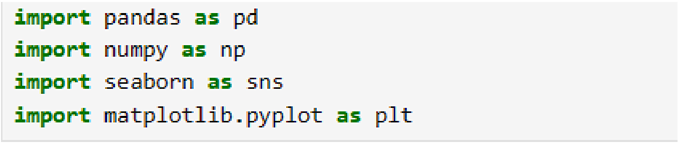
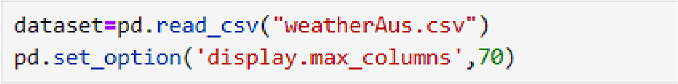
# Project output

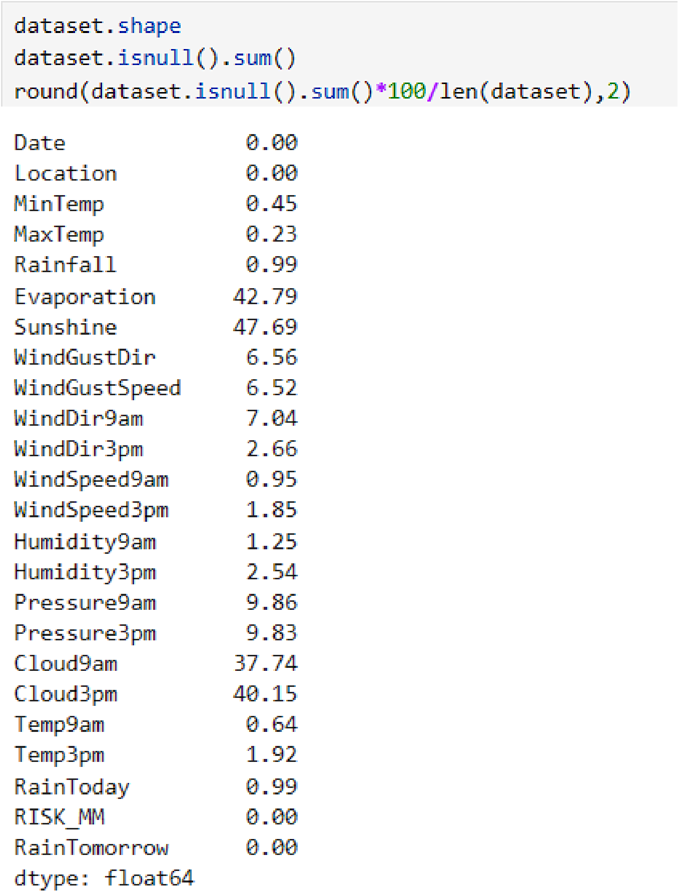
| | |
|---|---|
| Date | 12.11.2023 |
| Team id | Team-592242 |
| Project name | Machine learning approach for predicting rainfall |



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

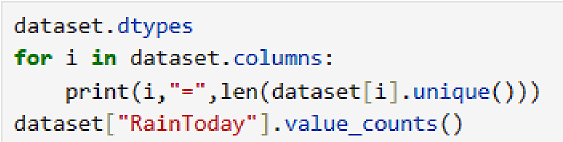


```
dataset=pd.read_csv("weatherAus.csv")
pd.set_option('display.max_columns',70)
```

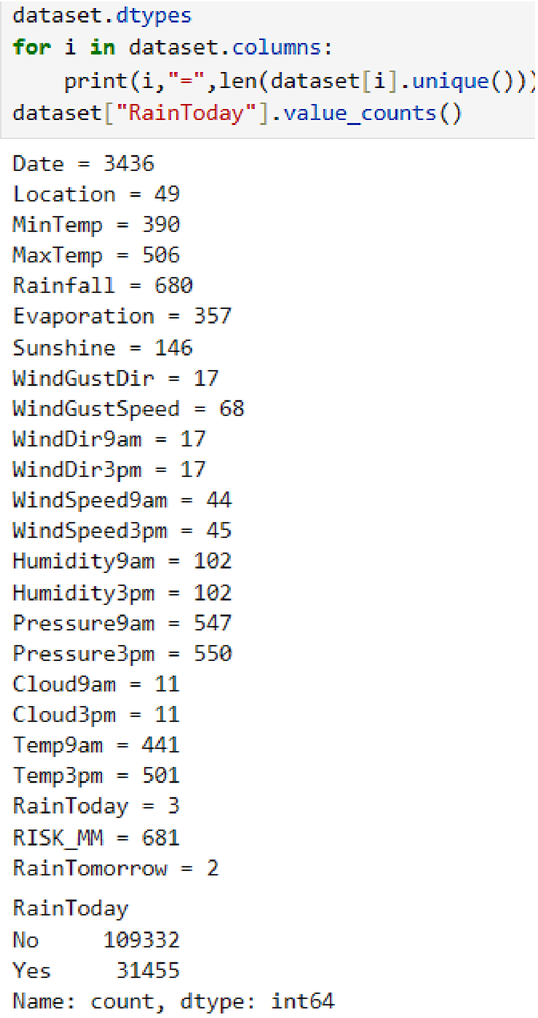


```
dataset.shape
dataset.isnull().sum()
round(dataset.isnull().sum()*100/len(dataset),2)
```

```
Date              0.00
Location          0.00
MinTemp           0.45
MaxTemp           0.23
Rainfall          0.99
Evaporation      42.79
Sunshine         47.69
WindGustDir       6.56
WindGustSpeed     6.52
WindDir9am        7.04
WindDir3pm        2.66
WindSpeed9am      0.95
WindSpeed3pm      1.85
Humidity9am       1.25
Humidity3pm       2.54
Pressure9am       9.86
Pressure3pm       9.83
Cloud9am         37.74
Cloud3pm         40.15
Temp9am           0.64
Temp3pm           1.92
RainToday         0.99
RISK_MM           0.00
RainTomorrow      0.00
dtype: float64
```



