A

Major Project

On

MENTAL HEALTH PREDICTION USING MACHINE LEARNING

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

A. Mahesh(197R5A0507)

Nusrath Jahan(197R5A0506)

B. Swathi(197R5A0508)

Under the Guidance of

K. PRAVEEN KUMAR

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CMR TECHNICAL CAMPUS
UGC AUTONOMOUS
(Accredited by NAAC, NIRF, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)
Recognized Under Section 2(f) & 12(B) of the UGC Act.1956, NIRF Rank Band 201-250
Kandlakoya (V), Medchal Road, Hyderabad-501401.
2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**MENTAL HEALTH PREDICTION USING MACHINE LEARNING** being submitted by **A. MAHESH (197R5A0507), NUSRATH JAHAN (197R5A0506) & B. SWATHI (197R5A0512)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

K. Praveen Kumar
Associate Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on

ACKNOWLEDGEMENT

A. MAHESH (197R5A0507)

NUSRATH JAHAN (197R5A0506)

B. SWATHI (197R5A0508)

ABSTRACT

The emotional, psychological and social welfare of a person is revealed by their mental health. It influences how an individual will think, feel or handle a situation. Positive mental health helps an individual to work productively and achieve their full potential. At each point in life, mental health is vital, from childhood to adulthood. Numerous factors contribute to mental health issues which lead to mental illness like stress, social anxiety, depression, obsessive compulsive disorder, drug addiction, workplace issues and personality disorders. The onset of mental illness should be determined without flaws for maintaining an appropriate life balance. We have collected data from online available datasets. The data has been label encoded for better prediction. The data is being subject to various machine learning techniques to obtain labels. These classified labels will then be used to build a model to predict the mental health of an individual. Our target population is in the working class i.e people above the age of 18. Once the model is built, it will be integrated to a website so that it can predict the outcome as per the details provided by the user.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

Mental Health Prediction using machine learning is used to help a person of his/her mental condition in a positive state or negative state. Positive state defines that the person is normal state that he can handle his things by his/her own. Negative state defines the person is not in a normal way that he/she is in some stress or financial issues or health issues these can define the person is in a Positive state or Negative state If he/she in negative state that they require treatment to solve their problem. In this we are using some database to define their state and whether they require treatment or not.

1.2 PROJECT PURPOSE

The Purpose of doing this project is that person can define their state and he/she can has to make changes in their life to live in a positive manner. In this project we need to fill the necessary data to require to show their mental health condition.

1.3 PROJECT FEATURES

The main features of this project are algorithms which we are using in this project that calculate the person is in positive state or negative state. We take dataset and calculate the result of that dataset and predict the person's mental state according to the dataset we provided. The other feature includes the predicting the each and every single dataset and calculating them we can see the results in the graph. The graph shows the predicting of each dataset and also each countries mental health conditions showing results in graph.

2. SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. the System is studied to the minute details and analysed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified.

Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

The Mental Health Prediction of a person which they cannot decide by themselves that they should depend upon the others like hospital or doctor to test weather their mental health but that requires some amount to consult the doctor or to check in hospital that both requires amount and also requires time to test and waiting for the results from the doctor or hospital. So that's why we are having some technology like Machine Learning that we can the check results of a person providing some information about their mental health condition. That's make a less time check the results by using some algorithms in machine learning

2.2 EXISTING SYSTEM

The existing systems gives the immense understanding of the mental health analysis amongst different target groups using different technology. The classification models performance can be improved using deep learning methods such as recurrent neural networks.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

• The prediction of mental Health treatment shows the results of the state of a person in which positive or negative condition.

• A person is having a negative kind of condition is always being in bad state and he is supposed to in serious manner of is health and having lot of issues in his personal life.

2.3 PROPOSED SYSTEM

To resolve the mental well-being machine learning technique play important role. It holds great promise to transform mental health care. Its tools also hold the potential to extend the current capabilities of clinicians, to deal with complex problems and ever-expanding information streams that stretch the limits of human ability. we have developed a framework for determining the mental health state of an individual. For further improvements the concept of Machine Learning can be used for very large dataset. Our proposed different levels of questionnaire and based on the results of that provide free checking of a person's mental state and help him by diagnosis prediction.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

• To ensure availability and accessibility of minimum mental health care for all in the foreseeable future, particularly to the most vulnerable and underprivileged sections of the population.

• To encourage the application of mental health knowledge in general health care and in social development

• To promote community participation in the mental health services development and to stimulate efforts towards self help in the community

• To know the major cause of mental illness through mental health analysis.

2.4 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This

is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility

- Technical Feasibility

- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.

- The cost of the hardware and software.

- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIOURAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?

- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- System   : i5 processor
- Hard disk   : 100 GB
- Ram   : 4 GB

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system   : Windows 7
- Coding   : Python
- Tool   : Jupyter notebook, Colab

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE
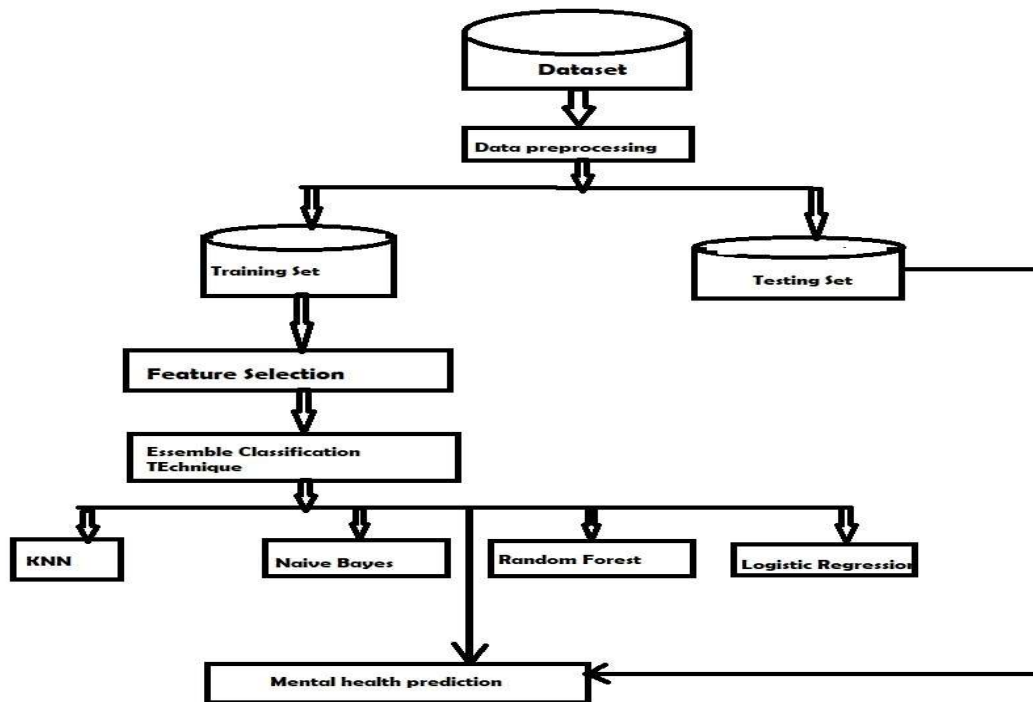
Fig: Architecture of Mental health prediction using machine learning

3.2 MODULES DESCRIPTION

DATA PREPROCESSING

- Data preprocessing is a technique that is used to convert raw data into a clean dataset.

- The data is gathered from different sources is in raw format which is not feasible for the analysis.

- In order to perform data preprocessing using Python, we need to import some predefined Python libraries they are :

- **Numpy:** Numpy Python library is used for including any type of mathematical operation in the code.

