MACS 301
Homework 9

Reid McIlroy-Young

March 15, 2017

1 Feminist

Part 1

To start with the data were loaded and a testing set of 30% of the data was created.

Part 1

To start with the data were loaded and a testing set of 30% of the data was created.

Part 2

The minimum MSE is 438.17, at 90 neighbours, figure 1 shows that after 30 the differences are minimal though. All models were fitted using all variables.

Part 3

The minimum MSE is 534.33, at 40 neighbours, figure 2 shows that after 30 the differences are minimal though. All models were fitted using all variables.

Part 3

The minimum MSE is 464.65, for the linear model. Although, figure 3 shows that both NNs, boosting and logit are also close. All models were fitted using all variables. This likely means that the relationship is nearly linear, which would also result in minkowski adjacency being a good metric.

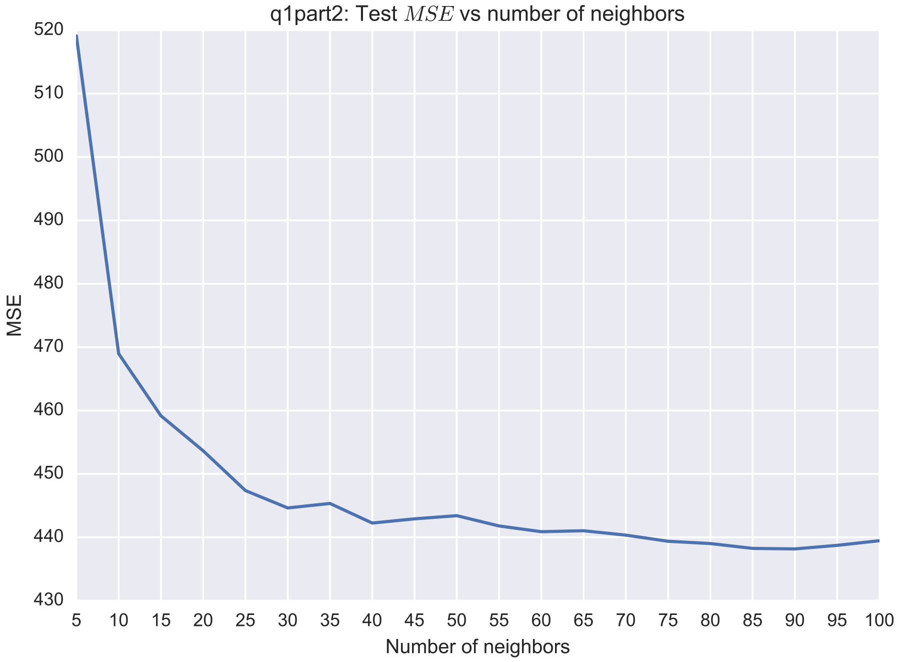

Figure 1: *q1part2*

2 Voters

Part 1

To start with the data were loaded and a testing set of 30% of the data was created.

Part 2

The minimum error rate is 0.29, at 10 neighbours, figure 4 shows that the error rate was decreasing still so more neighbours still might be better. All models were fitted using all variables.

Part 3

The minimum error rate is 0.28, at 10 neighbours, figure 5 shows that the error rate was decreasing still so more neighbours still might be better. All models were fitted using all variables.

