CHAPTER 1

INTRODUCTION

Mental health significantly influences individuals' emotions, thoughts, and behaviors, and is a critical component of overall well-being. In recent years, rising stress levels, lifestyle changes, and societal pressures have contributed to an increase in mental health issues such as anxiety, depression, and mood disorders across various age groups. Despite growing awareness, many individuals continue to suffer in silence due to stigma, lack of awareness, or delayed diagnosis. Early detection and timely intervention play a crucial role in managing mental health conditions and improving individuals' quality of life.

With the availability of survey data capturing behavioral, emotional, and workplace-related experiences, there is a growing opportunity to apply data science techniques to identify patterns and potential risk factors associated with mental health challenges. This project leverages such data to support predictive modeling for early mental health disorder detection.

1.1 Problem Statement

Mental health issues are often underdiagnosed or misdiagnosed due to the subjective nature of symptoms and the reliance on self-reporting. Traditional diagnostic methods are time-consuming and may not always capture early warning signs, especially in environments where mental health discussions are stigmatized. As a result, individuals may not receive the necessary care in time, leading to worsening conditions. There is a need for an efficient and data-driven approach to identify individuals at risk and promote early intervention based on behavioral and workplace indicators.

1.2 Objectives

- To analyze mental health survey data from individuals of varying countries, genders, and professions.
- To identify key behavioral, emotional, and workplace-related factors that influence mental health.
- To build and evaluate machine learning models for early detection of mental health challenges.

- To provide actionable insights that can support mental wellness strategies in personal and professional environments.

1.3 Scope and Limitations

Scope:

- The project focuses on analyzing structured survey data related to mental health indicators such as stress levels, mood swings, coping struggles, and treatment interest.
- It involves the application of machine learning algorithms to perform binary classification tasks (e.g., whether a person needs treatment).
- The study aims to provide a data-driven framework for early detection, not a clinical diagnosis.

Limitations:

- The dataset is self-reported, which may introduce bias or inaccuracies.
- The analysis is limited to the features available in the dataset and does not consider clinical or neurological data.
- Results may not generalize to populations outside the surveyed group or regions with different cultural attitudes toward mental health.
- The models are predictive tools and should not replace professional mental health evaluation or treatment.

CHAPTER 2

LITERATURE REVIEW

Sharma et al. (2025) proposed *NeuroVibeNet*, a novel multi-modal framework for early mental health disorder detection. The model combined Improved Random Forest and LightGBM for behavioral data, and a hybrid of SVM and KNN for voice data. Preprocessing included handling missing values, normalization, and outlier removal. Using a weighted voting mechanism, the model achieved 99.06% accuracy, validating the effectiveness of speech and behavior-based integration.

Rahman et al. (2019) conducted a systematic review of machine learning applications in mental health detection using Online Social Networks (OSNs). From 2770 screened studies, 22 were analyzed based on datasets, ML methods, and feature extraction techniques. The majority focused on text analysis from platforms like Twitter and Facebook, with some incorporating user consented data and questionnaires. The review highlighted the potential of OSN-based approaches as cost-effective alternatives to traditional methods, while also noting challenges such as algorithmic complexity, data quality, and the need for expert validation.

Khoo et al. (2024) conducted a systematic review of 184 studies focusing on machine learning approaches for mental health detection using passively sensed multimodal data. The review examined data from audio, video, social media, smartphones, and wearables. Key trends included the use of neural networks for model-level fusion and the increasing emphasis on feature-level integration from multiple sources. The study emphasized that individual context—such as demographics and personality traits—affects modality relevance, and highlighted the importance of choosing optimal data sources based on specific mental health disorders.

Costello (2016) explored the role of developmental epidemiology in the early detection and prevention of mental health problems in children and adolescents. The study differentiated between universal prevention, aimed at whole populations before symptoms emerge, and targeted prevention, focused on high-risk individuals. It highlighted how the timing of exposure, symptom onset, and intervention influences mental health outcomes. The paper emphasized that early detection, supported by effective service systems, can reduce long-term risks, although it also acknowledged potential downsides of premature or unnecessary interventions.

Rahman et al. (2020) conducted a survey on mental health detection using Online Social Networks (OSNs), highlighting OSNs like Twitter as rich sources of behavioral data. The study found that machine learning techniques, especially Support Vector Machine (SVM), are the most commonly used for classification tasks. English was the dominant language for analysis, with a strong reliance on text data. The review identified key challenges including language barriers, privacy restrictions, limited feature sets, and dependence on a single OSN. Future directions suggested incorporating multimedia content and multilingual analysis based on user location.

Hassantabar et al. (2023) developed MHDeep, a deep learning-based system for detecting schizoaffective, major depressive, and bipolar disorders using data from wearable medical sensors. The system utilizes physiological, motion, and environmental data collected from smartwatches and smartphones. MHDeep eliminates manual feature engineering by working directly on raw sensor data and enhances training with synthetic data generation. Using a grow-and-prune neural network approach, it achieved patient-level accuracy of 100% for schizoaffective and depressive disorders, and 90% for bipolar disorder, demonstrating the potential of wearable-based passive sensing in real-time mental health monitoring.

Tiwari et al. (2024) reviewed recent advancements in stress detection using EEG signals as a non-invasive and real-time method. The study focused on signal processing, feature extraction, and machine learning algorithms applied to EEG data for identifying stress. Key challenges discussed include variability in EEG patterns, noise reduction, and the need for personalized models. The review emphasized the potential of EEG-based systems for effective and timely mental health monitoring, highlighting opportunities for improving detection accuracy through advanced ML techniques.

Kim et al. (2020) conducted a bibliometric study analyzing research trends in machine learning applications for mental health detection using social media data. The review included 565 publications from 2015 to 2020, showing steady growth in this field. The study identified key sources, countries, institutions, and methodologies, with Lecture Notes in Computer Science and Journal of Medical Internet Research being the most productive outlets. Keyword co-occurrence networks were used to map major themes, and highly cited studies were examined for methodological insights, confirming social media's growing role in large-scale mental health analysis.

Liu et al. (2022) conducted a systematic review of 17 studies exploring machine learning methods for detecting depression through text data on social media. Most studies used supervised learning techniques, with a few employing unsupervised approaches. Depression identification was based on researcher inference, self-disclosure, or community participation. The review highlighted challenges such as data sampling, model generalizability, and ethical concerns including privacy. Despite limitations, the study concluded that ML-based depression detection on social media can be a valuable supplement to traditional mental health practices.

Garg (2023) presented a comprehensive survey analyzing over 92 research articles on mental health detection from social media posts using machine learning and deep learning techniques. The study proposed a taxonomy of research directions, covering stress, depression, and suicide detection. Key aspects included feature extraction, classification methods, AI model advancements, and the use of public datasets. The survey emphasized the growing importance of real-time, responsible AI in social computing and introduced an open-source repository to support further research and accessibility in this domain.

CHAPTER 3

SYSTEM ANALYSIS

3.1 System Requirements

System environment specifies the hardware and software configuration of the new system. Regardless of how the requirement phase proceeds it ultimately ends with the software requirements specification. A good SRS contains all the system requirements to a level of detail sufficient to enable designer to design a system that satisfies those requirements. The system specified in the SRS will assist the potential users to determine if the system meets their needs or how the system must be modified to meet their needs.

3.1.1 Software Environment

- Front End : Python
- IDE : Jupyter Notebook
- Operating System : Windows

1.Python

Python is an interpreted high level programming language for general purpose programming. Created by Guido Van Rossum and first released in 1991. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and the automatic memory management. It supports multiple programming paradigms, including object oriented imperative, functional and the procedural, and has a large and the comprehensive standard library. Python interpreters are available for many operating systems. It has a wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). Python is widely used high-level programming language for general-purpose programming. Apart from being an open source programming language, python is a great object-oriented, interpreted, and interactive programming language. Python combines remarkable power with very learn syntax. It has modules, classes, exceptions, very high-level dynamic data types, and dynamic typing. There are interfaces to many systems calls and libraries, as well as to various windowing systems.

