Sheryl Williams

DATA 71200 - White Paper

I chose the Electric Consumption data set from NYC OpenData provide by the New York City Housing Authority. The dataset provides monthly consumption and cost data by borough and development. I chose this dataset purely out of interest and curiosity as it didn't occur to me that this type of dataset would exist, and I was originally going to go with a NYC 311 dataset. The dataset included columns such as: Borough, Location, Vendor Name, Meter Number, Current Charges, and Consumption (KWH). I wanted to predict two possible scenarios: charges based on consumption usage and consumption usage/charge based on the borough.

During the data cleaning stage, I used the describe() function to perform an exploratory data analysis for values such as the standard deviation, the mean, minimum, and maximum values. I decided to drop the following values: those below 0 in the Current Charges dimension, any values that are equal to 0, and rows that had a missing value. Additionally, I ran the corr() function on the Consumption (KWH) dimension and dropped many of the columns that scored less than a 0.90 correlation. After that point, I chose to work with Borough, Current Charges, Consumption (KWH), and Charges (KWH). Furthermore, I had renamed the columns to a lower-case version—for precautionary measures. In the projects, I found that most of my challenges occurred during the machine learning stages of Project 2 and Project 3. Therefore, I found myself going back to the initial data cleaning stage and the training/testing split stage to adjust. Such as, dropping values outside the lower and upper limits of the standard deviation and values whose frequencies are less than a certain amount.

Another example occurred during Project 2, I received errors that the shapes of the datasets weren't equal. Another issue I ran into during Project 3 was the following error: "too many indices for array." Since I looked at multiple columns for the training set, I figured that these errors were a result of not reshaping the data. However, I received the following error: "list index out of range." At that point, I had removed the array element references to bypass the errors. I wasn't sure how to fix this error; however, programming with the breast_cancer dataset did provide some insight into the structure of the dataset. Especially, with how the columns are broken into parts of the code, e.g., the training set and the visual graphs.

Visualizing the data with the dataset helped tremendously with helping learn about the available data and identifying patterns. For example, data binning with histograms showed that consumption has a slight Gaussian distribution, but it is skewed to the left. While the Charges variables have an exponential distribution. Another example is in Project 2, using a scatter plot to visualize the relationship between the variables for consumption and charges revealed the outliers and the strong linear relationship in the dataset. The scatter matrix plot showed the outliers in the variables; however, I think a box-plot visualization would have done a better job at visualizing the extent of the outliers for the variables.

Additionally, from project 2, the line plot for the training and testing accuracy of k-Nearest Neighbors helped determined the best number of neighbors for the kNN model. By using the line plot alone, I would have assumed that 1 neighbor would be the best result for the model. However, I decided to play around with other n-neighbors such as 3 and 5. I thought about the potential of K=1 having high variance and proneness to errors, despite the high accuracy scores that were both calculated and visualized. Moreover, I had concerns of

overfitting at K=1, due to the high training score and low test accuracy. However, this didn't seem to be much of a case at K=3 and K=5, despite the low accuracy scores.

In Project 3, unfortunately, I can't speak much to the visuals on the electric consumption dataset—due to the errors I had mentioned previously. Although, on the breast_cancer dataset, I found it helpful in determining the best features to use in the models. For example, the matrix color bar graph for feature selection. While I can't speak for this improving the electric consumption dataset (I had more success visualizing this with the breast_cancer dataset), I can speak for the times I have used a PCA analysis in SAS and I see the power in identifying the most useful features in the dataset.

For supervised learning, I chose the k-Nearest Neighbors algorithm and Decision Trees. The former algorithm I picked because of its usage in both classification and regression analysis. Additionally, for classifying datapoint in terms of its similar measurements to another datapoint. This method allows for parameter tuning when picking the value of K. For the latter algorithm, I picked this because it can handle multi-dimensional data and can also be used for both classification and regression datasets. Furthermore, this algorithm divides the dataset into smaller datasets into a series of splitting into leaves and nodes. Using both the algorithms, I believe the algorithm performed better under the Decision Tree analysis, mainly because of the high variance in the kNN algorithm.

Looking at the algorithms separately, for the kNN algorithm, at K=1, there is a high difference between the training (0.99) and testing (0.39) accuracy score compared to when the value(s) is at K>1. I decided that the kNN model is strongest at K=3, so I ran that with the cross-validation algorithm and received low scores at around 0.30. Therefore, I don't think the data

used in this model is properly represented—possibly due to the variance at K=1 neighbors. For the Decision Tree algorithm, I used both the DecisionTreeClassifier and DecisionTreeRegressor and found that the classification model did better. This was also the same case for the Random Tree algorithms. Considering that for Project 2, my X training set focused on the Borough dimension as a one-hot encoding—I wonder how this would play out if I had manipulated the values as an ordinal column.

