Predicting Precipitation Trends in Denver, CO

Authors: Ryne Smith and Ryan James

Abstract

A dataset containing precipitation measurements over time will be analyzed for the state of Colorado. A single precipitation measurement station in Denver, Colorado will be the source for the data. The goal of the data analysis is to predict future precipitation trends by utilizing precipitation data from the past eleven years. In order to predict these future precipitation trends, a Random Forest Regressor model will be constructed and run on appropriate features/variables from the data. The best performing model earned an R^2 score of 0.768 meaning that the model is accurately predicting nearly 77% of the data.

Overview

Our Motivation: The main motivating factor in choosing precipitation measurements for the dataset is that we want to better understand the frequency and other behavioral factors about precipitation. Some examples of these behavioral factors are how often it rains/snows over specific periods of time or how hard or intense precipitation might be after a dry period. We find the behavior of precipitation over time to be very interesting and important because we hope to predict future weather trends based on past occurrences. Another reason that we decided to work with precipitation data is because we know that precipitation is commonly predicted among several industries so we will also be able to learn data analysis strategies from scholarly research articles. In summary, we feel that we have the passion and skill to accurately predict future precipitation trends in Denver.

The Dataset: We retrieved daily weather data from the NOAA's Climate Data Online website which provides access to archives of historical weather data worldwide.[1] Using this website we collected daily weather data in the Denver area from 2010-2021. Our dataset currently contains about 145,000 rows of data for a weather station located in Denver, Colorado. The precipitation measurement column is labeled "PRCP" and this column contains integers which represent the amount of rain collected in inches for each instance of rain. However, because the column contains several missing values (about 30,000), we will omit these rows and still have a large dataset to work with. The dataset also includes the date of the measurement and the geographic location and elevation of the weather station (currently constant for all rows).

Problem: Using the NOAA daily weather data for Denver, we want to create a model using linear regression that can predict the precipitation for the city at any point in the year. From this model, we could then attempt to draw conclusions about precipitation trends in Denver over the year. Using the year, month, and date as separate data features we will be able to predict trends on both a day-to-day and month-to-month basis as well as account for trends over the course of several years (i.e. are there

[1] https://www.ncdc.noaa.gov/cdo-web/

noticeable differences in precipitation between years?). Depending on time, we might also be able to expand our dataset to include multiple weather stations in the Denver area, which would allow us to use geographic data in our model, such as latitude/longitude and elevation. Using these features would allow us to view trends in precipitation based on the specific area within Denver and its elevation. Overall, the problem we would like to address is how to predict precipitation in a specific location within Denver given the time of year and (optionally) the specific location/elevation data.

Data Acquisition

We downloaded our dataset from an online source called the “National Climatic Data Center” which contains various climate data for every continent in the world. After looking through several of the options, we settled on a dataset that contains precipitation information for the different cities in Colorado. The precipitation data comes from individual sensors/units within these cities.

Another aspect of the data is that there are several columns which carry the same theme as a “flag” column. These columns or variables contain single letters that each have a corresponding meaning. Because our main goal is predicting precipitation trends, we plan to omit these columns as they are too specific to a certain day and that they contain a “NULL” value most of the time. One more column not included in these “flag” columns is the “MDSF” column. This column measures the amount of snowfall over multiple days in that it stands for multi-day snowfall. This column had very low values and did not correspond properly to the precipitation column because it did not total up inches over past days, rather it contained all values under 1 which made little sense. Instead of looking even further into MDSF, we decided it was not relevant to our goal of predicting precipitation so this column was omitted as well.

In regard to the latitude and longitude columns, these values change throughout our dataset indicating that different units within Denver are collecting data. We have decided to treat these different units as one because they are all reasonably close to each other (<100 feet) so the data will remain accurate and appropriate for the city of Denver.

Lastly for our data cleaning and methodology, we want to primarily focus on precipitation information from Denver, so we wrote a C++ script to omit all other cities. This omission also decreased our rows from around 900,000 to 140,000 which will help speed up our data analysis in that we won't be waiting around for long computational tasks. With all these changes made to our dataset, it will contain only the variables listed in the bulleted list below for the city of Denver.

