# Artificial Intelligence & Machine Learning

Project Report

Semester-IV (Batch-2022)

# MENTAL HEALTH PREDICTION MODEL

**Supervised By:**

Dr. Kirandeep Singh

**Submitted By:**

Danish Awasthi 2210990247

Bisman Dhillon 2210990230

**Department of Computer Science and Engineering**
**Chitkara University Institute of Engineering & Technology,**
**Chitkara University, Punjab**

## Introduction

Utilizing Machine Learning for Predicting Mental Disorder using Logistic Regression Machine Learning Algorithm.

## Background

The prevalence of mental health disorders, notably depression and anxiety, underscores the urgent need for proactive interventions. Traditional diagnostic approaches often lack efficiency and timeliness, leading to delayed treatment and exacerbated symptoms. Leveraging the power of machine learning, particularly logistic regression, offers a promising solution. By analyzing diverse datasets, including social media activity and clinical records, this project seeks to develop a predictive model capable of identifying individuals at risk of mental health conditions. This approach holds potential for early detection and intervention, ultimately contributing to improved mental health outcomes and more effective healthcare delivery.

## Objectives:

The objective of this project is to develop a machine learning model that accurately predicts the risk of mental health conditions, such as depression and anxiety, using data from specified sources, enabling early detection and intervention to improve mental health outcomes and provide actionable insights for healthcare providers.

- Develop a machine learning model using logistic regression to predict mental health conditions like depression and anxiety.
- Utilize diverse data sources, including social media activity and clinical records, to train the model effectively.
- Achieve high accuracy and reliability in predicting individuals at risk of mental health disorders.
- Enable early detection and intervention, potentially improving mental health outcomes and reducing the burden on healthcare systems.
- Provide actionable insights for healthcare providers and support systems to enhance mental health care delivery.

## Significance

The significance of this project lies in its potential to revolutionize mental health care by integrating advanced technology with clinical practice. By harnessing machine learning techniques such as logistic regression, we aim to create a predictive model capable of identifying individuals at risk of mental health conditions with unprecedented accuracy and timeliness. This approach not only facilitates early intervention and support but also enhances the efficiency of healthcare resources by targeting interventions where they are most needed. Ultimately, this endeavor holds promise for improving mental health outcomes on a large scale, addressing a pressing societal challenge with innovative solutions.

Firstly, such a model holds immense potential for early detection and intervention, thereby mitigating the adverse effects of mental health disorders. By accurately identifying individuals at risk, healthcare professionals can provide timely support and interventions, preventing the escalation of symptoms and improving overall mental well-being.

Moreover, the utilization of machine learning in mental health prediction signifies a paradigm shift towards more personalized and proactive healthcare approaches. By analyzing a wide range of data sources, including social media activity and clinical records, the model can capture nuanced patterns and risk factors that may not be apparent through traditional assessment methods. This holistic approach enables healthcare providers to tailor interventions to individual needs, enhancing the efficacy of treatment and support strategies.

Additionally, the development of such a model contributes to the advancement of mental health research and innovation. By leveraging cutting-edge technology and data analytics, researchers can gain deeper insights into the complex interplay of factors influencing mental health outcomes. This knowledge can inform the development of targeted interventions, prevention strategies, and policy initiatives aimed at addressing the growing mental health crisis.

Overall, the significance of this project extends beyond individual-level benefits to encompass societal and systemic improvements in mental health care delivery, research, and policy formulation.

# Problem Definition and Requirements

## Problem Statement:

- Objective: This project aims to develop a machine learning model capable of predicting mental health conditions, such as depression and anxiety, using diverse data sources. By leveraging machine learning algorithms like logistic regression, the goal is to provide early detection and intervention strategies, ultimately improving mental health outcomes.

- Challenges:
 Integrating heterogeneous data sources, including social media activity, clinical records, and demographic information
 Ensuring the privacy and ethical handling of sensitive health data.
 Addressing potential biases in data collection and model predictions.

- Scope: The project will focus on developing and evaluating the predictive model using historical data. The model's effectiveness will be assessed through rigorous evaluation metrics, with the aim of providing actionable insights for healthcare providers and support systems.

## Software Requirements:

1. **Programming Language**: Python will serve as the primary programming language, leveraging its extensive libraries for machine learning and data analysis, including scikit-learn, pandas, and NumPy.

2. **Development Environment**: Anaconda or a similar Python distribution will be utilized to manage dependencies and create virtual environments, ensuring reproducibility and ease of setup.

3. **Machine Learning Libraries**: Scikit-learn will be the core library for building and evaluating machine learning models, while additional libraries such as TensorFlow or PyTorch may be explored for advanced modeling techniques, particularly for deep learning.

4. **Data Visualization Tools:** Matplotlib and Seaborn will be employed for data visualization to gain insights into the dataset distribution and model performance, aiding in model interpretation and validation.

5. **Text Editor or Integrated Development Environment (IDE):** Jupyter Notebooks or IDEs like PyCharm will facilitate coding, experimentation, and documentation, providing an interactive environment for model development.

6. **Version Control**: Git will be used for version control, enabling collaboration, tracking changes, and managing project history efficiently, with platforms like GitHub serving as repositories for code and project management.

## Hardware Requirements:

- **Processor**: A multi-core processor is recommended to handle data preprocessing and model training efficiently.

- **Memory (RAM):** At least 8 GB of RAM is required to accommodate large datasets and machine learning algorithms, ensuring smooth execution without memory constraints.

- **Storage:** Adequate storage space is needed for storing datasets, code files, and model artifacts. SSD storage is preferred for faster data access and model training.

- **Graphics Processing Unit (GPU) (Optional)**: While not mandatory, a GPU (NVIDIA GeForce or AMD Radeon) can significantly accelerate model training, especially for deep learning algorithms, enhancing computational performance.

- **Operating System:** The project can be executed on Windows, macOS, or Linux-based systems, ensuring compatibility across different platforms for seamless deployment and execution.

## Datasets:

The dataset comprises 1250 records, each containing essential features relevant to mental health prediction. These features include demographic information such as age and gender, as well as contextual factors such as country and state of residence. Additionally, the dataset captures employment-related variables such as self-employment status, family history of mental health issues, and whether individuals have received treatment. Moreover, it encompasses workplace-related factors like the interference of mental health with work, company size, and remote work arrangements. Furthermore, it includes information on organizational support, such as access to mental health benefits and care options. This comprehensive dataset provides a rich and diverse source of information, enabling the development of robust machine learning models for predicting mental health risks and guiding appropriate interventions.