2. IDE

Jupyter Notebook Framework:

Jupyter notebook is an open source web application that allow users to create and share documents that contain the code, equations, visualizations and narrative text. The framework is widely used in data science and machine learning because it provide an interactive environment for developing and presenting code-based projects. It support several programming languages, including python, R, Julia and Scala, making it a popular choice for a wide range of applications. The notebook interface is accessible through a web browser, allowing users to run code and analyze data from anywhere with an internet connection.

3. Google Colab

Google Colab is a free cloud-based platform that allows users to write and execute Python code in a collaborative environment. It's 1 particularly well-suited for machine learning and data science tasks, offering access to powerful computing resources like GPUs and TPUs. Colab notebooks can be easily shared and edited in real-time, making it a great tool for learning, research, and collaboration.

3.1.2 Hardware Environment

- **RAM** **:** 7.688 GB
- **Disk space** **:** 240.29 GB
- **Processor** **:** Intel CORE i5
- **Display** **:** 1536 x 864

3.2 System Architecture

The system architecture consists of several key modules that work together for mental health prediction using machine learning techniques.

- Data Collection: Survey data is collected in CSV format.
- Data Preprocessing: Involves cleaning, encoding categorical data, and splitting into training and testing sets.
- Feature Selection: Identifies the most relevant features affecting mental health outcomes.

- Model Training: Machine learning models such as ANN, Random Forest, KNN, XG Boost, Ada Boost and Logistic Regression are trained and tested.
- Evaluation: Model performance is measured using metrics like accuracy, F1-score, and confusion matrix.
- Visualization: Power BI is used for country-wise and gender-based analysis.

3.3 Tools and Technologies used

- **Python:**
  Python is a powerful, general-purpose programming language that has gained widespread popularity in recent years. It is easy to learn and has a simple syntax, making it a popular choice for beginners.
- **Numpy:**
  Numpy is a numerical computing library for Python that provides a powerful array computing capability for Python.NumPy is built around a powerful N-dimensional array object,which allows for efficient storage and manipulation of large arrays of numerical data.
- **Pandas:**
  Pandas is a data analysis and manipulation library for Python.It provides data structures for efficiently storing and manipulating large datasets and tools for working with tabular data.
- **Scikit-learn:**
  Scikit-learn is a machine learning library for Python that provides a wide range of supervised and unsupervised learning algorithms. It is built on top of NumPy, Panda and Matplotlib and provides a consistent interface for working with different machime learning algorithms.
- **Tensorflow:**
  TensorFlow is an open-source deep learning framework developed by Google. It is designed to simplify the process of building, training, and deploying machine learning models, particularly deep neural networks. TensorFlow provides a comprehensive ecosystem of tools, libraries, and resources for developing and deploying machine learning applications efficiently.

- **Keras:**

  Keras is a high-level neural network library written in Python that serves as an interface to various deep learning frameworks, including TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK). Developed with a focus on enabling fast experimentation and prototyping of neural networks, Keras provides a user-friendly API that abstracts away much of the complexity of low-level deep learning frameworks.

- **Matplotlib:**

  Matplotlib is a data visualization library for python. It provides a range of tools for creating high-quality visualization, including line plot, scatter plots, bar charts, histograms, and more. Matplotlib is built on top of NumPy and provides a simple and flexible interface for creating complex visualizations.

- **Seaborn:**

  Seaborn is another data visualization library for Python. It provides a high-level interface for creating complex visualizations, including heatmaps, categorical plots and time series plots. It also provides tools for working with large datasets, including automatic data aggregation and binning.

- **Power BI**

  Power BI is a business analytics tool developed by Microsoft that enables users to visualize data and share insights through interactive dashboards and reports. It supports a wide range of data sources and offers powerful features for data transformation, aggregation, and visualization. It is especially useful for presenting machine learning results and survey analysis in a clear, user-friendly format.

3.4 Work Flow Diagram

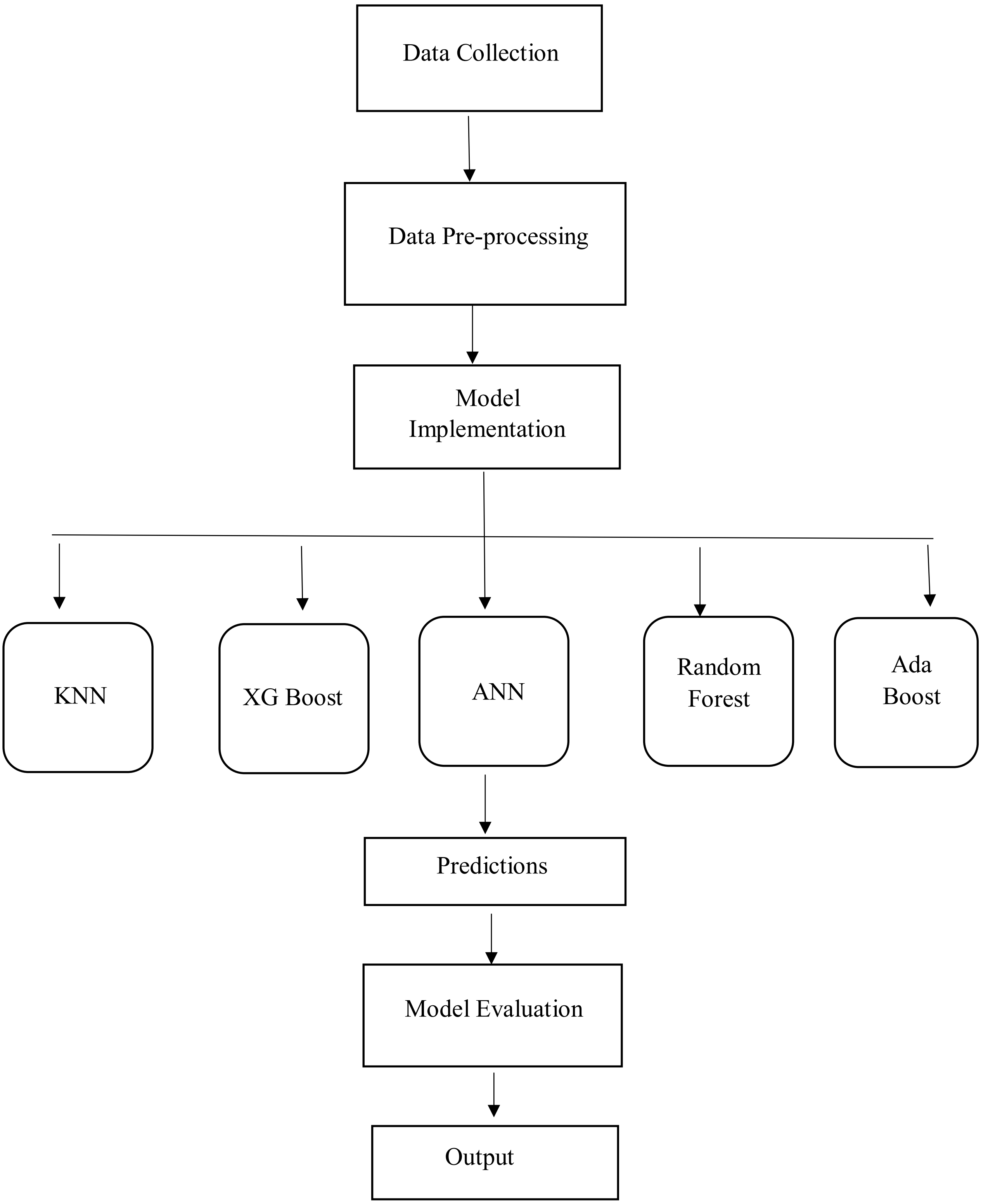

Figure 1 : Work flow

CHAPTER 4

METHODOLOGY

4.1 Data Collection

The dataset was sourced from publicly available repositories such as Kaggle, originally collected through an anonymous online survey focused on mental health in tech workplaces. Respondents voluntarily provided information regarding their mental well-being and related experiences. No personally identifiable information is included, ensuring privacy and ethical data use.