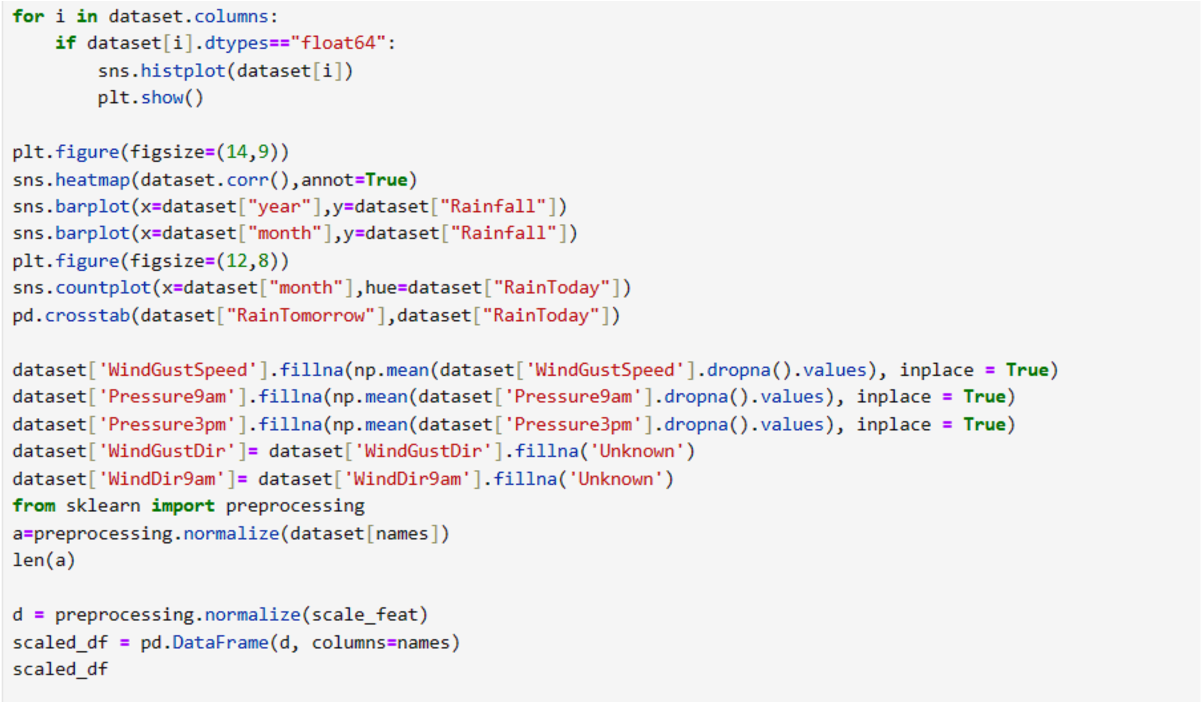
```
dataset.dtypes
for i in dataset.columns:
    print(i,"=",len(dataset[i].unique()))
dataset["RainToday"].value_counts()
```



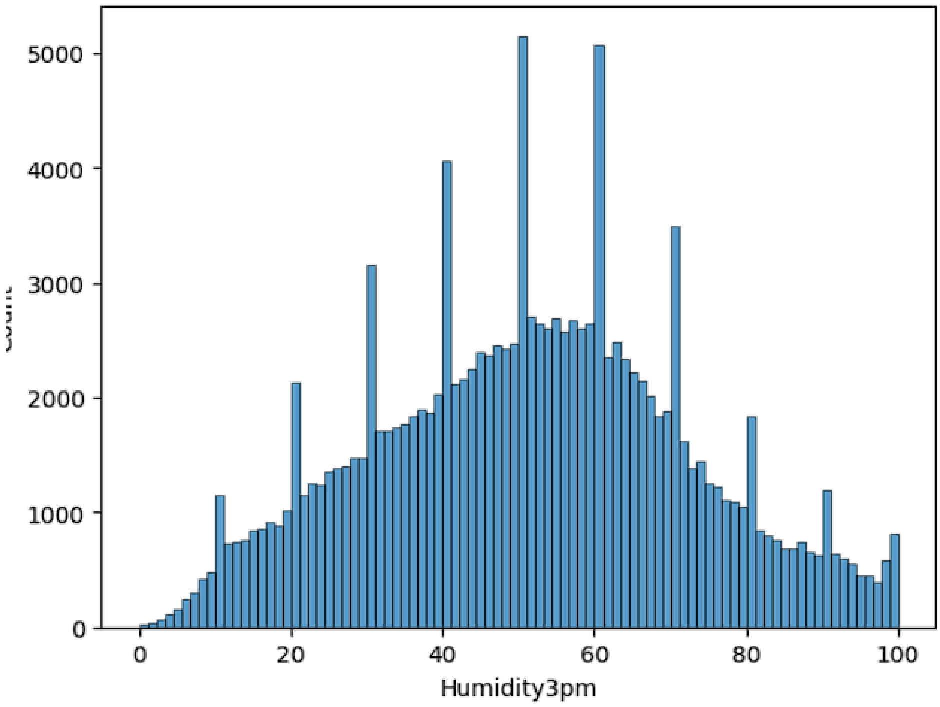
```
dataset.dtypes
for i in dataset.columns:
    print(i,"=",len(dataset[i].unique()))
dataset["RainToday"].value_counts()
```

```
Date = 3436
Location = 49
MinTemp = 390
MaxTemp = 506
Rainfall = 680
Evaporation = 357
Sunshine = 146
WindGustDir = 17
WindGustSpeed = 68
WindDir9am = 17
WindDir3pm = 17
WindSpeed9am = 44
WindSpeed3pm = 45
Humidity9am = 102
Humidity3pm = 102
Pressure9am = 547
Pressure3pm = 550
Cloud9am = 11
Cloud3pm = 11
Temp9am = 441
Temp3pm = 501
RainToday = 3
RISK_MM = 681
RainTomorrow = 2
```

