Tomato Plant Disease Detection Project

Forecasting

& Predictive Analytics

Presented By:

Name	Id
Zinab Rabie Fathy	20211045
Basmala Ahmed Mohamed	20201044
Reham Ashraf Fathy	20201079
Marwa Mahmoud Esmail	20201160
Zienab Mohy Eldin Mohamed	20180113

By Dr/Olivia

Contents

Chapter 1: Problem Definition

1.1 Problem Overview

Tomato plants are a crucial agricultural product worldwide, and their health directly impacts crop yield and quality. Diseases such as Early Blight, Late Blight, and others can severely reduce productivity. Timely and accurate detection of diseases is critical for minimizing losses and ensuring sustainable agricultural practices.

The goal of this project is to develop a system that can classify tomato plant health as either **Healthy** or **Diseased** using various features of the plants such as leaf spot size, leaf color index, temperature, and humidity. This system will provide actionable insights for farmers and agricultural professionals to make informed decisions about disease management.

1.2 Objective

The primary objective is to analyze the dataset and identify patterns that differentiate healthy plants from diseased ones. This involves:

- Understanding the relationship between features (e.g., leaf spot size, temperature) and the health status of the plant.
- Identifying key indicators of plant health.
- Creating a decision-making framework for disease detection based on the available data.

1.3 Business Context

Incorporating disease detection into agricultural practices has numerous benefits, including:

- **Reducing crop losses**: Early detection minimizes the spread of diseases.
- **Improving productivity**: Healthy plants lead to higher yields.
- **Optimizing resources**: Targeted interventions reduce the unnecessary use of pesticides.
- **Economic benefits**: Increased yield and quality lead to higher profits for farmers.

The project aligns with the broader goals of precision agriculture by leveraging data-driven techniques to enhance farming efficiency.

Chapter 2: Dataset

2.1 Data Description

The dataset contains the following features:

Feature	Description	Data Type	Range
Leaf_Spot_Size	Size of leaf spots in cm^2. Indicates the severity of disease.	Numeric (float)	0 to 10 cm^2
Leaf_Color_Index	Leaf color index (0 for healthy, 100 for highly diseased).	Numeric (float)	0 to 100
Temperature	Ambient temperature in degrees Celsius.	Numeric (float)	15°C to 35°C
Humidity	Relative humidity in percentage.	Numeric (float)	30% to 90%
Disease	Health status of the plant (target variable).	Categorical	Healthy/Diseased

2.2 Target Variable

The target variable is **Disease**, which categorizes the health of the plant as either:

- **Healthy**
- **Diseased** (includes Early Blight, Late Blight, and other diseases)

2.3 Dataset Summary Statistics

Feature Distribution Examples:

- **Leaf_Spot_Size**: Mean: 5.5 cm^2, Range: 1.76 to 8.85 cm^2.
- **Leaf_Color_Index**: Mean: 54.63, Range: 36.52 to 63.16.
- **Temperature**: Mean: 24.80°C, Range: 16.21°C to 27.20°C.
- **Humidity**: Mean: 63.89%, Range: 45.30% to 75.97%.

Target Variable Distribution:

- Diseased: **87%**
- Healthy: **13%**

Data Preparation

3.1 Data Quality Assessment

Issues Identified:

1. **Duplicate Records**: Identical rows may exist and need elimination.
2. **Outliers**: Extreme values in features such as temperature and humidity.
3. **Missing Values**: Some rows may have incomplete data.
4. **Attribute Standardization**: Features need to be scaled for consistency.

Data Cleansing Techniques:

1. **Elimination of duplicates**: Ensures unique records.
2. **Outlier handling**: Outliers are identified using statistical thresholds (e.g., Z-scores).
3. **Substitution of missing values**: Impute missing values using mean/median for numerical features.
4. **Standardization**: Rescale values for features like leaf spot size and temperature to ensure comparability.