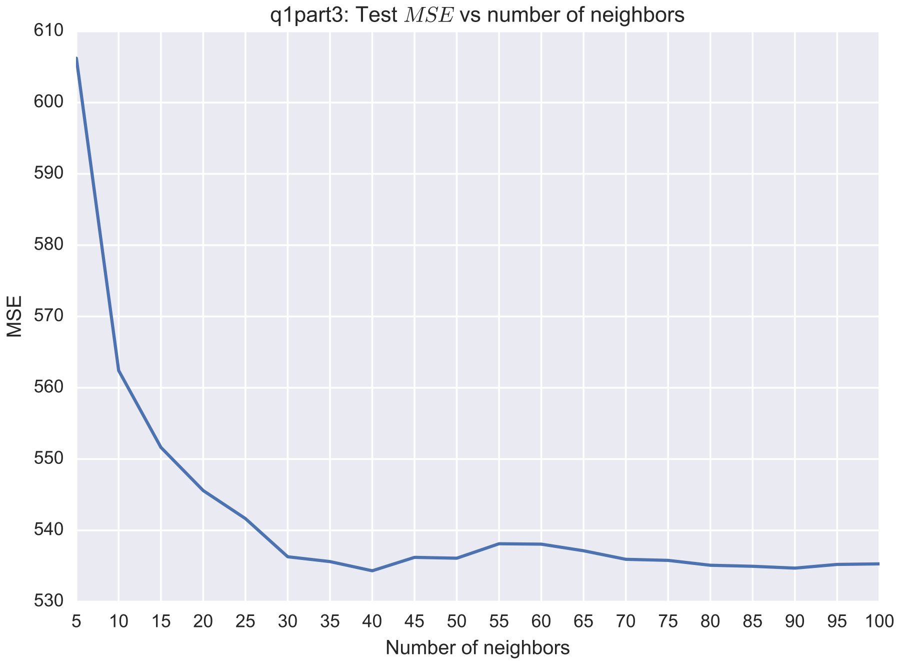

Figure 2: *q1part3*

Part 4

The minimum error rate is 0.27, for the 100-NN, weighted model. Although, figure 6 shows that boosting was also close. All models were fitted using all variables. Since the linear SVM did not do that well the sections are likely not linearly separable, but since 100-NN works well we can assumes there are a few clusters in minkowski space, they may just have a complicated boundary and be more than two. Boosting being good is just because AdaBoost is a really good technique.

3 Colleges

Figure 7 shows the results of projecting the data onto the first two PCA dimensions. I have also added a point for each of the original dimensions, these have all but the named dimension set to 0, with the selected one set to largest value seen in the dataset. We can see that *Outstate* maps almost perfectly to $PCA0$ and *Accept* to $PCA1$, although both negatively so. *Room.Board*, *F.Undergrad* and *Expend* also appear to be significant as they are not on the same $\frac{\pi}{4}$ angle from an axis and thus are effected by the PCA component.

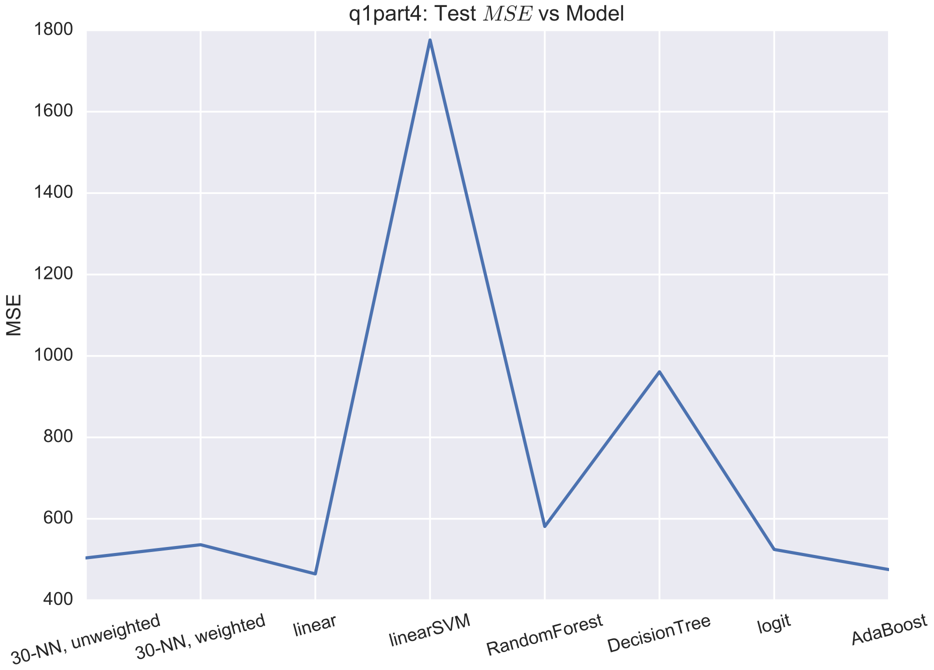

Figure 3: *q1part4*

4 States

Part 1

Figure 8 shows the results of projecting the data onto the first two PCA dimensions. I have also added a point for each of the original dimensions, these have all but the named dimension set to 0, with the selected one set to largest value seen in the dataset.