Dataset Description

- This dataset has been taken from kaggle.
- The dataset comprised of individual responses regarding mental health, including demographic details, workplace environment, stress levels, and treatment history.
- The dataset contains 2,92,364 rows and 17 columns.

	Timestamp	Gender	Country	Occupation	self_employed	family_history	treatment	Days_Indoors	Growing_Stress	Changes_Habits	Mental_Health_Hist
0	8/27/2014 11:29	Female	United States	Corporate	NaN	No	Yes	1-14 days	Yes	No	
1	8/27/2014 11:31	Female	United States	Corporate	NaN	Yes	Yes	1-14 days	Yes	No	
2	8/27/2014 11:32	Female	United States	Corporate	NaN	Yes	Yes	1-14 days	Yes	No	
3	8/27/2014 11:37	Female	United States	Corporate	No	Yes	Yes	1-14 days	Yes	No	
4	8/27/2014 11:43	Female	United States	Corporate	No	Yes	Yes	1-14 days	Yes	No	
...	...	...	...	...	...	...	...	...	...	...	
292359	7/27/2015 23:25	Male	United States	Business	Yes	Yes	Yes	15-30 days	No	Maybe	
292360	8/17/2015 9:38	Male	South Africa	Business	No	Yes	Yes	15-30 days	No	Maybe	
292361	8/25/2015 19:59	Male	United States	Business	No	Yes	No	15-30 days	No	Maybe	
292362	9/26/2015 1:07	Male	United States	Business	No	Yes	Yes	15-30 days	No	Maybe	
292363	2/1/2016 23:04	Male	United States	Business	No	Yes	Yes	15-30 days	No	Maybe	

292364 rows × 17 columns

Figure 2 - Dataset

Growing_Stress	Changes_Habits	Mental_Health_History	Mood_Swings	Coping_Struggles	Work_Interest	Social_Weakness	mental_health_interview	care_options
Yes	No	Yes	Medium	No	No	Yes	No	Not sure
Yes	No	Yes	Medium	No	No	Yes	No	No
Yes	No	Yes	Medium	No	No	Yes	No	Yes
Yes	No	Yes	Medium	No	No	Yes	Maybe	Yes
Yes	No	Yes	Medium	No	No	Yes	No	Yes
...	...	...	...	...	...	...	...	...
No	Maybe	No	Low	Yes	No	Maybe	Maybe	Not sure
No	Maybe	No	Low	Yes	No	Maybe	No	Yes
No	Maybe	No	Low	Yes	No	Maybe	No	No
No	Maybe	No	Low	Yes	No	Maybe	No	Yes
No	Maybe	No	Low	Yes	No	Maybe	No	Yes

Figure 3 – Dataset(Continuation)

Attributes:

- **Timestamp**: The date and time when the survey response was recorded.
- **Gender**: The gender identity of the respondent (e.g., Male, Female, etc.).
- **Country**: The country where the respondent resides or works.
- **Occupation**: The respondent's job category or professional role (e.g., Corporate, Business).
- **self_employed**: Indicates whether the respondent is self-employed (Yes, No, or missing data).
- **family_history**: Indicates if the respondent has a family history of mental illness (Yes or No).
- **treatment**: Indicates if the respondent has sought treatment for a mental health issue (Yes or No).
- **Days_Indoors**: Number of consecutive days the respondent stayed indoors (e.g., 1-14 days, 15-30 days).
- **Growing_Stress**: Indicates whether the respondent feels increasing stress over time (Yes or No).
- **Changes_Habits**: Indicates if there have been changes in the respondent's habits (e.g., eating, sleeping) due to mental health issues (Yes, No, or Maybe).
- **Mental_Health_History**: Indicates if the respondent has a prior history of mental health issues (Yes or No).
- **Mood_Swings**: The intensity or frequency of mood swings (Low, Medium, or High).

- **Coping_Struggles**: Indicates if the respondent is struggling to cope with stress or emotional challenges (Yes or No).
- **Work_Interest**: Indicates if the respondent has lost interest in work or feels disengaged (Yes, No, or Maybe).
- **Social_Weakness**: Indicates if the respondent feels socially withdrawn or isolated (Yes, No, or Maybe).
- **mental_health_interview**: Indicates if the respondent is willing or has attended a mental health interview at work (Yes, No, or Not sure).
- **care_options**: Indicates awareness or availability of mental health care options at the workplace (Yes, No, or Not sure).

4.2 Data Preprocessing Techniques

Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

- **Data Cleaning:** This step involves removing irrelevant or duplicate data, handling missing values, and correcting inconsistent data. For instance, if some data is missing or invalid, it needs to be either removed or imputed using methods such as mean or median imputation.

  In this project, the dataset was examined for missing or null values. It was found that the 'self_employed' column contained 5202 missing values, while all other columns were complete. To handle this, the missing values in 'self_employed' were replaced with the category 'Unknown', preserving the original structure of the data without introducing assumptions. This approach allows models to treat unknown cases distinctly during analysis. Additionally, the dataset was checked for duplicate entries, and 2313 duplicate rows were identified and removed to maintain the quality and accuracy of the dataset.
- **Data Transformation:** This step involves converting the data into a suitable format that can be used by the machine learning algorithm. Data transformation can include scaling, normalization, and encoding categorical variables. Scaling and normalization are techniques used to ensure that all variables are in the same range while encoding categorical variables involves converting categorical data into numerical data that can be used by algorithm.

The categorical features such as ‘self_employed’, ‘family_history’, treatment, Growing_Stress, Changes_Habits, ‘Mental_Health_History', ‘Mood_Swings’, Coping_Struggles, Work_Interest’, Social_Weakness, ‘mental_health_interview’, and ‘care_options’ were encoded using label encoding. Additionally, ‘Gender’, ‘Country’, ‘Occupation’, and ‘Days_Indoors’ were also converted to numerical values to ensure the dataset was suitable for machine learning analysis.

- **Feature Selection:** To improve model performance and reduce complexity, feature selection was performed using the feature importance scores from a Random Forest Classifier. Each feature's importance was evaluated based on how much it contributed to the prediction of the target variable. Features such as ‘Country’, ‘family_history’, ‘care_options’, ‘mental_health_interview’, ‘self_employed’, ‘Gender’, and ‘Occupation’ were identified as the most significant. On the other hand, features with very low importance—such as Days_Indoors, Coping_Struggles, and Growing_Stress—were dropped, as they contributed minimally to the model. This selection helped retain only the most relevant information for training, leading to a more efficient and focused model.
- **Data Spitting:** This step involves dividing the data into two or more sets, one for training the model and another for testing or validating the model. Data splitting is done to avoid overfitting and to ensure that the model can generalize well to new data. The dataset was split such that 20% of the data was reserved for the test set, while the remaining 80% was used for training the model. This is commonly represented with a test size ratio of 0.2.

Overall, data preprocessing is a crucial step in machine learning, as it helps to ensure that the model is trained on clean and relevant data. It involves several steps, including data cleaning, and integration, transformation, feature selection, and data splitting. By following steps, the quality of the data can be improved, and the accuracy and performance of the model can be enhanced.

