Multi-Output Career Prediction: Dataset, Method, and Benchmark Suite

Shruti Singh, Abhijeet Gupta, Samah S. Baraheem, Tam V. Nguyen
Department of Computer Science
University of Dayton
Dayton, Ohio
{singhs30, guptaa19, baraheems1, tamnguyen}@udayton.edu

***Abstract*— In this paper, we investigate the career path prediction of an individual in the future. This benefits a variety of application in the industry including enhancing human resources, career guidance, and keeping track of future trends. To this end, we collected a dataset via LinkedIn network, with the job position and the job domain for each individual. There are many attributes related to historical background for each individual. For the career prediction, we investigate six different multi-class multi-output classification methods. Via the benchmark suite, the best classifier achieves an accuracy rate of 91.21% and 95.97% for the job domain and the job position, respectively.**

Keywords— career prediction, dataset, multi-class multi-output classification, benchmark suite.

I. INTRODUCTION

Career choice has always been an important part of our lives. In the early days careers had limited choices and this can be best understood by Darwin's theory of natural selection and survival of the fittest [1]. Industries and jobs rise and fall and sometimes become extinct when the economy changes. Going back to Darwin's theory of evolution the term survival of the fittest describes how different species evolve. Species with better characteristics, like being stronger, faster, and brainier are adapted for survival. Since career choice is variable and is dependent on the current generation's economy, it is important to consider historical data to look for patterns and then make future career predictions. And that's how we can answer the question "Why historical data?". The above theory helps us make a connection between Darwin's theory and how business and careers evolve and sometimes die out. A career path has to compete with the latest trends, technologies, and needs of the consumer. The proposed method in this paper follows a similar concept, where we collect historical data to learn about the current economy's background. Now that we have seen the importance of historical data and background for our method let's explore the benefits of predicting career path. Career centers all around the world are loaded with data, like employment history, education history, candidate achievements, and the concentration of study/work. Consider the case when a student goes to a career counseling center to know the perfect career choice they can pursue. The career cell provides them with a questionnaire asking for their background details. It then analyzes their answers and matches their skills with the most suitable career path they can take. While this works, the approach in this paper automates the above process to make accurate predictions in less time. With our proposed method, a student/individual has to provide their historical data. And with the rising popularity, versatility of social media platforms almost everyone has a social media account, and presence. As with most job applications these days that ask for resume, information as well as links to social accounts, here we can we just need a person's updated LinkedIn profile [2]. This will serve as the basis of providing us with historical data, including educational and work history, years of experience, domain, positions, etc. And any future updates in a person's working/ education status will be updated on LinkedIn. Once we have this information we run it through the application implementing our proposed model to yield results into different labels. The model and the dataset are agile as they can always be modified to fit personal or commercial purposes. This included adding or removing any output predicted labels, or rather adding or removing features in our dataset and training the model to yield different output labels. While this is one of the many examples where career prediction can be used to make faster, more efficient decisions. Even of our data source platform changes in the future, it is easy to shift and include the changes in our dataset as it is not dependent on just one platform. This in turn makes decision-making more accurate, fast and up-to-date with the latest trends. Some other applications may include learning about the position and seeing comparing career growth in an organization or city/area by comparing the position level, years of experience, et al. Although the pair of the data source and model form versatile applications for different purposes, we discuss a more generalized approach. In this paper, we consider the input from the online career networking platform LinkedIn. Therefore, this disregards any cultural, or geographical bias. The LinkedIn data is copied to an excel manually (copy/paste for each data point) with an initial set of 26 features. Some of the examples of the data collected are Human Resources, Technology, Industrial, Economy, and Law to collect candidate information. And the output predicted is in two classes: Domain and Position. The domain class has six class labels, and the position class has eight class labels. Hence, given a candidate's LinkedIn profile as input (independent variable) to the model, it can predict the position name and the work domain in a candidate's career.

The remainder of this paper is organized as follows. Section II surveys the related work. The dataset and the classification

methods are introduced in Section III. Section IV presents the benchmark suite. Finally, Section V concludes this paper.

