FINDING PSYCHOLOGICAL INSTABILITY USING MACHINE LEARNING

A Project Report submitted

in partial fulfillment of

the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY
in
INFORMATION TECHNOLOGY

Submitted By

ADDULA SUGANDHI	**:17WJ1A1202**
BONKANPALLY SAHANA	**:17WJ1A1208**
POLA SHALINI	**:17WJ1A1240**
PULIJALA SHIRISHA	**:17WJ1A1242**

Under the Guidance of

Mr. K. ANIL KUMAR

Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS (Autonomous)
(Affiliated to JNTUH, Accredited by NBA)
Ibrahimpatnam, R. R. District, Telangana - 501506.

(2017-2021)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that this project report entitled **"Finding Psychological Instability Using Machine Learning"** submitted by **A. Sugandhi (17WJ1A1202), B. Sahana (17WJ1A1208), P. Shalini (17WJ1A1240), P. Shirisha (17WJ1A1242)** in partial fulfilment for the award of the degree of **Bachelor of Technology** in **Information Technology** prescribed by **Guru Nanak Institutions Technical Campus (Autonomous) affiliated to** Jawaharlal Nehru Technological University, Hyderabad during the academic year **2020-2021**.

Internal Guide	**Project Coordinator**	**HOD**	**Associate Director**
Mr.K.Anil kumar	Mrs.P.Padma	Dr.M.I.Thariq Hussan	Dr.Rishi Sayal

External Examiner

Vision of the Department

To be a premier Information Technology department in the region by providing high quality education.

Mission of the Department

M1: Nurture young individuals into knowledgeable, skillful and ethical professionals in their pursuit of Information Technology.

M2: Transform the students through excellent teaching learning process and sustain high performance by innovations.

M3: Extensive partnerships and collaborations with foreign universities.

M4: Develop industry-interaction for innovation and product development.

DEPARTMENT OF INFORMATION TECHNOLOGY

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: Produce industry ready graduates having the ability to apply academic knowledge across the disciplines and in emerging areas of Information Technology for higher studies, research, employability, product development and handle the realistic problems.

PEO 2: Graduates will have good communication skills, possess ethical conduct, sense of responsibility to serve the society and protect the environment.

PEO 3: Graduates will have excellence in soft skills, managerial skills, leadership qualities and understand the need for lifelong learning for a successful professional career.

PROGRAMME OUTCOMES (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

1. Ability to design, develop, test and debug software applications, evaluate and recognize potential risks and provide innovative solutions.
2. Explore technical knowledge in diverse areas of Information Technology for upliftment of society, successful career, entrepreneurship and higher studies.

ACKNOWLEDGEMENT

We wish to express our sincere thanks to Chairman **Sardar T.S. Kohli** and Vice-Chairman **Sardar G.S.Kohli** of **Guru Nanak Institutions** for providing us all necessary facilities, resources and infrastructure for completing this project work.

We express a whole hearted gratitude to our Managing Director **Dr.H.S. Saini** for providing strong vision in engineering education through accreditation policies under regular training in upgrading education for the faculty members.

We express a whole hearted gratitude to our Director **Dr.M. Ramalinga Reddy** for providing us the constructive platform to launch our ideas in the area of Information Technology and improving our academic standards.

We express a whole hearted gratitude to our Associate Director **Dr. Rishi Sayal** for providing us the conducive environment for carrying through our academic schedules and projects with ease.

We have been truly blessed to have a wonderful advisor **Dr.M.I. Thariq Hussan** HOD-Department of Information Technology for guiding us to explore the ramification of our work and we express our sincere gratitude towards him for leading me throughout the project work.

We specially thank our internal guide **Mr.K.Anil Kumar** Assistant Professor of Information Technology for his valuable suggestions and constant guidance in every stage of the project.

We express our sincere thanks to all the faculties of Information Technology department who helped us in every stage of our project by providing their valuable suggestions and support.

DECLARATION

We **A. Sugandhi (17WJ1A1202), B. Sahana (17WJ1A1208), P. Shalini (17WJ1A1240), P. Shirisha (17WJ1A1242)** hereby declare that the project report entitled "**Finding Psychological Instability Using Machine Learning** " under the esteemed guidance of **Mr.K.Anil Kumar**, Assistant Professor of Information Technology submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Information Technology. This is a record of bona fide work carried out by us and the results embodied in this project report have not been submitted to any other University or Institute for the award of Degree or Diploma.

Date :

Place : GNITC

A.SUGANDHI	**: 17WJ1A1202**
B.SAHANA	**: 17WJ1A1208**
P.SHALINI	**: 17WJ1A1240**
P.SHIRISHA	**: 17WJ1A1242**

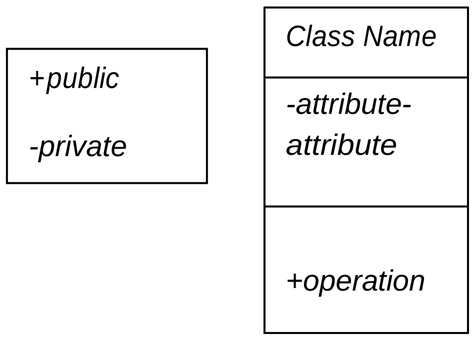
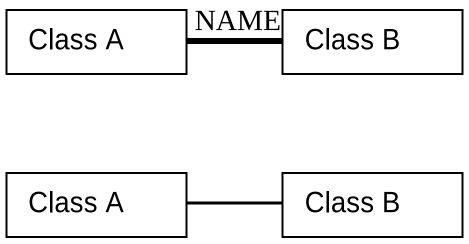
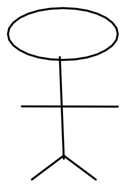
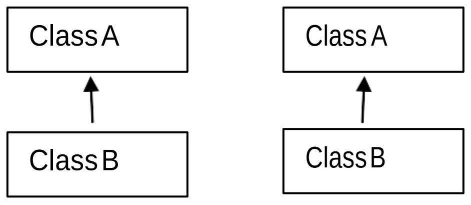
LIST OF CONTENTS

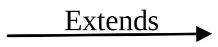
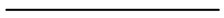
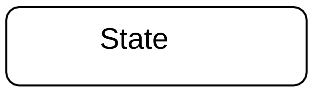
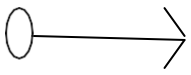
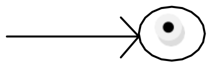
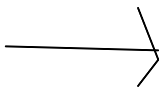
ABSTRACT

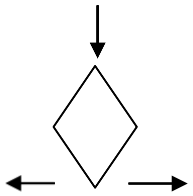
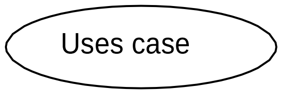
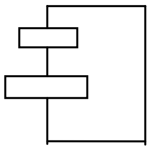
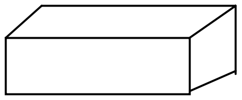
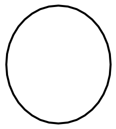
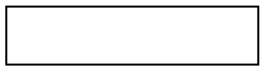
As we know that people around the globe work hard to keep up with this racing world. However, due to this each individual is dealing with different health issues, one of the most known issue is depression or stress which may eventually lead to death or other brutal activities. These abnormalities can be termed as the Bipolar disorder which can be treated by undergoing some treatment suggested by specialists. For this research, data has been collected from working people which comprises of all kinds of questions for despondent detection and the dataset has been run through some machine learning algorithms. Random Forest algorithm gives the highest accuracy as 87.02% compared to the other algorithms.

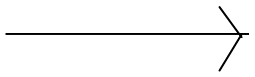
LIST OF FIGURES

LIST OF NOTATIONS

S.NO	NAME	NOTATION	DESCRIPTION
1.	Class	 +public -private Class Name -attribute- attribute +operation	Represents a collection of similar entities grouped together.
2.	Association		Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Aor		It aggregates several classes into a single classes
4.	Aggregation		Interaction between the system and external environment

5.	Relation *(uses)*	Uses	Used for additional process communication.
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the process.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.

12.	Decision box		Represents decision making process from a constraint
13.	Usecase		Interact ion between the system and external environment.
14.	Component		Represents physical modules which is a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.

18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message	Message 	Represents the message exchanged.

LIST OF ABBREVATIONS

S.NO	ABBREVATION	EXPANSION
1.	DB	Data Base
2.	ML	Machine Learning
3.	SL	Supervised Learning
4.	AI	Artificial intelligence
5.	ANNs	Artificial Neural Networks
6.	RF	Random Forest

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Mental health can influence everyday living, relations, and physical health. In any case, this connection additionally works the other way. Factors in individuals' lives, relational associations, and physical variables would all be able to add to mental health disturbances. Caring for mental issues can improve a person’s perspective over life in a positive way. Doing this can help in achieving harmony in life. Conditions, for example, stress, despondency, and nervousness would all be able to influence mental health and disturb an individual's everyday practice.

Despite the fact that the term mental health is in like manner use, numerous conditions that specialists perceive as mental issue have physical roots. Modifiable variables for mental health issue include:

- financial conditions, such whether work is accessible in the neighbourhood
- Occupation
- A person's level of social consideration
- Education
- Living quality
- Non-modifiable variables include:
- Gender
- Age

Mental disorders impact around 25 percent of elders; just about 6 percent are truly disabled and named having real mental sickness.

These disorders are habitually associated with endless physical infirmities, for instance, coronary disease and diabetes.

They in like manner increase the peril of physical injury and going through disasters, severity, and suicides. Suicide alone was at risk for 35,345 deaths in the U.S in 2019 (the latest year for which last data are available), making it the tenth driving explanation behind death. Among adolescents and young adults, suicide is responsible for extra deaths than the blend of harmful development, heart ailment, innate irregularities, respiratory disorder, influenza, , iron deficiency, and kidney and liver disease.

The treatment of mental affliction has been held somewhere around the inclination that disorders of feeling, thinking, and direct somehow need realness and rather reflect particular weakness or poor life choices. Most crisis offices are sick prepared to address the issues of patients amidst mental health emergencies.

Most protection plans see mental ailment and dependence as special cases to standard thought, not part of it.

Regardless of a general social move towards sympathy, our overall population in spite of everything will when all is said in done view the mentally wiped out and those with propensity as morally broken instead of as wiped out.

What is Machine Learning?

Machine Learning is a system of computer algorithms that can learn from example through self-improvement without being explicitly coded by a programmer. Machine learning is a part of artificial Intelligence which combines data with statistical tools to predict an output which can be used to make actionable insights.

The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results. Machine learning is closely related to data mining and Bayesian predictive modelling. The machine receives data as input and uses an algorithm to formulate answers.

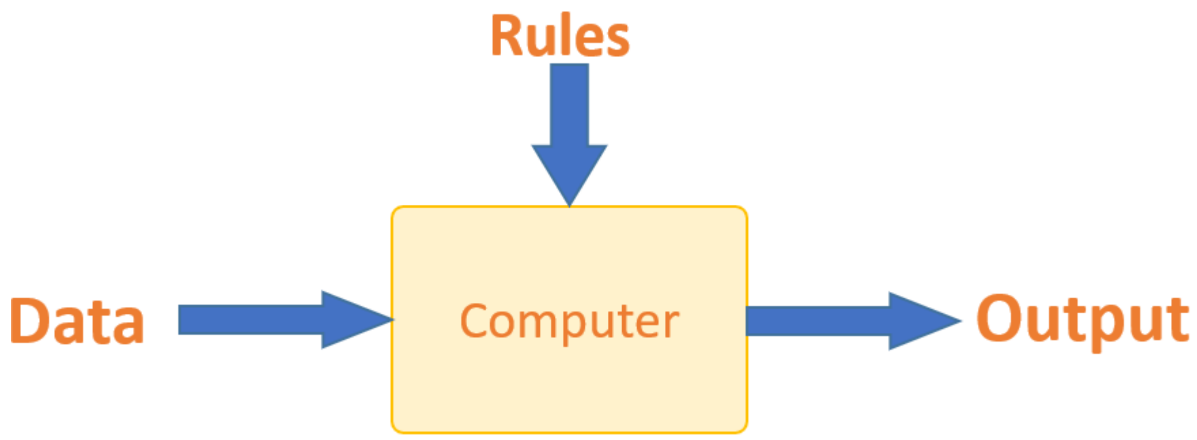
A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data.

Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.
Machine learning is also used for a variety of tasks like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

Machine Learning vs. Traditional Programming

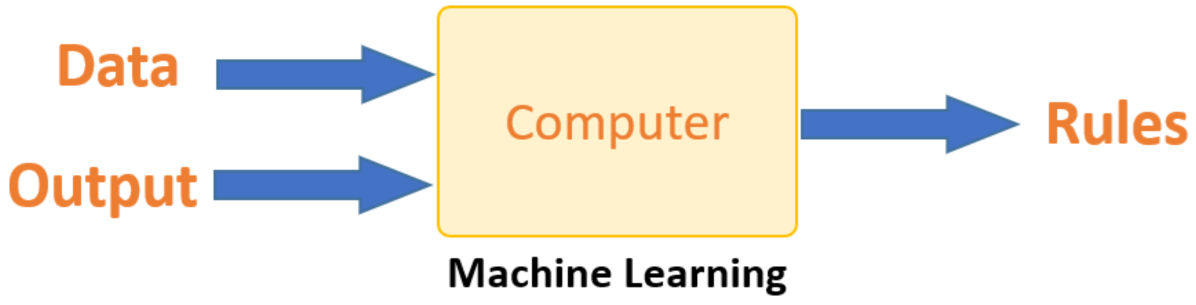
Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.



Traditional Programming

Machine learning is supposed to overcome this issue. The machine learns how the input and output data are correlated and it writes a rule. The programmers do not need to write new rules each time there is new data. The algorithms adapt in response to new data and experiences to improve efficacy over time.

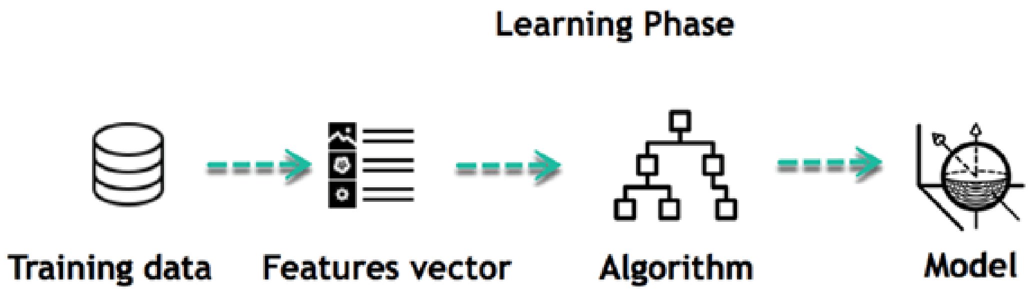


How does Machine Learning Work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector.** You can think of a feature vector as a subset of data that is used to tackle a problem.

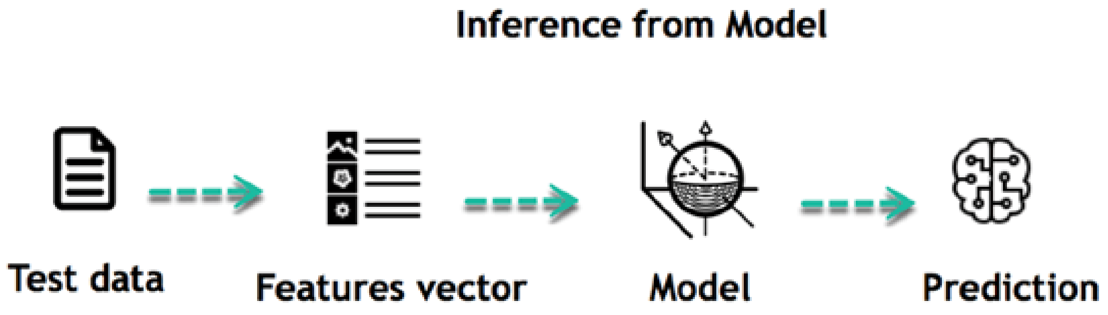
The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.



For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.



