Unsupervised Learning Comparisons and Insights

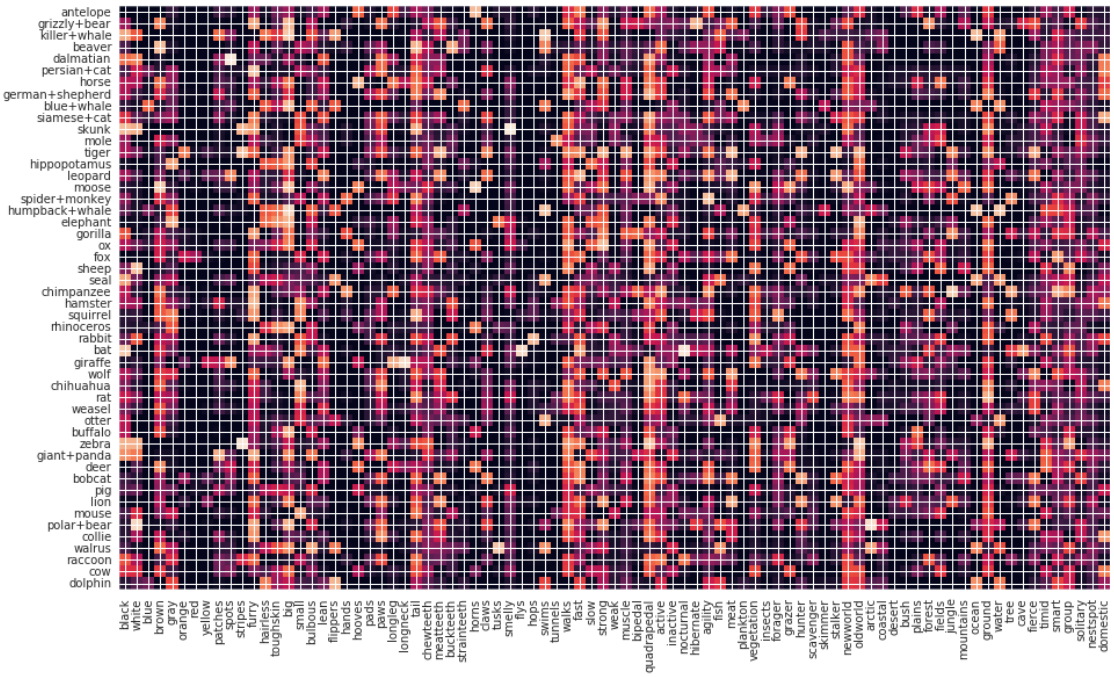
Scott Viteri
Massachusetts Institute of Technology
Cambridge, MA 02139
sviteri@mit.edu

Abstract

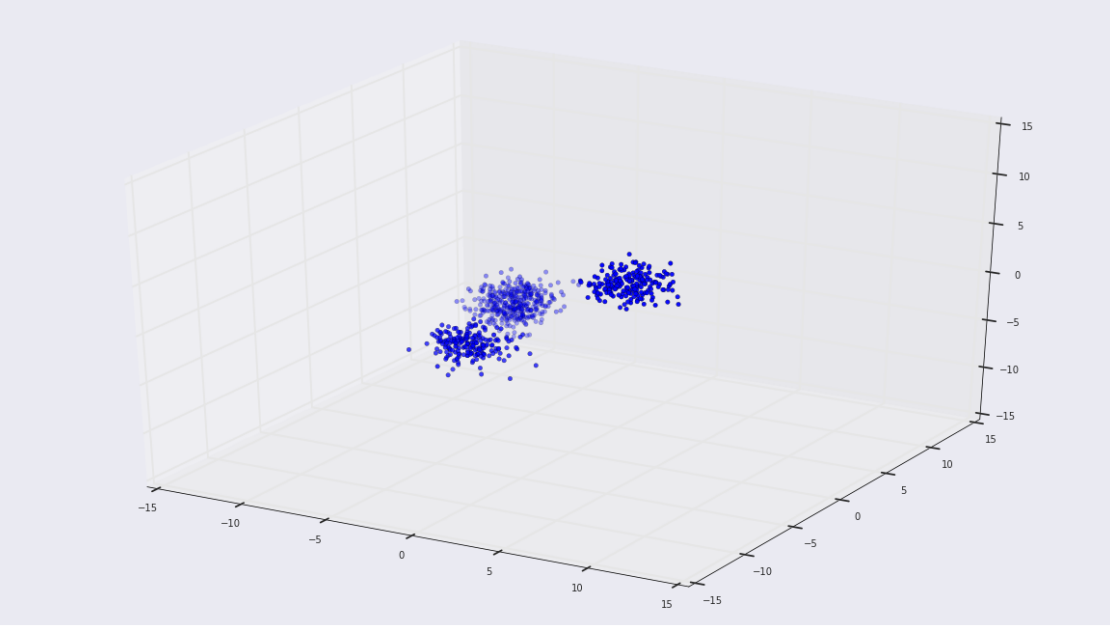
We apply the unsupervised learning methods of principle component analysis, multi-dimensional scaling, and gaussian mixture models to the animal feature dataset created by Osherson et al 1991. In the process, we compare and contrast structures found and generalization performance.

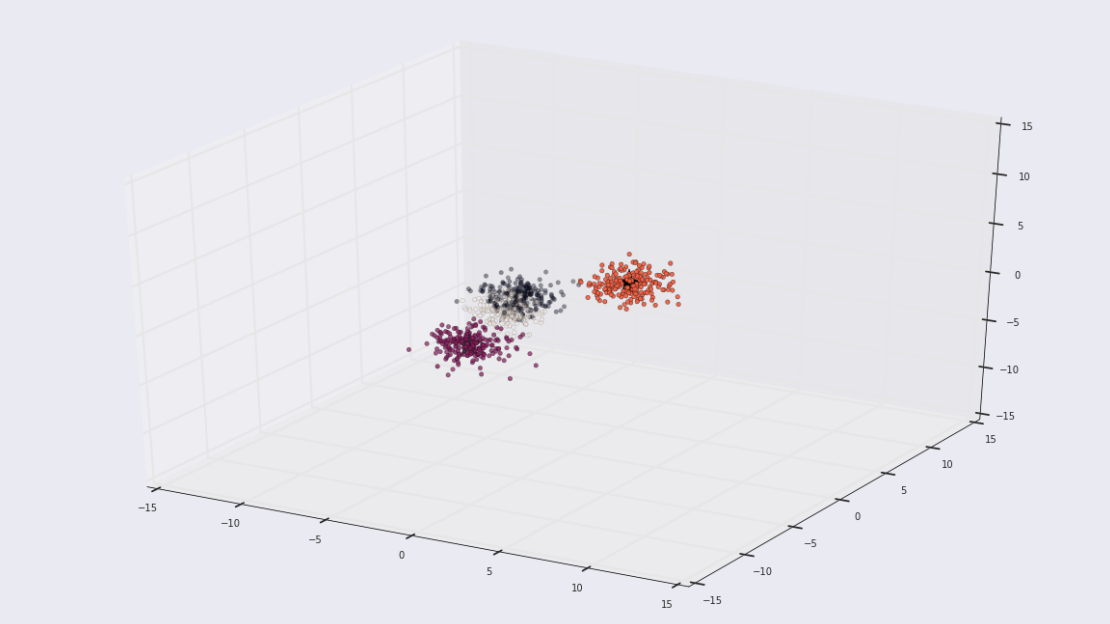
1 K-Means

First, we load the relevant animal data.

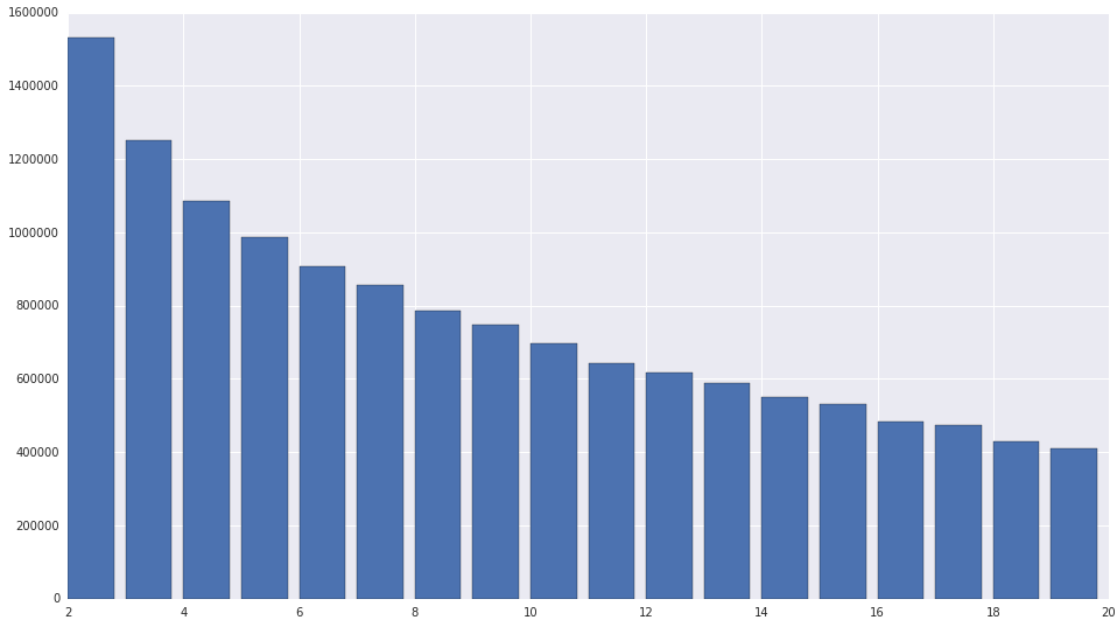


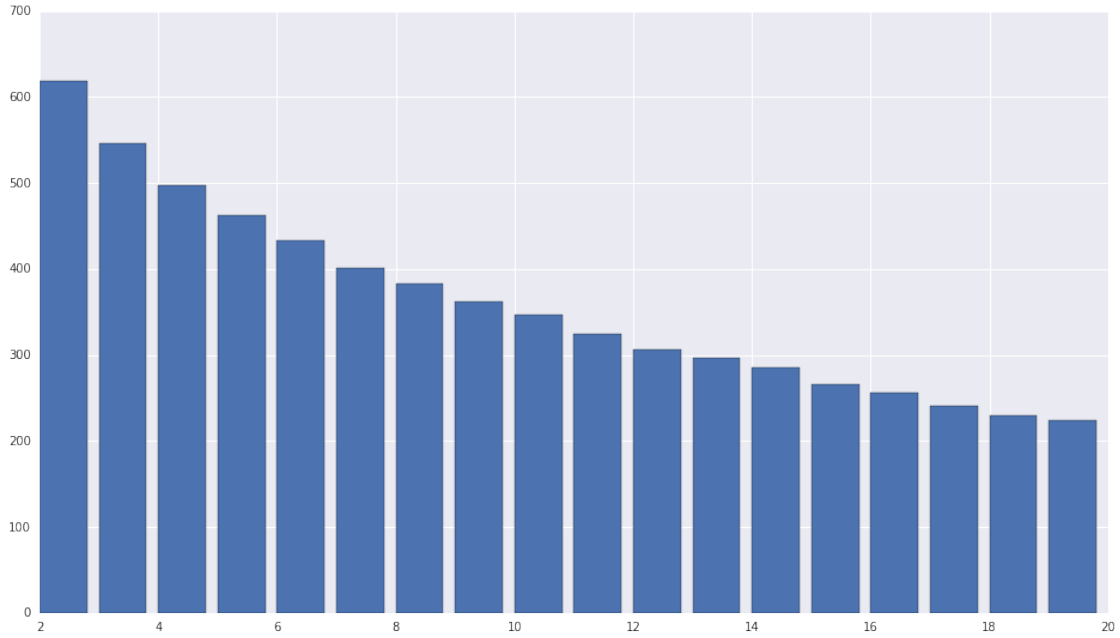
Then we demonstrate k-means clustering.





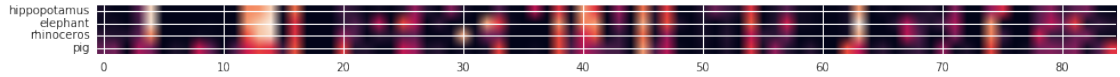
Then we plot the accuracy of clustering on the animal data as a function of the number of clusters k.

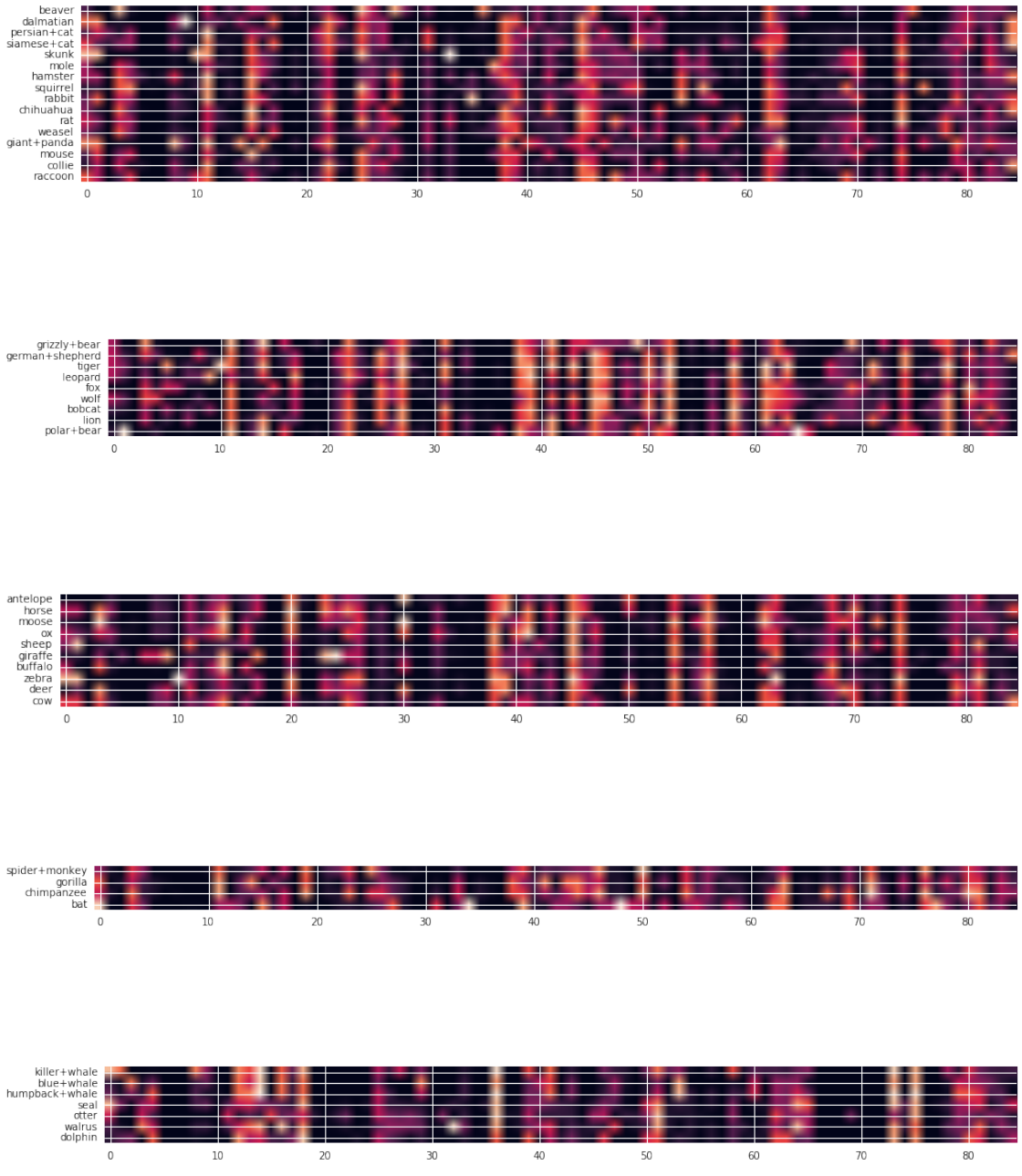




We learn that the k-means error rate for this data set drops off slowly with the number of variables. We set the future number of clusters to 6 as a tradeoff between small size and low error rate.