- **Pandas:** The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets.

- **Matplotlib** – Matplotlib is a Python 2D plotting library that is used to plot any type of charts in Python.

- **Pre-processing for this approach takes 4 simple yet effective steps:**

I. **Attribute Selection**: The attribute like serial no. is not required. The main attributes used for this study are GRE Scores, TOEFL Scores, CGPA, and University Ranking.

II. **Cleaning missing values**: The library used for the task is called Scikit Learn preprocessing. It contains a class called Imputer which will help us take care of the missing data.

III. **Feature Scaling** : It is performed during the data pre-processing to handle highly varying magnitudes or values or units.

- If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

- A feature scaling in machine learning model is based on Euclidean distance.
  Training and Test data Splitting the Dataset into Training set and Test Set: The next step is to split our dataset into two. Training set and a Test set.

- We will train our machine learning models on our training set and then we will test the models on our test set to examine how accurately it will predict.

- A general rule of the thumb is to assign 75% of the dataset to training set and therefore the remaining 25% to test set.

  DATA GATHERING

- Data gathering is the process of collecting and measuring information from countless different sources.

- In order to use the data we collect to develop practical machine learning solutions, it must be collected.

- The dataset has been collected from students of different colleges. The dataset collected consist of instances of students.

- To use the dataset in our code, we usually put it into a CSV file.

3.3 USECASE DIAGRAM
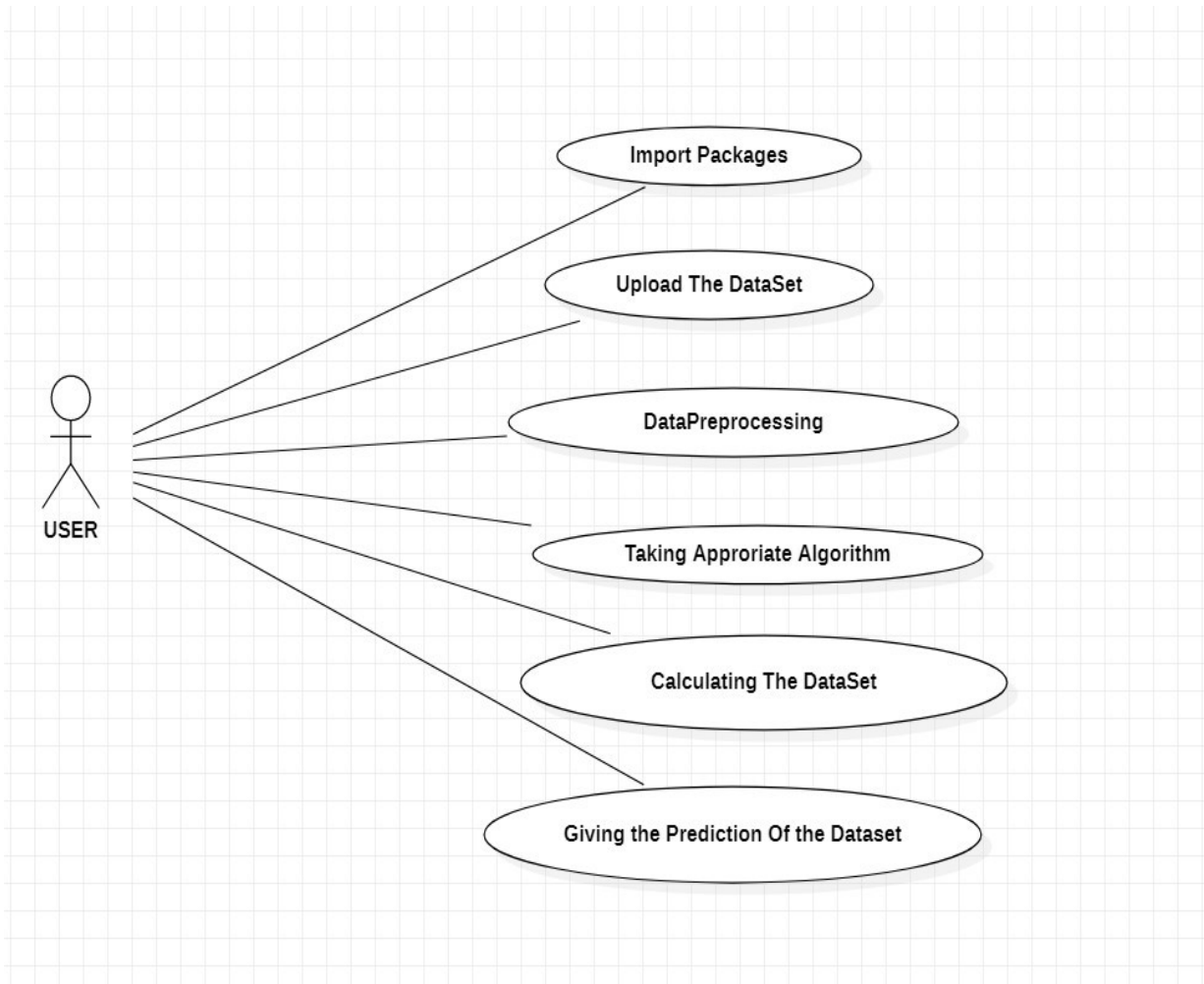

Fig: use case diagram for mental health prediction using machine learning
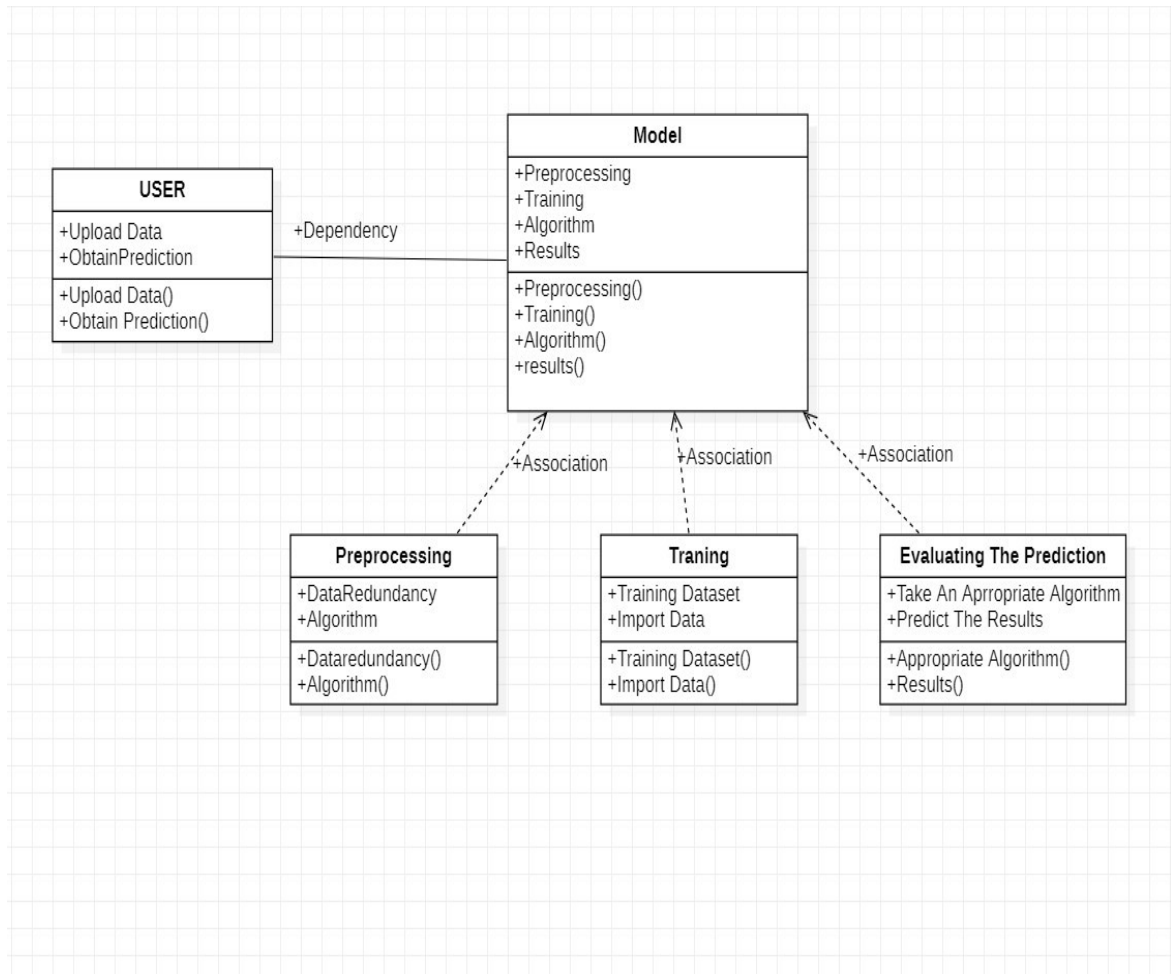
3.5 CLASS DIAGRAM

