MENTAL HEALTH PREDICTION USING MACHINE LEARNING

A dissertation submitted to the

THIRUVALLUVAR UNIVERSITY

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

Submitted *by*

V. YUVARANI 21022U09098

Under the guidance of

Mrs. S. KARTHIGA

Assistant Professor of Computer Applications

DEPARTMENT OF COMPUTER APPLICATIONS

SRI AKILANDESWARI WOMEN’S COLLEGE

WANDIWASH – 604408

APRIL/MAY-2025

MENTAL HEALTH PREDICTION USING MACHINE LEARNING

A dissertation submitted to the

THIRUVALLUVAR UNIVERSITY

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

Submitted *by*

V. YUVARANI 21022U09098

Under the guidance of

Mrs. S. KARTHIGA

Assistant Professor of Computer Applications

DEPARTMENT OF COMPUTER APPLICATIONS

SRI AKILANDESWARI WOMEN'S COLLEGE

WANDIWASH – 604408

APRIL/MAY-2025

MRS. S. KARTHIGA

ASSISTANT PROFESSOR IN COMPUTER APPLICATIONS,

SRI AKIILANDESWARI WOMENS COLLEGE,

WANDIWASH – 604 408

CERTIFICATE

This is to certify that the Project Work entitled **“MENTAL HEALTH PREDICTION USING MACHINE LEARNING”** submitted in partial fulfillment to qualify for the award of the degree of **BACHELOR OF COMPUTER APPLICATIONS** to the **THIRUVALLUVAR UNIVERSITY** during the academic year **2022-2025** is a bonafide record of the work carried out by **V. YUVARANI (21022U09098).**

INTERNAL GUIDE **HEAD OF THE DEPARTMENT**

Submitted for the Viva-Voce Examination held on_______________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

This is to certify that the Project Work entitled **“MENTAL HEALTH PREDICTION USING MACHINE LEARNING”** submitted in partial fulfillment to qualify for the award of the degree of **BACHELOR OFCOMPUTER APPLICATIONS** to the Thiruvalluvar University is the original work done under the guidance of **Mrs. S.KARTHIGA** during the period of her study in Department of BCA, Sri Akilandeswari Women’s College, Wandiwash and the dissertation has not formed the basis for the award of any degree fellowship or similar title of any candidate of any university.

SIGNATURE OF THE CANDIDATES

V. YUVARANI 21022U09098

SIGNATURE OF THE GUIDE **SIGNATURE OF THE HOD**

ACKNOWLEDGEMENT

With the blessing of **Mr. Ln. B. MUNIRATHINAM,** Founder, **Mr. Ln. M. RAMANAN,** Chairman and **Mrs. R. PRIYA RAMANAN,** Secretary of Sri Akilandeswari Women's College, Wandiwash for their kind support.

I wish to express my sincere thanks to **Dr. S. RUKMANI** Principal and **Ms. K. SUGUNA,** Vice-Principal, Sri Akilandeswari Women's College, Wandiwash, for their constant encouragement and valuable support.

I would like to express my heartfelt thanks to **Mrs. D. RAJASELVI**, Head of Department of BCA, for extending instrument facilities to carry out the project work. I would like to extend my sincere thanks to **Mrs. S.KARTHIGA**, for her motivation with a spirit of enthusiasm, for encouraging gracefully throughout the course of work and for suggesting relevant and enlightening solutions for problems.

I would like to express my heartfelt gratitude to all my faculty members of the Department Of Computer Applications for their moral support towards the completion of the work.

I express my deep thanks to my beloved parents and my friends for their good wishes, great concern and encouragement.

Above all, I devotionally thank almighty god for showering his constant blessings.

V. YUVARANI 21022U09098

MYMISOLUTIONANDTRAINING

57thStreet,VenkatRamanNagar,Korattur,Chennai–600080.

CIN:U92190TN2021PTC143111PAN:AABCY3842LGSTIN:33AAJCG0011B1ZL

mymitraining@gmail.com,malathimca29@gmail.com

TO WHOM SO EVER IT MAY CONCERN

This is to certify that Mr/Miss **V. YUVARANI (21022U09098)** Bachelor of Computer Applications final year students of ***Sri Akilandeswari Women's College, Vandavasi*** has done project work in the company on "**MENTAL HEALTH PREDICTION USING MACHINE LEARNING** " towards the fulfillment of the award of "*Bachelor of Computer Applications*" during the period of December 2024 to April 2025.

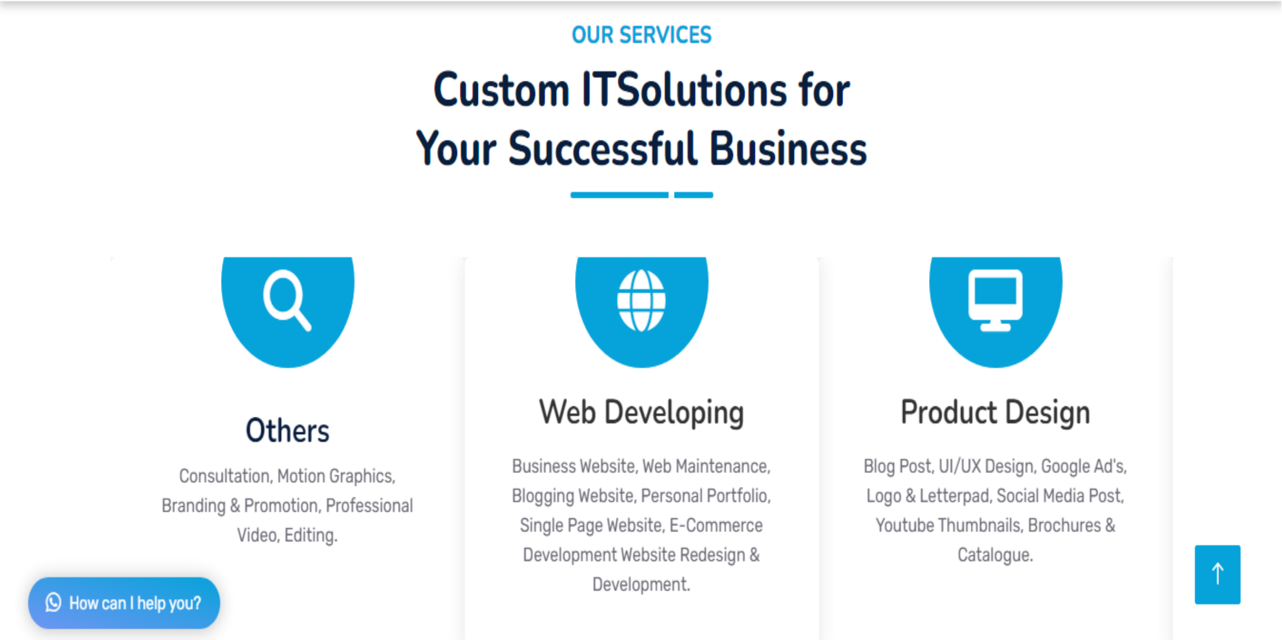
MYMI solutions and Training

Organization profile

MYMI solutions and training is a leading global IT Solutions company that offers web development, digital marketing and multimedia services. We are dedicated, passionate service providers offering best industry practices synced with technology expertise and business domain knowledge to drive the digital revolution.

Upgrade your IT Solutions by collaborating with a highly-skilled, experienced, handpicked team of experts. We ensure your project is in your hand on time at an affordable price.

Since, day one, we have believed in a mutual win by creating world-class digital Solutions. We paved our path in digital transformation through our proficiency in understanding business challenges and professional competence. We constantly strive to help our clients by harnessing the power of digital Solutions, analytics, and advanced technologies to scale clients' businesses.



OUR SERVICES

Custom ITSolutions for Your Successful Business

Others

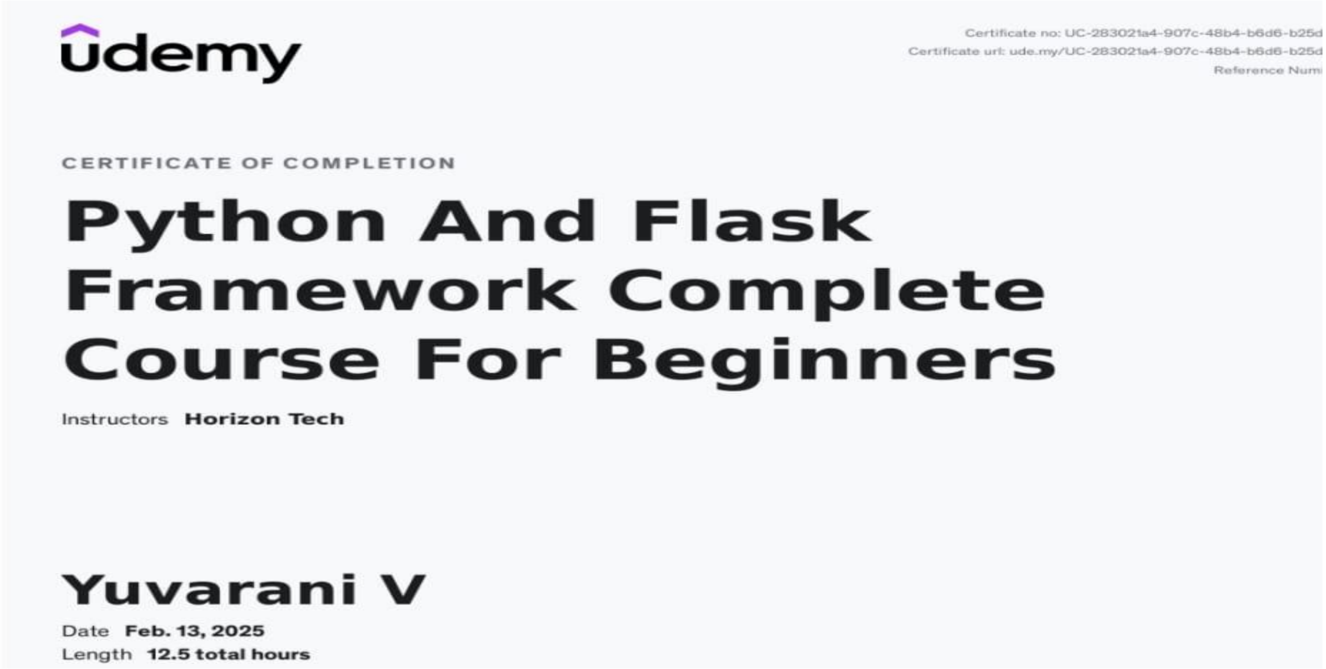
Consultation, Motion Graphics, Branding & Promotion, Professional Video, Editing.

Web Developing

Business Website, Web Maintenance, Blogging Website, Personal Portfolio, Single Page Website, E-Commerce Development Website Redesign & Development.