II. Related Work

Domain Prediction. Career counseling has always been a part of education. It helps them choose the next role in education. Thus many studies have been conducted at this level using various methods. And most of the research conducted has been focused on undergraduate students, or students belonging to STEM field. A methodology called the (Approach Cluster Centers Based) ACCBOX model [3] is proposed to model behavioral information of students belonging to different clusters. The effectiveness is then verified using this method of career choice prediction through experiments on students' behavior datasets. Another work proposed by VidyaShreeram and Muthukumaravel [4] uses four different machine learning models and compares their results with each other. The mode with the highest accuracy is deemed fit. The proposed method uses all four machine learning concepts that are, decision tree, SVM, random forest, and AdaBoost. Decision tree predictive modeling helps in statistics data mining and machine learning, whereas random forest follows a random decision forest and is an ensemble method of classification, regression, and other tasks. It handles missing values, maintains accuracy for a large proportion of data, and is less susceptible to overfitting. Here each student has identifiers that are represented as multidimensional items. Hyperplane divides one class with another, and this is where SVM comes in that finds hyperplanes using vectors and edges. SVM is hence the best method when using large amounts of arguments. And is suitable for less amount of training data, less than 2000. Lastly, AdaBoost uses the concept to combine weaker learning and form one strong rule. Finally, the dataset collected is information from various educational institutions, and the data is preprocessed to narrow it down to 16 features for the model to be trained on. After training the 16 features on the above-mentioned 4 classifiers, the random forest classifier performs the best as it yields the highest prediction accuracy of approximately 93%. The conclusions from this result can be used to send it to the various educational institution where they can utilize these concussions to identify and boost learning in low-performing students and for the recruiting system to select the highest-performing students. Other related works [5-8] include predictions on the following: if the student chooses a full-time job opportunity or goes for higher studies, and predict the position in a STEM field the student might take after their education. Liu and Tan [5] built a machine learning pipeline to automate predictions for students' choice of STEM career. Yamashita et al. [6] proposed NAOMI framework, which uses 1) multi-view embeddings, 2) job duration weight masking, and 3) neural collaborative reasoning for future career pathway prediction.

Job-based Prediction. This approach is different from the above student-focused groups. It focuses on the industrial sector where recruiters or employers seeking a job can use job descriptions to find the most suitable job title for that candidate. Baroliy et al. [9] proposed a learning module system quiz section for computer science engineers. These kinds of career recommender systems help students in picking a job role based on his/her performance and academic records. Huynh et al. [10] utilized four deep neural network models for IT job

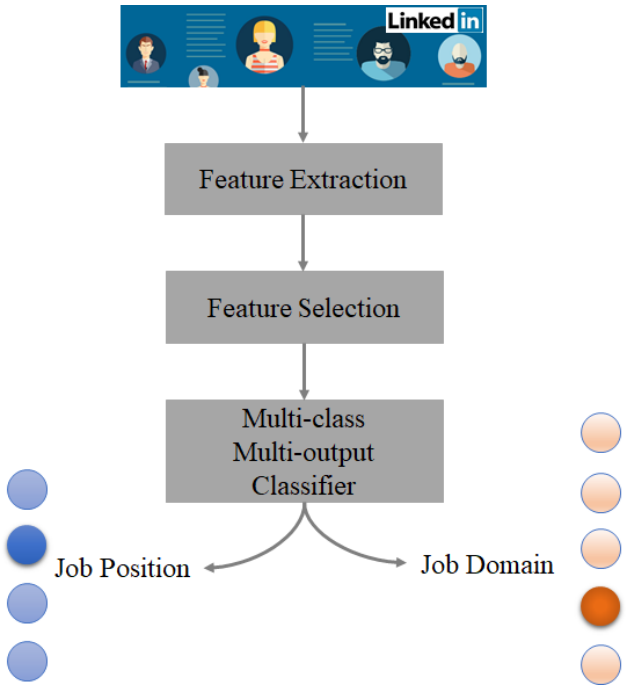

Fig. 1. The flowchart of our multi-output career prediction via historical information.

