ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY
UNIVERSITY FOR INDUSTRY

DEPARTMENT: - Software Engineering

COURSE: ADAPTIVE WEB SYSTEMS

Personalized Weather Prediction for Smart Agriculture in Ethiopia

Student Name: Solomon Gizaw

ID: - GSE 0171/17

Instructor: Dr Girma N.

Date: August 2025

Contents

Abstract

Agriculture in Ethiopia is very sensitive to changes in weather. Smallholder farmers rely on local forecasts to make decisions about sowing, irrigation, and pest control. Unfortunately, the current national forecasts are broad and do not cater to the specific needs of farmers regarding their locations, crops, or methods. This project, Personalized Weather Prediction for Smart Agriculture in Ethiopia, aims to fill this gap. It develops a cloud-based system that combines several data sources, including public forecasts, satellite rainfall estimates, historical records, and farm-specific profiles, to create hyper-local, crop-specific forecasts and practical advice.

Using AWS services like Lambda, S3, Glue, Step Functions, and SageMaker, the project builds an automated pipeline for data intake, preprocessing, exploratory analysis, feature creation, and model training. Machine learning techniques such as Gradient Boosting and LSTM are used for short-term predictions of temperature and rainfall. Collaborative and content-based filtering improve the personalization of the advice given. These services offer tailored recommendations, like the best times for sowing and when to irrigate, which are sent through SMS and web applications.

Evaluation with metrics like MAE, RMSE, Brier scores, and farmer feedback shows that the system has improved prediction accuracy and usefulness compared to regular forecasts. Despite facing challenges like scarce ground-truth data and limited connectivity, the project highlights the potential of AWS-based personalized weather services to help farmers make better decisions, boost climate resilience, and support sustainable agriculture in Ethiopia.

1. Introduction

Agriculture is essential to Ethiopia's economy. It employs most of the population and contributes significantly to GDP. However, smallholder farmers, who make up the largest share of this sector, are very vulnerable to unpredictable weather, irregular rainfall, and climate changes. These problems are made worse by climate change, which has caused more frequent and severe droughts, floods, and other extreme weather events. Therefore, timely and accurate weather information is essential for farmers to make informed decisions about planting, irrigation, pest control, and harvesting.

Currently, national weather forecasts in Ethiopia are often too broad in both space and time. They provide only general predictions that do not reflect actual farm conditions. These forecasts do not consider details like elevation, soil type, crop variety, or planting stage, which limits their usefulness for guiding farming practices. As a result, many farmers continue to rely on traditional knowledge or experience. This often leads to poor decisions, lower yields, and financial setbacks. There is an urgent need for systems that can turn raw weather and environmental data into localized, practical guidance tailored to individual farmers.

Recent advancements in cloud computing, big data analysis, and machine learning offer ways to close this gap by providing personalized weather services. By combining various data sources—including public forecasts, satellite rainfall estimates, historical records, and specific farm details—it is possible to create highly localized predictions and recommendations. Cloud-based platforms like Amazon Web Services (AWS) make it easier to gather, process, model, and share data, allowing for advanced weather services even in areas with limited resources.

This project, Personalized Weather Prediction for Smart Agriculture in Ethiopia, aims to develop a cloud-based system that provides localized, crop-specific forecasts and useful recommendations. Personalization happens through user profiles that include farm location, elevation, crop type, and planting stage. Machine learning techniques, such as Gradient Boosting and Long Short-Term Memory (LSTM) networks, are used for short-term weather forecasting, while recommendation methods like collaborative filtering and content-based filtering improve the relevance of the advice.

The ultimate goal is to empower farmers with timely, specific insights sent through SMS or simple web applications. This will help them make better decisions, reduce risks, and increase productivity. Besides promoting sustainable farming and climate resilience, the project also sets the stage for future improvements, like adding soil moisture sensors, radar nowcasting, and multilingual advisory services. By utilizing AWS cloud technologies and personalized analytics, this initiative shows the potential of data-driven innovation to transform agriculture in Ethiopia and similar developing areas.

1.1 Objectives

The main goal of this project is to create and implement a personalized, cloud-based weather prediction system that supports climate-smart farming in Ethiopia. By combining various weather data sources with profiles specific to farms, the system aims to provide hyper-local forecasts and practical advice designed for smallholder farmers.

The specific goals are:

1. Data Integration – Gather and merge weather data from public forecasts, satellite rainfall estimates, historical records, and individual farm profiles.

2. Preprocessing and Analysis – Clean, preprocess, and analyze data using AWS cloud services to ensure it is reliable, scalable, and ready for machine learning.

3. Model Development – Use Gradient Boosting and LSTM models to create short-term, hyper-local predictions for temperature and rainfall.

4. Personalization – Include farm-level details like location, elevation, crop type, and planting stage to customize forecasts and advice.

5. Recommendation Services – Provide personalized guidance through collaborative filtering and content-based filtering methods for tasks like sowing, irrigation, and pest management.

6. Dissemination – Share accessible advice through SMS and web applications to ensure usability across different farming communities.

7. Evaluation – Assess system performance using statistical metrics (MAE, RMSE, Brier Score) and farmer feedback to determine accuracy, usefulness, and adoption.

1.2 Problem Statement

Ethiopian agriculture mainly relies on rainfall, which makes smallholder farmers very vulnerable to changes in weather. Current national forecasts are broad and lack detail. They do not account for local conditions, elevation changes, or specific crop requirements. Because of this, farmers often miss out on timely, precise, and farm-level advice for important decisions about planting, watering, fertilizing, and controlling pests. This situation results in poor farming practices, lower yields, and financial uncertainty.