Product Design

Blog Post, UI/UX Design, Google Ad's, Logo & Letterpad, Social Media Post, Youtube Thumbnails, Brochures & Catalogue.



udemy

Certificate no: UC-283021a4-907c-48b4-b6d6-b25d
Certificate url: ude.my/UC-283021a4-907c-48b4-b6d6-b25d
Reference Num

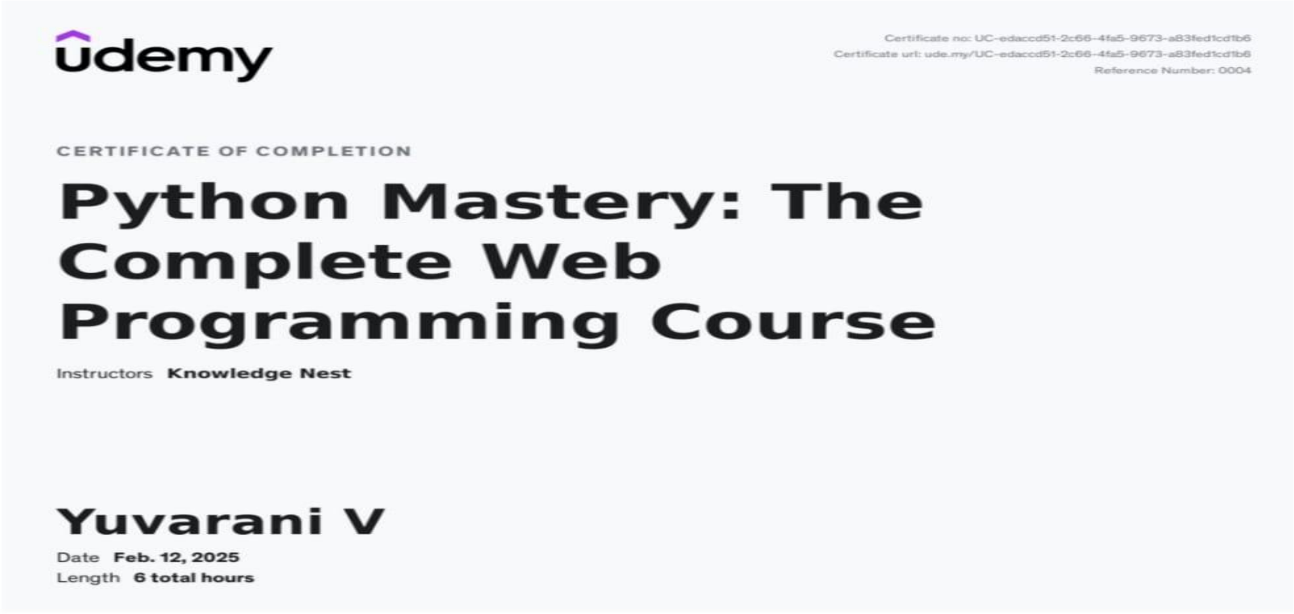
CERTIFICATE OF COMPLETION

Python And Flask Framework Complete Course For Beginners

Instructors **Horizon Tech**

Yuvarani V

Date **Feb. 13, 2025**
Length **12.5 total hours**



udemy

Certificate no: UC-edaccd51-2c66-4fa5-9673-a83fed1cd1b6
Certificate url: ude.my/UC-edaccd51-2c66-4fa5-9673-a83fed1cd1b6
Reference Number: 0004

CERTIFICATE OF COMPLETION

Python Mastery: The Complete Web Programming Course

Instructors **Knowledge Nest**

Yuvarani V

Date **Feb. 12, 2025**
Length **6 total hours**

-

MENTAL HEALTH PREDICTION USING MACHINE LEARNING

ABSTRACT

ABSTRACT

Professionals can help diagnose and treat patients more effectively by detection mental health issues early. In this article, we discuss the current status of AI in the mental health field and its potential applications in healthcare. Machine learning techniques can help address the basic mental health issues that people face, such as anxiety and depression. They can also detect patterns and provide helpful suggestions for addressing the problems. The attribute data has been reduced using Feature Selection algorithms. Various machine learning algorithms have been compared in terms of accuracy over the full set of attributes and a select set of attributes. Although various algorithms have been studied, further work is still needed to reduce the aperture between AI and mental health analysis.

TABLE OF CONTENT

LIST OF ABBREVIATIONS

LIST OF ABBREVIATIONS

1)	**AI** – Artificial Intelligence
2)	**ML** – Machine Learning
3)	**DS** – Data Science
4)	**NLP** – Natural Language Processing
5)	**EDA** – Exploratory Data Analysis
6)	**IOT** – Internet of Things
7)	**CSV** – Comma-Separated Values
8)	**JSON** – JavaScript Object Notation
9)	**SQL** – Structured Query Language
10)	**DBMS** – Database Management System
11)	**PCA** – Principal Component Analysis
12)	**LDA** – Linear Discriminant Analysis
13)	**ANOVA** – Analysis of Variance
14)	**SVM** – Support Vector Machine
15)	**KNN** – K-Nearest Neighbours
16)	**RF** – Random Forest
17)	**DT** – Decision Tree
18)	**LR** – Logistic Regression
19)	**CNN** – Convolutional Neural Network
20)	**RNN** – Recurrent Neural Network
21)	**LSTM** – Long Short-Term Memory
22)	**XG Boost** – Extreme Gradient Boosting
23)	**TF-IDF** – Term Frequency-Inverse Document Frequency
24)	**WHO** – World Health Organization
25)	**DSM** – Diagnostic and Statistical Manual of Mental Disorders
26)	**PH Q** – Patient Health Questionnaire

27)	**GAD** – Generalized Anxiety Disorder
28)	**HAM-D** – Hamilton Depression Rating Scale
29)	**BDI** – Beck Depression Inventory
30)	**CES-D** – Center for Epidemiologic Studies Depression Scale
31)	**HADS** – Hospital Anxiety and Depression Scale
32)	**MSE** – Mean Squared Error
33)	**RMSE** – Root Mean Squared Error
34)	**MAE** – Mean Absolute Error
35)	**R^2** – R-Squared (Coefficient of Determination)
36)	**TP** – True Positive
37)	**TN** – True Negative
38)	**FP** – False Positive
39)	**FN** – False Negative
40)	**AUC** – Area Under Curve
41)	**ROC** – Receiver Operating Characteristic

LIST OF FIGURES

INTRODUCTION

INTRODUCTION

Millions of people around the world are affected by one or more mental disorders that interfere in their thinking and behavior. A timely detection of these issues is challenging but crucial, since it could open the possibility to offer help to people before the illness gets worse. One alternative to accomplish this is to monitor how people express themselves, which is for example what and how they write, or even a step further, what emotions they express in their social media communications. In this study, we analyses two computational representations that aim to model the presence and changes of the emotions expressed by social media users. In our evaluation we use recent public data sets for the mental disorder: Depression. The obtained results suggest that the presence and variability of emotions, captured by the proposed representations, allow to highlight important information about social media users suffering from depression.

CHAPTER- 1

1. INTRODUCTION

1.1 DATA SCIENCE

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains. Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data. Data science can be defined as a blend of mathematics, business acumen, tools, algorithms and machine learning techniques, all of which helps in finding out the hidden insights or patterns from raw data which can be of major use in the formation of big business decisions.

1.2 DATA SCIENTIST

A data scientist is able to identify which questions need to be answered and where to locate the relevant data, as well as to mine, clean, and present data. Businesses use data scientists to source, manage, and analyze large amounts of unstructured data.

REQUIRED SKILLS FOR A DATA SCIENTIST:

- **Programming**: Python, SQL, Scala, Java, R, MATLAB.
- **Machine Learning**: Natural Language Processing, Classification, Clustering.
- **Data Visualization**: Tableau, SAS, D3.js, Python, Java, R libraries.
- **Big data platforms**: MongoDB, Oracle, Microsoft Azure, Cloudera.

1.3 ARTIFICIAL INTELLIGENCE (AI)

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving. Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, and speech recognition and machine vision.

AI applications include advanced web search engines, recommendation systems (used by YouTube, Amazon and Netflix), Understanding human speech (such as Siri or Alexa), self-driving cars (e.g. Tesla), and competing at the highest level in strategic game systems (such as chess and Go). AI research has tried and discarded many different approaches during its lifetime, including simulating the brain, modeling human problem solving, formal logic, large databases of knowledge and imitating animal behavior. In the first decades of the 21st century, highly mathematical statistical machine learning has dominated the field, and this technique has proved highly successful, helping to solve many challenging problems throughout industry and academia.

The various sub-fields of AI research are centered on particular goals and the use ofparticulartools.ThetraditionalgoalsofAIresearchincludereasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects. General intelligence (the ability to solve an arbitrary problem) is among the field’s long-term.

To solve these problems, AI researchers use versions of search and mathematical optimization, formal logic, artificial neural networks, and methods based on statistics, probability and economics. AI also drawspon computer science, psychology, linguistics, philosophy, and many other fields. As the hype around AI has accelerated, vendors have been scrambling to promote how their products a/nd services use AI. No one programming language is synonymous with AI, but a few, including Python, R and Java, are popular.

In this way, a chatbot that is fed examples of text chats can learn to produce life like exchange swith people, or an image recognition tool can learn to identify and describe objects in images by reviewing millions of examples. AI programming focuses on three cognitive skills: learning, reasoning and self correction.

Learning processes: This aspect of AI programming focuses on acquiring data and creating rules for how to turn the data into actionable information. The rules, which are called algorithms, provide computing devices with step-by-step instructions for how to complete a specific task.

Reasoning processes: This aspect of AI programming focuses on choosing the right algorithm to reach a desired outcome.

Self-correction processes: This aspect of AI programming is designed to continually fine-tune algorithms and ensure they provide the most accurate results possible.

AI is important because it can give enterprises insights into their operations that they may not have been aware of previously and because, in some cases, AI can perform tasks better than humans.

Particularly when it comes to repetitive, detail-oriented tasks like analyzing large numbers of legal documents to ensure relevant fields are filled in properly, AI tools often complete jobs quickly and with relatively few errors.

1.3 Natural Language Processing (NLP):

Natural language processing (NLP) allows Machine stored and Understand human language. A sufficiently powerful natural language processing system would enable natural-language user interfaces and the acquisition of knowledge directly from human-written sources, such as newswire texts. Some straight forward applications of natural language processing include information retrieval, text mining, question answering and machine translation. Many current approaches use word co-occurrence frequencies to construct syntactic representations of text. Keyword spotting strategies for search are popular and scalable but dumb; a search query for dog might only match documents with the literal word―dog and misses a document with the word―poodle. Lexical affinity strategies use the occurrence of words such as accident to assess the sentiment of a document. Modern statistical NLP approaches can combine all these strategies as well as others, and often achieve acceptable accuracy at the page or paragraph level. Beyond semantic NLP, the ultimate goal of ―narrative‖ NLP is to embody full understanding of common sense reasoning.

1.4 MACHINE LEARNING:

Machine learning is to predict the future from past data. Machine learning (ML) is a type of Artificial Intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data.

Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

Data scientists use many different kinds of machine learning algorithms to discover patterns in python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they learn ‖about data to make predictions: supervised and unsupervised learning.

Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function from input variables(X) to discrete output variables(y). In Machine Learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation.

This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.

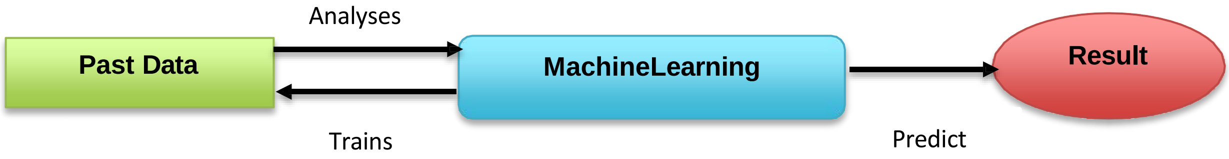


FIG1. PROCESS OF MACHINE LEARNING

Supervised Machine Learning is the majority of practical machine learning uses supervised learning. Supervised learning is where have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output is y = f(X). The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include logistic regression**,** multi-class classification**,** Decision Trees and support vector machines etc**.** Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. Supervised learning problems can be further grouped into Classification problems. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables.

PREPARING THE DATASET

CHAPTER- 2

PREPARING THE DATASET

This dataset contains 1784 records of features, which were then classified into 2 classes:

1. Depression
2. Not depression

To prepare the dataset for mental health prediction, we first collect relevant data from sources such as mental health surveys, clinical records, or social media posts. The collected data undergoes preprocessing, where missing values are handled, duplicates are removed, and text data is standardized. Feature selection is performed to extract meaningful attributes like sentiment analysis, behavioral patterns, and questionnaire responses. The dataset is then labeled into two categories: "Depression" for individuals showing signs of depression and "Not Depression" for those without symptoms. Finally, the data is split into training and testing sets to ensure proper model evaluation and accuracy.

PROPOSED SYSTEM

CHAPTER-3

PROPOSED SYSTEM

The proposed method is to build a ML model to predict mental health (depression). The dataset is first preprocessed and the columns are analyzed and then different machine learning algorithms would be compared to obtain the predictive model with maximum accuracy**.**

3.1 DATA WRANGLING

Data is loaded, checked for cleanliness, and then trimmed and cleaned for analysis.

