International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Job Recommendation System Implementation in Python vs. C++

Chanda, Rajesh Kumar Aggarwal

Abstract: ***Implementing a machine learning algorithm gives you a deep and practical appreciation for how the algorithm works. This knowledge can also help you to internalize the mathematical description of the algorithm by thinking of the vectors and matrices as arrays and the computational intuitions for the transformations on those structures. There are numerous micro-decisions required when implementing a machine learning algorithm, like Select programming language, Select Algorithm, Select Problem, Research Algorithm, Unit Test and these decisions are often missing from the formal algorithm descriptions.***

The notion of implementing a job recommendation (a classic machine learning problem) system using to two algorithms namely, KNN [3] and logistic regression [3] in more than one programming language (C++ and python) is introduced and we bring here the analysis and comparison of performance of each. We specifically focus on building a model for predictions of jobs in the field of computer sciences but they can be applied to a wide range of other areas as well. This paper can be used by implementers to deduce which language will best suite their needs to achieve accuracy along with efficiency We are using more than one algorithm to establish the fact that our finding is not just singularly applicable.

Keywords : algorithm comparison, C++ vs. python, data analysis, job recommendation, K-nearest neighbor, logistic regression, machine learning..

I. INTRODUCTION

With the advent of technology and digitization, one of the biggest reforms has been that of online job searching. Earlier it was a common scenario that even though there were available job opportunities but unemployment prevailed, mainly, due to the gap between the job seeker and the recruiter. This gap has been bridged by the internet.

Manuscript published on November 30, 2019.
* Correspondence Author
Chanda, Student*, Department of University School of Information, Communication and Technology (U.S.I.C.T), Guru Gobind Singh Indraprastha University (G.G.S.I.P.U), New Delhi, India.
Rajesh Kumar Aggarwal, Student, Department of University School of Information, Communication and Technology (U.S.I.C.T), Guru Gobind Singh Indraprastha University (G.G.S.I.P.U), New Delhi, India.

There is a cornucopia of job recommenders available that make the process of job searching a piece of cake. There is no longer the need to skim through the classified section to look for jobs. A job recommender system is provided with the various qualities an organization is looking in their prospective employees.

The job seeker may post their credentials online and the recommender will find the suitable match. We can build a recommender system using various algorithms and implement in any language of our choice, but will all of them result in a system of equal efficacy? This paper will focus on the problem of choosing the best method/algorithm and language for job recommender system. We will use **C++,** a general purpose, widely used programming language with imperative, object-oriented and generic programming features. C++ runs on lots of platform like Windows, Linux, Unix, Mac etc, and **Python** a widely used general-purpose, high-level programming language that allows programming in Object-Oriented and Procedural paradigm just like C++, however the programs are much smaller than other programming languages and its main strength lies in its flexibility for being used for machine learning, text processing, scientific computing, image processing etc.
We also examine which machine learning algorithm works better viz. KNN or logistic regression. There are some already existing findings on the comparison of languages such as:
Popularity of machine learning languages. Python leads the pack, with 57% of data scientists and machine learning developers using it and 33% prioritizing it for development. C/C++ is a distant second to Python, both in usage (44%) and prioritization (19%).
The most decisive factor when selecting a language is the type of project you'll be working on — your application area. Machine learning scientists working on sentiment analysis prioritize Python (44%).Artificial Intelligence (AI) in games (29%) and robot locomotion (27%) are the two areas where C/C++ is favored the most, given the level of control, high performance and efficiency required.
Second to the application area, the professional background is also pivotal in selecting a machine learning language. Python is prioritized the most by those for whom data science is the first profession or field of study (38%).

This indicates that Python has by now become an integral part of data science — it has evolved into the native language of data scientists. C/C++ is prioritized more by those who want to enhance their existing apps/projects with machine learning (20%) and less by those who hope to build new highly competitive apps based on machine learning (14%).

The paper is arranged as follows. In section II, formal definition of Job recommender and various matching criteria for evaluating similarity between jobs and job seekers. We summarize methods and materials used in section III. Section IV displays the successful machine learning models results for recommendations. Experimental results with some classic machine learning models in Section V, Finally Section VI concludes the paper.

