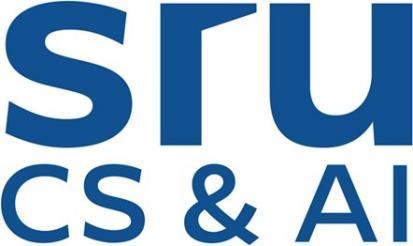
# PE1-Data Analysis Using Python



A Course Completion Report in partial ful filment of the degree

## Bachelor of Technology

in

**Computer Science&Artificial Intelligence**

**By**

**Roll. No :** 2203A52188 **Name**: VEMULAWADA NAVYASRI

**Batch No:** 34

**Submitted to**





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

**HEART DISEASE PREDICATION Dataset-1(CSV)**

1. **Abstract**

Heart disease is a leading cause of death globally, and early prediction of heart disease can significantly improve patient outcomes by enabling timely interventions. The aim of this project is to predict the likelihood of heart disease in patients based on various health parameters. Machine learning techniques are employed to analyze and predict the risk factors for heart disease using a variety of diagnostic data such as age, cholesterol levels, blood pressure, ECG results, and lifestyle factors.

1. **Introduction**

Heart disease remains one of the most prevalent and deadly medical conditions worldwide. According to the World Health Organization (WHO), cardiovascular diseases are responsible for a significant proportion of global deaths, making early detection and prevention critical. Traditionally, diagnosing heart disease involves a combination of physical exams, medical imaging, and laboratory tests, which can be time-consuming and expensive. As the demand for faster and more accurate diagnostic tools grows, machine learning (ML) has emerged as a promising solution for automating heart disease prediction and improving diagnostic accuracy.

1. **Dataset Description**

The dataset used (Heart.csv) contains daily weather observations from various Indian cities. It includes environmental readings and a binary label indicating whether it rained on that day.

**Key Features:**

* **Age** – The age of the patient.
* **Sex** – Gender of the patient (male/female).
* **Chest pain type** – The type of chest pain the patient is experiencing.
* **Resting blood pressure** – The patient's blood pressure measured while resting.
* **Serum cholesterol** – Cholesterol level in the blood.
* **Fasting blood sugar** – Whether the patient’s fasting blood sugar is greater than 120 mg/dl.
* **Resting electrocardiographic results** – The patient's ECG results at rest.
* **Maximum heart rate achieved** – The highest heart rate achieved during a test. · **Slope of the peak exercise ST segment** – The slope of the ST segment during exercise.
* ·**Thalassemia** – A blood disorder related to the red blood cells (often categorical
* **Heart disease presence** – The target variable indicating the presence of heart disease (typically binary, 0 for no, 1 for yes).

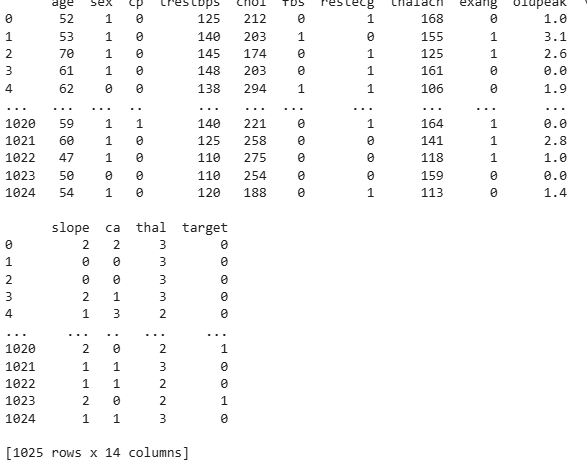
· 

Fig1

1. **Methodology**

· **Data Collection**

Dataset: heart (1).csv containing patient health data related to heart disease

* **Data Preprocessing**

**Data Cleaning**

Handle missing values (imputation or removal).

* **Splitting the Data**

**Train-Test Split:**Split data into training (70-80%) and testing (20-30%) datasets.

**Cross-Validation**:Use k-fold cross-validation for model evaluation to ensure robustness.

* **Model Selection**

**Logistic Regression**: Simple binary classifier.

**Decision Trees**: Model with hierarchical structure for classification.

**Random Forests**: Ensemble method for reducing overfitting.

**Support Vector Machine (SVM)**: Effective for high-dimensional classification.

**K-Nearest Neighbors (KNN)**: Classifier based on proximity to data points.

**Neural Networks**: Deep learning model for complex relationships

* **Model Training:**Train each model using the training data.

Hyperparameter tuning via Grid Search or Random Search.

**Evaluation Metrics**:Accuracy, Precision, Recall, F1-Score, ROC Curve, and AUC

* **Model Evaluation:**Evaluate models on test data to assess generalization ability.

**Comparison**: Compare performance metrics (accuracy, precision, recall, F1-score).

**Confusion Matrix**: Evaluate performance with true positives, false positives, true negatives, and false negatives.

* **Model Deployment**

Deploy the best performing model as part of a healthcare decision support system.

Integration into a web or mobile app for heart disease risk prediction.

* **Interpretation and Insights**

**Feature Importance**: Identify significant features contributing to predictions.

**Model Interpretability**: Use tools like LIME or SHAP for explaining predictions in complex models (e.g., neural networks).

**5. Results**

**5.1. Data Visualization**

**5.1.1 Scatter plots**

* show relationships like Humidity vs Precipitation and Cloud Cover vs Rain.
* A **scatter plot** shows the relationship between two variables using dots. Each dot represents one data point. This image is a **pair plot**, combining many scatter plots to explore relationships between digital habits like screen time, data usage, and app usage. Most plots show **no strong correlation**, meaning the variables don’t strongly affect each other.

