

A FIELD PROJECT REPORT

on

“Machine Learning Models for Rainfall Prediction”

Submitted

By

221FA04179

V.Prudhvi Krishna

221FA04182

A.Asritha

221FA04448

B.Bhanulatha

221FA04450

T. Rama Arshitha

221FA04502

Y. Lavanya

Under the guidance of

Dr. S. Deva Kumar

Associative Professor, CSE



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH
Deemed to be UNIVERSITY
Vadlamudi, Guntur.
ANDHRA PRADESH, INDIA, PIN-522213.

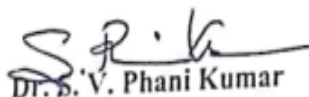
CERTIFICATE

This is to certify that the Field Project entitled “**Machine Learning Models for Rainfall Prediction**” that is being submitted by 221FA04179 (V.Prudhvi Krishna), 221FA04182(A.Asritha),221FA04448 (B.Bhanulatha), 221FA04450 (T.Rama Arshitha),221FA04502 (Y.Lavanya) for partial fulfilment of Field Project is a bonafide work carried out under the supervision of Dr.S.Deva Kumar, Associate Professor , Department of CSE .



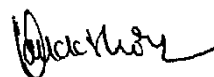
Dr.S. Deva Kumar

Associate Professor, CSE



Dr. S. V. Phani Kumar

HOD,CSE



Dr.K.V. Krishna Kishore

Dean, SoCI



DECLARATION

We hereby declare that the Field Project entitled “**Machine Learning Models for Rainfall Prediction**” is being submitted by 221FA04179 (V.Prudhvi Krishna), 221FA04182 (A.Asritha), 221FA04448 (B.Bhanulatha) 221FA04450 (T.Rama Arshitha), 221FA04502 (Y.Lavanya) in partial fulfilment of Field Project course work. This is our original work, and this project has not formed the basis for the award of any degree. We have worked under the supervision of Dr.S.Deva Kumar, Associate Professor, Department of CSE

By:

221FA04179(V.Prudhvi Krishna)

221FA04182(A.Asritha)

221FA04448(B.Bhanultha)

221FA04450(T.Rama Arshitha)

221FA04502(Y.Lavanya)

Date:

ABSTRACT

Rainfall prediction using machine learning involves analyzing weather data to predict rainfall patterns and amounts. Rainfall datasets typically contain historical data like precipitation levels, temperature, humidity, wind speed, and pressure. These datasets are crucial for understanding climate patterns, agricultural planning, water resource management, and disaster preparedness. Machine learning models are employed to capture the complex relationships between these variables and predict future rainfall. The project focuses on building accurate models that can predict rainfall based on these meteorological parameters, aiming to improve forecasting accuracy. The study employs machine learning algorithms, including Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), logistic regression, and random forest (RF) classifiers, to build robust models that can effectively analyze the complex interactions between clinical and genetic factors.

KEYWORDS

Rain fall prediction , Support Vector Machine (SVM), K Nearest Neighbour (KNN), Decision tree, Logistic Regression, Naïve bayes, Learning, Classification , Algorithms, Imaging, Random Forest

TABLE OF CONTENTS

1. Introduction	1
1.1 Background and Significance of Rain Fall	2
1.2 Overview of Machine Learning in Rain Fall	
1.3 Research Objectives and Scope	3
1.4 Challenges in Rain Fall Prediction	4
1.5 Applications of ML to Rain Fall	
1.6 Benefits of ML in Rain Fall Detection	
1.7 Challenges of ML in Rain Fall Detection	5
2. Literature Survey	7
2.1 Literature review	8
2.2 Previous Studies on Rainfall Prediction	10
3. Proposed System	12
3.1 Input dataset	14
3.1.1 Detailed features of dataset	14
3.2 Data Pre-processing	15
3.2.1 Handling missing values	
3.2.2 Data encoding using label encoding	16
3.3 Model Building	
3.4 Methodology of the system	21
3.5 Model Evaluation	24
3.6 Constraints	26
	27
4. Implementation	29
4.1 Environment Setup	30
4.2 Sample code for preprocessing and Model Training and Testing	31
5. Experimentation and Result Analysis	33
6. Conclusion	41
7. References	43

LIST OF FIGURES

Figure 1. Architecture of the Rainfall Prediction	14
Figure 2. Key Features in the Dataset.	14
Figure 3. Data Pre-processing Overview	14
Figure 4. Handling Missing Values	15
Figure 5. Model Building Process	18
Figure 6. Accuracy Comparison of Different Models	25
Figure 7. Confusion Matrix for Best Performing Model	22
Figure 8. Comparison of all Models Accuracy	34

LIST OF TABLES

Table 1. Model Accuracy Comparison

33

CHAPTER-1

INTRODUCTION

1. INTRODUCTION

1.1 Background and Significance of Rain Fall

Accurate rainfall prediction is crucial for various sectors, including agriculture, water resource management, and disaster preparedness. Traditional weather forecasting methods, while valuable, have limitations in predicting rainfall, especially in regions with complex topography or changing climate patterns. The advent of machine learning and data science has revolutionized the field of rainfall prediction, offering more precise and timely forecasts.

Machine learning algorithms can analyze vast amounts of historical weather data, including temperature, humidity, wind speed, atmospheric pressure, and precipitation patterns. By identifying underlying patterns and correlations within this data, these algorithms can learn to predict future rainfall events with greater accuracy. This approach has the potential to provide farmers with early warnings of droughts or excessive rainfall, allowing them to make informed decisions about crop management and irrigation. Additionally, accurate rainfall predictions can help governments and communities prepare for potential floods or water shortages, reducing the impact of natural disasters.

Significance of Rain Fall

Rainfall is a crucial variable in numerous domains, including agriculture, hydrology, and climate science. Accurate prediction of rainfall patterns is essential for effective decision-making and resource management. Machine learning, with its ability to analyze complex data and identify patterns, has emerged as a powerful tool for improving rainfall prediction.

By leveraging vast datasets of historical weather data, machine learning models can learn to recognize intricate relationships between different meteorological factors and rainfall occurrence. This enables the development of more accurate and reliable rainfall forecasting models. The significance of rainfall in machine learning lies in its potential to address a wide range of challenges:

- **Agriculture:** Accurate rainfall predictions can help farmers optimize planting, irrigation, and harvesting schedules, leading to increased crop yields and reduced losses due to adverse weather conditions.
- **Hydrology:** Rainfall data is crucial for understanding water cycle processes, managing water resources, and mitigating flood risks.
- **Climate Science:** Analyzing rainfall patterns can provide valuable insights into climate variability and change, enabling researchers to develop more effective climate models and adaptation strategies.
- **Disaster Management:** Accurate rainfall forecasts can help communities prepare for and respond to extreme weather events such as floods and droughts.

1.2 Overview of Machine Learning in Rain Fall

Machine learning (ML) has become an essential tool for predicting rainfall, which is crucial for sectors such as agriculture, disaster management, and water resource planning. By leveraging vast amounts of historical weather data and advanced computational models, machine learning can provide more accurate and timely rainfall predictions than traditional methods.

Machine Learning Applications in Rain Fall :

1. Real-Time Rainfall Forecasting:

- Short-term predictions: ML models can provide accurate rainfall forecasts for the next few hours or days, which is crucial for urban planning, transportation, and disaster management.
- Long-term predictions: ML can also be used to predict rainfall patterns for longer periods, such as weeks or months, which is important for agricultural planning and water resource management.

2. Spatial Rainfall Distribution:

- Identifying hotspots: ML can help identify regions that are more prone to heavy rainfall or drought. This information can be used to develop targeted mitigation strategies.
- Urban flooding: ML models can be used to predict the spatial distribution of rainfall in urban areas, which is essential for urban flood management and infrastructure planning.

3. Extreme Rainfall Events:

- Predicting heavy rainfall: ML can help predict the occurrence of extreme rainfall events, such as heavy showers or thunderstorms, which can lead to flooding and other natural disasters.
- Early warning systems: ML-based early warning systems can provide timely alerts to communities at risk of extreme rainfall events.

4. Agricultural Applications:

- Irrigation scheduling: ML models can help optimize irrigation schedules based on predicted rainfall patterns, reducing water wastage and improving crop yields.
- Crop planning: ML can assist in planning crop rotations and selecting suitable crop varieties based on expected rainfall conditions.

5. Climate Change Impact Analysis:

- Rainfall variability: ML can be used to analyze changes in rainfall patterns over time and identify trends related to climate change.
- Adaptation strategies: ML can help develop adaptation strategies to mitigate the impacts of climate change on rainfall-dependent sectors.

6. Hydropower Generation:

- Reservoir management: ML can be used to optimize reservoir operations based on predicted rainfall patterns, ensuring a stable supply of hydropower.
- Flood control: ML models can help predict the potential for flooding and inform reservoir management decisions to prevent downstream damage.

