4518 PROJECT

TEAM NARS 22

MENTAL

HEALTH

Mental Health Survey in tech space

Machine Learning in Mental Health

Proposed by:
Ahmed Jawad Rashid (190021107)
Adiba Isfara (190021235)
Nayeb Hasin (190021115)
Atiq Aziz Sadat (190021105)

Supervised by:
Mr. Asif Newaz
Faculty, Department of EEE, Islamic University of Technology

Abstract

Recent advances in artificial intelligence along with the high incidence of mental illness and the need for comprehensive mental health care, have increased research into the significant gains of machine learning (ML) for the early identification, diagnosis, and treatment of mental health issues. By recognizing mental health symptoms and risk factors, predicting the course of diseases, customizing and improving treatments, and understanding patterns of human behavior, ML approaches may open up new avenues. Despite the potential uses of ML in the field of mental health, this is a relatively new field of study, and creating practical ML-enabled applications is fraught with several intricate, interrelated difficulties.

Keywords:

Machine Learning, Mental Health.

Table of Content

Introduction

Machine learning is a subset of AI that analyzes and learns from data using algorithms. By analyzing large amounts of data and spotting patterns that are difficult for people to notice, it can affect the detection and treatment of mental health problems.

Detecting and forecasting the possibility of specific mental health diseases, such as depression or anxiety, is one method machine learning can be utilized to solve mental health problems. This can enable healthcare professionals act early and provide tailored therapy for patients who may be at danger. Personalized treatment plans can be generated using machine learning by analyzing data on an individual's symptoms, habits, and treatment responses. This can assist mental health providers adjust treatment to each patient's individual needs.

In addition, machine learning can be used to enhance mental health evaluations by evaluating data and detecting aspects that may impact an individual's mental health. This can help specialists in mental health make more educated treatment recommendations. Lastly, machine learning can be used to improve mental health care delivery by analyzing data on patient outcomes and treatment efficacy to uncover best practices and increase care delivery.

In this project, we attempted to work on a dataset containing the opinions of employees from various nations regarding how workplaces can generate mental health issues and which specific element might be at the root of mental health concerns in the workplace. In order to proceed with the model, we first did data preparation and feature engineering. We analyzed many attributes using EDA to determine which attribute correlates most to mental health difficulties. After performing data preprocessing, various models were implemented to generate predictions. The models were modified further to improve the accuracy of their predictions. Implementing advanced machine learning models and evaluating the model's efficacy by analyzing the classification report comprised the final evaluation.

Mental Health Intro

Mental health is a significant concern at present, as demonstrated by startling statistics from time to time. As the prevalence of mental health concerns has increased dramatically, it is difficult to deny that we should make mental health awareness a higher priority.

Mental health is an issue that affects individuals of all ages and walks of life. It is crucial that we prioritize mental health awareness and seek to eliminate the stigma associated with it. This involves offering assistance and services to individuals and their loved ones, campaigning for policy changes, and raising community understanding about mental health.

There are numerous ways in which technology, such as machine learning models, can be utilized to promote mental health awareness and treatment. Some mental health apps, for instance, employ machine learning to track users' moods and provide individualized suggestions for self-care and stress management. Other mental health platforms employ machine learning to find trends in user answers to different prompts, enabling them to deliver individualized treatment plans and suggestions.

It is vital to stress, however, that these technologies should be used as an addition to traditional mental health care and support, not as a replacement for them.

Purpose of our dataset

We are at the eve of the 5th industrial revolution, and the integration of the current economy and world order will depend heavily on mass customization, collaboration, and Networked Embedded Cognitive Systems. Therefore, it is important to learn about and use technology to improve our understanding of human behavior, and to take baby steps toward designing an algorithm that can recognize the unique characteristics of a patient and thus the treatment that will be most effective for that individual. Our data is consistent with the core ideas of the 5.0 revolution.

Scope of mental health

By "mental health," we mean a person's state of emotional, psychological, and social equilibrium. It has a huge impact on our daily lives and the way we feel, think, and act. Many people experience mental health problems at some point in their lives; these range in severity from moderate to severe, with diseases like depression, anxiety, and bipolar disorder being among the most frequent.

In some cases, people with mental health issues may not realize that they need treatment. This can be due to a lack of awareness about the signs and symptoms of mental health problems, or because they are not able to recognize the impact that their mental health is having on their daily lives. Early detection and treatment of mental health issues is important because it can help prevent the condition from worsening and improve a person's overall quality of life.

One way to detect and analyze mental health issues is through the use of machine learning algorithms. These algorithms can analyze data on a person's behavior, emotions, and other factors to identify patterns and indicators of mental health problems. By understanding the attributes that are most closely associated with mental health issues in the workplace, it may be possible to develop strategies to prevent these issues from occurring in the future. This could involve implementing support systems and resources for employees, providing training and education on mental health, or taking other proactive measures to address potential risk factors. Overall, the goal of using machine learning algorithms to analyze mental health issues is to improve the early detection and treatment of these conditions, and to help prevent them from occurring in the first place.

The Scope of the Dataset

For this project, we aimed to understand the factors that contribute to a person's mental health. We used a dataset from Kaggle, which is based on a 2014 survey that assesses attitudes towards mental health and the prevalence of mental health disorders in the tech industry.

Mental health is a crucial aspect of a person's overall emotional, psychological, and social well-being. It impacts how a person thinks, feels, and behaves, as well as how they cope with stress, interact with others, and make decisions.

The data set was derived from a survey that asked questions about factors that could have an impact on an individual's mental health on the job. Successful, high-performing teams and people possess the abilities of effective communication and inclusion. Absenteeism, productivity, and employee engagement can all suffer as a result of mental health issues in the workplace. For instance, sadness can cause workers to miss a large number of workdays annually, which can have a major impact on the company's bottom line.

To combat these problems, we've been working on a model of mental health first aid with the goal of spotting signs of mental health problems early on, before it's too late. The goal of Mental Health First Aid training is to equip participants with the knowledge and skills necessary to identify the signs of a mental health or substance use concern or crisis, offer immediate help, and refer the individual to the proper staff resources. A company's bottom line can benefit greatly if its employees acquire and apply these vital communication and support skills.

Target of the dataset

1. Constructing a model that predicts whether or not an employee would seek treatment for mental health issues.
2. Determining the most significant factors that contribute to mental health issues among tech workers.

Background overview:

To construct the machine learning model, we required the following basic background knowledge:

Target Variable: The target variable is the feature of a dataset that the user wants to comprehend or predict using the other features. Using the independent factors in the dataset, the user seeks to predict or explain the dependent variable.

Attributes: Attributes are the characteristics or features of the data items utilized in machine learning. These variables, including size, color, and form, are utilized to characterize and categorize the data items in the collection. These properties are used by machine learning algorithms to make predictions or classify target variables based on patterns and correlations discovered in the training data.

Outlier: An outlier is a data point that differs considerably from the other points in a dataset. It has an exceptionally high or low value relative to the rest of the data points in the collection. Outliers may be generated by errors in data collection or measurement, or by real-world phenomena that are inadequately represented in the dataset.

Several methods exist for identifying outliers in a dataset:

1. **Visual inspection:** Plotting the data and searching for points that are considerably different from the rest of the data is one technique to identify outliers.

2. **Box plots:** A box plot is a graphical representation of the data that shows the min value, Q1, median, Q3, and max values. Outliers are often represented as values outside the whiskers in the box plot.

3. **Z-Scores:** Z-scores quantify the dispersion of a set of data around the mean. Potential outliers are data points having a z-score of more than 3 or less than -3.

4. **Interquartile range (IQR):** The IQR is the difference between the data's Q1 and Q3. Data points outside the range of 1.5 times the IQR below the first quartile or above the third quartile are considered potential outliers.

It is crucial to assess carefully whether an outlier should be deleted from the dataset or preserved for study. Sometimes, outliers might provide valuable data insights and should not be instantly disregarded.

Encoding:

Encoding is the process of converting categorical variables to numeric values so that they can be easily incorporated into a machine learning model.

There are different types of categorical variables:

Nominal Categorical Variable:

A nominal categorical variable is a form of categorical variable in which the categories have no intrinsic order or ranking. Nominal categorical variables are merely names for the categories. They have no numerical significance.

One Hot Encoding:

To do one-hot encoding, we must first identify all of the distinct data types. For each category, a new binary column is added to the dataset. If a data point falls within that category, we place a 1 in the relevant column. Otherwise, a zero is appended.