3.2 DATA COLLECTION

- The data set collected for predicting given data is split into Training set and Test set.
- The 7:3 ratio is applied to split the Training set and Test set.
- The Data Model which was created using machine learning algorithms are applied on the training set and based on the test result accuracy, Test set prediction is done**.**

3.3 BUILDING THE CLASSIFICATION MODEL

ML algorithms prediction model is effective because of the following reasons:

- It provides better results in classification problem.
- It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.
- It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

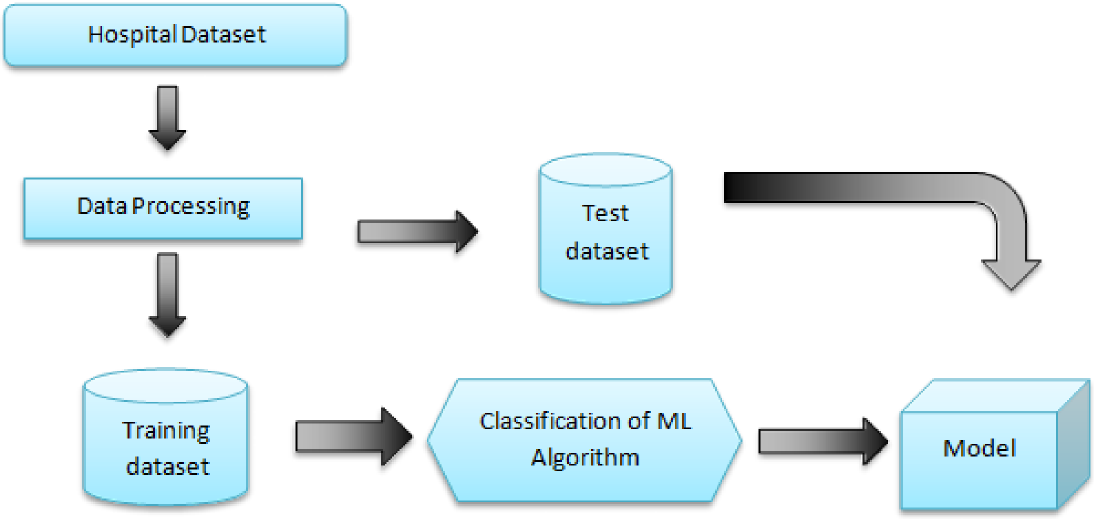


Fig2. Architecture of Proposed Model

3.4 Advantages:

These reports are to the investigation of applicability of Machine Learning Techniques for Mental Health prediction in operational conditions. Finally, it highlights some observations on future research issues, challenges, and needs.

LITERATURE SURVEY

CHAPTER-4

LITERATURE SURVEY

4.1 Review of Literature Survey

Title : Machine Learning-based Approach for Depression Detection in Twitter Using Content and Activity Features

Author: HATOON ALSAGRI, MOURAD YKHLEF

Year : 2020

Social media channels, such as Face book, Twitter, and Instagram, have altered our world forever. People are now increasingly connected than ever and reveal a sort of digital persona. Although social media certainly has several remarkable features, the demerits are undeniable as well. Recent studies have indicated a correlation between high usage of social media sites and increased depression. The present study aims to exploit machine learning techniques for detecting a probable depressed Twitter user based on both, his/her network behavior and tweets. For this purpose, we trained and tested classifiers to distinguish whether a user is depressed or not using features extracted from his/her activities in the network and tweets. The results showed that the more features are used; the higher are the accuracy and F-measure scores in detecting depressed users. This method is a data-driven, predictive approach for early detection of depression or other mental illnesses. This study's main contribution is the exploration part of the features and its impact on detecting the depression level.

This paper defines a binary classification problem as identifying whether a person is depressed, based on his tweets and Twitter profile activity. Different machine learning algorithms are exploited and different feature datasets are explored. Many preprocessing steps are performed, including data preparation and aligning, data labeling, and feature extraction and selection. The SVM model has achieved optimal accuracy metric combinations; it converts an extremely nonlinear classification problem into a linearly separable problem. Although the DT model is comprehensive and follows understandable steps, it can fail if exposed to brand- new data.

**Title : **Predicting the Utilization of Mental Health Treatment with Various Machine Learning Algorithms

Author: MEERA SHARMA, SONOK MAHAPATRA, ADEETHYIA SHANKAR

**Year : **2020

In 2017, about 792 million people (more than 10% of the global population) lived their lives with a mental disorder [24]– 78 million of which committed suicide because of it. In these unprecedented times of COVID-19, mental health challenges have been even further exacerbated as home environments have been proven to be major sources of the creation and worsening of poor mental health. Additionally, proper diagnosis and treatment for people with mental health disorders remains underdeveloped in modern-day's society due to the widely ever- present public stigma attached to caring about mental health. Recently there have been attempts in the data science world to predict if a person is suicidal (and other diagnostic approaches) yet all face major setbacks. To begin, big data has many ethical issues related to privacy and reusability without permission—especially in regards to using feeds from social media.

Additionally, people diagnosed with specific mental health conditions may not actually seek treatment, so data may be incorrect. In this research, we address both of these problems by using anonymous datasets to predict the answer to a different question—whether or not people are seeking mental health treatment. We also use a large variety of machine learning and deep learning classifiers and predictive models to predict with a high accuracy rate through statistical analysis. From our research, we were able to conclude that machine learning can be used to predict likelihood of individuals seeking treatment with a high degree of accuracy (76.3% - 82.5%) by utilizing a self-reported questionnaire.

Similarly, through a simple questionnaire that asks enough questions relevant to mental health, machine learning should also be able to determine if the person requires treatment. Despite stigma surrounding mental illness, individuals would be able to utilize machine learning to determine the correct course of action for their mental illness. As a result, these individuals would be more productive, reducing social and economic costs at the tech workplace.

**Title : ** Prediction of Mental Disorder for employees in IT Industry

Author: SANDHYA P, MAHEK KANTESARIA

**Year : ** 2019

Mental health is nowadays a topic which is most frequently discussed when it comes to research but least frequently discussed when it comes to the personal life. The wellbeing of a person is the measure of mental health. The increasing use of technology will lead to a lifestyle of less physical work. Also, the constant pressure on an employee in any industry will make more vulnerable to mental disorder. These vulnerabilities consist of peer pressure, anxiety attack, depression, and many more. Here we have taken the dataset of the questionnaires which were asked to an IT industry employee. Based on their answers the result is derived. Here output will be that the person needs an attention or not. Different machine learning techniques are used to get the results.

This prediction also tells us that it is very important for an IT employee to get the regular mental health check up to tract their health. The employers should have a medical service provided in their company and they should also give benefits for the affected employees There are many suggestions that employers and employees could keep in mind. Employers need to keep track of number of their employees having mental disorder. Employers should allow flexible work environment with flexible work scheduling and break timings. They should allow employees to work from home or have flexible place of work.

Title: Prediction of Mental Health Problems Among Children Using Machine Learning Techniques

Author: MS. SUMATHI M.R, DR. B. POORNA

**Year : ** 2016

Early diagnosis of mental health problems helps the professionals to treat it at an earlier stage and improves the patients' quality of life. So, there is an urgent need to treat basic mental health problems that prevail among children which may lead to complicated problems, if not treated at an early stage. Machine learning Techniques are currently well suited for analyzing medical data and diagnosing the problem. This research has identified eight machine learning techniques and has compared their performances on different measures of accuracy in diagnosing five basic mental health problems.

A data set consisting of sixty cases is collected for training and testing the performance of the techniques. Twenty-five attributes have been identified as important for diagnosing the problem from the documents. The attributes have been reduced by applying Feature Selection algorithms over the full attribute data set. The accuracy over the full attribute set and selected attribute set on various machine learning techniques have been compared. It is evident from the results that the three classifiers viz., Multilayer Perceptron, Multiclass Classifier and LAD Tree produced more accurate results and there is only a slight difference between their performances over full attribute set and selected attribute set Nowadays, a number of expert systems are utilized in medical domain to predict diseases accurately at an early stage so that treatment can be made effectively and efficiently. Also, expert systems are developed in the mental health domain to predict the mental health problem at an earlier stage. As a number of machine learning techniques are available to construct expert systems, it is necessary to compare them and identify the best that suits the domain of interest.

The research has compared eight machine learning techniques (classifiers) on classifying the dataset to different mental health problems. It is evident from the results that the three classifiers viz., Multilayer Perception, Multiclass Classifier and LAD Tree produce more accurate results than the others. The data set is very minimal and in future, the research may be applied for a large data set to obtain more accuracy. The classifiers need to be trained prior to the implementation of any technique in real prediction.

**Title : **Artificial Intelligence for Mental Health and Mental Illnesses: an Overview

Author: SARAH GRAHAM & COLIN DEPP & ELLEN E. LEE & CAMILLE NEBEKER & XIN TU HO-CHEOL KIM & DILIP V. JESTE

**Year : **2019

Artificial intelligence (AI) technology holds both great promise to transform mental health care and potential pitfalls. This article provides an overview of AI and current applications in healthcare, a review of recent original research on AI specific to mental health, and a discussion of how AI can supplement clinical practice while considering its current limitations, areas needing additional research, and ethical implications regarding AI technology. Recent Findings We reviewed 28 studies of AI and mental health that used electronic health records (EHRs), mood rating scales, brain imaging data, novel monitoring systems (e.g., smartphone, video), and social media platforms to predict, classify, or subgroup mental health illnesses including depression, schizophrenia or other psychiatric illnesses, and suicide ideation and attempts.

Collectively, these studies revealed high accuracies and provided excellent examples of AI's potential in mental healthcare, but most should be considered early proof-of-concept works demonstrating the potential of using machine learning (ML) algorithms to address mental health questions, and which types of algorithms yield the best performance. Summary As AI techniques continue to be refined and improved, it will be possible to help mental health practitioners re-define mental illnesses more objectively than currently done in the DSM-5, identify these illnesses at an earlier or prodromal stage when interventions may be more effective, and personalize treatments based on an individual's unique characteristics. However, caution is necessary in order to avoid over- interpreting preliminary results, and more work is required to bridge the gap between AI in mental health research and clinical care.

METHODOLOGY

CHAPTER-5

METHODOLOGY

5.1 OBJECTIVES

The goal is to develop a machine learning model for mental health (Depression) prediction and potentially replace the updatable supervised machine learning classification models by predicting results in the form of best accuracy by comparing supervised algorithm.

5.2 PROJECT GOALS

Exploration data analysis of variable identification:

- Loading the given dataset
- Import required libraries packages
- Analyze the general properties
- Find duplicate and missing values
- Checking unique and count values

Uni- variate data analysis:

- Rename, add data and drop the data
- To specify data type

Exploration data analysis of bi-variate and multi-variate:

- Plot diagram of pair plot, heat map, bar chart and Histogram

Method of Outlier detection with feature engineering:

- Pre-processing the given dataset
- Splitting the test and training dataset Comparing the Decision tree and Logistic regression model and random forest etc.

Comparing algorithm to predict the result:

- Based on the best accuracy.

5.3 SCOPE OF THE PROJECT

To detect the Mental Health Prediction, this is a classic text classification problem with a help of machine learning algorithm.

5.4 LIST OF MODULES

- Data Analysis of Visualization
- Data Pre-processing
- Comparing Algorithm with prediction in the form of best accuracy result

Deployment Using Flask.

FEASIBILITY STUDY

CHAPTER-6

6. FEASIBILITY STUDY

6.1 DATA WRANGLING

Quality control, open data and investigate and tidy up the data given. Deal with the record cautiously and settle on certain that the tidiness choice is legitimized

6.2DATA COLLECTION

Information was gathered together to anticipate the data given and partitioned into preparation phases and tests. Ordinarily the preparation and testing program is partitioned into 7: 3. Data model is created utilizing standard woodland, hardware, tree choice calculations, and Vector Classifier (SVC), and the trials depend on test results.

6.3 PREPROCESSING

At times, the data gathered might be fragmented, prompting irregularities. The data should initially be handled to work on the exhibition of the calculation.

6.4 BUILDING THE CLASSIFICATION MODEL

A high accuracy predicting model is effective at detecting outliers and irrelevant variables. It is also good at preprocessing a variety of continuous, categorical, and discrete variables. This produces an out-of-bag estimation error and is an easy-to- tune algorithm that has proven to be unbiased in many tests.

6.5 PREDICTIVE MODEL CONSTRUCTION

Data collection and lots of past data is required for machine learning. The data collection contains enough historical data and raw data. The raw data cannot be used without preprocessing the data. It is used for post-processing, sort of an algorithm with the model. Train and test this model to work and predict well with minimal error. The model is adjusted relative to time with improved accuracy.

