PREDICTING PHENOPHASE PRESENCE BASED ON WEATHER VARIABLES

Team 4: Andrea Newlands, Christian Ibanez Diaz, Linda Sylvester

Deepnote Project - Milestone 2 Fall 2023: Team 4

INTRODUCTION

As climate change progresses, there can be immense impacts on ecosystems as warmer seasons occur sooner in the year. Our project will aim to build tools to help predict the impact of climate change on plants. This can help with mitigation planning and quantifying the impact climate change can have.

Phenophases refer to the lifecycle events for plants such as first leaf, first bloom, bloom length, leaf fall, etc. Based on the findings of the Milestone I project[1], we were motivated to investigate using local weather variables (i.e. temperature, precipitation, solar radiation) to predict if given weather values are suitable for predicting a plant's first leaf and first bloom. In order to summarize the weather leading up to the phenophase, we used a 30 day rolling average. We focused mainly on temperature as many research articles cite temperature as the main driver of the timing of phenological events[2]. But we will also look at the additional weather variables, such as precipitation and solar radiation, in order to predict phenophases and categorize weather types.

We explored three different types of supervised learning models. Our supervised model aimed to use dummy variables and rolling 30-day weather values to predict the presence of a phenophase. The KNN model used 501 neighbors and performed well but had a large number of false positives, likely due to the fact that a phenophases like first leaf and first bloom can only occur once per plant per year even if other days may have the same or very similar weather values. Because of this, we are more concerned with recall than precision. We then moved on to Logistic Regression models. We first used a multinomial logistic regression to predict the phenophases, however this model performed particularly poorly, with a score of 0.66, a recall of 0.75, and a precision of 0.34. Because of the number of false negatives assigned in the incorrect phenophase, we decided to try two separate logistic regressions with a single phenophase being modeled in each one. These performed well, with the first leaf model having a recall of 0.83 and the first bloom model having a recall of 0.76. Finally, we created a random forest model. We experimented with different depths and concluded that a depth of 5 was best. This model returned a recall of 0.89 and a precision of 0.35.

For the unsupervised tasks, we explored the weather data through Principal Component Analysis (PCA) and performed clustering on the weather data with K-Means. The features that were most important in the daily weather, according to PCA, were maximum temperature (captured 55% variance in 1st component), solar radiation (captured 21% variance in the 2nd component) , and precipitation (captured 14% variance in the 3rd component) when including all weather features. Seasonal variation in daily weather affected the important features but they were still features that were related to our originally found components. Scaling made a difference in the amount of variation captured by PCA but this is to be expected as scaling is an important preprocessing step for that method. K-means clustering also requires scaling and the same scaler used for PCA was used for K-means. We found the daily weather clusters were related to seasonal variability. The 4 weather typologies found are closely related to a hot summer day, a cold winter day , a very cold day with snow on the ground, and a day with rain.

RELATED WORK

1) Impacts of Climate Extremes on Vegetation. Milestone I project.
   Link:https://docs.google.com/presentation/d/1KapxsH3-BioSDXi60knoKHEpZM4Muk2pixOM1frSBn8/edit?usp=sharing

   This Milestone I project explored correlations between temperature and precipitation extremes and plant phenologies. Our current project will build on the Milestone I findings to investigate whether the date of first leaf and first bloom can be predicted using weather variables.

2) Automated data-intensive forecasting of plant phenology throughout the United States
   Link: https://esajournals-onlinelibrary-wiley-com.proxy.lib.umich.edu/doi/10.1002/eap.2025
   Github: https://github.com/sdtaylor/phenology_forecasts
   Python Code: https://pyphenology.readthedocs.io/en/master/index.html

   This project looks at using climate models to forecast plant phenologies months in advance. Our project will explore the basics of using weather variables to see if/how they can predict the date of first leaf and first bloom for lilacs.

3) Deep Learning in Plant Phenological Research: A Systematic Literature Review
   Link: https://www.frontiersin.org/articles/10.3389/fpls.2022.805738/full

   This paper reviews the work that has been done in using deep learning in the phenological research space. It discusses the importance of phenological research in the plant sciences but also in climate science. A lot of work has been done using remote sensing with photography and satellite imagery, but less focus on using human observed data. Our project would add to the small space of using supervised and unsupervised learning on human observed data.

DATA SOURCE

Phenophases

This dataset contains the plant species, location, and day of the year for different observed phenophases such as first leaf, first bloom, and end bloom for the years 1965-2022. The variables used are mostly numerical with only state, genus, and phenophase description being nominal.
Source: USA National Phenology Network[4]
Link: https://data.usanpn.org/observations/

The website link above allows you to search and generate customized datasets from the National Phenology Database using different filters to specify dates, locations, species, and phenophases of interest. The dataset can then be downloaded in a csv format. The dataset was originally downloaded and processed for the Milestone I project.[1] We imported the dataset that was downloaded for Milestone I.

The phenology csv consisted of 354,233 observations with 27 features, covering 17 states in the Eastern United States, and the years 1965-2022. It contained many detailed phenophase categories and dates of observation for 343 genera.

From the Milestone I project, we knew we should focus on lilacs (genus *Syringa*) located in the Northeastern states. To match the time periods of our weather data we narrowed the phenophase data to the time periods between 1980 and 2020. The observations are recorded by volunteers, and because of this, there are fluctuations on the number of observations for some time periods and phenophases captured. The phenophases of First Leaf and First Bloom had a consistent number of observations compared to other options (see Figure 1).

