MA304 Multivariate Statistical Analysis
Multivariate Rainfall Prediction

Yuhang Huang 12213015
Haochen Ni 12213027
Shiqi Li 12211031

May 25, 2025

Abstract—This report investigates the application of **dimensionality reduction techniques** combined with **machine learning** (**ML**) models to improve **rainfall probability prediction** using **multivariate meteorological data**. **Principal Component Analysis** (**PCA**) and **Factor Analysis** (**FA**) are employed to extract key **latent features**, while classification models including **Logistic Regression** (**LR**), **Support Vector Machines** (**SVM**), and **Binary Neural Networks** (**BNNs**) are trained on both original and reduced datasets. **K-means clustering** is also conducted to reveal inherent patterns associated with rainfall events.

Empirical results demonstrate that **FA** significantly enhances model **interpretability** and **computational efficiency**, especially when combined with **LR**, which achieves the highest **accuracy** and **AUC** (Area Under the Curve). Cross-validation confirms the **robustness** and **generalizability** of the findings. This study highlights the importance of appropriate **dimensionality reduction strategies** in **predictive modeling** and provides insights for developing efficient and interpretable **rainfall forecasting systems**.

1 Introduction

1.1 Background

Rainfall prediction entails the utilization of various measurable meteorological parameters to determine the likelihood of precipitation within a specific region. Accurate and timely forecasts are of critical importance across a range of domains, including agriculture, transportation, water resource management, energy production, disaster mitigation, and emergency response.[1] Such forecasts contribute to minimizing economic losses, safeguarding the environment, and enhancing overall quality of life.

1.2 Literature Review

Since rainfall affects many aspects of human life, numerous rainfall prediction models have been developed in the past. Below is a brief overview of the widely adopted types of models classified by technology.

1.2.1 Extrapolation Techniques from Remote Sensing Observations

These techniques use **radar** or **satellite imagery** to extrapolate rainfall based on apparent motion detected in recent observations. They are generally divided into **object-based** and **pixel-based** methods.

Object-based methods treat rainfall systems as discrete entities and are effective in tracking thunderstorms, with examples including **TITAN**, **SCIT**, and **MAPLE** [2]. In contrast, **pixel-based methods** estimate motion at the pixel level, as implemented in tools like **PySTEPS** and **rainymotion**, which are particularly suitable for stratiform rain systems [3].

While these techniques provide high **spatial-temporal resolution**, their primary limitation lies in the inability to capture cloud **development processes**, leading to reduced accuracy for **convective systems** at extended **forecast lead times** [4].

1.2.2 Numerical Weather Prediction (NWP) Models

Numerical Weather Prediction (NWP) models solve systems of **partial differential equations** representing atmospheric **dynamics** and **thermodynamics**. These models assimilate meteorological observations

(e.g., **pressure**, **temperature**, **humidity**) to simulate future atmospheric states. Major NWP categories include **global circulation models (GCMs)**, **limited-area models (LAMs)**, and **convection-permitting models (CPMs)** [5]. **Ensemble prediction systems (EPSs)** enhance forecast reliability by generating multiple realizations through perturbations of initial conditions or model physics. While NWP models provide physically consistent forecasts for **medium- to long-range** predictions, their substantial **computational requirements** pose challenges for real-time applications [6].

1.2.3 Stochastic Models

Stochastic models characterize rainfall variability through statistical representations, making them particularly effective for **high-resolution** forecasting at **short temporal scales**. This category encompasses **point process models** (e.g., **Poisson**, **Neyman-Scott**, **Bartlett-Lewis**) [7], **time series approaches** including **ARMA** and **STARMA** [8], as well as **hybrid models** that independently parameterize rainfall occurrence and intensity [9]. These methods offer **computational efficiency** for small catchment applications, though their **Markovian assumptions** constrain predictive initiation capabilities (requiring existing storm observations) and typically demand extensive historical data for robust parameter estimation [10].

1.2.4 Deep Learning Models

Deep learning models utilize large-scale datasets from **radar**, **satellite**, and **ground station** networks to capture the nonlinear **spatiotemporal patterns** of rainfall. These models encompass several architectures: **convolutional neural networks (CNNs)** including **ConvLSTM** and **U-Net**, **recurrent neural networks (RNNs)** such as **PredRNN** and **TrajGRU**, and **generative adversarial networks (GANs)** exemplified by **DGMR** and **GA-ConvGRU**. While demonstrating superior **pattern recognition** capabilities and rapid **inference speed** for nowcasting applications, these models exhibit limitations including generation of **blurred outputs** during extreme events and lack of **physical constraints**, which may compromise their **interpretability** and **generalizability** beyond training conditions [11].

