MENTAL HEALTH PREDICTION USING ML

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

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

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(DS)

Submitted By

MANIDEEPIKA ENGLE	**21UK1A67B5**
SYED SHARIQ SHARIEF	**21UK1A6787**
NAYUDU BHAVANA	**21UK1A6776**
NERELLI NAGARJUNA	**21UK1A6777**

Under the guidance of

Ms.K.Soujanya

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING(DS)

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION
INDUSTRY ORIENTED MINI PROJECT

This is to certify that the UG Project Phase-1 entitled “Mental Health Prediction Using ML” is being submitted by MANIDEEPIKA ENGLE(21UK1A67B5),SYED SHARIQ SHARIEF(21UK1A6787),NAYUDU BHAVANA(21UK1A6776),NERELLI NAGARJUNA(21UK1A6777) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024- 2025.

Project Guide
Ms.K.Soujanya

HOD
Dr. K. Sharmila
(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.SYED MUSTHAK AHAMED,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr.K.SHARMILA**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, **K.SOUJANYA ,** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

MANIDEEPIKA ENGLE **(21UK1A67B5)**
SYED SHARIQ SHARIEF **(21UK1A6787)**
NAYUDU BHAVANA **(21UK1A6776)**
NERELLI NAGARJUNA **(21UK1A6777)**

ABSTRACT

This project focuses on utilizing machine learning (ML) techniques to predict mental health conditions, leveraging a variety of datasets that encompass social, psychological, and behavioral factors. The objective is to identify patterns and risk factors associated with mental health issues such as depression, anxiety, and stress. By employing advanced ML algorithms, the model aims to analyze large-scale data to provide accurate predictions and early detection of mental health conditions. This approach enables the development of personalized intervention strategies tailored to individual needs, thereby enhancing mental health support systems. The project also emphasizes the importance of data privacy and ethical considerations in handling sensitive health information. Ultimately, the goal is to improve mental health outcomes by facilitating timely and effective interventions, reducing the burden on healthcare systems, and promoting overall well-being through the integration of technology and mental health care.

TABLE OF CONTENTS:-

1.INTRODUCTION

1.1.OVERVIEW

The increase of mental health problems and the need for effective medical health care have led to an investigation of machine learning that can be applied in mental health problems. This report presents a recent systematic review of machine learning and deep learning approaches in predicting mental health problems. Furthermore, we will discuss the challenges, limitations, and future directions for the application of machine learning in the mental health field. We collect research articles and studies that are related in predicting mental health problems by searching reliable databases. Then, we categorize the collected research articles based on the mental health problems such as schizophrenia, bipolar disorder, anxiety and depression, posttraumatic stress disorder, and mental health problems among children. Discussing the findings, we reflect on the challenges and limitations faced by the researchers on machine learning in mental health problems. Additionally, we provide concrete recommendations on the potential future research and development of applying machine learning and deep learning in the mental health field. The main purpose of the Mental Health Prediction system is to predict whether a person needs to seek mental health treatment or not based on inputs provided by them.

1.2.PURPOSE

Mental Health First Aid teaches participants how to notice and support an individual who may be experiencing a mental health or substance use concern or crisis and connect them with the appropriate employee resources. Employers can offer robust benefits packages to support employees who go through mental health issues. That includes Employee Assistance Programs, Wellness programs that focus on mental and physical health, Health and Disability Insurance, or flexible working schedules or time off policies. Organizations that incorporate mental health awareness

help to create a healthy and productive work environment that reduces the stigma associated with mental illness, increases the organizations' mental health literacy, and teaches the skills to safely and responsibly respond to a co- worker's mental health concern. The main purpose of the Mental Health Prediction system is to predict whether a person needs to seek mental health treatment or not based on inputs provided by them. We will be using classification algorithms such as Logistic Regression, KNN, Decision tree, Random Forest, AdaBoost, Gradient Boost, and XGBoost. We will train and test the data with these algorithms. From this, the best model is selected and saved in pkl format. We will also be deploying our model locally using Flask.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

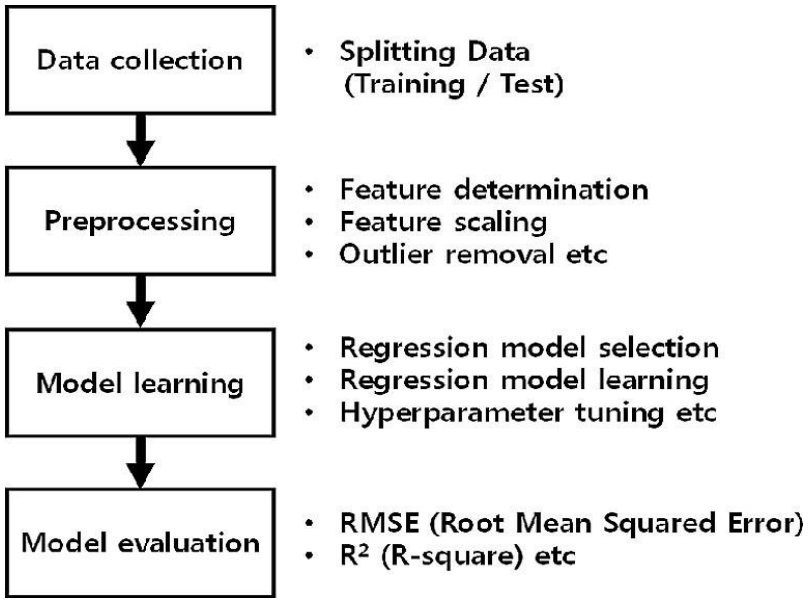
Mental illness is a health problem that undoubtedly impacts emotions, reasoning, and social interaction of a person. These issues have shown that mental illness gives serious consequences across societies and demands new strategies for prevention and intervention. To accomplish these strategies, early detection of mental health is an essential procedure. Mental illness is usually diagnosed based on the individual self report that requires questionnaires designed for the detection of the specific patterns of feeling or social interactions. With proper care and treatment, many individuals will hopefully be able to recover from mental illness or emotional disorder.

2.2 PROPOSED SOLLUTION

Machine learning is a technique that aims to construct systems that can improve through experience by using advanced statistical and probabilistic techniques. It is believed to be a significantly useful tool to help in predicting mental health. It is allowing many researchers to acquire important information from the data, provide personalized experiences, and develop automated intelligent systems. The widely used algorithms in the field of machine learning such as support vector machine, random forest.

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. Hardware / Software Designing Recommended

System Requirements

- **Processors**: Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAM Intel® Xeon® processor E5-2698 v3 at 2.30 GHz (2 sockets, 16 cores each, 1 thread per core), 64 GB of DRAM Intel® Xeon Phi™ processor 7210 at 1.30 GHz (1 socket, 64 cores, 4 threads per core), 32 GB of DRAM, 16 GB of MCDRAM (flat mode enabled)
- **Disk space**: 2 to 3 GB
- **Operating systems**: Windows® 10, macOS*, and Linux*

Minimum System Requirements

- **Processors**: Intel Atom® processor or Intel® Core™ i3 processor
- **Disk space**: 1 GB
- **Operating systems**: Windows* 7 or later, macOS, and Linux
- **Python* versions**: 3.9

Software requirements:

Anaconda navigator:

Anaconda is an open-source distribution for python and R. It is used for data science, machine learning, deep learning, etc. With the availability of more than 300 libraries for data science, it becomes fairly optimal for any programmer to work on anaconda for data science

Pycharm:

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

4.EXPERIMENTAL INVESTIGATION

Logistic Regression:

This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure.

DecisionTreeClassifier:

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy.

KNeighborsClassifier:

K-Nearest Neighbors, or KNN for short, is one of the simplest machine learning algorithms and is used in a wide array of institutions. KNN is a non-parametric, lazy learning algorithm. When we say a technique is non-parametric, it means that it does not make any assumptions about the underlying data. In other words, it makes its selection based off of the proximity to other data points regardless of what feature the numerical values represent. Being a lazy learning algorithm implies that there is little. Therefore, we can immediately classify new data points as they present themselves. To no training phase.

