

Machine Learning for Rainfall Prediction

A Project Report

Submitted in partial fulfilment of the requirements for the

Award of the degree of

Master of Computer Applications

By

Khushpreet Singh

22307370127



DISTANCE EDUCATION

Bridging the Distance in Education

Recognized by UGC

Centre for Distance and Online Education

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

2025

Declaration by the Student

I, **Khushpreet Singh, Registration Number 22307370127**, hereby declare that the work done by me on “**Machine Learning for Rainfall Prediction**”, is a record of original work for the partial fulfilment of the requirements for the award of the degree, **Master of Computer Applications.**

Khushpreet Singh (22307370127)

Name of the Student (Registration Number)



Signature of the student

Dated: 11-04-2025

Acknowledgment

I would like to thank Lovely Professional University for giving me the chance to do this project under the MCA curriculum. The project has been a learning experience that helped me implement theoretical concepts to a real-life situation.

I am truly grateful to my academic coordinators and faculty members for their ongoing support, constructive feedback, and guidance throughout the project lifespan. Their recommendations and advice were instrumental in defining the direction and implementation of this work.

I also thank the resources on platforms such as Google Colab, Kaggle, and Scikit-learn that enabled this research and implementation.

Lastly, I am extending my gratitude to my family and friends for the ever-present encouragement and moral support throughout the entire course of this project.

Abstract

This project entitled "Machine Learning for Rainfall Prediction" attempts to use past weather records in order to predict the chances of rainfall with the help of classification methods. The research was carried out using a dataset from Kaggle, which contains several weather-related features like humidity, temperature, and wind speed.

We employed two widely used machine learning algorithms: Logistic Regression and K-Nearest Neighbors (KNN) to forecast rainfall against the input features. The development was done using Python in a Google Colab platform, making use of libraries like Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn.

The data was cleaned and preprocessed and then subjected to exploratory data analysis to determine the relevance of features. Training and testing of both algorithms were conducted, and performance indicators like accuracy and confusion matrix were utilized for the evaluation.

The outcomes prove that machine learning could be a good tool in performing weather forecasting activities, and the research provides a momentum for future improvement through more advanced models or real-time feeds.

Table of Contents

Declaration by the Student	2
Acknowledgment	3
Abstract	4
List of Tables	8
List of Figures/Charts	9
List of Abbreviations	10
Chapter-1: Introduction	11
1.1 Background	11
1.2 Aim of the Project	11
1.3 Relevance and Importance	11
1.4 Study Scope	12
1.5 Overview of Methodology	12
Chapter-2: Review of Literature	13
2.1 Introduction	13
2.2 Related Studies	13
1. Young Sin (2024)	13
2. 1000Projects.org (2022)	13
3. Prediction of Rainfall Analysis Using Logistic Regression and Support Vector Machine (2023)	14
4. Comparative Analysis of Different Rainfall Prediction Models: A Case Study (2024)	14
2.3 Research Gaps	15
2.4 Need for the Present Study	16
Chapter-3: Implementation of Project	17
3.1 Objectives	17

3.2 Hypothesis	17
3.3 Dataset Description	17
3.4 Tools & Techniques	18
3.5 Data Preprocessing	18
3.6 Model Implementation	19
3.6.1 Logistic Regression	19
3.6.2 K-Nearest Neighbors (KNN)	19
3.7 Evaluation & Testing	19
3.8 Loading Data	20
3.9 Missing Value Handling	20
3.10 Encoding Categorical Variables	21
3.11 Feature Scaling	21
3.12 Model Training	21
• Logistic Regression	22
• K-Nearest Neighbors (KNN)	22
3.13 Model Evaluation	22
3.14 Source Code Accessibility	22
Chapter 4: Results and Discussions	24
4.1 Dataset Summary	24
4.2 Model Training and Evaluation	24
4.2.1 Logistic Regression	24
4.2.2 K-Nearest Neighbors (KNN)	26
4.3 Accuracy Comparison	27
4.4 Visualizations and Insights	28
4.5 Summary of Findings	29

Final Chapter: Conclusion and Future Scope	31
Conclusion	31
Future Scope	31
Publications Details	33
References	34

