Recommending industry jobs with machine learning

Pedro Seguel * [1] **Shih-Tsai Wei** * [1]

Abstract

This research focuses on the application of machine learning techniques using job advertisement and recruitment data to enhance the search strategy of candidates and help them to identify the gap between the job post requirement and their current ability. We expand current research by focusing on the role of the predictive character that hard and soft skills have over specific industry jobs among the same job domain. We conduct two experiments. First, we compare the performance of a Decision Tree, and K-Nearest Neighbor adapted for a multi-label classification problem, and, second, we analyze how the data size affects each model. With our preliminary results, we found that imbalance data on industry group labels affect the predictions. Also, K-Nearest Neighbors provides better performance than the decision tree, but the learning curve is worse. Finally, we propose a user interface design prototype. We provide ideas for future work to expand the current design and provide better performance.

1. Introduction

The internet has shaped the way people search for jobs nowadays (Jansen et al., 2005). The popularity of job market sites such as Indeed, Monster or LinkedIn provides an organized set of job offerings where employees can find their dream job and apply present or prepare for it in the future. Although these platforms help the potential candidate to evaluate specific jobs positions to apply, they don't provide an updated overview of the market trends that can help them to see the big picture.

We applied machine learning (ML) techniques using job advertisement and recruitment data to enhance the search strategy of candidates and help them to identify the gap between the job post requirement and their current ability. We expand current research by focusing on the role of the predictive character that hard and soft skills have over specific industry jobs among the same job domain. We extend prior research on job market analysis by using automated techniques of ML, using the soft skill in our dictionary, and by proposing a user interface for job-seekers.

We leveraged a labeled dataset based on 686 jobs posting information for marketing internships extracted from Indeed to train our multi-label classification models. Our dataset includes labels for 31 job skills, which are also grouped between soft and hard skills. Using a supervised approach, we will be able to explore how these diverse skills are predictors of getting a job in a desirable industry. Moreover, this project introduces a user interface prototype that can use this information to recommend people how to map their careers.

2. Related Work

Job market analysis using job advertisements Prior research has provided an in-depth analysis of specific job market trends using online job postings and advertisements for domain-specific professions (Ritzhaupt et al., 2010; Jansen et al., 2005; Sugar et al., 2012). For instance, Ritzhaupt et al. (2010) paper examines the multimedia competencies of educational technologists via combining insights from job posting analysis and a survey of professionals within the field. This in-deep study highlights changes in job trends, and how the general curriculum match with those job trends, showing the opportunities of using job posting data to understand variations on the demand side of the market. However, as insightful these studies are, limitations arise from the use of standard statistical techniques for analysis and coding of the data on these studies. Our approach is distinctive from this research because we plan to take an automated approach to use the text-data that is available related to the job search process, and we provide broader suggestions to job seekers about their suitable industries instead of specific job postings. This work will be beneficial especially to people who don't have a clear preference regarding the industry. Also, we will present a UI prototype for users to visually give our career suggestions and provide a better user experience.

[1] The University of Texas at Austin. Correspondence to: Pedro Seguel <ppseguel@utexas.edu>, Shih-Tsai Weid <stwei@utexas.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

Matching jobs and applicants Based on published papers on applied machine learning to the online job seeking process (Guo et al., 2016; Fazel-Zarandi & Fox, 2009; Poch et al., 2014), for instance, Poch et al. (2014) explored the use of supervised classifiers to learn implicit relations between resumes and jobs offers, which cannot be found with similarity or pattern based search skills, names of professions and degrees. While this type of research tackles the broader challenges of matching resumes with job descriptions and desirable characteristics for an employee, our project focus on a more specific area of the job searching process. Our interest is on the job-seeker that is planning its training for a future application. Consequently, we focus on automating the process of labeling a type of job using job skills features, rather than semantically matching an existing resume with a job offer. Also, by focusing on more specific distinctions between job skills (including hard and soft skills), we plan to visualize hidden patterns present on the job market under the same job title, or for our case marketing interns.

Job skills relatedness Finally, another interesting -and less frequent- line of work focus on the relatedness among job skills using text mining techniques (Hughes, 2015; Van-Duyet et al., 2017; Lau & Sure, 2002). Van-Duyet et al. (2017) develop a method called Skill2vec, which applies machine learning techniques in recruitment to enhance the search strategy regarding candidates possessing the appropriate skills. While Van-Duyet et al. (2017) creates a useful architecture to automate this process, it's based primarily on a general dictionary of technical skills. On the other hand, our work focuses on predictive classification techniques using labeled job skill data to explore the relationships of skills and industries. Also, we expand current research on job skill by adding soft, and hard skills to the analysis.