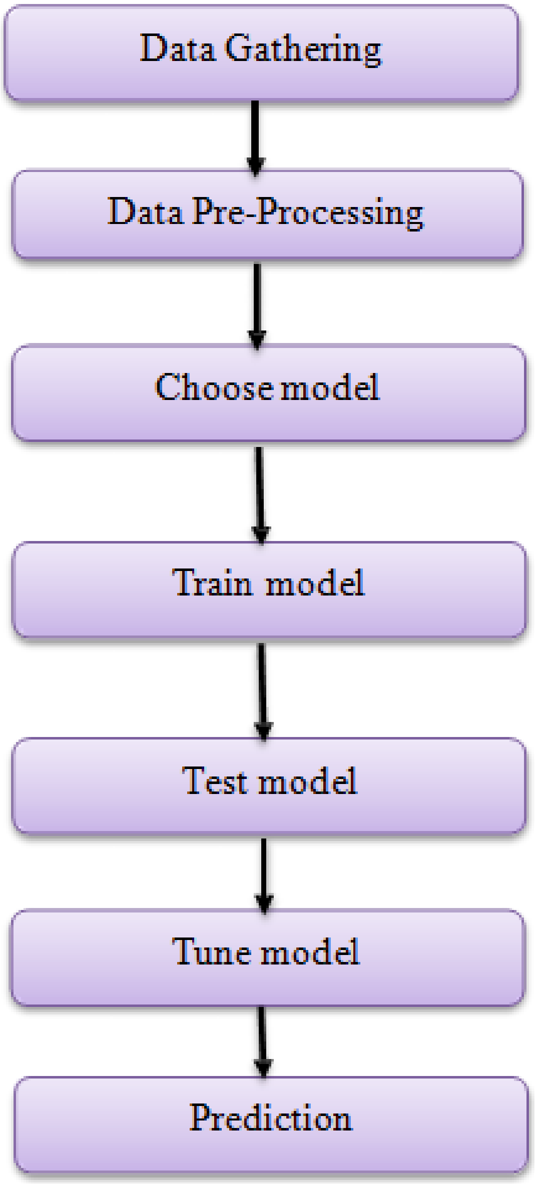


FIG3. PROCESS OF DATAFLOW DIAGRAM

PROJECT REQUIREMENTS

CHAPTER-7

PROJECT REQUIREMENTS

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Technical requirements
    A. Hardware requirements
    B. Software requirements

7.1 FUNCTIONAL REQUIREMENTS

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

7.2 NON-FUNCTIONAL REQUIREMENTS

Process of functional steps:

I. Problem define

II. Preparing data

III. Evaluating algorithms

IV. Improving results

V. Prediction the result

7.3 TECHNICAL REQUIREMENTS

SOFTWARE REQUIREMENTS:

Operating System	: Windows
Tool	: Anaconda with Jupyter Notebook

HARDWARE REQUIREMENTS:

Processor	: Pentium IV/III
Hard disk	: minimum 80 GB
RAM	: minimum 2 GB

PROJECT REQUIREMENTS

CHAPTER - 8

8. SOFTWARE DESCRIPTION

8.1 ANACONDA

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, Machine Learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system Conda. Anaconda distribution comes with more than 1,400 packages as well as the Conda package and virtual environment manager called Anaconda Navigator and it eliminates the need to learn to install each library independently. The open source packages can be individually installed from the Anaconda repository with the conda install command or using the pip install command that is installed with Anaconda.

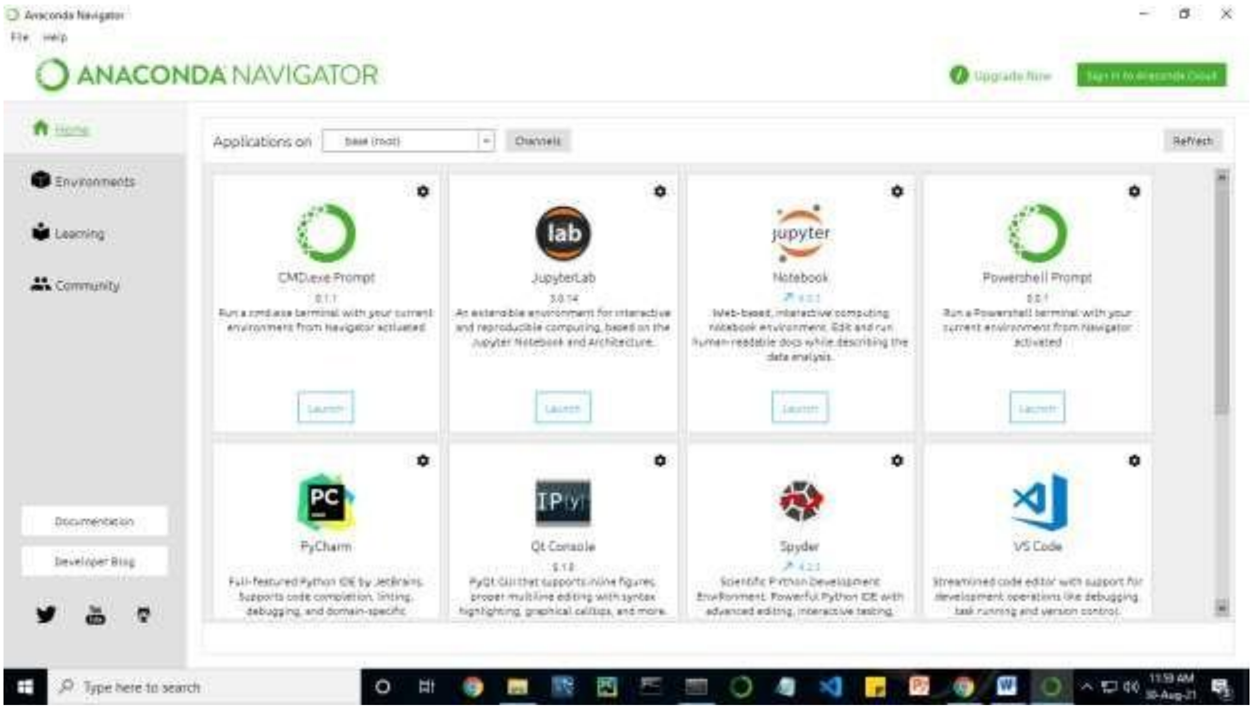
Pip packages provide many of the features of conda packages and in most cases they can work together. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, new environments can be created that include any version of Python packaged with conda.

8.1.1 ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. Anaconda is created by Continuum Analytics, and it is a Python distribution that comes preinstalled with lots of useful python libraries for data science. In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator. The following applications are available by default in Navigator:

- □ JupyterLab
- □ Jupyter Notebook
- □ Spyder
- □ PyCharm
- □ VSCode
- □ Glueviz
- □ Orange 3 App
- □ Rstudio
- □ AnacondaPrompt(Windowsonly)
- □ AnacondaPowerShell(Windowsonly)



CMD.exe Prompt: Run a cmd.exe terminal with your current environment from Navigator activated

JupyterLab: An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook: Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Powershell Prompt: Run a Powershell terminal with your current environment from Navigator activated

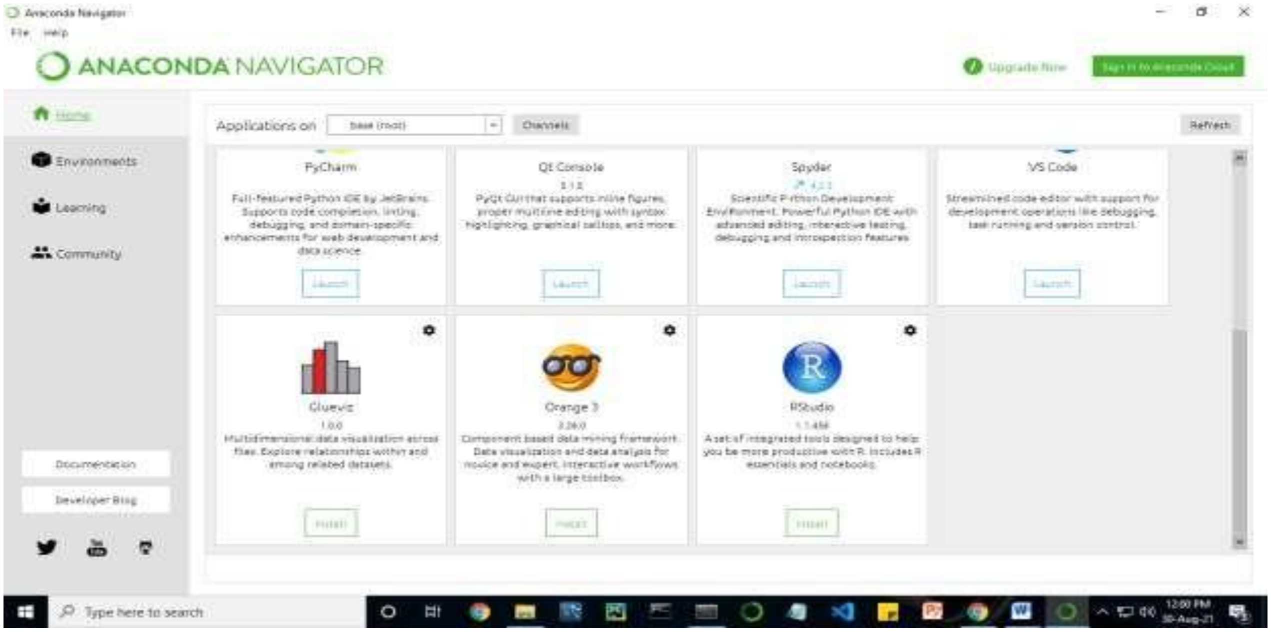
PyCharm: Full-featured Python IDE by JetBrains. Supports code completion, linting, debugging, and domain-specific

Qt Console: PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder: Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing,

VS Code: Streamlined code editor with support for development operations like debugging, task running and version control.

FIG4. ANACONDA NAVIGATOR (1)



Full-featured Python IDE by JetBrains. Supports code completion, linting, debugging, and domain-specific enhancements for web development and data science.

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Streamlined code editor with support for development operations like debugging, task running and version control.

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

FIG.5. ANACONDA NAVIGATOR (2)

8.1.2 CONDA:

Conda is an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs, and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multi- language projects. The Conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository. Anaconda is freely available, open source distribution of python and R programming languages which is used for scientific computations.

8.2 JUPYTER NOTEBOOK

This is the best place for you. It consists of much software which will help you to build your machine learning project and deep learning project. These software have great graphical user interface and can also use it to run your python script. These are the software carried by anaconda navigator. This website acts as meta documentation for the Jupyter ecosystem. It has a collection of resources to navigate the tools and communities in this ecosystem, and to help you get started.

Project Jupyter is a project and community whose goal is to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages. It was spun off from I python in 2014 by Fernando Perez. Notebook documents are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc…). Notebook documents are both human.

Installation:

The easiest way to install the Jupyter Notebook App is installing a scientific python distribution which also includes scientific python packages. The most common distribution is called Anaconda.

Running & Launching Jupyter Notebook App:

The Jupyter Notebook_App can be launched by clicking on the Jupyter Notebook icon installed by Anaconda in the start menu (Windows) or by typing in a terminal (cmd on Windows): Jupyter Notebook. This will launch a new browser window (or a new tab) showing the Notebook Dashboard, a sort of control panel that allows (among other things) to select which notebook to open. When started, the Jupyter_Notebook_App can access only files within its start-up folder (including any sub-folder). No configuration is necessary if you place your notebooks in your home folder or subfolders. Otherwise, you need to choose a Jupyter Notebook App start-up folder which will contain all the notebooks.

Save notebooks:

Modifications to the notebooks are automatically saved every few minutes. To avoid modifying the original notebook, make a copy of the notebook document (menu file -> make a copy…) and save the modifications on the copy.

Executing a notebook:

Download the notebook you want to execute and put it in your notebook folder (or a sub-folder of it).

- ❖ Launch the Jupyter Notebook App
- ❖ In the Notebook Dashboard navigate to find the notebook: clicking on its name will open

it in a new browser tab.

- Click on the menu *Help* -> User Interface Tour for an overview of the Jupyter Notebook App user interface.
- You can run the notebook document step-by-step (one cell a time) by pressing shift + enter.

File Extension:

An **IPYNB** file is a notebook document created by Jupyter Notebook, an interactive computational environment that helps scientists manipulate and analyze data using Python.

8.2.1 JUPYTER NOTEBOOK APP:

The Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet. In addition to displaying/editing/running notebook documents, the Jupyter Notebook App has a Dashboard (Notebook Dashboard), a control panel showing local files and allowing to open notebook documents or shutting down their kernels.

8.2.2 KERNEL

A notebook kernel is a computational engine that executes the code contained in a Notebook document. The ipython kernel, referenced in this guide, executes python code. Kernels for many other languages exist (official kernels). When a Notebook document is opened, the associated kernel is automatically launched. When the notebook is executed (either cell-by-cell or with menu Cell -> Run All), the kernel performs the computation and produces the results. Depending on the type of computations, the kernel may consume significant CPU and RAM. Note that the RAM is not released until the kernel is shut-down.

Notebook Dashboard:

The Notebook Dashboard is the component which is shown first when you launch Jupyter Notebook App. The Notebook Dashboard is mainly used to open notebook documents, and to manage the running kernels (visualize and shutdown). The Notebook Dashboard has other features similar to a file manager, namely navigating folders and renaming/deleting files.