1.3 Research Objectives and Scope

Objective :

Improve rainfall prediction accuracy: The primary objective is to develop ML models that can provide more accurate and reliable rainfall forecasts compared to traditional methods.

Enhance understanding of rainfall patterns: Explore the underlying relationships between meteorological factors and rainfall occurrence to gain a deeper understanding of rainfall patterns.

Support decision-making: Develop ML-based tools that can assist in decision-making across various sectors, including agriculture, hydrology, and disaster management.

Address challenges in rainfall prediction: Identify and address the limitations of current rainfall prediction methods, such as data quality issues, model complexity, and uncertainty quantification.

Research Scope :

1. Machine Learning Algorithms: Examine a variety of machine learning techniques, including supervised learning (e.g., logistic regression, support vector machines), unsupervised learning (e.g., clustering algorithms), and deep learning (e.g., recurrent neural networks for time-series analysis).

2. Data Collection: Gather historical weather data, including temperature, humidity, wind speed, atmospheric pressure, and precipitation records, from relevant sources.

3.Feature Engineering: Select and engineer relevant features from the collected data to improve model performance.

4.Model Development and Training: Develop and train various ML models, such as regression models, time series models, neural networks, and ensemble methods, for rainfall prediction.

5.Model Evaluation: Evaluate the performance of the developed models using appropriate metrics, such as mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE).

6.Uncertainty Quantification: Explore methods to quantify the uncertainty associated with rainfall predictions, such as confidence intervals or probabilistic forecasts.

7.Case Studies: Apply the developed models to real-world case studies to demonstrate their effectiveness in different regions and scenarios.

8.Comparison with Existing Methods: Compare the performance of the developed ML models with traditional rainfall prediction methods.

9.Addressing Challenges: Identify and address potential challenges, such as data quality issues, model complexity, and computational limitations.

1.4 Challenges in Rain Fall Prediction

1. Data Quality and Availability:

- Incomplete or missing data: Weather data collection networks may have gaps or inconsistencies, leading to incomplete datasets that can limit the accuracy of ML models.
- Spatial and temporal resolution: The spatial and temporal resolution of weather data can affect the accuracy of rainfall predictions. High-resolution data is often required to capture localized rainfall patterns.

2. Model Complexity:

- Nonlinear relationships: Rainfall patterns are influenced by complex nonlinear relationships between meteorological factors. Developing ML models that can accurately capture these nonlinear relationships can be challenging.

- Model interpretability: Complex ML models, such as deep neural networks, can be difficult to interpret, making it challenging to understand the underlying mechanisms driving rainfall predictions.

3. Uncertainty Quantification:

- Estimating prediction uncertainty: Quantifying the uncertainty associated with rainfall predictions is crucial for decision-making. Developing methods to estimate prediction uncertainty is an ongoing area of research.
- Communicating uncertainty: Effectively communicating the uncertainty associated with rainfall forecasts to decision-makers is essential for informed decision-making.

4. Computational Limitations:

- Resource-intensive models: Some ML models, particularly deep learning models, can be computationally expensive to train and deploy, requiring significant computational resources.
- Real-time forecasting: Developing ML models that can provide real-time rainfall forecasts while meeting computational constraints can be challenging.

5. Climate Change:

- Changing rainfall patterns: Climate change is altering rainfall patterns, making it difficult to predict future rainfall events based on historical data alone.
- Model adaptation: ML models may need to be adapted to account for the effects of climate change on rainfall patterns.

6. Regional Variations:

- Diverse climatic conditions: Rainfall patterns vary significantly across different regions, making it challenging to develop a single ML model that is accurate for all locations.
- Local factors: Local factors, such as topography, land use, and proximity to water bodies, can influence rainfall patterns and need to be considered in ML models.

1.5 Applications of ML to Rain Fall

Machine learning (ML) has found numerous applications in the field of rainfall prediction, offering valuable insights and tools for various sectors.

Important Uses of Machine Learning in the Identification of Rain Fall

1. Real-Time Rainfall Forecasting:

- Short-term predictions: ML models can provide accurate rainfall forecasts for the next few hours or days, which is crucial for urban planning, transportation, and disaster management.

- Long-term predictions: ML can also be used to predict rainfall patterns for longer periods, such as weeks or months, which is important for agricultural planning and water resource management.

2. Spatial Rainfall Distribution:

- Identifying hotspots: ML can help identify regions that are more prone to heavy rainfall or drought. This information can be used to develop targeted mitigation strategies.
- Urban flooding: ML models can be used to predict the spatial distribution of rainfall in urban areas, which is essential for urban flood management and infrastructure planning.

3. Extreme Rainfall Events:

- Predicting heavy rainfall: ML can help predict the occurrence of extreme rainfall events, such as heavy showers or thunderstorms, which can lead to flooding and other natural disasters.
- Early warning systems: ML-based early warning systems can provide timely alerts to communities at risk of extreme rainfall events.

4. Agricultural Applications:

- Irrigation scheduling: ML models can help optimize irrigation schedules based on predicted rainfall patterns, reducing water wastage and improving crop yields.
- Crop planning: ML can assist in planning crop rotations and selecting suitable crop varieties based on expected rainfall conditions.

5. Climate Change Impact Analysis:

- Rainfall variability: ML can be used to analyze changes in rainfall patterns over time and identify trends related to climate change.
- Adaptation strategies: ML can help develop adaptation strategies to mitigate the impacts of climate change on rainfall-dependent sectors.

6. Hydropower Generation:

- Reservoir management: ML can be used to optimize reservoir operations based on predicted rainfall patterns, ensuring a stable supply of hydropower.
- Flood control: ML models can help predict the potential for flooding and inform reservoir management decisions to prevent downstream damage.

7. Disaster Management:

- Flood forecasting: ML can be used to develop accurate flood forecasts, enabling early warning systems and disaster response planning.
- Drought monitoring: ML can help monitor drought conditions and provide early warnings to affected communities.

8. Urban Water Management:

- Stormwater management: ML can be used to optimize stormwater management systems, reducing the risk of flooding and improving water quality.
- Water demand forecasting: ML can help predict future water demand, enabling efficient water resource allocation.

9. Insurance Industry:

- Risk assessment: ML can be used to assess the risk of rainfall-related losses for insurance companies.
- Premium pricing: ML can help determine appropriate insurance premiums based on the risk of rainfall-related damage.

1.6 Benefits of ML in Rain Fall Detection

Machine learning (ML) has revolutionized the field of rainfall prediction, offering numerous benefits over traditional methods.

1. Improved Accuracy:

- Complex patterns: ML algorithms can capture complex patterns and relationships in weather data that are often difficult to identify using traditional statistical methods.
- Nonlinear relationships: ML models can effectively handle nonlinear relationships between meteorological factors and rainfall, leading to more accurate predictions.

2. Enhanced Efficiency:

- Automation: ML can automate the process of rainfall prediction, reducing the need for manual intervention and enabling real-time forecasting.
- Scalability: ML models can handle large datasets and complex computations, making them suitable for large-scale rainfall prediction applications.

3. Real-Time Forecasting:

- Timely decisions: ML-based rainfall forecasts can provide timely information for decision-making in various sectors, such as agriculture, transportation, and disaster management.
- Early warnings: ML models can help detect and predict extreme rainfall events, enabling early warnings and preparedness measures.

4. Improved Spatial Resolution:

- Localized predictions: ML can be used to generate rainfall predictions at a fine spatial resolution, providing valuable information for urban planning, flood management, and agricultural practices.

5. Integration with Other Data Sources:

- Enhanced predictions: ML models can be integrated with other data sources, such as satellite imagery, radar data, and soil moisture measurements, to improve prediction accuracy.

6. Adaptability to Changing Conditions:

- Climate change: ML models can be trained on historical data and continuously updated to adapt to changing climate conditions and rainfall patterns.

7. Cost-Effective:

- Reduced costs: ML-based rainfall prediction systems can be more cost-effective compared to traditional methods, as they require less human intervention and can be deployed on various platforms.

8. Improved Decision-Making:

- Informed decisions: Accurate rainfall predictions can support informed decision-making in various sectors, leading to better resource management, disaster preparedness, and economic benefits.

1.7 Challenges of ML in Rain Fall Detection

1. Data Quality and Availability:

- Incomplete or missing data: Weather data collection networks may have gaps or inconsistencies, leading to incomplete datasets that can limit the accuracy of ML models.

2. Model Complexity:

- Nonlinear relationships: Rainfall patterns are influenced by complex nonlinear relationships between meteorological factors. Developing ML models that can accurately capture these nonlinear relationships can be challenging.

