Madison Chamberlain

I tested a few different methods for generating a better prediction of the ForestCover data. I created a table with the accuracy and time it took for each method I tried, to determine which method was best. I tried PCA with various PCA numbers and Ks. I found that with PCA = 4, over 99% of the variation in the data could be explained, so there is no need to use a higher PCA. I tried different Ks as well. I found that K=20 generated a very similar accuracy to k=100, so to minimize time I used K=20. Next I tried the RandomForest method, and I tried this both in, and not in parallel. I found that the predictions using and not using parallel were very similar, but the parallel method took far less time, making it the superior method. Next I tried the adaboost function. I ran this with 10, 100 and 1,000 trees. As the number of trees increased, the accuracy decreased, and the time increased, which makes sense because boosting reduces bias, not variance. Although this may be more accurate when testing on different datasets, I did not feel that this form, or any form of boosting would optimize predictions for this assignment, given I am testing and training on subsets of the same dataset.

With that being said, the method I have chosen as my best method for generating accurate predictions for the ForestCover data set considering time as a component of generating a best method as well, is the RandomForest in parallel method. This was, on average, 94% accurate, although it did take 10 minutes to run. The PCA method was much more efficient with a runtime of just 20 seconds, but it was only 83% accurate, and I am afraid that using PCA may also be overfitting the data, which is why I have chose RandomForest in parallel as my favorite method for generating a categorical prediction for this large dataset.

Method:	**Accuracy:**	**Time:**
Lecture Notes:	0.698	3.7s
PCA = 4 K=20	0.834	20.3s
RandomForest reg	0.953	13m 42s
RandomForest parallel	0.943	10m 19s
AdaBoost n =10	0.558	11.3s