Visual Analytics 690V – Homework 3

- Ajay Shaan Shanmugam
ashanmugam@umass.edu
SPIRE: 31133955

Datasets used:

Wholesale customers data - No Missing Values.csv

Preferred Visualizations:

Scatterplots

The 2 notebooks contain visualizations of clusters in the dataset using two clustering techniques:

- KMeans
- DBSCAN

Notebook 1 (KMeans):

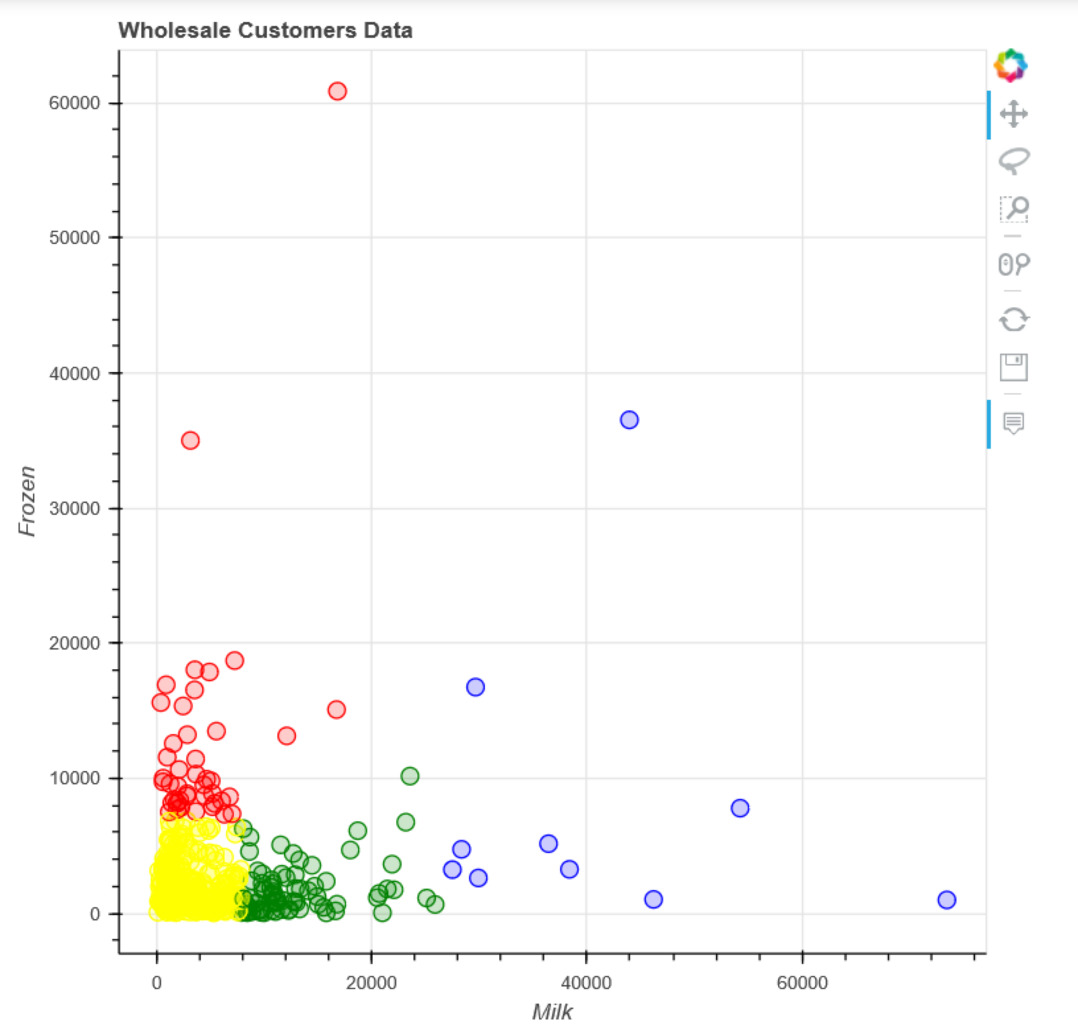
The KMeans clustering model has an iterative algorithm which does the following for every iteration. Prior to this, random centroids, one for each cluster, are selected.

1. Group data points based on their distances from these centroids.
2. Recalculate centroids to be the averages of the data points in each cluster.

It can be noticed that as the number of clusters increases, the number of iterations has to be increased to reach an optimum value.

KMeans is simple, computationally fast and centroid-based. However, it doesn't identify outliers in data, leading to them getting into one cluster or the other. This way, anomalies make their way into groups of quantifiable data points where they don't belong.