classification. This includes a single model TextCNN, two combination models (Bi-GRU-CNN and Bi-GRU-LSTMCNN), and a proposed ensemble model. In addition, they implement two pre-trained word embeddings into these models. TextCNN is proposed as it achieves the best results in studies of Natural Language Processing, including emotional recognition, and sentimental analysis. The Bi-GRU-CNN model is used in salary prediction problems to predict wages using data from job requirements, working time, and position. The Bi-GRU-LSTMCNN model is used to solve the Hate-speech detection problem. And finally, the fourth proposed method uses the Majority Voting method to increase the predictive efficiency of the classification model. The final classification of the problem is the combination of outputs of n different models by voting. There are three different job classification models and they predict y for each job description using majority voting. The experiments show that the Bi-GRU-CNN models outperform the Bi-GRULSTM-CNN and the TextCNN model. But the proposed ensemble method achieves the best performance yielding a 725 accuracy, with stable results in all metrics. This immensely helps job seekers and recruiters to find the best suitable job position. However, this work focuses on studying job predictions using deep neural network models considering job description/ requirements only.

III. Methodology

In this section, we first present our collected dataset. Then, we introduce our feature selection and multi-class multi-output classifiers. Figure 1 shows the flowchart of our work.

A. Dataset Collection

The first major step is to decide on the features that we can use to make our dataset. The features should be decided

Table 1. An excerpt of the data rows in our collected career dataset.

Marketing	Arts	Computer Science	Publication (0)	Publications (1-10)	Publications (10+)	Position (New)	Position (Experience)	Position (Expert)	Current Position
0	0	1	1	0	0	0	0	0	Legal Associate at publishing
0	0	0	1	0	0	1	0	0	Software Engineer at manufacturing
0	0	1	1	0	0	1	0	0	Software engineer at software
0	0	1	1	0	0	0	1	0	Software Engineer at a Tech company
0	0	1	1	0	0	0	0	0	Software Engineer at Retail
0	0	1	1	0	0	1	0	0	Chief economist at business

considering a person's background history including, work, and education history. We used the LinkedIn platform to collect a total of 26 features to form our dataset of 420 individuals. Some examples of the features are university, years of experience, papers written, etc. Our dataset was collected in three phases. First, we use LinkedIn to search and export the individual's details into an excel. But the exported report didn't contain the information we needed and just included the names. Therefore, we browsed our LinkedIn connections and visited their profile. Then we copy pasted each field and filled them into the 26 features in our dataset. We use '0' to mark a feature as negative and '1' to mark a feature as positive. As a result, we have an excel with 26 columns and 420 rows consisting of 0's and 1's. Second, we included the names of individuals, to verify the data we collected, and collected 420 entries manually. Finally, we decide on our class labels, and the fields we're going to predict. Since we collect a total of 26 features including years of experience and current position/company we can use it to create not one but two classes. The first, Position will predict the next career position the individual may change to, for example, Recruiter, or Software Engineer. The second, Domain will predict the industry the individual will work in, for example, Technology, Finance, and Marketing, among others. Table 1 shows an excerpt of the features in our collected career dataset.

Unlike other datasets for classification tasks [10, 11], regarding annotation, our dataset provides two outputs for each individual, namely, Position (8 class labels) and Domain (6 class labels). The dataset is randomly split to 70:30 for training and testing, respectively.

B. Feature Selection

Entropy is a measure of randomness in data. It is useful in the sense that it can tell us about the quality of data collected. Hence, to summarize Shannon's Entropy weighs the information based on the probability that an outcome will occur. As a result, systems with one very common event will have very less entropy than systems with equally probable events. The entropy denoted by H is computed as below:

$$H = -\sum_{i} p_i \log(p_i)$$

Similarly, joint entropy selects sets of features that have maximum joint entropy since these will be the least aligned. And these are the ones that will provide the most additional information, improving the quality of our dataset which in turn will be useful while training our dataset on a given model. The joint entropy is computed as:

$$H = -\sum_{i,j} p_{ij} \log(p_i)$$

Although a total of 26 features can be used to train the dataset on a given model. But by including variance, entropy, and joint entropy in our practice we can eliminate any redundant features that are not adding any value to our dataset. We first compute single entropies of the 26 features of 420 rows. Similarly, we compute the joint entropy of every two features and also with the class labels. As a result, we are able to eliminate the low entropy features. From the joint entropy information, we look at two things:

1. Joint Entropy among the features
2. Joint Entropy between feature and class

The joint entropy among features should be high, meaning the two features are not similar. When the joint entropy between a feature and the class is low, this means a feature is similar to the class we want to predict. Utilizing the above-mentioned method, we reduce the initial 26 features to a total of 11 features.