Plotted below are the six clusters.

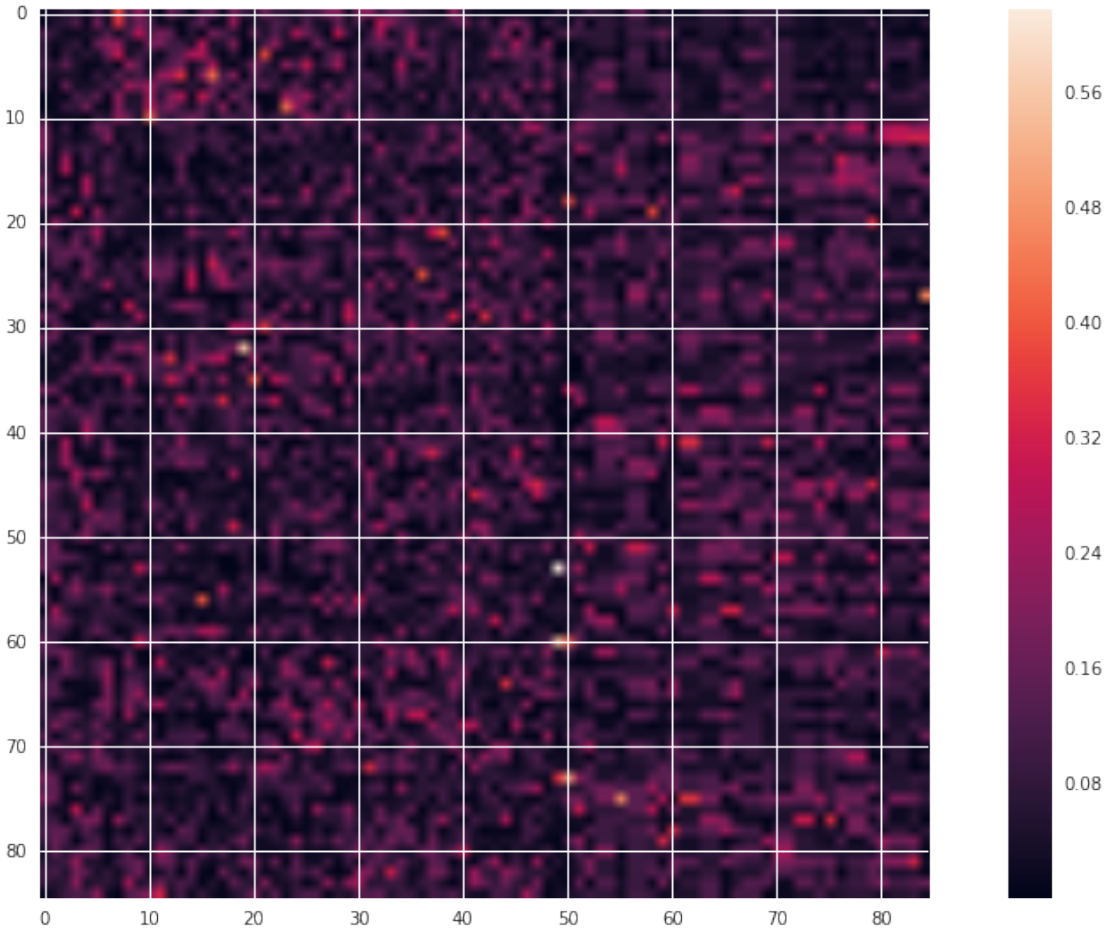




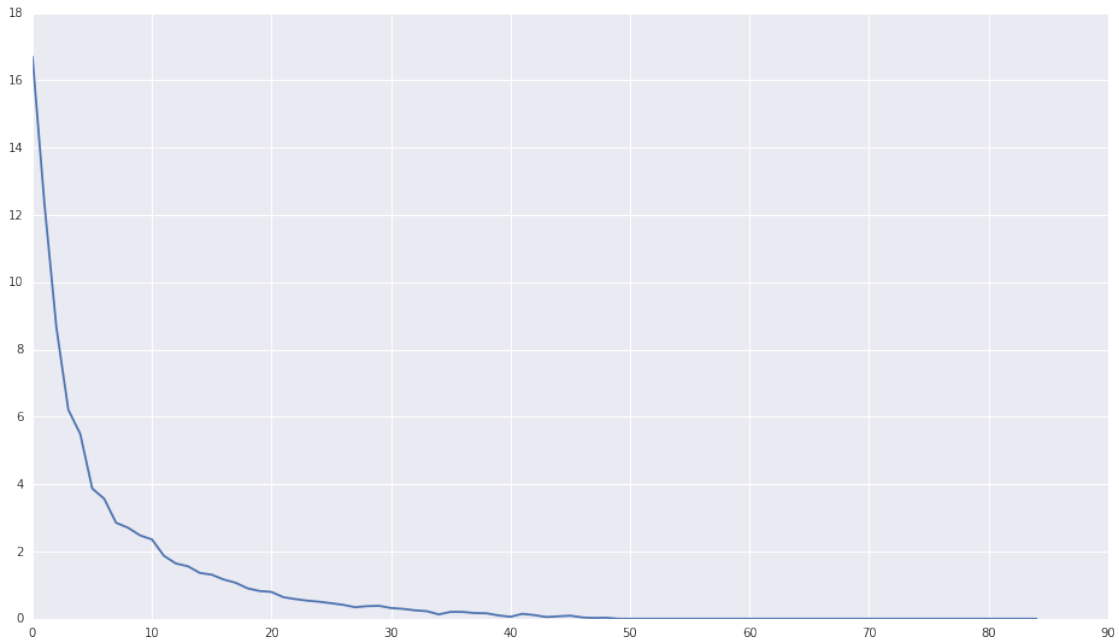
The clustering is reasonably intuitive with k-means. Perhaps we can get clearer separation by first running principle component analysis and projecting into the data into a lower dimensional subspace.

The steps are as follows: 1. Standardize data 2. Obtain the eigenvecs and eigenvals from the covariance or correlation matrix or perform SVD 3. Sort eigenvals in descending order and choose the k eigenvecs that corresond to the k largeset eigenvals (choose some k <=d) 4. Construct the projection matrix W from the selected k eigenvecs 5. Transform the original dataset X via W to obtain a k-dim feature subspace Y

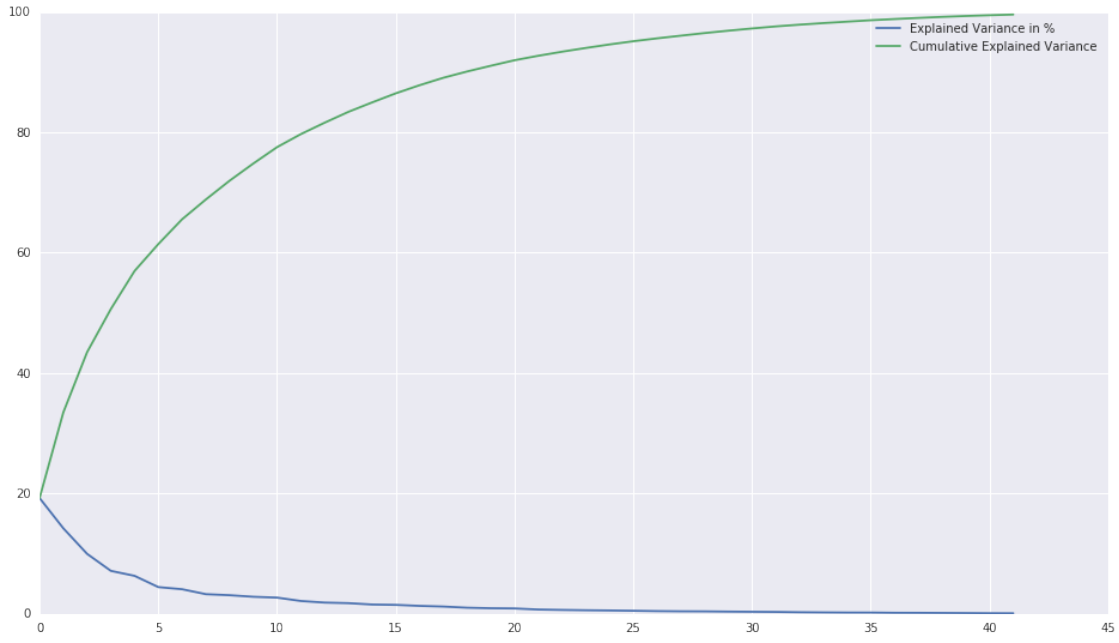
Performed steps 1 and 2 on data, and plotted eigenvals below.



3. Sort by largest eigenvalue, and plot below to verify sorted correctly.

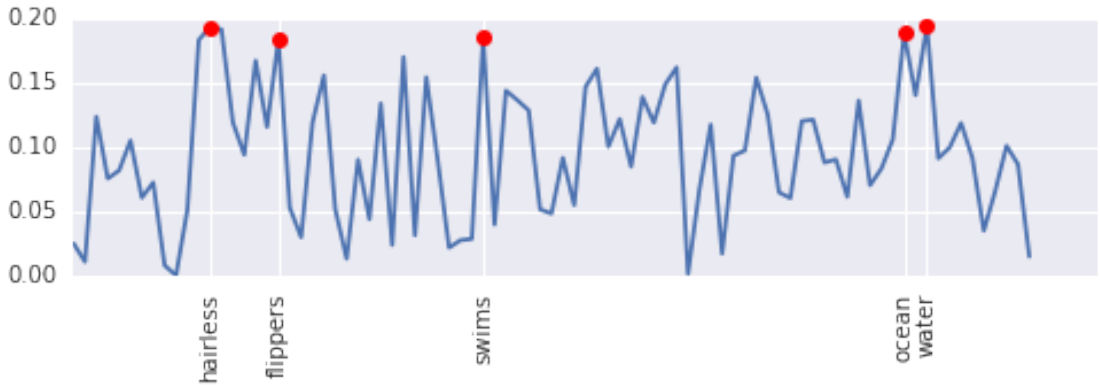


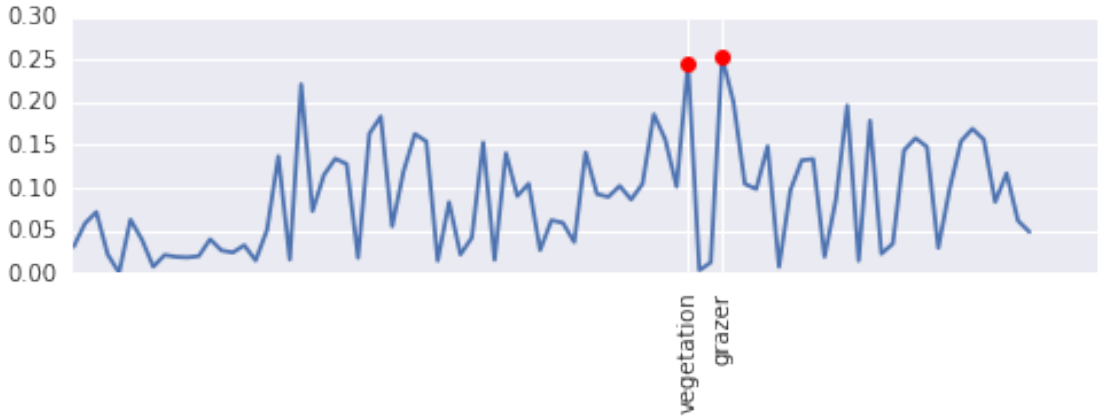
4. Constructed projection matrix, and plotted the amount of percentage of variance each 2D invariant eigenspace accounts for. The accumulated percentage is plotted above it.

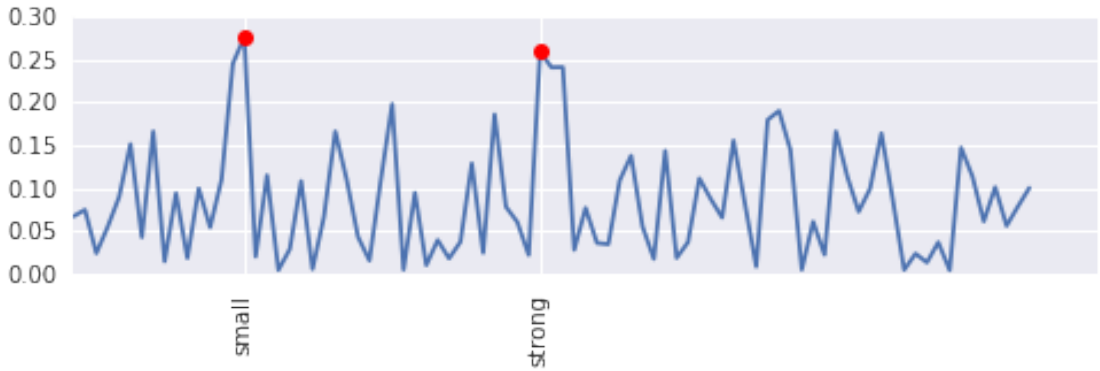


6 eigenvectors strikes a good tradeoff between information and simplicity. Plot the animal data in the projected feature space.

