Predictors of Mental Health Illness

Tejveer Singh, Ronit Kumar
CECS 590, Special Topics - Computer Science
13th May, 2018
Instructor : Dr. Roman Tankelevich

Abstract

This is a simple supervised learning model trained on Open Source Mental Illness dataset from a 2014 survey that measures attitudes towards mental health and frequency of mental health disorders in the tech workplace.

The dataset has various fields exploring the conditions tech workers face everyday. We use this dataset to train our model and see how various factors effect a worker's well being. Our model tries to use these factors to predict how likely an individual is to seek medical attention for mental health issues.

1. Introduction

Understanding how circumstances influence mental health in a workplace is one of the goals in every organization. It has been seen that organizations benefit when workers feel happy, motivated and valued. When employees love their jobs, it shows in their work.

In this project, we used an open source dataset[3] to see how conditions at work influence the well being of an employee. What are the most important reasons that contribute to mental health? The survey has data on worker's age, gender, if they had problems with mental health conditions before, etc.

Finally we observe that factors not relating to place or work, like age and gender, also affect mental health. We also look at what can be done to ensure workers are able to perform their work in an optimum manner.

The report is divided in sections that cover exploratory data analysis, data encoding, scaling, fitting and tuning. We test the performance of our model against earlier models and observe how correctly we were able to predict if someone had mental health conditions that could require treatment.

2. Background

The idea of this project and dataset have been taken from kaggle[1]. Kaggle provides us with many interesting ideas and a rough guide on how to proceed with the project. Kaggle has several kernels for this dataset and users have tried various approaches to see what gets them closer to their desired result. We have taken inspiration from many of these kernels and tried to implement one on our own.

We have used only python and its machine learning libraries for this project. We used jetBrains Pycharm Community Edition IDE to write and test our code. Since the survey was small, it did not require a lot of computational power and therefore can be easily emulated on most machines. We make use of supervised learning techniques like KNeighborsClassifier, Decision Tree classifier, Random Forests, Bagging, Boosting and Stacking functions. We also compare the results produced by these different models and the accuracy obtained. Finally we provide some observations on what could be done to improve the accuracy further.

3. Project

We implemented the project on a system with the following specifications -
OS : Linux 4.9.98-88.lts
IDE : PyCharm 2018.1.2 (Community Edition)
CPU: Intel Core i7-6500U @ 4x 3.1GHz
RAM: 7942MiB
GPU: Mesa DRI Intel(R) HD Graphics 520 (Skylake GT2)

3.1. Design

Our model consists of several steps. We start with examining the dataset. Our aim here is to determine the principal components that are effecting our results the most. Once we have an idea about what factors are influencing our predictions the most, we use it to train our models and see if we are successfully able to predict if a worker is at risk of suffering from mental health problems that would require treatment.

3.2. Evaluation

Our work can be broadly divided into :

- Cleaning the dataset.
- Exploratory Data Analysis
- Training the model
- Results

The original dataset had many fields that could not be used, like subjective feedbacks on work environment, etc. These columns were dropped because they could not be effectively classified or labeled for training.

3.3. Cleaning the dataset

We got rid of the variables like ”Timestamp”, “comments”, “state” because they were not really useful in our research. Also in places where the data had missing or NaN values, the average of known values was used, or the data was grouped into the most frequent class and labeled accordingly. Sometimes the data was also replaced with a default value. For example, many people left out the self-employed field. So it was marked NaN. It was changed to not self employed.