	Chittagong	Sylhet	Dhaka
Chittagong	1	0	0
Sylhet	0	1	0
Dhaka	0	0	1

In the example given, there are a total of three categories, resulting in three columns. Each relevant column is marked with a 1, while the others are left with the value 0.

Dummy Variable Trap:

The dummy variable trap is a condition that can occur when categorical variables are encoded with one-hot encoding in a machine learning model, whenever one variable can be derived from the rest variables then there is a problem faced known as multi-collinearity. To avoid the dummy variable trap, it is generally recommended to drop all of the dummy variable columns.

Example: If there are 3 dummy variables, then one column will be dropped. Using Sklearn's linear regression model, it is not necessary to drop the dummy variable column, as linear regression is aware of the dummy variable problem.

Ordinal Categorical Variable:

Ordinal Categorical Variable is concerned with the order or ranking of the variables. The grades in a specific subject might be A+, A, A- and so on. These depend on the order, which is why they are called ordinal categorical variables.

Label Encoding:
Ordinal classifications benefit from its use. Important categories are prioritized while assigning ranks.

A+	4
A	3
A-	2
B	1

The preceding table demonstrates that A+ is regarded as the highest grade, hence it is assigned the highest label.

Validating Model:

Training and Testing Data:

A. Train-Test Split Method

The train test split method of the scikit-learn library is a useful tool for dividing a dataset into training and testing sets. It lets users choose how much of the dataset to use for the training set, the testing set, and the random seed to make the results repeatable.

Training part of the dataset: Fitting the machine learning model requires it.

Testing part of the dataset: It is employed in the assessment of the fitness ML model.

Code:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Random State: Without the use of random state, the train and test sets for each execution are distinct, and the shuffling process is out of control.

Executions with random state=0 have the same train and test sets. It turns out that when we set random state to 42, we get the same train and test sets over multiple runs, but they're different from the train and test sets we got when we set random state to 0.

The performance score of a model is directly affected by the train and test sets. Because the train test split() procedure produces distinct train and test sets with varying random state integer values, the random state hyperparameter indirectly impacts the model's performance score.

B. K Fold Cross Validation and stratified cross validation:

To evaluate ML models with a small data set, a resampling method called cross-validation is employed. The number of categories into which a given data sample is to be partitioned is defined by a single parameter, k. As a result, k-fold cross-validation has become a common name for this procedure. In this process, each fold has one-fold that is a test set and the rest are training sets, and the average score is calculated by taking the average of the individual folds. This model may fail if there is a significant imbalance. In such instances, we employ the standard of stratified cross validation to prevent fluctuations in the projected outcome.

Handling imbalanced datasets: In instances where a model's accuracy is high but it cannot be executed correctly. This type of "naive" and misleading outcome is a result of the imbalanced dataset you are analyzing.

Confusion Matrix: The performance of a classification algorithm can be measured with a "confusion matrix," which is a kind of table. The quantity of correct and incorrect predictions made by the model are graphically shown.

These four values are what make up a table called a confusion matrix (The following is an example of a confusion matrix for a binary classification problem):

True Positives (TP): Cases where the model correctly predicted the positive class.

True Negatives (TN): Cases where the model correctly predicted the negative class.

False Positives (FP): Cases where the model incorrectly predicted the positive class.

False Negatives (FN): Cases where the model incorrectly predicted the negative class.

	True Positive	True Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Accuracy: (TP + TN) / (TP + TN + FP + FN)

Precision: TP / (TP + FP)

Recall: TP / (TP + FN)

F1 score: 2 * (Precision * Recall) / (Precision + Recall)

ROC: The performance of a binary classifier system as the discrimination threshold is changed is shown by the ROC curve. By plotting the TPR versus the FPR for varying cutoffs, we obtain the curve.

AUC: One way to evaluate a classifier's efficacy is by calculating its Area Under the ROC Curve (AUC). ROC area ranges from 0 to 1, with a larger number indicating better performance. The AUC for a random classifier is 0.5, while the AUC for a perfect classifier is 1.

Resampling the Training Data:

Under-sampling: Under sampling creates a balanced sample by decreasing the size of the class with a greater value. When adequate data is available, this strategy is employed.

Over-sampling: When inadequate data is available, oversampling is employed. It attempts to balance the dataset by increasing the sample size of unusual observation.There are two types of oversampling:

Random Oversampling: Random oversampling is a technique used to balance class distribution within a dataset by randomly repeating minority class observations.

SMOTE is an oversampling procedure that chooses a minority class observation and finds its k nearest neighbors. The algorithm then makes a new synthetic observation by randomly picking one of the k nearest neighbors and adding some noise to the feature values of the chosen observation.

Sampling:

Sampling facilitates rapid training and testing of a model on a subset of the data, enabling faster iterations and more efficient model tuning. There are different types:

Simple random sampling: Each data point in a dataset has an equal probability of being included in a sample drawn using a simple random sampling method.

Stratified sampling: Stratified sampling is a sampling technique in which the data is separated into homogeneous groups (strata) and a sample is drawn from each stratum. This technique is frequently employed to guarantee that the sample is representative of the full population.

Cluster sampling: Cluster sampling is a form of sampling in which the data is separated into clusters and a sample is collected from each cluster. This technique is frequently employed when data is grouped into natural groups or clusters.

Systematic sampling: Systematic sampling is a sampling technique in which the data is separated into a predetermined number of intervals and one data point is chosen from each interval.

C. Common Models in Practice:

Linear regression: The objective of the linear regression statistical model is to forecast a continuous dependent variable from a set of independent variables. Assuming a linear relationship between the independent variables and the dependent variable, the goal is to find the line of best fit that minimizes the distance between the observed data points and the line.

Logistic regression: Predicting a binary dependent variable from one or more independent factors is what logistic regression is all about. It is a form of linear regression useful in cases of classification problems with a binary outcome for the dependent variable (e.g., 0 or 1, yes or no, true or false).

SVM: For the SVM, or the Support Vector Machine An N-dimensional space is transformed into a hyperplane by the algorithm in order to widen the gap between the groups being classified. In the context of machine learning, a kernel function is a function that takes in two vectors as input and returns a scalar value indicating the similarity between those vectors. This function is commonly used in support vector machines (SVMs) and other kernel-based learning techniques.

KNN (K Nearest Neighbors): It is an algorithm for supervised machine learning. Both classification and regression issue statements can be resolved using the approach. To classify a new item using KNN, the method identifies the k nearest neighbors of the object in the training data and assigns the object to the class with the highest frequency among those neighbors. Various distance metrics, including Euclidean distance and Manhattan distance, can be used to estimate the distance between the objects.

Ensemble: An ensemble is an approach that combines the predictions of numerous models to produce more precise forecasts than any single model. Ensemble methods are frequently employed to produce state-of-the-art outcomes on a variety of problems, as they improve the stability and resilience of machine learning models.

GridSearchCV: GridSearchCV refers to the method of tweaking a model's hyperparameters to find its best settings. Choosing appropriate values for a model's hyperparameters is crucial to its effectiveness, as we've seen.

Methodology

The first step in developing an ML model is to specify the issue you're attempting to address. It's important to grasp the nature of the business issue, the nature of the data at hand, and the goals of the model. After identifying the issue at hand, the next step is to preprocess the data. As part of this process, you may have to clean the data, deal with missing values, and even normalize or scale the features. Separate the data into a training set and a test set. Both the training set and the test set are employed in the model-building and -testing processes. After picking a model, you'll want to teach it something with the data in the training set. In order to do this, an optimization technique must be used to determine the best values for the variables in the model.

Once the model has been trained, it must be evaluated based on how well it performs on a test set. This will show you how effectively the model generalizes to new data. Adjust the model's hyperparameters or choose a new model if its performance is inadequate. Once you're happy with the model's performance, you can put it into a production setting where it will make predictions on new data.

Phase 1

- Setting the table for forming data manipulation
- Importing the CSV file and necessary libraries
- Analyzing the dataset

Phase 2

- Understanding the data attributes through a purpose defined point of view
- Refining our goal for the final evaluation
- Understanding the purpose of study

Phase 3

- Pre-Processing the dataset

Phase 4

- Importing various sklearn toolkit to model our training datsets

- Hyper Test Tuning
- Finding the best model for our dataset

Phase 5

- Final evaluation of the work

Dataset source

Mental Health in Tech Survey: Survey on Mental Health in the Tech Workplace in 2014 (https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey)

Data Pre-Processing

Massive amounts of data are needed to train machine learning algorithms or models. Our model becomes more precise as the dataset grows. However, the vast amounts of data that we get from many sources are unprocessed raw data. It's going to be massive, and evaluating it might be quite complicated. Redundancy, noisy data, heterogeneity, null values, outliers, data type mistake, and many more issues can be present in raw data. Our model wouldn't be precise enough to make reliable predictions if we fed this data into the model. Data preparation is used in this situation.