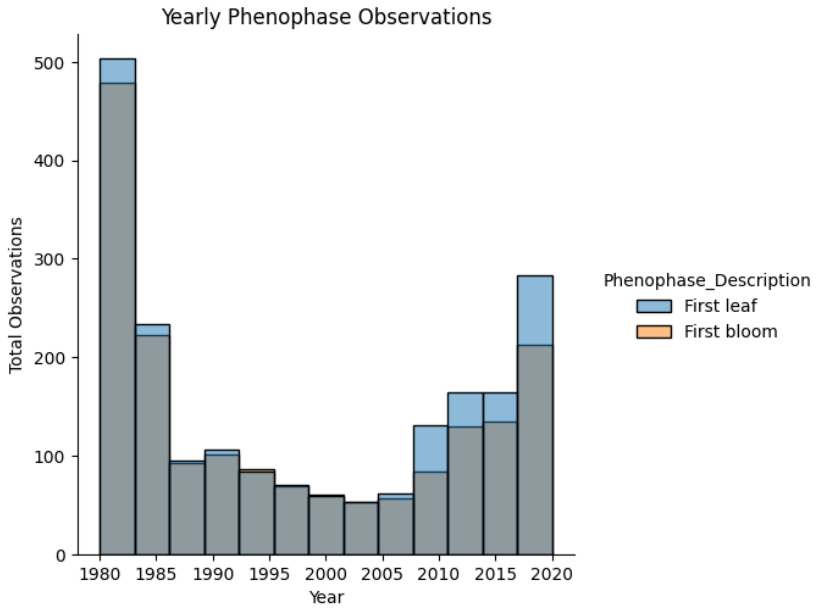

Figure 1.Phenophase Observations for Lilacs (*Syringa*) in the Northeast 1980-2020.

The preprocessing completed for this dataset was performed during cleaning and as we ran into data concerns during the exploratory data analysis (EDA) stage. Preprocessing consisted of:

1) Remove columns that are not required.
2) Filter data to only Genus *Syringa*.
3) Filter to only keep the phenophases of interest : 'First leaf','First bloom'
4) Create a Region column based on the State field. Subset data to states based on the region of interest: Northeast.

5) Remove entries that had incorrect geolocation information: Site_ID 26116
6) Renaming variables for consistency across the dataset.
7) Filter to keep only observations from 1980-2020, to match with weather data.
8) Remove entries for plants that have multiple days listed as first leaf date and first bloom date within the same year (meaning the plant experienced a first leaf/bloom phenophase multiple times in the same year). We did not trust the validity of those data collections.
9) Removing observations that were made after August, as any first leaf occuring in August doesn't make sense and we are not investigating fall blooms (which can happen).
10) Generate an 'interval' variable that measures the time length between first leaf and first bloom. After doing this we identified several observations that had a negative interval value (meaning the observer might have switched the dates for first leaf and first bloom) or a zero interval value (meaning the observer marked first leaf and first bloom occurring on the same day). We removed those observations from our dataset as we distrusted the validity.

The distribution for the final dataset is shown on the plots below, shown in Figure 2. Further discussion can be found in our Deepnote notebook. The description of the metadata can be found in Appendix A.

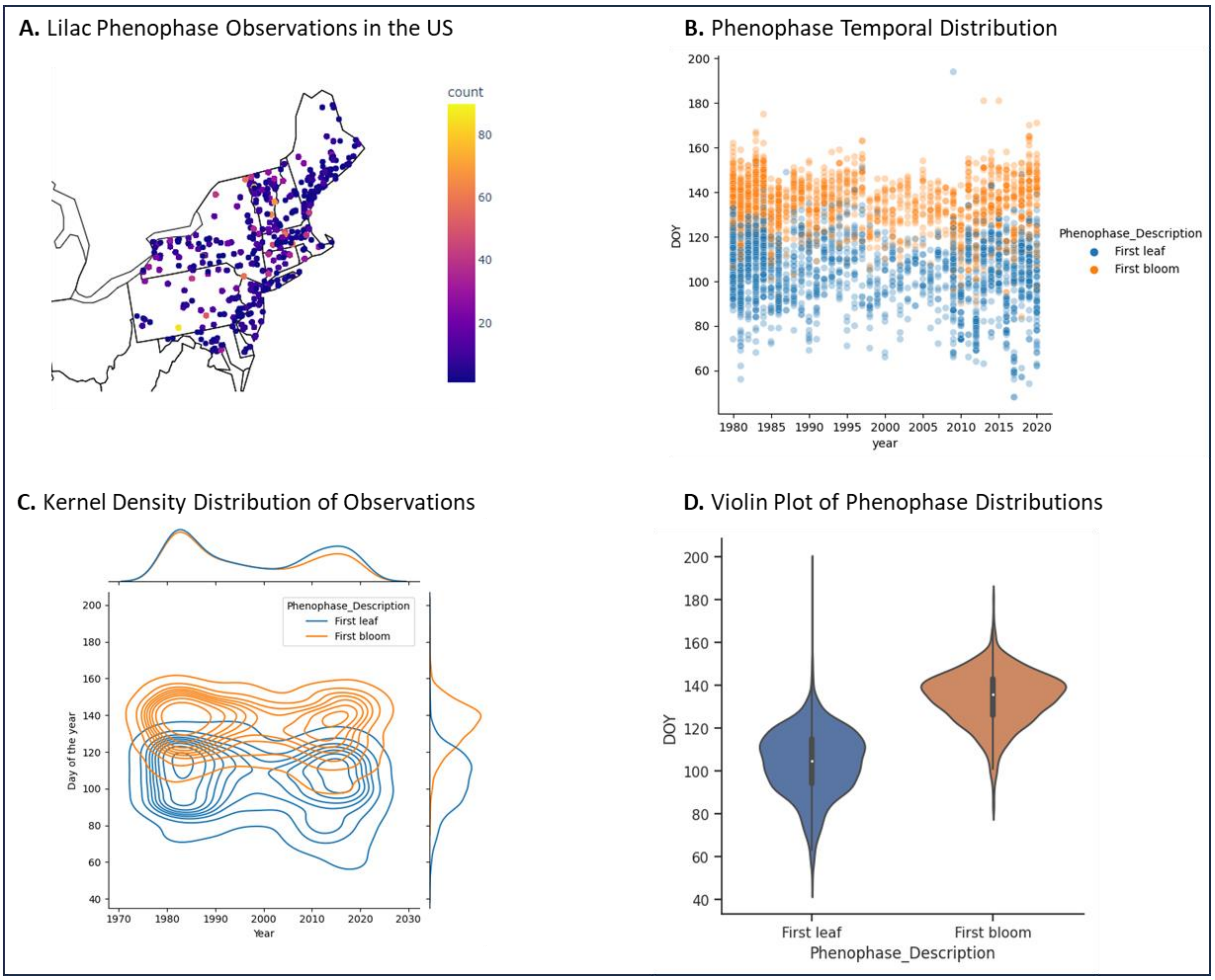

Figure 2. Geographic (A), Temporal (B, C, D), and Phenophase Type (B, C, D) Observation Distribution for *Syringa* 1980 - 2020.

