NBA GAME PREDICTION

David Hamilton Michael Yang Steffen Roehrsheim

PROJECT OBJECTIVES

A — Point Spread Prediction
Regression methods to model the points spread between teamss as close as possible, .

B — Classification
Predicting which team is going to win the mach up, and if possible try to predict the market line.

FEATRURES
Team stats from stats.nba.com on a game by game basis.

TIMEFRAME
Taking the last five season of data, 2012-2017. Using the first 3 for train, 1 validate, 1 test.

PROCCESSING
We choose the expanding window averaging for the features, staring from game one.

Format
For each training example, we take the home & away team and stack it as an vector.

ADVANCE STATS
For improving proformece, we also included stats like Offense rating, Opponent's Defensive Rebounds

METHODS

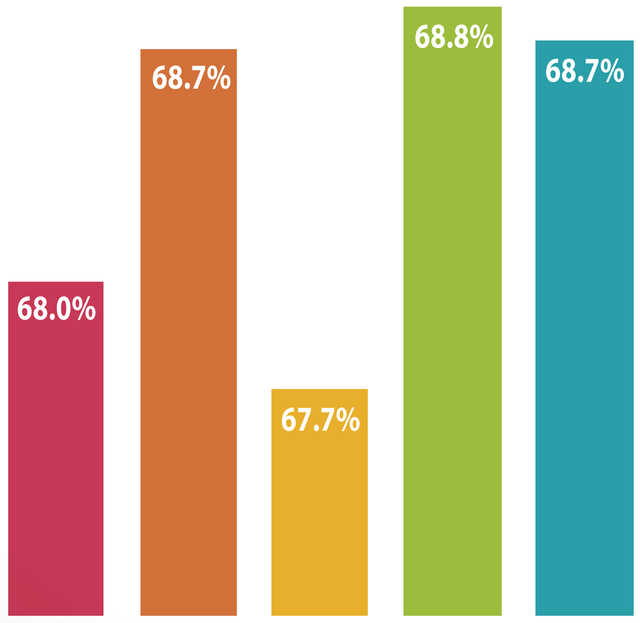
Linear Regression
Ridge Regression, Lasso Regression, Elastic-net

Hard Classification
Logistic Regression, SVM, Kernel-SVM

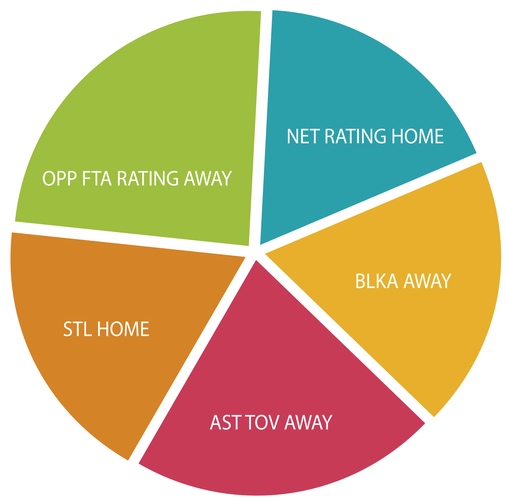
Bayesian Methods
Naive Bayesian, Bayesian Ridge Regression

Tree Models
Gradient Boosting Trees, Random Forest

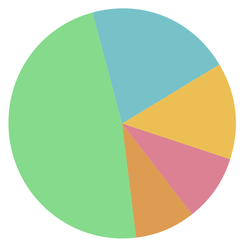
K-Neighbors
K-Neighbors Regression



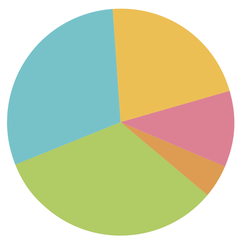
ERROR ANALYSIS



Most Wrong



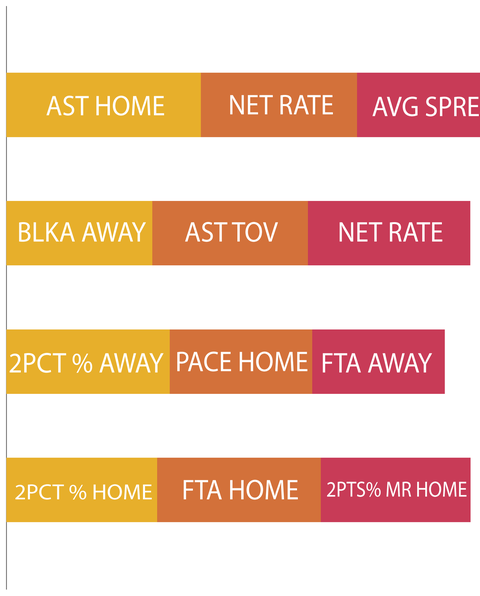
Most Right



STL HOME
AST TOV
BLKA AWAY
OPP TOV PCT AWAY
OPP FTA RATING AWAY
FTA AWAY

PCA ANALYSIS

From looking into the teams we consistantly predicted wrong, we isolated a couple of features, running only with and without those features gives roughly the same performance. The similar performance indicates there's a strong correlation in the features, thus we are looking at PCA.



94%

The first four principle components gives a 64.5% precision, reducing the dimension frome 125 to 4 features, we are still capable of maintaining 94% of our best performance.

We found the same important features in our error analysis as well as in the principle components, meaning the most varing directions in our data are also the most relavent.

FUTURE IMPORVEMENTS

A — Include player statistics, of perticular interest here would be to match up players against players to predict the would perform against each other.

B — We found that kernelizing is not helping our out of sample performance, so the non-linearity of this problem is not captured by common kernels, we need to find smart ways of combining features to achieve better performance.