8.3 WORKING PROCESS:

- Download and install anaconda and get the most useful package for machine learning in Python.
- Load a dataset and understand its structure using statistical summaries and data visualization.
- Machine learning models, pick the best and build confidence that the accuracy is reliable.
- Load a dataset and understand its structure using statistical summaries and data visualization.
- Machine learning models, pick the best and build confidence that the accuracy is reliable.

8.4 PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significantindentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a ―batteries included‖ language due to its comprehensive standard library.

SYSTEM ARCHITECTURE

CHAPTER - 9

9. SYSTEM ARCHITECTURE

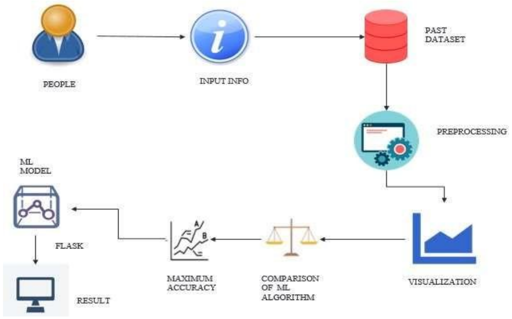


FIG6. SYSTEM ARCHITECTURE

9.1 WORK FLOW DIAGRAM

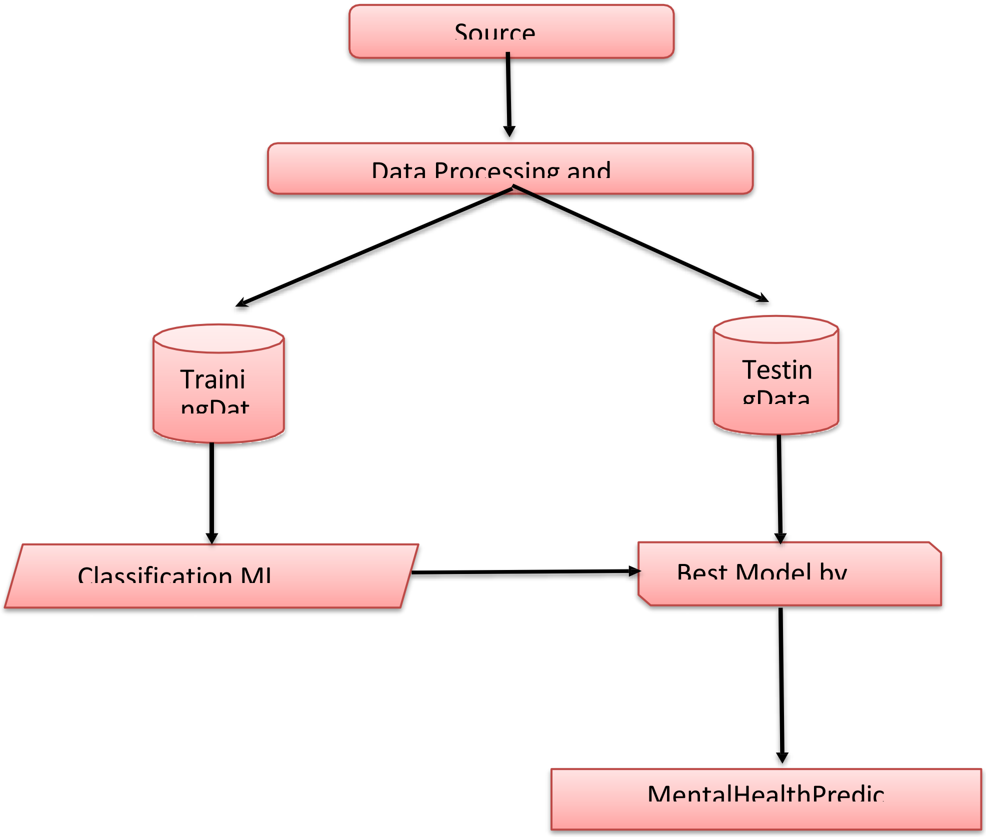


FIG7. WORKFLOW DIAGRAM

MODULE DESCRIPTION

CHAPTER - 10

10. MODULE DESCRIPTION

10.1 DATA PRE-PROCESSING:

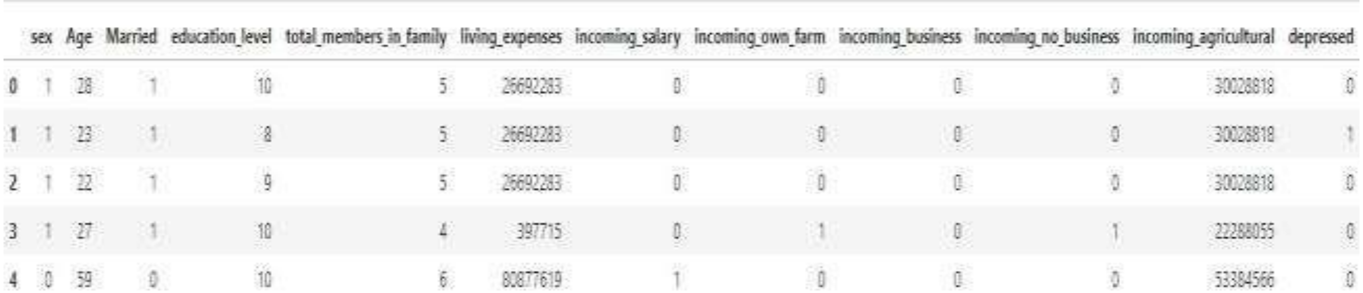
The confirmation methods used to identify AI blunder levels (ML) during AI can be viewed as near mistake estimations. On a fluke, how much data is ample enough to address the populace, approval might not be required for procedures. Issues emerge when working with media models that don't have to address the informational index. The information model used to give an unreasonable example of the model is reasonable for informational collection preparation when joining hyper parameters to show fragmented qualities and two qualities.

A number of different data cleaning tasks using Python’s Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling. Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It’s important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it’s important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

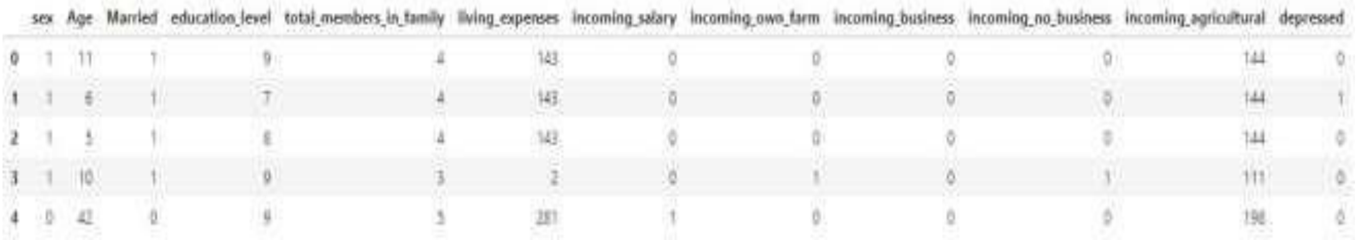
VARIABLE IDENTIFICATION WITH UNI-VARIATE, BI-VARIATE AND MULTI-VARIATE ANALYSIS:

- □ Import libraries for access and functional purpose and read the given dataset
- □ General Properties of Analyzing the given dataset
- □ Display the given dataset in the form of data frame
- □ Show columns
- □ Shape of the data frame
- □ To describe the data frame
- □ Checking Missing values of data frame
- □ Checking unique values of data frame
- □ Checking count values of data frame
- □ Rename and drop the given data frame
- □ To specify the type of values
- □ To create extra columns



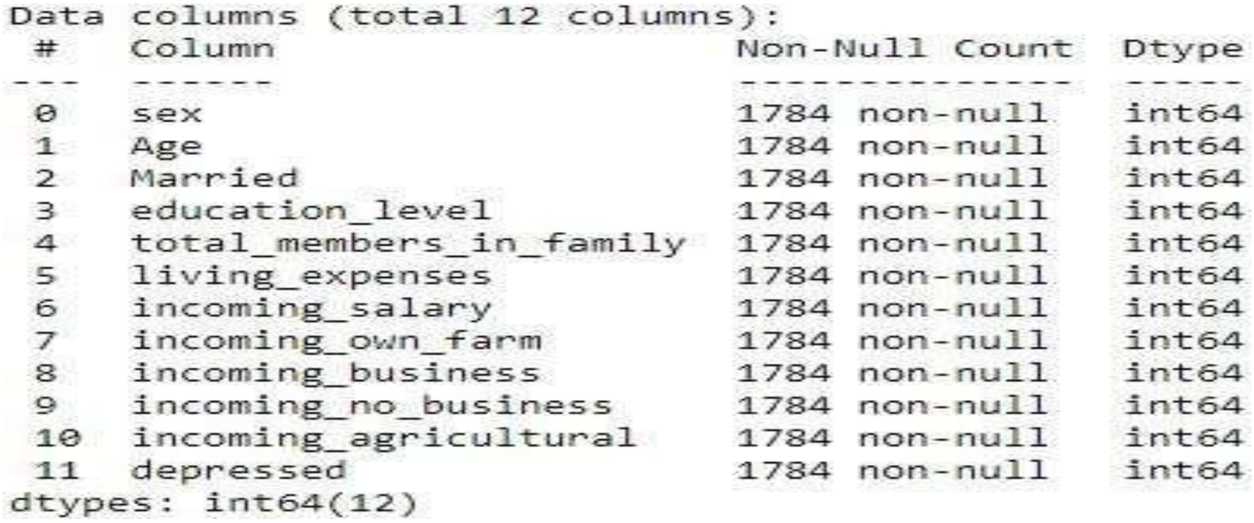
	sex	Age	Married	education_level	total_members_in_family	living_expenses	incoming_salary	incoming_own_farm	incoming_business	incoming_no_business	incoming_agricultural	depressed
0	1	28	1	10	5	26692283	0	0	0	0	30028818	0
1	1	23	1	8	5	26692283	0	0	0	0	30028818	1
2	1	22	1	9	5	26692283	0	0	0	0	30028818	0
3	1	27	1	10	4	397715	0	1	0	1	22288055	0
4	0	59	0	10	6	80877619	1	0	0	0	53384566	0

FIG8. BEFORE PREPROCESSING



	sex	Age	Married	education_level	total_members_in_family	living_expenses	incoming_salary	incoming_own_farm	incoming_business	incoming_no_business	incoming_agricultural	depressed
0	1	11	1	9	4	143	0	0	0	0	144	0
1	1	6	1	7	4	143	0	0	0	0	144	1
2	1	5	1	8	4	143	0	0	0	0	144	0
3	1	10	1	9	3	2	0	1	0	1	111	0
4	0	42	0	9	5	281	1	0	0	0	198	0

FIG9. AFTER PREPROCESSING



```
Data columns (total 12 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   sex                      1784 non-null   int64
 1   Age                      1784 non-null   int64
 2   Married                  1784 non-null   int64
 3   education_level          1784 non-null   int64
 4   total_members_in_family  1784 non-null   int64
 5   living_expenses          1784 non-null   int64
 6   incoming_salary          1784 non-null   int64
 7   incoming_own_farm        1784 non-null   int64
 8   incoming_business        1784 non-null   int64
 9   incoming_no_business     1784 non-null   int64
 10  incoming_agricultural    1784 non-null   int64
 11  depressed                1784 non-null   int64
dtypes: int64(12)
```

FIG10. DATA TYPE IDENTIFICATION

MODULE DIAGRAM

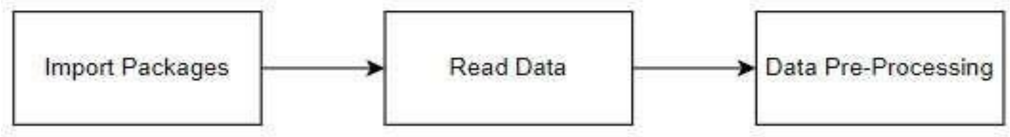


FIG11. MODULE (1)

GIVEN INPUT EXPECTED OUTPUT

Input: Data

Output: Removing noisy data

10.2 EXPLORATION DATA ANALYSIS OF VISUALIZATION

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- □ How to chart time series data with line plots and categorical quantities with bar charts.
- □ How to summarize data distributions with histograms and box plots.