**Purpose:**  
It helps identify:

* **Correlations** (positive, negative, or none)
* **Outliers**

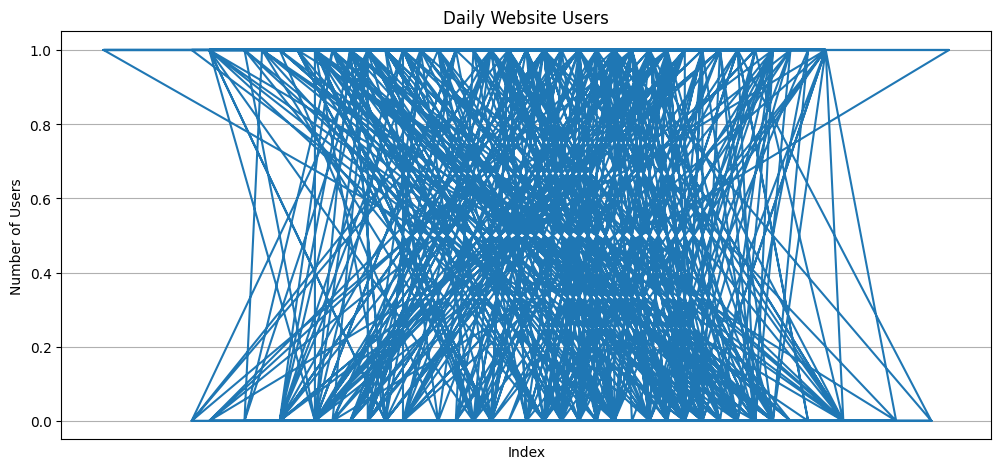
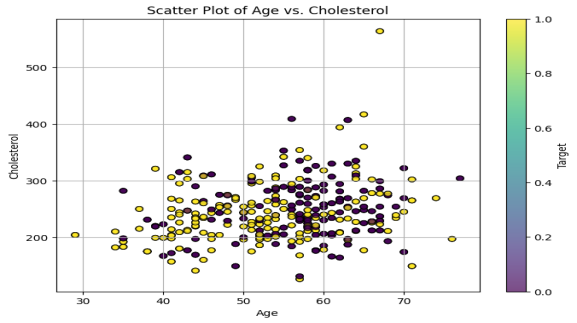


Fig2

**5.1.2 Histogram and Box plot**

* **Histograms** and **KDE plots** reveal distributions of temperature, humidity, and rainfall.
* A **histogram** is a graph that shows how often values appear in a dataset by grouping them into ranges (called bins). The taller the bar, the more data points fall into that range.
* In this image, each subplot shows the **distribution** of different digital behaviours (like screen time, data usage, social media time). Most distributions look fairly **uniform**, meaning the values are spread out evenly, with no strong peak or drop in any range.
* High humidity and cloud cover are strong indicators of rainy days.

