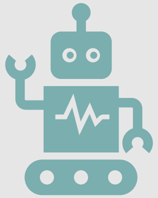
Weather predictions using Artificial Intelligence & Machine Learning

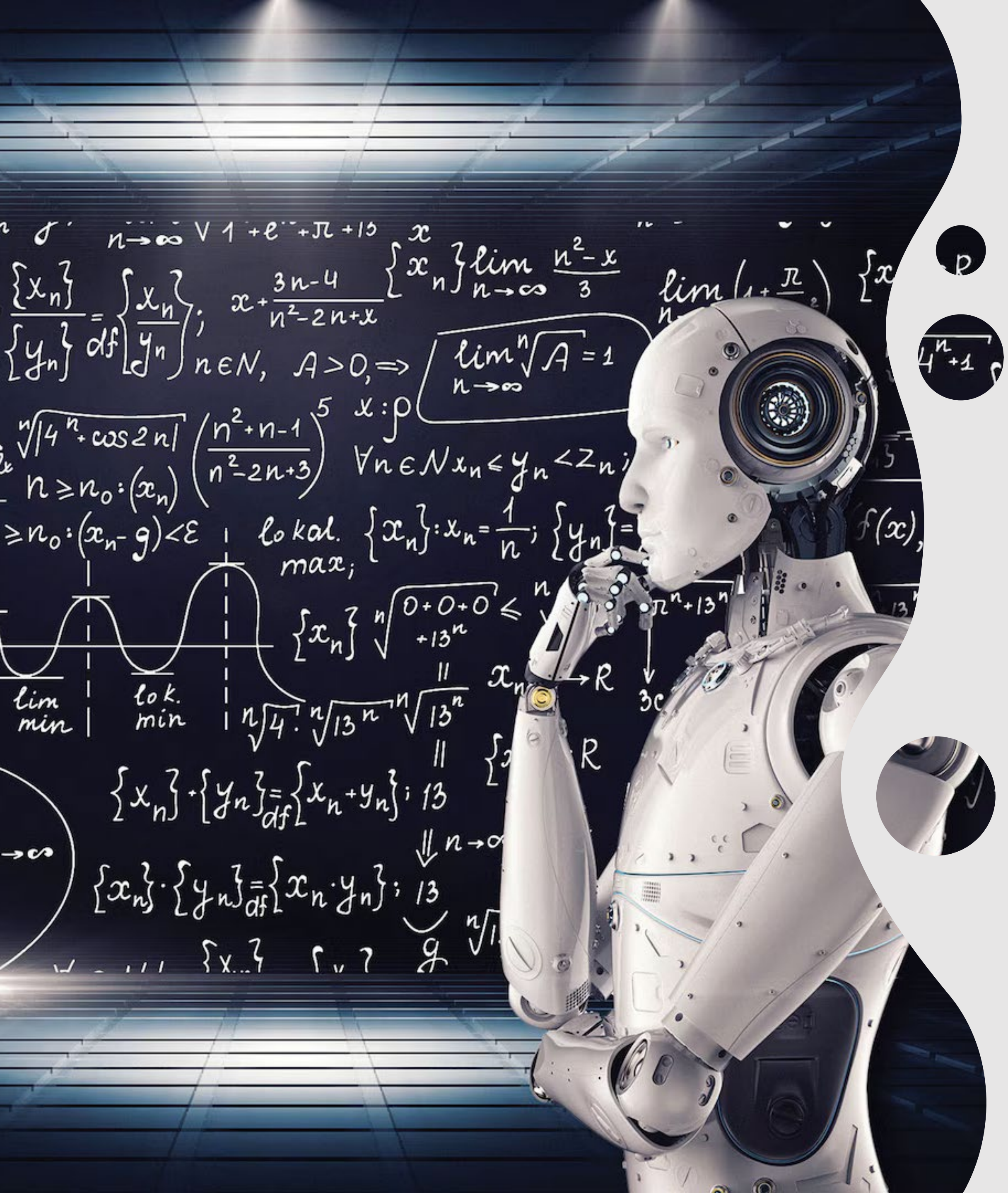
Presented By:
Eliran Haim

Introduction



What is Artificial intelligence (AI)?

What is Machine Learning (ML)?



Artificial Intelligence

- Artificial Intelligence (AI) is a branch of computer science focused on creating systems or machines capable of performing tasks that typically require human intelligence. These tasks include reasoning, problem-solving, learning, perception, language understanding, and decision-making. AI aims to replicate or simulate intelligent behavior using algorithms, data, and computational power.