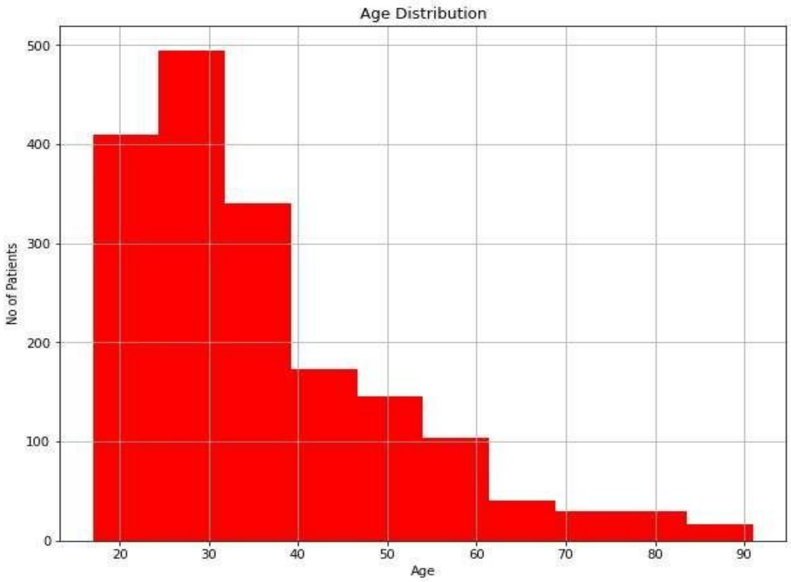


FIG12. AGE DISTRIBUTION

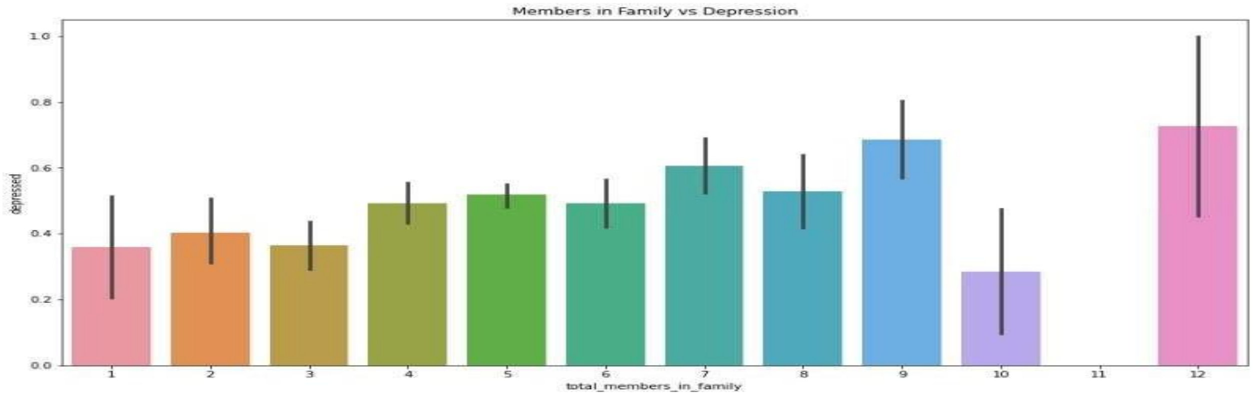


FIG13. MEMBERS IN FAMILY V/S DEPRESSION

MODULE DIAGRAM:

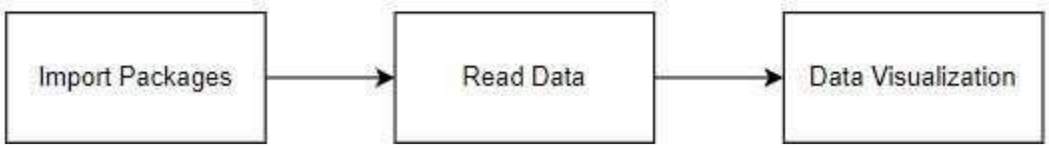


FIG14. MODULE (2)

GIVEN INPUT EXPECTED OUTPUT:

Input : Data

Output : Visualized data

10.3 COMPARING ALGORITHM WITH PREDICTION IN THE FORM OF BEST ACCURACY RESULT:

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resembling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 4 different algorithms are compared:

- Logistic Regression
- Random Forest
- Decision Tree Classifier
- Naive Bayes

The K-fold cross validation procedure is used to evaluate each algorithm, importantly configured with the same random seed to ensure that the same splits to the training data are performed and that each algorithm is evaluated in precisely the same way. Before that comparing algorithm, Building a Machine Learning Model using install Scikit-Learn libraries. In this library package have to done preprocessing, linear model with logistic regression method, cross validating by KFold method, ensemble with random forest method and tree with decision tree Classifier. Additionally, splitting the train set and test set. To predicting the result by comparing accuracy.

10.4 PREDICTION RESULT BY ACCURACY

Equation that predicts a value with independent predictors. It requires classification as variable data to predict a value anywhere between negative infinity and positive infinity**.**

True Positive Rate (TPR) = TP / (TP + FN) False Positive Rate (FPR) = FP / (FP + TN)

Accuracy: It is a simple ratio of precisely predicted results to all observations. Accuracy is a sizeable evaluation as long as false positives and false negatives are approximately equal**.**

Accuracy calculation:

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

The simple ratio of correctly predicted observations to all observations. It is a great measure as long as false positives and negatives are approximately equal.

Precision: Precisely predicted positive results to the grand predicted positive results.

Precision = TP / (TP + FP)

Recall: Proportion of positive observed values correctly predicted. (The model of genuine contributors will be a decent indicator)

Recall = TP / (TP + FN)

F1 score: This score is based on a weighted average of Precision and Recall. It is affected by both false positives and false negatives.

General Formula:

F- Measure = 2TP / (2TP + FP + FN)

F1-Score Formula:

F1 Score = 2*(Recall * Precision) / (Recall + Precision).

10.5 ALGORITHM AND TECHNIQUE EXPLANATION

Classification is a ML and statistics method based on supervised learning, where the data input is learnt by the computer program and then applies its new knowledge to identify new observations. The data set may either be bi-class (like identifying whether the person is male or female) or multi-class too. Labeled data is used by algorithms under supervised learning. Using the labeled data as a starting point, the algorithm determines which label to give to unlabeled data based on its patterns.

USED PYTHON PACKAGES:

sklearn:

- A package which includes ML algorithms. Modules used are train_test_split, Decision Tree Classifier or Logistic Regression and accuracy_score.

NumPy:

- Fast Numeric Python module for quick calculations.
- For reading and manipulating numpy arrays.

Matplotlib:

- Data visualization helps with identifying patterns in a dataset.
- Data frames help with easy data manipulation**.**

Pandas:

- To read and write different files.

10.5.1 LOGILSTIC REGRESSION

It is a ML calculation used to unwrap the conceivable outcomes of the progress stage. Strategically talking, the factors depend on two factors that encode data like 1 (yes) or 0 (no). At the end of the day, the inversion model shows that P (Y = 1) is a component of X.

In other words, the logistic regression model predicts p(y=1) as a function of x. logistic regression assumptions:

□ Binary logistic regression requires the dependent variable to be binary.

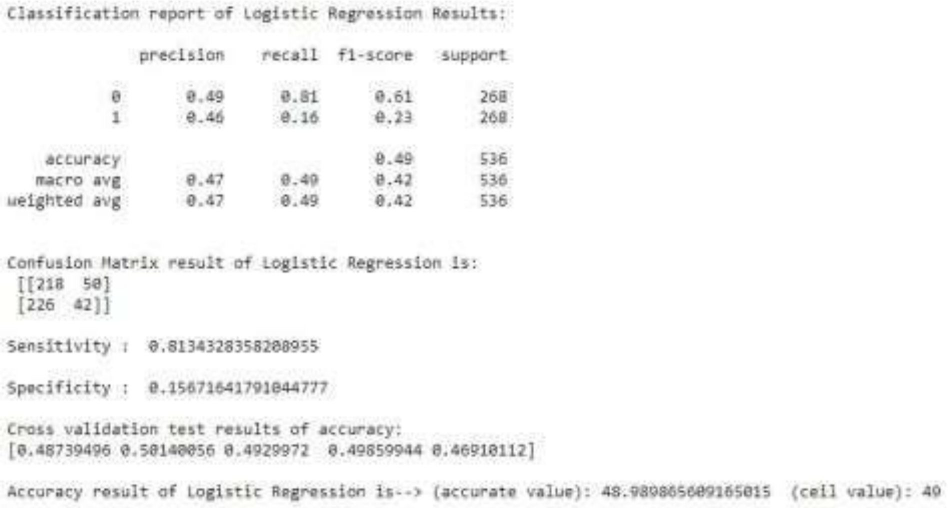
□ For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.

□ Only the meaningful variables should be included.

□ The independent variables should be independent of each other. That is, the model should have little.

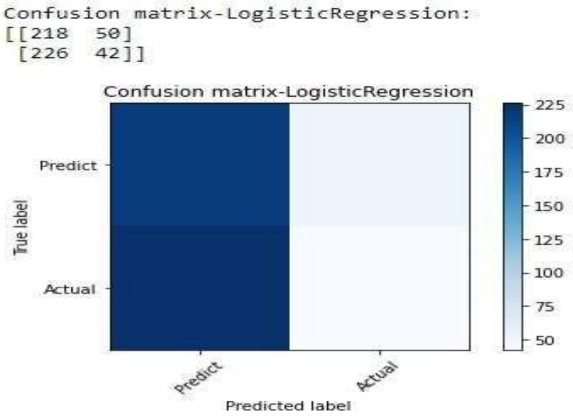
□ The independent variables are linearly related to the log odds.

□ Logistic regression requires quite large sample sizes.



```
Classification report of Logistic Regression Results:

              precision    recall  f1-score   support

           0       0.49      0.81      0.61       268
           1       0.46      0.16      0.23       268

    accuracy                           0.49       536
   macro avg       0.47      0.49      0.42       536
weighted avg       0.47      0.49      0.42       536


Confusion Matrix result of Logistic Regression is:
 [[218  50]
 [226  42]]

Sensitivity :  0.8134328358208955

Specificity :  0.15671641791044777

Cross validation test results of accuracy:
[0.48739496 0.50140056 0.4929972  0.49859944 0.46910112]

Accuracy result of Logistic Regression is--> (accurate value): 48.98986560916501 5  (ceil value): 49
```

FIG15. CLASSIFICATION REPORT OF LOGISTIC REGRESSION



```
Confusion matrix-LogisticRegression:
[[218  50]
 [226  42]]
```

FIG16. CONFUSION MATRIX OF LOGISTIC REGRESSION

MODULE DIAGRAM

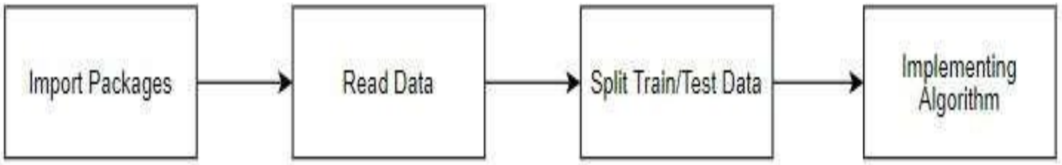


FIG17. MODULE (3)

GIVEN INPUT EXPECTED OUTPUT

Input: Data

Output: Getting accuracy

10.5.2 RANDOM FOREST CLASSIFIER

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

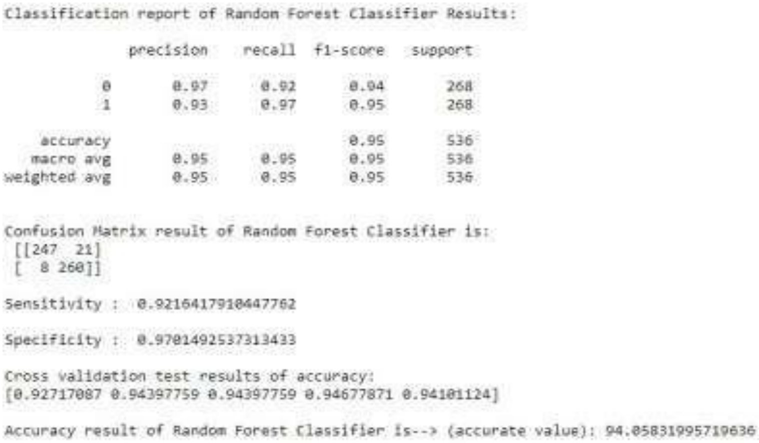
- □ Pick N random records from the dataset.
- □ Build a decision tree based on these N records.
- □ Choose the number of trees you want in your algorithm and repeat steps 1&2.