Fig: class diagram for mental health prediction using machine learning
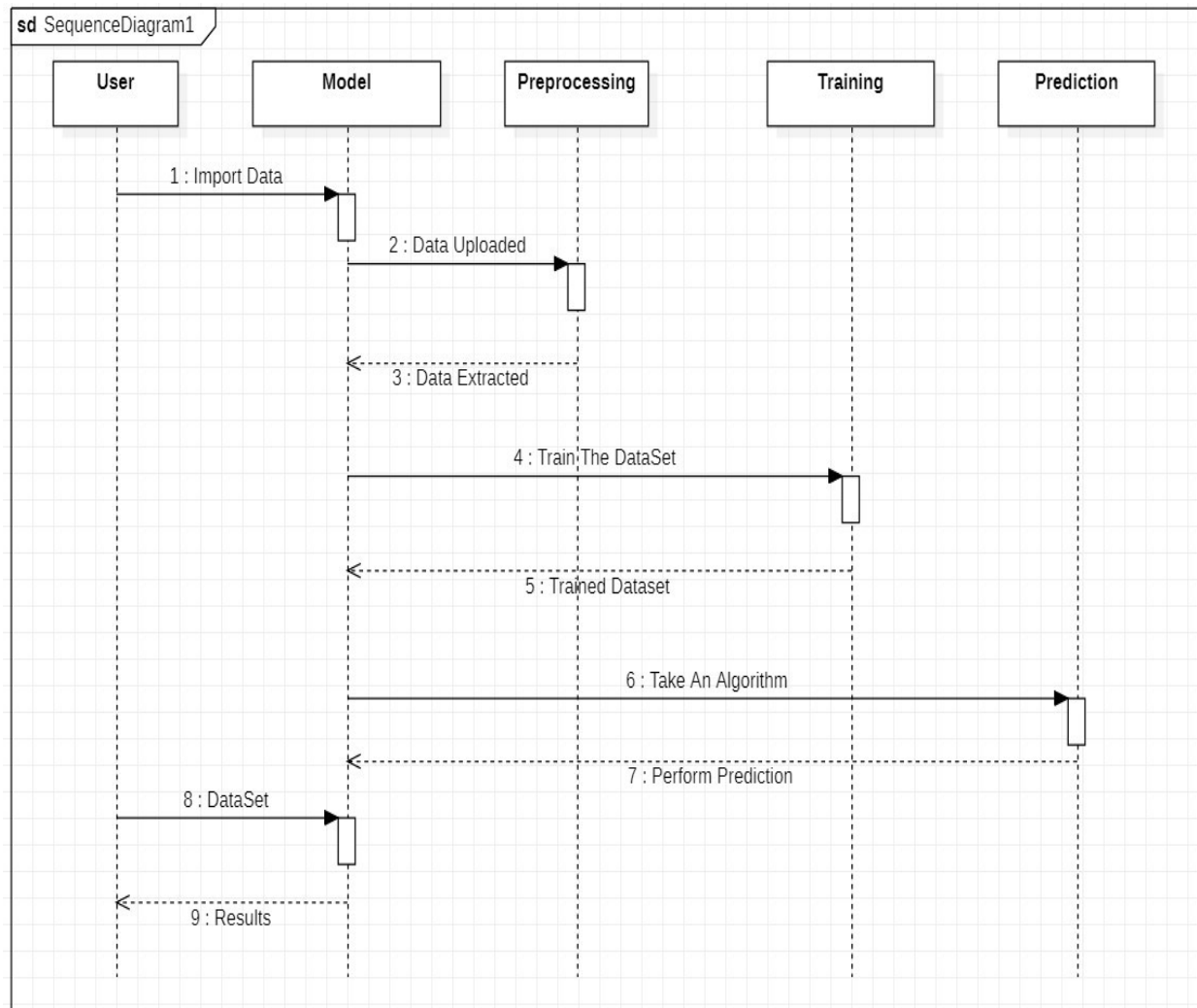
3.6 SEQUENCE DIAGRAM

Fig: sequence diagram for mental health prediction using machine learning
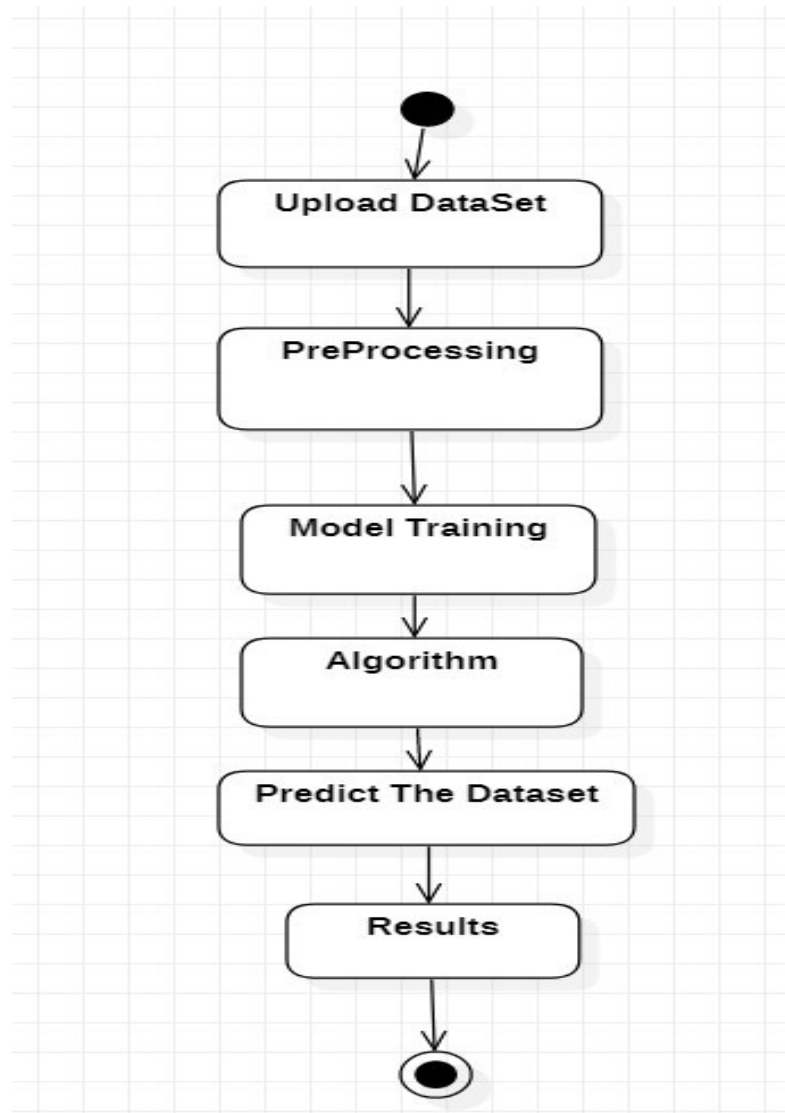
3.7 ACTIVITY DIAGRAM

