

# **CS 334: Machine Learning**

Emory University

Fall 2023

Aadi Waghray and Joshua Sun

---

## **CS334 Final Report**

Aadi Waghray and Joshua Sun

### **Abstract**

Abstract text

### **Introduction**

Precipitation prediction is a high priority for many groups for people from government entities to private corporations due to its far reaching consequences. Understanding when and how much precipitation will occur at a given location can aid conservation efforts, improving infrastructure, and allow for more efficient resource allocation for emergencies. Prior studies developing rainfall prediction models have fallen into two main categories: statistical and numerical. For our purposes, we consider only statistical models, which utilize machine learning to predict precipitation type or amount

based on past weather data. The studies used models and techniques such as SVMs, Extreme Gradient Boosting, Decision Trees, and LSTMs. [3, 1, 5] We planned to use similar models in our study.

Our study is using the NOAA quality controlled datasets [2] as well U.S. Local Climatological Data from NOAA[4]. The quality controlled dataset is split up into monthly, daily, hourly, and sub-hourly datasets. Each dataset collects various atmospheric and earth data from various weather stations across the US. Similarly, the USLCD does the same, with more detail, but much more missing data. Prior studies into predicting precipitation used only one source of data. Our study is unique in attempting to augment the quality controlled datasets with more features. We specifically focus on data collected from Brunswick, GA.

## Methodology

We first needed to preprocess our data. We first took the quality controlled data, and filtered the data to get rid of bad data. Conveniently, packaged with the data, were instructions on certain fields. For example, `ST_FLAG` is a field described such that, when the value is greater than 0, an error has occurred in the raw data gathering. Similarly, certain numeric fields, such as `P_PRECIP`, an extremely low value is written to the field if it is missing.

Continuing, certain fields are non-numeric, and needed to be removed, such as those fields. The fields removed without computation were

- WBANNO
- LST\_DATE
- LST\_TIME
- CRX\_VN
- SUR\_TEMP\_TYPE
- SOLARAD\_FLAG
- SOLARAD\_FLAG
- SOLARAD\_MAX\_FLAG
- SOLARAD\_MIN\_FLAG
- SUR\_TEMP\_FLAG
- SUR\_TEMP\_MAX\_FLAG
- SUR\_TEMP\_MIN\_FLAG

Clearly the flags were removed, as they only gave error information. The station name is the same for all pieces of data. We removed the local time because we already have UTC time to extract information. To extract more features, we converted UTC\_DATE and UTC\_TIME to a python `datetime` column labeled DATE.

For the LCD dataset, a similar process was followed. Labels that were strictly non-numeric were removed. Note that LCD dataset took measurements every 20 minutes, so we had to only take every third bit of data. `DATE` was created by then rounding the date to the nearest hour. We then merged the two datasets by matching the `DATE` columns of the datasets.

After doing so, we then created new features based off of the year, month, day, and hour. If there was missing data, for example, a missing precipitation value, we would randomly choose a non-missing value of the same day, month, and hour, but on a different year. This is a reasonable assumption that the weather will do similar things on the same date on different years.

We then did a Pearson Correlation Matrix to see which features were correlated. From the Pearson Correlation Matrix we removed features that were correlated greater than 0.5.

From our data, we can now use it in our models. Every model we used was from the `SKLearn` library. All the models used were regression models, as we are predicting the level of precipitation. For each model, we did hyperparameter tuning, and judged it based on the MSE. For hyperparameter tuning, we used the `RandomizedCV` from `SKlearn`. This by default uses 5 cross-validation sections. To test the efficacy of the model, we split the dataset into an 80-20 split of training-testing.

## Results

Results text

## Conclusions

Conclusions text

## Code

Code text

## References

- [1] Ari Yair Barrera-Animas et al. “Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting”. In: *Machine Learning with Applications* 7 (Mar. 2022), p. 100204. ISSN: 26668270. DOI: 10.1016/j.mlwa.2021.100204. URL: <https://linkinghub.elsevier.com/retrieve/pii/S266682702100102X> (visited on 12/07/2023).
- [2] Howard J. Diamond et al. “U.S. Climate Reference Network after One Decade of Operations: Status and Assessment”. In: *Bulletin of the American Meteorological Society* 94.4 (Apr. 1, 2013), pp. 485–498. ISSN: 1520-0477. DOI: 10.1175/BAMS-D-12-00170.1. URL: <https://journals.ametsoc.org/doi/10.1175/BAMS-D-12-00170.1> (visited on 12/07/2023).
- [3] Chalachew Muluken Liyew and Haileyesus Amsaya Melese. “Machine learning techniques to predict daily rainfall amount”. In: *Journal of Big Data* 8.1 (Dec. 2021), p. 153. ISSN: 2196-1115. DOI: 10.1186/s40537-021-00545-4. URL: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00545-4> (visited on 12/07/2023).
- [4] *Local Climatological Data (LCD) Publication*. URL: <https://www.ncei.noaa.gov/access/metadata/landing-page/bin/iso?id=gov.noaa.ncdc:C00128> (visited on 12/13/2023).
- [5] Atta-ur Rahman et al. “Rainfall Prediction System Using Machine Learning Fusion for Smart Cities”. In: *Sensors* 22.9 (May 4, 2022), p. 3504.

ISSN: 1424-8220. DOI: 10.3390/s22093504. URL: <https://www.mdpi.com/1424-8220/22/9/3504> (visited on 12/07/2023).