

Project review
on
Mental Health At Workplace

Sri Sai institute of technology and science

Under guidance of :

D.ESWARAIAH,
M.Tech.,

presented By:

Attar Faiza	- 18F71A0509,
Mayana Samreen khanam	- 18F71A0524,
Shaik Naziya	- 18F71A0513,
Syed Afreen	- 18F71A0503

Index

- Introduction
- Abstract
- Proposed system
- Advantages of proposed system
- Software requirements
- Exploratory data analysis
- Modelling
- Performance Metrics
- Deployment

What is Mental Health at Workplace?





Introduction

- Mental health effects your emotional , psychological and social wellbeing it effects how we think ,feel and act.it also helps determine how we handle stress relate to others and make healthy choices. In the workplace communication and inclusion are skills for successful high performing teams or employees.

Abstract

The idea of this project is to gather the personal data of employees from multiple data sources and use various machine learning algorithms on this data to extract important information. This model can be used by the organisations to check if the organization's employees are suffering from mental health. In this project random forest classifier is used to predict if a certain employee has mental issues or not.

Random forest classifier provides higher accuracy through cross validation. Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data.

Proposed system

- In the proposed system we will build a machine learning model by training the model with training_dataset and this model will do the work of taking care of employees for productivity.
- The employee's data is given input to the machine learning model.
- On the basis of the training_dataset , the model will predict whether an employee needs treatment on mental health or not.

Advantages of proposed system

- Initially the model will be trained only once with trained dataset and it will automatically predict if an employee needs treatment or not of any number of employees without any need of specialists.
- The model is unbiased.
- The model is accurate.

System Requirements

HARDWARE REQUIREMENTS :

- system : INTEL I5
- Hard disk : 500 gb
- Ram : 4 gb
- Operating system : windows 10

Software requirements :

- Web framework : flask
- Technologies : python, html, css, bootstrap
- Libraries : pandas , numpy , seaborn , matplotlib , scikit-learn

Hosting environment : Heroku

Modules

- EDA - by MAYANA SAMREEN KHANAM
- Modelling - by ATTAR FAIZA
- Performance Tuning - by SYED AFREEN
- Deployment - by SHAIK NAZIYA SULTHANA

DATA WRANGLING & EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis[EDA]

❖ EDA

After performing the cleaning and preprocessing of the data, we perform data analysis and visualizations on our dataset. We try to analyze our data more clearly to find any trends or patterns in the dataset.

❖ Data cleaning and Pre-processing:

One of the first steps is to make sure that the dataset we are using is accurate .the dataset should not have any missing values and if the data set does have missing values they should be replaced by appropriate values.

we performed data cleaning and pre-processing on age and gender.

TECHNIQUE

❖ LIBRARIES USED IN OUR PROJECT

we install and import all these libraries in python.

➤ Pandas

pandas is defined as an open-source library that provides high performance data manipulation in python.

They provides you with a huge set of important commands and features which are used to easy analyze your data.

It used for working with data set. It has functions for analyzing, cleaning, Exploring and manipulating data.

We get the insights about the data set using some functions in pandas such as head()

tail(), info(), describe(), sample().

There are several useful functions for detecting removing and replacing null values in pandas data frame, such as is isnull(), filla(), replace(), drop()

TECHNIQUES

➤ Numpy

Numpy (Numerical Python) is an open-source core Python library for scientific computations. It is a general-purpose array and matrices processing package.

Numpy is compatible with and used by many other popular python packages, including pandas and matplotlib.

Numpy makes many mathematical operations used widely in scientific computing fast and easy to use.

TECHNIQUES

➤ Matplotlib

Matplotlib is a low-level graph plotting library in python that serves as a visualization utility.

Matplotlib is opensource and we can use it freely.

Pyplot is a matplotlib module provides functions that interact with the figure.

Decorates the plot with labels, creates plotting area in figure.

Different plots can be plotted using this library they are

- 1.Bar Graph,
- 2.Pie Chart,
- 3.Boxplot,
- 4.Histogram,
- 5.Line Chart and
- 6.Scatter plot.

TECHNIQUES

➤ Seaborn

seaborn is a library mostly used for statistic plotting in python.

It is built on top of Matplotlib and provides beautiful default styles and colour palettes to make statistical plots more attractive.

Different Categorical of plot in seaborn

Plots are basically used for visualizing the relationship between variables.

Those variables can be either be completely numerical or a category like a group, class or division. Seaborn divides plot into the below categories-

Relational plots: This plot is used to understand the relation between two variables.

Categorical plots: The plot deals with categorical variables and how they can be visualized.

Distribution plots: This plot is used for examining univariate and bivariate distributions

Regression plots: The regression plots in seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.

Matrix plots: A matrix plot is an array of scatterplots.

Data Acquisition

- Attributes of data we used:
 - Age
 - Gender
 - Employment
 - Family-history
 - Work interference
 - Number of employees
 - Remote work or not
 - Tech company or not
 - Benefits of employees
 - Mental and physical health interview.

READING THE DATASET AND GETTING INSIGHTS ABOUT THE DATA

- ❑ Let's read the dataset using the Pandas module and print the 1st five rows

```
df = pd.read_csv('employees.csv')  
df.head()
```

- ❑ We get the insights about the data using the following functions

`df.shape` --- for the shape of the data

`df.describe()` -- for the distribution of data.

`df.info()` -- for the columns and their data types

WRANGLING

DATA WRANGLING

One of the first steps is to make sure that the dataset we are using is accurate. The dataset should not have any missing values and if the dataset does have missing values, they should be replaced by the appropriate value.

- ☐ **Handling Missing Value**
- ☐ **Outlier Treatment**

- **DATA VISUALIZATION**

Data Visualization is the process of analyzing data in the form of graphs or maps, making it easier to understand patterns in the data.