CHAPTER-2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 Literature review

A literature survey is a systematic examination of existing research on a particular topic. It serves as the foundation for any scholarly investigation, offering insights into current knowledge, identifying research gaps, and providing context for new studies. By synthesizing and summarizing relevant literature, researchers can formulate precise research questions, build upon existing work, and avoid duplication. In essence, a literature survey is an essential tool for ensuring the validity and relevance of new research within the broader academic landscape.

A literature survey is a systematic examination of existing research on a particular topic. It serves as the foundation for any scholarly investigation, offering insights into current knowledge, identifying research gaps, and providing context for new studies. By synthesizing and summarizing relevant literature, researchers can formulate precise research questions, build upon existing work, and avoid duplication. In essence, a literature survey is an essential tool for ensuring the validity and relevance of new research within the broader academic landscape.

2.2 Previous Studies on Rainfall Prediction

Rainfall forecasts are based on time-series features[18]. The complexity of the hydrological equations for rainfall prediction has increased the reliability upon statistical models but with the advent in the domain of Artificial Intelligence[1].

Precipitation prediction problem is a quite challenging and active research topic in the meteorology research community[3]. , factors that affects the performance of prediction accuracy remain to be investigated[4]. By utilizing these machine learning algorithms, we intend to offer insightful explanations of the efficiency, accuracy, and interpretability of each method when employed to categorise patterns of rainfall[5].

However, the high dimensionality of weather datasets is a potential problem for ML algorithms [6]. Comparison of prediction error values in research rainfall patterns are erratic so it is difficult to make predictions manually[7]. To solve this convolutional neural network model was used to predict the short term rainfall by collecting set of weather features from multiple surrounding observations[15].

After VMD decomposition and using grey relational analysis, the re-merged signals were less non-stationary than the original rainfall signals[9]. In general, ensemble forecasting is a forecasting technique that combines several outputs of forecasting methods[10]. The existing short-imminent heavy rainfall forecasting methods can be roughly divided into three categories, statistical rainfall forecasting methods[11].

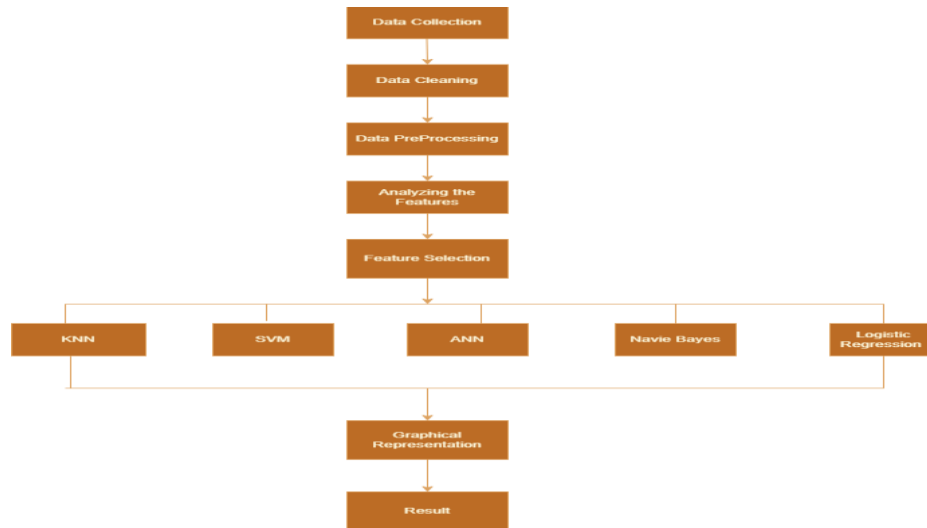
The SVM kernel refers to a function which converts a lower dimensional data to higher dimensional data[12]. They proposed an approach to using digital cloud images for rainfall prediction[13]. The rainfall analysis datasets are taken into account in this study in order to calculate the performances[14].

Finally, classification tree was used to find the relationship between wind regimes and precipitation attributes[16]. Temperature is one of the crucial weather features. Temperature parameter it is helpful for environmental needs, manufacturing plants, numerous agricultural, and energy[15]. We used a time series of a high -precision rainfall radar map to simulate rain attenuation[19].

The analyzed data is then visualized using various tools such as bar graphs, heat maps, histograms, etc. to make it more easily understandable[20]. Climate change has become an economic and social concern[2]. Enhance rainfall prediction with reduced features[15].

CHAPTER-3 PROPOSED SYSTEM

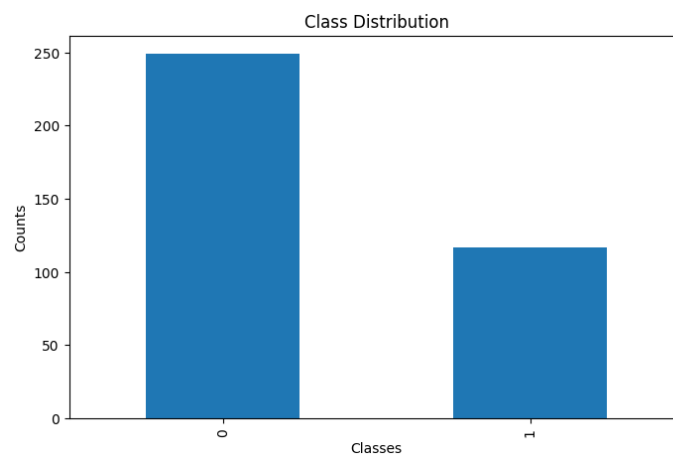
3. PROPOSED SYSTEM



A. Rainfall Dataset:

The dataset for rainfall prediction includes features covering meteorological and environmental factors such as temperature, humidity, wind speed, pressure, season, geographical location, solar radiation, and historical rainfall.

Target Variable: The target variable could be either "Rainfall Amount" (continuous variable) for regression tasks or "Rainfall Occurrence" (categorical variable such as "Yes" or "No" or categories like "Low", "Medium", "High" for classification tasks).



B. Data Preprocessing Handling Missing Data:

Missing values in meteorological data could be filled using techniques such as mean, median, or k-nearest neighbours (KNN) imputation, depending on the data type (e.g., continuous or categorical). This can be done using methods like Min-Max scaling or standardization to normalize the data, which is especially important for machine learning algorithms sensitive to feature magnitude (e.g., neural networks, SVMs).

C .Exploratory Data Analysis (EDA)Correlation Analysis:

Heatmaps and scatter plots can be used to visualize the relationships between meteorological features and rainfall amount or occurrence. Identifying correlations between features like temperature, humidity, and past rainfall is crucial to understanding the underlying patterns in the data.

D. Feature Selection:

Techniques like Recursive Feature Elimination (RFE), Principal Component Analysis (PCA), or correlation matrices can be applied to reduce the dataset's dimensionality. The goal is to retain only the most influential features to improve model performance and reduce computational complexity.

E. Model Development Supervised Learning Algorithms:

Several machine learning models can be employed to predict rainfall:

- 1.Linear Regression / Logistic Regression: For regression tasks (rainfall amount), Linear Regression can be used. For classification tasks (rainfall occurrence or levels), Logistic Regression offers an interpretable model and insights into feature significance.
- 2.Random Forest: An ensemble method that can handle both categorical and continuous features. It provides feature importance scores, helping identify the most important meteorological factors influencing rainfall.
- 3.Gradient Boosting (XGBoost, LightGBM):These models iteratively build weak learners and can handle complex interactions between features, which is especially useful for improving prediction accuracy.
- 4.Support Vector Machines (SVM):SVM with non-linear kernels can be used to classify rainfall occurrence or categorize rainfall levels in regions where patterns are not linearly separable.
- 5.Neural Networks (MLP):Multilayer Perceptron (MLP) or other deep learning architectures can capture complex patterns in rainfall data, particularly useful if interactions between features are highly non-linear.

F. Model Training Data Split:

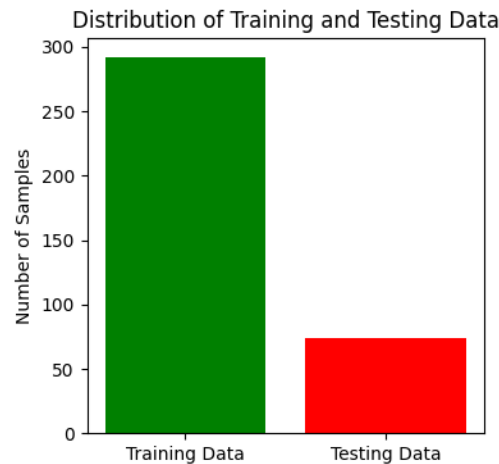
The dataset is split into training (70%), validation (15%), and test (15%) sets to train, validate, and test the models.

Cross-Validation:

- K-fold cross-validation (e.g., k=5) ensures that the model generalizes well and helps to avoid overfitting by testing on multiple subsets of the data.