To successfully fit our model, data must be cleaned or refined as part of the data pre-processing procedure. Data pre-processing is an essential component of machine learning because it gets the data ready for the model or algorithm so that it can operate as intended. The preparation of the data should be based on the kind of result we seek from our model.

In our dataset there are 24 features in total. They are the following:

I. Age
II. Gender
III. Country
IV. State
V. Self Employed
VI. Family History
VII. Treatment (Target Variable)
VIII. Work Interference
IX. Number of Employees
X. Remote Work
XI. Tech Company
XII. Benefits
XIII. Care Options
XIV. Wellness Program
XV. Seek Help
XVI. Anonymity
XVII. Leave
XVIII. Mental Health Consequence
XIX. Physical Health Consequence

XX. Coworkers
XXI. Supervisor
XXII. Mental Health Interview
XXIII. Physical Health Interview
XXIV. Mental vs. Physical
XXV. Observed Consequence

Among all these features most of them are categorical type data. To handle these categorical data we have done label encoding first.

Our target variable is either Yes or No. So, this would be binary encoding. And other features may require label encoding.

DATA PRE PROCESSING

```
In [49]: df.treatment.value_counts()

Out[49]: 1    637
         0    622
         Name: treatment, dtype: int64

In [50]: train_data=df
         train_data.treatment.value_counts()

Out[50]: 1    637
         0    622
         Name: treatment, dtype: int64

In [51]: from sklearn.preprocessing import LabelEncoder
         object_cols = ['gender', 'self_employed', 'family_history', 'treatment',
                'work_interfere', 'no_employees', 'remote_work', 'tech_company',
                'benefits', 'care_options', 'wellness_program', 'seek_help',
                'anonymity', 'leave', 'mental_health_consequence',
                'phys_health_consequence', 'coworkers', 'supervisor',
                'mental_health_interview', 'phys_health_interview',
                'mental_vs_physical', 'obs_consequence']
         label_encoder = LabelEncoder()
         for col in object_cols:
             label_encoder.fit(df[col])
             df[col] = label_encoder.transform(df[col])
```

Our target variable is either Yes or No. So, this would be binary encoding. And other features may require label encoding.

```
In [52]: df.columns

Out[52]: Index(['age', 'gender', 'self_employed', 'family_history', 'treatment',
                'work_interfere', 'no_employees', 'remote_work', 'tech_company',
                'benefits', 'care_options', 'wellness_program', 'seek_help',
                'anonymity', 'leave', 'mental_health_consequence',
                'phys_health_consequence', 'coworkers', 'supervisor',
                'mental_health_interview', 'phys_health_interview',
                'mental_vs_physical', 'obs_consequence'],
               dtype='object')

In [53]: list_col=['age', 'gender', 'self_employed', 'family_history', 'treatment',
                'work_interfere', 'no_employees', 'remote_work', 'tech_company',
                'benefits', 'care_options', 'wellness_program', 'seek_help',
                'anonymity', 'leave', 'mental_health_consequence',
                'phys_health_consequence', 'coworkers', 'supervisor',
                'mental_health_interview', 'phys_health_interview',
                'mental_vs_physical', 'obs_consequence']

         for col in list_col:
             print('{} :{} ' . format(col.upper(),train_data[col].unique()))
```

```
In [54]: from sklearn.preprocessing import LabelEncoder
         object_cols = ['gender', 'self_employed', 'family_history', 'treatment',
                'work_interfere', 'no_employees', 'remote_work', 'tech_company',
                'benefits', 'care_options', 'wellness_program', 'seek_help',
                'anonymity', 'leave', 'mental_health_consequence',
                'phys_health_consequence', 'coworkers', 'supervisor',
                'mental_health_interview', 'phys_health_interview',
                'mental_vs_physical', 'obs_consequence']
         label_encoder = LabelEncoder()
         for col in object_cols:
             label_encoder.fit(train_data[col])
             train_data[col] = label_encoder.transform(train_data[col])
```

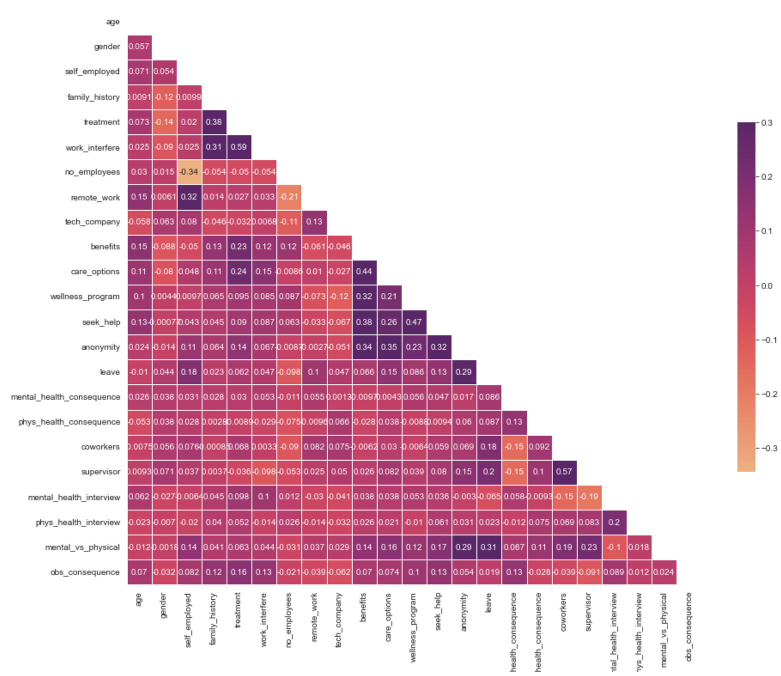
```
In [55]: df['treatment'].value_counts()

Out[55]: 1    637
         0    622
         Name: treatment, dtype: int64
```

```
In [56]: train_data=df
```

After our label encoding is completed, we gave used the correlation matrix to find out the best features possible to predicted the most accurate outcome.

```
In [57]: # Compute the correlation matrix
         corr = train_data.corr()

         mask = np.zeros_like(corr, dtype=np.bool)
         mask[np.triu_indices_from(mask)] = True

         f, ax = plt.subplots(figsize=(15, 15))

         # Draw the heatmap with the mask and correct aspect ratio
         sns.heatmap(corr, mask=mask, cmap='flare', vmax=.3, center=0,
                     square=True, linewidths=.5, cbar_kws={"shrink": .5}, annot = True)
```

Out[57]: <AxesSubplot:>



Exploratory Data Analysis

The initial stage after gather data from a particular source is exploratory data analysis, or EDA. In order to do so, we need to visualize our data is various graphs and charts to better understand which factors maybe important to our goal variable. With the use of statistical summaries and graphical representation, we utilized to identify trends, patterns, or to verify assumptions in this case.

EDA was carried out utilizing visualizations on our dataset, and the suitability for our variable's prediction model was assessed. It is possible to think of each column in the dataset as a survey question.

At first we get all the attribute column names and go one by one to ask surveying questions.

