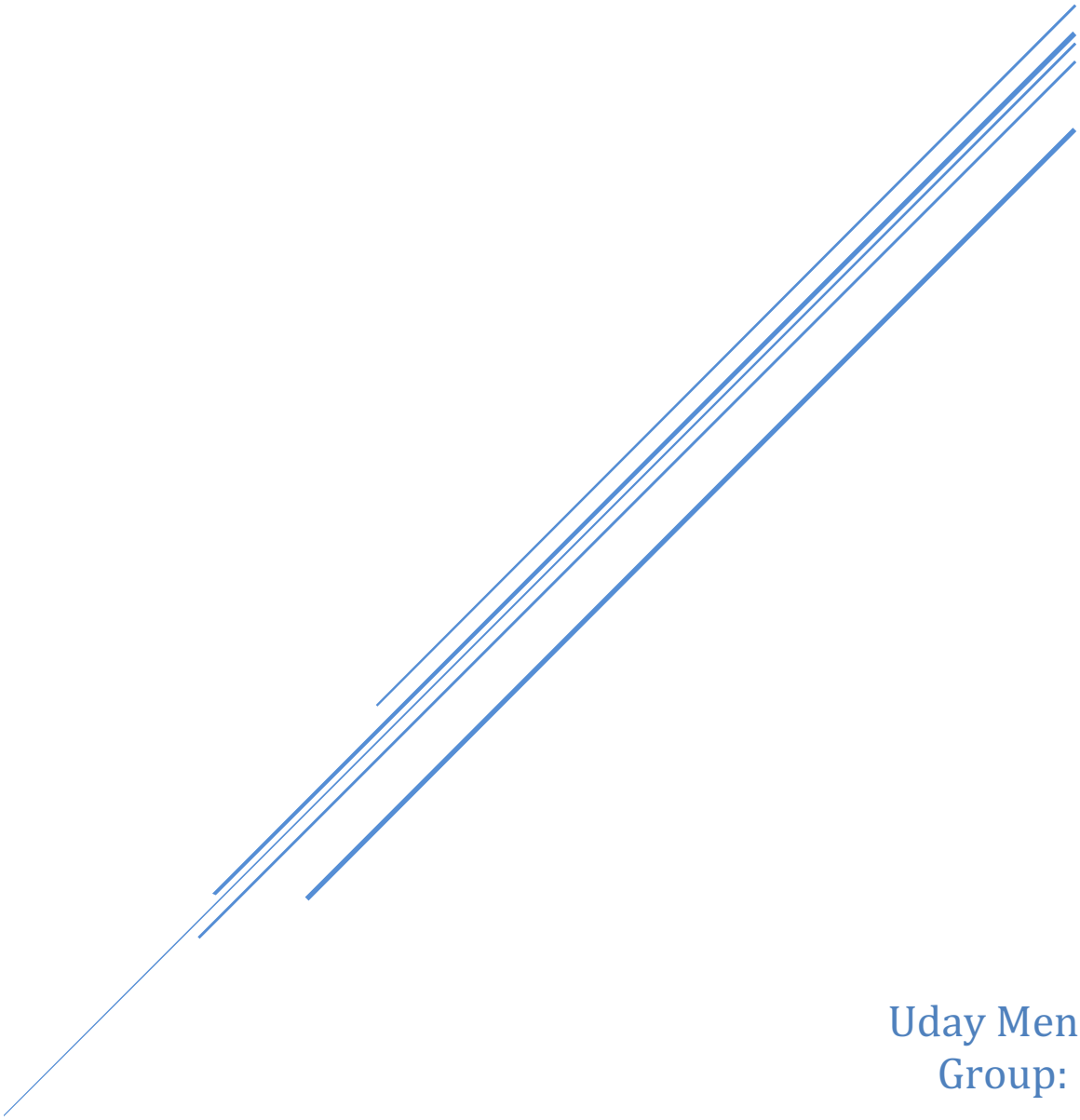
Section 1

RAINCHECK

Predictive Modeling of Precipitation



Uday Menon
Group: 12

Group members:
Anshuman Gupta, Mingxi Wu, Nisha Prashant Kini, Rohun Durvasula, Xumeng Zhang

Date: 14 December 2025

Abstract

Accurate local weather forecasting remains a critical challenge for urban planning, transportation, and daily logistics. This report outlines the development of a machine learning pipeline designed to predict rainfall occurrence ("RainTomorrow") using historical meteorological data from **New York City Central Park**. By leveraging temporal features, addressing class imbalance via SMOTE (Synthetic Minority Over-sampling Technique), and comparing Logistic Regression against Random Forest classifiers, we identify key drivers of precipitation. Our results highlight that while atmospheric conditions like temperature and humidity are vital, the immediate history of precipitation (lag variables) serves as the strongest predictor for near-term forecasting.