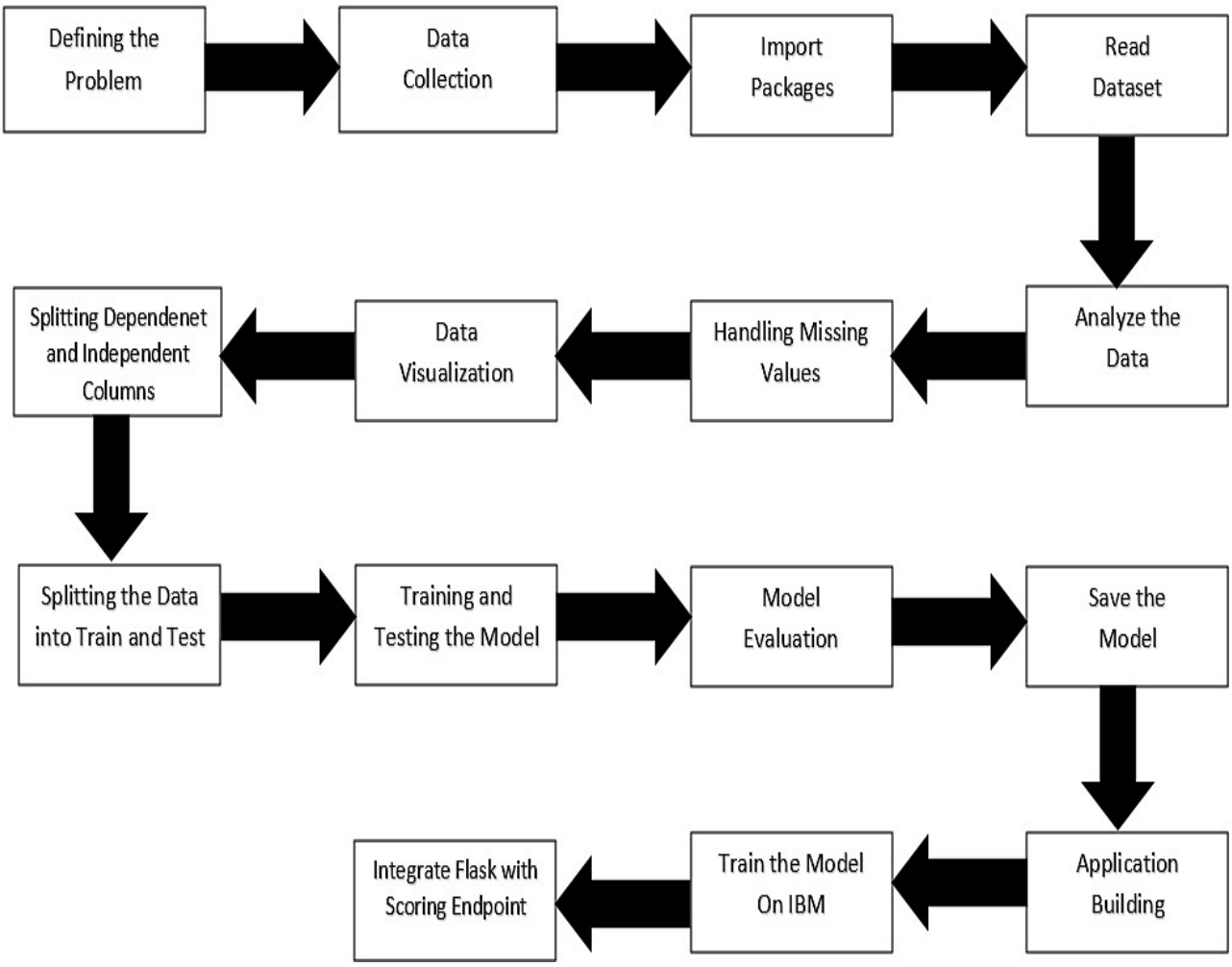
XGBClassifier:

The XGBoost stands for extreme Gradient Boosting, which is a boosting algorithm based on gradient boosted decision trees algorithm. XGBoost applies a better regularization technique to reduce overfitting, and it is one of the differences from the gradient boosting. The 'xgboost' is an open-source library that provides machine learning algorithms under the gradient boosting methods. The xgboost.XGBClassifier is a scikit-learn API compatible class for classification.

RandomForestClassifier:

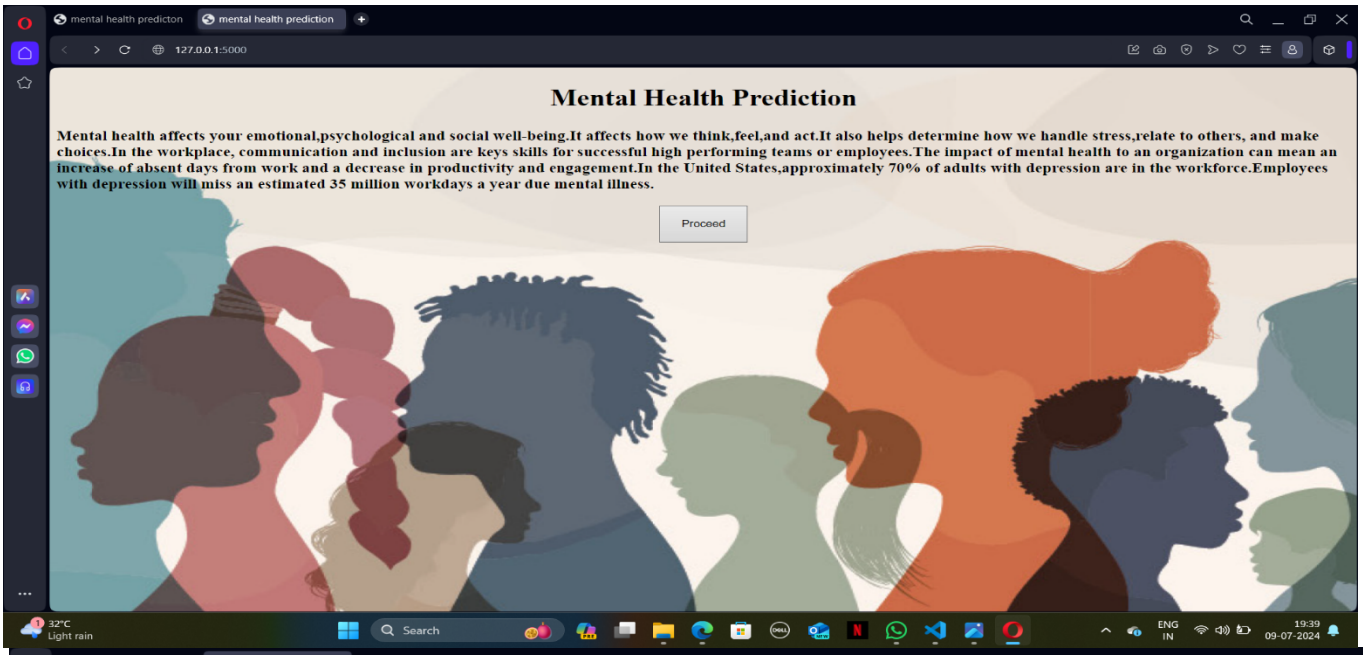
The Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction.