EDA

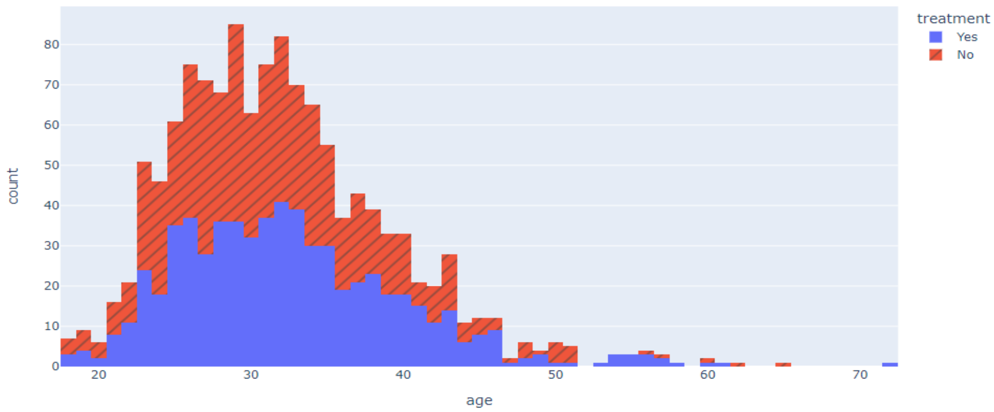
```
In [21]: health = df.copy()

In [22]: health.columns

Out[22]: Index(['age', 'gender', 'self_employed', 'family_history', 'treatment',
                'work_interfere', 'no_employees', 'remote_work', 'tech_company',
                'benefits', 'care_options', 'wellness_program', 'seek_help',
                'anonymity', 'leave', 'mental_health_consequence',
                'phys_health_consequence', 'coworkers', 'supervisor',
                'mental_health_interview', 'phys_health_interview',
                'mental_vs_physical', 'obs_consequence'],
               dtype='object')
```

Age Demographic receiving Mental Health Treatment

In order to determine if age may be used to distinguish between those who have received mental health care and those who have not, we first looked at the age demographic. So, we plotted a histogram and found our that different people from different age may or may not have taken any mental health treatment .

So, age is not a good feautre to use for training our model.

```
In [23]: px.histogram(health, x = 'age' , color = 'treatment' ,pattern_shape="treatment", nbins=100)
#As both distributions are merging , this won't be of much use
#for predicting classes
```

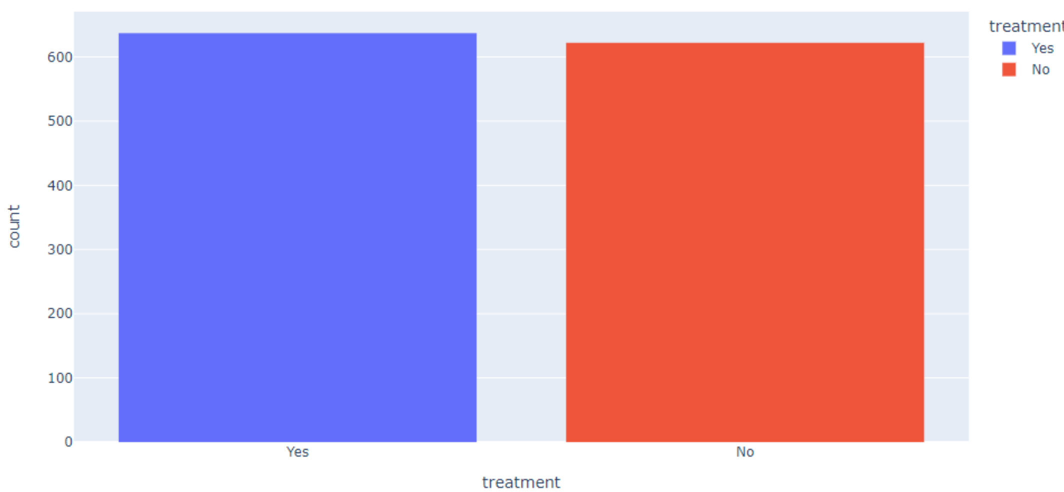


Have you sought treatment for a mental health condition?

In need of assistance are around 50% of people. This is a substantial section. We must make sure that every individual seeking aid is addressed since studies show that mental illness is a significant risk factor for suicide. Observations for creating models our objective variable is this.

No need for resampling exists since there is no class imbalance.

```
In [24]: plt.figure(figsize=(8,8))
px.histogram(health, x = 'treatment',color='treatment')
#From the histogram below we can see that a very large portion
#of our dataset have already shown tendencies of mental illness
#resulting in a major chance to commit suicide.
```




```
<Figure size 576x576 with 0 Axes>
```

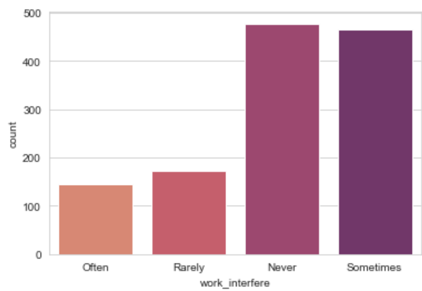
If you have a mental health condition, do you feel that it interferes with your work?

With a ratio of seldom, sometimes, and frequently, about 78% of respondents had encountered interruption at work.

About 45% of the time, mental health issues cause work-related interference. Nearly 80% of characters desire therapy, according to the narratives. The fact that some people still wish to receive treatment even when mental illness has never caused work-related issues is startling. It may be brought on by employment expectations that don't align with a worker's talents, resources, or demands.

If you are in charge of a tech company, you should think about giving employees who need treatment resources. This will improve employee satisfaction and undoubtedly raise productivity.

```
In [25]: sns.countplot(data = health , x = 'work_interfere', palette='flare')

         #About a big chunk of people feel that work intereference has a lot to do with
         #their mental state. So we must look into this

Out[25]: <AxesSubplot:xlabel='work_interfere', ylabel='count'>
```

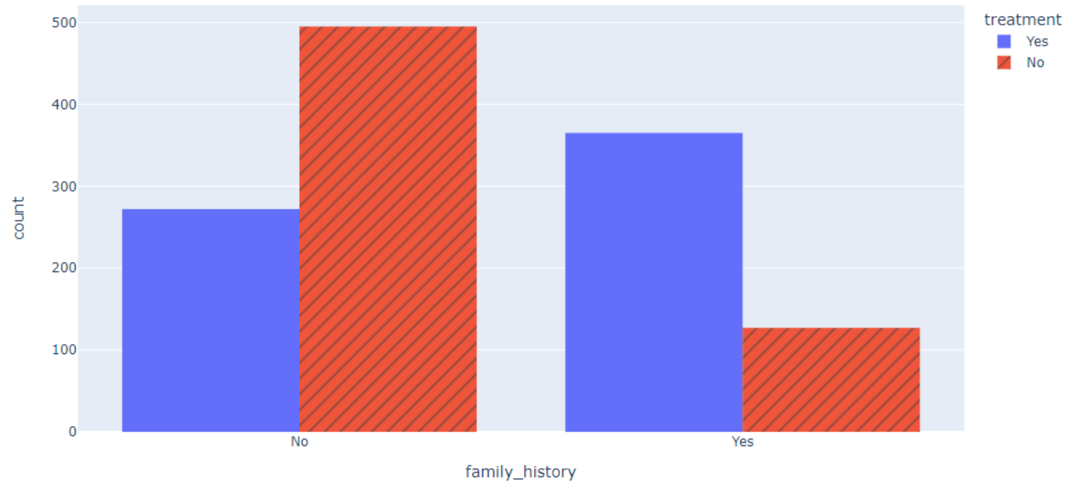



```
Do you have a family history of mental illness?
```

People who have mental illness in their families are more likely to seek therapy. 35% of those with no family history of mental illness also seek assistance. Observations for creating models

People who have a family history are more likely than those who don't to seek therapy.Family history will be a significant component.

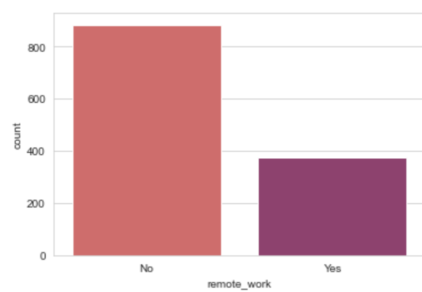
```
In [26]: px.histogram(health, x = 'family_history',color='treatment',barmode='group',pattern_shape="treatment")
         #family history is going to be an important feature
```



Do you work remotely (outside of an office) at least 50% of the time?

Whether they engage in remote work or not, about 50% of both kinds of persons seek therapy.There are somewhat more people who seek therapy who work remotely.The lack of social connection in distant mode may be to blame.

```
In [27]: sns.countplot(data = health , x = 'remote_work', palette='flare')
         #Due to lack of social interaction by working remotely, can trigger mental
         #health degradation

Out[27]: <AxesSubplot:xlabel='remote_work', ylabel='count'>
```



Does your employer provide mental health benefits?

