Christine Yao
Brown University
12/11/24

Mental Health in the Tech Industry

1. Introduction

Motivation

Mental health issues plague a large portion of the human population, and an even larger portion of workers in the tech field. Around 51% of workers in the tech field report facing distress and seeking treatment for their mental health issues yearly. The tech field is often a workplace filled with pressure, stress, burnout from long working hours and isolation, due to the nature of work in technology. People are paying more attention to the topic than in previous years, so companies have the potential to make mental health resources and work life balance better. A classification model for predicting if people need treatment could be used to help companies strongly identify which patients should be treated, implement better initiatives, and reallocate resources and money to the right places.

OSMI

The dataset was sourced from a mental health organization's website, OSMI [1]. OSMI is a non-profit organization focused on changing how mental health is addressed in the tech community. Founded by Ed Finkler, a developer and advocate for mental health awareness, the organization provides yearly surveys to assess the mental health attitudes of the people who work in technology. Open Sourcing Mental Illness (OSMI) has been hosting annual surveys to try to understand and spread awareness on this topic. Participants can click a link on the website to answer questions to the survey.

Previous Work

Previous work on modeling with the same dataset for a similar purpose has achieved a similar predictive power, mainly from notebooks that data scientists posted on Kaggle. Most models achieved an accuracy score of around 0.8, as with one Kaggler's dataset which ran Random Forest Classifier to predict the same target variable, and achieved a score of 0.797 [3]. In another Kaggler's notebook, the Logistic Regression model trained was able to get an accuracy score of 0.79 [4].

The purpose of this project is to develop a simple classification model that can predict if people need mental health treatment. There are 1259 observations and 27 features in the raw survey dataset.

2. Exploratory Data Analysis

First, the dataset was explored to get a general idea of the features, missing values, and unique values. Keeping in mind that the dataset is from a survey, each row in the dataset represents a response from an individual employee and each feature represents a survey question.

Data is missing in 4 out of 27 features with work_interfere, indicating if the respondent has a mental health diagnosis, if their mental health interfered with work, lacking 20% of data. Figure 1 shows that from the raw dataset, four categories are missing data.

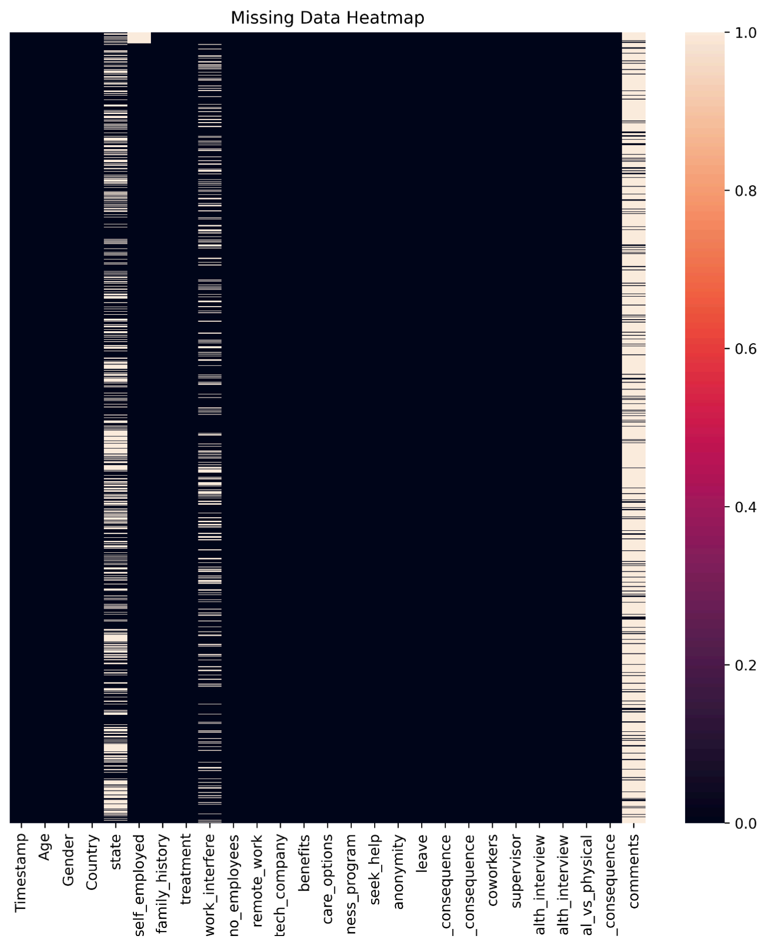

Figure 1. 4 out of 27 features contain missing numeric data.

