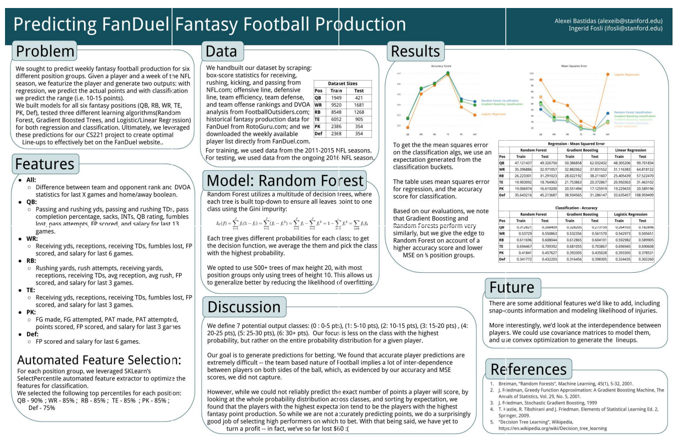
BLOCK POSTERS

Your settings for this Block Poster are:

Pages Wide	4
Orientation	PORTRAIT
Paper Format	LETTER
Border Setting	Top: 8, Right: 8, Bottom: 9, Left: 9

You can find tips on printing, assembling and putting up your poster at www.blockposters.com

Enjoy your Block Poster!



Predicting FanDuel

Problem

We sought to predict weekly fantasy football production
different position groups. Given a player and a week of t
season, we featurize the player and generate two output
regression, we predict the actual points and with classifi
we predict the range (i.e. 10-15 points).
We built models for all six fantasy positions (QB, RB, WR,
PK, Def), tested three different learning algorithms(Rand
Forest, Gradient Boosted Trees, and Logistic/Linear Regr
for both regression and classification. Ultimately, we leve
these predictions for our CS221 project to create optima
Line-ups to effectively bet on the FanDuel website..

Features

- **All:**
  - Difference between team and opponent rank and statistics for last X games and home/away boolea
- **QB:**
  - Passing and rushing yds, passing and rushing TDs completion percentage, sacks, INTs, QB rating, fu lost, pass attempts, FP scored, and salary for last

Fantasy Football Pro

for six
he NFL
s: with
cation

TE,
om
ession)
eraged
l

l DVOA
n.

s, pass
mbles
13

Data

We handbuilt our dataset by scraping: box-score statistics for receiving, rushing, kicking, and passing from NFL.com; offensive line, defensive line, team efficiency, team defense, and team offense rankings and DVOA analysis from FootballOutsiders.com; historical fantasy production data for FanDuel from RotoGuru.com; and we downloaded the weekly available player list directly from FanDuel.com.

Data	
Pos	**Trai**
QB	194
WR	952
RB	854
TE	605
PK	238
Def	236

For training, we used data from the 2011-2015 N
For testing, we used data from the ongoing 2016

Model: Random For

Random Forest utilizes a multitude of decision t
each tree is built top-down to ensure all leaves
class using the Gini impurity:

$$I_G(f) = \sum^{J} f_i(1-f_i) = \sum^{J} (f_i - {f_i}^2) = \sum^{J} f_i - \sum^{J} {f_i}^2 = 1 - \sum^{J}$$

oduction

set Sizes	
in	Test
9	421
0	1681
8	1268
2	905
6	354
8	354

NFL seasons.
5 NFL season.

rest

trees, where
point to one

$$\sum f_i^2 = \sum f_i f_k$$

Results

To get the the mean squares error on the classification algs, we use an expectation generated from the classification buckets.

The table uses mean squares error for regression, and the accuracy score for classification.

Based on our evaluations, we note that Gradient Boosting and Random Forests perform very

Pos	
QB	
WR	
RB	
TE	
PK	
Def	
Pos	

Alexei Bastidas (alexeib@stanford.edu)
Ingerid Fosli (ifosli@stanford.edu)

Regression - Mean Squared Error						
	Random Forest		Gradient Boosting		Linear Regression	
	Train	Test	Train	Test	Train	Test
	47.121437	49.326750	50.386858	62.032432	48.305206	99.701834
	35.396886	32.971057	32.882062	37.831552	31.116383	64.818122
	26.223301	31.291023	28.022192	38.211607	25.405439	57.522470
	19.903092	18.764963	21.753863	20.372867	20.950363	31.463102
	19.006974	16.619200	20.551494	17.125919	19.229433	20.589196
	35.643216	45.213687	38.934565	31.286147	33.635457	108.959499

Classification - Accuracy						
	Random Forest		Gradient Boosting		Logistic Regression	
	Train	Test	Train	Test	Train	Test

games.