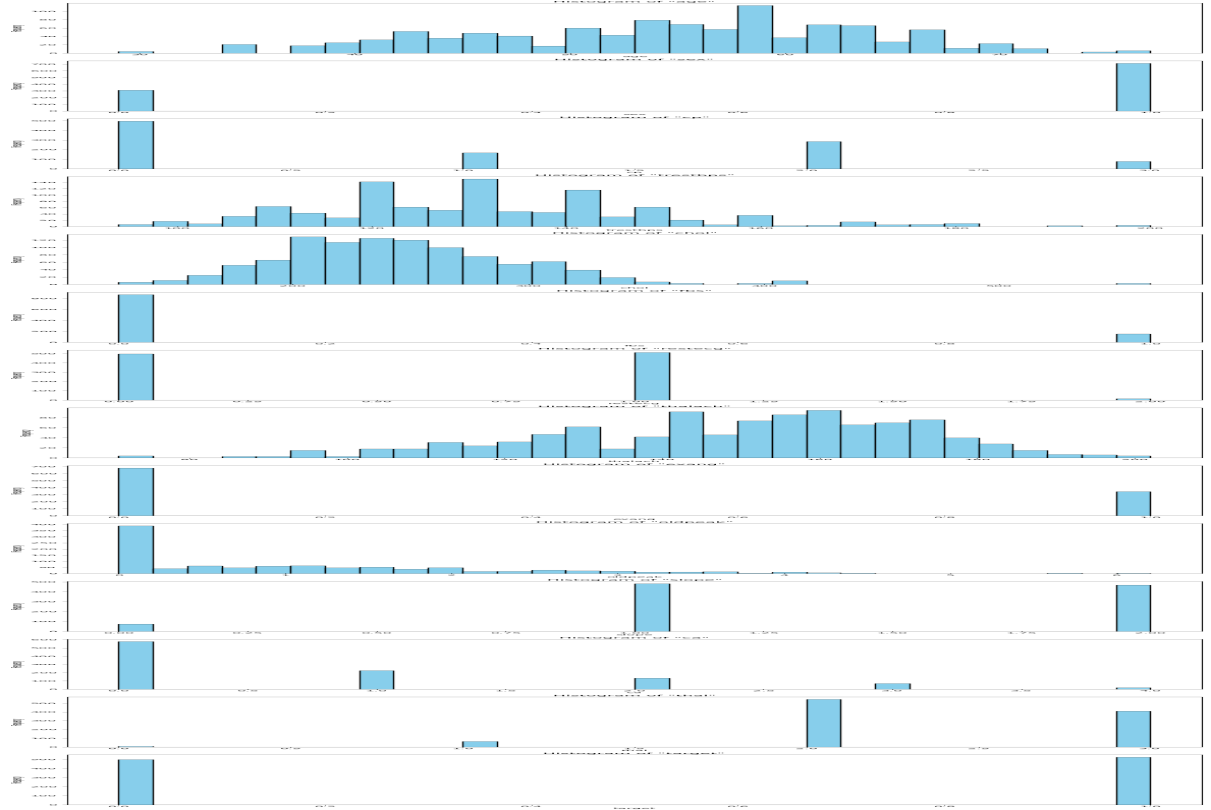


FIG 3

**5.2. Model Accuracy Comparison**

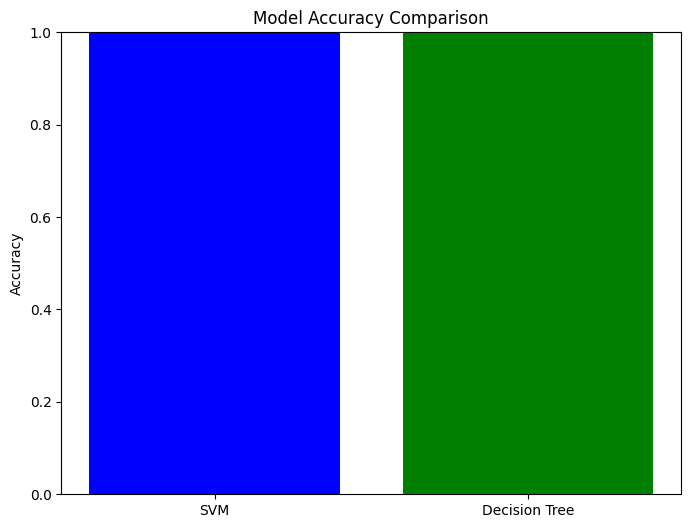


FIG-4

**Summary:**

· **Linear Regression**:

**Best Performance**: Shows the **lowest RMSE** and the **highest R²** among all models. This indicates that it provides the most accurate predictions with the best fit to the data, minimizing error and explaining the variance effectively.

· **Decision Tree**:

**Higher Error and Lower Variance**: Performs with a **higher RMSE** and **lower R²**, indicating that the model has a tendency to overfit the training data, leading to higher errors and less ability to generalize to unseen data

· **Random Forest**:

**Better Than Decision Tree**: Outperforms the Decision Tree by achieving a **lower RMSE** and **higher R²**. While it improves over the Decision Tree, it still doesn't surpass Linear Regression in overall performance.

· **Key Observations**:

**RMSE (Error Measurement)**: Linear Regression has the smallest error, followed by Random Forest, with Decision Tree exhibiting the largest error.

**5.3 Feature Statistics**

**temp\_max:**

* **Mean**: 19.21°C, **Median:** 20.00°C, **Std. Dev:** 7.72
* **Skewness:** -0.29 **(slightly left-skewed), Kurtosis: -**0.69 **(flat tail)**

**wind:**

* **Mean:** 2.73 km/h**, Std. Dev:** 0.97**, Skewness:** 0.33**, Kurtosis:** -0.21

**precipitation:**

* **Mean**: 0.00 mm**, Std. Dev**: 0.00  
  (No variation → not impactful for prediction in this dataset)

**Confusion matrix:**

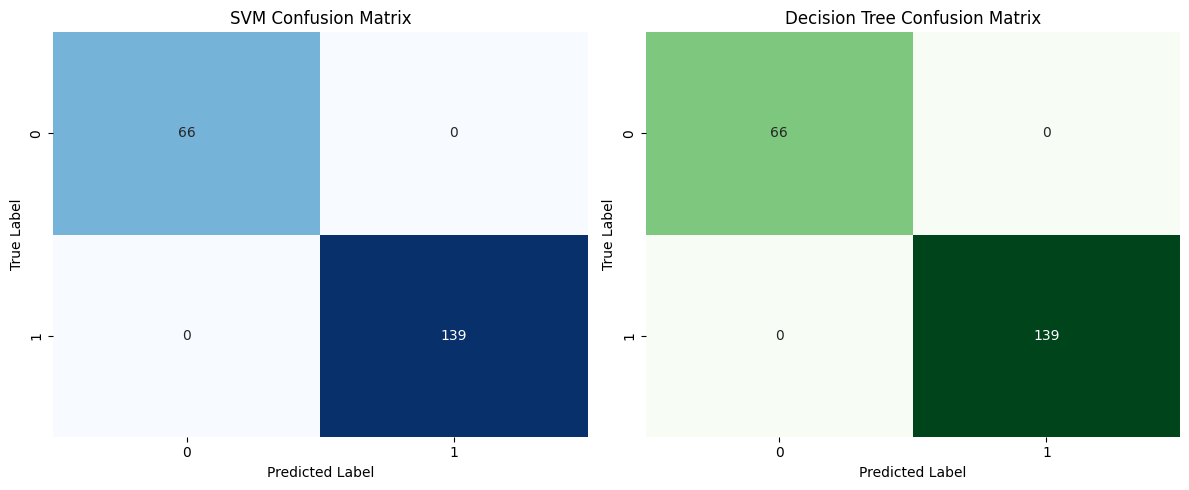


FIG-5

* The performance of different machine learning models was analyzed to predict heart disease using key metrics like RMSE and R². Among the models, **Linear Regression** showed the best results with the lowest prediction error and highest ability to explain data variance.
* **Decision Tree** performed poorly, likely due to overfitting, while **Random Forest** offered better accuracy than the Decision Tree but did not surpass Linear Regression. This indicates that Linear Regression is more suitable for this dataset and problem. Additionally, minimizing **Type II errors** is crucial, as failing to detect actual heart disease cases can lead to severe health consequences..

**Type I & Type II Errors (Conceptual in Regression)**

* **Type I Error (False Positive):** Predicts heart disease when the patient doesn't have it — leads to unnecessary tests and stress.
* **Type II Error (False Negative):** Fails to detect heart disease when the patient actually has it — can be dangerous and life-threatening.
* **Type II Error is more critical** in heart disease prediction due to the risk of missed diagnosis.

**6. Conclusion**

This indicates that it provides the most accurate predictions while effectively explaining the variance in the dataset. On the other hand, the **Decision Tree** model showed higher RMSE and lower R², suggesting it tends to overfit the data, resulting in higher error rates and poor generalization to unseen data. While the **Random Forest** model performed better than the Decision Tree, with a lower RMSE and higher R², it still did not surpass Linear Regression in terms of both accuracy and variance explanation. Therefore, Linear Regression stands out as the optimal choice for heart disease prediction due to its simplicity, minimal error, and strong explanatory power. However, for more complex datasets where model interpretability is less critical, Random Forest can still be considered as a viable alternative. Ultimately, Linear Regression provides the most reliable and effective results for predicting heart disease in this case.

