CAPSTONE PROJECT - 1
Tiny ML

Computer Hardware Software Workshop
COCSC19

Submitted by:-
Kanishk Kumar (2021UCS1627)
Shubham Yadav (2021UCS1630)
Kashish Goel (2021UCS1633)
CSE Section - 2
3rd Year

Title:-

Rainfall Prediction using Linear Regression

Objective:-

Utilizing TinyMl library,develop a project on a single board computer or microcontroller.

Idea:-

Training a machine learning model such that it will intelligently predict the amount of rainfall based on three variables, i.e temperature, humidity and surface level pressure readings.

Apparatus Required:-

We have implemented this project on an Online simulator (Wokwi) and the equipments used on Wokwi includes:-

- Arduino UNO board
- 2X16 LCD Screen
- DHT22 Sensor(Temperature and Humidity Sensor)
- Potentiometer(to Simulate Pressure)
- Connecting wires

Theory:-

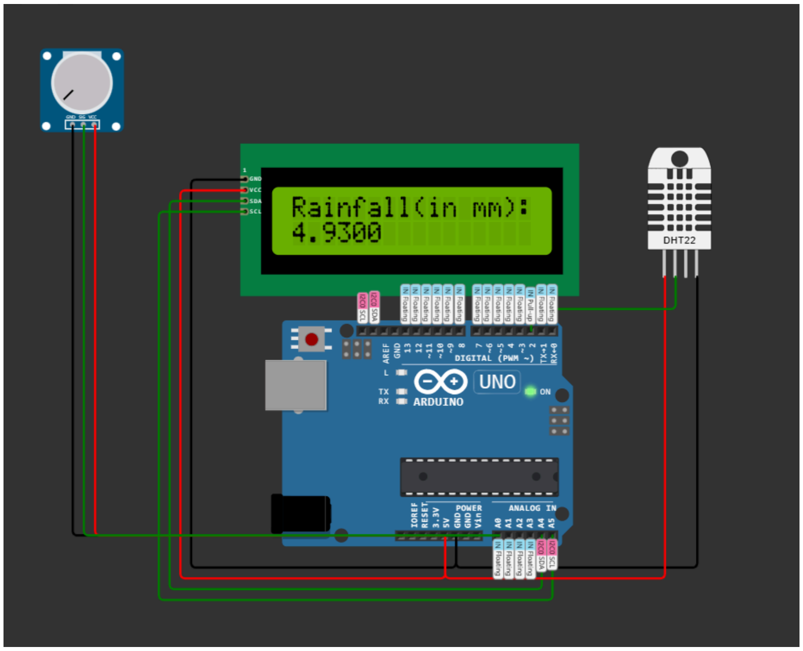
Rainfall Prediction is the application of science and technology to predict the amount of rainfall over a region. It is important to exactly determine the rainfall for the effective use of water resources, crop productivity and pre-planning of water structures.

In our project, we will use Linear Regression to predict the amount of rainfall.
Linear regression is a supervised learning algorithm that is used to model the relationship between a dependent variable and one or more independent variables. In this case, the dependent variable is the amount of rainfall, and the independent variables are the features that are used to predict it, such as temperature, humidity, pressure, etc.
The first step is to collect the historical data, which includes the amount of rainfall and the corresponding values of the independent variables. Once the data has been collected, it needs to be cleaned and preprocessed to remove any outliers or missing values. Next, the data is split into two sets: the training set and the testing set. The training set is used to train the model, while the testing set is used to evaluate its performance. To perform linear regression, we need to first define a hypothesis function

that maps the input variables to the output variable. In this case, the hypothesis function is a linear equation of the form:
y = b + b1x1 + b2x2 + … + bnxn
where
y is the predicted amount of rainfall,
x1, x2, …, xn are the input variables,
and b0, b1, b2, …, bn are the coefficients that are learned during training.

