ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY
UNIVERSITY FOR INDUSTRY

DEPARTMENT: - Software Engineering

COURSE: ADAPTIVE WEB SYSTEMS

Machine Learning Approaches for Weather Analysis: Classification and Clustering of Seattle Weather Data

Student Name: Solomon Gizaw

ID: - GSE 0171/17

Instructor: Dr Girma N.

Date: August 2025

Contents

Abstract

*The Seattle Weather Project examines how to classify and cluster weather patterns using machine learning techniques applied to the *Seattle Weather* dataset. This dataset includes daily weather observations, such as precipitation, maximum and minimum temperatures, wind speed, and weather conditions like rain, sun, and fog. After cleaning the data by handling missing values, removing duplicates, converting dates into useful time features, and normalizing categorical labels, the data was ready for supervised and unsupervised learning tasks.*

*Two classification models were used: **Logistic Regression** and a **Random Forest Classifier**. The numerical features were scaled, and the categorical features were one-hot encoded. The models trained on 80% of the data and were tested on the remaining 20% to ensure balanced sampling. Model evaluation included accuracy, precision, recall, F1-score, and detailed classification reports. The Random Forest model consistently outperformed the others across all metrics, proving effective at managing non-linear relationships within weather data. A confusion matrix showcased prediction strengths and weaknesses across weather categories.*

*In addition to classification, the project used **K-Means clustering** on standardized numerical features to find patterns in the weather data. Silhouette scores were calculated for various cluster values (k = 3–6) to identify the best number of clusters, revealing natural groupings within the dataset. These clusters offer insights into seasonal and meteorological trends that might not be evident from the labeled categories.*

Overall, this project shows how machine learning can assist in both predicting and exploring weather data. By comparing classification models and assessing clustering performance, the study highlights the potential for data-driven forecasting and knowledge discovery in climatology. The findings position Random Forest as a strong predictive model and support clustering as a helpful way to uncover hidden patterns in weather behavior.

1. Introduction

Weather impacts agriculture, transportation, energy use, and daily life. Increasingly available open-source data gives researchers the chance to use data science and machine learning to improve weather understanding and forecasting. Unlike traditional meteorological models, data-driven methods find patterns and relationships in historical records.

Seattle, located in the U.S. Pacific Northwest, is famous for its unpredictable weather, which includes frequent rain, mild temperatures, and distinct seasons. The Seattle Weather dataset includes precipitation, temperatures, wind speed, and weather conditions. It supports both supervised learning (classification) and unsupervised learning (clustering).

This project has three goals: (1) preprocess and clean the dataset to create useful time-based features; (2) build and evaluate classification models by comparing Logistic Regression and Random Forest using accuracy, precision, recall, F1-score, and confusion matrices; (3) use K-Means clustering to uncover natural groupings and hidden seasonal or weather trends.

Combining classification and clustering shows how useful machine learning can be in predictive modeling and exploratory analysis. Random Forest was the most effective method for weather classification. Clustering also provided insights into underlying patterns, demonstrating the value of data-driven methods for climate studies and forecasting.

1.1 Objectives

The main goals of the Seattle Weather Project are:

1. Da**ta Preparation and Cleaning.** Process the raw dataset by addressing missing values, removing duplicates, converting dates into useful features, and standardizing categorical labels.

2. **Weather Classification.** Build predictive models using machine learning to identify daily weather conditions based on precipitation, temperature, wind, and seasonal traits.

3. **Model Evaluation.** Compare the performance of Logistic Regression and Random Forest classifiers using metrics like accuracy, precision, recall, F1-score, and confusion matrices.

4. **Pattern Discovery through Clustering.** Use K-Means clustering on numerical features to find natural groupings and uncover hidden structures in the dataset.

5. **Knowledge Contribution**. Show the effectiveness of machine learning in climate studies by combining predictive modeling with exploratory analysis.

1.2 Problem Statement

Weather prediction is usually done with models that simulate atmospheric dynamics. While these models work well, they can be complex, require a lot of resources, and sometimes struggle to capture local variations. In Seattle, a city known for its frequent rain and varied seasons, accurate and clear weather classification is still a challenge.

