# Predicting Precipitation and Other Weather Conditions With Logistic Regression and Random Forest Classifiers

**Vadim Yezhkov**

08-February-2023

AI + X Scholars Program

**Abstract**

Nowadays, we predict the weather to get a better understanding of the days to come. An issue we have with our current algorithms is the inability to predict weather accurately for any time range further than ten days. Our intentions with this project were to create a model that could get an accuracy within 5% of our current weather prediction algorithms. We wanted the model to be accurate in both short-term and long-term conditions. Our current algorithms, while highly accurate in the short term, struggle in the long-term timespan. The importance of accuracy in this model can be boiled down to two main points; convenience, for planning activities, and safety, as perhaps an early warning system in the event of a possibly harmful or fatal weather event. Our method of achieving these goals was to first use a logistic regression model, and later a random forest classifier. In the end we reached a highest accuracy of 83%, which fell just short of our goal of 85%. Despite the inaccuracy, our model showed promising results, and if we had more time or a larger dataset, the accuracy could have been higher. In the end, the results seemed more to be a proof of concept, rather than a demonstration of accuracy and capability.

## 1. Introduction

The motivation behind this project were mainly three points; convenience, safety, and to further familiarize myself with the creation and training of models. To further elaborate on the more important of these three aspects, convenience in the sense of accuracy- the higher the accuracy of the model, and the better the predictions, the less situations there will be when you are expecting a sunny day but are caught in a rainstorm. By safety, I mean an early warning system- knowing that a dangerous weather event is coming even a week in advance could be the difference between life and death for some people, as they will have ample time to evacuate. The way we went about achieving these goals was by using a model that would first try to predict one of five types of weather for any given day, these being sun, rain, fog, snow, or drizzle. After that, we moved on to a simpler model that would attempt to confirm only one of these conditions, as in sunny or not. The way we could verify the practicality of our results is to compare them to existing weather models, and see their accuracies.

## 2. Background

Knowing the weather is useful for things other than planning a weekend outing. For example, people such as farmers use weather patterns to grow their crops more efficiently, as adverse weather conditions can easily wipe out an entire harvest. This was observed in 2022, when three consecutive failed rainy seasons left most of Kenya, Ethiopia, and Somalia without food. Another important concept is the actual severity of natural disasters. Often when one comes by, as many as thousands of people can die, and millions–if not billions–of dollars of damage are caused. An example of such a storm is Hurricane Ian, which caused 144 fatalities

and between 50-65 billion dollars in damages in September of 2022.While the cost of damages may be more difficult to minimize as quickly relocating permanent infrastructure is beyond our current abilities, we can try to minimize the casualties with an early warning system. While this seems like perhaps a relatively simple task on the surface, in reality, it is not. In chaos theory, the butterfly effect  is the sensitive dependence on initial conditions in which a small change in one state of a deterministic nonlinear system can result in large differences in a later state. An example of this was demonstrated In 1961, when mathematician and meteorologist Edward Norton Lorenz was running a computer model to redo a weather prediction from the middle of the previous run as a shortcut. He entered the initial condition “0.506” from the printout instead of entering the full precision “0.506127” value. The result was a completely different weather scenario. The way this ties in to our model is the unfathomable amount of factors that determine the weather. Something seemingly insignificant can change something as important as the time when a dangerous weather pattern occurs, making it nearly impossible to accurately predict when and where it would hit.

### 3. Dataset

Our dataset covered the weather in Seattle, Washington from January 1st, 2012 to December 31st, 2015. The data was organized in a way such that we have a data point for each day within the range. The data was split into the following separate columns: date, precipitation, maximum temperature, minimum temperature, wind speed, and weather type. The dataset was sourced from Kaggle, an online dataset bank. See table below.

| | date | precipitation | temp_max | temp_min | wind | weather |
|---|---|---|---|---|---|---|
| **0** | 2012-01-01 | 0.0 | 12.8 | 5.0 | 4.7 | drizzle |
| **1** | 2012-01-02 | 10.9 | 10.6 | 2.8 | 4.5 | rain |
| **2** | 2012-01-03 | 0.8 | 11.7 | 7.2 | 2.3 | rain |
| **3** | 2012-01-04 | 20.3 | 12.2 | 5.6 | 4.7 | rain |
| **4** | 2012-01-05 | 1.3 | 8.9 | 2.8 | 6.1 | rain |
| **...** | ... | ... | ... | ... | ... | ... |
| **1456** | 2015-12-27 | 8.6 | 4.4 | 1.7 | 2.9 | rain |
| **1457** | 2015-12-28 | 1.5 | 5.0 | 1.7 | 1.3 | rain |
| **1458** | 2015-12-29 | 0.0 | 7.2 | 0.6 | 2.6 | fog |
| **1459** | 2015-12-30 | 0.0 | 5.6 | -1.0 | 3.4 | sun |
| **1460** | 2015-12-31 | 0.0 | 5.6 | -2.1 | 3.5 | sun |