Part 2

Figure 9 shows the results of 2-means clustering projected onto the first two PCA dimensions. The plot shows the PCA space being nicely split in half around the $PCA0$ axis.

Part 3

Figure 10 shows the results of 3-means clustering projected onto the first two PCA dimensions. The plot shows the PCA space being nicely split into

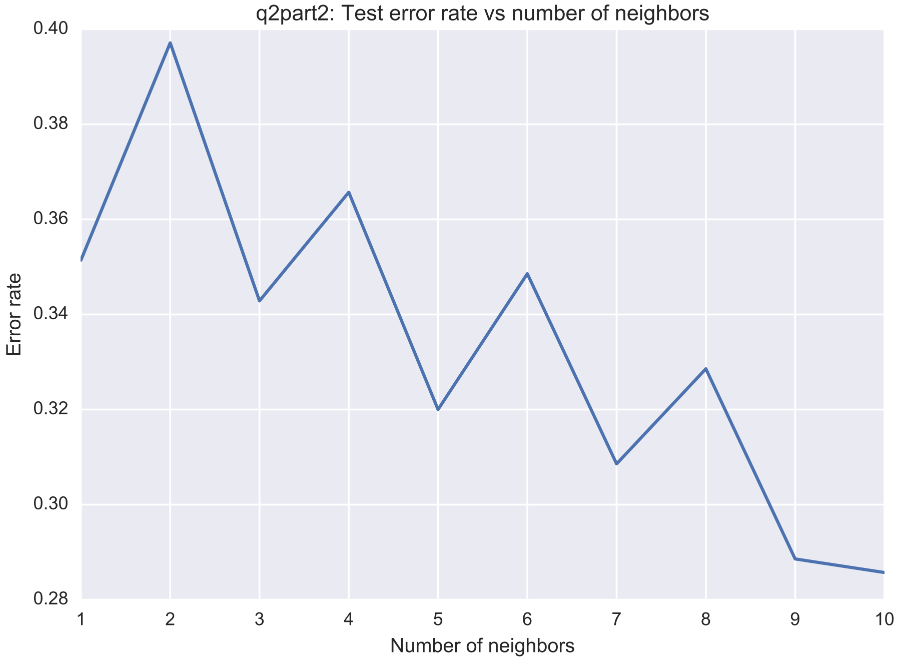

Figure 4: *q2part2*

thirds with one centred around the $PCA0$ axis and the other two on either side.

Part 4

Figure 11 shows the results of 4-means clustering projected onto the first two PCA dimensions. The plot shows the PCA space being nicely split into 4 strips with two of negative $PCA0$ and two with positive $PCA0$.

Part 5

Figure 12 shows the results of 3-means clustering of the first two PCA dimensions. It is not much different from Part 3, which suggests the projection did not affect the space much.