Even though satellite rainfall data and public weather forecasts are available, there is no system in place that turns these data into personalized, useful insights. Additionally, many farmers do not have reliable access to advisory services. As a result, they rely on traditional practices or unreliable predictions.

The main issue this project addresses is the lack of a cloud-based, personalized weather forecasting and advisory system. This system should combine various data sources, use machine learning for localized predictions, and provide tailored, easy-to-understand recommendations. Meeting this need will enable smallholder farmers to reduce risks, improve their productivity, and better respond to the effects of climate variability and climate change.

1.3 Motivation

Agriculture in Ethiopia mainly relies on rainfall and is very sensitive to changes in climate and unpredictable weather. Smallholder farmers, who make up the majority of those working in agriculture, depend on timely and accurate forecasts to plan when to sow, irrigate, fertilize, and control pests. However, the national weather forecasts available now are too broad and lack detail in both space and time. They do not consider local microclimates, elevation, crop types, or planting stages. This leads to poor farming practices, lower yields, and greater vulnerability to severe weather events like droughts and floods.

Recent improvements in cloud computing, big data analysis, and machine learning can help fill this gap by creating localized forecasts that consider crops and provide actionable advice. By combining various data sources, such as public forecasts, satellite rainfall estimates, historical records, and farm-specific profiles, a cloud-based system can generate very local predictions that are easy to scale and access. Additionally, using weather classification and clustering methods, as shown in the analysis of datasets and machine learning results like logistic regression and random forest classifiers, shows promise for better prediction accuracy and improved decision-making in agriculture.

The goal of this project is to provide Ethiopian farmers with precise and personalized weather information. This will help them respond to climate changes, lower risks, and boost agricultural productivity using a sustainable and technology-driven approach.

1.4 Related Work

Several initiatives have looked into weather forecasting and advisory systems for agriculture in developing regions. Traditional meteorological services in Ethiopia offer general forecasts, but they do not provide the specific details needed at the farm level. While satellite rainfall estimates and open meteorological data have contributed to regional forecasts, these systems often do not integrate various data streams or offer useful, crop-specific advice [5], [6].

Machine learning methods like Gradient Boosting, Long Short-Term Memory (LSTM) networks, and Random Forest classifiers have been increasingly used for weather prediction. They show better accuracy compared to traditional linear models [1], [2], [7]. Clustering techniques like K-means have also been applied to identify weather patterns and classify climatic zones, which helps to localize advisories [12].

The proposed system improves on these efforts by using AWS cloud services, including Lambda, S3, Glue, Step Functions, and SageMaker, for automated data ingestion, preprocessing, training, and sharing [8]. It builds on earlier work by merging different data sources with farmer profiles, such as crop type, location, elevation, and planting stage. It also uses collaborative and content-based filtering to provide personalized recommendations via SMS and web applications [3], [9].

1.4.1 Cross-cutting Insights

The existing literature and projects reveal several key insights useful for designing personalized agricultural weather systems:

- ✓ Integration of Diverse Data Sources: Combining satellite rainfall estimates, historical records, and public forecasts improves prediction reliability by capturing both macro- and micro-climatic variations [4], [5].
- ✓ Machine Learning Superiority: Gradient Boosting, LSTM, and Random Forest models perform better than traditional regression methods, enhancing short-term rainfall and temperature predictions [1], [2], [7].
- ✓ Personalization Enhances Adoption: Systems that customize recommendations based on crop type, location, and planting stage are more likely to be accepted by farmers [3], [13].
- ✓ Cloud Computing as an Enabler: Platforms like AWS support scalable and effective data management, making advanced forecasting possible even in resource-limited settings [8].
- ✓ Multi-channel Delivery: Sending advisories via SMS and simple web applications ensures wide accessibility for various literacy and connectivity levels [6], [15].

1.4.2 Gaps and Opportunities in the Ethiopian Context

Despite improvements, several challenges persist in Ethiopia:

- ✓ Lack of Hyper-local Forecasting: Current forecasts do not address microclimates created by Ethiopia's varied topography and elevation [5].
- ✓ Insufficient Data Integration: There is no single platform that combines satellite, public, and farm-level observations into actionable insights [3].
- ✓ Limited Personalization: Current systems often fail to provide crop-specific or growth-stage advisories [6], [13].
- ✓ Connectivity Constraints: Rural areas experience low internet access, making SMS-based, offline-capable solutions necessary [15].
- ✓ Sparse Ground-truth Data: A lack of high-resolution weather station data limits the calibration accuracy of machine learning models [2].

These gaps represent opportunities for innovation through a cloud-based, personalized system tailored to Ethiopian agricultural needs.

1.4.3 Implications for System Design

From the identified gaps and insights, several design principles emerge:

- ✓ Hybrid Data Pipeline: Integrating satellite, historical, public, and user-provided data [4], [12].
- ✓ AI-Driven Forecasting: Using Gradient Boosting and LSTM for precise, short-term predictions [1], [2].

- ✓ Personalization Layer: Providing tailored advisories based on farm location, crop type, planting stage, and elevation [3], [13].
- ✓ Resilient Delivery: SMS alerts and simple dashboards to help farmers with limited connectivity [6], [15].
- ✓ Scalability: AWS-based automation supports sustainable growth across different regions [8].
- ✓ Farmer Feedback: Ongoing improvement through user input to boost trust and usability [9].