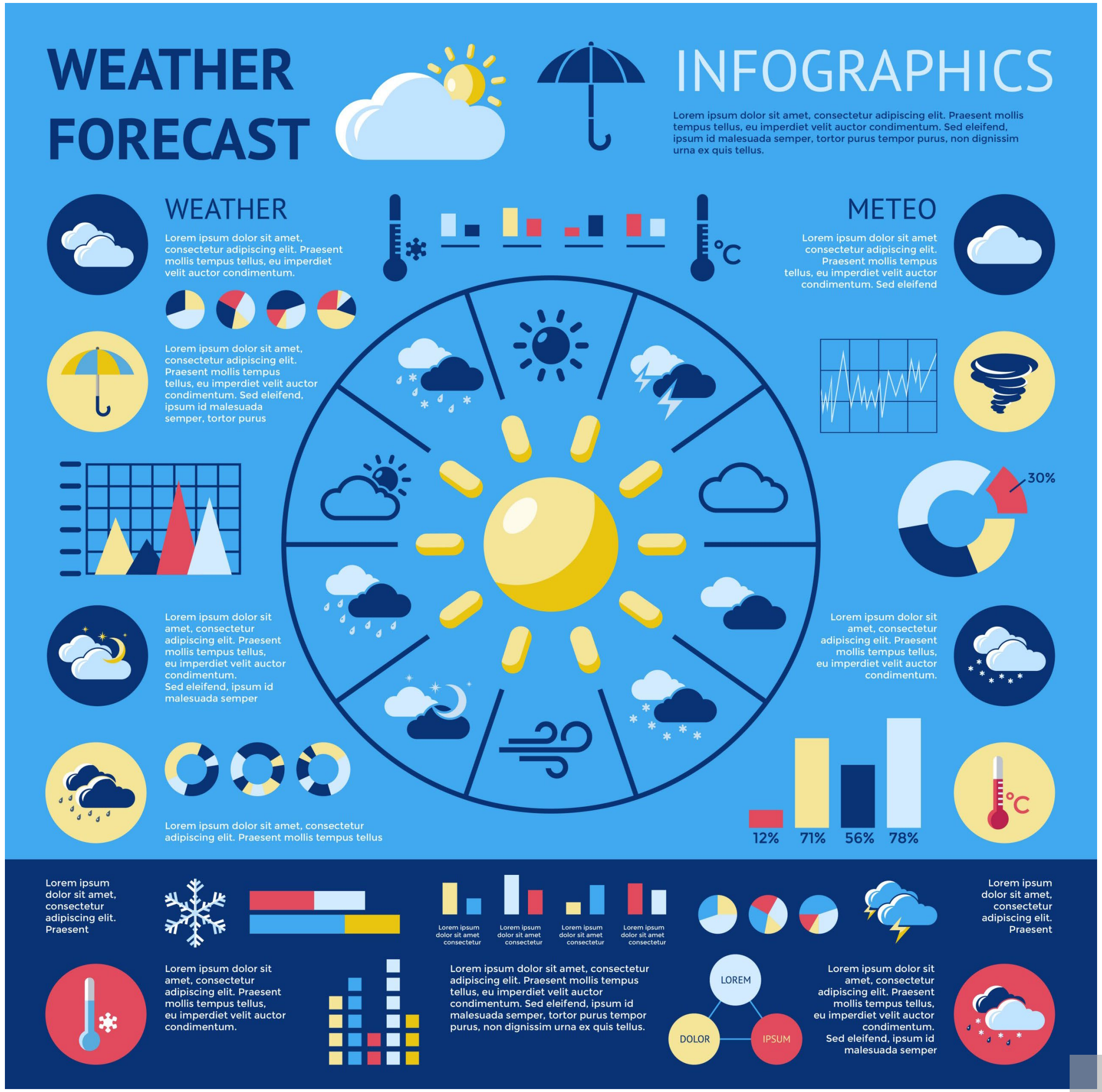
- Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that enables computers to learn from data without being explicitly programmed. Instead of following hardcoded rules, ML systems identify patterns in data, make decisions, and improve performance over time based on experience.

Netflix uses AI in the form of sophisticated recommendation systems to enhance the user experience by suggesting shows or movies you are likely to enjoy.