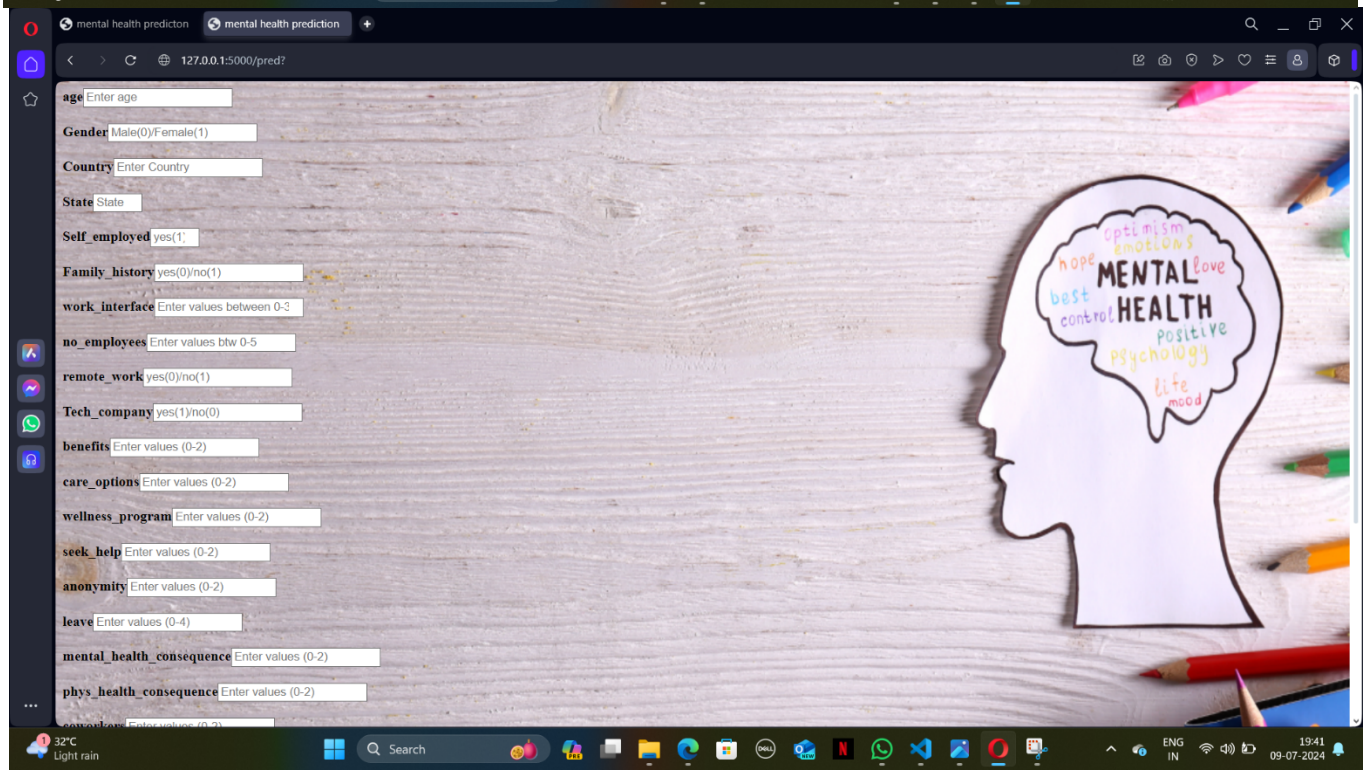
5.FLOWCHART

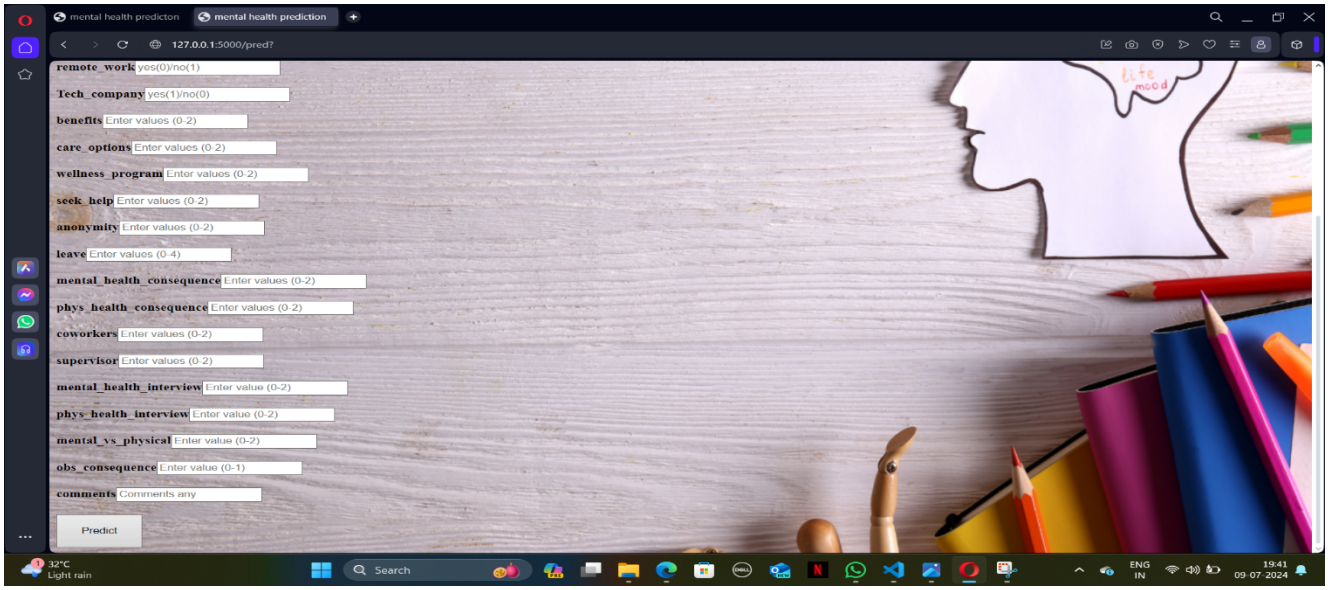

Defining the Problem
Data Collection
Import Packages
Read Dataset
Analyze the Data
Handling Missing Values
Data Visualization
Splitting Dependenet and Independent Columns
Splitting the Data into Train and Test
Training and Testing the Model
Model Evaluation
Save the Model
Application Building
Train the Model On IBM
Integrate Flask with Scoring Endpoint

6.RESULT

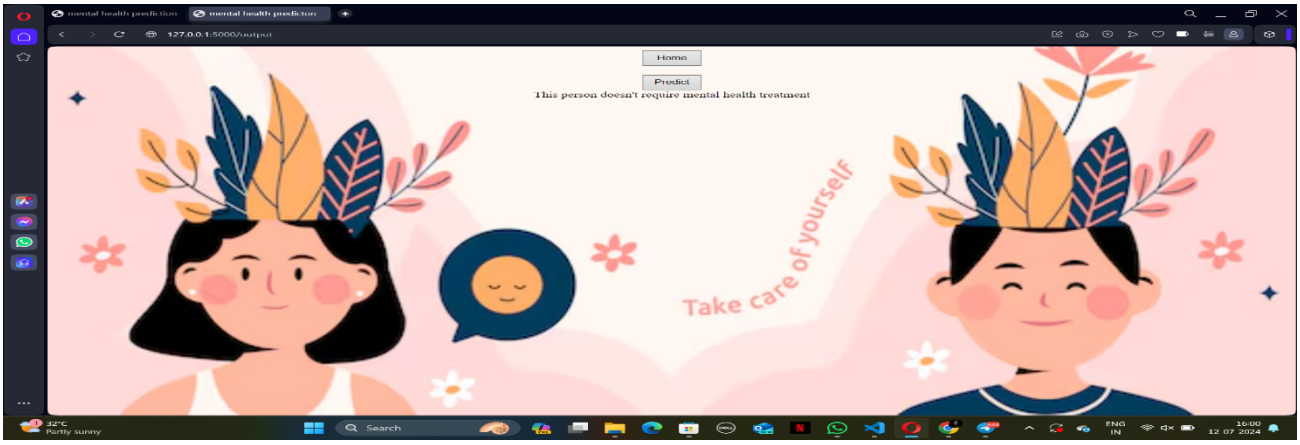
HOME PAGE

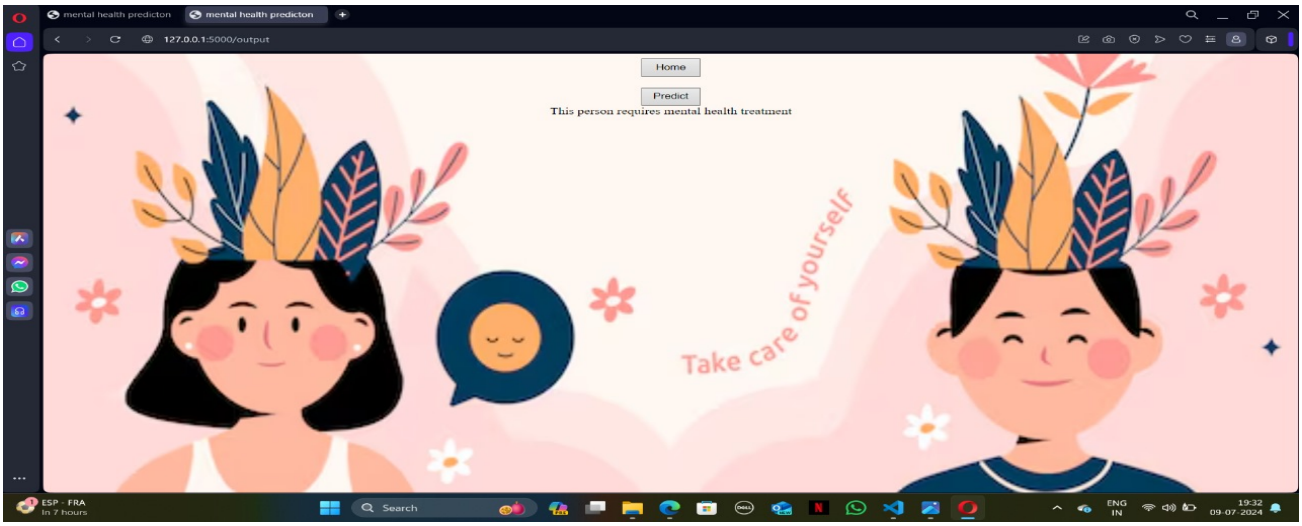

mental health predicton
mental health prediction
127.0.0.1:5000
Mental Health Prediction
Mental health affects your emotional,psychological and social well-being.It affects how we think,feel,and act.It also helps determine how we handle stress,relate to others, and make choices.In the workplace, communication and inclusion are keys skills for successful high performing teams or employees.The impact of mental health to an organization can mean an increase of absent days from work and a decrease in productivity and engagement.In the United States,approximately 70% of adults with depression are in the workforce.Employees with depression will miss an estimated 35 million workdays a year due mental illness.
Proceed


mental health predicton
mental health prediction
127.0.0.1:5000/pred?
age Enter age
Gender Male(0)/Female(1)
Country Enter Country
State State
Self_employed yes(1)
Family_history yes(0)/no(1)
work_interface Enter values between 0-3
no_employees Enter values btw 0-5
remote_work yes(0)/no(1)
Tech_company yes(1)/no(0)
benefits Enter values (0-2)
care_options Enter values (0-2)
wellness_program Enter values (0-2)
seek_help Enter values (0-2)
anonymity Enter values (0-2)
leave Enter values (0-4)
mental_health_consequence Enter values (0-2)
phys_health_consequence Enter values (0-2)
MENTAL HEALTH



PREDICTIONS





7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

There is an endless number of advantages of Machine learning. We can take a look at the one which are really helpful.