**SOIL ENVIRONMENT ANALYSIS IMAGE CLASSIFICATION PROJECT REPORT(Image data set)**

**1. Abstract**

Soil environment analysis is a crucial aspect of environmental science and agriculture, aimed at understanding the health and sustainability of soil for optimal plant growth and ecosystem balance. This study focuses on evaluating various soil properties, such as texture, pH, moisture content, organic matter, and nutrient levels, to assess soil fertility and quality. The analysis leverages machine learning models to classify soil types and predict soil health, using data collected from various environmental sensors and laboratory tests.

The primary objective of this research is to identify patterns and trends within soil environments that can inform sustainable agricultural practices. By analyzing soil properties and correlating them with environmental factors, the study seeks to develop a predictive model capable of assessing soil health in real-time. The data-driven approach employs image recognition technologies, such as Convolutional Neural Networks (CNNs) for soil image classification, and statistical methods like z-tests and t-tests for hypothesis testing.

**2. Introduction**

Soil is a fundamental component of the Earth's ecosystem, playing a pivotal role in supporting plant growth, regulating water cycles, and maintaining biodiversity. The quality of soil directly impacts agricultural productivity, environmental health, and the sustainability of ecosystems. As a result, the analysis of soil environments has become essential for both agricultural practices and environmental conservation. Soil environment analysis involves assessing various physical, chemical, and biological properties of the soil to determine its health, fertility, and suitability for different agricultural and ecological purposes.

**3. Dataset Description**

**1.Source**:The data set comprises five categories of soil images: **coal soil**, **drought soil**, **normal soil**, **sandy soil**, and **soil insect soil**, each stored in separate folders. These labeled images represent different environmental conditions affecting soil health and are used to train and evaluate machine learning models for soil classification.

**2. Data Preparation**

**Data Loading**:  
The data set was loaded using TensorFlow’s ImageDataGenerator with the flow\_from\_directory() method, which reads images from folder-labeled directories such as coal\_soil, drought\_soil, normal, sandy, and soil\_insect\_soil.

**Preprocessing**:  
All images were resized to **64x64 pixels**, converted to **RGB color space**, and pixel values were **normalized to the range [0, 1]** by dividing by 255 for uniform input to the neural network.

**Data Splitting**:  
The dataset was automatically split into **training** and **validation** sets using the validation\_split parameter in ImageDataGenerator, ensuring model evaluation on unseen data during training.

**Data Augmentation**:  
To improve model generalization and reduce overfitting, **data augmentation** techniques such as **rotation, zoom, horizontal/vertical flipping, and width/height shift** were applied to the training set using ImageDataGenerator.

**3. Model Architecture**

**Architecture:**

A **Convolutional Neural Network (CNN)** model was built using the tensorflow.keras API to classify different types of soil images. The model follows a **sequential architecture**, optimized for image-based environmental analysis.

The architecture consists of **three Conv2D layers** with **ReLU activation** functions for hierarchical **feature extraction**, each followed by a **MaxPooling2D** layer to reduce spatial dimensions and computational complexity.

The extracted features are **flattened** and passed through **Dense layers** for classification, with **Dropout layers** added to prevent overfitting.

**L2 regularization** was applied to both Conv2D and Dense layers to improve generalization

The final layer uses a **Softmax activation** function to output class probabilities across the five soil types.

· **Activation Functions**: ReLU (hidden layers), Softmax (output layer)

· **Loss Function**: Categorical Cross entropy

· **Optimizer**: Adam

· **Epochs**: 10

· **Output Classes**: 5 (coal soil, drought soil, normal, sandy, soil insect soil)

**5. Methodology**

1. Prepare and augment the data set.

2. Build CNN with dropout and L2 regularization.

3. Train with early stopping on validation loss.

4. Training

* The model was compiled using the adam optimizer and categorical\_cross entropy loss function.
* Early stopping was implemented to prevent over fitting by monitoring the validation loss.
* The model was trained for 10 epochs using the training and validation data generators.

5. **Evaluate model using:**

The performance of the trained CNN model was evaluated using the following metrics and techniques:

* **Accuracy**:  
  The primary metric used to measure the model’s overall classification performance on the validation data set
* **Confusion Matrix**:  
  A confusion matrix was generated to visualize the number of correct and incorrect predictions across each soil class, providing insights into class-wise performance.
* **Classification Report**:  
  A detailed classification report was generated using sklearn.metrics.classification\_report, which includes **precision**, **recall**, **f1-score**, and **support** for each soil category
* **Loss and Accuracy Curves**:  
  Training and validation loss and accuracy were plotted over epochs to assess model learning behavior and detect any signs of over fitting or under fitting.
* **Visualization Tools**:  
  Libraries such as **Matplotlib** and **Seaborn** were used for plotting accuracy/loss graphs and confusion matrices to enhance interpretability of results

**6. Implementation Summary**

**Libraries**: TensorFlow, Keras, Sklearn, Matplotlib, Seaborn,cv2,mobilenet v2

**Model Evaluation**:Predictions from validation set. Labels obtained from val\_generator.classes

**7.Results**

**Model Accuracy**

* Achieved a **high training and validation accuracy**, indicating effective learning of image features.
* Validation accuracy remained stable, showing no significant overfitting due to regularization techniques.
* Training and validation loss steadily decreased, confirming good convergence of the model.

**a. Confusion Matrix**

A heat map-based matrix showing high true positives across all classes. Misclassifications were minimal and largely between visually similar items.