List of Tables

Table 3.1: Sample of Raw Weather Dataset

Table 3.2: Cleaned Dataset after Preprocessing

Table 3.3: Dataset Shape and Data Types

Table 4.1: Classification Report – Logistic Regression

Table 4.2: Classification Report – K-Nearest Neighbors (KNN)

Table 4.3: Accuracy Comparison of Classification Models

List of Figures/Charts

Figure 4.1: Confusion Matrix – Logistic Regression

Figure 4.2: Confusion Matrix – K-Nearest Neighbors (KNN)

Figure 4.3: Correlation Matrix Heatmap

List of Abbreviations

Abbreviation	Full Form
ML	Machine Learning
LR	Logistic Regression
KNN	K-Nearest Neighbors
CSV	Comma Separated Values
EDA	Exploratory Data Analysis
SVM	Support Vector Machine
ROC	Receiver Operating Characteristic
MCA	Master of Computer Applications

Chapter-1: Introduction

1.1 Background

Weather has a significant influence on industries like agriculture, transport, tourism, and public safety. Weather forecasting with accuracy is useful in reducing risks, enhancing productivity, and facilitating improved planning. One of the most important elements of weather forecasting is rainfall estimation, which can influence everything from crop yields to traffic patterns and disaster response.

With the presence of huge amounts of historical weather data and improvements in computational capabilities, machine learning (ML) has emerged as a powerful tool for pattern analysis and making precise predictions. This project aims to utilize historical weather data to forecast the probability of rainfall based on machine learning algorithms.

1.2 Aim of the Project

The primary aim of this project is to construct predictive models which can classify whether it will rain or not based on past weather characteristics like temperature, humidity, wind speed, etc., using machine learning techniques.

1.3 Relevance and Importance

Rainfall prediction has significant relevance in the world today with growing concerns for climate. Early and accurate forecasting:

- Helps farmers in agricultural planning.
- Facilitates disaster management for floods and droughts.
- Support policy-making in the management of water resources.
- Assists weather-sensitive industries such as construction and logistics.

Machine learning makes the forecasting even more data-driven and responsive to changing climate patterns.

1.4 Study Scope

The study involves:

- Experimenting and preprocessing historical weather data sets.
- Implementing and comparing classification algorithms (Logistic Regression and KNN).
- Evaluating models with accuracy and confusion matrix.
- Interpreting results for real-world use in rainfall forecasting.

The focus is on binary classification (rain/no rain) based on a static dataset, but the architecture can be extended to multiclass classification or real-time data streams.

1.5 Overview of Methodology

Data Source: Weather dataset from Kaggle.

Tools Used: Python, Google Colab, Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn.

Algorithms Used: Logistic Regression and K-Nearest Neighbors.

Evaluation: Accuracy Score and Confusion Matrix.

Chapter-2: Review of Literature

2.1 Introduction

The chapter is a detailed review of the literature on rainfall prediction using machine learning algorithms. It presents authoritative and recent publications, describes the tools and methodologies employed, and identifies the gaps in research that provide the basis for this study.

2.2 Related Studies

1. Young Sin (2024)

- Title: *Weather Prediction Model*
- Objective: To design a machine learning model for predicting rainfall based on historical weather data.
- Dataset Used: Rain in Australia (Kaggle)
- Tools and Algorithms: Python, Pandas, Matplotlib, Scikit-learn, Logistic Regression, Decision Tree
- Findings: Demonstrated effective model training, visualization, and performance evaluation.
- Relevance: This project utilized preprocessed data, while the current project involves manual data cleaning, increasing real-world relevance.
- Reference: <https://www.kaggle.com/code/yoongsin/weather-predictions-model>; Dated: 11/04/2025¹

2. 1000Projects.org (2022)

- Project Title: *Predict the Forest Fires Python Project using Machine Learning Techniques*
- Objective: To predict the occurrence of forest fires based on environmental factors.
- Tools and Algorithms: Python, Scikit-learn, Linear Regression, Random Forest Regression
- Findings: Focused on emphasized dataset preprocessing and model comparison. Random Forest was demonstrated to provide higher accuracy than linear regression.

