Dane Hooker- Capstone Final Project

The goal of my capstone project is to build a classifier to predict the next play that will be run by the offensive team in an NFL football game. While this exercise would be fun and interesting for a fan, it would have immense value for the opposing team to know ahead of time what play the offense will run.

I plan to make several different classifiers. Not only will I test different methods of classification, but I will also train my models on different slices of the data set. For example, if I am attempting to predict the next play run by the Dallas Cowboys offense, I would develop models based on the following slices of data:

* All Cowboys offensive plays in the data set
* All Cowboys offensive plays under the current offensive coordinator
* All Cowboys offensive plays this season
* All Cowboys offensive plays versus my team in the data set
* All Cowboys offensive plays versus defenses similar to mine in the data set

My data source comes from Kaggle and provides detailed information about every play run in the NFL for the 2009-2018 seasons.

(<https://www.kaggle.com/maxhorowitz/nflplaybyplay2009to2016/download>)

I anticipate using the following techniques from the course:

* Data cleaning (prep the data for modeling)
* Data exploration (try to understand the data and look for patterns)
* Visualizations (try to understand the data and look for patterns)
* Clustering (try to find defenses similar to ‘mine’)
* Random forest classifier
* SVM
* KNN classifier
* Possibly ensemble modeling (test an ensemble against the strongest single classifier)
* Cross validation

I think the biggest challenge will be tuning the various classification models that I anticipate making. Related to this challenge is another, which is answering the question of how good is good enough? At what point do I stop tuning? A third challenge will be evaluating/learning from my clustering. I think it is fairly straightforward to evaluate clusters when there are only 2 input variables, but it becomes much more difficult when there are dozens of inputs.