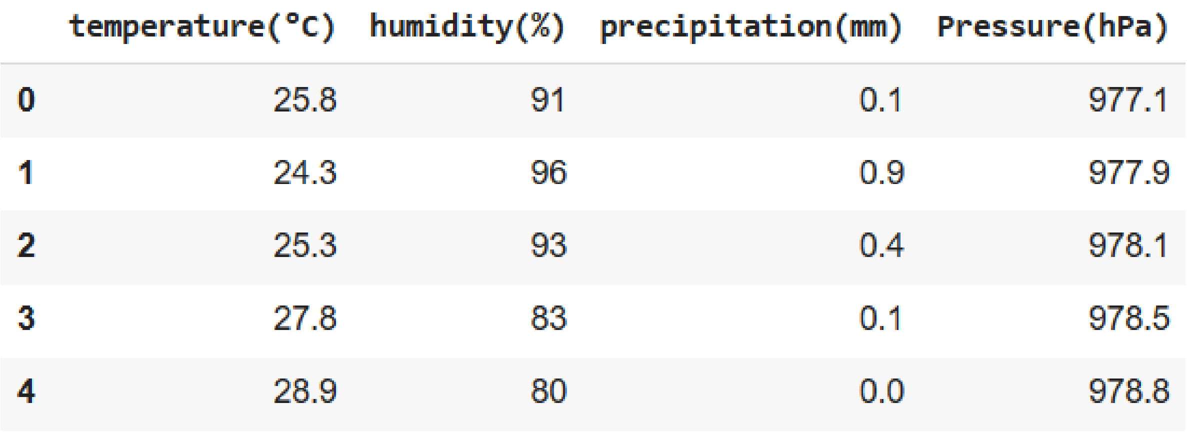
To train the model, we need to find the values of the coefficients that minimize the difference between the predicted values and the actual values in the training set. This is done by minimizing the mean squared error (MSE) using gradient descent or some other optimization algorithm. Once the model has been trained, it can be used to predict the amount of rainfall for new input values.
Inputs for new values will be taken from the respective sensors used in the project.

Circuit:-



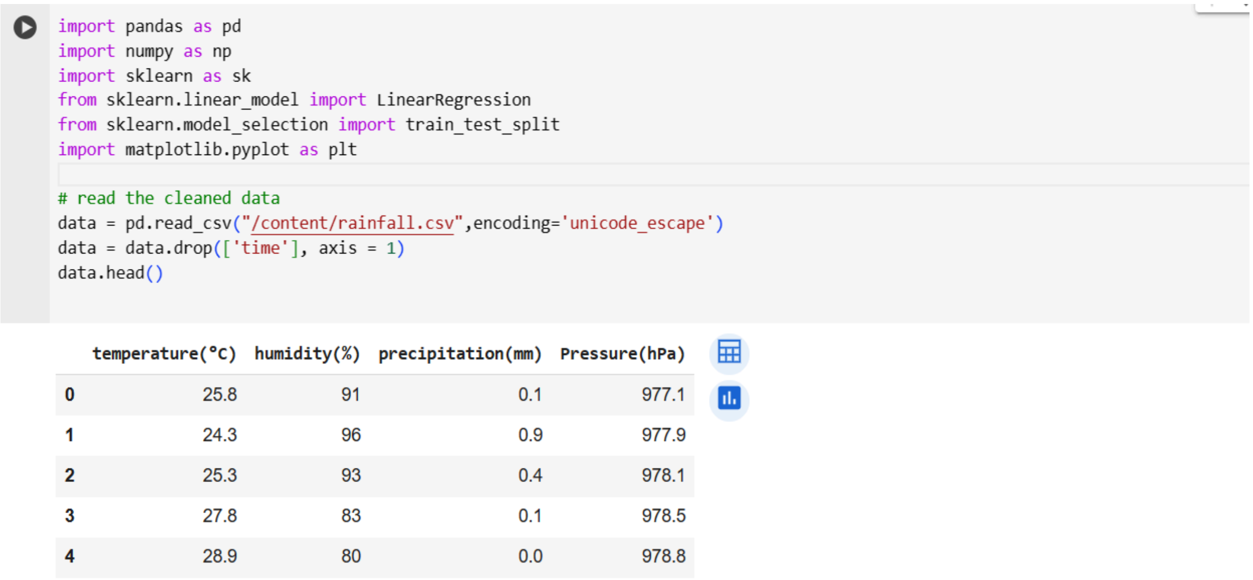
Dataset:-

The dataset comprises of hourly amount of precipitation in New Delhi at different average temperature,humidity and surface level pressure.



