Mental Health Detection using Machine Learning

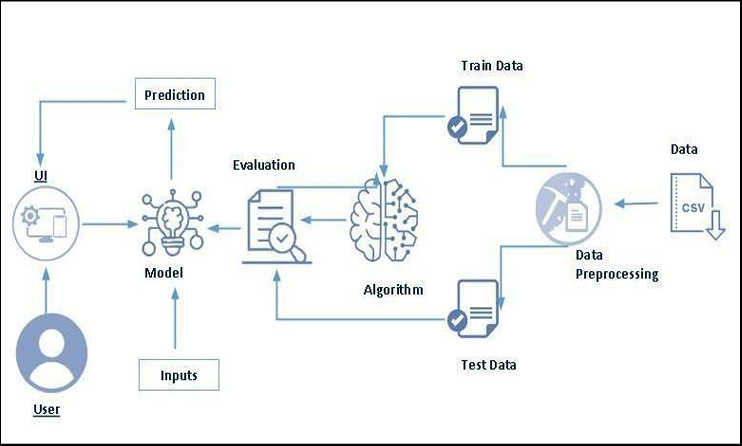
Introduction

Mental health has become a critical concern in today’s fast-paced and stressful environment. Early identification and intervention can significantly improve outcomes for individuals experiencing mental health challenges. This project aims to harness the power of Machine Learning (ML) to predict mental health conditions based on patient data, such as survey responses and electronic health records.

The core objective of this system is to develop a predictive model that can assist healthcare professionals in identifying individuals at risk of mental health disorders like anxiety, depression, and stress. By automating the analysis of large datasets through intelligent algorithms, the system enables timely and data-driven decision-making in clinical and wellness settings.

This documentation outlines the end-to-end architecture of the system, covering key stages including data preprocessing, model training and evaluation, and deployment. A user-friendly interface allows seamless interaction, enabling users to input relevant data and receive predictions in real time. The overall design focuses on accuracy, efficiency, and accessibility, making it a practical tool in mental health support and diagnostics.

Technical Architecture:



Project Flow:

* User interacts with the UI to enter the input.
* Entered input is analysed by the model which is integrated.
* Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

* Data collection
* Collect the dataset or create the dataset
* Data pre-processing
* Removing unnecessary columns
* Checking for null values
* Visualizing and analysing data
* Univariate analysis
* Bivariate analysis
* Descriptive analysis
* Model building
* Handling categorical values
* Dividing data into train and test sets
* Import the model building libraries
* Comparing the accuracy of various models
* Hyperparameter tuning of the selected model
* Evaluating the performance of models
* Save the model
* Application Building
* Create an HTML file
* Build python code

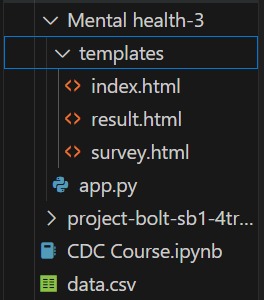
Prior Knowledge

We must have prior knowledge of the following topics to complete the course:

* ML Concepts
* Supervised learning: https://youtu.be/QeKshry8pWQ
* Unsupervised learning: https://youtu.be/D6gtZrsYi6c
* Decision tree: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm
* Random Forest: https://www.javatpoint.com/machine-learning-random-forest-algorithm
* KNN: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
* Xgboost: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/
* Evaluation Metrics: https://youtu.be/aWAnNHXIKww
* Flask Basics: <https://youtu.be/lj4I_CvBnt0>

Project Structure:

We have the project folder which has the files as shown below:



* the data folder contains the .csv file used for our project
* We are building a flask application that needs the HTML pages to be stored in the templates folder and a python script app.py for scripting.
* Notebook folder contains model training file MentalHealth.ipynb
* model.pkl is our saved model. We will use this model for flask integration.

Project Objective:

By the end of this project, we will:

* Know fundamental concepts and techniques used for machine learning.
* Gain a broad understanding of data.
* Have knowledge of pre-processing the data/transformation techniques and some visualization concepts before building the model
* Learn how to build a machine learning model and tune it for better performance
* Know how to evaluate the model and deploy it using flask

Milestone 1: Data Collection

1. Activity 1: Collect the dataset or create the dataset

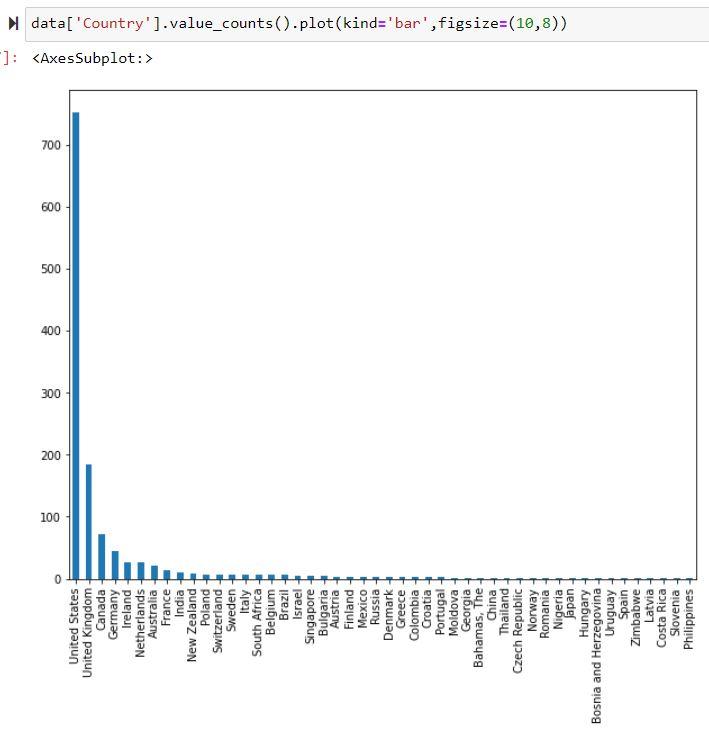
There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc. In this project, we have used survey.csv data. This data is downloaded from kaggle.com. Given below is the link to download the dataset.

Link: [Mental Health in Tech Survey | Kaggle](https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey)

Milestone 2: Data Pre-processing

1. Activity 1: Remove unnecessary columns

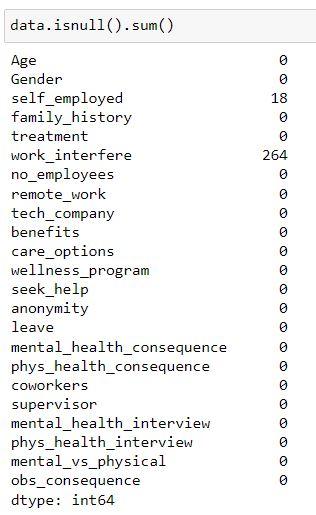
The below picture shows the distribution of countries



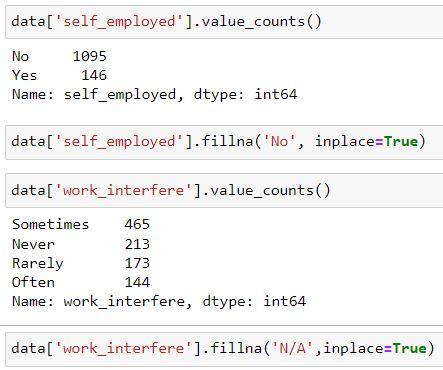
Since the countries are not evenly distributed, keeping this column will induce bias in our model.  So, we will be removing country and state columns.  We will also remove timestamp and comments columns as they do not contribute to providing relevant information.