Weather

This dataset contains numerical weather variables for each site location investigated in the Phenophase Dataset. The variables included are daily values for daylength, precipitation, solar radiation, snow water equivalent, maximum daily temperature, minimum daily temperature, and vapor pressure for the years 1980-2020.

Source: Daymet Daily Surface Weather and Climatological Summaries[5]

Link : https://daymet.ornl.gov/

We chose to use a gridded weather dataset to avoid the typical issues of inconsistent measurements and measuring techniques between weather station sites and the inevitability of the plant location being a far distance from the closest surface weather station. The Daymet dataset is computed from ground observations but the values are interpolated between the stations to create a full coverage of weather variables across all areas of the United States in a 1km x 1km grid.

We used an automated python script, provided by ORNL DAAC https://github.com/ornldaac/daymet-single-pixel-batch, to access and download the specific grid cells where the plant's observations were located. We downloaded and saved the python file in our shared Deepnote workspace. Following the examples, we created a text file with csv formatting, consisting of latitude, longitude, and file name (in our case, based on the site id) for the script to return daily data for all available years for that site as a single csv file. We ran the script in a terminal instance within our shared Deepnote workspace.

We joined the 412 csv files to create a single dataframe with daily weather variables for 1980-2020 of the weather variables mentioned above to create a large dataset of 6,165,580 daily observations.

Because this was a calculated gridded dataset we do not have to deal with the usual data issues found when dealing with ground weather stations. The preprocessing completed for this dataset was performed during the cleaning stage and consisted of:

1) Renaming a column to match with the phenophase dataset convention for day of year.
2) Creating a 30-day rolling average on precipitation, solar radiation, maximum temperature and minimum temperature on each individual site's data before we joined. We chose this time frame because the interval between first leaf and first bloom averaged around 30 days. We also chose to use the mean over the median as we wanted to capture some of the impact from the extremes in our data.
3) Concatenating all of the weather data from all sites into one large dataframe to join on our phenophase data.

Several descriptive statistics for the final weather dataset are shown on the plots below (Figure 3) and discussion about each can be found in our Deepnote notebook. The description of the metadata can be found in Appendix A.

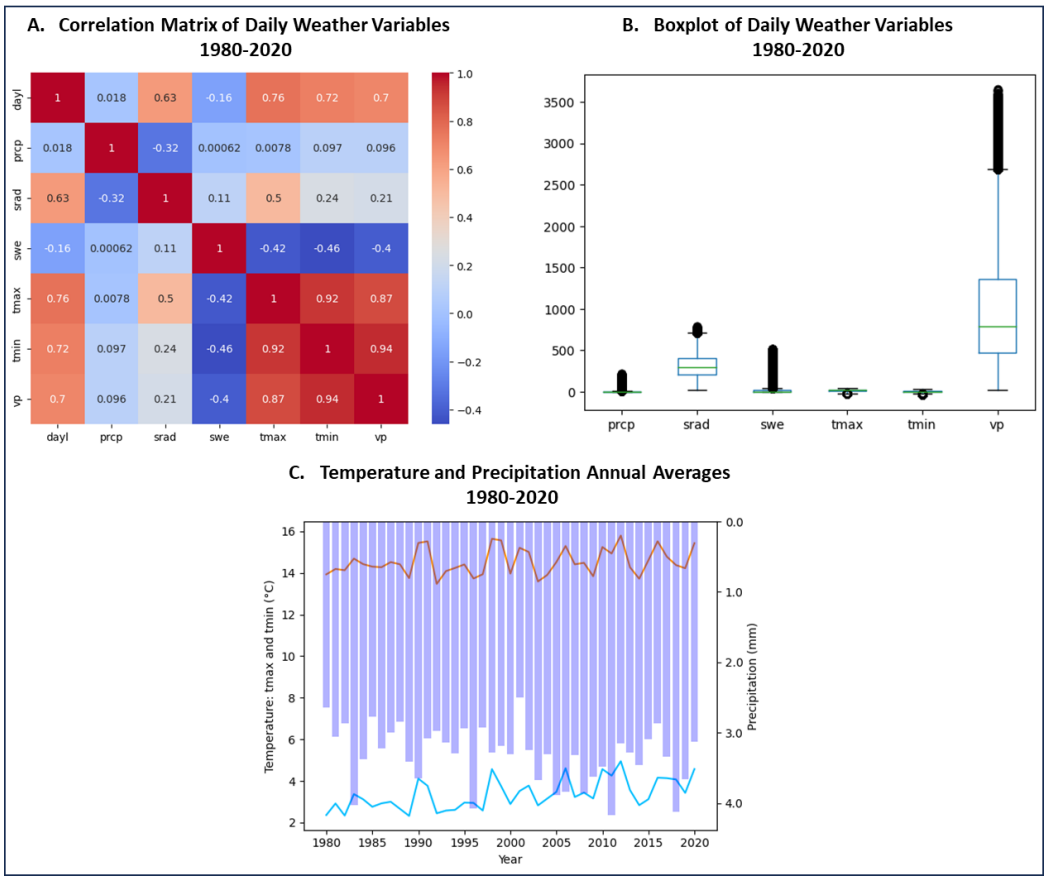

Figure 3. Weather Variables from Project Site Locations in the Northeast illustrating the correlation of variables (A), the distribution of data (B), and the annual averages and trends over time (C).

FEATURE ENGINEERING

Some of the feature engineering was described in the above Data Source section, such as defining the region, the rolling 30-day mean created to explore weather in our supervised learning methods, and the creation of the interval variable to describe the length of time between first leaf and first bloom.

To further describe the creation of the phenophase interval variable, the phenophase dataset was subset to just the features needed for determining the interval time frame and it was pivoted so that each row had both phenophase description and data for a single plant (based on Individual ID) for a single year. The interval was calculated by subtracting the day of year the first leaf occurred from the day of year the first bloom occurred.

We created two dummy variables for First Leaf and First Bloom presence using 0s and 1s. We also created another dummy variable for Phenophase presence using 0s, 1s, and 2s.

The description of the metadata for the engineered features can be found in Appendix A.