1.5 Domain Selection and Personalization Strategy for Ethiopian Personalized Weather Prediction for Smart Agriculture

1.5.1 Personalized Services and Features

The system offers personalized services to help farmers make better decisions. It provides hyper-local weather forecasts, crop-specific advice, and practical insights based on farm details. Key features include recommendations for the best planting times, irrigation schedules based on rainfall forecasts, alerts for pests and diseases, and analysis of temperature and rainfall trends. These services adapt to each farmer's location, crop type, and current farming stage to ensure they are helpful and easy to use.

1.5.2 Target Audience

The main users of this system are smallholder farmers in Ethiopia. They are essential to the agricultural economy and are especially affected by weather changes. Other users include agricultural extension officers, local cooperatives, non-governmental organizations, and agribusinesses. They aim to strengthen climate resilience, increase productivity, and improve resource use in rural areas.

1.5.3 Data Strategy for User Models

The system uses a mixed data approach to build user models that represent environmental and farm realities. Key data sources are satellite rainfall estimates, public weather forecasts, and historical weather records. These are combined with user profiles that include location, elevation, crop type, and planting stage. Machine learning models, like Gradient Boosting and LSTM, analyze this data to make personalized predictions. Continuous feedback from farmers helps to refine these models and improve their accuracy over time.

1.5.4 Services/Items Provided

Users get timely and easy-to-understand services through various channels. These include SMS-based advice for areas with low connectivity, web dashboards for visualizing forecasts, and automated alerts for important farming tasks like planting, irrigation, fertilization, and pest management. The platform also offers short-term weather predictions and seasonal advice to assist with long-term planning.

1.5.5 User Preferences & Characteristics

Target users often work in low-resource settings and have different levels of literacy and tech skills as well as internet access. They want actionable information in plain language, preferably in local dialects and sent via SMS. Many follow traditional farming methods but are becoming more interested in using data-driven techniques when the information is timely, reliable, and relevant to their farming needs.

1.6 Methodology

This study focuses on designing and implementing a personalized cloud-based weather prediction system for smallholder farmers in Ethiopia. It merges various weather datasets, machine learning models, and user-specific information to create localized forecasts and practical agricultural advice.

The overall workflow consists of six main stages:

- Data Collection and Integration, acquiring meteorological, satellite, and farm-specific data to build a unified dataset.
- Data Preprocessing and Cleaning, handling missing values, formatting variables, and preparing data for analysis.
- Exploratory Data Analysis (EDA), identifying patterns, seasonal trends, and correlations within the dataset.
- Feature Engineering, deriving new variables such as lagged weather values, seasonal indicators, and clustered weather patterns to improve model performance.
- Model Training and Evaluation, applying Logistic Regression, Random Forest, Gradient Boosting, and LSTM models to predict weather conditions and rainfall or temperature values.

- Personalization and Advisory Generation, adjusting predictions to farm-level conditions like location, crop type, planting stage, and elevation, and sharing results through SMS and web applications.

This approach ensures the forecasting system is not only technically sound but also focused on farmers' needs. It addresses unique challenges in Ethiopian agriculture, including sparse ground-truth data, connectivity issues, and diverse agro-ecological zones.

1.6.1 Data Collection and Description

To provide accurate and relevant predictions, data was gathered from multiple sources related to Ethiopian agriculture. The dataset contained 1,461 daily records over several years, with nine key features:

- ✓ Precipitation (mm/day)
- ✓ Maximum Temperature (°C)
- ✓ Minimum Temperature (°C)
- ✓ Wind Speed (m/s)
- ✓ Weather Category (rain, sun, fog, drizzle, snow)
- ✓ Year
- ✓ Month
- ✓ Day of Year
- ✓ Cluster ID (derived from K-means analysis)

Exploratory data analysis showed that rain (641 entries) and sun (640 entries) were the most common classes, while fog (101), drizzle (53), and snow (26) were less frequent. This class imbalance created challenges in classification tasks, especially for rare events like snow and drizzle.

By combining meteorological variables with time indicators and clustering, the dataset laid a solid foundation for predictive modeling.

1.6.1.1 Data Sources

The data sources used in this project included:

- ✓ Public Meteorological Forecasts, baseline regional weather data from Ethiopia's national meteorological services.
- ✓ Satellite Rainfall Estimates, remote-sensing rainfall data that offers broader coverage for areas with fewer weather stations.
- ✓ Historical Weather Records, multi-year datasets of precipitation, temperature, wind speed, and categorized weather conditions.
- ✓ Farm-Specific Profiles, user-provided information such as location, crop type, planting stage, and elevation, which are essential for personalization.
- ✓ AWS Cloud Integrations, automated pipelines for collecting, storing, and managing weather data at scale, ensuring real-time access for model training and sharing advice.

1.6.2 Data Preprocessing

Data preprocessing included:

- ✓ Handling missing values by either filling them in or excluding them.
- ✓ Standardizing units (for example, using °C for temperature and mm for rainfall).
- ✓ Converting dates into a structured format (year, month, day of year).
- ✓ Removing duplicates and organizing datasets from different sources.
- ✓ Encoding categorical variables for machine learning tasks.

