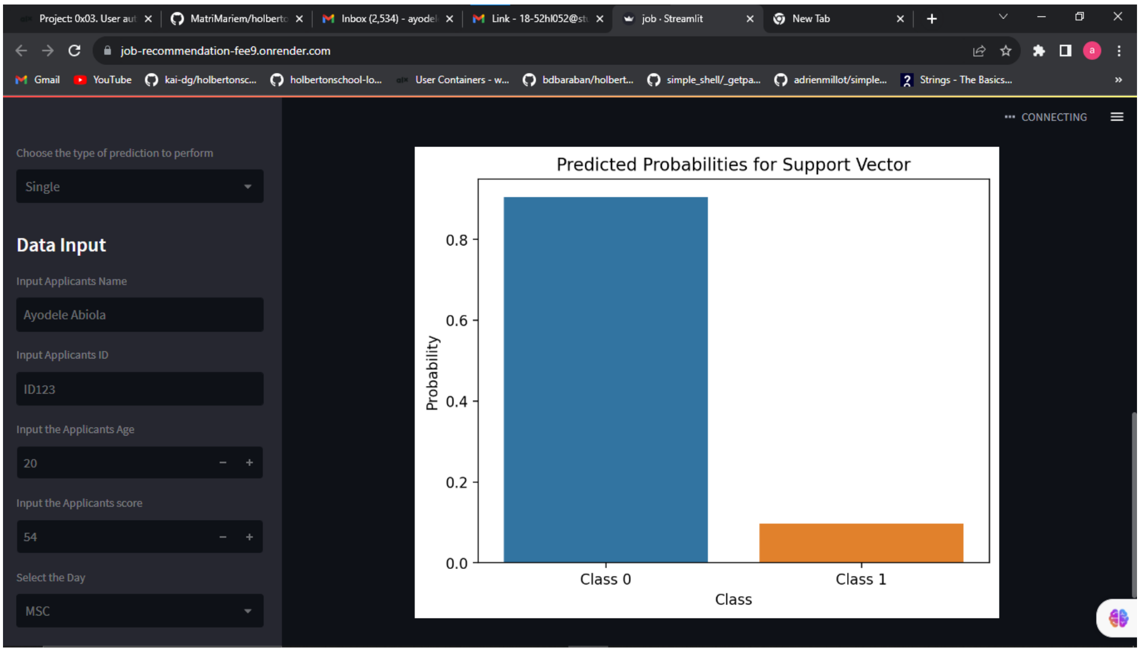

Figure 9. Visualization page for SVM

Note: The visualization page with the frequency in category (Class 0 denoted No, Class 1 denoted Yes). If the candidate qualifies for the job, the graph will be on class 1, otherwise the graph will move to class 2.

4.3. System Documentation

The system documentation of the job recommender system describes the activities and maintenance of the system at the end user's side. The maintenance of the system by the users is very important and crucial to avoid any future unforeseen occurrences or failures which may happen to the system if not properly maintained. The process of system documentation is called the 'User's Guide' which describes the operations of all interfaces (login, data input and prediction interface). The user's guide is summarized:

1. How to access the home page:
   Step 1: Enter domain name http://job.recommender-fee9.onrender.com on browser.
   Step 2: Press Enter keyboard.
2. How to login to the admin dashboard:
   Step 1: Select from the drop-down option to select Login or create account option
   Step 2: Enter your Email and password then click on submit.
3. How to select prediction choice:
   Step 1: The requirement criteria such as name, age, score and degree.
   Step 2: Streamlit automatically predicts if the candidate is qualified for the job based on the requirements needed for the prediction.
4. How to access the frequency distribution of predicted outcome:
   Stroll down to the bottom of the page, there are graph for the two models (RF and SVM) in which the visualization is based on their accuracy

E. Conclusions

The paper focuses on user-centric design and personalization to develop a job recommender system for recommending and matching applicants with appropriate jobs. It makes accurate predictions by comparing résumé and job descriptions using machine learning algorithms. The system identifies careers that the candidates will enjoy and assign to them based on the job descriptions. Any candidate can use this programme to learn more about their ideal professions and to develop both their soft talents and their hard skills. They will benefit from it because they won't have to waste time looking for work. They can also develop their expertise in that field and advance more quickly.

