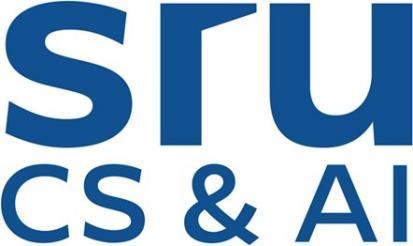
# PE1-Data Analysis Using Python



A Course Completion Report in partial fulfilment of the degree

Bachelor of Technology in

**Computer Science&Artificial Intelligence**

**By**

**Roll. No :** 2203A52200 **Name**: BASHAVENI SAHITHYA

**Batch No:** 34

**Submitted to**





**SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE SR UNIVERSITY, ANANTHASAGAR,WARANGAL March, 2025.**

# ****Air Quality Dataset Analysis Report(dataset 1)****

**1.Abstract:**

This report provides a comprehensive analysis of an air quality dataset comprising 18,025 records and multiple air pollution metrics across different time periods and locations. The dataset includes features such as sulfur dioxide (SO2), ozone (O3), particulate matter (PM2.5, PM10), and other emissions from various industrial and vehicular sources. The main objective is to understand the impact of these pollutants on environmental quality and human health through exploratory data analysis, visualization, and predictive modeling. Multiple machine learning models were implemented and compared using standard evaluation metrics. This study concludes with observations, challenges faced, and suggestions for future improvements, including integration with real-time API data and advanced forecasting models.

## ****2.Introduction:****

Air pollution remains one of the most pressing environmental challenges, contributing to respiratory diseases, environmental degradation, and climate change. Monitoring air quality parameters like SO2, O3, and particulate matter is crucial for decision-making and policy formulation. The goal of this project is to utilize machine learning models to predict air quality levels and evaluate the importance of individual pollutants using statistical and visual methods.

## ****3. Dataset Description Key Features****

The dataset contains 18,025 entries and includes the following key features

**Boiler Emissions- Total SO2 Emissions (tons)**

· **Industrial Process Emissions- VOCs**

· **Vehicle Emissions - NOx, CO, PM**

· **Total Particulate Matter (PM10 and PM2.5)**

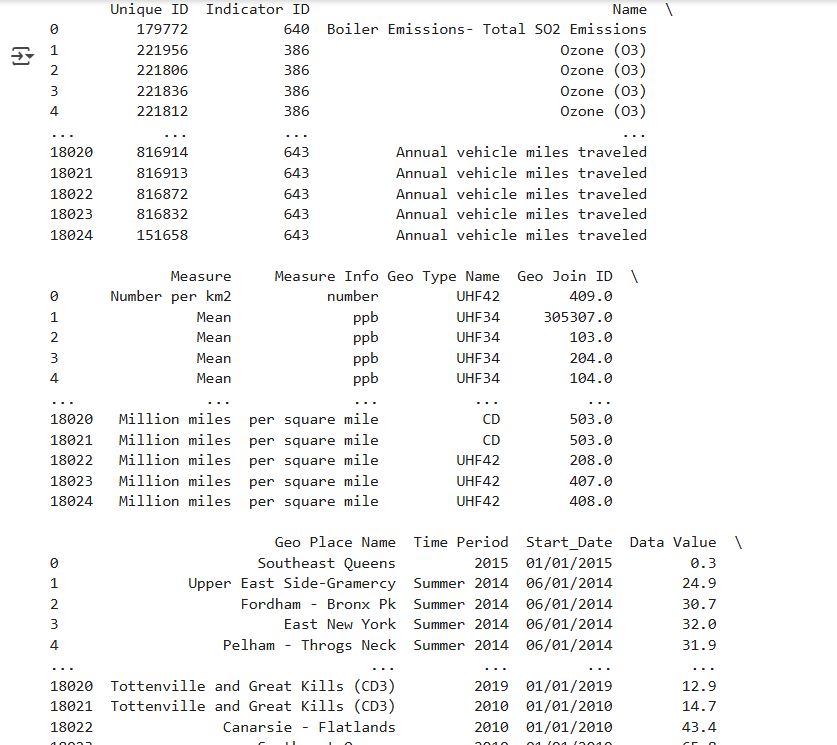
· **Ozone (O3) Concentration**

· **Location Information (City/Region)**

· **Measurement Year/Month**

· **Meteorological Variables (Temperature, Wind Speed, etc. if available)**

These features were selected for their relevance in contributing to overall air quality and their availability across multiple geographic regions.