II. JOB MATCHING AND SIMILARITY CRITERIA

Job seekers search for the job positions whereas the big companies or enterprises find appropriate candidates for the open positions. So the useful information is 'user-user' interactions, 'user-item' interactions and 'item-item' interactions. Now to achieve good matching between jobs and users, similarity calculation is to be performed. Various measures to calculate similarity index are present such as Constrained Pearson Correlation, Pearson Correlation [2], Spearman Rank Correlation [2], Cosine [2][11], Mean Squared Differences and many more have been used in the recommendation systems. For instance, the similarity between two users or two jobs or between user and a job can be obtained by any of the above mentioned methods, For example the similarity can be calculated by cosine method which is defined as

$$\text{CosineSim(a,b)} = \frac{\vec{a}.\vec{b}}{|\vec{a}||\vec{b}|} \text{ where } \vec{a}.\vec{b} \text{ is the dot product.}$$

'a' and 'b' can be user and job, user and user, job and job respectively.

CosineSim(a,b) gives the similarity measure score between a and b where a and b are the vectors in the dimensional space.

The correlation can also be determined by Pearson coefficient as follow:

$$\text{PCSim(a,b)} = \frac{\sum_{i \in O_{(a,b)}} (r_{(a,i)} - \bar{r}_a)(r_{(b,i)} - \bar{r}_b)}{\sqrt{\sum_{i \in O_{(a,b)}} (r_{(a,i)} - \bar{r}_a)^2} \sqrt{\sum_{i \in O_{(a,b)}} (r_{(b,i)} - \bar{r}_b)^2}},$$

Where $r_{(a,i)}$ refers to the user rating for objecti and $\bar{r}_a$ represents the average rating of user u. $O_{(a,b)}$represents the set of items which are rated by a as well as b.

III. METHODS

For the implementation of the algorithms the first and foremost requirement is that of data. It must be in a suitable format such that it is easy to understand and manipulate such as .txt or .csv We may use a simple text editor such as notepad or Microsoft Excel for this purpose. The data has been collected through the analysis of various already existing job advising websites. It consists of various variables that effect the final output of job recommendation such as:

expScore- assigns a score to a candidate on the basis of his/her previous experience,

degScore- gives a rating to the type of degree the candidate holds,

collegeScore- according to this parameter every college is given an overall ranking,

marksScore- value tantamount to the marks obtained by the candidate in college.

jobPredicted- the output parameter that gives the recommended job for the candidate according to his/her above credentials .

We may visualize the data through graphs and histograms and then implement our algorithms in a suitable environment. For implementation purposes, in python, we use an open source application such as Jupyter Notebook or a free computer program for scientific computing such as Anaconda ,whereas to implement in C++ we may use an IDE such as CodeBlocks. All these desktop application are available online and can be downloaded via internet and run a 64-bit, Intel i3 6th generation processor.

Once we have the requisite material we may write our code implementing the algorithms from scratch i.e. we may not use ready-made function calls, rather we may write everything from scratch. On execution we may analyze and compare the performance for the job recommendation system system using Mean Average Precision (MAP)[9] and RMSE.

IV. RESULT

The data can be analyzed by visualizing graphically through various functions available to us through the "matplotlib" library. From figure 1 we may observe the distribution with respect to the marksScore vs collegeScore for a part of our data set and Figure 2 depicts the distribution with respect to degScore vs expScore. We have considered three arbitrary job tyes namely jobType-A , jobTupe-B and jobType-C. Once we have setup our system we may simulate to predict the outcomes and analyze the time taken for the output as well as the accuracy. The graphs obtained are as follows:

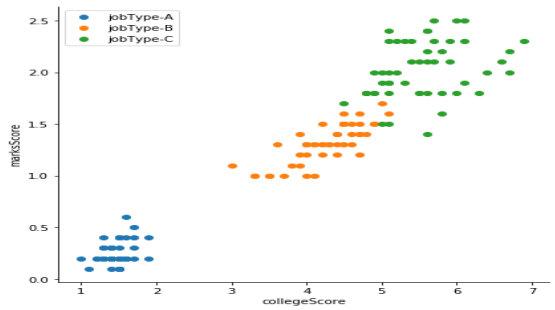

Figure:1

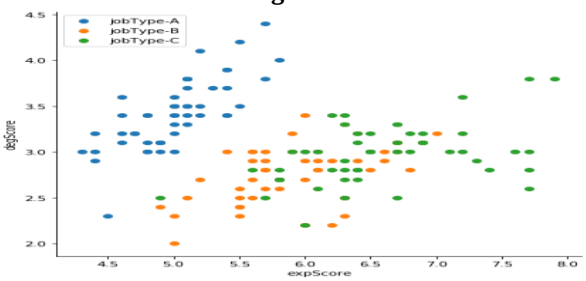

Figure:2

We get the following result:

Table I: Performance of algorithms implemented in C++

Algorithm	N-value	MAP	RMS Evaluation Index	Average Execution time
Logistic regression	2	0.6919	1.3976	5.5458s
	7	0.7833	1.4456	
	11	0.8207	1.629	
KNN	2	0.7981	0.9921	4.5092s
	7	0.8554	1.1193	
	11	0.9333	1.203	