4.3 Data Analysis

Data Analysis through different Visualization Techniques

Visualization refers to the graphical representation of information and data. By using visual elements like charts, graphs, maps, and other visual tools, visualization helps to understand trends, outliers, and patterns in data. The main goal of visualization is to make complex data more accessible, understandable, and usable.

4.3.1 Pie Chart

A pie chart is a type of data visualization that represents categorical data as slices of a circular pie. Each slice corresponds to a category's proportion or percentage of the whole, with the size of the slice indicating its relative magnitude. Pie charts are typically used to show the composition or distribution of a dataset, making it easy to compare parts to the whole. They are most effective when displaying a limited number of categories with clear differences in values.

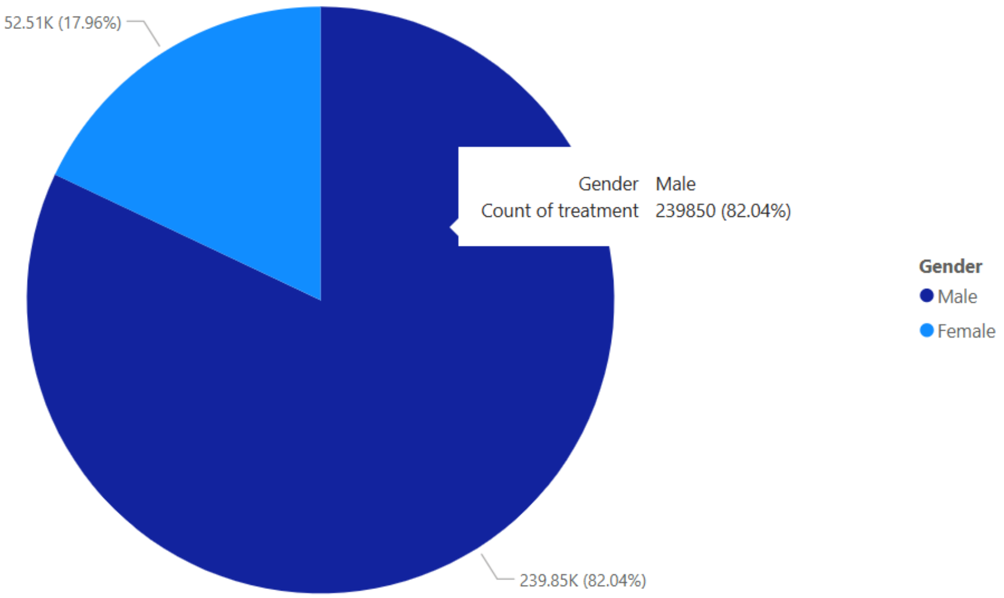

Figure 4 – Pie Chart

This pie chart visualizes the distribution of individuals receiving treatment based on gender. The chart shows that 82.04 percent, or 239,850 individuals, are male, while 17.96 percent, or 52,510 individuals, are female. The large difference between the two segments highlights a notable gender imbalance in treatment count, suggesting that males are substantially more represented in this dataset. This distribution may reflect underlying gender-related differences in the prevalence of the condition, access to care, or reporting behavior.

4.3.2 Clustered Column Chart

A clustered column chart is a type of data visualization used to compare values across multiple categories and subcategories. It displays data as vertical bars grouped side-by-side for each main category, allowing easy comparison between different series. This format is particularly effective for showing differences between groups over time or across conditions. Each cluster represents a primary category, and each column within the cluster represents a subcategory, making it easier to identify trends, variations, and patterns within the data.

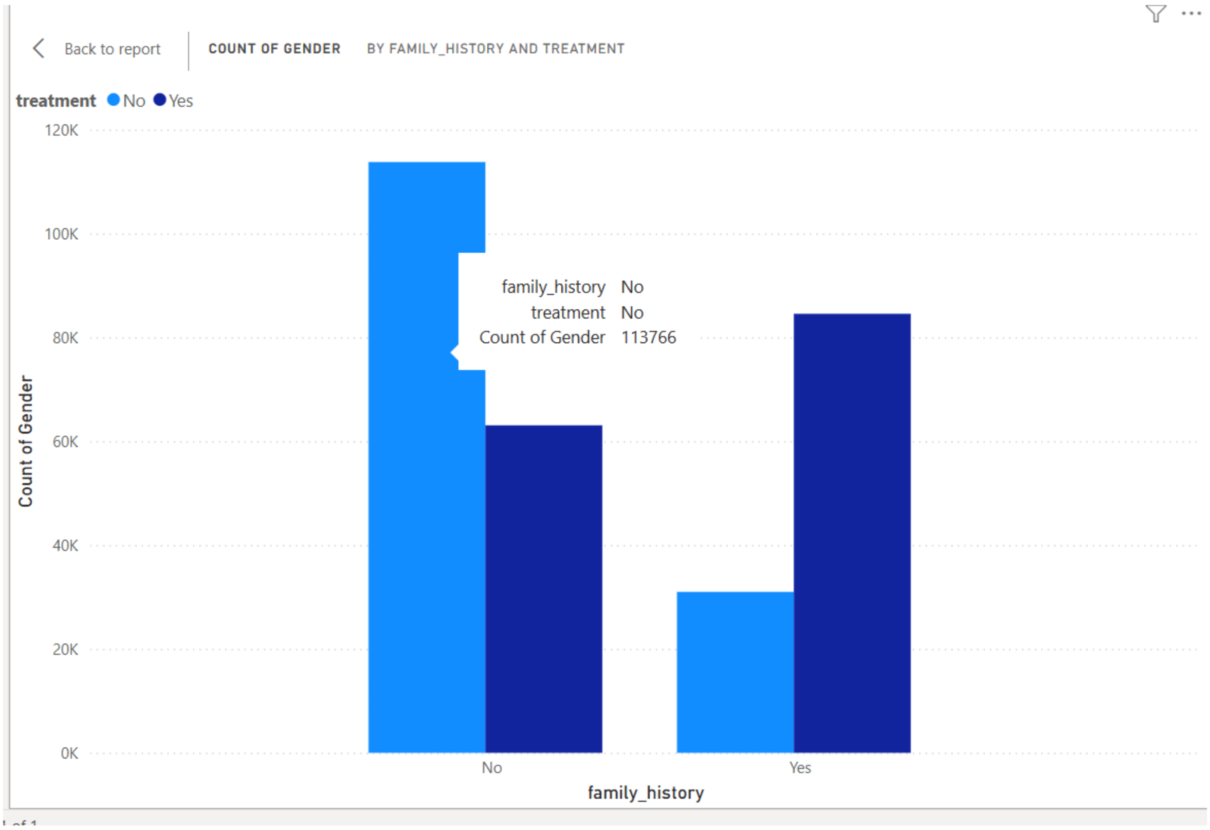

Figure 5 – Clustered Column Chart

This clustered column chart displays the count of individuals by gender, grouped by their family history of mental illness and whether they have received treatment. The chart is divided into two main groups: those with no family history and those with a family history. Within each group, the columns are split by treatment status—blue for individuals not receiving treatment and dark blue for those who have. Among individuals with no family history, the number not receiving treatment is significantly higher than those who did. In contrast, for individuals with a family history, more received treatment than not. This pattern suggests a potential correlation between family history and likelihood of seeking treatment.

4.3.3 Stacked Column Chart

A stacked column chart is a type of data visualization used to display the total and sub-category breakdowns of values across multiple categories. Each column represents a main category, and segments within the column represent different sub-categories stacked on top of each other. This allows for comparison of both the overall totals and the individual components within each category. Stacked column charts are useful for visualizing the contribution of various groups to a whole, making it easier to observe patterns, trends, and proportions across categories.