From the 27 columns, three columns- timestamp, comments and state were dropped for simplicity and redundancy. No rows are dropped. Of the remaining 24 columns, 23 were categorical or ordinal and only one feature, age, was numerical.

Some features with stand out relationships between them are family history and seeking treatment. As shown in Figure 2, there is a major increase in those who seek treatment between those with a family history of mental illness and those without. Mental illness could be hereditary, but it can also be due to familial awareness and acceptance of the issue.

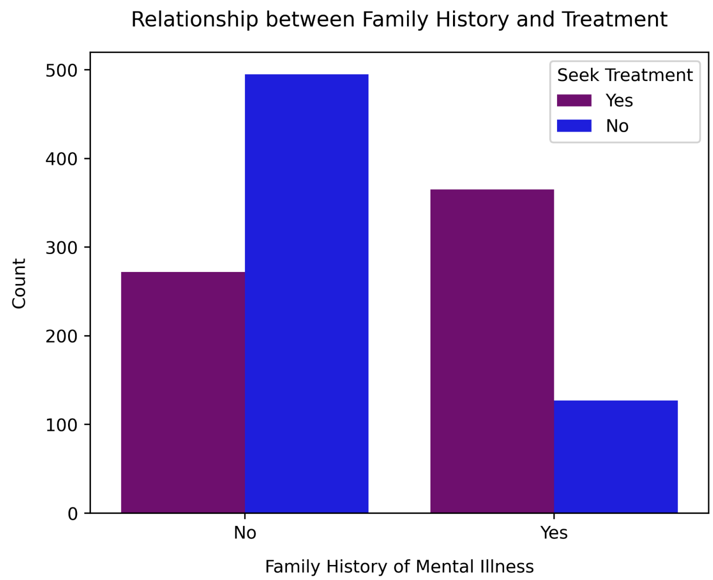

Figure 2. A family history of mental illness increases a person's chances of seeking treatment

Target Variable

The target variable in this dataset is 'treatment'. Each person's data is classified into two groups: 'Yes',sought treatment for mental health disorder, or 'No', did not seek treatment.

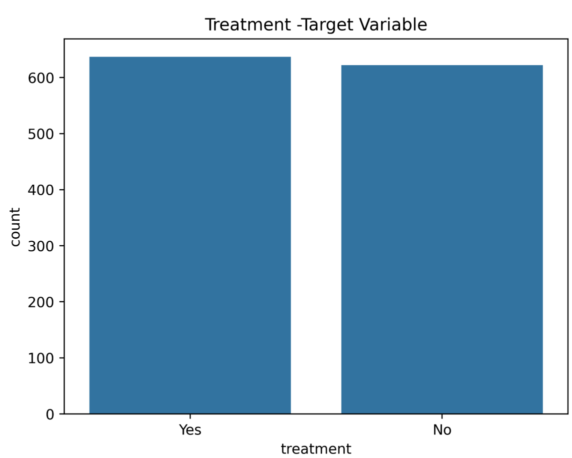

Figure 3. The target variable is balanced

3. Methods

Splitting Strategy

The initial strategy was to use a nesting splitting strategy to ensure reliable model evaluation and tuning. The train-test-split function split the dataset into 80% Train-Validation and 20% Test sets which is set aside for final evaluation. The remaining 80% train/val set is fed into a 5 KFold Split to reduce bias and variance stemming from overfitting on a single training set. A cross validation function was created to run the data over 10 different random states, allowing each random state's scores and best models to be saved in a test scores array and the average scores to be computed for each model.