There are various types of Visualizations:

- ❑ UNIVARIATE ANALYSIS
- ❑ BI-VARIATE ANALYSIS
- ❑ MULTI-VARIATE ANALYSIS

Methodology

We will use Matplotlib and Seaborn library for the data visualization. Some commonly used Graphs are

Univariate

- Bar plot
- Box plot
- Hist plot

Bivariate

- Bar plot
- Hist plot
- Scatter plot
- Box ploy

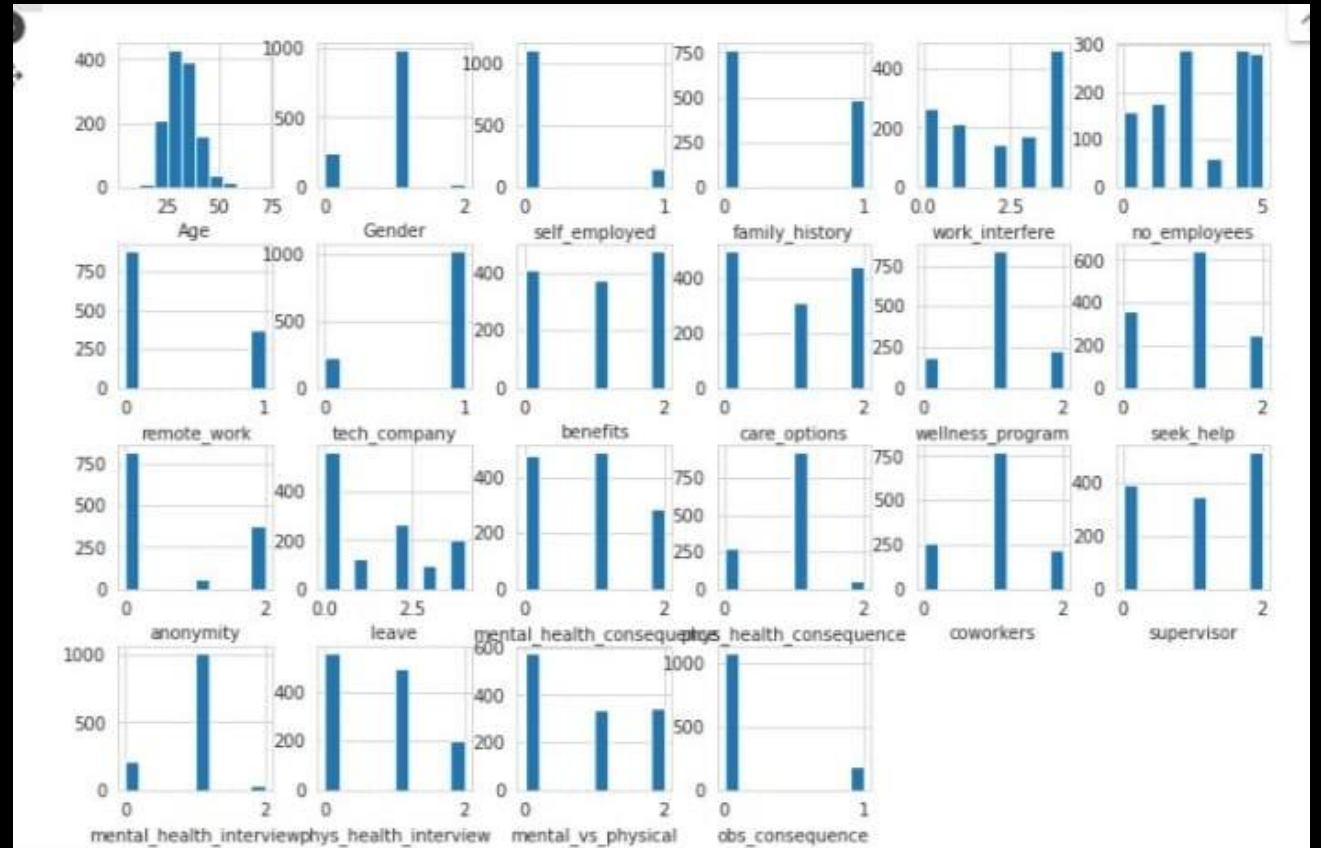
Multivariate

- Pair plot
- Heatmap

UNI-VARIANT

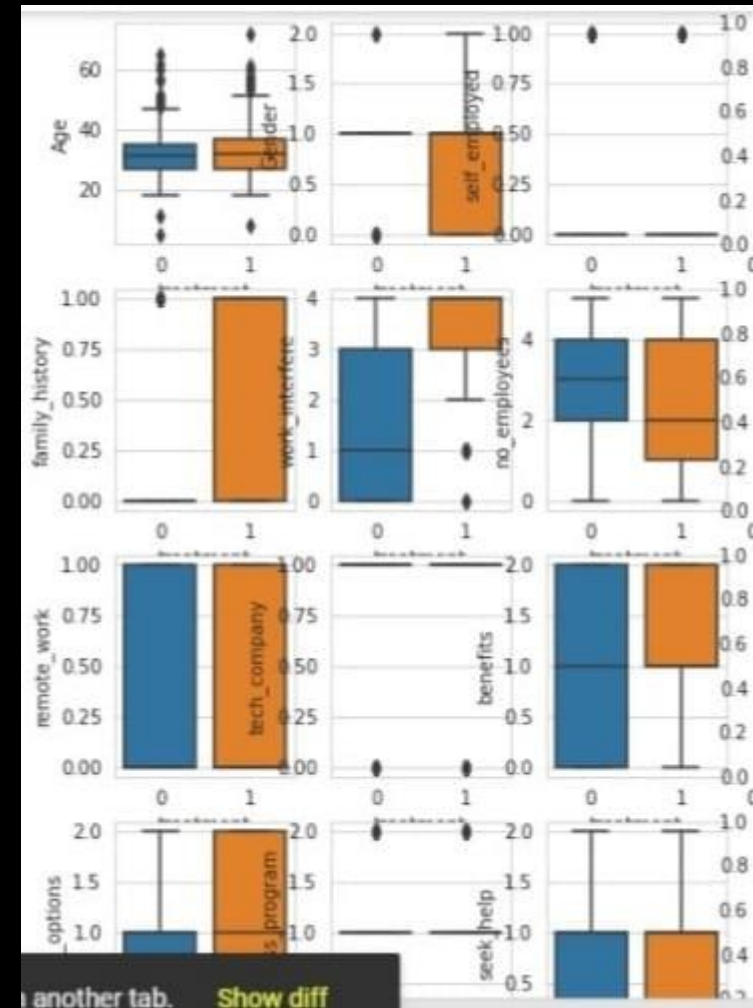
Univariate analysis is the simplest form of analyzing data. Uni means “one” so in other words your data has only one variable

A variable in univariate analysis is just a condition or subset that your data falls into. For example, the analysis might look at a variable of “age” or it might be more than one variable at a time otherwise it becomes bivariate analysis.



