**PCA**

To perform PCA we encoded ordinal non-numeric features with appropriate sequential integers, and used the label encoder to encode all nominal non-numeric features. We then applied PCA without limiting the number of components to get a sense of the data. The quick tailing off of variance in the dataset explained by each subsequent component suggests that not many features in the housing training dataset are strongly correlated. We then limited the number of components for the analysis. It is worth noting that a reduction in the dimensionality of the data by half -- from 80 columns in the dataset to 40 components -- still encompasses ~80% of the total variation in the dataset. If one wishes to drastically reduce dimensionality we would suggest using 15 components, which is sufficient to incorporate ~50% of the total variance.

**K Means Model**

To create a k means model, we first plotted all the numerical columns against each other to find some sort of clustering pattern. Sale Price vs. Garage Area gave a somewhat prominent cluster, although there was overlapping. This led to a problem when using the K means classifier. The model chose random variables and approximated the closest points as a part of the cluster of those variables. We noticed that the model classified several variables correctly. The problem was when there was overlapping. The MS Zoning column was used as the hyper parameter because it created quite prominent clusters indicating that sale price and garage area, proportional to the actual area of the house, were dependent on the zone the house was located in. When trying to fit data into this model, we realized that the k means classifier works well when the data has clear-cut clusters. Overlapping boundaries or points prevent the classifier from being as accurate as possible since this model classifies all the points near the chosen centroid as one cluster.

**Hierarchical Clustering**

Originally, we had attempted to use the entire dataset (minus the “target” feature) in the hierarchical clustering model to see if it would result in clusters that approximated that of the groups that could be found in the “target” features. Some of the features we tried this on were “Neighborhood” and “MSSubClass”. However, this performed poorly, resulting in dendrograms that were dominated by one large cluster. The problem may be that these features have too many types (e.g. there are 25 types of neighborhoods), such that they cannot be differentiated clearly. Another issue is that these features may, in general, correlate weakly with the other clusters and hence not form distinct clusters.

We took a different tack by limiting the dataset to quality / condition – related features to try and find clusters for the ‘OverallQual’ column. The data was first ordinally encoded such that the numeric value was proportional to the quality ratings (i.e. 2 for Poor, 6 for Excellent.) Then we used a standard scaler on the data to scale it to unit variance, which typically improves the results obtained from hierarchical clustering. This worked well, and as can be seen in the generated diagram, the length of the vertical lines shows that there is a substantial distance between clusters, i.e. there is a strong

clustering in the dataset. The dendrogram reflects 4 main clusters, which we would say corresponds roughly with the ‘OverallQual’ data:

| Rating | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Count | 2 | 3 | 20 | 116 | 396 | 374 | 318 | 167 | 43 | 18 |

We can divide the data according to the above groupings, with the assumption that ratings with low counts are grouped with the next higher rating, until the total group is relatively similar in size to the other clusters. This is why we group ratings 1-4 and 8-10 together. This is an approximation or guess on our part, and it may be the case that ratings 1-5 actually form a cluster instead. Nevertheless, we can see that ‘OverallQual’ can be roughly divided to about 5 clusters, which is close to the dendrogram’s 4 clusters.