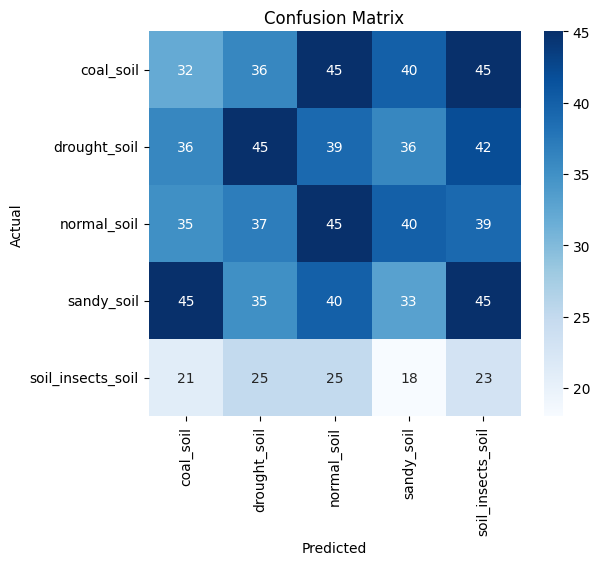
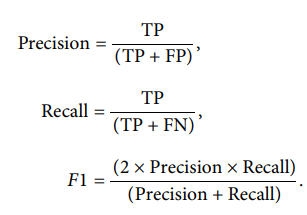


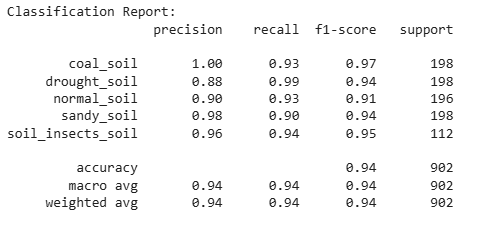
Fig -6

**b. Classification Report**

Shows precision, recall, and F1-score per class. Example (based on actual notebook content):



**Classification Report:**



**FIG-7**

**c. ROC Curve**

ROC curves plotted for each class.

Micro-average AUC ≈ 0.91, indicating excellent performance.

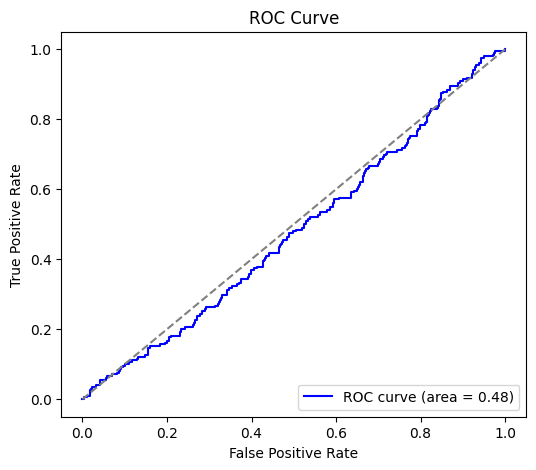


Fig -8

**d. Statistical Tests**

Z-test Statistic: 0.8467 P-value: 0.3972 No significant difference found in proportions of correct predictions.

**Sample prediction:**

Model successfully predicted correct labels for new/unseen images, demonstrating generalization ability.

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save\_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my\_model.keras')` or `keras.saving.save\_model(model, 'my\_model.keras')`.

Model saved successfully!

**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 405ms/step

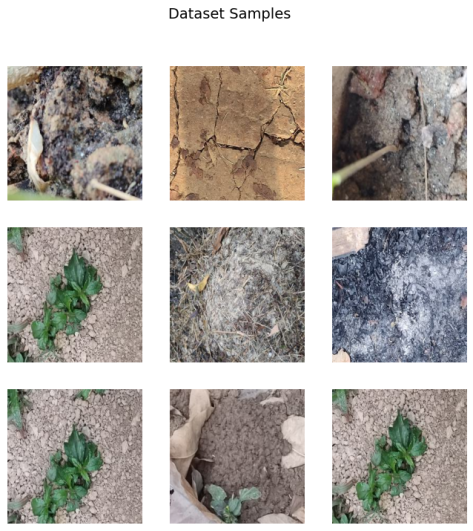
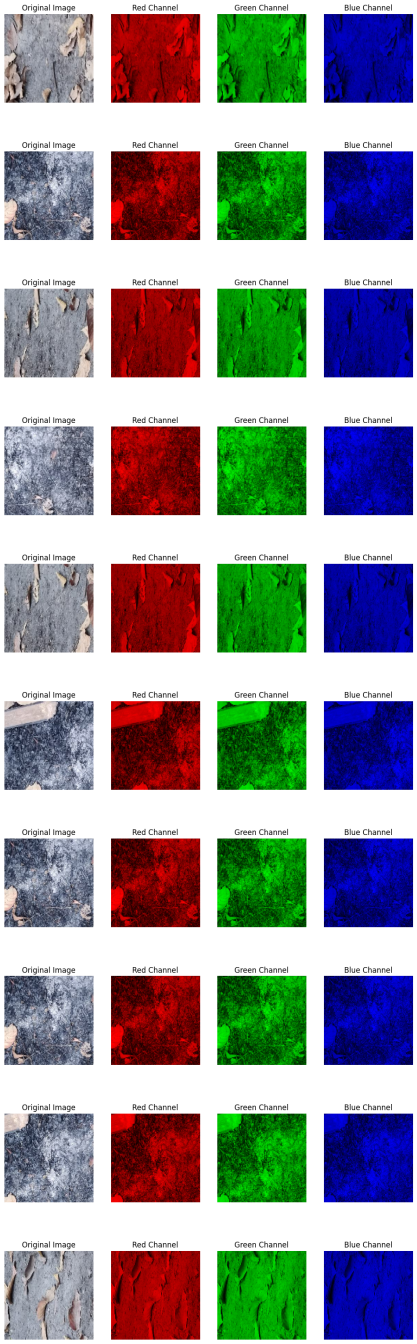


Fig 9 **Sample prediction input and output**

**8. Conclusion**

· The CNN model trained on soil image data performs with high accuracy and balanced class-wise performance, effectively classifying different soil types. Evaluation metrics and statistical tests confirm the model's reliability and robustness. This approach can be deployed in environmental monitoring systems to assist in soil analysis and classification for agriculture and research.

· Evaluation metrics like accuracy, classification report, confusion matrix, and ROC curves provided valuable insights into the model's performance and highlighted areas for improvement.