PART A SUPERVISED LEARNING

Methods Description

We first set up our notebook with the necessary packages and read in our joined dataframe that consisted of our three dummy variable columns and the averaged weather data. The features chosen to include in the model were based on subject domain knowledge. The literature discusses the importance temperature, solar radiation, and precipitation has on plant growth. These three features were also

discovered to play an important role in the first three principal components when using unsupervised methods to explore the weather data. We then split the dataset into training, validation, and test sets. The training sets were rebalanced using SMOTE because there was a large imbalance in the 0 class and the 1s/2s.

Our first model was a KNN. K-nearest Neighbors is an instance-based learning algorithm meaning that when classifying a new instance (Phenophase: 0 or 1's), it examines the 'k' training samples that are closest to the instance in the feature space and assigns a label based on a majority vote. This method is non-parametric as it does not make any assumptions about the underlying data distribution, the classification is based on the entire dataset rather than summarized parameters. To calculate the closest instance this method uses a distance parameter (Euclidean distance the most common one). In addition, the scikit implementation allows for parameter tuning on the weights assigned to the neighborhood observations, which can be either: 1) uniform 2) distance where: "closer neighbors of a query point will have a greater influence than neighbors which are further away"[6], or a callable one. We compared the model's performance using both uniform and distance and decided to use uniform weighting for our final model.

Our second model was a logistic regression. Logistic Regression is a statistical method commonly used in binary classification problems. A multinomial logistic regression can be used for classifications with more than two levels. A logistic regression models the log-odds of the probability of the event and estimates the parameters of a logistic. The coefficients represent the change in the log odds of the outcome for a one-unit change in the predictor variable. We used the implementation from scikit learn, in which we explored different solver algorithms, which vary in their application of regularization strength and type.

We then created the decision tree and random forest models. These methods use ensemble methods and "combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator"[7]. A single tree creates splits based on an evaluation criterion until the model can return a single class assignment. A random forest uses multiple trees to explore different subsets of data and will output the class that is the mode of result of the individual trees. For our implementation in scikit we used 100 estimators due to processing time and tuned the depth of the trees.

In the section for each model, we looped through multiple options for a parameter to train a model and evaluated it using the validation dataset. This allowed us to compare the effect of the parameter on model performance. We then chose which parameters we would use for the final model, created it, and evaluated it using the test dataset. Once all three models were built, we used 5-fold cross validation to compare the results of each model. Then, for our best performing model, the random forest, we created a graph on feature importance to visualize how the different weather variables affected the phenophases.

Supervised Evaluation

We used score, confusion matrices, and precision/recall/F1-score to evaluate our models' performance. For the logistic regression models for first leaf and first bloom prediction separately, we also use ROC curves as an evaluation metric.

Score was used to evaluate how accurately the model was able to predict classifications. Confusion matrices allowed us to see more detail in where our models were misclassifying and played a large role in helping us hypertune our parameters. While we still looked at precision and F1-score, we were mainly concerned with recall. A plant could only have a first leaf or first bloom once per year, so days with equally suitable weather conditions would still be given a 0. Our goal for the model was to predict if weather conditions were suitable for a given phenophase and thus felt that false positives were not an inherent failure of the model. Instead, we aimed to minimize the number of false negatives our model returned, and thus

recall was very important in evaluating our models, though precision was still taken into account when evaluating between models with similar recall.

Table 1. Average of Metric across 5-Fold Cross Validation (Standard Deviation)

Model	Score	Precision	Recall	F1 Score
KNN	0.816 (0.001)	0.352 (0.001)	0.892 (0.008)	0.335 (0.001)
Logistic Regression	0.665 (0.002)	0.342 (0.000)	0.754 (0.013)	0.283 (0.001)
Random Forest	0.813 (0.002)	0.352 (0.001)	0.900 (0.005)	0.335 (0.002)

We used the random forest model to return the feature importance. The maximum temperature was found to have the highest importance, followed by minimum temperature and solar radiation. Precipitation was found to have very little importance (See Figure 4).

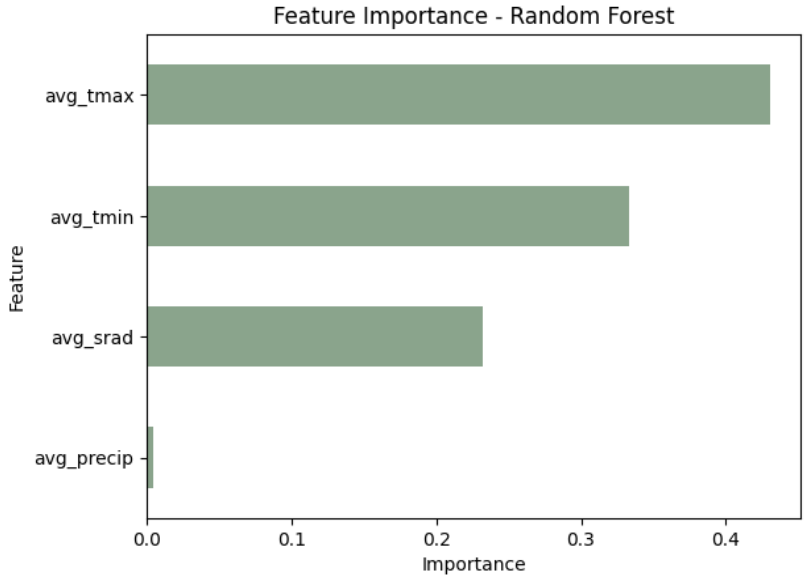

Figure 4: Feature Importance of Weather Variables in Determining Phenophase

We explored using different depths in the trees of the random forest. The model's score on the validation set was higher the higher the depth, as did precision. The recall was best around a depth of 4-6, but did decrease again at larger depths.

We chose to emphasize recall over precision due to the way our data was collected and the goal of our models, but given similar recalls did still try to take precision into account when hypertuning. In our random forest model, we chose to only use 100 estimators due to the amount of time it took for the model to train. We also limited our logistic regression to 1000 iterations due to similar concerns. With uncertainty over the effects of climate change, it will also be important to use future data to check for data shifts and update the model.

