Prediction of Mental Health Disorder

Keval Patel, Data Scientist
Kp14@myscc.ca

Table of Contents

Introduction

Mental Health is one of the most import part of tech industries. It is very important and critical problem how to measure mental health of employers so, stakeholders can comprehend its effect on their business and try to improve that by giving employees what they needed furthermore they try to feel their employees that nobody is separated from everyone else. If they get results of Mental Health Disorder so, these results are additionally useful assets for raising open mindfulness, disgrace busting and pushing for better medical services [1].

Mental Health Disorders is also known as Mental Health illness and it denotes health condition of mental state that influence your state of mind which directly affect your mood, rational process and performances. In case of Mental illness persons find themselves into schizophrenia, incorporate despondency, uneasiness, dietary problems, addictive practices and so on. Many people occasionally have Mental health concerns, but it become Disorder or illness when determined signs and indications becomes very frequent which affect their ability to work and it can affect your commonplace lifecycle. In most cases, it can be only managed by mixture of medical drugs recommend by doctor or psychotherapy where they can talk about their problems which are bothering them. It's hard to find Symptoms of Mental illness because it is depending on disorder patient have and occasionally signs and indication of disorder could be appearing as physical problems such as pain in different part of body, headaches and so on [2].

We can see number of incidents based on Mental illness as per Research conducted by National Alliance on Mental Health Illness (NAMI), Only in US 1 in 5 adult experience mental health disorder every year, In young age 1 in 6 youngsters feel mental illness year and people between age 10 - 34 commit suicide which lead to death and all they are suffering of Mental Health disorder specially in US. It also effects their bodies, according to this research people who are in depression they 40% higher risks to develop heart problems and decrease metabolism of their bodies [3].

Mental Health in Tech Survey was published by Open Sourcing Mental Illness (OSMI) which is non-profit organization to raise awareness, educate people and provide resources to tech communities to promote mental health. They started this mission in 2013, and so far, they conduct survey every year and publish the result of the survey [4]. they have lot of options to analysis data on their site you can see direct dashboards on their website which shows results.

Related Work

As vital part of overall wellbeing, Mental Health disorder is barrier in tech companies. In University of Waterloo, they made tremendous progress in area of clustering by using density-based Clustering which describes effect of Mental Health in workplace. Firstly, they use Principal Component Analysis (PCA) for dimension reduction and then they use DBSCAN clustering because it helps to identifies outlier in data. For selecting radius, they use Radius to Nearest

Neighbors (RNN). Furthermore, they also use Support Vector Machine (SVM) and Naïve Bayes to solve Classification problems after all that they get 66% accuracy on test Sets [5].

Natalie Poon from General Assembly of Data science publish her work on perzi.com on 29th November 2017, her goal was to find whether or not person will pursue treatment for a mental health condition. In most of her work she did Exploratory Data Analysis (EDA) and at the end for prediction, she used only decision tree for that, and she got 69.5% accuracy on test set [6].

The Fluffy Mammal published his work in Towards Science on 18th March 2017, He used OSMI – Survey- 2016 another part of our dataset, started with Exploratory Data Analysis and find relation between target and features. Besides he started with Model and variable selection with logistic regression and decision tree and he use brierScore to check performance of their models at the end, He got 57.4% of diagnoses correctly [7].

When I compare these works with my work then I can say that, I tried some different approaches to solve the same problem and we got better results.

Methods

In order to achieve our desired goal, I tried many methods, which can be mainly summarized as the following steps:

Data Cleaning and Preprocessing: Firstly, I important packages which are necessary for data cleaning and preprocessing which are NumPy, pandas, preprocessing form sklearn, train_test_split, matplotlib.pyplot, seaboarn, and Counter. Then I import our data set which I downloaded from Kaggle [8]. After that, I start cleaning our dataset. When I take look at data, I have seen some unexpected values in age and Gender columns so, for that first, I find unique values in the age column and then I remove that unnecessary values which are around 8 values. Then I did the same thing for Gender column but this time I replace values in 3 classes instead of removing them because there are a lot of values such as 'Cis Male', 'CisFemale','Malr' which could be typing mistakes and I don't want to lose that much data so, I replace all values in Male, Female, and others. Furthermore, I find how much null values I have and Then I drop 2 columns which has null values "comments" and "state" and also remove "Timestamp" which has almost 3 months of data that doesn't define anything. When I finish with that, I have null values in "work_interfere" and "self_employed" and I don't want to remove that so, I replace null values with the string "NaN", I create one function where if integers columns are there the function will not put any values but if columns have string values then function put "NaN" instead of null values [9]. After that, I create Label Dictionary by Label encoder from sklearn and it creates unique value for each string in column and store in Dictionary so, whenever I need to define that values I can define them and for further model building it's important to use label encoder