- **WR:**
  - Receiving yds, receptions, receiving TDs, fumbles scored, and salary for last 6 games.
- **RB:**
  - Rushing yards, rush attempts, receiving yards, receptions, receiving TDs, avg reception, avg rush scored, and salary for last 3 games.
- **TE:**
  - Receiving yds, receptions, receiving TDs, fumbles scored, and salary for last 3 games.
- **PK:**
  - FG made, FG attempted, PAT made, PAT attempte points scored, FP scored, and salary for last 3 gan
- **Def:**
  - FP scored and salary for last 6 games.

Automated Feature Selectio

For each position group, we leveraged SKLearn's SelectPercentile automated feature extractor to optimiz features for classification.

We selected the following top percentiles for each positi QB - 90% ; WR - 85% ; RB - 85% ; TE - 85% ; PK - 85% ; Def - 75%

lost, FP

, FP

lost, FP

ed,
nes

n:

e the

ion:

$\sum_{i=1}$ $\sum_{i=1}$ $\sum_{i=1}$ $\sum_{i=1}$ $\sum_{i=1}$

Each tree gives different probabilities for each c
the decision function, we average the them and
with the highest probability.

We opted to use 500+ trees of max height 20, w
position groups only using trees of height 10. Th
to generalize better by reducing the likelihood o

Discussion

We define 7 potential output classes: (0 : 0-5 pts
20-25 pts), (5: 25-30 pts), (6: 30+ pts). Our focus
probability, but rather on the entire probability

Our goal is to generate predictions for betting. W
extremely difficult -- the team based nature of F
between players on both sides of the ball, which
scores, we did not capture.

However, while we could not reliably predict the
looking at the whole probability distribution acr
found that the players with the highest expectat
fantasy point production. So while we are not a
good job of selecting high performers on which
turn a profit -- in fact, we've so far lost $6

$\sum_{i \neq k}$

class; to get
pick the class

ith most
his allows us
of overfitting.

Random Forests perform very
similarly, but we give the edge to
Random Forest on account of a
higher accuracy score and lower
MSE on ⅚ position groups.

QB
WR
RB
TE
PK
Def

s), (1: 5-10 pts), (2: 10-15 pts), (3: 15-20 pts) , (4:
s is less on the class with the highest
distribution for a given player.

We found that accurate player predictions are
Football implies a lot of inter-dependence
n, as evidenced by our accuracy and MSE

e exact number of points a player will score, by
ross classes, and sorting by expectation, we
tion tend to be the players with the highest
ccurately predicting points, we do a surprisingly
to bet. With that being said, we have yet to
50 :(

Fu

There
snap-

More
player
and us

Re

1. Bre
2. J. Fr
Ann
3. J. Fr
4. T. H
Spr
5. "De
http

	0.312821	0.268409	0.328205	0.273159	0.264103	0.182898
	0.53729	0.550863	0.532356	0.561570	0.542973	0.505651
	0.611696	0.608044	0.612865	0.604101	0.592982	0.589905
	0.694467	0.709392	0.681055	0.703867	0.696945	0.690608
	0.41841	0.457627	0.393305	0.435028	0.393305	0.378531
	0.341772	0.432203	0.316456	0.398305	0.324435	0.302260

ture

are some additional features we'd like to add, including counts information and modeling likelihood of injuries.

interestingly, we'd look at the interdependence between rs. We could use covariance matrices to model them, se convex optimization to generate the lineups.

eferences

iman, "Random Forests", Machine Learning, 45(1), 5-32, 2001.
riedman, Greedy Function Approximation: A Gradient Boosting Machine, The nals of Statistics, Vol. 29, No. 5, 2001.
riedman, Stochastic Gradient Boosting, 1999
Hastie, R. Tibshirani and J. Friedman. Elements of Statistical Learning Ed. 2, inger, 2009.
cision Tree Learning", Wikipedia, os://en.wikipedia.org/wiki/Decision_tree_learning