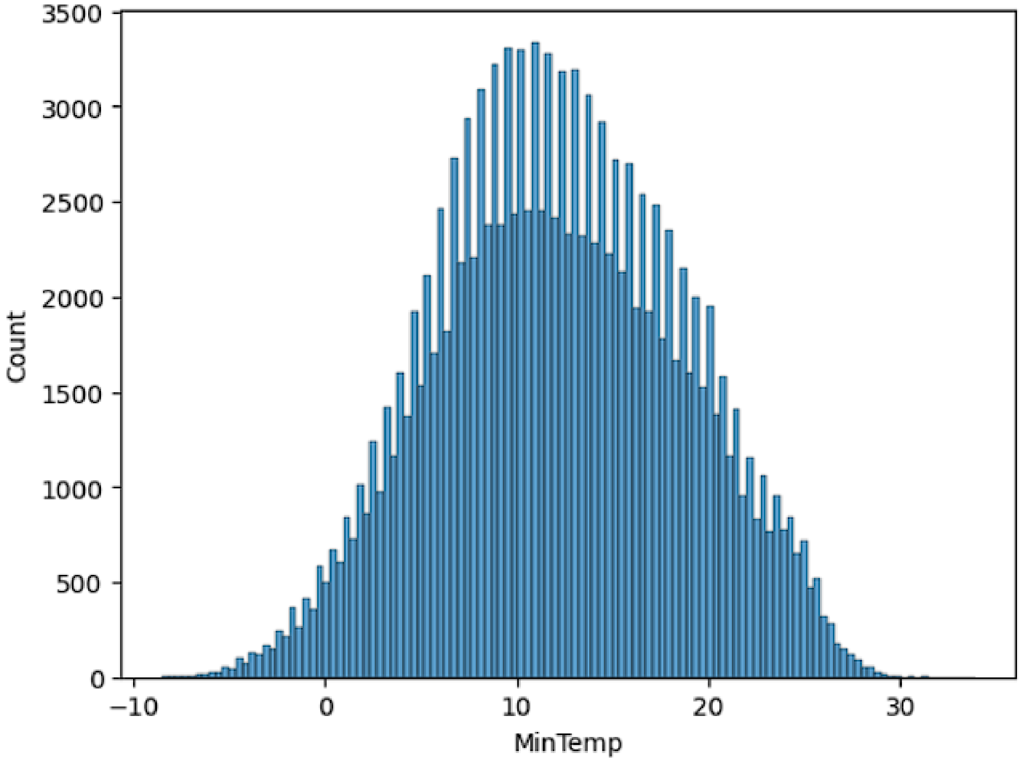
```
RainToday
No     109332
Yes     31455
Name: count, dtype: int64
```