because machine learning model cannot identify string values by itself I have to define them. After all data cleaning and preprocessing I save that data so, in future if I need that data, I don't have to do the same process again. In the end, part of preprocessing I define X as all the features except treatment which is our target variable and y as treatment then split data by train_test_split which by default creates 75% training data and 25% test data.

Analysis of Data: As part of Analysis, I did some analysis which are meaningful and help to find appropriate results.

- In First plot (Fig.1), I check, where I get the data and in the plot, I select top 10 countries where most of the data were collected from the United States which is around 60% and after that United Kingdom which is around 16% after 7 most countries it remains around the same. So, I conclude that most of the data collected from the United States [10].

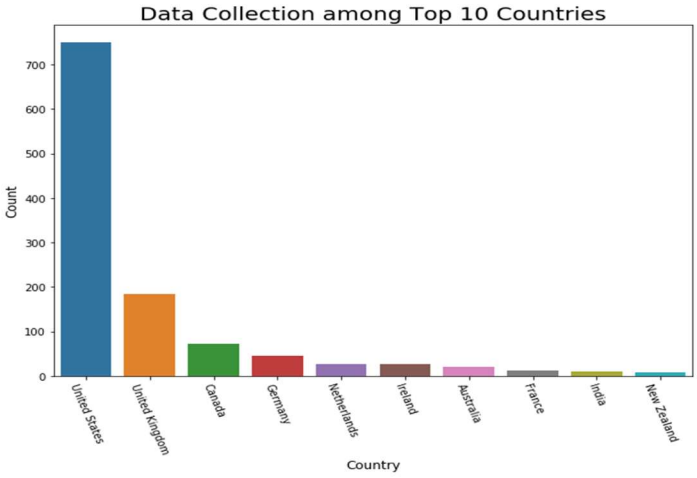


Fig.1: Data Collection for Top 10 Countries

- In the second plot (Fig.2), I create a Correlation matrix of data where I can see the relation between two features In, this matrix darker blue color means there is a strong relationship between those two features I mention scale over there so, white color means there no relation between those features. As I can see there is a very good relationship between our target variable treatment and work_interfere and this the reason I don't want to remove null values from that column because sometimes null values can help to learn model [9].

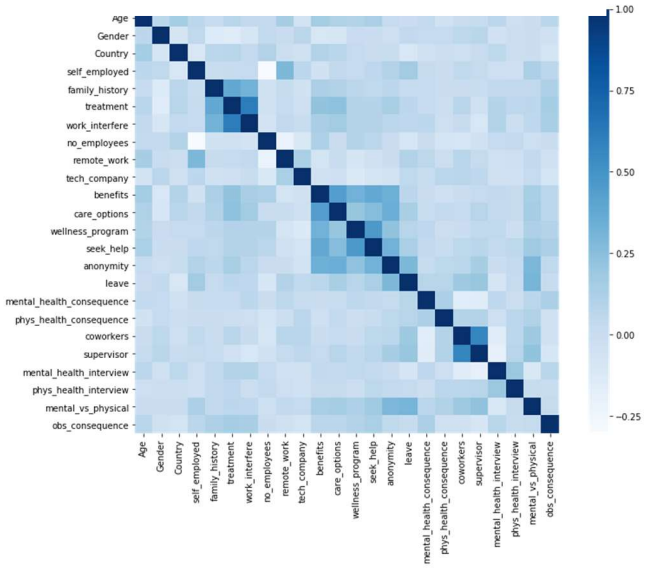
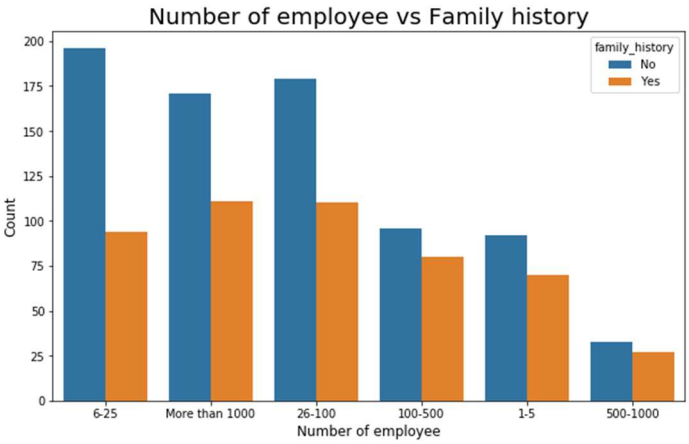