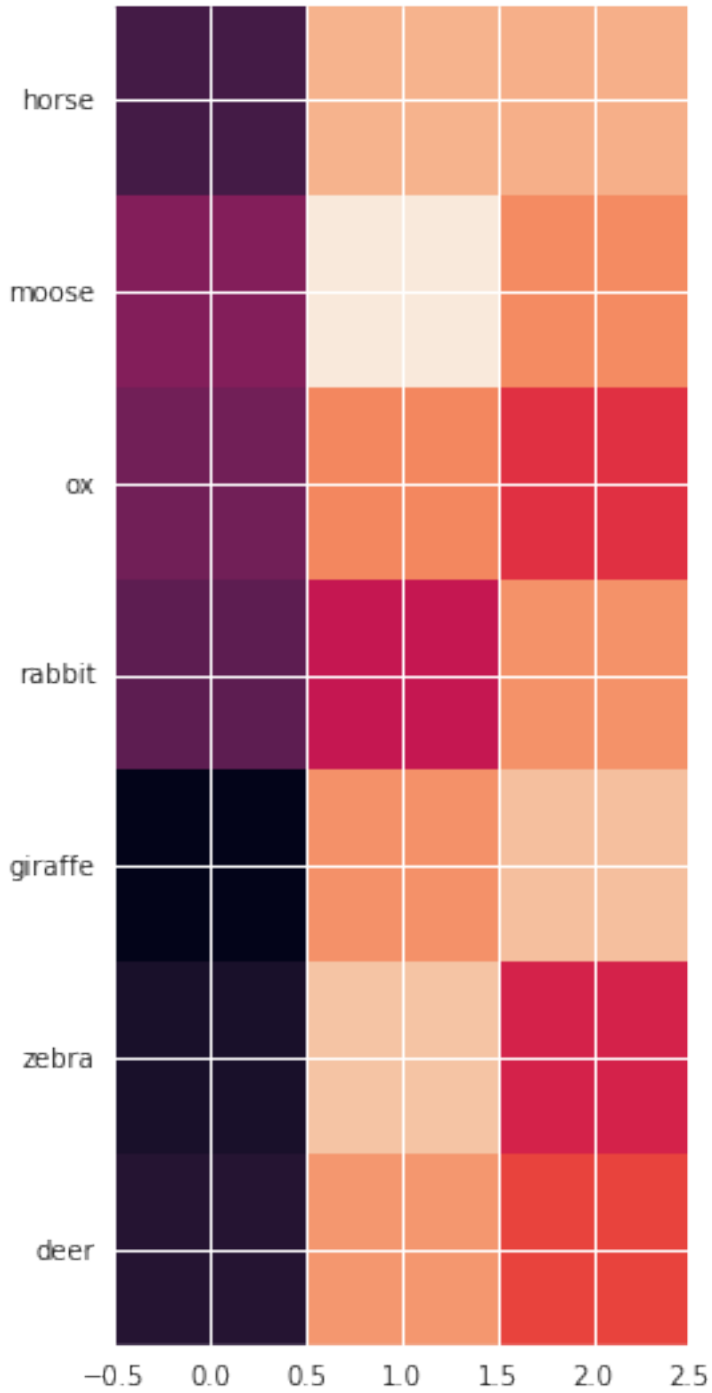
The x axis no longer corresponds directly to labelled features. Rather they correspond to linear combinations of labelled features. Plotted below are the first few eigenvectors, and the corresponding most relevant features in those eigenvectors.

