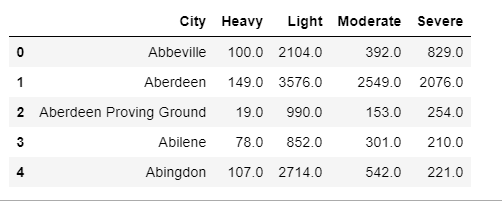
**Data Science II: Second Milstone**

**Data set: US Weather Events**

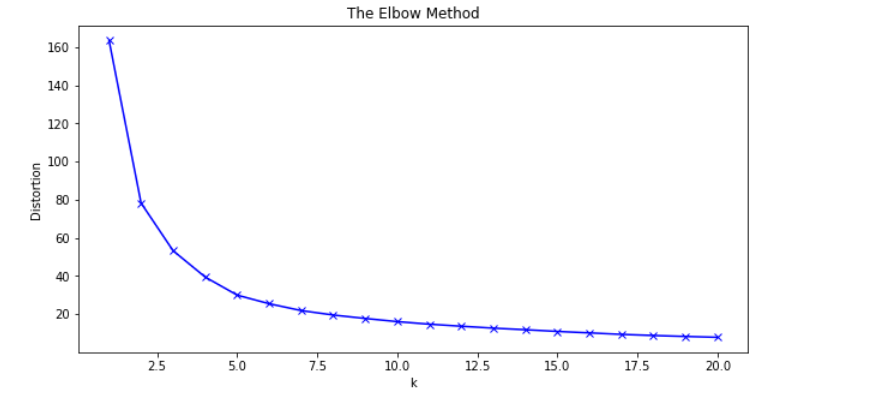
**Modeling algorithm utilized:** k-Mean, Hierarchal Clustering, DBScan

The main objective for our data modeling studies is to find any pattern or places that could be impacted by high severity weather. In order to achieve this goal, we will be utilizing several unsupervised learning to cluster the severity of the weather based on given cities in the dataset.

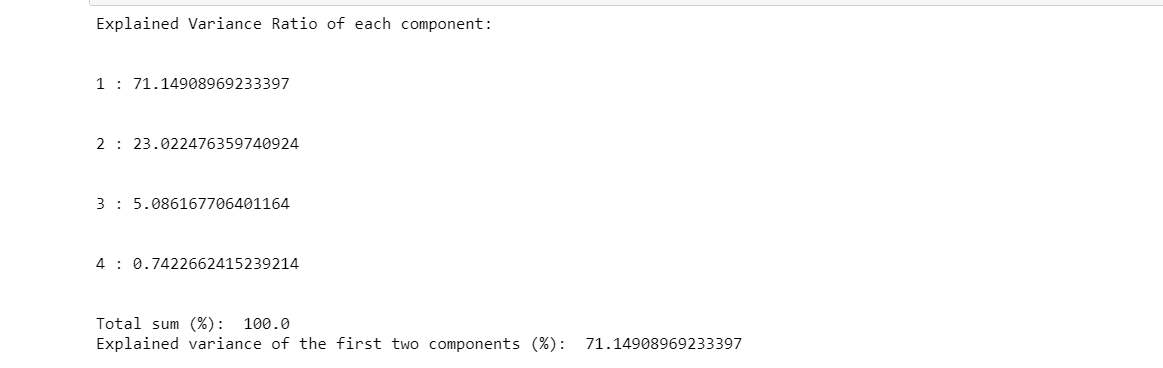
**K-MEAN**

The first unsupervised learning model that we use is **k-Mean.** This model was handled by Chantha Mak. In order to get the K-mean clustering to work, we created another table by dividing each type of severity into separated columns and we also included a city column to represent each instances of the severity record. In order to achieve this, for every city in our dataset, we calculate the sum of each severity within that city and used the sum of each severity as our new data entries. In addition to the new table, we decided to drop the severity instances of UNK (Unknown) and Others since they are not useful to our findings and acted as outliers. Below is the head of the table that we came up with for clustering purposes.