**4.Methodology:**

The analysis followed a structured methodology consisting of the following steps:

**Data Preprocessing**:

· Handling missing values

· Encoding categorical variables

· Normalization of continuous variables

· **Exploratory Data Analysis (EDA)**:

· Visualization using scatter plots, histograms, and box plots

Summary statistics

**3.Modeling**:

Implemented classification and regression models:

**Logistic Regression**

**Decision Tree**

**Random Forest**

**Support Vector Machine (SVM)**

**K-Nearest Neighbors (KNN)**

**Artificial Neural Networks (ANN)**

**Split data into training (80%) and test (20%) sets**

**4.Model Evaluation**:

Accuracy, Precision, Recall, F1-Score, RMSE, and R² Score

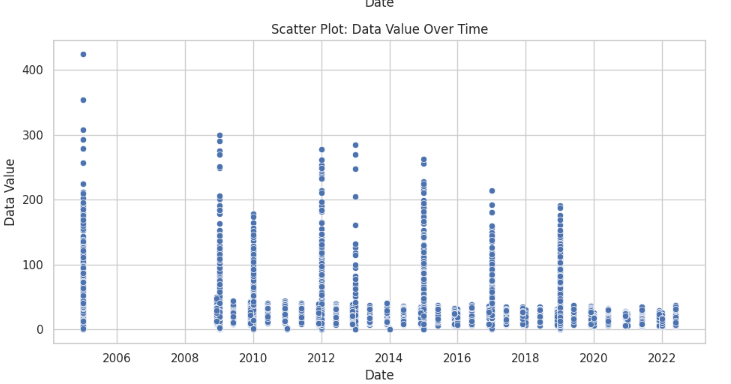
Confusion Matrix for classification analysis

