**Meteorological Timeseries Data**

The meteorological timeseries dataset was compiled by Christoph Minixhofer and uploaded to Kaggle. This dataset includes data from the [NASA Langley Research Center (LaRC) POWER Project](https://power.larc.nasa.gov/) which is funded through the NASA Earth Science/Applied Science Program. The features in the dataset are listed below, and are collected daily.

* FIPS Code
* Observation Date
* Precipitation (mm day-1)
* Surface Pressure (kPa)
* Specific Humidity at 2 Meters (g/kg)
* Temperature at 2 Meters (C)
* Dew/Frost Point at 2 Meters (C)
* Wet Bulb Temperature at 2 Meters (C)
* Max Temperature at 2 Meters (C)
* Minimum Temperature at 2 Meters (C)
* Temperature Range at 2 Meters (C)
* Earth Skin Temperature (C)
* Wind Speed at 10 Meters (m/s)
* Maximum Wind Speed at 10 Meters (m/s)
* Minimum Wind Speed at 10 Meters (m/s)
* Wind Speed Range at 10 Meters (m/s)
* Wind Speed at 50 Meters (m/s)
* Maximum Wind Speed at 50 Meters (m/s)
* Minimum Wind Speed at 50 Meters (m/s)
* Wind Speed Range at 50 Meters (m/s)

**US Drought Monitor Data (Target Feature)**

[The US Drought Monitor](https://droughtmonitor.unl.edu/) collects drought scores for counties throughout the US on a scale from No Drought - D4. In the dataset these values are floating point values ranging from 0-5, 0 corresponding to No Drought and 5 corresponding to D4. These values are the county average and are collected weekly.

Timeline

Description automatically generated

**Soil Data**

The features in the soil dataset are from the [Harmonized World Soil Database](https://power.larc.nasa.gov/), which contains data generated by the NASA Shuttle Radar Topographic Mission (SRTM). This includes digital elevation data (DEMs) for over 80% of the globe with 3 arc second (approximately 90 meter) resolution at the equator.

The terrain slopes of a given county (FIPS code) are included as a percentage of the entire area that each slope category represents. The slope categories add up to a value of 1 (100%). Similarly, slope aspects for a given county are included and add up to a value of 1 (100%). Other features represent land and soil characteristics.

* 0% ≤ slope ≤ 0.5%
* 0.5% ≤ slope ≤ 2%
* 2% ≤ slope ≤ 5%
* 5% ≤ slope ≤10%
* 10% ≤ slope ≤15%
* 15% ≤ slope ≤ 30%
* 30% ≤ slope ≤ 45%
* Slope > 45%
* North Aspect
* East Aspect
* South Aspect
* West Aspect
* Unknown Aspect
* Mapped Water Bodies
* Sparsely Vegetated Land
* Built-up Land
* Grass/Scrub/Woodland
* Forest Land
* Rain-fed Cultivated Land
* Irrigated Cultivated Land
* Total Cultivated Land
* Nutrient Availability
* Nutrient Retention Capacity
* Rooting Conditions
* Oxygen Availability to Roots
* Excess Salts
* Toxicity
* Workability

**Notes and Methodology**

Changing the problem to a classification problem:

* I will still aggregate the previous weather as new features.
* I will make the new target feature the drought score for 30 days (or 60 days?) into the future, the current score will become one of the predictor features
* I will turn date into month and convert it to a categorical feature
* Change present score to be categorical
  + 0-1 = 0(low drought)
  + 1-2 = 1
  + 2-3 = 2
  + 3-4 = 3
  + 4 = 4

One consideration about this data is that it is time heterogeneous. The sampling frequency of the target feature “score” is weekly, while the sampling frequency of the meteorological data is daily.

* I will create new features which aggregate the weekly and monthly sum of precipitation, and the weekly and monthly average for the other features.
* I will then remove the daily features so that each row corresponds to a separate week, each row will still contain the daily information in the form of the weekly/monthly sums and averages.

Exploratory Data Analysis techniques to implement

* Pd.crosstab

Another consideration is that this is a multivariate time series problem, which is more complicated than an univariate time series problem.

* I will first create a simple univariate time series model which only uses the ‘score’ feature, this will create a baseline that I can compare future model performance against.
* I will then introduce the weekly and monthly sums/averages features, turning the time-series problem into a multivariate one. I will then compare the performance of this to the baseline.
* I will then introduce the non-timeseries soil data into the model, and assess performance by comparing against the baseline and the sum/average model. **Merge this into main dataframe during beginning.**

Start with the simplest approaches:

1. Predicting the last value observed
   1. Univariate with just the score for a baseline?
   2. Using sktime we can use the NaiveForecater(strategy=”last”)
   3. Evaluate loss with sMAPE
2. Predict the last value observed in the same season
   1. Univariate with just the score for a baseline?
   2. Using sktime we can use the NaiveForecater(strategy=”seasonal\_last”)
   3. Evaluate loss with sMAPE