1. Activity 2: Handling null values and dealing with wrongly entered data

To check for null values, .isnull() function is used along with .sum() function to the dataframe.

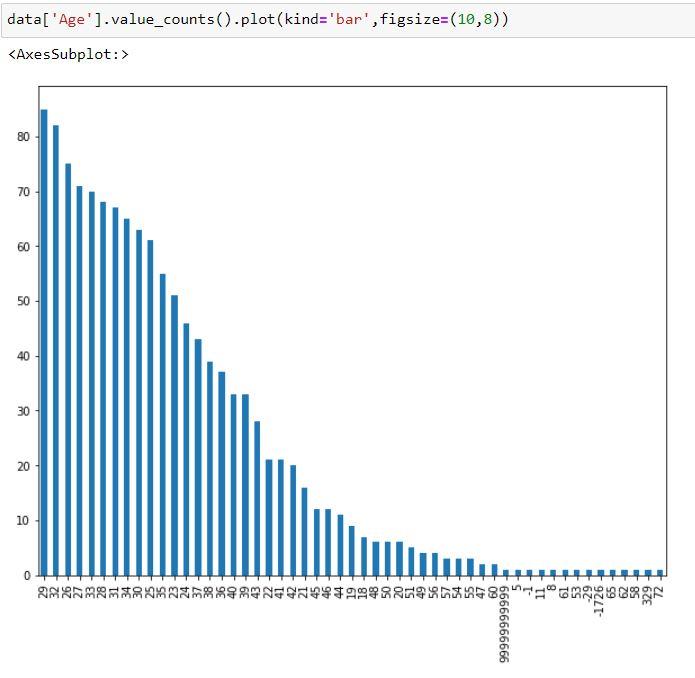


We observe that 2 columns - self\_employed and work\_interfere contain null values.  Let us fill the self\_employed column with ‘No’ and work\_interfere column with ‘N/A’ in place of null values.  In order to do this, we will make use of .fillna() function



Now our dataset is free of null values.

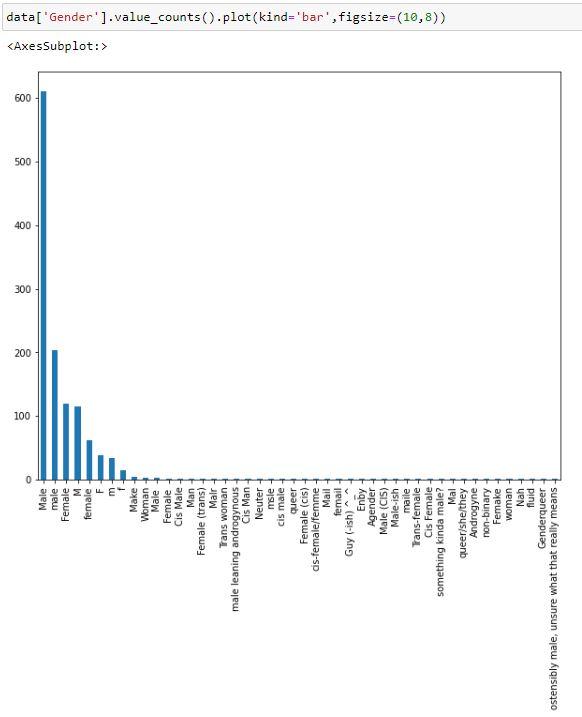
Let us now handle data that might have been entered wrongly. Consider the Age column of our dataset.



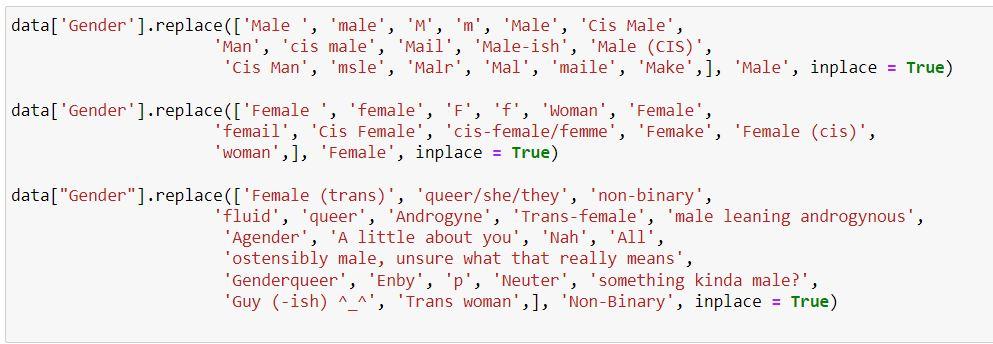
If we observe the Age column, it is seen that some impractical values have been entered.  So, let us remove the rows with ages greater than 60 and less than 18 using the code below.



Next, consider the Gender column of our dataset.



It is observed that different names are used for the same category of gender.  Let us group them into 3 major categories- Male, Female and Non-Binary using the .replace() function.



Milestone 3: Data Analysis and Visualization

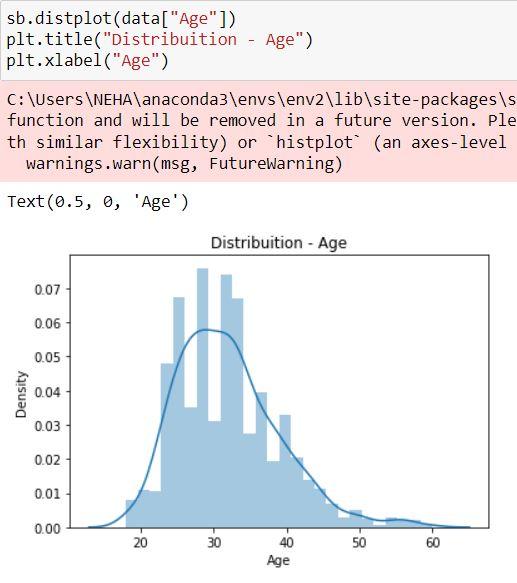
Let us understand the data properly with the help of some visualization techniques and some analyzing techniques.

Note: There are a lot of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

1. Activity 1: Univariate Analysis

In simple words, univariate analysis is understanding the data with a single feature.

* Seaborn package provides a function called distplot, which helps us to find the distribution of specific features in our dataset.  Let us observe the distribution of age.