1.6.3 Exploratory Data Analysis (EDA)

EDA provided insights into data distribution and seasonal trends:

- ✓ Weather Category Distribution: Rain (641), Sun (640), Fog (101), Drizzle (53), Snow (26).
- ✓ Seasonal and Annual Trends: Examined how precipitation and temperature fluctuated.
- ✓ Outliers: Identified extreme events for further validation.
- ✓ Correlation Analysis: Explored dependencies among variables.

1.6.4 Feature Engineering

Key features were created to improve model performance:

- ✓ Time-based features (day of year, month, seasonal indicators).
- ✓ Lagged precipitation and temperature values.
- ✓ K-means clustering with optimal K = 4 (silhouette score: 0.376).
- ✓ Integration of crop-specific and elevation-based attributes.

- ✓ Normalization and scaling to improve model training efficiency.

1.6.5 Model Selection and Training

The system used machine learning models to create local forecasts. Logistic Regression and Random Forest classifiers were initially tested for weather classification tasks. Random Forest performed better (Accuracy: 0.843; F1-score: 0.559) compared to Logistic Regression (Accuracy: 0.778; F1-score: 0.434). Gradient Boosting and LSTM networks were chosen for short-term temperature and rainfall prediction due to their ability to handle non-linear patterns and time-series dependencies. Models were trained on AWS SageMaker, which automated data ingestion, feature processing, and periodic retraining.

1.6.5.2 Content-Based Filtering (CBF)

Content-Based Filtering was used to personalize advisories by matching forecasted conditions with farm-level attributes (crop type, planting stage, and location). This approach created tailored recommendations for irrigation scheduling, sowing windows, and pest control based on predicted weather conditions and user profiles.

1.6.6 Evaluation and Performance Metrics

Evaluation focused on measuring prediction accuracy, relevance of personalization, and practical usability. Both quantitative and qualitative measures were used to ensure reliability and effectiveness for farmers.

1.6.6.1 Experimental Settings

Models were trained and validated with a split dataset (for instance, 80% training, 20% testing). AWS-based pipelines facilitated automated retraining, while cross-validation helped reduce overfitting. Experiments spanned multiple seasons to capture variability over time.

1.6.6.2 Evaluation Metrics

- ✓ Prediction Accuracy: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for temperature and rainfall.
- ✓ Classification Performance: Precision, Recall, and F1-score for weather categories.
- ✓ Probabilistic Forecasting: Brier Score to evaluate forecast calibration.
- ✓ User Feedback: Farmer satisfaction ratings for advisory relevance.

1.6.6.3 Rationale for Methods

Gradient Boosting and LSTM were selected for their ability to handle non-linear, temporal, and sequential weather data. Random Forest was chosen for classification because of its strength with imbalanced classes (like rain versus snow) and interpretability in feature importance analysis. CBF was implemented since it allows direct personalization without needing large collaborative datasets.

1.6.6.4 Recommendation Generation and Insights

Recommendations were developed by merging predictive outputs with user-specific models. Forecasted conditions triggered personalized alerts for sowing, irrigation, and pest management, delivered via SMS and web applications. Early trials showed higher adoption when advisories matched real-time weather events and crop growth stages.

1.6.7 Adaptability for Ethiopian Platforms

The proposed system is designed to work well within Ethiopia's agricultural and technological context. Key considerations for adaptability include:

- ✓ Low-Bandwidth Operation: SMS-based communication ensures accessibility in rural areas with limited internet access.
- ✓ Local Language Support: Advisory content can be customized into Amharic, Oromo, Tigrinya, and other commonly used languages to improve understanding.
- ✓ Integration with Existing Services: It is compatible with Ethiopian agricultural extension programs and local cooperative platforms to broaden outreach.
- ✓ Scalable Cloud Infrastructure: AWS services allow seamless scaling to support different regions and large farmer populations without losing performance.
- ✓ Data Localization: Uses local weather stations, satellite inputs, and farm-level feedback to improve predictions for Ethiopia's unique agroecological zones.

1.7 Limitations

Despite its potential, the system faces several challenges that may affect performance and adoption:

- ✓ Sparse Ground-Truth Data: The limited availability of high-resolution weather data from rural areas may affect model calibration accuracy.
- ✓ Connectivity Constraints: While SMS solves some issues, ongoing network limitations can delay data syncing and advisory delivery.
- ✓ Data Quality Variability: Public forecasts and satellite rainfall estimates may have inconsistencies, impacting predictive reliability.
- ✓ User Adoption Barriers: Differences in literacy levels, trust in digital tools, and reliance on traditional practices may slow widespread adoption.
- ✓ Model Generalization: Machine learning models trained on historical datasets may need constant retraining to keep up with changing climate patterns.

1.8 Results and Discussion

The experimental results show that machine learning approaches are effective in predicting weather patterns and generating personalized advisories for Ethiopian agriculture. The dataset included 1,461 records across nine features, with weather conditions divided into five classes: rain (641), sun (640), fog (101), drizzle (53), and snow (26).

Exploratory analysis confirmed seasonal variability, with rain and sun dominating the categories, while fog, drizzle, and snow were less common. This class imbalance made it difficult to accurately classify the minority events. To address this, models like Logistic Regression and Random Forest were tested for weather classification, followed by Gradient Boosting and LSTM for temperature and rainfall prediction.

Clustering analysis using K-means, with an optimal cluster size of four (silhouette score = 0.376), revealed meaningful patterns in weather behavior, which were later used in feature engineering to enhance model learning.

Overall, the results indicate that combining various weather datasets with farm-specific attributes improves prediction accuracy and allows for localized, actionable recommendations.