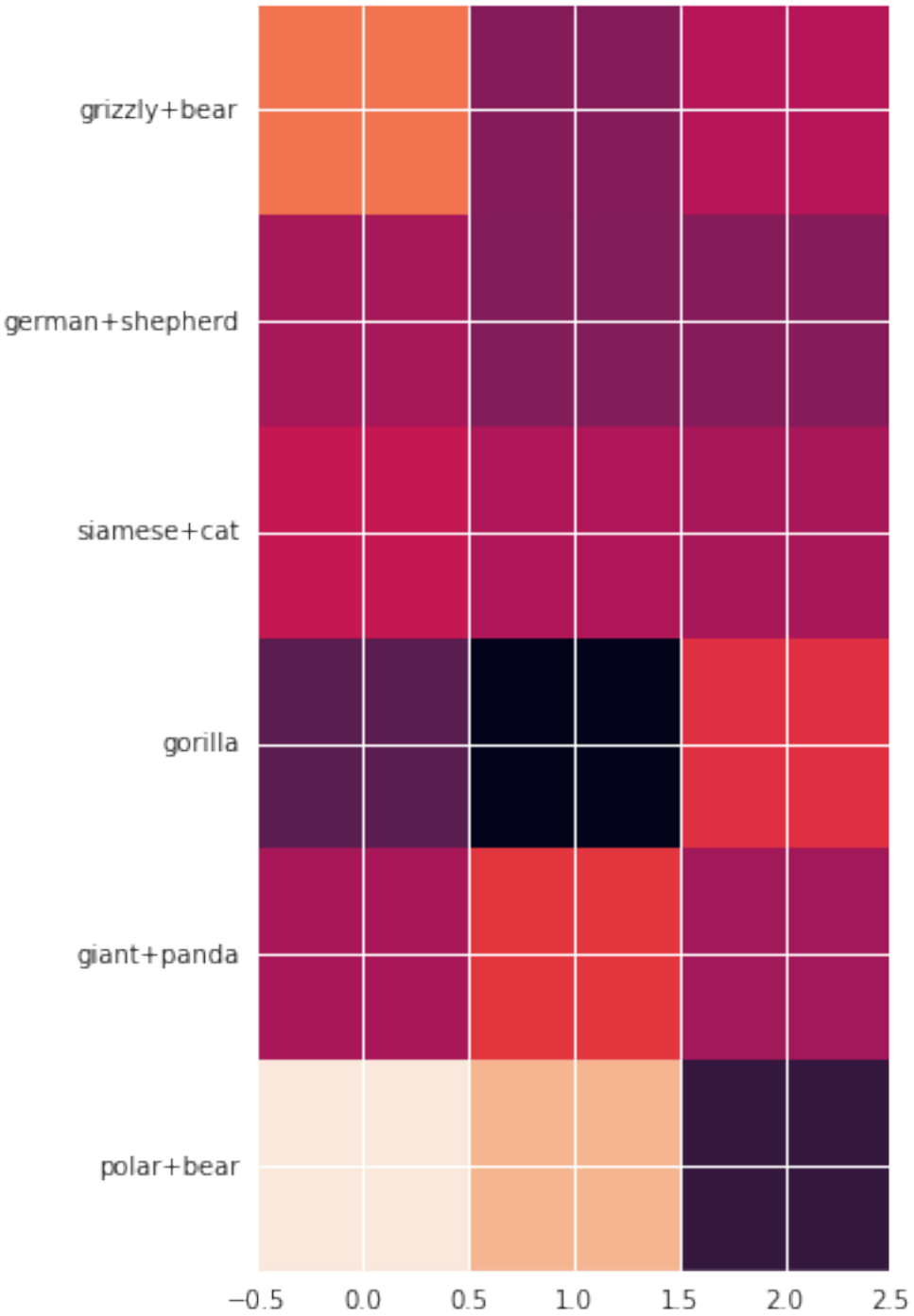


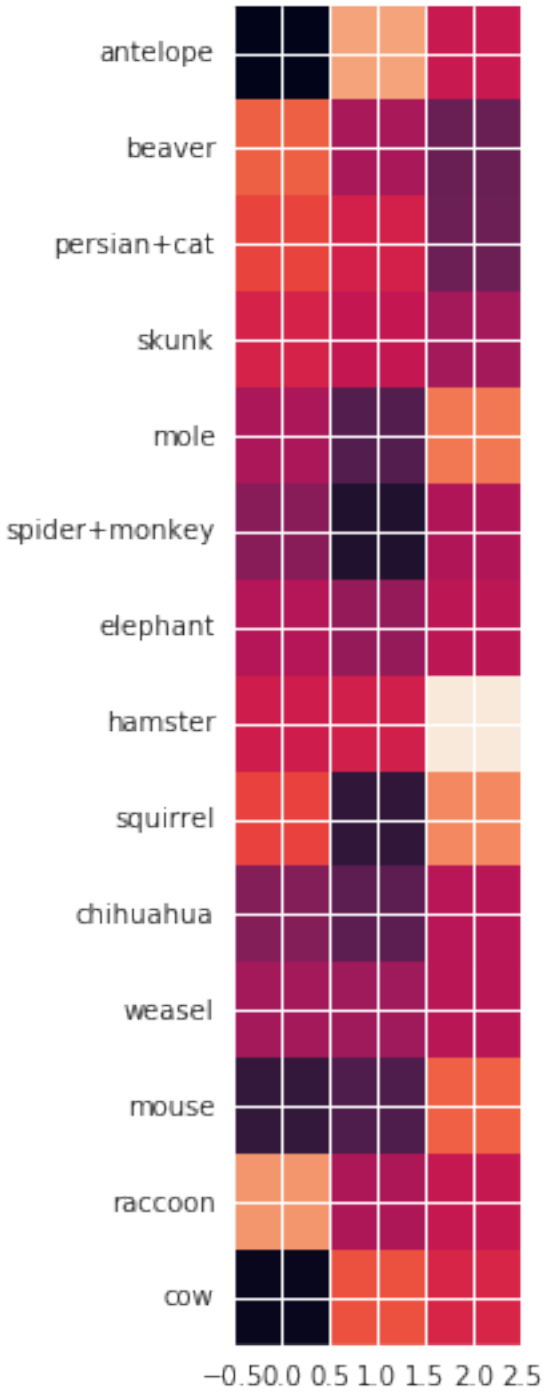


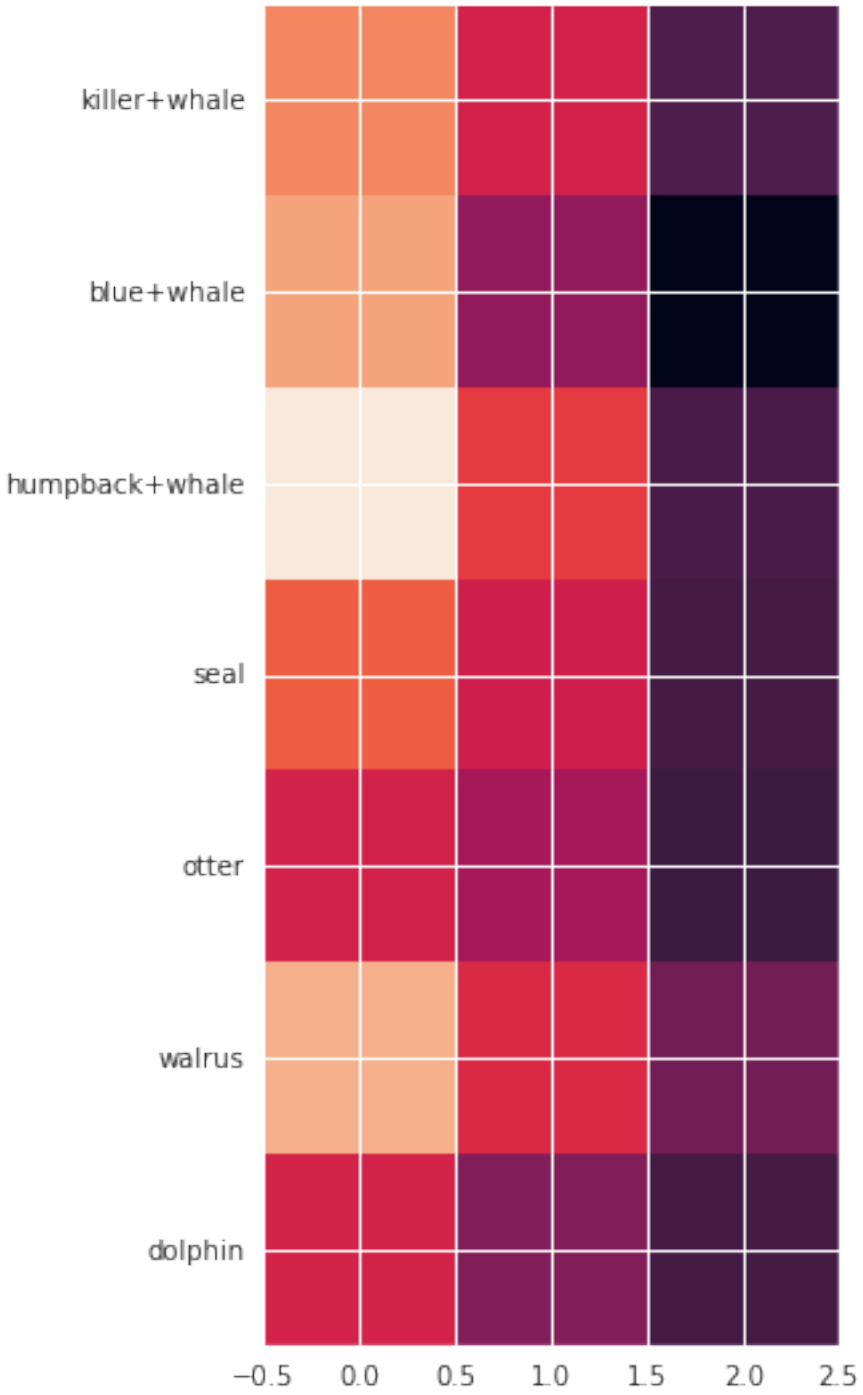


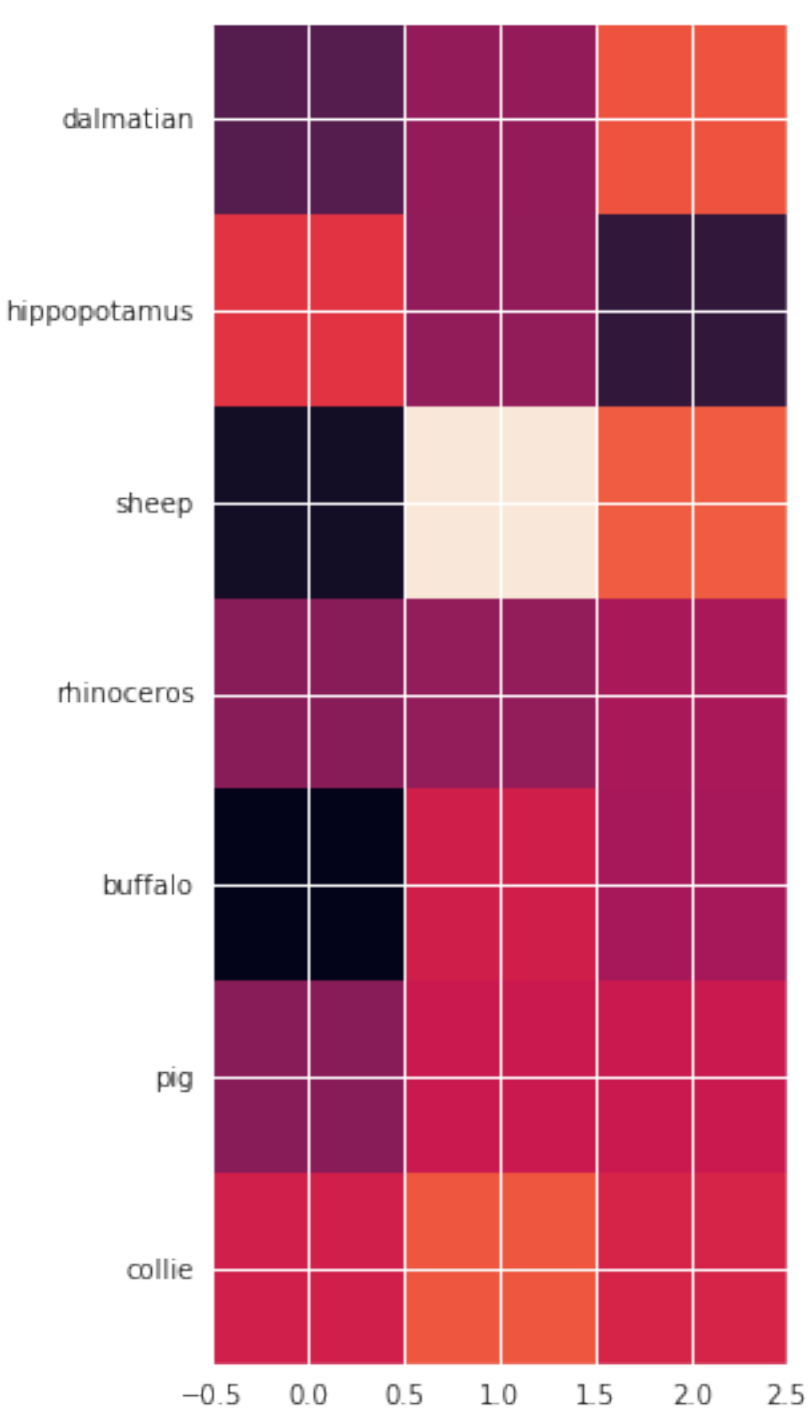
Plotted next are k-means clusters of 6, used on the reduced-dimensional data.The data has been reduced to 3 dimensions, but the output is less intuitive than the k-means case. Maybe this was not the appropriate dimensionality reduction.

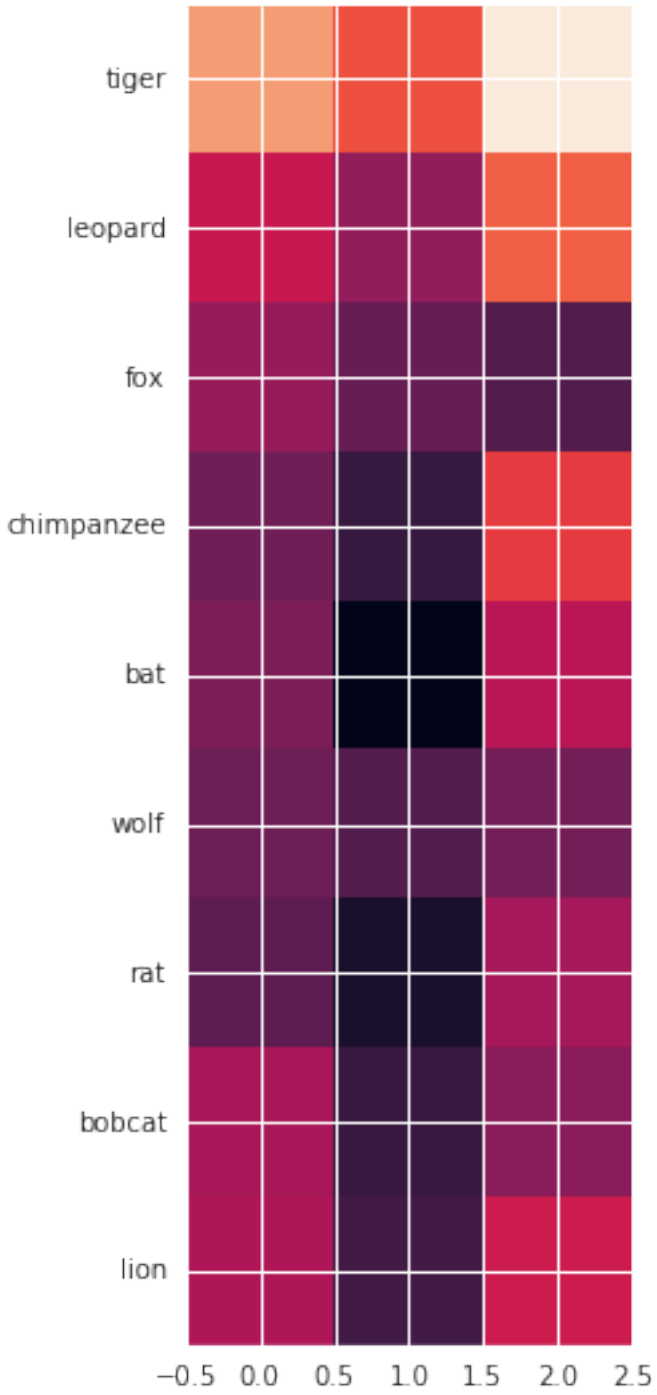

horse
moose
ox
rabbit
giraffe
zebra
deer
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


grizzly+bear
german+shepherd
siamese+cat
gorilla
giant+panda
polar+bear
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


antelope
beaver
persian+cat
skunk
mole
spider+monkey
elephant
hamster
squirrel
chihuahua
weasel
mouse
raccoon
cow
−0.5 0.0 0.5 1.0 1.5 2.0 2.5


killer+whale
blue+whale
humpback+whale
seal
otter
walrus
dolphin
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


dalmatian
hippopotamus
sheep
rhinoceros
buffalo
pig
collie
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

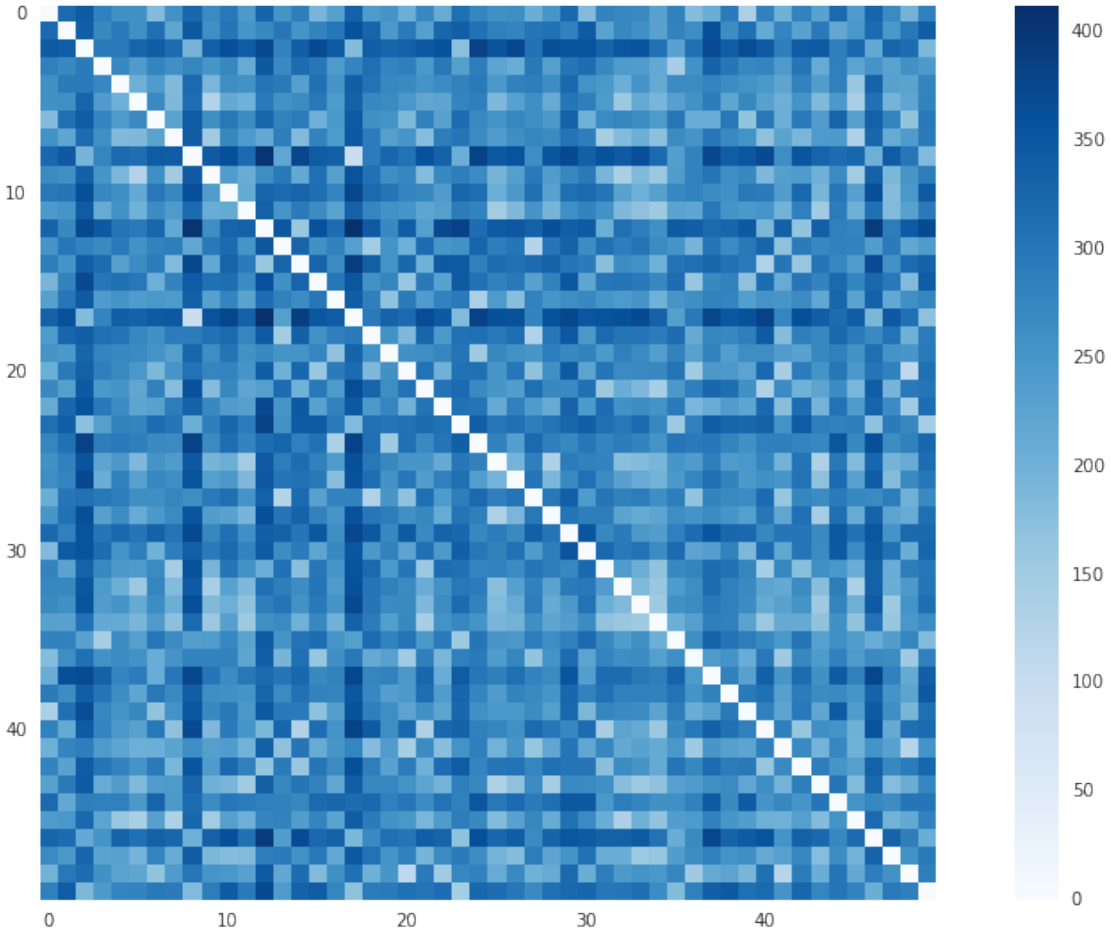


PCA is betting that there is a lower dimensional subspace that contains a decent amount of the data. But it seems that is not the case for our animals data set.

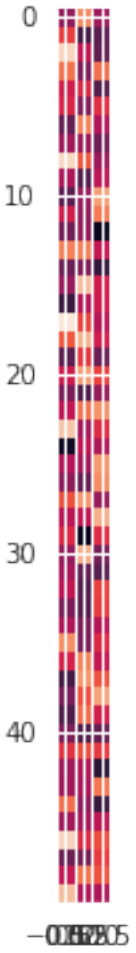
Try MDS.

First get pairwise distances.

Try a dimensionality reduction with 3 components.

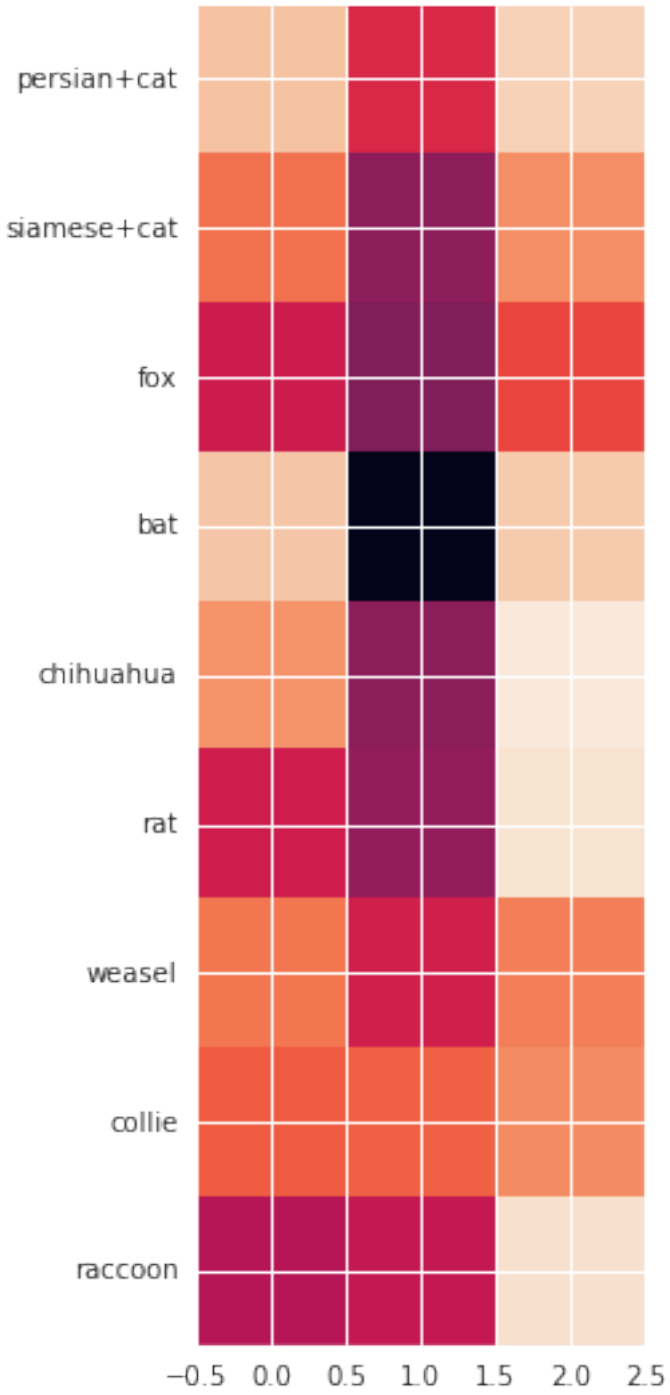


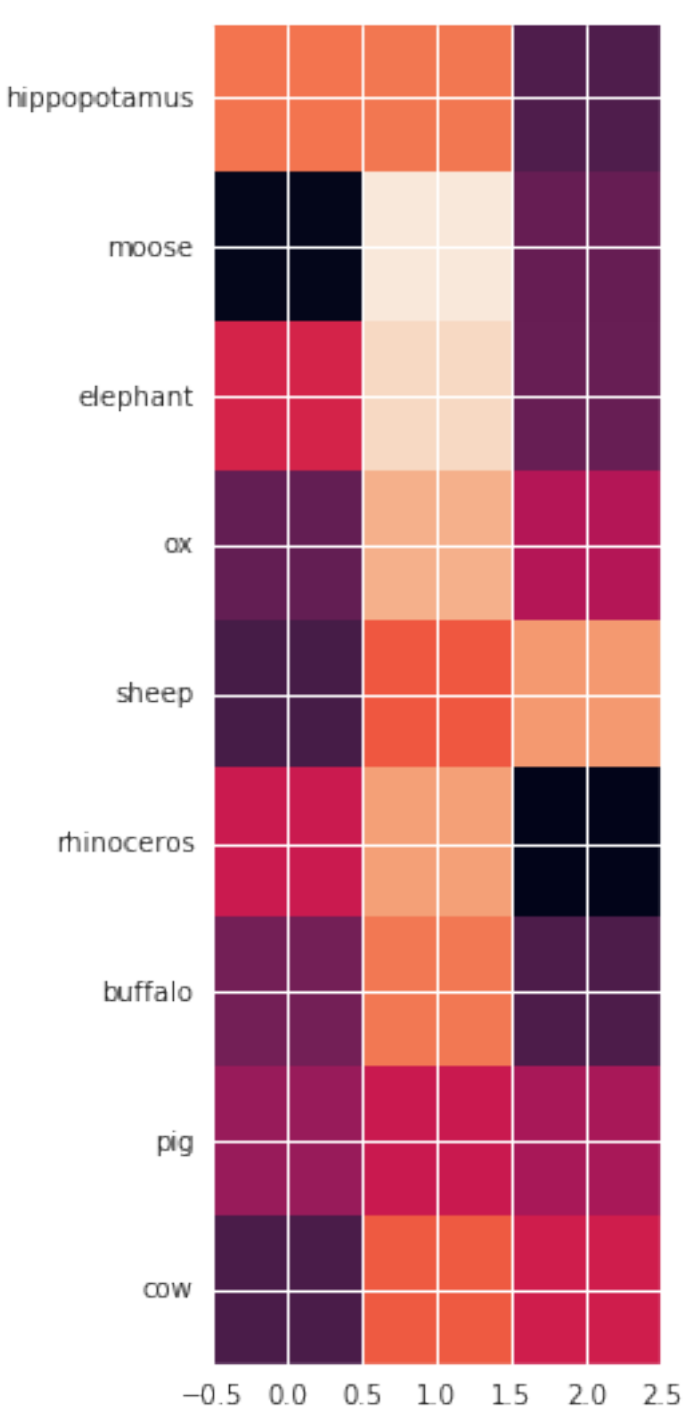
Do dimensionality reduction.

