**Graphical user interface

Description automatically generated**

**Pathrise Fellowship Placement Analysis**

Yadu Krishnan Sarathchandran

4/26/2021

**Introduction**

Pathrise is an online program that provides 1-on-1 mentorship, training, and advice to help job seekers get the best possible jobs in tech. Every two weeks, Pathrise welcomes a new cohort of fellows. If a candidate is interested in joining our program and successfully passes all stages of our admission process, they receive an offer to join Pathrise and become a fellow. The first 2 weeks in the program are called a free trial period and a fellow can withdraw within this free trial period without any penalty. After 2 weeks, a fellow needs to sign an ISA (Income Share Agreement) with us if they want to stay in the program. The entire program lasts up to a year, including 8 weeks of the core curriculum. If a fellow is unable to find a job within a year after joining Pathrise, his/her contract is terminated. However, there might be some exceptions. For instance, if someone was on a break, we may extend their contract for the period of the break.

On average, for fellows who stay with us after their free trial period, it takes about 4 months to receive a final job offer. However, there is a lot of variation in fellows’ outcomes. Being able to predict how fast every single fellow is going to find a job is crucial for our business. In this exercise we provide you with a sample of information we collected from our fellows from the moment they joined our program. Please don’t expect this data set to be perfect, it never happens in the real world.

**The objective of this analysis is to derive insights around if a fellow will ultimately be placed at a company and how long until a placement can be expected.**

For this purpose, we are provided a dataset containing information collected from the Pathrise fellows.

**Exploratory Data Analysis**

We are clear on what the important business questions are that need answering from our analysis. Let’s break them down into two:

1. Do the fellows get placed after joining the program?
2. How fast do they get placed?

Let’s change the business questions to data science/machine learning questions:

1. Predict if a future fellow is more likely to be placed or not placed in a company at the end of the program. This is a binary classification problem (0 or 1). We will elaborate on this soon.
2. Predict the time for a future fellow to be placed in a company after joining the program. This is a regression problem.

Let’s look at the dataset, which is a comma separated values (CSV) format. We load the dataset using Python pandas library’s ‘read\_csv’ input command.

Here’s how it looks for a few columns and rows.

Table

Description automatically generated

We can see that the ‘pathrise\_status’ is given as active, lost, placed etc. ‘primary\_tracks’ are given as SWE(Software), Design, PSO, etc and so on. The two important features here are **1) ‘placed’ – which gives 0 or 1** (meaning placed or not placed) and **2) ‘program\_duration\_days’ – which gives the number of days** from starting the program. These are **the target variables** for our data science questions.

**The Placement Dilemma**

To answer our first question, we only need ‘**placed’ as the target variable**, all others are **predictor variables** and we **drop the ‘program\_duration\_days’** feature.

The features for the dataset are 'id', 'pathrise\_status', 'primary\_track', 'cohort\_tag', 'program\_duration\_days', 'placed', 'employment\_status ','highest\_level\_of\_education', 'length\_of\_job\_search', 'biggest\_challenge\_in\_search', 'professional\_experience', 'work\_authorization\_status', 'number\_of\_interviews', 'number\_of\_applications', 'gender', and 'race'.

Except 'number\_of\_interviews' and 'number\_of\_applications', all of the features are **categorical**. Therefore we need to have some pre-processing before starting our model building. We drop the ‘id’ column right away as it doesn’t provide any information.

Now, we see NaN’s in the dataset, this tells us that there are a lot of **missing values** in our data. Therefore, we need to devise strategies to either fills or remove those candidate rows. Let’s have a look here:

Text

Description automatically generated

We can employ **ordinal encoding (label encoding)** for columns such as 'length\_of\_job\_search' and 'professional\_experience'.

For 'professional\_experience':

'Less than one year': 1, '1-2 years': 2, '3-4 years': 3, '5+ years': 5

For 'length\_of\_job\_search':

'Less than one month': 1, '1-2 months': 2, '3-5 months': 3, '6 months to a year': 4, 'Over a year': 5

Also, **instead of dropping the missing values**, we **replace the missing values with the median values** (because of the **skewness of the data**) for these columns.