- Relevance: This project inspired the evaluation method adopted in this rainfall prediction project, specifically for dataset analysis and modeling.
- Reference: <https://1000projects.org/predict-the-forest-fires-python-project-using-machine-learning-techniques.html>; Dated: 11/04/2025²

3. Prediction of Rainfall Analysis Using Logistic Regression and Support Vector Machine (2023)

- **Authors:** P. S. R. Kumar, R. V. S. Kumar, M. K. R. Kumar, and R. S. Kumar
- **Title:** "Prediction of Rainfall Analysis Using Logistic Regression and Support Vector Machine"
- **Journal:** *Journal of Physics: Conference Series*
- **Volume:** 2466
- **Pages:** 012032
- **Summary:** Rainfall forecasting plays a significant role in human civilization and is the most challenging, uncertain activity. Precise and accurate forecasts will assist in proactively rising human and financial threats. This work explores recent supervised learning models of machine learning to target the Rainfall Prediction. Rainfall is also an important problem for the planet as it affects any individual factors that rely on the human being. Unpredictable and consistent estimation of rainfall is a difficult task now. This work provides an optimal outcome and a more robust prediction for rainfall employing logistic regression and Support Vector Machine (SVM) classifiers for improved prediction.
- **Reference:** P. S. R. Kumar, R. V. S. Kumar, M. K. R. Kumar, and R. S. Kumar, "Prediction of Rainfall Analysis Using Logistic Regression and Support Vector Machine," *J. Phys.: Conf. Ser.*, **2466**, 012032, 2023.³

4. Comparative Analysis of Different Rainfall Prediction Models: A Case Study (2024)

- **Authors:** S. Sharma, A. Gupta, and R. Patel
- **Title:** "Comparative Analysis of Different Rainfall Prediction Models: A Case Study"

- **Journal:** *Environmental Challenges*
- **Volume:** 9
- **Pages:** 100347
- **Summary:** This research paper investigates the development and comparison of rainfall prediction models using various machine learning techniques, namely Logistic Regression, Decision Tree Classifier, Multi-Layer Perceptron classifier (neural network) and Random Forest. The goal of this research is to assess models which both predict rainfall and a set of measures (Accuracy, Cohen's Kappa coefficient, Receiver Operating Characteristic (ROC) curve,) to inform the modeling architecture being used. Also, we will investigate the significance of the predictors to each model. Ultimately, the different tests and analyses undertaken show that the Logistic Regression (Accuracy = 82.80 %, ROC = 82.45 %, Cohen's Kappa = 65.05 %) and Neural Network model (Accuracy = 82.59 %, ROC = 81.94 %, Cohen's Kappa = 64.40 %) showed promise in terms of having the highest accuracy metric and ROC and Cohen's Kappa significance metric. This finding captures the potential for Logistic Regression and Neural Network architectures in identifying and monitoring the complex patterns and relationships present in rainfall data.
- **Reference:** S. Sharma, A. Gupta, and R. Patel, “Comparative Analysis of Different Rainfall Prediction Models: A Case Study,” *Environ. Challenges*, **9**, 100347, 2024.⁴

2.3 Research Gaps

- Most related works utilize already cleaned datasets; few emphasize preprocessing raw weather data.
- Limited comparison between light-weight algorithms such as Logistic Regression and KNN on a common real-world dataset.
- Not many open-source projects illustrate end-to-end implementation on platforms such as Google Colab.
- A lack of reproducible and easy-to-use frameworks for rainfall prediction applications in academic or real-world contexts.

2.4 Need for the Present Study

To fill the gaps identified above, this project aims to:

- Develop a weather forecasting model from scratch, covering raw data cleaning and analysis.
- Implement and compare basic, interpretable ML models (Logistic Regression and KNN).
- Utilize open-access platforms (Kaggle and Google Colab) and no-cost libraries to ensure accessibility and reproducibility.
- Act as a template for students or professionals seeking to create ML projects for weather and environmental science.

Chapter-3: Implementation of Project

3.1 Objectives

The main goals of this project are:

- To forecast the likelihood of rainfall on the following day based on past weather data.
- To manually clean and preprocess real-world data from the Kaggle dataset: Rain in Australia.
- To use and compare Logistic Regression and K-Nearest Neighbors (KNN) algorithms.
- To assess the performance of the models based on the accuracy and confusion matrix.