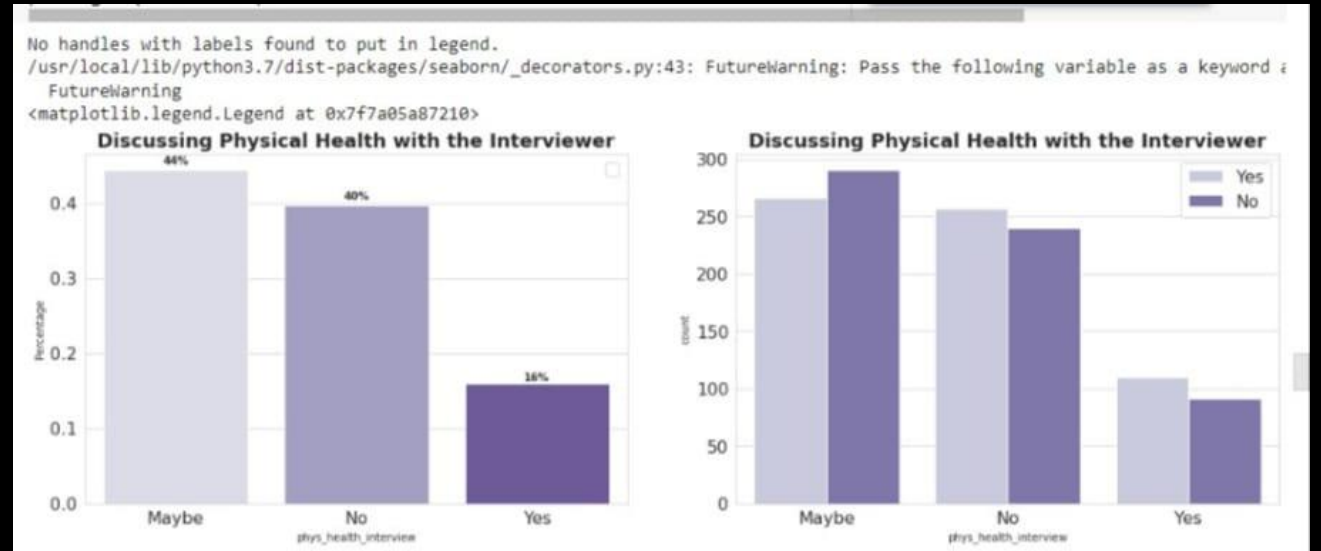
BI-VARIANT

Bi – variate analysis is performed to find the relationship between each variable in the data set and the target variable of interest or using 2 variables and finding the relationships between them



BAR GRAPH

This is a bar graph where the input column (physical_health_interviewe) show the effect and influence on output column(treatment).



CO-RELATION

❑ Correlation coefficients are used to measure the strength of the linear relationship between two variables.

❑ A correlation coefficient greater than zero indicates a positive relationship while a value less than zero signifies a negative relationship.



Modelling

❖ FEATURES SELECTION

After performing the data cleaning and visualizations, we implemented our machine learning algorithms on the features of the dataset.

The features selected of the dataset are gender, age, work_interface, self_employed, tech_company, wellness_program, coworkers, Supervisor, leave, remote_work, no_employees and etc...

X is the set of input features from the data set.

y is the output feature from the data set.

The following code is used for selecting the features.

```
X=df.drop(columns='treatment')
```

```
y=df['treatment']
```

❖ SCIKIT-LEARN LIBRARY

Scikit-learn is an open -source Machine Learning Python package that offers functionality supporting supervised and unsupervised learning.

Additionally, it provides tools for model development, selection and evaluation as well as many other utilities including data pre processing functionality .

❑ supervised learning

supervised learning is an approach to creating Artificial intelligence(AI) where a computer algorithm is trained on input data have been labeled for a particular.

❑ Unsupervised learning

unsupervised learning is use of Artificial intelligence(AI) algorithms to identify patterns in datasets containind data points that are neither classified nor labeled.

❖ Methods in Scikit-Learn Package

fit_transform()

It joins the fit() and transform() method for the transformation of the dataset.

It is used on the training data ,so that we can scale the training data and also learn the scaling parameters.

fit()

This method calculates the parameters μ (mean) and σ (standard deviation) and saves them as internal objects.

transform()

Using these same parameters, using this method we can transform a particular dataset. Used for pre-processing before modeling.

predict()

Use the above-calculated weights on the test data to make the predictions.

❖ SPLITTING

The next step is building the machine learning model. While building the machine learning model, first we need to split our dataset into 2 parts i.e.: training data and test data.

The **Sklearn train test split** function helps us create our training and test data. We import this function from the model_selection package in scikit-learn library.

We have split the training data and we apply our machine learning algorithms on the features of the dataset.

The Syntax for splitting is given below

```
from sklearn.model_selection import train_test_split  
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=.2,stratify=y,random_state=1)
```

❖ Scaling

Scaling is a technique to standardize the independent features present in data in a fixed range. It is performed during the data pre-processing to handle highly varying units.

Techniques to perform Scaling are

- ❑ Standard Scalar
- ❑ Min-Max Scalar

❑ STANDARD SCALAR

It is very effective technique which rescales a feature value so that it has distribution with 0-Mean value and 1-Variance.

❑ MIN-MAX SCALAR

This technique rescales a feature or observation value between 0 and 1. The Scikit learn provides the implementation of scaling in a preprocessing package. We import MinMaxScaler or StandardScaler from preprocessing package to perform scaling.

Machine learning algorithms used

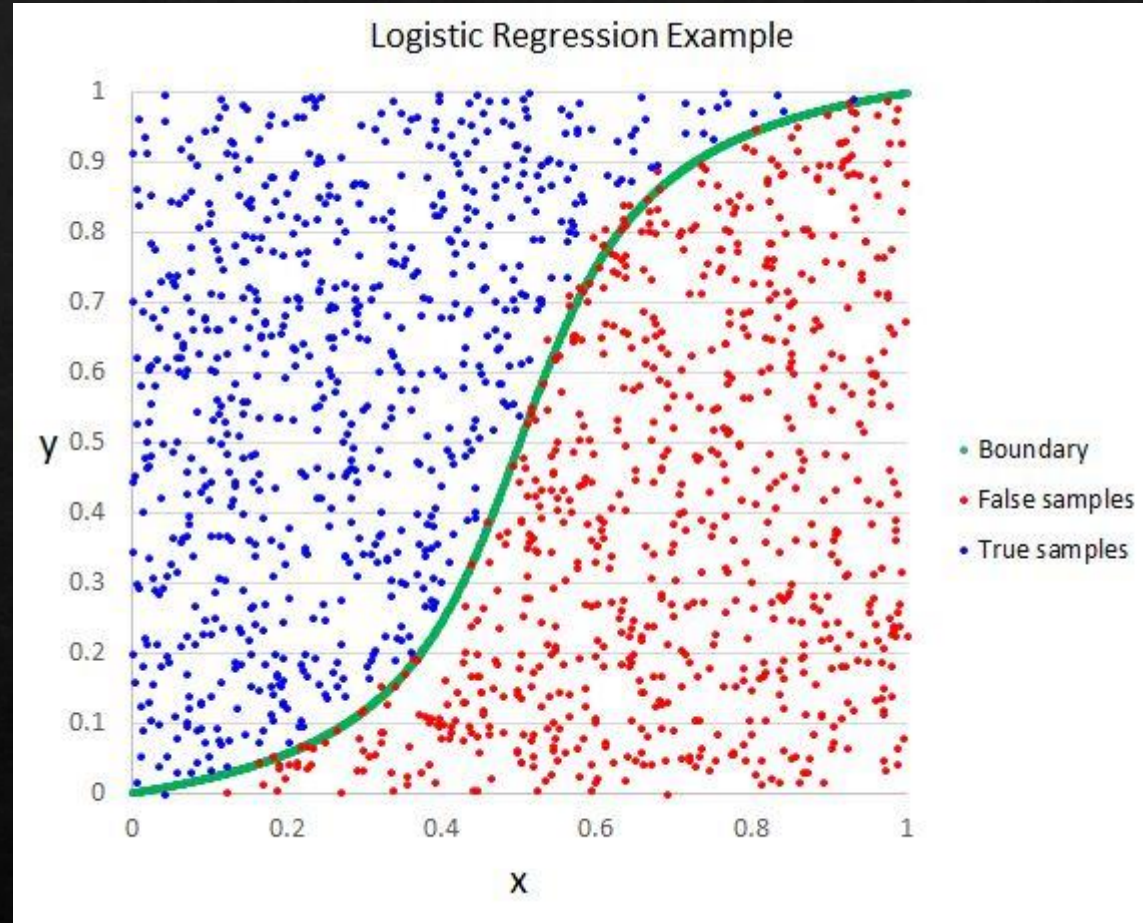
- Logistic regression
- Decision tree
- K-nearest neighbor
- Random forest classifier

