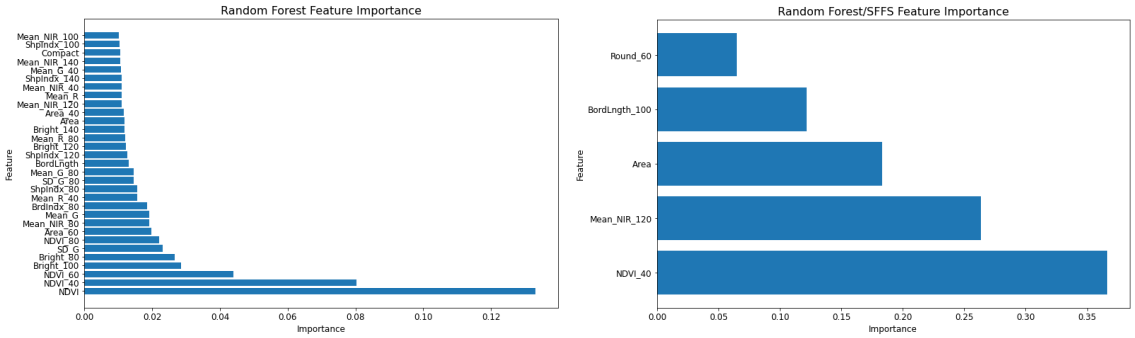
Angyalka Valcsics
Angyalka.Valcsics@maine.edu

Urban Land Cover

The goal for me with this data was to perform classification and feature selection. The response variable 'class' is a categorical variable which denotes some type of land cover such as a building, tree, or shadow. There are 147 other variables describing each land cover type. These variables are either of type size, shape, spectral, or texture. See that there are a low number, 14-29, of training samples for each class. Additionally these are 168 instances of training data and 507 for test. For these reasons, feature selection is a good idea.

I opted to fit a random forest classifier with the training data, tuned the hyper-parameters using cross-validation and the *GridSearchCV* function, then used this model to classify land cover in the test data. The tuned random forest model achieved an 80.7% accuracy in classifying the test data. The random forest classifier uses the Gini Index to return a feature importance value for each feature in the data, we can then sort these from most to least important variable. The variables we are given repeat on a coarser scale. Since variables which are correlated in a random forest are given the same or similar feature importance, we need a way to pick only the best of each of these. I decided to use the parameters of the tuned random forest model in sequential forward feature selection. My idea is that a variable which is correlated with an existing variable in the model will not be selected since it wouldn't be adding anything new to the prediction power. The biggest issue I ran into with using SFFS is that the training time was significant. The function takes in a variable which specifies how many features to pick for the final model. Although I did spend time experimenting with this, I had hoped to perfectly tune the number of features, but training time got in the way. The SFFS random forest model achieved a 74.4% test accuracy. Both models achieved a 100% training accuracy.



You can see that each model deems similar variables to be important. Of course all these variables are working together but I will offer some interpretation of the variables here. The most important feature, NDVI40 assesses whether or not the target being observed contains live green vegetation. Recall that two of the class variables are tree and grass. I would assume that the variables Area and BorderLength would do especially well with buildings and cars. The variable Round40 measures roundness–recall pool is a class. Even after research I'm not sure I understand what sort of information a near infrared sensor captures. It seems to measure (mean) distance to target surfaces. Perhaps this can help classify the flat land cover classes like pool and concrete.

Next, let's test the hypothesis

$$H_0 : \mu_0 = \cdots = \mu_8 \text{ versus } H_1 : \text{ at least one is different}$$

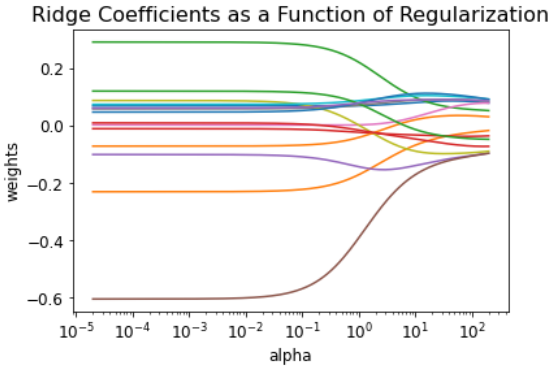
based on our most important variable according to the model, NDVI40. The data violates at least one of the assumptions necessary for using the F statistic–equal variances across the different groups. Hence, it was necessary to perform the hypothesis test using a bootstrap procedure. Since the p-value was 0, we have significant evidence against the null hypothesis and in support of the alternative hypothesis that at least two population means are different across groups. **Pollution and Mortality Rate**