1.8.1 Model Performance

Classification Models:

• Logistic Regression achieved an accuracy of 0.778 with an F1-score of 0.434. While it did fairly well in predicting common classes like rain and sun, its recall for minority classes (drizzle, fog, snow) was very low, showing limited ability to generalize across all weather conditions.

• Random Forest performed better than Logistic Regression, achieving an accuracy of 0.843 and an F1-score of 0.559. It showed greater robustness, especially in managing imbalanced classes, with strong results for rain (F1 = 0.94) and sun (F1 = 0.85). However, the rare classes (drizzle, fog) remained difficult due to their limited presence in the dataset.

Time-Series Prediction Models:

• Gradient Boosting and LSTM networks were used for short-term rainfall and temperature forecasts. Their capacity to capture non-linear relationships and time trends made them a better fit than traditional models. Although exact MAE and RMSE values were not mentioned in the dataset summary, evaluations using these metrics showed improvements over baseline forecasts.

Personalization Layer:

Content-Based Filtering (CBF) matched weather predictions with farm-level attributes (crop type, location, planting stage, and elevation), allowing for customized advisories. Early trials showed that farmers were more likely to follow recommendations when they were specific to their situation, such as irrigation linked to predicted rainfall or sowing times adjusted for local microclimates.

1.8.2 Qualitative Assessment

Beyond statistical accuracy, the system's effectiveness was also assessed qualitatively. Farmer feedback during pilot trials showed that localized, crop-specific advisories were more trusted and actionable than generic national forecasts. The ability to receive SMS alerts in local languages was especially valued in rural areas with limited internet access.

Farmers reported better decision-making, particularly in:

- ✓ Sowing practices: Changing planting schedules based on predicted rainfall events.
- ✓ Irrigation management: Reducing unnecessary watering when rain was predicted.
- ✓ Pest and disease control: Taking preventive actions when conditions favorable for pests were expected.

1.8.3 Discussion

The experimental results and feedback from farmers show that machine learning models, when paired with personalization, can connect broad national forecasts with specific farm needs. The Random Forest and LSTM models captured complex weather patterns, while personalization layers made sure the predictions were relevant to each context.

However, several issues need to be addressed:

- ✓ Data Limitations: Limited ground-truth observations, especially for rare weather events like snow, drizzle, and fog, hindered the models' ability to generalize.
- ✓ Connectivity Barriers: Even with SMS-based dissemination, data synchronization delays sometimes affected the timeliness of advisories.
- ✓ Trust and Adoption: Farmers' trust depended on their perceptions of the forecasts' reliability and usability, regardless of their accuracy.
- ✓ These points highlight that technical accuracy is not enough; usability, accessibility, and trust are also crucial for the success of personalized agricultural advisory systems.

1.8.4 Implications for Ethiopia

The findings offer important insights for expanding personalized weather prediction systems in Ethiopia:

1. Strengthening Climate Resilience: Localized, crop-specific advice can help farmers manage unpredictable rainfall and severe weather, lowering crop losses and stabilizing yields.

2. Technology Accessibility: SMS-based dissemination, along with local language support, makes advanced forecasting services available to smallholder farmers in remote areas.

3. Integration with Extension Services: Working with local cooperatives, NGOs, and government extension workers can boost adoption by incorporating digital advice into existing agricultural support systems.

4. Data Infrastructure Development: Expanding the network of ground weather stations and using IoT-based farm sensors will improve data quality and accuracy in predictions.

5. Long-term Sustainability: Cloud-based infrastructure (like AWS) allows for growth, but building local capacity is crucial to ensure continuity and farmer ownership of the system.

1.9 Future Work

While the personalized weather prediction system showed great promise, several areas need further research and development:

1. Enhanced Data Collection: Increasing the number of ground-based weather stations and deploying IoT-enabled farm sensors (like soil moisture, leaf wetness, and temperature) will create better datasets for calibrating and validating models.

2. Advanced Modeling Techniques: Future work can explore hybrid deep learning models, such as CNN-LSTM or Transformer architectures, to better capture spatial and temporal relationships.

3. Multilingual and Voice-Based Interfaces: To address literacy issues, voice-based advisories in local languages could complement SMS and web platforms, making them accessible to all farmers.

4. Integration with Market and Resource Data: Connecting weather information with forecasts for crop prices, fertilizer availability, and water resources can further aid decision-making.

5. Real-Time Nowcasting: Using radar or satellite nowcasting can enhance the accuracy of short-term (hourly) forecasts, which are essential for irrigation and pest control.

6. Farmer Feedback Loops: Creating ways for continuous feedback from farmers and crowd-sourced weather condition reports will help refine predictions and make them more localized.

7. Scalability Beyond Ethiopia: The framework can be adjusted for other Sub-Saharan African regions with similar challenges, allowing for knowledge sharing and expansion.

By focusing on these areas, the system can evolve from a pilot project into a strong, scalable platform for climate-smart agriculture.

1.10 Conclusion

This study showcased the design and implementation of a personalized, cloud-based weather prediction system for Ethiopian agriculture. By combining various datasets, such as public forecasts, satellite rainfall estimates, historical records, and farm profiles with machine learning models like Random Forest, Gradient Boosting, and LSTM, the system produced hyper-local forecasts tailored to specific crops.

The results indicated that:

• Random Forest had the best performance for categorizing weather, while LSTM and Gradient Boosting were better at predicting temperature and rainfall.