Siri, Apple’s virtual assistant, leverages AI for tasks like voice recognition, natural language processing (NLP), and intelligent decision-making.

Why use AI and ML for Weather predictions?

- AI and ML are essential for weather prediction because they handle complex patterns in vast datasets, enabling more accurate and real-time forecasts.



```cpp
#include <iostream>
#include <fstream>
#include <vector>
#include <sstream>
#include <cmath>

using namespace std;

// Function to read CSV data
vector<vector<double>> readCSV(const string& filename) {
    vector<vector<double>> data;
    ifstream file(filename);

    if (!file.is_open()) {
        cerr << "Error: Unable to open file " << filename << endl;
        return data;
    }

    string line;
    while (getline(file, line)) {
        vector<double> row;
        stringstream ss(line);
        string value;
        while (getline(ss, value, ',')) {
            row.push_back(stod(value));
        }
        data.push_back(row);
    }
    file.close();
    return data;
}

// Linear Regression: Train Model
pair<vector<double>, double> trainLinearRegression(const vector<vector<double>>& data) {
    int n = data.size();
    int features = data[0].size() - 1; // Exclude the target column
    vector<double> weights(features, 0);
    double bias = 0;

    // Training Parameters
    double learning_rate = 0.01;
    int epochs = 1000;

    for (int epoch = 0; epoch < epochs; ++epoch) {
        vector<double> gradients(features, 0);
        double bias_gradient = 0;

        for (const auto& row : data) {
            double prediction = bias;
            for (int j = 0; j < features; ++j) {
                prediction += weights[j] * row[j];
            }
            double error = prediction - row[features];
            bias_gradient += error;

            for (int j = 0; j < features; ++j) {
                gradients[j] += error * row[j];
            }
        }

        // Update weights and bias
        for (int j = 0; j < features; ++j) {
            weights[j] -= learning_rate * gradients[j] / n;
```

```cpp
        // Update weights and bias
        for (int j = 0; j < features; ++j) {
            weights[j] -= learning_rate * gradients[j] / n;
        }
        bias -= learning_rate * bias_gradient / n;
    }

    return { weights, bias };
}

// Predict Temperature
double predictTemperature(const vector<double>& features, const vector<double>& weights, double bias) {
    double prediction = bias;
    for (size_t i = 0; i < features.size(); ++i) {
        prediction += weights[i] * features[i];
    }
    return prediction;
}

int main() {
    string filename = "weather_data.csv"; // Ensure you have a CSV file with your data
    auto data = readCSV(filename);

    if (data.empty()) {
        cerr << "No data found!" << endl;
        return 1;
    }

    // Split data into training and testing sets (80-20 split)
    int train_size = static_cast<int>(0.8 * data.size());
    vector<vector<double>> train_data(data.begin(), data.begin() + train_size);
    vector<vector<double>> test_data(data.begin() + train_size, data.end());

    // Train Linear Regression Model
    auto result = trainLinearRegression(train_data); // Get weights and bias as a pair
    vector<double> weights = result.first;          // Extract weights
    double bias = result.second;                    // Extract bias

    // Test Model
    cout << "Testing Model...\n";
    for (const auto& row : test_data) {
        vector<double> features(row.begin(), row.end() - 1);
        double actual = row.back();
        double predicted = predictTemperature(features, weights, bias);
        cout << "Actual: " << actual << ", Predicted: " << predicted << endl;
    }

    return 0;
}
```

Conclusion

In this project, we demonstrated how Machine Learning, specifically linear regression, can be applied to weather prediction. By leveraging historical weather data, we trained a model to identify patterns and make accurate temperature forecasts based on features like humidity and wind speed. This approach highlights the potential of AI and ML to process complex datasets efficiently, adapt over time, and deliver precise predictions.

Compared to traditional methods, our model showcases the benefits of flexibility and scalability, emphasizing the importance of ML in addressing challenges like extreme weather conditions and climate change.

As future improvements, integrating larger datasets and more advanced models, such as neural networks, could further enhance the accuracy and scope of predictions, enabling more robust and long-term forecasting solutions. This project underscores the transformative role of AI and ML in creating smarter, data-driven solutions for global challenges.

Sources

https://www.nationalacademies.org/news/2024/01/how-ai-is-shaping-weather-research-and-forecasting-an-interview-with-amy-mcgovern#:~:text=NOAA%20also%20has%20a%20real,when%20you%20search%20for%20weather.