## Features:

The dataset encompasses various features pertinent to mental health prediction, offering insights into different aspects of individuals' lives and work environments. Here's some information on the key features:

1. *Age*: Provides demographic information about individuals' age, which can indicate different stages of life and potential risk factors associated with mental health.

2. *Gender*: Captures gender identity, acknowledging the diverse spectrum of gender expressions and its potential impact on mental health experiences.

3. *Country and State*: Reflects geographical location, which may influence access to healthcare resources, cultural attitudes towards mental health, and workplace policies.

4. *Self-Employed*: Indicates whether individuals are self-employed, which could influence access to mental health benefits and support structures typically available in traditional employment settings.

5. *Family History*: Identifies whether individuals have a family history of mental health issues, which can contribute to genetic predispositions and risk assessment.

6. *Treatment*: Indicates whether individuals have received treatment for mental health conditions, providing insights into help-seeking behaviors and previous interventions.

7. *Work Interference*: Reflects the extent to which mental health issues interfere with individuals' work, highlighting potential challenges in maintaining productivity and well-being.

8. *Number of Employees*: Describes the size of the company or organization, which may impact the availability of mental health resources and support programs.

9. *Remote Work*: Indicates whether individuals have the option to work remotely, which can affect their access to in-person support and the work-life balance.

10. *Tech Company*: Identifies whether individuals work in the technology industry, where unique workplace cultures and stressors may influence mental health experiences.

11. *Benefits*: Reflects the availability of mental health benefits, such as insurance coverage for therapy and counseling services, which can facilitate access to care.

12. *Care Options*: Describes the range of care options available to individuals, including employer-provided resources and external support networks.

13. *Wellness Program*: Indicates whether the workplace offers wellness programs aimed at promoting mental health and overall well-being.

14. *Seek Help*: Reflects individuals' willingness to seek help for mental health concerns, which can vary based on cultural norms and organizational support.

15. *Anonymity*: Indicates whether individuals feel comfortable seeking help anonymously, which can impact their willingness to disclose mental health issues.

16. *Leave*: Reflects the availability and flexibility of leave policies for addressing mental health needs, including sick leave and mental health days.

17. *Consequences*: Describes the perceived consequences of disclosing mental health issues in the workplace, including potential stigma and discrimination.

## Proposed Design / Methodology

**• Data Preprocessing:**

- Handling Missing Values: Missing values will be imputed using appropriate techniques such as mean, median, or mode imputation.

- Feature Scaling: Features will be scaled using StandardScaler to ensure that no feature dominates due to its scale.

- Encoding Categorical Variables: Categorical variables will be encoded using techniques like one-hot encoding or label encoding for compatibility with machine learning algorithms.

-

**Feature Selection:**

Correlation Analysis: Pearson correlation coefficient will be computed to identify highly correlated features and remove redundant ones.

**• Model Development:**

1. Logistic Regression:

Logistic Regression will be implemented to predict the probability of diabetes based on the input features. Regularization techniques like L1 (Lasso) or L2 (Ridge) regularization may be applied to prevent overfitting.

**Model Evaluation:**

The models will be evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrix to assess their performance in predicting diabetes status. Cross-validation techniques like k-fold cross-validation will be employed to ensure robustness and avoid overfitting.

**Hyperparameter Tuning:**

Grid search or random search will be conducted to optimize hyperparameters for each algorithm, such as regularization strength in logistic regression.

## Model Interpretation:

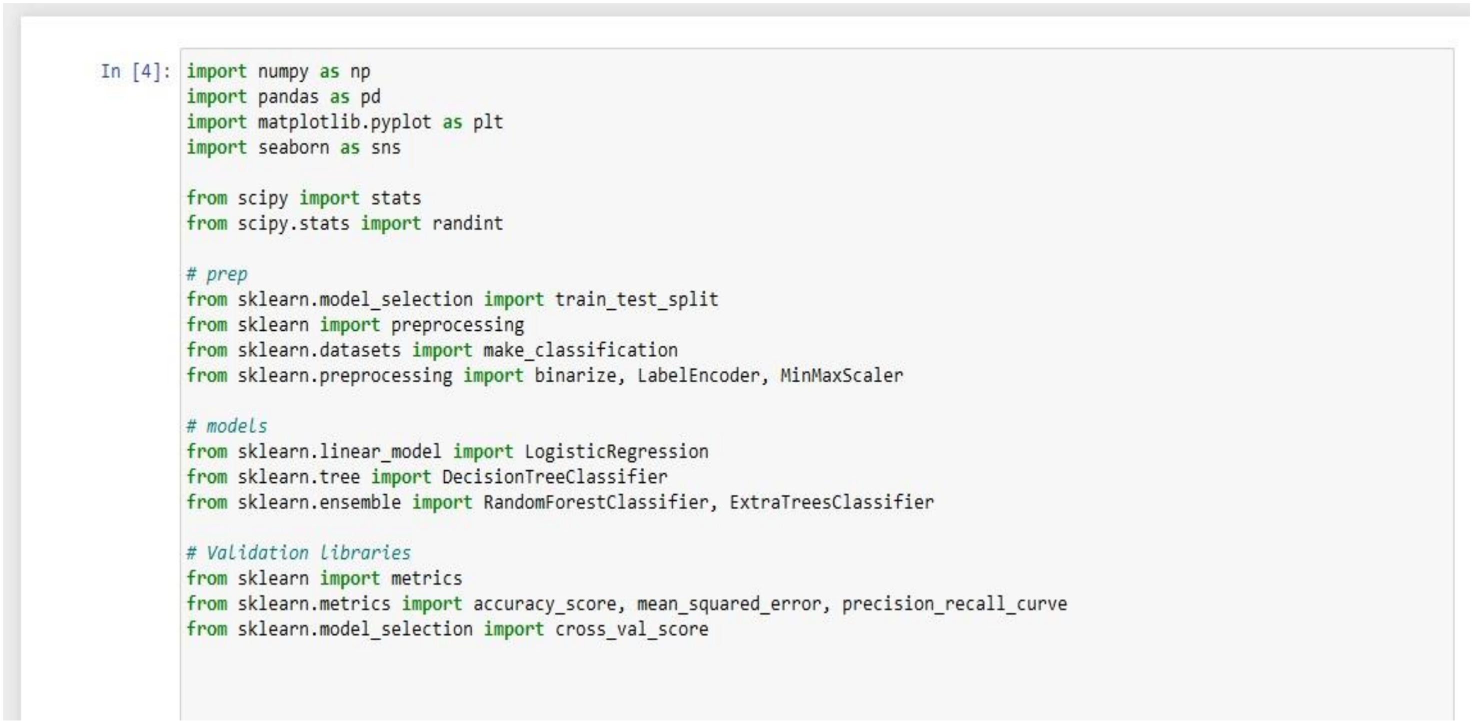
The trained models' decision boundaries and feature importance will be visualized to provide insights into their behavior and aid in model interpretation.