1. Activity 2: Bivariate Analysis

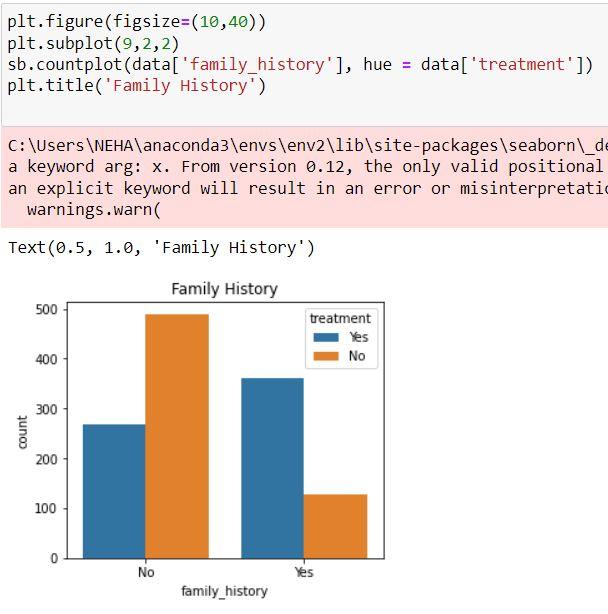
We use bivariate analysis to find the relation between two features.  Here we are visualising the relationship of various features with respect to treatment, which is our target variable.

* Employment type and treatment



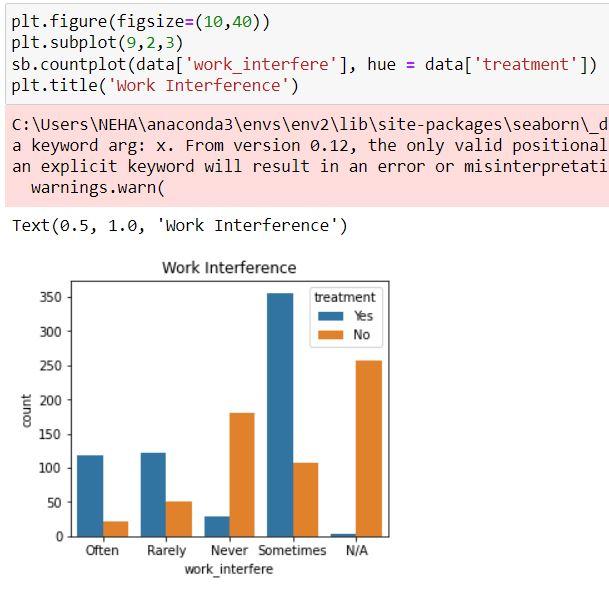
We observe that though there is a vast difference between people who are self-employed or not, the number of people who seek treatment in both the categories is more or less similar.

* Family history and treatment



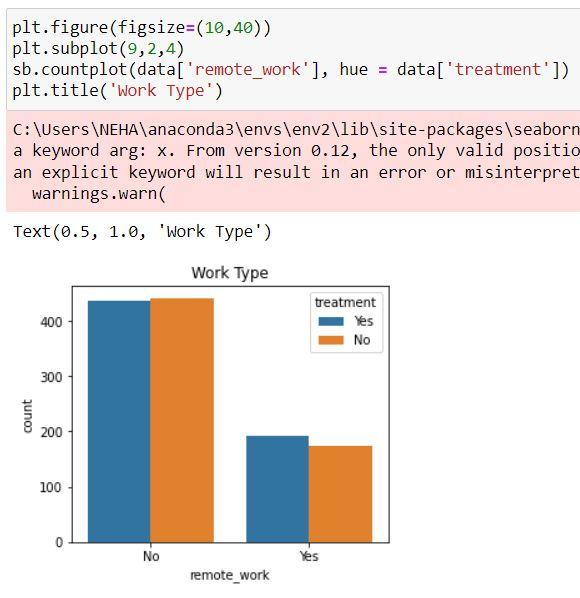
We observe that treatment is directly proportional to family history.  Hence this is an important factor.

* Work interference and treatment



We observe that the people who chose Sometimes were the largest who wanted to get treatment.  These group of people are the ones who are reluctant to choose either of the extreme categories.

* Work type and treatment



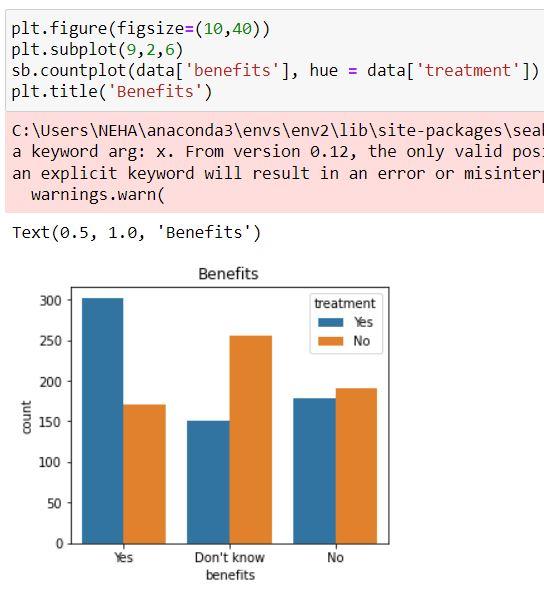
We observe that the number of people who seek treatment in both the categories is more or less similar and it does not affect our target variable.

* Company and treatment



We can conclude that irrespective of the field the company of the people falls in, mental health is a big issue.

* Benefits and treatment



We see that a large group among the people who wanted mental health benefits wanted to seek treatment and also a significant number of people who said No too, wanted to seek treatment.

* Care options and treatment



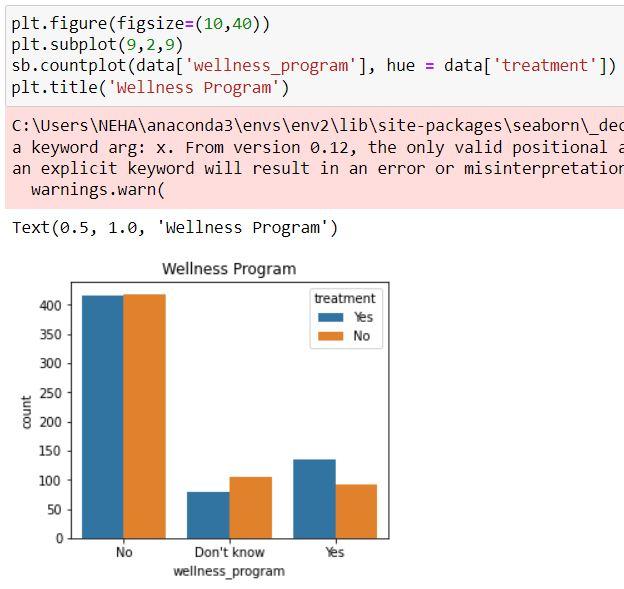
This graph is quite similar to the benefits column.

* Mental vs Physical health



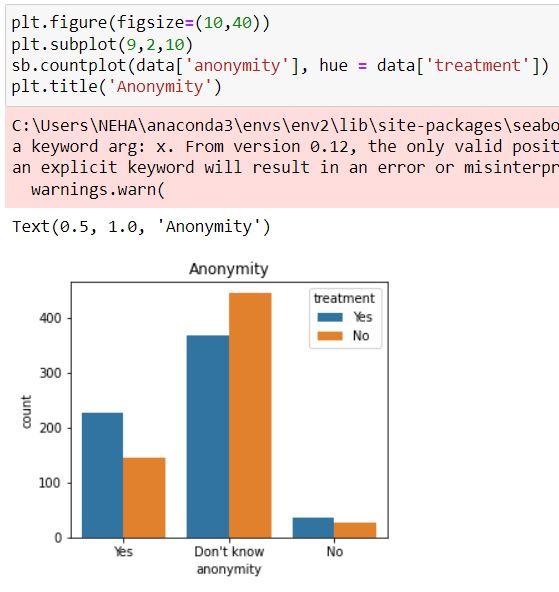
We observe that half of the people are not aware of the importance given to mental health as compared to physical health, whereas almost equal parts of the other halves answered Yes and No.