The advantages of Machine learning tell us how using Machine learning would benefit us. So, let’s have a Look at the advantages of Machine Learning.

-->Automation of Everything

-->Wide Range of Applications

-->Scope of Improvement

-->Efficient Handling of Data

-->Best for Mental Health

DISADVANTAGES:

Similar to the advantages of Machine Learning, we should also know the disadvantages of Machine Learning. If you don't know the cons, you won't know the risks of Machine Learning. So; let’s have a look at these disadvantages.

-->Possibility of High Error

-->Algorithm Selection

-->Data Acquisition

-->Time and Space

-->Internet Issues

8.APPLICATIONS

The applications of mental health prediction using machine learning (ML) are diverse and far-reaching, significantly enhancing mental health care across various sectors.

- **Early Detection and Diagnosis:**

ML models can identify early signs of mental health conditions such as depression, anxiety, and PTSD, facilitating timely interventions and improving patient outcomes.

- **Personalized Treatment Plans:**

By analyzing individual risk factors, ML algorithms can help develop tailored treatment strategies, enhancing the effectiveness of interventions and ensuring that patients receive the most appropriate care.

- **Resource Allocation:**

Healthcare systems can use predictive analytics to efficiently allocate resources, prioritizing care for high-risk individuals and optimizing the distribution of mental health services.

- **Preventive Care:**

Predictive models can identify individuals at risk of developing mental health issues, enabling the implementation of preventive measures and reducing the

incidence of severe mental health conditions.

9.CONCLUSION

Many different techniques and algorithms had been introduced and proposed to test and solve the mental health problems. There are still many solutions that can be refined. In addition, there are still many problems to be discovered and tested using a wide variety of settings in machine learning for the mental health domain. As classifying the mental health data is generally a very challenging problem, the features used in the machine learning algorithms will significantly affect the performance of the classification.

The existing studies and research show that machine learning can be a useful tool in helping understand psychiatric disorders. Besides that, it may also help distinguish and classify the mental health problems among patients for further treatment. Newer approaches that use data that arise from the integration of various sensor modalities present in technologically advanced devices have proven to be a convenient resource to recognize the mood state and responses from patients among others.

It is noticeable that most of the research and studies are still struggling to validate the results because of insufficiency of acceptable validated evidence, especially from the external sources. Besides that, most of the machine learning might not have the same performance across all the problems. The performance of the machine learning models will vary depending on the data samples obtained and the features of the data.

Moreover, machine learning models can also be affected by preprocessing activities such as data cleaning and parameter tuning in order to achieve optimal results.

Hence, it is very important for researchers to investigate and analyze the data with various machine learning algorithms to choose the highest accuracy among the machine learning algorithms.

10.FUTURE SCOPE

We can extend the Mental Health Prediction Project by using different constraints. Various machine learning approaches can be implemented to predict or detect a disease at its early stages so that the treatment for it would be less complex and it would increase the probability of the patient being cured. As a result of these approaches, different types of disease can been detected but With diverse accuracy levels depending on factors such as the used algorithm, feature set, training dataset, and so on.

11.BIBILOGRAPHY

The bibliography on mental health prediction using machine learning includes key works that explore various methodologies and applications. Notable sources include:

1. "**Machine Learning for Mental Health: A Systematic Review**"by D. Shatte, D. Hutchinson, and J. Teague (JMIR Mental Health, 2019) - This paper reviews ML applications in mental health.

2. **"Predicting Mental Health Disorders Using Machine Learning Models"** by J. Torous, M. Kiang, and S. Gutheil (Current Psychiatry Reports, 2021) - This article discusses ML's role in diagnosing and predicting mental health issues.

3. "Artificial Intelligence in Mental Health Services" by A. Ben-Zeev, D. E. Brian, and R. J. Wang (Psychiatric Services, 2020) - This work examines AI's integration into mental health services.

These sources provide comprehensive insights into the advancements and challenges of using machine learning for mental health prediction.

12.APPENDIX

Model building :

1)Dataset
2)Google colab and VS code Application Building
   1. HTML file (Index file, Home file, Out put file)
   1. CSS file
   2. Models in pickle format

SOURCE CODE:

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>mental health prediction</title>
</head>
<style>

   body {
  background-image: url("https://media.licdn.com/dms/image/D4D12AQHua22Rc22oWA/article-cover_image-shrink_720_1280/0/1654029354457?e=2147483647&v=beta&t=FqrsIlRQIHugIG5sFpifDalkITh5CTR8DolGuKwFY-E");
  background-repeat: no-repeat;
  background-size: 1540px 1070px;
}
```

```
</style>
<body>

 <div id="bg">
 <form action="{{url_for('output')}}" method="post">
  <label for=""><b>age</b></label><input type="number" placeholder="Enter age" max="100" name="a"><br><br>
  <label for=""><b>Gender</b></label><input type="number" placeholder="Male(0)/Female(1)" max="100"
name="b"><br><br>
  <label for=""><b>Country</b></label><input type="number" placeholder="Enter Country" max="100"
name="c"><br><br>
  <label for=""><b>State</b></label><input type="number" placeholder="State" max="1" min="0"
name="d"><br><br>
  <label for=""><b>Self_employed</b></label><input type="number" placeholder="yes(1)/no(0)" id="" min="0"
max="1" name="e"><br><br>
  <label for=""><b>Family_history</b></label><input type="number" placeholder="yes(0)/no(1)" name="f"><br><br>
  <label for=""><b>work_interface</b></label><input type="number" placeholder="Enter values between 0-3"
name="g"><br><br>
  <label for=""><b>no_employees</b></label><input type="number" placeholder="Enter values btw 0-5"
name="h"><br><br>
  <label for=""><b>remote_work</b></label><input type="number" placeholder="yes(0)/no(1)" name="i"><br><br>
  <label for=""><b>Tech_company</b></label><input type="number" placeholder="yes(1)/no(0)"
name="j"><br><br>
  <label for=""><b>benefits</b></label><input type="number" placeholder="Enter values (0-2)" name="k"><br><br>
  <label for=""><b>care_options</b></label><input type="number" placeholder="Enter values (0-2)"
name="l"><br><br>
  <label for=""><b>wellness_program</b></label><input type="number" placeholder="Enter values (0-2)"
name="m"><br><br>
  <label for=""><b>seek_help</b></label><input type="number" name="n" placeholder="Enter values (0-
2)"><br><br>
  <label for=""><b>anonymity</b></label><input type="number" name="o" placeholder="Enter values (0-
2)"><br><br>
  <label for=""><b>leave</b></label><input type="number" name="p" placeholder="Enter values (0-4)"><br><br>
  <label for=""><b>mental_health_consequence</b></label><input type="number" name="q" placeholder="Enter
values (0-2)"><br><br>
  <label for=""><b>phys_health_consequence</b></label><input type="number" name="r" placeholder="Enter
values (0-2)"><br><br>
  <label for=""><b>coworkers</b></label><input type="number" name="s" placeholder="Enter values (0-
2)"><br><br>
  <label for=""><b>supervisor</b></label><input type="number" name="t" placeholder="Enter values (0-
2)"><br><br>
  <label for=""><b>mental_health_interview</b></label><input type="number" name="u" placeholder="Enter value
(0-2)"><br><br>
  <label for=""><b>phys_health_interview</b></label><input type="number" name="v" placeholder="Enter value (0-
2)"><br><br>
  <label for=""><b>mental_vs_physical</b></label><input type="number" name="w" placeholder="Enter value (0-
2)"><br><br>
  <label for=""><b>obs_consequence</b></label><input type="number" name="x" placeholder="Enter value (0-
1)"><br><br>
  <label for=""><b>comments</b></label><input type="number" placeholder="Comments any" max="100"
name="y"><br><br>
```

```
   <button type="submit" style="height:50px; width:100px">Predict</button>
  </form>
  </div>
</body>
</html>
```

HOME.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>mental health prediction</title>
</head>
<style>
   body  {
  background-image:
url("https://magazine.medlineplus.gov/images/uploads/main_images/Teens_are_talking.jpg");
  background-repeat: no-repeat;
  background-size: 1540px 750px;
}
</style>
<body>

   <center>

         <h1>Mental Health Prediction</h1>
   </center>
   <p><h3>Mental health affects your emotional,psychological and social well-being.It affects how we
think,feel,and act.It also helps determine how we handle stress,relate to others, and make choices.In the workplace,
communication and
      inclusion are keys skills for successful high performing teams or employees.The impact of mental health to
an organization can mean an increase of absent days from work and a decrease in productivity and engagement.In
the United
      States,approximately 70% of adults with depression are in the workforce.Employees with depression will miss
an estimated 35 million workdays a year due mental illness.
   </h3></p>
   <center>
     <form action="{{url_for('pred')}}">

   <button type="submit" style="height:50px; width:100px">Proceed</button>
     </form>

</center>

</body>
```

```
</html>
```