Dataset for Precipitation in Denver, Colorado

- Station: Serial code for precipitation unit
- Name: City Name in which the precipitation unit is located
- Latitude: Latitude measurement of precipitation unit
- Longitude: Longitude measurement of precipitation unit
- Elevation: Elevation measurement of precipitation unit
- Date: The day in which the precipitation unit records the data
- PRCP: The amount of precipitation (rain/snowfall) measured in inches

From these variables listed above, we will focus our regression workflow as well as predictive models on the variable “PRCP”. The “PRCP” column either contains an empty value or an integer.

The most significant limitation in our dataset is that after we split the day, month, and year into three separate columns, we noticed that there are multiple values for ‘PRCP’ on the same day. This is actually a discrepancy in our data in that the documentation clearly stated that the ‘PRCP’ values were taken on daily intervals. To resolve this issue, we created a function to sum the ‘PRCP’ values when there are multiple of them for a particular day. Our justification for summing these values is that precipitation is measured as the amount of inches of rain/snow that fall; so, when there are multiple values for a single day, the precipitation unit must have picked up multiple storms. In getting an accurate value for the precipitation on that day, the only reasonable strategy is to sum all of the ‘PRCP’ values into a single row for a single day.

As the above paragraph discussed, we plan to omit blank entries for ‘PRCP’ and sum multiple ‘PRCP’ values for a single day. The drawback of this strategy is that we

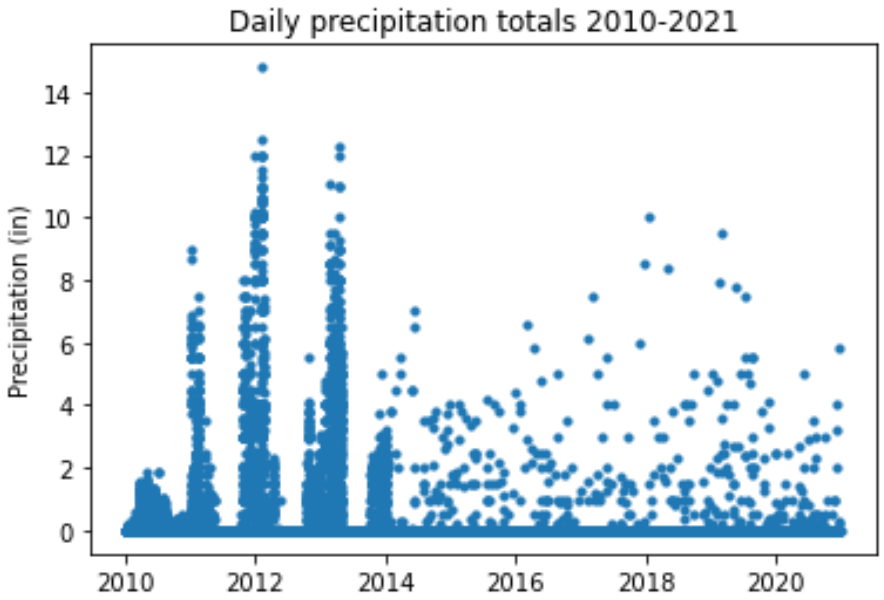
will have less data to work with in that our dataset will likely go from ~100,000 to <50,000 rows (about 40,000 rows). The benefit of this strategy is that our dataset will be much more condensed and more simple to perform analysis on. If our dataset ends up not having a significant amount of rows, we will disregard our previous plan of only studying one station. There are multiple stations in our dataset that we can add back in with the purpose of performing more robust and relevant analysis if needed.