Based on a job seeker's qualifications, aptitude test's result, age, and work experience, the job recommender system offers personalized job recommendations. The system represents a more effective tool that assists job seekers to find the right opportunities according to their unique preference. The development has been designed in such a way that it is adaptable across various fields. The developed software offers individualized jobs based on each person's experiences and interests contained on the résumé, therefore mitigate high-level prejudices from human resources in the hiring process. MoresoMore so, the developed software is designed to accommodate future updates that will increase the system and satisfy more user requirements. We suggest that system maintenance must be routine for more effective utilization of the software.

Although job recommender systems have the potential and ability to increase the efficacy and efficiency of the hiring process, they have significant limitations that must be resolved to ensure unbiased and fair suggestions. The developed systems have drawbacks, too, like the potential for bias and the requirement for ongoing updates to guarantee accuracy. We identify the following likely limitation for the system. There is likelihood of bias in the utilized data for training the algorithms, which could lead to suggestions that were unfair. The suitability of the position recommendation depends on the accessibility and standard of the data used to train the model. The algorithm might not be precise if the data is inaccessibility, inaccurate or out-of-date. There is also the risk of overfitting the designed. This may occur if the model is very complicated or the training data for the model was insufficient and may likely make the model perform well on training data but poorly on new data. Lastly, inability to capture some non-cognitive qualities such as communication and teamwork, which are essential for job success but not considered by job recommender systems.

Despite these limitations, the outcome still leads to more fulfilling and successful job matches while respecting user privacy and preferences. Therefore, we suggest that corporations, higher institutions, and ministries should adopt the software to ease job requirements process.

F. References

[1] Abel, F., Benczúr, A.A., Kohlsdorf, D., Larson, M., & Pálovics, R. (2016). RecSys Challenge 2016: Job Recommendations. Proceedings of the 10th ACM Conference on Recommender Systems.

[2] Afoudi,Y., Lazaar, M. & Al Achhab, M. (2021). Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial

neural network. Simulation Modelling Practice and Theory. Volume 113, December 2021, 102375. https://doi.org/10.1016/j.simpat.2021.102375
[3] Al-Otaibi, S. T., & Ykhlef, M. (2012). A survey of job recommender systems. International Journal of the Physical Sciences Vol. 7(29), pp. 5127-5142. DOI: 10.5897/IJPS12.482
[4] Bhatia, V., Rawat, P., Kumar, A., & Shah, R. A. (2019). End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT. Information Retrieval, https://doi.org/10.48550/arXiv.1910.03089
[5] Bobadilla, J., Ortega, F., Hernando, A. & Gutiérrez, A. 2013). Recommender systems survey. Knowledge-Based Systems. Volume 46, pages 109-132, https://doi.org/10.1016/j.knosys.2013.03.012
[6] Bolandraftar, M. & Imandoust, S. B. (2017). Theoretical Background for the K-Nearest Neighbour (KNN) Approach in Economic Event Prediction. International Journal of Engineering Research and Applications, 3(5), 605.
[7] Buschner, S., Schirru, R., Zieschang, H., & Junker, P. (2014). Providing recommendations for horizontal career change. International Conference on Knowledge Management and Knowledge Technologies.
[8] Celma, O. (2010) Music recommendation. In music recommendation and discovery (pp. 43–85). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13287-2
[9] Diaby, M., Viennet, E., & Launay, T. (2014). Exploration of methodologies to improve job recommender systems on social networks. Social Network Analysis and Mining, 4, 1-17.
[10] Diaby, M., Viennet, E., & Launay, T. (2018). An online social network-based job recommendation system is the next generation of recruitment tools. ASONAM, 821–828.
[11] Eckhardt, A, & Laumer, S. (2017). Integrating recommender systems in an IT-supported staff recruitment strategy can assist in finding the needle in the haystack. CPR 2009, 7-12.
[12] Faliagka, E., Iliadis, L. S., Karydis, I., Rigou, M., Sioutas, S., Tsakalidis, A. K. & Tzimas. (2014). Looking for the truth behind a well-written CV through online consistent ranking on e-recruitment. Artificial Intelligence Review, 42(3), 515-528.
[13] Fogelman-Soulie, F., Fournier, R., Bernardes, D., Diaby, M. & Viennet, E. (2018). A Survey and Social Formalism for Recommender Systems. SIGKDD Explorations, 16(2),2037.
[14] García-Sánchez, F., Colomo-Palacios, R. & Valencia-García, R. (2020). A social-semantic recommender system for advertisements. Information Processing & Management, Volume 57, Issue 2, March 2020, 102153. https://doi.org/10.1016/j.ipm.2019.102153.
[15] Lin, Y., Lei, H., Addo, P. C., & Li, X. (2016). Matching résumé to jobs using machine learning. Preprint for arXiv:1607.07657, 1–8.
[16] Maier, C., Laumer, S., Eckhardt, A., & Weitzel, T. (2012). When Social Networking Turns to Social Overload: Explaining the stress, Emotional Exhaustion, and Quitting Behavior from Social Network sites' Users. European Conference on Information Systems.
[17] Malinowski, J., Keim, T., Wendt, O. & Weitzel, T. (2016). Bilateral

