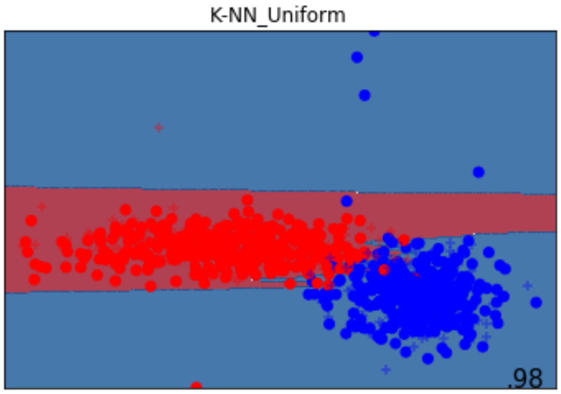
Ross Spencer

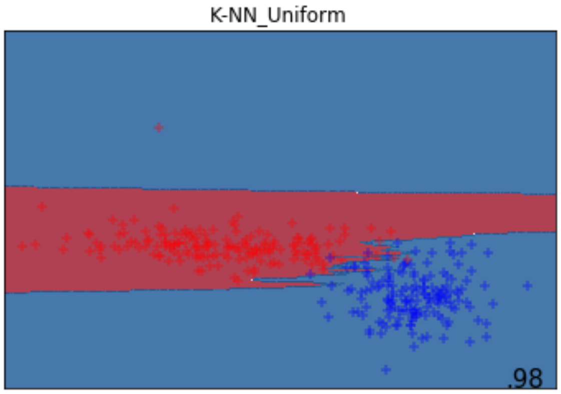
HW2

Dataset	Accuracy of PGC	Accuracy of PGC (Diagonal)	Accuracy of KNN
2D Training (M = 20)	97.27273%	97.27273%	**97.57575%**
2D Training (M = 21)	98.48484%	**98.78788%**	**98.78788%**
2D Training (M = 5)	97.27273%	97.27273%	**97.87878%**
7D Training (M = 20)	**100%**	**100%**	**100%**
7D Training (M = 21)	**100%**	**100%**	99.39394%
7D Training (M = 5)	**100%**	**100%**	99.69697%
HS Training (M = 20)	11.81818%	3.03030%	**81.51515%**
HS Training (M = 21)	10.90909%	3.33333%	**82.42424%**
HS Training (M = 5)	12.12121%	2.72727%	**79.39394%**

When I first began using KNN, I played around with the 2D dataset since it was the only one I could visualize and found that K=3 provided the best estimate (K=2 shown for comparison, n=4 was overfitted and had pockets).

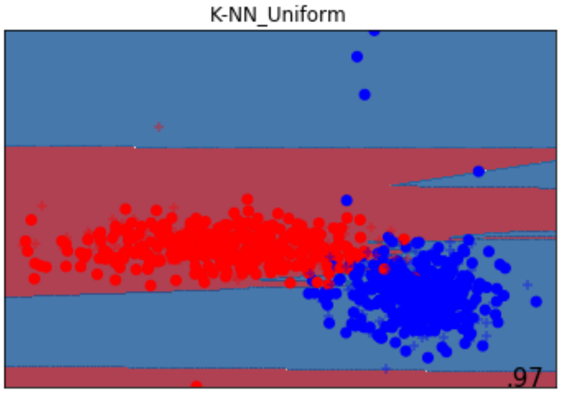
K=3





(K=3 with just the validation points displayed)

K=2



2. When training the probabilistic generative classifier, how does the full covariance compare to diagonal covariance in performance for each of the data sets? Why? When training KNN classifier, what happens as you vary k from small to large? Why?

For both the 2D and 7D datasets, the probabilistic generative classifier performed around the same since our covariance matrices were small and didn't need as much data as HyperSpectral to estimate. For the HyperSpectral dataset, the diagonal probabilistic generative classifier definitely underperformed the one with full variance (~3% vs ~11%), probably since it completely ignored the possibility that some of the 147 variables weren't independent and didn't have enough data to make up for that.

For the KNN, trying multiple different values of k showed that if k is too small accuracy goes down and if k is too large accuracy goes down as well. I've included some of my tests for 2D and HS, leaving out 7D as 7D still had 96.969696% accuracy at k=300 and was more robust to the change in number of neighbors. The value k=3 seemed to provide the best tradeoff among smaller k values for both 2D and HS, but I included k as large as 300 in my table for fun. If the number of neighbors is small, our KNN will be too easily affected by noise. On the other hand, as the number of neighbors becomes too large, the "smoother" the decision regions will be, which could get rid of some of the underlying detail in our distribution.

K	Accuracy of KNN (2D)	Accuracy of KNN (HS)
1	96.96969	80.90909
2	97.27273	76.66666
3	97.57576	81.51515
4	96.66666	79.39393
15	96.06060	80.90909
100	92.12121	72.72727
200	91.81818	63.03030
300	91.51515	60.30303

3. Determine which classifier(s) you would use for each data set and give an explanation of your reasoning. Hint: This should incorporate some discussion based on results from cross-validation.

I was getting a weird error that my covariance matrices in the HyperSpectral probabilistic generative classifier were singular and had a determinant too close to 0, so clearly in this case it's probably not the best fit without rescaling or standardizing my data. The full covariance matrix for that case is 147x147, and I think there just wasn't enough data in the training to properly estimate it. Somehow the diagonal covariance case faired even worse. The KNN classifier still did fairly well in this case, and far above the other 2, however!

For the 2D dataset and the HyperSpectral dataset, I think my cross validation table shows that KNN is the best classifier for these datasets. Similarly, cross validation shows that the PG is the best performing for the 7D dataset, even though KNN is very close.