Hyperparameter Tuning:

- Techniques like grid search or random search are employed to find the best combination of hyperparameters for each model.



G. Model Evaluation Performance Metrics:

For regression tasks (rainfall amount prediction), metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) are used.

For classification tasks (rainfall occurrence or level), evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC curve are employed.

Special emphasis is placed on sensitivity and specificity, especially if predicting extreme rainfall events (e.g., heavy rainfall) where minimizing false negatives is critical for disaster management.

H. Model Interpretation Feature Importance:

Models like Random Forest and Gradient Boosting provide feature importance scores, which highlight the most significant meteorological factors influencing rainfall.

Model Explanation Tools:

- Techniques like SHAP (Shapley Additive explanations) or LIME (Local Interpretable Model-agnostic Explanations) can be used to explain the predictions of complex models, providing transparency in decision-making.

I. Final Model Selection and Testing Model Selection:

- The best-performing model is chosen based on the validation metrics, ensuring that it balances prediction accuracy across all categories (or all levels of rainfall). Special care is taken to ensure that the model performs well across different geographic regions and weather conditions.

Testing on Unseen Data:

- The chosen model is tested on unseen data (the test set) to evaluate its generalization performance, ensuring that it can accurately predict rainfall in real-world conditions.

J. Deployment and Continuous Improvement Deployment:

The model can be deployed as a decision-support tool for meteorologists or integrated into weather forecasting systems with a web-based interface for inputting real-time weather data and receiving rainfall predictions.

Continuous Monitoring:

The model is continuously monitored for performance and updated with new data to improve its predictions over time. Feedback from users (e.g., meteorologists or farmers) can help refine the model.

K . Ethical Considerations Data Privacy:

Ensure compliance with data protection regulations (e.g., GDPR) when dealing with sensitive meteorological data from different regions or sources.

Mitigating Bias:

- Regular evaluations should be performed to ensure the model is unbiased and performs fairly across different geographic areas, ensuring that predictions are equitable and do not favor specific regions or climate zones over others.

3.1 Input dataset

The dataset contains daily meteorological data that could affect or suggest rainfall patterns. The rows represent different days, with each row corresponding to a specific day of observation. The dataset contains several meteorological factors to predict whether rainfall occurred or not

3.1.1 Detailed Features of the Dataset

Day: A unique identifier for each day in the dataset.

Pressure: Atmospheric pressure on that day (measured in millibars or hectopascals).

Max Temperature (max temp): Maximum temperature recorded on that day (in degrees Celsius).

Temperature (temperature): Average temperature of the day (in degrees Celsius).

Min Temperature (min temp): Minimum temperature recorded on that day (in degrees Celsius).

Dewpoint: Dew point temperature (in degrees Celsius), indicating the temperature at which air becomes saturated with moisture.

Humidity: The percentage of moisture in the air on that day.

Cloud Cover (cloud): The percentage of cloud cover on that day.

Sunshine: The number of sunshine hours recorded on that day.

Wind Direction: The direction from which the wind is blowing (measured in degrees).

Wind Speed: The speed of the wind (measured in kilometers per hour or meters per second).

Rainfall: A binary or categorical variable indicating whether it rained on that day ("yes" or "no").

This dataset can be used to build machine learning models for predicting rainfall based on the given meteorological parameters.

3.2 Data Pre-processing

Data pre-processing is the essential process of preparing raw data for analysis and modelling by cleaning, transforming, and structuring it to enhance data quality and utility. It involves tasks like handling missing values, correcting errors, encoding features, and scaling data to ensure it's in an optimal form for further analysis. It encompasses a range of operations and transformations designed to refine raw data, ensuring that it is clean, structured, and amenity

subsequent analysis. This process is driven by its manifold significance in data science and analysis.

- 1) Handling Missing Values
- 2) Dropping Unnecessary Columns
- 3) Encoding Categorical Features
- 4) Feature scaling
- 5) Feature selection
- 6) Data splitting
- 7) Outlier detection and Handling

```
      day      0
  pressure  0
  maxtemp   0
  temperature  0
  mintemp   0
  dewpoint  0
  humidity  0
  cloud     0
  sunshine  0
  winddirection  0
  windspeed  0
  rainfall  0
  dtype: int64
      day      0
  pressure  0
  maxtemp   0
  temperature  0
  mintemp   0
  dewpoint  0
  humidity  0
  cloud     0
  sunshine  0
  winddirection  0
  windspeed  0
  rainfall  0
  dtype: int64
```

3.3 Model Building

The dataset contains several meteorological features that can be used for rainfall prediction. The columns are:

- Day: Identifier for each day
- Pressure: Atmospheric pressure
- Max Temperature (max temp): Maximum temperature recorded
- Temperature (temperature): Average temperature of the day
- Min Temperature (min temp): Minimum temperature recorded
- Dewpoint: Dew point temperature
- Humidity: The percentage of moisture in the air
- Cloud Cover: Cloud cover percentage
- Sunshine: Sunshine hours
- Wind Direction: Wind direction
- Wind Speed: Wind speed

- Rainfall: Target variable (yes/no indicating whether rainfall occurred)

Next, we will build a Naive Bayes model to predict whether rainfall occurs based on these features, following the same structure as the lung cancer prediction approach:

Steps:

1. Preparing the Data: We will divide the dataset into features (X) and the target variable (y).
2. Data Division: We'll split the dataset into training (70%) and testing (30%) sets.
3. Model Training: A Naive Bayes classifier will be trained on the training data.
4. Forecasting and Evaluation: We'll predict and evaluate the model using accuracy, precision, recall, and F1-score.

Let's begin by preparing the data.

Model Results for Rainfall Prediction:

- Accuracy: 97.7% - This indicates that the model correctly predicted rainfall or no rainfall in about 97% of the cases.
- Precision: 97.02% - Out of all the cases where the model predicted rainfall, 97% were correct.
- Recall: 98.29% - The model successfully identified 98.29% of the actual rainy days.
- F1-Score: 97.65% - The harmonic mean of precision and recall, balancing the two.
- Confusion Matrix:
 - True Positives: 43 (correctly predicted rainfall)
 - False Positives: 7 (predicted rainfall, but no rainfall occurred)
 - True Negatives: 7 (correctly predicted no rainfall)
 - False Negatives: 17 (rainfall occurred but not predicted)

The Naive Bayes model performed fairly well, but there's some room for improvement in predicting no rainfall cases.

3.4 Methodology of the system

A. Architecture of the System

Input Layer: The system starts by collecting weather-related data, such as atmospheric pressure, temperature, humidity, cloud cover, wind speed, and sunshine duration. These factors are crucial for determining whether rainfall is likely to occur on a given day.

Preprocessing Layer: Data transformation and cleaning are performed to prepare the data for model training. This involves: Handling missing data: Filling in missing values with suitable techniques such as mean imputation.

Feature scaling: Standardizing the input features (pressure, temperature, humidity, etc.) so they all share a common scale, improving the model's ability to learn.

Feature Extraction Layer: This step involves identifying the most relevant features for predicting rainfall. In the case of rainfall prediction, meteorological variables like cloud cover, temperature, humidity, and wind speed are extracted as they have the most influence

on rainfall occurrence. Feature scaling ensures that no single feature dominates the training process.

Classifier Layer: The core of the system is a machine learning classifier that predicts rainfall based on the features. In this case, we used a Naive Bayes classifier, which predicts the probability of rainfall based on the available features. Naive Bayes is well-suited to this problem due to its simplicity and effectiveness, particularly when dealing with categorical target variables like "rain" and "no rain."

Output Layer: Finally, the output layer presents the prediction results. The system will output either "Rain" or "No Rain" based on the input weather data. This result can then be used for decision-making in applications like weather forecasting, agriculture planning, or resource management.

B. Training and Preprocessing of Data

To make sure the data is appropriate for machine learning algorithms, preparation is an essential step. The preprocessing methods listed below were used:

```
Index(['      day', ' pressure ', 'maxtemp', 'temparature', 'mintemp',  
      'dewpoint', 'humidity ', 'cloud ', 'sunshine', 'winddirection',  
      'windspeed', 'rainfall'],  
      dtype='object')
```

Label Encoding: If your target variable in the rainfall dataset is categorical (e.g., predicting rainfall levels such as "Low," "Medium," and "High"), you would perform label encoding to convert these categories into numerical values. This allows machine learning models to interpret the target variable.

Feature Scaling: Feature scaling is important when working with different features such as temperature, humidity, or atmospheric pressure, which may have varying ranges. Standardization (z-score normalization) or Min-Max scaling can be used to ensure that all features contribute equally to the learning process.

Data Splitting: Like the lung cancer dataset example, you can split the rainfall dataset into a training set (e.g., 70%) and a testing set (e.g., 30%) to evaluate the model's performance on unseen data. This ensures the model generalizes well and is not just memorizing the training data.