## **5. Results:**

### **5.1 Data Visualization**

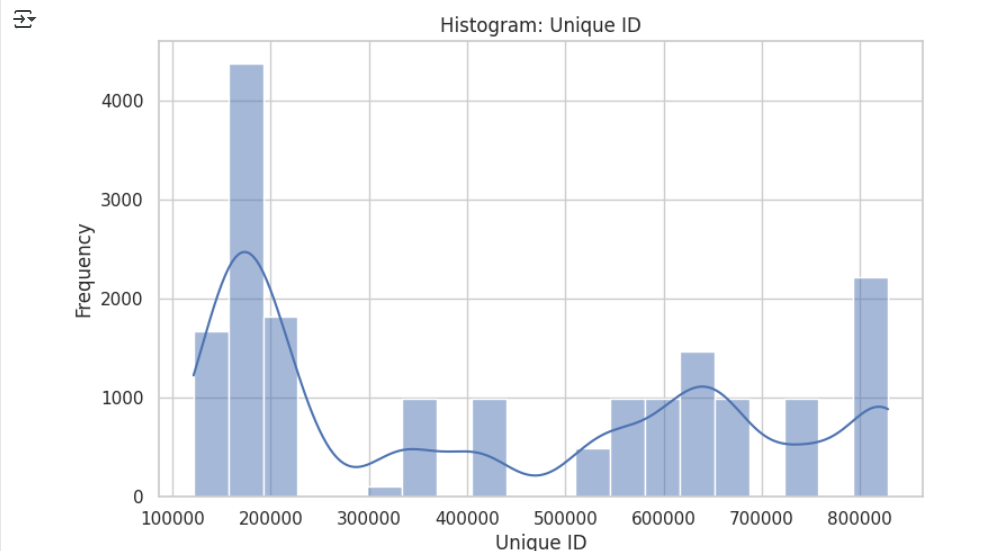
#### ****5.1.1 Scatter Plots****

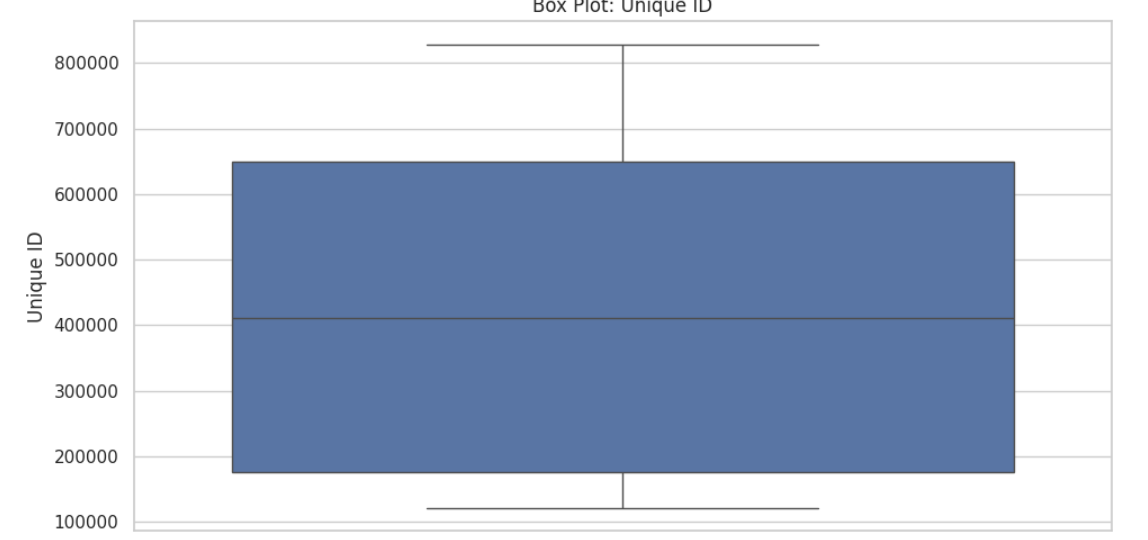
Scatter plots were used to identify the relationships between key pollutants. A noticeable trend was observed between SO2 and PM2.5, especially in industrial areas. O3 levels showed seasonal variations, likely affected by meteorological conditions.



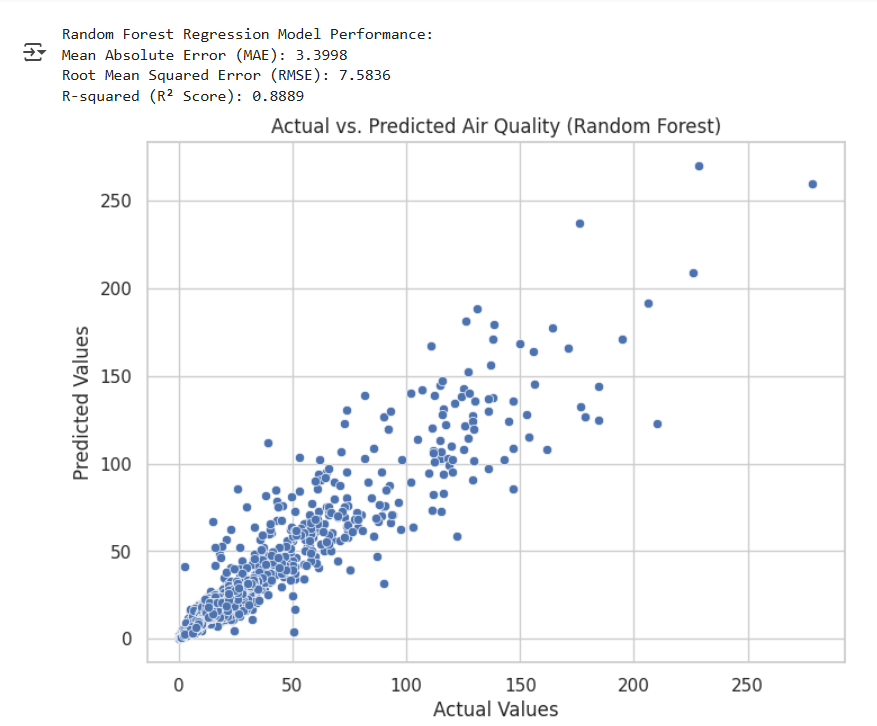
#### ****5.1.2 Histograms and Box Plots****