C. Multi-class and Multi-output Classification Methods

This subsection covers functionality related to multi-learning problems, like multiclass and multioutput classification. Since we defined our dataset to have two classes as previously mentioned, Domain and Positions, we need to use a multiclass and multioutput algorithm to train our dataset. Below is an insight into the multiclass and multioutput algorithms.

Multilabel Classification (closely related to multioutput classification) is a classification task labeling each sample with m labels from $n_{classes}$ possible classes, where m can be 0 to $n_{classes}$ inclusive. This can be thought of as predicting properties of a sample that are not mutually exclusive. Formally, a binary output is assigned to each class, for every sample. Positive classes are indicated with 1 and negative classes with 0 or -1. It is thus comparable to running n classes binary classification tasks. This approach treats each label independently whereas multilabel classifiers may treat multiple classes simultaneously, accounting for correlated behavior among them. For example, prediction of the topics relevant to a text document or video. The document or video may be about one of 'religion', 'politics', 'finance' or 'education', several of the topic classes or all of the topic classes.

Multiclass Classification is a classification task with more than two classes. Each sample can only be labeled as one class. For example, classification using features extracted from a set of images of fruit, where each image may either be of an orange, an apple, or a pear. Each image is one sample and is labeled as one of the 3 possible classes. Multiclass classification assumes that each sample is assigned to one and only one label - one sample cannot, for example, be both a pear and an apple.

Multioutput Classification: This strategy consists of fitting one classifier per target. This allows multiple target variable classifications. The purpose of this class is to extend estimators to be able to estimate a series of target functions (f_1,f_2,f_3…, f_n) that are trained on a single X predictor matrix to predict a series of responses (y_1,y_2,y_3…, y_n).

Multiclass-Multioutput Classification: Multiclass-multioutput classification (also known as multitask classification) is a classification task that labels each sample with a set of non-binary properties. Both the number of properties and the number of classes per property are greater than 2. A single estimator thus handles several joint classification tasks. This is both a generalization of the multilabel classification task, which only considers binary attributes, as well as a generalization of the multiclass classification task, where only one property is considered. For example, the classification of the two properties: Positions, and Domain. The property Positions have the following classes: Recruiter, Manager, Software Engineer, Supervisor, Director, Analyst, Project Manager, and Student. And the property Domain has the following classes: Internet Publishing, Retail, Education, Marketing, Business consulting, and Finance. Each individual has 11 features providing background details on past work and education. And a label is output for both properties and each label is one of the possible classes of the corresponding property. Multitask classification is similar to the multioutput classification task with different model formulations.

Table 2. The number and the dimension of targets according to classification problem type.

Classification	Number of targets	Target cardinality
Multiclass	1	>2
Multi-label	>1	2 (0 or 1)
Multiclass and multi-output	>1	>2

Table 2 shows the differences between the different classification problems. The number of targets indicates the number of targets to be predicted. Meanwhile, the target cardinality shows the dimension of the target. In this work, we consider multiclass and multi-output classifiers since they well fit the given problem. We use scikit-learn library [12] for the implementation.

IV. BENCHMARK SUITE

A. Dataset and Classifiers

Dataset: In this paper, we evaluate the proposed method on the newly collected career dataset that has 11 features and 2 classes. The two classes: Positions and Domain have 8 and 6 class labels respectively. We use the accuracy rate as the main metric.

Classifiers: We consider six following multiclass and multi-output classifiers.