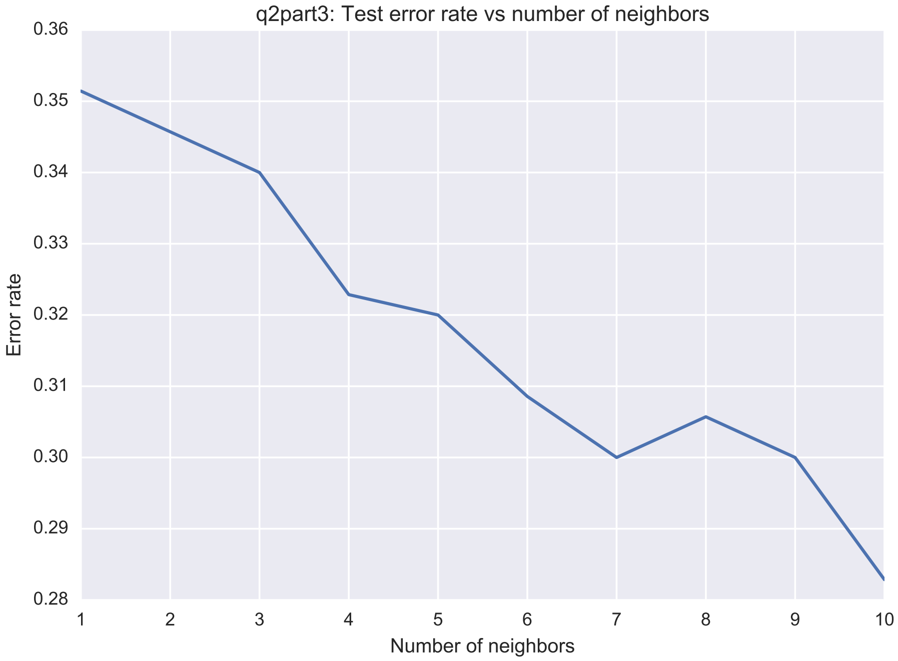

Figure 5: *q2part3*

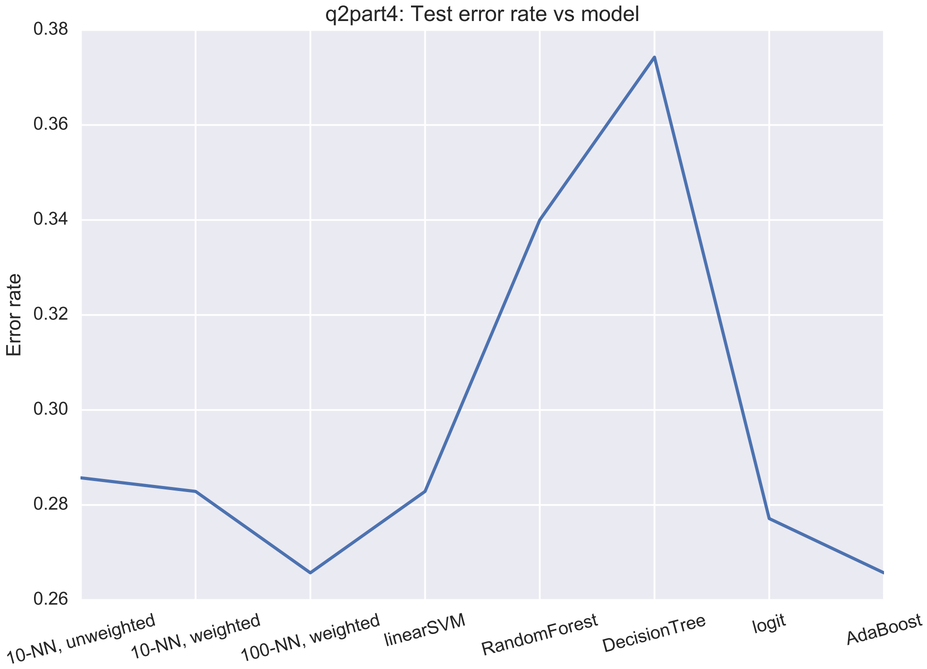

Figure 6: *q2part4*

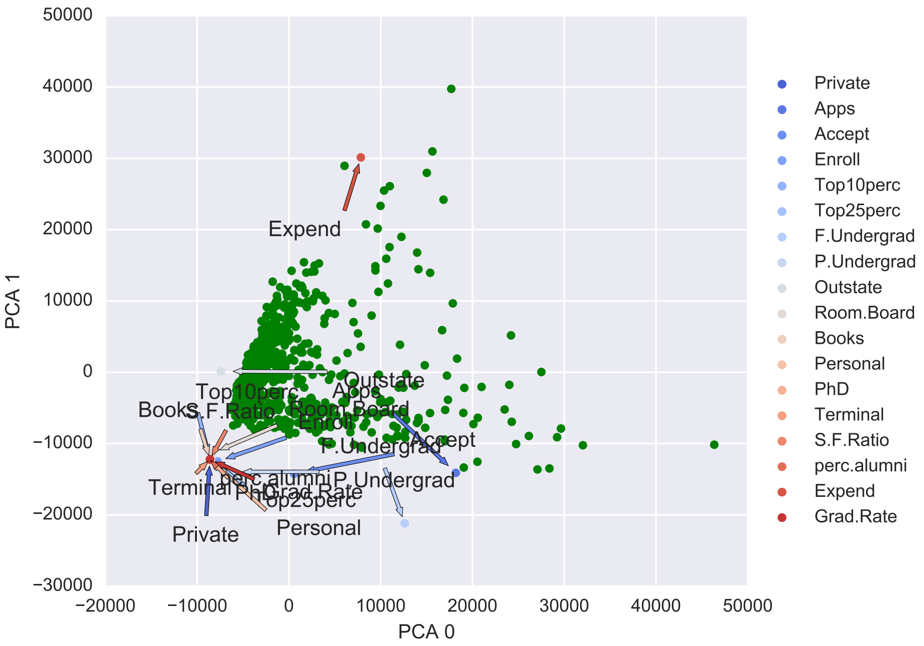

Figure 7: *question3, labels were