Drought Prediction using Rudimentary Meteorological & Soil Variables

Chad Delany

December 2022

Data Sourced from: Kaggle.org, North American Drought Monitor (NOAA)

Introduction:

With increasing climate change and an overall increase in global temperature, the occurrences of drought are expected to become more prevalent and occur in areas that historically over the last several hundred years have been less likely to experience drought. Drought impacts agriculture directly but also impacts water supply to urban areas as well. With a shifting pattern of drought, using long-term historical patterns is becoming increasingly unreliable to predict present-day conditions. Infrastructure to support both agriculture and urban areas has been developed around historic drought patterns and will need to adapt to permanently shifted present-day drought patterns. As well, the countries' economies throughout the world have become inherently intertwined and drought in one part of the globe has significant impact on the rest of the world.

Problem:

The United States has an abundance of weather and soil data compared to other countries. Using basic weather and rudimentary soil data, can we accurately predict droughts in the United States? Can a model be developed that has an accuracy greater than 80% within the next two months? Can these results be generalized to other countries with less available data resources?

Data Wrangling:

The data originally came in four csv files, one for soil data and three for meteorological data. The three csv files for meteorological data were separated by time period. The training set of data was 2000 – 2016. The validation and test set were 2017 – 2018 and 2019 – 2020. The soil

data were a set of static variables. Location for the data were all the counties in the United States. The meteorological data was daily. The drought scores were weekly. The drought scores were derived from a category of severity ranging from 0 to 5. These integers were converted into floating point numbers based on an interpolation of the drought score location and the location of the county meteorological variables, as well as using the percentage of land in that drought category compared to the overall land within the county. The data sets were very clean and did not require any imputation for missing values. The overall initial training set contained 16 years of 18 daily meteorological variables, a weekly drought score, and 30 rudimentary soil variables for 3,143 counties in the US. This training data set contained approximately 2.3 million rows of data. The test and validation sets contained the same information for a four-year timespan.

Exploratory Data Analysis:

I began exploring the training data by looking at the correlation between meteorological variables. There were a significant number of correlated variables (see figure 1) evident from the heatmap and pairwise plots. I created a list of variables that had a correlation higher than 0.92. I also plotted different meteorological variables versus time and verified the seasonal shifts typically expected from such variables. I did a Principal Component Analysis to determine any inherent dimensionality within the meteorological dataset. There initially appeared to be some significance to 6 dimensions but became unclear as the analysis continued. I also explored correlation within the soil dataset, creating a heatmap (see figure 2) and pairwise plot. There were also a significant number of variables within the soil dataset that were highly correlated. Exploration of the weekly drought scores showed that the target variable had a skewed distribution with the significant majority of drought scores being less severe and very few drought scores indicating severe drought (see figure 3).

I also merged the soil data with the test meteorological data based on location and did a PCA analysis. I was unable to process the entire dataset to conduct the PCA analysis, so I split the dataset in half and did PCA analysis on the two sets of data. The analysis was unclear but inherent data structure may be around 8 to 10 dimensions (see figure 4).

The size of the dataset was straining my available resources. I switched from Windows to Linux to reduce operating system overhead and installed more RAM to reduce processing time. I created multiple Jupyter notebooks to minimize system crashes and reduce the occurrence of running multiple processes.

