Part 5

KNN or K Nearest Neighbors is a non-parametric algorithm that attempts to describe the data space by finding the k number of neighbors that most closely resemble the data point we want to find the result of. Since the model is lazy, we only do calculations at runtime. For classification we can take a point and find the k closest neighbors. Then using majority votes, we see of what type is the closest and then assign that type to the data point. Logistical regression on the other hand uses the mean of the data that describes the other points. The main downsides of KNN is that if our sample size or dataset is too large we end up with a large calculation time since each point has to be checked for its distance from the data value.

Decision Trees are similar to KNN in the sense that the algorithm is non-parametric but goes about making a prediction through a series of choices that were made when the algorithm was created. Decision trees are great for finding the choice that best separates two types of data. This is chosen through the use of entropy and information gained for each choice. Additionally, one can end up with an arbitrarily large number of leaf nodes and by pruning the tree we can cut back on how many decisions have to be made on a single data point to receive an answer. Since we attempt to make the best choice at each level the algorithm is considered greedy and can take a while to train on large data sets. Two major downsides to decision trees are that they are very prone to outliers as well as loss of information on continuous data. When we have outliers in the data the tree is very likely to split that data off and form a branch of their own making the tree as a whole less efficient. When it comes to continuous data, we are forced to use less than or greater than signs meaning that we lose the ability see gradual change in the data. The closest we get to seeing gradual change Is by splitting the data multiple

times along the line, but even then, we are making the tree less efficient and still can't fully capture the data.

Both PCA and LDA are dimensionality reduction techniques that are vital to ensure that our data doesn't contain to many features that don't help describe our data. PCA or Principal Component Analysis is useful for when you have multiple features that have a linear relationship. If we were to use 10 features that are all related similarly, we are increasing our number of dimensions unnecessarily. LDA or Linear Discriminant Analysis is a similar linear separation technique that is both supervised and takes in the labels of the classes in order to ensure maximum separation between different classes. These are valuable techniques that lets us go further with smaller amounts of data as well as putting it into a form that is better understood by model.