	temperature(°C)	humidity(%)	precipitation(mm)	Pressure(hPa)
0	25.8	91	0.1	977.1
1	24.3	96	0.9	977.9
2	25.3	93	0.4	978.1
3	27.8	83	0.1	978.5
4	28.9	80	0.0	978.8

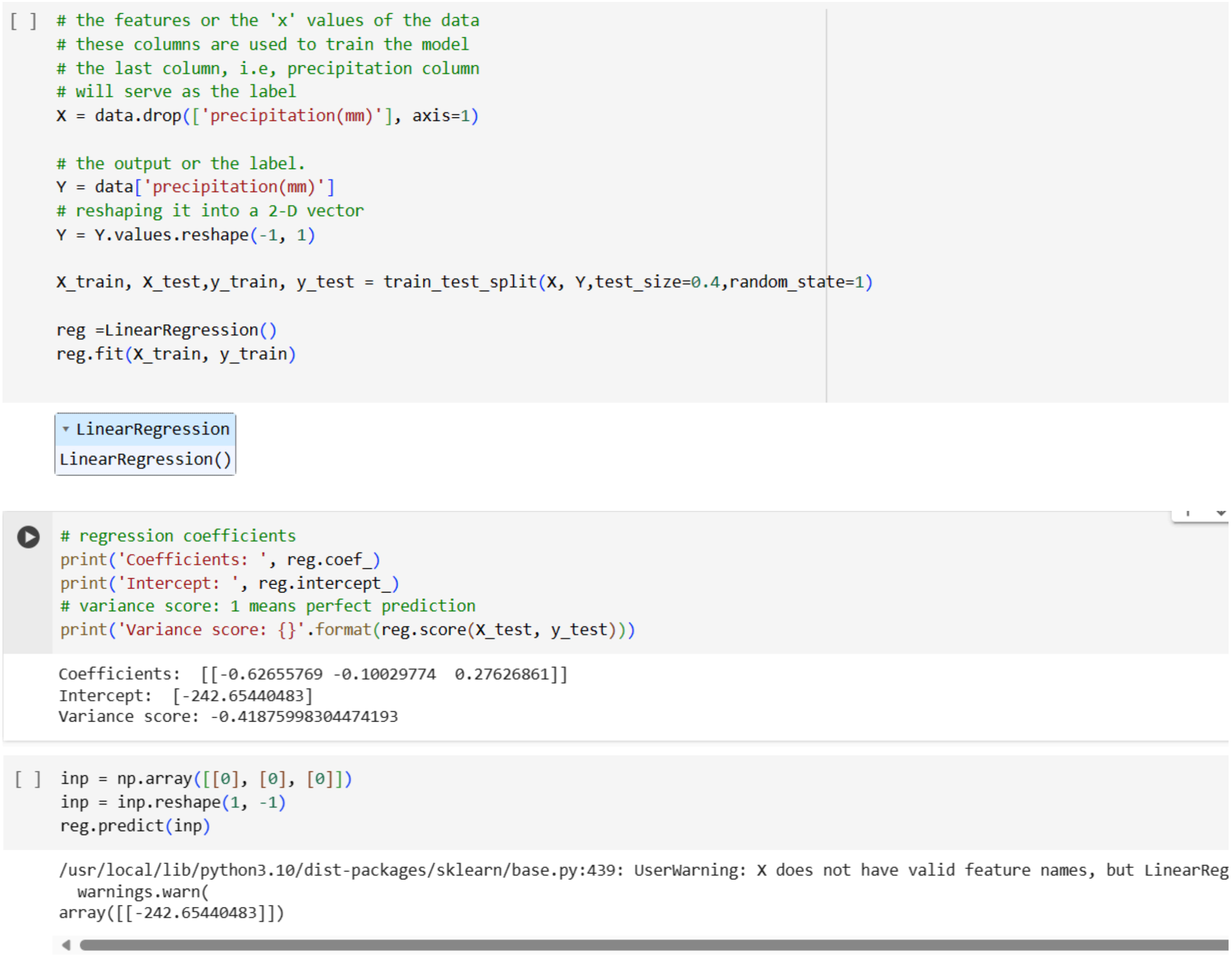
Python Code:-



```
import pandas as pd
import numpy as np
import sklearn as sk
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

# read the cleaned data
data = pd.read_csv("/content/rainfall.csv",encoding='unicode_escape')
data = data.drop(['time'], axis = 1)
data.head()
```