1461 rows × 6 columns

In order to make the data more usable, we had to preprocess some of the columns within the dataset. To do so, we first made a separate column and encoded the weather data in a one-hot

format. After that, we made a separate column to store the dates as numbers as opposed to strings (ie. 0, 1, 2 as opposed to 2012-01-01, 2012-01-02 etc.).

When it came to using our data, we used a split of 80% training data and 20% testing data, independent of time, though more complex models might have used one period as training data and the following period for testing. Initially, we made it predict the weather conditions, but we ran into a problem when we were unable to calculate precision and recall with multiple classes as the output. To combat this issue, we decided to shift to a binary option of sunny versus not sunny, as opposed to selecting an option out of five.

One potential weakness of the dataset is the distribution of the data. When sorted, out of the 1461 data points, 640 were sunny, 641 were rain, 101 were fog, 26 were snow, and 53 were drizzle. The majority of the dataset was in either sunny or rain, which caused the model to be able to guess either of those conditions and still have a relatively high score.

**4. Methodology / Models**

To start out, we used a Logistic Regression model. Logistic Regression implements regularized logistic regression using the 'liblinear' library, and the 'lbfgs' solver.. The regularization is applied by default. It can handle both dense and sparse inputs. An issue with this type of model is that the underlying C implementation uses a random number generator to select features when fitting the model. It is thus not uncommon to have slightly different results for the same input data. We input data such as precipitation, temperature, and wind, and based on that, the model outputs a one-hot of the weather condition it thought it was. This yielded around a 77% accuracy, as the model would mainly predict either rainy or sunny weather, due to the distribution of the data.

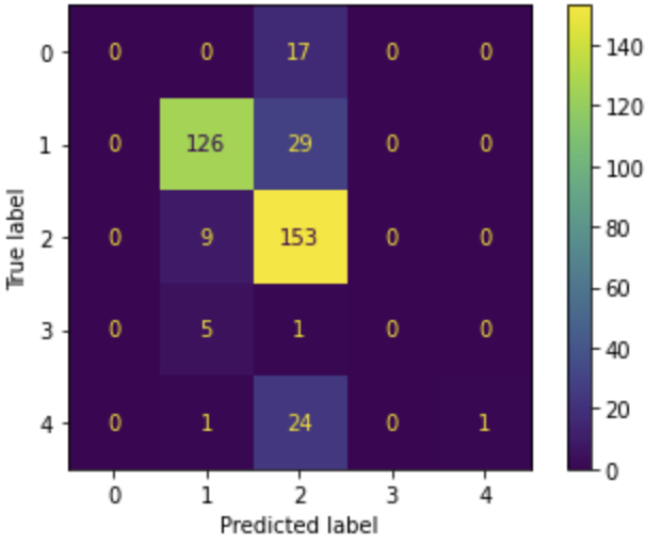
After this, we moved on to using a random forest regressor model. A random forest model is an estimator that fits a number of classifying decision trees on various sub-samples of the dataset, and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter. The features are always randomly permuted at each split, therefore, the best found split may vary, even with the same training data. We also changed the parameter "n_estimators", which controls the number of "trees" in the forest. The value is 100 by default. This yielded a higher accuracy of about 83%, which was an increase, but still fell short of the goal of 85%, likely due to the lack of distribution of data in the dataset.

Regardless of what model we used, I feel that we would have run into the same issue; the dataset being so focused on rainy and sunny weather. This is likely due to the other options being so niche, as they were: fog, snow, and drizzle. Since it can only snow during winter, that already greatly limits the amount of data points with this weather condition. Fog is dependent on a very

particular circumstance, as it only forms when cold air moves over warm water. When the cold air mixes with the warm moist air over the water, the moist air cools, and when humidity reaches 100%, fog forms. Finally, drizzle, which is classified by the droplet size. In order for precipitation to be classified as drizzle instead of rain, the droplet size has to be less than 0.5mm in diameter. Larger drops are more commonly observed, leading to the weather to be classified as rain instead of drizzle.