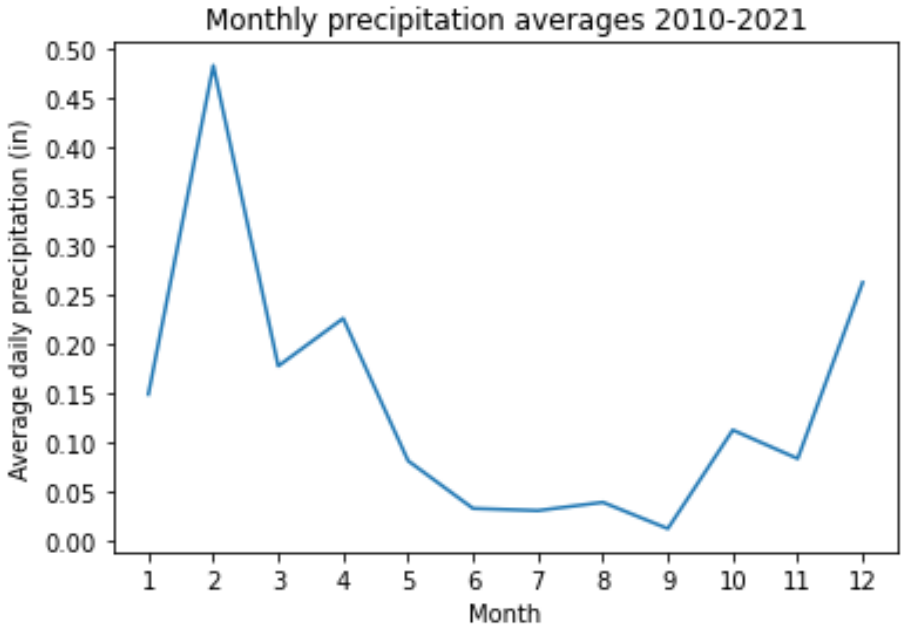
Data Preprocessing

As discussed above, one of the first things we noticed when examining our dataset is that some rows had a null PRCP value, making up about 21.14% of all 140,000 rows in the dataset. We decided that the best option would be to remove these rows as our dataset was already large enough that this wouldn't cause issues, and assuming that these values should be 0 or some other number could cause our model to be inaccurate. Removing rows with null PRCP values led to a reduction in size of our dataset to about 100,000 rows. Another issue discovered during the initial data exploration/cleansing phase was that in some cases we had multiple rows for one station on a particular day, despite the dataset being a "daily summaries" collection. Some of these duplicate rows had different PRCP values as well, so we had to decide how to approach this. The documentation for our dataset made no mention of this so we assumed that the rows represented multiple measurements taken throughout the day, despite the name of the dataset. Although this was an assumption we noticed that most stations had about 35 rows per day before processing the dataset, which would correspond to a fairly consistent 45 minute interval between readings. Because of this we decided to reduce each duplicate combination of STATION and DATE to one row, and have the new PRCP column be equal to the sum of the duplicate rows' PRCP columns. All other columns of interest, (LATITUDE, LONGITUDE, ELEVATION) were constant for each duplicate so we simply kept those values. Removing these duplicates reduced our dataset size to just over 40,000 rows. We also kept backups of our dataset before and after the change to easily revert this in the future if our assumption proves to be wrong. One other modification to our dataset at this time was to create three new columns containing integers (DAY, MONTH, YEAR) based on the DATE column to make

it easier to visualize the data and feed it into the model in the future. We then converted the DATE column to a datetime object to make it directly applicable to time series graphs on our dataset.

At this point, we looked at several visualizations/statistics of our dataset, shown below in the daily scatter plot, monthly line graph, and variable information grid:





	YEAR	MONTH	DAY	LATITUDE	LONGITUDE	ELEVATION	PRCP
count	43140.000000	43140.000000	43140.000000	43140.000000	43140.000000	43136.000000	43140.000000
mean	2011.798632	6.574154	15.735304	39.722157	-104.976779	1645.629569	0.142794
std	1.796270	3.556578	8.814216	0.066413	0.091851	32.604203	0.776578
min	2010.000000	1.000000	1.000000	39.570280	-105.153300	1572.500000	0.000000
25%	2010.000000	3.000000	8.000000	39.679172	-105.046300	1619.400000	0.000000
50%	2012.000000	7.000000	16.000000	39.726300	-104.966400	1645.900000	0.000000
75%	2013.000000	10.000000	23.000000	39.758800	-104.942500	1662.100000	0.000000
max	2021.000000	12.000000	31.000000	39.894700	-104.657500	1793.100000	14.800000

The majority of the time, these visualizations follow the expected trends in that PRCP, being a measure of melting snow/ice or rainfall, should logically be larger in the fall/spring/winter and much lower in the summer, which the monthly precipitation averages graph seems to show. The daily precipitation totals scatter plot however seems to show two unexpected trends; first, the precipitation totals for 2010 seem extremely low compared to the rest of the dataset, and secondly the values for 2014-2021 seem much more random than the preceding years. The latter is likely due to the removal of rows with a null PRCP value from the database, since the percentage of rows without values in 2014-2021 was actually almost three times as high as the percentage from 2010-2014. Likely, these rows would be expected to have a nonzero PRCP value if they were recorded and would have made the trend seen in 2010-2014 more noticeable in 2014-2021. The low precipitation values in 2010 are likely due to some sort of difference in how the weather stations collected/recorded data before 2011, so all of the data from 2010 has been discarded to keep consistency in the data.

