Assignment 3

Problem 1 -Linear and kNN Regression

Dataset

The two datasets kind of threw me through a loop for a while, as I could not figure out how to pick which features I wanted to keep. I ultimately decided on leaving out dates because of formatting issues, and there were already features that included that information. Another feature I didn't think I needed was the ‘index’. This just looked like a count, and so it provided no value as a feature to be included. As stated in the assignment we couldn't use the last three rows, so those were removed before processing.

Hourly

This problem dealt with solving for linear and k Nearest Neighbor(kNN) regressions algorithms, which posed quite the confusion to myself. The biggest concern I had while solving this problem was not knowing if my values were correct. Every time I ran the application my output was not what I was expecting. For example, I was getting really low ‘Residual Sum of Squares’ (RSS) numbers (9.457) for kNN regression, which didn’t seem correct, however, when compared to linear regression numbers for the hourly bike ridership it was almost the same. This gave me confidence my answers must be on the right track.

Daily

The daily outputs were a little more of what I was thinking in terms of correct numbers. This made me believe that with the more granular the measurements the less the numbers would be. For instance, if we had time of ridership in seconds they would most likely be smaller than the hourly RSS scores. Another thing I noticed was the variance score of both datasets. It seems there was a big variance, which could only be attributed the randomness of when customers actually ride bikes.

Problem 2 - kMeans

Dataset

The dataset for the seeds was a lot easier to work with. The only thing I needed to change was to not include the feature in column eight.

Seed Feature Clustering

Out of all of the classifiers and regression algorithms clustering makes the most sense to me conceptually. Basically, it's taking your dataset and splitting them up into the features you supplied, or if some features can be merged then they're clustered together. The outcomes of this dataset are pretty close to the ground truth. My script produced **Clusters (result of k-means)** with a count of **{1: 75, 0: 74, 2: 61}**. Compared to its Ground Truth of **{1: 70, 2: 70, 3: 70}** I would say those results are pretty close. Thus, making the script and code highly accurate.