Logistic Regression

Logistic regression is a supervised learning technique it solves the classification problem. it gives the results in binary either true or false.

It uses sigmoid function, used to model the data in logistic regression.

$$F(x) = 1 / (1 + e^{-x})$$



❑ Syntax for using logistic regression

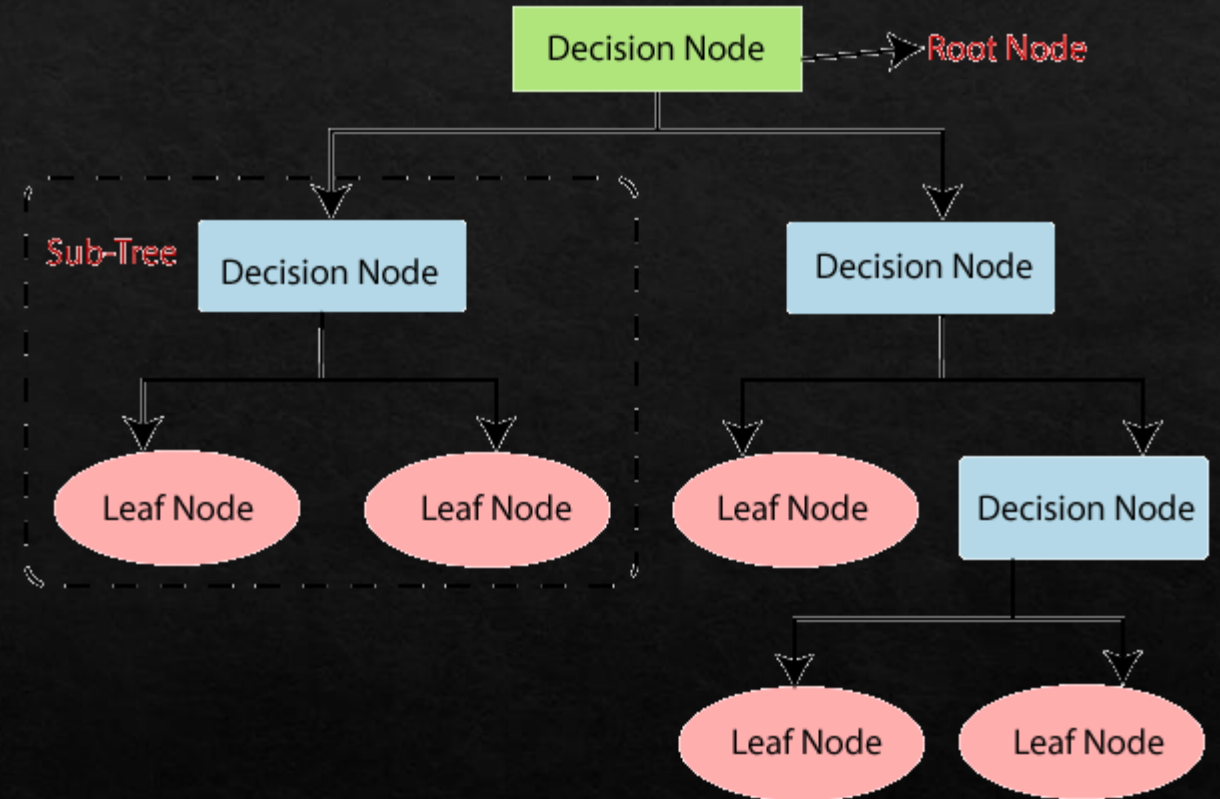
```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
model_logistic=Logistic Regression()
model_logistic.fit(X_train_scaled,y_train)
```

Decision Tree

Decision tree is one the most popular machine learning algorithms used mostly in classification problems but can also be used for regression type of problems.

It is Tree structure classifier where internal nodes represent the feature of the data set, branches represent the decision rules & each leaf node represents the outcome.

The split of the nodes can either happen by calculating Gini impurity (calculate the measure of impurity) or information gain (calculate the change in the entropy)



□ Syntax for Decision Tree

```
from sklearn.tree import DecisionTreeClassifier  
model_decisiontree = DecisionTreeClassifier(criterion="entropy", random_state=100,  
                                             max_depth=3, min_samples_leaf=5)  
model_decisiontree.fit(X_train, y_train)
```

K-Nearest Neighbor

KNN algorithms assume the similarity between the new case and the available cases and put the new case into the category that is most similar to available categories.

KNN algorithm can be used for regression as well as for classification. But, it is mostly used for classification problems.