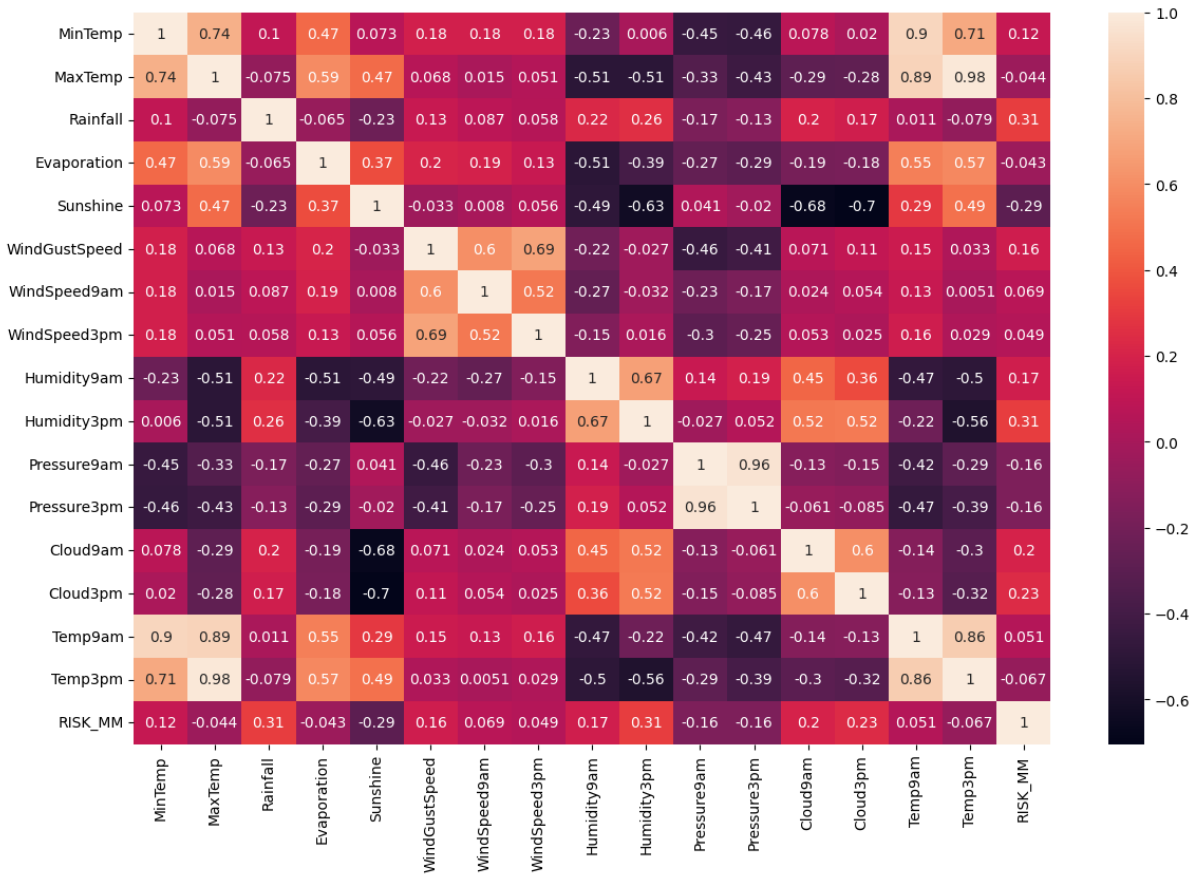
```
for i in dataset.columns:
    if dataset[i].dtypes=="float64":
        sns.histplot(dataset[i])
        plt.show()

plt.figure(figsize=(14,9))
sns.heatmap(dataset.corr(),annot=True)
sns.barplot(x=dataset["year"],y=dataset["Rainfall"])
sns.barplot(x=dataset["month"],y=dataset["Rainfall"])
plt.figure(figsize=(12,8))
sns.countplot(x=dataset["month"],hue=dataset["RainToday"])
pd.crosstab(dataset["RainTomorrow"],dataset["RainToday"])

dataset['WindGustSpeed'].fillna(np.mean(dataset['WindGustSpeed'].dropna().values), inplace = True)
dataset['Pressure9am'].fillna(np.mean(dataset['Pressure9am'].dropna().values), inplace = True)
dataset['Pressure3pm'].fillna(np.mean(dataset['Pressure3pm'].dropna().values), inplace = True)
dataset['WindGustDir']= dataset['WindGustDir'].fillna('Unknown')
dataset['WindDir9am']= dataset['WindDir9am'].fillna('Unknown')
from sklearn import preprocessing
a=preprocessing.normalize(dataset[names])
len(a)

d = preprocessing.normalize(scale_feat)
scaled_df = pd.DataFrame(d, columns=names)
scaled_df
```



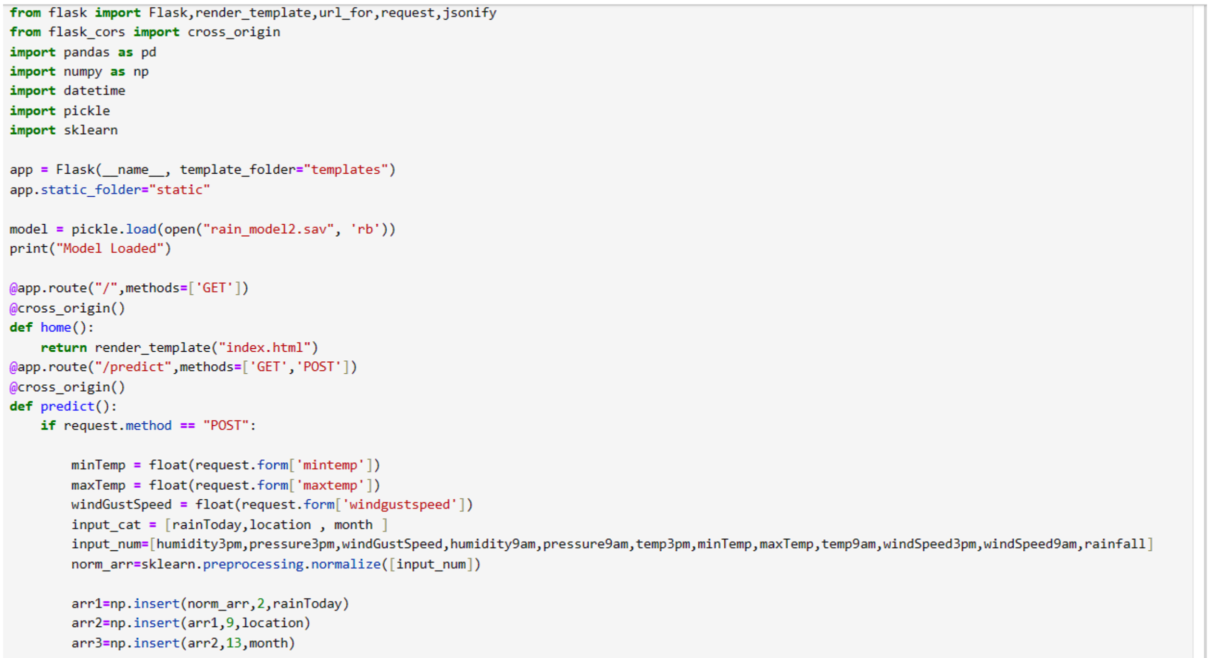
Count
0
1000
2000
3000
4000
5000
0
20
40
60
80
100
Humidity3pm