Preprocessing Steps

Features such as "timestamp" of survey, "comments", and "state" were difficult or unnecessary for the machine learning process, and dropped. After dropping these features, the new dataset comprised 1259 rows and 24 features. Next, the missing data was handled. Mode imputation was used and set the missing values to "No" for self_employed. The other feature with missing values was the work_interfere column which lacked 20.97% of responses. The high number could be because the question targets workers that have a mental illness and did not apply to those without. "Not Applicable" was imputed into all Nan values [4].

The dataset consisted of a large range of ages, and some ages were unreasonable for the survey demographic. Ages that fell outside of typical working ages, from 18-80, were set to the median age of the data points [Fig 4].

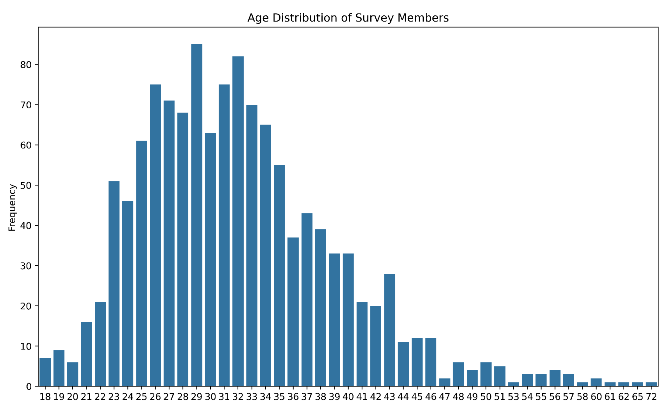

Figure 4. After preprocessing, the range of ages in the dataset fall within the expected age range

For data preprocessing, four ordinal features with categories that had an order to them were encoded using an Ordinal Encoder with defined category ordering. Then, one numerical feature (Age) was encoded with MinMaxScaler to normalize values into the range [0, 1]. The rest of the nine categorical features were encoded using OneHotEncoder, which creates binary indicator variables for each category. The handle unknown = 'unknown' parameter prevents errors when the test set has unseen categories. All of the features were collected and the preprocessed through

Sklearn's preprocessor and pipeline methods. The preprocessing steps are combined into a column transformer, ensuring that transformations for different feature types are applied simultaneously and consistently.

Machine Learning Pipeline

There were five machine learning algorithms trained on the dataset: Logistic Regression, Decision Tree, KNN, SVM, and XGBoost. For each algorithm, many parameters were tuned as shown in Table 1. The algorithm's optimal parameters are highlighted in blue and were found by gathering the most frequent parameters in each machine learning algorithm's set of scores and parameters gathered from the 10 random states.

Table 1. ML Models and their Corresponding Hyperparameters

Machine Learning Model	**Hyperparameter Values**
Logistic Regression	C: [0.1, 1.0, 10.0 ,50, 100] Penalty: l1 Solver: liblinear
Random Forest Classifier	Max_depth: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] Max_features: [0.25, 0.5, 0.75, 1.0]
Support Vector Classifier	Gamma: [1e-3, 1e-1 , 1e1 , 1e3 , 1e5] C: [1e-2 , 1e-1 , 1e0 , 1e1 , 1e2]
KNearest Neighbors Classifier	n_neighbors: [3, 5, 7, 9, 11] weights: ['uniform', 'distance'] metric: ['euclidean' , 'manhattan']
XGBoost Classifier	N_estimators: [50 , 100 , 150 , 300] Max_depth: [3 , 5 , 7 , 30 , 50]

Uncertainties due to splitting and non deterministic ML methods were addressed by training and testing over 10 different random states. The evaluation metric chosen is accuracy since the target class is binary classification and is balanced. In addition, the true positives and negatives are important because organizations can allocate resources to those who need treatment and save resources for those that do not need treatment whereas help those that need it.

4. Results

Model Comparison and Best Model Analysis

XGBoost classifier was the most predictive model with a score of 0.83 compared to the other models. Logistic regression performed the worst, although not far from the other models. When compared to the baseline accuracy of 0.51, all accuracy scores are above the majority class baseline accuracy. The standard deviation indicates how much variation there is in the model's performance across different test sets from the 10 random states, which ensures consistency in the model's performance and that it generalizes well across different data splits. Evaluating the model using standard deviation can assess the model's robustness and ability to work on unseen data.