In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote**.**

GIVEN INPUT EXPECTED OUTPUT

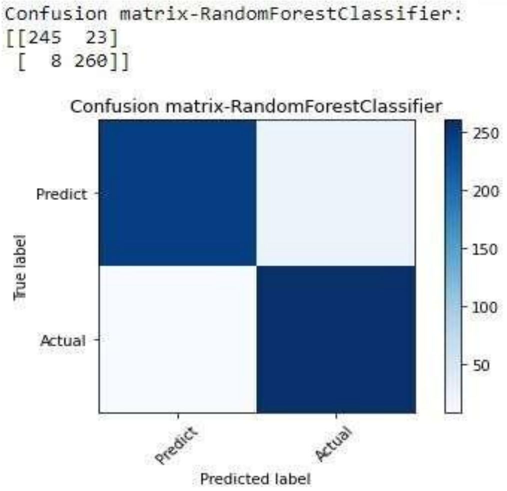
Input: Data

Output: Getting accuracy



```
Classification report of Random Forest Classifier Results:

              precision    recall  f1-score   support

           0       0.97      0.92      0.94       268
           1       0.93      0.97      0.95       268

    accuracy                           0.95       536
   macro avg       0.95      0.95      0.95       536
weighted avg       0.95      0.95      0.95       536

Confusion Matrix result of Random Forest Classifier is:
 [[247  21]
 [  8 260]]

Sensitivity :  0.9216417910447762

Specificity :  0.9701492537313433

Cross validation test results of accuracy:
[0.92717087 0.94397759 0.94397759 0.94677871 0.94101124]

Accuracy result of Random Forest Classifier is--> (accurate value): 94.05831995719636
```

FIG18. CLASSIFICATION REPORT OF RANDOM FOREST



```
Confusion matrix-RandomForestClassifier:
[[245  23]
 [  8 260]]
```

FIG19. CONFUSION MATRIX OF RANDOM FOREST

MODULE DIAGRAM

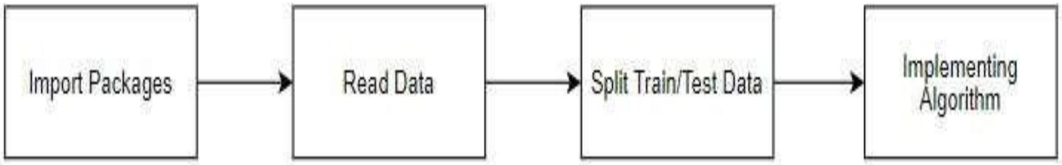


FIG20. MODULE (4)

CONCLUSION AND FUTURE WORK

CHAPTER - 11

11. CONCLUSION AND FUTURE WORK

CONCLUSION

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score will be find out. This application can help to find the Prediction of mental health**.**

FUTURE WORK

Mental health prediction to connect with cloud model and to optimize the work to implement in Artificial Intelligence environment.

APPENDICES

CHAPTER - 12

12. APPENDICES

A. SAMPLE CODE

Module – 1

Pre-Processing

```
#import library packages import pandas as pd import numpy as np
import warnings warnings.filterwarnings("ignore")
#Load given dataset
data = pd.read_csv("depressed.csv")
Before drop the given dataset:
data.head()
#shape
data.shape
After drop the given dataset:
df = data.dropna() df.head()
#shape
df.shape
#columns
df.columns
#To describe the dataframe
df.describe()
#Checking datatype and information about dataset
df.info()
Checking duplicate values of dataframe df.sex.unique() df.Age.unique()
df.Married.unique() df.education_level.unique() df.incoming_agricultural.unique()
df.total_members_in_family.unique() df.incoming_own_farm.unique()
print("Age of patient ranges :", sorted(df['Age'].unique()))
```

```
df.depressed.unique()
```

Before Pre_Processing:

```
df.head()
```

After Pre_Processing:

```
from sklearn.preprocessing import LabelEncoder
var = ['sex', 'Age', 'Married', 'education_level', 'total_members_in_family',
'living_expenses', 'incoming_salary', 'incoming_own_farm', 'incoming_business',
'incoming_no_business',
'incoming_agricultural', 'depressed']
```

Module – 2 Visualization

```
#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import warnings warnings.filterwarnings('ignore')
data = pd.read_csv("depressed.csv") df = data.dropna()
df.columns pd.crosstab(df.Married,df.depressed)
#Histogram Plot of Age distribution df["Age"].hist(figsize=(10,8), color="red") plt.title("Age
Distribution") plt.xlabel("Age")
family", y = "depressed", ax=ax, data=df)
plt.title("Members in Family vs Depression")
#Propagation by variable
def PropByVar(df, variable):
dataframe_pie = df[variable].value_counts()
ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize = 12)
ax.set_title(variable + ' \n', fontsize = 15)
return np.round(dataframe_pie/df.shape[0]*100,2) PropByVar(df, 'education_level')
```

```
fig, ax = plt.subplots(figsize=(15,6)) sns.boxplot(df.Age, ax =ax) plt.title("Age distribution")
plt.show()
sns.pairplot(df) plt.show()
fig, ax = plt.subplots(figsize=(15,6))
sns.violinplot(y = df['Age'], x = df['depressed'], ax=ax) plt.title("Depressed Persons and their age")
plt.show()
# Heatmap plot diagram
fig, ax = plt.subplots(figsize=(15,10)) sns.heatmap(df.corr(), ax=ax, annot=True)
```

Spliting Train/Test:

```
#preprocessing, split test and dataset, split response variable
X = df.drop(labels='depressed', axis=1)
def qul_No_qul_bar_plot(df, bygroup): dataframe_by_Group = pd.crosstab(df[bygroup],
columns=df["depressed"], normalize = 'index')
dataframe_by_Group = np.round((dataframe_by_Group * 100), decimals=2) ax =
dataframe_by_Group.plot.bar(figsize=(15,7));
vals = ax.get_yticks() ax.set_yticklabels(['{:3.0f}%'.format(x) for x in vals]);
ax.set_xticklabels(dataframe_by_Group.index,rotation = 0, fontsize =
15);
ax.set_title('Depression or not by given attributes (%) (by ' +
dataframe_by_Group.index.name + ')\n', fontsize = 15)
ax.set_xlabel(dataframe_by_Group.index.name, fontsize = 12) ax.set_ylabel('(%)', fontsize = 12)
ax.legend(loc = 'upper left',bbox_to_anchor=(1.0,1.0), fontsize= 12) rects = ax.patches
# add data labels
for rect in rects:
height = rect.get_height() ax.text(rect.get_x() + rect.get_width()/2,
height + 2, str(height)+'%', ha='center', va='bottom', fontsize = 12)
```

Module – 3

Logistic Regression

```
#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import warnings warnings.filterwarnings('ignore')
#Load given dataset
data = pd.read_csv("depressed.csv") df=data.dropna()
df.columns
X = df.drop(labels='depressed', axis=1)
#Response variable
y = df.loc[:,'depressed']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)
Logistic Regression :
In [ ]:
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score logR= LogisticRegression()
logR.fit(X_train,y_train)
predictR = logR.predict(X_test)
print("")
print('Classification report of Logistic Regression Results:')
print("") print(classification_report(y_test,predictR))
print("") cm=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Logistic Regression is:\n',cm) print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1]) print('Sensitivity : ', sensitivity ) print("")
```

```
specificity = cm[1,1]/(cm[1,0]+cm[1,1]) print('Specificity : ', specificity) print("")
accuracy = cross_val_score(logR, X, y, scoring='accuracy') print('Cross validation test results of
accuracy:') print(accuracy)
#get the mean of each fold
print("")
LR=accuracy.mean() * 100
print("Accuracy result of Logistic Regression is-->(accuratevalue):",LR," (ceil
value):",math.ceil(LR))
def graph():
import matplotlib.pyplot as plt data=[LR]
alg="Logistic Regression" plt.figure(figsize=(5,5)) b=plt.bar(alg,data,color=("b"))
plt.title("Accuracy comparison of Depression",fontsize=15) plt.legend(b,data,fontsize=9)
graph()
TN = cm[0][0]
FN = cm[1][0]
TP = cm[1][1]
FP = cm[0][1]
print("True Positive :",TP) print("True Negative :",TN) print("False Positive :",FP) print("False
Negative :",FN) print("")
TPR = TP/(TP+FN) TNR = TN/(TN+FP) FPR = FP/(FP+TN) FNR = FN/(TP+FN)
print("True Positive Rate :",TPR) print("True Negative Rate :",TNR) print("False Positive Rate
:",FPR) print("False Negative Rate :",FNR) print("")
PPV = TP/(TP+FP) NPV = TN/(TN+FN)
print("Positive Predictive Value :",PPV) print("Negative predictive value :",NPV)
def plot_confusion_matrix(cm2, title='Confusion matrix- LogisticRegression',
cmap=plt.cm.Blues):
target_names=['Predict','Actual']
plt.imshow(cm2, interpolation='nearest', cmap=cmap) plt.title(title)
plt.colorbar()
```

```
tick_marks = np.arange(len(target_names)) plt.xticks(tick_marks, target_names, rotation=45)
plt.yticks(tick_marks, target_names) plt.tight_layout()
plt.ylabel('True label')
```

Module – 4

Random Forest Algorithm

```
#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import warnings warnings.filterwarnings('ignore')
#Load given dataset
data = pd.read_csv("depressed.csv") df=data.dropna()
df.columns
#According to the cross-validated MCC scores, the random forest is the best-performing model, so
now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,
roc_auc_score
X = df.drop(labels='depressed', axis=1)
#Response variable
y = df.loc[:,'depressed']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)
RandomForestClassifier:
In [ ]:
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score rfc = RandomForestClassifier()
rfc.fit(X_train,y_train)
```

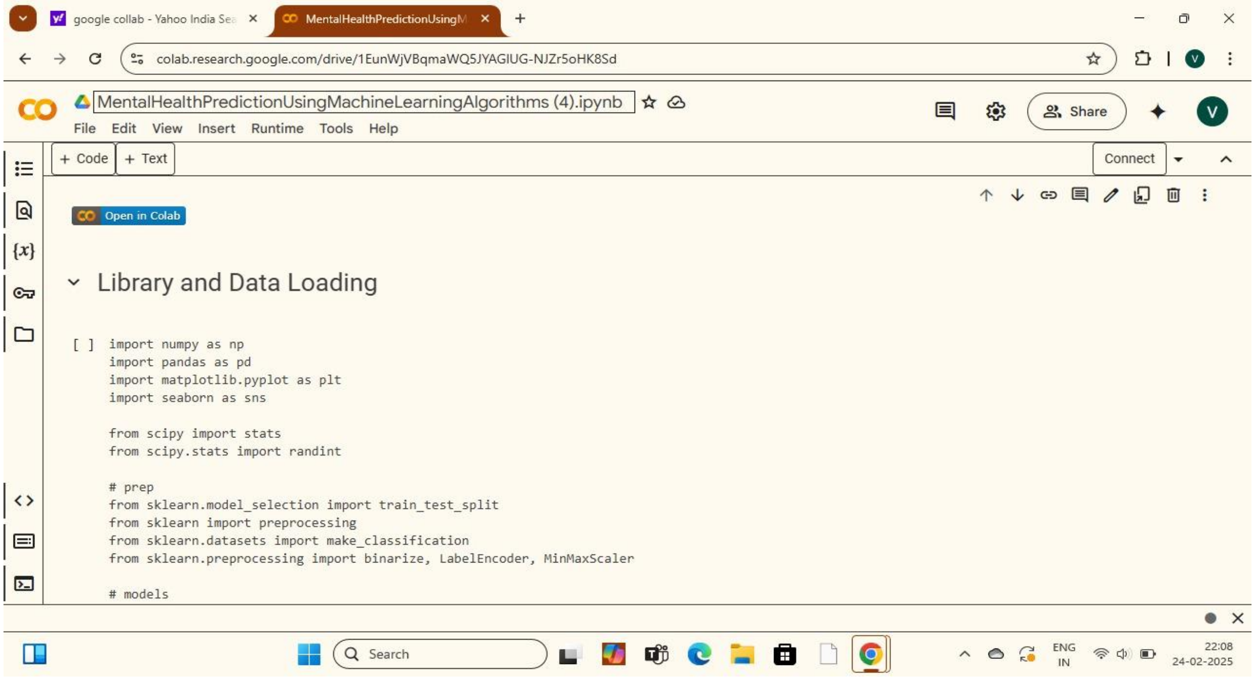
```
predictR = rfc.predict(X_test)
print("")
print('Classification report of Random Forest Classifier Results:')
 print("")
print(classification_report(y_test,predictR))
print("") cm=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Random Forest Classifier is:\n',cm) print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1]) print('Sensitivity : ', sensitivity ) print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1]) print('Specificity : ', specificity) print("")
accuracy = cross_val_score(rfc, X, y, scoring='accuracy') print('Cross validation test results of
accuracy:') print(accuracy)
#get the mean of each fold
print("")
LR=accuracy.mean() * 100
print("Accuracy result of Random Forest is-->(accuratevalue):",LR," (ceil value):",math.ceil(LR))
def graph():
import matplotlib.pyplot as plt data=[LR]
alg="Random Fores tClassifier" plt.figure(figsize=(5,5)) b=plt.bar(alg,data,color=("b"))
plt.title("Accuracy comparison of Depression",fontsize=15) plt.legend(b,data,fontsize=9)
graph()
TP = cm[0][0]
FP = cm[1][0]
FN = cm[1][1]
TN = cm[0][1]
print("True Positive :",TP) print("True Negative :",TN)
```