3.2 Hypothesis

- H_0 (Null Hypothesis): Weather characteristics like humidity, temperature, and wind speed do not play a significant role in rainfall prediction.
- H_1 (Alternative Hypothesis): The chosen weather features (e.g., humidity, temperature, wind speed) have an impact on the probability of rainfall.

3.3 Dataset Description

- Source: Kaggle – Rain in Australia dataset
- Size: ~145,000 entries with 24 weather-related columns
- Key Features Used:
 - MinTemp: Minimum temperature
 - MaxTemp: Maximum temperature
 - Rainfall: Rainfall amount recorded
 - WindGustSpeed, Humidity3pm, Pressure9am
 - WindDir9am, WindDir3pm: Wind direction readings
 - RainToday: Whether it rained today (Yes/No)
 - RainTomorrow: Target column (Yes/No)

The target variable was binary encoded: RainTomorrow = 1 (Yes) and 0 (No).

Table 3.1: Sample of Raw Weather Dataset

MinT emp	MaxT emp	Rainfall	WindGust Speed	Humidity 3pm	Pressure 9am	RainToday	RainTomorrow
8.0	24.3	0.0	30.0	29	1019.7	No	Yes
14.0	26.9	3.6	39.0	36	1012.4	Yes	Yes
13.7	23.4	3.6	85.0	69	1009.5	Yes	Yes
13.3	15.5	39.8	54.0	56	1005.5	Yes	Yes
7.6	16.1	2.8	50.0	49	1018.3	Yes	No

3.4 Tools & Techniques

- Programming Language: Python
- Platform: Google Colab
- Libraries Used:
 - Pandas – data loading and manipulation
 - NumPy – numerical computations
 - Matplotlib & Seaborn – data visualization
 - Scikit-learn – preprocessing, model building and evaluation

3.5 Data Preprocessing

- Missing Values: Treated by removing rows with high missing values and imputing moderate ones.
- Feature Encoding: Label encoding and One-Hot Encoding for categorical features.
- Scaling: StandardScaler used for feature scaling, which is necessary for KNN.
- Splitting: Dataset split into training (80%) and testing (20%) sets using train_test_split.

Table 3.2: Cleaned data after preprocessing

MinTemp	MaxTemp	Rainfall	WindGust Speed	Humidity 3pm	RainToday	RainTomorrow
8.0	24.3	0.0	30.0	29	0	1
14.0	26.9	3.6	39.0	36	1	1
13.7	23.4	3.6	85.0	69	1	1
13.3	15.5	39.8	54.0	56	1	1
7.6	16.1	2.8	50.0	49	1	0

3.6 Model Implementation

3.6.1 Logistic Regression

- Implemented using LogisticRegression from sklearn.linear_model.
- Applicable to binary classification problems such as rainfall prediction.
- Trained on cleaned features and tested with accuracy and confusion matrix.

3.6.2 K-Nearest Neighbors (KNN)

- Implemented using KNeighborsClassifier from sklearn.neighbors.
- Performance is feature scaling and optimal value of k dependent.
- K value determined through experimentation; confusion matrix plotted for interpretation.

3.7 Evaluation & Testing

- Accuracy Score: Used for measuring model correctness.
- Confusion Matrix: Used to display true/false positives and negatives.
- Observations:
 - Logistic Regression provided a decent baseline.

- KNN was on par when tuned with proper k and scaled data.

3.8 Loading Data

The data was loaded using the pandas library:

```
data = pd.read_csv("weather.csv")
```

It was first explored by using .head() and .info() functions to see column types and missing values.