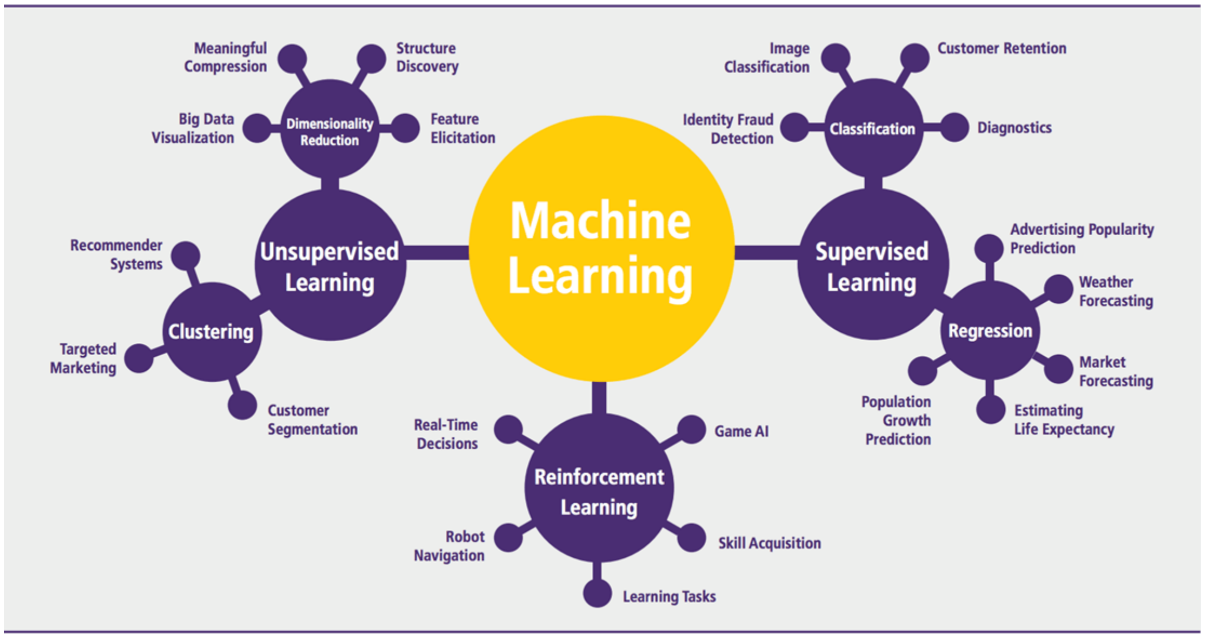
The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying

9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

Machine Learning Algorithms and Where they are Used?



Machine learning Algorithms

Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

- Classification task
- Regression task

Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer , it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above Machine learning example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

Algorithm Name	**Description**	**Type**
Linear regression	Finds a way to correlate each feature to the output to help predict future values.	Regression
Logistic regression	Extension of linear regression that's used for classification task The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colours)	Classification
Decision tree	Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a colour, each possible colour becomes a new branch) until a final decision output is made	Regression Classification

Algorithm Name	Description	Type
Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyper lane that optimally divided the classes. It is best used with a non-linear solver.	Regression(not very common) Classification
Random forest	The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction.	Regression Classification
Ada Boost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
Gradient-boosting trees	Gradient-boosting trees is a state-of-the-art classification/ regression technique. It is focusing on the error committed by the previous trees and tries to correct it.	Regression Classification

Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

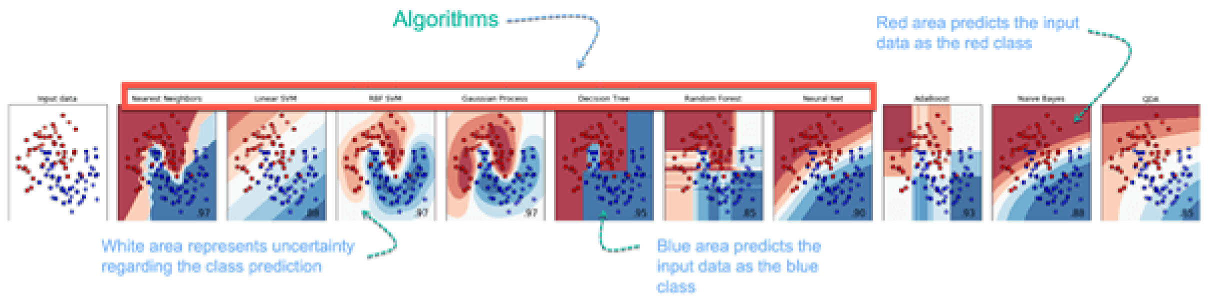
Algorithm	Description	Type
K-means clustering	Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans)	Clustering
Gaussian mixture model	A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters)	Clustering
Hierarchical clustering	Splits clusters along a hierarchical tree to form a classification system. Can be used for Cluster loyalty-card customer	Clustering
Recommender system	Help to define the relevant data for making a recommendation.	Clustering
PCA/T-SNE	Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances.	Dimension Reduction

How to Choose Machine Learning Algorithm

Machine Learning (ML) algorithm:

There are plenty of machine learning algorithms. The choice of the algorithm is based on the objective.

In the Machine learning example below, the task is to predict the type of flower among the three varieties. The predictions are based on the length and the width of the petal. The picture depicts the results of ten different algorithms. The picture on the top left is the dataset. The data is classified into three categories: red, light blue and dark blue. There are some groupings. For instance, from the second image, everything in the upper left belongs to the red category, in the middle part, there is a mixture of uncertainty and light blue while the bottom corresponds to the dark category. The other images show different algorithms and how they try to classified the data.



Challenges and Limitations of Machine Learning

The primary challenge of machine learning is the lack of data or the diversity in the dataset. A machine cannot learn if there is no data available. Besides, a dataset with a lack of diversity gives the machine a hard time. A machine needs to have heterogeneity to learn meaningful insight. It is rare that an algorithm can extract information when there are no or few variations. It is recommended to have at least 20 observations per group to help the machine learn. This constraint leads to poor evaluation and prediction.

Application of Machine Learning

Augmentation:

- Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

Automation:

- Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

Finance Industry

- Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

Government organization

- The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

Healthcare industry

- Healthcare was one of the first industry to use machine learning with image detection.

Marketing

- Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

Example of application of Machine Learning in Supply Chain

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track , report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

Example of Machine Learning Google Car

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

Why is Machine Learning Important?

Machine learning is the best tool so far to analyse, understand and identify a pattern in the data. One of the main ideas behind machine learning is that the computer can be trained to automate tasks that would be exhaustive or impossible for a human being. The clear breach from the traditional analysis is that machine learning can take decisions with minimal human intervention.

Take the following example for this ML tutorial; a retail agent can estimate the price of a house based on his own experience and his knowledge of the market.

A machine can be trained to translate the knowledge of an expert into features. The features are all the characteristics of a house, neighbourhood, economic environment, etc. that make the price difference. For the expert, it took him probably some years to master the art of estimate the price of a house. His expertise is getting better and better after each sale.

For the machine, it takes millions of data, (i.e., example) to master this art. At the very beginning of its learning, the machine makes a mistake, somehow like the junior salesman.

Once the machine sees all the example, it got enough knowledge to make its estimation. At the same time, with incredible accuracy. The machine is also able to adjust its mistake accordingly.

Most of the big company have understood the value of machine learning and holding data.

McKinsey have estimated that the value of analytics ranges from $9.5 trillion to $15.4 trillion while $5 to 7 trillion can be attributed to the most advanced AI techniques.

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Overview

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset of handwritten digits has often been used.

Machine learning approaches

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

Other approaches have been developed which don't fit neatly into this three-fold categorisation, and sometimes more than one is used by the same machine learning system. For example topic modelling, dimensionality reduction or meta learning.

As of 2020, deep learning has become the dominant approach for much on going work in the field of machine learning.

History and relationships to other fields

The term machine learning was coined in 1959 by Arthur Samuel, an American IBMer and pioneer in the field of computer gaming and artificial intelligence. A representative book of the machine learning research during the 1960s was the Nilsson's book on Learning Machines, dealing mostly with machine learning for pattern classification. Interest related to pattern recognition continued into the 1970s, as described by Duda and Hart in 1973. In 1981 a report was given on using teaching strategies so that a neural network learns to recognize 40 characters (26 letters, 10 digits, and 4 special symbols) from a computer terminal.

Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E. "This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?"

Modern day machine learning has two objectives, one is to classify data based on models which have been developed, The other purpose is to make predictions for future outcomes based on these models. A hypothetical algorithm specific to classifying data may use computer vision of moles coupled with supervised learning in order to train it to classify the cancerous moles. Where as, a machine learning algorithm for stock trading may inform the trader of future potential predictions.

Artificial intelligence

Machine Learning as subfield of AI

Part of Machine Learning as subfield of AI or part of AI as subfield of Machine Learning
As a scientific endeavour, machine learning grew out of the quest for artificial intelligence. In the early days of AI as an academic discipline, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what was then termed "neural networks"; these were mostly perceptron's and other models that were later found to be reinventions of the generalized linear models of statistics. Probabilistic reasoning was also employed, especially in automated medical diagnosis. However, an increasing emphasis on the logical, knowledge-based approach caused a rift between AI and machine learning. Probabilistic systems were plagued by theoretical and practical problems of data acquisition and representation. By 1980, expert systems had come to dominate AI, and statistics was out of favour. Work on symbolic/knowledge-based learning did continue within AI, leading to inductive logic programming, but the more statistical line of research was now outside the field of AI proper, in pattern recognition and information retrieval. Neural networks research had been abandoned by AI and computer science around the same time. This line, too, was continued outside the AI/CS field, as "connectionism", by researchers from other disciplines including Hopfield, Rumelhart and Hinton. Their main success came in the mid-1980s with the reinvention of back propagation.

Machine learning (ML), reorganized as a separate field, started to flourish in the 1990s. The field changed its goal from achieving artificial intelligence to tackling solvable problems of a practical nature. It shifted focus away from the symbolic approaches it had inherited from AI, and toward methods and models borrowed from statistics and probability theory.

As of 2020, many sources continue to assert that machine learning remains a subfield of AI. The main disagreement is whether all of ML is part of AI, as this would mean that anyone using ML could claim they are using AI. Others have the view that not all of ML is part of AI where only an 'intelligent' subset of ML is part of AI.

The question to what is the difference between ML and AI is answered by Judea Pearl in The Book of Why. Accordingly ML learns and predicts based on passive observations, whereas AI implies an agent interacting with the environment to learn and take actions that maximize its chance of successfully achieving its goals.

Data mining

Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of (previously) unknown properties in the data (this is the analysis step of knowledge discovery in databases). Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as "unsupervised learning" or as a pre-processing step to improve learner accuracy. Much of the confusion between these two research communities (which do often have separate conferences and separate journals, ECML PKDD being a major exception) comes from the basic assumptions they work with: in machine learning, performance is usually evaluated with respect to the ability to reproduce known knowledge, while in knowledge discovery and data mining (KDD) the key task is the discovery of previously unknown knowledge. Evaluated with respect to known knowledge, an uninformed (unsupervised) method will easily be outperformed by other supervised methods, while in a typical KDD task, supervised methods cannot be used due to the unavailability of training data.

Optimization

Machine learning also has intimate ties to optimization: many learning problems are formulated as minimization of some loss function on a training set of examples. Loss functions express the discrepancy between the predictions of the model being trained and the actual problem instances (for example, in classification, one wants to assign a label to instances, and models are trained to correctly predict the pre-assigned labels of a set of examples).

Generalization

The difference between optimization and machine learning arises from the goal of generalization: while optimization algorithms can minimize the loss on a training set, machine learning is concerned with minimizing the loss on unseen samples. Characterizing the generalization of various learning algorithms is an active topic of current research, especially for deep learning algorithms.

Statistics

Machine learning and statistics are closely related fields in terms of methods, but distinct in their principal goal: statistics draws population inferences from a sample, while machine learning finds generalizable predictive patterns. According to Michael I. Jordan, the ideas of machine learning, from methodological principles to theoretical tools, have had a long pre-history in statistics. He also suggested the term data science as a placeholder to call the overall field.

Leo Breiman distinguished two statistical modelling paradigms: data model and algorithmic model, wherein "algorithmic model" means more or less the machine learning algorithms like Random forest.

Some statisticians have adopted methods from machine learning, leading to a combined field that they call statistical learning.

Theory

A core objective of a learner is to generalize from its experience. Generalization in this context is the ability of a learning machine to perform accurately on new, unseen examples/tasks after having experienced a learning data set. The training examples come from some generally unknown probability distribution (considered representative of the space of occurrences) and the learner has to build a general model about this space that enables it to produce sufficiently accurate predictions in new cases.

The computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science known as computational learning theory. Because training sets are finite and the future is uncertain, learning theory usually does not yield guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common. The bias–variance decomposition is one way to quantify generalization error.

For the best performance in the context of generalization, the complexity of the hypothesis should match the complexity of the function underlying the data. If the hypothesis is less complex than the function, then the model has under fitted the data. If the complexity of the model is increased in response, then the training error decreases. But if the hypothesis is too complex, then the model is subject to over fitting and generalization will be poorer.

In addition to performance bounds, learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results. Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time.

Approaches

Types of learning algorithms

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

Supervised learning

A support vector machine is a supervised learning model that divides the data into regions separated by a linear boundary. Here, the linear boundary divides the black circles from the white.

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

Types of supervised learning algorithms include active learning, classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email.

Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labelled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more pre designated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

Semi-supervised learning

Semi-supervised learning falls between unsupervised learning (without any labelled training data) and supervised learning (with completely labelled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that unlabelled data, when used in conjunction with a small amount of labelled data, can produce a considerable improvement in learning accuracy.

In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.

Reinforcement learning

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov decision process (MDP). Many reinforcement learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

Self-learning

Self-learning as a machine learning paradigm was introduced in 1982 along with a neural network capable of self-learning named crossbar adaptive array (CAA). It is a learning with no external rewards and no external teacher advice. The CAA self-learning algorithm computes, in a crossbar fashion, both decisions about actions and emotions (feelings) about consequence situations. The system is driven by the interaction between cognition and emotion. The self-learning algorithm updates a memory matrix W =||w (a,s)|| such that in each iteration executes the following machine learning routine:

```
In situation s perform an action a;
Receive consequence situation s';
Compute emotion of being in consequence situation v(s');
Update crossbar memory w'(a,s) = w(a,s) + v(s').
```

It is a system with only one input, situation s, and only one output, action (or behaviour) a. There is neither a separate reinforcement input nor an advice input from the environment. The back propagated value (secondary reinforcement) is the emotion toward the consequence situation. The CAA exists in two environments, one is the behavioural environment where it behaves, and the other is the genetic environment, wherefrom it initially and only once receives initial emotions about situations to be encountered in the behavioural environment. After receiving the genome (species) vector from the genetic environment, the CAA learns a goal-seeking behaviour, in an environment that contains both desirable and undesirable situations.

Feature learning

Several learning algorithms aim at discovering better representations of the inputs provided during training. Classic examples include principal components analysis and cluster analysis. Feature learning algorithms, also called representation learning algorithms, often attempt to preserve the information in their input but also transform it in a way that makes it useful, often as a pre-processing step before performing classification or predictions. This technique allows reconstruction of the inputs coming from the unknown data-generating distribution, while not being necessarily faithful to configurations that are implausible under that distribution. This replaces manual feature engineering, and allows a machine to both learn the features and use them to perform a specific task.

Feature learning can be either supervised or unsupervised. In supervised feature learning, features are learned using labelled input data. Examples include artificial neural networks, multilayer perceptron’s, and supervised dictionary learning. In unsupervised feature learning, features are learned with unlabelled input data. Examples include dictionary learning, independent component analysis, auto encoders, matrix factorization and various forms of clustering.

Manifold learning algorithms attempt to do so under the constraint that the learned representation is low-dimensional. Sparse coding algorithms attempt to do so under the constraint that the learned representation is sparse, meaning that the mathematical model has many zeros. Multi linear subspace learning algorithms aim to learn low-dimensional representations directly from tensor representations for multidimensional data, without reshaping them into higher-dimensional vectors. Deep learning algorithms discover multiple levels of representation, or a hierarchy of features, with higher-level, more abstract features defined in terms of (or generating) lower-level features. It has been argued that an intelligent machine is one that learns a representation that disentangles the underlying factors of variation that explain the observed data.

Feature learning is motivated by the fact that machine learning tasks such as classification often require input that is mathematically and computationally convenient to process. However, real-world data such as images, video, and sensory data has not yielded to attempts to algorithmically define specific features. An alternative is to discover such features or representations thorough examination, without relying on explicit algorithms.