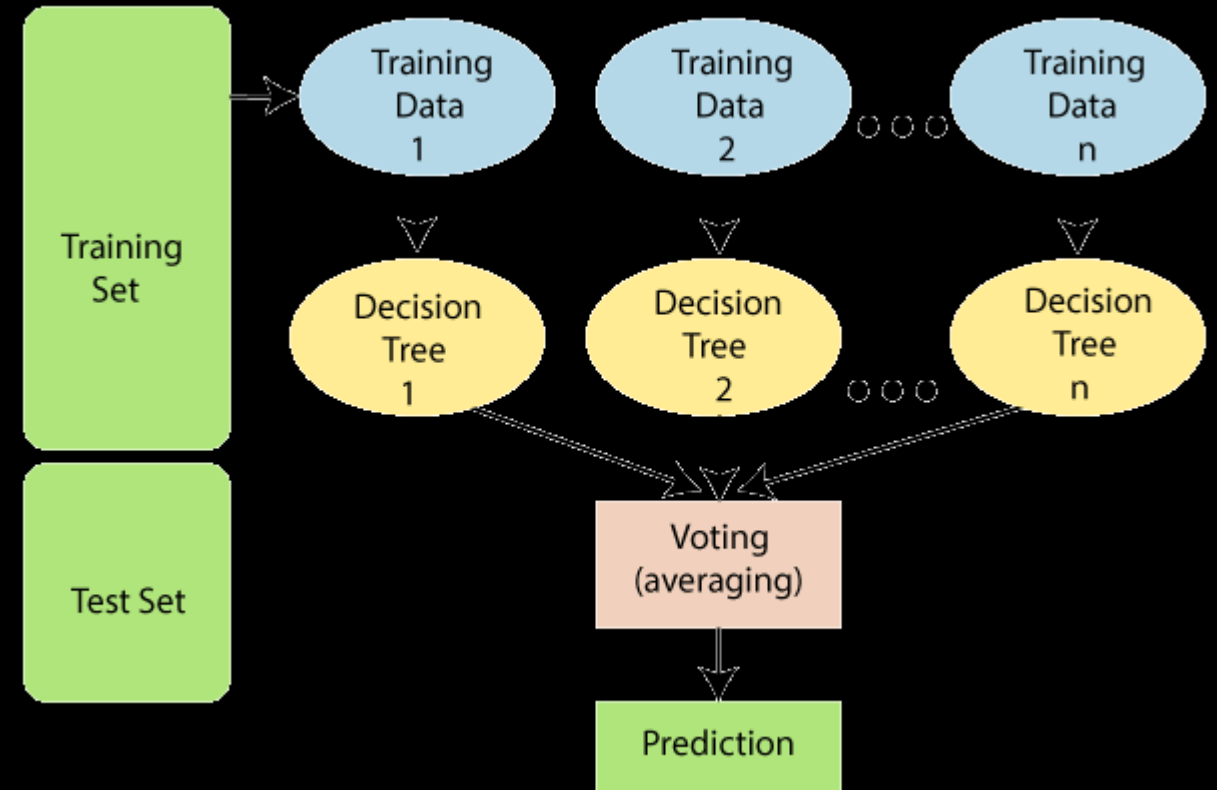
❑ Syntax of K-Nearest Neighbour

```
from sklearn.neighbors import KNeighborsClassifier  
Knn = KNeighborsClassifier()  
Knn.fit(X_train,y_train)
```

Random Forest Classifier

Random forest is a classifier that contains a number of decision trees on various subsets of the given datasets and takes the average to improve the predictive accuracy of the data set

The greater number of the trees in the forest leads to higher accuracy and prevents the problem of over fitting



❑ Syntax for using random forest

```
from sklearn.ensemble import RandomForestClassifier
model_randomforest=RandomForestClassifier(n_estimators=100,random_state=42,
                                           n_jobs=-1,max_depth=5,oob_score=True,min_samples_split=2)
model_randomforest.fit(X_train,y_train)
```


PERFORMANCE METRICS

PERFORMANCE METRICS

❖ CONFUSION MATRIX

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

`confusion_matrix` function of `sklearn.metrics`

The Confusion Matrix

		Actual	
		Positive	Negative
Predicted	Positive	TRUE POSITIVE	FALSE POSITIVE Type 1 error
	Negative	FALSE NEGATIVE Type 2 error	TRUE NEGATIVE

PERFORMANCE METRICS

❖ ACCURACY

Accuracy is the most prominent factor for the evaluation of a machine learning model. It is most common performance metric for classification algorithms.

On the basis of accuracy, we decide whether our machine learning model is applicable in real world or not. If the accuracy of an algorithm implemented is high, then eventually it means the system is more closer to real world.

We can use **accuracy_score function of sklearn.metrics** to compute accuracy of our classification model.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

PERFORMANCE METRICS

❖ **Classification Report**

- **Precision**
- **Recall**
- **F1-score**
- **support**

PERFORMANCE METRICS

Precision

Precision may be defined as the number of correct documents returned by our ML model.

- We can easily calculate it by confusion matrix with the help of following formula –

$$\text{Precision} = \text{TP} / \text{TP} + \text{FP}$$

Recall

Recall may be defined as the number of positives returned by our ML model.

We can easily calculate it by confusion matrix with the help of following formula-

$$\text{Recall} = \text{TP} / \text{TP} + \text{FN}$$

PERFORMANCE METRICS

F1-Score

This score will give us the harmonic mean of precision and recall. Mathematically, F1 score is the weighted average of the precision and recall. The best value of F1 would be 1 and worst would be 0. We can calculate F1 score with the help of following formula –

$$F2 \text{ Score} = 2PR/P+R$$

Support

Support may be defined as the number of samples of the true response that lies in each class of target values.

PERFORMANCE METRICS

As mentioned below we calculate the performance metrics for classification problems

```
y_predict = model.predict(X_test_scaled)
```

```
From sklearn.metrics import accuracy_score, classification_report
```

```
accuracy_score(y_test,y_predict)
```

```
classification_report(y_test,y_predict)
```

