






# MULTI CLASS CLASSIFICATION

CCQUATMET  
Quantitative Methods



# DATASET

Multi\_class.csv

	uuid	tave	tmin	tmax	heat_index	pr	wind_speed	rh	solar_rad	uv_rad
<b>0</b>	CATMS000000	1	1	1	1	0	0	0	0	0
<b>1</b>	CATMS000001	1	0	1	0	0	1	0	1	0
<b>2</b>	CATMS000002	1	0	0	0	0	0	0	1	0
<b>3</b>	CATMS000003	1	0	1	0	0	0	0	1	1
<b>4</b>	CATMS000004	1	0	0	0	0	1	0	1	0
<b>...</b>	...	...	...	...	...	...	...	...	...	...
<b>654</b>	CATMS00028E	1	1	0	1	0	0	0	1	1
<b>655</b>	CATMS00028F	1	0	0	1	0	0	0	1	1
<b>656</b>	CATMS000290	1	1	0	1	0	1	1	1	0
<b>657</b>	CATMS000291	1	1	0	1	0	1	0	1	1
<b>658</b>	CATMS000292	1	1	0	1	0	0	1	1	1

# CONTEXT AND PERSPECTIVE

This study focuses on understanding how environmental factors such as rainfall, temperature, and humidity influence the patterns of dengue outbreaks. The dataset includes daily measurements of average, minimum, and maximum temperature, heat index, precipitation, wind speed, relative humidity, solar radiation, and UV radiation over a series of observations. The goal is to analyze these climate drivers to identify early predictors of dengue risk, as variations in these weather variables are known to affect mosquito breeding and virus transmission. By examining these environmental patterns, the study aims to improve dengue outbreak forecasting and inform public health interventions.

# DATA UNDERSTANDING

To investigate the influence of climate drivers on dengue outbreaks, the dataset includes nine (9) key environmental attributes:

- tave (Average Temperature): Represents the daily average temperature measured in degrees Celsius.
- tmin (Minimum Temperature): The lowest temperature recorded each day in degrees Celsius.
- tmax (Maximum Temperature): The highest temperature recorded each day in degrees Celsius.
- heat\_index: A combined measure of air temperature and relative humidity, indicating how hot it feels to humans.
- pr (Precipitation): The amount of rainfall measured daily, usually in millimeters.

# DATA UNDERSTANDING

To investigate the influence of climate drivers on dengue outbreaks, the dataset includes nine (9) key environmental attributes:

- wind\_speed: The average speed of wind recorded daily, typically in meters per second or kilometers per hour.
- rh (Relative Humidity): The percentage of moisture in the air relative to the maximum it can hold at the same temperature.
- solar\_rad (Solar Radiation): The amount of solar energy received per unit area, measured daily.
- uv\_rad (UV Radiation): The intensity of ultraviolet radiation from the sun measured daily.

# DATA PREPARATION

The following steps are done to prepare the data set for association:

1. Download the data set from Kaggle and rename it based on the required filename (Multi\_class.csv).
2. Upload in the Python directory.
3. Check the dimensions and data types of the data.
4. Encode all categorical data using label encoder (if there's any).
4. Assign values for X (predictor variables) and y (target variable).
5. Split the data in train (70%) and test (30%)

# DATA PREPARATION

```
In [2]: print(data.shape) #display the dataset dimension
```

(659, 10)

```
In [3]: data.dtypes #display the data types
```

```
Out[3]:  uuid          object
         tave          int64
         tmin          int64
         tmax          int64
         heat_index     int64
         pr            int64
         wind_speed     int64
         rh            int64
         solar_rad      int64
         uv_rad         int64
         dtype: object
```

```
In [4]: data.info() #display the data structures
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 659 entries, 0 to 658
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   uuid                  659 non-null    object
 1   tave                  659 non-null    int64
 2   tmin                  659 non-null    int64
 3   tmax                  659 non-null    int64
 4   heat_index            659 non-null    int64
 5   pr                    659 non-null    int64
 6   wind_speed            659 non-null    int64
 7   rh                    659 non-null    int64
 8   solar_rad             659 non-null    int64
 9   uv_rad                659 non-null    int64
dtypes: int64(9), object(1)
memory usage: 51.6+ KB
```

```
In [5]: data.describe() #display the summarized info of dataset
```

Out[5]:

[illegible]

# MODELING

1. Get the best multi class classifier model based on the model's accuracy using accuracy score, f1, precision and recall, confusion matrix.
2. Build the model in Python.