## Implementation:

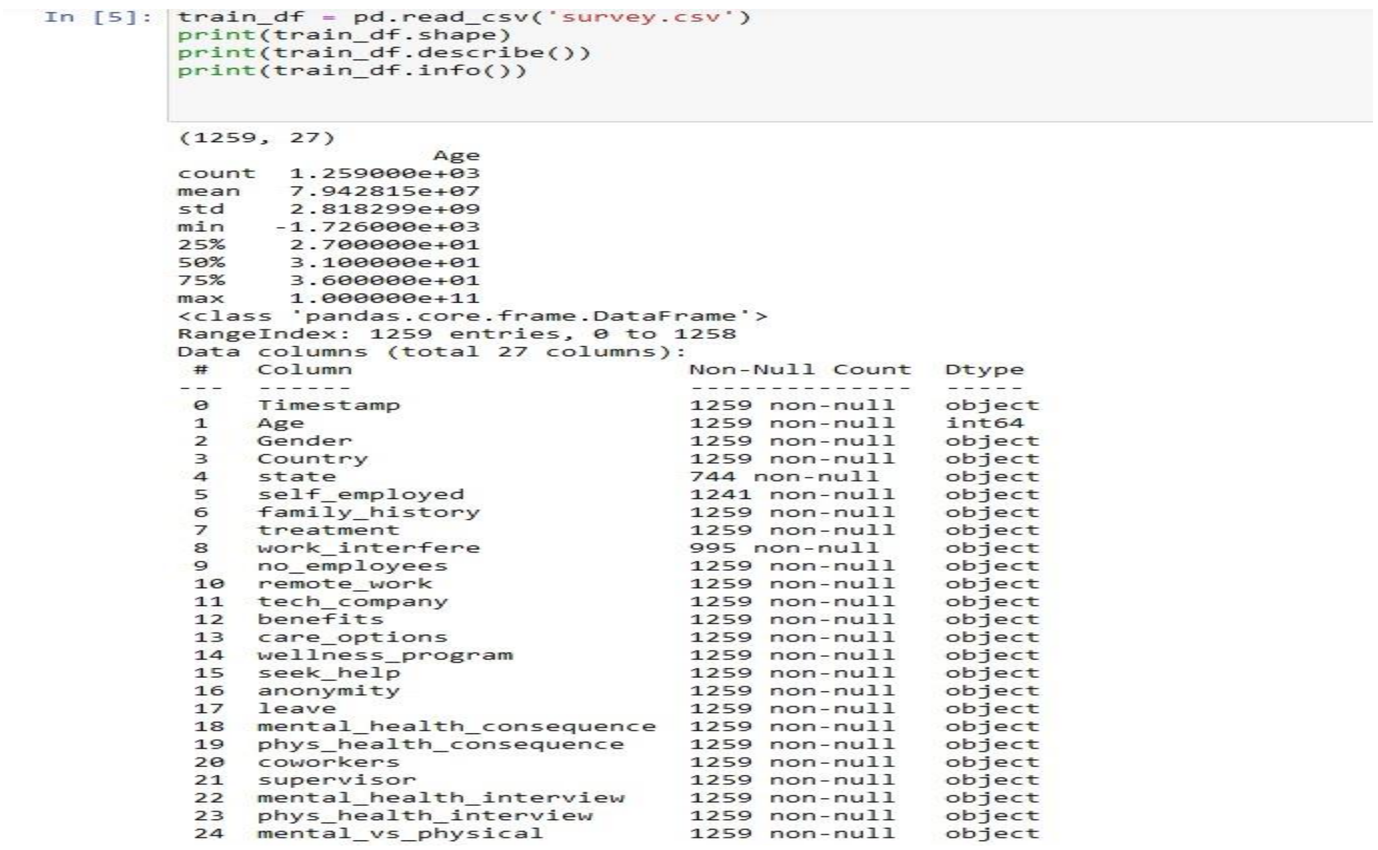
The proposed design and methodology will be implemented using Python programming language and relevant libraries such as scikit-learn, pandas, and matplotlib. Jupyter Notebooks or IDEs like PyCharm will facilitate code development, experimentation, and documentation. The provided imports such as StandardScaler, train_test_split, LogisticRegression, and evaluation metrics will be utilized in the implementation.