3. Methods

We decided to use a domain-specific dataset that could provide rich text information from the real job posting information, as well as a set of labeled skill data coded from those text descriptions. We will use Woods (2017) dataset on marketing internships job postings from a diverse set of enterprises in the U.S.A., which includes 760 job postings collected at the beginning of February 2017 from Indeed.com. We plan to use the list of 31 hard- and soft-skills features which had been coded in the original dataset by its author. The skills were coded using the hot encoding process, treating each skill as a dummy variable and assigning number 1 when the skills were present and 0 when it was not.

Since the targeted labels for the industry were not available on the original dataset, we coded them by hand using the following protocol. First, we coded the industry information from the LinkedIn profile from each company named on the dataset. Since each company had to select and self-identified themselves with one industry category among the total 149 different options created by LinkedIn, we interpreted this information as an accurate self-report of industry. Also, since the options are fixed and the industry is well self-defined, the resources are unified and reliable. Later, we deleted part of the companies data which don't have official company Linkedin page because we couldn't decide the industries they belonged. A total of 686 job postings were used for our analysis. At last, we recorded these industry labels to an industry group label of 17 main groups used by LinkedIn (Table 1) (LinkedIn, 2019). Take Diebergs as an example; according to their Linkedin official web page, its industry is retail. So, after checking the industry code using the LinkedIn protocol, we coded into two groups: good and manufacture. This process provided us with a reliable and condensed group of 17 targeted classes for our study. The distribution for each one of these group labels can be seen in table 2.

We addressed the job industry group classification as a Multi-Label Classification problem (Jain, 2017), which means each instance can be assigned with multiple industry group categories. This process works well for our dataset since the predicting properties of a data-point are not mutually exclusive. Also, we coded our set of target labels using the steps described above using a hot encoding process, which means that we coded each industry group as a column and assign values in the same way that it was done for the skill related variables.

We used a transformation problem method (Jain, 2017) to make our multi-label data interpretable in a more efficient way. We used the Label Powerset method to transform our industry group labels in an adapted multi-class classification problem that the machine could handle more efficiently. This method of transformation creates a binary classifier for every industry group combination in our training dataset. We also tested our results with other approaches such as Binary Relevance, and Classifier Chains, but this one provided the best results. Also, it allows incorporating the correlation among the labels, being interpretable as a multi-label problem despite the transformation process. We use Scikit-multilearn (Scikit-multilearn, n.d; Szymański & Kajdanowicz, 2017), a BSD-licensed library for multi-label classification that is built on top of the well-known scikit-learn ecosystem, to transform our dataset for analysis.

We run and compared a Decision Tree and K-Nearest Neighbors learning models, which are algorithms that are relatively more explainable than other models, and this explainability could be used to provide feedback to potential users on a future interface. More detail on the model selection and tuning processes on the sections of the experiments.

Table 1. Industry code based on LinkedIn (2019)

Group	Industry	Company
good, manufacture	Retail	Dierbergs
organizational, services	Individual and Family Services	Children's Bureau
corporate, media	Marketing and Advertising	A2 Advertising
...	...	...

Table 2. Number of data for each group

Industry groups	Number of data
agr	2
art	13
cons	57
corp	129
edu	13
fin	75
good	110
gov	37
hlth	44
leg	2
man	116
med	124
org	39
rec	100
serv	69
tech	120
tran	23

In terms of evaluation metrics, since multi-label classification differs from a single-label setting we will use distinctive label-wise margin performance measures proposed by the literature (Wu & Zhou, 2017) that will be addressed with more detailed in our experiment sections.

Figure 1 is a flowchart illustrating the steps of our machine learning system. Also, all our data and code can be found on the following repository: `https://github.com/ppseguel/jobskillmatching`

4. Experimental Design

4.1. Experiment 1

We compared the performance of a Decision Tree, and K-Nearest Neighbors classifiers to analyze the data. There were two parts to this experiment. First, we optimized hyper-parameter(s) for each classification model. Second, based on the first result, we run a comparative analysis of optimized classification models using evaluation metrics and comparing the results. Finally, we compared the results for each industry group labels.

Datasets. We used an adapted version of the Woods (2017) dataset on marketing internships job postings based on the protocol described in the method section. We analyzed 17 group labels (dummy variables) using 31 skill dummy variables. We used Label Powerset method to transform the multi-label data before analysis based on Scikit-multilearn library. Also, skill features were normalized using MinMaxScaler function in sklearn, which transforms features by scaling each feature to a given range. We split the data between 75% for the training data (514) and 25% for samples in testing data (172).