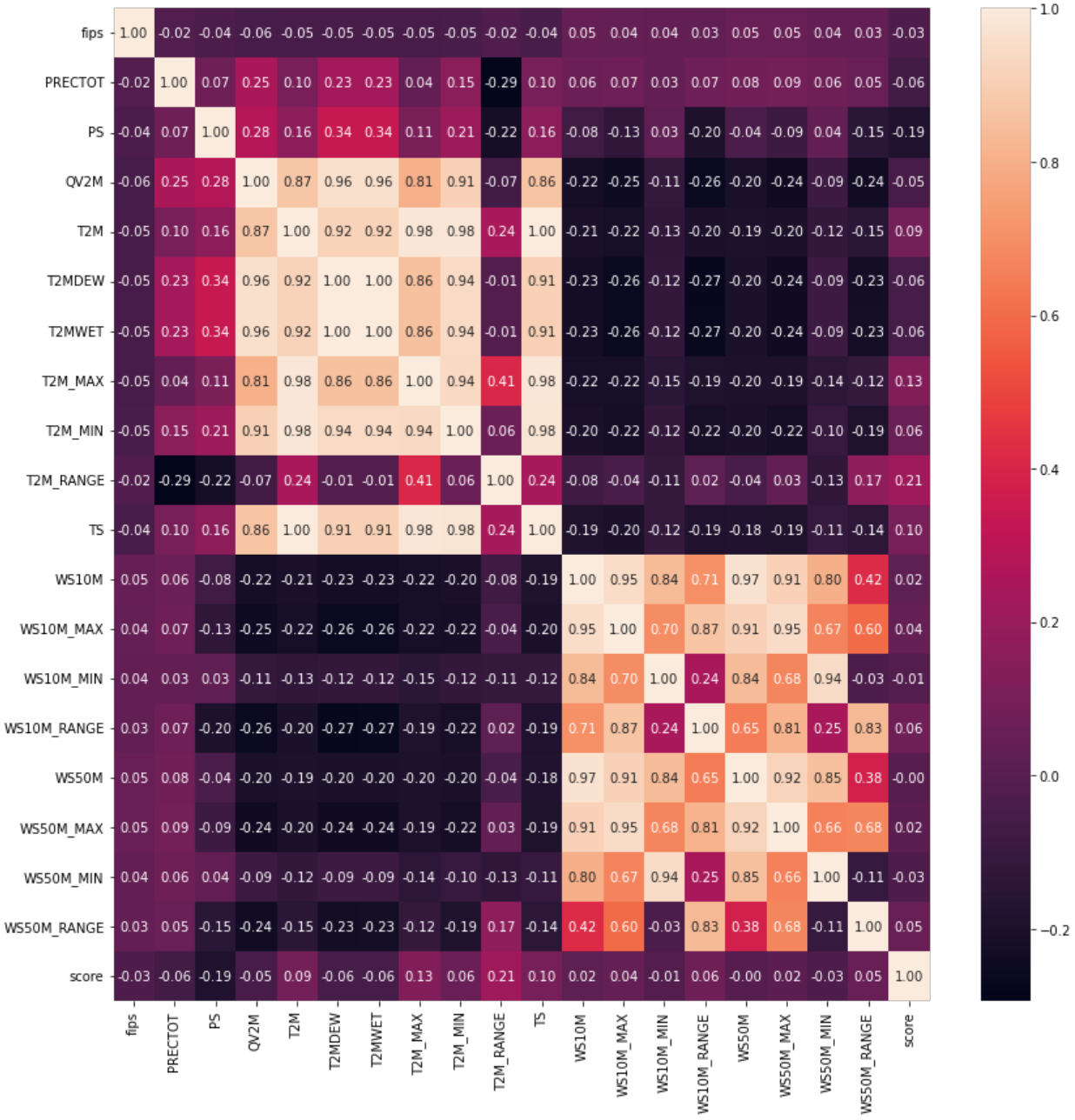

Figure 1. Heatmap of correlation of meteorological variables.