1.2.5 Blended Models

Blended models synergistically combine multiple modeling approaches to capitalize on their complementary strengths. Representative implementations include: the integration of **radar extrapolation** with **NWP** (e.g., **STEPS**, **INCA**), coupling of **stochastic models** with **NWP** frameworks (e.g., **PRAISE-MET**), and **post-processing** of NWP outputs using deep learning techniques. Emerging **physics-informed machine learning (PIML)** methodologies, such as **PINNs**, **LPT-QPN**, and **LUPIN**, directly embed physical laws within data-driven architectures. These hybrid approaches demonstrate enhanced **forecast accuracy**, extended **prediction horizons**, and reduced **computational demands**, positioning them as a transformative paradigm for developing robust **nowcasting systems** [12].

1.3 Structure of this Report

This report is organized into four main sections. **Section 2** details the methodological framework, presenting three key components: (1) **dimensionality reduction** techniques including **Principal Component Analysis (PCA)** and **Factor Analysis (FA)**, (2) **K-means clustering** for pattern discovery, and (3) classification algorithms comprising **Logistic Regression (LR)**, **Support Vector Machines (SVM)**, and **Binary Neural Networks (BNNs)**, with complete theoretical foundations and implementation specifications.

Section 3 conducts comprehensive empirical analysis using the **Kaggle rainfall dataset**, structured in four phases: (1) dataset characteristics and preprocessing, (2) dimensionality reduction outcomes, (3) clustering-derived insights, and (4) comparative performance evaluation of prediction models across multiple metrics.

Section 4 synthesizes the research findings, highlighting the superior performance of **FA-LR integration**, discussing practical implications for meteorological forecasting systems, and outlining promising research directions. The analysis is supplemented with **interactive visualizations** and **model architecture schematics** in the appendices to enhance reproducibility and insight generation.

2 Methods

2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique that projects high-dimensional data onto a lower-dimensional subspace while preserving maximum variance. Given a centered data matrix $\mathbf{X}_{\text{centered}} = \mathbf{X} - \boldsymbol{\mu}$, where $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ is the mean vector, PCA computes the covariance matrix $\boldsymbol{\Sigma} = \frac{1}{n}\mathbf{X}_{\text{centered}}^{\top}\mathbf{X}_{\text{centered}}$. Through eigendecomposition $\boldsymbol{\Sigma} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\top}$, where $\mathbf{V}$ contains the eigenvectors and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d)$ contains the eigenvalues ($\lambda_1 \geq \cdots \geq \lambda_d$), PCA selects the top-$k$ eigenvectors $\mathbf{V}_k$ corresponding to the largest eigenvalues. The reduced-dimensional representation is obtained by $\mathbf{Z} = \mathbf{X}_{\text{centered}}\mathbf{V}_k \in R^{n\times k}$, where the columns of $\mathbf{Z}$ are called principal components.

2.2 Factor Analysis

Factor Analysis (FA) models observed variables $\mathbf{x} \in R^p$ as linear combinations of latent common factors $\mathbf{f} \in R^k$ ($k \ll p$) and unique factors $\boldsymbol{\epsilon} \in R^p$, expressed as $\mathbf{x} = \boldsymbol{\mu} + \mathbf{L}\mathbf{f} + \boldsymbol{\epsilon}$ where $\mathbf{L}$ is the factor loading matrix and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$ with diagonal $\boldsymbol{\Psi}$. The covariance matrix decomposes as $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^{\top} + \boldsymbol{\Psi}$, with parameters estimable via principal factor method (extracting top-k eigenvectors of sample covariance $\mathbf{S}$ as $\mathbf{L} = [\sqrt{\lambda_1}\mathbf{v}_1 \cdots \sqrt{\lambda_k}\mathbf{v}_k]$) or maximum likelihood estimation (maximizing $\mathcal{L} = -\frac{n}{2}[\log|\boldsymbol{\Sigma}| + \text{tr}(\mathbf{S}\boldsymbol{\Sigma}^{-1})]$), often followed by factor rotation (e.g., Varimax) for enhanced interpretability.