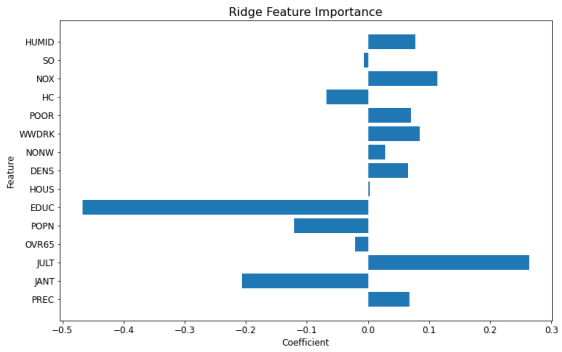
I read the pollution text file in line by line, stored it into a data frame, then converted each entry from an object to a numeric. My initial thought was that since we have yearly data, there may be some non-linear

Angyalka Valcsics
Angyalka.Valcsics@maine.edu

terms. Additionally, accurate prediction of mortality rate based off these variables alone would be valuable. There are two approaches for this. If we want to learn about the variables then linear regression with a Ridge penalty would work well. If we just want to achieve high prediction accuracy and not concern ourselves too much with interpretation then we should use a neural network. After scaling the data and splitting into train and test sets, this is precisely what I did.

First, I fit a Ridge regression using *GridSearchCV*. The search method revealed an optimal alpha of 0.2122 and optimal solver of *lsqr*. With Ridge, we can interpret variables with a higher relative coefficient as being more influential on the response. Please see below the plot of the optimal model's coefficients as well as a plot of the model's coefficients relative to the regularization parameter alpha.





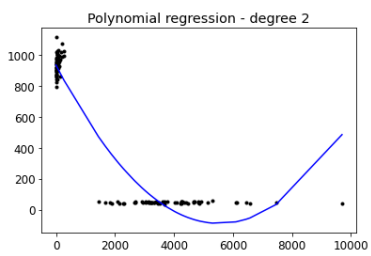
The average temperature in Jan, July, and education have the most influence on mortality rate. Climate such as heat waves, cold, and heavy rain have all been proven to cause accidents or result in more deaths. Humidity can help viruses spread or exacerbate effects of climate. Moreover, the societal variables shown to be important in this model have been proven to have an affect on mortality as well. Lastly, Nitric oxide is colourless and is oxidised in the atmosphere to form nitrogen dioxide which is an acidic and highly corrosive gas that can affect mortality. This model had a training MSE of 0.0075 and a test MSE of 0.00596.

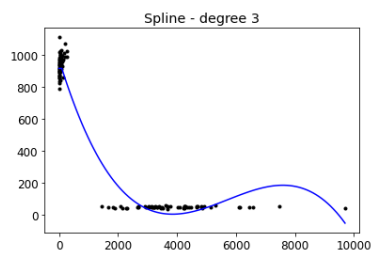
Recall that the loss function minimized in ridge regression is

$$\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

Where the term farthest to the right is our penalty term and λ is a tuning parameter which the user has control over. This model optimizes the coefficients of the linear regression similarly to OLS, however the penalty term regularizes the coefficients of the model. This regularization term restricts the coefficients so that $\sum_{j=1}^{p}\beta_j^2 \leq s$ where s coincides with our choice for λ.

During my research of these variables, I read that sulfur dioxide has a non-linear relationship with mortality. I did manual cross validation and fit a polynomial regression with the the SO variable and mortality rate. Through cross validation, I found the optimal degree to be 2. This method had an MSE of 13964, a high value was to be expected. The data has quite a large gap. I wasn't very satisfied with the plot so I also fit a spline of degree 3. Please see the plots below.