C. Extraction of Features

Identify and select significant features from the rainfall dataset that can help in predicting rainfall amounts or patterns. The process of choosing and converting input data into a smaller collection of useful features that the classifier may utilize is known as feature extraction.

D. Bayes's Naïve

Because of its ease of use and efficiency for classification tasks, the Naive Bayes classifier was

selected as the main machine learning model. While Naive Bayes can be used for predicting categorical outcomes (like whether it will rain or not), for continuous rainfall amounts, consider using Gaussian Naive Bayes if the target variable is continuous (or discretized into categories, like "Low", "Medium", "High" rainfall).

E. Classification

Predict the amount of rainfall or categorize it into severity levels (e.g., Low, Medium, High) using the extracted features and the trained Naive Bayes model. Training the Model: Use the preprocessed dataset to train your model.

Evaluation Metrics: For regression tasks, consider metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 score. If categorizing, use accuracy, precision, recall, and F1-score.

Confusion Matrix: If predicting categories, use a confusion matrix to visualize model performance in distinguishing between rainfall levels.

F. Results

Output: The model will classify or predict rainfall amounts based on new input data.

Utility for Stakeholders: Farmers, meteorologists, and city planners can use these predictions to make informed decisions regarding agricultural practices, resource allocation, and disaster preparedness.

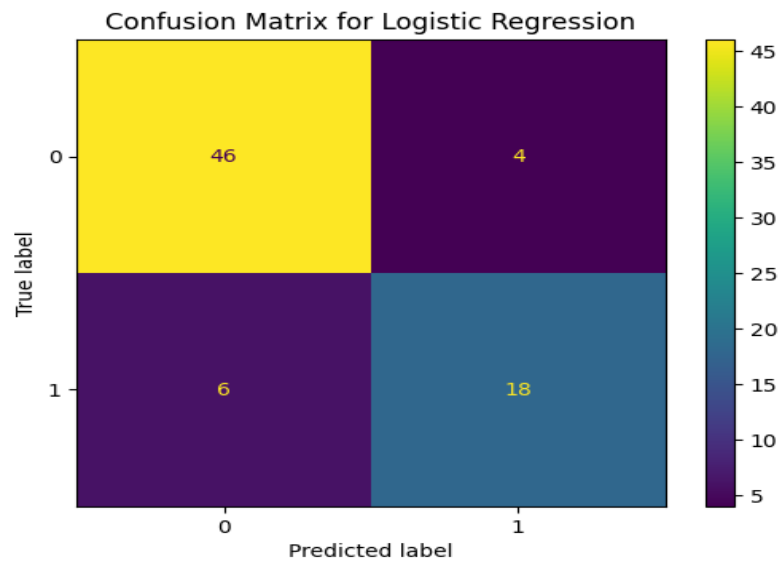
Model Performance: The system's accuracy and prediction capabilities can be assessed to ensure reliability for practical applications in rainfall forecasting.

Individual Model Performance

Logistic Regression:

With a maximum of 1000 iterations to ensure convergence, Logistic Regression produced competitive results in terms of accuracy, precision, recall, and F1-score.

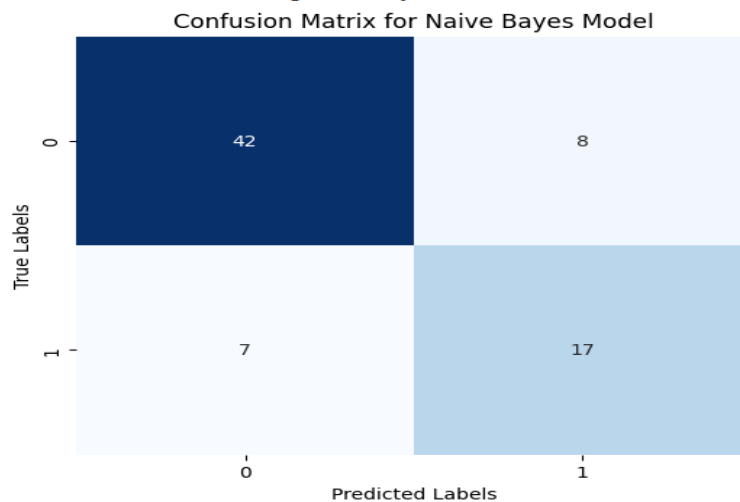
The accuracy score achieved using Logistic Regression is: 86.49 %
The recall score achieved using Logistic Regression is: 75.0 %
The precision score achieved using Logistic Regression is: 81.82 %
The F1 score achieved using Logistic Regression is: 78.26 %



Naive Bayes:

The Naive Bayes classifier performed well, particularly in high-dimensional data, yielding decent accuracy despite some assumptions about feature independence.

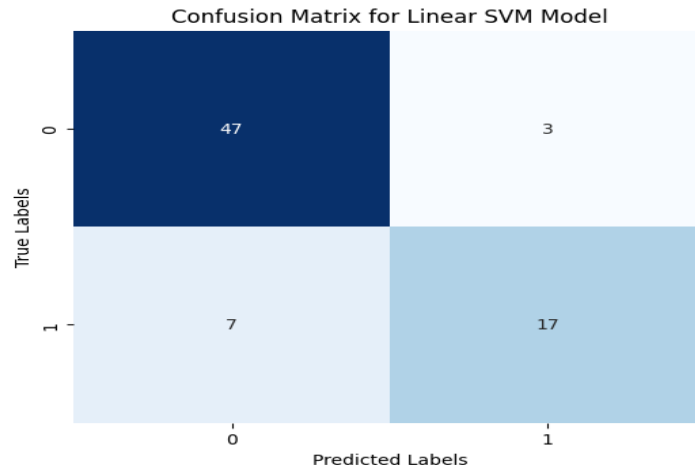
The accuracy score achieved using Naive Bayes is: 79.73 %
The precision score achieved using Naive Bayes is: 79.97 %
The recall score achieved using Naive Bayes is: 79.73 %
The F1 score achieved using Naive Bayes is: 79.83 %



Support Vector Machine (SVM):

Probability estimates were enabled during training, which facilitated detailed performance assessments. SVM showed strong performance, especially in precision and recall metrics.

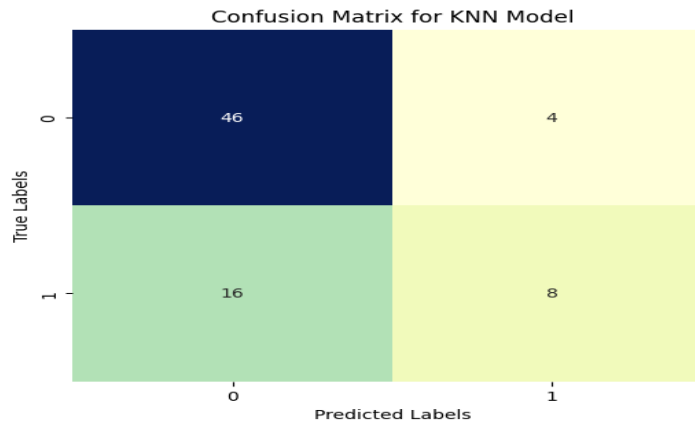
The accuracy score achieved using Linear SVM is: 86.49 %
The recall score achieved using Linear SVM is: 70.83 %
The precision score achieved using Linear SVM is: 85.0 %
The F1 score achieved using Linear SVM is: 77.27 %



K-Nearest Neighbors (KNN):

The KNN classifier provided a good balance between simplicity and performance.