2.3 K-means Clustering

K-means clustering partitions n observations $\{\mathbf{x}_i \in R^d\}_{i=1}^{n}$ into k disjoint clusters $\{C_j\}_{j=1}^{k}$ by minimizing the within-cluster sum of squares (WCSS): $\min \sum_{j=1}^{k}\sum_{\mathbf{x}\in C_j} \|\mathbf{x} - \boldsymbol{\mu}_j\|^2$, where $\boldsymbol{\mu}_j$ represents the centroid of cluster j. The algorithm iteratively assigns points to nearest centroids ($C_j = \{\mathbf{x} : \|\mathbf{x} - \boldsymbol{\mu}_j\| \leq \|\mathbf{x} - \boldsymbol{\mu}_l\| \; \forall l \neq j\}$) and updates centroids as the mean of assigned points ($\boldsymbol{\mu}_j = \frac{1}{|C_j|}\sum_{\mathbf{x}\in C_j}\mathbf{x}$) until convergence. The solution depends on initial centroid positions, typically addressed via multiple random initializations (k-means++ initialization provides improved seeding). The objective function is non-convex, guaranteeing only local optima.

2.4 Logistic Regression

Logistic regression models the probability $p(\mathbf{x})$ of a binary outcome $y \in \{0, 1\}$ using the logistic function $\sigma(z) = (1 + e^{-z})^{-1}$, where the log-odds are linear in the predictors $\mathbf{x} \in R^d$: $\log\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \boldsymbol{\beta}^\top \mathbf{x} + \beta_0$. The model parameters $\boldsymbol{\beta} \in R^d$ and intercept $\beta_0 \in R$ are typically estimated via maximum likelihood estimation (MLE), maximizing the log-likelihood function $\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} [y_i \log p(\mathbf{x}_i) + (1 - y_i) \log(1 - p(\mathbf{x}_i))]$, with $p(\mathbf{x}_i) = \sigma(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0)$. The decision boundary is linear in the feature space, given by $\boldsymbol{\beta}^\top \mathbf{x} + \beta_0 = 0$.

2.5 Support Vector Machines (SVM)

SVM learn a maximum-margin hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$ in feature space $\mathcal{X} \subseteq R^d$ by solving the convex optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i \in \{1, ..., n\} \end{aligned}$$

where $y_i \in \{-1, 1\}$ are class labels, ξ_i are slack variables for soft-margin classification, and $C > 0$ controls the trade-off between margin maximization and classification error. The dual formulation via Lagrange multipliers $\boldsymbol{\alpha}$ reveals the kernel trick: $\max_{\boldsymbol{\alpha}} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, where $K(\cdot, \cdot)$ is a Mercer kernel enabling nonlinear decision boundaries. Support vectors are the points with $\alpha_i > 0$, lying on or inside the margin.

2.6 Binary Classification Neural Network

We uses a binary classification neural network model with the probability $p(\mathbf{x})$ of a positive class ($y = 1$) using a sigmoid output activation $\sigma(z) = (1 + e^{-z})^{-1}$. The network architecture consists of:

$$f(\mathbf{x}) = \sigma\left(\mathbf{W}_L \phi(\mathbf{W}_{L-1} \cdots \phi(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \cdots + \mathbf{b}_{L-1}) + b_L\right) \tag{1}$$

where $\mathbf{W}_l$ are weight matrices, $\mathbf{b}_l$ are bias terms, and ϕ is the ReLU activation function $\max(0, x)$. The model is trained by minimizing the binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log p(\mathbf{x}_i) + (1-y_i)\log(1-p(\mathbf{x}_i))\right] \tag{2}$$

Key implementation components include:

- Data standardization: $\mathbf{X} \leftarrow (\mathbf{X} - \boldsymbol{\mu})/\boldsymbol{\sigma}$
- Batch normalization layers for stable training
- Dropout regularization (e.g., $p = 0.2$) to prevent overfitting
- Adam optimizer with learning rate scheduling
- Evaluation metrics: accuracy, AUC-ROC, and confusion matrix

The decision threshold is typically set at $p(\mathbf{x}) > 0.5$, with the model architecture (e.g., $128 \to 64 \to 32 \to 1$ units) and hyperparameters tuned via validation performance.