Fig: activity diagram for mental health prediction using machine learning

4. IMPLEMENTATION

IMPLEMENTATION

4.1 SAMPLE CODE

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter

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

#Neural Network
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import RandomizedSearchCV

#Bagging
from sklearn.ensemble import BaggingClassifier, AdaBoostClassifier
from sklearn.neighbors import KNeighborsClassifier

#Naive bayes
from sklearn.naive_bayes import GaussianNB
#Stacking
from mlxtend.classifier import StackingClassifier

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input
directory

# from subprocess import check_output
# print(check_output(["ls", "./input"]).decode("utf8"))

# Any results you write to the current directory are saved as output.
```

```
#reading in CSV's from a file path
train_df = pd.read_csv('./input/survey.csv')


#Pandas: whats the data row count?
print(train_df.shape)

#Pandas: whats the distribution of the data?
print(train_df.describe())

#Pandas: What types of data do i have?
print(train_df.info())
(1259, 27)

#missing data
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)

#dealing with missing data
#Let's get rid of the variables "Timestamp",“comments”, “state” just to make our lives easier.
train_df = train_df.drop(['comments'], axis= 1)
train_df = train_df.drop(['state'], axis= 1)
train_df = train_df.drop(['Timestamp'], axis= 1)

train_df.isnull().sum().max() #just checking that there's no missing data missing...
train_df.head(5)
# Assign default values for each data type
defaultInt = 0
defaultString = 'NaN'
defaultFloat = 0.0

# Create lists by data tpe
intFeatures = ['Age']
stringFeatures = ['Gender', 'Country', 'self_employed', 'family_history', 'treatment', 'work_interfere',
          'no_employees', 'remote_work', 'tech_company', 'anonymity', 'leave',
'mental_health_consequence',
          'phys_health_consequence', 'coworkers', 'supervisor', 'mental_health_interview',
'phys_health_interview',
          'mental_vs_physical', 'obs_consequence', 'benefits', 'care_options', 'wellness_program',
          'seek_help']
floatFeatures = []

# Clean the NaN's
for feature in train_df:
    if feature in intFeatures:
        train_df[feature] = train_df[feature].fillna(defaultInt)
    elif feature in stringFeatures:
        train_df[feature] = train_df[feature].fillna(defaultString)
```

```
    elif feature in floatFeatures:
        train_df[feature] = train_df[feature].fillna(defaultFloat)
    else:
        print('Error: Feature %s not recognized.' % feature)
train_df.head(5)
#clean 'Gender'
#Slower case all columm's elements
gender = train_df['Gender'].str.lower()
#print(gender)

#Select unique elements
gender = train_df['Gender'].unique()

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())
#complete missing age with mean
train_df['Age'].fillna(train_df['Age'].median(), inplace = True)

# Fill with media() values < 18 and > 120
s = pd.Series(train_df['Age'])
s[s<18] = train_df['Age'].median()
train_df['Age'] = s
s = pd.Series(train_df['Age'])
s[s>120] = train_df['Age'].median()
train_df['Age'] = s

#Ranges of Age
train_df['age_range'] = pd.cut(train_df['Age'], [0,20,30,65,100], labels=["0-20", "21-30", "31-65", "66-100"], include_lowest=True)

#There are only 0.014% of self employed so let's change NaN to NOT self_employed
#Replace "NaN" string from defaultString
train_df['self_employed'] = train_df['self_employed'].replace([defaultString], 'No')
print(train_df['self_employed'].unique())
['No' 'Yes']
```

```python
def logisticRegression():
    # train a logistic regression model on the training set
    logreg = LogisticRegression()
    logreg.fit(X_train, y_train)

    # make class predictions for the testing set
    y_pred_class = logreg.predict(X_test)

    print('########## Logistic Regression ##############')

    accuracy_score = evalClassModel(logreg, y_test, y_pred_class, True)

    #Data for final graph
    methodDict['Log. Regres.'] = accuracy_score * 100
```

```python
def Knn():
    # Calculating the best parameters
    knn = KNeighborsClassifier(n_neighbors=5)

        # From https://github.com/justmarkham/scikit-learn-
    videos/blob/master/08_grid_search.ipynb
    #tuningCV(knn)
    #tuningGridSerach(knn)
    #tuningMultParam(knn)

    # define the parameter values that should be searched
    k_range = list(range(1, 31))
    weight_options = ['uniform', 'distance']

    # specify "parameter distributions" rather than a "parameter grid"
    param_dist = dict(n_neighbors=k_range, weights=weight_options)
    tuningRandomizedSearchCV(knn, param_dist)

    # train a KNeighborsClassifier model on the training set
    knn = KNeighborsClassifier(n_neighbors=27, weights='uniform')
    knn.fit(X_train, y_train)

    # make class predictions for the testing set
    y_pred_class = knn.predict(X_test)

    print('########## KNeighborsClassifier ##############')

    accuracy_score = evalClassModel(knn, y_test, y_pred_class, True)

    #Data for final graph
    methodDict['KNN'] = accuracy_score * 100
def treeClassifier():
    # Calculating the best parameters
    tree = DecisionTreeClassifier()
    featuresSize = feature_cols.__len__()
    param_dist = {"max_depth": [3, None],
            "max_features": randint(1, featuresSize),
            "min_samples_split": randint(2, 9),
            "min_samples_leaf": randint(1, 9),
            "criterion": ["gini", "entropy"]}
```

```
    tuningRandomizedSearchCV(tree, param_dist)


    # train a decision tree model on the training set
    tree = DecisionTreeClassifier(max_depth=3, min_samples_split=8, max_features=6,
criterion='entropy', min_samples_leaf=7)
    tree.fit(X_train, y_train)


    # make class predictions for the testing set
    y_pred_class = tree.predict(X_test)


    print('########### Tree classifier ###############')


    accuracy_score = evalClassModel(tree, y_test, y_pred_class, True)