Sparse dictionary learning

Sparse dictionary learning is a feature learning method where a training example is represented as a linear combination of basic functions, and is assumed to be a sparse matrix. The method is strongly NP-hard and difficult to solve approximately. A popular heuristic method for sparse dictionary learning is the K-SVD algorithm. Sparse dictionary learning has been applied in several contexts. In classification, the problem is to determine the class to which a previously unseen training example belongs. For a dictionary where each class has already been built, a new training example is associated with the class that is best sparsely represented by the corresponding dictionary. Sparse dictionary learning has also been applied in image de-noising. The key idea is that a clean image patch can be sparsely represented by an image dictionary, but the noise cannot.

Anomaly detection

In data mining, anomaly detection, also known as outlier detection, is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data. Typically, the anomalous items represent an issue such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are referred to as outliers, novelties, noise, deviations and exceptions.

In particular, in the context of abuse and network intrusion detection, the interesting objects are often not rare objects, but unexpected bursts of inactivity. This pattern does not adhere to the common statistical definition of an outlier as a rare object, and many outlier detection methods (in particular, unsupervised algorithms) will fail on such data unless it has been aggregated appropriately. Instead, a cluster analysis algorithm may be able to detect the micro-clusters formed by these patterns.

Three broad categories of anomaly detection techniques exist. Unsupervised anomaly detection techniques detect anomalies in an unlabelled test data set under the assumption that the majority of the instances in the data set are normal, by looking for instances that seem to fit least to the remainder of the data set. Supervised anomaly detection techniques require a data set that has been labelled as "normal" and "abnormal" and involves training a classifier (the key difference to many other statistical classification problems is the inherently unbalanced nature of outlier detection). Semi-supervised anomaly detection techniques construct a model representing normal behaviour from a given normal training data set and then test the likelihood of a test instance to be generated by the model.

Robot learning

In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies and imitation.

Association rules

Association rule learning is a rule-based machine learning method for discovering relationships between variables in large databases. It is intended to identify strong rules discovered in databases using some measure of "interestingness".

Rule-based machine learning is a general term for any machine learning method that identifies, learns, or evolves "rules" to store, manipulate or apply knowledge. The defining characteristic of a rule-based machine learning algorithm is the identification and utilization of a set of relational rules that collectively represent the knowledge captured by the system. This is in contrast to other machine learning algorithms that commonly identify a singular model that can be universally applied to any instance in order to make a prediction. Rule-based machine learning approaches include learning classifier systems, association rule learning, and artificial immune systems.

Based on the concept of strong rules, Rakesh Agrawal, Tomasz Imieliński and Arun Swami introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule {\displaystyle \{\mathrm {onions,potatoes} \}\Rightarrow \{\mathrm {burger} \}}\{{\mathrm {onions,potatoes}}\}\Rightarrow \{{\mathrm {burger}}\} found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, they are likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as promotional pricing or product placements. In addition to market basket analysis, association rules are employed today in application areas including Web usage mining, intrusion detection, continuous production, and bioinformatics. In contrast with sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

Learning classifier systems (LCS) are a family of rule-based machine learning algorithms that combine a discovery component, typically a genetic algorithm, with a learning component, performing either supervised learning, reinforcement learning, or unsupervised learning. They seek to identify a set of context-dependent rules that collectively store and apply knowledge in a piecewise manner in order to make predictions.

Inductive logic programming (ILP) is an approach to rule-learning using logic programming as a uniform representation for input examples, background knowledge, and hypotheses. Given an encoding of the known background knowledge and a set of examples represented as a logical database of facts, an ILP system will derive a hypothesized logic program that entails all positive and no negative examples. Inductive programming is a related field that considers any kind of programming language for representing hypotheses (and not only logic programming), such as functional programs.

Inductive logic programming is particularly useful in bioinformatics and natural language processing. Gordon Plotkin and Ehud Shapiro laid the initial theoretical foundation for inductive machine learning in a logical setting. Shapiro built their first implementation (Model Inference System) in 1981: a Prolog program that inductively inferred logic programs from positive and negative examples. The term inductive here refers to philosophical induction, suggesting a theory to explain observed facts, rather than mathematical induction, proving a property for all members of a well-ordered set.

Models

Performing machine learning involves creating a model, which is trained on some training data and then can process additional data to make predictions. Various types of models have been used and researched for machine learning systems.

Artificial neural networks

An artificial neural network is an interconnected group of nodes, akin to the vast network of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

Artificial neural networks (ANNs), or connectionist systems, are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules.

An ANN is a model based on a collection of connected units or nodes called "artificial neurons", which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit information, a "signal", from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called "edges". Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

Deep learning consists of multiple hidden layers in an artificial neural network. This approach tries to model the way the human brain processes light and sound into vision and hearing. Some successful applications of deep learning are computer vision and speech recognition.

Decision trees

Decision tree learning uses a decision tree as a predictive model to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data mining, and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data, but the resulting classification tree can be an input for decision making.

Support vector machines

Support vector machines (SVMs), also known as support vector networks, are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. An SVM training algorithm is a non-probabilistic, binary, linear classifier, although methods such as Platt scaling exist to use SVM in a probabilistic classification setting. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Regression analysis

Regression analysis encompasses a large variety of statistical methods to estimate the relationship between input variables and their associated features. Its most common form is linear regression, where a single line is drawn to best fit the given data according to a mathematical criterion such as ordinary least squares. The latter is often extended by regularization (mathematics) methods to mitigate over-fitting and bias, as in ridge regression. When dealing with non-linear problems, go-to models include polynomial regression (for example, used for trend line fitting in Microsoft Excel, logistic regression (often used in statistical classification) or even kernel regression, which introduces non-linearity by taking advantage of the kernel trick to implicitly map input variables to higher-dimensional space.

Bayesian networks

A simple Bayesian network. Rain influences whether the sprinkler is activated, and both rain and the sprinkler influence whether the grass is wet.

A Bayesian network, belief network, or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independence with a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Efficient algorithms exist that perform inference and learning. Bayesian networks that model sequences of variables, like speech signals or protein sequences are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.

Genetic algorithms

A genetic algorithm (GA) is a search algorithm and heuristic technique that mimics the process of natural selection, using methods such as mutation and crossover to generate new genotypes in the hope of finding good solutions to a given problem. In machine learning, genetic algorithms were used in the 1980s and 1990s. Conversely, machine learning techniques have been used to improve the performance of genetic and evolutionary algorithms.

Training models

Usually, machine learning models require a lot of data in order for them to perform well. Usually, when training a machine learning model, one needs to collect a large, representative sample of data from a training set. Data from the training set can be as varied as a corpus of text, a collection of images, and data collected from individual users of a service. Over fitting is something to watch out for when training a machine learning model. Trained models derived from biased data can result in skewed or undesired predictions. Algorithmic bias is a potential result from data not fully prepared for training.

Federated learning

Federated learning is an adapted form of distributed artificial intelligence to training machine learning models that decentralizes the training process, allowing for users' privacy to be maintained by not needing to send their data to a centralized server. This also increases efficiency by decentralizing the training process to many devices. For example, Gboard uses federated machine learning to train search query prediction models on users' mobile phones without having to send individual searches back to Google

1.2 SCOPE OF THE PROJECT

The proposed system considers the stress detection among the tech people. The dataset considered is a survey among the working people, which considered all possible question for stress detection.

1.3 OBJECTIVE

These abnormalities can be termed as the Bipolar disorder which can be treated by undergoing some treatment suggested by specialists. For this research, data has been collected from working people which comprises of all kinds of questions for despondent detection and the dataset has been run through some machine learning algorithms. Random Forest algorithm gives the highest accuracy as 87.02% compared to the other algorithms.

1.4 PROBLEM STATEMENT

After population outburst in India, the proportion of specialists to patients is 1: 1810 and time spent by a specialist for the patient is under two minutes. Despondency is the main source of disability worldwide. For all intents and purposes 80% of people with cerebral disorders remain misdiagnosed in making countries with practically 1 million people taking their lives each year. Similarly, as demonstrated by the WHO, 1 out of 14 comprehensive encounters disquiet. The WHO says that strain disorders are the most generally perceived mental disorders worldwide with express dread, huge troublesome issue and social dread being the most notable anxiety disorders.

1.5 EXISTING SYSTEM

- Sridharan et al. presented the detection diagnostics on online social media with the assistance of Convolution Neural Networks (CNN) where accentuation was to get information posted by different clients while also ensuring algorithm protects the security with the assistance of separating agents which deal with information.
- M N Stollar, M Lechh, S J Stollar, N B Allen approach utilizes an upgraded spectral move off parameters for detection of the depression side effects from discourse signals on the clinical dataset obtained. The classification of these highlights is done with the assistance of basic SVM classifier. In past investigation, gender dependence has improved depression classification either best for females, males and fluctuated amongst highlights. In this examination depression detection was more viable in males than females.

1.5.1 EXISTING SYSTEM DISADVANTAGES

- The current works contemplated the downturn among the substance from social media, whereas the working individuals stress just not considered.
- Less accuracy.
- Prone to misleading results

1.5.2 LITERATURE SURVEY

TITLE **:** Mental Disorder Detection : Bipolar Disorder Scrutinization Using Machine Learning

AUTHOR **:** Ranjana Jadhav; Vinay Chellwani; Sharyu Deshmukh; Hitesh Sachdev

YEAR 2019

DESCRIPTION

In today's modernized lifestyle there is an increase in stress and pressure, consequently people are facing mental issues and abnormalities. However, the utility of these abnormalities in distinguishing individual Bipolar disorder patients from Mood Disorder or healthy controls and stratify patients based on overall illness, burden has not been investigated in a large cohort. This research uses Machine Learning approach to screen Bipolar Disorder using the Mood Disorder Questionnaire (MDQ). The data set was fed to Decision Tree Classifier which determines the most significant feature in the dataset and makes it the decision factor at that level of the decision tree. The testing samples are compared with the decision factor at each level and this decision takes each test case to a defined class (Screened Positive or Screened Negative). The Mood Disorder Questionnaire is a feasible method for detection of Bipolar Disorder.

TITLE **:** Intelligent data mining and machine learning for mental health diagnosis using genetic algorithm

AUTHOR **:** Azar, Ghassan & Gloster, Clay & El-Bathy, Naser & Yu, Su&Neela, Rajasree & Alothman, Israa

YEAR 2015

DESCRIPTION

Inappropriate diagnosis of mental health illnesses leads to wrong treatment and causes irreversible deterioration in the client's mental health status including hospitalization and/or premature death. About 12 million patients are misdiagnosed annually in US. In this paper, a novel study introduces a semi-automated system that aids in preliminary diagnosis of the psychological disorder patient. This is accomplished based on matching description of a patient's mental health status with the mental illnesses illustrated in DSM-IV-TR, Fourth Edition Text Revision. The study constructs the semi-automated system based on an integration of the technology of genetic algorithm, classification data mining and machine learning. The goal is not to fully automate the classification process of mentally ill individuals, but to ensure that a classifier is aware of all possible mental health illnesses could match patient's symptoms. The classifier/psychological analyst will be able to make an informed, intelligent and appropriate assessment that will lead to an accurate prognosis. The analyst will be the ultimate selector of the diagnosis and treatment plan.

TITLE **:** A Framework for Classifying Online Mental Health-Related Communities With an Interest in Depression

AUTHOR **:** B. Saha, T. Nguyen, D. Phung and S. Venkatesh

YEAR 2016

DESCRIPTION

Mental illness has a deep impact on individuals, families, and by extension, society as a whole. Social networks allow individuals with mental disorders to communicate with others sufferers via online communities, providing an invaluable resource for studies on textual signs of psychological health problems. Mental disorders often occur in combinations, e.g., a patient with an anxiety disorder may also develop depression. This co-occurring mental health condition provides the focus for our work on classifying online communities with an interest in depression. For this, we have crawled a large body of 620 000 posts made by 80 000 users in 247 online communities. We have extracted the topics and psycholinguistic features expressed in the posts, using these as inputs to our model. Following a machine learning technique, we have formulated a joint modelling framework in order to classify mental health-related co-occurring online communities from these features. Finally, we performed empirical validation of the model on the crawled dataset where our model outperforms recent state-of-the-art baselines.

TITLE **:** Detecting Cognitive Distortions Through Machine Learning Text Analytics

AUTHOR **:** T. Simms, C. Ramstedt, M. Rich, M. Richards, T. Martinez and C. Giraud-Carrier

YEAR 2017

DESCRIPTION

Machine learning and text analytics have proven increasingly useful in a number of health-related applications, particularly in the context of analysing online data for disease epidemics and warning signs of a variety of mental health issues. We follow in this tradition here, but focus our attention on cognitive distortion, a precursor and symptom of disruptive psychological disorders such as anxiety, anorexia and depression. We collected a number of personal blogs from the Tumblr API, and labelled them based on whether they exhibited distorted thought patterns. We then used LIWC to extract textual features and applied machine learning to the resulting vectors. Our findings show that it is possible to detect cognitive distortions automatically from personal blogs with relatively good accuracy (73.0%) and false negative rate (30.4%).

TITLE **:** Machine Learning Framework for the Detection of Mental Stress at Multiple Levels

AUTHOR **:** Subhani, Ahmad & Mumtaz, Wajid & MOHAMAD SAAD, MOHAMAD NAUFAL & Kamel, Nidal& Malik, Aamir.

YEAR 2017

DESCRIPTION

Mental stress has become a social issue and could become a cause of functional disability during routine work. In addition, chronic stress could implicate several psychophysiological disorders. For example, stress increases the likelihood of depression, stroke, heart attack, and cardiac arrest. The latest neuroscience reveals that the human brain is the primary target of mental stress, because the perception of the human brain determines a situation that is threatening and stressful. In this context, an objective measure for identifying the levels of stress while considering the human brain could considerably improve the associated harmful effects. Therefore, in this paper, a machine learning (ML) framework involving electroencephalogram (EEG) signal analysis of stressed participants is proposed. In the experimental setting, stress was induced by adopting a well-known experimental paradigm based on the montreal imaging stress task. The induction of stress was validated by the task performance and subjective feedback. The proposed ML framework involved EEG feature extraction, feature selection (receiver operating characteristic curve, t-test and the Bhattacharya distance), classification (logistic regression, support vector machine and naïve Bayes classifiers) and tenfold cross validation. The results showed that the proposed framework produced 94.6% accuracy for two-level identification of stress and 83.4% accuracy for multiple level identification. In conclusion, the proposed EEG-based ML framework has the potential to quantify stress objectively into multiple levels. The proposed method could help in developing a computer-aided diagnostic tool for stress detection.

1.6 PROPOSED SYSTEM

- The proposed system considers the stress detection among the tech people. The dataset considered is a survey among the working people, which considered all possible question for stress detection.
- The designed approach utilizes the ML algorithm for stress identification; Random forest is used on the dataset for learning and detection. The proposed approach of Random Forest is the suitable algorithm for mental disorder prediction.
- The proposed system was developed in Python and libraries which are necessary are used. The dataset is downloaded from kaggle. The data is then divided into training dataset and testing dataset. ML algorithms which are apt for this problem are used.

1.6.1 PROPOSED SYSTEM ADVANTAGES

- The proposed system applied on the dataset and they have given a good accuracy.
- Good efficiency in detecting the psychological instability.

CHAPTER 2

PROJECT DESCRIPTION

2.1 GENERAL

The code was developed in Python and libraries which are necessary are used. The dataset is downloaded from kaggle. The data is then divided into training dataset and testing dataset. ML algorithms which are apt for this problem are used.

2.2 METHODOLOGIES

2.2.1 MODULES NAME:

This project having the following five modules:

- Data Collection
- Dataset
- Data Preparation
- Model Selection
- Analyse and Prediction
- Accuracy on test set
- Saving the Trained Model

2.2.2 MODULES EXPLANATION AND DIAGRAM

- **Data Collection:**

This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform.

There are several techniques to collect the data, like web scraping, manual interventions and etc.

The dataset used in this Finding Psychological Instability Using Machine taken from kaggle and some other source

Link: https://www.kaggle.com/osmi/mental-health-in-tech-survey/data

- **Dataset:**

The dataset consists of 1000 individual data. There are 27 columns in the dataset, which are described below.

Timestamp

Age

Gender

Country

State: If you live in the United States, which state or territory do you live in?

Self- employed: Are you self-employed?

Family-history: Do you have a family history of mental illness?

Treatment: Have you sought treatment for a mental health condition?

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

No-employees: How many employees does your company or organization have?