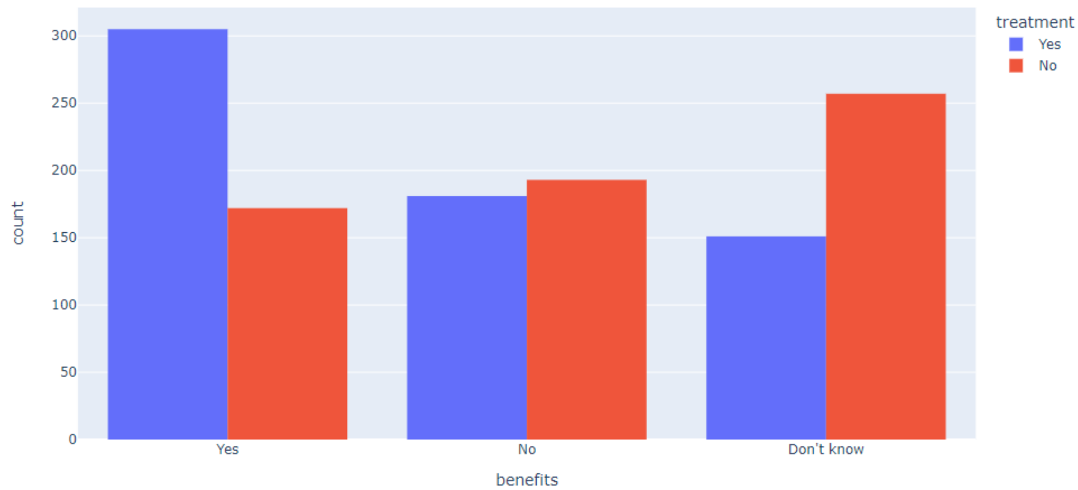
We can see that about 38% of respondents said their workplace offered them mental health benefits, whereas a sizable portion (32%) didn't even know if they had this benefit.

In the second graph, we can observe that 63% of those who responded "YES" to the question about mental health advantages claimed they were seeking medical attention. Thus, it is clear that the employer is making better use of its resources.

Even if you consider the expense, you should still go for it because the staff use it effectively.

Surprisingly, over 45% of those who rejected the company's offer of mental health benefits nevertheless desire to receive therapy for their mental health.

```
In [29]: px.histogram(health, x = 'benefits',color='treatment',barmode='group')
         #So we can see the employer resources are utilized to a larger extent.
         #Even if you think about the cost , you should definitely go for it,
         #because it is efficiently utilized by the employees.
```

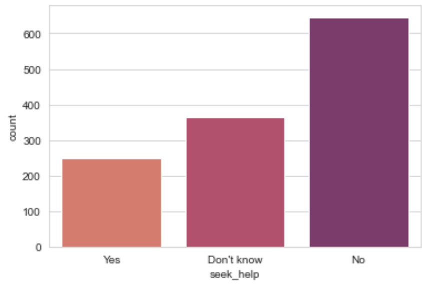


Do you know the options for mental health care your employer provides?

25% of employees are unsure whether the employer offers care alternatives, while 40% of employees receive no care options.

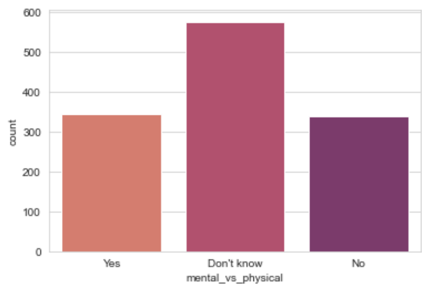
We can observe that 60% of employees whose employers don't offer healthcare choices are seeking help. These organizations must deal with this problem. This can support our argument that people with care alternatives are genuinely seeking therapy.

In [32]:
```
sns.countplot(data = health , x = 'seek_help', palette='flare')
```
Out[32]: <AxesSubplot:xlabel='seek_help', ylabel='count'>



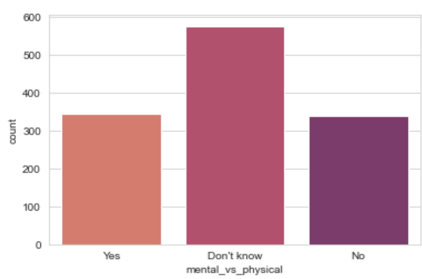
Do you feel that your employer takes mental health as seriously as physical health?

In [33]:
```
sns.countplot(data = health , x = 'mental_vs_physical', palette='flare' )
```
Out[33]: <AxesSubplot:xlabel='mental_vs_physical', ylabel='count'>

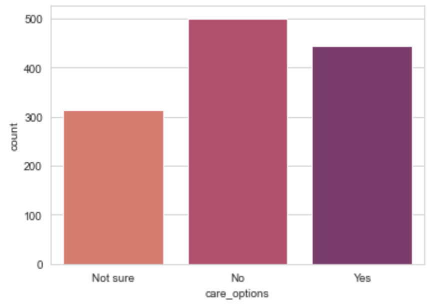


Employees are more likely to seek therapy than the other two groups if they believe their organization doesn't take mental health seriously or if they are unsure.

In [34]:
```
sns.countplot(data = health , x = 'mental_vs_physical', palette='flare' )
```
Out[34]: <AxesSubplot:xlabel='mental_vs_physical', ylabel='count'>

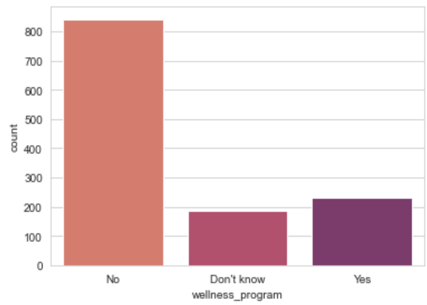


```
In [30]: sns.countplot(data = health , x = 'care_options',palette='flare')
Out[30]: <AxesSubplot:xlabel='care_options', ylabel='count'>
```



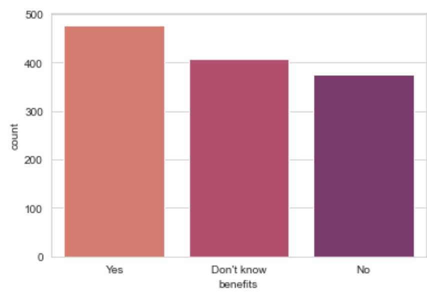
Has your employer ever discussed mental health as part of an employee wellness program?

```
In [31]: sns.countplot(data = health , x = 'wellness_program', palette='flare')
Out[31]: <AxesSubplot:xlabel='wellness_program', ylabel='count'>
```



Does your employer provide resources to learn more about mental health issues and how to seek help?

```
In [28]: sns.countplot(data = health , x = 'benefits',palette='flare')
Out[28]: <AxesSubplot:xlabel='benefits', ylabel='count'>
```



The majority of respondents' companies didn't include mental health as part of their employee wellness initiatives.

About 50% of those who are unaware of the program are looking for assistance. This implies that businesses should describe the benefits they offer for mental health. The employee wellness program should cover mental health for businesses. This must not be disregarded.

Models:

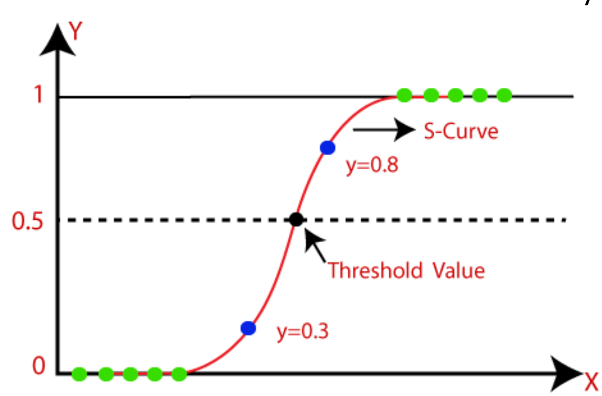
There is no one-and-only-way to determine which algorithm is best suited for a given machine learning model. Instead, the method differs based on criteria such as the amount of data being processed, the nature of that data, the computational cost, and the amount of time available to analyze it, among others. We have narrowed down our dataset and issue statement to a manageable set of variables in order to better select the best model.

1. ***Precision of the Output***
2. ***Computation and Training Time***
3. ***Linearity***
4. ***Amount of feature variable***
5. ***Volume of Training Data***

Models that we have used for our datasets are illustrated bellow along with their short intuition:

A. Logistic Regression:

Logistic regression is widely used in the area of Machine Learning known as Supervised Learning. It use a pre-established set of independent components to make predictions about a categorical dependent variable. Model outcomes with a categorical dependent variable can be predicted with the help of logistic regression. So, the final value must be either a discrete or a categorical. The figures it returns are probabilistic estimates that could be anywhere from 0 to 1, rather than a precise number. You can use words like "Yes" and "No," "0" and "1," "true" and "false," etc. While linear and logistic regression are conceptually similar, they differ significantly in application. Logistic regression can be used to solve classification problems, while linear regression is more effective when dealing with regression problems. Logistic regression cannot be used for non-linear issues because it is based on a linear decision surface. As practice, information is rarely presented in a table with clearly delineated columns and rows.