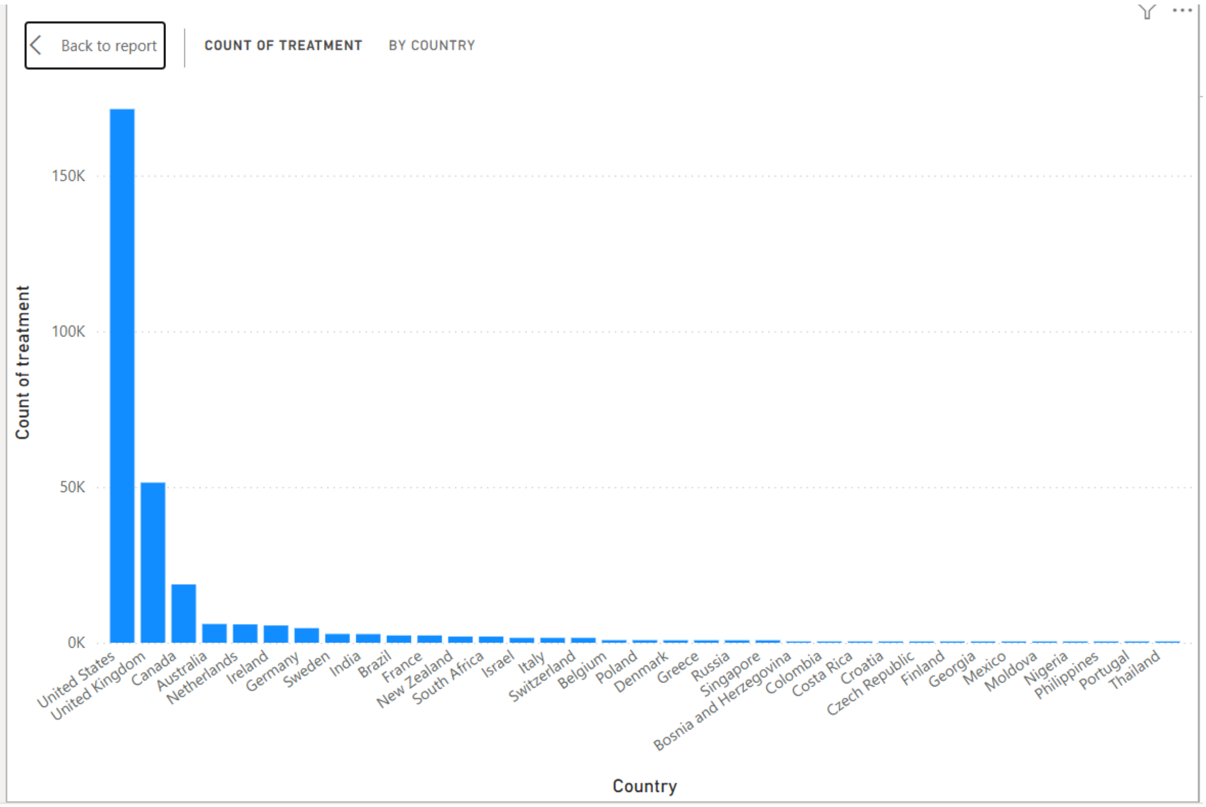

Figure 6 – Stacked Column Chart

This stacked column chart displays the count of individuals receiving treatment, grouped by country. Each column represents a country, and the height of the column shows the total count of treatment cases. The stacking within each column, though appearing as a single color here, typically represents sub-categories such as gender, treatment type, or another dimension if applied. The chart clearly highlights that the United States has the highest number of treatment cases, followed by the United Kingdom and Canada. A steep decline is observed in other countries, indicating a major concentration of the dataset in a few regions. This visualization helps in understanding the country-wise distribution and emphasizes regional disparities in treatment data.

4.4 Mental Health Treatment Distribution in India Based on Family History and Gender

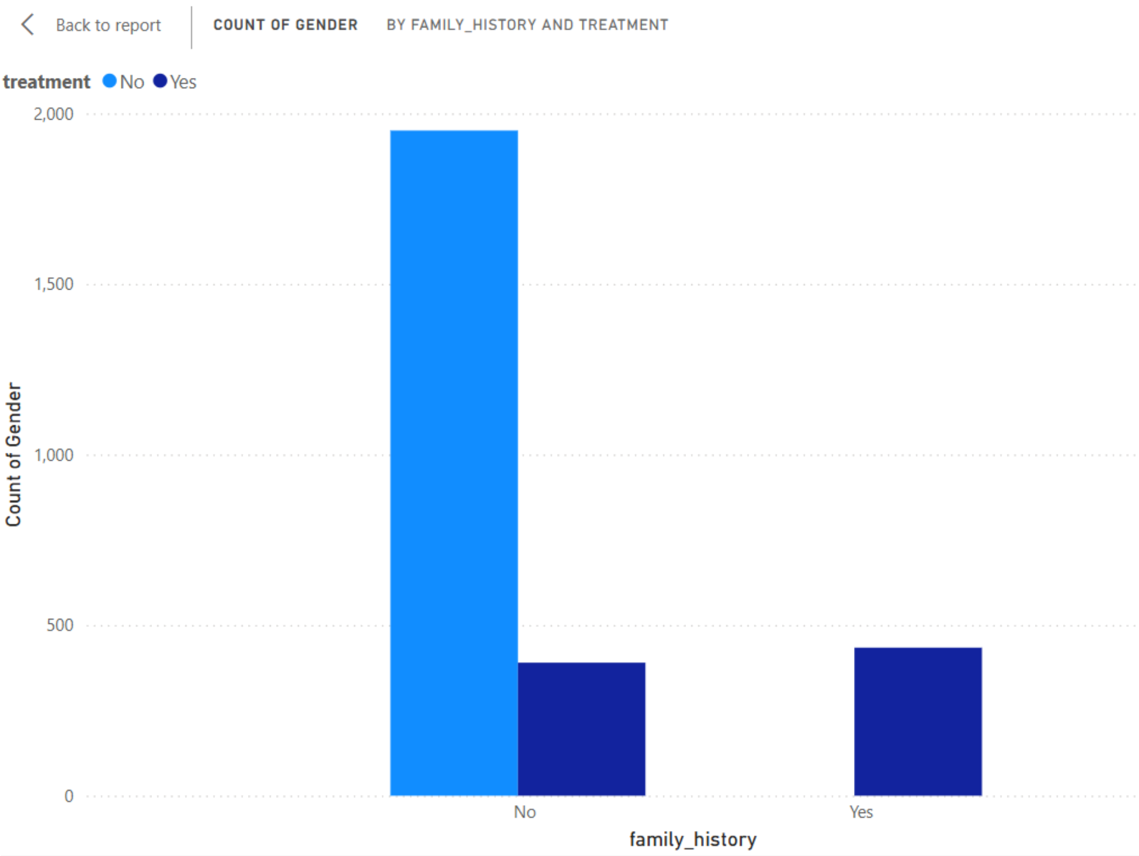

Figure 7 - Clustered Column Chart of Treatment by Family History in India

This visualization represents mental health treatment distribution in India, segmented by family history of mental illness and gender. It highlights that a higher number of individuals without a family history have not received treatment, while those with a family history show a greater proportion undergoing treatment. This suggests that awareness or inherited risk may influence treatment-seeking behavior in the Indian population.

CHAPTER 5
MODEL IMPLEMENTATION AND EVALUATION

5.1 Model Implementation

The Mental Health Detection project is a classification-based approach aimed at identifying potential mental health conditions using both classical machine learning models and deep learning techniques. Models such as K-Nearest Neighbors (KNN), Random Forest, AdaBoost, XG Boost, and Artificial Neural Networks (ANN) were implemented to analyse and classify the data. These models were evaluated using a comprehensive set of performance metrics including accuracy, precision, recall, F1-score, and ROC-AUC score to assess the quality of predictions. Additionally, the confusion matrix was utilized to visualize the performance of each classifier in terms of true positives, false positives, true negatives, and false negatives, helping to understand the strengths and limitations of each model more clearly. The integration of neural networks enabled the system to capture complex, non-linear patterns in the data, offering improved predictive performance in certain cases. This project highlights the practical application of classification techniques in the field of mental health, a critical area in countries like India where early detection and support systems are still developing.