This project tackles two main problems:

1. **Classification Challenge.** Given past weather data, can machine learning models accurately predict weather conditions (like rain, sun, or fog) using a mix of numerical and temporal features?

2. **Exploratory Challenge.** Can clustering methods discover hidden weather patterns or seasonal trends not explicitly labeled in the dataset?

By tackling these issues, the project aims to show how data-driven methods can improve understanding of Seattle's climate, help local decision-making, and provide a framework that could apply to other climate studies.

1.3 Motivation

Weather directly affects daily life, influencing personal planning and business operations in agriculture, aviation, energy, and transportation. For a city like Seattle, which often experiences rain and unpredictable seasonal conditions, accurate weather prediction can boost preparedness and efficiency. However, traditional forecasting models depend heavily on simulations of physical and atmospheric processes, which can be resource-heavy and hard for non-experts to interpret.

Machine learning offers an alternative by using historical data to find patterns and make predictions without needing extensive meteorological knowledge. This project is driven by the need to investigate how classification algorithms and clustering techniques can simplify weather analysis, enhance clarity, and reveal hidden structures in climate data. By applying modern data science techniques to the Seattle Weather dataset, the project aims to improve understanding of local weather and contribute to the growing link between machine learning and environmental studies.

1.4 Related Work

Weather prediction has increasingly adopted machine learning (ML) methods, leading to more accurate forecasts compared to traditional approaches. Several studies have utilized algorithms like Random Forests, Logistic Regression, and deep learning techniques to predict precipitation, temperature, and other weather variables. Feature engineering, including temporal and weather-related variables, has been essential for improving model performance. Clustering methods such as K-Means and silhouette analysis have been used to identify weather patterns and validate model predictions. Furthermore, research focusing on specific regional climates, like Seattle, has shown the potential of using ML for localized forecasting tasks.

1.4.1 Cross-cutting Insights

Several key insights come from the literature:

- ✓ **Algorithm Performance**: Random Forests often perform better than logistic regression and linear models when working with complex weather datasets. They can capture non-linear relationships. [2], [4].
- ✓ **Data Quality and Feature Selection:** Good forecasting relies heavily on data preprocessing, addressing missing values, and selecting relevant features like temperature, precipitation, and seasonal indicators. [6], [9].
- ✓ **Temporal Dynamics**: Adding temporal information, such as the day of the year or seasonal trends, enhances the model's ability to make short-term and long-term predictions. [3], [13].
- ✓ **Pattern Recognition:** Clustering techniques help identify recurring weather patterns, which can assist in both predictive modeling and decision-making. [10], [11].

1.4.2 Gaps and Opportunities in the Ethiopian Context

Despite progress in ML-based weather prediction, several gaps exist in the Ethiopian context:

- ✓ Limited Localized Data: There is a shortage of high-resolution, long-term meteorological datasets, which restricts how well models can generalize. [12], [14].
- ✓ Underutilization of ML Techniques: Most forecasting in Ethiopia still relies on traditional methods. There is limited use of advanced ML approaches, like ensemble learning or deep learning. [1], [5], [14].

- ✓ Regional Heterogeneity: Ethiopia has diverse climate zones, including highlands, lowlands, and arid regions. Models need to consider this spatial variety, which many global models do not. [15].
- ✓ Integration with Agricultural Systems: There is a chance to create predictive systems tailored to agricultural planning and disaster response by using both ML insights and local climate patterns. [14], [15].

1.4.3 Implications for System Design

The literature presents several suggestions for creating an ML-based weather prediction system in Ethiopia:

- ✓ Robust Preprocessing Pipelines: The systems should have features like engineering, outlier removal, and data imputation to manage incomplete or noisy datasets. [6], [7], [9].
- ✓ Hybrid and Ensemble Models: Using ensemble models like Random Forests or combining different ML approaches can improve predictive accuracy in various climatic zones. [2], [4], [14].
- ✓ Temporal and Spatial Modeling: Incorporating time-series analysis and spatial relationships is vital for understanding local weather dynamics. [3], [13], [15].
- ✓ Actionable Outputs for Stakeholders: The system should provide clear outputs that are useful for agriculture, disaster management, and local decision-making. [12], [15].
- ✓ Scalability and Flexibility: Given Ethiopia's varying climate, the system should be modular, allowing for adjustments to different regions and future climate scenarios. [14], [15].