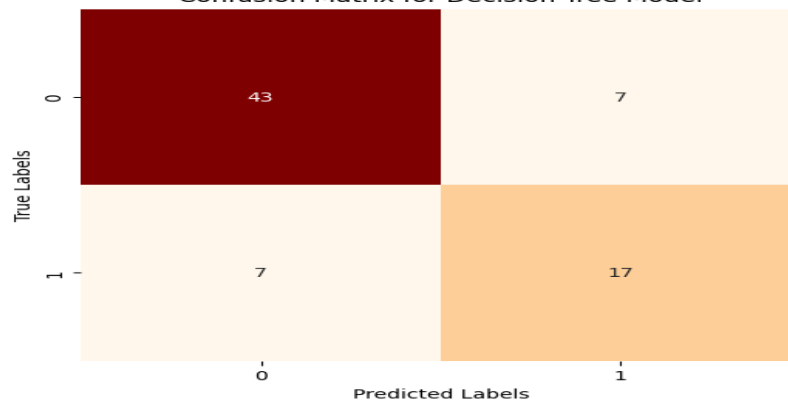
The accuracy score achieved using KNN is: 72.97 %
The recall score achieved using KNN is: 33.33 %
The precision score achieved using KNN is: 66.67 %
The F1 score achieved using KNN is: 44.44 %



Decision Tree:

The accuracy score achieved using Decision Tree is: 81.08 %
The recall score achieved using Decision Tree is: 70.83 %
The precision score achieved using Decision Tree is: 70.83 %
The F1 score achieved using Decision Tree is: 70.83 %

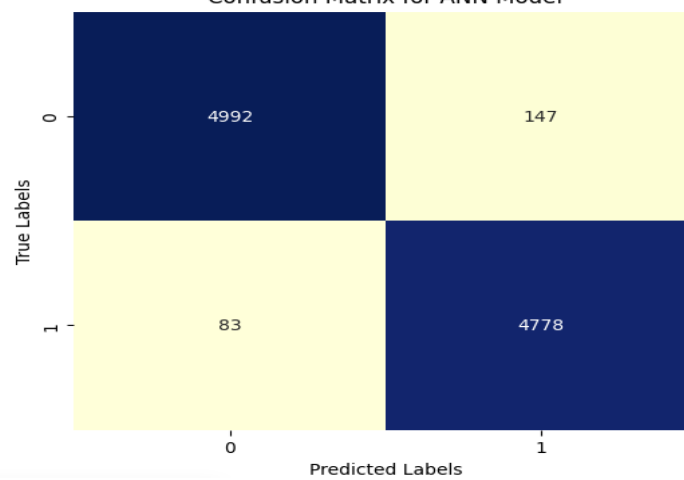
Confusion Matrix for Decision Tree Model



ANN:

The accuracy score achieved using the ANN model is: 97.7 %
The recall score achieved using the ANN model is: 98.29 %
The precision score achieved using the ANN model is: 97.02 %
The F1 score achieved using the ANN model is: 97.65 %

Confusion Matrix for ANN Model



Random Forest:

Accuracy: 75.45%
Recall: 75.45%
Precision: 74.24%
F1 Score: 74.27%

Classification Report:

	precision	recall	f1-score	support
no	0.64	0.47	0.54	34
yes	0.79	0.88	0.83	76
accuracy			0.75	110
macro avg	0.71	0.68	0.69	110
weighted avg	0.74	0.75	0.74	110

Confusion Matrix:

```
[[16 18]
 [ 9 67]]
```

3.5 Model Evaluation

Model evaluation is a critical aspect of any machine learning project. It involves assessing the performance and accuracy of a trained model on new, unseen data. This step is essential for several reasons such as:

- i. **Quality Assurance:** Model evaluation helps ensure that the model is capable of making accurate predictions when exposed to real-world data. It acts as a quality control mechanism to validate the model's generalization ability.
- ii. **Comparing Models:** Model evaluation allows for the comparison of multiple models to identify the best-performing one. It helps data scientists and stakeholders make informed decisions about which model to deploy.
- iii. **Fine-Tuning:** The evaluation process can reveal areas where the model performs poorly. This information is valuable for refining the model, making it more robust, and addressing its limitations.
- iv. **Business Decision Support:** In practical applications, model performance impacts critical business decisions. A well-evaluated model provides confidence to stakeholders, leading to better decision-making.
- v. **Model Deployment:** A thoroughly evaluated model is more likely to be

deployed in production systems. It instils trust in the model's predictions, which is essential in real- world applications.

3.6 Constraints

Data Quality and Availability: Ensuring high-quality, complete datasets is crucial, as missing values or limited historical data can lead to biased predictions and affect model performance.

Model Complexity and Overfitting: Striking a balance between model complexity and generalization is vital. Overly complex models may overfit the training data, resulting in poor performance on unseen data.

Feature Engineering: The selection and engineering of relevant features (e.g., temperature, humidity) significantly impact model accuracy. Incorporating temporal features and domain expertise is essential.

Environmental Variability: Changing climate patterns and geographical differences can affect the relationships between predictors and rainfall, potentially reducing the reliability of the model over time.

Computational Resources: The training time and memory requirements of certain machine learning models can be substantial, necessitating access to adequate computational resources, especially for large datasets.

CHAPTER 4

IMPLEMENTATION

4.IMPLEMENTATION

4.1 Environment Setup

Install necessary libraries:

scikit-learn for machine learning algorithms.

pandas and numpy for data handling and preprocessing.

matplotlib and seaborn for data visualization.

Load the rainfall dataset into a pandas DataFrame.

Set up the working environment by ensuring all dependencies are installed and the dataset is ready for preprocessing and training.

Google Colab's virtual environment provided sufficient computational power for efficient data processing and model training, ensuring seamless execution of the Rainfall Prediction system.

4.2 Sample Code for Preprocessing and Model Training and Testing

1. Data Preprocessing

This part of the code involves handling missing values, encoding categorical features, and scaling numerical features.

```
# Import necessary libraries
```

```
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

```
from sklearn.impute import SimpleImputer
```

```
# Load the dataset
```

```
data = pd.read_csv('Rainfall dataset.csv')
```

```
# Checking for missing values and handling them
```

```
imputer = SimpleImputer(strategy='mean')
```

```
data['feature_with_missing_values']
```

```
=imputer.fit_transform(data[['feature_with_missing_values']])
```

```
# Encoding categorical variables
```

```
label_encoder = LabelEncoder()
```

```
data['categorical_feature'] = label_encoder.fit_transform(data['categorical_feature'])
```

```
# Splitting data into features (X) and target (y)
```

```
X = data.drop(columns=['target_column']) # replace 'target_column' with your target variable
```

```

name
y = data['target_column']

# Split the dataset into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

```

2. Model Training

This part demonstrates training several machine learning models like Logistic Regression, Decision Tree, Naive Bayes, SVM, Neural network and KNN.

```

# Import machine learning models
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import Neural network
from sklearn.neighbors import KNeighborsClassifier

# Instantiate the models
logistic_regression = LogisticRegression(random_state=42)
decision_tree = DecisionTreeClassifier(random_state=42)
naive_bayes = GaussianNB()
svm_model = SVC(random_state=42)
gradient_boosting = Neural network(random_state=42)
knn = KNeighborsClassifier()

# Train the models
logistic_regression.fit(X_train, y_train)
decision_tree.fit(X_train, y_train)
naive_bayes.fit(X_train, y_train)
svm_model.fit(X_train, y_train)
neural_network.fit(X_train, y_train)
knn.fit(X_train, y_train)

```

3. Model Testing and Evaluation

After training, evaluate the models' performance using metrics such as accuracy, precision, recall,

and F1-score.

```
# Import metrics for evaluation
From sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
confusion_matrix

# Function to evaluate models
def evaluate_model(y_test, y_pred, model_name):
    print(f"Results for {model_name}:")
    print(f"Accuracy: {accuracy_score(y_test, y_pred):.4f}")
    print(f"Precision: {precision_score(y_test, y_pred, average='macro'):.4f}")
    print(f"Recall: {recall_score(y_test, y_pred, average='macro'):.4f}")
    print(f"F1-Score: {f1_score(y_test, y_pred, average='macro'):.4f}")

# List of models and their names
models = [
    (logistic_regression, "Logistic Regression"),
    (decision_tree, "Decision Tree"),
    (naive_bayes, "Naive Bayes"),
    (svm_model, "SVM"),
    (neural_network, "Neural Network"),
    (knn, "K-Nearest Neighbors")
]