Multi Class classification is a supervised learning algorithm that categorizes new observations into one of multiple classes. After checking on the accuracy measures using accuracy score, f1, precision, recall, and confusion matrix, and making predictions to identify which model between Naïve Bayes, Decision Tree, Logistic Regression, and Support Vector Machine (SVM) can be best used for the data set.



# MODELING

```
In [6]: X = data[['tave', 'tmin', 'tmax', 'heat_index', 'wind_speed', 'rh', 'solar_rad', 'uv_rad']]
y = data['pr']
```

```
In [8]: X #display X
```

```
Out[8]:
```

	tave	tmin	tmax	heat_index	wind_speed	rh	solar_rad	uv_rad
0	1	1	1	1	0	0	0	0
1	1	0	1	0	1	0	1	0
2	1	0	0	0	0	0	1	0
3	1	0	1	0	0	0	1	1
4	1	0	0	0	1	0	1	0
...	...	...	...	...	...	...	...	...
654	1	1	0	1	0	0	1	1
655	1	0	0	1	0	0	1	1
656	1	1	0	1	1	1	1	0
657	1	1	0	1	1	0	1	1
658	1	1	0	1	0	1	1	1

659 rows × 8 columns

```
In [9]: y #display y
```

```
Out[9]:
```

0	0
1	0
2	0
3	0
4	0
...	...
654	0
655	0
656	0
657	0
658	0

Name: pr, Length: 659, dtype: int64

# MODELING

```
In [10]: from sklearn.model_selection import train_test_split #import sci-kit learn library to split train & test data
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.30,random_state=0)
```

```
In [11]: X_train.shape, y_train.shape #display X_train and y_train shape
```

```
Out[11]: ((461, 8), (461,))
```

```
In [13]: X_test.shape, y_test.shape #display X_test and y_test shape
```

```
Out[13]: ((198, 8), (198,))
```

```
#display the accuracy results
results = pd.DataFrame({
    'Model': ['Gradient Boosting','Decision Tree', 'K Nearest Neighbors',
             'Random Forest', 'Naive Bayes','Support Vector Machine'],
    'Score': [acc_gb, acc_dt, acc_knn,
             acc_rf, acc_gnb, acc_svm]})
result_df = results.sort_values(by='Score', ascending=False)
result_df = result_df.set_index('Score')
result_df.head(6)
```

Model	
Score	
76.26	Gradient Boosting
76.26	Decision Tree
76.26	Random Forest
76.26	Support Vector Machine
64.65	K Nearest Neighbors
50.51	Naive Bayes

# MODELING

```
In [29]: y_pred_gb #display prediction
```

```
Out[29]: array([1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0,
                0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0,
                0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
                0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
                0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1,
                0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
                1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
                0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0],
                dtype=int64)
```

```
In [31]: #import libraries for root-mean-square error
import numpy as np
from sklearn import metrics
from sklearn.metrics import mean_squared_error
import math

#display the root-mean-square error
MSE=np.square(np.subtract(y_test,y_pred_gb)).mean()
RMSE = (math.sqrt(MSE) * 100)
print(RMSE)
```