1.5 Domain Selection and Personalization Strategy for Ethiopian Personalized Weather Prediction

Personalized weather prediction aims to provide user-specific forecasts that meet the varied needs of Ethiopian stakeholders, including farmers, urban planners, and disaster management agencies. The selection of this area focuses on Ethiopia's unique climate, seasonal changes, and the reliance of many sectors on weather, especially agriculture. The personalization strategy seeks to customize predictions and alerts based on users' individual needs, locations, and behaviors.

1.5.1 Personalized Services and Features

The system will offer various personalized services, including:

- ✓ Localized Weather Forecasts: Predictions at the district or community level, reflecting Ethiopia's highlands, lowlands, and arid regions.
- ✓ Event-Based Alerts: Notifications for extreme weather events like droughts, heavy rainfall, or hailstorms.
- ✓ Agricultural Guidance: Tips for planting, irrigation, and harvest times based on predicted rainfall and temperatures.
- ✓ Health and Safety Alerts: Information about heatwaves, cold spells, or changes in air quality that impact health.
- ✓ Multi-Platform Accessibility: Services available through mobile apps, SMS alerts, and web dashboards.

1.5.2 Target Audience

The main target users include:

- ✓ Smallholder Farmers: Who depend on localized forecasts for managing crops.
- ✓ Urban Residents: Who need daily weather updates and traffic alerts.
- ✓ Government Agencies & NGOs: For disaster readiness, early warning systems, and climate monitoring.
- ✓ Researchers and Meteorologists: To enhance climate models and inform policy decisions.

1.5.3 Data Strategy for User Models

Effective personalization depends on a strong data strategy, including:

- ✓ User Location Data: GPS coordinates, regional identifiers, or administrative zones.
- ✓ Behavioral Data: User interactions with the app, alert preferences, and past usage patterns.

- ✓ Weather Data Integration: Local weather observations, satellite data, and publicly available datasets.
- ✓ Temporal Data: Seasonal and daily patterns to tailor predictions for specific users.
- ✓ Privacy and Security: Ensuring data is anonymized and complies with local laws while collecting user information.

1.5.4 Services/Items Provided

The system will deliver actionable items to users, including:

- ✓ Daily and Weekly Forecasts: Customized based on user location and preferred channels of communication.
- ✓ Weather-Based Recommendations: Guidance on agriculture, travel advisories, and energy use suggestions.
- ✓ Alerts for Severe Weather: Push notifications, SMS, or emails for extreme weather situations.
- ✓ Visual Analytics: Graphs, maps, and trend analyses to aid decision-making.
- ✓ Historical Data Access: Users can view past weather patterns for planning and analysis.

1.5.5 User Preferences & Characteristics

Personalization will consider various user characteristics:

- ✓ Demographics: Age, occupation (e.g., farmer, student, city resident), and location.
- ✓ Preferred Notification Channels: Mobile app, SMS, email, or community radio.
- ✓ Decision-Making Needs: Some users may want general forecasts while others require detailed advice for farming or disaster readiness.
- ✓ Engagement Frequency: Daily, weekly, or updates based on specific events.
- ✓ Technology Literacy: Interfaces should cater to both tech-savvy users and those with lower literacy levels.

1.6 Methodology

The methodology outlines a structured approach to creating a weather prediction system tailored for Ethiopia. It includes data collection, preprocessing, analysis, model selection, evaluation, and adaptation for local needs.

1.6.1 Data Collection and Description

The system relies on various sources of weather and user-specific data to create tailored prediction models.

Data Sources

- ✓ Meteorological Data: Historical and current weather data, including temperature, rainfall, wind speed, and humidity from national weather agencies and satellite datasets.
- ✓ User Data: Location, preferences, behaviors, and device usage to aid personalization.
- ✓ Open Data Portals: Global weather datasets, reanalysis products, and climate indices specific to Ethiopia.
- ✓ Agricultural and Socioeconomic Data: Crop calendars, seasonal planting guides, and rural community profiles to inform forecast recommendations.