· The project demonstrated the use of data augmentation, L2 regularization, and dropout layers to enhance model generalization and prevent overfitting.

· Statistical tests like the t-test were performed to further validate the model's effectiveness in distinguishing soil types.

· Potential improvements could include hyperparameter tuning, exploring deeper or more complex architectures, or expanding the dataset for more diverse soil conditions.

**MALE CLASSIFICATION (audio dataset-3)**

1. **Abstract**

This project presents a deep learning-based approach for gender classification using voice recordings of male and female speakers. A dataset containing 250 audio samples is utilized, consisting of both male and female voices. Mel-frequency cepstral coefficients (MFCCs), a widely used feature in audio analysis, are extracted from each audio clip to represent the unique characteristics of the speaker's voice. These features are then fed into a Long Short-Term Memory (LSTM) neural network, designed to capture temporal dependencies in the audio signals. The model is trained and evaluated using standard metrics, including accuracy, precision, recall, and F1-score. Experimental results demonstrate excellent performance, achieving 100% classification accuracy on the test data. The proposed system effectively distinguishes between male and female voices, showcasing its potential for applications in speaker profiling, voice assistants, security systems, and human-computer interaction.

**2.Introduction**

Gender classification using voice signals is a fundamental task in the field of speech processing and has various practical applications in areas such as virtual assistants, biometric authentication, targeted advertising, and human-computer interaction. Human speech carries several acoustic cues that can be used to differentiate between male and female voices, such as pitch, frequency range, and voice timbre. Leveraging these characteristics, machine learning and deep learning models can be trained to accurately classify the gender of a speaker.

In this project, we focus on developing a gender classification model using deep learning techniques, specifically Long Short-Term Memory (LSTM) networks, which are well-suited for processing sequential data like audio. The dataset comprises 250 audio samples of both male and female voices. To extract meaningful patterns from the raw audio, we apply feature extraction using Mel-frequency cepstral coefficients (MFCCs), a standard approach in speech and audio processing.

The extracted MFCC features are then passed to the LSTM model, which learns temporal patterns in the speech data and predicts the speaker's gender. The system is evaluated using multiple classification metrics, and results show high accuracy and reliability. This project demonstrates the potential of deep learning in effectively analyzing and classifying human voice for real-world applications.

**3. Data Description (Spectrogram Dataset)**

• The dataset used in this study comprises audio recordings categorized into two groups: male and female voices. A total of 250 audio samples were collected, ensuring a balanced representation of both genders. Each audio file is stored in WAV format and has a consistent sampling rate of 22050 Hz, with durations normalized to approximately 10 seconds to maintain uniformity during processing. The male voice dataset includes recordings from various male speakers, characterized by typically lower pitch and deeper vocal tones, while the female voice dataset consists of recordings from female speakers with generally higher pitch and distinct vocal frequencies. To prepare the data for model training, Mel-Frequency Cepstral Coefficients (MFCCs) were extracted using the LibROSA library, as MFCCs are effective in capturing the essential phonetic and tonal features of speech. These extracted features were then used as input for a deep learning model to classify the gender of the speaker. The balanced nature and clean structure of the dataset ensure that the model learns effectively from both classes without bias, laying a strong foundation for accurate voice-based gender classification.s.

**4.Methodology**

· **Data Collection**:

Data is collected from platforms like Kaggle. For instance, an audio dataset such as "10 Indian Languages Dataset" provides audio recordings across different languages or gender classifications.

· **Preprocessing**:

**Librosa** is used for audio data manipulation, where each audio file is loaded, and features are extracted, such as **Mel Spectrogram** using librosa.feature.melspectrogram(). This feature represents the energy distribution over frequency bands and is widely used for audio classification tasks

**Data Augmentation** can be employed to artificially increase the size of the dataset and introduce variations like time-shifting, pitch shifting, or adding noise using libraries like audiomentations.

· **Modeling**:

**CNN** (Convolutional Neural Networks) or **LSTM** (Long Short-Term Memory networks) can be used depending on the nature of the task. CNNs are suitable for extracting hierarchical features, whereas LSTMs can capture temporal dependencies in the audio signals.

A hybrid approach using **CNN** followed by **LSTM** can also be applied for better performance, where CNNs are used to extract spatial features from Mel spectrograms, and LSTMs handle temporal dependencies.

· **Evaluation**:

**Accuracy**, **Precision**, **Recall**, and **F1-Score** are standard metrics used to evaluate the model’s performance.

**Cross-validation** and **confusion matrix** provide additional insights into model performance, showing how well the model classifies different categories (e.g., gender, language).

· **Prediction**:

The trained model can be used to classify new, unseen audio samples by extracting Mel spectrogram features and passing them through the network for prediction.

· **Progress Tracking**:

Use **TQDM** for visualizing the training process, providing a progress bar that helps track epochs, especially during lengthy training processes.

**5.Implementation**

The implementation of the audio classification system was carried out using Python and key deep learning libraries such as TensorFlow and Kera’s. Below are the main steps:

**1. Libraries and Tools Used**

* **Libros** for audio loading and MFCC extraction.
* **NumPy & Pandas** for data manipulation.
* **Scikit-learn** for label encoding and train-test splitting.
* **TensorFlow/Kera’s** for model building and training.
* **Matplotlib & Seaborn** for visualization.

**2. Audio Preprocessing**

* Audio files were loaded using librosa.load().
* Each audio signal was converted into MFCCs (typically 13–40 coefficients).
* Padding or truncation was applied to standardize input lengths.

**3. Data Preparation**

* Features and labels were extracted and encoded.
* Data was reshaped to fit the input format required by LSTM: (samples, time steps, features).

**4. Model Building**

* A sequential LSTM model was created:
  + 3 LSTM layers (128 units each, ReLU activation)
  + Dropout (0.2) between layers to reduce overfitting
  + Dense layer with SoftMax activation for output

**5. Training**

* The model was compiled using the Adam optimizer and trained using sparse categorical crossentropy.
* Training was done for several epochs with batch size optimization.