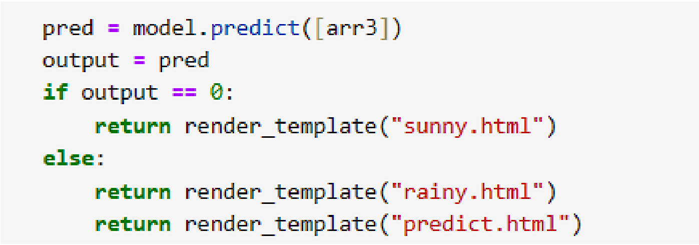
Count
0
500
1000
1500
2000
2500
3000
3500
−10
0
10
20
30
MinTemp



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

from sklearn.model_selection import train_test_split as tts
x_train,x_test,y_train,y_test=tts(x,y,test_size=0.2,random_state=100)

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix

def result(algo):
    var=algo
    var.fit(x_train,y_train)
    y_pred=var.predict(x_test)
    return accuracy_score(y_pred,y_test) , f1_score(y_pred,y_test) ,confusion_matrix(y_pred,y_test)
result(RandomForestClassifier())
result(DecisionTreeClassifier())
```

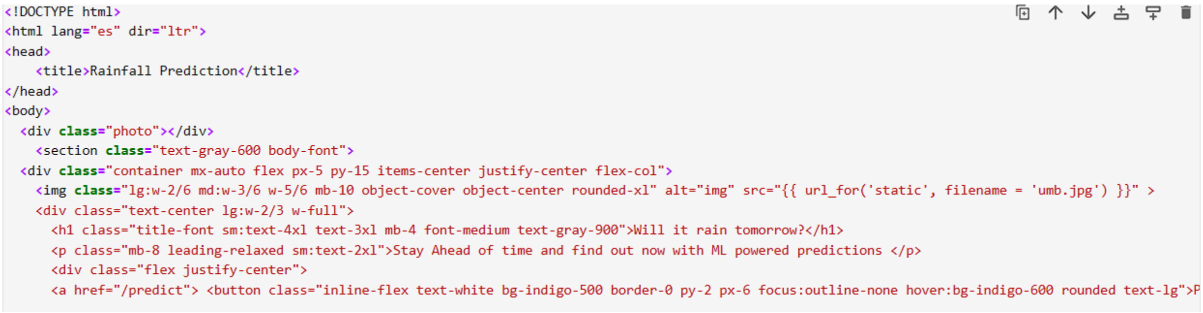


```python
from flask import Flask,render_template,url_for,request,jsonify
from flask_cors import cross_origin
import pandas as pd
import numpy as np
import datetime
import pickle
import sklearn

app = Flask(__name__, template_folder="templates")
app.static_folder="static"

model = pickle.load(open("rain_model2.sav", 'rb'))
print("Model Loaded")

@app.route("/",methods=['GET'])
@cross_origin()
def home():
    return render_template("index.html")
@app.route("/predict",methods=['GET','POST'])
@cross_origin()
def predict():
    if request.method == "POST":

        minTemp = float(request.form['mintemp'])
        maxTemp = float(request.form['maxtemp'])
        windGustSpeed = float(request.form['windgustspeed'])
        input_cat = [rainToday,location , month ]
        input_num=[humidity3pm,pressure3pm,windGustSpeed,humidity9am,pressure9am,temp3pm,minTemp,maxTemp,temp9am,windSpeed3pm,windSpeed9am,rainfall]
        norm_arr=sklearn.preprocessing.normalize([input_num])

        arr1=np.insert(norm_arr,2,rainToday)
        arr2=np.insert(arr1,9,location)
        arr3=np.insert(arr2,13,month)
```