3.2 Transformation Steps:

- Scale features such as **Leaf_Spot_Size** and **Leaf_Color_Index** to a 0–1 range.
- Convert temperature to Fahrenheit for alternative analysis if needed.
- Create a binary encoding for the **Disease** variable: 0 = Healthy, 1 = Diseased.

Data Exploration

4.1 Descriptive Statistics

- **Mean, Median, Mode**: Understand central tendencies of numeric features.
- **Variance and Standard Deviation**: Gauge feature variability.
- **Feature Correlations**: Identify relationships between features such as leaf color index and disease status.

4.2 Data Visualization Techniques

1. **Histograms**:
    - Show distribution of features like leaf spot size and temperature.
    - Example: Histogram of Leaf_Color_Index with separate bars for Healthy vs Diseased plants.
2. **Scatter Plots**:
    - Visualize relationships between variables (e.g., Humidity vs Leaf_Color_Index).
    - Highlight clusters of healthy and diseased plants.
3. **Box Plots**:
    - Identify outliers in features such as temperature and leaf spot size.
    - Example: Box plot comparing humidity for healthy vs diseased plants.
4. **Heatmaps**:
    - Visualize correlations among features (e.g., strong correlation between Leaf_Color_Index and Disease).
5. **Pie Charts**:
    - Show percentage distribution of Healthy vs Diseased plants.

Chapter 3: Algorithms Implementation

In this chapter, we describe the implementation of various classification algorithms applied to the dataset. The selected algorithms include Decision Trees, Rule Induction, Naive Bayes, and Neural Networks. Each algorithm was implemented using the RapidMiner platform, leveraging its robust tools for data preprocessing, training, and evaluation.

1. **Decision Trees**
    - Decision Trees create a tree-like model of decisions, enabling clear interpretability and effective classification.
    - The algorithm was applied to the dataset with default hyperparameters, and the resulting model's performance was evaluated using accuracy, precision, and recall metrics.
2. **Rule Induction**
    - Rule Induction generates classification rules based on the dataset. This approach is straightforward and ensures high interpretability.
    - The model was trained and tested using RapidMiner to extract useful patterns and measure its classification effectiveness.
3. **Naive Bayes**
    - Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, assuming independence among predictors.
    - The algorithm was executed on the dataset, and performance metrics such as accuracy and class precision were obtained.
4. **Neural Networks**
    - Neural Networks simulate the human brain's learning process by using interconnected layers of nodes (neurons).
    - The implementation involved using a standard feedforward neural network model. The results were evaluated based on the model's ability to classify instances accurately.

Chapter 4: Results, Comparison, and Conclusion

This chapter presents the results of the classification models, followed by a comparative analysis of their performance. Recommendations based on the findings are also provided.

4.1 Results

The results of each algorithm are summarized in the table below:

Algorithm	Accuracy (%)	Class Precision (Diseased)	Class Precision (Healthy)	Class Recall (Diseased)	Class Recall (Healthy)
Decision Trees	99.00	99.15	97.83	99.72	93.75
Rule Induction	97.00	96.55	100.00	100.00	81.25
Naive Bayes	85.00	84.85	100.00	100.00	6.25
Neural Networks	85.50	90.61	36.84	93.18	29.17

4.2 Comparison

- **Best Algorithm:** Decision Trees emerged as the best-performing algorithm with the highest accuracy of 99.00% and balanced precision and recall for both classes.
- **Worst Algorithm:** Naive Bayes demonstrated the weakest performance, particularly in class recall for the "Healthy" category, achieving only 6.25%.
- **Other Observations:**
    - Rule Induction performed well overall but had lower recall for the "Healthy" class compared to Decision Trees.
    - Neural Networks, though effective in detecting diseased cases, struggled with precision and recall for the "Healthy" category.

4.3 Conclusion

Based on the analysis, the Decision Tree algorithm is recommended for this classification problem due to its superior performance metrics across all categories. While Rule Induction is a viable alternative, further optimization of the Neural Network model could enhance its performance. Naive Bayes, given its limitations with this dataset, is not recommended.