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

Tech-company: Is your employer primarily a tech company/organization?

Benefits: Does your employer provide mental health benefits?

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

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

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

Anonymity: Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources?

Leave: How easy is it for you to take medical leave for a mental health condition?

Mental health consequence: Do you think that discussing a mental health issue with your employer would have negative consequences?

Physical health consequence: Do you think that discussing a physical health issue with your employer would have negative consequences?

Co-workers: Would you be willing to discuss a mental health issue with your co-workers?

Supervisor: Would you be willing to discuss a mental health issue with your direct supervisor(s)?

Mental health interview: Would you bring up a mental health issue with a potential employer in an interview?

Physical health interview: Would you bring up a physical health issue with a potential employer in an interview?

Mental vs. physical: Do you feel that your employer takes mental health as seriously as physical health?

Obs-consequence: Have you heard of or observed negative consequences for co-workers with mental health conditions in your workplace?

Comments: Any additional notes or comments

➢ Data Preparation:

We will transform the data. By getting rid of missing data and removing some columns. First we will create a list of column names that we want to keep or retain.

Next we drop or remove all columns except for the columns that we want to retain.

Finally we drop or remove the rows that have missing values from the data set.

➢ Model Selection:

While creating a machine learning model, we need two dataset, one for training and other for testing. But now we have only one. So let's split this in two with a ratio of 80:20. We will also divide the data frame into feature column and label column.

Here we imported train_test_split function of sklearn. Then use it to split the dataset. Also, test_size = *0.2*, it makes the split with 80% as train dataset and 20% as test dataset.

The random_state parameter seeds random number generator that helps to split the dataset.

The function returns four datasets. Labelled them as train_x, train_y, test_x, test_y. If we see shape of this datasets we can see the split of dataset.

We will use Random Forest Classifier, which fits multiple decision tree to the data. Finally I train the model by passing train_x, train_y to the *fit* method.

Once the model is trained, we need to Test the model. For that we will pass test_x to the predict method.

Random Forest is one of the most powerful methods that is used in machine learning for regression problems. The random forest comes in the category of the supervised regression algorithm. This algorithm is carried out in two different stages the first one deals with the creation of the forest of the given dataset, and the other one deals with the prediction from the regression.

➢ Analyse and Prediction:

In the actual dataset, we chose only 10 features :

Age

Gender

Family-history: Do you have a family history of mental illness?

No-employees: How many employees does your company or organization have?

Self-employed: Are you self-employed?

Benefits: Does your employer provide mental health benefits?

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

Anonymity: Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources?

Leave: How easy is it for you to take medical leave for a mental health condition?

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

➢ Accuracy on test set:

We got a accuracy of 82.1% on test set.

➢ Saving the Trained Model:

Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or . pkl file using a library like pickle .
Make sure you have pickle installed in your environment.
Next, let's import the module and dump the model into . pkl file

2.2.3 GIVEN INPUT EXPECTED OUTPUT:

- **INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

- **OBJECTIVES**

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

- **OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

CHAPTER 3

REQUIREMENTS ENGINEERING

3.1 GENERAL

The program scripts will be implemented further with the optimization techniques and compared with the base results i.e. with default parameters. The spam detection engine should be able to take email datasets as input and with the help of text mining and optimized supervised algorithms, it should be able to classify the mail as ham or spam. Figure-1 represents the process that is followed to implement the model.

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

HARDWARE

• Processor	:	Pentium Iv 2.6 Ghz, Intel Core 2 Duo.
• Ram	:	512 Mb Dd Ram
• Monitor	:	15” Color
• Hard Disk	:	40 GB

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

SOFTWARE

• Front End	:	J2EE (JSP, Servlet)
• Back End	:	MY SQL 5.5
• Operating System	:	Windows 7
• IDE	:	Eclipse

3.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behaviour, and outputs. The proposed system considers the stress detection among the tech people. The dataset considered is a survey among the working people, which considered all possible question for stress detection. The designed approach utilizes the ML algorithm for stress identification; SVM, DT and Random forest are used on the dataset for learning and detection. The proposed approach finds the suitable algorithm for mental disorder prediction.

3.5 NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

- **Usability**

The system is designed with completely automated process hence there is no or less user intervention.

- **Reliability**

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

- **Performance**

This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

- **Supportability**

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having JVM, built into the system.

- **Implementation**

The system is implemented in web environment using struts framework. The apache tomcat is used as the web server and windows xp professional is used as the platform. Interface the user interface is based on Struts provides HTML Tag

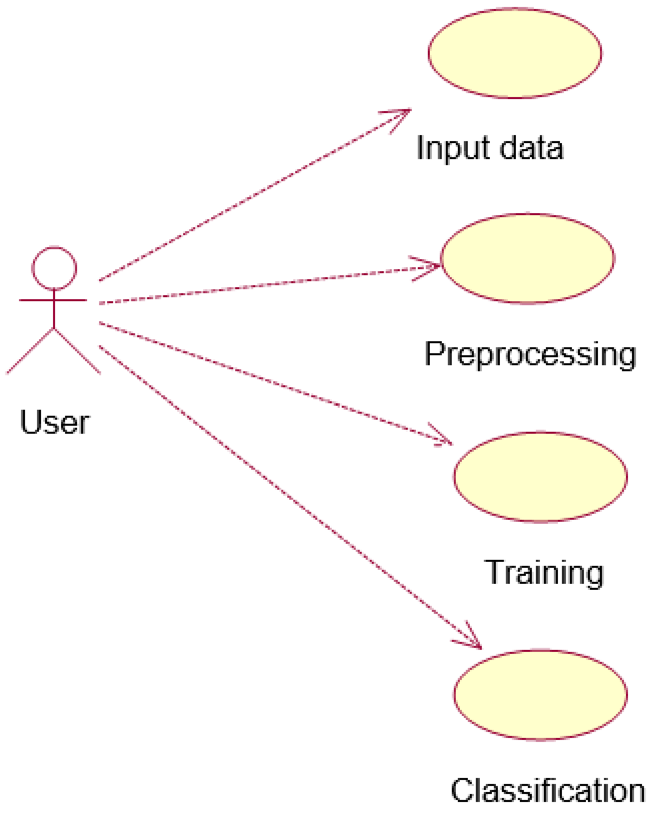
CHAPTER 4

DESIGN ENGINEERING

4.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

4.1.1 Use Case Diagram

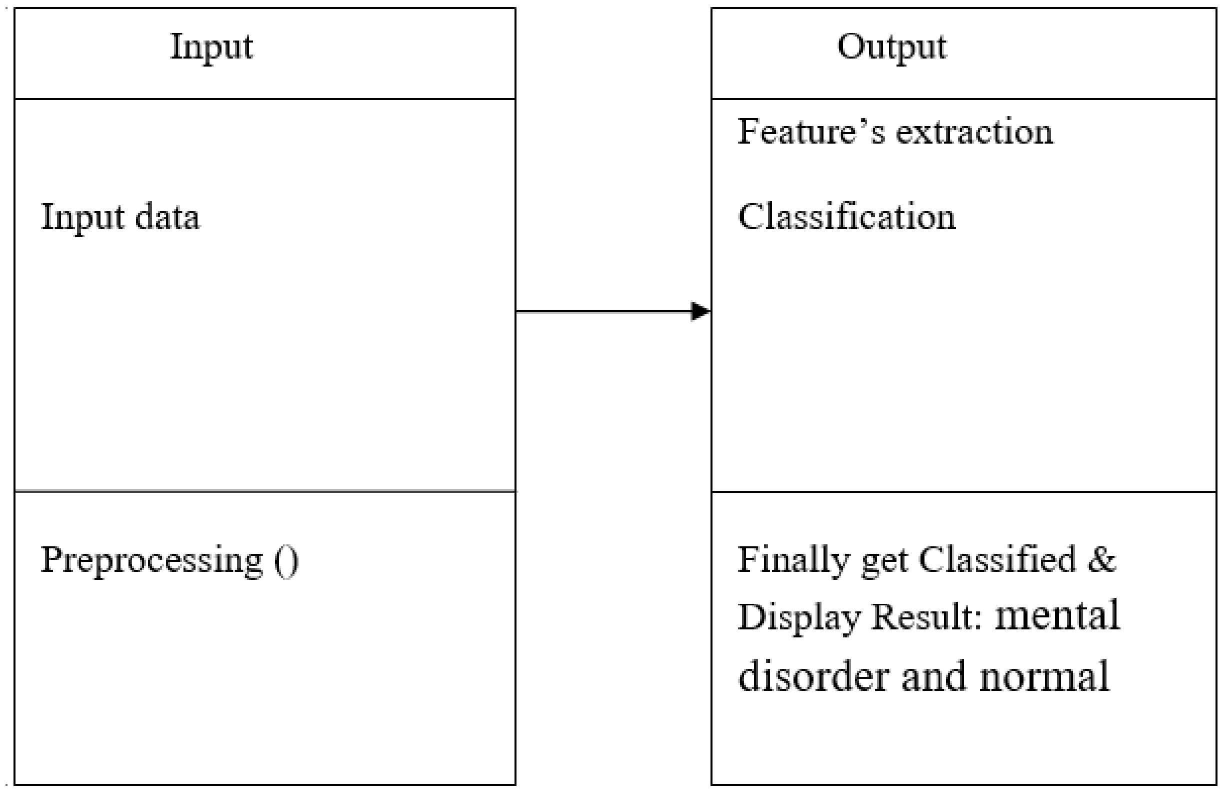


EXPLANATION:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

.

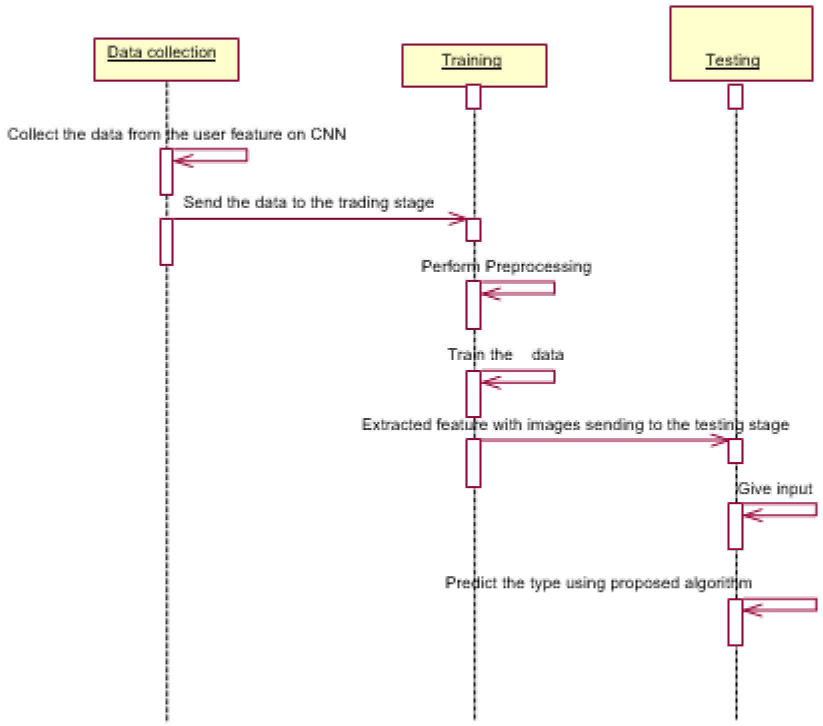
4.1.2 Class Diagram



EXPLANATION

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

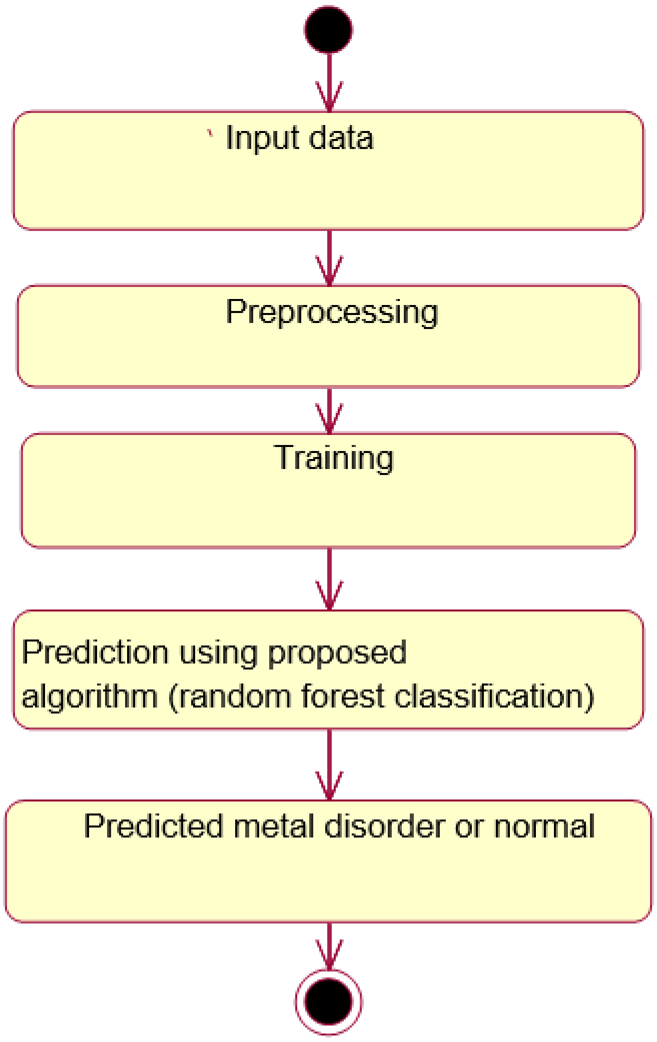
4.1.3 Sequence Diagram


Data collection
Training
Testing
Collect the data from the user feature on CNN
Send the data to the trading stage
Perform Preprocessing
Train the data
Extracted feature with images sending to the testing stage
Give input
Predict the type using proposed algorithm

EXPLANATION:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

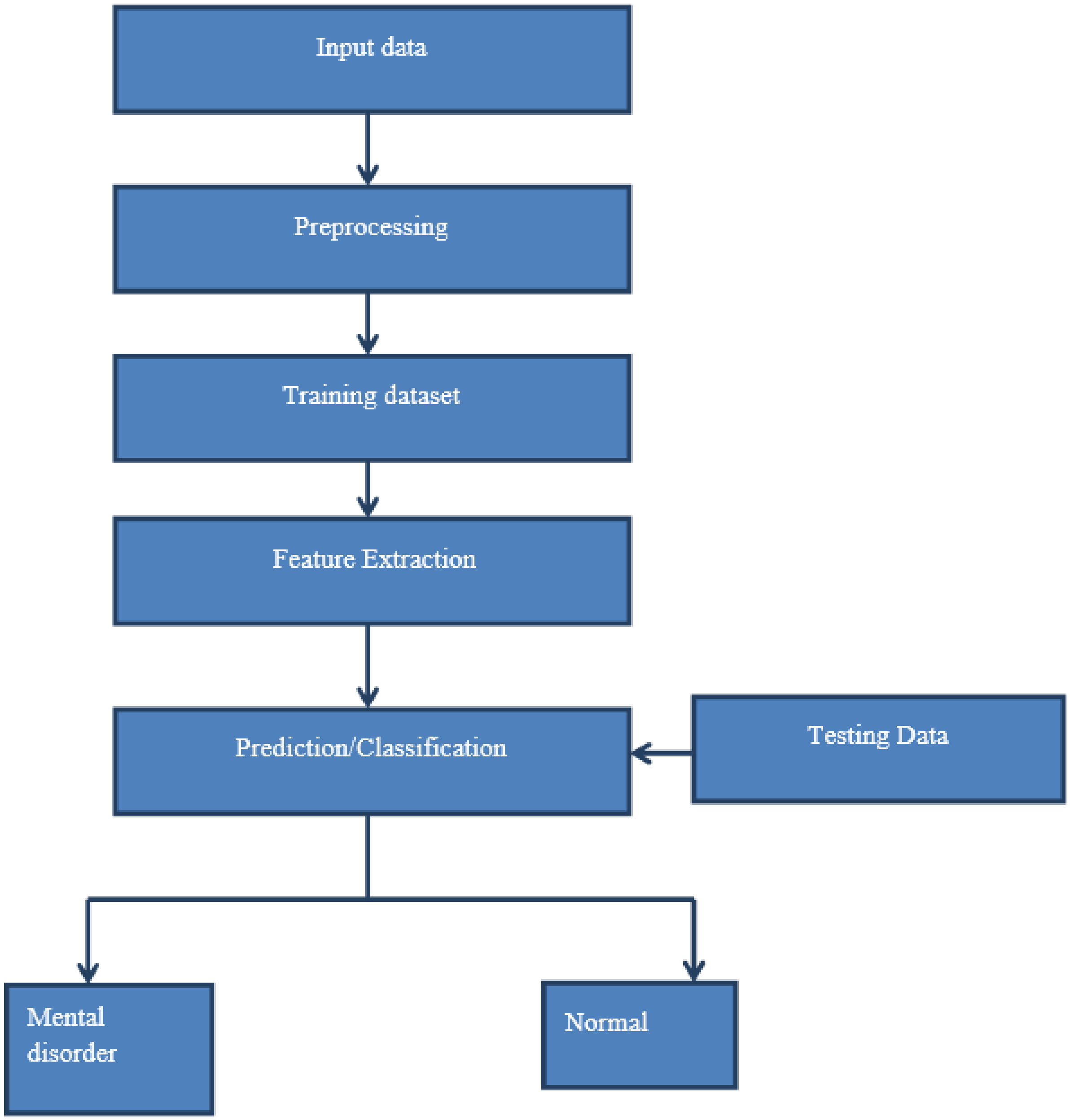
4.1.4 Activity Diagram


Input data
Preprocessing
Training
Prediction using proposed algorithm (random forest classification)
Predicted metal disorder or normal