PERFORMANCE METRICS

- We can use **classification_report function of sklearn.metrics** to get the classification report of our classification model.
- After applying all the performance metrics on four algorithms on our dataset, we can see that RANDOM FOREST CLASSIFIER gives us the highest accuracy out of all four algorithms.
- This is the classification report we got when we used the random forest classifier.

```
[91] y_predict = model.predict(X_test_scaled)

[92] from sklearn.metrics import accuracy_score, classification_report
     print('accuracy_score :', accuracy_score(y_test, y_predict))

accuracy_score : 0.8207171314741036

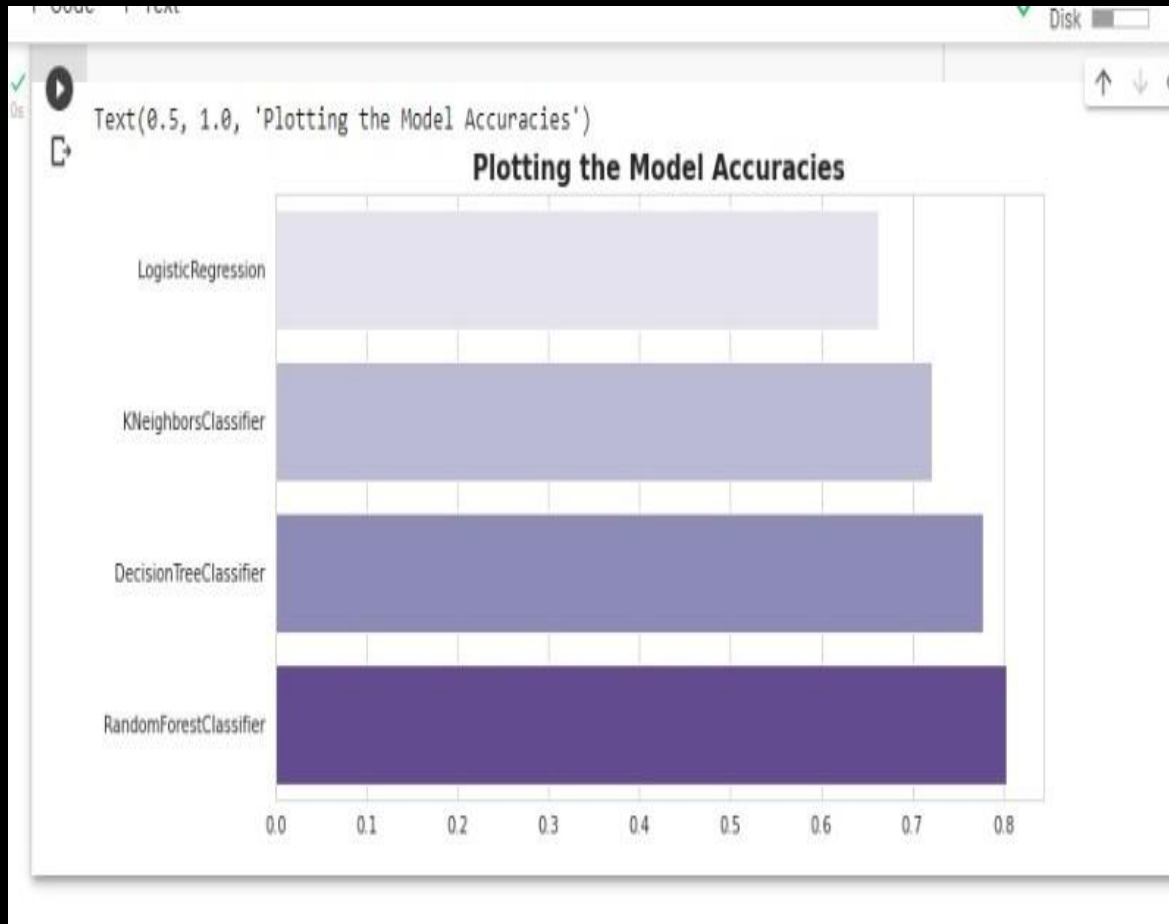
[93] print('classification_report :\n', classification_report(y_test, y_predict))

classification_report :
              precision    recall  f1-score   support

      0               0.84       0.79       0.81        124
      1               0.81       0.85       0.83        127

   accuracy                   0.82        251
  macro avg              0.82       0.82       0.82        251
 weighted avg              0.82       0.82       0.82        251
```


Accuracies Comparision



<i>models</i>	<i>accuracy</i>
Logistic regression	0.8037
K-neighbours Classifier	0.6631
Decision tree classifier	0.7214
Random forest classifier	0.8271

DEPLOYMENT

Deployment

- In order to deploy the trained model for the organization's to use off, we would need an application with the simple user interface which employees can utilize.
- Thus, here we made a simple web interface using HTML, CSS & BOOTSTRAP.
- Lastly, we wanted to predict the results for the obtained values from the user for which we made use of the FLASK framework to integrate the backend and front end
- And we generated the pickle file for our model to generate the prediction for the input data
- In next step, we have built a UI for a user to input his data so that once he enters the information for all the inputs. The model processes the data and will recommend if the employee needs treatment or not

HTML

- Html stands for HyperText Markup languages.
- It is used to design web pages using a markup language.
- Html is the combination of Hypertext and markup language.
- Hypertext defines the link between the web page.
- A markup language is used to define the text document within tag which is used to define the structure of web pages. This language is used to annotate text.
- so the machine can understand it and manipulate text accordingly. Most markup languages are human-readable.

CSS

- css stands for cascading style sheets.css describes how html elements are to be displayed on screen,paper, or in other media.External style sheets are stored in css files.
- More importantly, css enables you to do independent of the html that makes up each web page it prescribes colors,fronts, spacing and much more.
- css lets developers and designers define how it behaves including how elements are positioned in the browser.

BOOTSTRAP

- Bootstrap is a **free and open-source CSS framework directed at responsive, mobile-first front-end web development**. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap.

Flask

- Flask is a python-based microframework used for developing small scale websites. flask is very easy to make Restful API's using python.
- Flask is a small and lightweight python web framework that provides useful tools and features that make creating web application in python easier.
- It gives developer flexibility and is a more accessible framework for new developers.
- Flask , a backend this compact and controlled is capable of handling all the data processing required to support a full-featured frontend tracking app.

DEPLOYMENT

□ SET UP PROJECT

- Step 1: open command prompt.
- Step2: Install virtual environment.
pip install virtualenv
- Step3: Give the name to the virtual environment.
virtualenv my_env
- Step4 :Activate the virtual environment.
my_venv/scripts/activate
- Step5: Install Flask Module
Pip install flask

DEPLOYMENT

- Now create an app that hosts the application.

```
app = Flask(__name__)
```

- Then create *route* that calls a Python function that maps to the browser.