Failure Analysis

One way prediction failed is that the logistic regression model performed poorly. This may have been because the model was able to distinguish between the presence of a phenophase and the absence, but was not as capable of distinguishing which specific phenophase it was. As an improvement, we used the individual dummy variables for first leaf and first bloom to create two separate logistic regressions, which performed much better and had scores of 0.81 and 0.77, respectively.

A second way prediction failed is that a single decision tree overfits the training data because it memorizes the training data exactly. A single tree performs almost perfectly when tested against the data it was trained with, which can hurt its ability to generalize to other datasets. This was solved by using a random forest and capping the maximum depth of the tree.

Another failure in prediction is first leaf and first bloom only occur once a year so nearby days with equally suitable weather conditions may be flagged 0, resulting in many false positives. One option to help mitigate this is to include day of the year as a feature, however we left this out due to the assumption that climate change will shift phenophase starts and thus make day of year an ineffective predictor over time. If day of year were to be included, it may be necessary to further slice the data to more recent years and to regularly retrain the model.

PART B UNSUPERVISED LEARNING

Methods Description

For the Unsupervised Learning tasks we wanted to see how dimensionality reduction would work on the weather variables and also see which variables were most important in the explained variance of the result. We used Principal Component Analysis (PCA) through the sklearn library, which uses Singular-Value Decomposition to reduce the matrix of variables. It will perform a linear transformation of the original dataset to create a new approximate version of the original but with less dimensions. By doing so, it captures the variation that existed in the original data and from there one can see which features contribute to the most variance. This will allow us to see which of the seven weather variables were the more important features to this method.

We used PCA as a method of exploring the data so we did not need to split the data. The hyperparameters for PCA include the number of components to keep. We chose this optimal number by initially running all the variables and then interpreting a scree plot but also by visually viewing the cumulative explained variance for each component. We further explored our data in PCA by subsetting by season to see how the results would change.

We then wanted to use clustering techniques to see if daily weather could be clustered into specific weather typologies. K-means is a partitional clustering technique and it will categorize each weather day as belonging to a specific cluster. With K-Means the user has to declare the number of clusters the algorithm will assign the data. This was explored through several metrics. Because we do not have ground truth labels to help test the validity of the clusters, we used the Davies-Bouldin score, the Calinski-Harabasz score, and viewed the sum of squared distances for each cluster.

We looked at the hyperparameters for K-Means in sklearn. We kept the initialization method as k-means++ to help speed up the convergence. This means the initial cluster centers are set based on the probability distribution of the centroids' contribution to the overall inertia. The number of runs the k-means algorithm tries before determining the best output is set to 1, because of computing space in Deepnote and the size of our data. The maximum number of iterations for the algorithm to iterate through a single run was set to 100, also because of size and space constraints.

We chose to work with all of the weather variables that were available to us. In both workflows, the original data features needed to be scaled as a preprocessing task before running any of the methods on the data. Scaling is important to standardize all the features so that one feature does not overwhelm another

simply due to differences in the range of scale. It will also help with comparisons when variables are in different units. It is especially important to use scaled data when dealing with distance based algorithms like K-means. In the section on PCA, we explored the different scalers, StandardScaler, RobustScaler, and MinMaxScaler, on the number of principal components and cumulative explained variance.

Unsupervised Evaluation

PCA:

We began the PCA by exploring how scalers would affect the amount of variance captured by each component in PCA. This is further discussed in the Sensitivity Analysis below. StandardScaler was ultimately used and the first 3 components explained over 80% of the variance in the data. With those parameters we ran the model and retrieved the heatmap (shown in Figure 5) showing how much each feature influenced each principal component. The higher the magnitude (from the absolute value) the higher the importance. For the first component we can see that temperature, specifically maximum temperature, has the greatest magnitude. And we can also see the grouping/correlation between daylength and vapor pressure. Precipitation has the least amount of influence on the 1st PC. Second PC shows that the amount of solar radiation has the most influence and then the third component is influenced most by precipitation.

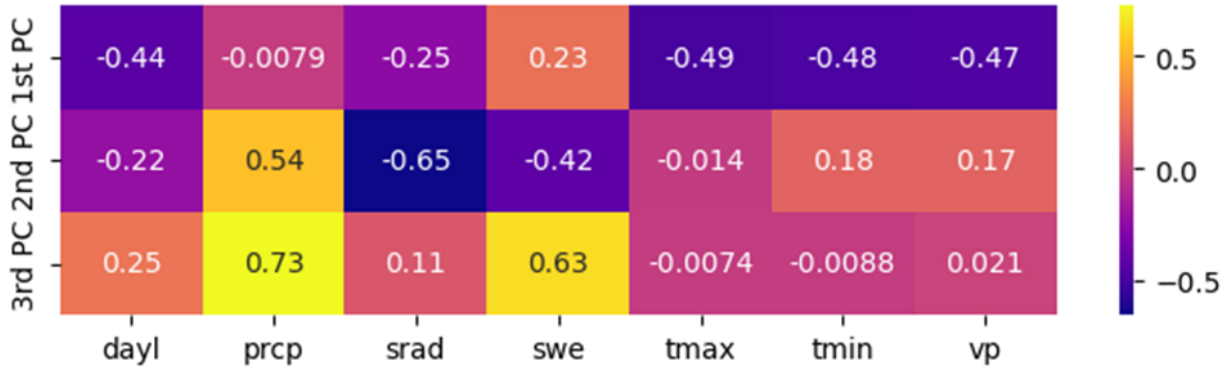

Figure 5. Heatmap of Feature Importance in Principal Component Results

We also explored the seasonal components of the weather data and ran PCA on each season. Instead of scaling the data over the whole year, we wanted to see how the principal components are created when scaled for the seasons. Temperature still had the most influence on the first principal component but whether it was maximum or minimum depended on the season. Solar radiation and daylength were most important for all the second components and precipitation, or lack thereof, was the main influencer of the third component. Snow water equivalent is important in the summer and fall maybe because the values are always zero in summer and fall? Table 2 contains the results of which features influenced which components.