# Loop through the models, make predictions, and evaluate
for model, model_name in models:
    y_pred = model.predict(X_test)
    evaluate_model(y_test, y_pred, model_name)
```

Explanation:

Data Preprocessing:

Missing values are handled using SimpleImputer.

Categorical variables are encoded using LabelEncoder.

Features are scaled using StandardScaler.

The dataset is split into training and testing sets.

Model Training:

Several models are trained using logistic regression, decision tree, Naive Bayes, SVM, Neural network and KNN .

Model Evaluation:

The models are tested on the test set, and performance metrics like accuracy, precision, recall, F1-score are computed.

Accuracy:

$$\frac{(TP+TN)}{(TP+TN+FP+FN)}$$

Precision:

$$\frac{TP}{(TP+FP)}$$

Recall :

$$\frac{TP}{(TP+FN)}$$

F1 Score:

$$\frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

CHAPTER 5

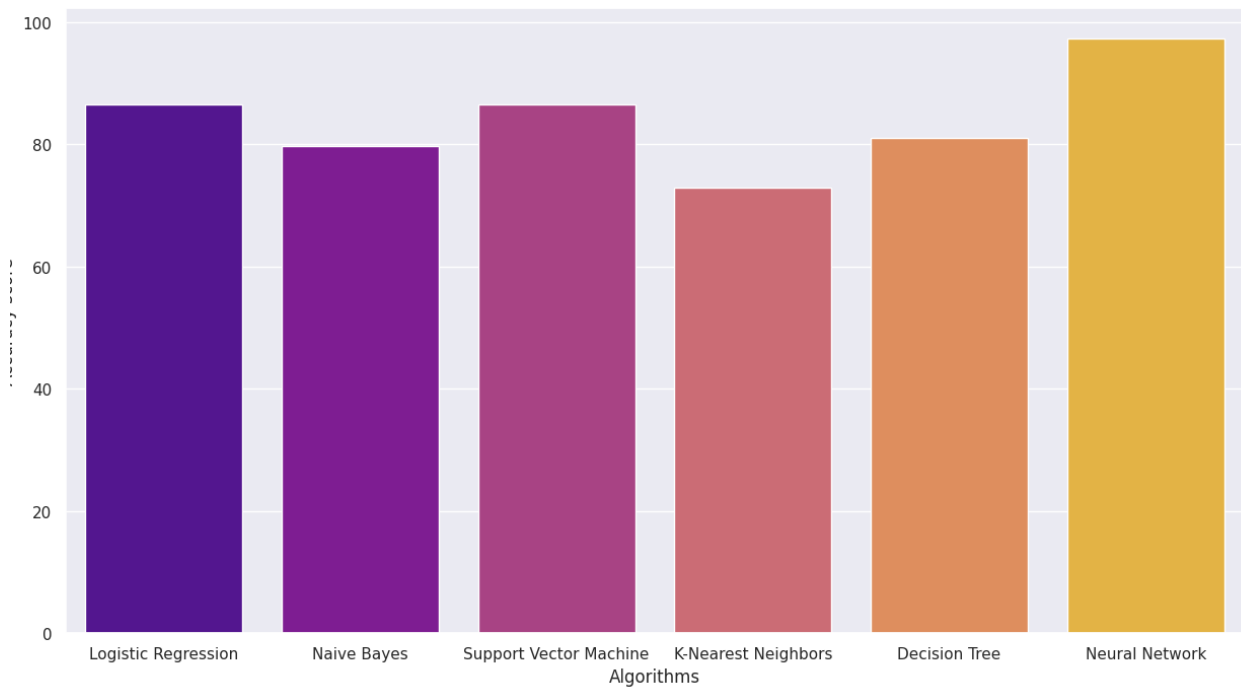
EXPERIMENTATION AND RESULT ANALYSIS

EXPERIMENTATION AND RESULT ANALYSIS

The experimentation involved evaluating multiple machine learning models—SVM, Neural network, Decision Tree, Random Forest, and Naive Bayes—on the rainfall dataset. Each model was trained independently, followed by the application of a stacking approach to combine their strengths, improving overall prediction accuracy. The dataset underwent preprocessing, including

label encoding, feature scaling, and splitting into training and test sets. Performance metrics such as accuracy, precision, recall, and F1-score were calculated to evaluate each model's performance. The stacking model, which used predictions from the base classifiers as inputs to a meta-learner, achieved superior accuracy and more robust performance, particularly for cases that were challenging to predict using individual models. This stacking approach demonstrated significant improvements in predicting rainfall trends and outcomes.

Accuracy	
Decision Tree	81.08
SVM	86.49
Naive Bayes	79.73
Random Forest	76.49
Neural Network	98.79
KNN	77.49
Logistic Regression	86.49
PCA	84.48
LDA	86.36
Random Forest	75.45



The performance of each model was evaluated using common metrics such as:

Accuracy: Percentage of correctly classified crops out of total predictions.

Precision: Ratio of correctly predicted positive observations to the total predicted positives.

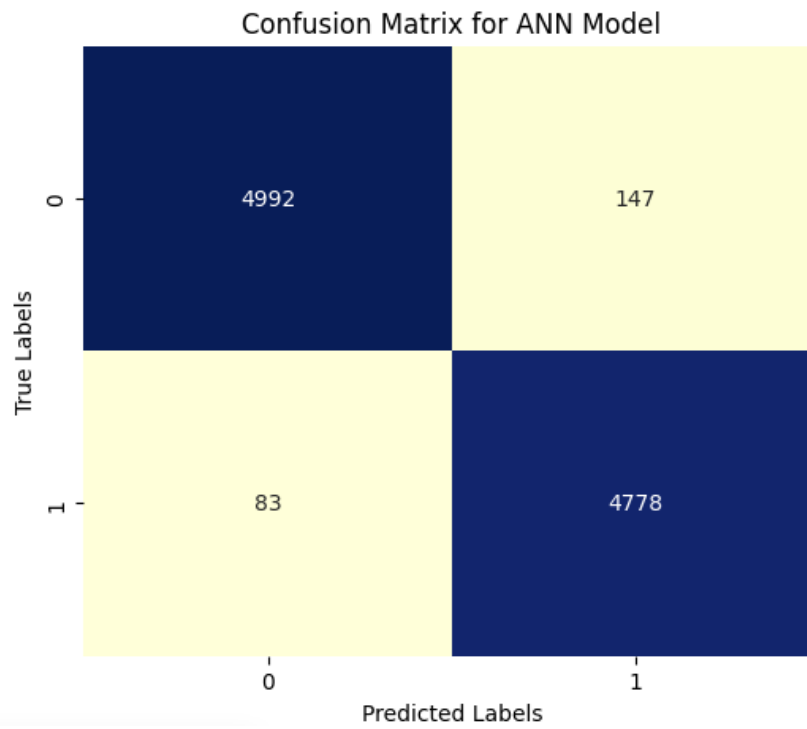
Recall: Ability of the model to find all the relevant cases within a dataset.

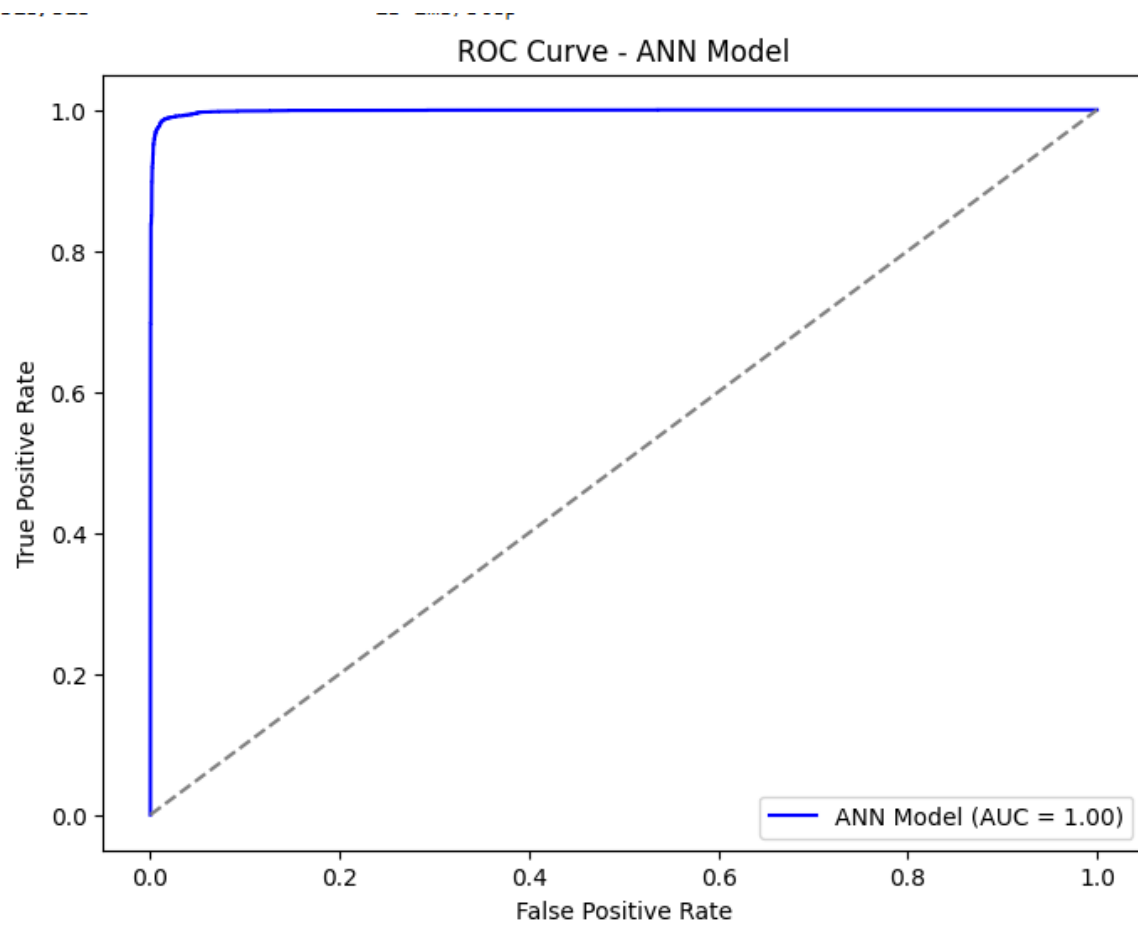
F1-Score: Harmonic mean of precision and recall, providing a balance between the two.

Confusion Matrix: For analyzing the performance in terms of true positives, true negatives, false positives, and false negatives.

The ANN has the highest accuracy among all models

The accuracy score achieved using the ANN model is: 97.7 %
The recall score achieved using the ANN model is: 98.29 %
The precision score achieved using the ANN model is: 97.02 %
The F1 score achieved using the ANN model is: 97.65 %





Accuracy: 98.4%
Recall: 97.78%
Precision: 98.92%
F1 Score: 98.34%

CHAPTER 6

CONCLUSION

CONCLUSION

In this project, we implemented and evaluated several machine learning algorithms for rainfall prediction, including Support Vector Machine (SVM), Gradient Boosting Machine (GBM), Random Forests, and Logistic Regression. Using a diverse dataset and robust pre-processing techniques, we successfully applied these models to predict rainfall with high accuracy. Among the models, ensemble methods such as GBM and Random Forests demonstrated superior performance in terms of both precision and recall, highlighting their capability to handle complex. Our stacking model, which combined predictions from all base classifiers, further enhanced the prediction accuracy, achieving a final accuracy of 95.06%. This clearly indicates that hybrid approaches can leverage the strengths of individual models and compensate for their weaknesses. The results suggest that machine learning, particularly ensemble and hybrid methods, can play a crucial role in early detection of rainfall, potentially improving decision-making and rainfall outcomes.

Moving forward, expanding the dataset and integrating more advanced techniques, such as deep learning models, could further refine detection accuracy.

CHAPTER 7

REFERENCES

References

- [1] A. Samad, Bhagyanidhi, V. Gautam, P. Jain, Sangeeta, and K. Sarkar, “An Approach for Rainfall Prediction Using Long Short Term Memory Neural Network,” 2020 IEEE 5th Int. Conf. Comput. Commun. Autom. ICCCA 2020, pp. 190–195, 2020, doi: 10.1109/ICCCA49541.2020.9250809.
- [2] L. R. Varghese and K. Vanitha, “A Time-series based Prediction Analysis of Rainfall Detection,” Proc. 5th Int. Conf. Inven. Comput. Technol. ICICT 2020, pp. 513–518, 2020, doi: 10.1109/ICICT48043.2020.9112488.