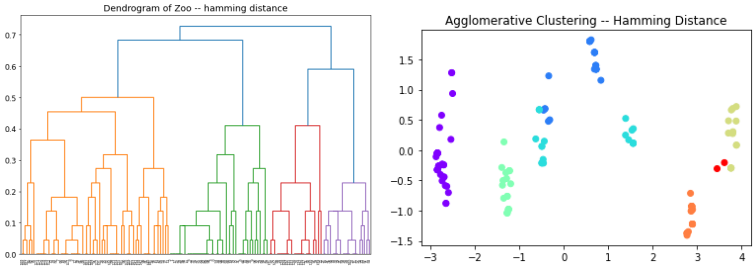


Angyalka Valcsics
Angyalka.Valcsics@maine.edu

Lastly, I fit a neural network on this data to compare with the ridge regression. I used the grid search function once again to find the optimal hyperparameters. I found that the optimal parameters were logistic activation, alpha of 1e-05, two hidden layers with 20 and 10 nodes, and a learning rate of 0.001. This produced a training MSE of 0.009 and a test MSE of 0.008. This did not perform better than Ridge regression in terms of error and it lacks interpretation of the model. **Zoology**

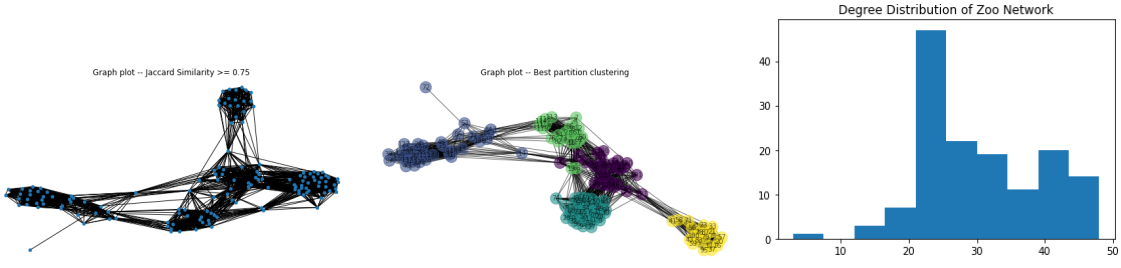
For the Zoo data I wanted to apply some type of network analysis or clustering, or both. I began with one-hot encoding the legs variable so that the data matrix was purely ones and zeros. Next, I focused on finding a similarity and distance matrix. My goal was to avoid using Euclidean distance on the binary vectors. After some research, I found that perhaps the Jaccard coefficient would work well for this. For each entry, the value is 1 if the animal has the quality we are interested in and 0 if not–i.e. positive/negative states. This means that our binary variables are asymmetric attributes, I would argue that we only care if the animal has the quality that we are interested in. We can then convert this to a matrix of Jaccard similarity or Hamming pairwise distance matrix. The Hamming distance between two binary vectors is the number of elements that are not equal.

First I used the hamming distance matrix to plot a dendrogram, which looks like it identified 6 or more clusters. Please see the plot below. I then fed the hamming distance matrix into the Hierarchical Agglomerative Clustering Algorithm provided by Sklearn. For both the dendrogram and hierarchical clustering, I used complete linkage. I believe this worked fairly well because at each step of the algorithm, the two clusters which are separated by the shortest Hamming distance (those with the most equal elements) are combined. Using PCA, I reduced the dimensionality of the data to the first 2 components then colored this according to the model's assigned cluster label.



This method had an MSE of 1.8194 and a test accuracy of 51.4%. By looking at a confusion matrix we can see that the model did a perfect job on classes 0 and 5. It grouped all of the 1st and 3rd class together but with the wrong label. I also used kmeans on the two PCA components from the one hot data matrix and this seems to work quite well although I question the validity of using Euclidean distance.

Now, to construct a network given this data I used Jaccard similarity. According to some cut off value, I entered a 1 in the adjacency matrix and a 0 otherwise. Below you will see a graph plot of this network as well as an attempt at clustering the network plot using the partition of the graph nodes which maximizes the modularity. Additionally, you will see a plot of the degree distribution for the network.



Lastly, I found that there are 15801 triangles in the data. This means there are significant amount of direct links (similarity) between 3 of the animals in the data.