3 Empirical Analysis

3.1 Data Description

This study employs the meteorological dataset from the Kaggle competition *"Binary Prediction with a Rainfall Dataset"* [13], designed to investigate the predictive capability of multidimensional atmospheric measurements. The dataset comprises **2,191** temporally and spatially distinct meteorological observations, each capturing comprehensive atmospheric state variables including **thermodynamic parameters** (temperature, humidity, and atmospheric pressure) and **wind field characteristics** (speed and directional components). Each observation is paired with a **binary precipitation indicator** denoting rainfall occurrence, serving as the target variable for supervised learning.

The dataset provides complete snapshots of atmospheric conditions at specific spatiotemporal coordinates, with its multidimensional nature enabling simultaneous analysis of **meteorological patterns** and their **precipitation relationships**. This structure makes the dataset particularly suitable for evaluating the proposed dimensionality reduction and classification framework, as it captures both the predictor variables and ground truth required for model training and validation.

3.2 Analysis Steps and Results

We performed dimensionality reduction using PCA and Factor Analysis, then developed predictive models based on the original data, PCA-transformed data, and FA-transformed data. For logistic regression, hyperparameters were automatically selected via RandomizedSearchCV with AUC as the optimization metric. SVM were implemented using Gaussian kernel functions, while neural networks were trained with binary cross-entropy loss functions and the Adam optimizer. For K-means clustering, we evaluated variable combinations using accuracy as the selection criterion.

3.2.1 Dimensionality Reduction

In this study, we conducted **principal component analysis** and **factor analysis** to explore the underlying structural features of the dataset.

Based on the results of **PCA**, we selected the first three **principal components**, which together explained **85%** of the total variance. This indicates that these principal components can effectively represent the main information of the original data. In the **factor analysis**, we identified two significant **factors**. The first factor exhibited significant positive loadings on **temperature** and **dew point** variables, and a significant negative loading on **pressure**. Based on these loading patterns, we defined the first factor as the **"Temperature Factor"**. The second factor showed significant positive loadings on **humidity** and **cloud** variables, and a significant negative loading on **sunshine**. Therefore, we named the second factor the **"Humidity Factor"**.

This method of naming and interpreting factors is based on the **direction** and **magnitude** of the variable loadings, which helps us better understand the **latent structure** of the data and provides a solid foundation for subsequent analysis and interpretation. Through this systematic approach, we are able to simplify complex multivariate data into a few key factors, thereby enabling more effective data analysis and interpretation.

Table 1: factor loading estimation after rotation

Variables	Principal Components			Principal Factors	
	f_1	f_2	f_3	f_1^*	f_2^*
Pressure	-0.365	-0.164	-0.072	-0.830	-0.010
Maxtemp	0.414	0.021	0.033	0.960	-0.210
Temperature	0.416	0.058	0.041	0.980	-0.160
Mintemp	0.412	0.084	0.046	0.980	-0.110
Dewpoint	0.393	0.187	0.020	0.960	0.050
Humidity	-0.033	0.564	-0.120	0.090	0.690
Cloud	-0.144	0.574	-0.042	-0.120	0.910
Sunshine	0.215	-0.517	0.047	0.290	-0.840
Winddirection	0.316	0.083	0.205	0.670	-0.070
Windspeed	-0.174	0.079	0.964	-0.320	0.160
Cumulative Proportion					
PCA (3 components)	0.5507, 0.7688, 0.8551			—	
FA (2 factors)	—			0.6767, 0.9486	

3.2.2 K-means Clustering

We performed **K-means clustering** on the original data, and the results indicated that the **cluster associated with rainfall** tends to have higher **air pressure**, **humidity**, and greater **cloud cover** and **wind speed**, while the **cluster associated with rainy conditions** tends to have higher **temperatures**, **dew points**, and stronger **solar radiation**. Moreover, the **wind directions** for **rainy** and **non-rainy conditions** tend to be **opposite**, which aligns with common understanding.