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

```
def index():
```

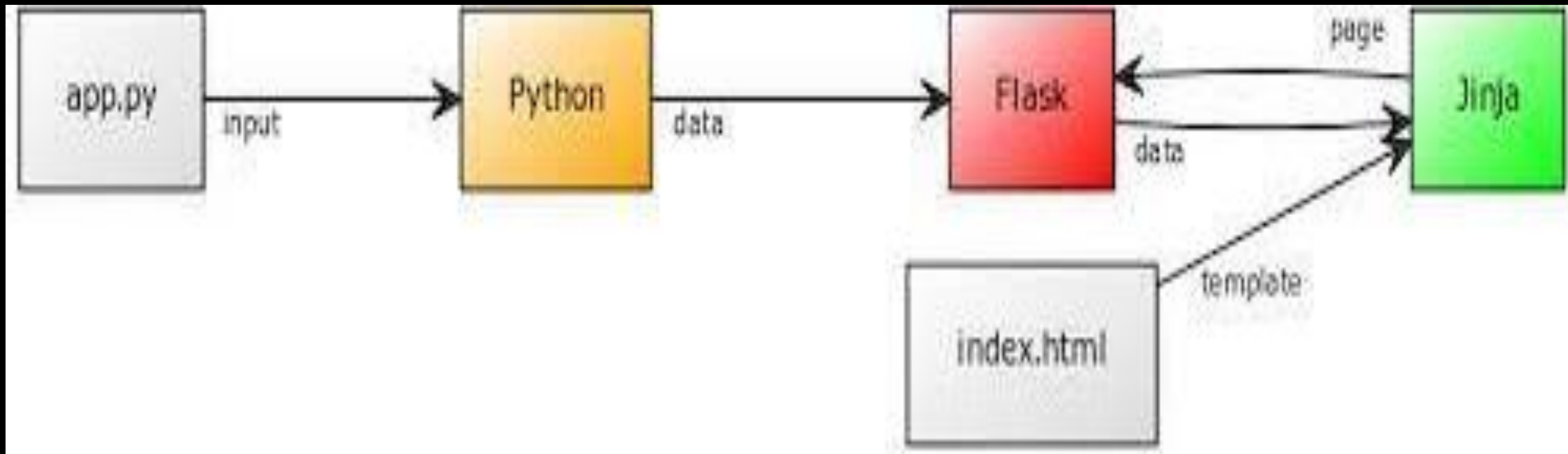
- Use the return statement to return the function .
- To run the function use the below code

```
app.run(debug=True)
```

- Now run the python file in terminal, you will get the URL.
- Enter the url link in your web browser to see the website.

DEPLOYMENT

- We can use either Html or jinja2 template engine to return the output of a function on which the flask is based.
- Instead of returning hardcoded HTML from the function, we have used jinja2
- Html file can be rendered by using the **render_template()** function.



DEPLOYMENT

A web application often requires a static file such as a **javascript** file or a **CSS** file supporting the display of a web page.

Usually, the web server is configured to serve them for you, but during the development, these files are served from *static* folder in your package or next to your module and it will be available at */static* on the application.

DEPLOYMENT

- Generate the pickle file for our model to predict the input data.

In the next step, we have built a user interface.

- The model will process the data and will recommend the appropriate type of treatment is necessary or not in such a condition..

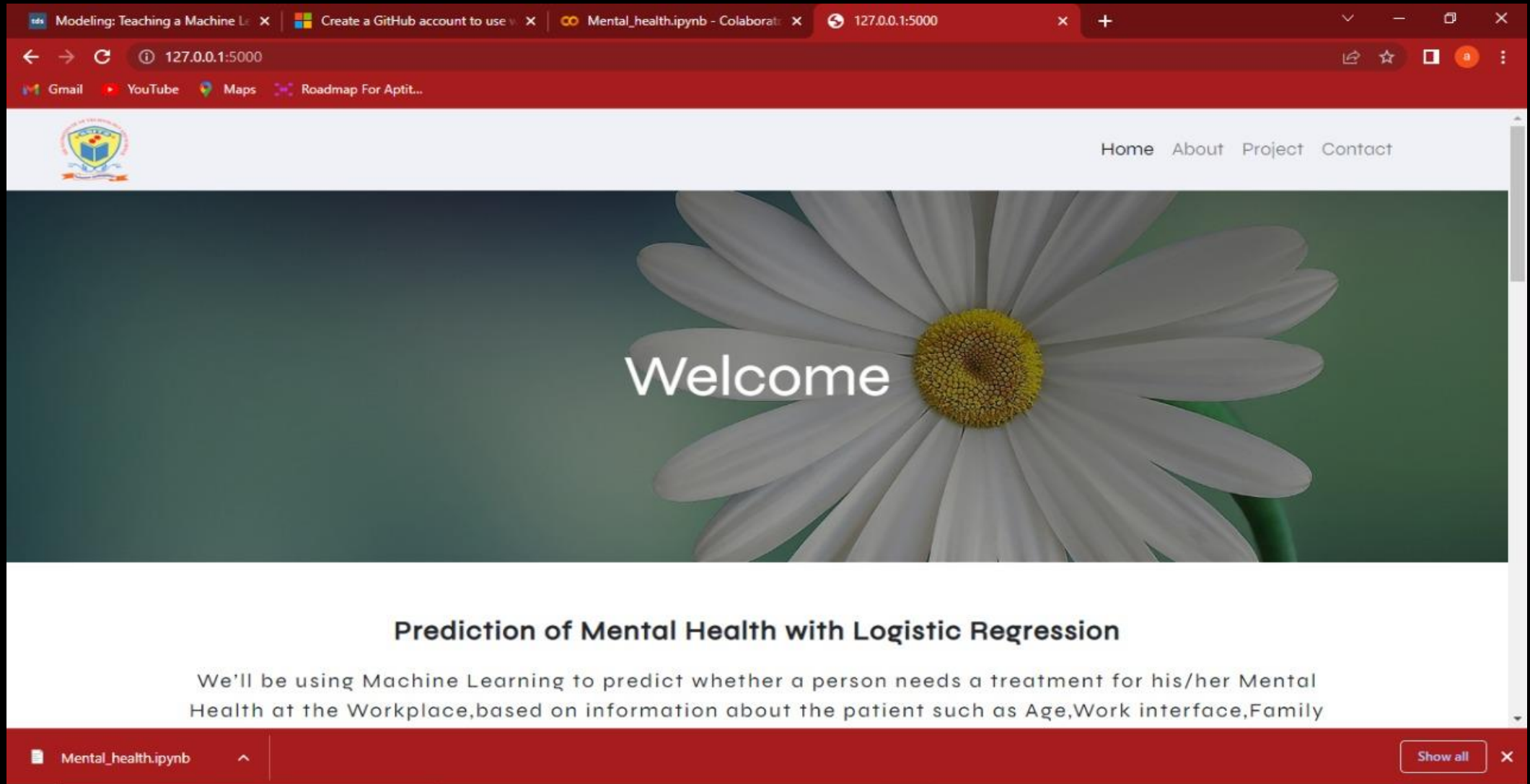
DEPLOYMENT

- ❑ To Deploy the project in the CLOUD, we are using **HEROKU Platform**.
- ❑ and operate applications entirely in the cloud.
- ❑ To Deploy the Project Using Heroku we need to follow the following Steps.
- ❑ In the terminal go to Project Location and install the gunicorn which is a python based HTTP Server.
- ❑ Now go to your Folder and Create the folder **Procfile** without extensions.
- ❑ In Terminal use this command **pip freeze > requirements.txt**
- ❑ Install the Git and **HerokuCli** in your system.
- ❑ Now Initialize the git in terminal. To initialize use **git init**
- ❑ To add files use **git add**
- ❑ Commit the git **git commit -m "msg"**

DEPLOYMENT

- Login into Heroku, use **heroku login**
- Now create the application using **heroku AppName**
- Add a remote to your local repository
herokugit: remote -a AppName
- Push the files to Heroku
git push heroku master
- Now application is deployed into the cloud.