## RESULTS:

### Importing the dependencies/Libraries

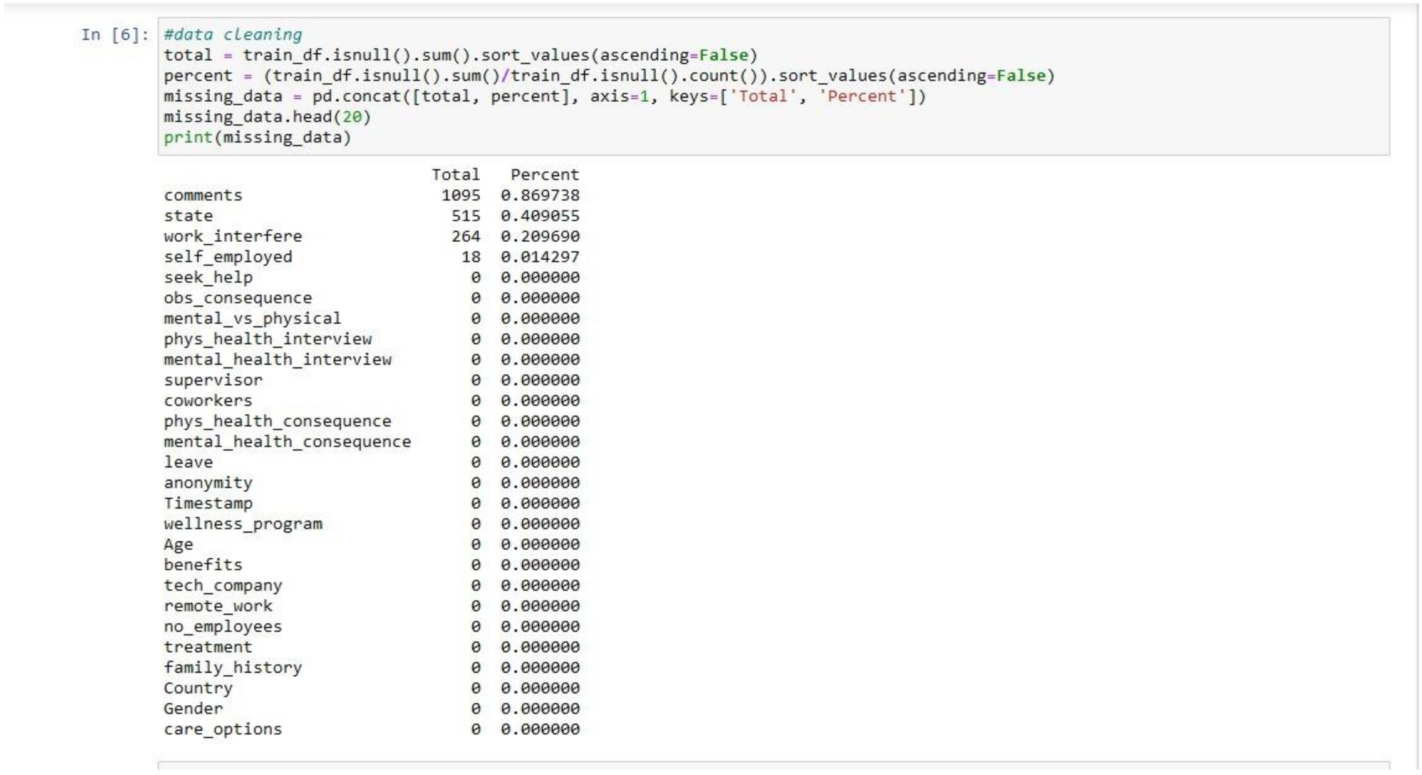


```
In [4]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns

        from scipy import stats
        from scipy.stats import randint

        # prep
        from sklearn.model_selection import train_test_split
        from sklearn import preprocessing
        from sklearn.datasets import make_classification
        from sklearn.preprocessing import binarize, LabelEncoder, MinMaxScaler

        # models
        from sklearn.linear_model import LogisticRegression
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier

        # Validation libraries
        from sklearn import metrics
        from sklearn.metrics import accuracy_score, mean_squared_error, precision_recall_curve
        from sklearn.model_selection import cross_val_score
```



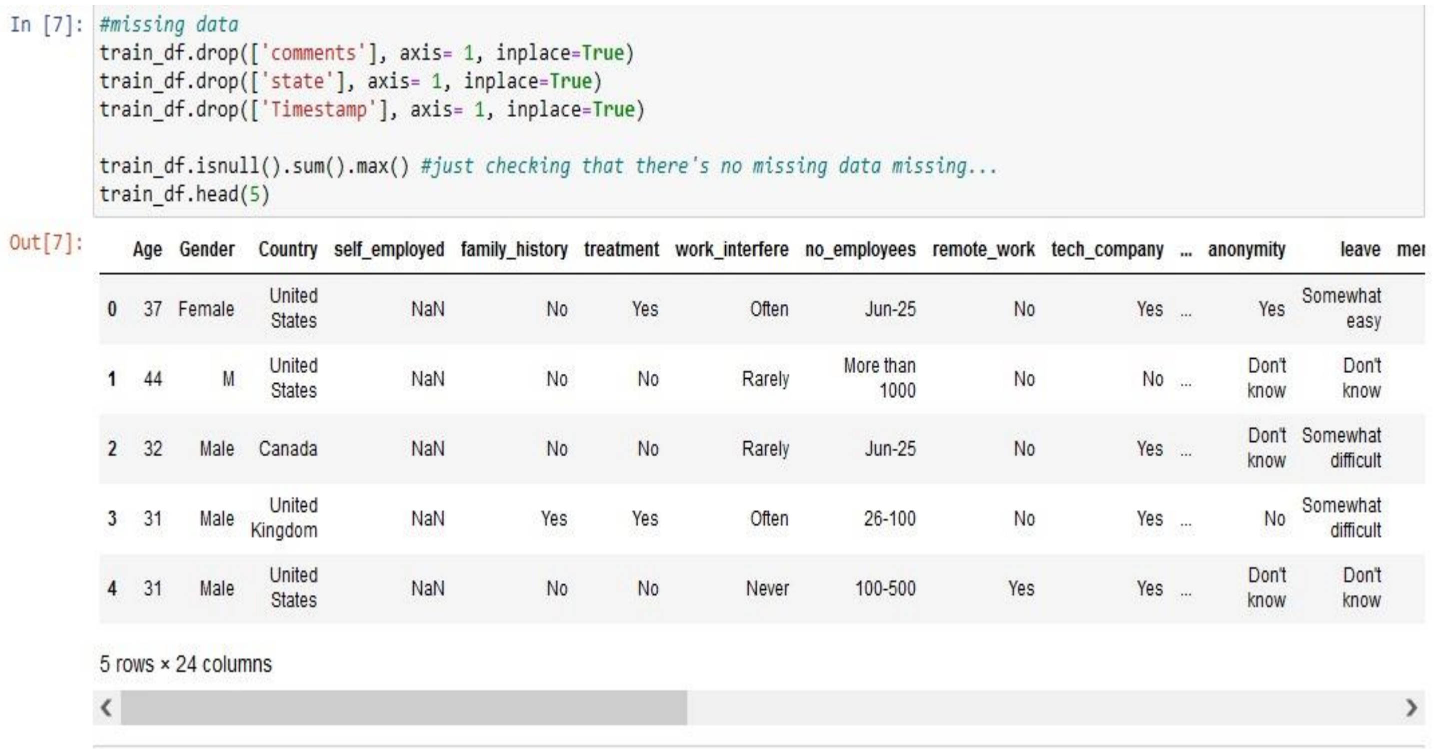
```
In [5]: train_df = pd.read_csv('survey.csv')
        print(train_df.shape)
        print(train_df.describe())
        print(train_df.info())

(1259, 27)
                Age
count  1.259000e+03
mean   7.942815e+07
std    2.818299e+09
min   -1.726000e+03
25%    2.700000e+01
50%    3.100000e+01
75%    3.600000e+01
max    1.000000e+11
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1259 entries, 0 to 1258
Data columns (total 27 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Timestamp                  1259 non-null   object
 1   Age                        1259 non-null   int64
 2   Gender                     1259 non-null   object
 3   Country                    1259 non-null   object
 4   state                      744 non-null    object
 5   self_employed              1241 non-null   object
 6   family_history             1259 non-null   object
 7   treatment                  1259 non-null   object
 8   work_interfere             995 non-null    object
 9   no_employees               1259 non-null   object
 10  remote_work                1259 non-null   object
 11  tech_company               1259 non-null   object
 12  benefits                   1259 non-null   object
 13  care_options               1259 non-null   object
 14  wellness_program           1259 non-null   object
 15  seek_help                  1259 non-null   object
 16  anonymity                  1259 non-null   object
 17  leave                      1259 non-null   object
 18  mental_health_consequence  1259 non-null   object
 19  phys_health_consequence    1259 non-null   object
 20  coworkers                  1259 non-null   object
 21  supervisor                 1259 non-null   object
 22  mental_health_interview    1259 non-null   object
 23  phys_health_interview      1259 non-null   object
 24  mental_vs_physical         1259 non-null   object
```