	temperature(°C)	humidity(%)	precipitation(mm)	Pressure(hPa)
0	25.8	91	0.1	977.1
1	24.3	96	0.9	977.9
2	25.3	93	0.4	978.1
3	27.8	83	0.1	978.5
4	28.9	80	0.0	978.8



```
[ ] # the features or the 'x' values of the data
    # these columns are used to train the model
    # the last column, i.e, precipitation column
    # will serve as the label
    X = data.drop(['precipitation(mm)'], axis=1)

    # the output or the label.
    Y = data['precipitation(mm)']
    # reshaping it into a 2-D vector
    Y = Y.values.reshape(-1, 1)

    X_train, X_test,y_train, y_test = train_test_split(X, Y,test_size=0.4,random_state=1)

    reg =LinearRegression()
    reg.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
# regression coefficients
print('Coefficients: ', reg.coef_)
print('Intercept: ', reg.intercept_)
# variance score: 1 means perfect prediction
print('Variance score: {}'.format(reg.score(X_test, y_test)))
```

```
Coefficients:  [[-0.62655769 -0.10029774  0.27626861]]
Intercept:  [-242.65440483]
Variance score: -0.41875998304474193
```

```
[ ] inp = np.array([[0], [0], [0]])
    inp = inp.reshape(1, -1)
    reg.predict(inp)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearReg
  warnings.warn(
array([[-242.65440483]])
```

Arduino Code:-

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT22
float a = 0.62655769;
float b = 0.10029774;
float c = 0.27626861;
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
int pressurePin = A0;
float pressureValue;

void setup() {
  lcd.init();
  lcd.backlight();
  dht.begin();
  lcd.setCursor(0, 0);
  lcd.print("Weather");
   lcd.setCursor(0, 1);
  lcd.print(" Monitoring ");
  delay(2000);
  lcd.clear();
}

void loop() {
  delay(2000);
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  pressureValue = analogRead(pressurePin);
  float pressure = map(pressureValue, 0, 1023, 970, 980);
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temperature);
  lcd.print("C");
  delay(2000);
  lcd.setCursor(0, 0);
  lcd.print("Humidity: ");
  lcd.print(humidity);
  lcd.print("%");
  delay(2000);
  lcd.setCursor(0, 0);
  lcd.print("Pressure(in hPa): ");
  lcd.setCursor(0, 1);
  lcd.print(pressure);
  delay(2000);
  lcd.setCursor(0, 0);
  float rainfall = -a*temperature - b*humidity + c*pressure - 242.65;
  lcd.print("Rainfall(in mm): ");
  lcd.setCursor(0, 1);
  lcd.print(rainfall);
  delay(3000);
  lcd.clear();
}
```

Result:-

We have successfully implemented the Rainfall Prediction Model using Linear Regression. It is important to exactly determine the rainfall for the effective use of water resources, crop productivity and pre-planning of water structures.

Video Demonstration

Online Simulation