**ATOC5860 – Application Lab #6**

**Machine Learning with Weather Data**

**Spring 2022**

**Notebook #1**

**Questions to guide your analysis of Notebook #1:**

**1) Start with 4 clusters. Cluster the data at 17 UTC (mid-day in Colorado). What is the seasonal occurrence of the 4 clusters? Do the 4 clusters correspond to Fall, Winter, Spring, and Summer? Why or why not?**

Chart, scatter chart

Description automatically generated

I would say cluster 3 corresponds to Summer pretty well but the other 3 clusters don’t really fit well into the other seasons. From these clusters, it seems like it would be easy to use temperature at this time of day to tell if you are in Summer or another time of year but it would be hard to specify what the other part of the year is.

**2) Based on 2D and 3D scatter plots of the cluster centers and the data – Which weather variables help (or NOT help) define the clusters?**

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

From all of the plots above, it looks like the wind speed and temperature do a decent job at separating out three clusters and the fourth cluster is kind of in between all of the other ones. For the 2D plot with relative humidity and wind direction, the gray and blue clusters are a little distinctive but not super. As for the 3D plot with wind direction, temperature, and relative humidity, I think it does well at isolating the gray cluster but not the other ones.

**3) What do the clusters show during the time period from September 5-15, 2020 (Labor Day 2020)? Are the cluster assignments consistent with the weather experienced over that time period? Are there other date ranges that you would like to check out?**

From the first plot in this document, we can see a gap in the data in the summer cluster for day 249-254 of 2020. These dots are shown in purple and appear in the cluster that looks the most like the winter instead of the summer one. This makes sense because this is when we had an extreme temperature change from warm summer temperatures to really cold winter temperatures and snow.

**4) Re-run the analysis. But now use three clusters instead of four clusters. Compare your cluster analyses for 4 clusters and 3 clusters. Do the results for 4 clusters or 3 clusters make more sense to you based on your analysis and also your experience living in Boulder, Colorado? Which number of clusters provides a better fit to the data?**

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

From all of the plots above, I think 3 clusters makes more sense than 4. In the first plot, you can see a very distinctive summer cluster and a distinctive winter-ish cluster. In the plot with wind speed and temperature, the three clusters are pretty distinct. They are slightly less distinct in the plot with relative humidity and wind direction. The 3D plot looks like the red and blue are pretty distinct and the gray is not very distinctive.

When you try just two clusters, you get:

Chart

Description automatically generated

This does not do a great job fitting the data. Thus, I think 3 clusters is the best choice for this dataset.

**Notebook #2**

**Questions to guide your analysis of Notebook #2 – See also questions at the end of supervised.ipynb:**

**1) Which machine learning model performs the best to predict rainfall? What metrics did you use to make this assessment?**

Table

Description automatically generated

From the table above, it looks like the Neural Network model has the highest recall, which represents the proportion of precipitating hours that are correctly predicted by the model, thus I think this model does the best at predicting rainfall. However, the Singular Vector Machine also has a high recall and an even higher prediction example so I would day this is the second best model for predicting rainfall with this dataset.

**2) Describe the difference between accuracy and recall. Why did we choose to use accuracy, recall, and predicted precipitation probability as a way to compare models? In forecasting: when is a false positive (you said it would rain, it didn’t rain) preferred over a false negative (you said it wouldn’t rain, it did rain)?**

In this case, accuracy is the proportion of precipitating hours or non-precipitating hours that are correctly predicted by the model. Recall is the proportion of precipitating hours that are correctly predicted by the model. Thus, the accuracy assesses whether or not the model prediction is correct, regardless of the specifics of the prediction whereas recall is the model’s ability to correctly predict the desired event (rainfall in this case). We chose accuracy, recall, and predicted precipitation probability to compare models because they tell us how well the model identifies precipitation events and non-precipitation events. Thus, they give us an indication of how well the model is performing in general.

**3) One important "gotcha" in a machine learning workflow or pipeline is the order of data preparation. Why should one should perform the train-test split before feature scaling and rebalancing? *Hint: think about using a trained model for future predictions.* Do you want your scaling of the testing data to depend on the training data? Why perform a test-train split at all?**

It is important to perform the test-train split before feature scaling and rebalancing so that you do not accidentally seed information about the testing data into the training data or vice versa. This could give the model information about the testing data that may cause it to not learn as robustly and would make it more difficult to test the model’s performance since it may be using information about the testing data that was accidentally included in the training to perform well. Conversely, having information about the training data in the testing data might also skew the model’s performance because then it is indirectly being tested on data that it was already trained on. This would make it difficult to assess the robustness of the model.

**4) Collinearity, or non-zero correlation among features, results in a model that is overly complex, reduces the statistical significance of the fit of the model, and prevents one from correctly identifying the importance of features. *Are there features included in our machine learning models to predict rain in the Christman dataset that are collinear?* If so, how do you think we should address this collinearity? A couple of suggestions: If we don't have that many features, we could use our meteorological expertise to simply remove one of the features that shares collinearity with other features. Another way to address collinearity is to use feature regularization, or add weights that penalize features that add noise, ultimately reducing model complexity.**

For our datasets, I think the temperature, dewpoint temperature, and relative humidity all contain similar information. Also, the temperature and pressure have similar information so one could argue that one of these variables could be removed, such as the temperature.

I think it would be very interesting to see how removing one feature that is well-correlated with another variable would impact the model’s performance. I think this decision depends on what your goal is with the model you are developing because this will determine if you want to give it as much information and variables as possible or if you want to only keep as few as needed to retain all the necessary information.