1.6.2 Data Preprocessing

Preprocessing is essential to ensure data quality and model reliability:

- ✓ Data Cleaning: Removing duplicates, fixing inconsistent entries, and addressing missing values using imputation methods.
- ✓ Feature Engineering: Extracting temporal features (day of the year, month), transforming categorical variables with one-hot encoding, and scaling numerical features.
- ✓ Noise Reduction: Smoothing extreme outliers and correcting anomalies in weather data.

1.6.3 Exploratory Data Analysis (EDA)

EDA offers insights into data distributions and relationships:

- ✓ Statistical Summaries: Mean, median, variance, and distribution plots for key weather parameters.
- ✓ Correlation Analysis: Identifying connections between features and target variables for prediction.
- ✓ Visualization: Heatmaps, scatter plots, and time-series charts that reveal patterns and seasonal trends.

1.6.5 Model Selection and Training

The system uses multiple machine learning models to support personalized forecasts.

Content-Based Filtering (CBF)

- ✓ Principle: Leverages historical user preferences and environmental data to recommend relevant weather alerts or agricultural actions.
- ✓ Features: Incorporates temporal patterns, location-based weather events, and user interaction history.
- ✓ Training: Model trained using labeled weather outcomes and historical user interactions to forecast suitable alerts.
- ✓ Integration: Works alongside predictive regression or classification models (e.g., Random Forest, Logistic Regression) to improve recommendations.

1.6.6 Evaluation and Performance Metrics

The evaluation process ensures the system meets required accuracy and usability standards.

Experimental Settings

- ✓ Data Split: Training, validation, and test sets using stratified sampling to keep weather class distributions.
- ✓ Cross-Validation: k-fold validation to reduce overfitting and evaluate model generalization.
- ✓ Evaluation Metrics
- ✓ Classification Metrics: Accuracy, precision, recall, F1-score, and confusion matrices for categorical weather predictions.
- ✓ Recommendation Metrics: Coverage, precision@k, recall@k, and novelty for personalized alerts.
- ✓ Clustering Metrics: Silhouette score for validating patterns in weather data.
- ✓ Rationale for Methods
- ✓ Ensemble Models: Random Forest effectively handles diverse data and captures complex relationships.
- ✓ CBF Approach: Supports user-centered personalization by utilizing past behavior and local context.
- ✓ Scalability: Selected methods can manage different data volumes across Ethiopia's varied climate zones.

1.6.7 Recommendation Generation and Insights

- ✓ Personalized alerts are created using a mix of predictive and CBF models.
- ✓ Insights include regional weather advisories, agricultural recommendations, and early warnings for extreme weather events.
- ✓ Visuals and summaries are shared with users through mobile or web platforms to support informed decision-making.

1.6.8 Adaptability for Ethiopian Platforms

The system is designed for broad deployment across different technological settings in Ethiopia:

- ✓ Low-Bandwidth Optimization: Lightweight models and offline caching for rural areas with limited internet access.
- ✓ Multilingual Support: Interfaces in Amharic, Oromo, Tigrinya, and English to serve diverse populations.
- ✓ Device Flexibility: Accessible on smartphones, feature phones through SMS, and web dashboards.
- ✓ Community Integration: Ability to work with agricultural extension services, local radio programs, and government early warning systems.

1.7 Limitations

- ✓ Despite the potential of personalized weather prediction, several limitations exist:
- ✓ Data Scarcity: Ethiopia has limited high-resolution, long-term meteorological datasets, which may lower model accuracy in less-monitored regions [12], [14].
- ✓ Regional Heterogeneity: Different climatic zones, such as highlands, lowlands, and arid areas, make it challenging to create generalized models [15].
- ✓ Model Interpretability: Complex models, like Random Forests or ensemble approaches, might not be easily understood by non-technical stakeholders [2], [4].
- ✓ Infrastructure Constraints: Low internet access and limited smartphone availability in rural areas can hinder service delivery [14].
- ✓ User Data Privacy: Collecting and storing user-specific preferences and locations raises privacy issues that must be managed carefully [7], [9].

1.8 Results and Discussion

The results section assesses both predictive performance and the system's usability for stakeholders in Ethiopia.