OUTPUT.HTML

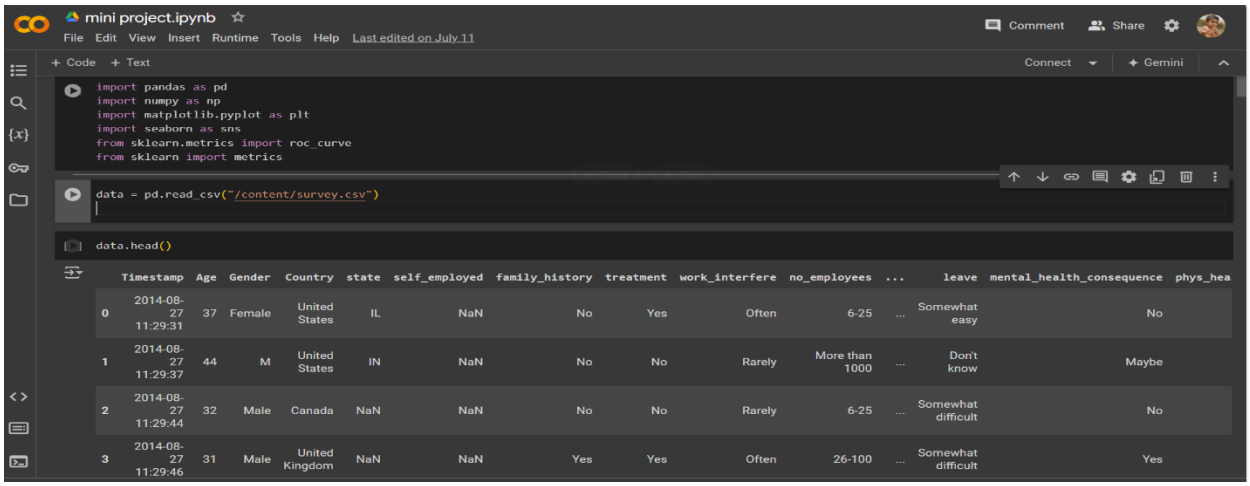
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>mental health predicton</title>
</head>
<style>
   body{
      background: url('https://img.freepik.com/free-vector/flat-world-mental-health-day-background_23-
2149631906.jpg?size=626&ext=jpg&ga=GA1.1.2113030492.1720310400&semt=ais_user');
      background-position: center;
      background-repeat: no-repeat;
      background-size: 1540px 1400px;
   }
</style>
<body>
   <div id="bg">
   <center>
      <center>
         <form action="{{url_for('home')}}">
            <button type="submit" style="height:30px; width:70px">Home</button>
         </form>

     </center>

</br>
       <form action="{{url_for('pred')}}">
           <button type="submit" style="height:30px; width:70px">Predict</button>
       </form>



   </center>

   <center>{{y}}</center>

   </div>
</body>
</html>
```

APP.PY

```
from flask import Flask,render_template,request,url_for
```

```
import pickle,joblib
import pandas as pd
app=Flask(_name_)

model = pickle.load(open("Scaler2.pkl", "rb"))
ct=joblib.load('feature_values')


@app.route('/')
def home():
   return render_template("home.html")
@app.route('/pred')
def pred():
   return render_template("index.html")

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

def output():

  age=request.form["a"]
  gender=request.form["b"]
  country=request.form["c"]
  state=request.form["d"]
  self_employed=request.form["e"]
  family_history=request.form["f"]
  work_interfere= request.form["g"]
  no_employees=request.form["h"]
  remote_work=request.form["i"]
  tech_company=request.form["j"]
  benefits=request.form["k"]
  care_options=request.form["l"]
  wellness_program =request.form["m"]
  seek_help=request.form["n"]
  anonymity= request.form["o"]
  leave=request.form["p"]
  mental_health_consequence=request.form["q"]
  phys_health_consequence = request.form["r"]
  coworkers=request.form["s"]
  supervisor= request.form["t"]
  mental_health_interview=request.form["u"]
  phys_health_interview= request. form[ "v"]
  mental_vs_physical=request.form["w"]
  obs_consequence=request.form["x"]
  comments = request.form["y"]
  data = [[age,gender, country, state, self_employed, family_history,work_interfere,no_employees, remote_work,
         tech_company, benefits, care_options, wellness_program,seek_help,anonymity, leave,
        mental_health_consequence,phys_health_consequence, coworkers, supervisor,
        mental_health_interview,phys_health_interview, mental_vs_physical,obs_consequence, comments]]
  feature_cols=['Age', 'Gender', 'Country', 'State', 'self_employed', 'family_history'
```

,'work_interfere','no_employees', 'remote_work', 'tech_company',



	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	no_employees	...	leave	mental_health_consequence	phys_hea
0	2014-08-27 11:29:31	37	Female	United States	IL	NaN	No	Yes	Often	6-25	...	Somewhat easy	No	
1	2014-08-27 11:29:37	44	M	United States	IN	NaN	No	No	Rarely	More than 1000	...	Don't know	Maybe	
2	2014-08-27 11:29:44	32	Male	Canada	NaN	NaN	No	No	Rarely	6-25	...	Somewhat difficult	No	
3	2014-08-27 11:29:46	31	Male	United Kingdom	NaN	NaN	Yes	Yes	Often	26-100	...	Somewhat difficult	Yes	

'benefits', 'care_options', 'wellness_program', 'seek_help',
'anonymity', 'Leave', 'mental_health_consequence',
'phys_health_consequence', 'coworkers', 'supervisor',
'mental_health_interview', 'phys_health_interview',
'mental_vs_physical', 'obs_consequence', 'comments']

```
 #pred=model.predict(ct.transform(pd.DataFrame(data,columns=feature_cols)))
 #pred=model.predict(ct.transform(pd.DataFrame(data)))
 pred = model.predict(data)
 pred = pred[0]
 if pred:
    return render_template("output.html",y="This person requires mental health treatment ")
 else:
    return render_template("output.html",y="This person doesn't require mental health treatment")

if _name_ =="_main_":
  app.run(debug=True)
```

CODE SNIPPETS

```
data.tail()
```

mini project.ipynb

File Edit View Insert Runtime Tools Help Last edited on July 11

Comment Share

+ Code + Text

Connect Gemini

```
data.shape
```

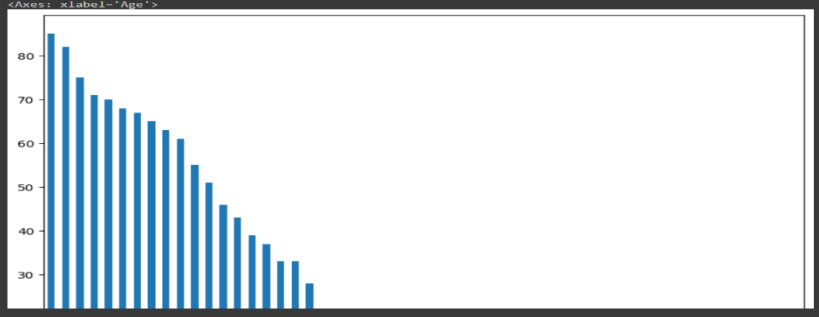
```
(1259, 27)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1259 entries, 0 to 1258
Data columns (total 27 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Timestamp                  1259 non-null   object
 1   Age                        1259 non-null   int64
 2   Gender                     1259 non-null   object
 3   Country                    1259 non-null   object
```

```
data['Age'].value_counts().plot(kind='bar',figsize=(10,8))
```

```
<Axes: xlabel='Age'>
```




```
Name: count, dtype: int64
```

```
data['Gender'].value_counts().plot(kind='bar',figsize=(10,8))
```

mini project.ipynb

File Edit View Insert Runtime Tools Help Last saved at 12:42 AM

Comment Share

+ Code + Text

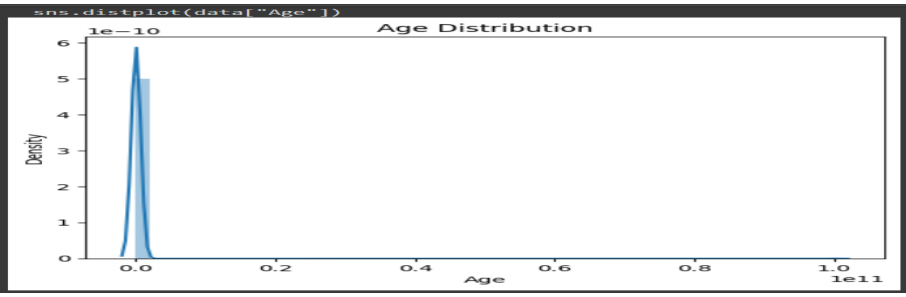
Connect Gemini

```
[ ] data['Gender'].replace(['Male', 'male', 'M', 'm', 'Male', 'Cis Male',
    'Man', 'cis male', 'Mail', 'Male-ish', 'Male (CIS)',
    'Cis Man', 'msle', 'Malr', 'Mal', 'maile', 'Make'], 'Male', inplace=True)

[ ] data['Gender'].replace(['Female', 'female', 'F', 'f', 'Woman', 'Female','femail','Cis Female','cis-female/femme','Femake','Female (cis)','woman'], 'Female', inplace