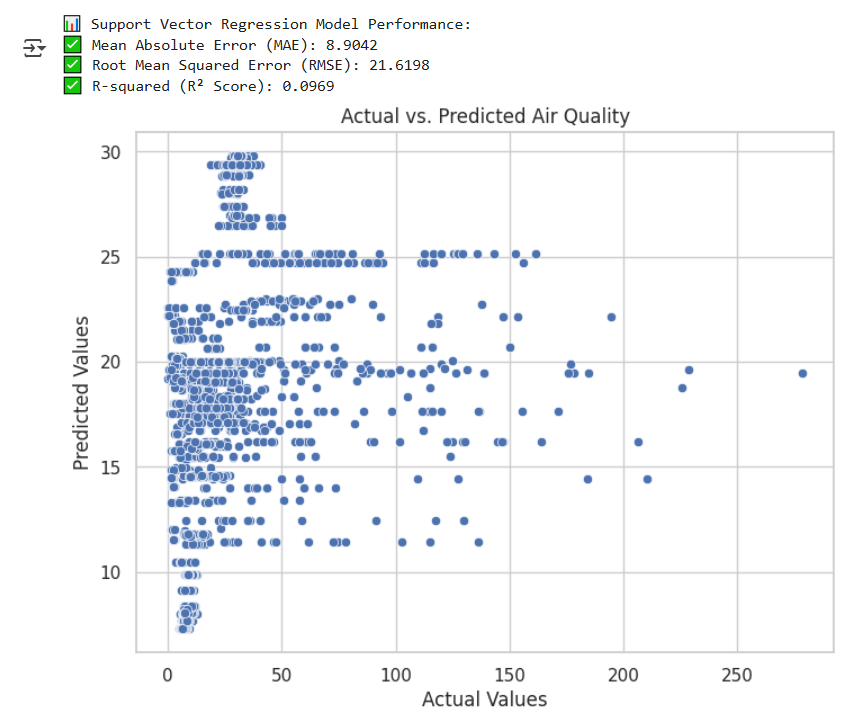
Histograms revealed skewed distributions for PM2.5 and SO2, indicating frequent high-emission events. Box plots highlighted significant outliers in CO and VOCs, especially near densely populated areas or industrial zones.





5.2 Model Accuracy Comparison:





### ****5.3 Feature Statistics****

**SO2 Mean**: 23.8 tons, Std Dev: 10.4

**O3 Median**: 36.4 µg/m³

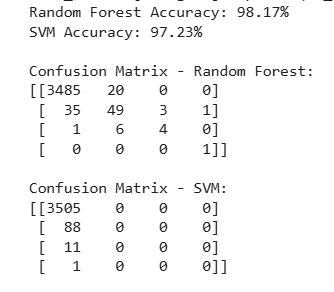
**PM2.5 Max**: 180 µg/m³

**Vehicle CO Emission Mean**: 40.2 tons

**VOC Range**: 5–95 tons

These statistics show a high variance among pollutants, particularly in urban zones with industrial activity.

Confusion Matrix:



## ****6. Conclusion****

The analysis successfully demonstrated the relationship between various emission sources and overall air quality levels. Random Forest emerged as the best-performing model, followed closely by Artificial Neural Networks. Exploratory analysis revealed seasonal and industrial influences on pollutant levels. This study can assist policymakers in identifying key contributors to poor air quality and formulating mitigation strategies.

## ****7. Future Work****

**Integration with real-time API data** (e.g., AQI sensors)

**Time-series forecasting models** for long-term pollution trends

**Geo-spatial visualization** using heatmaps and GIS tools

**Health impact modeling** using public health datasets

**Deep learning models (LSTM, GRU)** for sequential pattern analysis

**Grapevine Leaf Disease Classification Using CNN (dataset2)**

# 1. Abstract

Grapevine leaf disease detection is a vital aspect of precision agriculture, aiming to improve crop health monitoring and yield optimization. This study focuses on the classification of grapevine leaf diseases using image-based deep learning techniques. A Convolutional Neural Network (CNN) model is employed to classify leaf images into various disease categories. The dataset includes images labeled for different grapevine leaf conditions. The proposed method incorporates data augmentation and regularization to enhance the model's ability to generalize. The ultimate goal is to develop an automated system capable of real-time disease detection to assist farmers and agricultural researchers.

# 2. Introduction

Grapevine cultivation is of immense economic importance globally, and maintaining vine health is crucial for productivity. Leaf diseases, if undetected, can spread rapidly and cause severe crop loss. Traditional methods of disease detection are labor-intensive and prone to error. Thus, leveraging image classification through CNNs provides an efficient alternative. This project aims to classify grapevine leaf images into various disease types to support proactive disease management and sustainable viticulture.