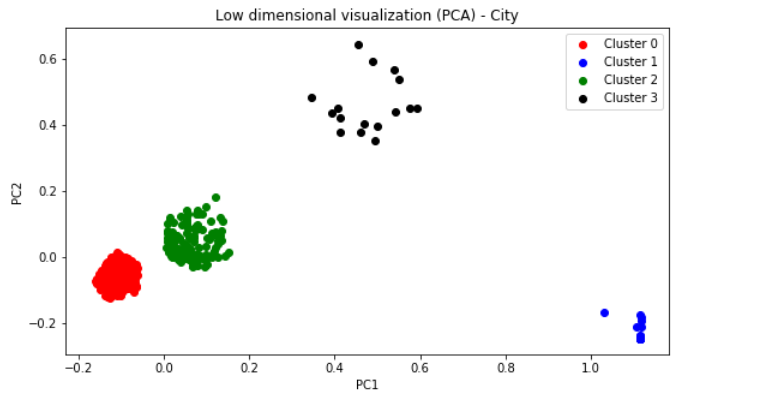
Parameter Tuning for KMean:

In order to obtain the best clustering result for our kmean algorithm, we applied the elbow method to identify the ideal k value that we can use to get the best clustering result. Elbow method essentially just run a loop of k range where k would be the number of clusters for each of the kmean model that we ran in the loop and then we append the clustering result to an array. Finally, we visualization the result with the sum squared distance as the y axis and the k value as the x axis. The ideal k value resides at the elbow of the graph. In our case, we found out that k=4 yield the optimal result.

Visualization:

Since we are dealing with a lot of data with high dimension, in order to fit all the data for visualization purposes, we have to lower the dimensionality of the data by using Principal Component Analysis (PCA). PCA helps with reducing the run time of the algorithm and also helps with visualization. With PCA, we are calculating a matrix that would summarize the relation of how our variables relate to one another. We first needed to scaled the data to standardized the dataset features onto a (0,1) scaled so that we could yield the optimal pe rformance from the clustering algorithm. Below is the result of your PCA run.

As we can see, we were able to reserve 71 percent of the original dataset using the first two components, hence, reducing the dimensionality of the data.

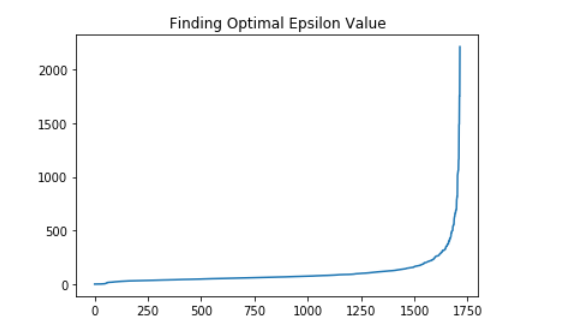
Clustering Result Visualization:

As we can see, k mean was very effective when it comes to clustering the US weather data set.

**DBScan:**

In addition to K-Mean clustering, we applied DBScan. DBScan is an unsupervised learning algorithm that clusters data based on density and are used to classify unlabeled data. With DBScan, we utilized the sklearn library DBscan algorithm that is based off Euclidean distances. With DBScan, we have two main parameter that will play a big part in how our clustering result will turns out. The two parameters are minimum neighbors that a cluster must have to be considered a core point and the epsilon which is a threshold that determine if two points are considered neighbors based on their distances.

Parameter tuning for DBSCAN:

One way to determine the optimal epsilon value for our algorithm is using the elbow method by apply nearest neighbor algorithm. We calculate the distance for each point to its closest neighbor with n\_neighbor=3. The optimal value for epsilon relies on the point of maximum curvature of the graph generated by the elbow method. After we ran the nearest neighbor algorithm, our result suggested that epsilon value of ~250 is the ideal value. And when it comes to the minPts parameter, there’s no particular way to find the best value. We were just testing a range of numbers from 10-100. We don’t want to pick a number that is too small as it would classify a bunch of noises into a cluster group and we don’t want that. We decided to pick min\_sample of 50 as it gives the best result within the 10-100 range. Below is the elbow method graph that we used to find the best epsilon value for our model.