Capstone Project 1

Introduction:

For my Capstone Project, I decided to work with a company named Tule. Tule helps growers make irrigation decisions. The company provides its customers with these deliverables: the Actual Evapotranspiration (i.e., the water use of the crop field), the water stress level of the crop field, the amount of water applied to the crop field, and irrigation recommendations.

Tule installs a hardware device in farmer's fields. The hardware device itself measures two things: 1) the Actual Evapotranspiration (abbreviated as Actual ET), and the amount of water applied to the crop field. The other two deliverables (plant water stress and irrigation recommendations) are derived from the Actual ET and applied irrigation amounts. For our work here, the irrigation recommendations are not under consideration. We will instead be focusing on the plant water stress.

In order to determine the plant water stress, Tule uses the following physical model that describes how plants use water. The amount of water a plant uses (i.e., the Actual ET) is influenced by the weather, the size of the canopy, and the water stress level.

$$\text{Actual ET} = \text{Weather} * \text{Canopy Size} * \text{Water Stress}$$

The weather component can be estimated using another model called Reference ET. Reference ET is computed using readily available weather data (i.e., wind speed, solar radiation, vapor pressure, and air temperature). In other words, it is easy to obtain.

$$\text{Actual ET} = \text{Reference ET} * \text{Canopy Size} * \text{Water Stress}$$

Tule provides its customers with plant water stress measurements by rearranging the above equation.

$$\text{Water Stress} = \text{Actual ET} / [(\text{Reference ET}) * (\text{Canopy SIze})]$$

Tule needs the Canopy Size in order to provide the Water Stress Measurements. In order to get the Canopy Size, Tule makes the assumption that there is no plant water stress during the spring period of the growing season, because winter rainfall has been stored in the soil. Therefore, the model now looks as follows.

$$\text{Canopy Size} = \text{Actual ET} / [(\text{Reference ET}) * (\text{Water Stress})],$$

$$\text{Where Water Stress} = 1, \text{ so}$$

$$\text{Canopy Size} = \text{Actual ET} / [(\text{Reference ET}) * 1)], \text{ so}$$

$$\text{Canopy Size} = \text{Actual ET} / \text{Reference ET}$$

Canopy size is a challenging parameter because typically there is a cover crop grown in orchards/vineyards. This cover crop is responsible for some of the water use. As the plants bud break and begin to enter the growing season, the grower will mow the cover crop and it will eventually die off. The crop then grows until it reaches a local maximum. This local maximum is

what is used for the Canopy Size variable. In essence, Tule is looking for the state transition from the state where the canopy is actively growing to the state where the canopy is no longer growing. It is difficult to identify the state transition, because Tule sensors measure the combined canopy size of both the cover crop and the economic crop. The ecosystem-scale canopy size (i.e., cover crop and economic crop) measurement is called the Plant Response Index (PRI). In addition, it can be tricky to determine as plant growth rate is highly subject to water input (rain), variable weather, etc. Currently, the way the state transition is found is by a person looking at a graph for a local maximum and making an informed decision on when they think the canopy size has reached a steady-state.

Data:

- General
  - A tower is a Tule sensor. For each field we work in, there is one tower.
  - Tower_id is the master key in our database. Each tower we install has a tower_id. The different tables refer to each other via the tower_id
- Towers
  - About: this table has the metadata associated with each Tule sensor and the field it is located in.
  - Fields
    - tower_id: id for the tower. This is the same as tower_id on other tables. It is the master key.
    - crop: the type of crop (text)
    - metacrop: the superset crop name (i.e., tree, annual, vine). Generally, the pattern of canopy growth is similar for each metacrop. (text)
    - between_row_feet: the space between rows. The closer the rows, the larger the max canopy size. (feet)
    - slope: the slope of the field (degrees)
    - aspect: the aspect of the field (degrees)
    - region: the geographical region the sensor is located (text)
    - subregion: the geographical subregion the sensor is located (text)
    - Installed_at: the date the tower was installed (date)
    - young_plant: is the plant a young plant? Note: this would be better if it was the planting date, because many of our young_plant values are stale (boolean)
    - trellis_type: vineyard trellis type. Only applicable to vines. Within a region, max canopy size tends to group around trellis type. (text)