```python
    pred = model.predict([arr3])
    output = pred
    if output == 0:
        return render_template("sunny.html")
    else:
        return render_template("rainy.html")
        return render_template("predict.html")
```

## Index.HTML

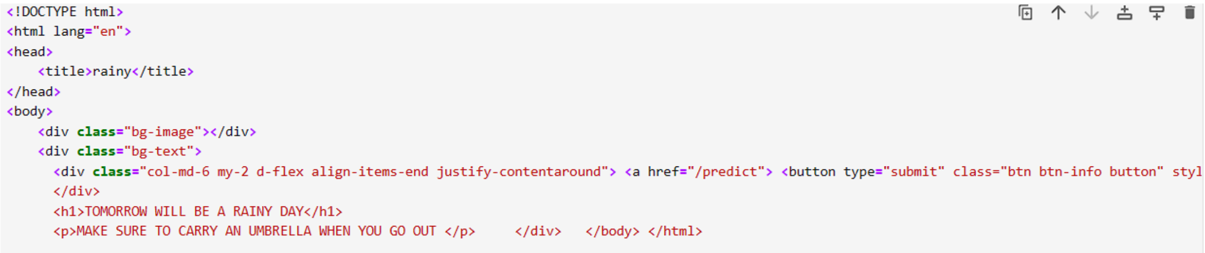


```html
<!DOCTYPE html>
<html lang="es" dir="ltr">
<head>
    <title>Rainfall Prediction</title>
</head>
<body>
  <div class="photo"></div>
    <section class="text-gray-600 body-font">
  <div class="container mx-auto flex px-5 py-15 items-center justify-center flex-col">
    <img class="lg:w-2/6 md:w-3/6 w-5/6 mb-10 object-cover object-center rounded-xl" alt="img" src="{{ url_for('static', filename = 'umb.jpg') }}" >
    <div class="text-center lg:w-2/3 w-full">
      <h1 class="title-font sm:text-4xl text-3xl mb-4 font-medium text-gray-900">Will it rain tomorrow?</h1>
      <p class="mb-8 leading-relaxed sm:text-2xl">Stay Ahead of time and find out now with ML powered predictions </p>
      <div class="flex justify-center">
      <a href="/predict"> <button class="inline-flex text-white bg-indigo-500 border-0 py-2 px-6 focus:outline-none hover:bg-indigo-600 rounded text-lg">P
```

## Prediction.Html

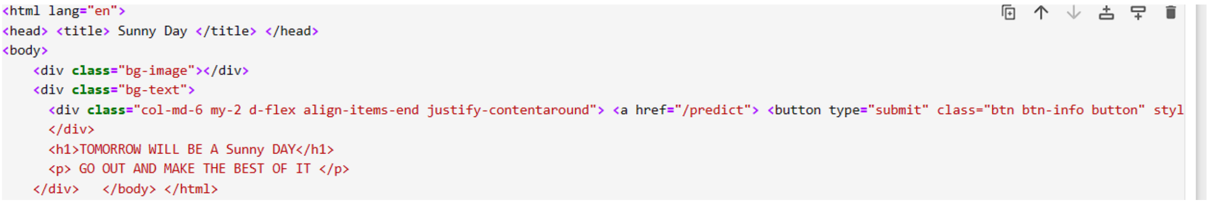
```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href=
{{url_for('static',filename='predictor.css')}}>
    <title>Rain Prediction</title>
</head>
<body>
    <section id="prediction-form">
    </br>
    <div class="col-md-6 my-2 d-flex align-items-end justify-content-around">
    <a href="./"> <button type="submit" class="btn btn-info button" style="margin-right: 100%;">Back</button></a>
    </div>
        <form class="form" action="/predict", method="POST">
            <h1 class="my-3 text-center">Predictor</h1>
            <div class="row">
                <div class="col-md-6 my-2">
                    <div class="md-form">
        <label for="mintemp" class="mintemp"> Minimum temprature (°C)</label>
        <input type="text" class="form-control" id="mintemp" name="mintemp">
</div> </div>
                <div class="col-md-6 my-2">

                    <div class="md-form">
        <label for="mintemp" class="mintemp"> Minimum temprature (°C)</label>        <
        input type="text" class="form-control" id="mintemp" name="mintemp">
</div> </div>
                <div class="col-md-6 my-2">
                <div class="md-form">
                        <label for="temp9am" class="temp9am">Temperature 9am (°C)</label>
                        <input type="text" class="form-control" id="temp9am" name="temp9am">
                    </div>
    <div class="col-md-6 my-2 d-flex align-items-end justify-content-around">
<button type="submit" class="btn btn-info button" style="margin-left:
90%;">Predict</button>
         </div> </div> </form>   </section> <div>
 <h1><center> {{ prediction }} </center></h1> </div> </body> </html>
```

Rainy Day:

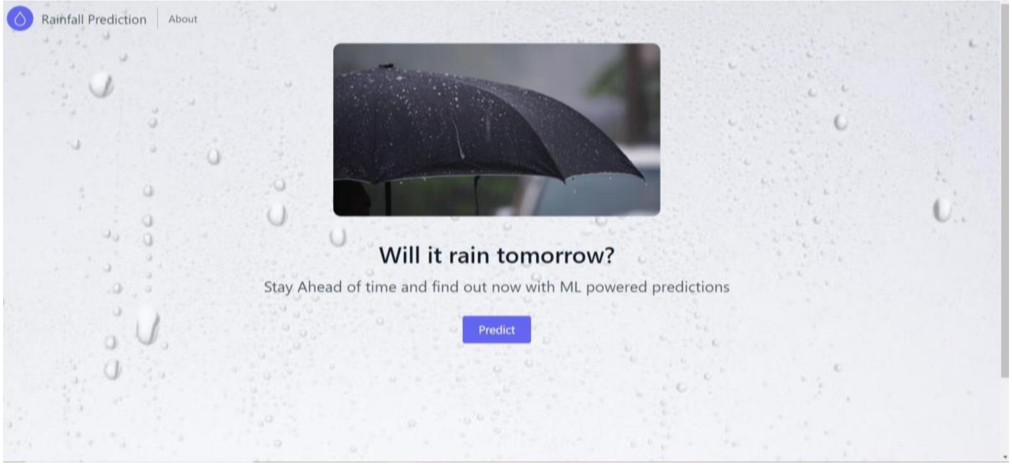


