Mini Project Report

Entitled

Weather Prediction Using Machine Learning

Submitted to the Department of Electronics Engineering in Partial Fulfilment for the Requirements for the Degree of

Bachelor of Technology
(Electronics and Communication)

: Presented & Submitted By :

Kavish Gajjar, Rahul Makwana, Meet Kataria

Roll No. (U21EC120, U21EC128, U21EC145)
B. TECH. VI (EC), 6th Semester

: Guided By :

Dr. Kishor Upla
Assistant Professor, SVNIT

(Year: 2023-24)

DEPARTMENT OF ELECTRONICS ENGINEERING

SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY

Surat-395007, Gujarat, INDIA.

Sardar Vallabhbhai National Institute Of Technology

Surat - 395 007, Gujarat, India

DEPARTMENT OF ELECTRONICS ENGINEERING

CERTIFICATE

This is to certify that the Mini-Project Report entitled “**Weather Prediction Using Machine Learning**” is presented & submitted by Kavish Gajjar, Rahul Makwana, Meet Kataria, bearing Roll No. U21EC120, U21EC128, U21EC145, of B.Tech. VI, 6th Semester in the partial fulfillment of the requirement for the award of B.Tech. Degree in Electronics & Communication Engineering for academic year 2023-24.

They have successfully and satisfactorily completed their **Mini-Project** in all respects. We, certify that the work is comprehensive, complete and fit for evaluation.

Dr. Kishor Upla
Assistant Professor & Project Guide

Abstract

This project presents a comprehensive approach to Weather Prediction by harnessing the power of Machine Learning, specifically Linear Regression, in conjunction with Arduino Nano board technology. The system is designed to provide accurate forecasts of rain or other weather conditions based on real-time data collected from sensors measuring parameters such as humidity and temperature. Leveraging the capabilities of I2C communication, the Arduino Nano board communicates with a laptop where the collected data is used to train the regression algorithm. Through iterative analysis and model refinement, the algorithm generates predictions regarding the likelihood of rain or specific weather outcomes. The results are displayed in real-time on the screen, offering users immediate access to actionable weather forecasts. By integrating machine learning with IoT technology, this project aims to offer a scalable and adaptable solution for localized weather prediction, with potential applications ranging from agriculture to urban planning and beyond. Additionally, the system's modular design allows for easy expansion and customization, facilitating future enhancements and optimizations as needed.

Table of Contents

List of Figures

List of Abbreviations

IoT	Internet of Things
LCD	Liquid Crystal Display
I2C	Inter-Integrated Circuit
ADC	Analog to Digital Conversion
LDR	Light Dependent Resistor

Chapter 1
Linear Regression

Linear Regression is a foundational statistical method used for modeling the relationship between a dependent variable and one or more independent variables. It serves as a fundamental tool in the realms of both statistics and machine learning, owing to its simplicity, interpretability, and wide applicability. By fitting a linear equation to observed data, Linear Regression provides insights into the nature of the relationship between variables and enables prediction and inference.

1.1 Understanding Linear Regression

Linear Regression seeks to establish a linear relationship between the independent variables (features) and the dependent variable (target). [1] Mathematically, it can be represented as:

$$y = Bo + B1 \times x1 + B2 \times x2 + B3 \times x3 + ... + Bn \times xn + E \tag{1.1}$$

- y is the dependent variable.
- x1,x2,....,xn are the independent variables.
- Bo, B1, B2,....,Bn are the coefficients (parameters) of the model.
- E is the error term

1.1.1 Example

Consider a scenario where we want to predict the performance of students (y) based on two independent variables: hours of study (x1) and attendance(x2). The linear regression equation for this scenario would be:

$$performance = Bo + B1 \times hours\_of\_study + B2 \times attendance + E \tag{1.2}$$

1.2 Advantages of Linear Regression

- **Simplicity**: Linear Regression is intuitive and easy to understand, making it accessible to individuals with varying levels of statistical background.

- **Interpretability**: The coefficients in the linear equation provide clear insights into the impact of each independent variable on the dependent variable.
- **Computational Efficiency**: Training and inference with linear regression models are computationally efficient, particularly with large datasets.
- **Flexibility**: Linear Regression can be extended to incorporate regularization techniques such as Ridge Regression and Lasso Regression, enhancing its robustness and generalization performance.