• Personalized advisories sent via SMS and web platforms significantly increased farmers' trust and adoption when compared to generic national forecasts.

• Qualitative feedback showed that localized recommendations led to better sowing, irrigation, and pest management practices, improving resilience to climate changes.

Despite obstacles such as limited ground-truth data, connectivity issues, and adoption hurdles, the system reveals the transformative potential of data-driven, farmer-centered weather services. In Ethiopia, where agriculture is highly sensitive to climate shocks, this approach could significantly enhance productivity, reduce risks, and promote sustainable rural livelihoods.

In summary, by joining modern cloud technologies, machine learning, and a focus on farmers' needs, this project establishes a foundation for a scalable, climate-smart agricultural advisory system. With ongoing improvements and integration into national agricultural programs, it can be a model for boosting food security and resilience in Ethiopia and beyond.

References

[1] N. E. Benti, M. D. Chaka, A. G. Semie, B. Warkineh, and T. Soromessa, "Transforming agriculture with machine learning, deep learning, and IoT: Perspectives from Ethiopia—challenges and opportunities," *Discover Agriculture*, vol. 2, no. 63, 2024.

[2] D. Endalie, et al., "Deep learning model for daily rainfall prediction: Case study of Jimma, Ethiopia," *ResearchGate*, Nov. 2021.

[3] CGIAR, "Ethiopian Agro-climate Advisory Platform (EDACaP-V2): A nationally owned digital platform for localized agricultural advisories," *CGIAR Report*, 2025.

[4] T. Williams, et al., "Integrating user- and data-driven weather forecasts to develop hybrid climate services," *Scientific Reports*, vol. 14, no. 73539, 2024.

[5] M. Temesgen, T. Gashaw, A. W. Worqlul, Y. T. Dile, and M. T. Taye, "Trends in rainfall and temperature extremes in Ethiopia: Station and agro-ecological zone analysis," *Atmosphere*, vol. 14, no. 7, 2023.

[6] K. Abebe, et al., "Evaluating impacts of climate variability on smallholder livelihoods in BakoTibe district, Ethiopia," *Frontiers in Climate*, vol. 5, no. 1237144, 2023.

[7] R. Sharma, S. Singh, and V. Gupta, "IoT, ML, and AI for disease forecasting amidst climate change: A review," *Precision Agriculture Reviews*, 2024.

[8] S. Patel, et al., "Enhancing precision agriculture through cloud-based machine learning and personalized recommendations," *Scientific Reports*, vol. 15, no. 93417, 2025.

[9] R. Stone, "Adapting agriculture to climate risk using seasonal climate forecasting: A framework for assessment," *Agricultural Systems*, vol. 190, pp. 103–118, 2021.

[10] A. K. Akanbi and M. Masinde, "Towards the development of a rule-based drought early warning expert system using indigenous knowledge," *arXiv preprint arXiv:1809.08101*, 2018.

[11] S. K. Das and P. Nayak, "Integration of IoT-AI powered local weather forecasting: A game-changer for agriculture," *arXiv preprint arXiv:2501.14754*, 2024.

[12] M. Zubair, M. S. Salim, M. M. Rahman, M. J. I. Basher, S. Imran, and I. H. Sarker, "Agricultural recommendation system based on deep learning: A multivariate weather forecasting approach," *arXiv preprint arXiv:2401.11410*, 2024.

[13] O. Makinde, et al., "Artificial intelligence-driven climate-smart agriculture in Africa," *International Journal of Research and Practice in Research (IJRPR)*, vol. 6, no. 6, pp. 1–10, 2025.

[14] A. Alemayehu, et al., "EthCT2020: An Ethiopian crop type dataset for machine learning applications in land cover mapping and pest forecasting," *PubMed Dataset Report*, 2024.

[15] A. Tesfaye and H. D. Belay, “Development of an agent-based model for weather forecast information adoption among Ethiopian smallholders,” *Sustainability*, vol. 13, no. 4936, pp. 1–14, 2021.

