Key: blue = discussed 1/30

green = topics for 1/28 meeting

red = discussed during 1/14 meeting

~~strike-through~~ = implemented

highlight = things to do

Technology:

* Amazon cloud computing
  + Interface? Unix or?
    - Re: generalized code with shared reference documents for multiple executions on multiple processors.
  + Pycloud
  + Keep writing robust, function based code; functions that accept a single argument for parallelizing

Algorithm:

* Creating features of k week averages
  + number to create, choices of k
    - ~~do them all.~~
* Dimension reduction!
  + PCA; princomp; fitting a hyperplane that maximizes variance of projection on to plane
    - Things to know?
    - ‘referring only to numeric variables’
      1. 0/1?
      2. No categorical variables
    - Na.action – try both
      1. Impute and na.fail b/c something must have gone wrong **OR**
      2. No impute and na.omit
      3. Impute with rf
    - scores vs. newdata?
      1. Score is betta
    - data vs. x?
    - scores – eigenvalues; choose number of components based on all greater than 5% of maximum
* Handling missing data
  + Data available only for later seasons in a given dataset
    - Bigger issue
      1. ~~Begin by imputing with median (simple to start)~~
      2. **MISS Forest technique; random forest imputation**
         * Have paper, read
  + Features only available in later weeks in a given season
    - **Ultimately, must make different prediction rules for different portions of the season to eliminate this issue – does not ‘make sense’ to impute for the k week average features**
      1. Prediction for the first game of season? Entirely different prediction rule…look to exploit historical bias in lines for H/A teams in first game of season? Use prev season record, last L games of last season win %, etc.? Change to a classification problem using prev season data and point spread; estimated robustness to +/- 1 point line difference? Else?
    - For now, impute with previous season median for prediction rules that do not accept missing values
      1. How about with the median for entire data set?
         * **Is it now/eventually important to impute for each fold in CV, as to not bias training data?**

**Imagine early seasons, feature exists just not recorded; has relationship to other features that are recorded; use imputation to fill in; prediction rule with missing values as outcome**

* Inclusion of novel features:
  + **# weeks from/until bye week**
    - **Indicator variable**
    - **Days since previous game**
  + **home team k wk datum when at home average; away team**
    - **winning margin, win %**
    - **tells of how well team plays at home/away**
  + To use in place of k week averages for categorical variables
    - Dummy variable expansion; average 0s and 1s, pct
  + Point Spread
    - **get better historical line data!**
      1. ~~For now use closing line~~
         * ~~As move closer to game time the line will get closer the closing line; prediction gets more accurate~~
  + 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
  + 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
  + More seasons -> less variance because more data
  + Loop over number of seasons as a later stage part of analysis
  + How about holding things fixed and CV on one dimension, hold all but another dim fixed, CV, etc
    - Use hold one fixed for all internal and external parameters; spend comp time on grid search on external; then internal if needed
* Regression vs two-stage classification/regression
  + Loop later; for now just regression
* Boosted trees, random forests, linear methods, closest neighbors, blending
  + Run full process separately for different methods
  + Hard code non-generalizable issues in separate source files, using shared functions where practical
* Tuning parameter validation
* **Choice of error metric**
  + RMSE
    - ~~Use RMSE for now; different stages will require different error metrics~~
    - Hesitant to use margin of victory error as the only summary statistic

4th quarter points for/against team with a large lead are often not representative of the ‘closeness’ of a game. A close game with a 14 point swing in the 4th quarter and a 14 point final margin; a blowout with the losing team getting garbage points with other team in prevent defense.

* + - 1. Binning margins to create an initial classification problem, followed by regression on the classes for a margin estimate.
         * Doesn’t address the entire issue.
      2. **Inclusion of the line feature may actual help mitigate things; the prediction rule tends to be largely influenced by point spread with smaller changes based on other features.**
    - **Keep this in mind for later stages of analysis; for now RMSE; get mvp and go from there**
  + 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
    - Split into quarters.
      1. Compare with first 4 wks and rest of them
* 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

Fit model with entire feature set (increasing variance, decreases bias); fit model with PCA (increasing bias, decreasing variance)

Fit a model with n seasons; fit a model with 2n seasons; if prediction on a new season are substantially different for the two models, then more training is better than more validation data