	tower_id	crop	metacrop	betweenrowft	slope	aspect	region	subregion	installed_at	young_plant	trellis_type
0	4068	grape	vine	5.0	4	85	NCoast	napa	2019-04-03 18:42:35	True	vsp
1	3490	grape	vine	8.0	0	297	NCoast	napa	2018-04-19 22:57:42	False	vsp
2	1297	grape	vine	10.0	1	252	NCoast	napa	2016-05-20 19:16:29	False	lyre
3	61	grape	vine	7.0	0	0	NCoast	napa	2014-04-30 04:00:00	False	vsp
4	20	grape	vine	8.0	0	0	NCoast	napa	2014-04-21 04:00:00	False	lyre

- Tower_seasons table
  - About: This table has the information about when a field reached its max canopy and the size of the canopy.
  - Fields
    - tower_id: id for the tower
    - year: the year that the max canopy value and date correspond to (date)
    - actual_max_canopy: the size of the canopy (i.e., the crop coefficient) (dimensionless)
    - max_canopy_date: the date that the field reached its max canopy size. In other words the date of state-transition (date)

	tower_id	year	actual_max_canopy	max_canopy_date	parsed_dates
2	3512	2019	0.458826	2019-07-08	2019-07-08
13	825	2019	0.659543	2019-07-01	2019-07-01
26	2048	2019	0.660164	2019-06-17	2019-06-17
31	3994	2019	0.700000	2019-05-27	2019-05-27
32	5442	2019	0.900000	2019-10-21	2019-10-21

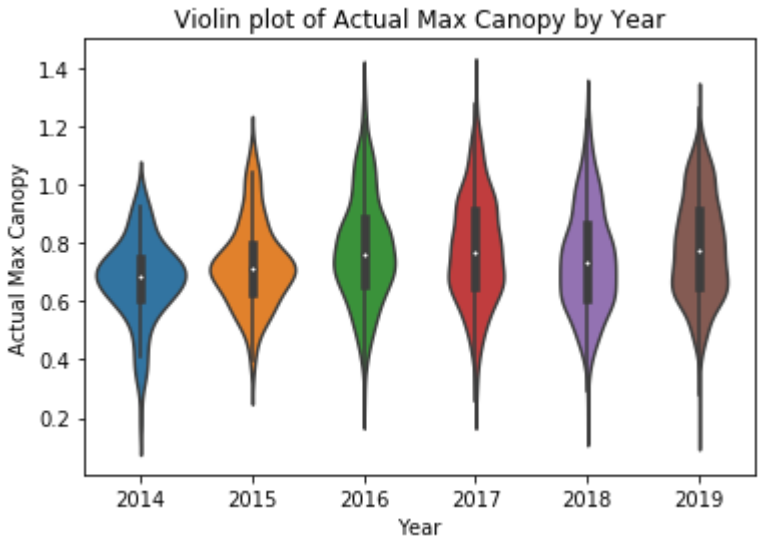
- Tower_weekly
  - About: this table has the weekly PRI data. PRI is calculated as the Actual ET measured by a Tule sensor, divided by Reference ET. PRI tells us about the size of the field's canopy. In the winter, the field's canopy is just the cover crop. In the spring, the field's canopy is both the cover crop and crop. In the summer, the field's canopy is just the crop. The max canopy value is selected from a weekly average value, so the data here are the weekly average values.
  - Caveat: For almost every tower_id and year, one of the weekly values in this table corresponds to the max canopy value in the tower_seasons table. ***For some towers, we manually overwrote the max canopy value using an admin tool. Please discard the towers where one of the weekly values in tower_weekly does not match the actual_max_canopy value in tower_seasons.***
  - Fields
    - tower_id: id for the tower.
    - date: the beginning date of the seven-day period over which the average weekly PRI was calculated
    - pri: this is the ecosystem canopy size. The max canopy value is calculated, generally, from a local maximum of this variable. PRI is calculated as the average weekly value of Actual ET / Reference ET. (dimensionless)
    - irrigation_mm: this is the amount of water the grower applied to the field. It is the weekly sum. (mm)
    - actual_precip_mm: this is the rainfall that the field received. It is the weekly sum. (mm)
  - (jan 1 through aug 31)

	tower_id	date	irrigation_mm	actual_precip_mm	pri
0	10	2019-01-07	0.0	14.204270	1.183220
1	10	2019-01-14	0.0	85.758867	1.158736
2	10	2019-01-21	0.0	0.000000	1.063871
3	10	2019-01-28	0.0	43.022252	1.088352
4	10	2019-02-04	0.0	19.813417	1.166881

