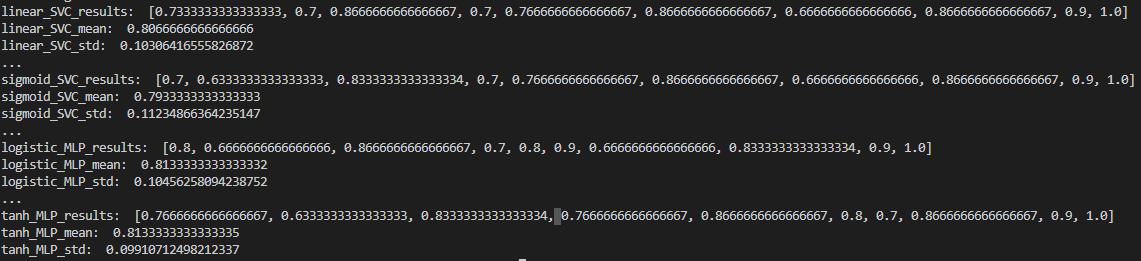
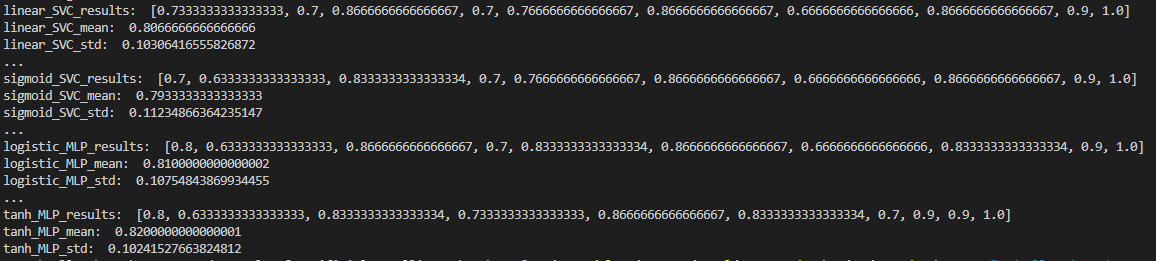
Nick Moore

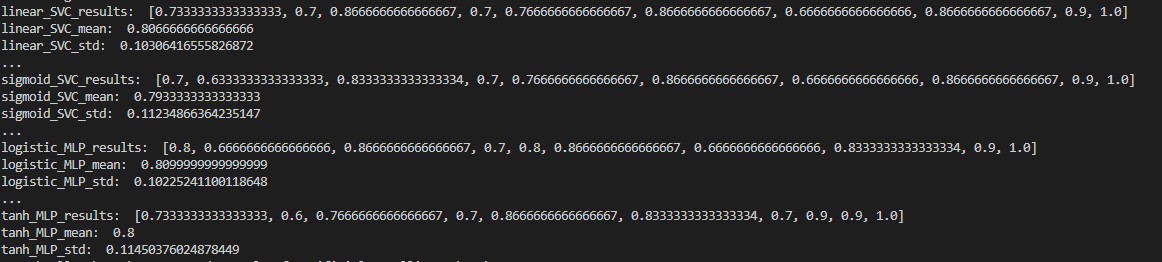
1390458

For my analysis I ran my code 6 times. 3 times with shuffle off for a base and 3 times with shuffle on.

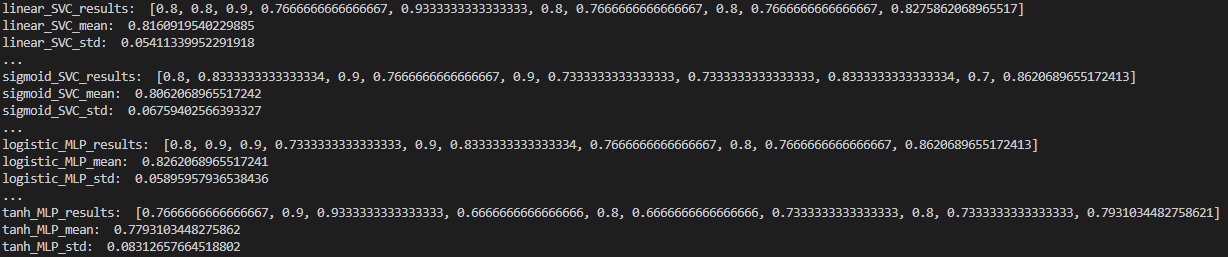
3 times with shuffle off:

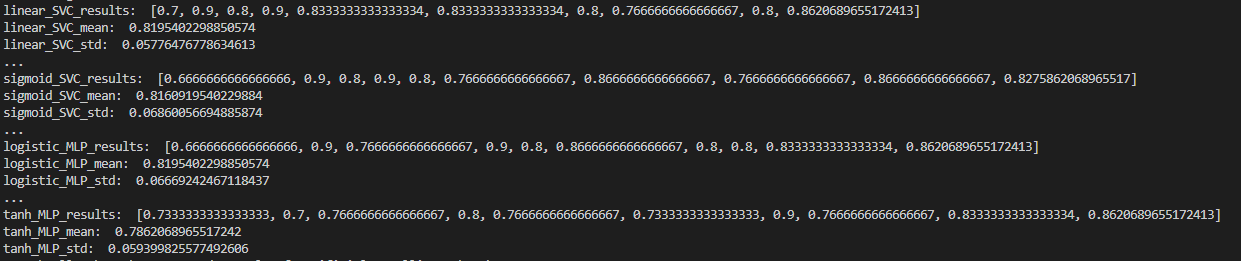


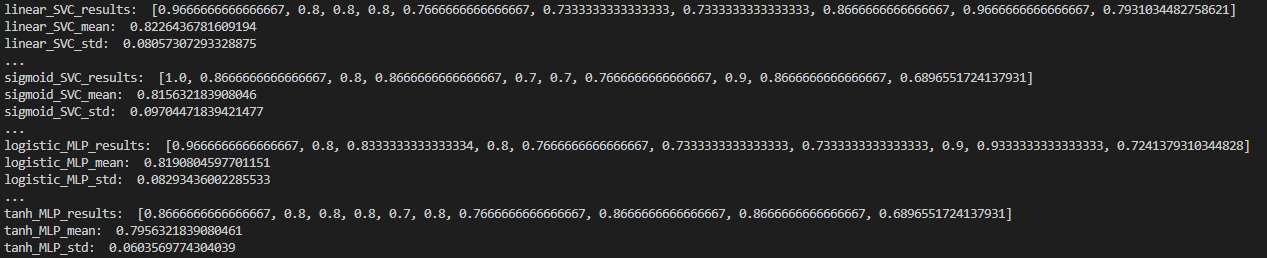




3 times with shuffle on:







Looking at these results its clear that the SVC is much more consistent than MLP however the MLP can have higher or lower results than SVC depending upon the order of the given dataset and the random seed used during calculations. From these results the best SVC obtained was 0.8226 while the best MLP was 0.8262. Now what was interesting in the none shuffle runs Tanh MLP consistently outperformed Logistic MLP with only 1 exception. However when the order of the dataset was shuffled Logistic MLP greatly outperformed Tanh in all runs. While for SVC it followed the same trend in both runs with Linear beating Sigmoid in all instances.

Now I believe that the reason why Linear SVC is better than the Sigmoid SVC is due to the data set we are looking at. There are only 2 possible results for prediction. Either the person died or they did not die. Without results falling in between these possibilities a curved line does not hold much advantage over a straight line. In fact the more curved the line becomes the less accurate it would become. I tested with rbf and found that the probability of success dropped drastically due to this linear inclination with sigmoid being a far straighter and better approach to this data set than rbf.

Now for MLP I am torn on how to feel about it. Overall it took a much longer time to train and generate results. However this time investment could either lead to a better or worse result than SVC. Yet again in this case sigmoid which creates a much straighter line than tanh consistently came out better. However as I stated the generation of the NN did not make this 100% consistent only giving it a strong inclination towards this trend. I would say about 70-80% of the time Logistic MLP will be better than Tanh MLP for this data set. Now this I imagine is due to what the NN put more weight into deciding its line or plane.

One import thing to take away from this is the importance of the random seed being used when generating the NN for MLP. As we can see it can dictate if the NN if it is more or less accurate than its SVC counterpart. So I believe if given enough computation power and time to check a multitude of different seeds to find what maximizes the results MLP will in this data set always come out on top. However if you do not have the time then SVC will give a quicker result given that you have a understanding of what your data’s prediction tendency is going to need to be.

It is also important to remember that just cause you got a good result doesn’t mean it can not be improved by changing various variables. Above I talked about changing the random seed for the MLP NN generation. But also as we can see in these results what order you feed your algorithm to train it can also have a large impact on how well it performs. Take Linear SVC for example with out shuffling it got a result of 0.8066 while the best result from shuffling the dataset for Linear SVC got us 0.8226. There are a lot of components to training the NN and you can never be certain that one specific way of giving it data will be best until you conduct through testing of the different possibilities. Which is always why it takes so long to train Ais. The different formulas and approaches greatly affect the end result even if in this specific data set it was only a difference of about 0.05 from worst to best.