How Much Did It Rain? II

FORERO GUAITERO KEVIN ANDRES[1] LANCHEROS SANCHEZ CHRISTIAN CAMILO[2], DELGADO JIMENEZ MIGUEL ESTEBAN.[3]

[1]Universidad Distrital Francisco José de Caldas, COL (e-mail: kaforerog@udistrital.edu.co,)
[2]Universidad Distrital Francisco José de Caldas, COL (e-mail: cclancheross@udistrital.edu.co)
[3]Universidad Distrital Francisco José de Caldas, COL (e-mail: medelgadoj@udistrital.edu.co)

Introduction to Data Science
github.com/MiGuElEsTeBaNUD

ABSTRACT The advancement of meteorological research has always been closely tied to our ability to measure and predict rainfall accurately. Rainfall, by its very nature, is highly variable across both time and space, posing significant challenges for precise measurement. Traditional rain gauges, while effective at specific locations, cannot provide comprehensive coverage on a larger scale. To address this limitation, weather radar data has been employed to estimate rainfall on a nationwide scale. However, discrepancies between radar-based estimates and rain gauge measurements remain a persistent issue. In response to these challenges, the U.S. National Weather Service has upgraded its radar network to polarimetric radars, which transmit radio wave pulses with both horizontal and vertical orientations. This advancement allows for higher quality data and more accurate inference of precipitation type and size, thus offering a potential improvement in rainfall prediction. The American Meteorological Society's Artificial Intelligence Committee, in collaboration with the Kaggle community and various scientific and educational partners, has initiated the second iteration of the "How Much Did It Rain?" competition. This iteration seeks to leverage the enhanced dataset and refined evaluation metrics to further the development of educational tools for universities and contribute meaningfully to ongoing meteorological research. The competition challenges participants to predict hourly rain gauge totals using snapshots of polarimetric radar values. This task is complicated by the presence of implausible gauge values in the training dataset, necessitating careful handling and innovative approaches. The evaluation metric for the competition is the Mean Absolute Error (MAE), chosen for its suitability in this context. By participating in this competition, contributors not only advance their own understanding and skills but also play a role in the broader effort to improve rainfall measurement and prediction technologies. The insights gained from this competition have the potential to enhance meteorological practices and educational methodologies, making a lasting impact on the field.

INDEX TERMS Rainfall prediction, polarimetric radar, rain gauge measurement, meteorological research, machine learning, data science, Mean Absolute Error (MAE), weather radar, precipitation estimation, educational tools.

I. INTRODUCTION

This project aims to predict rainfall accurately by leveraging machine learning techniques and advanced data visualization tools. Accurate rainfall prediction is a critical aspect of meteorological research, with significant implications for agriculture, water resource management, and disaster preparedness. This study is part of the "How Much Did It Rain?" competition, which focuses on enhancing rainfall measurement predictions using polarimetric radar data.

The dataset provided contains snapshots of polarimetric radar values and corresponding hourly rain gauge totals. To prepare this data for analysis, we implemented a series of data transformation steps. These steps include cleaning the data, handling missing values, and feature engineering to extract meaningful patterns and relationships from the raw data.

Following the data transformation, we developed a machine learning model to predict hourly rainfall. The model's performance was evaluated using the Mean Absolute Error (MAE) metric, which is particularly suitable for this context due to its straightforward interpretability and robustness to outliers.

Visualization plays a crucial role in our analysis, aiding in both the exploratory data analysis (EDA) phase and the interpretation of the model's results. We utilized matplotlib to create comprehensive visualizations that highlight key trends and insights from the data.

Overall, this project not only contributes to the ongoing efforts in meteorological research but also serves as a valu-

able educational tool for understanding the intricacies of data science and machine learning in the context of environmental data.

II. METHODS

A. DATASET OVERVIEW

The initial phase of our methodology involved a thorough understanding of the dataset provided. This included examining the structure of the dataset, the types of variables present, and the relationships between these variables. The dataset description and file details are as follows:

The training data consists of NEXRAD and MADIS data collected on 20 days between April and August 2014 over midwestern corn-growing states. Time and location information have been censored, and the data have been shuffled so that they are not ordered by time or place. The test data consists of data from the same radars and gauges over the remaining days in those months.