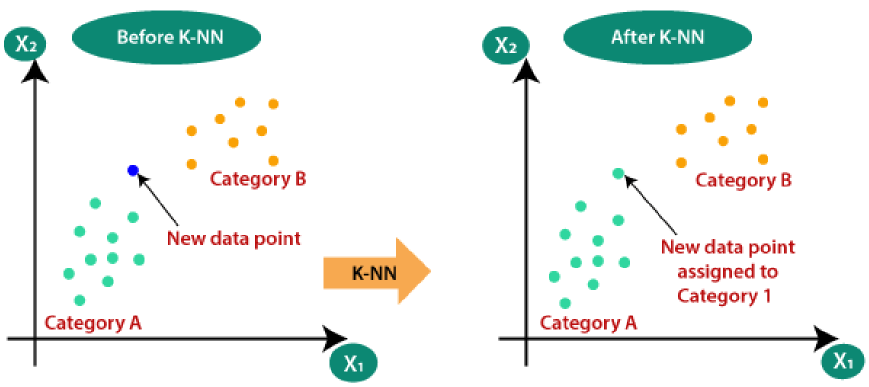


B. KNeighbours Classifiers

KNN is another supervised learning-based method. Using the K-NN technique, a new data point is assigned a classification based on its similarity to previously recorded data. As a result, the K-NN technique can be used to classify new data in a timely manner with a high degree of precision.

KNN algorithm can be separated into some steps:

1. Choose the Kth neighbor's number.
2. Determine the Euclidean distance between K neighbors.
3. Pick the K closest neighbors based on the Euclidean distance estimate.
4. Count the number of data points in each category among these k neighbors.
5. Assign the additional data points to the category where the neighbor count is at its highest.



Source : https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

C. Decision Tree

Decision tree algorithms are one type of supervised learning algorithms. In contrast to other supervised learning approaches, the decision tree technique can handle both classification and regression problems.

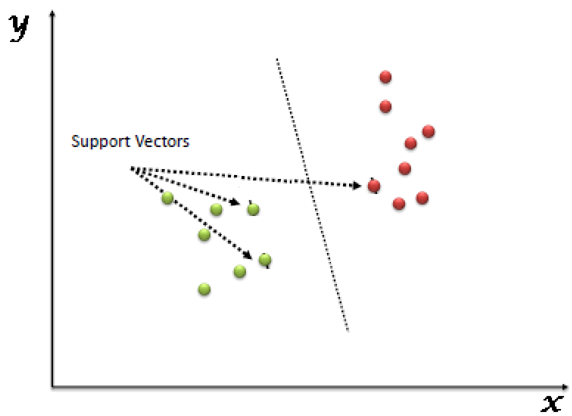
The goal of using a Choice Tree is to create a training model that can predict the target variable's class or value based on simple decision rules inferred from past data. Foreseeing a record's classification in a decision tree starts at the node closest to the root of the tree. We compare the values of the root attribute with those of the attribute in the record. We then go to the next node in the graph based on the results of the comparison, which may be the current node or a different one.

There are two types of decision trees

1) **Categorical Variable Decision Tree**
2) **Continuous Variable Decision Tree**

D. SVM

The "Support Vector Machine" (SVM) analysis is a supervised machine learning technique useful for solving classification and regression problems. However, its main use is in problems of categorization. The SVM algorithm reads each data point as a coordinate in an n-dimensional space (where n is the number of features), with the value of each feature being that coordinate. Our next step is to identify the hyper-plane that makes the most discernible divide between the two sets of data, and thereby classify the information.



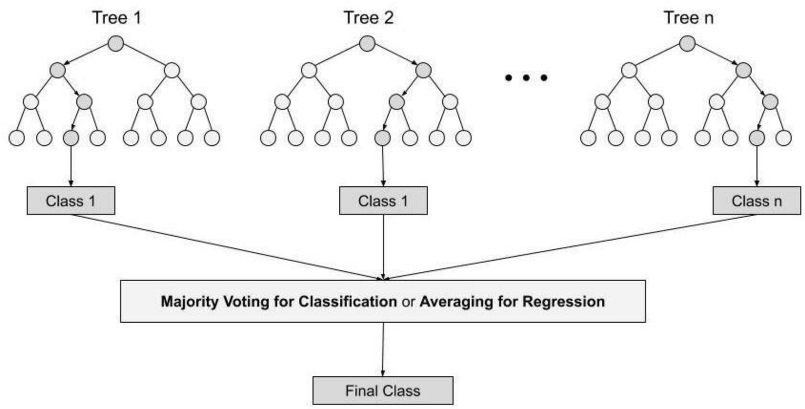
Source: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

E. Random Forest

Random forest is a common supervised machine learning approach used for classification and regression issues. Data from several samples are used to construct decision trees, with the mean used for classification and a majority vote determining regression.

The capability to analyze datasets with both continuous variables (for use in regression) and categorical variables is a key characteristic of the Random Forest Algorithm (for use in classification). To put it another way, it does a better job than other methods of categorizing problems. There are some steps followed in random forest algorithm:

1. N records at random are selected from a dataframe of k records.
2. Distinct trees are built for every sample.
3. A result is generated by each decision tree.
4. For classification and regression, the final result is based on either majority voting or averaging.



Source: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Ensemble Machine Learning Approach

The goal of creating an ensemble is to improve upon the performance of a given classifier by combining the outputs of multiple underperforming classifiers into a single, more robust model. Here, separate classifiers all cast ballots, with the majority decision being the final prediction label. When compared to a single classifier or a simple classifier, the accuracy provided by an ensemble is much higher. Ensemble methods can achieve parallelization by dividing each base learner over multiple processors. Ensemble learning approaches are meta-algorithms that combine various machine learning techniques into a single prediction model, with the goal of improving performance. Examples of ensemble methods that can decrease variability, bias, or improve predictions include the bagging strategy, the boosting approach, and the stacking approach.

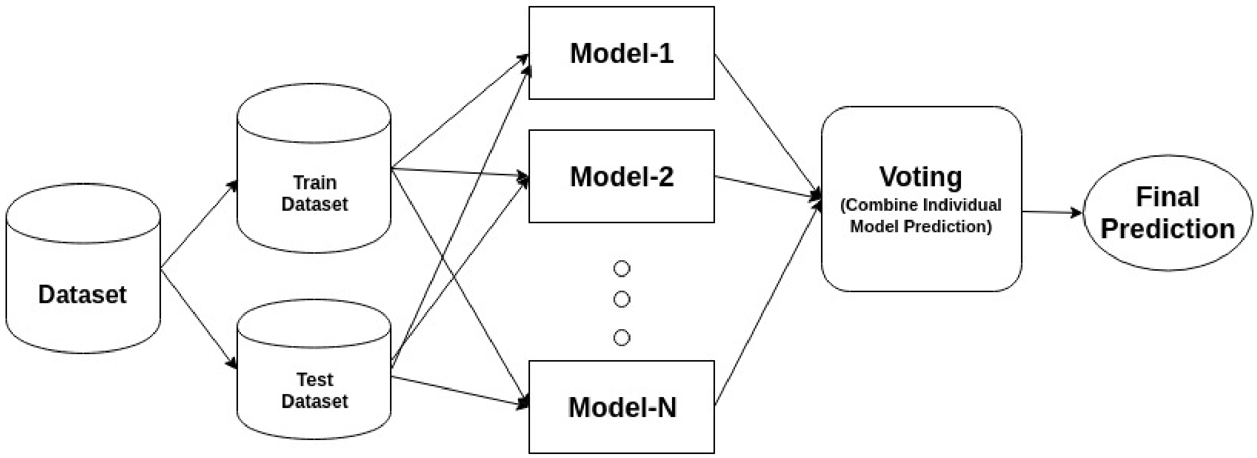
There are numerous ensemble methods, including:

Boosting is a type of ensemble strategy that includes training weak models and combining their predictions to create a robust model. The weak models are trained consecutively, with each model attempting to repair the errors made by its predecessor. Such boosting techniques as AdaBoost and Gradient Boosting are examples.