Table 2: Clustering mean of 10 original variables

Cluster	pressure	maxtemp	temperature	mintemp	dewpoint
1	0.8904	-1.0616	-1.0738	-1.0703	-1.0146
2	-0.5902	0.7037	0.7118	0.7095	0.6725

Cluster	humidity	cloud	sunshine	winddirection	windspeed
1	0.0347	0.2631	-0.4589	-0.7327	0.4163
2	-0.0230	-0.1744	0.3042	0.4851	-0.2760

We performed **K-means clustering** on the data after **factor analysis**, and the results showed that the **cluster associated with rainfall** tends to have a higher **Humidity Factor**. The **Temperature Factor**, however, does not significantly differ in mean values between the **rainy** and **non-rainy clusters**, with the **rainy cluster** tending to have a **negative** Temperature Factor.

Cluster	Factor1	Factor2
1	-0.0808	0.5516
2	0.1799	-1.2275

Figure 1: Clustering mean of 2 factors.

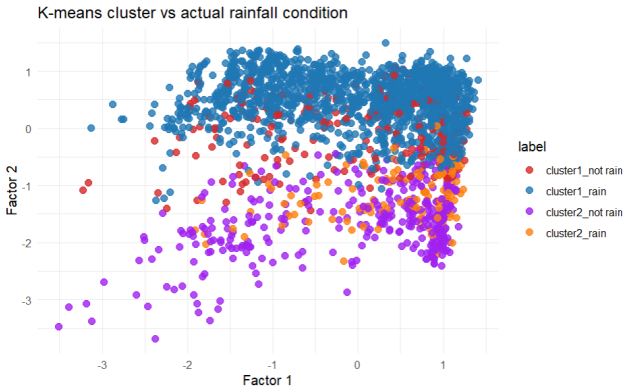

Figure 2: K-means cluster vs actual rainfall condition

3.2.3 Prediction Analysis

This study investigates the impact of different **dimensionality reduction methods** on **model performance** by comparing models built on **original data**, **principal component analysis (PCA)**, and **factor analysis**

(FA), and selects the **best-performing models** for further analysis. In **logistic regression**, **FA** significantly outperforms both the original data and **PCA** in terms of **test set accuracy** (**0.8676**) and **AUC** (**0.9144**), indicating that **FA** effectively preserves the data's potential **discriminative features** and improves overall model performance. The results for the **support vector machine** show more complexity: **PCA** exhibits a notable improvement in **F1-score** (**0.898**) compared to the original data (**0.6943**), but its **AUC** (**0.824**) is lower than that of the original data (**0.8898**). While **PCA** can filter **noise** and accelerate **computation**, it may also cause **information loss**, leading to reduced **generalization capability**. The **binary neural network** maintains **stable performance** across the three scenarios, with **test set accuracy** (**0.85–0.86**) and **F1-score** (**0.85**) remaining constant, indicating **insensitivity** to dimensionality reduction methods. However, **FA** significantly optimizes **computational efficiency**, reducing **training time** from **0.146** to **0.0561** (a **61.5%** decrease).

Method	**Judge Criteria**	**Origin Data**	**PCA**	**FA**
LR	Test Set Accuracy	0.863	0.8562	0.8676
	AUC	0.9042	0.8988	0.9144
SVM	Test Set Accuracy	0.8493	0.8402	0.8425
	AUC	0.8898	0.824	0.832
	F1-Score	0.6943	0.898	0.8996
Binary Neural Network	Test Set Accuracy	0.86	0.85	0.85
	F1-Score	0.85	0.85	0.85
	Average Time per Epoch	0.146	0.089	0.0561

Table 3: Performance comparison of different methods with different dimensionality reduction methods.

Overall, **FA** demonstrates consistent advantages in most scenarios, especially for tasks requiring high **precision** and **efficiency**. The **stability** of the **neural network** suggests that its inherent **feature extraction** capabilities may reduce the necessity of **dimensionality reduction**, but the significant improvement in **computational efficiency** still supports the priority of **FA**. Cross-model comparisons show that **logistic regression** combined with **FA** dominates in terms of overall performance, with its **AUC** and **accuracy** significantly surpassing those of **SVM** and the **neural network**, making it

the preferred solution for tasks requiring high **discriminative power**.

The **variable importance plots** for both the **original data logistic regression model** and the **logistic regression model with factor analysis** reveal insightful patterns. For the original logistic regression model, **cloud cover** and **dew point** exhibit significant positive contributions to **rainfall prediction**, while **sunshine duration** and **temperature** show moderate negative contributions. These findings align with both real-world **meteorological understanding** and the clustering results derived from **K-means analysis**. In contrast, for the logistic regression model incorporating **factor analysis**, it is evident that the **"Humidity Factor"** plays a substantial positive role in predicting rainfall, indicating its strong **discriminative power** in capturing the *latent structure* of the predictors.

