### Technology:

* GitHub version control
* Amazon cloud computing; parallelized feature search code?

Algorithm:

* **Creating features of k week averages**
  + number to create, choices of k
* **Handling missing data**
  + Data available only for later seasons in a given dataset
  + Features only available in later weeks in a given season
    - Exclude features with k greater than the earliest week in the training data minus one
      * Lose a lot of potentially meaningful data – if a rule is created for weeks 6-12, there will be no k wk averages with k greater than 5 in the entire feature set
    - Include all features for all rules
      * End up replacing missing values with something, introduce questionable information
        + If replace with all week average, perhaps predictive power of the true k wk average is masked or increased due to predictive power of the replacement all week average; all week average is already a feature anyways so why include it in the data in the first place?, etc.
    - If week eight is lowest in train data, a seven week average will be NA in week eight for teams with a bye. There have been seasons with bye weeks in week 1.
    - IDEA
      * Include all features in feature selection
      * When feature with k > min(prediction weeks) – 1 is proposed
        + All week average is used when k > prediction week – 1
        + Else, use k week average
      * Get two summary statistics, one for weeks =< k and another for weeks > k
      * The resulting prediction rule uses the min(week – 1, k) week average of a feature
    - FOR NOW
      * Exclude for k -2 > min prediction week in train data
      * This way teams with a bye early in season will still have full data available
      * Excluding more features, introducing less noise
    - For example: the eight week average of rushing yards per game for week six predictions
      * **Should the feature be empty or should it contain the average over the previous max(eight, current week – 1) weeks?**
        + Currently the algorithm (featureHelper function) uses the max() approach.
* Inclusion of novel features:
  + To use in place of k week averages for categorical variables
    - Dummy variable expansion; average 0s and 1s, pct
  + WWTS
    - use current data closing lines plus/minus noise
      * three classes: WWTS + d, LWTS – d, within +/- d WTS
  + Point Spread as a feature
    - Use covers.com closing lines for now but **get better historical line data!**
  + team strength
  + previous season summary data
  + score weighted features; e.g. passing yards in second half blowouts less important
  + close game through 3 quarters – eventual margin of victory more or less important?
    - Correlation of fourth quarter performance with future point spreads. Does a team that performs well in the fourth quarter in a given game have a higher or lower likelihood of performing well in future games?
  + Games won when trailing after 3 quarters; more or less likely to perform well in later weeks?
  + Blowout binary variable by week; leading by x or more after 3 quarters and win by y or more
    - Categorical variable; big win, close game, big loss
  + DVOA as a week of game feature
  + One decision rule for early season, one for late
    - For early season, prev season data, prev season DVOA
  + Average winning margin last k games
    - Scaling? Not important/relevant in gbm/random forests
* **Number of previous seasons to include in model training for a given season**
  + Importance weighting previous season training summary statistic
* Regression vs two-stage classification/regression
* Boosted trees, random forests, linear methods, closest neighbors
  + blending
* Tuning parameter validation
* **Choice of error metric**
  + RMSE
  + bet recommendation accuracy
  + bet recommendation accuracy and week/season variance
  + multi-class bet recommendation total value/weighted accuracy
  + **How to predict for all weeks with one decision rule? Better to use multiple rules?!** Lose data by splitting up the rule; increase bias, reduce variance
    - Predict for all weeks, summary statistic down-weighted for games ‘early’ in the season
* Bet recommendation system summary statistic
  + number of classes and breakpoints

Feature Selection Algorithm

* Propose base feature set with size drawn from a distribution
* Evaluate CV summary statistic and store for reference
* Store feature set to ensure no duplicates
* Store summary statistic for each feature in the set
* Based on summary statistic for the set and feature specific score based on cumulative performance, choose to keep or discard each feature with some probability.
  + For discarded features, choose to replace with other features or not with some probability
* Choose to increase the size of the feature set with some probability
  + Choose feature/s to add with some probability
* Repeat

### Market

* Competition
* Marketing sources
* Community building

### Team

* McAuliffe agreement paperwork
* Hussein conversation
* Kirch equity/role