```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>rainy</title>
</head>
<body>
    <div class="bg-image"></div>
    <div class="bg-text">
      <div class="col-md-6 my-2 d-flex align-items-end justify-contentaround"> <a href="/predict"> <button type="submit" class="btn btn-info button" styl
      </div>
      <h1>TOMORROW WILL BE A RAINY DAY</h1>
      <p>MAKE SURE TO CARRY AN UMBRELLA WHEN YOU GO OUT </p>     </div>   </body> </html>
```

Sunny day

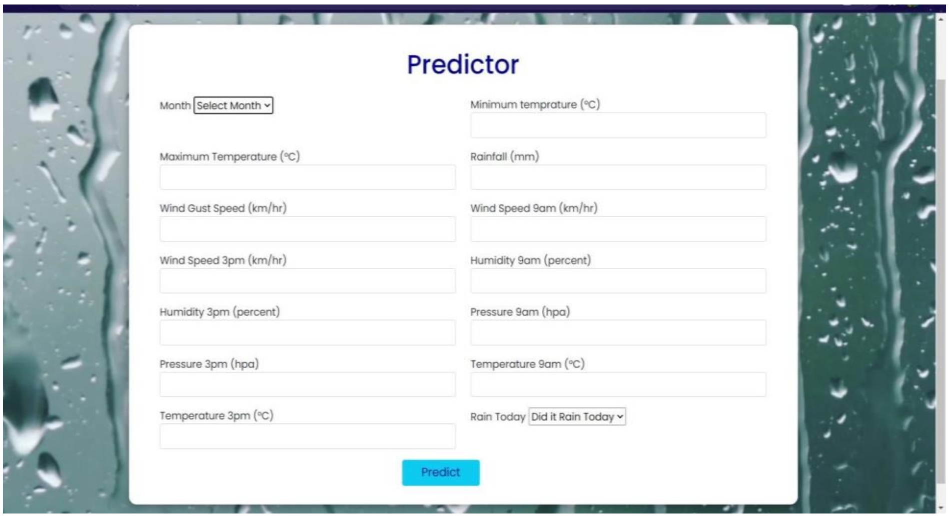