* Wellness program and treatment



We observe that almost half of the people who said No want to seek treatment, which means their company has to take steps in arranging for the same.

* Anonymity and treatment



We observe that most people either answered yes or they are not aware if their anonymity will be protected.

* Leave and treatment



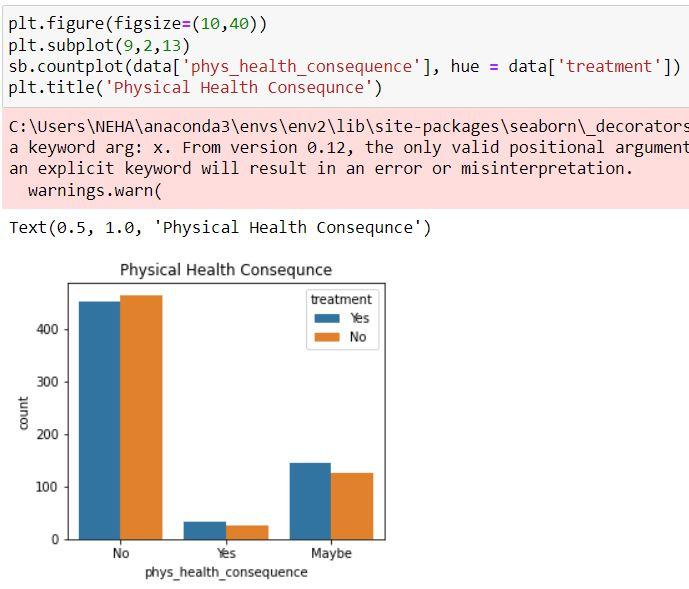
We see that around half of the total people don’t know how easy it is to get a leave due to a mental health condition and they are the ones who want to seek treatment the most.

* Mental health consequence and treatment



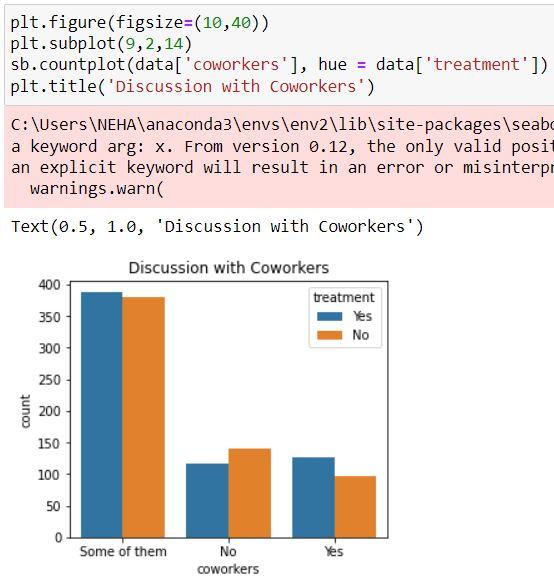
We observe that the majority answered either No or Maybe but around 1/3rd of the people answered yes and they want to seek treatment.

* Physical health consequence and treatment



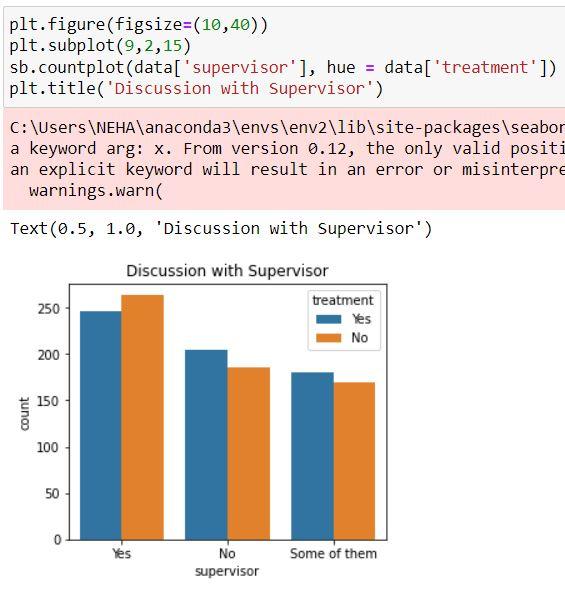
As completely opposed to the above results, a very small number of people answered yes to this question, which means that a major number of people do not face negative consequences by discussing their physical health conditions with their employers.

* Discussion with coworkers and treatment



We can observe that more than half of the people were willing to discuss their mental health problems with some or all of their coworkers.

* Discussion with supervisor and treatment



This graph again, is in contrast to the above one.  Though majority of people are comfortable with discussing their physical health problems with their supervisor, about 1/3rd of them isn’t comfortable with it.

* Mental health discussion during interview and treatment



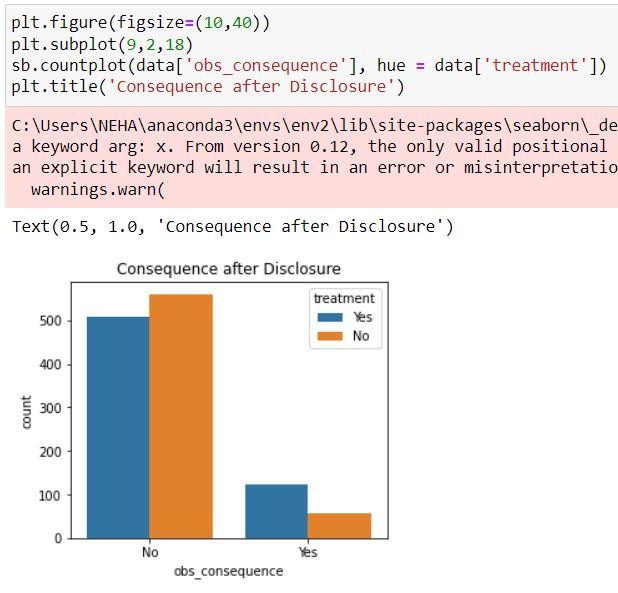
We observe that very few people are comfortable in bringing up their mental health issue in front of a potential employer.

* Physical health discussion during interview and treatment



Though this graph is similar to the above one with respect to not being comfortable in bringing up a physical health issue with a potential employer, a good number of people may be willing to or are willing to bring it up.

* The consequence after disclosure and treatment



We observe that majority of people have not faced any negative consequences after discussing their mental health conditions in their workplace.

1. Activity 3: Descriptive Analysis

Descriptive analysis is to study the basic statistical features of data. We can achieve it by using the .describe() function. With this describe function we can understand the unique, top, and frequent values of categorical features. Also, we can find mean, std, min, max, and percentile values of numerical features.



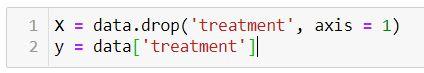
Milestone 4: Model Building

1. Activity 1: Handling Categorical Values

As we can see our dataset has categorical data.  Before training our model, we must convert the categorical data into a numeric form.

