ml

by PREMI ANGAJALA ANGAJALA

Submission date: 13-May-2021 08:29PM (UTC+0100)
Submission ID: 152610520
File name: ML_Report.docx (33.66K)
Word count: 2356
Character count: 12273

Predicting the treatment rate provided in the mental health of the tech-employees.

Abstract

This dataset focuses on the prediction of the treatment rate to the **IT employees** based on their mental health situation and history and other aspects using machine learning techniques. According to the chosen dataset, employees from different regions of the world are entered in the spreadsheet of 1260 rows * 27 columns providing the least number of entries stating the undermining importance given to the mental health diagnosis has been given. In order determine the importance of mental health, we will use this dataset in understanding each employee's current data and history drilling down to get the information on the predictability on the treatment provided. First, we will pre-process and refine the dataset removing the duplicates and null values. We will then train (80%) and test (20%) the dataset. Training and testing the dataset helps the model to get accuracy of the model. We are using random forest, SVM, K-N-N and XG Boost algorithms to determine the percentage prediction of the variable 'y'= treatment in this dataset. The objective to drill down on the 'treatment' defines the need for considering the necessity to fill in the need for mental health of every individual in our society, especially for the employees who are constantly affected by stress and anxiety from time to time. This **tech survey** dataset is an example to set how credible **mental health** is, which is not considered as much of an illness and is ignored most of the times.

Introduction

This dataset consists of the data of the employees in the technical industry starting from their basic information like age, gender and location to the details of the type of work environment and mental health and physical health data of every employee. We will import all the required libraries and categorize every aspect and it's necessity to drill down on our prediction value 'y' which is 'treatment' column. Our 'x' value consists of 26 column which are the following, 'Timestamp', 'Age', 'Gender', 'Country', 'state', 'self_employment', 'family_history', 'work_interfere', 'no_employees', 'remote_work', 'tech_company', 'benefits', 'care_options', 'wellness_program', 'seek_help', 'anonymity', 'leave', 'mental_health_consequece', 'physical_health_consequence', 'coworkers', 'supervisor', 'mental_health_interview', 'physical_health_interview', 'obs_consequence' and 'comments'. We will first check for the null and duplicate values for each column and remove them and read the data. In order to fill in the null values, we will use the mode value of that column. Then we will remove the columns that will help increase the percentage of the predictability as much as we can in the pre-processing not waiting till the feature selection part. Once, the dataset is refined we will use label-encoder to replace the variable constraints with numerical values which is compatible to run our algorithms. After this, we will train our model and test it for errors. Once this part is cleared, we will proceed with the feature selection and find the correlation between the column and remove one of the correlated columns that will be addressed in the heat map. Finally, we will use random forest algorithm, SVM algorithm, K-N-N algorithm and XG Boost algorithm one by one where we train and test the data in each case and drill down to get an accuracy on the prediction for our 'y' value.

This hypothesis will clear the attention paid on the mental health of the employees in the technical industry. The motivation behind this is to bring light to the idea of having the best of mind in every field. The deliverables of the profession will improve on a regular basis, scope for the creative outlook will be enhanced and every person, irrespective of the profession will carry on with a healthy state of mind. But in the reality, taking a break from work in the name of mental health is not regularly granted and brainstorming for new ideas can get stressful in a work environment. Building a set of databases especially for mental health and constantly conducting workshops to see the progress in the organizations will give a robust outcome. In a long run using artificial intelligence can store the data and implement ideas on cumulative grounds to get better performance results promising the best mental health results.

Machine Learning Model

We will start with importing the libraries into our working model. They are pandas, numpy, seaborn and matplotlib. We started with these libraries to pre-process our dataset. Let us explore the data in the following in table.

Data Exploration

S. No	Variable (Column Name)	Datatype	Description
1	Timestamp	Int64	Timestamp noted when this survey is taken by every employee.
2	Age	Int64	Age of the employee
3	Gender	Object	Gender of the employee
4	Country	Object	Country of the employee
5	state	Object	State of the employee
6	self_employment	Object	If the employee is self-employed or not
7	family_history	Object	If employee's family history consists of any mental health issue or not
8	treatment	Object	If the employee is receiving necessary treatment or not