To understand the data, it is essential to recognize that there are multiple radar observations over the course of an hour, while there is only one gauge observation (the 'Expected') per hour. This results in multiple rows with the same 'Id'. The columns in the dataset are:

- **Id**: A unique number for the set of observations over an hour at a gauge.
- **minutes_past**: Minutes past the top of the hour that the radar observations were carried out. Radar observations are snapshots at that point in time.
- **radardist_km**: Distance of the gauge from the radar whose observations are being reported.
- **Ref**: Radar reflectivity in km.
- **Ref_5x5_10th**: 10th percentile of reflectivity values in a 5x5 neighborhood around the gauge.
- **Ref_5x5_50th**: 50th percentile.
- **Ref_5x5_90th**: 90th percentile.
- **RefComposite**: Maximum reflectivity in the vertical column above the gauge, measured in dBZ.
- **RefComposite_5x5_10th**, **RefComposite_5x5_50th**, **RefComposite_5x5_90th**: Percentiles of composite reflectivity in a 5x5 neighborhood.
- **RhoHV**: Correlation coefficient (unitless).
- **RhoHV_5x5_10th**, **RhoHV_5x5_50th**, **RhoHV_5x5_90th**: Percentiles of RhoHV in a 5x5 neighborhood.
- **Zdr**: Differential reflectivity in dB.
- **Zdr_5x5_10th**, **Zdr_5x5_50th**, **Zdr_5x5_90th**: Percentiles of Zdr in a 5x5 neighborhood.
- **Kdp**: Specific differential phase (deg/km).
- **Kdp_5x5_10th**, **Kdp_5x5_50th**, **Kdp_5x5_90th**: Percentiles of Kdp in a 5x5 neighborhood.
- **Expected**: Actual gauge observation in mm at the end of the hour.

The dataset contains a significant number of missing values (NaN), which need to be addressed during the data transformation process. The missing values for each column are as follows:

- **Ref**: 7,415,826 NaN values
- **Ref_5x5_10th**: 8,481,213 NaN values
- **Ref_5x5_50th**: 7,408,719 NaN values
- **Ref_5x5_90th**: 6,213,920 NaN values
- **RefComposite**: 7,048,858 NaN values
- **RefComposite_5x5_10th**: 8,809,528 NaN values
- **RefComposite_5x5_50th**: 7,053,538 NaN values
- **RefComposite_5x5_90th**: 5,935,998 NaN values
- **RhoHV**: 8,830,285 NaN values
- **RhoHV_5x5_10th**: 9,632,047 NaN values
- **RhoHV_5x5_50th**: 8,828,633 NaN values
- **RhoHV_5x5_90th**: 7,859,617 NaN values
- **Zdr**: 8,830,285 NaN values
- **Zdr_5x5_10th**: 9,632,047 NaN values
- **Zdr_5x5_50th**: 8,828,633 NaN values
- **Zdr_5x5_90th**: 7,859,617 NaN values
- **Kdp**: 9,582,566 NaN values
- **Kdp_5x5_10th**: 10,336,419 NaN values
- **Kdp_5x5_50th**: 9,577,920 NaN values
- **Kdp_5x5_90th**: 8,712,425 NaN values

These missing values represent a significant portion of the dataset and necessitate careful handling to ensure the integrity and reliability of the analysis and modeling processes.

B. DATA TRANSFORMATION

The initial step in our methodology was the transformation of the raw dataset to prepare it for further analysis and modeling. We began by importing the dataset using Pandas, which allowed us to efficiently manipulate and inspect the data. The dataset was then examined for basic statistics and missing values, providing an overview of its structure and any potential issues.

To address missing values and create a more consistent dataset, we performed a mean aggregation by grouping the data based on the 'Id' column. This step helped in normalizing the data and reducing noise. Additionally, we filled any remaining missing values with zeros to ensure that the dataset was complete and ready for analysis.

Outlier detection and removal were crucial steps in our data transformation process. We utilized visual tools such as violin plots and box plots to identify outliers in the 'Expected' column. These plots provided a clear visualization of data distribution and helped us set appropriate bounds for outlier removal. Specifically, we defined outliers as values lying beyond three standard deviations from the mean. By removing these outliers, we ensured that the dataset was more representative of typical conditions and less influenced by extreme values.