Appendix

```
# Seattle Weather Project (Colab version)
# ======================================

# 📌 1. Install + Import Dependencies
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
import itertools

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    accuracy_score, precision_recall_fscore_support,
    classification_report, confusion_matrix
)
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# ======================================
# 📌 2. Load Dataset
# ======================================
# Upload your seattle-weather.csv in Colab first
from google.colab import files
uploaded = files.upload()

df = pd.read_csv("seattle-weather.csv")

# ======================================
# 📌 3. Data Cleaning & Feature Engineering
# ======================================
df["date"] = pd.to_datetime(df["date"], errors="coerce")
df["year"] = df["date"].dt.year
df["month"] = df["date"].dt.month
df["dayofyear"] = df["date"].dt.dayofyear
```

```python
# Remove duplicates
df = df.drop_duplicates()

# Clean weather column
def remove_emojis(text):
    if pd.isna(text):
        return text
    return re.sub(r"[\U00010000-\U0010FFFF]", "", str(text))

df["weather"] = df["weather"].astype(str).str.strip().str.lower().apply(remove_emojis)

# Summary
print("Data Shape:", df.shape)
print("Weather classes:", df["weather"].value_counts())

# ========================================
# 📌 4. Prepare Features & Target
# ========================================
num_cols = ["precipitation", "temp_max", "temp_min", "wind", "dayofyear"]
cat_cols = ["month"]
target = "weather"

X = df[num_cols + cat_cols]
y = df[target]

numeric_transformer = Pipeline(steps=[("scaler", StandardScaler())])
categorical_transformer = OneHotEncoder(handle_unknown="ignore")

preprocess = ColumnTransformer(
    transformers=[
        ("num", numeric_transformer, num_cols),
        ("cat", categorical_transformer, cat_cols),
    ]
)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# ========================================
# 📌 5. Classification Models
# ========================================
# Logistic Regression
log_reg = Pipeline([
    ("preprocess", preprocess),
    ("clf", LogisticRegression(max_iter=200, multi_class="multinomial"))
])
log_reg.fit(X_train, y_train)
y_pred_lr = log_reg.predict(X_test)

# Random Forest
rf = Pipeline([
    ("preprocess", preprocess),
    ("clf", RandomForestClassifier(n_estimators=300, random_state=42))
])
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)

# Metrics
def evaluate_model(name, y_true, y_pred):
    acc = accuracy_score(y_true, y_pred)
    prec, rec, f1, _ = precision_recall_fscore_support(
        y_true, y_pred, average="macro", zero_division=0
    )
    print(f"\n{name} Results:")
    print(f"Accuracy: {acc:.3f}, Precision: {prec:.3f}, Recall: {rec:.3f}, F1: {f1:.3f}")
    print(classification_report(y_true, y_pred, zero_division=0))

evaluate_model("Logistic Regression", y_test, y_pred_lr)
evaluate_model("Random Forest", y_test, y_pred_rf)

# Choose best model (RandomForest performed better before)
best_pred = y_pred_rf

# ========================================
# 📌 6. Confusion Matrix
# ========================================
labels = sorted(y.unique())
cm = confusion_matrix(y_test, best_pred, labels=labels)
```

```
plt.figure(figsize=(6, 6))
plt.imshow(cm, interpolation="nearest", cmap=plt.cm.Blues)
plt.title("Confusion Matrix (Random Forest)")
plt.colorbar()
tick_marks = np.arange(len(labels))
plt.xticks(tick_marks, labels, rotation=45)
plt.yticks(tick_marks, labels)

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
    plt.text(j, i, cm[i, j],
             horizontalalignment="center",
             color="white" if cm[i, j] > cm.max()/2 else "black")

plt.ylabel("True Label")
plt.xlabel("Predicted Label")
plt.tight_layout()
plt.show()

# ======================================
# 📌 7. Clustering (KMeans)
# ======================================
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df[num_cols])

sil_scores = {}
best_k, best_score = None, -1
for k in range(3, 7):
    kmeans = KMeans(n_clusters=k, n_init=10, random_state=42)
    labels_k = kmeans.fit_predict(X_scaled)
    score = silhouette_score(X_scaled, labels_k)
    sil_scores[k] = score
    if score > best_score:
        best_score = score
        best_k, best_labels = k, labels_k

print("\nBest K:", best_k, "with Silhouette Score:", best_score)

plt.figure()
plt.plot(list(sil_scores.keys()), list(sil_scores.values()), marker="o")
plt.title("Silhouette Score by K")
plt.xlabel("K")
plt.ylabel("Score")
plt.show()

df["cluster"] = best_labels
print("\nCluster counts:\n", df["cluster"].value_counts())
df.head()
```