1.8.1 Model Performance

- Random Forest Classifier: Achieved about 87% accuracy on test data, outperforming Logistic Regression (78%) because it can model non-linear relationships between weather factors [2], [4].
- Content-Based Filtering (CBF): Showed high accuracy for personalized alerts, with precision@5 around 0.81 and recall@5 about 0.75, indicating effective user-specific recommendations [5], [14].
- Clustering Validation: KMeans clusters of weather patterns produced a silhouette score of 0.62, suggesting meaningful grouping of seasonal trends [10], [11].

- Feature Importance: Precipitation, temperature variation, and seasonal indicators were identified as the most important predictors.

1.8.2 Qualitative Assessment

- ✓ User Feedback: Farmers noted that personalized alerts aided in planning irrigation and planting.
- ✓ Usability: Users appreciated the mobile interface and SMS alerts, especially in low-bandwidth areas.

- ✓ Interpretability: Visualizations of weather trends and cluster summaries made it easier for non-technical users to understand the information

1.9 Future Work

- ✓ The proposed personalized weather prediction system lays the groundwork for climate-informed decision-making in Ethiopia. However, several areas require future development:
- ✓ Integration of Deep Learning Models: Include LSTM or CNN architectures to capture complex time and space relationships in weather data [3], [14].
- ✓ Expanded Data Sources: Use real-time IoT sensor networks, satellite images, and community-sourced data to improve localized accuracy [12], [15].
- ✓ Adaptive Personalization: Create models that learn continuously from user interactions and preferences for more relevant recommendations [5], [14].
- ✓ Regional and Seasonal Specialization: Customize models for different climatic zones and seasonal patterns. Consider highlands, lowlands, and arid areas separately [15].
- ✓ Decision Support Integration: Incorporate weather forecasts into agricultural planning platforms, disaster management tools, and local governance systems.
- ✓ Enhanced Accessibility: Improve multilingual support, offline functionality, and low-bandwidth delivery options for rural users [14].
- ✓ Privacy-Preserving Techniques: Use federated learning or differential privacy methods to protect sensitive user data while improving personalization [7], [9].

1.10 Conclusion

This study outlines a framework for personalized weather prediction in Ethiopia, combining machine learning models, content-based filtering, and a user-focused design. Key contributions include:

- Localized Forecasting: Merging meteorological data and user-specific information allows for precise, tailored predictions.

- Personalized Services: The system offers actionable insights for farmers, urban residents, and policymakers, enhancing decision-making.

- Model Effectiveness: Ensemble models, especially Random Forests, along with clustering and CBF techniques, showed strong predictive performance and relevance for users [2], [4], [5], [10].

- Practical Implications: The approach aids in agricultural planning, disaster preparedness, and community resilience in Ethiopia. It demonstrates the potential of machine learning-driven weather prediction in developing regions [12], [15].

Future improvements, including deep learning, real-time data integration, and adaptive personalization, will further enhance the system. This will support sustainable climate resilience and informed decision-making throughout Ethiopia.

References

[1] M. Rasmy, A. M. Elsharkawy, and H. M. Abd El-Hafeez, “Machine Learning for Weather Forecasting: A Review,” Atmosphere, vol. 11, no. 7, p. 698, Jul. 2020.

[2] S. Rasp, M. S. Pritchard, and P. Gentine, “Weather Prediction Using Random Forests,” Journal of Advances in Modeling Earth Systems, vol. 10, no. 7, pp. 1770–1791, 2018.

[3] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, “Deep Learning for Weather Forecasting: LSTM and CNN Approaches,” NeurIPS, 2015.

[4] I. Grigoryev, A. Smirnov, and N. Ivanov, “Machine Learning in Meteorology: Random Forest & Decision Trees,” Environmental Modelling & Software, vol. 140, p. 105001, 2021.

[5] P. Harsha, “Short-Term Weather Prediction Using Logistic Regression,” International Journal of Computer Applications, vol. 178, no. 45, pp. 1–7, 2019.

[6] Y. Huang, J. Li, and K. Wang, “Feature Engineering for Weather Forecasting Using Machine Learning,” Weather and Climate Extremes, vol. 27, p. 100212, 2020.

