Predicting NBA Shot Results

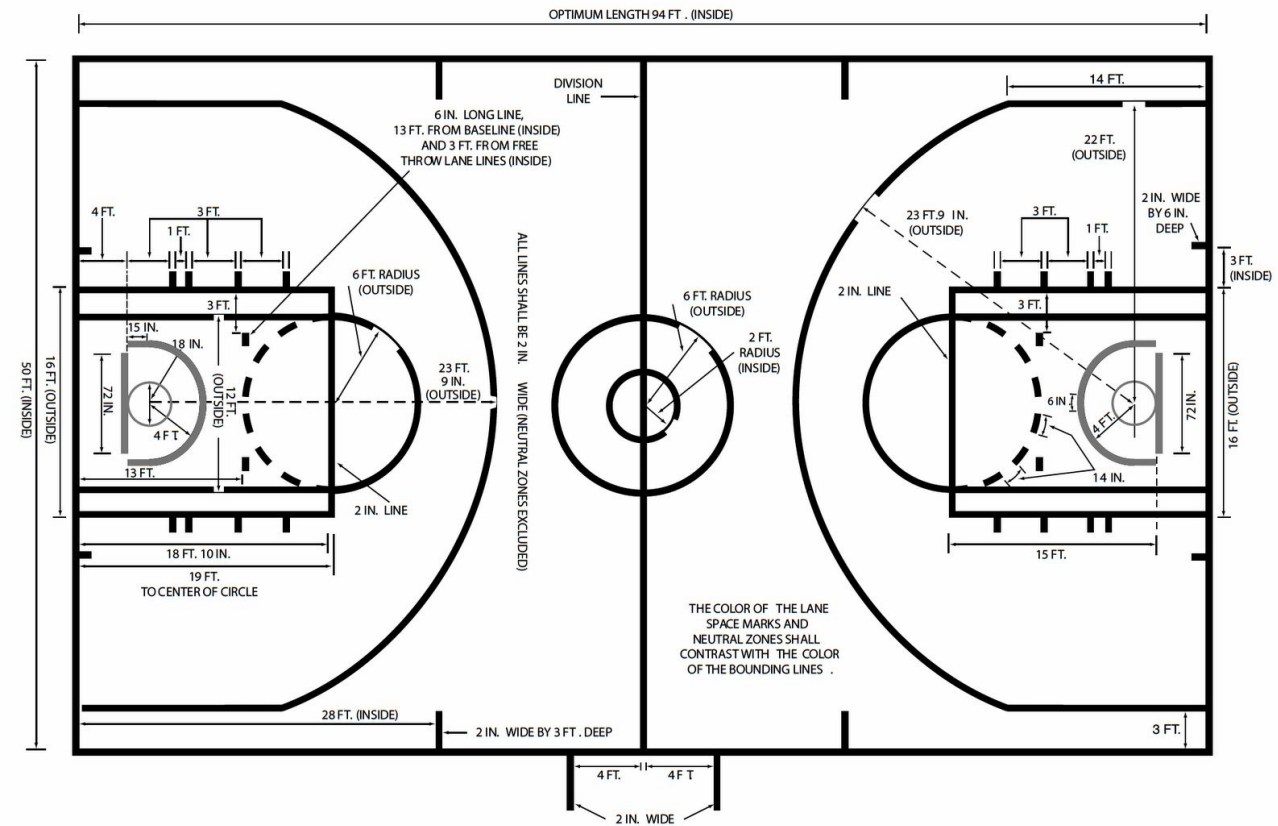
By Gilbert Zhang

Agenda

- Problem Statement
- Data Source and Scope
- EDA and Feature Engineering
- Machine Learning Models
- