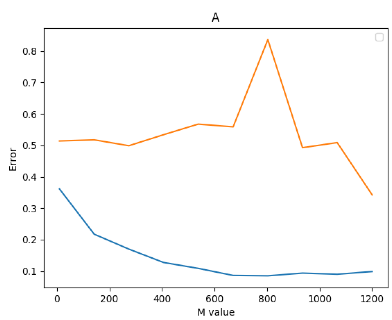
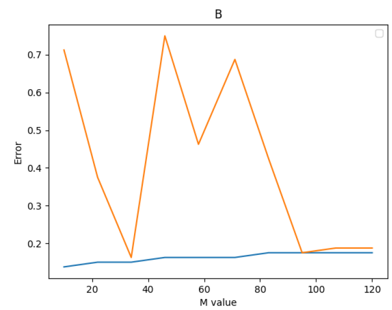
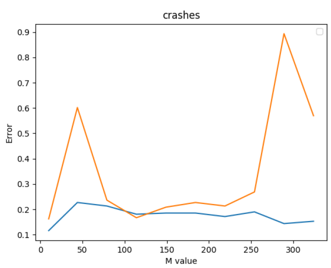
Christian Zinck

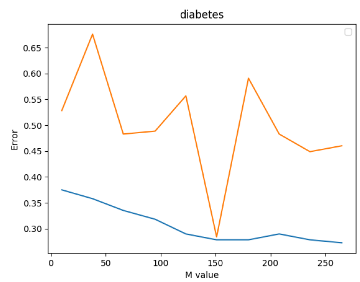
PP3

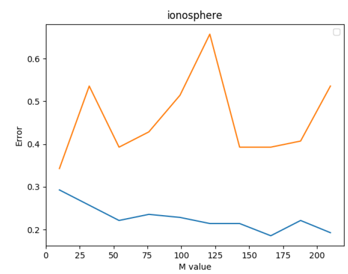
Task 1:





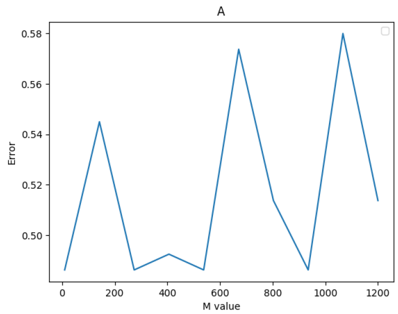


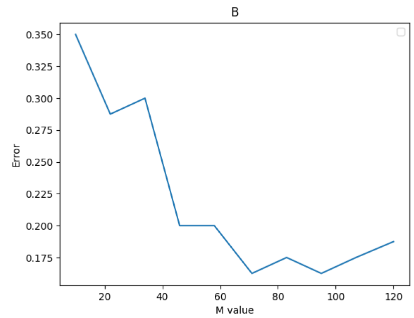


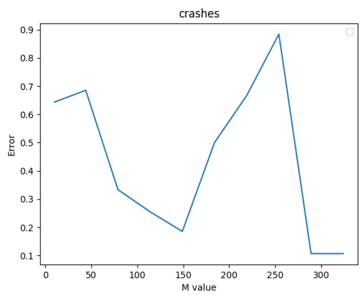


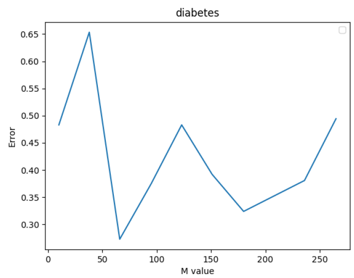
In the above graphs, the orange line is the learning curve for the discriminative model, Bayesian Linear Regression, and the blue line is the learning curve for the generative model using a shared covariance matrix. The generative learning curve is fairly typical: test error decreases with an increase training set size, except in dataset B because B is derived from a model with different class covariances. Therefore, adding more data from B will not improve accuracy because it cannot be modeled well by the shared covariance generative model. The discriminative learning curve is so wild that, despite many hours of debugging and testing, I am still convinced is wrong. Accordingly, the only trend I can glean from the graphs is that the generative model performed consistently better than the discriminative model.