Table 2: Most influential features of PCA results based on temporal subset

Principal Component	All	Winter	Spring	Summer	Fall
PC1	Tmax	Tmin	Tmin	Tmin	Tmax
PC2	Srad	Dayl	Srad	Srad	Srad
PC3	Prcp	Prcp	Swe	Swe	Swe

K-Means:

We explored the clustering of weather days using K-Means and determined the number of clusters to use via the Davies-Bouldin score and Calinksi-Harabasz score, because we did not have ground truth labels for our dataset. The best number of clusters, determined by looking at both scoring methods, was 4 clusters. The sum of squared distances metric was also used but not a clear indicator of what number of clusters would be best with our particular dataset.

We plotted the cluster centroids for each feature to understand what type of weather day is being described by each cluster. The plot is shown in Figure 6.

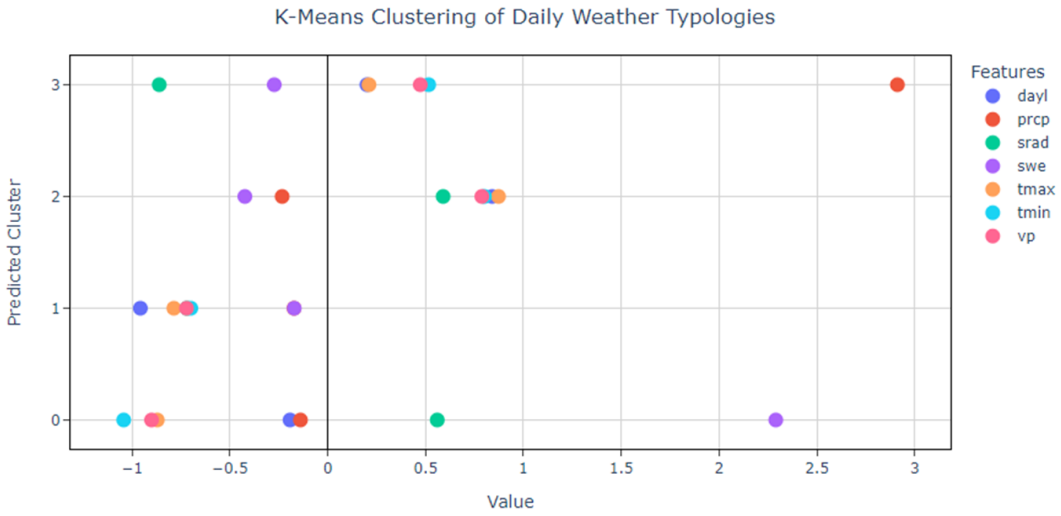

Figure 6. Cluster Centroids for Each Feature in Each K-Means Cluster

The easiest way to begin interpreting it is to look at where the amount of daylight falls within the chart. Since daylight is correlated with seasons, we have an initial starting point to see that it appears all 4 seasons were captured. This is confirmed when looking at associated temperature values. The 4 weather typologies appear to be a hot summer day (Cluster 2), a cold winter day (Cluster 1), a very cold day with snow on the ground (and value of daylight possibly suggests early spring) (Cluster 0), and a day with rain (Cluster 3).

Sensitivity Analysis

For the sensitivity analysis for an unsupervised task, we looked at the different results from scaling the data, as to how much variance is explained by each component in PCA. And we also looked at Sparse PCA.

Figure 7 below shows the individual and cumulative explained variance found by using PCA on all features but with different scalers used on the original dataset. Standard Scaler is where the mean of each feature will be rescaled to 0 and the variance will be 1. The Robust Scaler is robust to outliers as it uses the

mean and quartiles to scale. And the MinMax Scaler scales so that all feature values are between 0 and 1 and the distribution of the data stays the same.

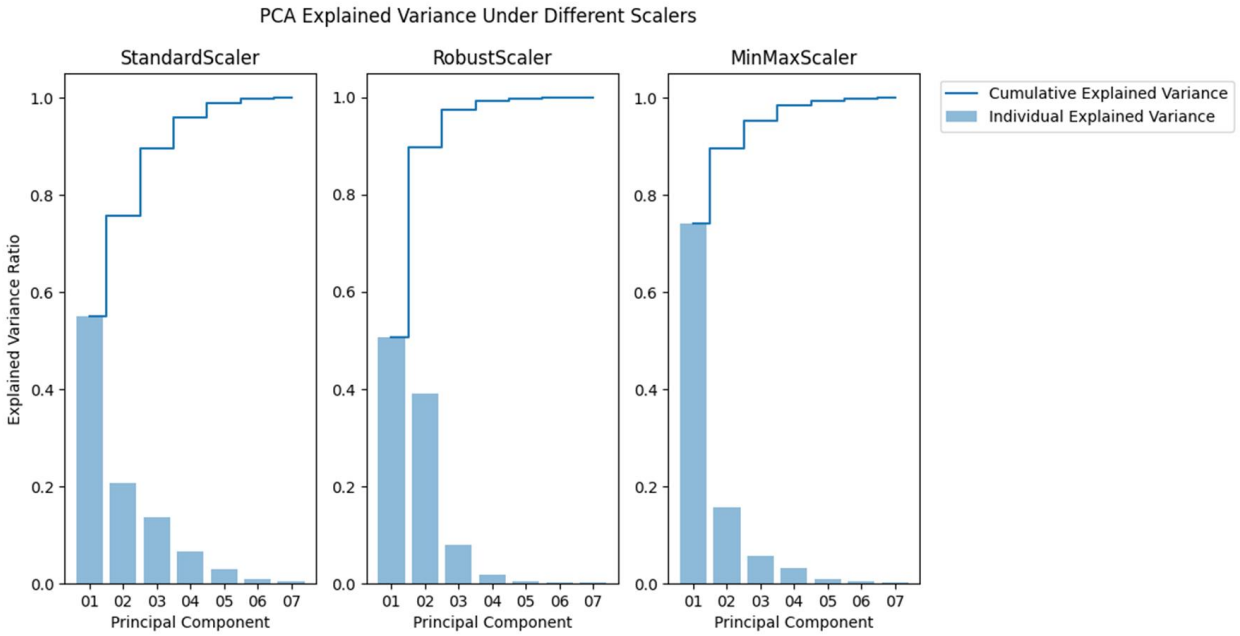

Figure 7: Individual and Cumulative PCA Explained Variance Under Different Scalers

The choice of scaler changes how the data is distributed and will change how the variance is captured by the components as shown in Table 3. The choice of StandardScaler for our analysis was chosen as it rescales the mean to zero and provides unit variance which is necessary for PCA. RobustScaler could also be used as it is more robust to the outliers. But MinMax does not meet the scaling criteria for PCA.

