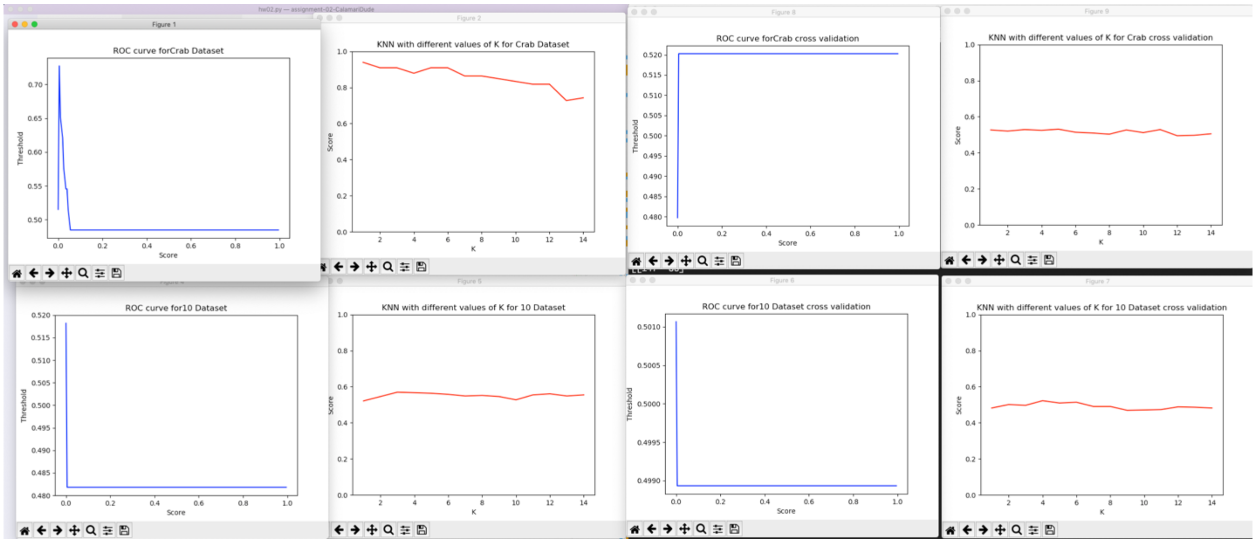
```
********* Crab Dataset **********
Classification report for MAP
           precision    recall  f1-score   support

       0.0       1.00      1.00      1.00        34
       1.0       1.00      1.00      1.00        32

   accuracy                           1.00        66
  macro avg       1.00      1.00      1.00        66
weighted avg       1.00      1.00      1.00        66

Confusion Matrix (Rows are true, Prediction are columms)
[[34  0]
 [ 0 32]]



Best k value =  1
Classification report for KNN
           precision    recall  f1-score   support

       0.0       0.94      0.94      0.94        34
       1.0       0.94      0.94      0.94        32

   accuracy                           0.94        66
  macro avg       0.94      0.94      0.94        66
weighted avg       0.94      0.94      0.94        66

Confusion Matrix (Rows are true, Prediction are columms)
[[32  2]
 [ 2 30]]
********* 10 Dataset **********
Classification report for MAP
           precision    recall  f1-score   support

       0.0       0.90      1.00      0.95       171
       1.0       1.00      0.89      0.94       159

   accuracy                           0.95       330
  macro avg       0.95      0.94      0.95       330
weighted avg       0.95      0.95      0.95       330

Confusion Matrix (Rows are true, Prediction are columms)
[[171   0]
```