Baselines. We followed two steps to conduct this part. First, we optimized hyperparameters for each model. On the Decision Tree, we found the optimal hyperparameters for the split criterion by testing gini and entropy and for the tree depth by checking a range of 1-19 depth values. For K-Nearest Neighbor, we found the optimal hyperparameters by testing a range of 1-3 different values for the Power parameter for the Minkowski metric (p) and the number of nearest neighbors (n neighbors) by testing a range of 1-11 values.

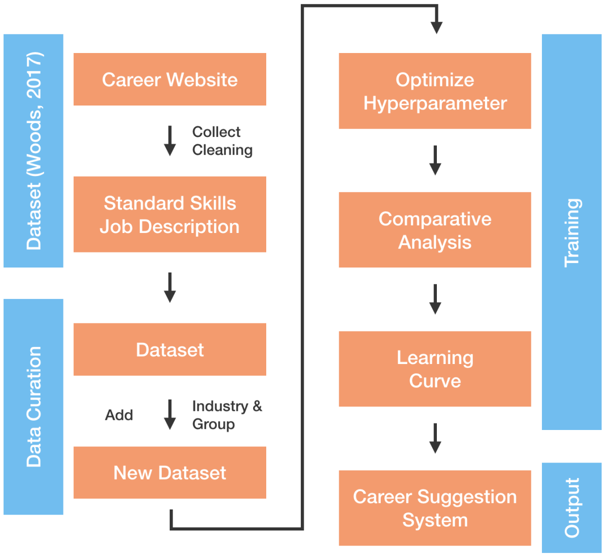


Figure 1. Flowchart

Second, we conducted a comparative analysis by using the best hyperparameters for the models reporting predictive performance metrics.

Evaluation Metrics. We used adapted measurements of accuracy scores and balanced F-scores (F-measure) to compare the multi-label classifiers. The accuracy classification score was calculated from the set of labels. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and the worst score at 0. We reported values for macro-F1 (F-measure averaging on each label) and micro-F1 (F-measure averaging on the prediction matrix). Finally, we report recall and precision scores based on the micro averages. These measures are commonly used to addressed performance inf multi-label and multi-class problems (Wu & Zhou, 2017).

We used the Sklearn library that provides implementations that allow the analysis of the overall score of the classifier, by enabling each sample to give a weighted contribution to the total score, and for each label.

Finally, we plotted the confusion matrix for each targeted label to present the prediction results of the classifiers.

4.2. Experiment 2

On this experiment we will test how the size of training data affects the performance of the model using learning curves.

Datasets. We will use the same dataset that we used in Experiment 1.

Baselines. We will plot the learning curves for the two models with the best parameters in experiment 1, using the same normalization process and the same Label Powerset transformation method for the Decision Tree, and K-Nearest Neighbor classifiers. To create each curve, we will vary the amount of training data (at least ten different training sizes) in relation to two evaluation metrics.

Evaluation Metrics. We will report our results with Label Ranking Loss and Hamming Loss using Sklearn package ranking measures (Sklearn, n.d), which are suitable for multi-class and multi-label classification problems. The label ranking loss function computes the ranking loss, which averages over the samples the number of label pairs that are incorrectly ordered. The hamming loss function calculates the average Hamming loss or Hamming distance between two sets of samples. According to Wu & Zhou (2017) ranking loss can be interpreted as the average fraction of reversely ordered label pairs of each instance, and the hamming loss as the fraction of misclassified labels. In both cases, the lowest achievable ranking is zero.

5. Experimental Results

With our preliminary results, we found that imbalance data on industry group labels affect the predictions. Also, K-Nearest Neighbors provides better performance than the decision tree, but the learning curve is worse. At last, overfitting occurs no matter which analysis we use. More details are shown below.

5.1. Experiment 1

After checking for the best hyper-parameters for our machine learning models, we identified that the Decision Tree performs the best under the entropy metric and on a maximum depth value of 15. On the case of the K-Nearest Neighbors model, we observed that the best parameters were a parameter for the Minkowski metric of 1 and the number of nearest neighbors equal to 1.

According to our results in Table 3, we can see that the K-Nearest Neighbor model performed relatively better than the Decision Tree, either in accuracy, micro-F1, and macro-F1 scores. However, the overall reported performance is not relatively high. It should be noticed that we are using only 31 skills as features for this model, without other supporting data, and that the performance could increase with more instances and more skill features in the future.