Table 3: Comparison and Sensitivity Analysis of Scaling Choice on PCA Variance Ratio

Principal Component	Standard Scaler	Robust Scaler	MinMax Scaler	% Change Standard & Robust	% Change Standard & MinMax
01	0.55	0.51	0.74	-8.01	34.34
02	0.21	0.39	0.16	88.79	-24.50
03	0.14	0.08	0.06	-42.26	-59.19
04	0.07	0.02	0.03	-71.83	-50.87
05	0.03	0.00	0.01	-86.25	-67.55
06	0.01	0.00	0.00	-80.14	-48.31
07	0.00	0.00	0.00	-81.84	-20.09

We briefly looked at Sparse PCA. It can be useful for clearly defining which features are influential in the components as it forces the non-influential to zero. Comparing the original PCA to the Sparse PCA, the resulting Sparse PCA heatmap (see Figure 8) do not provide more guidance than the original. Increasing the alpha parameter in the Sparse PCA implementation will push more features towards zero, however in

the few features we have we are able to see which features this would occur. Sparse PCA is best used on data where there are many more features.

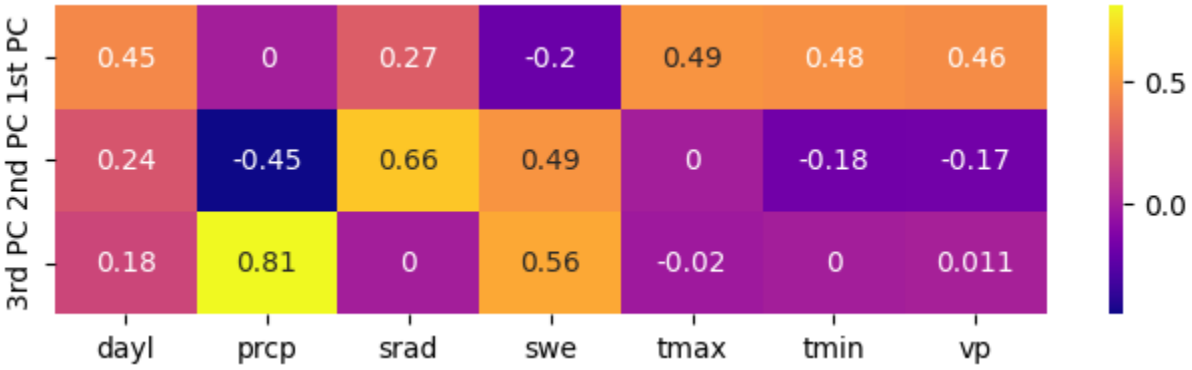

Figure 8: Heatmap of Feature Importance in Sparse Principal Component Results

DISCUSSION

Supervised Learning

We were surprised that the logistic regression had such a low score considering the KNN and random forest both performed similarly. We were also surprised that the feature analysis put very little importance on precipitation.

One challenge that we faced was the length of time it took to train a model, which was compounded by loops used for hyperparameter tuning. To combat this, we ran models while doing other things and checking back in periodically rather than trying to sit and wait until the models finished running. In the instance of the random forest model, we chose to only use 100 iterators due to the amount of time it would take for the model to run if we used more.

Another challenge was that it is hard to judge a model's precision as a large amount of it could be weather suitable for phenophases even though the original dataset did not indicate an actual occurrence on that day as previously discussed, but there is also likely a number of them that are just failures of the model. For example, weather in fall may mimic spring weather and so the model may predict a first leaf of bloom during that time even though it would be highly unlikely (although possible). Not being able to distinguish between the reasons for false positives made it hard to consider precision while evaluating our models. When deciding between two possible model parameters with equal or nearly equal recall, we then gave more weight to the model with the higher precision.

With more time, we could explore more complex models, like using more trees in our random forest. In addition, better processing resources would let us evaluate our models quicker and thus have more time for investigating alternative model options.

Unsupervised Learning

We were surprised at just how much PCA was able to tell us about our weather dataset and that subsetting by season would also give us more information. This should not have been a surprise as scaling the data based on seasonal values will change the scale and variation from the original dataset. But it was an important reminder to think about the data structure and what the data mean.

Processing time and the size of the weather dataset were initially issues. This mostly came down to the computing environment. We adjusted some parameters, such as number of iterations and initializing K-means so as to speed up convergence. We learned to read what was being held in the computing environment's memory and learned to manage resources better.

There are many ways the solutions could be extended, particularly with more time. We could look at using PCA and the first leaf, first bloom data to view the important weather components (30-day rolling mean) on that phenophase day. We computed the 30-day mean of certain weather variables and it would be interesting to see how the data would cluster and how the variance would change based on that aggregation. The difference in means, where we take the difference of the daily feature and the rolling average to look at the departure from normal, would provide an interesting comparison of K-means to look for clustering outside of the seasonal patterns. Looking at the seasonal data within K-means clustering would provide us more information about the daily weather typologies within the seasons and perhaps use that information with the first leaf and first bloom over the years. We could also explore the geographical distribution and timing of weather variables as well.

ETHICAL CONSIDERATIONS

The phenological data does contain potentially identifiable information in the form of latitude and longitude. As the observation program is mostly completed by volunteers, the coordinates may be that of places of personal property. Users and contributors did have to sign an agreement when submitting information to the database that the data could be freely shared to all.

The information gained through supervised learning does not contain the coordinates and is an aggregation of the data. There are no perceived ethical issues in sharing the models and results of how weather impacts the prediction of a phenophase date.

The weather data never contained personal identifiable information. The information gained through the unsupervised learning methods, such as the principal components and clusters of weather days, do not contain information that could be, or lead to, an ethical issue.

STATEMENT OF WORK

- Project Proposal: All
- Project Management: Andrea
- Data Download: Andrea, Linda
- Data Cleaning: All
- EDA: Christian, Linda
- Supervised Learning: Andrea, Christian
- Phenophase Interval: Christian
- Unsupervised Learning: Linda
- Final Report: All

We were all communicative and available throughout the project. We all shared ideas and offered constructive feedback. We were able to accomplish more together than if we had worked separately.