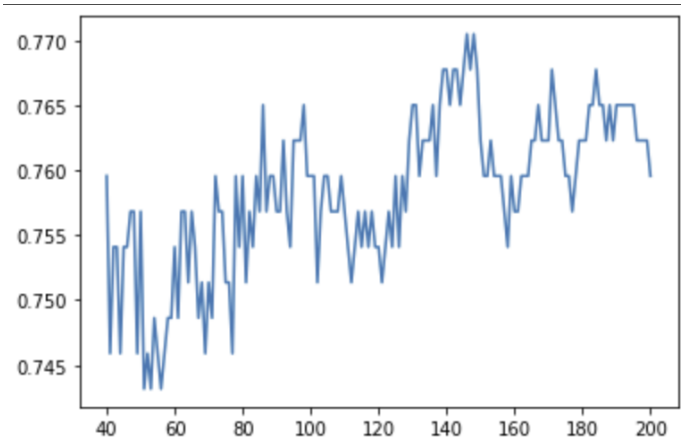
**5. Results and Discussion**

After running the model, the initial results were rather underwhelming, as we achieved an accuracy of 77% instead of our goal of 85%. Below is a confusion matrix displaying our results.




As shown in the matrix, the flaws of the dataset's distribution as previously stated, played a role in the predictions our models made. The predicted label "2" shows up the most, which was the label for sunny weather. The second most predicted label, "1", was the label for rain. All of the other labels were predicted a combined total of one time. The unbalanced distribution of data in the dataset caused the model to lean heavily into predicting only sunny or rainy conditions. Another source of error for our model was not being able to hash a numpy array. This was problematic, as we could not calculate precision and recall, and we could not find a solution to this error.

After we switched from a classifier model to a random forest model, we tried to tune it, in an attempt to increase our accuracy. We did this by creating a for loop to generate a model with an increasing amount of estimators. As shown by the graph below, overfitting occurred past 150 estimators, which gave us a baseline for how many to use. Our overall increase in accuracy was about 5%.




When comparing our results to our theory, with a final highest accuracy of 83%, we only fell 7% short of the current approximate 90% accuracy. Our models showed promising results for the amount of time spent on them, as well as the limited dataset used. If a larger team of dedicated researchers worked on a project similar to this one, as well as had access to a larger dataset, it is likely that they could achieve an accuracy to rival our current algorithms.

## 6. Conclusions

In conclusion, our model barely fell short of our goal. Using the random forest regressor yielded better results than using the logistic regression model, likely due to the highly customizable nature of the forest regressor. The model suffered from the inability to be trained with a variety of data. While the dataset did provide sufficient data, most data points were relatively indifferent. To improve our current model in the future, I would look into either a larger or more diverse dataset, especially one that is more specific about its labels. If a larger dataset is unable to be accessed or obtained, trying to sample only a certain amount of points from each weather condition for the training data could help spread the guesses of the model. Looking into a more complex model, or further tuning the random forest regressor could also increase the accuracy. I feel like if we set the flaws of the model and dataset aside, the highest realistic accuracy is around 95%-98%, just due to the ever changing climate, whose rate of change is being boosted by the burning of fossil fuels, and further pollution of the atmosphere. A perfect accuracy, even with a steady and stable climate would still be impossible to achieve, due to the aforementioned butterfly effect. For the amount of time we had, and the dataset we used, I am happy with the results that the model procured.

## Acknowledgments

I would like to thank Darnell Granberry for extended assistance with this project. His contributions were significant, and without them, this project could have had a very different outcome. I also would like to give out my gratitude to the people that made and labeled our dataset.

**References**

*Severe drought threatens 13 million with hunger in Horn of Africa | UN News United Nations*. United Nations. Available at: https://news.un.org/en/story/2022/02/1111472 (Accessed: 08-Feb-2023).

N.O.A.A. US Department of Commerce, "Hurricane Ian: September 30, 2022," *National Weather Service*, 28-Oct-2022. [Online]. Available: https://www.weather.gov/ilm/HurricaneIan. [Accessed: 08-Feb-2023].

*"Linear Model Logistic Regression," scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. [Accessed: 08-Feb-2023].

*"Random Forest Regressor," scikit*. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html [Accessed: 08-Feb-2023].

US Department of Commerce, N.O.A.A. (2015) *How fog forms*, *National Weather Service*. NOAA's National Weather Service. Available at: https://www.weather.gov/lmk/fog_tutorial#:~:text=Evaporation%20or%20Mixing%20Fog&text=Steam%20fog%20forms%20when%20cold,reaches%20100%25%20and%20fog%20forms. [Accessed: 08-Feb-2023].