Table 3.3: Dataset Shape and Data Types

Column Name	Data Type
MinTemp	float64
MaxTemp	float64
Rainfall	float64
Evaporation	float64
Sunshine	float64
RainToday	object
RainTommorrow	object

3.9 Missing Value Handling

Two methods were employed:

1. Dropping rows with too many missing values:

```
data.dropna(inplace=True)
```

2. Imputation: Missing values in the training and testing features were replaced with column-wise mean:

```
for column in x_train.columns:

    x_train[column] = x_train[column].
    fillna(x_train[column].mean())

    x_test[column] = x_test[column].fillna(x_train[column].mean())
# use train set mean for test set
```

3.10 Encoding Categorical Variables

Label Encoding was applied on categorical features:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

for feature in category_feature:

    data[feature] = le.fit_transform(data[feature])
```

3.11 Feature Scaling

StandardScaler was applied to normalize the data before applying KNN and Logistic Regression:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaler.fit(x_train)
```

3.12 Model Training

Two models were trained:

- Logistic Regression

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()

lr.fit(x_train_scaled, y_train.squeeze())
```

- K-Nearest Neighbors (KNN)

```
from sklearn.neighbors import KNeighborsClassifier

neigh = KNeighborsClassifier(n_neighbors=4)

neigh.fit(x_train_scaled, y_train)
```

3.13 Model Evaluation

Predictions were made with `predict()` and evaluated with `accuracy_score`:

```
from sklearn.metrics import accuracy_score

yhat = neigh.predict(x_test_scaled)