EXPLANATION:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

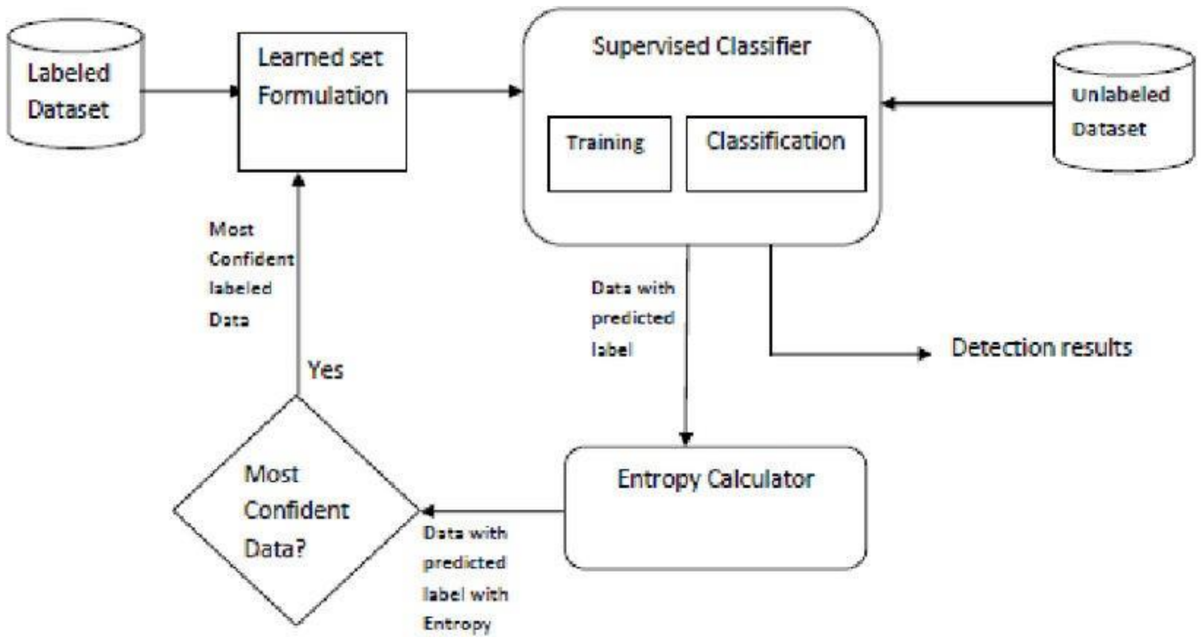
4.1.5 Data Flow Diagram:



EXPLANATION:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modelling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

4.2 System Architecture



CHAPTER 5

DEVELOPMENT TOOLS

5.1 GENERAL

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.

- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt −

```
$ python

Python2.4.3(#1,Nov112010,13:34:43)

[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type"help","copyright","credits"or"license"for more information.

>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result –

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods −

Sr.No	Methods & Description
1	**GET** Sends data in unencrypted form to the server. Most common method.
2	**HEAD** Same as GET, but without response body
3	**POST** Used to send HTML form data to server. Data received by POST method is not cached by server.
4	**PUT** Replaces all current representations of the target resource with the uploaded content.

5	**DELETE** Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

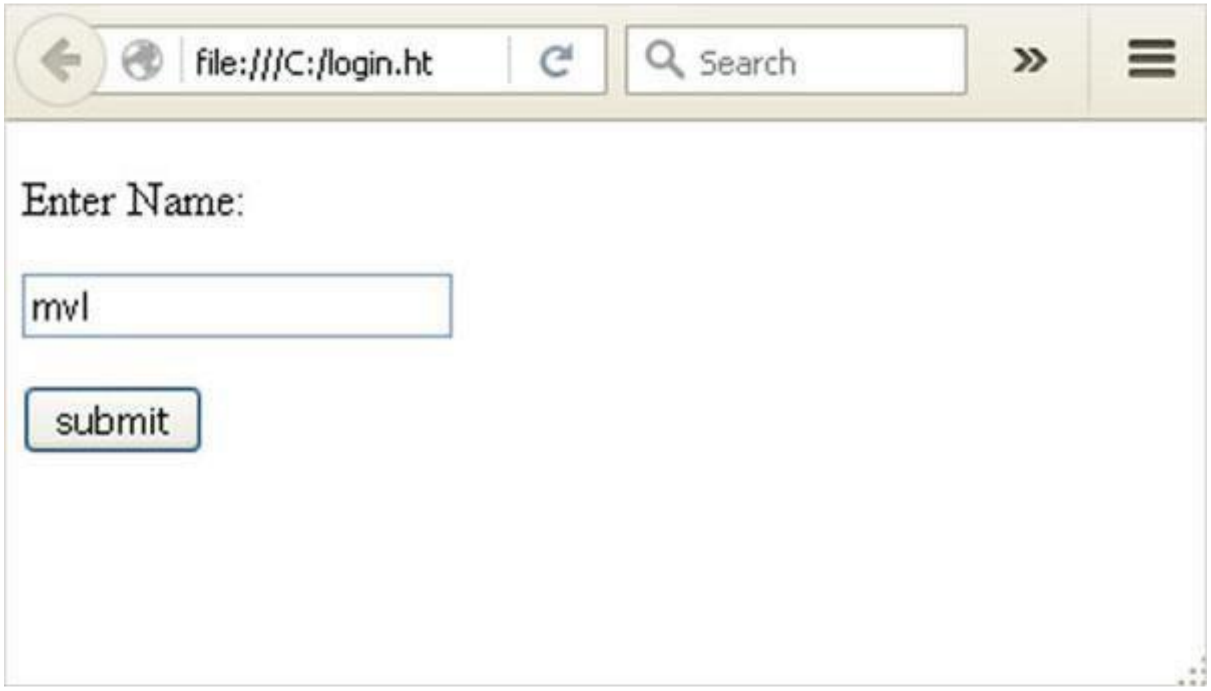
```
<html>

<body>

<formaction="http://localhost:5000/login"method="post">

<p>Enter Name:</p>

<p><inputtype="text"name="nm"/></p>

<p><inputtype="submit"value="submit"/></p>

</form>

</body>

</html>
```

Now enter the following script in Python shell.

```
from flask importFlask, redirect,url_for, request

app=Flask(__name__)

@app.route('/success/<name>')
```

```
def success(name):

return'welcome %s'% name

@app.route('/login',methods=['POST','GET'])

def login():

ifrequest.method=='POST':

user=request.form['nm']

return redirect(url_for('success',name= user))

else:

user=request.args.get('nm')

return redirect(url_for('success',name= user))

if __name__ =='__main__':

app.run(debug =True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.

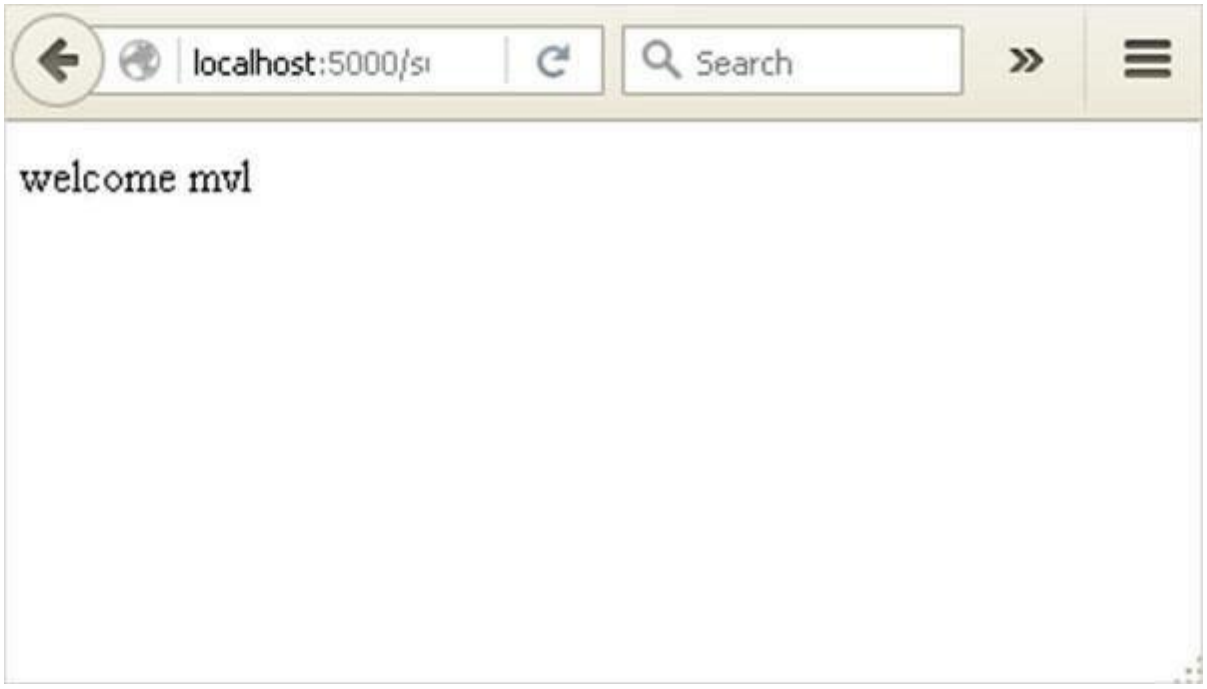


Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of 'nm' parameter obtained from the form data is obtained by −

```
user = request.form['nm']
```

It is passed to **'/success'** URL as variable part. The browser displays a **welcome** message in the window.



welcome mvl

Change the method parameter to **'GET'** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of 'nm' parameter is now obtained by −

```
User = request.args.get('nm')
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to 'nm' parameter is passed on to '/success' URL as before.

What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: https://www.python.org/

Python Quick start

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

helloworld.py

```
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> print("Hello, World!")
Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

C:\Users\Your Name>python myfile.py

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important.

Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:
  print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

Example

```
if 5 > 2:
print("Five is greater than two!")
```

Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python:

```
#This is a comment.
print("Hello, World!")
```

Docstrings

Python also has extended documentation capability, called docstrings.

Docstrings can be one line, or multiline.

Python uses triple quotes at the beginning and end of the docstring:

Example

Docstrings are also comments:

```
"""This is a multiline docstring."""
print("Hello, World!")
```

CHAPTER 6

IMPLEMENTATION

6.1 GENERAL

The Implementation is nothing but source code of project.

6.2 IMPLEMENTATION

Coding:

app.py:

```
import numpy as np

import pandas as pd

from flask import Flask, request, jsonify, render_template, redirect, flash, send_file

from sklearn.preprocessing import MinMaxScaler

from werkzeug.utils import secure_filename

import pickle

from sklearn.decomposition import PCA

app = Flask(_name_) #Initialize the flask App

model = pickle.load(open('survey.pkl', 'rb'))

@app.route('/')

@app.route('/')
```

```
@app.route('/index')

def index():

    return render_template('index.html')

@app.route('/abstract')

def abstract():

    return render_template('abstract.html')

@app.route('/future')

def future():

    return render_template('future.html')

@app.route('/login')

def login():

    return render_template('login.html')

@app.route('/upload')

def upload():

    return render_template('upload.html')

@app.route('/preview',methods=["POST"])

def preview():

    if request.method == 'POST':

        dataset = request.files['datasetfile']
        df = pd.read_csv(dataset,encoding = 'unicode_escape')
        df.set_index('Id', inplace=True)
        return render_template("preview.html",df_view = df)
@app.route('/home')
def home():
    return render_template('test.html')
@app.route('/predict',methods=['POST'])
def predict():
        features = [float(x) for x in request.form.values()]
    final_features = [np.array(features)]
    y_pred = model.predict(final_features)
    if y_pred[0] == 1:
        label="Mental disorder"
    elif y_pred[0] == 0:
        label="Normal person"
    return render_template('test.html', prediction_text=label)
@app.route('/chart')
def chart():
    return render_template('chart.html')
        if _name_ == "_main_":

    app.run(debug=False)
```