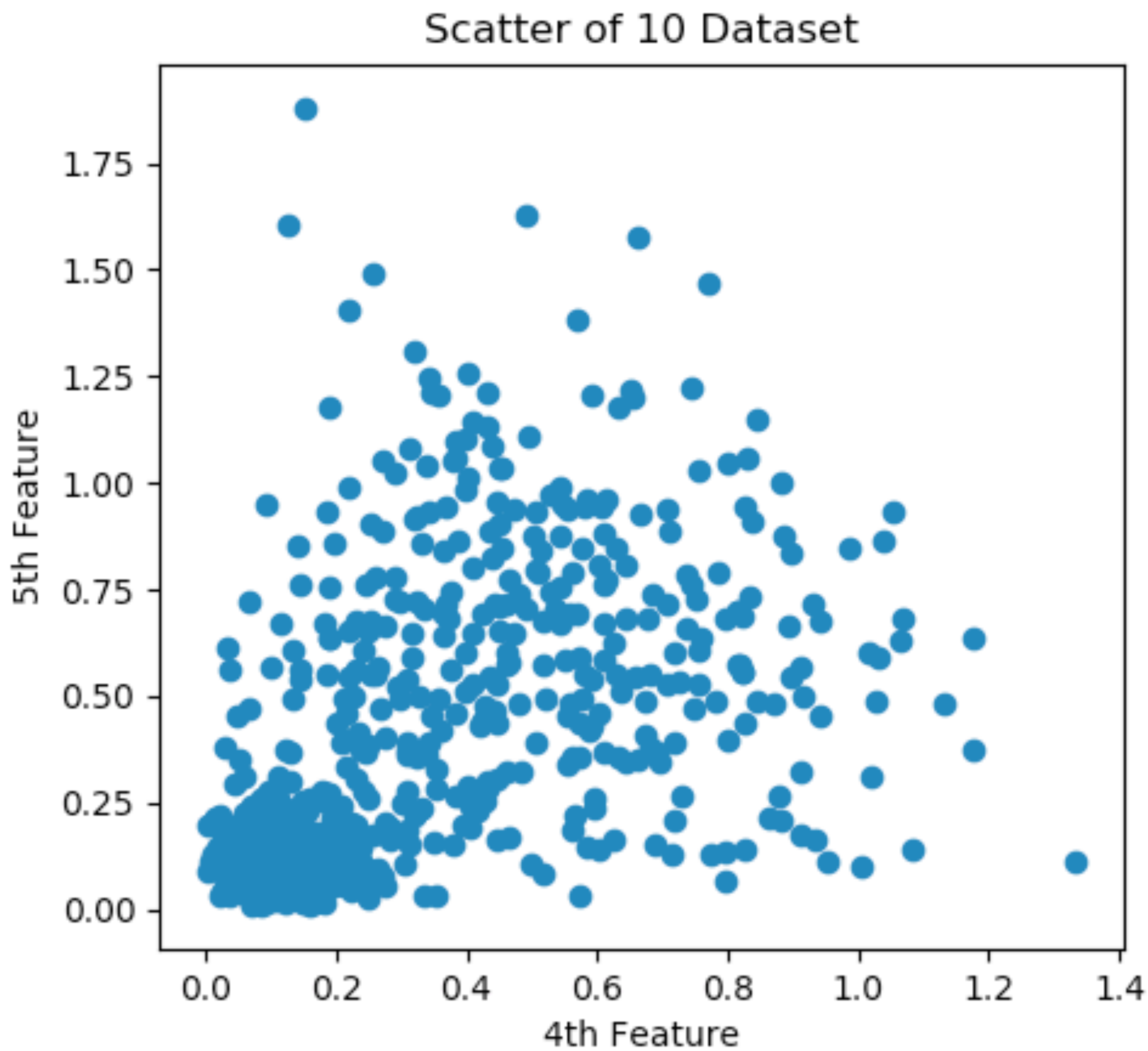
1.
2. The problem encountered was that the covariance matrix was singular, so all I did was perturb it by identity * epsilon to make it singular and invertible again.

3. When K becomes large, the results become very underfit, and the algorithm score goes down. You can see this in the KNN for the crab dataset. In the 10d datasets all the scores were around guess, so there is no noticeable difference in scores since they are all near .5



4. Knn gives worse classification becomes if you look a the clusters in this diagram, the data isn't entirely separable, knowing the variation of the two classes, which MAP does, it helps it determine, "oh, if one class has very small variance and a data point is outside this cluster/not near the mean, then it most likely not from this cluster." Whereas KNN doesn't know what to do since it sees the datapoints as being mixed and therefore

starts guessing.



5. I would use the MAP estimator for the 10D dataset AND the Crab dataset since in our cross validation it is clear that KNN overfits the data. When we do a 30/70 split (training/validation) on just our training data we see that knn does significantly worse at around <.6 for both datasets. When this kind of difference occurs between splits, it is an indicator of overfitting. Therefore, MAP is more robust to our dataset, and KNN is not as robust. So we should use MAP for both datasets.

6.

6A

we wish to maximize $\hat{\lambda} = \underset{\lambda}{\arg\max}\, p(\lambda \mid X)$

derivative of product rule for log $\Rightarrow \frac{d}{d\lambda}\left(-n\lambda - \underbrace{\sum_{i=1}^{n} \log(x_i!)} + \log(\lambda)\sum_{i=1}^{n} x_i +\right.$

$0 = -n + \frac{1}{\lambda}\sum_{i=1}^{n} x_i$

$\lambda n = \sum^{n} x_i$

$\boxed{\lambda = \bar{x}}$

maximizing $\log(x)$ is same as max $\max(x)$

6b $P(x' \mid X) = \dfrac{P(x' \mid \lambda) \cdot P(\lambda \mid x)}{P(\lambda \mid x', x)}$

$$= \frac{\dfrac{\cancel{\lambda^{x'}}\ \cancel{e^{-\lambda}}}{x'!} \times \dfrac{(N+\beta)^{N\bar{x}+\alpha}}{\Gamma(N\bar{x}+\alpha)}\ \cancel{\lambda^{n\bar{x}+\alpha-1}}\ \cancel{e^{-(n+\beta)\lambda}}}{\dfrac{(N+1+\beta)^{N\bar{x}+x'+\alpha}}{\Gamma(N\bar{x}+x'+\alpha)}\ \cancel{\lambda^{N\bar{x}+x'+\alpha-1}}\ e^{-(N\cancel{+1}+\beta)\lambda}}$$

$$P(x' \mid x) = \frac{(N+\beta)^{N\bar{x}+\alpha}}{x'!\,\Gamma(N\bar{x}+\alpha)} \times \frac{\Gamma(N\bar{x}+x'+\alpha)}{(N+1+\beta)^{N\bar{x}+x'+\alpha}}$$

$$= \left(\frac{N+\beta}{N+\beta+1}\right)^{N\bar{x}+\alpha} \left(\frac{1}{n+1+\beta}\right)^{x'} \times \frac{\Gamma(N\bar{x}+x'+\alpha)}{x'!\,\Gamma(N\cancel{\bar{x}}+\alpha)}$$

=(N-1)!

Sources:
https://www.statlect.com/fundamentals-of-statistics/Poisson-distribution-maximum-likelihood
https://www.youtube.com/watch?v=iCxrimJzzDo