9	work_interfere	Object	If work is interfering with the mental health or not
10	no_employees	Object	Partially numbers are involved, yet an object.
11	remote_work	Object	Is remote work is allowed or not
12	tech_company	Object	If the employee is working at a tech related company or not
13	Benefits	Object	Are there any mental-health benefits or not
14	care_options	Object	Are there any care options or not
15	wellness_program	Object	Is the company conducting any wellness programs or not
16	seek_help	Object	Is the employee willing to seek profession mental health counselling or not
17	anonymity	Object	Is the anonymity protected or not
18	Leave	Object	Is it easy to obtain leave or not
19	mental_health_consequece	Object	Are there any consequences suffered from mental health issues or not
20	physical_health_consequence	Object	Are there any consequences suffered from physical health issues or not
21	coworkers	Object	Are your coworkers aware of your condition or not
22	supervisor	Object	Is your supervisor aware of your condition or not
23	mental_health_interview	Object	Are there any mental health related interviews

			conducted at your office or not
24	physical_health_interview	Object	Are there any physical health related interviews conducted at your office or not
25	mental_vs_physical	Object	Is mental health affecting the physical health or not
26	obs_consequence	Object	Are there any observations made on your progress or not
27	Comments	Object	Comments of every employee after taking the survey

Pre-processing

Finding the missing values

First, we read the dataset and checked for missing and duplicate values. The columns 'state', 'self_employed', 'work-interfere' and 'comments' have missing values. Now after going through the dataset, we will fill the null values for the columns we plan on keeping in the model. We can use Mean, Median or Mode values, whichever is best suited for our dataset to fill the null values for each column. We are using Mode value as it will help us with the best predictability constraint for our model (I have run the code with mean and median which did not provide the best accuracy). We chose the columns 'self_employed' and 'work-interfere' to remove the null values.

Dropping the columns

Now we will remove the columns which will not help our model moving forward, using the drop function. We will drop 'Timestamp' (as this will not help us in categorizing the employees), 'Country' (as the survey is taken by the employees all over the world and there will be too many categories), 'state' (state will divide countries column and the classification gets complicated. So, we are removing location in total) and 'comments' (comments are not important in the classification. It is better for understanding the type of mental health issues are out there, however, to practically run the model it is not necessary).

Categorizing the unorganized columns

Now, we ensure each column is categorized well before label-encoding. Starting from the age, we have different age groups of employees that participated in this survey. First, we need to remove the rows that has no credibility. The age mentioned by certain employees are humanly impossible and using drop function we will drop them. Now, we will divide the age into sets

like, less than or equal to 30years, greater than 30years and less than 50 years and 50 years and above.

In the column ‘gender’, every employee either spelt it wrong or used either upper case or lower-case letters. Now, we will categorize gender by adding them with values like ‘male’, ‘female’ and ‘other’. This will bring the data together and make label-encoding easier.

Label-encoding

Now that are data is well categorized and organized, we will now go by each column and assign a numerical to each result in that column for us to proceed further. We will read the dataset and check if we have any missing values left. Once we are sure, we will start encoding. The following are the encoded constraints.

S. No	Variable (Column name) and constraints	Label encoding values
1	Age: [<=30, >30= <50, >50]	[1., 0., 2.]
2	Gender: ['Female', 'Male', 'Other']	[0, 1, 2]
3	self_employment: ['No', 'Yes’]	[0, 1]
4	family_history: ['No', 'Yes’]	[0, 1]
5	treatment: ['Yes', 'No']	[1, 0]
6	work_interfere: ['Often', 'Rarely', 'Never', 'Sometimes']	[1, 2, 0, 3]
7	no_employees: ['Jun-25', 'More than 1000', '26-100', '100-500', '01-May', '500-1000']	[4, 5, 2, 1, 0, 3]
8	remote_work ['No', 'Yes']	[0, 1]
9	tech_company: ['Yes', 'No']	[1, 0]
10	Benefits: ['Yes', "Don't know", 'No']	[2, 0, 1]
11	care_options: ['Not sure', 'No', 'Yes']	[1, 0, 2]
12	wellness_program: ['Yes', "Don't know", 'No']	[1, 0, 2]
13	seek_help: ['Yes', "Don't know", 'No']	[2, 0, 1]
14	Anonymity: ['Yes', "Don't know", 'No']	[2, 0, 1]
15	Leave: ['Somewhat easy', "Don't know", 'Somewhat difficult', 'Very difficult', 'Very easy']	[2, 0, 1, 3, 4]
16	mental_health_consequece: ['No', 'Maybe', 'Yes']	[1, 0, 2]
17	physical_health_consequence: ['No', 'Yes', 'Maybe']	[1, 2, 0]
18	Coworkers: ['Some of them', 'No', 'Yes']	[1, 0, 2]
19	Supervisor: ['Yes', 'No', 'Some of them']	[2, 0, 1]
20	mental_health_interview: ['No', 'Yes', 'Maybe']	[1, 2, 0]

21	physical_health_interview: ['Maybe', 'No', 'Yes']	[0, 1, 2]
22	mental_vs_physical: ['Yes', "Don't know", 'No']	[2, 0, 1]
23	obs_consequence: ['No', 'Yes']	[0, 1]

Training and Testing the data

First, we import the train_test_split library with the test value 0.3. Now, splitting the data into two parts, for training 70% and testing 30% we run the model.