At this stage, we've reduced our dataset to only include all combinations of STATION and DATE. So the possible features are ELEVATION, LATITUDE, LONGITUDE, DATE, DAY, MONTH, YEAR, PRCP, and date/geographic location columns. Our initial dataset also included snow data as well as tags to describe the weather, which we decided were outside the scope of our project. These selected features were chosen either because they will be included as inputs to the regression model or because they are useful for interacting with the dataset and visualizing the

data. As explained before, our target column will be PRCP which is why it is not included in our possible feature combination set.

Algorithm Explanation

After plotting the time series graph, “Monthly Precipitation Averages 2010 - 2021”, we see that the data behaves in a reasonable fashion in terms of future regression analysis. In other words, the data does not appear to be skewed or awkward in that it comfortably fluctuates over time and stays consistent meaning there are no significant rises, falls, or long periods of flatlining. Given this healthy behavior, our next step is to predict future trends in the data through statistical modeling.

We will be using sklearn’s Random Forest Regressor to attempt to predict future precipitation values. This model should work well for this data because it is less prone to overfitting and being affected by noise than a standard regression model. To implement the model we’ll split the data into training/testing sets to fit and score the model. In addition, we will test multiple feature sets to determine which produce the highest-scoring model, and do further testing with the best models from that group.

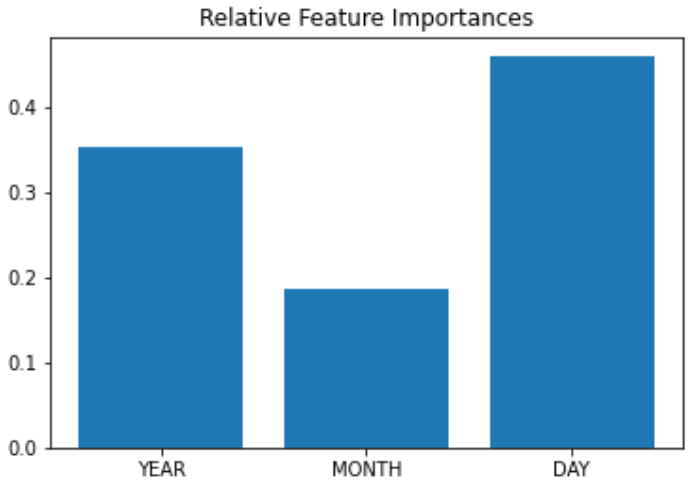
One problem with the random forest model is that it is unable to predict target values outside of the range in the training set; the max/min values from the training set can’t be exceeded. Since our dataset includes multiple years’ worth of data, however, this shouldn’t be a problem since a precipitation value would have to be record-breaking to exceed the range of the training set, and since our model works based on average values it would be unreasonable to expect outliers such as those to be predicted.

Results and Evaluation

To begin our data analysis, we took the entire dataset and identified possible feature attributes as well as a target. The possible features we identified are YEAR, MONTH, DAY, LATITUDE, LONGITUDE, ELEVATION. The target, as before, is PRCP. We then initialized a Random Forest Regression model with exactly 500 estimators as when we used more, the metric scores were very similar (1000 and 1500 estimators were tested). To identify the best set of features to use, we iterated through all possible combinations of the feature set and trained/tested a model with each possible subset.

We scored each model with sklearn's R^2 score metric to identify the best-performing feature subsets. The R^2 score is a measure from [0:1] where the closer the value is to 1, the better the predictive model fits or predicts the data. Some combinations of the features like [DAY, LONGITUDE] received a negative score, which indicated that a horizontal line would be a more accurate fit to the data than the model. Furthermore, this test identified the model with the feature set [YEAR, MONTH, DAY, LATITUDE, LONGITUDE] as the best performing model. However, four other models had very similar R^2 scores, so to account for the effects of randomness in the model performance, we took each of the 5 top performing models and trained/tested them and ran the R^2 score test 4 more times, averaging the R^2 scores for each model. In this test, the [YEAR, MONTH, DAY] model performed the best, with some difference between it and the next 4 models. Because of this we decided to use the [YEAR, MONTH, DAY] model scoring at 0.768 instead of the [YEAR, MONTH, DAY, LATITUDE, LONGITUDE] model scoring at 0.738. To clarify, both of the scores shown before is the R^2 score for the corresponding model.