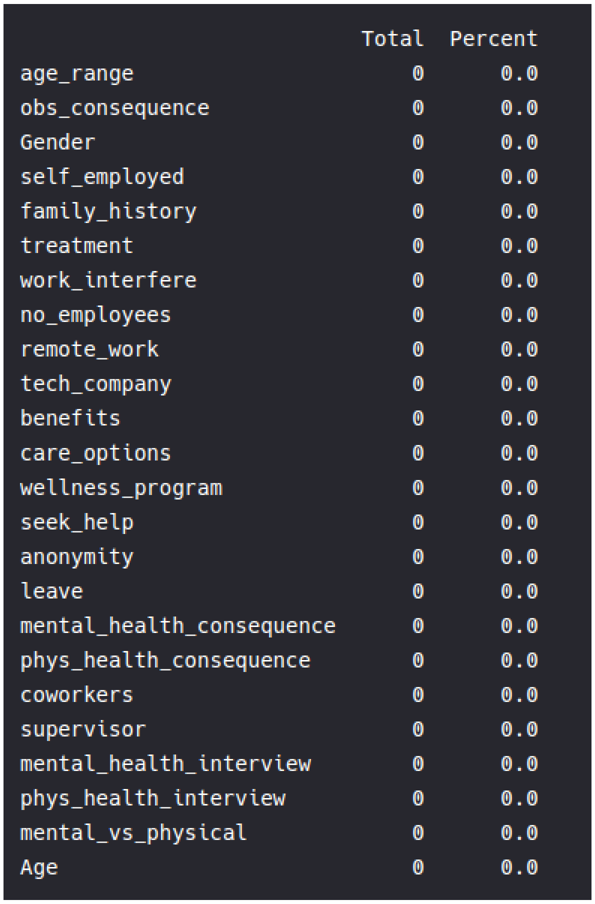
3.4. Exploratory data analysis

We confirmed that our data does not contain any missing or NaN values. Now we can run some operations to see which factors are influencing mental health conditions the most. We generate a covariance matrix to see which data attributes closely relate to each other.

It can be observed that features like wellness-program and seek-help co-relate highly, meaning that the presence of a wellness program contributes positively to the desire to seek help.

Further analysis reveals some interesting trends in our data. According to this survey, women are at a higher risk of having mental health problems but women also find it harder to find or receive help. Also the chances of mental health problems are higher for people who identify themselves as trans.

We have decided to scale our data to age. It gives us a nice overview of the distribution and how different age groups are affected differently.



```
                            Total  Percent
age_range                       0      0.0
obs_consequence                 0      0.0
Gender                          0      0.0
self_employed                   0      0.0
family_history                  0      0.0
treatment                       0      0.0
work_interfere                  0      0.0
no_employees                    0      0.0
remote_work                     0      0.0
tech_company                    0      0.0
benefits                        0      0.0
care_options                    0      0.0
wellness_program                0      0.0
seek_help                       0      0.0
anonymity                       0      0.0
leave                           0      0.0
mental_health_consequence       0      0.0
phys_health_consequence         0      0.0
coworkers                       0      0.0
supervisor                      0      0.0
mental_health_interview         0      0.0
phys_health_interview           0      0.0
mental_vs_physical              0      0.0
Age                             0      0.0
```

Figure 1. There are no missing values.

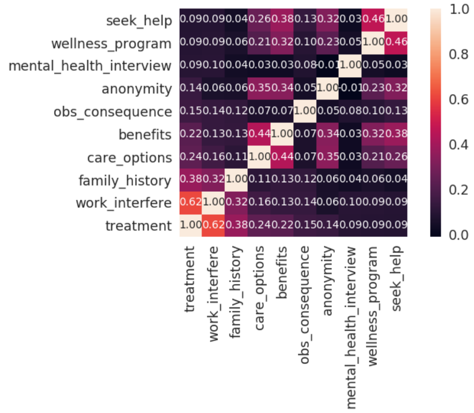


Figure 2. Covariance matrix with probabilities.

3.5. Training the model

We start with tuning our model. We use the following methods for parameter tuning for kNN. We have set our range to 31 and used RandomizedSearchCV Cross

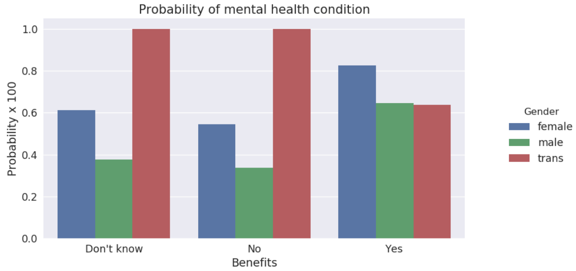


Figure 3. Probability w.r.t work benefits.

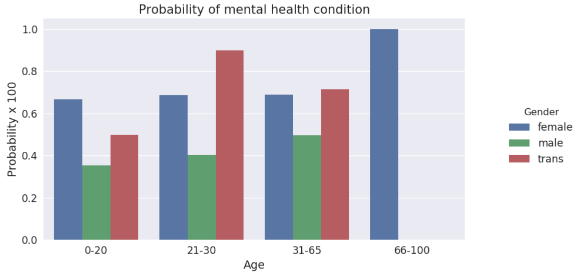


Figure 4. Probability w.r.t age.