Table 2. Average accuracy and standard deviation of the test scores for each ML Algo

Model	Average Accuracy	Standard Deviation
Logistic Regression	0.7160	0.0209
Random Forest	0.8389	0.0204
KNN	0.7767	0.0198
SVM	0.8297	0.0197
XGBoost	0.8389	0.0197

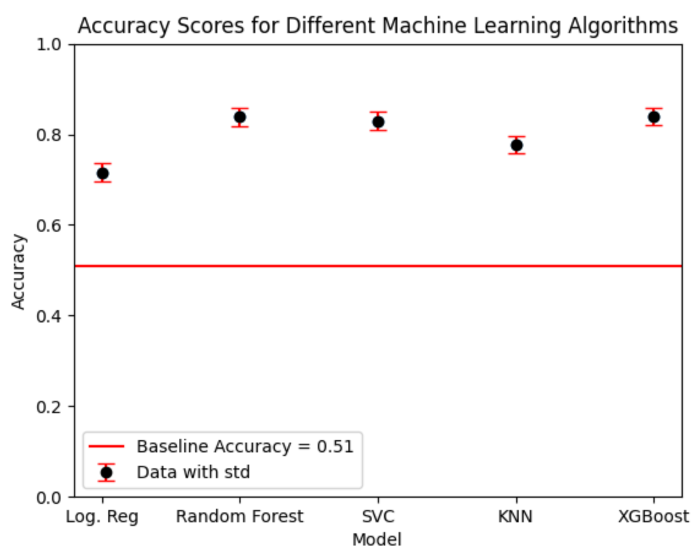

Figure 5. XGBoost model performed the best in terms of accuracy

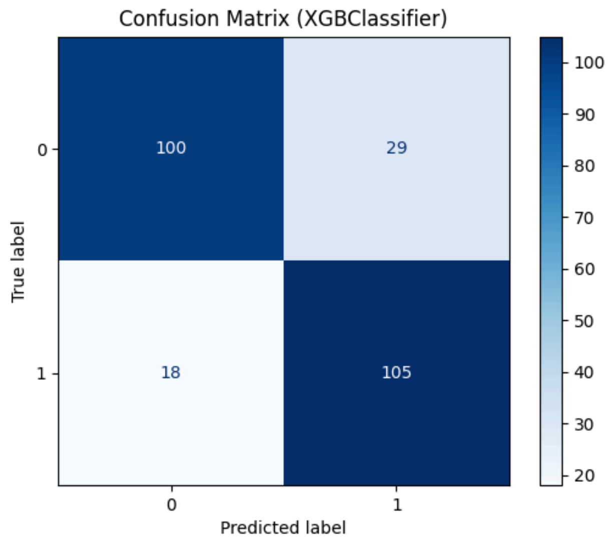

Figure 6. A look into the specifics of predicted classes w/confusion matrix using the best XGBoost model

Feature Analysis

To calculate feature importances, first, the data was resplit and preprocessed. During model training, preprocessing is automatically handled by the Pipeline. However, to calculate feature importances, preprocessing was applied outside the pipeline for interpretability.

Permutation feature analysis calculated feature importance on the best performing model, the XGBoost Classifier model.

Figure 7 shows the top 10 most important features using permutation feature importance across all data points through randomly reshuffling and calculating the difference in test score