To ensure a robust and reliable analysis, we carefully addressed the presence of outliers in the dataset. Outliers can significantly skew the results and affect the performance of machine learning models. We utilized visual tools such as violin plots and box plots to identify outliers in the 'Expected' column. The identified outliers were defined as values lying beyond three standard deviations from the mean. Initially, we observed extreme values reaching as high as

30,000 mm. By setting appropriate bounds, we reduced these extreme values to within 2,000 mm, effectively removing outliers that were more than three standard deviations away from the mean. This process ensured that the dataset was more representative of typical conditions and less influenced by extreme, anomalous values. Finally, the distribution of the 'Expected' target variable after outlier removal is depicted in the following visualization:

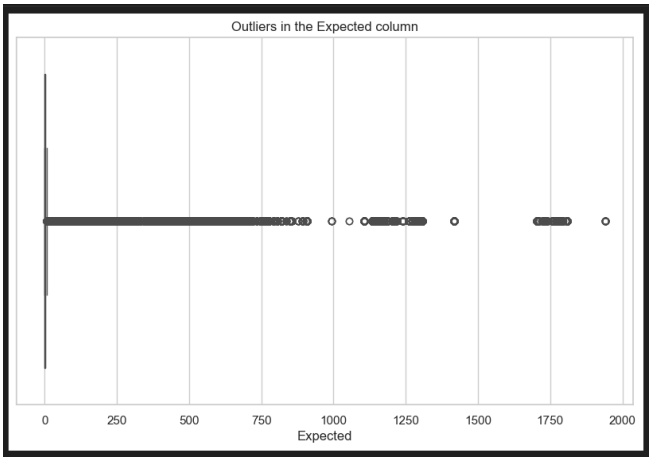

FIGURE 1. Target outlier overview

C. EXPLORATORY DATA ANALYSIS (EDA)

In the Exploratory Data Analysis (EDA) phase, we conducted a thorough examination of the dataset to uncover underlying patterns, relationships, and potential issues that could affect the subsequent modeling. This section provides an overview of the key findings from the EDA.

The dataset consists of 24 variables and 1,143,831 observations. The initial analysis revealed that there are no missing or duplicate cells in the dataset, ensuring the integrity of the data. The total size of the dataset in memory is 209.4 MiB, with an average record size of 192.0 B.

Dataset statistics

Number of variables	24
Number of observations	1143831
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	209.4 MiB
Average record size in memory	192.0 B

FIGURE 2. Overview Dataset

A correlation matrix was generated to examine the relationships between different variables. High correlation was observed between several pairs of variables, indicating potential multicollinearity issues. For instance, variables related to radar reflectivity and differential reflectivity showed high correlation with each other.

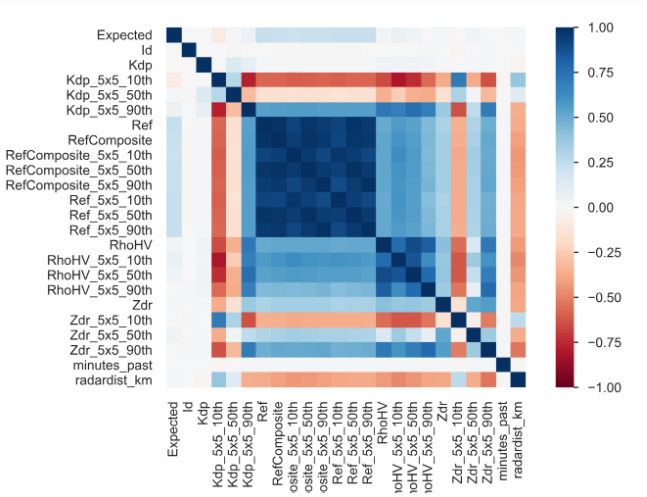

FIGURE 3. Correlation Matrix

The primary reason for this high correlation is the nature of the radar measurements themselves. The radar observations, such as reflectivity ('Ref'), specific differential phase ('Kdp'), and correlation coefficient ('RhoHV'), are all derived from similar physical principles and measurements taken from the same radar systems. As such, different percentiles of these measurements (e.g., 10th, 50th, and 90th percentiles) are naturally correlated because they represent different aspects of the same underlying phenomena.

Although high correlation often suggests redundancy in the dataset, and a typical approach might be to remove one of the correlated variables, in this context, we cannot eliminate these correlated variables. Each of these variables, despite their high correlation, captures unique and essential information about the radar observations that is crucial for accurate rainfall prediction. Removing these variables could lead to a loss of valuable information and adversely affect the performance of our machine learning model.