    #Data for final graph
    methodDict['Tree clas.'] = accuracy_score * 100
```
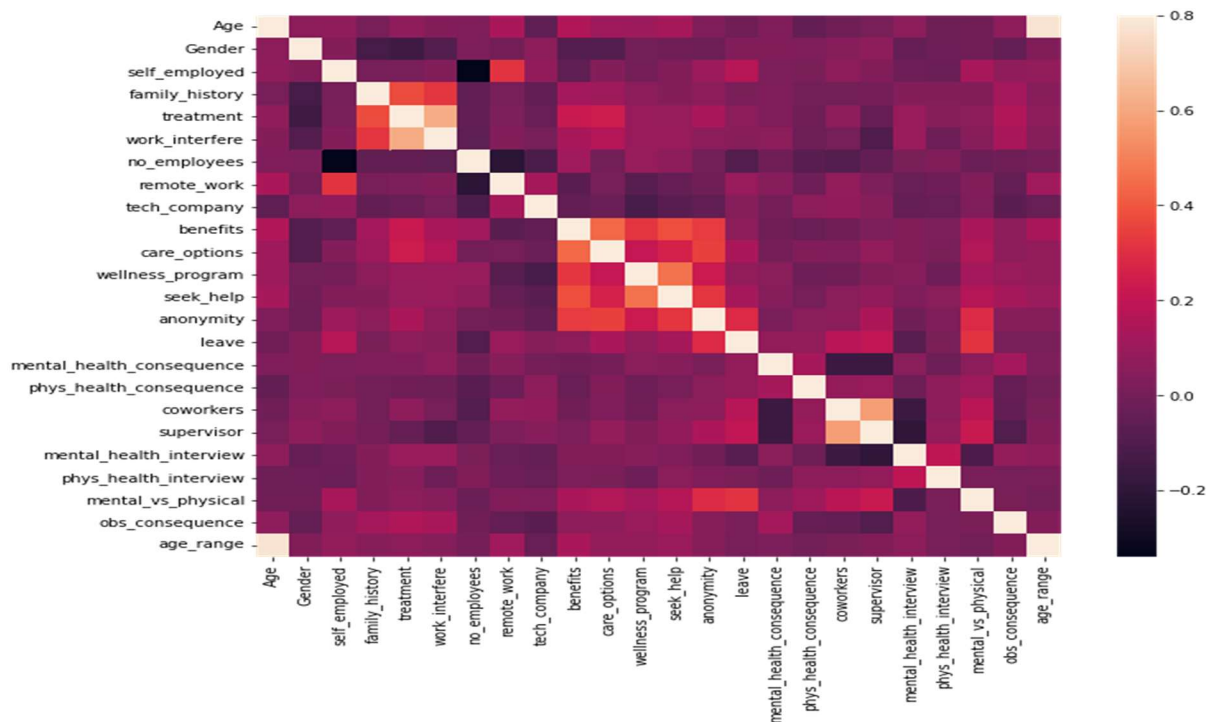
5. SCREENSHOTS

5. SCREENSHOTS

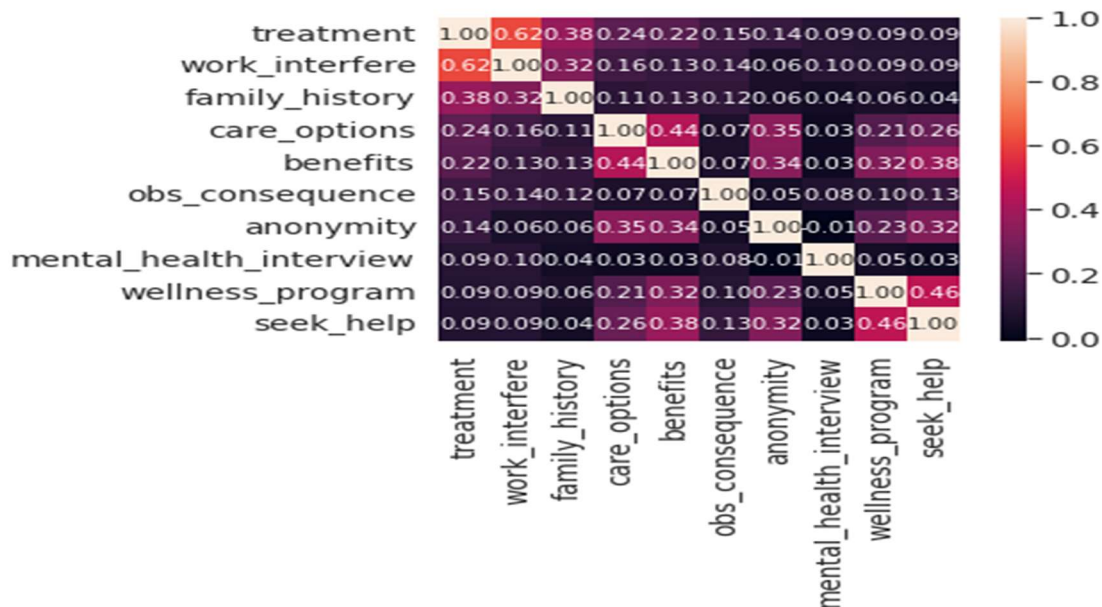Fig 5.1: Correlation matrix

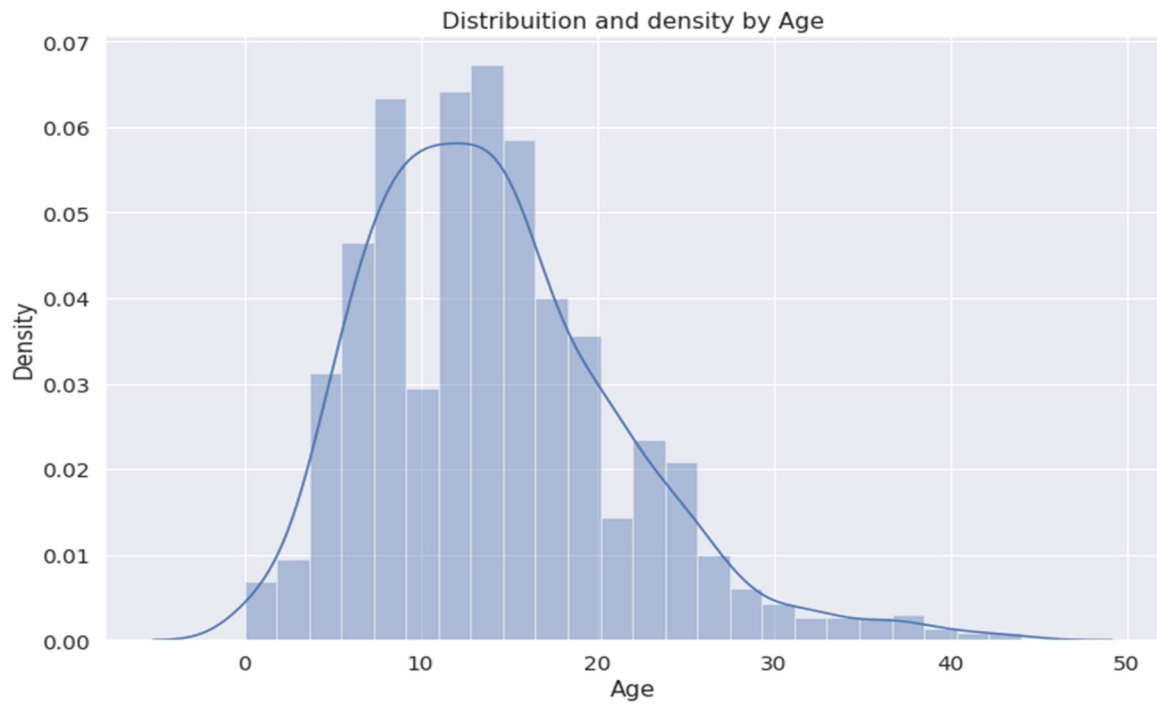Fig:5.2 Variability Comparison Between Categories Of Variables

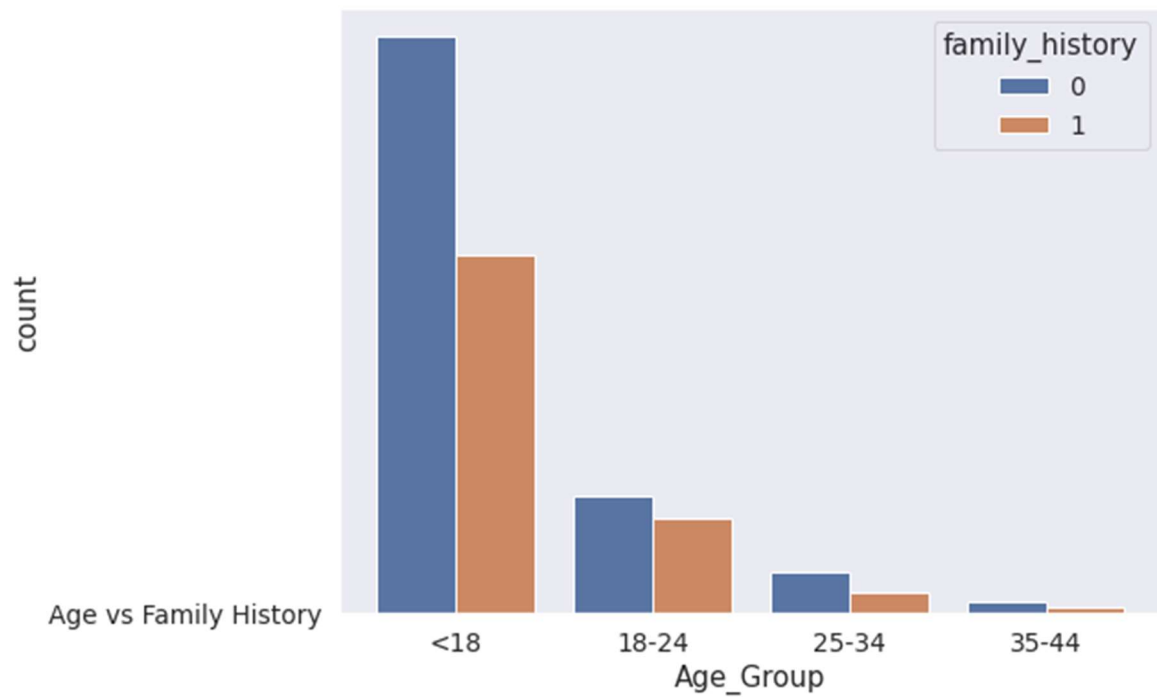Fig 5.3: Distribution and density by age
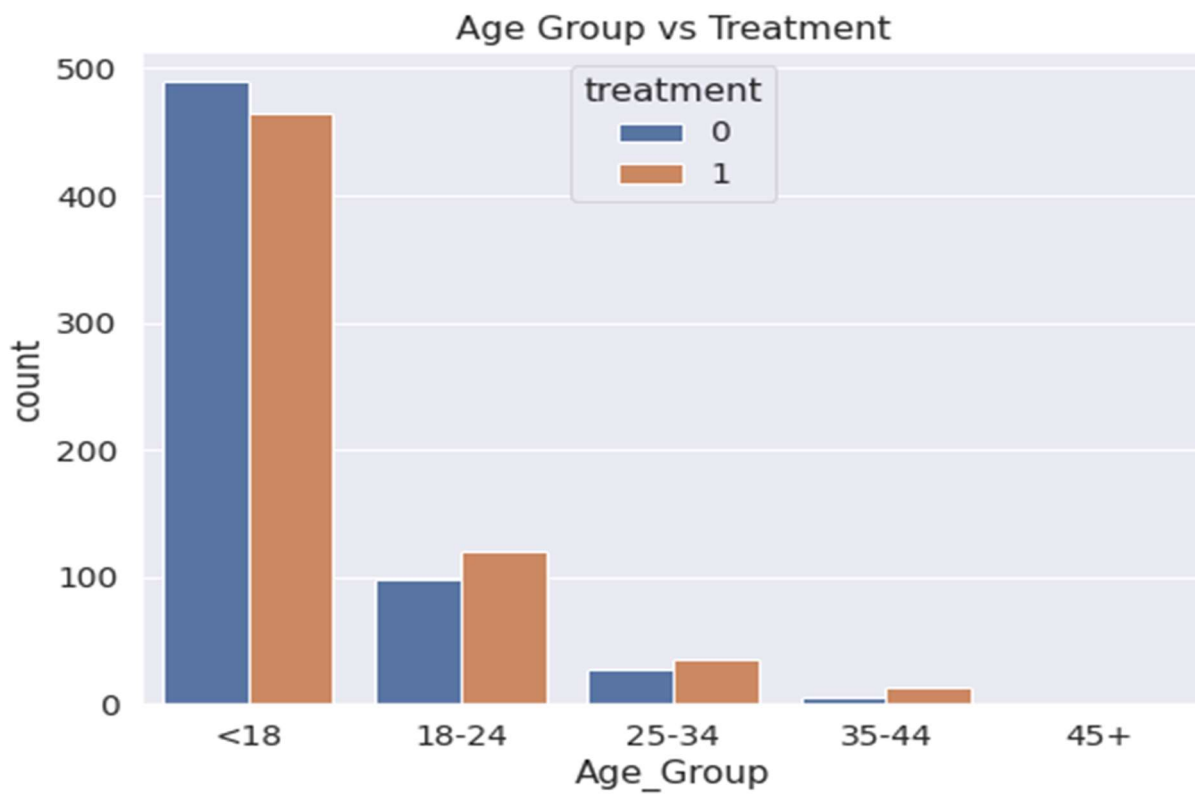

Fig 5.4: Age vs family history
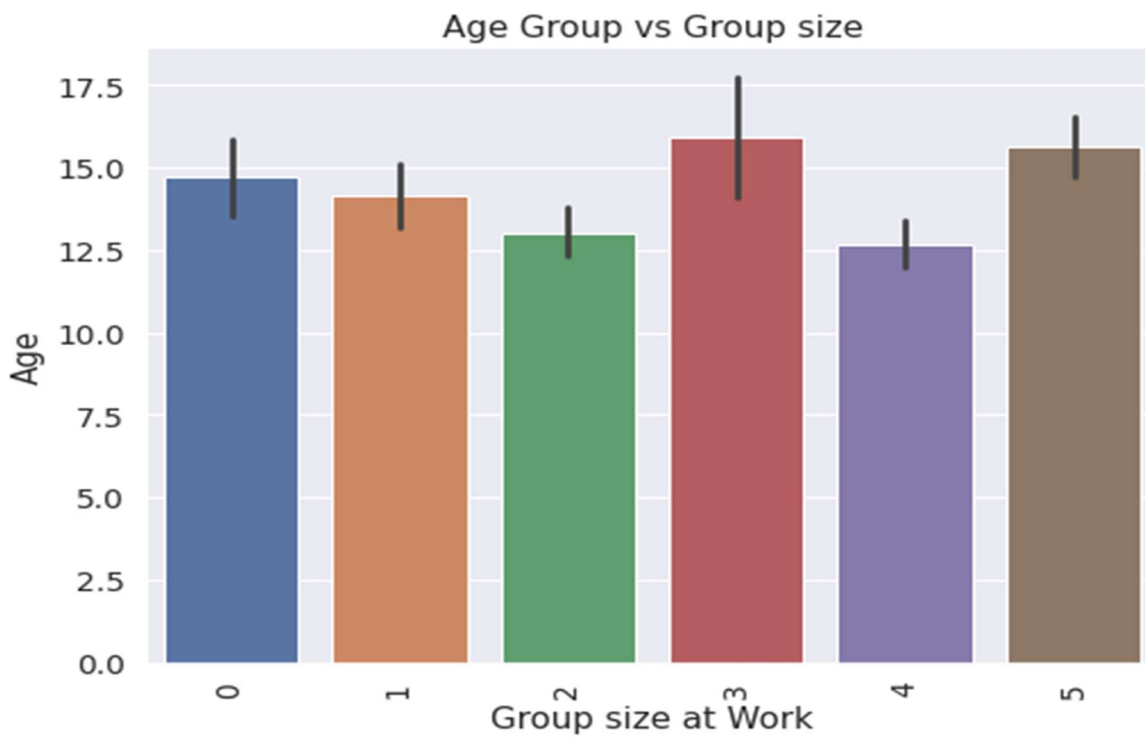
Fig 5.5: Age group vs treatment
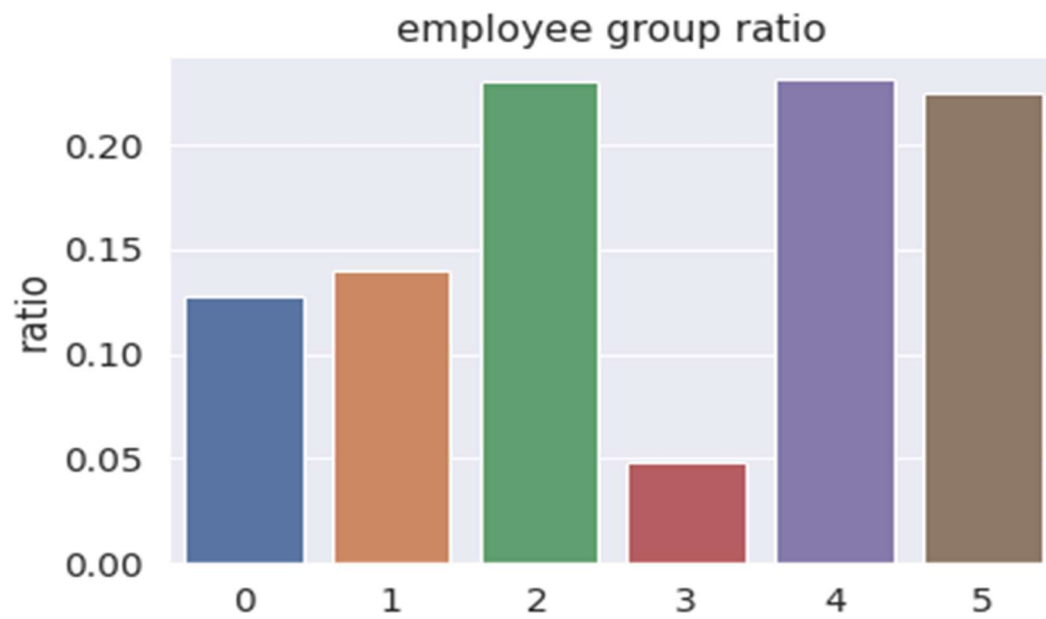

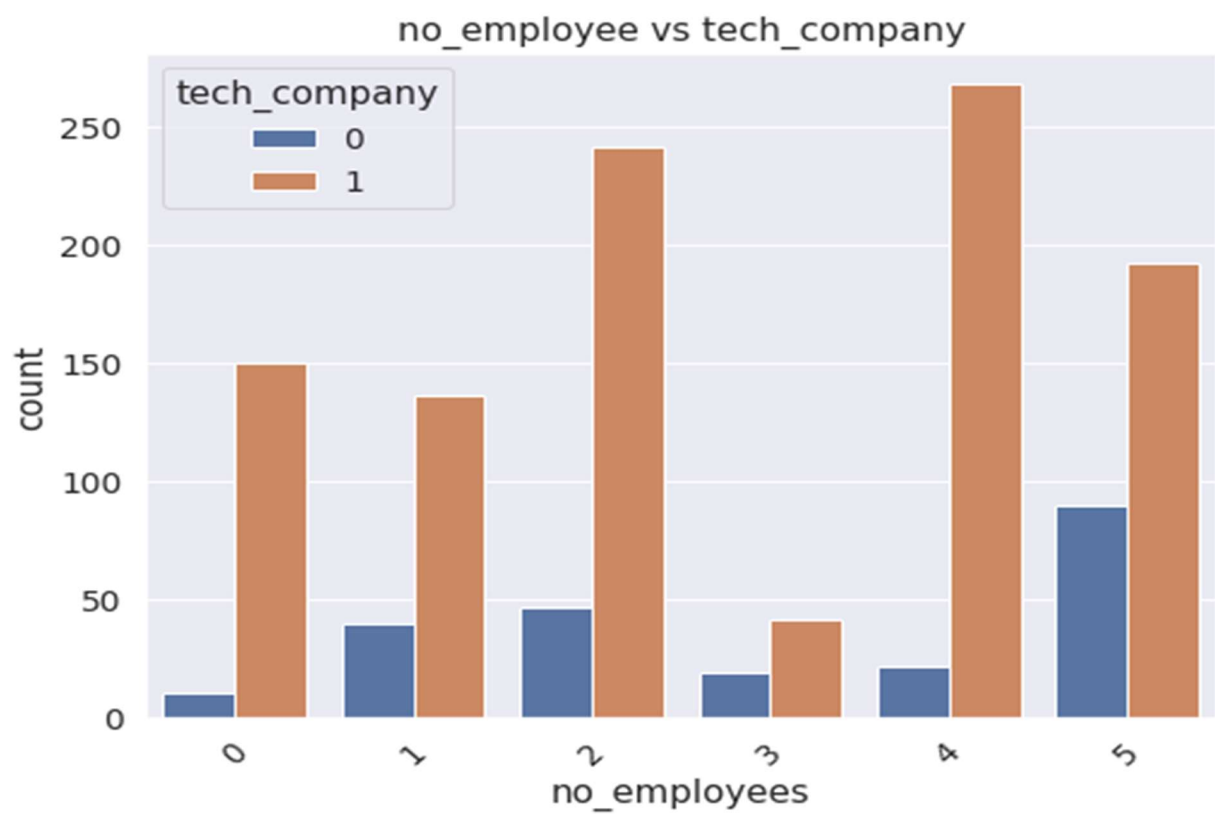Fig 5.6: Age group vs group size

Fig 5.7: Employee group ratio

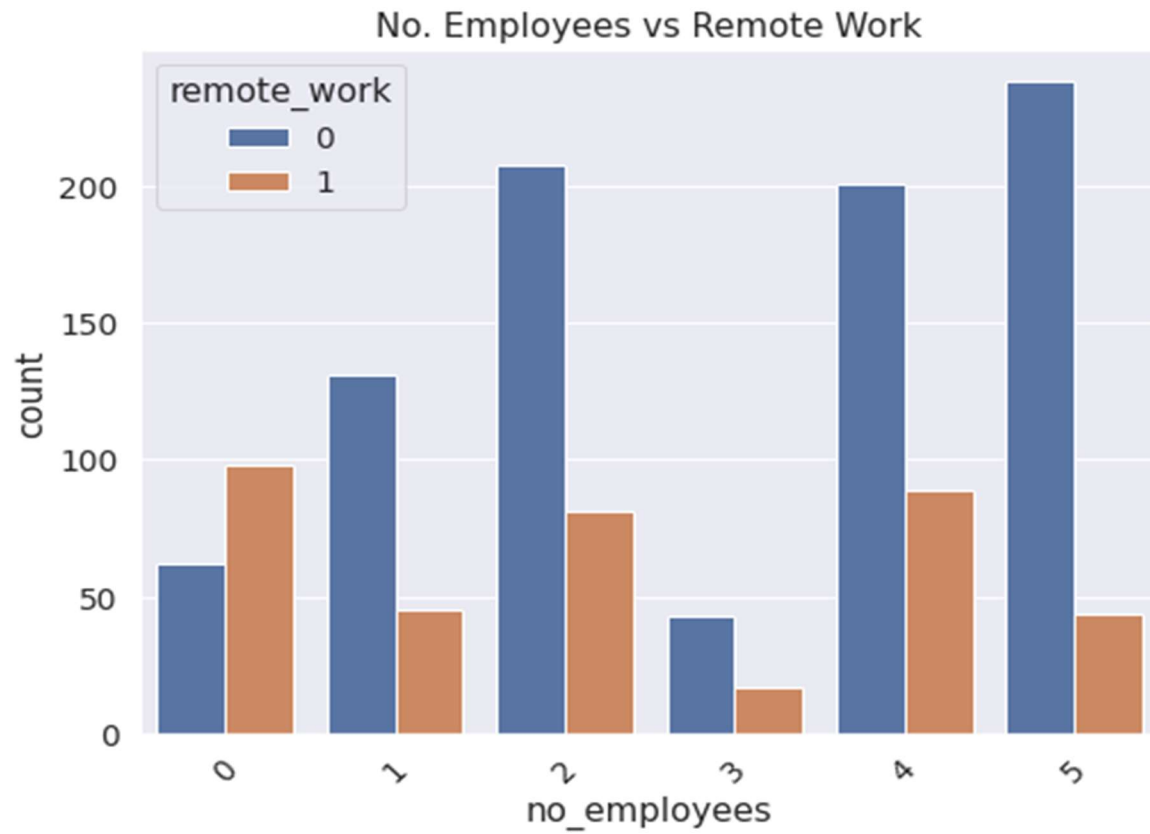Fig 5.8: Number of employees vs tech company
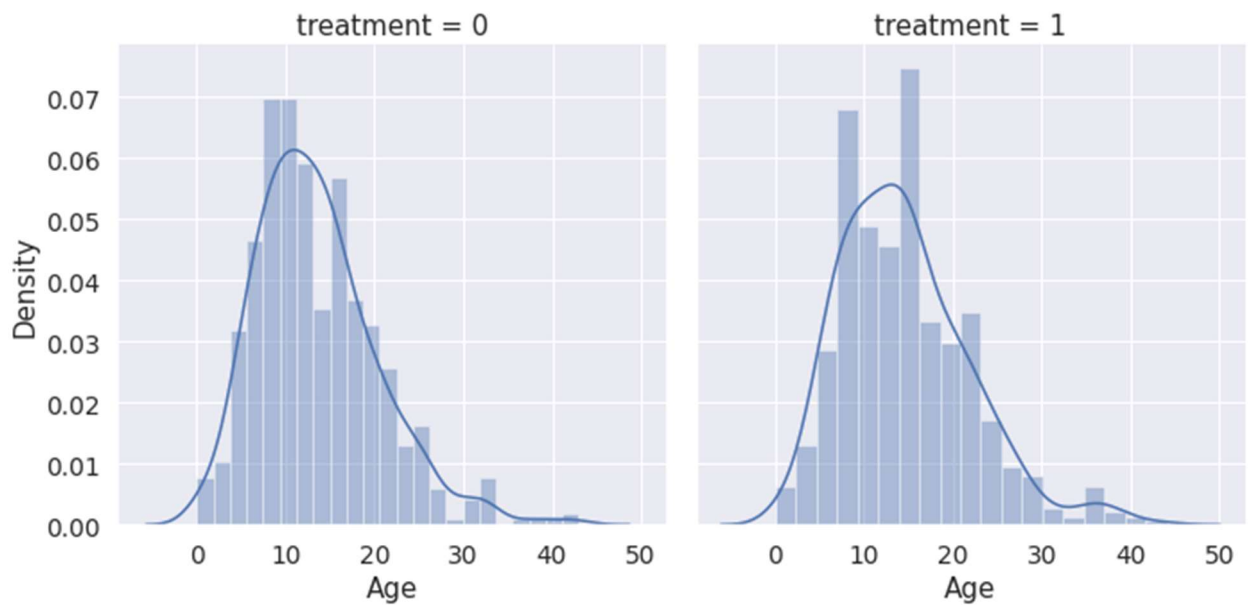
Fig 5.9: Number of employees vs remote work

Fig 5.10: Treatment Comparison

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover very conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.

- Invalid Input : identified classes of invalid input must be rejected.

- Functions : identified functions must be exercised.