[ ] data["Gender"].replace(['Female (trans)', 'queer/she/they', 'non-binary', 'Fluid', 'queer', 'Androgyne', 'Trans-female', 'male leaning androgynous', 'Agender', 'A
```

```
import seaborn as sns
sns.distplot(data["Age"])
plt.title("Age Distribution")
plt.xlabel("Age")
plt.show()
```

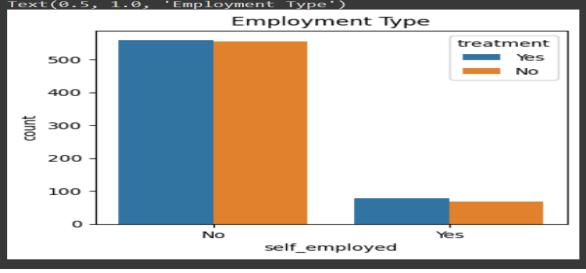
```
<ipython-input-20-45c49f9722e1>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data["Age"])
```

```
sns.distplot(data["Age"])
```



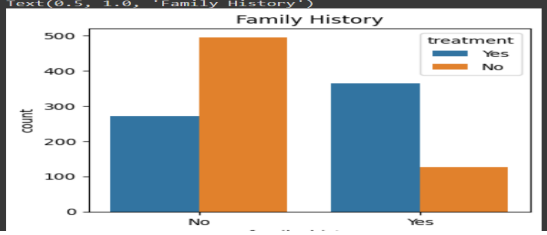

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,1)
sns.countplot(data=data, x='self_employed', hue='treatment')
plt.title('Employment Type')
```

```
Text(0.5, 1.0, 'Employment Type')
```




```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10,40))
plt.subplot(9,2,2)
sns.countplot(data=data,x='family_history',hue='treatment')
plt.title('Family History')
```

```
Text(0.5, 1.0, 'Family History')
```




```
plt.figure(figsize=(10,40))
plt.subplot(9,2,3)
sns.countplot(data=data,x='work_interfere',hue='treatment')
plt.title('Work Interference')
```

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,4)
sns.countplot(data=data,x='remote_work',hue='treatment')
plt.title('Work Type')
```

```
[ ] plt.figure(figsize=(10,40))
    plt.subplot(9,2,6)
    sns.countplot(data=data,x='benefits',hue='treatment')
    plt.title('benefits')
```

```
[ ] plt.figure(figsize=(10,40))
    plt.subplot(9,2,7)
    sns.countplot(data=data,x='care_options',hue='treatment')
    plt.title('care options')
```

```
Text(0.5, 1.0, 'care options')
```

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,8)
sns.countplot(data=data,x='mental_vs_physical',hue='treatment')
plt.title('equal importance to mental and physical health')
```

```
Text(0.5, 1.0, 'equal importance to mental and physical health')
```

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,9)
sns.countplot(data=data,x='wellness_program',hue='treatment')
plt.title('wellness program')
```

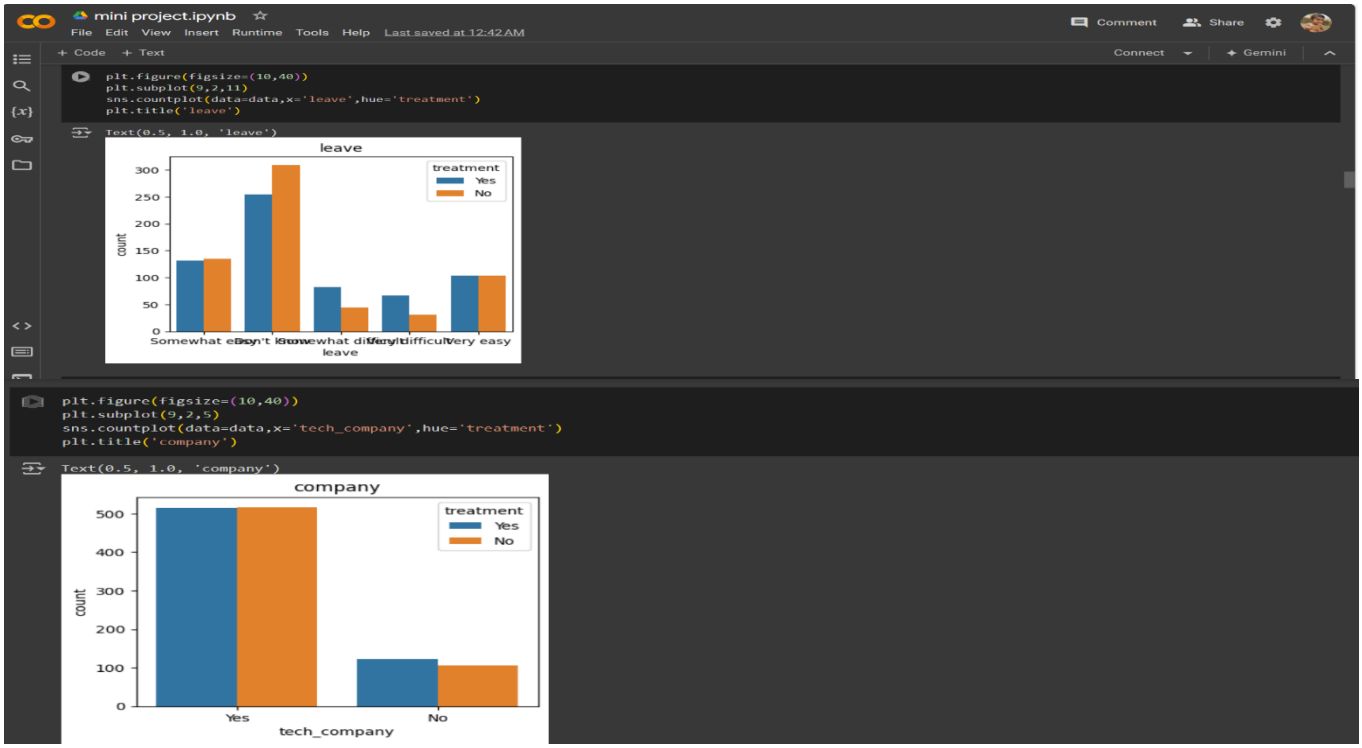
```
Text(0.5, 1.0, 'wellness program')
```

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,10)
sns.countplot(data=data,x='anonymity',hue='treatment')
plt.title('Anonymity')
```

```
Text(0.5, 1.0, 'Anonymity')
```

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,12)
sns.countplot(data=data,x='mental_health_consequence',hue='treatment')
plt.title('mental health consequence')
```

```
Text(0.5, 1.0, 'mental health consequence')
```


mini project.ipynb
File Edit View Insert Runtime Tools Help Last saved at 12:42 AM
Comment
Share
Connect
Gemini
+ Code
+ Text
plt.figure(figsize=(10,40))
plt.subplot(9,2,11)
sns.countplot(data=data,x='leave',hue='treatment')
plt.title('leave')
Text(0.5, 1.0, 'leave')
leave
treatment
Yes
No
count
300
250
200
150
100
50
0
leave

plt.figure(figsize=(10,40))
plt.subplot(9,2,5)
sns.countplot(data=data,x='tech_company',hue='treatment')
plt.title('company')
Text(0.5, 1.0, 'company')
company
treatment
Yes
No
count
500
400
300
200
100
0
Yes
No
tech_company

mini project.ipynb

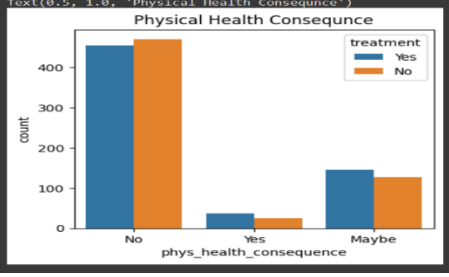
File Edit View Insert Runtime Tools Help Last saved at 12:42 AM

Comment Share Connect Gemini

+ Code + Text

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,13)
sns.countplot(data=data,x='phys_health_consequence',hue='treatment')
plt.title('Physical Health Consequnce')
```

```
Text(0.5, 1.0, 'Physical Health Consequnce')
```




```
plt.figure(figsize=(10,40))
plt.subplot(9,2,14)
sns.countplot(data=data,x='coworkers',hue='treatment')
plt.title('Discussion with coworkers')
```

```
Text(0.5, 1.0, 'Discussion with coworkers')
```

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,15)
sns.countplot(data=data,x='supervisor',hue='treatment')
plt.title('Discussion with supervisor')
```

```
Text(0.5, 1.0, 'Discussion with supervisor')
```

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,16)
sns.countplot(data=data,x='mental_health_interview',hue='treatment')
plt.title('Discussion with Interviewer(Mental )')
```