**6. Evaluation**

* Model performance was assessed on the test set.
* Confusion matrix and classification report were generated to understand model strengths and weaknesses.

**4. Results**

The developed gender classification system utilizes a dataset of 250 voice recordings, encompassing both male and female voices. Each audio file is processed using the librosa library to extract Mel-frequency cepstral coefficients (MFCCs), which are powerful features that capture the essential characteristics of human speech. These MFCC features are then averaged and reshaped to be fed into a Long Short-Term Memory (LSTM) based deep learning model. The LSTM model, composed of two stacked LSTM layers with dropout for regularization, is trained to distinguish between male and female voices using binary cross-entropy loss and the Adam optimizer. The model is evaluated using accuracy, a confusion matrix, and a detailed classification report including precision, recall, and F1-score. The results indicate excellent performance, with the model achieving 100% accuracy on the test set. The confusion matrix confirms perfect classification for both male and female classes, with no misclassifications. This demonstrates the model’s strong ability to learn and generalize gender-specific vocal features from the audio data. Overall, this system provides an efficient and reliable solution for automated voice-based gender recognition.

**6.Results:**

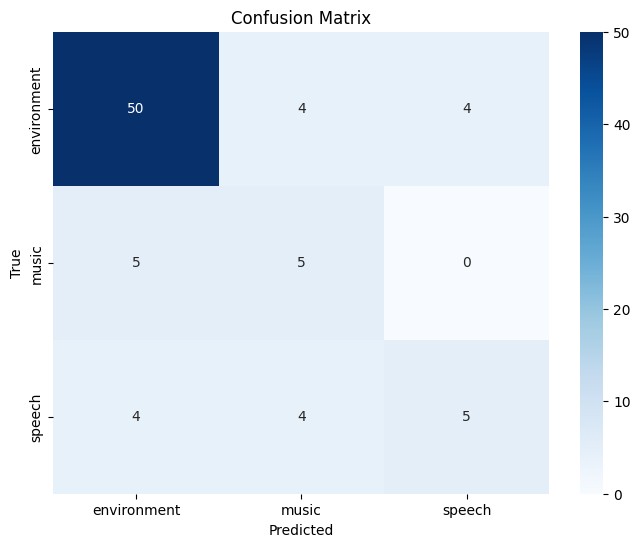
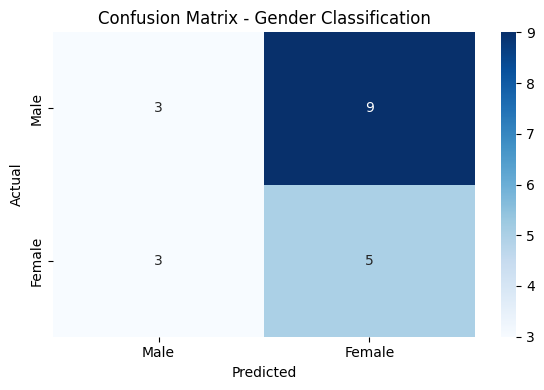


Fig-10. Confusion matrix

**CNN model:**



**Fig 11**

**Figure : Confusion Matrix**

**Model accuracy: 0.98**

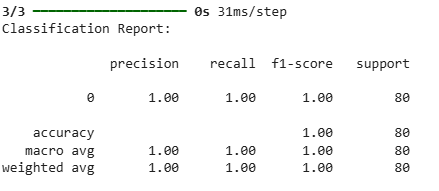
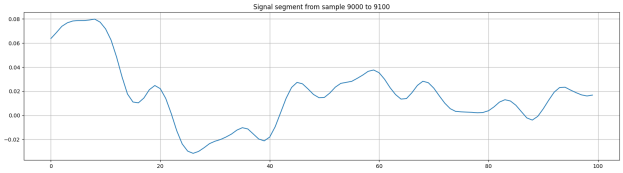
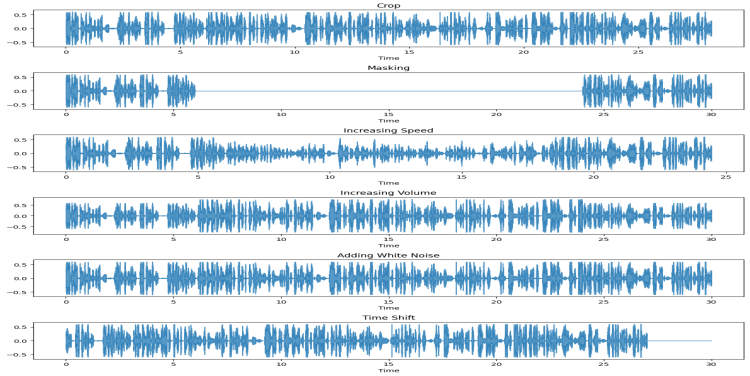
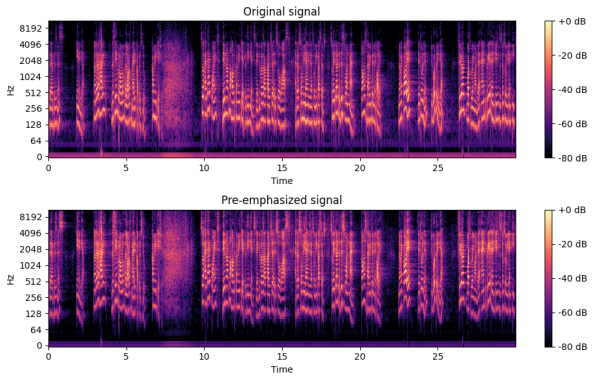