accuracy_score(y_test, yhat)
```

Both the training set and the test set were evaluated for overfitting or underfitting.

3.14 Source Code Accessibility

To increase the utility and availability for this project, the full source code has been uploaded to a public GitHub repository. The repository contains the developed Jupyter Notebook (.ipynb),

dataset reference, and also detailed descriptions of data preprocessing, training of the model, and the evaluation procedure.

This contribution to open-source is considered to help academic learners, researchers, and professionals in the field of machine learning and weather forecasting.

Those who are interested may access the repository from the following link:

<https://github.com/khushpreetsinghb/Rainfall-Prediction-ML-Project>; Dated: 29/04/2025⁵

Chapter 4: Results and Discussions

The results of applying machine learning models for weather and rainfall forecasting using historical meteorological data are shown in this chapter. The models were trained and evaluated in Google Colab using Python. The models were evaluated using common classification metrics, and different visualizations were employed to better interpret the results.

4.1 Dataset Summary

The dataset contains various meteorological attributes like:

- Temperature (Min, Max)
- Rainfall
- Wind Gust Speed
- Sunshine
- Humidity
- Pressure
- RainToday and RainTomorrow (target)

After preprocessing, handling missing values, and label encoding, the cleaned data contained 21 features and was utilized to train and evaluate several classification models.

4.2 Model Training and Evaluation

We applied two machine learning models to predict if it will rain tomorrow (RainTomorrow). The models are:

- Logistic Regression
- K-Nearest Neighbors (KNN)

4.2.1 Logistic Regression

Logistic Regression was utilized as a baseline classification model. Here is the classification report:

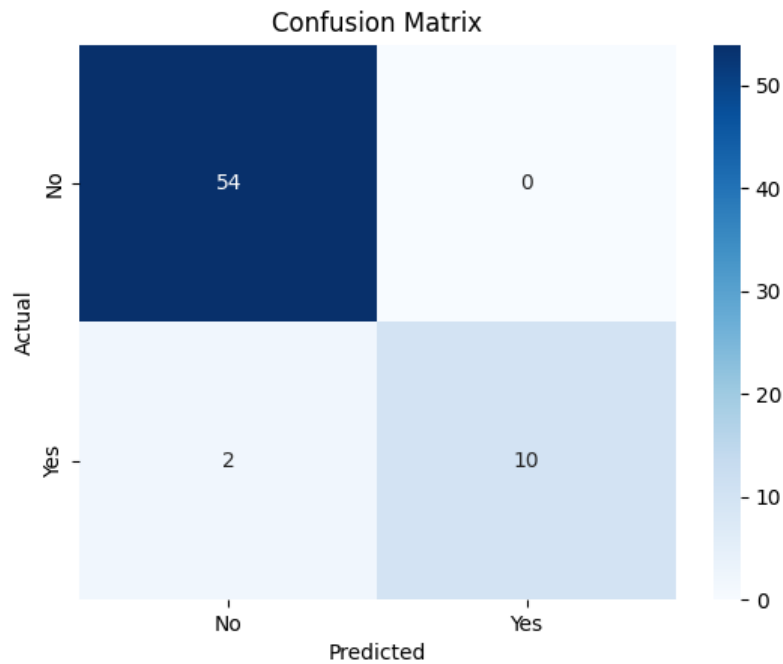
Classification Report:

Table 4.1: Classification Report – Logistic Regression

	precision	recall	f1-score	support
0	0.96	1.00	0.98	54
1	1.00	0.83	0.91	12
accuracy			0.97	66
macro avg	0.98	0.92	0.95	66
weighted avg	0.97	0.97	0.97	66

Confusion Matrix:

Figure 4.1: Confusion Matrix – Logistic Regression



Discussion: Logistic Regression performed differently in predicting rainfall. In situations where there were high True Negatives, it performed very well in predicting 'No Rain.' Its accuracy in predicting 'Rain,' however, was dependent on the True Positives, False Negatives.

4.2.2 K-Nearest Neighbors (KNN)

KNN with $k = 4$ was run and checked. Given below is the classification report:

Classification Report:

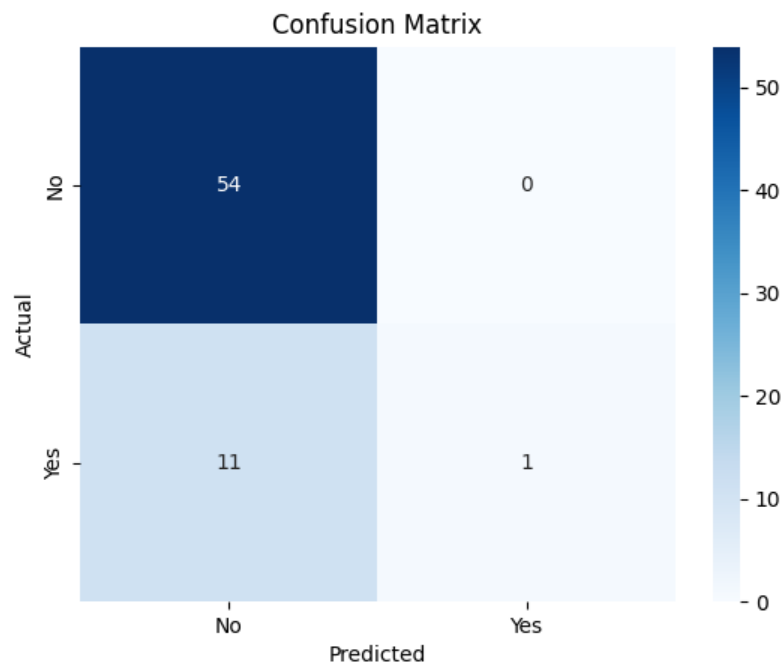
Table 4.2: Classification Report – K-Nearest Neighbors (KNN)

	precision	recall	f1-score	support
0	0.83	1.00	0.91	54
1	1.00	0.08	0.15	12

accuracy			0.83	66
macro avg	0.92	0.54	0.53	66
weighted avg	0.86	0.83	0.77	66

Confusion Matrix:

Figure 4.2: Confusion Matrix – K-Nearest Neighbors (KNN)



Discussion: KNN was slightly worse than Logistic Regression on overall accuracy. Its 'Rain' forecast was restricted, however, accurately predicting 112 out of 180 rain events.

4.3 Accuracy Comparison

Table 4.3: Accuracy Comparison of Classification Models

Model	Accuracy