5.1.1 K-Nearest Neighbor

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning techniques. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confus
```

```
model = KNeighborsClassifier(n_neighbors=5)
model.fit(x_train, y_train)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

```
y_pred = model.predict(x_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7261984163631078
```

```
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_lr))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Confusion Matrix:
 [[19863  9032]
 [ 8321 21257]]

Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.72      0.72     28895
           1       0.73      0.73      0.73     29578

    accuracy                           0.73     58473
   macro avg       0.73      0.73      0.73     58473
weighted avg       0.73      0.73      0.73     58473
```

5.1.2 XG Boost

XG Boost (Extreme Gradient Boosting) is an optimized machine learning algorithm that is widely used for both classification and regression tasks. It is an implementation of the Gradient Boosting framework, specifically designed to improve its performance and efficiency by making use of several advanced techniques.

```
import xgboost as xgb
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
xgb_model = xgb.XGBClassifier(use_label_encoder=False, eval_metric='logloss')
```

```
xgb_model.fit(x_train, y_train)
```

```
y_pred_xgb = xgb_model.predict(x_test)
```

```
print("XGBoost Accuracy: ", accuracy_score(y_test, y_pred_xgb))
```

```
XGBoost Accuracy:  0.7842251979546115
```

```
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_xgb))
print("Classification Report:\n", classification_report(y_test, y_pred_xgb))
```

```
Confusion Matrix:
 [[20900  7995]
 [ 4622 24956]]
Classification Report:
               precision    recall  f1-score   support

           0       0.82      0.72      0.77     28895
           1       0.76      0.84      0.80     29578

    accuracy                           0.78     58473
   macro avg       0.79      0.78      0.78     58473
weighted avg       0.79      0.78      0.78     58473
```

5.1.3 Ada Boost Classifier

AdaBoost (Adaptive Boosting) is an ensemble learning algorithm used for classification tasks. It combines multiple weak learners, typically decision trees, to form a strong classifier. Each new model focuses on correcting the errors made by the previous ones. AdaBoost improves accuracy by giving more weight to misclassified instances and adjusting the model accordingly. It is effective for binary classification problems and helps reduce bias and variance, making it suitable for structured datasets.

```
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

# Split your data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# Initialize and train AdaBoost model
model = AdaBoostClassifier(n_estimators=100, random_state=42)
model.fit(x_train, y_train)

# Predict on test set
y_pred = model.predict(x_test)

# Evaluate performance
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f}")

print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.7016

Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.70      0.70     28895
           1       0.71      0.70      0.70     29578

    accuracy                           0.70     58473
   macro avg       0.70      0.70      0.70     58473
weighted avg       0.70      0.70      0.70     58473
```

5.1.4 Random Forest Classifier

Random Forest Classifier is an ensemble learning method used for classification tasks. It builds multiple decision trees during training and combines their outputs to make a final prediction. Each tree is trained on a random subset of the data and features, which helps reduce overfitting and improves model generalization. Random Forest is known for its high accuracy, robustness, and ability to handle both numerical and categorical data effectively. It is widely used for binary and multi-class classification problems.

```
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier()
```

```
rf_model.fit(x_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier()
```

```
y_pred_rf = rf_model.predict(x_test)
```

```
print("Random Forest Accuracy: ", accuracy_score(y_test, y_pred_rf))
```

```
Random Forest Accuracy:  0.7585552306192602
```

```
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_rf))
print("Classification Report:\n", classification_report(y_test, y_pred_rf))
```

```
Confusion Matrix:
 [[20659  8236]
 [ 5882 23696]]
Classification Report:
               precision    recall  f1-score   support

           0       0.78      0.71      0.75     28895
           1       0.74      0.80      0.77     29578

    accuracy                           0.76     58473
   macro avg       0.76      0.76      0.76     58473
weighted avg       0.76      0.76      0.76     58473
```

5.1.5 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) is a type of deep learning model that mimics the working of the human brain to recognize patterns and relationships in data. It consists of layers of interconnected nodes, including input, hidden, and output layers. Each node processes inputs using weights, biases, and activation functions to produce an output. ANN is capable of learning from data through backpropagation and is widely used for classification, regression, and prediction tasks in various domains such as healthcare, finance, and natural language processing.

```
# Evaluate on test data
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Accuracy: {accuracy*100:.2f}%')

# Predict and show report
y_pred = (model.predict(X_test) > 0.5).astype("int32")
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
1795/1795 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.8031 - loss: 0.3981
Test Accuracy: 80.31%
1795/1795 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step
[[23682  4813]
 [ 6496 22442]]
              precision    recall  f1-score   support

           0       0.78      0.83      0.81     28495
           1       0.82      0.78      0.80     28938

    accuracy                           0.80     57433
   macro avg       0.80      0.80      0.80     57433
weighted avg       0.80      0.80      0.80     57433
```

5.2 Evaluation

In this mental health detection project, the performance of various machine learning and deep learning models was evaluated using standard classification metrics. The primary focus was on how accurately the models could predict whether an individual is likely to seek or require mental health treatment based on survey data. Accuracy was considered to understand the overall correctness of the models, while precision and recall were important to assess how well the models handled positive predictions—particularly in identifying those genuinely in need of support. The F1-score provided a balanced view by combining both precision and recall, which is especially useful when dealing with uneven class distributions. These metrics helped compare the effectiveness of different models and revealed that while models like XG Boost showed high overall performance, the deep learning model (ANN) delivered better generalization and balance between precision and recall, making it more suitable for real-world mental health prediction tasks.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Model Performance Table

In this study, both traditional machine learning models and a deep learning model were implemented to predict mental health conditions based on survey data. The models evaluated include K-Nearest Neighbors (KNN), AdaBoost, Random Forest Classifier, XG Boost, and Artificial Neural Network (ANN).

The following table presents the accuracy scores obtained by each model:

Model Performance Table:

Model	Accuracy
ANN	80.31%
XG Boost	78.42%
Random Forest Classifier	75.85%
KNN	72.61%
Ada Boost	70.16%

Table 1 – Model Performance Table

From the results, it is evident that the deep learning model (ANN) achieved the highest accuracy at 80.31%, showing its capability to capture complex patterns in the data. Among the traditional machine learning models, XGBoost outperformed the others, achieving 78.42%, which highlights the strength of gradient boosting in handling structured data.

The findings indicate that while deep learning approaches like ANN provide better results for this dataset, advanced machine learning models such as XG Boost also deliver high performance and are suitable for mental health prediction tasks.