There are multiple encoding techniques to convert the categorical columns into numerical columns. For this project we will be using ordinal encoding for our features and label encoding for our target.

So, for that we first need to divide our data into features and target.



Here X contains our features and y contains our target.

The next step is to apply respective encoding techniques on features and target.  To apply ordinal encoding on our features, we will be using column transformer.



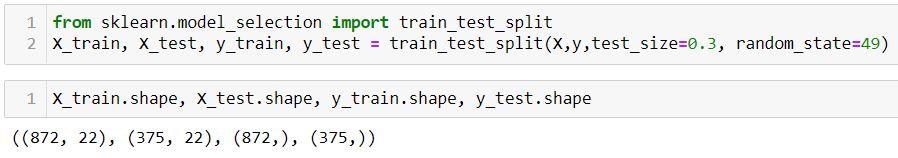
We need to save the column transformer instance so that we can use it during our model deployment.



After executing the above lines, ct will be saved in a file known as feature\_values.

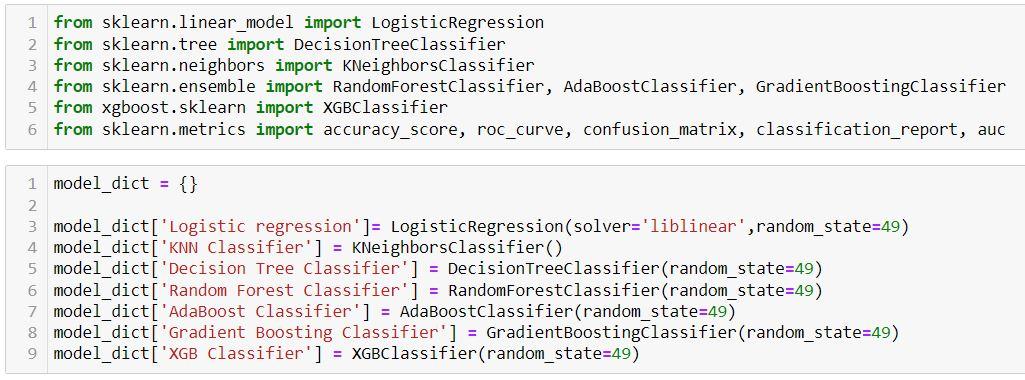
1. Activity 2: Splitting data into Train and Test

For splitting the data into train and test sets, we are using the train\_test\_split() function from sklearn. As parameters, we are passing X, y, test\_size, random\_state.

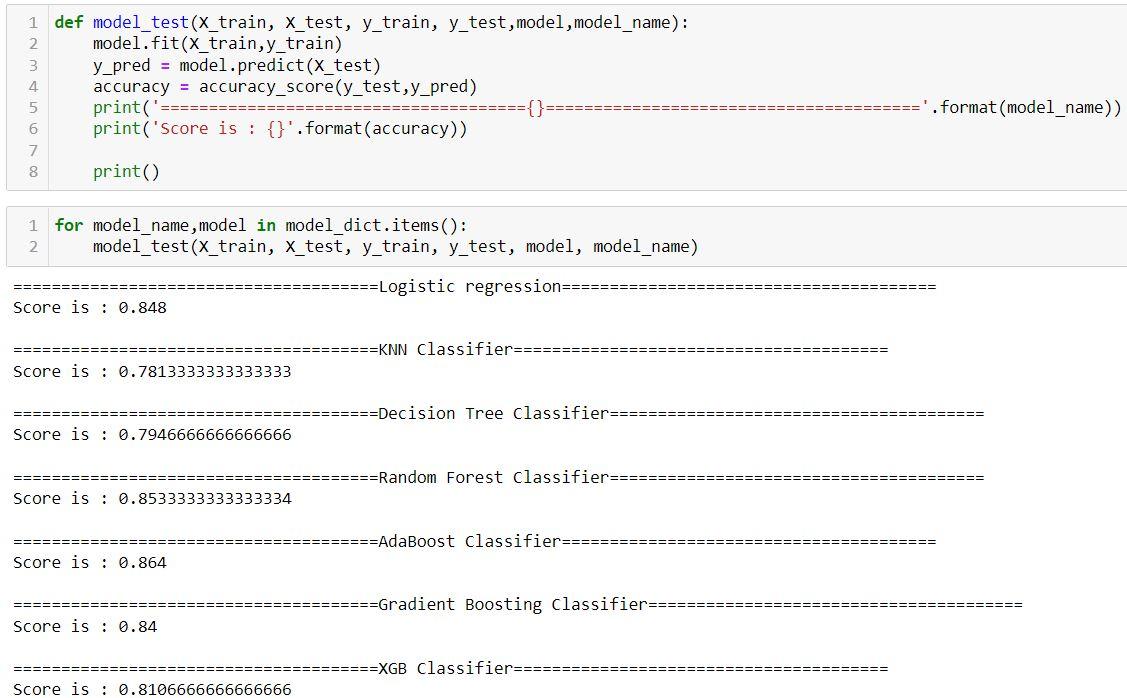


1. Activity 3: Comparing Accuracy of various models

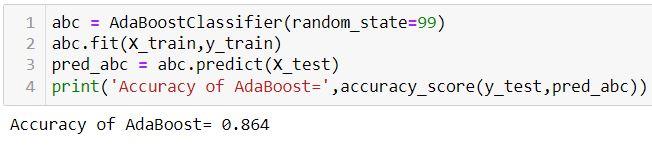
We will be considering multiple models to train our data and choose the one that performs the best.  So, we need to import the necessary libraries and create a dictionary of our models.



Next, we will define a function known as model\_test() that accepts 6 parameters - X\_train, X\_test, y\_train, y\_test, model, model\_name.  We will obtain y\_pred by using .predict() function and compute the accuracy score for every model by iterating through the dictionary.



From the above results, it is clear that AdaBoost Classifier provides the best accuracy.  So let us create a different variable to fit and make predictions using the model.



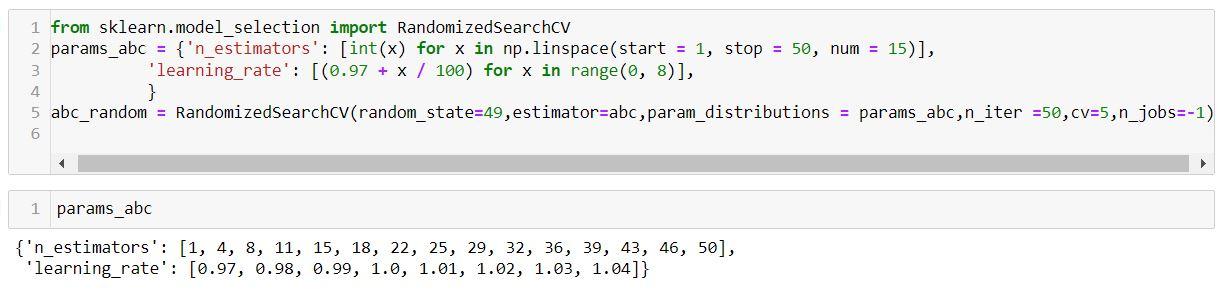
1. Activity 4: Hyperparameter tuning of selected model

To further improve the model performance, we are going to carry out a process known as hyperparameter tuning.  Every model will have multiple hyperparameters.  Please find the hyperparameters for AdaBoost Classifier from the below link:

[AdaBoostClassifier-docs](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html#:~:text=An%20AdaBoost%20%5B1%5D%20classifier%20is,focus%20more%20on%20difficult%20cases).