1.3 Disadvantages of Linear Regression

- **Assumption of Linearity**: Linear Regression assumes a linear relationship between variables, which may not always hold true in complex real-world scenarios.
- **Sensitivity to Outliers**: Linear Regression can be sensitive to outliers, affecting the model's performance and predictive accuracy.
- **Limited Complexity**: Linear Regression models have limited capacity to capture complex nonlinear relationships between variables, potentially leading to underfitting of the data.
- **Multicollinearity**: When independent variables are highly correlated, it can lead to issues of multicollinearity, where the coefficients become unstable and difficult to interpret. [2]

1.3.1 Example

In a dataset where the relationship between the independent and dependent variables is nonlinear, such as predicting the price of a house based on its age and square footage, a linear regression model may struggle to accurately capture the underlying patterns, leading to suboptimal predictions.

Linear Regression remains a versatile and widely-used technique for modeling relationships between variables. While it offers simplicity, interpretability, and computational efficiency, it also has limitations, particularly in handling nonlinear relationships, outliers, and multicollinearity.

Chapter 2
Hardware Implementation

Weather prediction plays a crucial role in various aspects of human life, from agriculture to transportation and disaster management. Machine learning techniques, particularly Linear Regression, offer promising avenues for accurate weather forecasting. This project aims to develop a weather prediction system using Linear Regression, implemented on a laptop, and integrated with an Arduino Nano board for real-time data collection and display. The system will utilize sensors such as humidity and temperature sensors to gather environmental data, which will be processed by the regression algorithm to predict weather conditions. The results will be communicated back to the Arduino Nano board and displayed on an LCD screen, providing users with accessible and timely weather forecasts.

2.1 Implementation on Arduino Nano Board

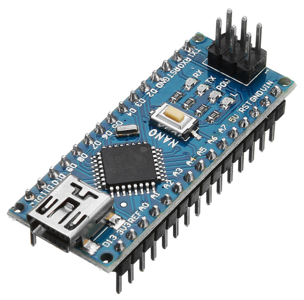


Figure 2.1: Arduino Nano

The Arduino Nano board acts as the intermediary between the physical environment and computational processing, facilitating data collection from sensors and displaying prediction results on an LCD screen. The hardware setup includes a humidity sensor, measuring moisture content with analog voltage output, and a temperature sensor, enhancing prediction accuracy. Communication with the laptop occurs via I2C serial communication, allowing seamless data transfer. The LCD screen provides a user-friendly interface for viewing weather forecasts, controlled by the Arduino Nano board. [3] [4]

2.2 Working of Sensors

2.2.1 Humidity Sensor

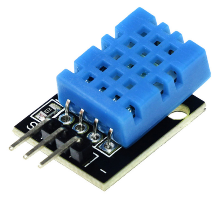


Figure 2.2: Humidity Sensor

The humidity sensor operates based on changes in electrical conductivity or capacitance in response to variations in moisture levels. It typically consists of a moisture-sensitive material such as a polymer, whose electrical properties change with humidity. As the humidity increases, the resistance or capacitance of the sensor changes, resulting in a corresponding change in the output voltage. This voltage is then read by the Arduino Nano board using its analog-to-digital converter (ADC) and converted into a numerical value representing the relative humidity. [5]

2.2.2 LDR Sensor (Light Dependent Resistor)

Figure 2.3: LDR Sensor

An LDR sensor can be integrated for additional environmental sensing. The LDR sensor's resistance varies inversely with the intensity of light falling on it. This property makes it suitable for detecting changes in ambient light conditions, which can be use-

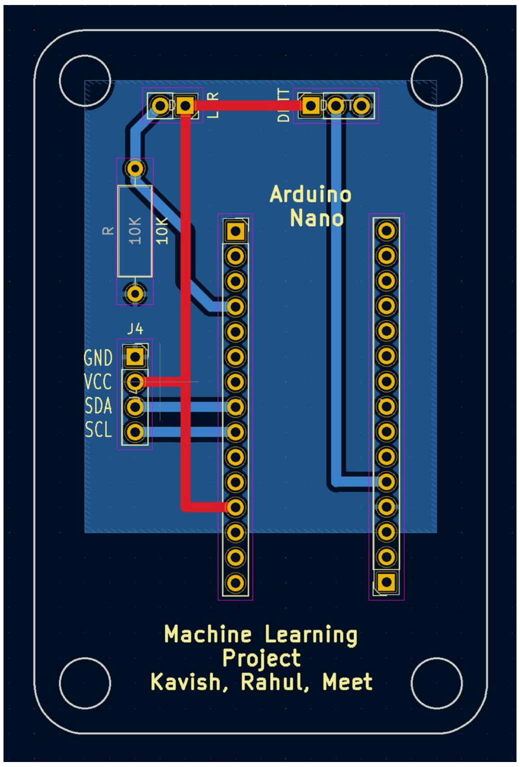
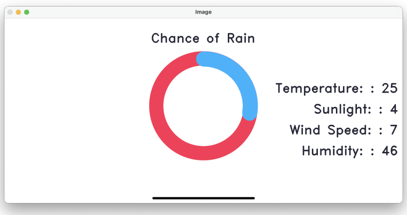