Data Cleaning:

1. Remove NAs from each tower.
2. Specific cleaning steps for tower weekly data caveat:
    a. Narrow down to columns of interest.
    b. Merge previous dataframe with tower_seasons.
    c. Create a new column 'different' which calculates difference between max canopy and PRI. Remember we want values that are NOT different aka the difference should equal 0.
    d. How many cases did equal 0 (should be close to 600)?
    e. Add the other variables of interest back in.
    f. Perform a merge with other tables to combine all relevant information.

Visualizations:

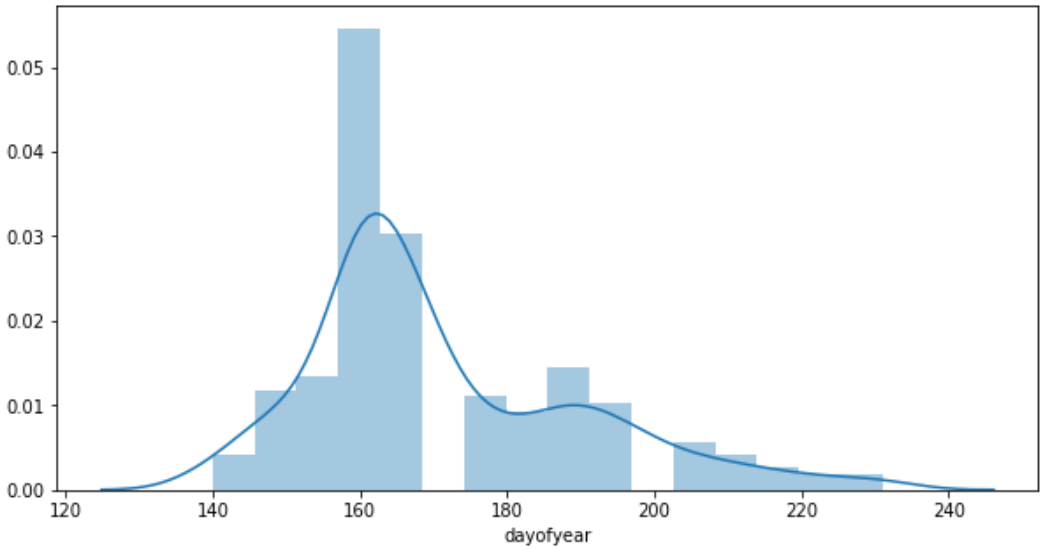


Overview of Methods:

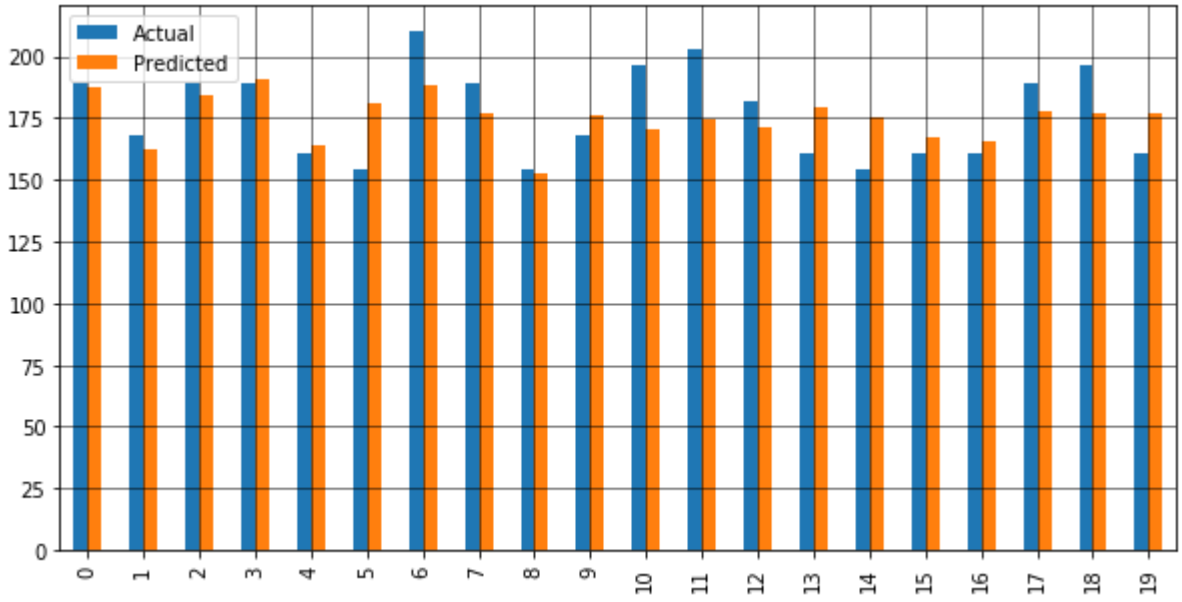
1. Simple linear regression using the fully merged “final table”. Convert the date to a number (day of year) and use the other variables to predict the date.
2. Write a function which calculates and plots time series data with local minima and maxima highlighted for each tower.
3. Merge towers with tower_seasons, perform basic cleaning steps, and calculate the average day of max canopy date by tower and crop, as well as standard deviation. Use these values to create a plot which can inform the range of days for each crop’s max canopy date.
4. Create an ARIMA model with forecasting for each region.

Simple Linear Regression:

On the final table generated after all the initial cleaning steps, convert the actual max canopy date to a number representing the day of the year, and using other variables perform a simple linear regression to see if we can predict actual max canopy date. Predictors include: tower id, year, pri, irrigation, actual precipitation, between row feet, slope, aspect, crop, metacrop, region and subregion. These latter four are categorical variables and dummy variables were used.



The plot below shows the actual vs. predicted values for the towers (only 20 are shown here).



Mean Absolute Error: 12.53573782082062
Mean Squared Error: 269.1132291454739
Root Mean Squared Error: 16.40467095511135
Coefficient of Determination: 0.24603196000344388

Interpretation:

Adjusted r-squared: This is a fairly low r-squared in that it described about 25% of the variance in the model which means it is missing a large amount of what is going on in the data.

The MAE, MSE and RMSE are all very large. This indicates poor fit of the model.

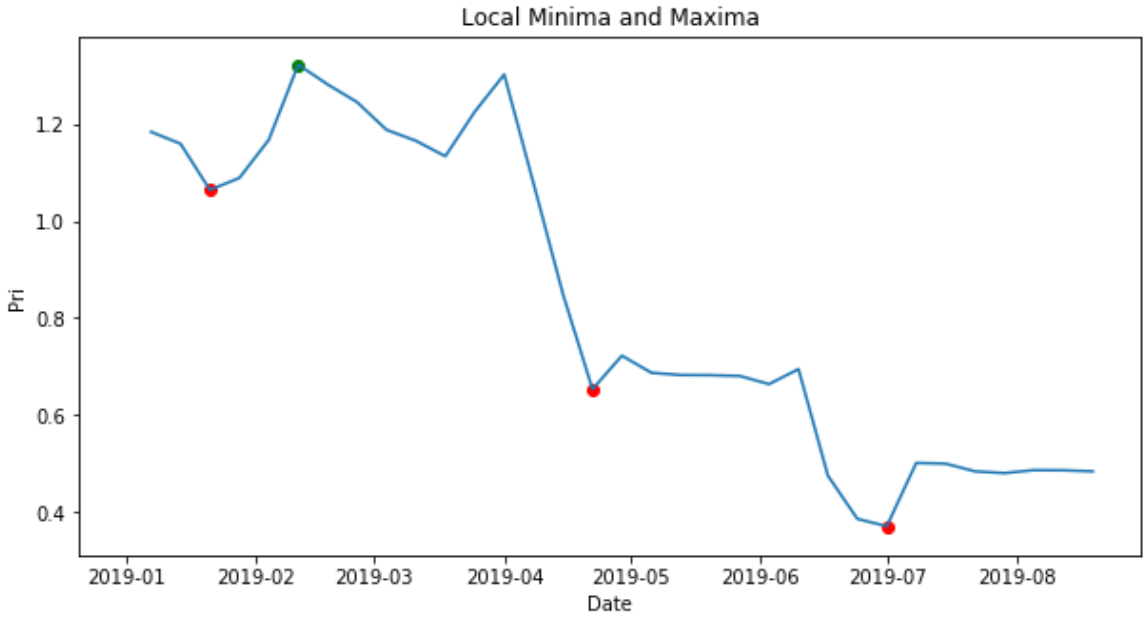
- Mean Absolute Error: The measure of distance between two continuous variables.
- Mean Squared Error: The average square difference between the estimated values and the actual values.
- Root Mean Square Error: Standard deviation of the residuals.