CHAPTER 7

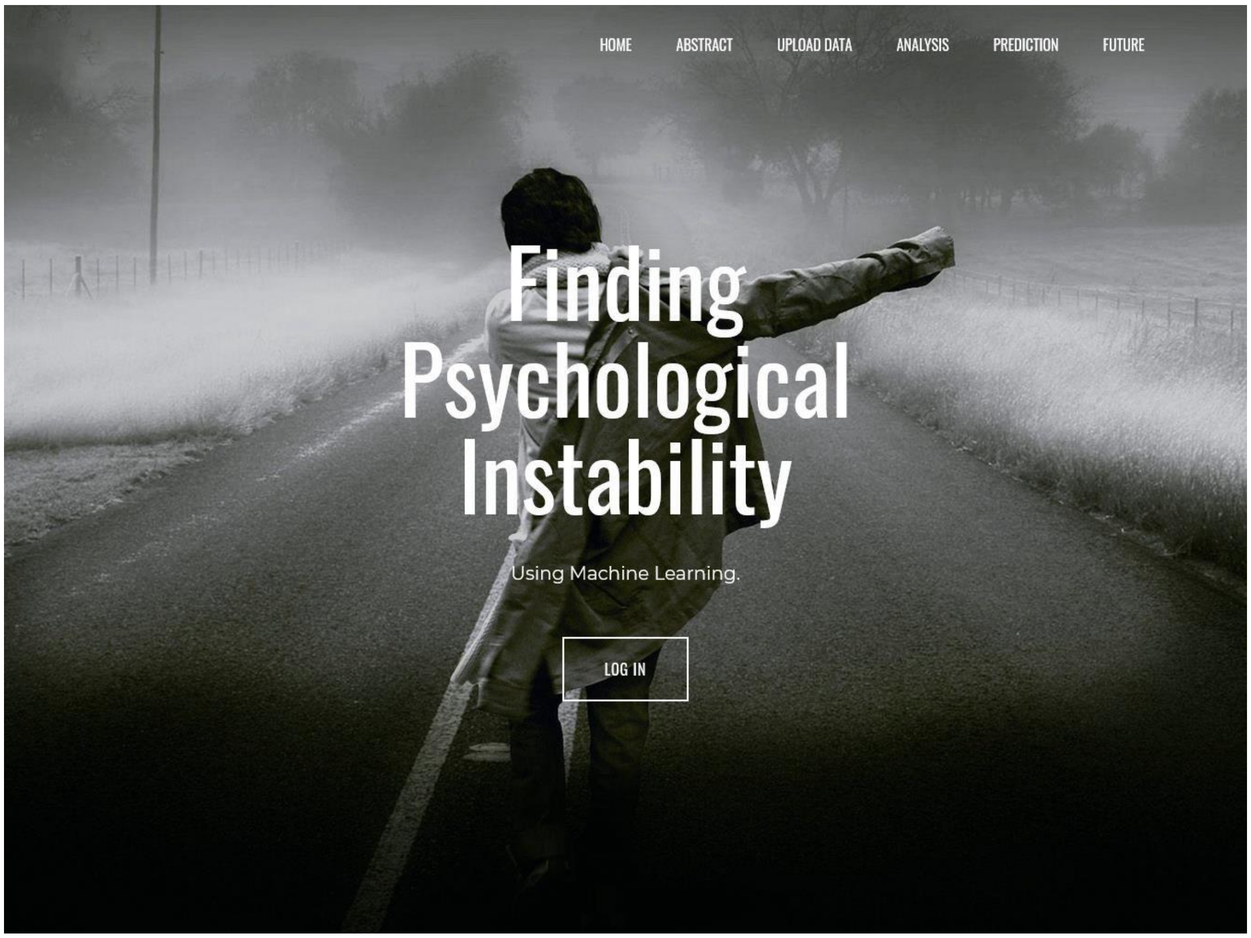
SNAPSHOTS

7.1 GENERAL

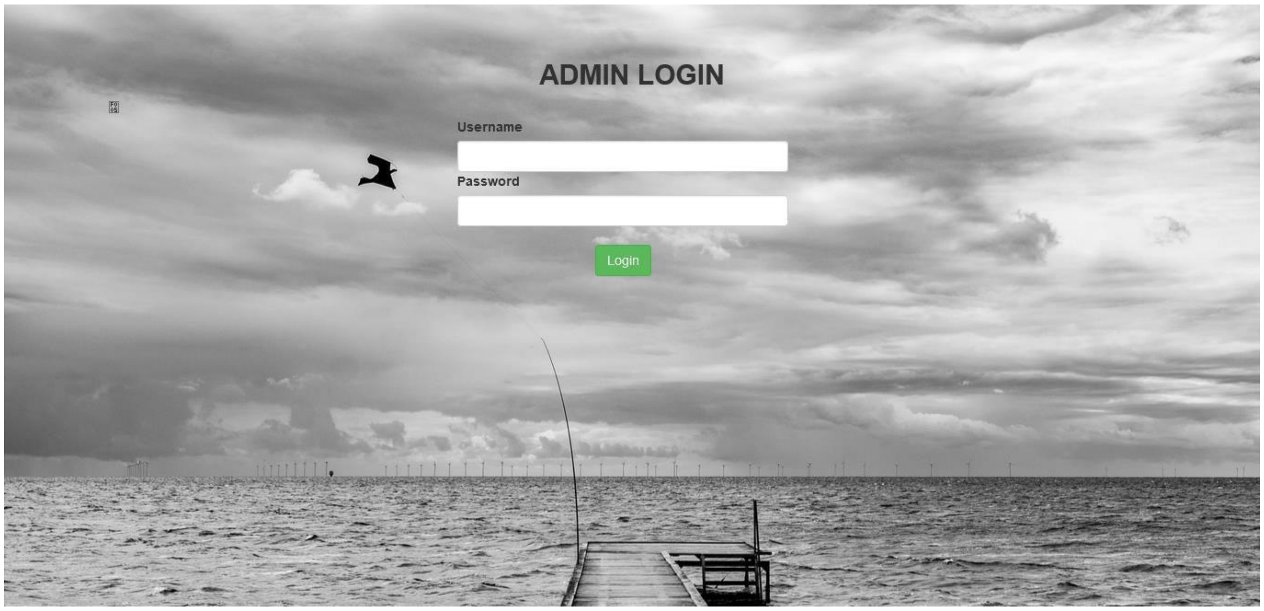
This project is implements like web application using Python and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

7.2 VARIOUS SNAPSHOTS

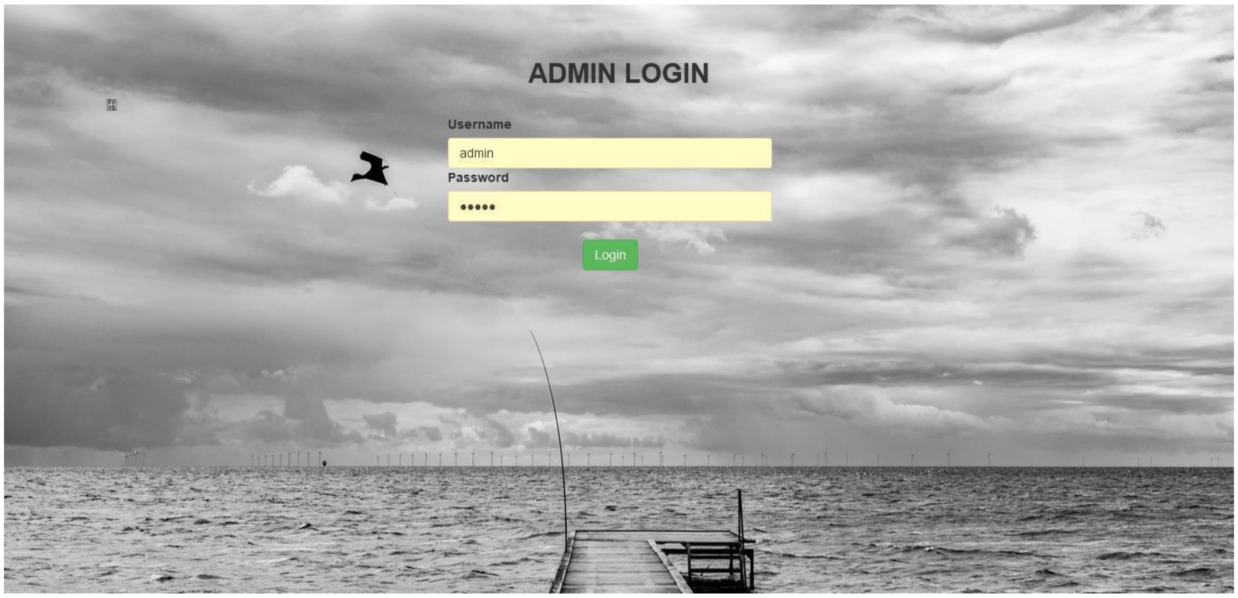
- Home Page



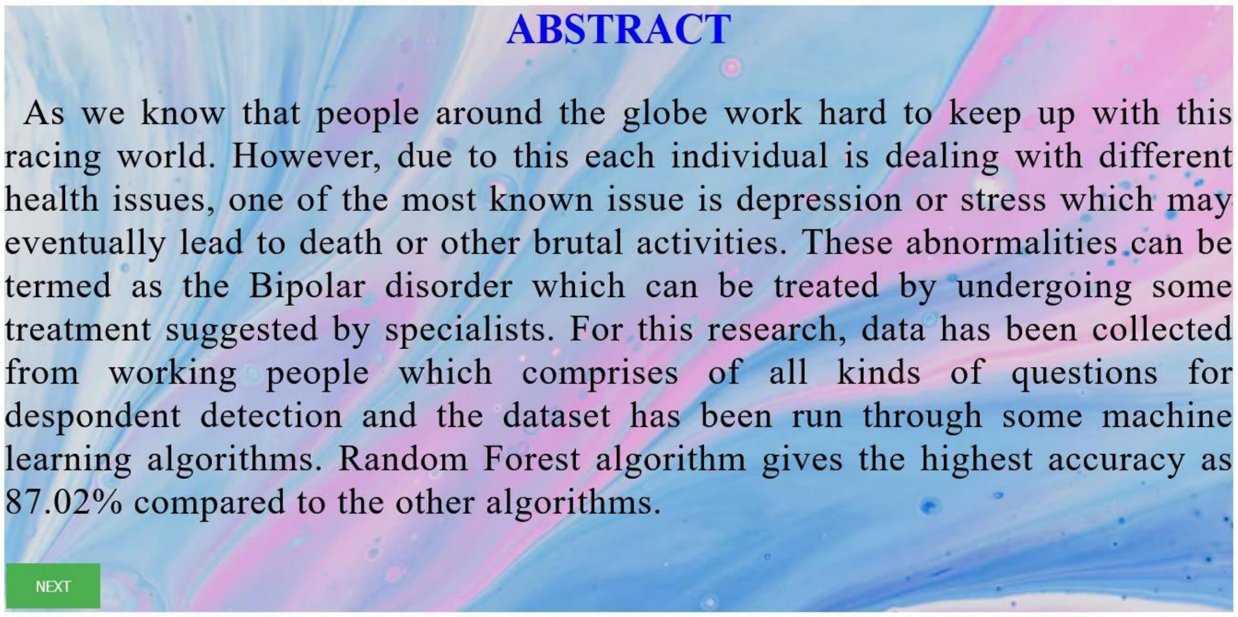
- Login Page



- Enter the Username and Password



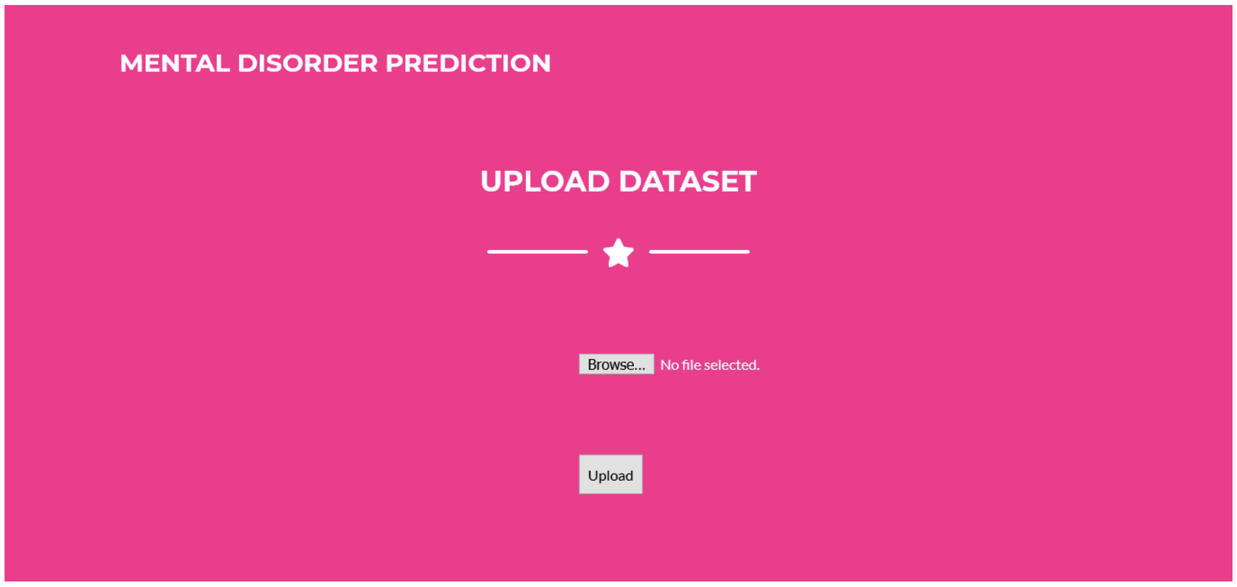
- Abstract Page



ABSTRACT

As we know that people around the globe work hard to keep up with this racing world. However, due to this each individual is dealing with different health issues, one of the most known issue is depression or stress which may eventually lead to death or other brutal activities. These abnormalities can be termed as the Bipolar disorder which can be treated by undergoing some treatment suggested by specialists. For this research, data has been collected from working people which comprises of all kinds of questions for despondent detection and the dataset has been run through some machine learning algorithms. Random Forest algorithm gives the highest accuracy as 87.02% compared to the other algorithms.