WELCOME PAGE



The screenshot shows a web browser window with multiple tabs. The active tab is titled "127.0.0.1:5000". The browser's address bar also shows "127.0.0.1:5000". The page features a logo on the left and navigation links (Home, About, Project, Contact) on the right. The main content area has a large white daisy flower on a green background with the word "Welcome" in white text. Below this, the title "Prediction of Mental Health with Logistic Regression" is displayed, followed by a paragraph explaining the project's goal: "We'll be using Machine Learning to predict whether a person needs a treatment for his/her Mental Health at the Workplace, based on information about the patient such as Age, Work interface, Family". The bottom of the browser window shows a red status bar with the file name "Mental_health.ipynb" and a "Show all" button.

Modeling: Teaching a Machine Learning | Create a GitHub account to use | Mental_health.ipynb - Colaboratory | 127.0.0.1:5000

127.0.0.1:5000

Gmail YouTube Maps Roadmap For Aptit...

Home About Project Contact

Welcome

Prediction of Mental Health with Logistic Regression

We'll be using Machine Learning to predict whether a person needs a treatment for his/her Mental Health at the Workplace, based on information about the patient such as Age, Work interface, Family

Mental_health.ipynb Show all

Input page

Mental_health.ipynb - Colaborat... 127.0.0.1:5000/form

127.0.0.1:5000/form

Gmail YouTube Maps Roadmap For Aptit...

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

☐ Yes

☐ No

☐ May be

Do you feel that your employer takes mental health as seriously as physical health?

☐ Yes

☐ no

☐ Don't know

Have you heard of or observed negative consequences for coworkers with mental health conditions in your workplace?

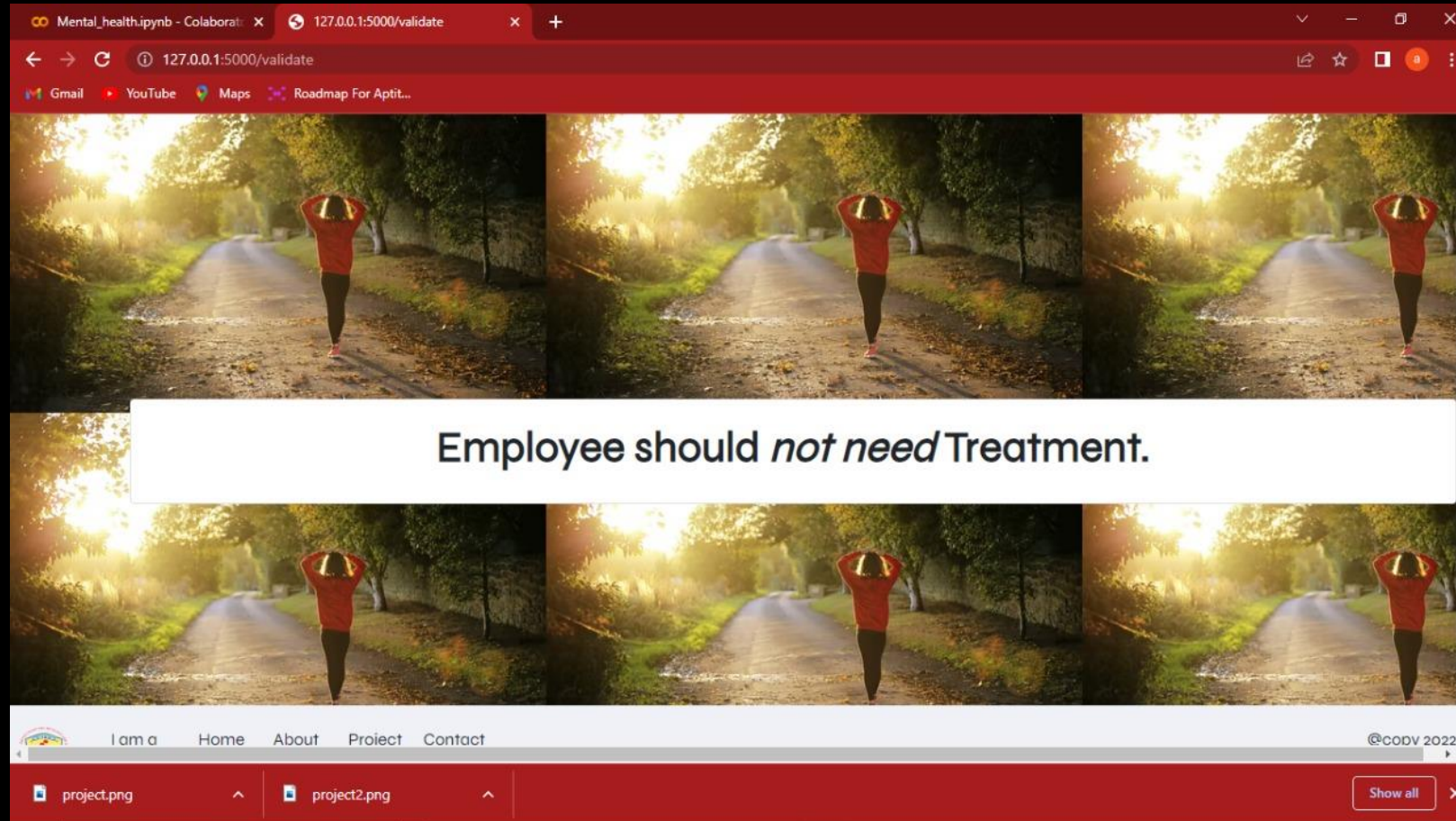
☐ YES

☐ NO

Predict

project.png project2.png Show all

Output page



Future Scope

- This applicant can be implemented as a mobile app and be made available free for the anyone to download which they can use it to predict if they need to consult the doctor or not.
- The mental issues of employees are frequently increasing cause of all the global issues like virus outbreaks and many more.it has been seen that mental problems are more in this decade than last decade.

CONCLUSION

- we made this project to promote the mental health awareness of the employees. because, mental health is the most important factor in the production.
- Using machine learning algorithms, we trained the model with 4 following algorithms namely logistic regression, Decision Tree,K-neighbours classifier & Random Forest. And we got highest accuracy of 82% with random forest classifier. So we selected this model as best model for future prediction.

Thank you