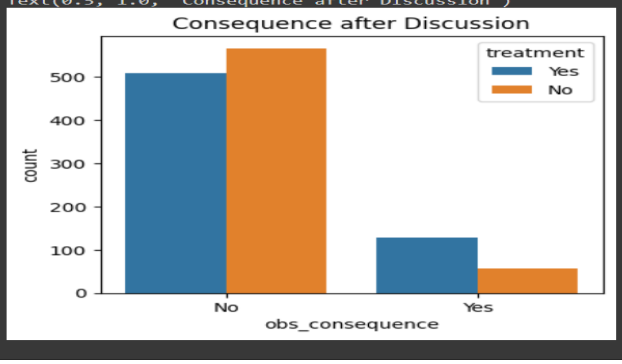
```
Text(0.5, 1.0, 'Discussion with Interviewer(Mental )')
```

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,17)
sns.countplot(data=data,x='phys_health_interview',hue='treatment')
plt.title('Discussion with Interviewer(Physical)')
```

```
Text(0.5, 1.0, 'Discussion with Interviewer(Physical)')
```

```
plt.figure(figsize=(10,40))
plt.subplot(9,2,18)
sns.countplot(data=data,x='obs_consequence',hue='treatment')
plt.title('Consequence after Discussion')
```

```
Text(0.5, 1.0, 'Consequence after Discussion')
```



mini project.ipynb

File Edit View Insert Runtime Tools Help Last saved at 12:42 AM

Comment Share Connect Gemini

+ Code + Text

```
data.describe(include='all')
```

	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	no_employees	remote_work	...	leave	mental_health_consequen
count	1.259000e+03	1259	1259	744	1259	1259	1259	995	1259	1259	...	1259	12
unique	NaN	7	48	45	2	2	2	4	6	2	...	5	
top	NaN	Male	United States	CA	No	No	Yes	Sometimes	6-25	No	...	Don't know	N
freq	NaN	988	751	138	1113	767	637	465	290	883	...	563	4
mean	7.942815e+07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	Na
std	2.818299e+09	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	Na
min	-1.726000e+03	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	Na
25%	2.700000e+01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	Na
50%	3.100000e+01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	Na
75%	3.600000e+01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	Na
max	1.000000e+11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	Na

11 rows × 26 columns

```
[ ] data.tail()
```

```
[ ] x=data.drop(['treatment'],axis = 1)
    y=data['treatment']
```

```
x.head()
```

	Age	Gender	Country	state	self_employed	family_history	work_interfere	no_employees	remote_work	tech_company	...	leave	mental_health_consequence	phy
0	37	Female	United States	IL	No	No	Often	6-25	No	Yes	...	Somewhat easy	No	
1	44	Male	United States	IN	No	No	Rarely	More than 1000	No	No	...	Don't know	Maybe	
2	32	Male	Canada	NaN	No	No	Rarely	6-25	No	Yes	...	Somewhat difficult	No	
3	31	Male	United Kingdom	NaN	No	Yes	Often	26-100	No	Yes	...	Somewhat difficult	Yes	
4	31	Male	United States	TX	No	No	Never	100-500	Yes	Yes	...	Don't know	No	

5 rows × 25 columns

```
import pandas as pd # Import pandas library

# Assuming 'data' is your Pandas DataFrame
y = data['treatment']

# Convert the NumPy array 'y' back to a Pandas Series if you need to use .head()
y = pd.Series(y)

y.head()
```

```
0    Yes
1     No
2     No
3    Yes
4     No
Name: treatment, dtype: object
```

```
[ ] from sklearn.compose import ColumnTransformer
    from sklearn.preprocessing import LabelEncoder,OrdinalEncoder
```

```
[ ] x=data.drop(['treatment'],axis = 1)
    y=data['treatment']
    x = pd.DataFrame(x)
```

```
[ ] ct =  ColumnTransformer([('oe',OrdinalEncoder(),['Gender','self_employed','family_history','work_interfere','no_employees','remote_work','tech_company','benefits',
```

```
[ ] oe = OrdinalEncoder()
    for col in x.columns:
      if x[col].dtype == 'object':
        x[col] = oe.fit_transform(x[col].values.reshape(-1,1))
```

```
x.head()
```

	Age	Gender	Country	state	self_employed	family_history	work_interfere	no_employees	remote_work	tech_company	...	leave	mental_health_consequence	phys_he
0	37	0.0	45.0	10.0	0.0	0.0	1.0	4.0	0.0	1.0	...	2.0	1.0	
1	44	3.0	45.0	11.0	0.0	0.0	2.0	5.0	0.0	0.0	...	0.0	0.0	
2	32	3.0	7.0	NaN	0.0	0.0	2.0	4.0	0.0	1.0	...	1.0	1.0	
3	31	3.0	44.0	NaN	0.0	1.0	1.0	2.0	0.0	1.0	...	1.0	2.0	
4	31	3.0	45.0	37.0	0.0	0.0	0.0	1.0	1.0	1.0	...	0.0	1.0	

5 rows × 25 columns

```
[ ] x["obs_consequence"].value_counts()
```

```
obs_consequence
0.0    1075
1.0     184
Name: count, dtype: int64
```

```
[ ] le=LabelEncoder()
    y=le.fit_transform(y)
```

```
import joblib
joblib.dump(ct,'feature_values')
```

```
['feature_values']
```

```
[ ] from sklearn.model_selection import train_test_split
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=49)
```

```
[ ] x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((1007, 25), (252, 25), (1007,), (252,))
```

```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.neighbors import KNeighborsClassifier
    from sklearn.ensemble import RandomForestClassifier,AdaBoostClassifier,GradientBoostingClassifier
    from xgboost.sklearn import XGBClassifier
    from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

mini project.ipynb

File Edit View Insert Runtime Tools Help Last saved at 12:42 AM

Comment Share

+ Code + Text

Connect Gemini

```
model_dict={}

model_dict['LogisticRegression']=LogisticRegression(solver='liblinear',random_state=49)
model_dict['KNN classifier']=KNeighborsClassifier()
model_dict['DecisionTreeClassifier']=DecisionTreeClassifier(random_state=49)
model_dict['RandomForestClassifier']=RandomForestClassifier(random_state=49)
model_dict['AdaBoostClassifier']=AdaBoostClassifier(random_state=49)
model_dict['GradientBoostingClassifier']=GradientBoostingClassifier(random_state=49)
model_dict['XGBClassifier']=XGBClassifier(random_state=49)
```

```
from sklearn.impute import SimpleImputer

# Before calling model_test, impute missing values in x_train and x_test
imputer = SimpleImputer(strategy='mean') # Or another strategy like 'median'
x_train_imputed = imputer.fit_transform(x_train)
x_test_imputed = imputer.transform(x_test)

for model_name,model in model_dict.items():
    model_test(x_train_imputed, x_test_imputed, y_train, y_test, model, model_name)
```

```
score is :0.5198412698412699
score is :0.5952380952380952
score is :0.7857142857142857
score is :0.7817460317460317
score is :0.8333333333333334
score is :0.8333333333333334
score is :0.8373015873015873
```

```
abc=AdaBoostClassifier(random_state=99)
abc.fit(x_train_imputed,y_train)
pred_abc=abc.predict(x_test_imputed)
print('Accuracy of AdaBoost=',accuracy_score(y_test,pred_abc))
```

```
Accuracy of AdaBoost= 0.8333333333333334
```

```
[ ] from sklearn.model_selection import RandomizedSearchCV
params_abc = {'n_estimators': [int(x) for x in np.linspace(start=1, stop=50, num=15)],
              'learning_rate': [(0.97 + x / 100) for x in range(0, 8)], }
abc_random = RandomizedSearchCV(random_state=49, estimator=abc, param_distributions=params_abc,
                                n_iter=50, cv=5, n_jobs=-1)
```

```
[ ] params_abc
```

```
{'n_estimators': [1, 4, 8, 11, 15, 18, 22, 25, 29, 32, 36, 39, 43, 46, 50],
 'learning_rate': [0.97, 0.98, 0.99, 1.0, 1.01, 1.02, 1.03, 1.04]}
```

```
[ ] abc_random.fit(x_train_imputed,y_train)
```

RandomizedSearchCV
estimator: AdaBoostClassifier
AdaBoostClassifier

```
[ ] abc_random.best_params_
```

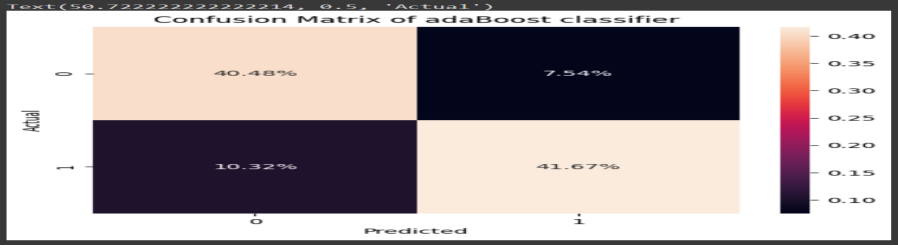
```
{'n_estimators': 29, 'learning_rate': 1.0}
```

```
[ ] abc_tuned=AdaBoostClassifier(random_state=49,n_estimators=11,learning_rate=1.02)
    abc_tuned.fit(x_train_imputed,y_train)
    pred_abc_tuned=abc_tuned.predict(x_test_imputed)
    print('Accuracy of AdaBoost(tuned)=',accuracy_score(y_test,pred_abc_tuned))
```

```
Accuracy of AdaBoost(tuned)= 0.8214285714285714
```