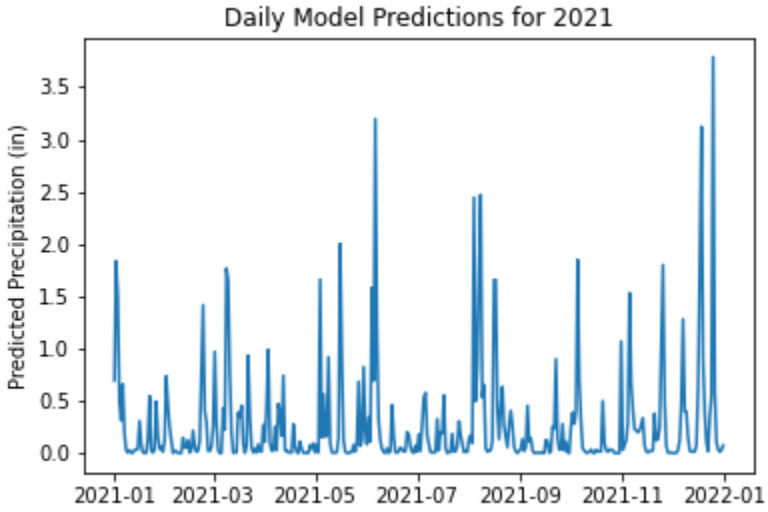
Next we looked at the importance weights to see which features have the most significant impact on the data. A figure with (y-axis = importance (percentage of explained variance), x-axis = feature) is shown below:



As we can see from the histogram, the day feature is the most significant feature with a relative weight approaching 0.5 within our model. We also see that the month column is the least important which implies that month-to-month precipitation behavior is not

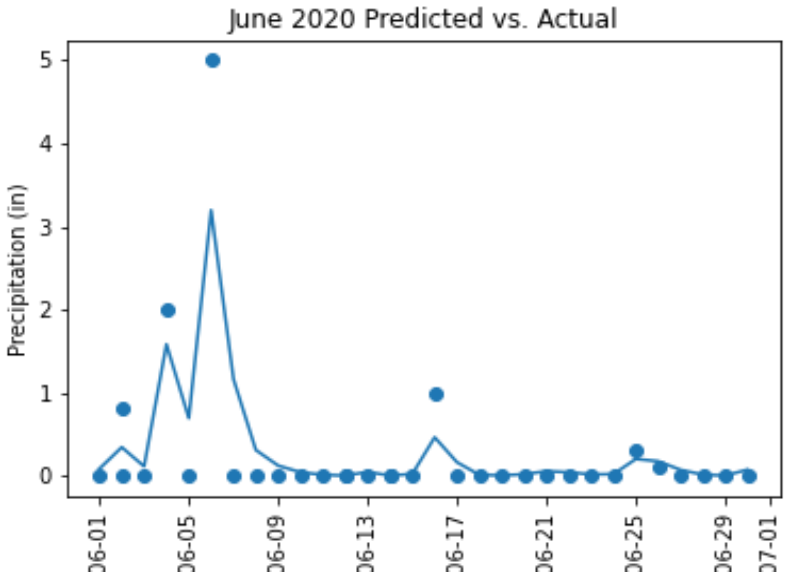
incredibly useful in predicting future trends. While the importance results are interesting, this was more of an exploratory strategy to get a better understanding of the most significant features.