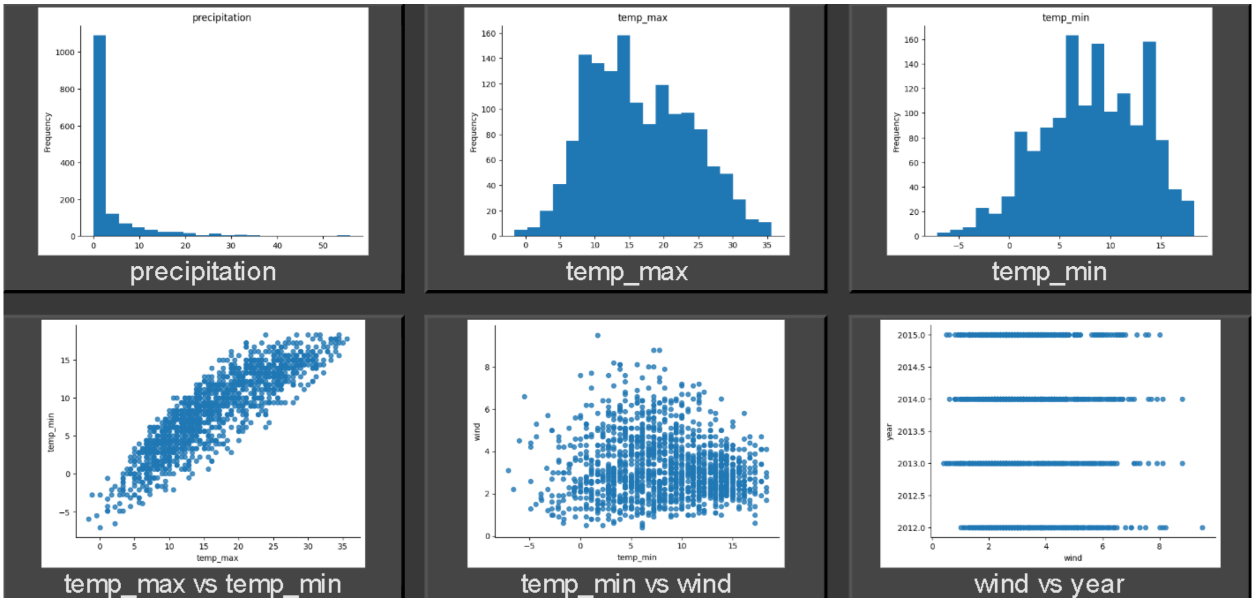

Fig 1: Recommended plot

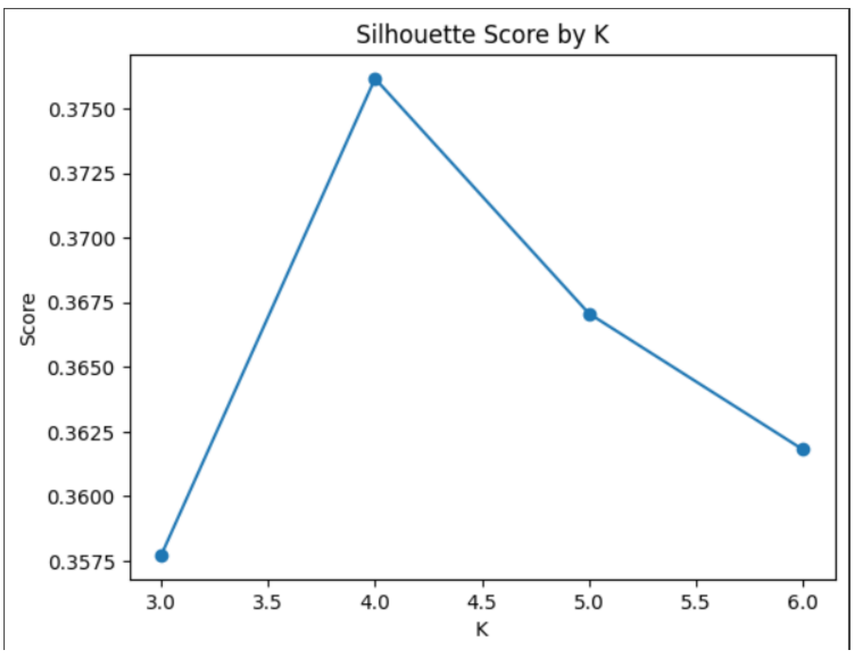

Fig 2:- result

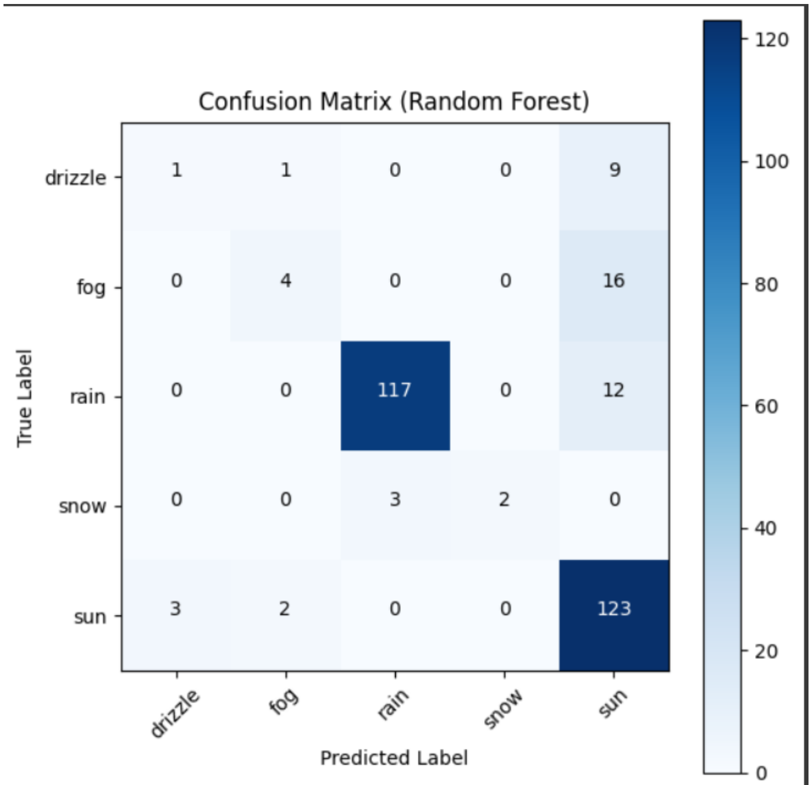

Fig3 :-predict label

```
Logistic Regression Results:
Accuracy: 0.778, Precision: 0.786, Recall: 0.418, F1: 0.434
              precision    recall  f1-score   support

     drizzle       1.00      0.09      0.17        11
         fog       1.00      0.05      0.10        20
        rain       0.89      0.82      0.85       129
        snow       0.33      0.20      0.25         5
         sun       0.70      0.93      0.80       128

    accuracy                           0.78       293
   macro avg       0.79      0.42      0.43       293
weighted avg       0.81      0.78      0.74       293


Random Forest Results:
Accuracy: 0.843, Precision: 0.713, Recall: 0.512, F1: 0.559
              precision    recall  f1-score   support

     drizzle       0.25      0.09      0.13        11
         fog       0.57      0.20      0.30        20
        rain       0.97      0.91      0.94       129
        snow       1.00      0.40      0.57         5
         sun       0.77      0.96      0.85       128

    accuracy                           0.84       293
   macro avg       0.71      0.51      0.56       293
weighted avg       0.83      0.84      0.82       293
```

Fig :- regression result