Of these, we will be tuning n\_estimators and learning\_rate.

For hyperparameter tuning, we can either use GridSearchCV or RandomizedSearchCV.  RandomizedSearchCV is more fast, efficient and preferred so we will be using it for our project.

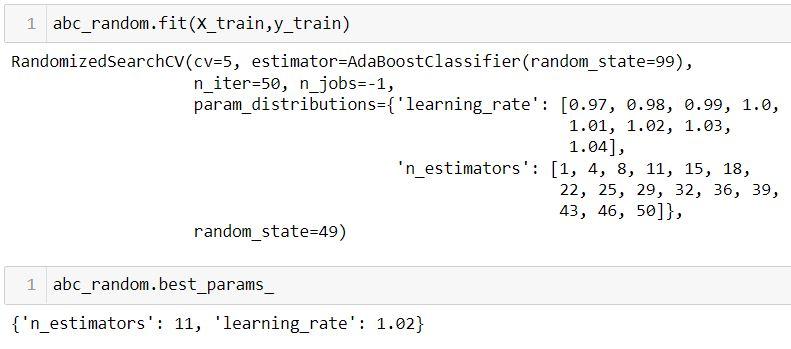


For n\_estimators, we are taking 15 equally spaced values from 1 to 50 and for learning rate, we are trying various values close to 1.

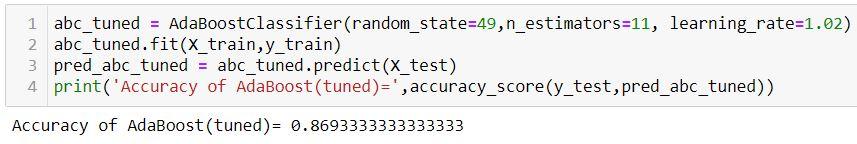
There are various parameters to be passed in RandomizedSearchCV.  To know what each parameter signifies, please refer to the below link:

[RandomizedSearchCV-docs](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html)

Next, let us fit our data and check what are the best hyperparameters using the .best\_params\_ attribute.



So, our model will perform the best if n\_estimators are equal to 11 and learning\_rate is equal to 1.02.  Let us add these values to train our model, make predictions and check accuracy.



We observe that the accuracy has increased approximately by 0.5%.  Though this is not a very great improvement, it is at least better than our previous model.

1. Activity 5: Evaluating Performance of Models

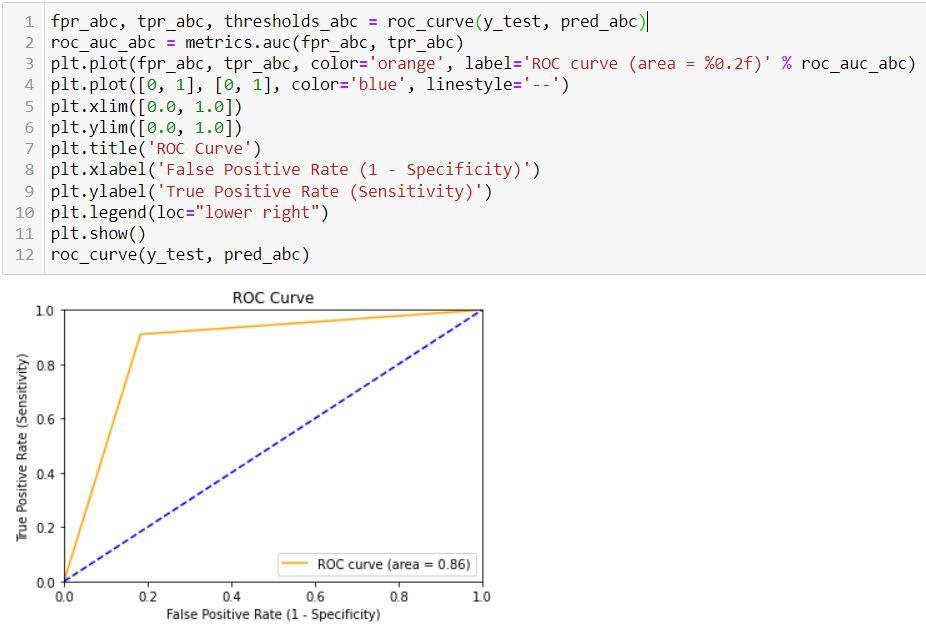
We will compare the confusion matrix, ROC curve and classification report for both models.

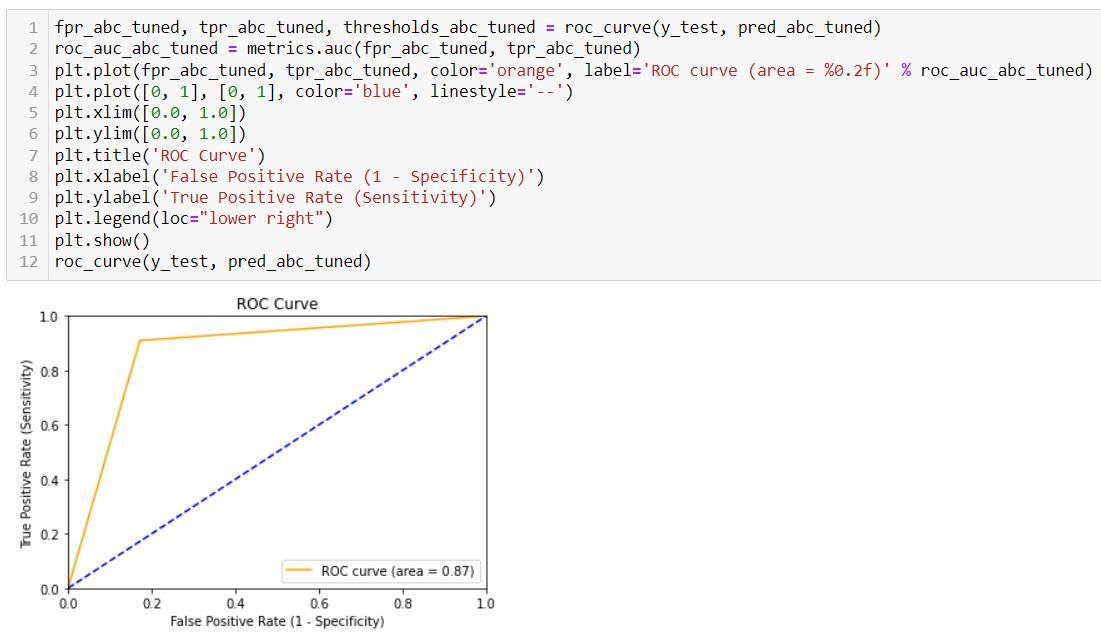
In order to obtain these, we will be using the confusion\_matrix(), roc\_curve() and classification\_report() functions from sklearn.metrics.