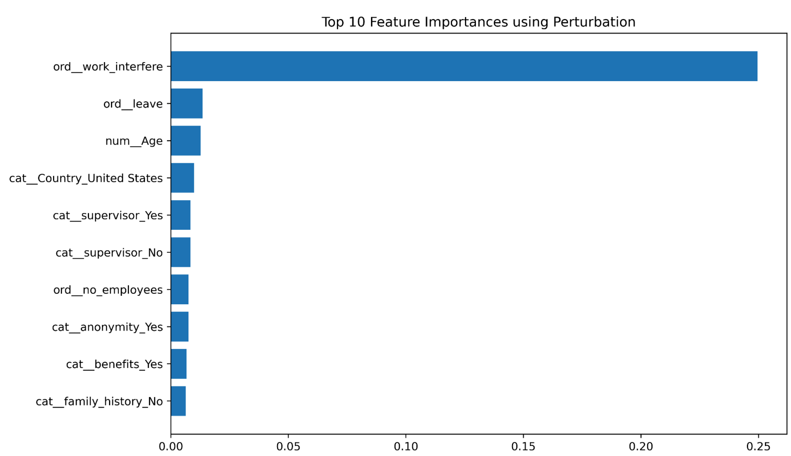

Figure 7. Globally important features using perturbation

The most surprising aspect of feature analysis was that the most important feature proved to be work_interfere, which from previous data analysis, was the feature with the most missing values that was not dropped, and imputed with a separate unknown category. This means that choosing the right method of imputing or dropping the data points was crucial.

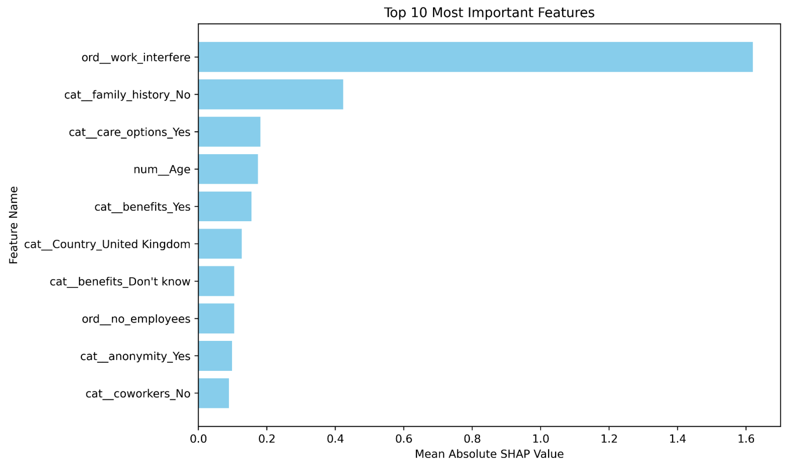

Figure 8. Mean absolute SHAP values to calculate global feature importances

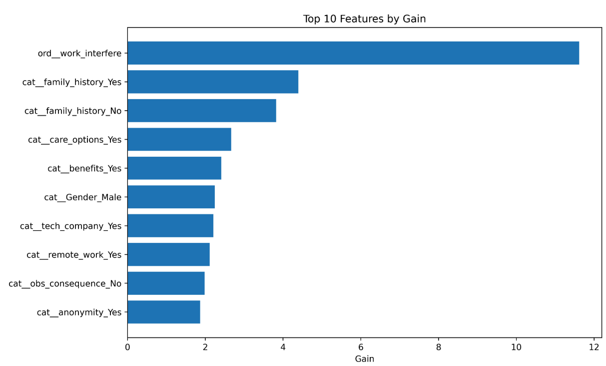

Figure 9. Global feature importances with gain method

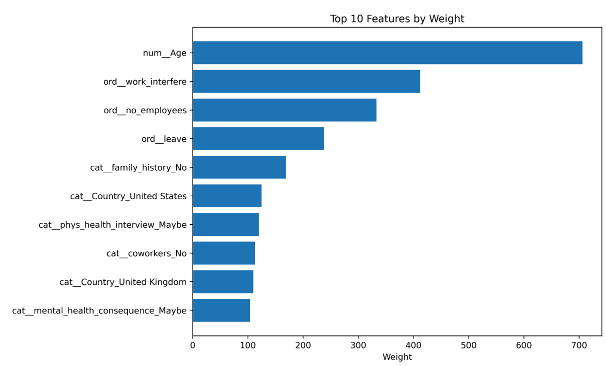

Figure 10. Global feature importances-weight method

Lastly, figures 9 and 10 show the top 10 most important features according to the gain and weight metric.

Local

The next three graphs show the impact of each feature to the output or predicted variable for a single observation. Person #1, person #101, and person #201 have different situations so their predicted output is different, as are the most important features and depend on the impact value/ input value of each feature. Each feature has a base value of around 0 but skewed to the right due to there being a slightly larger population of people in class 1. In the force plot, the more influential features have longer bars. For the specific person in figure 11.1, their answer to if mental illness interfered with their work contributed the most in boosting the prediction higher towards class 1. While in figure 11.3, this specific feature was also the most important but pushed the prediction towards class 0.