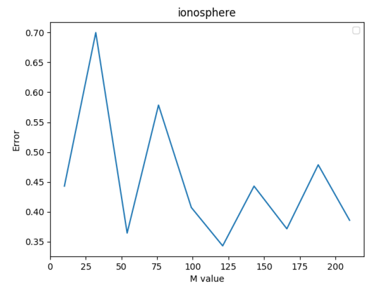
Task 2:









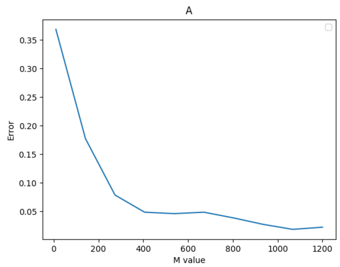


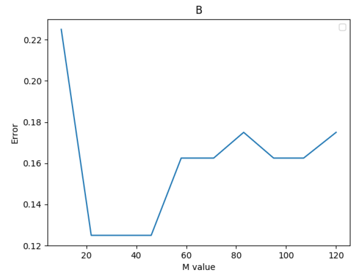
Optimal Alpha Values

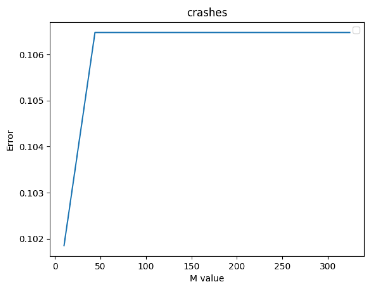
A	B	crashes	diabetes	ionosphere
0.15879911107	0.000000147223	0.734776306102	0.540341155471	0.787959097163

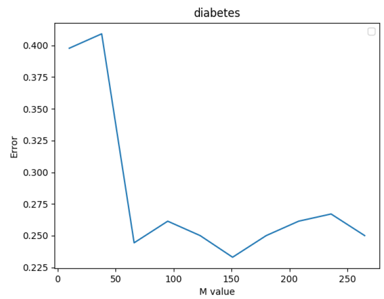
Although my learning curves for Bayesian linear regression with model selection are also disastrous (except B!), I do believe that I have found accurate alpha values. The alpha values of datasets A and B are significantly lower than the alpha values of the UCI datasets because A and B are linearly separable, while the others are not. I couldn't help but think that the extremely low regularization used for B is responsible for the somewhat normal looking learning curve, while all the other more heavily regularized learning curves are all over the place, so I ran my task 1 with alpha = 0.000000001 instead of alpha = 1 and got much better looking curves, which in general had higher error initially, but quickly overtook the generative model, so it seems that the best alpha value for my implementation is as close to 0 as possible to avoid singular matrices. This is supported by the fact that as training set size increases, the value of alpha decreases.

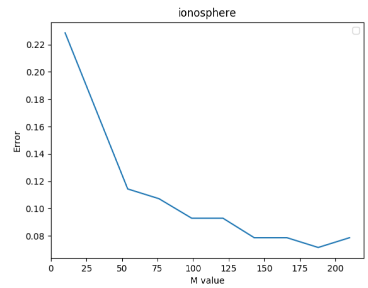
Task 3:











In general, Gaussian Process Classification is outperformed by the other methods on small training set size, but achieves minimal error across all models for each dataset with a very steep learning curve. I am confused by the crashes learning curve, but I must assume that it immediately achieved an approximately optimal error rate with a training set size of 10, which is rather strange to me. I think that GPC is supposed to be computationally slower than the other 2 models but that was not the case at all. It takes about as much time as the generative model, if not less, both of which are much faster than Bayesian linear regression (potentially due to a faulty implementation). Also, I know that I wasn't supposed to do GPC on dataset A, but it didn't take long at all and had such a nice learning curve that I had to include it.