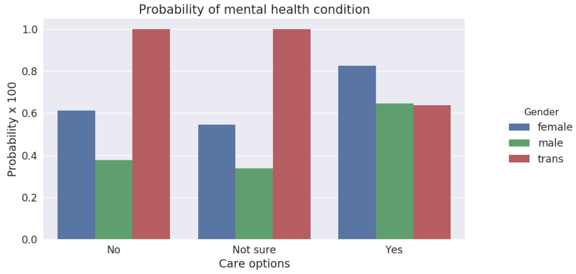


Figure 5. Probability w.r.t when care options are available.

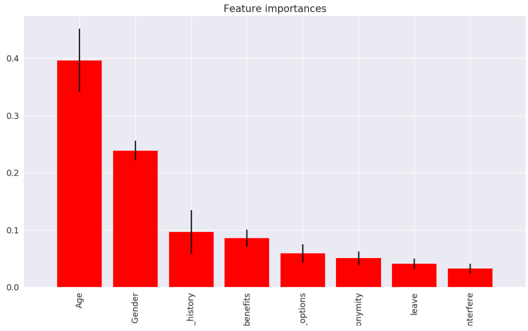


Figure 6. Most important features of our dataset.

Validation for parameter tuning of our n-neighbors parameter. After training the model we found that values around 19 seem to produce the best results.

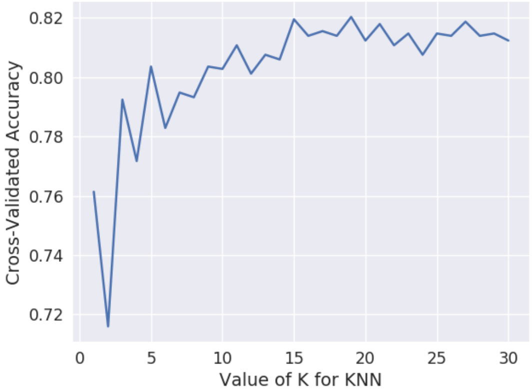


Figure 7. Accuracy for different k-values.

We now perform further data analysis by using various ensemble methods. Our purpose here is to hit and try several techniques and see which provides us the best results. We have reserved 30% of the data for cross validation. The following data attributes appear to influence our predictions the most : *'Age'*, *'Gender'*, *'family_history'*, *'benefits'*, *'care_options'*, *'anonymity'*, *'leave'* and *'work_interfere'*.

We have used the following functions to make predictions. As can be seen, not all methods perform with similar accuracy. The results obtained have been described below.

- **Logistic Regression**
  Accuracy: 0.798941798941799
  Null accuracy:
  0 191
  1 187
  Classification Accuracy: 0.798941798941799
  Classification Error: 0.20105820105820105
  False Positive Rate: 0.2617801047120419
  Precision: 0.7630331753554502
  AUC Score: 0.799591231066439
  Cross-validated AUC: 0.875357422875064

- **KNeighborsClassifier**
  Accuracy: 0.8042328042328042
  Null accuracy:
  0 191
  1 187
  Classification Accuracy: 0.8042328042328042
  Classification Error: 0.1957671957671958

False Positive Rate: 0.2931937172774869
Precision: 0.7511111111111111
AUC Score: 0.8052747991152673
Cross-validated AUC: 0.8788016433051714

- **Decision Tree classifier**
  Accuracy: 0.8068783068783069
  Null accuracy:
  0 191
  1 187
  Classification Accuracy: 0.8068783068783069
  Classification Error: 0.19312169312169314
  False Positive Rate: 0.3193717277486911
  Precision: 0.7415254237288136
  AUC Score: 0.8082285746283282
  Cross-validated AUC: 0.8844362039170506

- **Random Forests**
  Accuracy: 0.8121693121693122
  Null accuracy:
  0 191
  1 187
  Classification Accuracy: 0.8121693121693122
  Classification Error: 0.1878306878306878
  False Positive Rate: 0.3036649214659686
  Precision: 0.75
  AUC Score: 0.8134081809782457
  Cross-validated AUC: 0.8931609623015874