Fig.2: Correlation matrix of data

- In third plot (Fig.3), I check that How many Employees working on the company and Family history of those Employees to check if anyone from their family has or had mental health disorder and this is very important because if the family of that employee has mental illness then it increase the chances of that employee have mental health disorder.



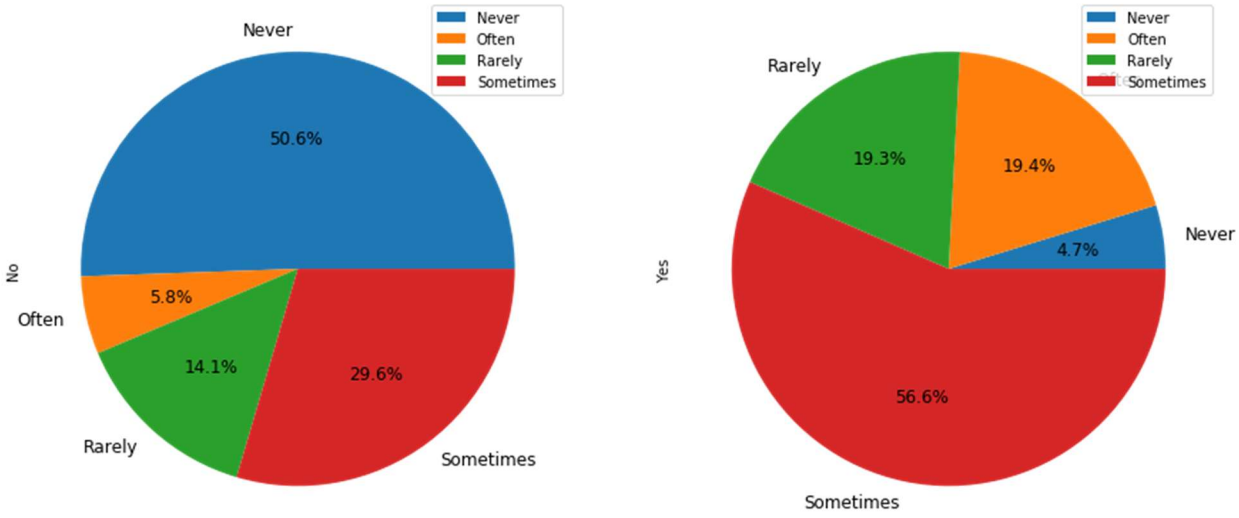
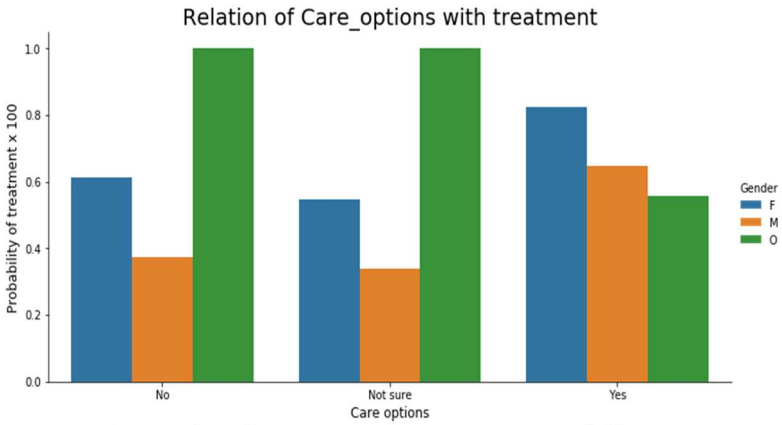


Fig.4: Effect of Mental health on work by treatment

- In Fig.4, By this pie-plots, I can see that how mental health effect on employees work, and I create pie-plots with treatment column so, I define the statement what are the consequences when an employee had treatment at the very right side, first pie-plot define employees who didn't have treatment and 50.6% employees think that mental health is not going to effect on their work but there are some employees who think that mental health often effects on their work. On the other hand, People who had treatment around 56.6% believe that mental health is still affecting their work and there are only 4.7% employees who think that after treatment mental health never affects their work. So, by all this, I deduce that people who had treatment they still believe that mental health is going to affect their work at the same time people who didn't have any treatment they trust that mental health is not going to affect their work.