## Data Cleaning



```
In [6]: #data cleaning
        total = train_df.isnull().sum().sort_values(ascending=False)
        percent = (train_df.isnull().sum()/train_df.isnull().count()).sort_values(ascending=False)
        missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
        missing_data.head(20)
        print(missing_data)

                                 Total   Percent
        comments                  1095  0.869738
        state                      515  0.409055
        work_interfere             264  0.209690
        self_employed               18  0.014297
        seek_help                    0  0.000000
        obs_consequence              0  0.000000
        mental_vs_physical           0  0.000000
        phys_health_interview        0  0.000000
        mental_health_interview      0  0.000000
        supervisor                   0  0.000000
        coworkers                    0  0.000000
        phys_health_consequence      0  0.000000
        mental_health_consequence    0  0.000000
        leave                        0  0.000000
        anonymity                    0  0.000000
        Timestamp                    0  0.000000
        wellness_program             0  0.000000
        Age                          0  0.000000
        benefits                     0  0.000000
        tech_company                 0  0.000000
        remote_work                  0  0.000000
        no_employees                 0  0.000000
        treatment                    0  0.000000
        family_history               0  0.000000
        Country                      0  0.000000
        Gender                       0  0.000000
        care_options                 0  0.000000
```

## Missing Data



```
In [7]: #missing data
        train_df.drop(['comments'], axis= 1, inplace=True)
        train_df.drop(['state'], axis= 1, inplace=True)
        train_df.drop(['Timestamp'], axis= 1, inplace=True)

        train_df.isnull().sum().max() #just checking that there's no missing data missing...
        train_df.head(5)
```

Out[7]:

| | Age | Gender | Country | self_employed | family_history | treatment | work_interfere | no_employees | remote_work | tech_company | ... | anonymity | leave | mer |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 37 | Female | United States | NaN | No | Yes | Often | Jun-25 | No | Yes | ... | Yes | Somewhat easy | |
| 1 | 44 | M | United States | NaN | No | No | Rarely | More than 1000 | No | No | ... | Don't know | Don't know | |
| 2 | 32 | Male | Canada | NaN | No | No | Rarely | Jun-25 | No | Yes | ... | Don't know | Somewhat difficult | |
| 3 | 31 | Male | United Kingdom | NaN | Yes | Yes | Often | 26-100 | No | Yes | ... | No | Somewhat difficult | |
| 4 | 31 | Male | United States | NaN | No | No | Never | 100-500 | Yes | Yes | ... | Don't know | Don't know | |

5 rows × 24 columns

```
In [10]: #Clean 'Gender'
         gender = train_df['Gender'].unique()
         print(gender)
```

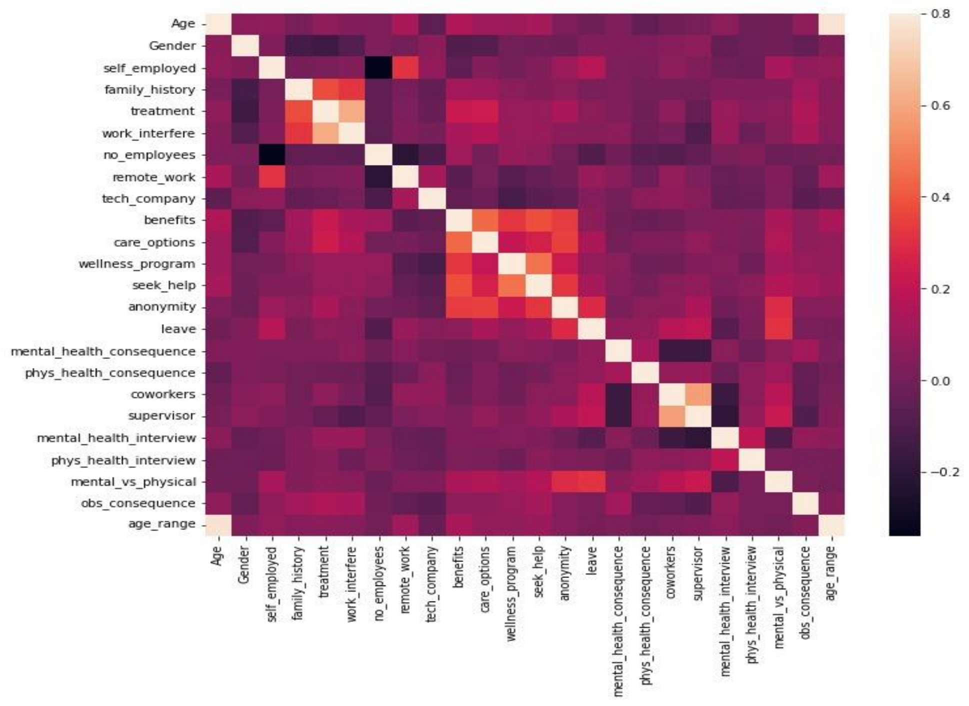
```
['Female' 'M' 'Male' 'male' 'female' 'm' 'Male-ish' 'maile' 'Trans-female'
 'Cis Female' 'F' 'something kinda male?' 'Cis Male' 'Woman' 'f' 'Mal'
 'Male (CIS)' 'queer/she/they' 'non-binary' 'Femake' 'woman' 'Make' 'Nah'
 'All' 'Enby' 'fluid' 'Genderqueer' 'Female ' 'Androgyne' 'Agender'
 'cis-female/femme' 'Guy (-ish) ^_^' 'male leaning androgynous' 'Male '
 'Man' 'Trans woman' 'msle' 'Neuter' 'Female (trans)' 'queer'
 'Female (cis)' 'Mail' 'cis male' 'A little about you' 'Malr' 'p' 'femail'
 'Cis Man' 'ostensibly male, unsure what that really means']
```

```
In [11]: #gender grouping

         male_str = ["male", "m", "male-ish", "maile", "mal", "male (cis)", "make", "male ", "man","msle", "mail", "malr","cis man",
         trans_str = ["trans-female", "something kinda male?", "queer/she/they", "non-binary","nah", "all", "enby", "fluid", "genderc
         female_str = ["cis female", "f", "female", "woman",  "femake", "female ","cis-female/femme", "female (cis)", "femail"]

         for (row, col) in train_df.iterrows():

             if str.lower(col.Gender) in male_str:
                 train_df['Gender'].replace(to_replace=col.Gender, value='male', inplace=True)

             if str.lower(col.Gender) in female_str:
                 train_df['Gender'].replace(to_replace=col.Gender, value='female', inplace=True)

             if str.lower(col.Gender) in trans_str:
                 train_df['Gender'].replace(to_replace=col.Gender, value='trans', inplace=True)

         #Get rid of bullshit
         stk_list = ['A little about you', 'p']
         train_df = train_df[~train_df['Gender'].isin(stk_list)]

         print(train_df['Gender'].unique())
```

```
['female' 'male' 'trans']
```