Module – 5

Flask Deploy

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import joblib
app = Flask(name)
model = joblib.load('RF.pkl')
@app.route('/')
def home():
return render_template('index.html')
@app.route('/predict',methods=['POST'])
def predict(): '''
For rendering results on HTML GUI '''
int_features = [(x) for x in request.form.values()] final_features = [np.array(int_features)]
print(final_features)
prediction = model.predict(final_features) print(prediction)
output = prediction[0]
if output == 1:
return render_template('index.html', prediction_text='Person in depression')
else:
return render_template('index.html', prediction_text='Person not in depression')
print(output)
```

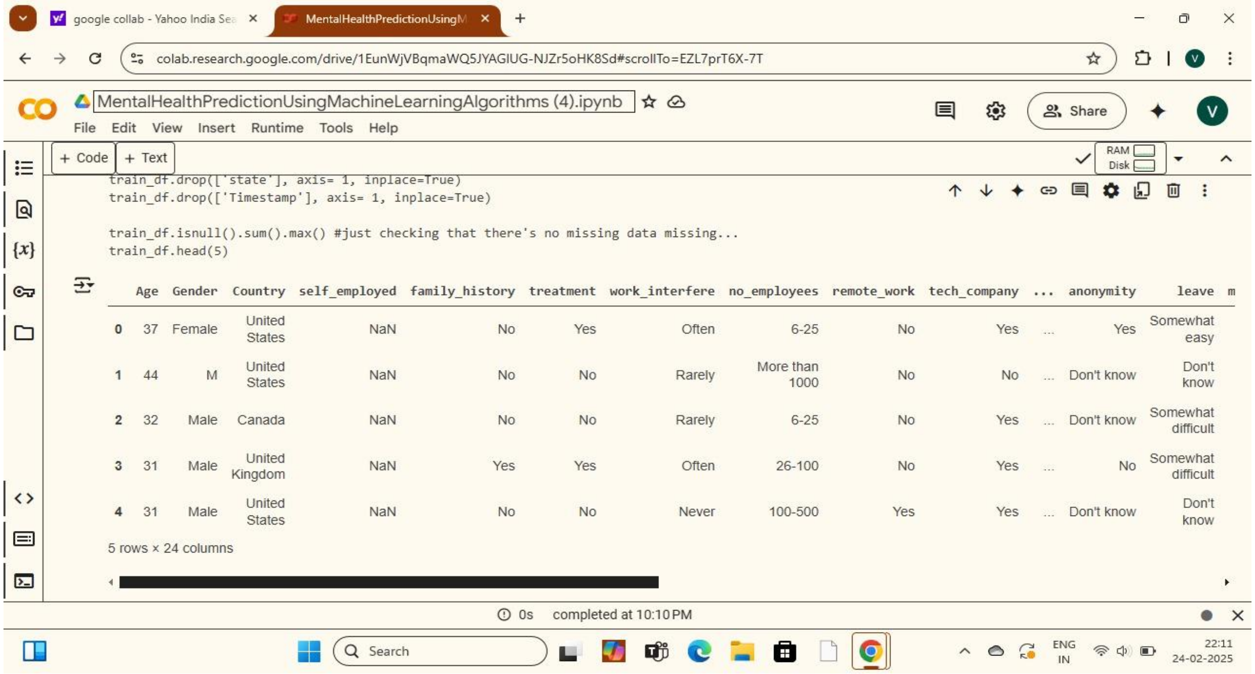
A. SCREENSHOTS



google collab - Yahoo India Sea
MentalHealthPredictionUsingM
colab.research.google.com/drive/1EunWjVBqmaWQ5JYAGIUG-NJZr5oHK8Sd
MentalHealthPredictionUsingMachineLearningAlgorithms (4).ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
Connect
Open in Colab
Search
ENG IN
22:08
24-02-2025

Library and Data Loading

```
[ ]  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
```

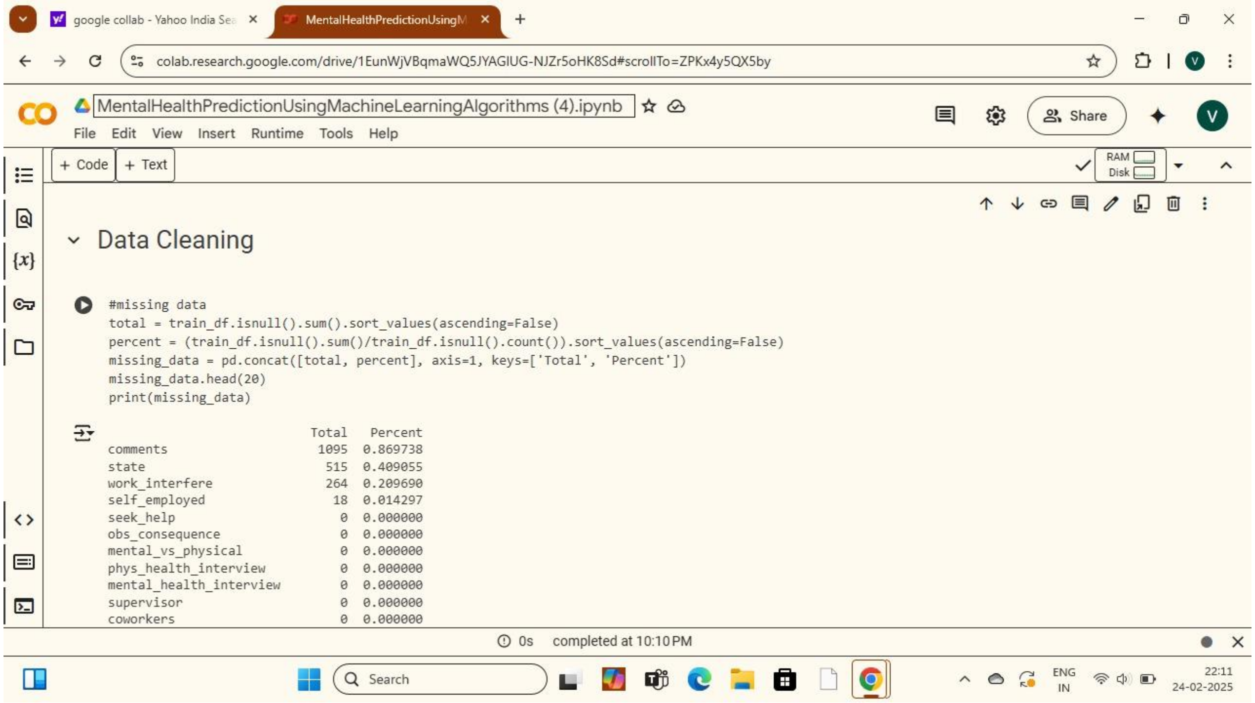


google collab - Yahoo India Sea
MentalHealthPredictionUsingM
colab.research.google.com/drive/1EunWjVBqmaWQ5JYAGIUG-NJZr5oHK8Sd#scrollTo=EZL7prT6X-7T
MentalHealthPredictionUsingMachineLearningAlgorithms (4).ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
RAM
Disk
0s completed at 10:10 PM
Search
ENG IN
22:11
24-02-2025

```
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)
```

	Age	Gender	Country	self_employed	family_history	treatment	work_interfere	no_employees	remote_work	tech_company	...	anonymity	leave	m
0	37	Female	United States	NaN	No	Yes	Often	6-25	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	6-25	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

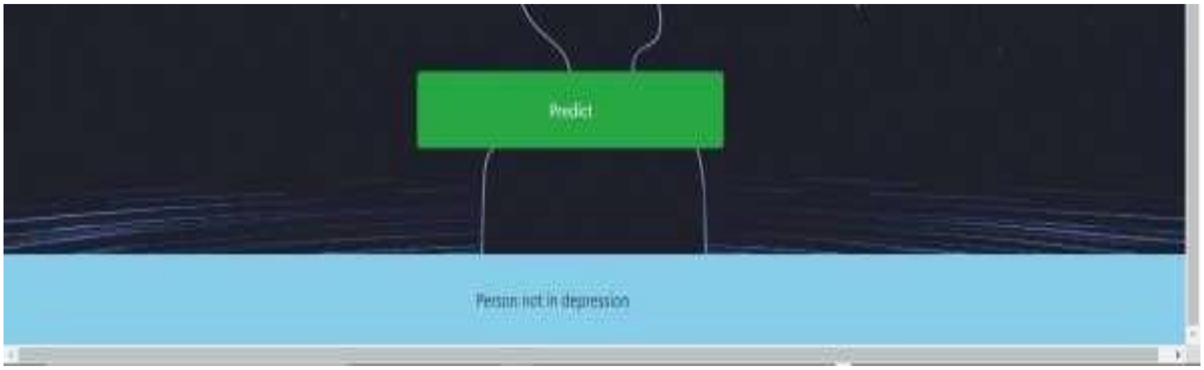


google collab - Yahoo India Se
MentalHealthPredictionUsingM
colab.research.google.com/drive/1EunWjVBqmaWQ5JYAGIUG-NJZr5oHK8Sd#scrollTo=ZPKx4y5QX5by
MentalHealthPredictionUsingMachineLearningAlgorithms (4).ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
RAM
Disk
0s completed at 10:10 PM
Search
ENG
IN
22:11
24-02-2025

Data Cleaning

```
#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)
```

```
                         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
```



Person not in depression

B. REFERENCES

1. R. Kessler, E. Bromet, P. Jonge, V. Shahly, and Marsha., ―The burden of depressive illness,‖ Public Health Perspectives on Depressive Disorders,2017.
2. W. H. Organization, ―Mental health: Fact sheet,‖ https://www.euro.who.int/en/health-topics/noncom municablediseases/mental- health, 2019.
3. M. Renteria-Rodriguez, ―Salud mental en méxico,‖ NOTA-INCyTUNU´ MERO 007, 2018.
4. S. Guntuku, D. Yaden, M. Kern, L. Ungar, and J. Eichstaedt, ―Detecting depression and mental illness on social media: an integrative review,‖ Current Opinion in Behavioral Sciences, 2017.
5. G. Coopersmith, M. Dredze, and C. Harman, ―Quantifying mental health signals in twitter,‖ Workshop on Computational Linguistics and Clinical Psychology, 2014.
6. M. De Choudhury, M. Gamon, S. Counts, and E. Horvitz, ―Predicting depression via social media,‖ In Proceedings of the 7th International AAAI Conference on Weblogs and Social Media, 2013.
7. S. Tsugawa, Y. Kikuchi, F. Kishino, K. Nakajima, Y. Itoh, and H. Ohsaki,―Recognizing depression from twitter activity,‖ In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015.
8. M. Trotzek, S. Koitka, and C. Friedrich, ―Word embeddings and linguistic metadata at the clef 2018 tasks for early detection of depression and anorexia,‖ Proceedings of the 9th International Conference of the CLEF Association, CLEF 2018, Avignon, France, 2018.
9. E. A. R´ıssola, M. Aliannejadi, and F. Crestani, ―Beyond modelling: Understanding mental disorders in online social media,‖ Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, 2020.
10. S. Burdisso, M. Errecalde, and M. Montes-y Go´mez, ―A text classification framework for and effective early depression detection over social media streams,‖ Expert Systems With Applications, Vol. 133, 2019.
11. D. Preotiuc-Pietro, J. Eichstaedt, G. Park, M. Sap, L. Smith, V. Tobolsky, H. Schwartz, and L. Ungar, ―The role of personality, age and gender in tweeting about mental illnesses,‖ In

Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology, 2015.

12. R. Ortega-Mendoza, A. Lopez-Monroy, A. Franco-Arcega, and M. Montes-Y-Go´mez,―Peimex at erisk2018: Emphasizing personal information for depression and anorexia detection,‖ Proceedings of the 9th International Conference of the CLEF Association, CLEF 2018, Avignon, France, 2018.