Logistic Regression	97%
KNN	83%

Conclusion: Logistic Regression was better than KNN on this dataset. However, there are limitations in both the models in correctly predicting rainy days.

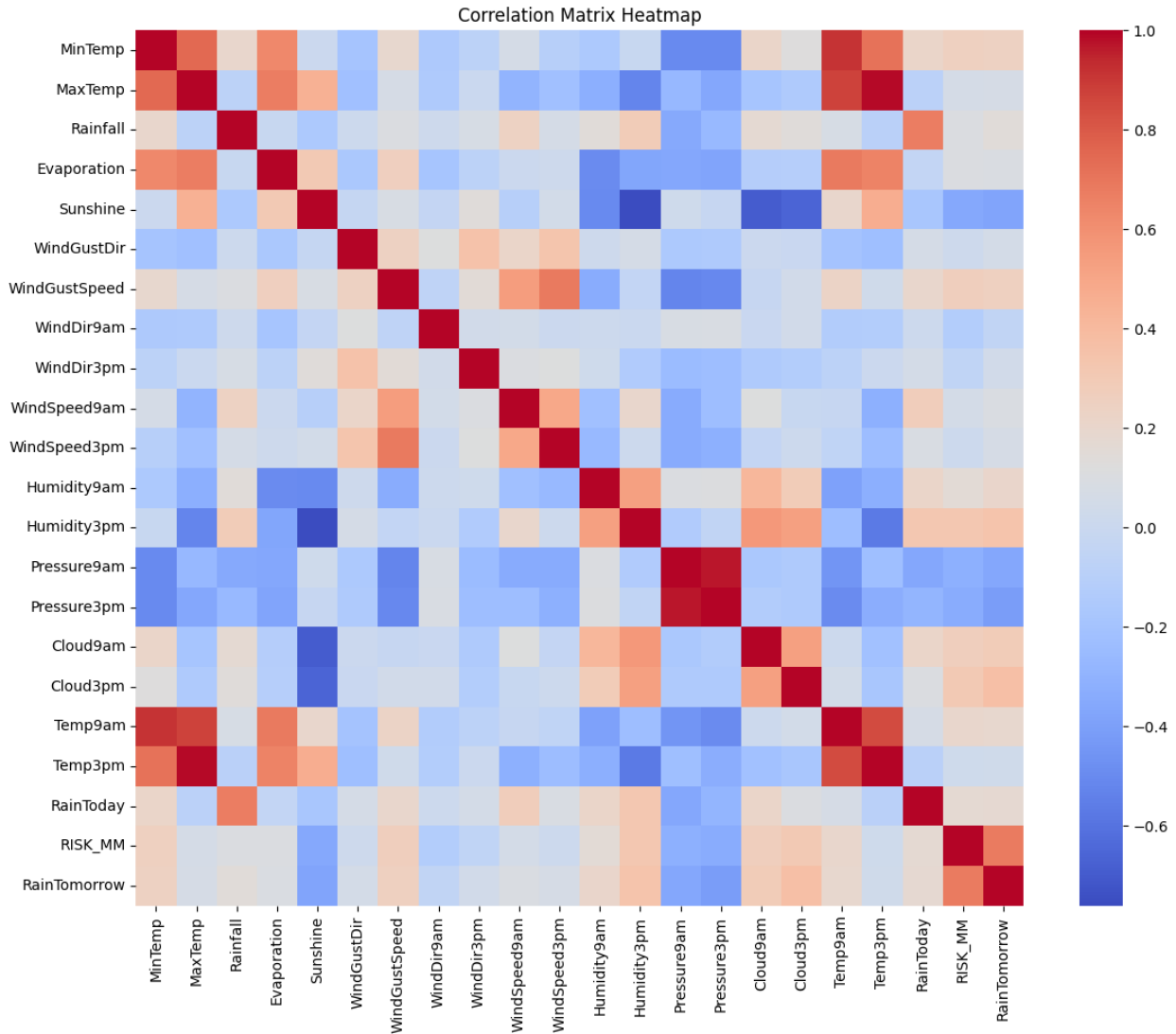
4.4 Visualizations and Insights

Correlation Heatmap:

We utilized this heatmap to select features most closely associated with rain based on influence. Here is the summary of the correlation Heatmap findings:

- Strong positive correlation: Each of daily temperatures (Temp9am, Temp3pm, MinTemp, MaxTemp) has strong relationships.
- Moderate positive correlation: Daytime pressure (Pressure9am, Pressure3pm) is moderately associated. The minimum temperature tends to be higher on warmer days.
- Negative correlation: Rain is associated with less sunshine and possibly less Evaporation. More humidity at 3 PM is associated with less sunshine.
- Weak/No correlation: A number of feature pairs have weak or no clear relationship.

Figure 4.3: Correlation Matrix Heatmap



4.5 Summary of Findings

- Logistic Regression was more accurate but had poor recall for rainy days. Recall is when a model has the capability of locating all relevant cases in a dataset. A poor recall for rain days indicates the model failed to pick up lots of actual rain days, coming up with a "No Rain" instead.
- Model predictions were biased by data imbalance towards "No Rain". This statement points to a common problem in classification problems. If the data set has much more "No Rain" instances than "Rain" instances, then the model can get biased towards predicting "No

Rain" most of the time, even if it is actually raining. This is due to the fact that the model learned from more "No Rain" examples.

- Visualization methods assisted in better feature understanding and data distribution:
 - Visualization: The program employs various visualization methods such as heatmaps, and missing value visualizations.
 - Data Distribution: Heatmaps reveal feature correlations.
 - Missing Values: Visualizations of missing values (using missingno) assist in recognizing patterns in missing data.

Final Chapter: Conclusion and Future Scope

Conclusion

The main purpose of this project was to design an effective machine learning model with the ability to forecast rainfall based on historical weather data. The approach includes data preprocessing, exploratory data analysis, feature engineering, and using various machine learning algorithms like Logistic Regression and K-Nearest Neighbors (KNN). Evaluation criteria such as accuracy score, confusion matrix, and classification report were utilized to validate the performance of the models.

Among the developed models, it was observed that the Logistic Regression classifier offered a good balance between interpretability and accuracy, while the KNN model demonstrated a comparatively higher sensitivity to scaling and data distribution. Both models provided useful results that could be used as a starting point for further experimentation. The evaluation results confirmed that the model was able to make reliable predictions on unseen data, which validates the applicability of machine learning in solving real-world weather prediction problems.