Figure 2.5: PCB Design for Weather Prediction

ful for supplementary weather data or for determining daytime and nighttime conditions. [5]

2.3 Displaying Results on Screen



Chance of Rain

Temperature: : 25
Sunlight: : 4
Wind Speed: : 7
Humidity: : 46

Figure 2.4: Screen Display

The measured values of Humidity, Temperature, Light Intensity and Wind speed are communicated to the computer and the results are depicted on the computer screen. [6]

Chapter 3
Code

3.1 Arduino Code

```
#include <SimpleDHT.h>
int pinDHT11 = 2;
SimpleDHT11 dht11(pinDHT11);
const int ldr_pin = A6;
const int wind_pin = A2;
void setup() {
  Serial.begin(115200);
  pinMode(ldr_pin, INPUT);
  pinMode(wind_pin, INPUT);
}
void loop(){
  int ldr = analogRead(ldr_pin);
  int wind = analogRead(wind_pin);
  int ldr_map = map(ldr, 100, 1000, 0, 10);
  int wind_map = map(wind, 0, 1023, 0, 57);
  byte temperature = 0;
  byte humidity = 0;
  if(ldr_map<10){
    Serial.print("0");
  }
  Serial.print(ldr_map);
  Serial.print((int)temperature);
  Serial.print((int)humidity);
  if(wind_map<10){
    Serial.print("0");
  }
  Serial.print(wind_map);
  delay(1500);
}
```

3.2 Python Script

```
import serial
import time
import cv2
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder


data = pd.read_csv(r"/Users/kavishgajjar/Downloads/Weather_Data
(1).csv")

data = data.drop(['Date', 'Temp9am', 'MinTemp', 'MaxTemp',
'Evaporation','WindGustSpeed', 'WindGustDir', 'WindDir9am', 'WindDir3pm',
'WindSpeed9am', 'Humidity9am', 'Pressure9am', 'Pressure3pm', 'Cloud9am']
, axis=1)

lb = LabelEncoder()
data['RainToday'] = lb.fit_transform(data['RainToday'])
x = data.iloc[:, [0,3,4,5]].values;
y = data.iloc[:, [1,2]];
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
0.25, random_state = 0)

regression = LinearRegression()
regression.fit(x_train, y_train)

ser = serial.Serial('/dev/tty.usbserial-110', 115200, timeout=1)
time.sleep(2)
while True:
    line = ser.readline().decode("utf-8")
    ldr = int(line[0:2])
    temp = int(line[2:4])
    hum = int(line[4:6])
    wind = int(line[6:8])
    img = cv2.imread(r"/Users/kavishgajjar/Downloads/swiftui-circular
    -progress-bar-progress-indicator.png")
    y_pred = regression.predict([[temp, ldr, wind, hum]])
    rain_prob = y_pred[0,1]
    font = cv2.FONT_HERSHEY_DUPLEX
    font_scale = 2
    text_color = (56, 40, 39)  # BGR color (Blue, Green, Red)
    thickness = 4
    font2 = cv2.FONT_HERSHEY_DUPLEX
    font_scale2 = 2
    text_color2 = (56, 40, 39)  # BGR color (Blue, Green, Red)
    thickness2 = 3
    text2 = f"Chance of Rain"
    text_size2, _ = cv2.getTextSize(text2, font2, font_scale2,
    thickness2)
    text_x = (img.shape[1] - text_size2[0]) // 2
    text_y = (img.shape[0] + text_size2[1] - 700) // 2
    cv2.putText(img, text2, (text_x, text_y), font2, font_scale2,
    text_color2, thickness2)
    variables_dict = {
        'Temperature: ': temp,
        'Sunlight: ': ldr,
        'Wind Speed: ': wind,
        'Humidity: ': hum
    }
    for j, (label, value) in enumerate(variables_dict.items()):
        # Construct text string
        text = f"{label}: {value}"
        text_size, _ = cv2.getTextSize(text, font, font_scale,
        thickness)
        text_x = (img.shape[1] - text_size[0]) - 20
        text_y = (img.shape[0] - 4*text_size[1] + j*200) // 2
        cv2.putText(img, text, (text_x, text_y), font,
        font_scale, text_color, thickness)
    val = np.interp(float(rain_prob),[0, 1], [-90, 270])
    cv2.ellipse(img, (960, 420), (225, 225), 0, -90, val,
    (255, 180, 0), 72)
    cv2.imshow("Image", img)
    cv2.waitKey(1)
```