```
<html lang="en">
<head> <title> Sunny Day </title> </head>
<body>
    <div class="bg-image"></div>
    <div class="bg-text">
      <div class="col-md-6 my-2 d-flex align-items-end justify-contentaround"> <a href="/predict"> <button type="submit" class="btn btn-info button" styl
      </div>
      <h1>TOMORROW WILL BE A Sunny DAY</h1>
      <p> GO OUT AND MAKE THE BEST OF IT </p>
    </div>   </body> </html>
```



Will it rain tomorrow?

Stay Ahead of time and find out now with ML powered predictions



Predictor

Month Select Month

Minimum temprature (°C)

Maximum Temperature (°C)

Rainfall (mm)

Wind Gust Speed (km/hr)

Wind Speed 9am (km/hr)

Wind Speed 3pm (km/hr)

Humidity 9am (percent)

Humidity 3pm (percent)

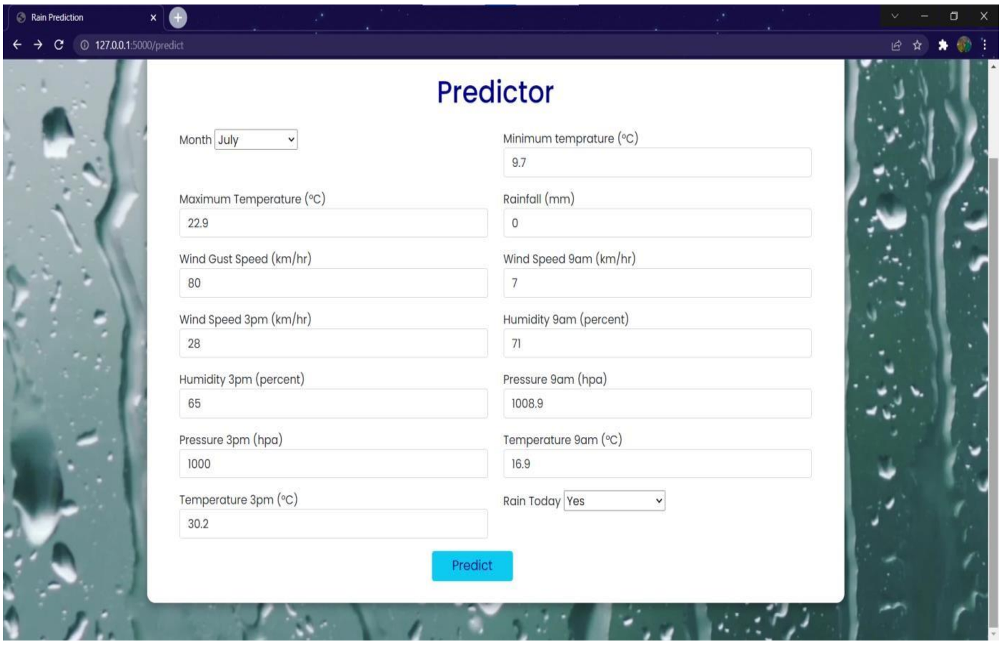
Pressure 9am (hpa)

Pressure 3pm (hpa)

Temperature 9am (°C)

Temperature 3pm (°C)

Rain Today Did it Rain Today



Predictor

Month July

Minimum temprature (°C) 9.7

Maximum Temperature (°C) 22.9

Rainfall (mm) 0

Wind Gust Speed (km/hr) 80

Wind Speed 9am (km/hr) 7

Wind Speed 3pm (km/hr) 28

Humidity 9am (percent) 71

Humidity 3pm (percent) 65

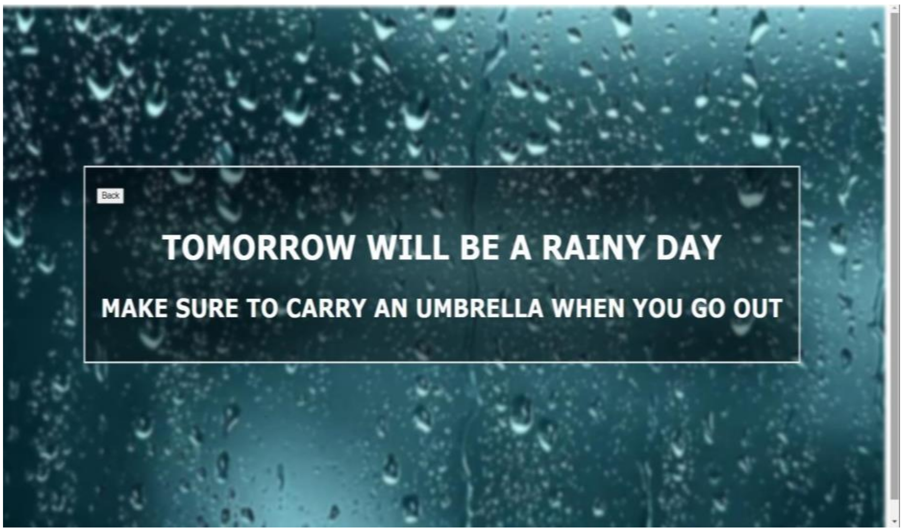
Pressure 9am (hpa) 1008.9

Pressure 3pm (hpa) 1000

Temperature 9am (°C) 16.9

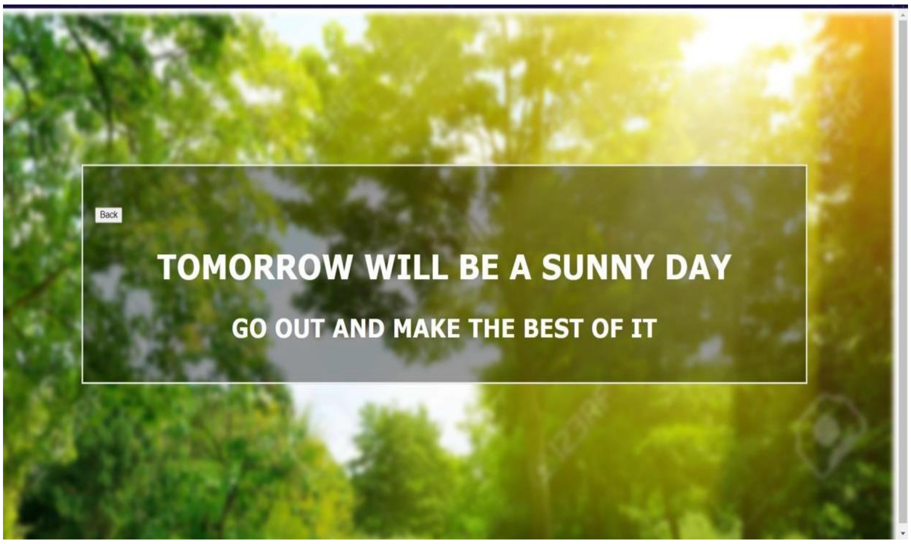
Temperature 3pm (°C) 30.2

Rain Today Yes



TOMORROW WILL BE A RAINY DAY

MAKE SURE TO CARRY AN UMBRELLA WHEN YOU GO OUT



TOMORROW WILL BE A SUNNY DAY

GO OUT AND MAKE THE BEST OF IT