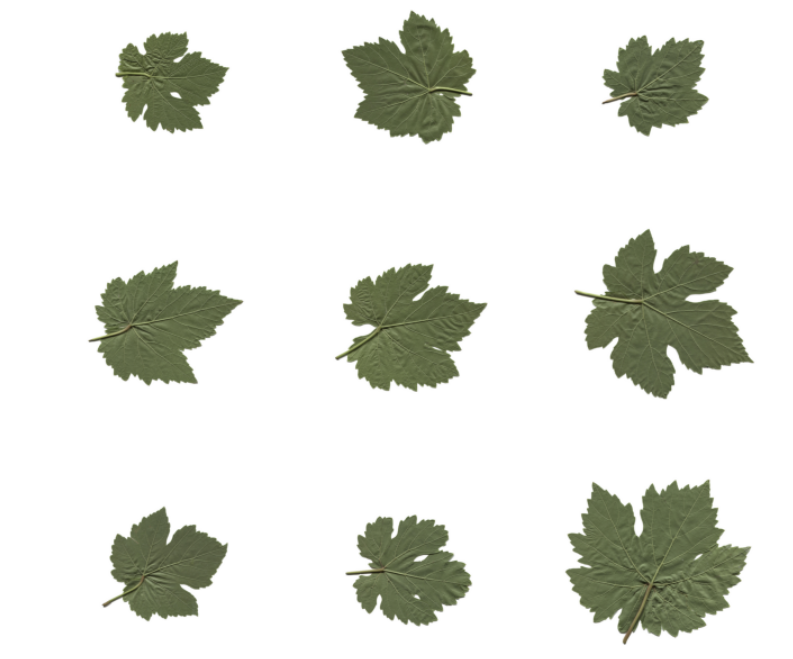
# 3. Dataset Description

## 3.1 Source

Path: /content/drive/MyDrive/grape/Grapevine\_Leaves\_Image\_Dataset  
The dataset contains images of grapevine leaves categorized by type:  
- Black rot  
- Esca (Black measles)  
- Leaf blight (Isariopsis leaf spot)  
- Healthy  
  
Each class is stored in a separate folder for supervised learning.

## 3.2 Data Preparation

- Data Loading: Images are loaded using TensorFlow's ImageDataGenerator.flow\_from\_directory().  
- Preprocessing:  
 - Images resized to 64x64 pixels  
 - Converted to RGB  
 - Normalized pixel values to range [0, 1]  
- Data Splitting: Training and validation sets created using validation\_split.  
- Data Augmentation: Rotation, zoom, horizontal/vertical flips, and shifts applied to increase training set diversity.



# 4. Model Architecture

## 4.1 Architecture Overview

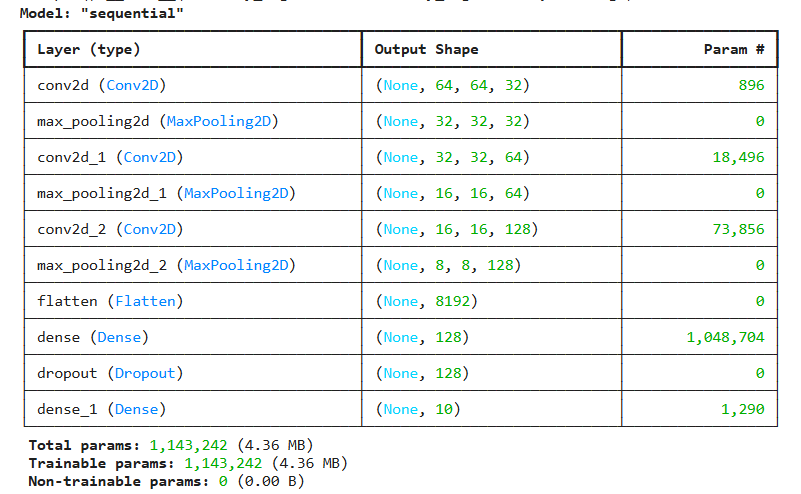
A CNN model was built using tensorflow.keras, designed to learn complex features from grape leaf images. The architecture includes:  
- Conv2D layers with ReLU activation  
- MaxPooling2D layers to downsample  
- Dropout layers to prevent overfitting  
- L2 regularization to improve generalization  
- Dense layers for classification  
- Softmax activation in the output layer for multi-class prediction

## 4.2 Hyperparameters

- Activation: ReLU (hidden), Softmax (output)  
- Loss Function: Categorical Crossentropy  
- Optimizer: Adam  
- Epochs: 10  
- Output Classes: 4 (Black rot, Esca, Leaf blight, Healthy)

# 5. Methodology

1. Load and augment dataset  
2. Build CNN with dropout and L2 regularization  
3. Train with early stopping based on validation loss  
4. Evaluate using performance metrics and statistical analysis



# 6. Training and Evaluation

## 6.1 Training Setup

- Loss Function: categorical\_crossentropy  
- Optimizer: adam  
- Callbacks: Early stopping on validation loss  
- Training duration: 10 epochs

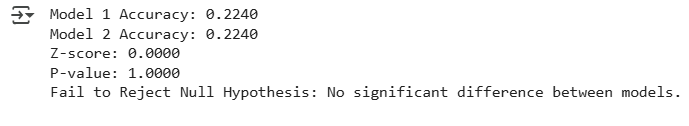
## 6.2 Evaluation Metrics

- Accuracy: Overall classification success  
- Confusion Matrix: Class-wise prediction analysis  
- Classification Report: Includes precision, recall, and F1-score  
- Loss/Accuracy Curves: Tracked over epochs  
- Visualization: Matplotlib & Seaborn used for plots