48.72101572973796

```
In [30]: y_test #display actual values
```

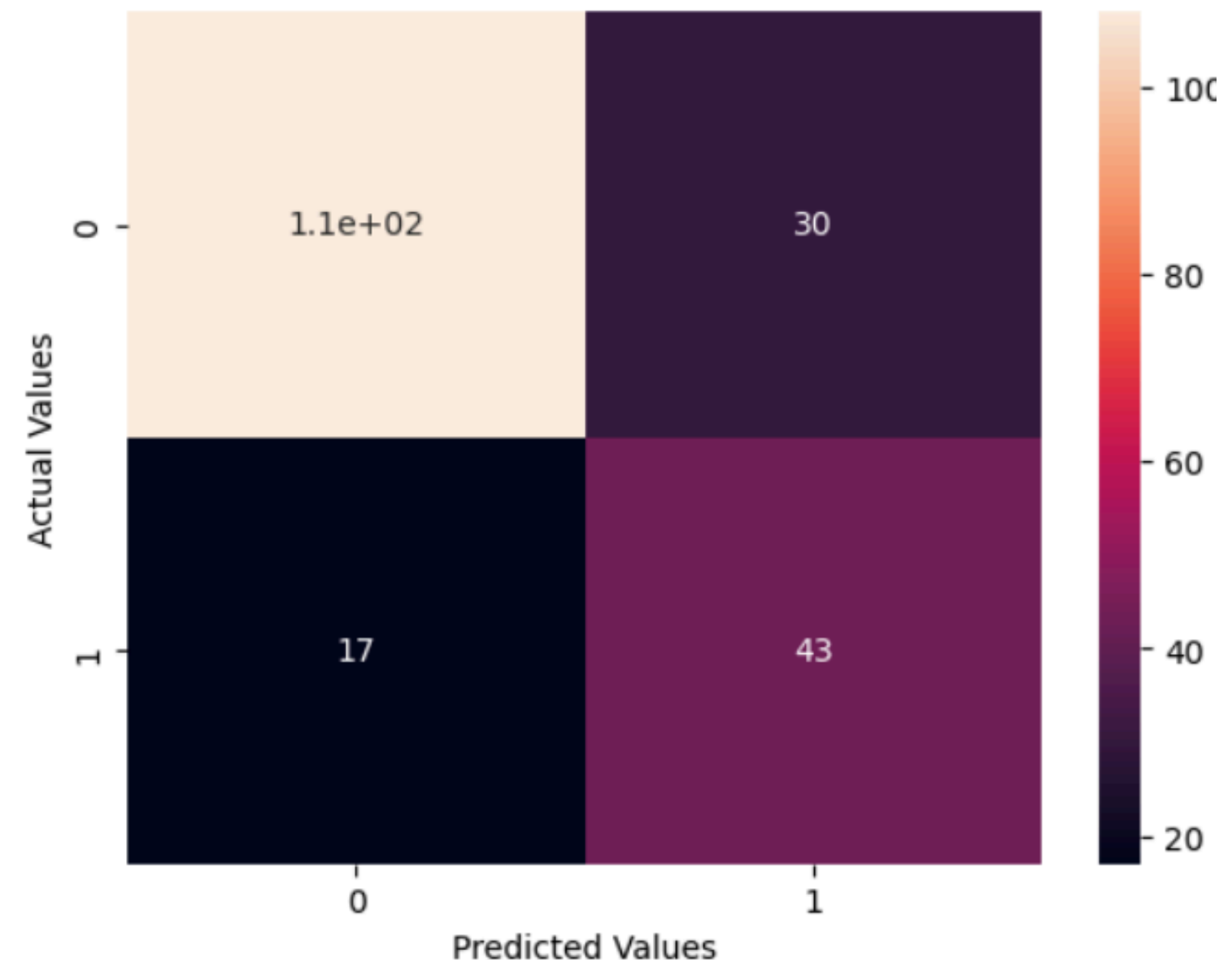
```
Out[30]: 540    1
          103    0
          14    0
          525    0
          385    0
          ..
          132    0
          173    1
          178    0
          609    1
          316    0
```

# EVALUATION

Based on the generated one-vs-rest confusion matrix, we were able to identify the true positive, true negative, false positive, and false negative with the following values:

True Positive: The label belongs to the class, and it is correctly predicted. Two (4) Innovators belong to the class, and it is correctly predicted.