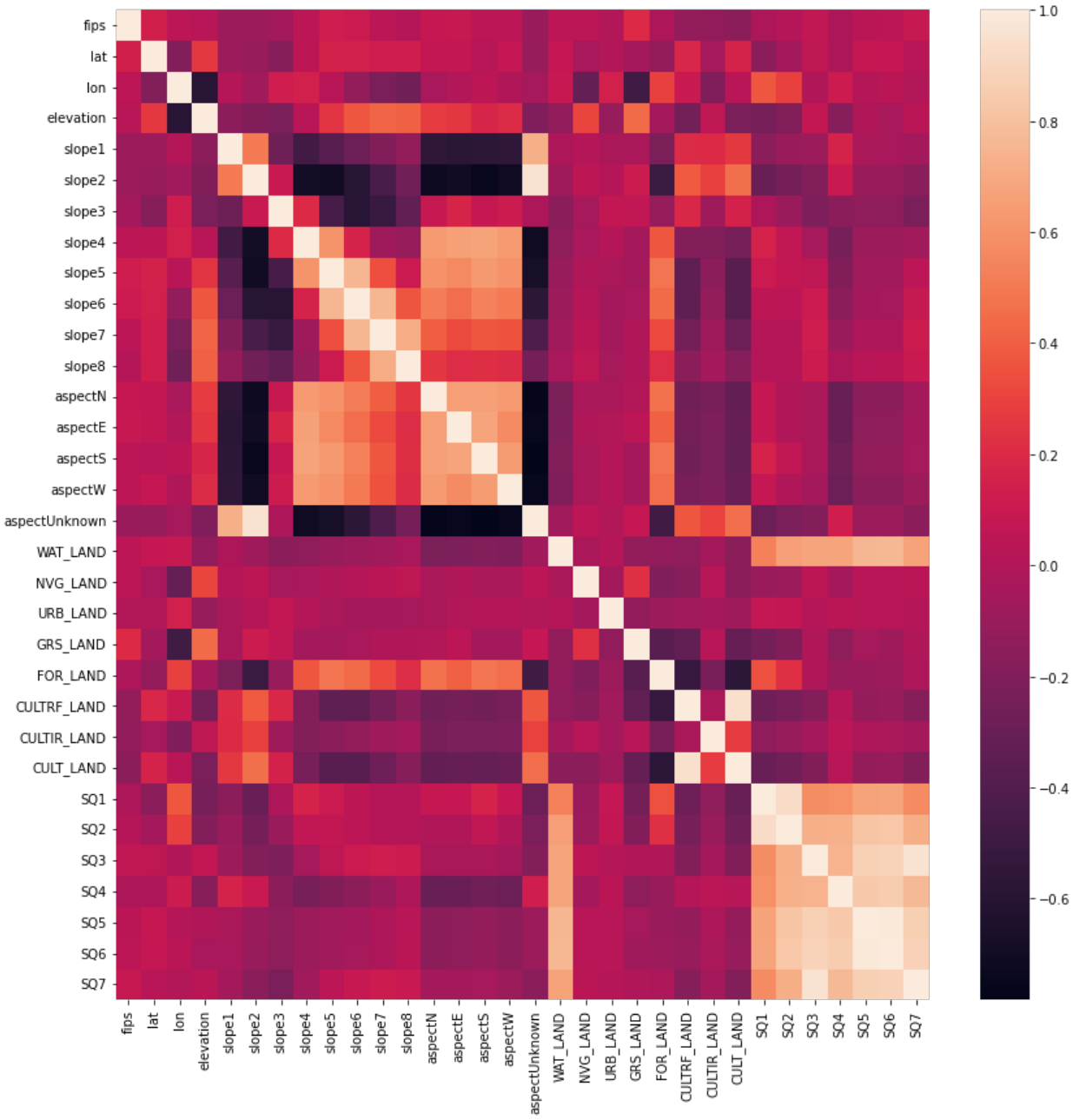

Figure 2. Heatmap of correlation of soil variables.