Another thing I found interesting for the Decision Tree algorithm is that higher values for max_depth improved the classification models. Coincidentally, for the regression models, the testing score decreased. Changing the n_estimators and random_state did not have much of an effect on the dataset. On another note, when I performed the Naive Bayes and Support Vector Machine algorithms—this was due to array and indices errors that were mentioned previously. Moreover, I didn't want to do a Linear Model algorithm as I knew from Project 1, the model performs well (over 0.90 accuracy score).

As I mentioned previously, I can't say if the principal component analysis (PCA) for feature selection improved my models from my supervised learning project due to errors that I ran into—I don't believe I scaled my data properly, among other previously discussed errors. Given the electric consumption dataset, the PCA method reduces the dimension of the large dataset to make it simpler and easier to explore. The reduced dataset contains most of the important information of the original dataset.

As for the errors, a prominent one is "list index out of range" which meant I was access an index value inside a Python list that does not exist. This occurred when I attempted to use the mglearn.discrete_scatter() function. After attempting to correct this error by adding a mask

(per the error message), I then received the following error: “index 1 is out of bounds for axis 1 with size 1” which I believe is due to improper scaling of my X and y data splits.

In terms of the PCA results for the breast_cancer dataset, three different algorithms were looked out: k-means clustering, hierarchical clustering, and DBSCAN clustering. Given the number of features (30 of them) in the dataset, I think performing a PCA analysis is appropriate to perform especially when one of its strengths is a reduction in computational costs. Furthermore, the 95% of variance steps helped reinforced this matter for achieving reduced dimensionality. While performing this method, I received a 0.95 cumulative sum of the explained variance. For the scatter plot for the first and second principal component analysis, the features that signified a malignant result tend to score farther right of the first principal component indicating a stronger importance compared to the features that signified a benign result.

In k-Means clustering, the algorithm groups similar items in the form of clusters. For the algorithm, there are 3 distinct cluster groups that are divided up amount Feature 0 and Feature 1. As such, next is to pick the number of clusters that will be significant for the algorithm. The plot appears to be an elbow curve that decreases sharply from 1 to 2 and plateaus out after n=5. Therefore, the ideal cluster number should be between 2 and 5 to group similar features.

In hierarchical clustering, this method groups unlabeled points and groups them together based on similar characteristics. This is then broken down in a dendrogram. With the cluster of datapoint shown in the plotted dendrogram, the plotted distance is shown indicating the total variance between clusters. In the case from Project 3, the highest dimensionality is 2, then 3, then 4, then so on and so forth. Moreover, the features are grouped together by

similarities and selecting feature that would help improve models. Moreover, I see this method as a way of improving k-Means and k-Nearest Neighbors algorithms.

Now for DBSCAN clustering, this technique is a density-based clustering that determines the distance between different values based on their shapes and sizes. From the results, it doesn't look like this method fared well due to the high number of dimensions in the dataset and there does not appear to be any discernible clusters. It's possible that the breast_cancer dataset is not dense enough a DBSCAN analysis. This is the same when the data is scaled; however, there is a dark blue group close to Feature 0. I'm presuming that these are possible outliers. Adjusting the min_samples and eps values had no effects on the algorithm. Moreover, when comparing the algorithms used for this dataset, it appears that DBSCAN is not an appropriate due to the calculated adjusted Rand Index of 0—indicating our randomness in the selection independent of the samples and clusters. This is opposed to the 0.50 possible matches in k-Means and hierarchical clustering.

Personally, I think that the electric consumption dataset is appropriate for a PCA analysis. I thought relabeling the Borough variable to a one-hot encoding variable would help with the analysis; however, this resulted in an error referencing the need for a 1D array. Therefore, I recategorized it as an ordinal variable, with minor success, but not enough to gather information and decide from the visual graphs. As I mentioned previously, I believe I need to rescale and reshape the data properly for it to work.

Overall, I believe I should focus more on data pre-processing and clean up. Honestly, I underestimated that step and feel that majority of the errors I ran into is due to my approaches in organizing my data. Plus, the lack of scaling my data at different points of the projects. As

such, working on these three projects provided hindsight into the importance of this stage before going into the models. Additionally, I'm planning to work on manipulating numpy and pandas data in Python—I have a feeling I mixed the code for those two concepts throughout these projects.

Another thing, I would probably play around with the size of my dataset. The electric consumption dataset has over 300k row values. While, I didn't have issues due to the size, the Agglomerative/Hierarchical model crashed the kernel every time I tried to run it. No errors were associated with it and given the success of the algorithm running with the breast_cancer dataset, I would try this with a smaller dataset next time. And perhaps next time, I'll have a better answer to which model predicts electric consumption better—especially which variables or dimensions would help predict the model.