Table II: Performance of algorithms implemented in Python

Algorithm	N-value	Accuracy	Average Execution time (1000 loops)
Logistic Regression	2 7 11	96.6667% 97.1193% 95.9956%	1.7ms ± 252µs per loop
KNN	2 7 11	93.8775% 98.3261% 96.1177%	33.5 ms±5.26µs per loop

V. CONCLUSION

This paper provides a conclusive means to analyze the performance of various machine learning algorithms implemented in python and c++ language. Through this paper one may be able to make a more informed decision regarding choice of language and algorithm to use. According to the findings, the average execution time for algorithms implemented in python is less and the Accuracy is better as compared to the algorithms implemented in c++. So we may conclude that if the main point of concern is to make the job recommendation system economical in terms of time and accuracy then python should be the choice. In python, out of logistic regression and KNN, later gives a better performance in terms of accuracy, however it takes more time thus there is a trade-off between the two parameters. To implement in python, we need to learn the way to use various functions that the libraries offer because it is because of them that we get a better performance. Thus, it needs knowledge in addition to the basic object-oriented principles of the language. This is not required in C++.The implementation uses the basic object-oriented concepts like classes, objects and inheritance and data structures.

VI. DICUSSION AND LIMITATION

The main idea behind the model is that the algorithm shall learn from the past pattern of the data and when the user inputs his credentials the algorithm may give a suitable output based on the previous learning. However, a case may arise when the user may be incapable to provide appropriate value to the input parameters, such as the candidate may be missing the marks for one of his semesters, in this case our model will be rendered incapable to provide an accurate prediction.

To train the model we divide the data available into training as well as testing data. But the biggest conundrum is to find the ratio in which to divide the data set. We tried various combinations and found training set should be around 67% of the complete data set. If we increase the percentage then we don't get conclusive results from the testing dataset and if we decrease it then the model may not get properly trained, thus reducing the accuracy.

Our models suffers from another problem of cold-start, in which the system cannot predict any logical consequence for the user about which it has not sufficient knowledge yet.

There could be a '**New Community'**- refers to the start-up of a recommender when although the initial dataset is available it lacks the interaction with any user or

'**New item'-** a new item is added in the system and hasn't had any interactions yet or '**New User'**- a new user has registered on the system and hasn't provided any information yet, thus the model cannot provide any personalized recommendations.

REFERENCES

1. Shaha T. Al-Otaibi1* and Mourad Ykhlef2, "A survey of job recommender systems", International Journal of the Physical Sciences Vol. 7(29), 26 July 2012, pp. 5127-5142.
2. Zheng Siting, Hong Wenxing*, Zhang Ning, Yang Fan, "Job Recommender Systems: A Survey", The 7th International Conference on Computer Science & Education (ICCSE 2012) July 14-17, 2012. Melbourne, Australia
3. Machine Learning Course by Andrew Ng, Coursera.
4. Walid Shalaby1, BahaaEddin AlAila2, Mohammed Korayem3, Layla Pournajaf3, Khalifeh AlJadda3, Shannon Quinn2, and Wlodek Zadrozny1, "Help Me Find a Job: A Graph-based Approach for Job Recommendation at Scale", 2017 IEEE International Conference on Big Data (BIGDATA).
5. Sarah Guido, Andreas Müller, Introduction to Machine Learning with Python-A Guide for Data Scientists,2016, O'Reilly Media,285
6. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th international conference on World Wide Web, pp. 285–295, ACM, 2001..
7. Minh-Luan Tran, Anh-Tuyen Nguyen, Quoc-Dung Nguyen, Tin Huynh, "A Comparison Study for Job Recommendation", KICS-IEEE International Conference on Information and Communications with Samsung LTE & 5G Special Workshop (2017).
8. Pazzani M.J., Billsus D. (2007) Content-Based Recommendation Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg
9. Vivek Singh Chawla and R.K Aggarwal, "Exploring Machine Learning Algorithms for Job Recommendation System" International Conference on Recent Developments in Computer & Information Technology (ICRDCIT), International Society for Scientific Research and Development (ISSRD), New Delhi, 12thJune 2019.
10. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, 2019 , O'Reilly Media, Inc.
11. J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, "Evaluating collaborative fltering recommender systems," ACM Transactions on Information Systems 22 (2004) 5-53.

AUTHORS PROFILE

Chanda is a final year bachelors in technology student from University School of Information, Communication, and Technology, Guru Gobind Singh Indraprastha Univresity, Delhi.
She has a strong acumen in research work.
She has been a research intern at NIT, Kurukshetra.