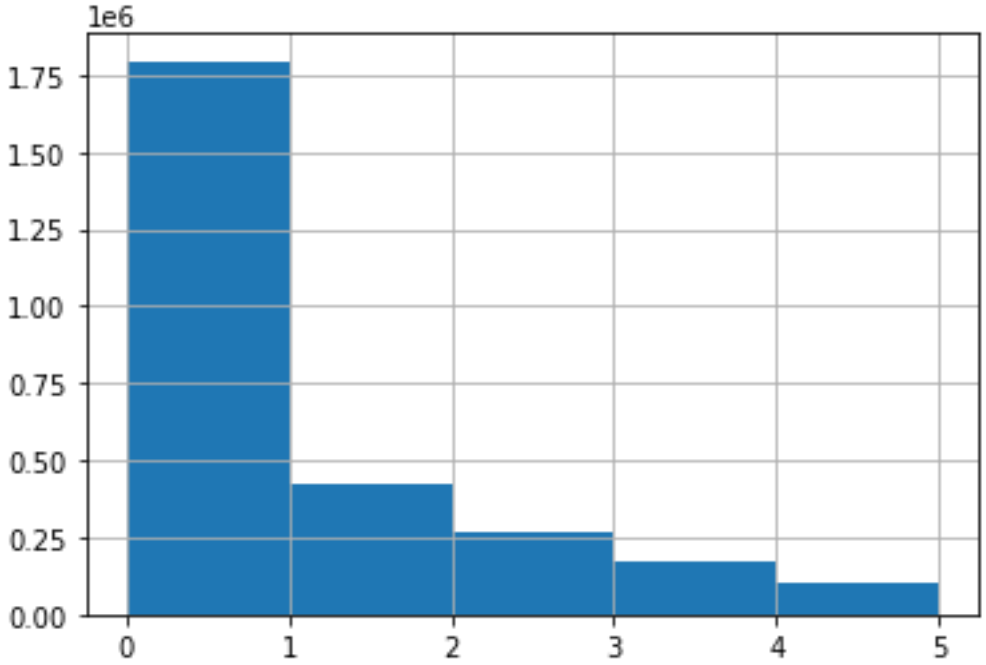

Figure 3. Skewed distribution of drought scores.

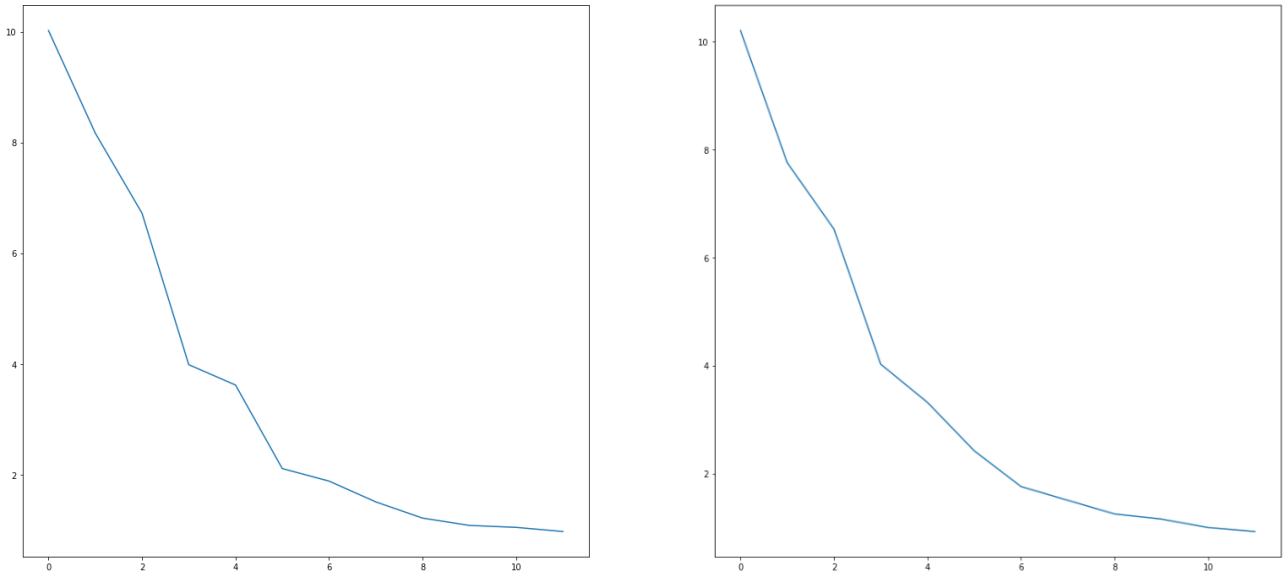

Figure 4. PCA distribution versus number of dimensions for the first half and second half of the combined soil and meteorological dataset.

Pre-processing & Training Data

The soil dataset was merged with the meteorological data based on location to create a complete variable set. The validation and test set were combined to create a single test set. Since the drought scores were weekly and there were issues with running the analysis with available resources, I collapsed the daily meteorological variables into weekly values from the preceding week before the drought score. Since I was uncertain which aspect of the preceding week's meteorological variables might be the most predictive, I took the mean, maximum, and minimum value for each meteorological variable based on the week preceding the drought score. I then scaled all the independent variables to a mean of zero and standard deviation of one to help facilitate the evaluation of multiple types of machine learning models. Since this was still impacting my available resources, this processing was done in two batches based on time period.