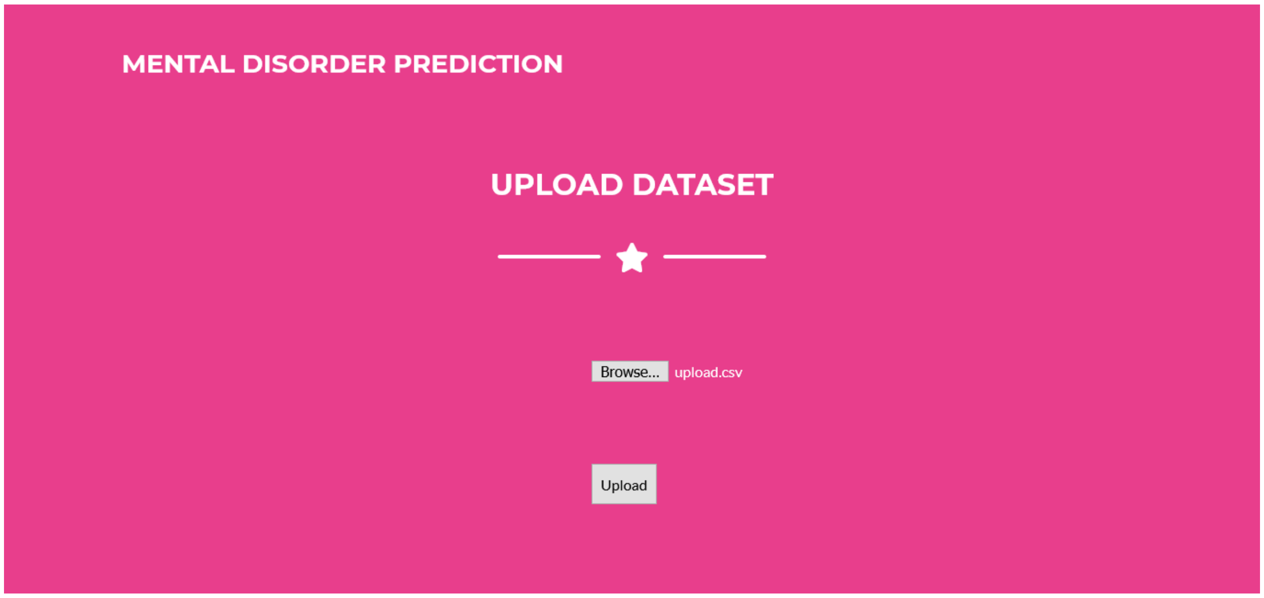
- Upload Page



MENTAL DISORDER PREDICTION

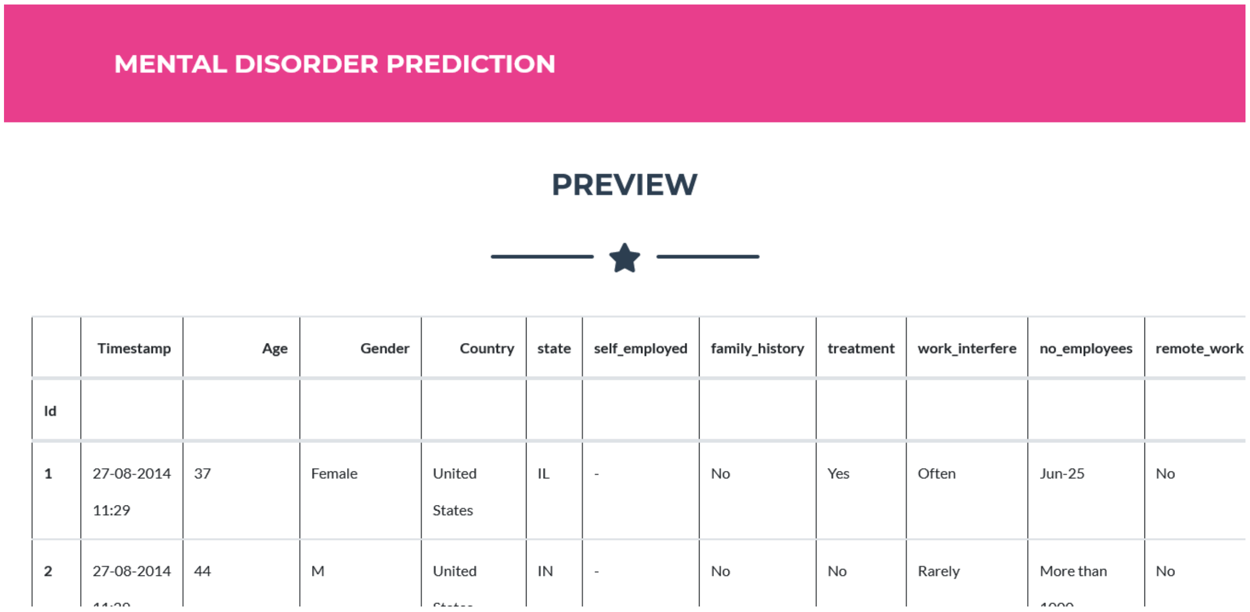
UPLOAD DATASET

- Upload the Dataset


MENTAL DISORDER PREDICTION

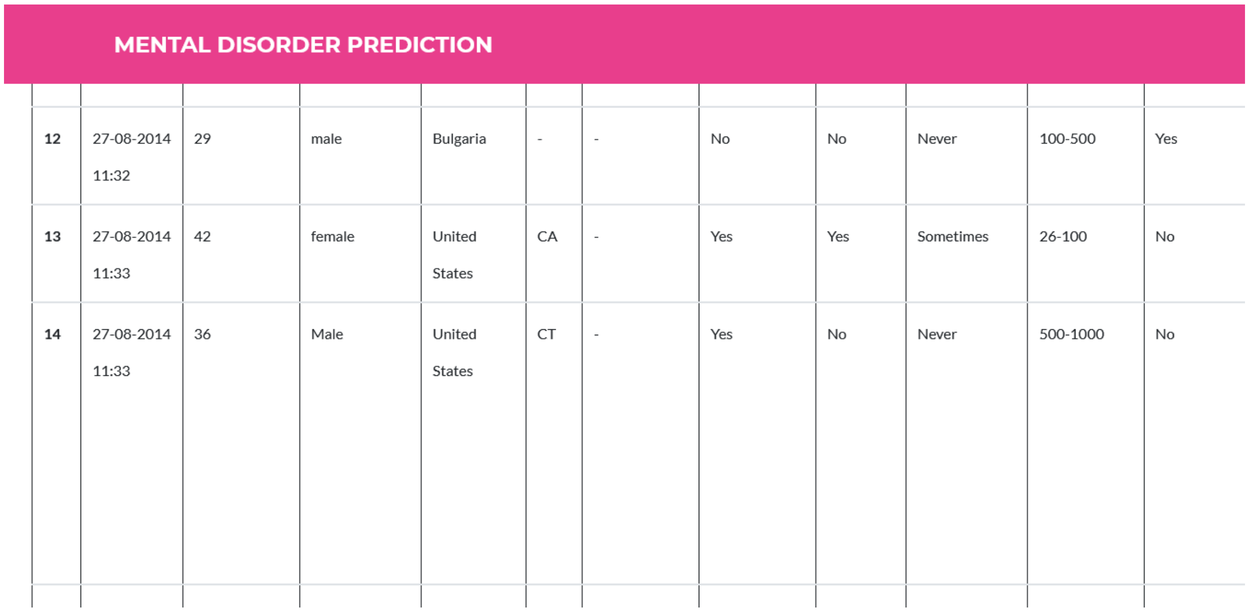
UPLOAD DATASET

- Preview of the dataset


MENTAL DISORDER PREDICTION

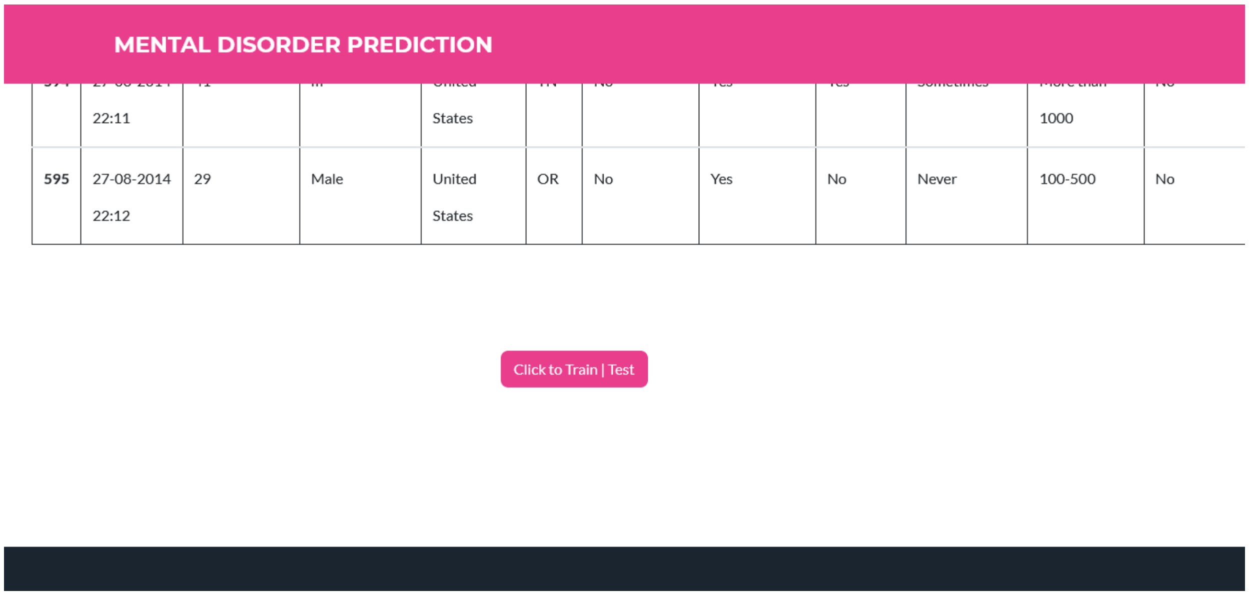
PREVIEW

	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	no_employees	remote_work
Id											
1	27-08-2014 11:29	37	Female	United States	IL	-	No	Yes	Often	Jun-25	No
2	27-08-2014	44	M	United	IN	-	No	No	Rarely	More than	No


MENTAL DISORDER PREDICTION

12	27-08-2014 11:32	29	male	Bulgaria	-	-	No	No	Never	100-500	Yes
13	27-08-2014 11:33	42	female	United States	CA	-	Yes	Yes	Sometimes	26-100	No
14	27-08-2014 11:33	36	Male	United States	CT	-	Yes	No	Never	500-1000	No

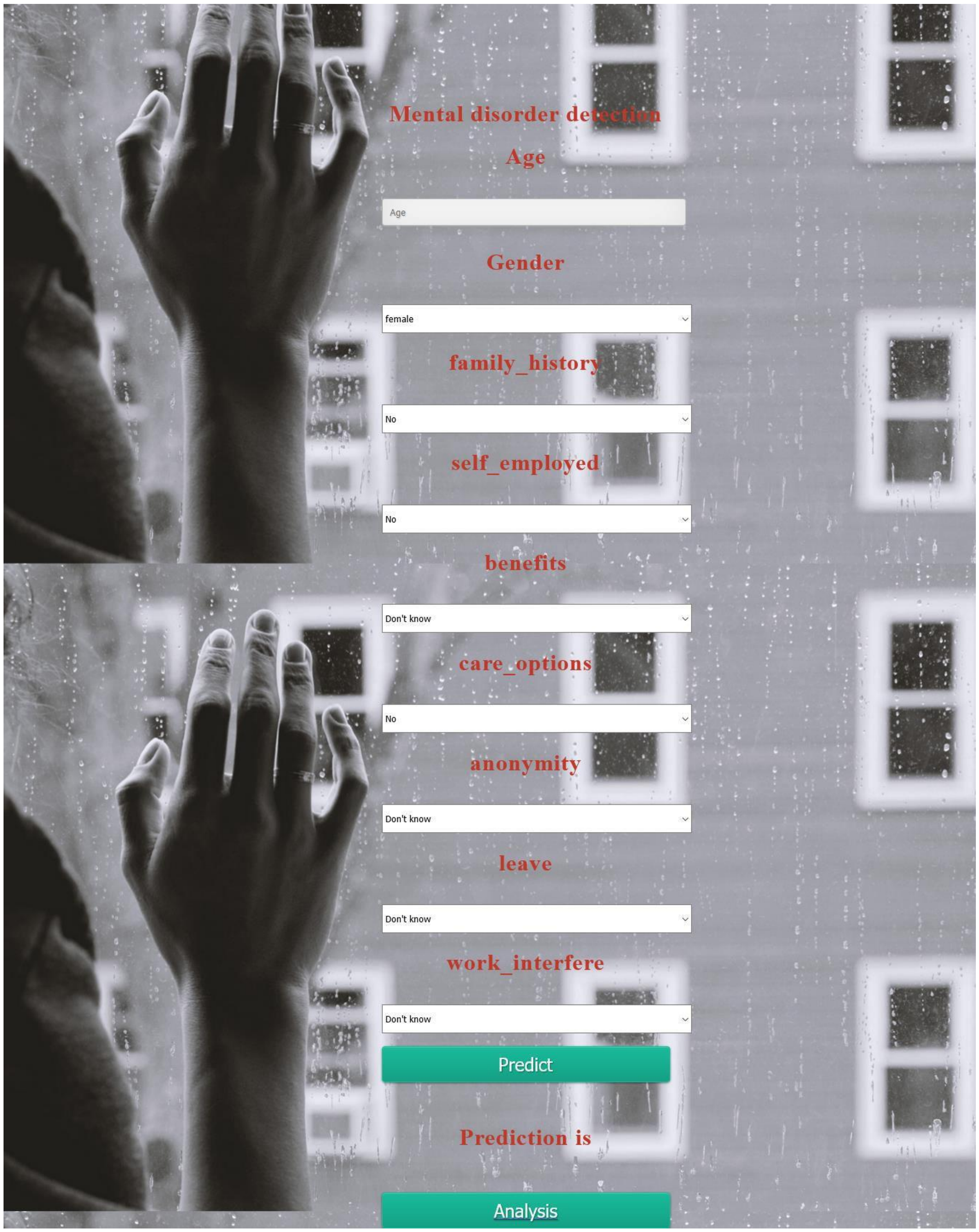
- At the end of the preview click to Train/Test


MENTAL DISORDER PREDICTION

	22:11			States						1000	
595	27-08-2014 22:12	29	Male	United States	OR	No	Yes	No	Never	100-500	No

