Analysis

After making some graphs, comparing the various predictors to each other and to how those specific predictors determine the grouping of A or B seems pretty unsuccessful. After making a few scatterplots, the data seems extremely difficult to separate in a binary fashion due to how dense it is and how points in different groups are very frequently overlapping the points in group A and B are. While there are some areas that have pretty high densities of only group A or B points, but having such a high degree of overlap in multiple sections of the graphs makes it seem like the data is non-separable at all or very difficult to separate

Methods

To train and evaluate models, I used first, a train test split on the data and split it into the training set and test with test size of .2 Additionally, all predictors were z scored in order to normalize all of our points. Three separate models were used to on this data to determine which one is the best. First was the support vector machine model, which functions by projecting our data onto a predetermined amount of dimensions and using a slack variable to add some leniency so achieve linear separability or near-linear separability For the hyper tuning of the SVM model we hyper-tuned three parameters: the C value, the gamma value and the kernel We wanted to check which was better between two types of kernels, linear and radial basis function.

The linear kernel functions by not projecting the data onto higher dimensions and works best only if the data is linearly separable in two dimensions, and it functions similarly to logistic regression. The radial basis function kernel projects the data onto higher dimensions and even possibly infinite dimensions until it is most linearly separable. The gamma value determines how much of an influence two data points have on each other and the higher the value, the lower the influence. For gamma hyper tuning we used the values of: 0.001,0.01, 0.1, 0.5, 1,2,5. The third hyper parameter we tuned for the SVM model was the C value which is the slack variable or the margin of error essentially. With a smaller C value we are allowed more incorrect predictions and a larger C value raises the penalty for incorrect predictions. For C, the grid search chose 50, for gamma it chose .01 and for kernel it picked the radial basis function, which makes sense as the data was definitely not linearly separable in two dimensions. For logistic regression, I did no hyper parameter tuning and ran it as is.

For KNN the only parameter I hyper tuned with grid search was the n_neighbors and what that value is responsible is for how many neighbors of a certain class a point must have to be classified in that class, in our case, A or B. The grid search result was that the ideal number of neighbors was 18, and that makes sense seeing how in some areas, tightly clustered our data is and how in other areas, it is not tightly clustered. Additionally, I think this is due to the fact that the data seems so linearly inseparable on the 2d plane.

Results

Overall, the three models performed very similarly according to the metrics I took, which was quite surprising to me, but one model's metrics stood slightly above the other two models'.The model that performed the best was the SVM model and it had 78.25% accuracy on the training data and 76% accuracy on the testing data. This tells us our model was slightly overfit, but overall still is good accuracy in the field. The ROC AUC or area under the curve was 86% for the training data and 84% for the testing data which shows that the model has strong predicting power and is doing a good job at predicting whether a data point is in group A or B. The other two models performed similarly well but both the logistic regression and KNN models performed about a percent or two worse on every single metric and that is why the SVM is the best model to use for production. To add on to why SVM is best for this type of data, it became clear as explained earlier that after plotting out parts of the data that it didn't seem linearly separable and the advantage of SVM is that it can project the data into as many dimensions as necessary until it is linearly separable so we can have the strong prediction power on unseen data from our training set. To summarize, the model I would recommend to be used in prediction is the SVM model, primarily because it is the model that was able to predict on unseen data with the best and due to the fact that I think it can do very well predicted on further unseen but similar data because of the ability of the SVM model to project data into further dimensions.

Reflections

I think the most interesting thing I learned from the assignment is how the SVM model has the power to make data that looks like points randomly placed on a graph and randomly sorted into A&B groups and extract some meaning from it. Even though the other models performed similarly, SVM was a little bit better than both of them and that was very interesting for me. I didn't really struggle with anything specifically, I feel like the SVM part of the project was just like the classwork we did and that the KNN and logistic regression parts were very similar to what we did in 392 so I just had to brush off some of those skills and use them. At the beginning of the assignment, I kind of struggled with figuring how the models would perform when all I had was the data and graphs because it seemed so all over the place and random.