Modeling:

Before implementing different models, I created another heatmap to visualize the severity of correlation between variables (see figure 5). This was the bases for feature selection after initially running models on all available variables. There were significant issues with trying to run standard sci-kit learn models on this dataset. I implemented the use of Google Colab Pro+ to gain access to faster CPUs, although this did not increase the availability of RAM and imposed time-out scenarios when the process took more than 24 hours to complete. It was not possible to run different models on this dataset with the available resources. I learned the RAPIDS framework from NVIDIA that has a similar but smaller library of models than scikit learn. The RAPIDS framework allows access to GPU parallel processing. When implemented, models that took several days to run were able to run in 5 to 15 minutes. There were other issues with using RAPIDS that related to a less well-developed library with less support. For example, I wrote my own function to implement cross validation. There were still issues with running cross validation with the available resources, so instead I used multiple runs with different random seeds, which lead to other issues. I initially started by modeling the continuous drought score against the independent variables with different types of regression. These regressions did not create an accurate enough model (see table 1). I converted the continuous drought scores back into their original form by converting them into distinct drought categories from 0 to 5. I ran several classification models. The Random Forest model provided the best model (see table 2). I ran iterations of the Random Forest model to tune the hyperparameters. The resulting model confusion matrix on the training data is in table 3. I did feature selection based on removing variables that had a correlation higher than 0.92 and reran the Random Forest models. The accuracy of the model decreased, and that model's confusion matrix is in table 4. I applied this model to the test dataset and the model confusion matrix on the test dataset is in table 5. This model unfortunately overtrained on the training dataset and while the overall accuracy of the model was robust, the model performed poorly in the severe

drought category, which also had the least amount of data. Also, when attempting to determine the most important input variables the RAPIDS system crashed with the available resources.