Bagging is a type of ensemble method that includes training many models independently on random subsets of the training data and aggregating their predictions by voting or averaging. Bagging is applicable to decision trees, neural networks, and other model types. Random Forest is an example of bagging algorithm.

Random Forest: Random Forests are a sort of ensemble learning method of decision tree algorithm in which numerous models are trained and their predictions are integrated to provide a final forecast.

Stacking: Stacking is a form of ensemble method involving the training of numerous models on the same data and producing the final prediction based on the outputs of the individual models.



Source: https://www.datacamp.com/tutorial/adaboost-classifier-python

F. AdaBoost

The AdaBoost classifier integrates the best features of numerous less-than-stellar classifiers into a single powerful, pinpoint-accurate model. The core idea of Adaboost is to train the data sample and fine-tune the classifier weights in each iteration until accurate predictions of unusual events are achieved. Any machine learning method that makes use of training set weights can serve as the underlying classifier.

Adaboost should meet two conditions:

1. It is recommended to train the classifier interactively on many weighted training cases.
2. In order to provide the most suitable match for these circumstances, it repeatedly tries to reduce training error.

G. XGBoost

XGBoost is a collective learning method. The results of a single machine learning model may not always be reliable. Ensemble learning provides a structured strategy for integrating the predictive abilities of multiple learners. The result is a unified model that can account for the results from multiple models.

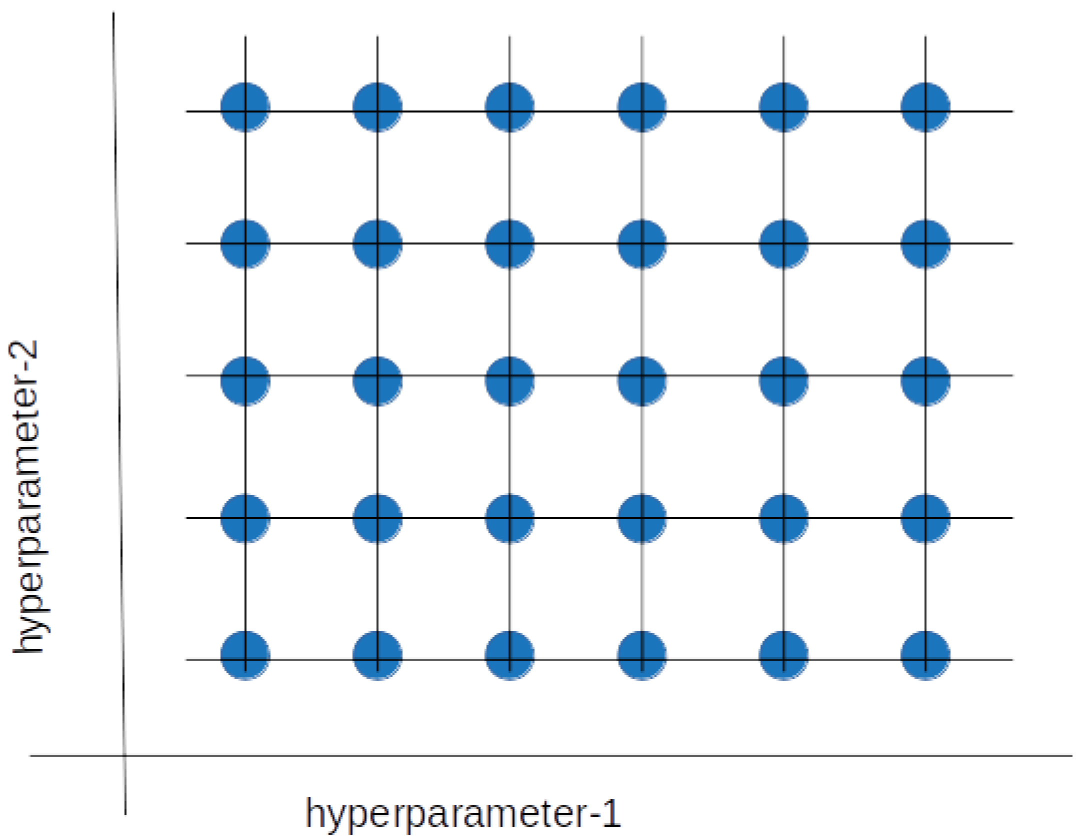
It's possible for the foundation learners, also known as the ensemble's models, to be derived from different learning methods, or they could have been created using the same technique. Bagging and boosting are two often used ensemble learners. There are many statistical models that can benefit from these two techniques, but decision trees have been the most prevalent in practice.

Predictions:

Still, before we can use our data to make predictions, we need to fine-tune the hyperparameters of our algorithms. Specific classes of variables, called hyperparameters, have a significant impact on the models' learning processes. Hyperparameter tuning is necessary for manipulating a machine learning model's output. If our hyperparameters aren't optimized to minimize the loss function, our forecasted model parameters will produce subpar results. This is another evidence that our model is not perfect. In actual use, the accuracy or confusion matrix will degrade further.

We use a method called grid search CV to fine-tune our information. We fit the model with every possible combination of the discrete hyperparameter values after first creating a grid of them. The results of each model set are recorded, and the optimal set is selected.

Hyper Test Tuning



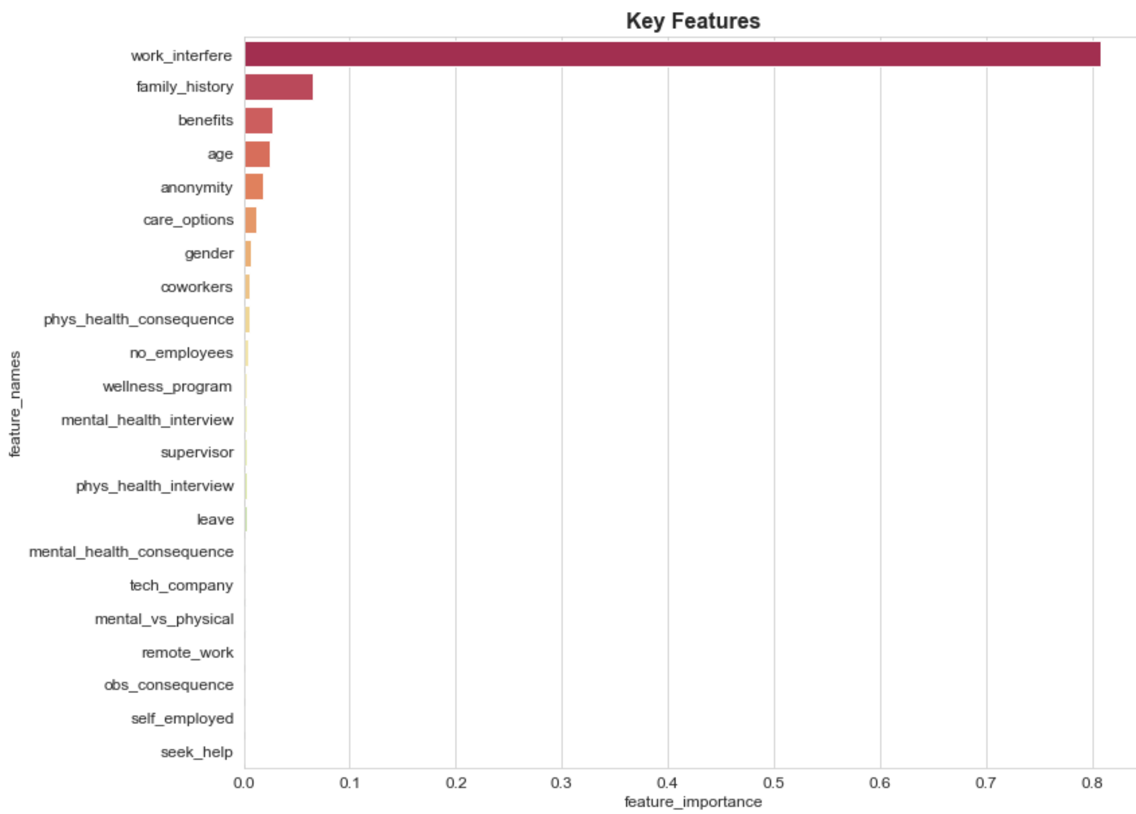
	model	best_score	best_params
0	XGBClassifier	0.836543	{'colsample_bylevel': 0.5, 'colsample_bynode':...

	model	best_score	best_params
0	svm	0.801387	{'C': 10, 'kernel': 'linear'}
1	random_forest	0.822946	{'max_depth': 3, 'max_features': 8, 'n_estimat...
2	logistic_regression	0.787763	{'C': 1}

Figure: Hyperparameter tuning of different models

The best parameters were used as the machine learning model's input after hypertuning.

