# SVM and Ensemble Methods

## Support Vector Machines:

Support vector machines are generated by an algorithm that finds a hyperplane in any dimensional space that divides the data given to it. This hyperplane is regularized to find the optimal hyperplane by ensuring that the margin between the hyperplane and the nearest vectors is maximized. The support vectors are just the data points that fall on the edge of the margin, and therefore define the shape/orientation of the hyperplane. This method allows the creation of a linear decision boundary and fits a line that could be used for continuous regression. Mathematical tricks can also expand the model to fit more complex data as well.

A kernel is a mapping of the existing input data to some n-dimensional space. The magic of the kernel is that the function for finding this maximalized margin in SVM requires the dot product of vectors in the data. The kernel function serves as a mathematical trick that saves the algorithm from actually computing the data in n dimensions, and instead applies the conversion to the result of the dot product. SVM can then take this mapped dot product and simulate finding a hyperplane in the higher dimensional space. In the case of a linear kernel, we are just finding the dot product normally. In the case of something more useful like a polynomial kernel, we are mapping the dot product result to something that may not originally be linear.

The most interesting kernel, the Radial Kernel, is able to apply a mapping to the dot product that simulates the relationship between points in infinite dimensions! With the versatility of the kernel and the ingenuity of the overall method of SVM, we have great advantages and some disadvantages. While the algorithm is computationally complex, that complexity leads to its versatility. Its use of a decision boundary means that SVMs deal with outliers well, and work better than logistic regression on well-separated data. Overall its wide applications make it seem like a very strong algorithm that you would really be able to use (for both classification and regression) in any situation.

## Ensemble Methods Analysis:

Ensemble methods are a series of machine learning methods that combine or refine other categories of machine learning algorithms. Ensemble focuses on the ideas of bagging, which is repeatedly sampling data to reduce variance, and boosting, which reduces bias by weighting algorithms and the harder to place data so that a group of models will focus on making those accurate. Different algorithms due this in different ways, though decision trees are a common one. Decision trees will be refined to such precision by so many different things that they will no longer be easily interpretable, but can get surprising accuracy.

Random Forests are the prime example of decision trees in ensemble learning. This method utilizes bagging to focus on subsets of predictors in varying orders so that when it makes a large number of different trees, it can find certain trees that may be more accurate than others. Random

Forests are built to avoid always picking the most influential predictor first, to increase the differences between trees.

XGBoost is a fast algorithm that works by using decision trees as well, using multiple cores on a computer so that it can handle extremely large data sets while still being fast. XGBoost uses boosting over bagging, so that the models learn one after the other instead of all at once, such as in Random Forests.

CaretEnsemble is a package that is built for combining multiple models, not just decision trees. A number of different models can be used, so that the best of both linear regression and K-Nearest-Neighbors can be combined, for example. It will consult all the models given to determine the most likely result of a test.

Each of these have their pros and cons. Random forests spread out to find the best possible way to approach data and focus there, while XGBoost finds the more difficult problems and focuses on being able to improve the confidence of those edges. Random forests can get large and out of hand, and can fall to the standard sorts of decision tree faults, but in its improved results it is also no longer interpretable.

XGBoost can more easily fall to overfitting, though random forests are also susceptible. While XGBoost is commonly hailed as an extremely accurate method, some things still cannot be decided, and there are instances where it can't provide anything more helpful than other methods. XGBoost is also built for large amounts of data, and since it focuses on prior models errors, outliers in that data can become a problem.

CaretEnsemble is more unique as it uses many different user-defined models to pull together, so it is difficult to list an exact weakness, as it could combine logistic regression with random forests, and reduce both of their pros and cons. However, in combining the assorted models, there is weighting and skew problems that using too similar data models would provide. It becomes a matter of user conscientiousness.