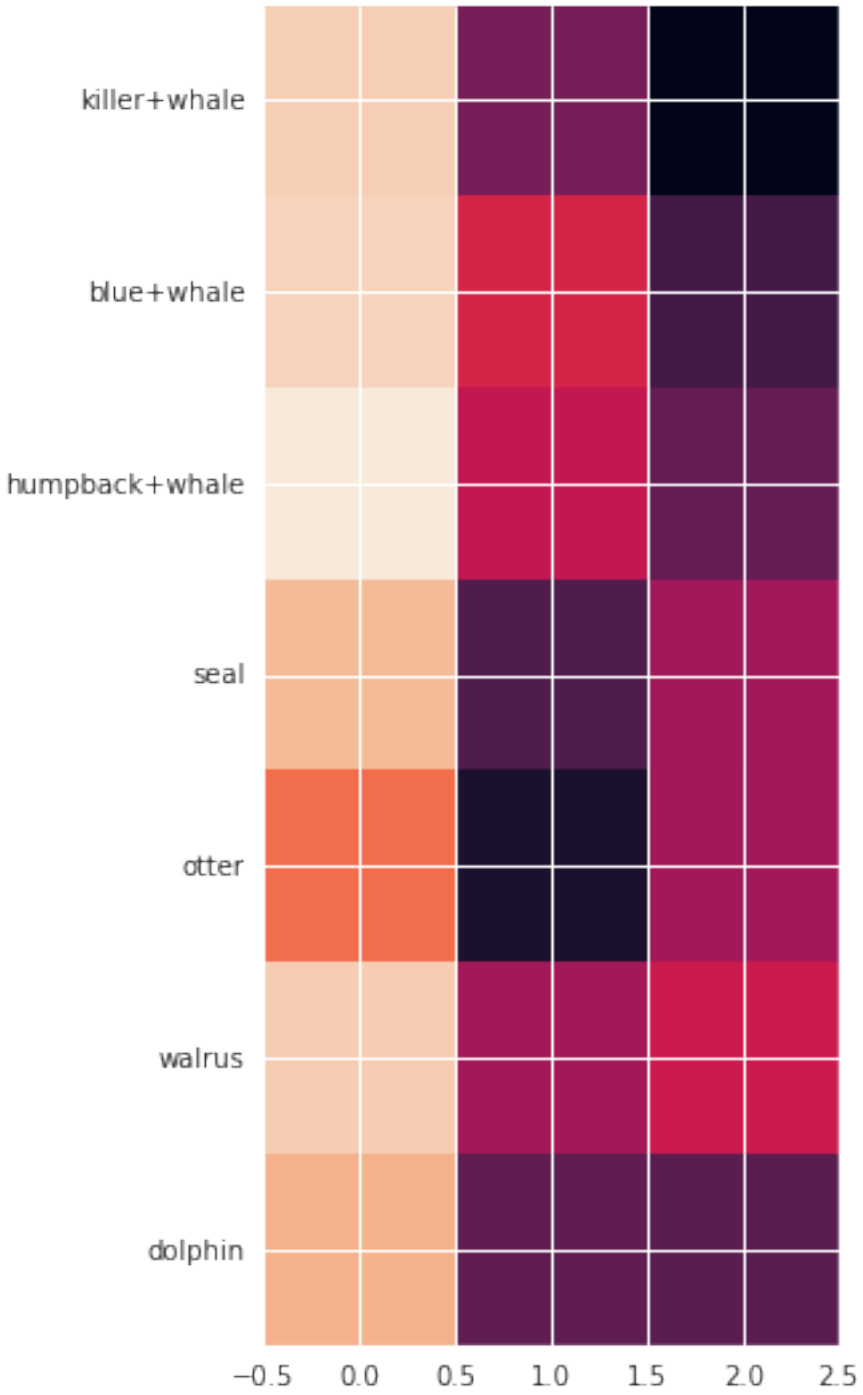


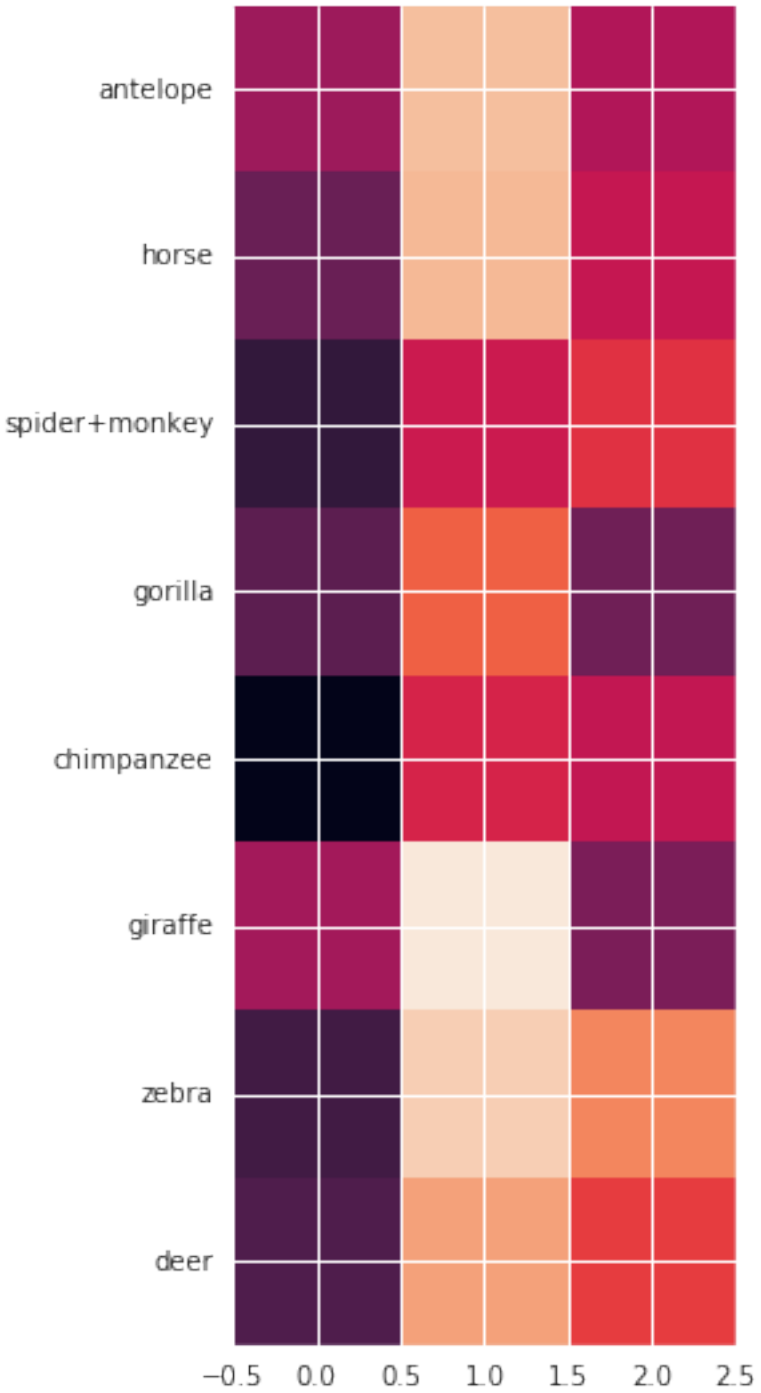
Use k-means on the MDS-reduced data. This is doing much better than PCA with 3 components. This seems to hint that the animal data is more affected by pairwise distances between points than by improper rotation.

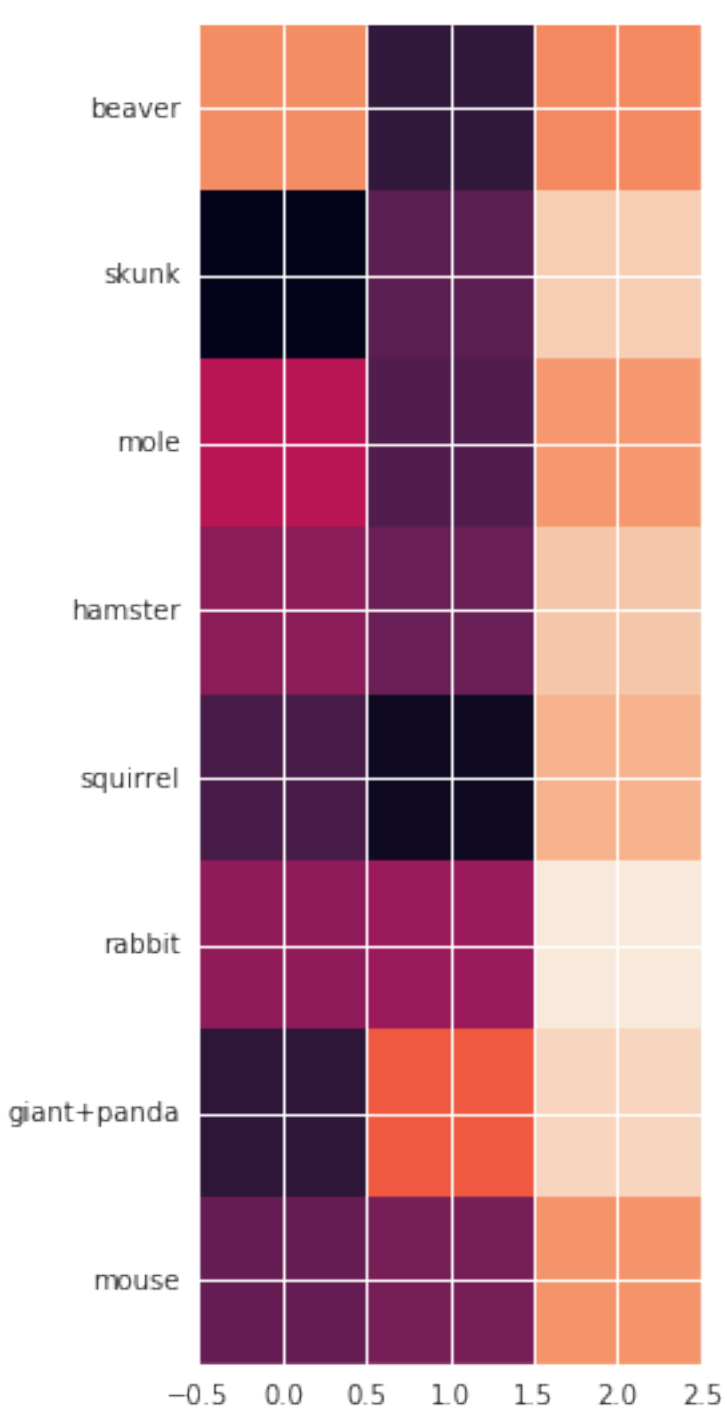
Still the MDS clustering is not perfect, and it seems strange that it groups spider monkey with horse and zebra.