randomly positioned because matplotlib does not like annotations*

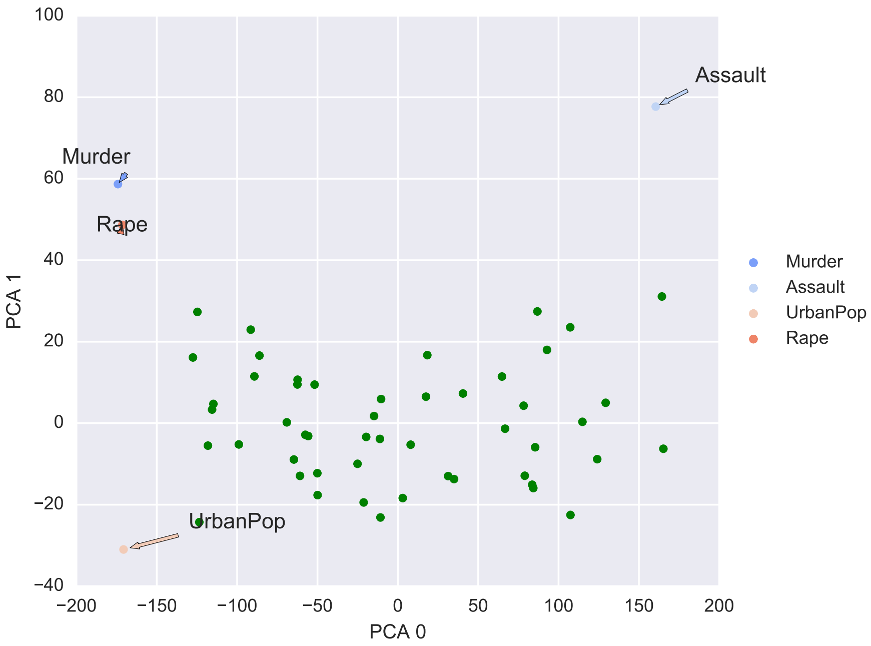

Figure 8: *q4part1, labels were randomly positioned because matplotlib does not like annotations*