For example, in the following KMeans plot, it's apparent that actual outliers are tagged into clusters (reds and blues) when they're too unrelated to truly make one.



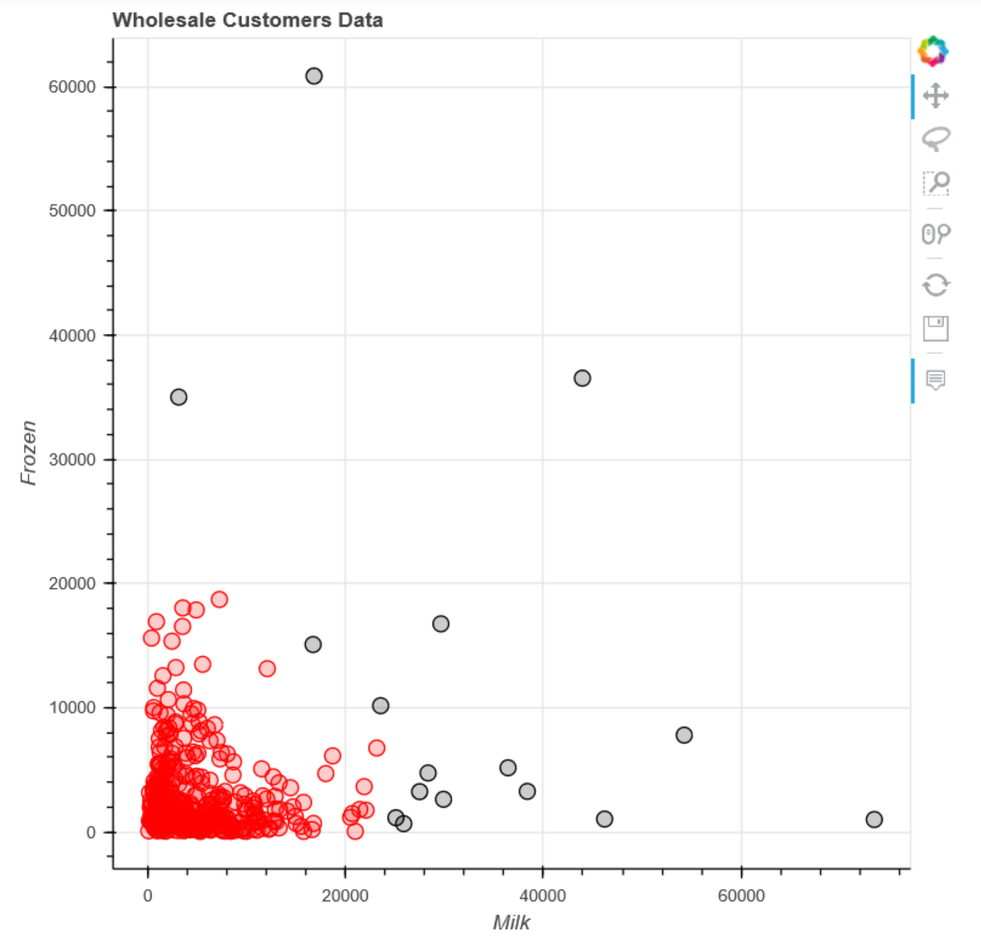
Notebook 2 (DBSCAN):

The DBSCAN clustering model works as follows.

1. If a random data point has more than 'minimum number of samples' in its 'radius', it's considered a core point and all points within its radius are grouped into a cluster.
2. The cluster is expanded by checking if the points in it are core points. At each iteration, the points outside this realm are tagged outliers.
3. When all said points are exhausted, repeat with a new random data point.

Unlike the KMeans model, the DBSCAN handles outliers and label them so. In the following plot, the outliers are labelled in black circles.

Another observation is that for this dataset, the density-based DBSCAN is a good technique for determining outliers. The DBSCAN doesn't prompt for the number of clusters from the user, but instead creates clusters on its own based on density of data points, i.e data points' reachability of neighbors.



It is also observed that when the radius is low, DBSCAN identifies clusters even if they are surrounded by other clusters.

Another major advantage is that it gives one the freedom to tweak values of radius and minimum number of samples to obtain better clustering results. This could be particularly effective for someone with domain knowledge who can make a good judgement on how to define data point neighborhoods.

It can be noticed that the radius is inversely proportional to the number of outliers, and the minimum number of samples is directly proportional to the number of outliers.