These values are all quite large, indicated predictions and actual data are not close. Although for some towers the values are very similar. The inaccuracy of the model is likely due to large amounts of variability in the actual max canopy dates.

Local Minima and Maxima

For this section I created a function which allows you to determine local minima and maxima, the number of terms considered is modifiable based on an order term. This function is particularly useful for visualizing individual tower data and finding the localized minima and maxima.

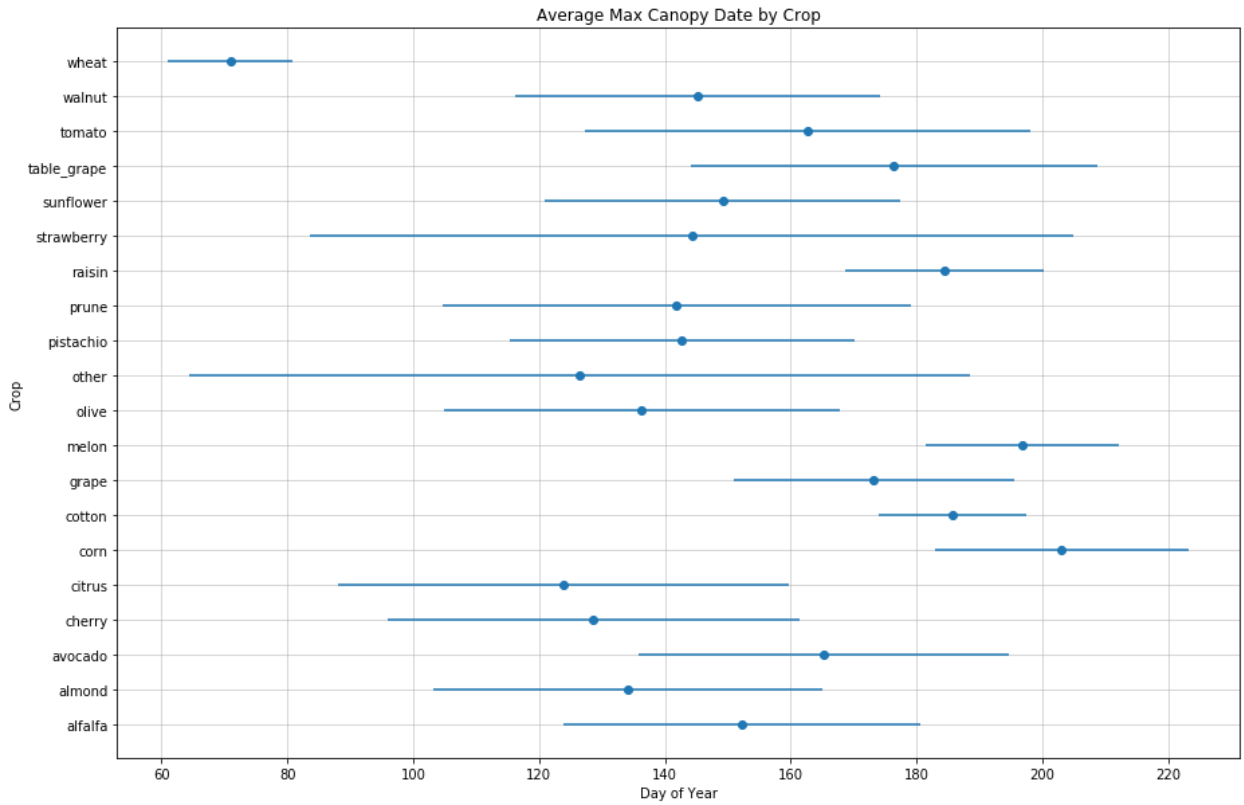
I wrapped it into a function which allows the user to view this plot for each tower_id. This can be a great way to get a quick look at the minima and maxima in the time series and make a possible decision for the maximum canopy date or a steady-state transition. For cases where the user wants to look at the change in pri value for each tower, this function can very quickly help you plot it and decide when the max canopy date is.