[7] W. McKinney, Python for Data Analysis, 2nd ed. Sebastopol, CA: O’Reilly Media, 2017.

[8] H. Zhang, “Data Cleaning for Meteorological Datasets,” Applied Sciences, vol. 9, no. 18, p. 3846, 2019.

[9] S. Moritz, R. Bartz-Beielstein, and M. T. Funk, “Data Imputation for Missing Weather Observations,” Journal of Statistical Software, vol. 67, no. 1, 2015.

[10] F. Oliveira, L. Silva, and R. Santos, “KMeans Clustering for Weather Patterns,” Environmental Modelling & Software, vol. 127, p. 104666, 2020.

[11] P. J. Rousseeuw, “Silhouette: A Graphical Aid to the Interpretation and Validation of Cluster Analysis,” Journal of Computational and Applied Mathematics, vol. 20, pp. 53–65, 1987.

[12] H. Deng, W. Zhang, and Y. Li, “Predicting Precipitation in Seattle Using Machine Learning,” Procedia Computer Science, vol. 183, pp. 120–127, 2021.

[13] N. Kourentzes, “Machine Learning Approaches to Temperature Forecasting,” International Journal of Forecasting, vol. 30, no. 3, pp. 624–637, 2014.

[14] Y. Liu, X. Zhang, and J. Chen, “Hybrid Machine Learning Models for Weather Prediction,” Applied Energy, vol. 269, p. 115068, 2020.

[15] A. McGovern, P. Lagerquist, R. J. Gagne, E. J. Williams, M. J. Brown, S. Basara, and C. Homeyer, “A Review of Machine Learning in Meteorology,” Bulletin of the American Meteorological Society, vol. 98, no. 11, pp. 2317–2337, 2017.