## Heat Map

```
In [17]: corrmat = train_df.corr()
         f, ax = plt.subplots(figsize=(12, 9))
         sns.heatmap(corrmat, vmax=.8, square=True);
         plt.show()
```

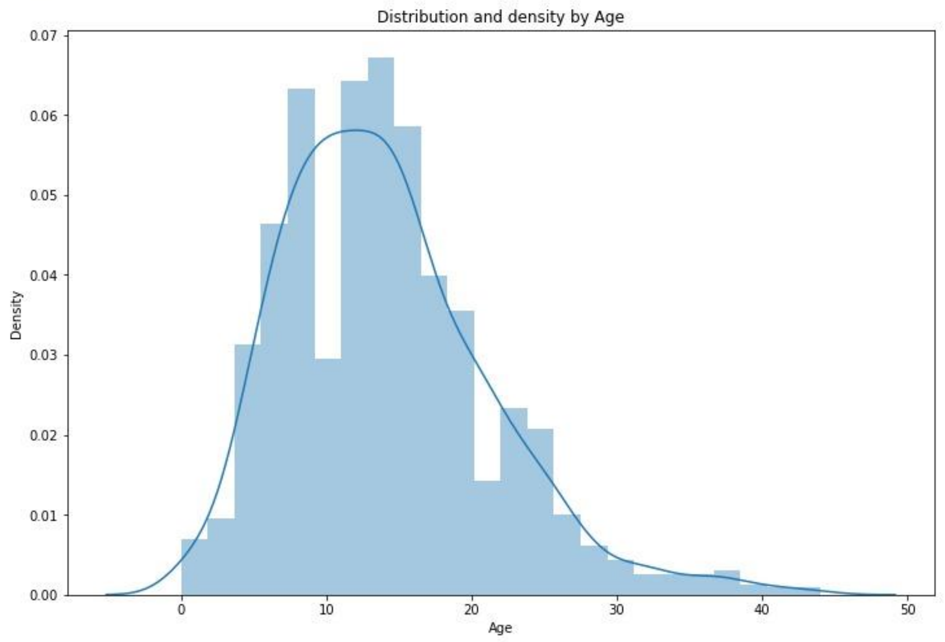


## Histogram

```
In [18]: # Distribution and density by Age
         plt.figure(figsize=(12,8))
         sns.distplot(train_df["Age"], bins=24)
         plt.title("Distribution and density by Age")
         plt.xlabel("Age")
```

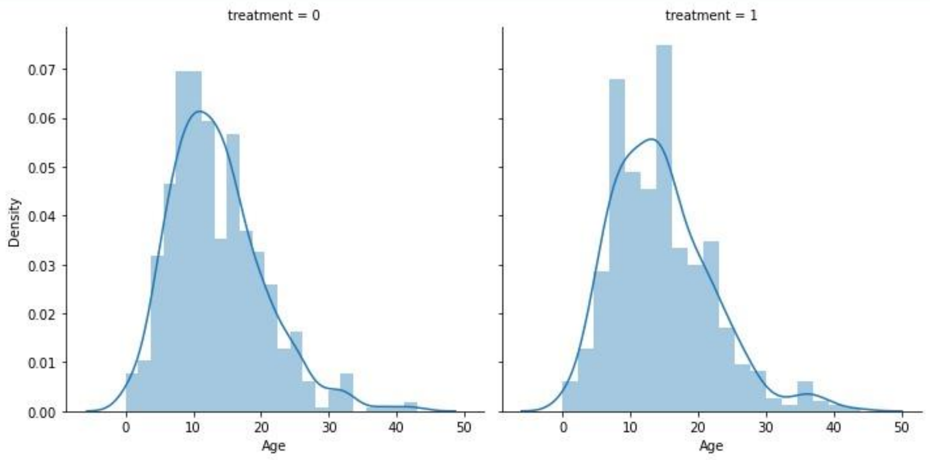
```
C:\Users\BISMAN DHILLON\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecat
ed function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
Out[18]: Text(0.5, 0, 'Age')
```