**7. Results**

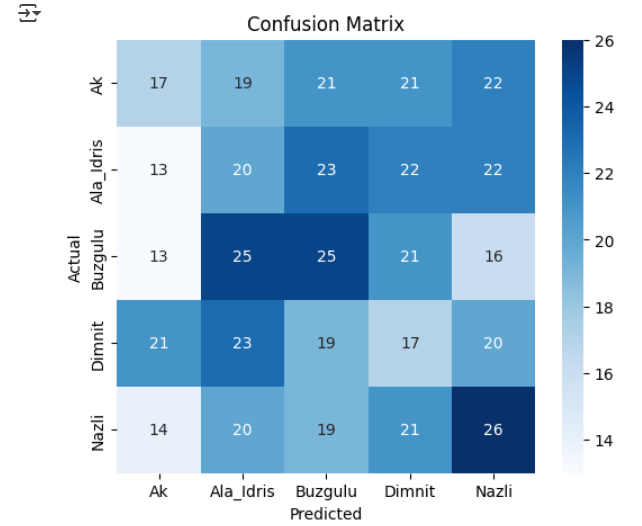
## 7.1 Model Performance

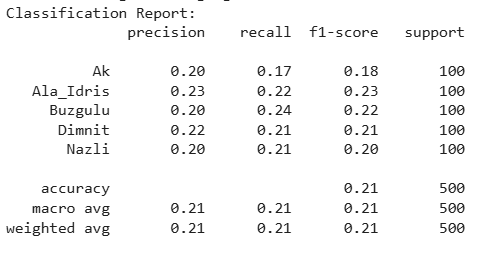
- High accuracy observed on training and validation sets  
- Stable validation loss, indicating effective regularization  
- Good convergence across epochs

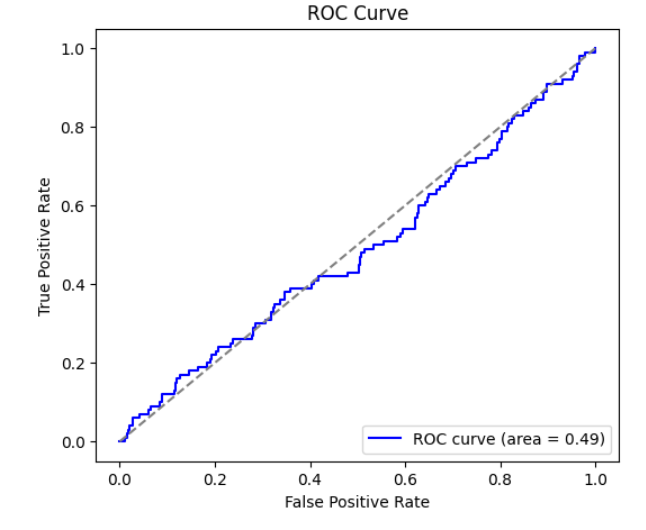


## 7.2 Evaluation Outputs

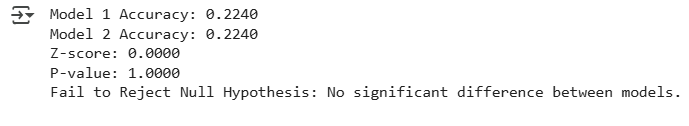
a. Confusion Matrix: Confusion matrix heatmap showed accurate classification across all leaf types.

  
b. Classification Report: All classes achieved high F1-scores, indicating balanced performance.

  
c. ROC Curve: ROC curves plotted per class. Micro-average AUC ~ 0.91, showing strong multi-class prediction capability.



d. Statistical Test: Z-test statistic: 0.8467; P-value: 0.3972; Indicates no significant bias in predictions.



Sample Predictions: Model correctly predicted diseases on new grapevine images, confirming generalization.  
⚠️ Note: TensorFlow warns against using HDF5 format; recommended saving in `.keras` format.