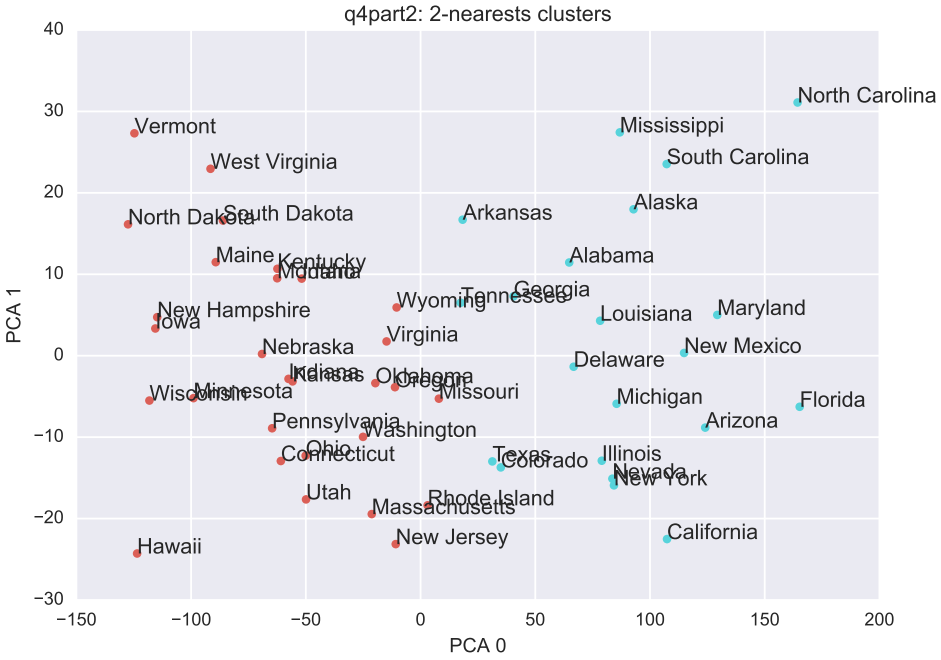

Figure 9: *q4part2*

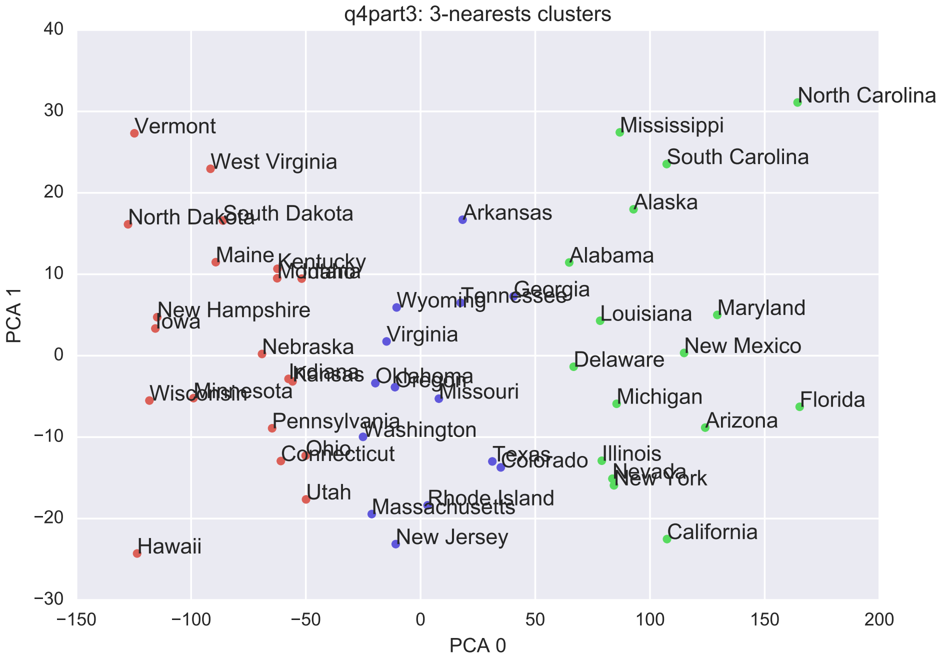

Figure 10: *q4part3*