- *Decision Tree Classifier*: A decision tree classifier is a non-parametric supervised learning method used for, in this case, classification. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.
- *Extra Tree Classifier*: Extra trees differ from classic decision trees in the way they are built. When looking for the best split to separate the samples of a node into two groups, random splits are drawn for each of the max features randomly selected features, and the best splits among those are chosen. When the number of max features is set to 1, this amounts to building a totally random decision tree.
- *Extra Trees Classifier*: An extra-trees classifier. This class implements a meta-estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
- *K-Neighbors Classifier:* It is a classifier implementing the k-nearest neighbor vote. Neighbors-based classification is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the points. The k- neighbors classification in *K Neighbors Classifier* is the most commonly used technique. In our case, the optimal choice

Table 3. The accuracy rates of six multi-class and multi-output classifiers. The best performance is marked in **boldface** font.

Classification	Random Forest	Decision Tree	Extra Tree	Extra Trees	K-Neighbors	Radius
Domain	**91.21**	90.6	89.65	85.87	90.28	60.65
Position	**95.97**	93.23	90.5	90.7	92.62	45.67

for the value of k can be calculated by using the rule of thumb: $k = \sqrt{N}$ where N is the number of data points. Computing k in our case comes out to be 20. However, while trying values between 1 and 20, the value 1 resulted in the most accurate results. Hence, for our dataset, we choose 1 as the ideal value of k.

- *Radius Neighbors Classifier*: In cases where the data is not uniformly sampled, radius-based neighbors classification in Radius Neighbors Classifier can be a better choice. The user specifies a fixed radius r, such that points in sparser neighborhoods use fewer nearest neighbors for the classification. For high-dimensional parameter spaces, this method becomes less effective due to the so-called "curse of dimensionality".
- *Random Forest Classifier*: A random forest classifier. A random forest is a meta-estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

B. Experimental Results

The first classification model tested was the Random Forest classifier. Since Random Forest uses an ensemble learning method that combines predictions from multiple machine learning algorithms it makes an accurate prediction than a single model. The 300 training samples and 1000 estimators as the input were tested against 120 samples and this classification yielded prediction as a vote by the trees in the forest weighted by their probability estimates. Hence, the predicted class is the one with the highest mean probability estimate across the trees. The predicted class labels were then compared with testing samples to calculate the accuracy classification score. Table 3 shows the accuracy rates of six multi-class and multi-output classifiers. As can be seen in the table, the top performance is 91.21% for Domain and 95.97% for Position.

As a closer look at Extra Tree and Decision Tree classifiers, both are tree classifiers and hence have several advantages like interpretability and data robustness. The Extra Tree achieves 89.65% for Domain and 90.5% for the Position. Meanwhile, the Decision Tree classifier yields 90.6% for Domain and 93.23% for the Position class label. Although the accuracy score is remarkably well and comparable with the Random Forest classifier, the downside of using a tree classifier for future scope and with a larger dataset is the problem of overfitting, resulting in poor prediction performance on unseen data.

Next, we adopt another ensemble classifier called the Extra Trees classifier. This differs from the Extra Trees classifier in the sense that it implements meta-estimators that fit randomized decision trees on various sub-samples of the database and differs from the Random Forest classifier in the sense that it uses averaging to improve the predictive accuracy and also controls overfitting. The resulting accuracy score of the Extra Trees classifier is 85.87% for Domain and 90.7% for the Position class label. Although the Extra Trees is faster than Random Forest, it randomly chooses the split point and does not calculate the optimal one.

The remaining classifiers that we used to test the career dataset are the Radius Neighbors and K- Nearest Neighbors classifier, which implements a vote among neighbors within given radius neighbors. The number of neighbors set to 3 for KNN and 2 for Radius Neighbors as the input parameters, yielded a 60.65% accuracy score for Domain and 45.67% for Position for Radius Neighbors and an accuracy score of 90.28% for Domain and 92.62% for the Position class label. Clearly, the Radius Neighbors classifier fails to compete with the other classifiers, but KNN gives a competitive accuracy as compared to Random Forest. Although, the downside for KNN is that it works well for small dimensions, but not for higher dimensions, and in addition takes a lot of memory to run.