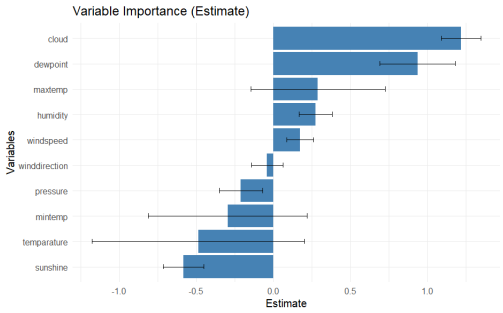

Figure 3: variable importance of Logistic Regression

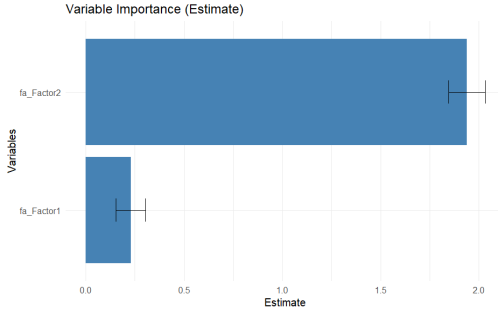

Figure 4: factor importance of Logistic Regression

3.3 Verification of Results

After performing **cross-validation**, the **prediction accuracy** and **AUC** of the **logistic regression model** were found to be highly consistent with the results obtained from the previous single data split. Specifically, the cross-validation yielded an average prediction accuracy of **0.864** and an average **AUC** of **0.89**, while the test set achieved a prediction accuracy of **0.868** and an **AUC** of **0.91**. These results indicate that the performance metrics obtained from cross-validation closely align with those from the test set, thus further validating the **robustness** and **reliability** of the chosen **evaluation criteria**. Based on these stable performance indicators, it can be concluded that the **logistic regression model** demonstrates good **generalization capability** and can be used as an effective tool for subsequent

analysis and prediction.

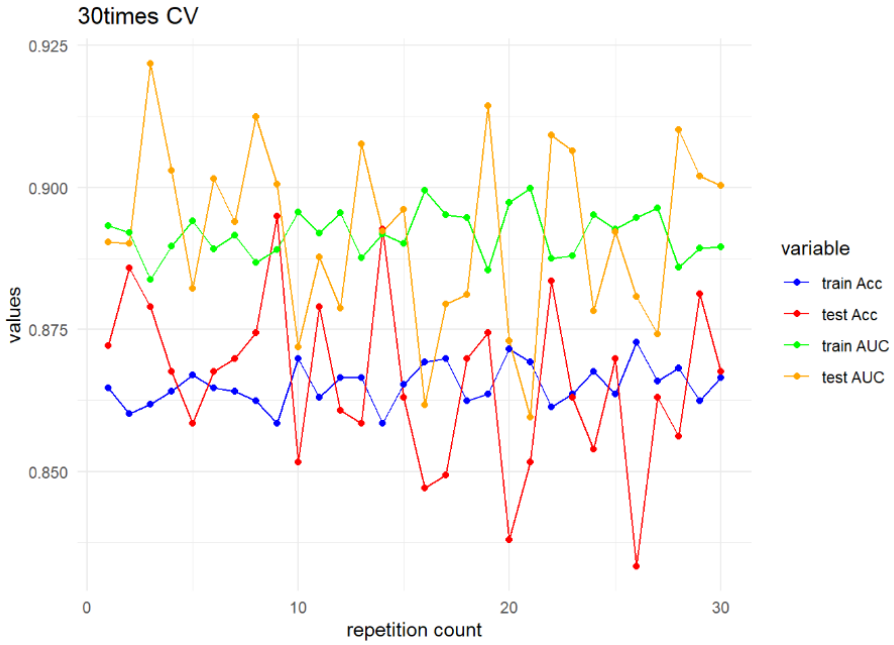

Figure 5: 30 times cross-validation

4 Conclusion

This study systematically evaluated the application of **dimensionality reduction techniques** (**PCA** and **FA**) combined with **machine learning models** for predicting **rainfall probability** in meteorological data. The main findings and insights are summarized as follows:

1. **Effectiveness of Dimensionality Reduction: PCA** successfully retained **85%** of the total variance through three **principal components**, demonstrating its capability to reduce dimensionality while preserving key information. **Factor analysis** identified two interpretable latent factors: the *"Temperature Factor"* and the *"Humidity Factor"*. These factors align with **meteorological principles** and significantly enhance model interpretability.