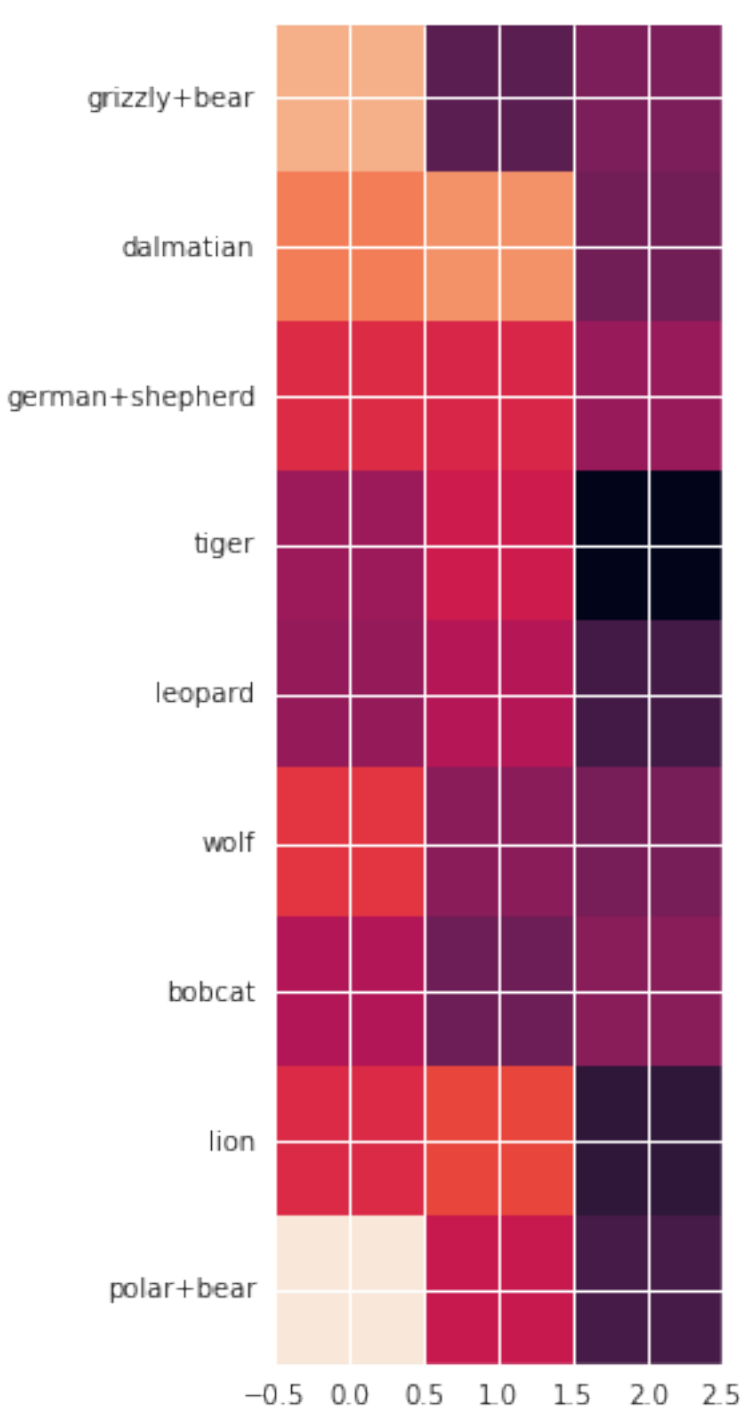

persian+cat
siamese+cat
fox
bat
chihuahua
rat
weasel
collie
raccoon
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


hippopotamus
moose
elephant
ox
sheep
rhinoceros
buffalo
pig
cow
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


killer+whale
blue+whale
humpback+whale
seal
otter
walrus
dolphin
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

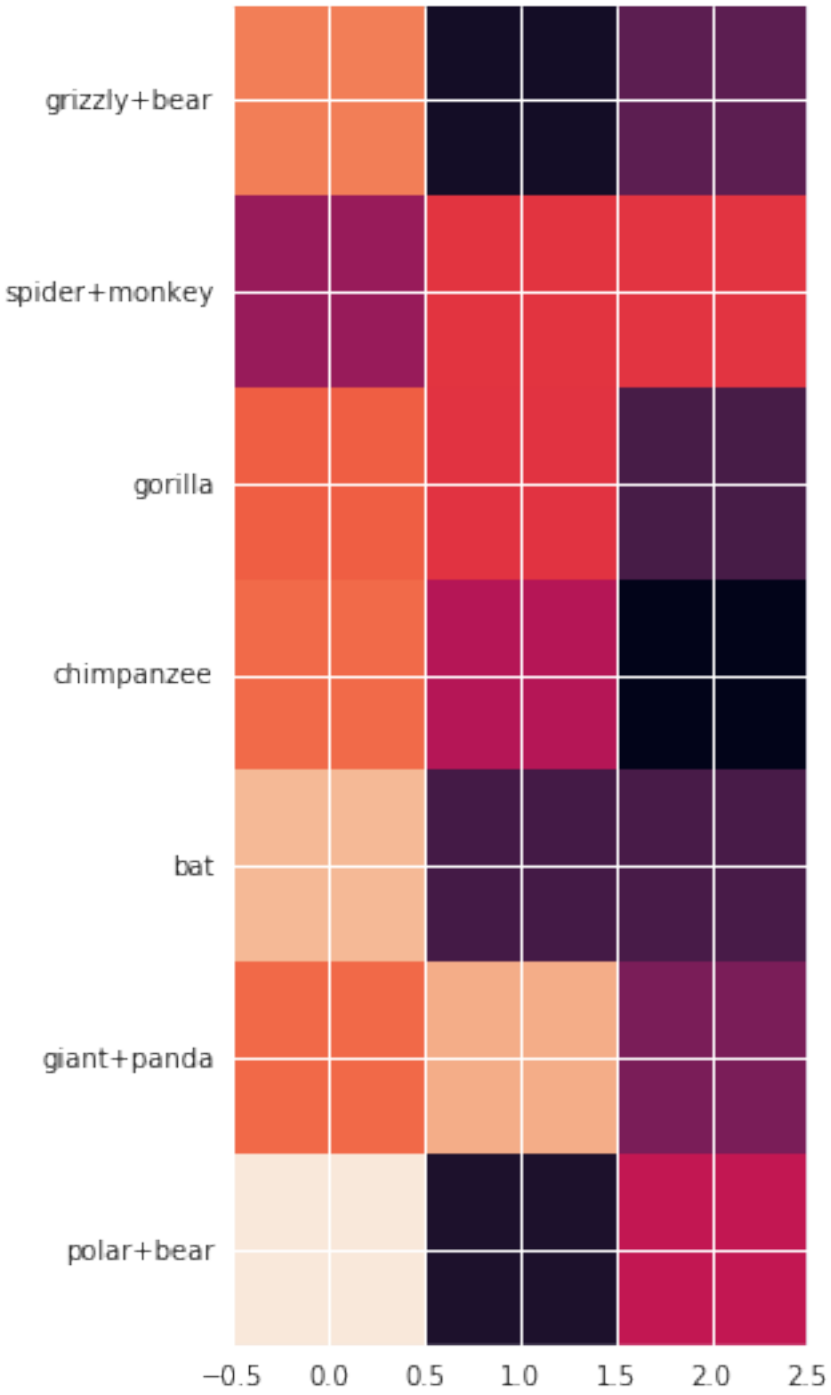

antelope
horse
spider+monkey
gorilla
chimpanzee
giraffe
zebra
deer
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

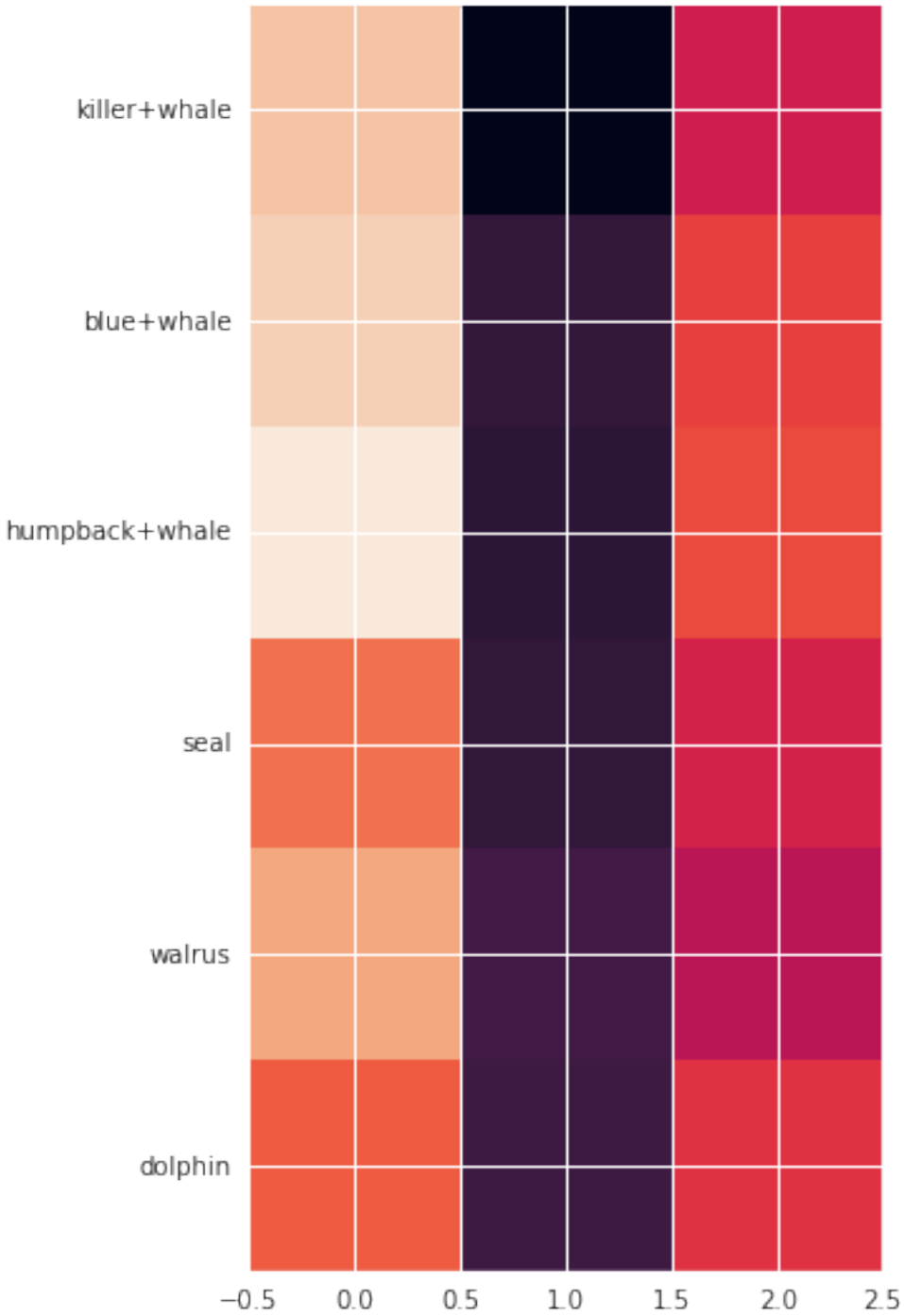

beaver
skunk
mole
hamster
squirrel
rabbit
giant+panda
mouse
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

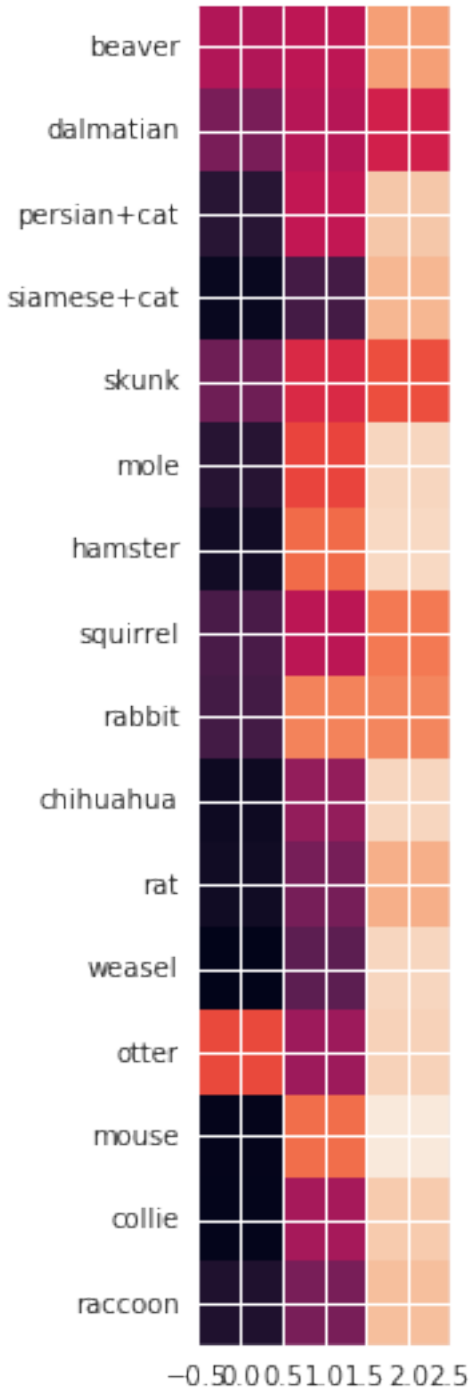


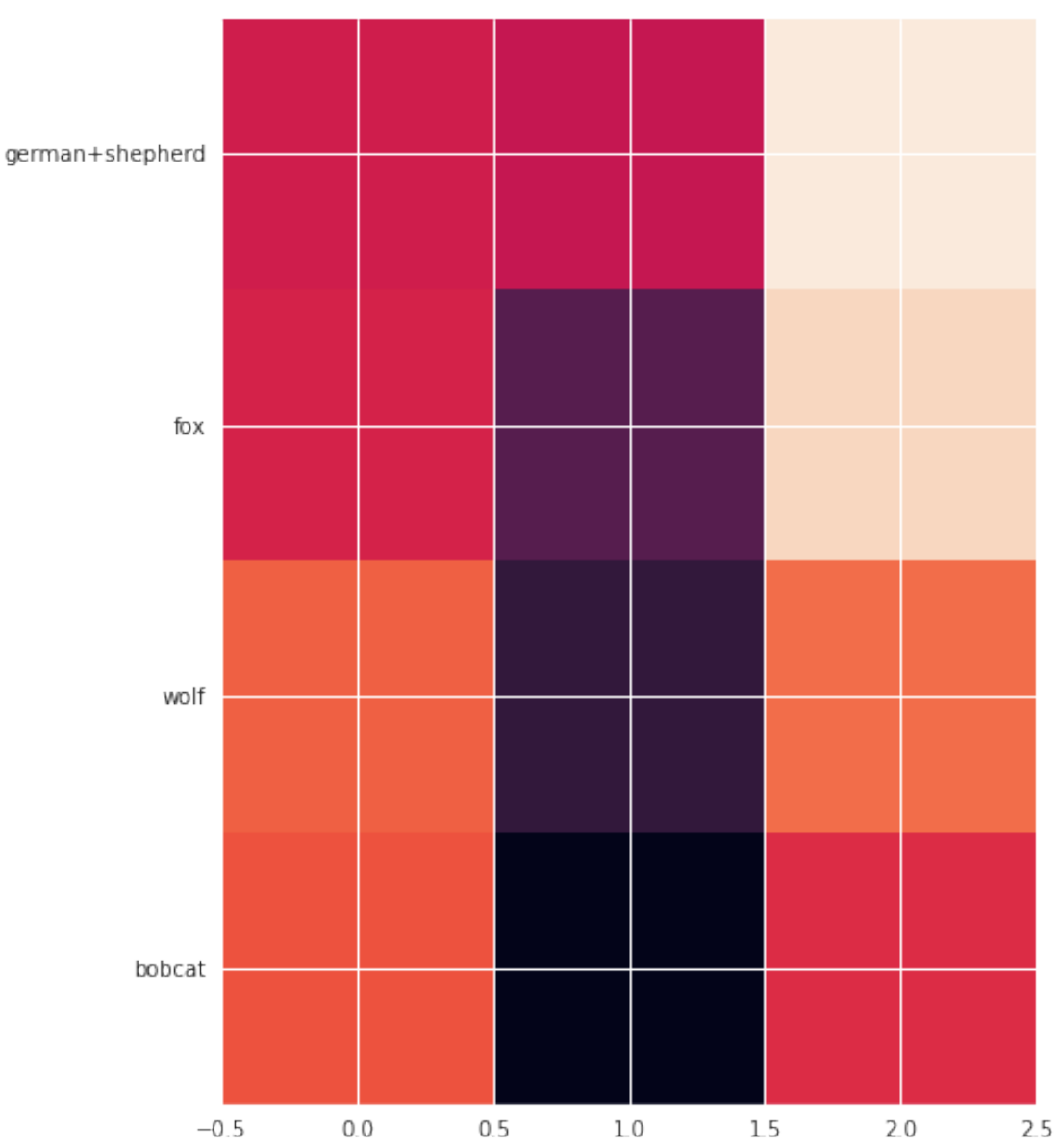
Since the previous attempts has been unsuccessful, it is reasonable to think that a linear embedding will not work given the shape of the animal data. Try a 3 components of locally linear embedding of the data.