recommendation approach for matching job candidates and employers. HICSS 2006.
[18] Mitova, E., Blassnig, S., Strikovic, E., Urman, A., Hannak, A., de Vreese, C. H., & Esser, F. (2022). News recommender systems: a programmatic research review. Annals of the International Communication Association, 47(1), 84–113. https://doi.org/10.1080/23808985.2022.2142149
[19] Mooney, R. J. & Roy, L. (2000). Content-based book recommending using learning for text categorization. Proceedings of the fifth ACM conference on Digital libraries, June 2000, Pages 195 – 204. https://doi.org/10.1145/336597.336662
[20] Mrzic, E., & Zaimovic, T. (2020). Data Science Methods and Machine Learning Algorithm Implementations. School of Economics and Business, Department of Information Systems Development, University of Sarajevo, Sarajevo, Bosnia and Herzegovina.
[21] Mustafal, N., Ibrahim, A. O., Ahmed, A. & Abdullah, A. (2020). Collaborative Filtering: Techniques and Applications. 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), Khartoum, Sudan
[22] OpenAI. (2022). GPT-3.5 Language Model (software for computers). Retrieved from https://openai.com/technology/chatpt/ on April 5, 2023.
[23] Paparrizos, I. K·, Cambazoglu, B. B. & Gionis, A. (2011). Machine learned job recommendation. RecSys '11: Proceedings of the fifth ACM conference on Recommender systems, Pages 325 – 328 DOI: 10.1145/2043932.2043994
[24] Priyanka, K., Tewari, A. S. & Barman, A. G. (2015). Personalised book recommendation system based on opinion mining technique. Global Conference on Communication Technologies (GCCT), Thuckalay, India, 2015, pp. 285-289, doi: 10.1109/GCCT.2015.7342668.
[25] Rhanoui, M., Mikram, M., Yousfi, S., Kasmi, A. & Zoubeidi, N. (2022). A hybrid recommender system for patron driven library acquisition and weeding. Journal of King Saud University - Computer and Information Sciences, Volume 34, Issue 6, Part A, June 2022, Pages 2809-2819. https://doi.org/10.1016/j.jksuci.2020.10.017
[26] Riyahi, M. & Sohrabi, M. K. (2020). Providing effective recommendations in discussion groups using a new hybrid recommender system based on implicit ratings and semantic similarity. Electronic Commerce Research and Applications, Volume 40, March–April 2020, 100938. https://doi.org/10.1016/j.elerap.2020.100938
[27] Schafer J, Konstan J, Riedl J (1999). Recommender Systems in ECommerce. In Proceedings of the First ACM Conference on Electronic Commerce (EC 1999), Denver, Colorado, USA, ACM pp. 158-166
[28] Taneja, S., Gupta, C., Goya, K., & Gureja, D. (2017). An Improved K-Nearest Neighbour Algorithm Using Clustering and Information Gain. The fourth international conference on advanced communication and computing technologies.
[29] Wei, K., Huang, J., & Fu, S. (2007). A survey of e-commerce recommender systems, in: 2007 international conference on service systems and service management, IEEE, 2007, pp. 1–5.

[30] Xu, M., & Liu, S. (2019). Semantic-enhanced and context-aware hybrid collaborative filtering for event recommendation in event-based social networks. IEEE Access 7:17493–17502. https://doi.org/10.1109/ACCESS.2019.2895824
[31] Yalcin, E., Ismailoglu, F. & Bilge, A. (2021). An entropy empowered hybridized aggregation technique for group recommender systems. Expert Syst Appl 166:114111. https://doi.org/10.1016/j.eswa.2020.114111
[32] Yu, M., He, W., Zhou, X., Cui, M., Wu, K. & Zhou, W. (2022). Review recommendation system. Southwest University for Nationalities Press, Chengdu.
[33] Z., Junshu, W., Nan, J., & Yehua, S. (2018). A Point-of-interest Recommendation Method Based on Hawkes Process. DOI:10.11947/j.AGCS.2018.20170552.