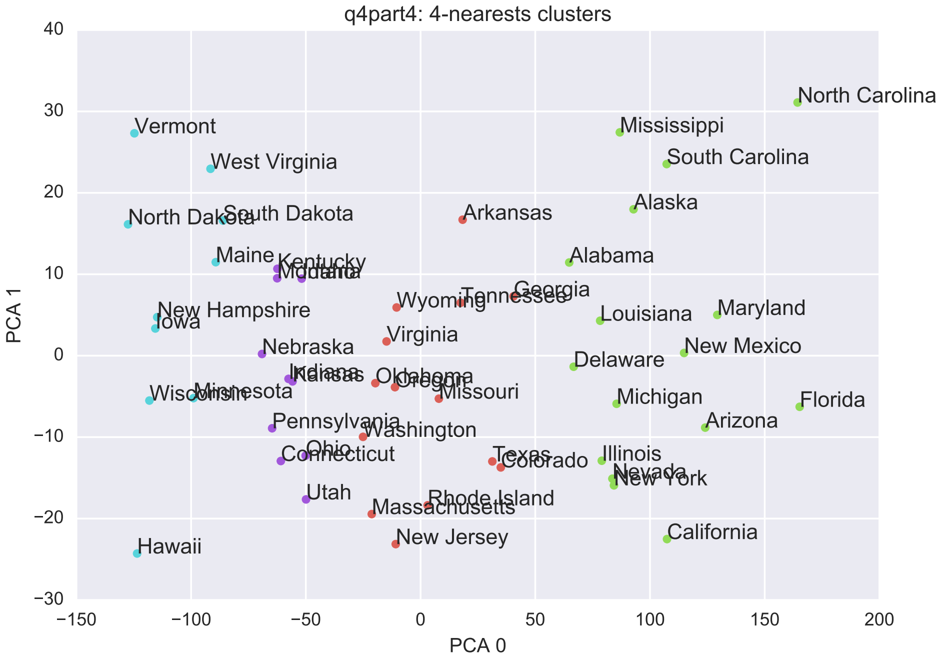

Figure 11: *q4part4*

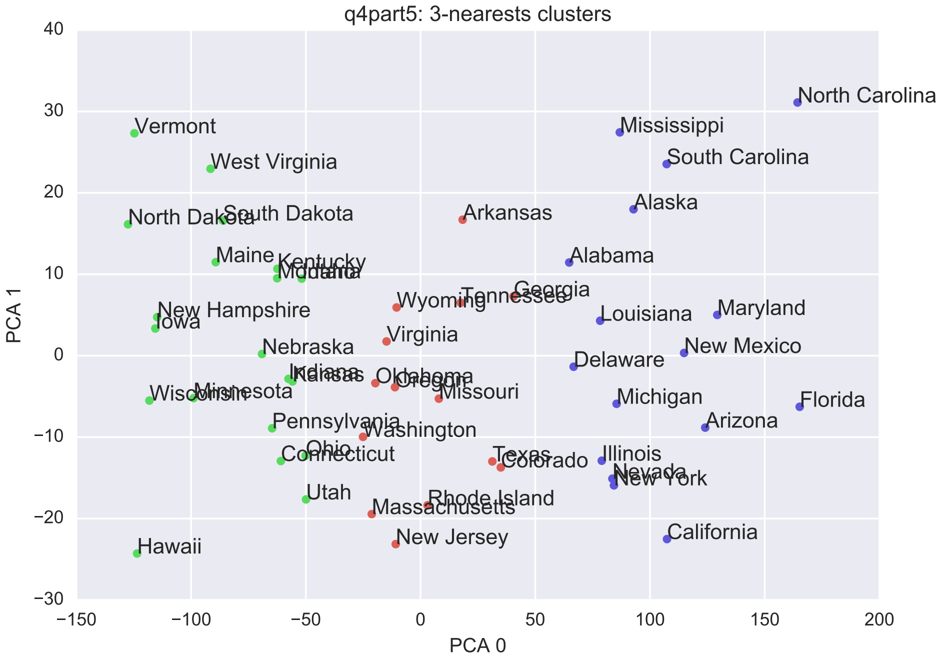

Figure 12: *q4part5*