Confusion matrix:

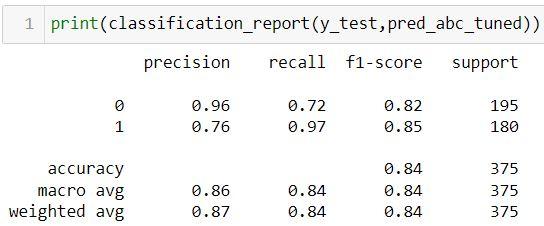
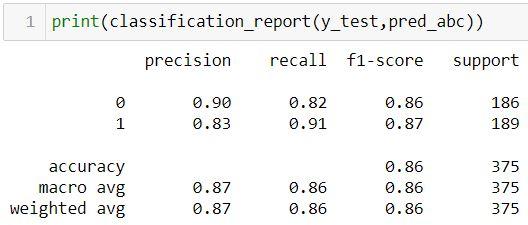


ROC Curve:





Classification report:



1. Saving the Model

The final step is saving our model.  We can do it by using pickle.dump().



Milestone 5: Application Building

In this section, we will be building a web application that is integrated into the model we built. A UI is provided for the uses where he has to enter the values for predictions. The entered values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server-side script

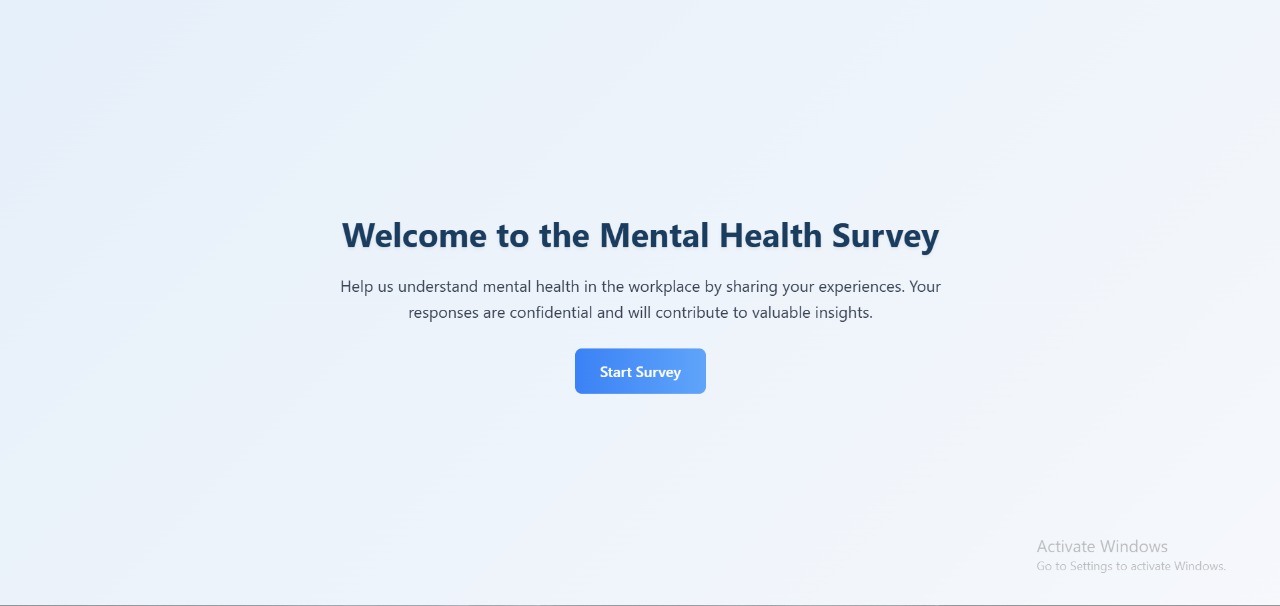
1. Activity 1: Building Html Pages

For this project create three HTML files namely

* home.html
* index.html
* output.html

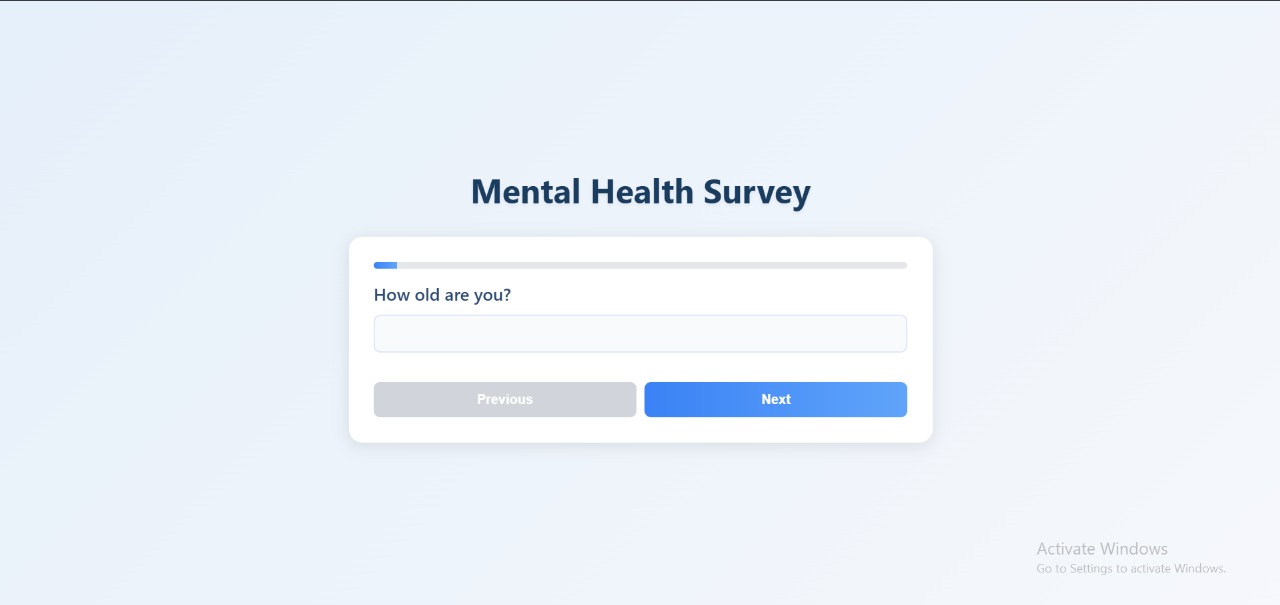
and save them in the templates folder.