Average Max Canopy Date:

This graph shows the average max canopy date with standard deviation for each crop.

This graph was generated using a dataframe containing tower id, with the number of years there was data for the tower, the average canopy date and the standard deviation of the max canopy date. This data was further broken down by crop to generate the graph. Some crops had very large ranges for max canopy date. As more data is collected or as outlier identification is applied, these windows will likely shrink to a more useful range. Ultimately, it is helpful to have a visual aid when evaluating max canopy date. This graph gives an idea of when we can at least expect the max canopy date to be reached. This graph could be modified as well to plot by region and subregion if you want to get a slightly different view of max canopy date.



ARIMA with Forecasting:

Data preparation:

1. Combine cleaned tower and tower_weekly data sets to get region information for the weekly measurements.
2. Group by date and use median as aggregate function.
3. Subset into regions.

1. Data must be stationary. To confirm this test using the Augmented Dickey Fuller test. ($p > 0.05$ thus data is not stationary and we must adjust the three different components of the model).
2. Find out the order of differencing (d) portion of the ARIMA using the autocorrelation plots.
    a. Looks like d = 1 because the first lag is out of the significance range.
    b. If series is under differenced we can always add an AR term, if it is over differenced we can add an MA term.

3. Find out the number of lags / AR term (p) using partial autocorrelation plots and counting the number of lags above the significance level.
. Only one is past significance level, so p = 1.
4. Find out the number of lagged forecast errors / order of MA terms (q) again using the autocorrelation plots. (q is 1 or 2).
5. Plot the model using the determined d,p,q terms.
6. We can also use auto arima to try out multiple possible permutations of the p,d,q terms to see which one has the best AIC, however I preferred to manually change it and assess significance of terms, AIC and forecasting plot.
7. Split data into train and test (a little tricky due to the limited number of points). For my data I aggregated the dates using median so that I could have more information from multiple towers in the same region. Ultimately this led to a much better model.
8. Plot forecast against prediction to see how it looks. In the case described in this write up (and in general) it was usually pretty close.
9. Various metrics were calculated to evaluate performance. Most important being MAPE which was quite low. This value indicates the accuracy of model predictions (90+% in my case). ME, MSE and RMSE were all very low as well.

'mape': 0.0884392534284488,
'me': -0.06753894179821472,
'mae': 0.06753894179821472,
'mpe': -0.0884392534284488,
'rmse': 0.07197847644923835,
'acf1': 0.015273876972158568,
'corr': -0.09076953803168887,
'minmax': 0.0884392534284486

The aim of this is to make forecasting predictions for pri values. Data was grouped by region and dates were aggregated by median. Data was tested for stationarity using the Dickey-Fuller test. To determine the appropriate d,p,q terms for the model the graphs for autocorrelation and partial autocorrelation were used. Multiple permutations were compared, evaluating coefficient significance, AIC and the appearance of the graph to make a final decision. To test the model we split the data into train and test sets. The mean average percent error was very low indicating a high performance accuracy for prediction on the data (95% +).

There is considerable potential for this model to make forecasting predictions for max canopy date based on region, subregion or crop. Ultimately, as more data is accumulated the predictions will improve. However, the forecasting gives a good baseline for any predictions and can help inform future decisions. Also, the aggregation of data by region in this case increases the power of the forecasting. Data can be aggregated in different ways depending on the user's interest, and this can be done easily.