When we analyzed closer at the performance of the algorithms across all labels (Table 4), we noticed that the performance varies significantly across them. These results are related to the imbalanced properties of our data, which means that our industry groups are not represented equally

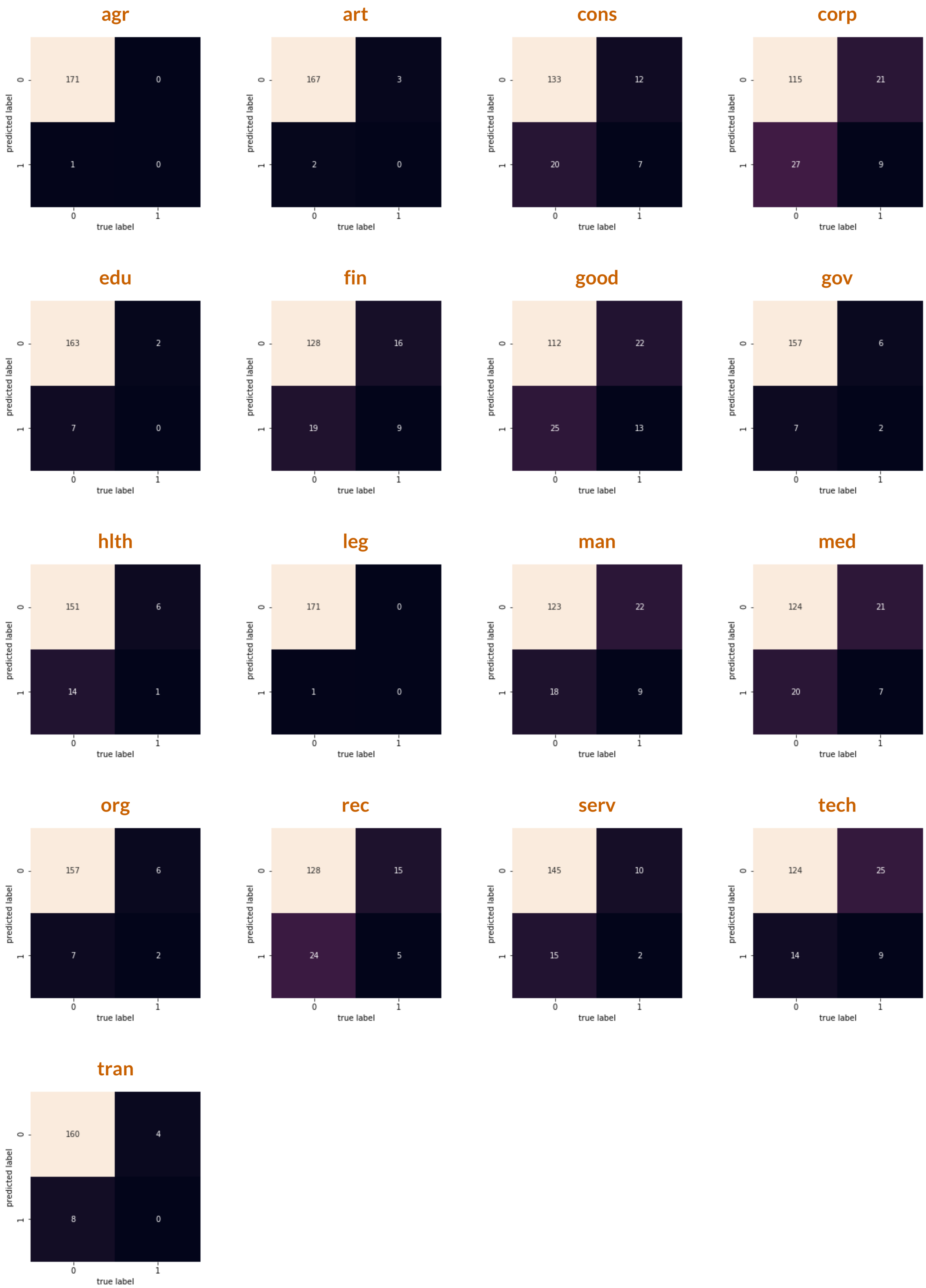


Figure 2. Confusion matrices for the Decision Tree model

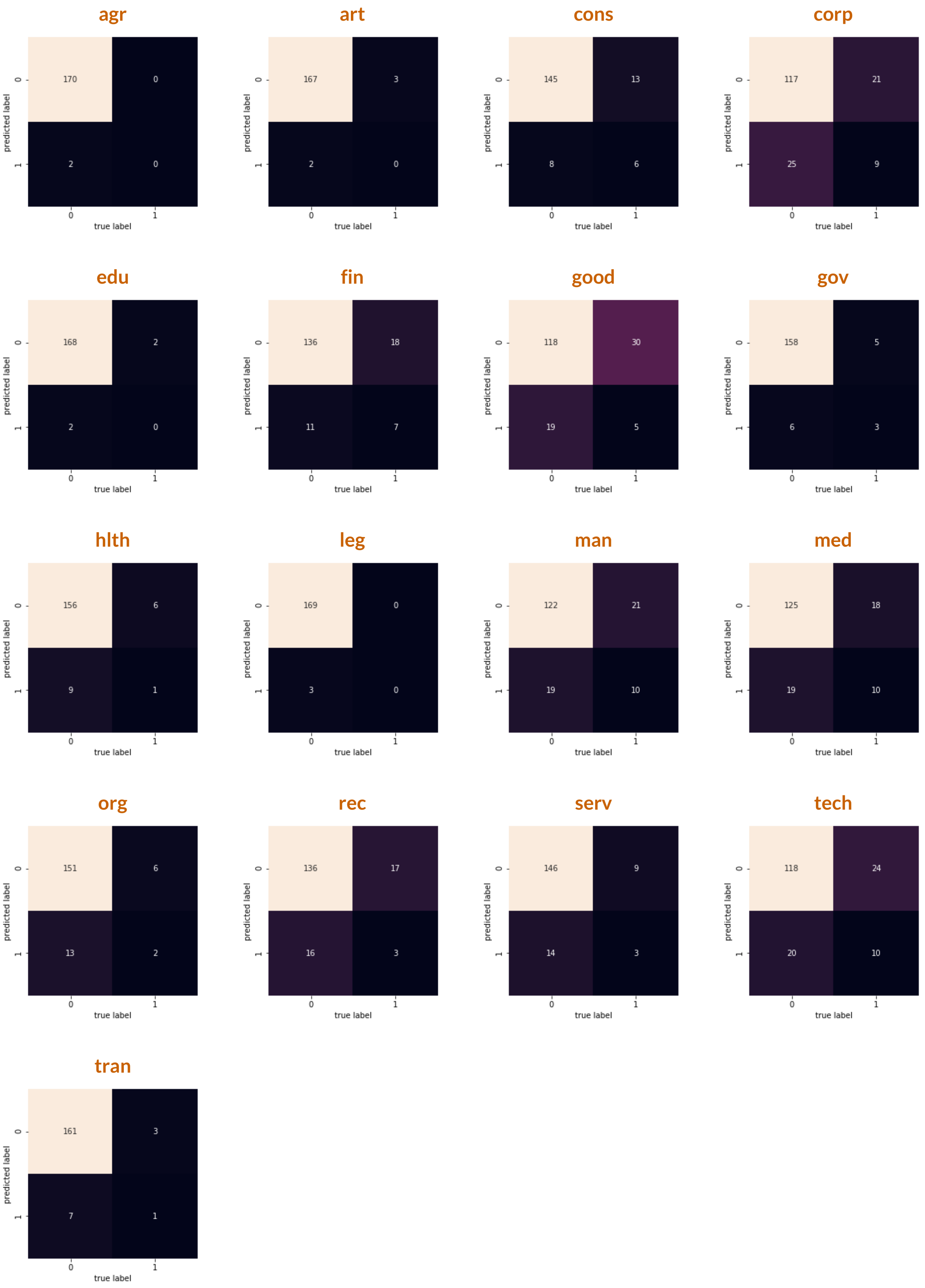


Figure 3. Confusion matrices for the K-Nearest Neighbors model

Table 3. Performance metrics

Performance metric	Decision Tree	K-nearest neighbors
accuracy	0.20	0.23
f1-micro	0.24	0.26
f1-macro	0.17	0.19
recall-micro	0.24	0.26
precision-micro	0.25	0.26

(Table 2).

In terms of the confusion matrices for each label (Fig 2 and Fig 3), we can see, in general, all the matrices show extremely high values of true negative (TN) but very low values of true positive (TP) cases. Although the values of false negative (FN) and false positive (FP) varies, some labels such as corporate (corp), good, medicine (med) have quite high values in FN and FP.

If we look closer, we can see that ”corp” has lower recall on Decision Tree (0.26) than K-Nearest Neighbors (0.42). However, ”org” has higher recall on Decision Tree (0.22) than K-Nearest Neighbors (0.13). ”good” has higher recall and precision on Decision Tree (0.42 and 0.37) than K-Nearest Neighbors (0.21 and 0.14). ”rec” has higher precision on Decision Tree (0.25) than K-Nearest Neighbors (0.15).