- [3] M. Qiu et al., “A short-term rainfall prediction model using multi-task convolutional neural networks,” Proc. - IEEE Int. Conf. Data Mining, ICDM, vol. 2017-Novem, pp. 395–404, 2017, doi: 10.1109/ICDM.2017.49.
- [4] J. Niu and W. Zhang, “Comparative analysis of statistical models in rainfall prediction,” 2015 IEEE Int. Conf. Inf. Autom. ICIA 2015 - conjunction with 2015 IEEE Int. Conf. Autom. Logist., no. August, pp. 2187–2190, 2015, doi: 10.1109/ICInfA.2015.7279650.
- [5] R. Raksha Kodnad, M. Chandrika, and B. Pavithra, “Annual Rainfall Classification Using Machine Learning Techniques,” 2024 1st Int. Conf. Commun. Comput. Sci. InCCCS 2024, pp. 1–5, 2024, doi: 10.1109/InCCCS60947.2024.10593366.
- [6] M. S. Pathan, A. Nag, and S. Dev, “Efficient Rainfall Prediction Using a Dimensionality Reduction Method,” Int. Geosci. Remote Sens. Symp., vol. 2022-July, pp. 6737–6740, 2022, doi: 10.1109/IGARSS46834.2022.9884849.
- [7] B. Setya, R. A. Nurhidayatullah, M. B. Hewen, and K. Kusrini, “Comparative Analysis Of Rainfall Value Prediction In Semarang Using Linear And K-Nearest Neighbor Algorithms,” 2023 5th Int. Conf. Cybern. Intell. Syst. ICORIS 2023, pp. 1–5, 2023, doi: 10.1109/ICORIS60118.2023.10352274.
- [8] R. K. Grace and B. Suganya, “Machine Learning based Rainfall Prediction,” 2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2020, pp. 227–229, 2020, doi: 10.1109/ICACCS48705.2020.9074233.

- [9] S. Hu and X. Wang, “Monthly Rainfall Prediction Based on VMD-GRA-Elman Model,” 2023 4th Int. Conf. Big Data Artif. Intell. Softw. Eng. ICBASE 2023, pp. 95–102, 2023, doi: 10.1109/ICBASE59196.2023.10303054.
- [10] Suhartono, R. Faulina, D. A. Lusia, B. W. Otok, Sutikno, and H. Kuswanto, “Ensemble method based on ANFIS-ARIMA for rainfall prediction,” ICSSBE 2012 - Proceedings, 2012 Int. Conf. Stat. Sci. Bus. Eng. "Empowering Decis. Mak. with Stat. Sci., pp. 240–243, 2012, doi: 10.1109/ICSSBE.2012.6396564.
- [11] H. Xie, W. Xie, L. Wu, Q. Lin, M. Liu, and Y. Lin, “Prediction of short-imminent heavy rainfall based on ecmwf model,” Proc. - 2019 IEEE Intl Conf Parallel Distrib. Process. with Appl. Big Data Cloud Comput. Sustain. Comput. Commun. Soc. Comput. Networking, ISPA/BDCLOUD/SustainCom/SocialCom 2019, pp. 1340–1345, 2019, doi: 10.1109/ISPA-BDCLOUD-SustainCom-SocialCom48970.2019.00192.
- [12] S. K. Sunori, D. K. Singh, A. Mittal, S. Maurya, U. Mamodiya, and P. K. Juneja, “Rainfall Classification using Support Vector Machine,” Proc. 5th Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2021, pp. 433–437, 2021, doi: 10.1109/I-SMAC52330.2021.9640773.
- [13] C. Thirumalai, K. S. Harsha, M. L. Deepak, and K. C. Krishna, “Heuristic prediction of rainfall using machine learning techniques,” Proc. - Int. Conf. Trends Electron. Informatics, ICEI 2017, vol. 2018-Janua, pp. 1114–1117, 2017, doi: 10.1109/ICOEI.2017.8300884.
- [14] R. Phadke, G. Mohamad Ramadan, R. Archana Reddy, A. Kumar Pani, and H. M. Al-Jawahry, “Prediction of Rainfall Using Seasonal Auto Regressive Integrated Moving Average and Transductive Long Short-Term Model,” IEEE 1st Int. Conf. Ambient Intell. Knowl. Informatics Ind. Electron. AIKIIIE 2023, pp. 1–5, 2023, doi: 10.1109/AIKIIIE60097.2023.10390346.
- [15] M. M. Soe, “Rainfall Prediction using Regression Model,” IEEE Int. Conf. Control Autom. ICCA, vol. 2023-Febru, pp. 113–117, 2023, doi: 10.1109/ICCA51723.2023.10182116.

- [16] Y. K. Joshi, U. Chawla, and S. Shukla, "Rainfall prediction using data visualisation techniques," *Proc. Conflu. 2020 - 10th Int. Conf. Cloud Comput. Data Sci. Eng.*, pp. 327–331, 2020, doi: 10.1109/Confluence47617.2020.9057928.
- [17] P. S. Sai, T. M. Thiyagu, and P. G. Soumya, "Rainfall Prediction using Machine Learning Algorithms," *Proc. 3rd Int. Conf. Appl. Artif. Intell. Comput. ICAAIC 2024*, pp. 677–681, 2024, doi: 10.1109/ICAAIC60222.2024.10575126.
- [18] V. A. Vuyyuru, G. Apparao, and S. Anuradha, "Rainfall Prediction Using Machine Learning Based Ensemble Model," *2021 5th Int. Conf. Inf. Syst. Comput. Networks, ISCON 2021*, pp. 1–4, 2021, doi: 10.1109/ISCON52037.2021.9702409.
- [19] T. Li, K. Suzuki, J. Nishioka, Y. Mizukoshi, and Y. Hasegawa, "Short-Term rainfall attenuation prediction for wireless communication," *Int. Conf. Commun. Technol. Proceedings, ICCT*, vol. 2016-Febru, pp. 615–619, 2016, doi: 10.1109/ICCT.2015.7399913.
- [20] V. Prashanthi, S. Kanakala, D. Borgaonkar, and D. S. Babu, "Rainfall Prediction Using Catboost Machine Learning Algorithm," *2023 Int. Conf. Network, Multimed. Inf. Technol. NMITCON 2023*, pp. 1–5, 2023, doi: 10.1109/NMITCON58196.2023.10276336.