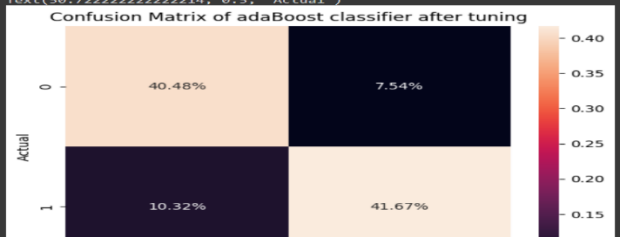
```
[ ] cf_matrix=confusion_matrix(y_test,pred_abc_tuned)
    sns.heatmap(cf_matrix/np.sum(cf_matrix),annot=True,fmt='.2%')
    plt.title('Confusion Matrix of adaBoost classifier')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
```

```
[ ] Text(50.722222222222214, 0.5, 'Actual')
```

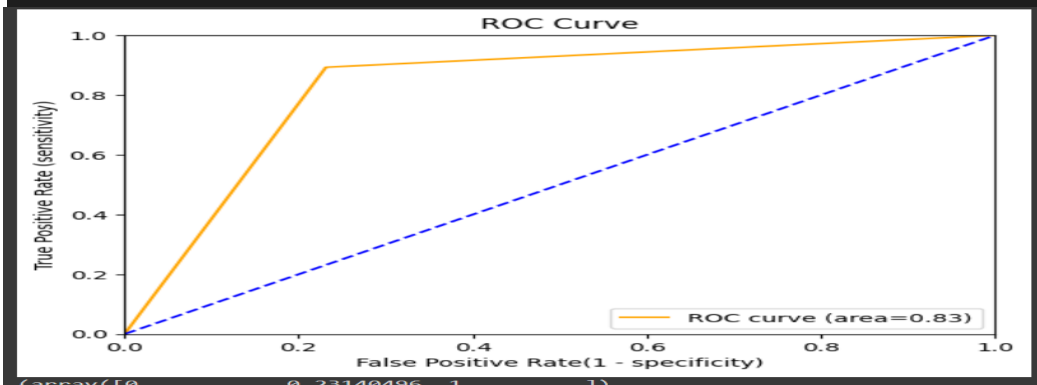


```
cf_matrix=confusion_matrix(y_test,pred_abc_tuned)
sns.heatmap(cf_matrix/np.sum(cf_matrix),annot=True,fmt='.2%')
plt.title('Confusion Matrix of adaBoost classifier after tuning')
plt.xlabel('Predicted')
plt.ylabel('Actual')
```

```
Text(50.72222222222214, 0.5, 'Actual')
```

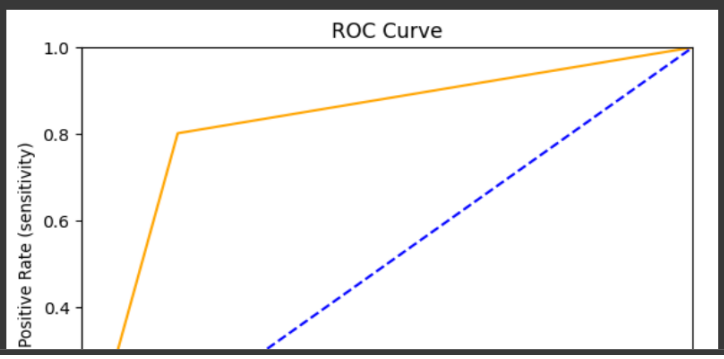


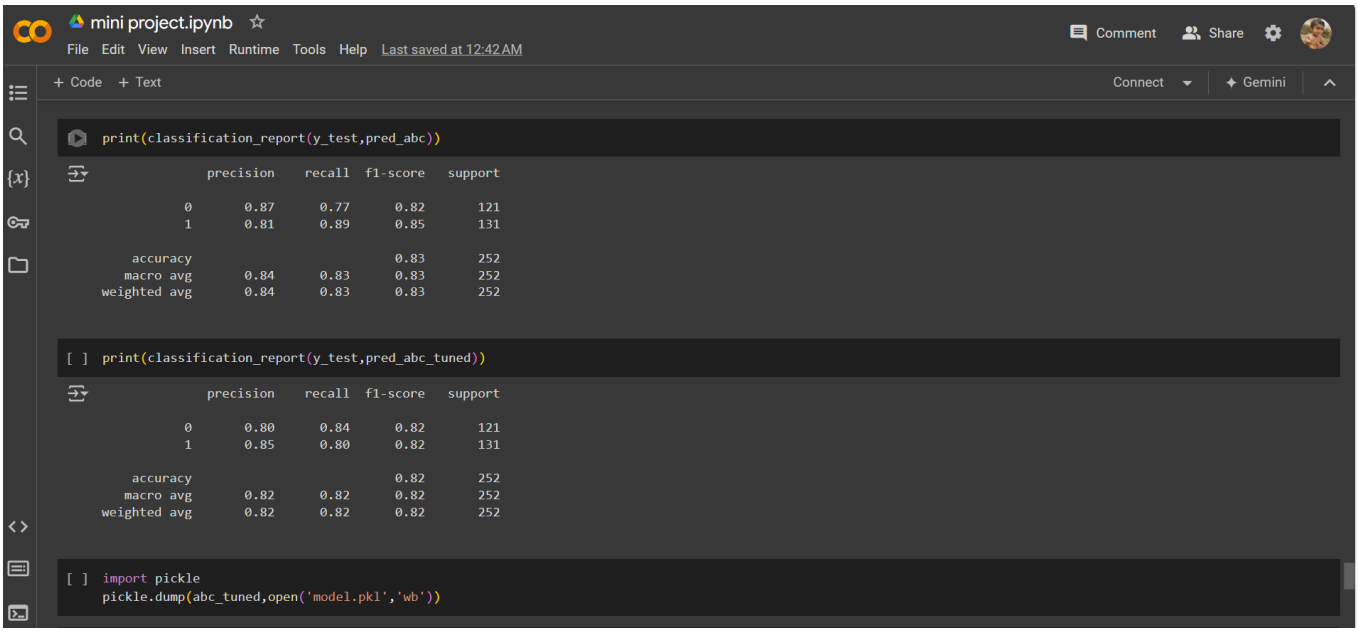

```
fpr_abc, tpr_abc, thresholds_abc = roc_curve(y_test, pred_abc)
roc_auc_abc = metrics.auc(fpr_abc, tpr_abc)
plt.plot(fpr_abc, tpr_abc, color="orange", label='ROC curve (area=%0.2f)'%roc_auc_abc)
plt.plot([0,1],[0,1],color='blue',linestyle='--')
plt.xlim(0.0,1.0)
plt.ylim(0.0,1.0)
plt.title('ROC Curve')
plt.xlabel('False Positive Rate(1 - specificity)')
plt.ylabel('True Positive Rate (sensitivity)')
plt.legend(loc="lower right")
plt.show()
roc_curve(y_test,pred_abc)
```




```
(array([0.        , 0.23140496, 1.        ]),
 array([0.        , 0.89312977, 1.        ]),
 array([2, 1, 0]))
```

```
fpr_abc_tuned, tpr_abc_tuned, thresholds_abc_tuned = roc_curve(y_test, pred_abc_tuned)
roc_auc_abc_tuned = metrics.auc(fpr_abc_tuned, tpr_abc_tuned)
plt.plot(fpr_abc_tuned, tpr_abc_tuned, color="orange", label='ROC curve (area=%0.2f)'%roc_auc_abc_tuned)
plt.plot([0,1],[0,1],color='blue',linestyle='--')
plt.xlim(0.0,1.0)
plt.ylim(0.0,1.0)
plt.title('ROC Curve')
plt.xlabel('False Positive Rate(1 - specificity)')
plt.ylabel('True Positive Rate (sensitivity)')
plt.legend(loc="lower right")
plt.show()
roc_curve(y_test,pred_abc_tuned)
```




mini project.ipynb
File Edit View Insert Runtime Tools Help Last saved at 12:42 AM
Comment
Share
+ Code + Text
Connect
Gemini
print(classification_report(y_test,pred_abc))
precision recall f1-score support
0 0.87 0.77 0.82 121
1 0.81 0.89 0.85 131
accuracy 0.83 252
macro avg 0.84 0.83 0.83 252
weighted avg 0.84 0.83 0.83 252
[ ] print(classification_report(y_test,pred_abc_tuned))
precision recall f1-score support
0 0.80 0.84 0.82 121
1 0.85 0.80 0.82 131
accuracy 0.82 252
macro avg 0.82 0.82 0.82 252
weighted avg 0.82 0.82 0.82 252
[ ] import pickle
pickle.dump(abc_tuned,open('model.pkl','wb'))