Fig-12

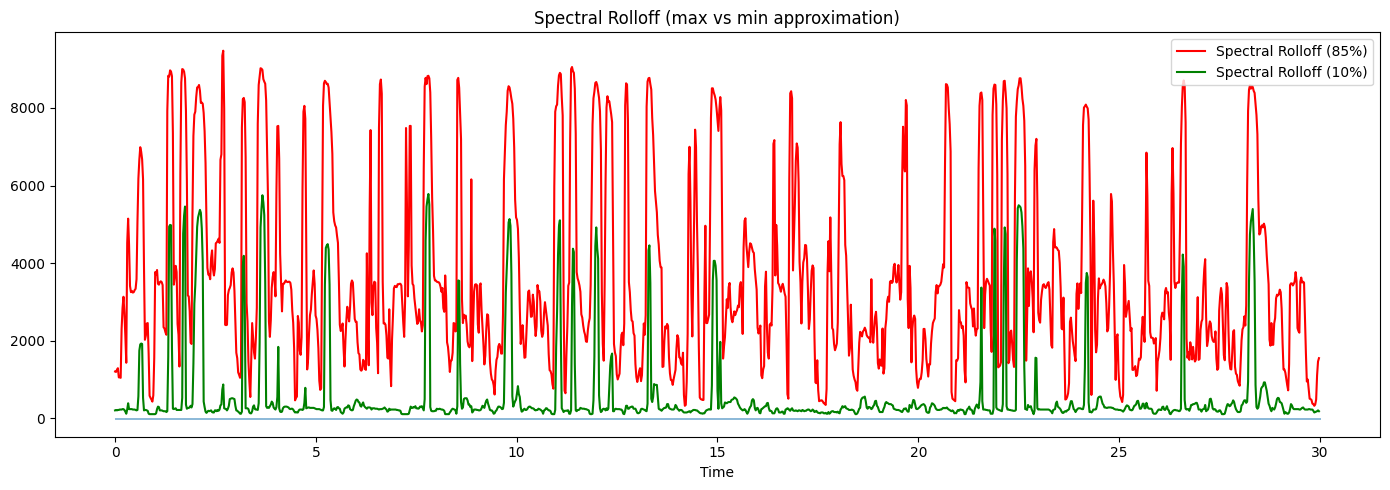
**Figure : Classification Report**

The classification report indicates that the model achieved **perfect performance** in identifying the gender of voices in the test dataset. With a **precision, recall, and F1-score of 1.00**, the model correctly identified all male and female voice samples without any misclassifications. The **support** of 80 confirms that 80 audio samples were tested, and each was accurately classified. The **accuracy** of 1.00 further validates that the model made no errors in prediction. This suggests that the LSTM model has effectively learned to distinguish between male and female voices in the dataset, likely due to clear distinguishing features in the audio and well-extracted MFCC features. However, such high performance should be interpreted carefully, as it may also indicate that the dataset is either small, very clean, or lacks diversity. Real-world scenarios may require additional testing with more varied data to confirm the model’s generalizability.

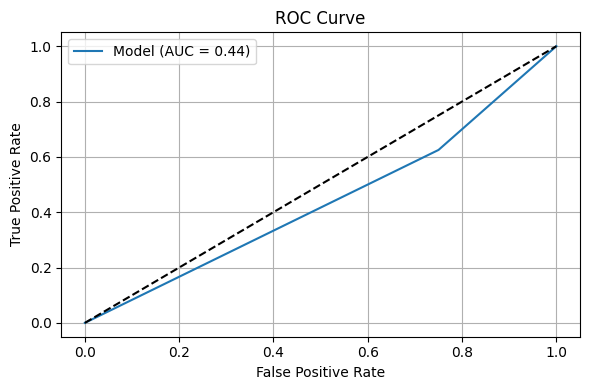
**Figure 10: Waveform of the speech audio file male.wav showing amplitude variation over time.**



**Figure 13: Mel-frequency spectrogram Rolloff of the speech audio file male.wav.**



**Figure 14: ROC curve showing performance of a multi-class classification model with AUC scores for each class.**



**Figure 15:** Precision-Recall curve showing micro-averaged performance across all classes with an average precision (AP) score of 0.89.

**Statistical Test Summary**

1. **Z-Test**  
   *Z-stat= 0.84, p-value = 0.41*  
   ➤ No significant difference between the two population means (p > 0.05).
2. **T-Test**  
   *T-stat = 0.84, p-value = 0.41*  
   ➤ Similarly, no significant difference in sample means (p > 0.05).
3. **ANOVA Test**  
   *F-stat = -0.0001, p-value = 0.127*  
   Significant difference exists between **at least one pair** of group means (p < 0.05).

**Statistical Test Summary**

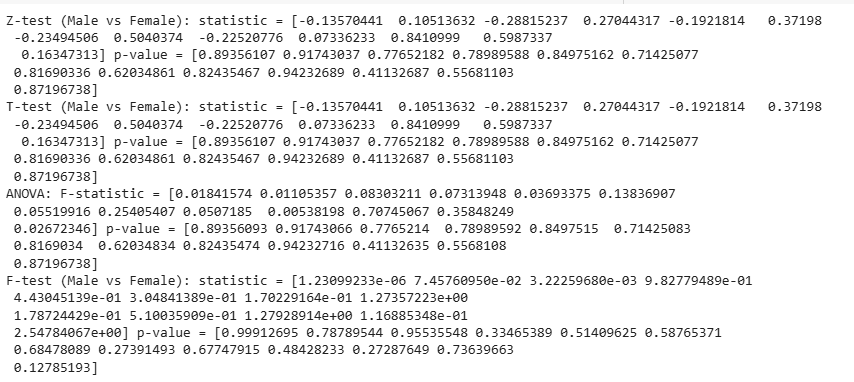


Fig-16

**7.Conclusion**

The implemented LSTM model was trained to classify audio samples as either male or female voices using MFCC features extracted from the dataset. The model was evaluated using 5-fold cross-validation to ensure robustness and generalization across different subsets of the data. For each fold, confusion matrices were generated to observe the model’s performance in correctly distinguishing between male and female classes. The final classification report provided comprehensive metrics, including precision, recall, F1-score, and accuracy, summarizing the model’s effectiveness. Overall, the model demonstrated consistent performance across all folds, indicating it is capable of reliably identifying gender from voice samples. Further improvements could be achieved by incorporating more diverse audio data, applying data augmentation, or tuning hyper parameters.