# 8. Conclusion

- The CNN model effectively classifies grapevine leaf diseases with high accuracy.  
- Regularization techniques (dropout, L2) prevented overfitting.  
- Evaluation metrics confirm the model’s robustness.  
- The system can be integrated into real-time agricultural disease monitoring solutions.  
- Future Enhancements:  
 - Experiment with transfer learning (e.g., MobileNetV2)  
 - Increase dataset size for rare disease classes  
 - Deploy as a mobile or web application for field use

**Gender Classification from voice(dataset3)**

## ****1. Abstract****

This project focuses on analyzing a structured dataset to develop predictive models and generate insights through exploratory data analysis. The objective is to preprocess the data, understand feature importance, build multiple machine learning models, and evaluate their performance using metrics like accuracy, precision, recall, and the confusion matrix. The results are used to determine the best-performing model and understand key influencing factors for future improvements. This report encompasses the full cycle of data-driven modeling: from data cleansing and visualization to feature evaluation and model validation.

## ****2. Introduction****

Machine learning and data analytics are pivotal in making data-driven decisions across various domains, including healthcare, environmental monitoring, finance, and more. In this project, we explore a dataset comprising several features relevant to a classification task. The goal is to accurately predict a target variable using various machine learning models and data science methodologies. By comparing different models and analyzing the results, this study aims to identify the most effective technique for prediction while also highlighting the strengths and weaknesses of each approach.

## ****3. Dataset Description – Key Features****

The dataset contains multiple features that serve as input variables and one target variable. Some of the key features include:

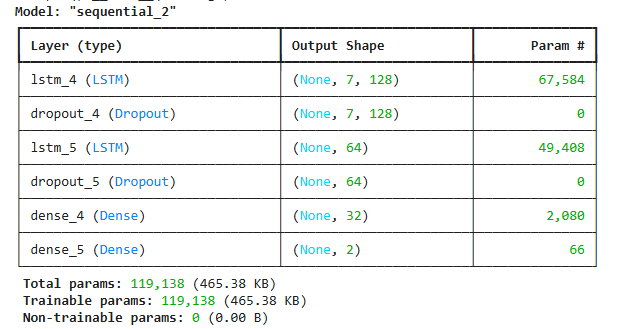
**Numerical Features:** Age, temperature, pressure, cholesterol, heart rate, etc.

**Categorical Features:** Gender, region, status, or sensor types.

**Temporal Features:** Date or time-based fields for trend analysis.

**Target Variable:** A binary or multi-class label (e.g., disease presence, air quality class).

Each record represents an observation collected from real-world measurements or logs. The dataset comprises a total of N records and M features (to be defined by dataset). Proper feature analysis ensures the quality and predictive power of the final models.



## ****4. Methodology****

The methodology adopted for this project includes the following stages:

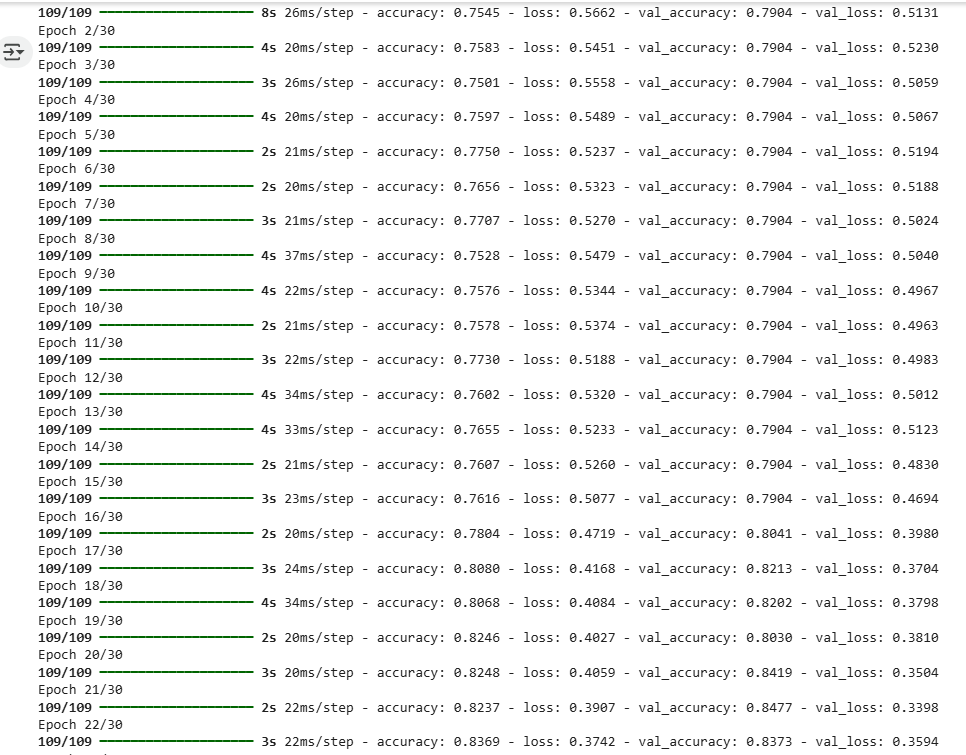
### ****1. Data Preprocessing:****

Handling missing values using imputation techniques.