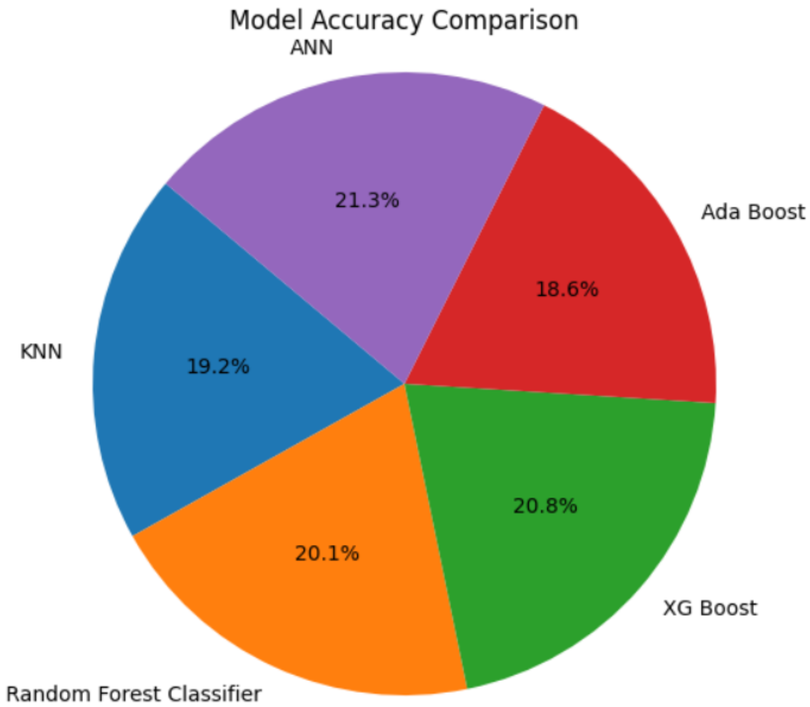

Figure 8 – Model Accuracy Comparison(Pie Chart)

6.2 Evaluation Matrix

CLASSIFIERS	PRECISION	RECALL	F1-SCORE
K-NEAREST NIGHBORS	0.72	0.72	0.72
RANDOM FOREST CLASSIFIER	0.78	0.71	0.75
XG BOOST	0.82	0.72	0.77
ADA BOOST	0.72	0.72	0.72
ARTIFICIAL NEURAL NETWORK	0.80	0.84	0.82

Table 2 – Evaluation matrix table

6.3 Model Evaluation Using ROC-AUC Scores

The ROC (Receiver Operating Characteristic) curve is a graphical representation used to evaluate the performance of classification models. It plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) at various threshold levels. A model with a curve closer to the top-left corner indicates better performance.

The AUC (Area Under the Curve) summarizes the ROC curve into a single value, where an AUC of 1.0 represents a perfect model and 0.5 indicates no discriminative ability. In this project, the ROC curve is used to compare the effectiveness of different machine learning algorithms including XGBoost, Random Forest, KNN, and AdaBoost in predicting mental health treatment needs. XGBoost achieved the highest AUC score, indicating superior classification capability.

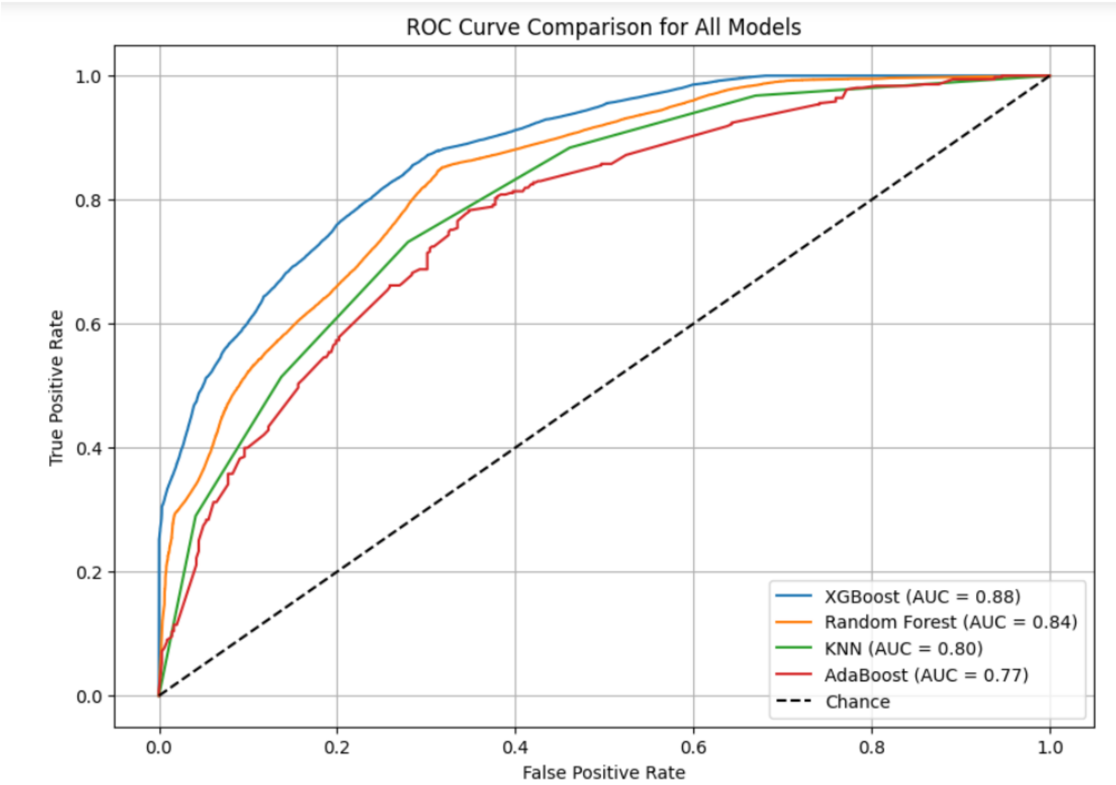

Figure 9 – ROC Curve Comparison for all models

The ROC curve of the ANN model demonstrates strong classification performance, with an AUC (Area Under the Curve) value of 0.91. This indicates that the model has a high ability to distinguish between the positive and negative classes. The curve rising steeply towards the top-left corner shows good sensitivity and low false positive rate.

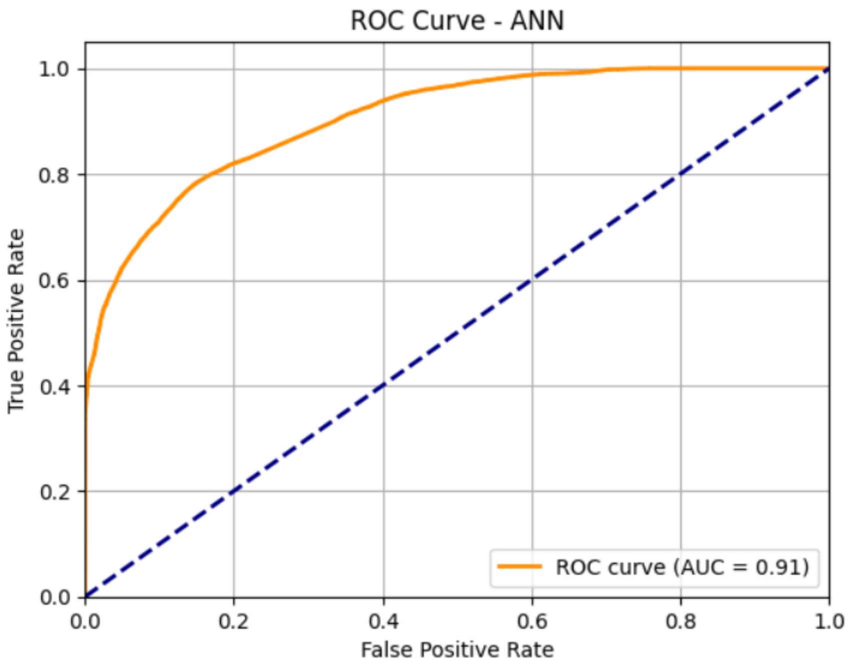

Figure 10 – ROC curve for ANN

The ROC-AUC analysis reveals that the Artificial Neural Network (ANN) model achieved the highest AUC score among all the models, indicating superior performance in distinguishing between individuals who require mental health treatment and those who do not. The ANN's AUC of 0.91 reflects its strong ability to balance sensitivity and specificity, making it the most effective model for this classification task. This demonstrates the advantage of deep learning techniques in capturing complex patterns in mental health data.