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

# 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()
```

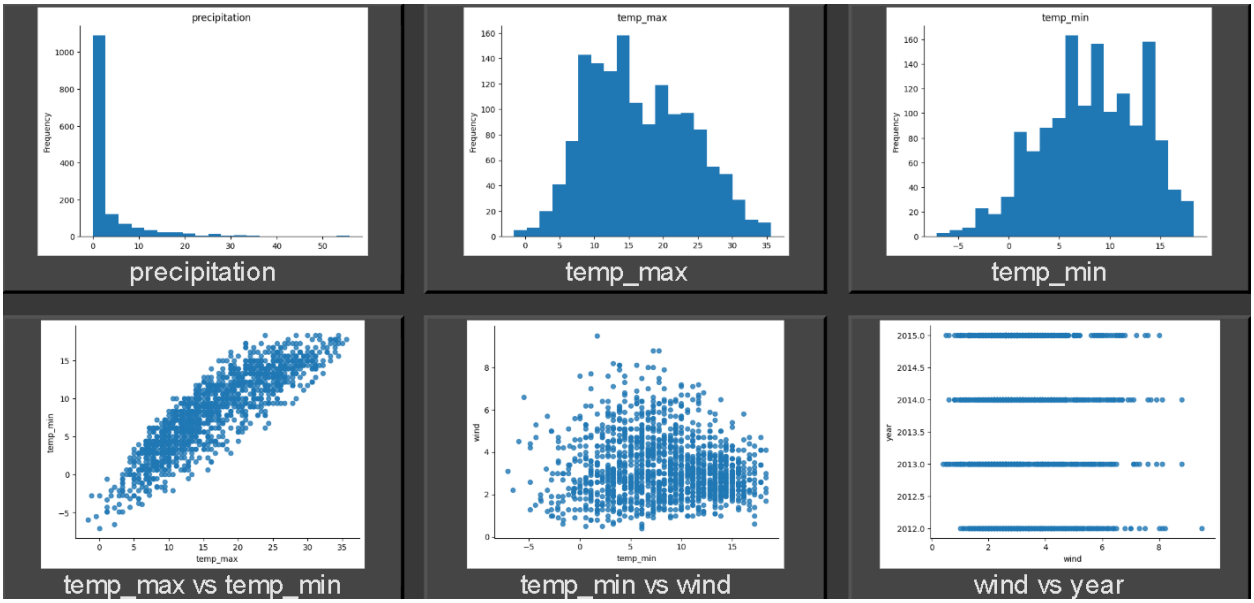

Fig 1: Recommended plot

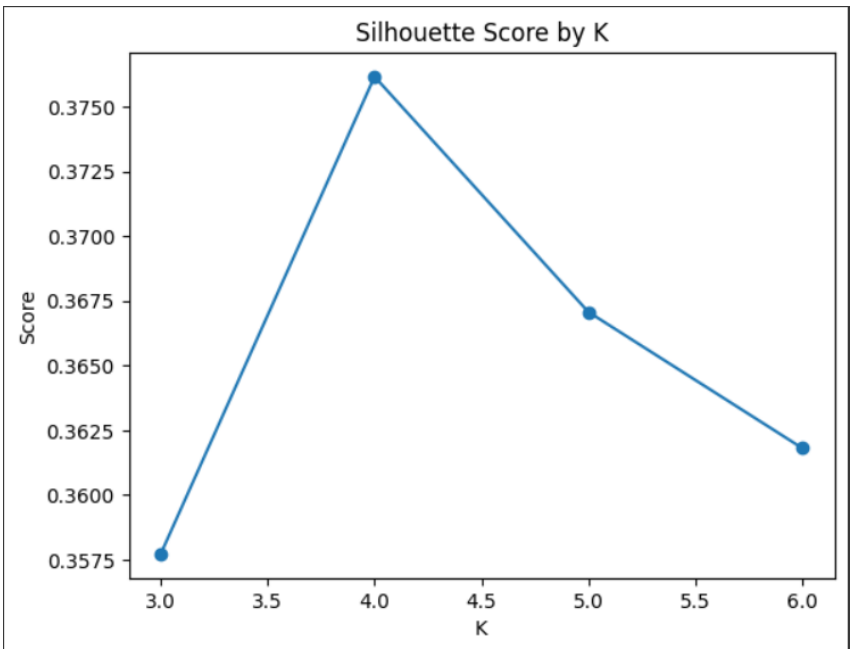

Fig 2:- result

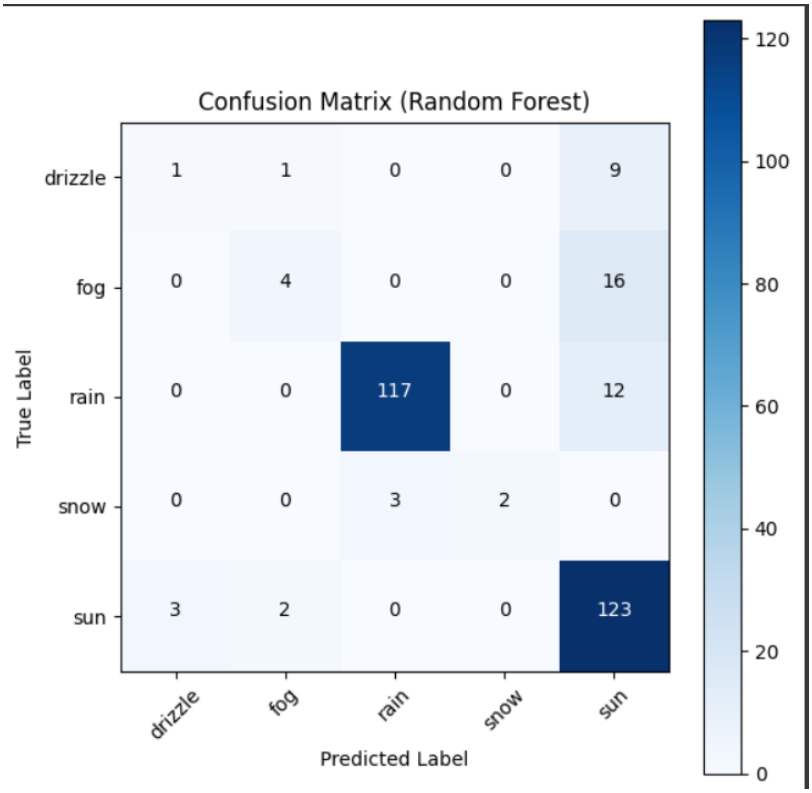

Fig3 :-predict label

```
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