Click to Train | Test

- Enter the details below based on test data and predict



Mental disorder detection

Age

Age

Gender

female

family_history

No

self_employed

No

benefits

Don't know

care_options

No

anonymity

Don't know

leave

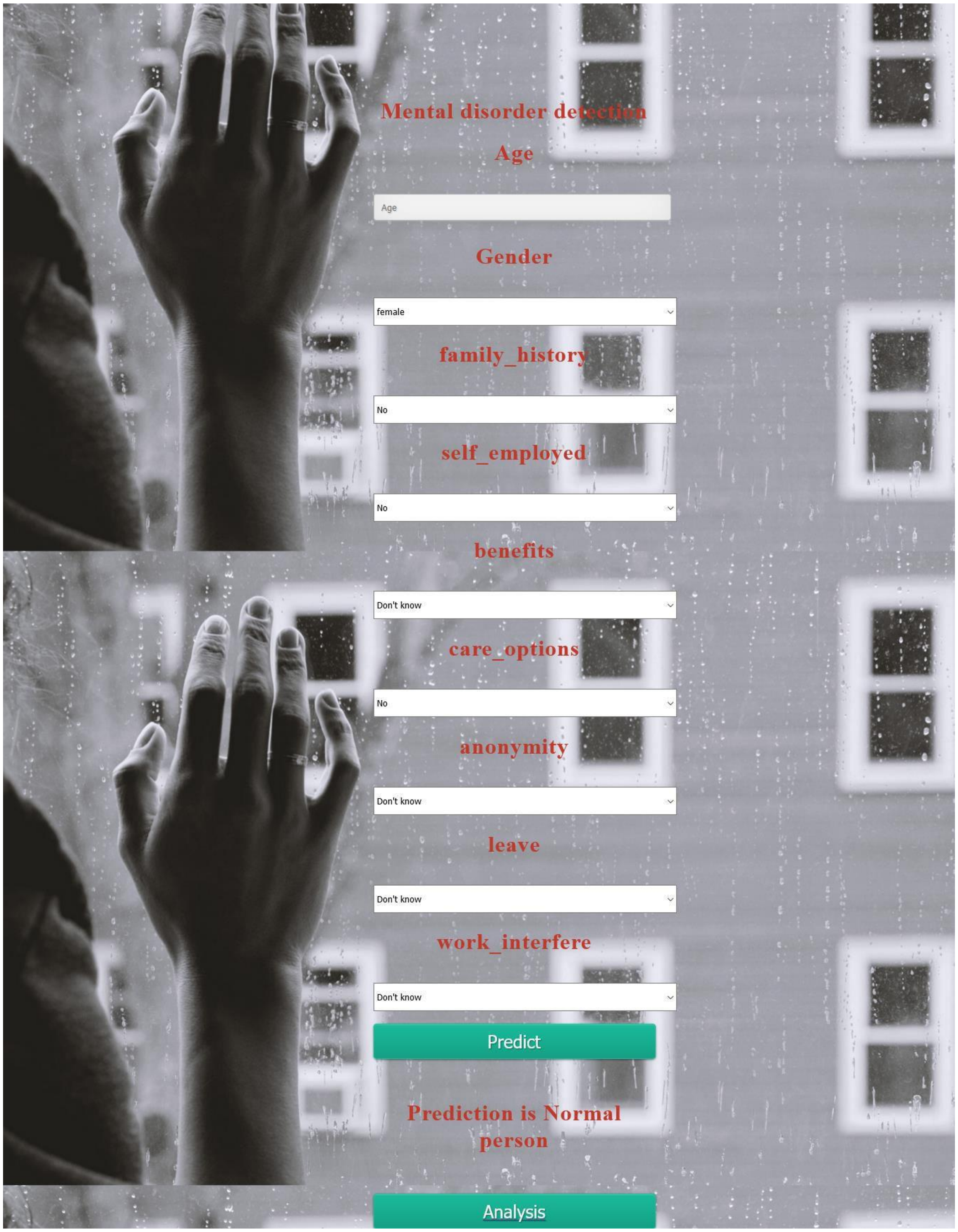
Don't know

work_interfere

Don't know

Prediction is

- The below are some of the predictions



Mental disorder detection

Age

Age

Gender

female

family_history

No

self_employed

No

benefits

Don't know

care_options

No

anonymity

Don't know

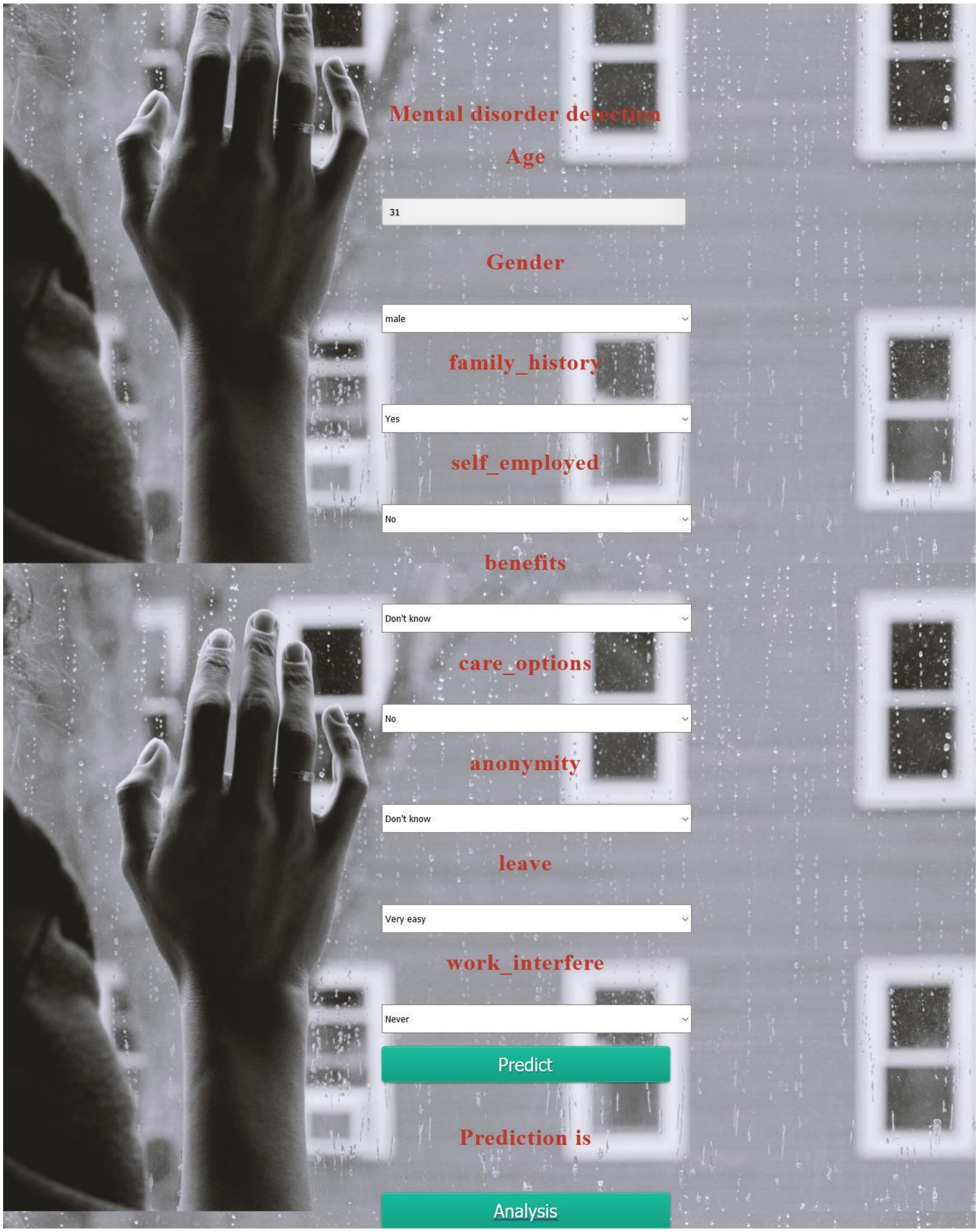
leave

Don't know

work_interfere

Don't know

Prediction is Normal person


Mental disorder detection

Age

31

Gender

male

family_history

Yes

self_employed

No

benefits

Don't know

care_options

No

anonymity

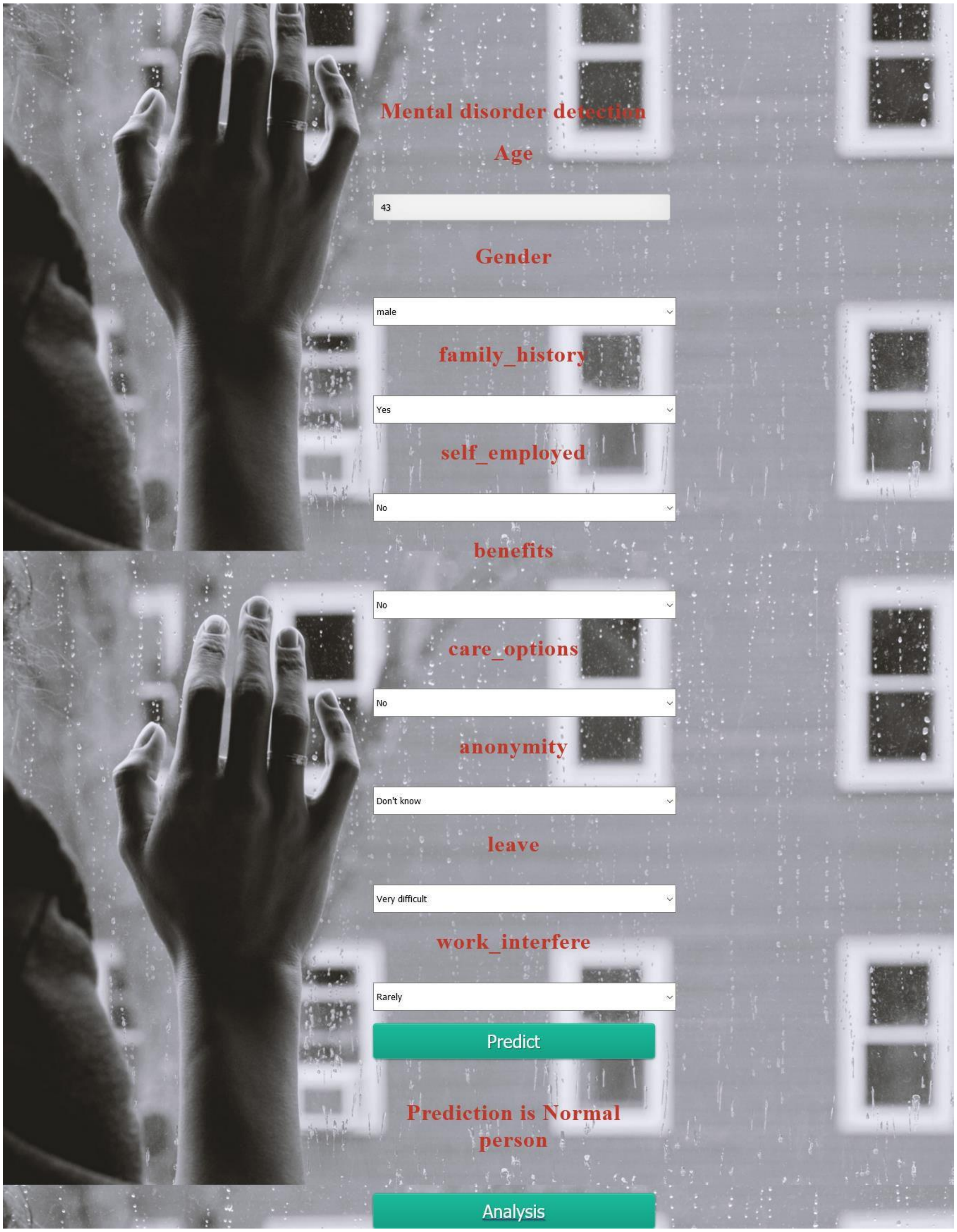
Don't know

leave

Very easy

work_interfere

Never



Mental disorder detection

Age

43

Gender

male

family_history

Yes

self_employed

No

benefits

No

care_options

No

anonymity

Don't know

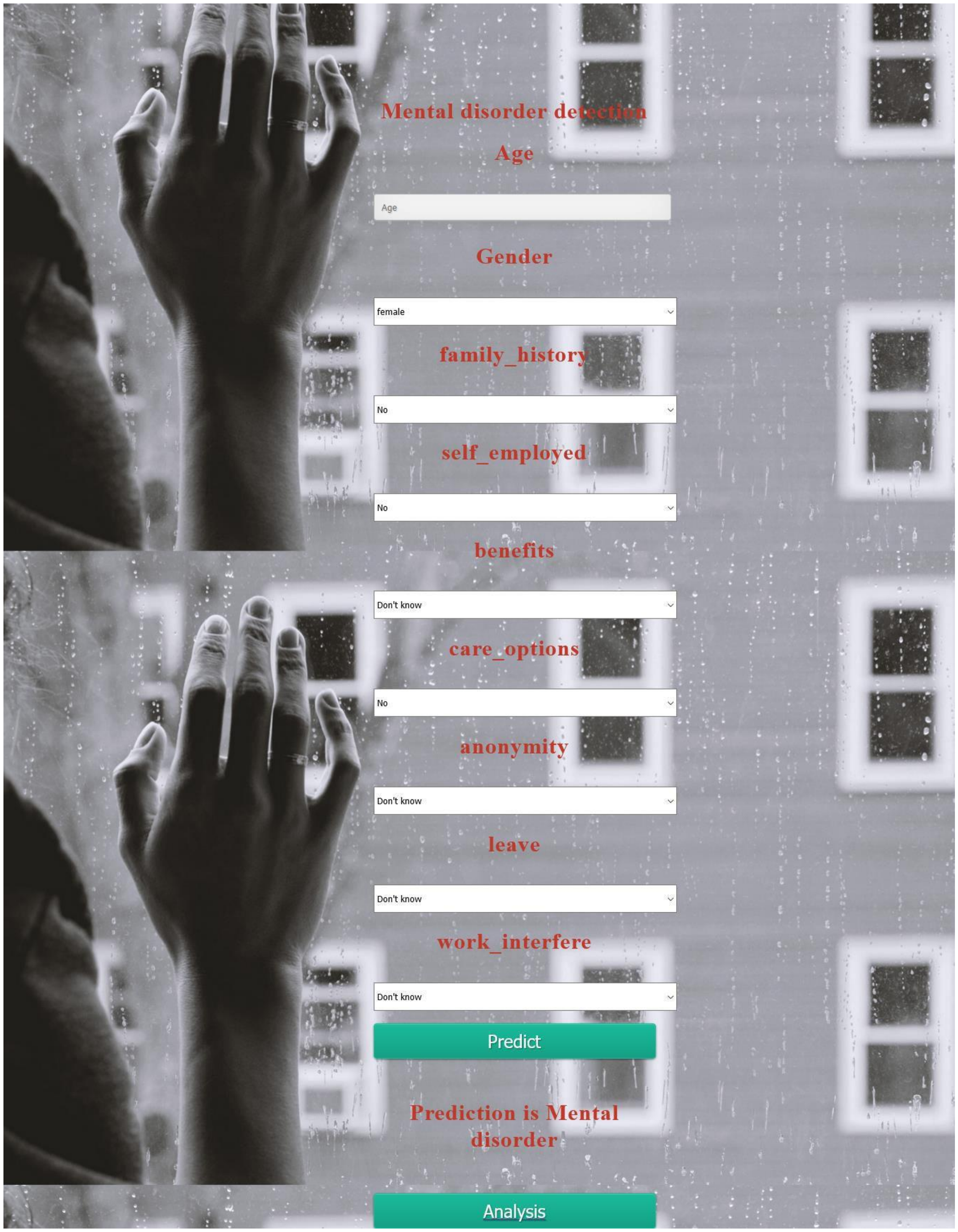
leave

Very difficult

work_interfere

Rarely

Prediction is Normal person


Mental disorder detection

Age

Age

Gender

female

family_history

No

self_employed

No

benefits

Don't know

care_options

No

anonymity

Don't know

leave

Don't know

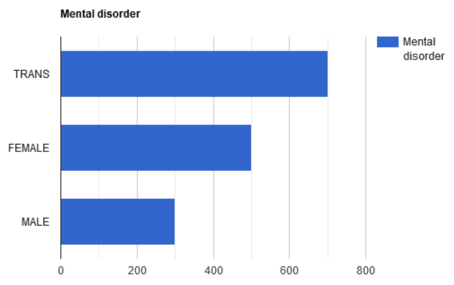
work_interfere

Don't know

Prediction is Mental disorder

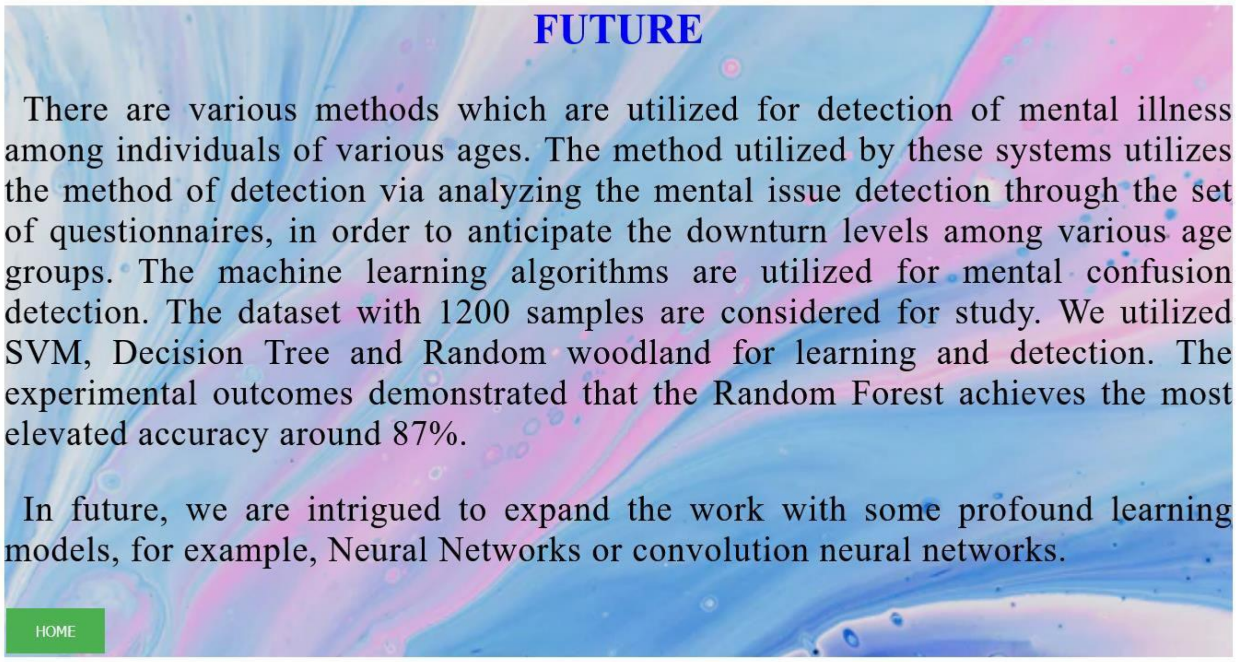
- Chart Page

  Based on the results the below chart represents the number of persons according to their gender who has mental disorder



FUTURE

- Future Page



FUTURE

There are various methods which are utilized for detection of mental illness among individuals of various ages. The method utilized by these systems utilizes the method of detection via analyzing the mental issue detection through the set of questionnaires, in order to anticipate the downturn levels among various age groups. The machine learning algorithms are utilized for mental confusion detection. The dataset with 1200 samples are considered for study. We utilized SVM, Decision Tree and Random woodland for learning and detection. The experimental outcomes demonstrated that the Random Forest achieves the most elevated accuracy around 87%.

In future, we are intrigued to expand the work with some profound learning models, for example, Neural Networks or convolution neural networks.

CHAPTER 8
SOFTWARE TESTING

8.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every 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.

8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

8.3 Types of Tests

8.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input 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.

8.3.2 Functional test

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 centred 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.

8.3.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.3.4 Performance Test

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

8.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

8.3.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- ➢ The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- ➢ The Route add operation is done only when there is a Route request in need
- ➢ The Status of Nodes information is done automatically in the Cache Updation process

8.3.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

8.4 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

8.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 9

APPLICATION

9.1 GENERAL

The system is developed using Python language with required libraries. Implemented using three machine learning algorithms on the given dataset for mental disorder detection shows that Random forest model outperforms other models. SVM and Random forest algorithms have high accuracy compared to other Decision Tree algorithm.

9.2 FUTURE ENHANCEMENT

In future, we are intrigued to expand the work with some profound learning models, for example, Neural Networks or convolution neural networks.

CHAPTER 10

CONCLUSION & REFERENCE

10.1 CONCLUSION

There are various methods which are utilized for detection of mental illness among individuals of various ages. The method utilized by these systems utilizes the method of detection via analyzing the mental issue detection through the set of questionnaires, in order to anticipate the downturn levels among various age groups. The machine learning algorithms are utilized for mental confusion detection. The dataset with 1200 samples are considered for study. We utilized SVM, Decision Tree and Random woodland for learning and detection. The experimental outcomes demonstrated that the Random Forest achieves the most elevated accuracy around 87%.

10.2 REFERENCE

[1] Mental Disorder Detection : Bipolar Disorder Scrutinization using Machine Learning, published in 2019.

[2] Intelligent data mining and machine learning for mental health diagnosis using genetic algorithmAzar, Ghassan &Gloster, Clay & El-Bathy, Naser & Yu, Su&Neela, Rajasree&Alothman, Israa. (2015). Intelligent data mining and machine learning for mental health diagnosis using genetic algorithm. 201-206. 10.1109/EIT.2015.7293425

[3] A Framework for Classifying Online Mental Health-Related Communities With an Interest in Depression B. Saha, T. Nguyen, D. Phung and S. Venkatesh, "A Framework for Classifying Online Mental Health-Related Communities With an Interest in Depression," in IEEE Journal of Biomedical and Health Informatics, vol. 20, no. 4, pp. 1008-1015, July 2016.

[4] Detecting Cognitive Distortions Through Machine Learning Text AnalyticsT. Simms, C. Ramstedt, M. Rich, M. Richards, T. Martinez and C. Giraud-Carrier, "Detecting Cognitive Distortions Through Machine Learning Text Analytics," 2017 IEEE International Conference on Healthcare Informatics (ICHI), Park City, UT, 2017, pp. 508-512.

[5] Machine Learning Framework for the Detection of Mental Stress at Multiple Levels Subhani, Ahmad & Mumtaz, Wajid & MOHAMAD SAAD, MOHAMAD NAUFAL & Kamel, Nidal& Malik, Aamir. (2017). Machine Learning Framework

for the Detection of Mental Stress at Multiple Levels. IEEE Access. PP. 1-1. 10.1109/ACCESS.2017.2723622.

[6] Prediction of Mental Health Problems Among Children Using Machine Learning TechniquesSumathi, Ms & B., Dr. (2016). Prediction of Mental Health Problems Among Children Using Machine Learning Techniques. International Journal of Advanced Computer Science and Applications. 10.14569/IJACSA.2016.070176.