Chapter 4
Future Enhancements

4.1 Potential Improvements and Extensions to the Project

To enhance scalability and adaptability, implement a scalable architecture capable of handling larger datasets and accommodating more complex models, possibly exploring cloud-based solutions for distributed computing. For user interface and visualization, prioritize the development of a user-friendly interface for configuring settings and visualizing data, leveraging interactive visualization tools like graphs and charts to offer intuitive insights into weather patterns. Additionally, focus on integrating the weather prediction system with external systems such as irrigation controllers or smart home automation systems, facilitating seamless communication to automate actions based on weather forecasts.

4.2 Exploration of other Machine Learning algorithms for Weather Prediction

To broaden the predictive capabilities, delve into decision tree-based algorithms like Random Forests for weather forecasting, juxtaposing their performance against linear regression to determine the optimal algorithm for the dataset. Additionally, investigate neural network architectures like backward and recurrent neural networks, assessing their ability to capture intricate nonlinear relationships within weather data, thus augmenting the predictive accuracy and robustness of the system. [7]

4.3 Consideration of additional Sensors and data sources for improved accuracy

Integrating atmospheric pressure sensors to measure air pressure, crucial for weather forecasting, and incorporate this data into the prediction model to enhance accuracy, particularly for predicting weather pattern changes like storms or cyclones. Additionally, include wind speed and direction sensors to capture wind patterns' impact on weather conditions, improving predictions of phenomena such as windstorms or hurricanes.

Explore integrating satellite imagery and radar data to supplement ground-based sensor measurements, providing broader geographical coverage and more comprehensive weather forecasts, especially for regions with limited sensor coverage.

By exploring these potential enhancements and extensions, the weather prediction project can evolve into a more robust and versatile system capable of providing more accurate and actionable forecasts for various applications and users.

Conclusion

In this project, we have successfully implemented a weather prediction system using machine learning techniques, specifically Linear Regression, integrated with Arduino Nano for real-time data collection and display. By training the dataset on a laptop, we have developed a regression algorithm capable of predicting the percentage chances of rain or other weather conditions based on environmental parameters such as humidity and temperature. The Arduino Nano board serves as the interface between the physical world and computational model, detecting sensor data and facilitating communication with the laptop via I2C serial communication. The predicted weather outcomes are then transmitted back to the Arduino Nano board and displayed on the screen, providing users with accessible and timely weather forecasts.

Through this project, we have demonstrated the feasibility of integrating Machine Learning algorithms with Embedded Systems for practical applications such as weather prediction. The system offers a cost-effective and scalable solution for localized weather forecasting, with potential applications in agriculture, transportation, and disaster management. Overall, this project showcases the intersection of Machine Learning, IoT, and Embedded Systems, paving the way for innovative solutions in weather prediction and beyond.

References

[1] T. Patil and D. K. Shah, “Weather forecasting analysis using linear and logistic regression algorithm,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 08, no. 06, pp. 2557–2564, 2021.

[2] GeeksforGeeks. (2024) Linear Regression in Machine Learning. [Online]. Available: https://www.geeksforgeeks.org/ml-linear-regression/

[3] Arduino Official Documentation. (2021) Arduino Nano. [Online]. Available: https://docs.arduino.cc/hardware/nano/

[4] P. Scherz and S. Monk, *Practical Electronics for Inventors*. New York: McGraw-Hill Education, 2016.

[5] Dejan, How To Mechatronics. (2017) Temperature & Humidity Sensor Module. [Online]. Available: https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/

[6] MicroDigisoft. (2022) Interfacing LCD with Arduino Uno. [Online]. Available: https://microdigisoft.com/interfacing-lcd-with-arduino-nano-board/

[7] Akkio, “Using machine learning for accurate weather forecasts in 2023,” 2023, accessed: April 5, 2024. [Online]. Available: [[invalidURLremoved]]