Identify the key features that lead to mental health problems in tech space:



Inconvenience caused by one's job is the primary cause. Concerns about mental health should prompt employers to inquire as to whether or not workers are able to do their duties. The care options (programs and benefits) provided by an employer may also play a role in shaping an employee's decision to seek treatment.

Scores:

Metrics used for assessment shed light on how well a model works. It is vital that evaluation metrics can differentiate between various model outcomes. Classification reports are used as metrics for assessing machine learning programs. This approach will show you the precision, recall, F1 Score, and reliability of your trained classification model. In the context of machine learning, it is one of the metrics used to evaluate a model's efficacy in making classifications. It shows your model's precision, recall, F1 score, and validity. This helps us see how well our trained model is performing in general. To make sense of the machine learning model's classification report, you need to be well-versed in all of the metrics presented within.

A. Classification report:

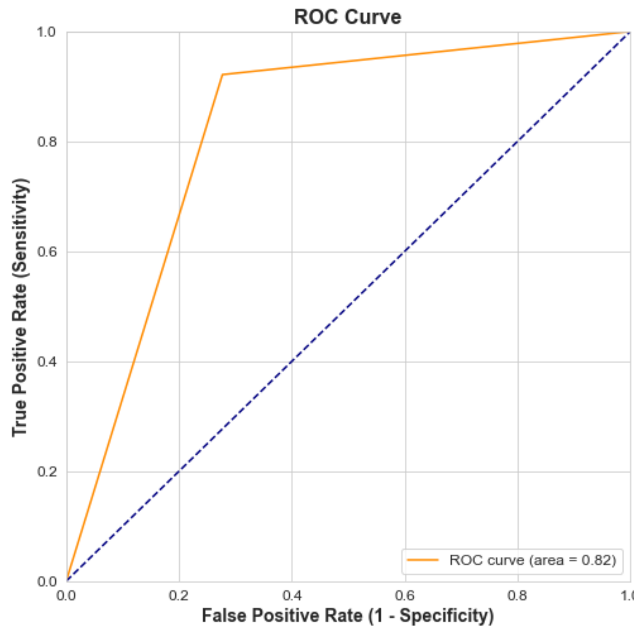
	precision	recall	f1-score	support
0	0.93	0.70	0.79	128
1	0.75	0.94	0.84	127
accuracy			0.82	255
macro avg	0.84	0.82	0.82	255
weighted avg	0.84	0.82	0.82	255

The percentage of accurately anticipated positive real values, or recall, was 70% for those who hadn't undergone therapy and 94% for those who had.

Accuracy can be defined as the rate at which observed results match those that would be considered correct. In this instance, we achieved an accuracy of 0.93 and a precision of 0.75. Due to the discrepancy in our data, we will use f1-scores in this case instead of accuracy and recall ratings.

B. ROC Curve:

ROC Curve area=0.82



C. Confusion matrix:

Given the uneven distribution of our data, we have opted to use the confusion matrix as our classification matrix here. Using the matrix, we can deduce that our true positive and true negative rates are 34.5 and 46.27%, respectively, while our false positive and false negative rates are roughly 15.68% and 3.50%, respectively. Due to the fact that our true positive and true negative values are larger than our false negative and false positive values, we may conclude that our forecast is relatively accurate.

Challenge Factors

1. Dataset entries were informal, thus lacking critical parameters to better analyze the target variable
2. The dataset is not competent with more advanced artificial intelligence tools currently deployed
3. High reliance on selective attributes, e.g., work_interference, family_history etc.

Key takeaway

- We have a developed a model which can predict whether a employee seeks mental health treatment or not.
- Work interference has the largest contribution. Whether the employee's mental health issues interfering with the work is the thing that the company should ask for its employees.
- Family history and care options (programs and benefits) provided by company is also influential in employees who want to get treatment.

- For all the remaining features, there has been a little contribution.
- noticing/knowing some of these features beforehand can even help support an individual who may be experiencing a mental health issue and connect them with the appropriate employee resources.

Key Takeaways from Reputed Research Papers

1. **Machine Learning in Mental Health: A Systematic Review of the HCI Literature to Support the Development of Effective and Implementable ML Systems:** The creation of a voice-based stress indicator that may be used as an automatic mental health monitor, A controlled storytelling study (ST) vs a reliving study (RL) with ecological validity, In order to detect social anxiety and depression from extended audio samples, we developed a unique feature modeling technique within a weakly supervised learning framework. The integration of self-reports and physiological measures of skin conductance has allowed for the development of a multi-modal method to estimating changes in the intensity of PTSD symptoms (Thieme, Belgrave and Doherty, 2020).

Conclusion: *The datasets of audio, accelerometer, audio+activity, cellular devices are beyond the scope of manipulation using data science and the research of the team for this semester. In the days to come, we hope to make progress on more complex ideas.*

2. **Mental Health Prediction Models Using Machine Learning in Higher Education Institution:** This paper describes the use of machine learning models for predicting student mental health in higher education settings. First, they forecast that stress, sadness, and anxiety will be the three most common mental health issues among college students and provide background information on these conditions. They show how the DASS-21 data can be utilized for modeling by assigning labels to the individuals based on their scores on various criteria. In the meantime, the WHOQOL criteria were fed into a model of health issues using a feature selection method. To do this, various machine learning techniques are used to model the health issue (Mutalib, 2021).

   Conclusion: *The datasets and the model used in this research was a good indicator of how particular attributes are utilized based on their utilitarian perspective when compared to target variable. And how different models used have different accuracies and predictions. And this laid the cornerstone of confidence for our team to dive into experimenting with the dataset that we had chosen.*

Contribution

During the course of the project, we took a decentralized approach, with everyone doing their own work before bringing their findings to weekly team meetings. Afterwards, we'll evaluate how well the team did by looking at how each member performed throughout the project and how their individual scores and approaches to solving problems compared to the final product. We made an effort to make sure everyone on the team is familiar with the steps necessary to complete a project independently, and we focused our efforts on meeting the demands of the team and the sectors to which we had been individually assigned. This means that we all worked on the project separately and conducted our own study into various sectors in order to get the broadest possible conclusions. Then, in the next team meeting, we reviewed our findings. It won't be wise for us to pinpoint any particular effort to one specific member since the approach we took was extremely decentralized and it was a true team effort.

Conclusion

Machine learning cannot be used ethically or appropriately to infer conclusions about mental health from a dataset. Understanding mental health is difficult since it depends on a person's unique set of experiences, perspectives, and patterns of behavior. Concerns have been raised that the use of a machine learning algorithm to assess a dataset related to mental health could lead to incorrect or even harmful conclusions and suggestions. Instead, the analysis and interpretation of mental health data should involve professionals in the fields of psychology, psychiatry, and social work. They have the ability to provide assistance and care that is tailored to the specific needs of each client by taking into account the individual's life history.

The use of machine learning for the study of mental health data should be approached with caution, and appropriate consultation with mental health practitioners should be sought before beginning any such analysis. However, the results of our dataset and the machine learning analysis that we conducted showed promise, so the HR department of a technology company can conclude with confidence that, yes, Work Interference is a field which can be improved in the office space, and better policies and work procedures can be developed and included in the employee handbook to improve the mental health of the workplace and slightly increase overall productivity. To sum up, we were able to accomplish the goals we set out to accomplish at the outset of this documentation, which were to move closer to the heart of Revolution 5.0 and develop an algorithm to pinpoint the specific characteristics of a patient that will lead to the most effective course of treatment.

Reference list

Thieme, A., Belgrave, D. and Doherty, G., 2020. Machine learning in mental health: A systematic review of the HCI literature to support the development of effective and implementable ML systems. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *27*(5), pp.1-53.

Abd Rahman, R., Omar, K., Noah, S.A.M., Danuri, M.S.N.M. and Al-Garadi, M.A., 2020. Application of machine learning methods in mental health detection: a systematic review. *Ieee Access*, *8*, pp.183952-183964.

Mutalib, S., 2021. Mental health prediction models using machine learning in higher education institution. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(5), pp.1782-1792.

Cho, G., Yim, J., Choi, Y., Ko, J. and Lee, S.H., 2019. Review of machine learning algorithms for diagnosing mental illness. *Psychiatry investigation*, *16*(4), p.262.