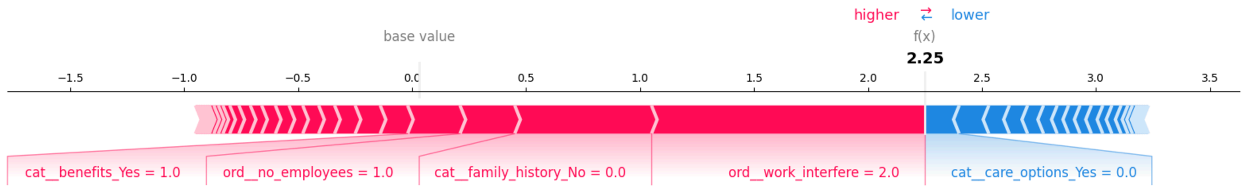

Figure 11.1. Force Plot for index 0 , person #1

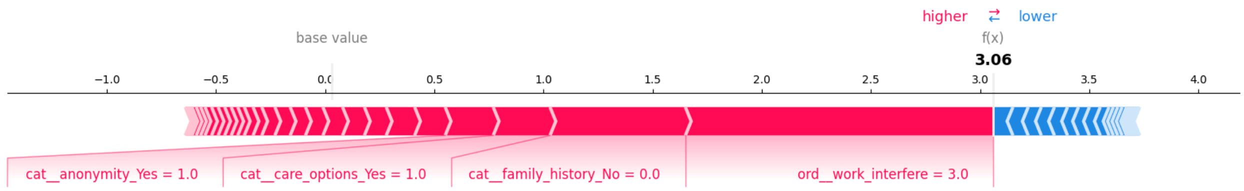

Figure 11.2. Force Plot for index 100, person #101

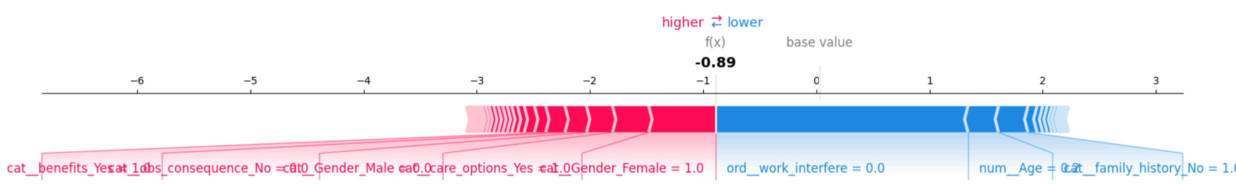

Figure 11.3. Force Plot for index 200, person #201

5. Outlook

Places of improvement to the model could be understanding and tuning parameters with more caution. The parameters used were typical parameters and could be expanded to include more. From dropping the State column the viewer is unable to see how that column may affect the prediction performance of the model, and if it is an important feature overall. It would also be worth looking into how to derive the perturbation features and feature importances directly from the data processed during the machine learning function with KFold CV and 10 random states. This dataset was run through a separate machine learning pipeline in order to get feature importances. Running it through the original would definitely be better for comparison and consistency. In addition, XGBoost can feature importances internally, a feature unused in this project but would be good to compare feature importances derived from this.

The survey design also has some flaws that could be improved on, such as asking less open ended questions, or only allowing those working in tech fields to submit an answer. Giving this as a mandatory survey at a couple of companies so that more than 2000 respondents would also increase the number of respondents and allow for quality answers and data analysis.

References

[1] https://osmihelp.org/research.html (data source)
[2]https://www.kaggle.com/code/devraai/unveiling-mental-health-secrets-in-tech-industry#Predictive-Modeling
[3] https://www.kaggle.com/code/abhishekpattanayak/mental-health-analysis#Logistic-Regression
[4]https://medium.com/learning-data/iterative-imputer-for-missing-values-in-machine-learning-32bd8b5b697a

Github Project Link

https://github.com/cyao98/mental-health-prediction