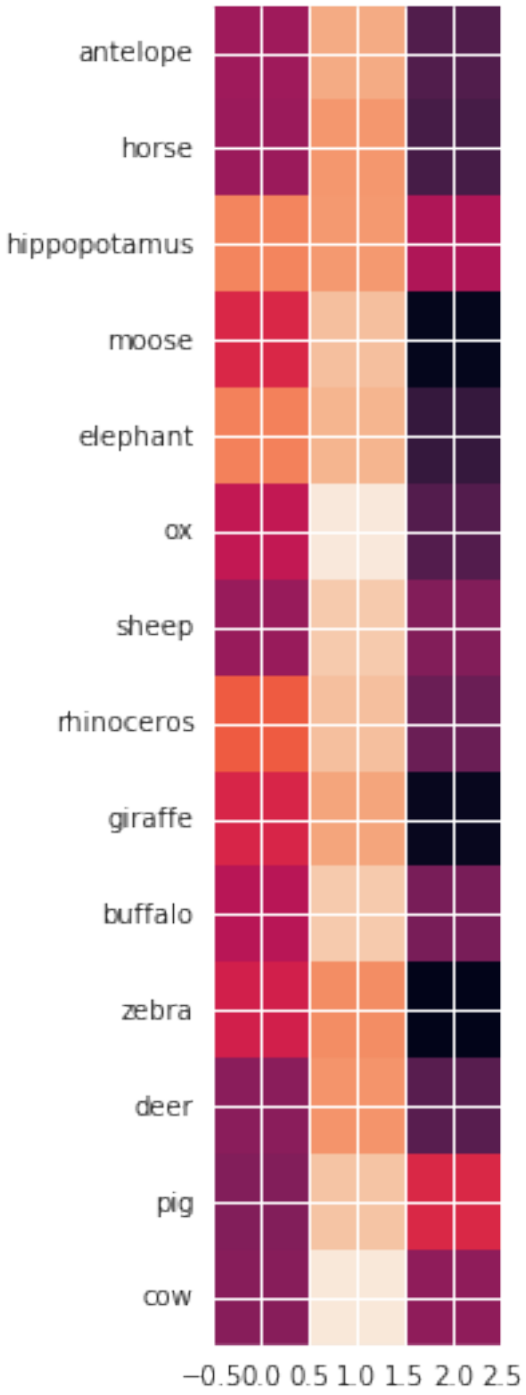
Try to cluster reduced representation.

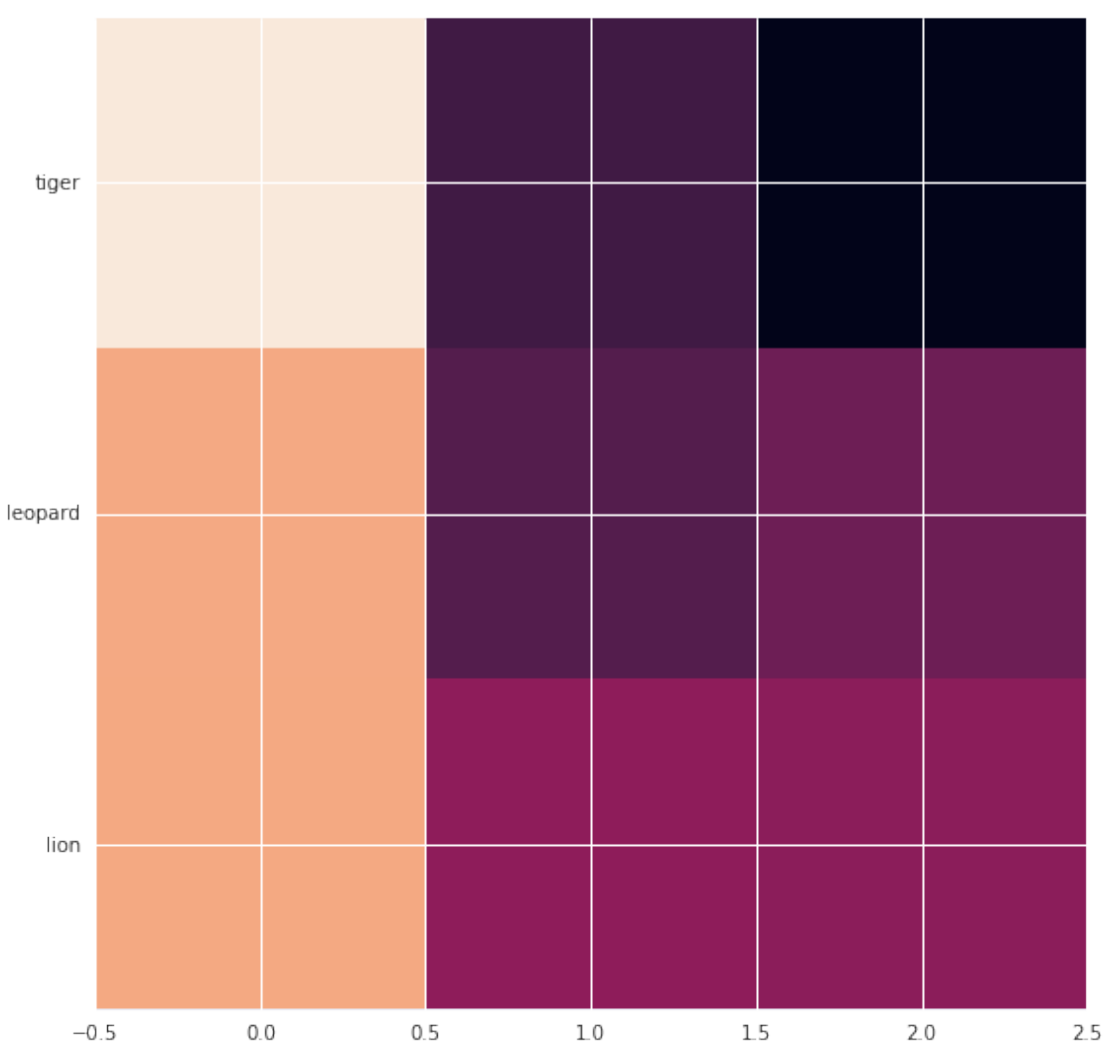

grizzly+bear
spider+monkey
gorilla
chimpanzee
bat
giant+panda
polar+bear
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


killer+whale
blue+whale
humpback+whale
seal
walrus
dolphin
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


beaver
dalmatian
persian+cat
siamese+cat
skunk
mole
hamster
squirrel
rabbit
chihuahua
rat
weasel
otter
mouse
collie
raccoon
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


german+shepherd
fox
wolf
bobcat
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

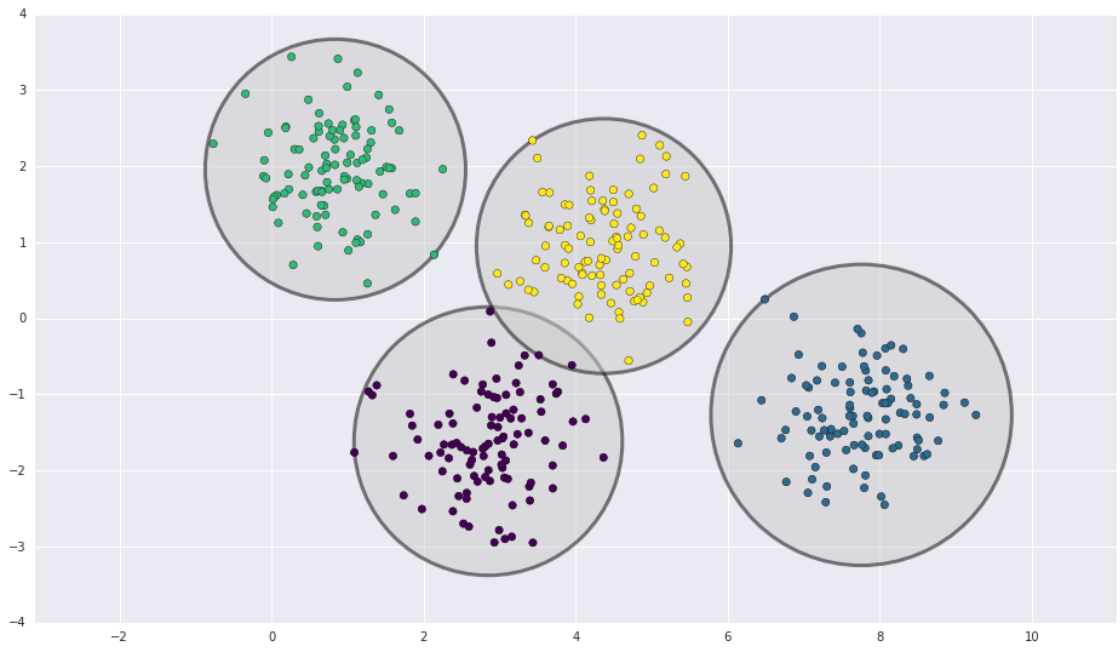

antelope
horse
hippopotamus
moose
elephant
ox
sheep
rhinoceros
giraffe
buffalo
zebra
deer
pig
cow
−0.5 0.0 0.5 1.0 1.5 2.0 2.5



Using 3 components, this method did a much better job of clustering animals in a way that matches my inuition. But for a large number of components, like 8, the preditions one again become unintuitive.

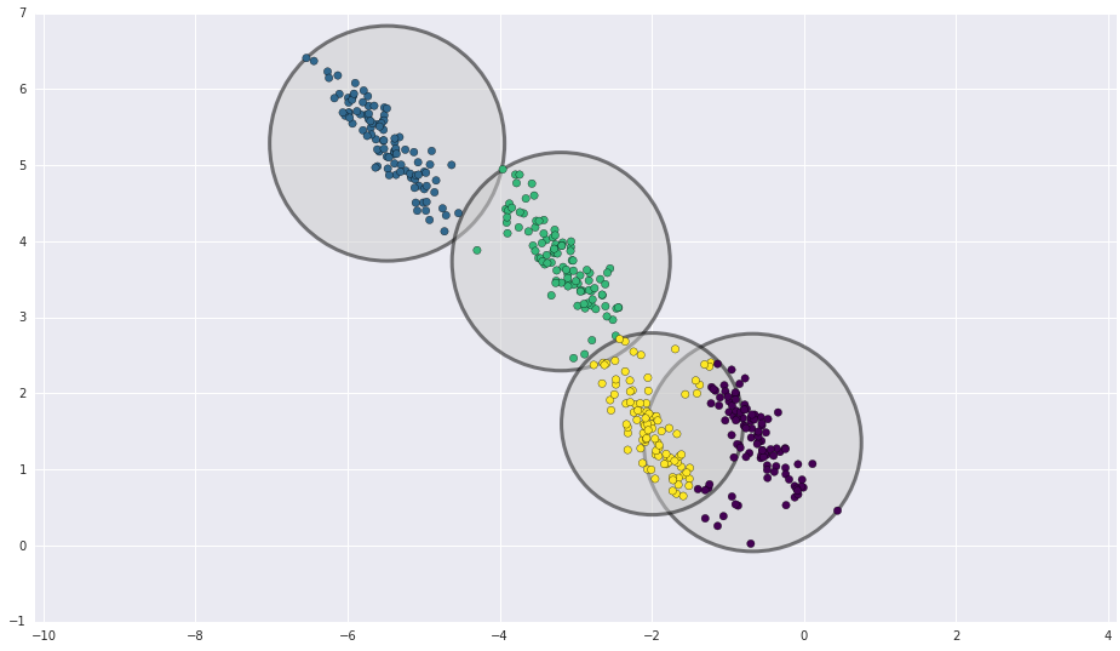
Lastly, try GMM to compare with k-means clustering. k-means is nice and simple, but its non-probabilistic nature and its use of a single distance from a point to a cluster limits its possible use cases. It can lead to poor performance in many real-world situations. GMM can be viewed as an extension of k-means.o

Let's view k-means again.



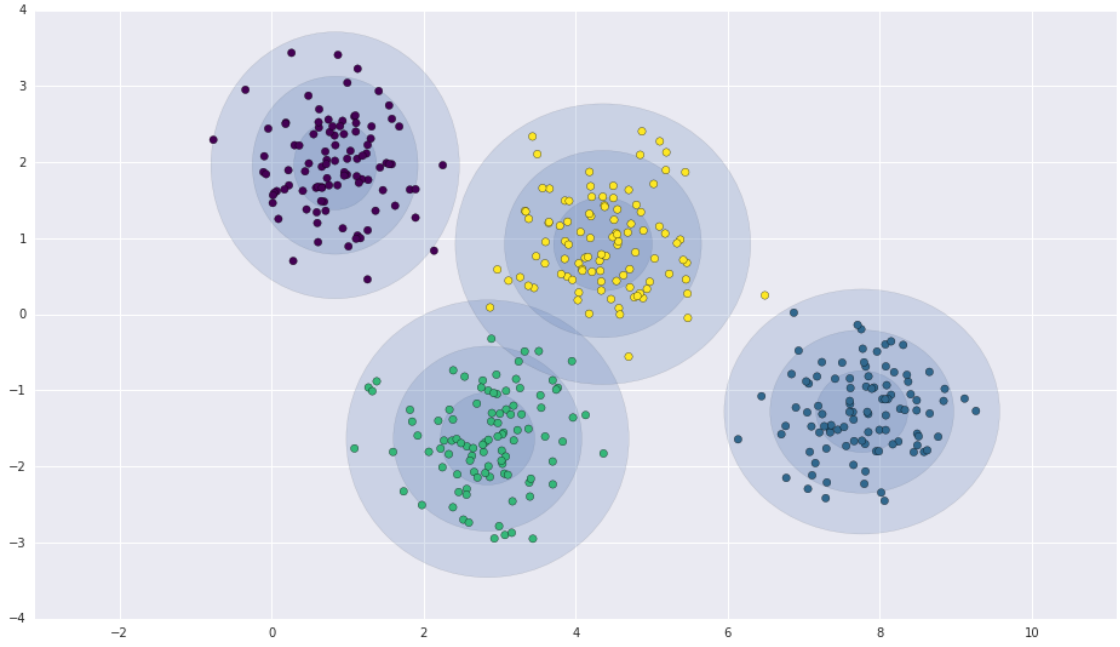
Plotted below is a case where k-means will struggle. It cannot change the shape of its clusters from circles.