We then visualized the predictive power of our Random Forest Regression model and plotted the predicted precipitation values for each day of 2021. Keep in mind that the Random Forest Regression model utilizes the [DAY, MONTH, YEAR] features. The resulting figure is shown below:

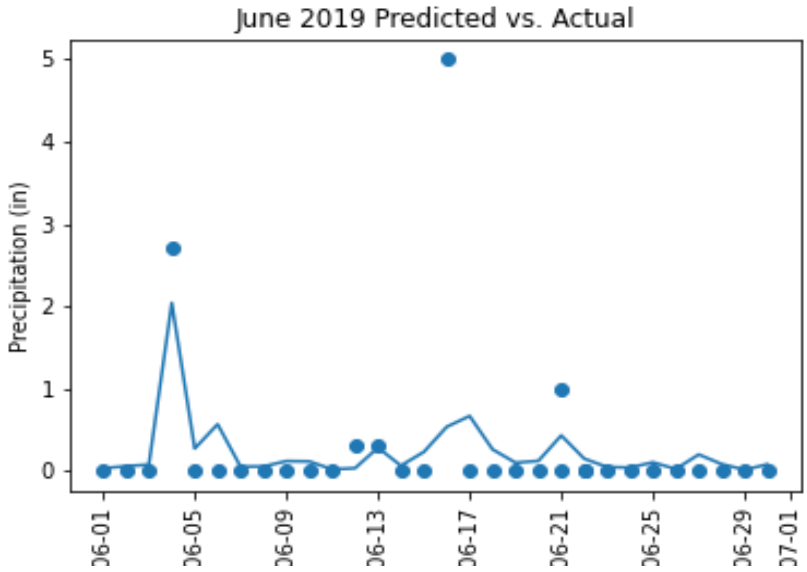


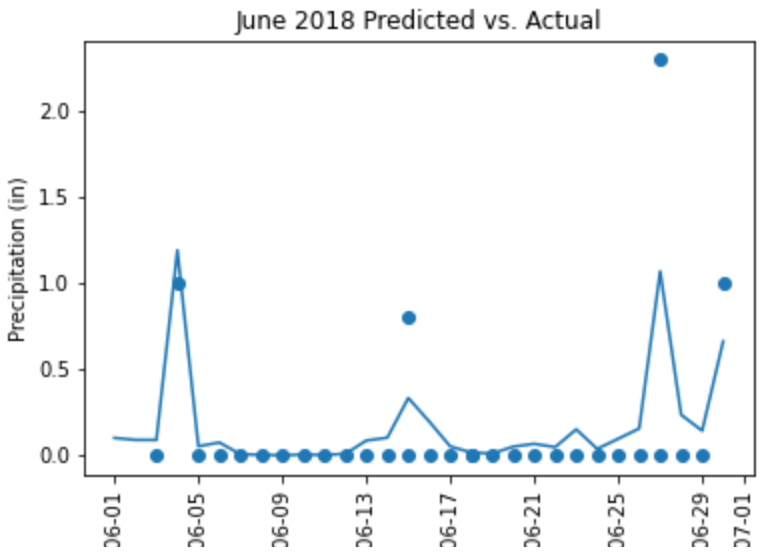
As shown the line graph above, there are several spikes and drops throughout the year. This makes sense when compared to the visualization of our data from earlier in that precipitation is a very volatile variable. One way to test the accuracy of this model would be to analyze the amount of precipitation over the next month or the rest of 2021 and compare that to the predicted values from the model.

While comparing the actual precipitation versus the expected precipitation for a certain time period is a great way to test the predictive power of this model, it doesn't show consistency, which we would expect in a cyclical system such as yearly precipitation. In order to visualize model consistency we can, for example, look at the model predictions versus actual data for the same month over several years. Below is a plot of our predicted values (blue line) versus the actual values (blue points) for June 2020:



From the line graph above, our model is accounting for all data points within the model. This most likely removes the possibilities of the model over or under fitting our data in that the predicted line does not over or under account for any data points. This trend shows where there are outlier data points in that the predictive line significantly traveled to those points, but then returns to the mean points at 0 precipitation. Now we will plot the same information but for June 2019 (first image) and June 2018 (second image) shown below:





The model seems to be predicting the actual values in a similar way. The predicted line or best fit line is accounting for every data point; however, it is not over or under accounting for any data points either which is optimal. A potential case of overfitting is the largest value data point near the beginning of the graph in that we see the predicted line shoot up quite quickly. However, we would expect the best fit line to spike because that data point is a strong outlier compared to the rest of the data points.

Overall, our model right now is showing positive signs in that it is predicting expected values accurately. While further tests still need to be run, we can say with some confidence that our Random Forest Regression model appears to be sufficient in predicting precipitation trends.