- In Fig. 5, I create a bar plot where I can define that whether employees know about their medical benefits or not, and if they know then what is the probability of, they use medical benefits in their treatment. As I can see there are people who didn't specify their gender, or they don't want to specify their gender and they don't know about benefits and they use medical benefits in their treatment there is the possibility that they don't understand the question or they got information from someone by which they use their medical benefits and at the same time people who know about their medical benefits they didn't use so, I can also conclude that they didn't have any medical situation so, they can use the medical benefits [9].



Classifier Models [11]**:** In this Part, I build Classifier models from sklearn.

- For K-nearest neighbors I set number of nearest neighbor 5 and all default parameters already set by sklearn.
- For DecisionTreeClassifier I set min_sample_split = 6 and for max_depth I run for loop to try different number as max_depth and I found with max_depth=9 I got better results all default parameters already set by sklearn.
- For RandomForestClassifier I use ParameterGrid search to find best parameters for random forest and I found {'n_estimators': 200, 'min_samples_split': 12, 'max_depth': 7} as best parameters all default parameters already set by sklearn.
- For Support Vector Classifier I set kernel as 'linear' and all default paramters alredy set by sklearn.
- For Gaussian Naive Bayes I use all the parameters set by sklearn.

Note: Please check scikit-learn documentation for further information about classifier models.

Word Cloud [13]**:** For creating word Cloud, primary I need to install wordcloud library so, for that, I install wordcloud on anaconda prompt with help of pip. Then I import wordcloud in jupyter notebook for words I use comments parts of the original data set which I imported first without cleaning. After that I create DataFrame where I remove rows that have null values in the comments column besides, I select only two columns "treatment" and "comments". Then I create a series of comments where treatments are Yes and join all the words with space and create a list and give name comment_Yes. In the next step, I create a word cloud with fig size = (15,10).

Build Models using NLP [12]**:** For Neutral Language Processing first I must create that kind of data from comments. Firstly, I took the same comments DataFrame which I create for world Cloud and replace Yes with 1 and No with 0 in treatment Column. I have “Maybe” in the treatment column, so I remove that because it creates one more class in prediction which directly reduces the probability of random prediction. In the end, I create new DataFrame with treatment and comments where treatments are only “Yes” and “No”.

To perform NLP on this data I need to import nltk library so, for that, I need to install nltk on anaconda prompt with help of pip. Initially, I import re tool for text cleaning and then import nltk inside nltk I choose stopwords and proterStemmer for removing extra characters, and stemming the word means to choose the root of the word. Then I create a list named corpus and do all the transformation of words and append into the list. Then I use CountVectorizer from sklearn and select 1500 maximum words which count words in a list, and it can select a maximum of 1500 words. After that I use that tokenize data with our target variable treatment and put into train_test_split which split the data into training and testing.

I use same classifiers which I used in Classifiers Models but In decision tree I got better result at max_depth=8 and for Random Forest I got {'n_estimators': 100, 'min_samples_split': 2, 'max_depth': 2} as best parameters in ParameterGrid search.

Results

In Result, I mention all the results which I got from our classifier model for the original data with cleaning and modified data which contain comments and treatment columns with some specific modification to develop data on which I can use Neural Language Processing and I also mention another results for Classifiers on which I perform Neural Language Processing.