Let’s describe the data below and see what to understand from it. We see that **there are outliers** in the candidates. The 95% values have only given 7 interviews, yet the maximum is 20. The 95% values have only applied to 130 jobs, yet the maximum is 1000. The clear outliers can be exclude by limiting the data to the 95% values. We can Table

Description automatically generatedalso see that the mean of the placement is 0.37.

**It might appear like more fellows fail to find a job than find one. This is not correct, at least not yet! This data is for all the fellows in the program, including active fellows and expired fellows. Active fellows are also given ‘placed’ =0. Hence, the wrong inference**.

We will look at this again after data cleaning and wrangling.

Now, we drop features such as 'id', 'program\_duration\_days', 'race', 'gender', 'cohort\_tag'. ‘id’ and ‘cohort\_tag’ are dropped because they aren’t helpful information. Dropping 'program\_duration\_days' was explained earlier for the first question. **race** and **gender** might be good ‘features’ for our question. However, using them to build a predictive model maybe **unethical/harmful** along the way as they are complicated ideas and can induce a **certain bias** into our models.

The shapes of the dataset after this and then dropping the missing values:

Before removing missing values, shape: (2438, 11)

After removing missing values, shape: (1736, 11)

The dataset has shrunk from 2438 instances to 1736. This is **not a small change**, however due to the **overwhelming number of nominal categories** in our features, any other method of filling the missing values might not help and introduce bias. We notice that we have only 11 features, a relatively small number.

Let’s divide the dataset into two categories: 1) With current fellows who aren’t yet placed, and 2) previous fellows who are either placed or not placed. **We can’t do any analysis with the first group as we lack any target variable for them**. Therefore we use them as the input for our predictive model. Thus **our training and validation sets need to be collected from the second dataset.**

df\_test=df[(df['pathrise\_status']=='Active')|(df['pathrise\_status']=='Break')]

df=df[(df['pathrise\_status']!='Active')&(df['pathrise\_status']!='Break')]

Let’s use the useful **pandas crosstab** to look at the fellows’ placement as a function of each feature to derive insights regarding their placements below by univariate analysis.

Table

Description automatically generatedTable

Description automatically generated

Table

Description automatically generatedTable

Description automatically generatedTable

Description automatically generated

Table

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generated

Some observations are in order:

1. highest\_level\_of\_education: **High-school graduates** dominate the placement from first look. **GED or equivalent** underperforms. However, the **sample size is low for both classes**, so we take it with a pinch of salt. **Doctorate candidates perform well**, bachelor's and master's degree candidates are 50/50 to be placed in a univariate analysis.
2. ‘primary\_track’: SWE dominates. Data candidates do better than Design. Not enough sample size for Marketing or Web. **PSO underperforms** compared to all others.
3. ‘length\_of\_job\_search’: As expected, the longer you search for the job, the higher the chances that you’ll be placed.
4. ‘employment\_status’: **Part-timers are dominating**, maybe due to the available time to participate in the program or look for job? Contractors are slightly less likely to be placed, this might be expected because of the nature of the work and time-constraints. The sampling size is low to make any predictions. Students and unemployed do well to get placed ultimately.
5. biggest\_challenge\_in\_search: Candidates with **Resume gap or lack of technical skills** are **underperforming** here. Maybe Pathrise can focus more to help them. The rest are doing good. Candidates whose big challenges are 1) behavioral interviewing. 2) lack of relevant experience, 3) Getting past final round interviews, or 3) Technical interviewing are all **getting placed** in a higher rate.
6. ‘gender’: Male candidates dominates the female candidates with **38% to 33%** in getting placed. Non-binary candidates or candidates who decline to self-identify are not getting placed enough. Even while considering the low sampling, it’s problematic. A **deeper analysis** might be needed, which is **beyond the scope** of this project.
7. ‘race’: Middle Eastern or Arab American and South Asian or Indian American races are more likely to be placed among the race groups. African Americans are under-placed. We require a detailed study to know **how to serve candidates from the groups better so that we can increase the likelihood of their placement**. **Race and gender however won’t be used in our predictive model** as it might induce **bias**. The exclusion of race and gender from the dataset will **decrease the accuracy** of the model, however that is a compromise we should be willing to make.
8. ‘professional\_experience’: Nothing surprising here, candidate with more experience are more likely to be placed.
9. ‘work\_authorization\_status’: Not much difference between groups, however candidates needing an **H1B visa are understandably placed the least**. Maybe **special attention and modifications** in the program to fit their needs? Also, **people without work authorization do not get placed**.