Based on the accuracy score results from all the six multiclass-multioutput classifiers and taking into consideration the future scope for a larger dataset, the Random Forest classifier is a clear winner. Hence, when trained on a dataset of 300 data and tested on 120 training samples, it outperformed the rest with an accuracy score of 91.21% for Domain (8 class labels) and 95.97% (6 class labels) for Position class labels, hence predicting the most accurate results in the class label subset. Furthermore, we can use this model to increase the scope of our class labels, and always include or exclude certain labels with time. As the career dataset collected from LinkedIn is not only helpful in learning about the latest and most popular job domains, but gathering accurate information with a variety of features such as historical data, it is a reliable source. The classification prediction not only predicts the job domain but also the position, and it is malleable in a way that it can be altered to fit the needs of additional class labels to handle complex scenarios.

V. CONCLUSION

In this paper, we explore the prediction of an individual's career path using the individual's historical data. We collect a dataset including information like the university they attended, their current position, and the company from a reliable data source which is only expected to expand in the future. We further investigate six multiclass-multioutput classifiers to test and

verify the classification methods. The benchmark suite shows that Random Forest achieves the best accuracy score of 91.21% for Domain and 95.97% for Position. The experiments conducted conclude a clear winner classifier that can be used to handle even larger datasets for the future scope.

Since we validated the prediction with two class labels Domain and Position, the future scope can include even more class labels to expand the scope of this research. Indeed, this research is flexible to further future modifications and experiments, where it expects to further enhance the aim and the results of the scope.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF) under Grant 2025234.

REFERENCES

[1] Herbert Spencer (1864). Principles of Biology, Volume 1. Williams and Norgate. p. 444.

[2] Linkedin. https://www.linkedin.com. Last access on December 10, 2022.

[3] Min Nie, Zhaohui Xiong, Ruiyang Zhong, Wei Deng, "Career Choice Prediction Based on Campus Big Data—Mining the Potential Behavior of College Students", MDPI, April 2020.

[4] N. VidyaShreeram, A. Muthukumaravel (2021), "Student Career Prediction Using Machine Learning Approaches" Research Scholar, Department of Computer Applications, Bharath Institute of Higher Education and Research, Chennai, Tamilnadu, India.

[5] Ruitao Liu, Aixin Tan, "Towards Interpretable Automated Machine Learning for STEM Career Prediction", Journal of Educational Data Mining, Volume 12, No 2, 2020.

[6] Michiharu Yamashita, Yunqi Li, Thanh Tran, Yongfeng Zhang, Dongwon Lee, "Looking Further into the Future: Career Pathway Prediction", WSDM Computational Jobs Marketplace 2022, February 25, 2022, Tempe, AZ.

[7] Rucha Hemant, Rangnekar, Khyati Pradeep Suratwala, Sanjana Krishna, Sudhir Dhage, "Career Prediction Model Using Data Mining and Linear Classification", 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA).

[8] Lingmei Cai, Xin Wang, "Prediction and Influencing Factors of College Students' Career Planning Based on Big Data Mining", School of Tourism and Exhibition, Hefei University, Hefei Anhui 230601, China, 2022.

[9] Anmol Baroliy, N. Shakya, A. Dwivedi, Shikha Sehrawat, "Career Prediction using Machine Learning Algorithms", International Journal for Research for Applied Science and Engineering Technology, 2020.

[10] Tin Van Huynh, Kiet Van Nguyen, Ngan Luu-Thuy Nguyen, and Anh Gia-Tuan Nguyen, "Job Prediction: From Deep Neural Network Models to Applications", University of Information Technology, Ho Chi Minh City, Vietnam, January 2021.

[11] Thomas Sherk, Minh-Triet Tran, Tam V. Nguyen, "SharkTank Deal Prediction: Dataset and Computational Model", International Conference on Knowledge and Systems Engineering, 2019, pp. 1-6.

[12] Multiclass and multioutput algorithms 1.12. Multiclass and multioutput algorithms — scikit-learn 1.1.3 documentation https://scikit-learn.org/stable/modules/multiclass.html Last access on December 10, 2022.