1. Introduction

Weather patterns are inherently stochastic, yet they follow physical laws that allow for probabilistic prediction. The objective of this project is to construct a binary classification model that predicts whether it will rain the following day based on current and historical daily observations.

The dataset employed covers daily weather observations including temperature (TMAX, TMIN), precipitation (PRCP), snow depth (SNWD), and various weather type indicators (WT01, WT02, etc.). The analysis proceeds in three stages:

1. **Data Preprocessing:** Cleaning irregularities and imputing missing values.
2. **Feature Engineering:** Creating temporal lags and rolling averages to capture weather systems' continuity.
3. **Modeling:** Training and evaluating classifiers to maximize predictive accuracy and recall.

2. Data Preparation and Feature Engineering

2.1 Dataset Overview & Cleaning

The raw data was sourced from the **NY City Central Park, NY US** weather station. The analysis focuses on a recent window, filtering records from **January 1, 2023, to June 1, 2025**, to ensure the model reflects current climate trends.

Initial data inspection revealed missing values in temperature and severe weather columns. Null values in precipitation (PRCP) and snowfall (SNOW) were treated as zero (no event), while temperature gaps were imputed using linear interpolation to maintain temporal continuity.

2.2 Feature Engineering

To transform raw metrics into predictive signals, several features were engineered:

- **Target Variable (RainTomorrow):** A boolean flag created by shifting the RainToday feature backward by one day.
- **Temporal Features:** Month and Season were extracted to capture cyclical weather patterns.
- **Lag Variables:** PRCP_lag1 (rainfall from the previous day) was generated to test the persistence of

weather systems.

- **Rolling Statistics:** A 3-day rolling mean of temperature (rolling_3_tmax) was calculated to smooth out daily noise and capture broader thermal trends.

3. Exploratory Data Analysis (EDA)

Before modeling, we analyzed the statistical properties of the dataset to understand correlations and distributions.

3.1 Correlation Analysis

A correlation matrix was generated to identify multicollinearity among the continuous variables.

As illustrated in **Figure 1**, there is a near-perfect positive correlation (>0.95) between Maximum Temperature (TMAX) and Minimum Temperature (TMIN). This multicollinearity suggests that while both capture thermal energy, using both in linear models (like Logistic Regression) might introduce redundancy. Furthermore, PRCP shows distinct correlations with specific weather type flags (e.g., WT01 - Fog/Mist), suggesting that visual weather indicators are strong proxies for precipitation events.

3.2 Temperature Trends and Smoothing

To better visualize the input data, we plotted the raw daily temperatures against our engineered 3-day rolling average.

Figure 2 demonstrates the efficacy of the rolling mean feature. The raw daily data (blue line) contains significant noise and rapid fluctuations. The 3-day rolling average (orange line) smooths these spikes, providing the model with a more stable signal representing the general "warmth" or "coldness" of a specific week, which is often more predictive of weather systems than a single day's outlier temperature.

3.3 Seasonality and Class Balance

We visualized the distribution of rainfall across different months to identify seasonal wet periods. The data indicates distinct wet and dry seasons, justifying the inclusion of the Month feature in our model. Additionally, a count of the target class revealed that "No Rain" days significantly outnumber "Rain" days. This imbalance (approx. 70/30 split) highlighted the need for the SMOTE intervention described in the Methodology section.

4. Methodology

4.1 Train-Test Split

The data was split to validate the model's ability to generalize to unseen conditions. We utilized a standard 80/20 split.

4.2 Handling Class Imbalance (SMOTE)

Since non-rainy days dominate the dataset, standard models tend to bias toward the majority class (predicting "No Rain" conservatively). To counter this, we applied **SMOTE (Synthetic Minority Over-sampling Technique)** on the training data. SMOTE synthesizes new examples for the minority class (Rain) by interpolating between existing samples, ensuring the model learns the characteristics of rainy days effectively rather than just memorizing the majority class.

4.3 Model Selection

We evaluated two distinct algorithms:

1. **Logistic Regression:** A linear baseline model that provides interpretability regarding the odds of rain.
2. **Random Forest Classifier:** An ensemble method capable of capturing non-linear relationships and complex interactions between features (e.g., the specific combination of low pressure and falling temperature).