5.2. Experiment 2

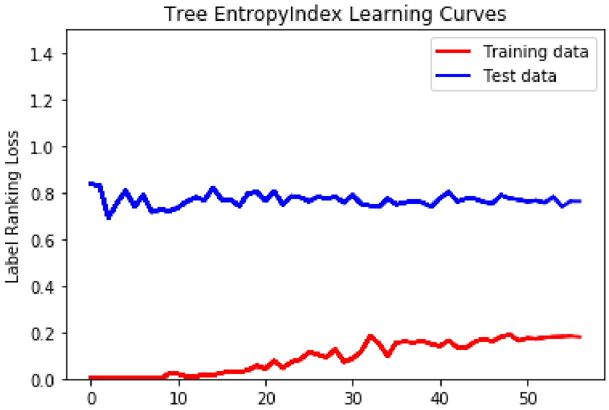


Figure 4. Tree Learning Curves LRL

In our second experiment, we tested if the size of the training data could affect the performance of our models. We plotted the learning curve of decision tree and K-Nearest Neighbors with label ranking loss and hamming loss metrics. According to the four combination results (Figures 4 and 7), in general, four curves are overfitted in training. First of all, low error on four training data which are shown in red.

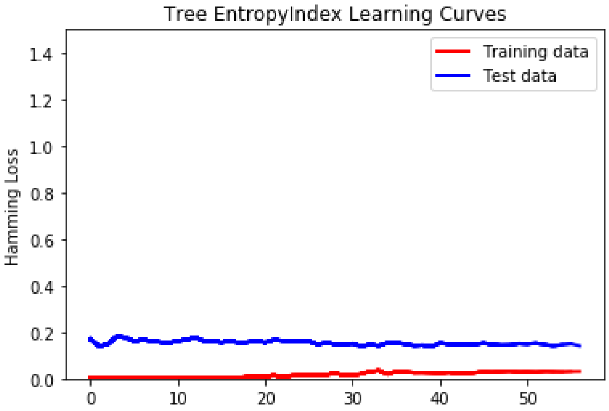


Figure 5. Tree Learning Curves HL

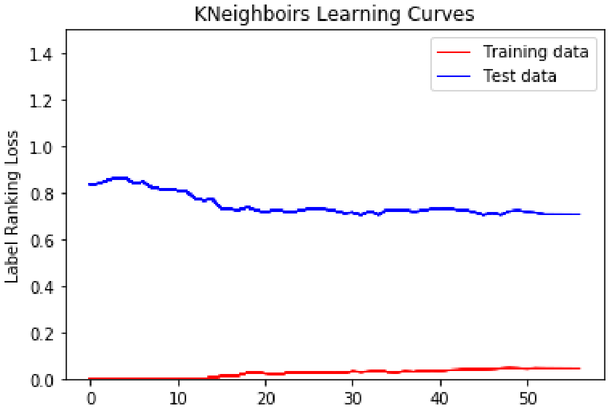


Figure 6. K-nearest neighbors LRL

Second, they all represent low bias and high variance, and the error of test data didn’t decrease much, which shows in blue. That means the model has incorrect assumptions that do not apply well to new examples. To solve the overfitting problem, we expect that the coverage for both training and testing data set should be higher or decrease the numbers of features to lower the variance.

However, if we compare the results of the two algorithms, we will see that the decision tree (Fig 4 and Fig 5) performs

Table 4. F1 scores for each label

Industry groups	Decision Tree	K-nearest neighbors
agr	0	0
art	0	0
cons	0.26	0.36
corp	0.32	0.28
edu	0	0
fin	0.23	0.33
good	0.29	0.17
gov	0.29	0.35
hlth	0.09	0.12
leg	0	0
man	0.31	0.33
med	0.3	0.35
org	0.21	0.17
rec	0.12	0.15
serv	0.07	0.21
tech	0.22	0.31
tran	0.15	0.17

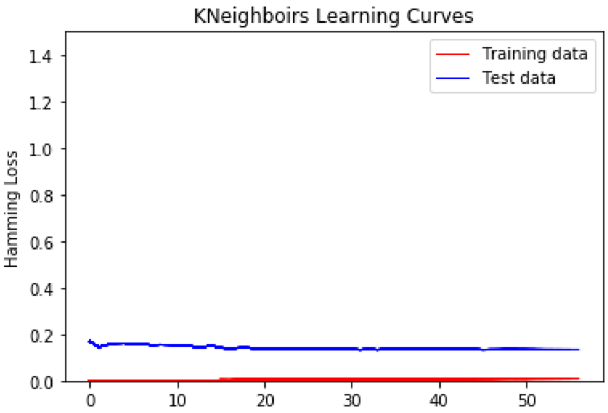


Figure 7. K-nearest neighbors HL

better than the K-Nearest Neighbors (Fig 6 and Fig 7). In K-Nearest Neighbors, the error metrics of training data always low while the error of training data of decision tree increase gradually. That means that increasing the training data might be helpful for us to get better results.