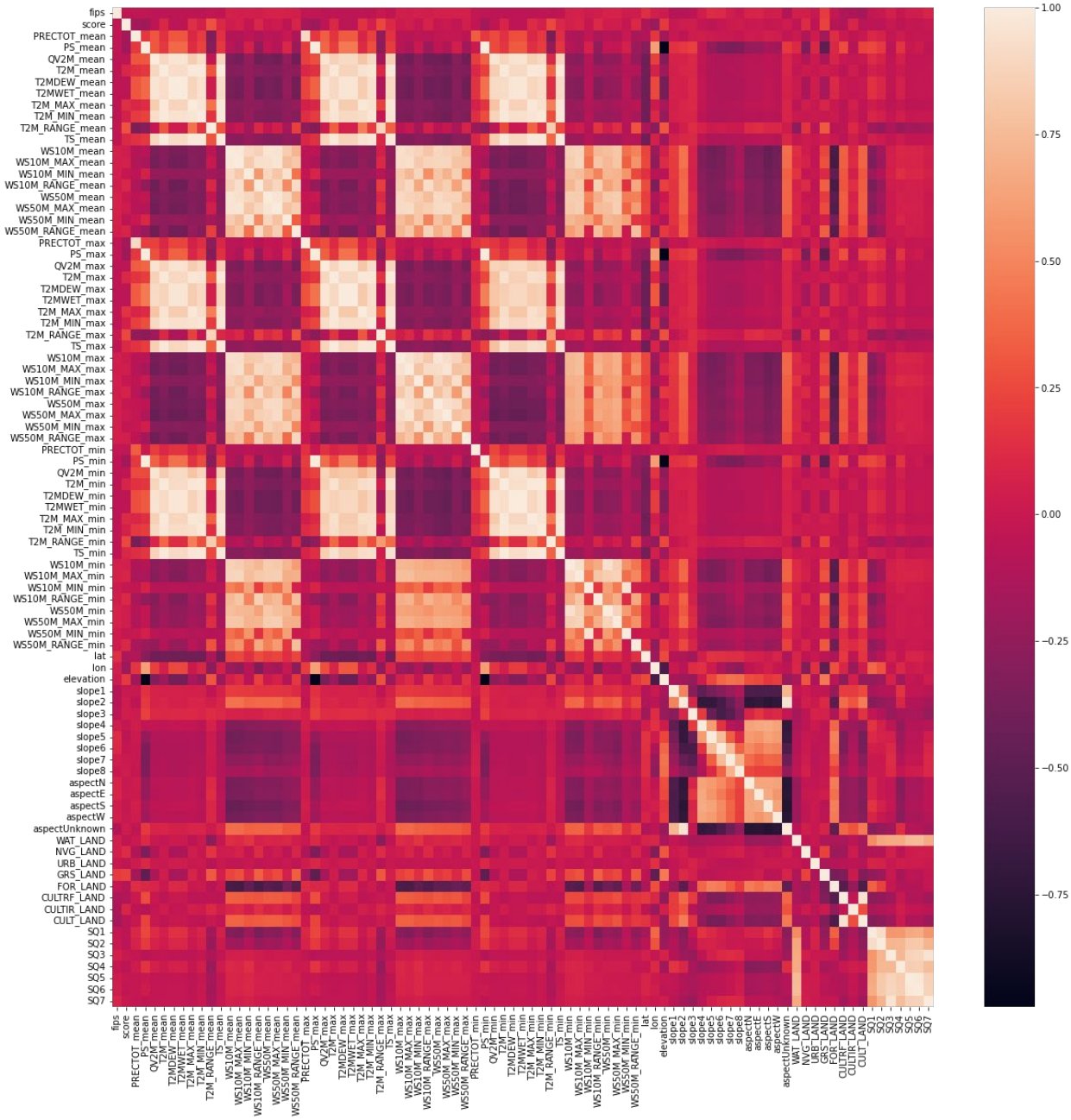

Figure 5. Heatmap of correlation of all soil and derived meteorological variables.

Table 1. Comparison of the regression models' accuracy.

Model & Metric	R**2	MSE	RMSE	MAE
Dummy Regression	0.000	1.497	1.224	0.975
Linear Regression	0.215	1.176	1.084	0.819
Ridge Regression	0.154	1.266	1.125	0.872
ElasticNet Regression	0.074	1.387	1.178	0.929
Nearest Neighbor Regression	0.468	0.796	0.892	0.574
Random Forest Regression	0.714	0.429	0.655	0.434

Table 2. Comparison of the classification models' accuracy.

Model & Metric	ROC AUC	Total Accuracy	Mean Accuracy per Class	Accuracy per Class STD
Logistic Regression	0.844	60%	27%	16%
Nearest Neighbor	1.472	68%	55%	9%
Random Forest	1.720	77%	75%	5%

Table 3. Random Forest model accuracy on training dataset using all variables.

Class	Accuracy
0	78.8%
1	73.5%
2	68.9%
3	68.7%
4	74.3%
5	83.7%
Total	77.0%

Table 4. Random Forest model accuracy on training dataset using select variables.

Class	Accuracy
0	70.3%
1	61.5%
2	58.9%
3	59.1%
4	66.2%
5	76.2%
Total	68.8%

Table 5. Random Forest model accuracy on test dataset using all variables.

Class	Accuracy
0	76.1%
1	17.2%
2	14.6%
3	4.8%
4	0%
5	0%
Total	73.7%

Conclusion and Next Steps:

This process did produce a viable model and demonstrated the usefulness of random forests for this problem. It was not able to highlight a few, key variables. The next steps to improve this model would be:

- Allocate more resources so that the training can be done on the entire training dataset.
- Subset the training dataset and rerun the models to allow for a standard cross validation procedure and allow tools that determine important input variables to be determined.
- Incorporate ordinality information into the classification schema.
- Incorporate a time series analysis that capitalizes on the time nature of the data.
- Use a recurrent neural network to build a time series model.

The initial overall accuracy of the Random Forest model is 74%. With an additional allocation of time and resources, these models could absolutely reach an accuracy above 80%. This is especially true when the cardinality of drought scores is incorporated into the models and ever more importantly the information contained within the timeseries. Additionally, a recurrent neural network may be able to leverage deep learning available within such a large dataset. Given the changing climate and the inherent integration of economies throughout the present-day world, understanding and accurately predicting drought is an important first step in adapting to the current changing conditions of our environment and maintaining a viable global economy. Being able to predict drought from simple variables and not overly complex models, would allow them to be applied worldwide.