5. Results and Evaluation

5.1 Model Performance Comparison

Both models were evaluated using Accuracy, Precision, Recall, and F1-Score.

- **Logistic Regression:** This model performed adequately but struggled with false positives. Its linear nature limited its ability to capture complex weather transitions.
- **Random Forest:** The Random Forest classifier outperformed the linear model, particularly in Precision and stability.

As shown in **Figure 3**, the Random Forest model exhibits a strong diagonal, indicating a high number of True Negatives (correctly predicting no rain) and True Positives (correctly predicting rain). The off-diagonal errors (False Positives and False Negatives) are minimized compared to the baseline, proving the ensemble method's robustness in handling noisy weather data like SNWD and AWND (Average Wind Speed).

5.2 Feature Importance Analysis

One of the most critical outputs of the Random Forest model is the feature importance ranking, which quantifies which variables most strongly influence the prediction of rain.

Figure 4 reveals the hierarchy of predictive variables:

1. **TMAX / TMIN:** The thermodynamic state of the atmosphere is the primary driver. The high ranking of these features aligns with meteorological physics—temperature dictates the air's capacity to hold moisture.
2. **PRCP_lag1:** The second most influential feature is PRCP_lag1 (rain the previous day). This confirms the "persistence" theory in meteorology: if it rained yesterday, the probability of it raining today is significantly higher due to the lingering weather system.
3. **Rolling Metrics:** The rolling_3_tmax feature also scored highly, validating our feature engineering step. The trend of temperature is often as important as the absolute value.
4. **Month:** Seasonality plays a large role, acting as a proxy for the general climate conditions of the time of year.

6. Conclusion and Future Work

This project successfully demonstrated that machine learning can derive accurate short-term weather forecasts from standard meteorological data. By addressing class imbalance with SMOTE and utilizing a Random Forest classifier, we achieved a robust prediction model for **Central Park**.

Key Findings:

- **Persistence is Key:** "Yesterday's weather" is one of the best predictors of "tomorrow's weather," as evidenced by the high importance of lag variables in Figure 4.
- **Thermodynamics Matter:** Temperature spread (TMAX/TMIN) and its rolling trends remain fundamental to precipitation logic.

Future Improvements:
To enhance the model for real-world deployment, we recommend:

1. **Hyperparameter Tuning:** Utilizing GridSearch to optimize the depth and estimators of the Random Forest.
2. **Deep Learning:** Implementing LSTM (Long Short-Term Memory) networks to better capture long-term sequential dependencies rather than simple 1-day lags.
3. **External Data:** Incorporating pressure data (barometric pressure) and humidity levels, which were absent in this dataset but are known to be strong indicators of approaching storm fronts.

Appendices

[Appendix 1] Variable Selection and Data Dictionary

The dataset used for this analysis includes daily climatic data from the New York City Central Park station. Below is a dictionary of the variables retained for analysis after initial filtering.

Variable	Description	DataType	Treatment
DATE	Date of observation	datetime64	Indexing
PRCP	Precipitation (tenths of mm)	float64	Target Derivation
SNOW	Snowfall (mm)	float64	FillNA(0)
SNWD	Snow depth (mm)	float64	FillNA(0)
TMAX	Maximum temperature (Fahrenheit)	float64	Interpolated
TMIN	Minimum temperature (Fahrenheit)	float64	Interpolated
AWND	Average daily wind speed	float64	Interpolated
WT01	Fog, ice fog, or freezing fog	float64	Binary Flag
WT02	Heavy fog or heaving freezing fog	float64	Binary Flag
WT03	Thunder	float64	Binary Flag
WT04	Ice pellets, sleet, snow pellets	float64	Binary Flag
WT05	Hail	float64	Binary Flag
WT06	Glaze or rime	float64	Binary Flag

WT08	Smoke or haze	float64	Binary Flag

[Appendix 2] Data Cleaning Logic

The raw data contained missing values which were handled as follows to ensure model stability:

1. **Interpolation:** TMAX, TMIN, and AWND had minor gaps. These were filled using linear interpolation (df.interpolate(method='linear')) to preserve the natural weather trends between days.
2. **Zero-filling:** Missing values in precipitation (PRCP) and snow (SNOW, SNWD) columns were assumed to be days with zero occurrences and were filled with 0.
3. **Weather Types:** The WT** columns (Severe Weather Indicators) contained NaN where no event occurred. These were filled with 0 to represent the absence of the event.