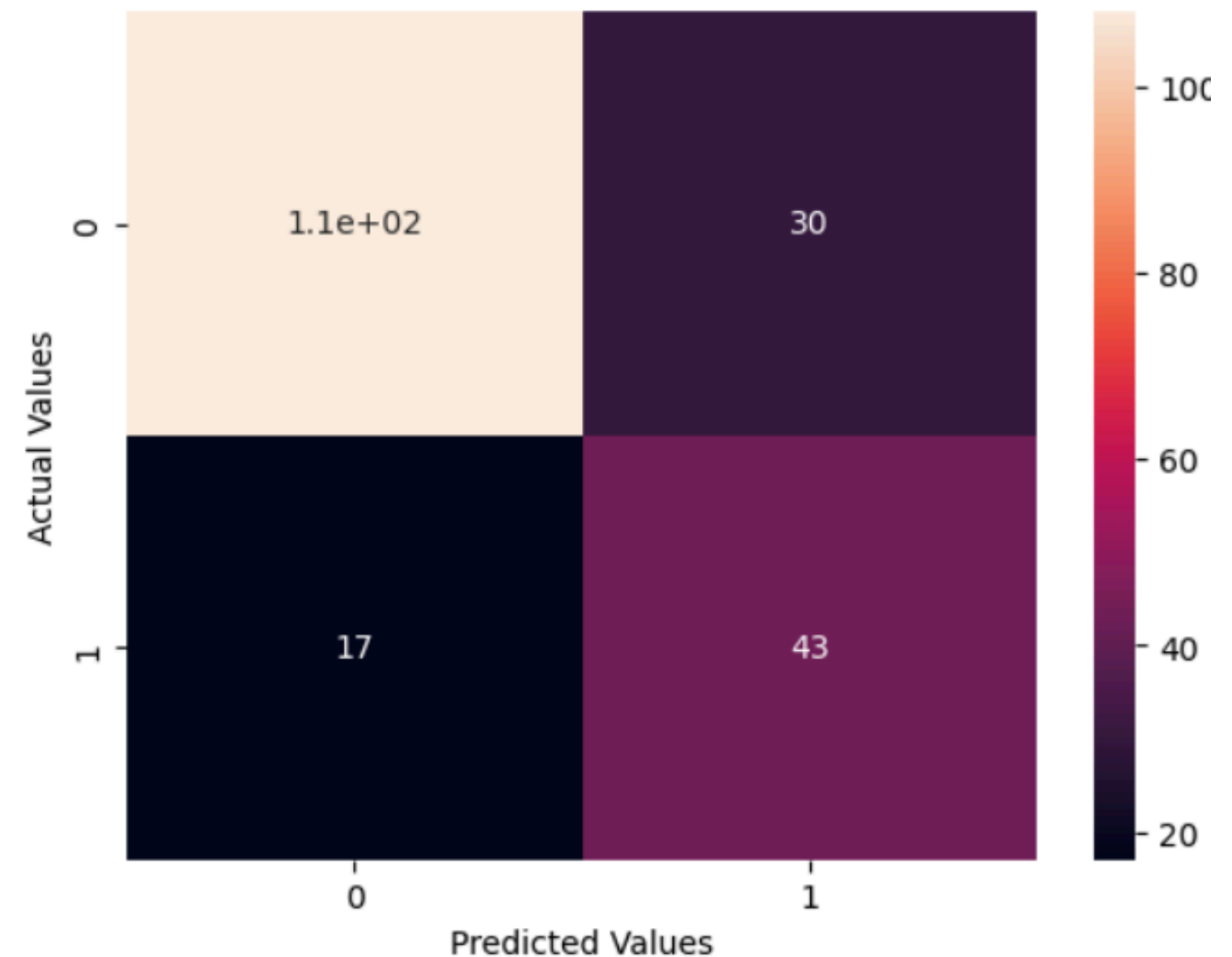
False Positive: The label does not belong to the class, but the classifier is predicted as positive. Four (4) values (50, 60, 80, 100) do not belong to the class, but they are incorrectly predicted as positive.



# EVALUATION

Based on the generated one-vs-rest confusion matrix, we were able to identify the true positive, true negative, false positive, and false negative with the following values:

True Negative: The label does not belong to the class and is correctly rejected



# EVALUATION

Based on the generated one-vs-rest confusion matrix, we were able to identify the true positive, true negative, false positive, and false negative with the following values:

False Negative: The label does belong to the class, but it is predicted as negative. Two (2) values (17, 20) belong to the class, but they are missed in the predictions.