Here we can see in our first experiments, that the K-Nearest Neighbor models performed relatively better than the decision tree. Although our performance is not relatively high, it should be noticed that we are using only 31 skills as features for this model, without other support data. In the future, this data can be increased. When we give a closer look at the performance of our algorithms across all labels, we can notice that the performance varies a lot across them. This result is probably related to the fact that we are dealing with imbalanced data, which means that our industry groups are not represented equally.

6. User Interface

On this section, we provide a general overview of how we envision the user interface that can be built under the predictions of the tested models.

There are a few main features designed on our prototype. First, users can search for the skills in our database, and we will provide them the labels we have. Second, users can add as many hard- or soft-skills as they want (Fig 8). The user can select these labels thinking that those are desirable or actual skills. Third, our machine will calculate the based on our model and dataset. At last, we will show users a suitable industry for them (Fig 9) with a level of accuracy. If they are interested in any of the industry, they can go to see more to get more information.

In short, users can input their current soft and hard skills or add some skills they would like to have in the future. Our system will provide them a possible career path.

A demo can be found in the following link:

https://drive.google.com/a/utexas.edu/file/d/1nPKv-00Z7KEsL0c3VYeNhieUax_clWDM/view?usp=sharing

7. Conclusions

According to our experiments and the training of machine learning models, our models have low levels of prediction of targeted industry labels, and they are all over-fitted. Furthermore, we can see that although the K-Nearest Neighbors performs better in the first experiment, the learning curves were worse than the decision tree on the Multi-label confusion matrix.

In our future work, we will address in five different areas. First, we will explore the differences among labels, and how to fixed problems with the imbalance multi-label data. So far, our dataset includes 17 groups. However, the number of data in each group varies. Some of them has less than 15 data such as agriculture(agr), art, education(edu), and legal(leg) (Table 2). The imbalance multi-label data is a key result of the low accuracy in training. Moreover, we might test more models and methods to deal with multi-label classification problems. For instance, ensemble approaches (ensemble of base multi-label classifiers) could provide better results than the current models.

Second, we will increase the sample of job posting data scraped from Indeed.com, since covering more data for every group may be helpful in our model training. This process will probably also increase the amount of skills features from the current 31. Also, we believe that our approaches can be tested across different domain and job titles beyond marketing intern market, including full-time jobs or contractor positions.

Third, we will compare results with the complete job posting text data, using techniques such as the bag of words of Tf-idf. We believe this analysis will prove useful to compare predictions focused on skill data extracted from the text and predictions made by the whole text data for each instance. This comparison could help us understand if there is an emerging pattern within the text (style, use of words, word counts) across industry jobs.

At last, we believe there is potential to explore label relations using the current methods of Network Science, also applicable to analyze relationships between skills (hard and soft). This type of analysis is supported by Scikit-multilearn library (Scikit-multilearn, n.d; Szymański & Kajdanowicz, 2017).

Our research expands current efforts leveraging ML to provide suggestions on career development to people are given specific skills sets, including soft skill data. We believe that our preliminary results can be used for the design of prototypes of interfaces to assist job-seekers. Although the limitations of our evidence and our data, we believe that this is a significant step to develop AI solutions that can help people who are struggling in deciding or changing their career path.

If you would like to have a quick view of our research, you can access to our video https://www.youtube.com/watch?v=qFGBhNw67J8&feature=youtu.be which shows our vision and idea.

References

Fazel-Zarandi, M. and Fox, M. S. Semantic matchmaking for job recruitment: an ontology-based hybrid approach. In *Proceedings of the 8th International Semantic Web Conference*, volume 525, 2009.

Guo, S., Alamudun, F., and Hammond, T. Résumatcher: A personalized résumé-job matching system. *Expert Systems with Applications*, 60:169–182, 2016.

Hughes, S. How We Data-Mine Related Tech Skills kernel description. https://insights.dice.com/2015/03/16/how-we-data-mine-related-tech-skills/, 2015. Accessed: 2019-04-15.

Jain, S. Solving multi-label classification problems (case studies included). https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/, 2017. Accessed: 2019-04-23.

Jansen, B. J., Jansen, K. J., and Spink, A. Using the web to look for work: Implications for online job seeking and recruiting. *Internet research*, 15(1):49–66, 2005.

Lau, T. and Sure, Y. Introducing ontology-based skills management at a large insurance company. In *Proceedings of the Modellierung*, pp. 123–134. Citeseer, 2002.

LinkedIn. Industry Codes. https://developer.linkedin.com/docs/reference/industry-codes, 2019. Accessed: 2019-04-23.

Poch, M., Bel, N., Espeja, S., and Navio, F. Ranking job offers for candidates: learning hidden knowledge from big data. In *LREC*, pp. 2076–2082, 2014.