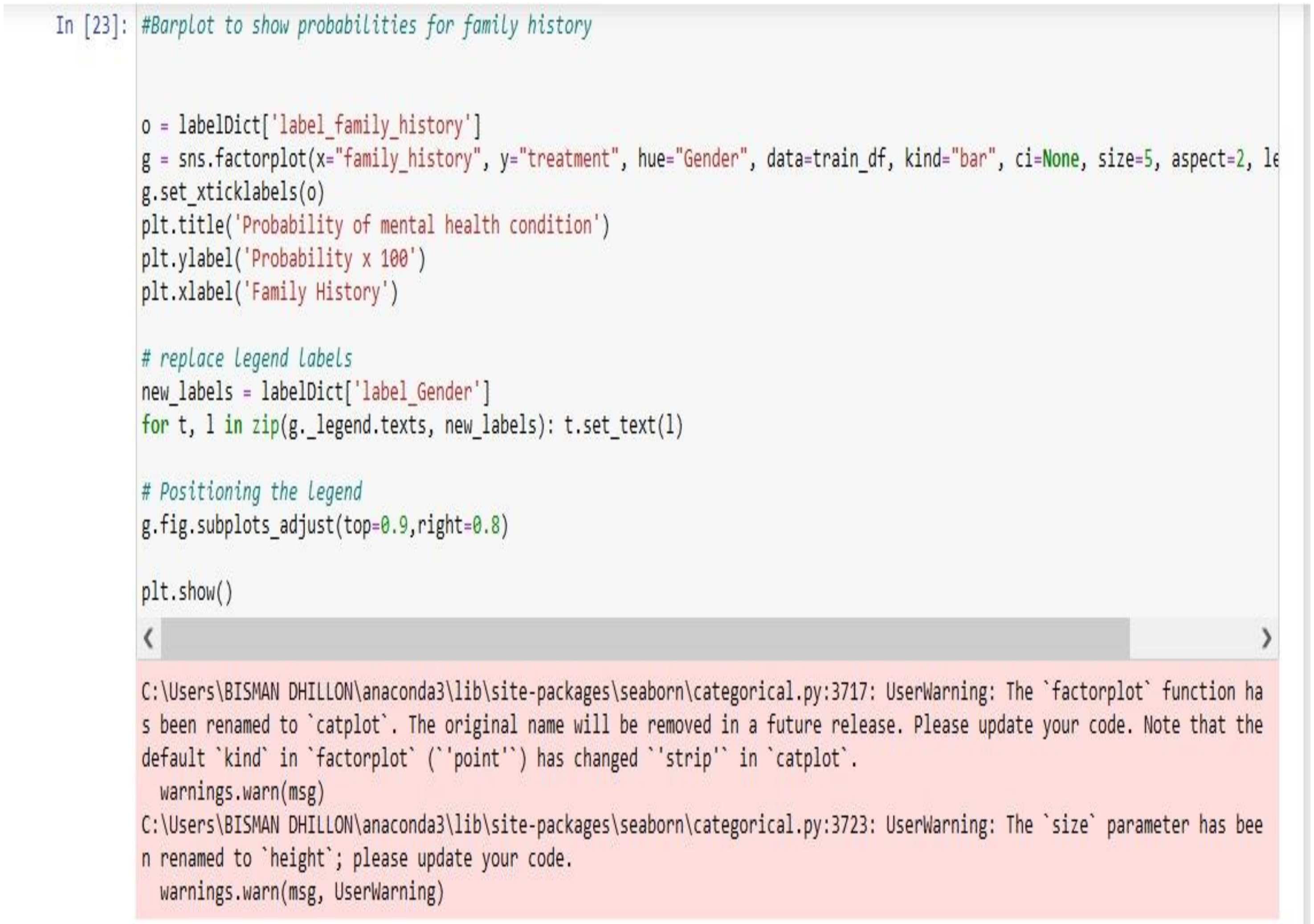


```
In [19]: #seperate by treatment
         g = sns.FacetGrid(train_df, col='treatment', size=5)
         g = g.map(sns.distplot, "Age")
```

```
C:\Users\BISMAN DHILLON\anaconda3\lib\site-packages\seaborn\axisgrid.py:337: UserWarning: The `size` parameter has been re
named to `height`; please update your code.
  warnings.warn(msg, UserWarning)
C:\Users\BISMAN DHILLON\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecat
ed function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\BISMAN DHILLON\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecat
ed function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

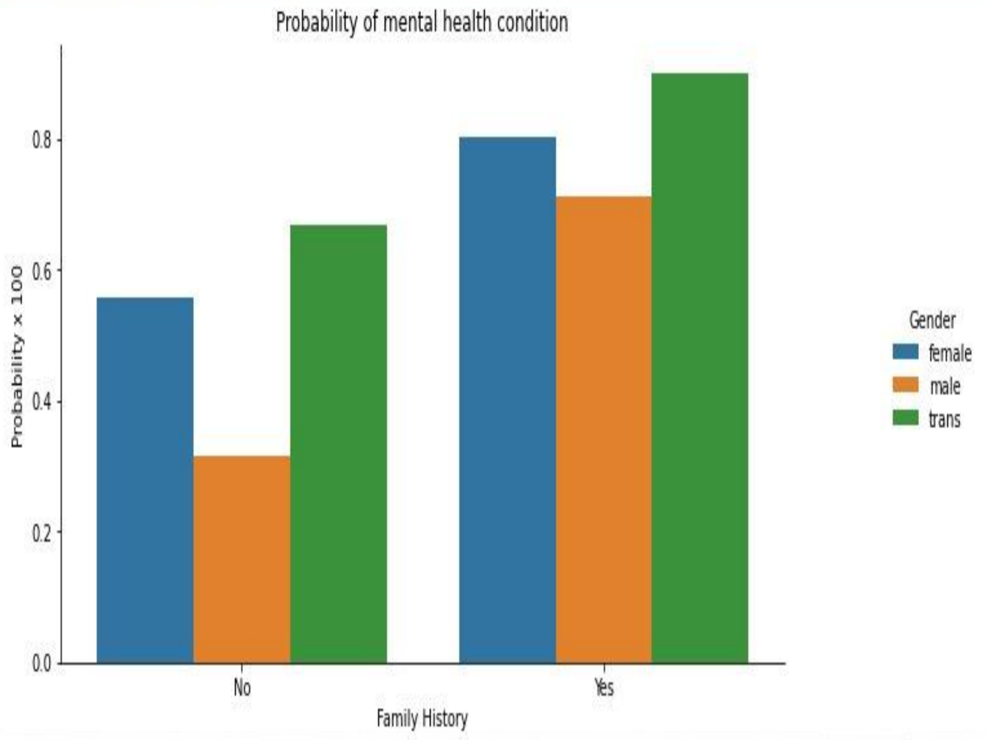


## Bar Plot



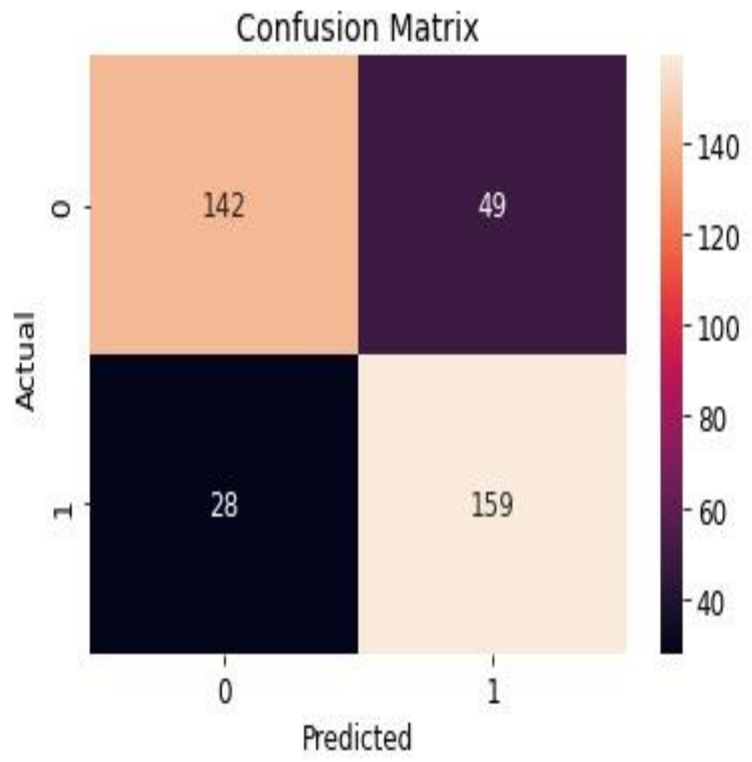
```
In [23]: #Barplot to show probabilities for family history