Results for Classifier Models:

Models	Accuracy	
	Train	Test
K-Neighbor Classifier	85%	74%
Decision Tree Classifier	91%	77%
Random Forest Classifier	88%	83%
Support Vector Classifier	81%	79%
Gaussian Naive Bayes Classifier	79%	77%

Results for Models in which I use NLP:

Models	Accuracy	
	Train	Test
K-Neighbor Classifier	76%	63%
Decision Tree Classifier	78%	70%
Random Forest Classifier	61%	73%
Support Vector Classifier	98%	68%
Gaussian Naive Bayes Classifier	97%	51%

Discussion

When I decided to work on mental health disorders, I need a dataset that can help me to achieve my goals. After a little bit struggling, I find some dataset on Kaggle created by OSMI and I select 2014 because I think that this is the initial stage of this mission, they have some unstructured data and unclean data so, I retrieve more information and learn more about data cleaning [8]. When I started working on the dataset and I find very hard to clean data, but I find one kernel on Kaggle, and Megan Risdal has done some work which helps me to clean data [9]. After Cleaning data, I started working on the analysis of data, I get some ideas from the same kernel and some of them I create by my understanding which results in the statement I want to define. Then I started working on the Classifier model which I learn from another class of Machine learning-2 and I use this book [11]. After a lot of trials and errors, I got satisfied with the accuracy and better performance over models. I was seeing my old projects and I saw word cloud and I got an idea to use the word cloud on comments to see how a person reacts when he/she has mental health disorder and if you are more interested about word cloud then you could check out this [13]. At vary last I worked with NLP in our marketing Analytics classes so, I think that I can also use this kind of data which could be legendary idea because If I could build a perfect model which uses NLP to check whether patients have a mental illness or not then by some sentences said by a person is enough to predict Mental Health of that person. It has some consequences but still, it could help in the industry as well as hospitals, and here is the similar work which I did in marketing analytics [12].

Conclusion

In Conclusion, after analyzing data, we can accomplish that people who had treatment at some point of life they still think that Mental Health can affect their personal life. Besides when we do training and testing on clean data, I got 83% accuracy in Classifier without NLP also, by seeing a model performance with help of confusion matrix I can say that models better can predict more accurate results. In the end, when I use NLP on data and run classification model then we got 73% accuracy in Random Forest but It's not performing well on the training set, and performance is worst. In NLP, the Decision tree has 70% accuracy and by seeing the confusion matrix I can say that performance of the model is better than other models. It could be more helpful in the future if we have more comments given by a Person who has Mental Health Disorder and comments given by a person who has a normal Mental Health condition so, we can classify that data and get expected results.

References

[1] Mental Health By the Numbers: https://www.nami.org/learn-more/mental-health-by-the-numbers

[2] Mayo Clinic Mental illness: https://www.mayoclinic.org/diseases-conditions/mental-illness/symptoms-causes/syc-20374968

[3] About OSMI: https://osmihelp.org/about/about-osmi

[4] Density-based Clustering of Workplace Effects on Mental Health: https://www.researchgate.net/profile/Shruti_Appiah/publication/320310808_Density-based_Clustering_of_Workplace_Effects_on_Mental_Health/links/59dd28be458515f6efef24a7/Density-based-Clustering-of-Workplace-Effects-on-Mental-Health.pdf

[5] An Analysis on a Mental Health in Tech Survey: https://prezi.com/1udt0hbffcre/an-analysis-on-a-mental-health-in-tech-survey/

[6] Data and Mental Health: The OSMI Survey 2016: https://towardsdatascience.com/data-and-mental-health-the-osmi-survey-2016-39a3d308ac2f

[8] Mental Health in Tech Survey – Dataset: https://www.kaggle.com/osmi/mental-health-in-tech-survey

[9] Predictors of mental health illness: https://www.kaggle.com/kairosart/machine-learning-for-mental-health-1

[10] Mental Health in Tech Survey Data Visualization: https://www.kaggle.com/xingobar/mental-health-in-tech-survey-data-visualization

[11] Introduction to Machine Learning with Python: A Guide for Data Scientists: https://books.google.ca/books?id=1-4lDQAAQBAJ&printsec=frontcover&redir_esc=y#v=onepage&q&f=false

[12] Sentiment Analysis of Restaurant Reviews: https://www.kaggle.com/apekshakom/sentiment-analysis-of-restaurant-reviews

[13] Generating WordClouds in Python: https://www.datacamp.com/community/tutorials/wordcloud-python

Appendices

I mention my files over here and you will find all the information inside that files I annotated well so, anyone can understand code and I also mention necessary packages. You can find all files as mention below.

- Data_Cleaning.ipynb
- Machine_Learning.ipynb
- Natural_Language_Processing_&_wordcloud.ipynb
- Related_Analysis.ipynb