REFERENCES

1. Newlands, A, Sylvester, L, Shulack, J. Fall 2022. Impacts of Climate Extremes on Vegetation. SIADS 593 Milestone I Project. https://docs.google.com/presentation/d/1KapxsH3-BioSDXi60knoKHEpZM4Muk2pixOM1frSBn8/edit?usp=sharing
2. Taylor, S D, and White, E P. 2020. Automated data-intensive forecasting of plant phenology throughout the United States. Ecological Applications 30(1):e02025. 10.1002/eap.2025
3. Katal N, Rzanny M, Mäder P and Wäldchen J. 2022. Deep Learning in Plant Phenological Research: A Systematic Literature Review. Front. Plant Sci. 13:805738. doi: 10.3389/fpls.2022.805738
4. a) Caprio, J.M., Schwartz M.D. and the USA National Phenology Network. 2013. Lilac and Honeysuckle Data for the United States, 1956-2013. Tucson, Arizona, USA: USA-NPN. Dataset accessed 2022-10-4 at http://dx.doi.org/10.5066/F78S4N1V
   b)USA National Phenology Network. 2022. Plant and Animal Phenology Data. Data type: Individual Phenometrics. 1950-2022 for Region: Eastern United States. USA-NPN, Tucson, Arizona, USA. Data set accessed 2022-10-4 at http://doi.org/10.5066/F78S4N1V
5. Thornton, M.M., R. Shrestha, Y. Wei, P.E. Thornton, S-C. Kao, and B.E. Wilson. 2022. Daymet: Daily Surface Weather Data on a 1-km Grid for North America, Version 4 R1. ORNL DAAC, Oak Ridge, Tennessee, USA. https://doi.org/10.3334/ORNLDAAC/2129
6. K-Neighbors Classifiers." Scikit-learn, scikit-learn developers, 2023-10-23, .https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier.
7. Ensemble methods." Scikit-learn, scikit-learn developers, 2023-10-23, https://scikit-learn.org/stable/modules/ensemble.html#random-forests

   Referenced throughout the project:
   MADS Coursera Coursework

APPENDIX A

Metadata

1) Phenophase Dataframe
   Source: USA NPN. Individual Phenometrics Datafield Descriptions.
   https://data.usanpn.org/observations and then click on Metadata

Column Name	Units	Description	Type
Site_ID	(No units)	The unique identifier of the site at which the series was recorded.	Numerical
Latitude/ Longitude	(Degree)	The latitude / longitude of the site at which the series was recorded. Calculated from the Google Maps API with a datum of WGS84 (https: //developers.google.com/maps), unless a plausible user-defined lat/long was submitted. Information about the datum and source of the lat/long value can be found in the "Site" ancillary data file.	Numerical
State	(No units)	The U.S. state or territory, Mexican state or Canadian province in which the site is located. The state is calculated from lat/long. A value of "-9999" indicates the site does not fall within the boundaries of North America.	Nominal
Genus	(No units)	The taxonomic genus of the organism for which the series was recorded. Taxonomy follows that in the Integrated Taxonomic Information System (http: //itis.gov).	Nominal
Individual_ID	(No units)	The unique identifier of the individual plant or the animal species at a site for which the series was recorded.	Numerical
Phenophase_Description	(No units)	The descriptive title of the phenophase for which the series was recorded.	Text
Column Name	**Units**	**Description**	**Type**

Year	(No units)	The year of the first "yes" record of the series.	Numerical
First_Yes_Month	(No units)	The month of year, ranging from 1 to 12, of the first "yes" record of the series.	Numerical
First_Yes_Day	(No units)	The day of the month of the first "yes" record of the series.	Numerical
DOY	(No units)	The day of year, ranging from 1 to 366, of the last "yes" record of the series	Numerical

2) Weather Dataframe
Source: ORNL Daymet. https://daymet.ornl.gov/single-pixel-tool-guide

Column Name	**Units**	**Description**	**Type**
Year	(no units)	Year, repeated for each day in the year.	Numerical
Yday	(no units)	Integer representing day of year, values ranging from 1-365. NOTE: All Daymet years are 1 – 365 days, including leap years. The Daymet database includes leap-days. Values for December 31 are discarded from leap years to maintain a 365-day year so yday 365 = December 31 for non-leap years or December 30 for leap-years.	Numerical
Dayl	(s/day)	Duration of the *daylight period* for the day. This calculation is based on the period of the day during which the sun is above a hypothetical flat horizon.	Numerical
Column Name	**Units**	Description	Type
Prcp	(mm/day)	Daily total precipitation, sum of *all forms* converted to water-equivalent.	Numerical

Srad	(W/m^2)	Incident shortwave radiation flux density, taken as an average over the daylight period of the day. NOTE: Daily Total Radiation (MJ/m^2/day) can be calculated: ((srad (W/m^2) * dayl (s/day)) / l,000,000)	Numerical
Swe	(kg/m^2)	Snow water equivalent. The amount of water contained within the snowpack.	Numerical
Tmax	(deg c)	Daily maximum 2-meter air temperature.	Numerical
Tmin	(deg c)	Daily minimum 2-meter air temperature.	Numerical
Vp	(Pa)	Water Vapor Pressure (in pascals). Daily average partial pressure of water vapor. (can derive mean relative humidity, mean absolute humidity, and mean heat index using vp)	Numerical

3) Engineered Data Features

Column Name	Units	Description	Type
Region	(No units)	Determined by State. northeast = ['ME', 'VT', 'NH', 'MA', 'RI', 'CT', 'NJ', 'DE', 'MD', 'PA', 'NY'] southeast = ['VA, NC, SC, GA, FL, AL']	Nominal
Interval	(day)	Number of days between the first leaf and first bloom.	Numerical
Avg_[precip, tmax, tmin, srad]	See units for specific feature in weather data	Average value based on the past 30 days.	Numerical
First_leaf	(dummy variable)	0 or 1 indicating presence of First Leaf phenophase	Nominal

First_bloom	(dummy variable)	0 or 1 indicating presence of First Bloom phenophase	Nominal
Phenophase	(dummy variable)	0 if no phenophase presence, 1 if First Leaf, 2 if First Bloom	Nominal