Let’s see how our home.html page looks like:



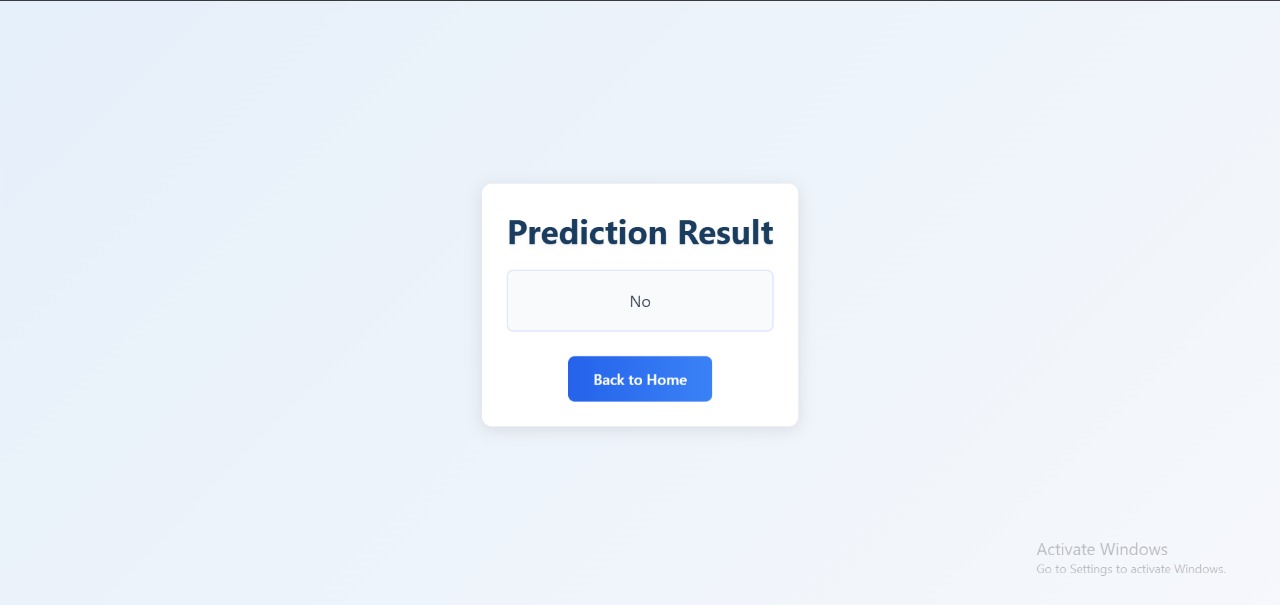
Now when you click on proceed button, you will get redirected to index.html

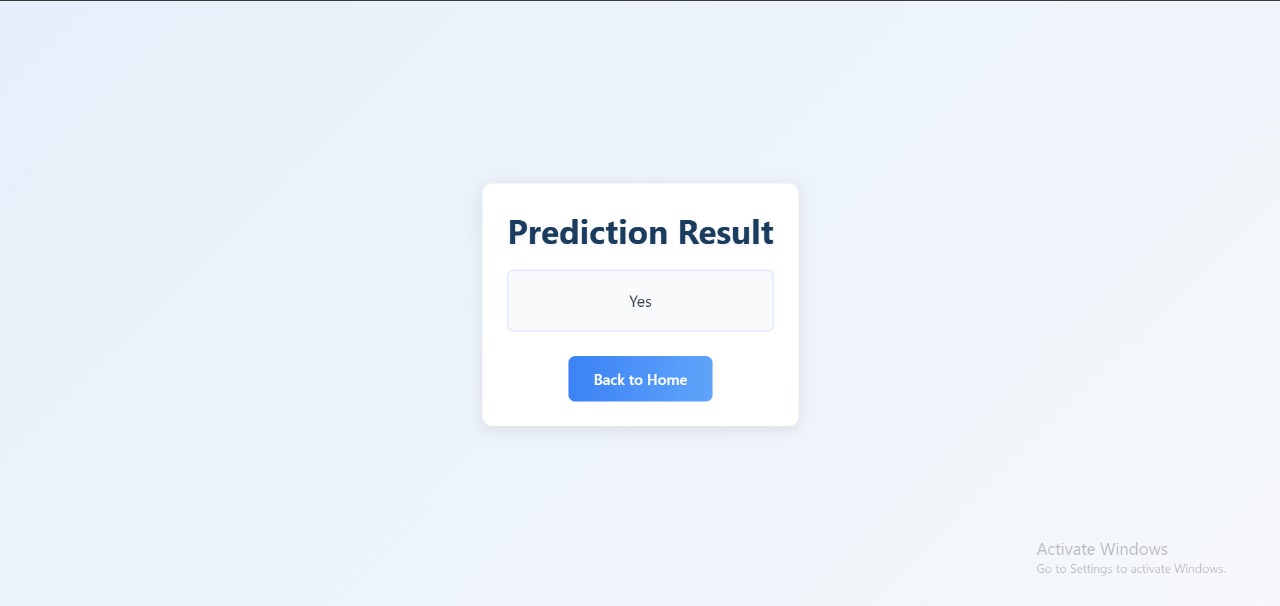
Let us look how our index.html page looks like:



Now when you click on predict button from left bottom corner you will get redirected to output.html

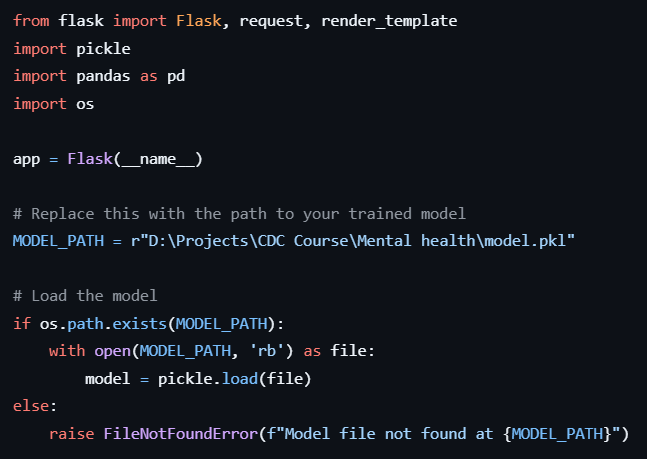
Let us look how our output.html page looks like:



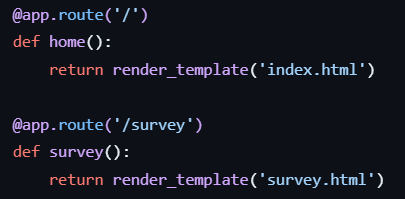


1. Activity 2: Build Python Code

Import the required libraries and load model and ct



Render home.html and index.html pages:



The values entered in can be retrieved using the POST Method.

Retrieves the value from UI:



Here we are routing our app to output() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the output.html page earlier.

Main Function:



1. Activity 3: Run the Model

* Open anaconda prompt from the start menu
* Navigate to the folder where your python script is.
* Now type “python app.py” command
* Navigate to the localhost where you can view your web page.
* Click on the proceed button, enter the inputs, click on the predict button, and see the result/prediction on the web.



Conclusion:

The Mental Health Prediction System using Machine Learning provides a data-driven approach to proactively identify individuals at risk of mental health disorders such as depression, anxiety, and stress. By leveraging survey responses and electronic health records, the system is able to uncover patterns that are often overlooked in traditional assessments.

Throughout the development cycle—from data collection and preprocessing to model training, evaluation, and deployment—we implemented robust methodologies to ensure accuracy, reliability, and ethical considerations. Multiple machine learning algorithms were tested and compared using evaluation metrics like accuracy, precision, recall, and F1-score, ensuring the selection of the best-performing model for real-world application.

The deployed model, integrated into a user-friendly web interface, enables healthcare professionals and researchers to make informed, timely decisions that could significantly improve patient outcomes. With increasing mental health challenges in society, this project demonstrates how technology can play a transformative role in enhancing public health support systems.

As a next step, further model refinement and real-world testing with diverse datasets can help improve generalizability and fairness. This initiative paves the way for scalable mental health screening tools that prioritize both accuracy and compassion.