Feature Selection

Now we will eliminate one of the two columns ('supervisor' and 'coworkers) that are highly correlated which will affect our target variable 'y'= treatment. We eliminate 'supervisor' column for more accuracy. Moving on, we eliminate 'treatment', 'supervisor', 'phys_health_consequence', 'tech_company', 'no_employees' and 'Gender' columns that has low dependency on our target variable.

Algorithms

Random Forest Algorithm

Random forest is a learning algorithm that is supervised. It creates a "forest" out of an ensemble of decision trees, which are normally trained using the "bagging" process. The bagging method's basic premise is that combining different learning models improves the overall outcome.

We first import libraries for random forest algorithm and import hyper-parameter tuning for randomized search cross validation for more accuracy and run the model. Now, we train and test the data and import libraries for confusion matrix and classification report for the accuracy of our target value giving an accuracy of 75%.

```
[[140  45]
 [ 49 142]]
              precision    recall  f1-score   support

           0       0.74      0.76      0.75       185
           1       0.76      0.74      0.75       191

    accuracy                           0.75       376
   macro avg       0.75      0.75      0.75       376
weighted avg       0.75      0.75      0.75       376

0.75
```

Support Vector Machine (SVM) Algorithm

These data points are fed into a support vector machine, which generates the hyperplane that best separates the tags. This line serves as a decision boundary: anything falling on one side will be classified as blue, and anything falling on the other will be classified as red.

First, we import libraries for SVM algorithm and train and test the dataset, and then importing libraries for confusion matrix and classification report for the accuracy of our target value giving an accuracy of 75%.

```
[[143  42]
 [ 52 139]]
              precision    recall  f1-score   support

           0       0.73      0.77      0.75       185
           1       0.77      0.73      0.75       191

    accuracy                           0.75       376
   macro avg       0.75      0.75      0.75       376
weighted avg       0.75      0.75      0.75       376

0.75
```

k-nearest-neighbors (KNN) Algorithm

The KNN algorithm is a simple, supervised machine learning algorithm that can be used to solve classification and regression problems. It is simple to set up and use, but it has the disadvantage of being noticeably slower as the amount of data in use increases.

First, we import libraries for KNN algorithm and use 'for' loop in Python language to run the model. Now, we train and test the data and import libraries for confusion matrix and classification report for the accuracy of our target value giving an accuracy of 72%.

```
[[134  51]
 [ 52 139]]
              precision    recall  f1-score   support

           0       0.72      0.72      0.72       185
           1       0.73      0.73      0.73       191

    accuracy                           0.73       376
   macro avg       0.73      0.73      0.73       376
weighted avg       0.73      0.73      0.73       376

0.726063829787234
```

XG Boost Algorithm

XG Boost is a machine learning algorithm that has recently dominated Kaggle competitions for structured or tabular results. XG Boost is a high-speed and high-performance implementation of gradient boosted decision trees.

First, we import libraries for XG Boost algorithm and train and test the dataset, and then importing libraries for confusion matrix and classification report for the accuracy of our target value giving an accuracy of 71%.

```
[[132  53]
 [ 57 134]]
              precision    recall  f1-score   support

           0       0.70      0.71      0.71       185
           1       0.72      0.70      0.71       191

    accuracy                           0.71       376
   macro avg       0.71      0.71      0.71       376
weighted avg       0.71      0.71      0.71       376
```

0.7074468085106383

Conclusion

In conclusion to my report, I believe there are many actions that could be taken to prevent me ntal health issues. My research made me understand that there are not many databases that ad dress this issue and with proper collection of information of the people globally which just ha s to be additional column for people to freely address any sort of mental trouble, this could be controlled. The algorithms which brought the accuracy for treatment, 70-75% is not on a huge database even though the survey is taken by people from different parts of the world.

ml

ORIGINALITY REPORT

11%	5%	6%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	github.com Internet Source	2%
2	Submitted to University of Essex Student Paper	2%
3	"Pertinent Exploration of Privacy Preserving Perturbation Methods", International Journal of Recent Technology and Engineering, 2020 Publication	2%
4	Submitted to University of Greenwich Student Paper	2%
5	Submitted to University of Edinburgh Student Paper	2%
6	Submitted to University of Warwick Student Paper	1%
7	Submitted to Sim University Student Paper	1%
8	Submitted to RMIT University Student Paper	<1%

9	"Proceedings of International Conference on Trends in Computational and Cognitive Engineering", Springer Science and Business Media LLC, 2021 Publication	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		