- **Bagging**
  Accuracy: 0.7698412698412699
  Null accuracy:
  0 191
  1 187
  Classification Accuracy: 0.7698412698412699
  Classification Error: 0.23015873015873012
  False Positive Rate: 0.25654450261780104
  Precision: 0.7525252525252525
  AUC Score: 0.7701234706162332
  Cross-validated AUC: 0.8528599830389145

- **Boosting**
  Accuracy: 0.8174603174603174
  Null accuracy:
  0 191
  1 187
  Classification Accuracy: 0.8174603174603174
  Classification Error: 0.18253968253968256
  False Positive Rate: 0.28272251308900526
  Precision: 0.7610619469026548
  AUC Score: 0.8185317915838397
  Cross-validated AUC: 0.8740853414618535

- **Stacking**
  Accuracy: 0.7592592592592593
  Null accuracy:
  0 191
  1 187
  Classification Accuracy: 0.7592592592592593
  Classification Error: 0.2407407407407407
  False Positive Rate: 0.2513089005235602
  Precision: 0.75
  AUC Score: 0.7593722877061343
  Cross-validated AUC: 0.8410388264848951

4. Results

It took our system about thirty minutes to complete execution of all tasks. We based our work on one of the kernels provided on kaggle[1]. We made adjustments to our tuning parameters to use more features and try more cross-validation patterns. Although it improved the overall accuracy of some models, it did not provide any significant improvements in any model. We can see the differences in results below :

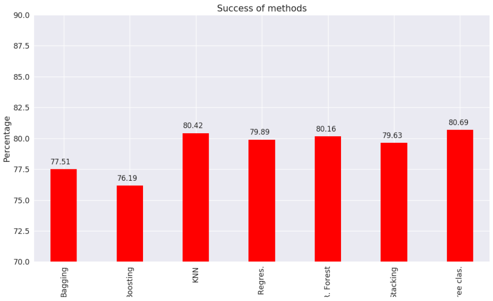


Figure 8. Prediction accuracy of our model.

5. Summary

We used Mental Health in Tech Survey dataset from 2014. We performed some exploratory data analysis. In that we found that there is high co-relation between age, gender and work environment and mental health issues. We used this information to train our models to see if we can successfully predict if a person needs to be treated for a mental health condition. This is a supervised learning model. We used various ensemble and regression techniques to build our evaluation models. These models showed good accuracy. We acquired

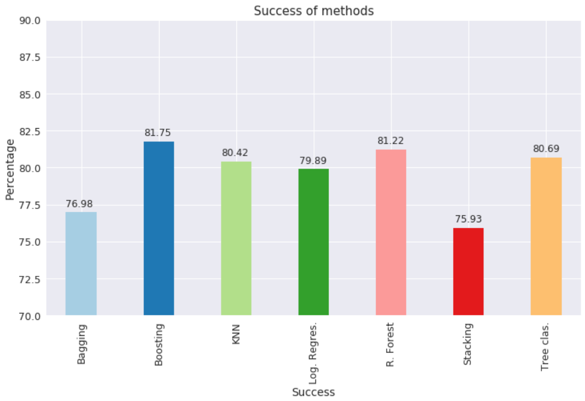


Figure 9. Prediction accuracy of earlier model.

the best prediction accuracy of 80.69% with Tree Classification. The earlier model acquired best prediction accuracy of 81.75% with Boosting.

6. Conclusions

We saw that tuning our parameters plays a very important role in how our model builds its predictions. We assumed that training for more features and allowing the model to train for larger feature combinations would produce better results. After trying many times, we concluded that this is not the case. For our models, the Boosting function gave us the maximum prediction accuracy. Although an accuracy of about 80% is not considered sufficient. Our future scope is to reach an accuracy of around 90%.

A newer version of this dataset is now available. This dataset is more comprehensive and therefore it would be interesting to see how abundance of data affects the quality of predictions.

Therefore, we can also extend our work by using the new dataset and the insights learned from doing this project.

References

[1] https://www.kaggle.com/kairosart/machine-learning-for-mental-health/notebook

[2] https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/

[3] https://www.kaggle.com/osmi/mental-health-in-tech-survey