         o = labelDict['label_family_history']
         g = sns.factorplot(x="family_history", y="treatment", hue="Gender", data=train_df, kind="bar", ci=None, size=5, aspect=2, le
         g.set_xticklabels(o)
         plt.title('Probability of mental health condition')
         plt.ylabel('Probability x 100')
         plt.xlabel('Family History')

         # replace legend labels
         new_labels = labelDict['label_Gender']
         for t, l in zip(g._legend.texts, new_labels): t.set_text(l)

         # Positioning the legend
         g.fig.subplots_adjust(top=0.9,right=0.8)

         plt.show()
```

```
C:\Users\BISMAN DHILLON\anaconda3\lib\site-packages\seaborn\categorical.py:3717: UserWarning: The `factorplot` function ha
s been renamed to `catplot`. The original name will be removed in a future release. Please update your code. Note that the
default `kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.
  warnings.warn(msg)
C:\Users\BISMAN DHILLON\anaconda3\lib\site-packages\seaborn\categorical.py:3723: UserWarning: The `size` parameter has bee
n renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



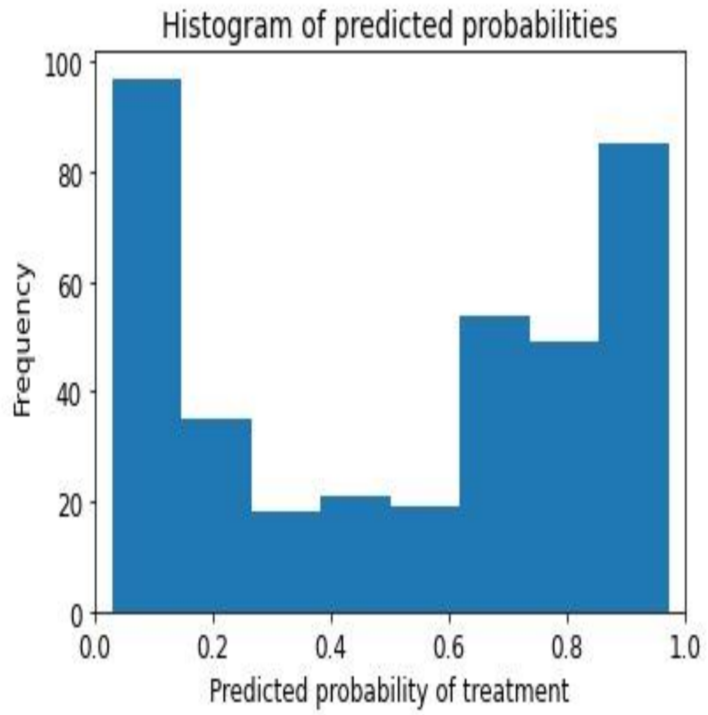
```
In [38]: logisticRegression()
```

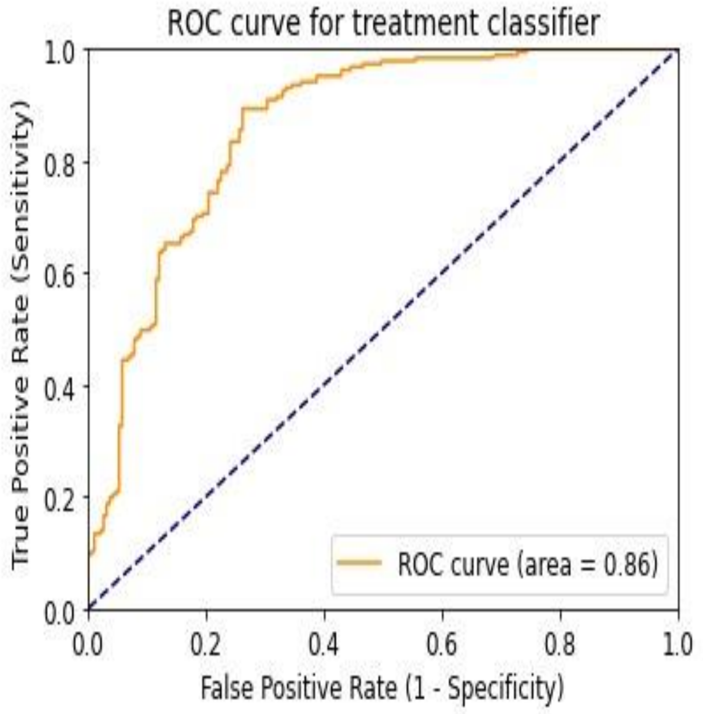
```
Accuracy: 0.7962962962962963
Null accuracy:
 0    191
1    187
Name: treatment, dtype: int64
Percentage of ones: 0.4947089947089947
Percentage of zeros: 0.5052910052910053
True: [0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 1 0 1 0 0 0 1 1 0 0]
Pred: [1 0 0 0 1 1 0 1 0 1 0 1 1 0 1 1 1 1 0 0 0 0 1 0 0]
```



```
Classification Accuracy: 0.7962962962962963
Classification Error: 0.20370370370370372
False Positive Rate: 0.25654450261780104
Precision: 0.7644230769230769
AUC Score: 0.7968614385306716
Cross-validated AUC: 0.8753623882722146
```

```
[0.92500057 ]]
```





```
[[142  49]
 [ 28 159]]
```

## References:

https://www.youtube.com/

https://www.geeksforgeeks.org/python-for-machine-learning/?ref=shm