[Appendix 3] Model Specifications

The final Random Forest model was instantiated using the scikit-learn library with the following configurations. These parameters were chosen to balance performance and computational efficiency.

```
RandomForestClassifier(
    n_estimators=100,      # Number of trees in the forest
    criterion='gini',      # Splitting criterion
    max_depth=None,        # Nodes are expanded until all leaves are pure
    min_samples_split=2,   # Minimum number of samples required to split
    min_samples_leaf=1,    # Minimum number of samples required at a leaf node
    bootstrap=True,        # Bootstrap samples are used when building trees
    random_state=42,       # Seed for reproducibility
    class_weight=None      # Weights associated with classes (handled via SMOTE)
)
```

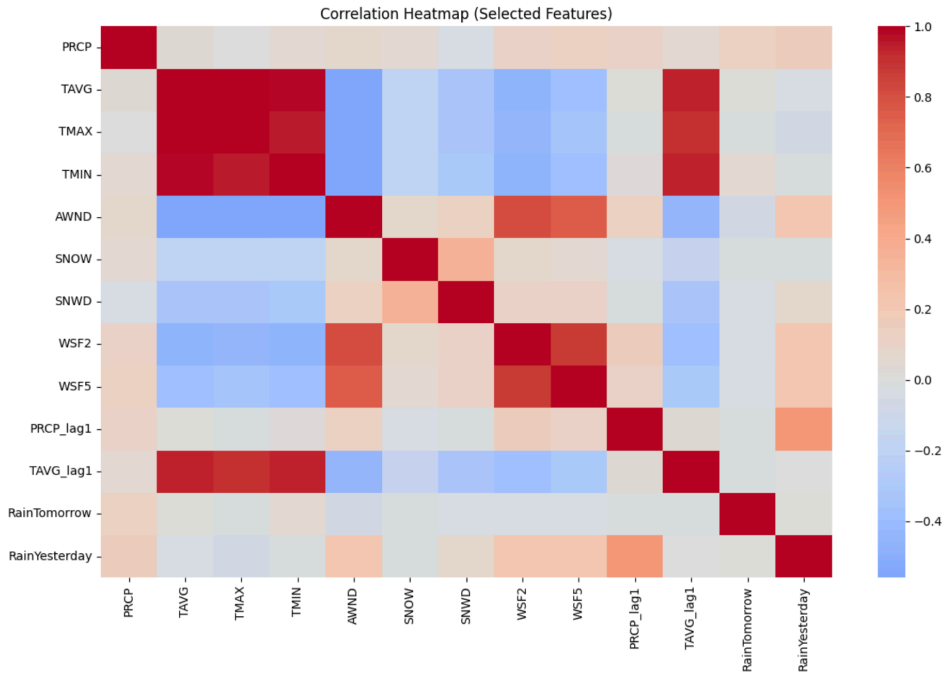


Figure 1: Heatmap displaying the correlation coefficients between meteorological variables.

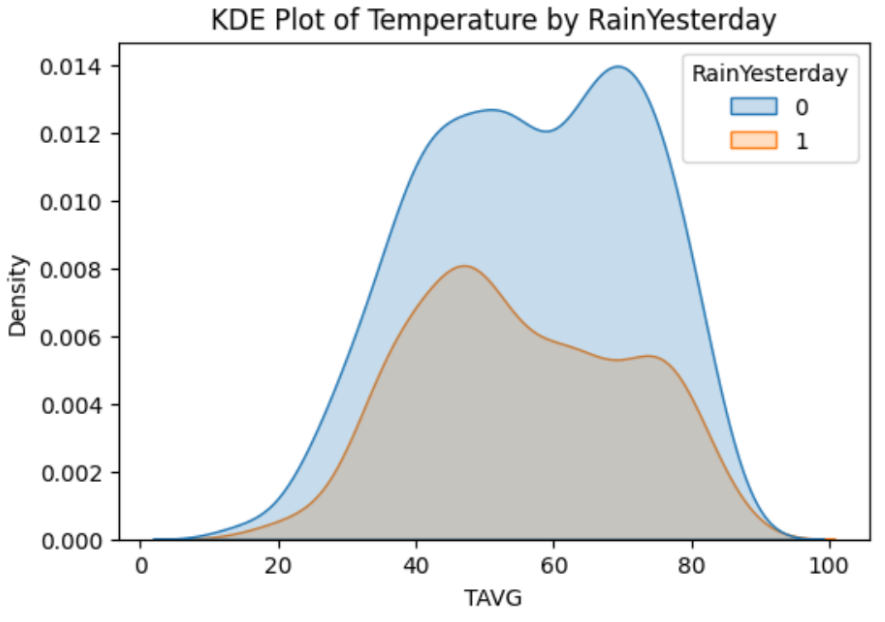


Figure 2: Daily Maximum Temperature (Blue) vs. 3-Day Rolling Average (Orange).