5 of the features showed above are nominal variables. We can’t build a model with nominal variables as features. One solution to this problem is using **One-Hot-Encoding**. It is a representation of categorical variables as binary vectors. We map the categorical values into integer values. Each integer value is then represented as a binary vector, which contains all zeros except the index of the integer, which is 1. See below to see a One-Hot-Encoded dataset.

Table

Description automatically generated

After this, all the new datasets (encoded from the existing nominal feature columns) created are joined to the primary data. Then we remove the parent nominal feature columns from our new dataset.

The resulting dataset has 42 features and 1332 rows of training data. When we look at the number of classes (0 or 1 in ‘placed’) in the data, it seems that **it is pretty much balanced**. We use ‘placed’ as label and remove it from the training data. Now we have 41 columns in training data. However, we can play with dimensionality reduction technique such as **PCA to reduce the dimensions** and improve the accuracy of our models. Another problem to figure out: our testing data (active fellows) doesn’t have labels to know the ground truth. Therefore, we split the training data using the ‘train\_test\_split’ method in Scikit-learn to create a small dataset to test our predictive model’s accuracy. The next step is **scaling the data**, this is needed because different columns have different type of data with different means and variances. Finally, we are ready to proceed with our model on the dataset.

Chart, line chart

Description automatically generatedThe first and the simplest model we test is the **decision-tree-classifier**. **5-fold cross-validation** of the model provided accuracy close to 60%. There are a few possible reasons why the accuracy is lower. Our dataset isn’t perfect, it’s a real-world, messy data. The training data is smaller than usual.

Let’s also look at other classifier performances, for example, how does a random forest classifier and logistic regression classifier perform? See below.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

We choose the **Random Forest Classifier** to make prediction as it performed well in cross-validation. The **accuracy and confusion matrix** for an RF-classifier with **60 trees** are shown below.

Chart

Description automatically generated

1. **The Placement Anticipation**

The next task is a **regression problem**. It is fairly straight-forward from our exploratory data analysis. **How does the ‘program\_duration’ changes for fellows?** This time we **only look at fellows who are already placed**. The **target variable is the ‘program\_duration’** of all ‘placed’ fellows. Let’s revisit some plots from our EDA section to see if we have any decisions to make.

Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

Chart, histogram

Description automatically generated

It doesn’t look easy to make a decision based on the plots, so we proceed directly to the predictive model based on regression techniques. Note that we have used the same One-Hot-Encoding with all the nominal features for this task too. After using PCA to possibly reduce the dimensions and scaling the data, we use linear and logistic regression as well as SGD (Stochastic gradient Descent) Classifier. to train the data and make predictions for the testing data. The scores are 0.11, 0.83, and 0.73 respectively.

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

**Conclusion**

We have successfully transferred two important business questions to simple, solvable Data Science/Machine Learning problems in this analysis. We explored the data and found meaningful relations between features. We then implemented several state-of-the-art data science and machine learning techniques to answer two relevant business questions. Despite the smaller size of the training data, our results and ML models can add value to the ‘Pathrise’ program.

There are avenues where Pathrise can improve more, such as:

1. Provide more Visa related support to H1B holders, which could help them and generate more revenue for Pathrise from the income sharing.
2. Help candidates with Resume Gap and lack of technical skills.
3. Part-time employees get placed more, so does students.
4. Doctoral candidates gets placed more than other classes, hiring them promises revenue and growth.

Our ML classifiers and regression algorithms have done reasonably well given the smaller sizes of the data. These techniques can be improved further as we collect more data in the future.

**References**

1. **https://scikit-learn.org/stable/index.html**
2. https://www.kaggle.com/pmarcelino/comprehensive-data-exploration-with-python