- Output : identified classes of application outputs must be exercised.

- Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.3 TESTCASES

6.3.1 UPLOADING DATASET

Test Case ID	Test Case Name	Purpose	TestCase	Output
1	User Upload The DataSet	Use for Prediction	The User upload the dataset for prediction	Calculates the Dataset Given

6.3.2 DETECTION

Test Case ID	Test Case Name	Purpose	input	Output
1	Prediction	To Predict the mental health	A dataset is given as input	Mental health Prediction of a person

7. CONCLUSION

7. CONCLUSION

7.1 PROJECT CONCLUSION

Many different techniques and algorithms had been introduced and proposed to test and solve the mental health problems. There are still many solutions that can be refined. In addition, there are still many problems to be discovered and tested using a wide variety of settings in machine learning for the mental health domain. As classifying the mental health data is generally a very challenging problem, the features used in the machine learning algorithms will significantly affect the performance of the classification. The existing studies and research show that machine learning can be a useful tool in helping understand psychiatric disorders. Besides that, it may also help distinguish and classify the mental health problems among patients for further treatment. Newer approaches that use data that arise from the integration of various sensor modalities present in technologically advanced devices have proven to be a convenient resource to recognize the mood state and responses from patients among others. It is noticeable that most of the research and studies are still struggling to validate the results because of insufficiency of acceptable validated evidence, especially from the external sources. Besides that, most of the machine learning might not have the same performance across all the problems. The performance of the machine learning models will vary depending on the data samples obtained and the features of the data. Moreover, machine learning models can also be affected by pre-processing activities such as data cleaning and parameter tuning in order to achieve optimal results.

7.2 FUTURE SCOPE

We will investigate and analyse the data with various machine learning algorithms to choose the highest accuracy among the machine learning algorithms. Not only that, challenges and limitations faced by the us need to be managed with proper care to achieve satisfactory results that could improve the clinical practice and decision-making.

8. BIBILIOGRAPHY

8. BIBILIOGRAPHY

8.1 REFERENCES

1. G. Miner, L. Miner, M. Goldstein et al., Practical Predictive Analytics and Decisioning Systems for Medicine: Informatics Accuracy and Cost-Effectiveness for Healthcare Administration and Delivery Including Medical Research, Academic Press, Inc., Cambridge, MA, USA, 1st edition, 2016.

2. M. Hamilton, "Development of a rating scale for primary depressive illness," British Journal of Social and Clinical Psychology, vol. 6, no. 4, pp. 278–296, 1967.
   View at: Publisher Site | Google Scholar

3. American Psychiatric Association, Diagnostic and Statistical Manual of Mental Disorders, American Psychiatric Association, Reston, VA, USA, 2013.

4. M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," Science, vol. 349, no. 6245, pp. 255–260, 2015.
   View at: Publisher Site | Google Scholar

8.2 WEBSITES

1. https://www.hindawi.com/journals/acisc/2022/9970363/

2.https://www.researchgate.net/publication/357624760_Mental_Health_Prediction_Using_Machine_Learning_Taxonomy_Applications_and_Challenges