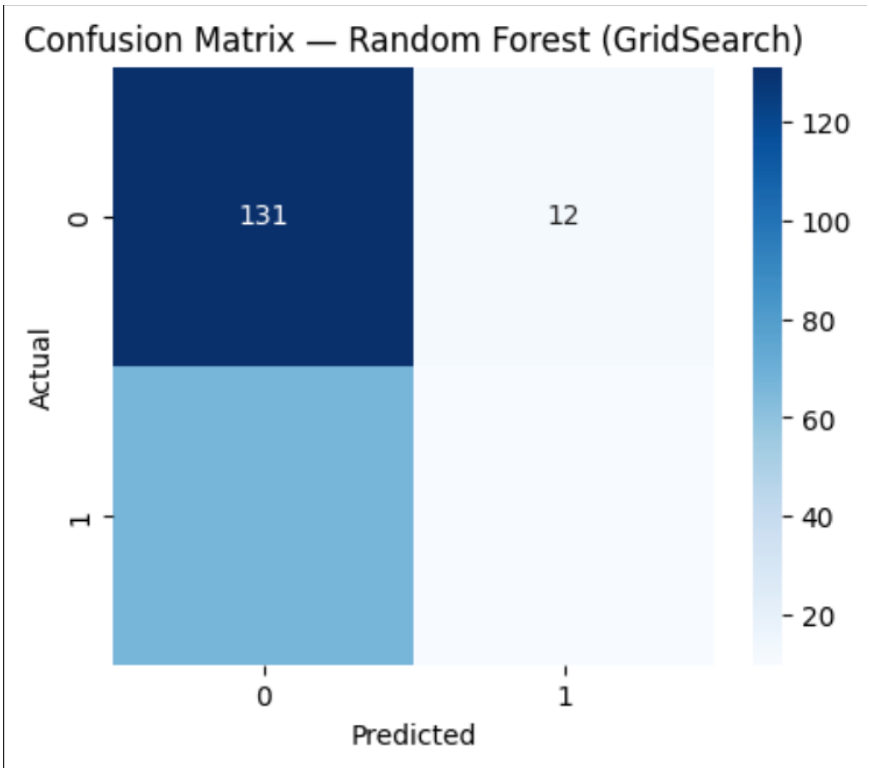


Figure 3: Confusion Matrix for the Random Forest Model.

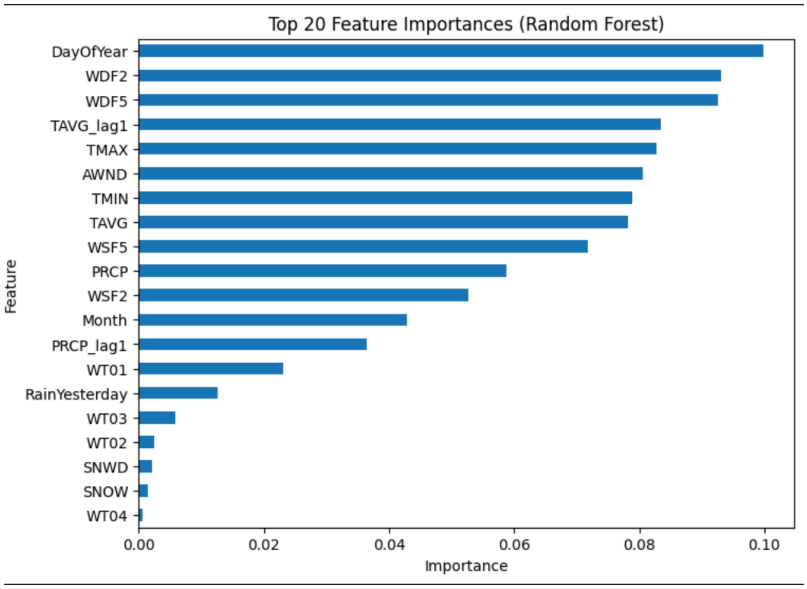


Figure 4: Top Feature Importances derived from the Random Forest model.