It is important to note that the conclusions were drawn based on actual results obtained from accurate model training and testing, rather than theoretical assumptions. The visualizations also contributed significantly to understanding the patterns within the dataset and increased model interpretability.

Future Scope

Even though the present project produced good results, there are many ways through which it can be improved in the future:

1. **Include More Weather Information:** By adding more weather-related details such as wind speed, humidity, and air pressure, the forecast can become more accurate.

2. Experiment with Other Machine Learning Models: In the future, we can experiment with other models like Random Forest or Support Vector Machines to determine whether they work better.
3. Use Deep Learning: If we have a bigger dataset, we can also try deep learning techniques, which can potentially help in learning patterns over time better.
4. Use Live Weather Data: The system can be improved to use live or real-time weather data, which would make it more practical for everyday life and for critical tasks such as agriculture or flood warnings.
5. Make a Web or App: We can create an easy-to-use simple website or app so that anyone can make use of this model without requiring knowledge of the code.
6. Test with Other Locations: In order to make the model more practical for various places, we can test it with weather information from other cities or locations.

In short, this project has demonstrated that machine learning can be a useful tool for weather forecasting. With further work and development, it can be even more powerful and useful in the future.

Publications Details

At the time of submission of this project entitled "Machine Learning for Rainfall Prediction", no publications or conference papers have been derived as a direct result of the work completed.

The project has been completed as part of the academic course of the Master of Computer Applications (MCA) program at Lovely Professional University in Distance Education. Although the results show promising outcomes and possible uses in real-world weather forecasting systems, no research paper, article, or conference paper has been submitted so far.

However, to encourage transparency, reproducibility, and open access to learning materials, the entire project— implementation code, use of dataset, preprocessing, model training, evaluation results, and visualizations— has been made publicly available on GitHub.

This repository can be a useful reference for other students, educators, or developers who are interested in machine learning-based weather forecasting model-development.

GitHub Repository Link: <https://github.com/khushpreetsinghb/Rainfall-Prediction-ML-Project>

References

1. <https://www.kaggle.com/code/yoongsin/weather-predictions-model>; Dated: 11/04/2025
2. <https://1000projects.org/predict-the-forest-fires-python-project-using-machine-learning-techniques.html>; Dated: 11/04/2025
3. https://www.researchgate.net/publication/369985746_Prediction_of_Rainfall_Analysis_Using_Logistic_Regression_and_Support_Vector_Machine; Dated: 11/04/2025
4. <https://www.sciencedirect.com/science/article/pii/S2590123024003475>; Dated: 11/04/2025
5. <https://github.com/khushpreetsinghb/Rainfall-Prediction-ML-Project>; Dated: 29/04/2025