6.4 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true value are known. It is particularly useful for understanding the types of errors a model is making.

6.4.1 Confusion Matrix for ANN

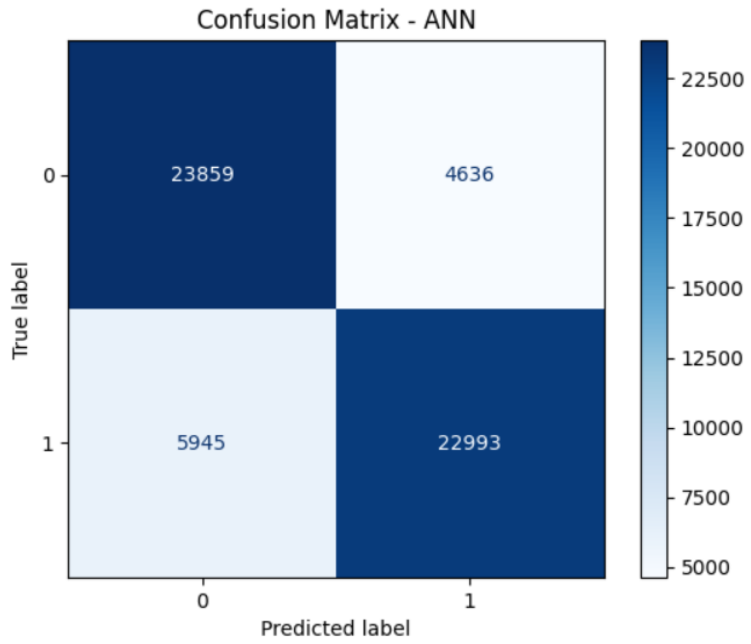

Figure 11 – Confusion matrix for ANN

The confusion matrix of the ANN model indicates that it correctly classified 23,859 instances of class 0 and 22,993 instances of class 1. However, it misclassified 4,636 class 0 instances as class 1 and 5,945 class 1 instances as class 0. Overall, the ANN model shows balanced performance with a strong ability to distinguish between both classes.

6.4.2 Confusion Matrix for XG Boost

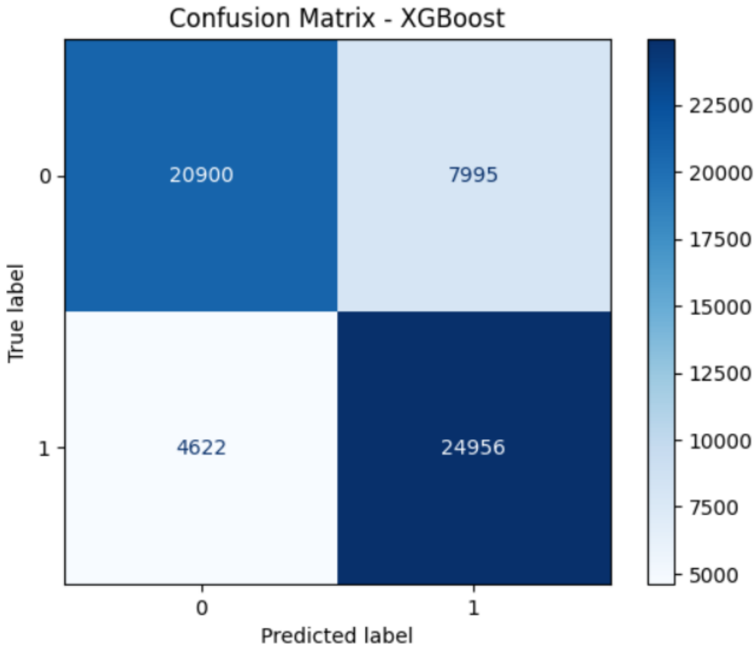

Figure 12 – Confusion matrix for XG Boost

The confusion matrix for the XG Boost model shows strong classification performance, correctly predicting 20,900 instances of class 0 and 24,956 instances of class 1. However, the model misclassified 7,995 class 0 instances as class 1 and 4,622 class 1 instances as class 0. Overall, the model demonstrates a good balance between sensitivity and specificity, indicating reliable predictive capability for mental health classification.

CHAPTER 7

CONCLUSION

This project focused on predicting mental health treatment needs using survey data by applying a range of machine learning and deep learning techniques. After thorough preprocessing and exploratory analysis, models such as K-Nearest Neighbors (KNN), Random Forest, AdaBoost, XGBoost, and Artificial Neural Networks (ANN) were implemented and evaluated. Among these, the ANN model delivered the best performance, achieving an accuracy of 80.31%, highlighting its capability to learn complex, non-linear patterns present in mental health data. XGBoost also performed competitively with an accuracy of 78.42%, confirming the strength of ensemble-based decision tree models in classification tasks. Evaluation metrics such as confusion matrices, precision, recall, F1-score, and ROC-AUC were used to measure and compare model effectiveness. Notably, the ANN model achieved the highest AUC score of 0.91, indicating its superior discriminative power between individuals who may or may not require mental health treatment. Overall, the results underscore the value of predictive modeling in mental health analytics, offering a data-driven approach to support awareness, early intervention, and informed decision-making by healthcare professionals and organizations.

CHAPTER 8

FUTURE SCOPE

In the future, this mental health prediction system can be enhanced by incorporating real-time data from mobile apps, social media, or wearable devices to enable early and continuous monitoring. Specifically in a country like India, where mental health awareness and access to care are still developing, such predictive tools can play a vital role in bridging gaps in early diagnosis and support. Expanding the model to classify specific disorders and integrating regional languages, cultural factors, and location-based insights can make the system more relevant and accessible to the Indian population. Additionally, deploying this solution as a web or mobile application can empower individuals, especially in rural or underserved areas, to assess their mental well-being and seek timely help.

BIBLIOGRAPHY

- Sharma, S.K., Alutaibi, A.I., Khan, A.R. *et al.* Early detection of mental health disorders using machine learning models using behavioral and voice data analysis. *Sci Rep* 15, 16518 (2025). https://doi.org/10.1038/s41598-025-00386-8
- Abd Rahman, Rohizah, et al. "Application of machine learning methods in mental health detection: a systematic review." *Ieee Access* 8 (2020): 183952-183964.
- Khoo, Lin Sze, et al. "Machine learning for multimodal mental health detection: a systematic review of passive sensing approaches." *Sensors* 24.2 (2024): 348.
- Costello, E. J. (2016). Early Detection and Prevention of Mental Health Problems: Developmental Epidemiology and Systems of Support. *Journal of Clinical Child & Adolescent Psychology*, *45*(6), 710–717. https://doi.org/10.1080/15374416.2016.1236728
- Abd Rahman, Rohizah, et al. "A survey on mental health detection in online social network." *International Journal on Advanced Science, Engineering and Information Technology* 8.4-2 (2018): 1431-1435.
- Hassantabar, Shayan, et al. "Mhdeep: Mental health disorder detection system based on wearable sensors and artificial neural networks." *ACM Transactions on Embedded Computing Systems* 21.6 (2022): 1-22.
- Tiwari, Amit Kumar, Shivani Pandey, and Raghvendra Kumar Pandey. "Mental health detection using machine learning." *Artificial Intelligence and Information Technologies* (2024): 315-320.
- Kim, Jina, Daeun Lee, and Eunil Park. "Machine learning for mental health in social media: bibliometric study." *Journal of Medical Internet Research* 23.3 (2021): e24870.
- Liu, Danxia, et al. "Detecting and measuring depression on social media using a machine learning approach: systematic review." *JMIR Mental Health* 9.3 (2022): e27244.
- Garg, Muskan. "Mental health analysis in social media posts: a survey." *Archives of Computational Methods in Engineering* 30.3 (2023): 1819-1842.