David Park

CS 4375.003

SVM works by creating a hyperplane that separates the data into certain classes. It finds the best line or hyperplane by finding the data points that are closest to a possible hyperplane. With these data points, called support vectors, SVM then calculates the largest distance between a possible hyperplane and the support vectors called the margin. Once the margin is maximized, the hyperplane that corresponds to these specific margins becomes the optimal hyperplane. Problems start to arise when a dataset is too complicated to separate linearly. Kernel functions directly resolve this problem by transforming the data into some higher dimension space depending on the data. My impressions of the strengths of SVM are that it is very flexible. SVM can be both used for classification and regression which makes it very convenient and straight forward. However, it should be noted that SVM also does not work well when the data has lots of noise or when features are overlapping.

Random forest works by creating multiple decision trees which are then used at the end to predict a class. Decision trees are nodes that branch at each decision creating a tree-like structure. Each of these trees are separate from each other and generate an output. Together these trees create the forest. The algorithm for random forest chooses its prediction for classification by selecting the most voted output and for regression by the average voted output. When comparing the simple decision tree and random forest to adaboost and xgboost, it is important to note that they work in fundamentally different ways. Adaboost and xgboost work by boosting, which is when a single learner gets inputted the training data. Then it outputs to a subsequent learner that focuses on the errors of the previous learner. Random forest works by bagging, which is when a set of similar machine learning algorithms run in parallel each with a different subset of the training data. The strengths of decision trees are that they are very simple and fast, but

they are possibly not as accurate as the other techniques because it is the most basic. For the random forest, it is generally more accurate than adaboost because it creates full decision trees, while adaboost creates stumps rather than full trees. However, this also means that it will most likely take longer to run which is evidenced in my run times. Xgboost has the advantage of running very fast compared to adaboost and random forest. However, depending on the data, it may be less accurate compared to the other two. This was not evidenced in my data since my data seemed to fit the algorithm of xgboost very well.