2. **Model Performance Comparison: Logistic Regression** emerged as the optimal approach, with **FA** effectively preserving **discriminative features** critical for rainfall prediction. **Support Vector Machine** exhibited trade-offs in performance, while **Neural Networks** demonstrated robustness across all data representations, with **FA** significantly improving **computational efficiency**.

3. **Insights from Cluster Analysis: K-means clustering** revealed significant differences in **meteorological patterns**: the **rainfall cluster** was associated with high **humidity**, **cloud cover**, and specific **wind directions**, whereas the **non-rainfall cluster** exhibited characteristics of high **sunshine** and **temperature**. These patterns are consistent with the factors extracted by **FA** and actual meteorological dynamics.

4. **Limitations and Future Directions:** Further exploration of **hybrid methods** is recommended to balance **data-driven insights** with **physical laws**.

In summary, this study established **factor analysis** as an effective method for dimensionality reduction in meteorological data, especially when combined with **logistic regression**. The results highlight the importance of selecting dimensionality reduction strategies based on model characteristics and operational requirements to optimize prediction performance and computational efficiency. Future research should focus on enhancing interpretability and developing **hybrid modeling frameworks** to advance robust rainfall prediction systems.

References

[1] Sarmad Dashti Latif, Nur Alyaa Binti Hazrin, Chai Hoon Koo, Jing Lin Ng, Barkha Chaplot, Yuk Feng Huang, Ahmed El-Shafie, and Ali Najah Ahmed. Assessing rainfall prediction models: Exploring the advantages of machine learning and remote sensing approaches. *Alexandria Engineering Journal*, 82:16–25, 2023.

[2] Michael Dixon and Gerry Wiener. Titan: Thunderstorm identification, tracking, analysis, and nowcasting—a radar-based methodology. *Journal of Atmospheric and Oceanic Technology*, 10(6):785 – 797, 1993.

[3] M. Grecu and W.F. Krajewski. A large-sample investigation of statistical procedures for radar-based short-term quantitative precipitation forecasting. *Journal of Hydrology*, 239(1):69–84, 2000.

[4] Stefano Sebastianelli, Fabio Russo, Francesco Napolitano, and Luca Baldini. On precipitation measurements collected by a weather radar and a rain gauge network. *Natural Hazards and Earth System Sciences*, 13:605–623, 2013.

[5] Shejule Priya Ashok and Sreeja Pekkat. A systematic quantitative review on the performance of some of the recent short-term rainfall forecasting techniques. *Journal of Water and Climate Change*, 13(8):3004–3029, 07 2022.

[6] World Meteorological Organization (WMO). *Standardized Precipitation Index User Guide*. Geneva, 24 2012. Accessed: [Insert Date].

[7] Davide De Luca and A. Petroselli. Storage (stochastic rainfall generator): A user-friendly software for generating long and high-resolution rainfall time series. *Hydrology*, 8, 05 2021.

[8] Davide De Luca. *Rainfall Nowcasting Models for Early Warning Systems*. 01 2013.

[9] A. Bárdossy and G. G. S. Pegram. Copula based multisite model for daily precipitation simulation. *Hydrology and Earth System Sciences*, 13(12):2299–2314, 2009.

[10] Anthony Lawrance and P. Lewis. A new autoregressive time series model in exponential variables (near(1)). *Advances in Applied Probability*, 13:826–845, 12 1981.

[11] F. Cioffi, L. Tieghi, M. Giannini, and S. Pirozzoli and. Flash flood prediction in st. lucia island through a surrogate hydraulic model. *Journal of Applied Water Engineering and Research*, 12(3):297–310, 2024.

[12] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Comput. Surv.*, 55(4), November 2022.

[13] Kaggle. Playground series - season 5, episode 3. `https://www.kaggle.com/competitions/playground-series-s5e3`, 2025. Accessed: 2025-05-23.