K-means does pretty well on the animal dataset anyway, because of the shape and high-dimensionality of the data.



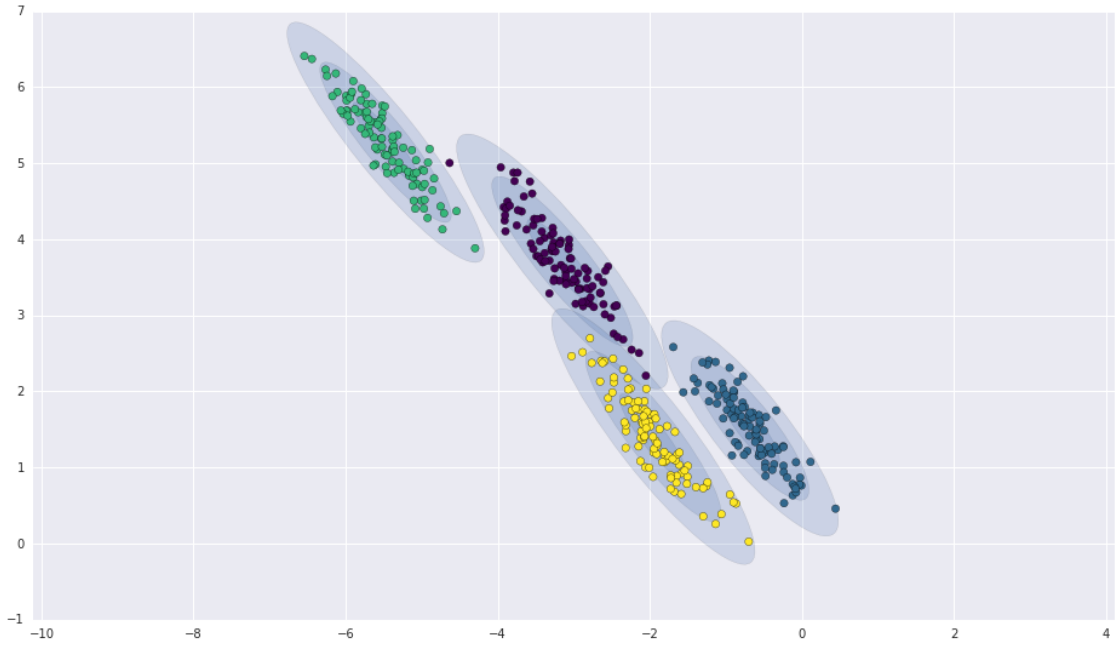
Test the Gaussian Mixture model for a simple case.

Plot the probability distributions around the clusters.

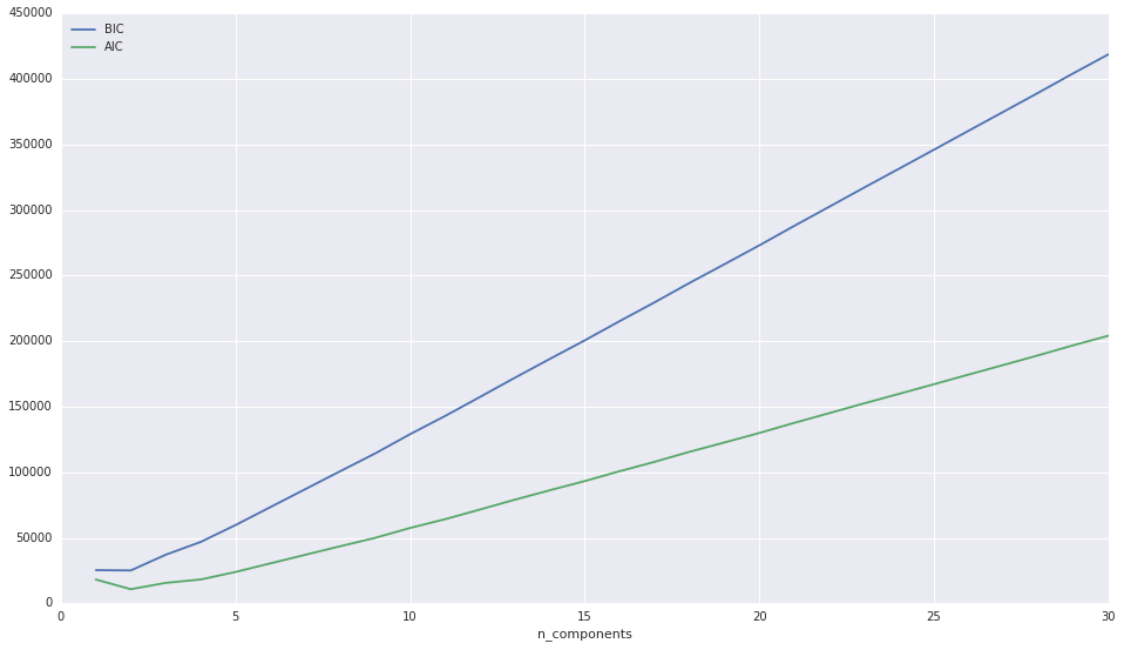


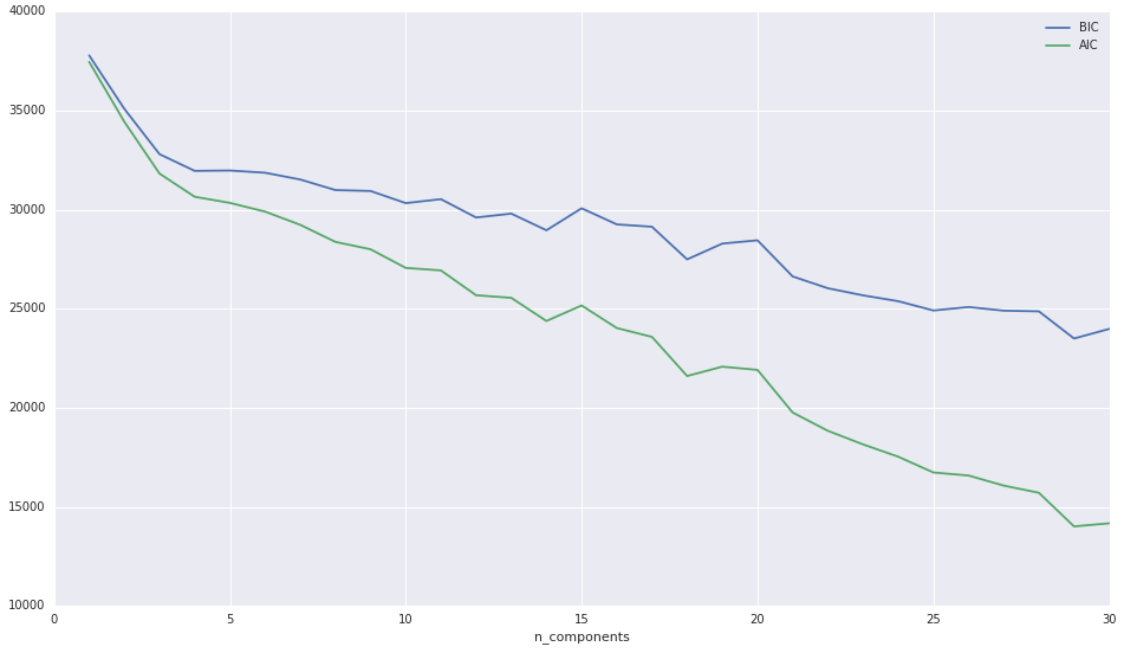
Do for problem with non-circular clusters.

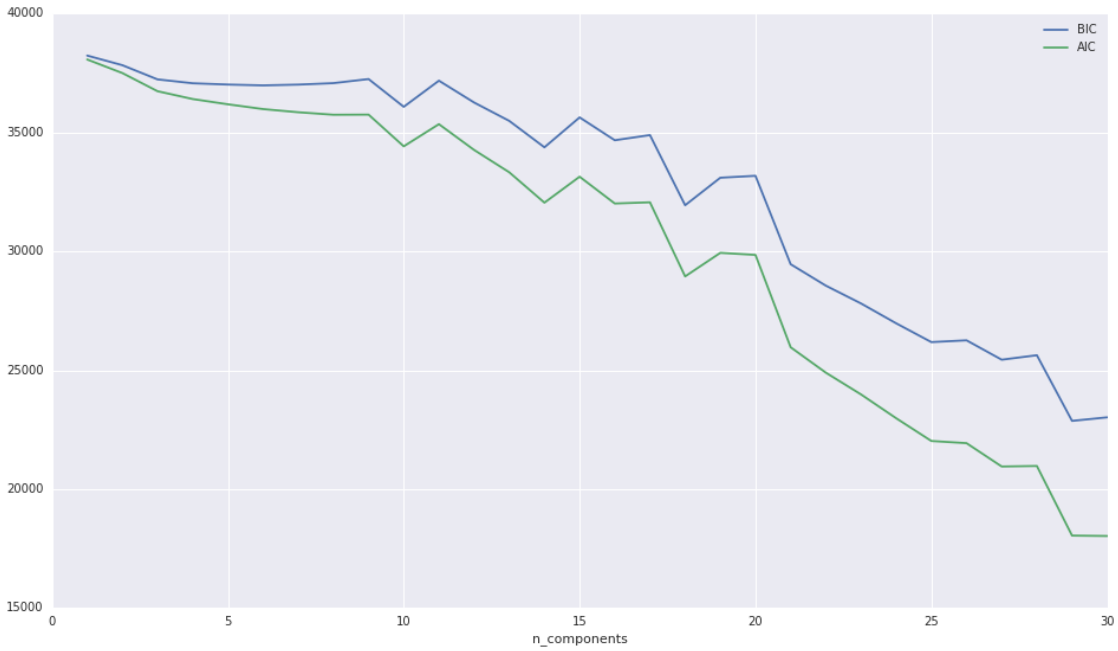
Do for problem with very non-circular clusters



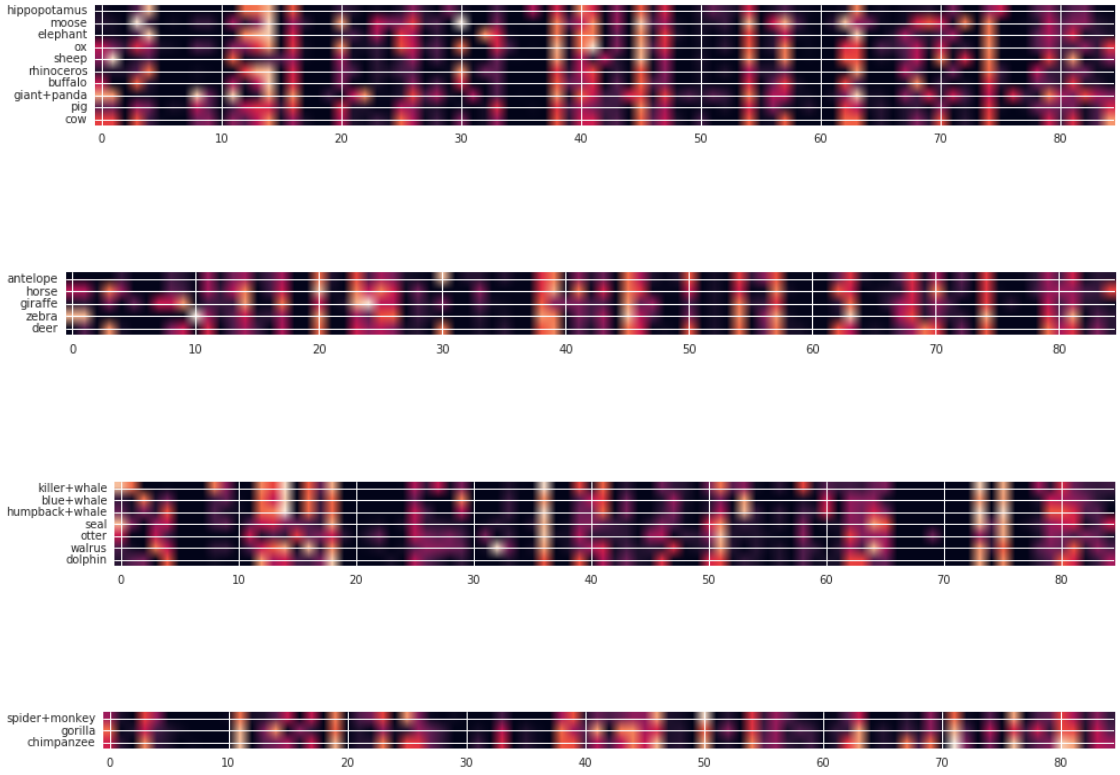
AIC-Akaike information criterion and BIC-Bayesian information criterion are plotted below. These tests are inconclusive for the different types of GMM (full, diag, and spherical)

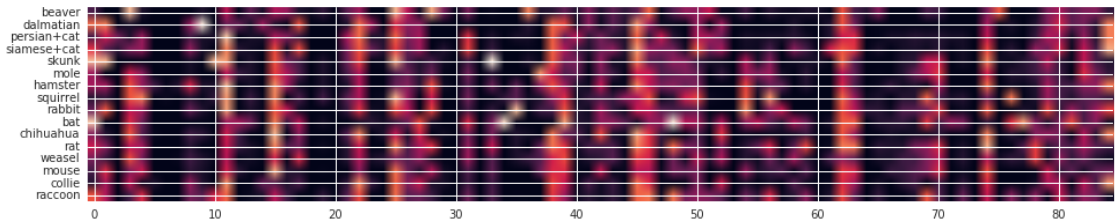

BIC
AIC
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
0
5
10
15
20
25
30
n_components


BIC
AIC
40000
35000
30000
25000
20000
15000
10000
0
5
10
15
20
25
30
n_components



Try grouping animals based on 6 components of GMM. The results are pretty reasonable!
It would most likely be redundant to do something like PCA on top of this.





References

[1] D. N. Osherson, J. Stern, O. Wilkie, M. Stob, and E. E. Smith. "Default probability". Cognitive Science, 15(2), 1991.
[2] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. "Learning systems of concepts with an infinite relational model". In AAAI, 2006.