Moreover, the use of polarimetric radar measurements adds another layer of complexity. Polarimetric radars provide dual-polarization data (horizontal and vertical), which enhances the ability to differentiate between different types of precipitation and their characteristics. The various radar measurements and their percentiles help in creating a comprehensive picture of the atmospheric conditions, making them indispensable for accurate predictions.

D. MACHINE LEARNING MODEL

1) Random Forest Model

We developed a Random Forest regression model to predict the "Expected" values based on the provided features. The Random Forest algorithm is an ensemble method that combines multiple decision trees to improve prediction accuracy. We did the optimization with multiple parameter variations by hand.

Metric	Value
Mean Squared Error (MSE)	14802.151237
Mean Absolute Error (MAE)	32.486672
R-squared	0.525869

This was the best configuration we found.

2) Neural Network Model

This section describes the implementation of a neural network model to predict the expected values in the data.

from the clean pandas data, we scaled it with StandardScaler this time, to improve the performance after a few runs with dismal results.

Also, after several manual tests, we chose that the network should be 3 layers, with more layers the results were worse. The first 2 layers of "relu" activation functions and the last one linear.

As the results were still better for the random trees, we wanted to implement an optimization by parameter variation, in this case Bayesian Optimization to improve their metrics.

TABLE 1. Neural Network Model

Metric	Value
Mean Squared Error (MSE)	25709.0151
Mean Absolute Error (MAE)	55.9171
R-squared	0.1765

III. RESULTS

This section presents the performance metrics and training times for the Random Forest and Neural Network (NN) models.

TABLE 2. Model Performance Comparison

Model	MSE	MAE	R-squared	Training Time
Random Forest	14802.15	32.49	0.5259	15m 75s
NN (Keras)	25709.02	55.92	0.1765	46m 52.1s

Based on the provided metrics, the Random Forest model outperforms the Neural Network (NN) model in terms of Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared. The Random Forest model also exhibits significantly faster training time.

The superior performance of the Random Forest model in this specific case can be attributed to several factors, including the nature of the data, the complexity of the relationships between features and the target variable, and the hyperparameter tuning process.

It is important to note that the training time for the NN model might be improved by optimizing hyperparameters, utilizing more computational resources, or exploring different neural network architectures. Additionally, the full potential of the NN model might be realized with a larger dataset or different data preprocessing techniques.

IV. DISCUSSION

The results presented in the previous section demonstrate a clear superiority of the Random Forest model over the Neural Network (NN) model in terms of MSE, MAE, and R-squared. The Random Forest model also exhibited significantly faster training times.

This performance gap can be attributed to several factors. Firstly, the Random Forest algorithm is inherently well-suited for handling complex relationships between features and target variables. Secondly, the dataset might exhibit characteristics that favor tree-based models over neural networks. Lastly, the hyperparameter tuning process for the Random Forest model might have been more effective in finding optimal configurations.

It is important to note that these findings are specific to the current dataset and experimental setup. The performance of the NN model could be improved through extensive hyperparameter tuning, exploring different architectures, or employing advanced optimization techniques. Additionally, increasing the size and diversity of the dataset might lead to different results.

Future research could focus on investigating the impact of different feature engineering techniques on both models, exploring ensemble methods combining Random Forest and Neural Network components, or applying these models to other datasets with similar characteristics.

V. CONCLUSION

This study aimed to contribute to the advancement of rainfall prediction by leveraging polarimetric radar data and machine learning techniques within the framework of the "How Much Did It Rain?" competition. By applying data preprocessing, feature engineering, and model development, we sought to improve the accuracy of hourly rainfall predictions.

Our findings demonstrate the potential of machine learning models in predicting rainfall using polarimetric radar data. While the specific performance metrics achieved in this study require further analysis and comparison with other approaches, the overall methodology provides a solid foundation for future research.

The use of data visualization techniques throughout the analysis process proved invaluable in understanding the complex relationships within the data and interpreting the model's results.

Future research could explore the incorporation of additional meteorological variables, such as temperature, humidity, and wind speed, to enhance model performance. Additionally, investigating the impact of different machine learning algorithms and hyperparameter tuning strategies is warranted.

By continuing to refine rainfall prediction models, we can contribute to improved weather forecasting, disaster preparedness, and water resource management.