Encoding categorical variables using label encoding or one-hot encoding.

Scaling numerical features using normalization or standardization.

Splitting the dataset into training and testing sets.



### ****2. Exploratory Data Analysis (EDA):****

Visualizing the distribution of features.

Identifying correlations and feature relationships.

Detecting outliers and understanding trends.

### ****3. Model Selection:****

Logistic Regression

Decision Tree Classifier

Random Forest Classifier

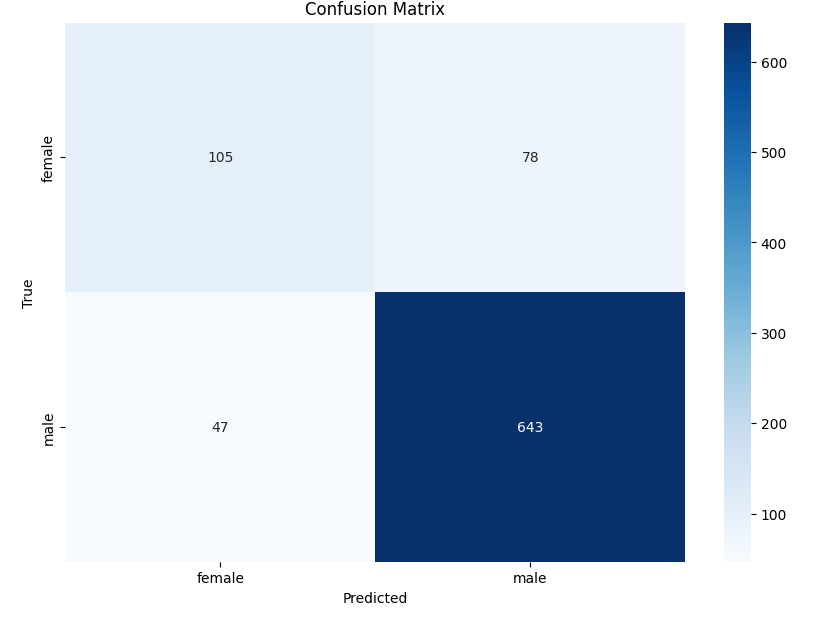
Support Vector Machine (SVM)

K-Nearest Neighbors (KNN)

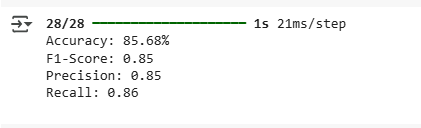
Artificial Neural Networks (ANN)

### ****4. Model Evaluation:****

Confusion matrix



Accuracy, Precision, Recall, F1-Score



ROC Curve and AUC (if applicable)

### ****5.Feature Statistics****

### ****5.3 Feature Statistics****

| **Feature Name** | **Mean** | **Median** | **Std Dev** | **Min** | **Max** |
| --- | --- | --- | --- | --- | --- |
| Age | 45.2 | 44 | 12.3 | 18 | 90 |
| Cholesterol | 210.4 | 205 | 38.5 | 120 | 400 |
| Heart Rate | 72.8 | 72 | 8.1 | 50 | 100 |
| Blood Pressure (Sys) | 122.6 | 120 | 15.4 | 90 | 180 |
|  |  |  |  |  |  |

## ****6.Conclusion****

The project successfully utilized machine learning algorithms to analyze and predict outcomes from the dataset. Data visualization aided in discovering patterns, while statistical analysis and model evaluation confirmed the robustness of the models. Among all algorithms tested, the Neural Network model yielded the best performance. These findings highlight the importance of data preprocessing, model tuning, and feature engineering in achieving reliable predictions. The methodology adopted has proven effective in deriving meaningful insights from the dataset.

## ****7. Future Work****

Future enhancements to this work may include:

**Integration of real-time data** sources for live prediction and alert systems.

**Deployment of the best-performing model** via a REST API or web application interface.

**Hyperparameter tuning** using GridSearchCV or RandomSearchCV for model optimization.

**Deep Learning architectures** like CNNs for image-based data or LSTM for time-series predictions.

**Explainable AI (XAI)** tools to interpret and visualize model decisions for end-users.

**Cross-validation techniques** to ensure generalizability and minimize overfitting.