Ritzhaupt, A., Martin, F., and Daniels, K. Multimedia competencies for an educational technologist: A survey of professionals and job announcement analysis. *Journal of Educational Multimedia and Hypermedia*, 19(4):421–449, 2010.

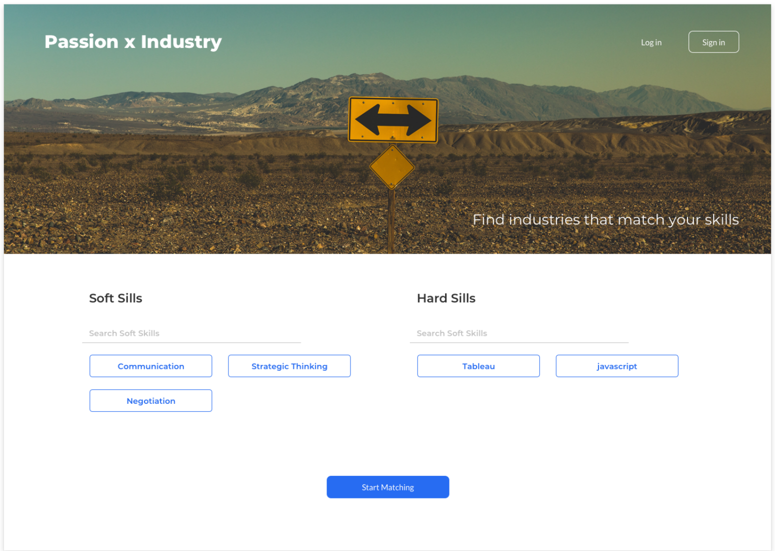


Figure 8. Prototype feature: Add skills

Scikit-multilearn. 1.12. multiclass and multilabel algorithms. `http://scikit.ml/index.html`, n.d. Accessed: 2019-04-23.

Sklearn. 3.3.3. multilabel ranking metrics. `https://scikit-learn.org/stable/modules/model_evaluation.html#scoring-parameter`, n.d. Accessed: 2019-04-23.

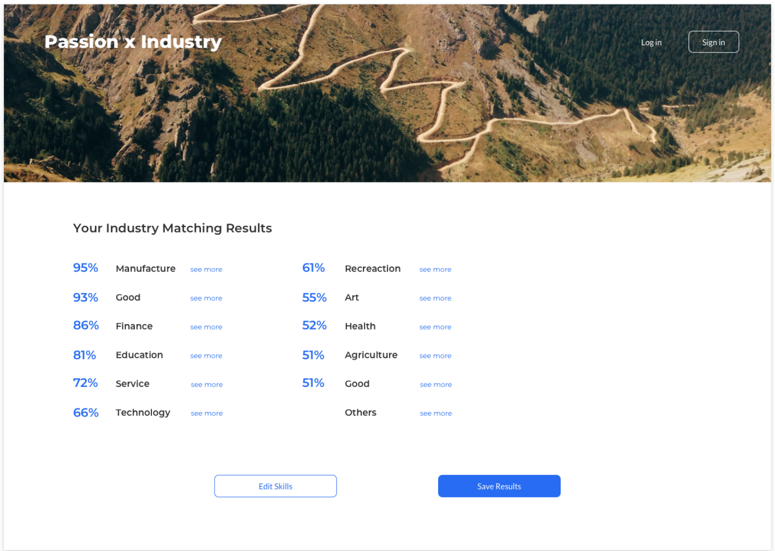
Sugar, W., Hoard, B., Brown, A., and Daniels, L. Identifying multimedia production competencies and skills of instructional design and technology professionals: An analysis of recent job postings. *Journal of Educational Technology Systems*, 40(3):227–249, 2012.

Szymański, P. and Kajdanowicz, T. A scikit-based Python environment for performing multi-label classification. *ArXiv e-prints*, February 2017.

Van-Duyet, L., Quan, V. M., and An, D. Q. Skill2vec: Machine learning approach for determining the relevant skills from job description. *arXiv preprint arXiv:1707.09751*, 2017.

Woods, R. Marketing internship postings. `https://data.world/rdowns26/marketing-internship-postings`, 2017. Accessed: 2019-04-23.

Wu, X.-Z. and Zhou, Z.-H. A unified view of multi-label performance measures. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3780–3788. JMLR. org, 2017.



Your Industry Matching Results

95%	Manufacture	see more	61%	Recreaction	see more
93%	Good	see more	55%	Art	see more
86%	Finance	see more	52%	Health	see more
81%	Education	see more	51%	Agriculture	see more
72%	Service	see more	51%	Good	see more
66%	Technology	see more		Others	see more

Figure 9. Prototype feature: Industry matching results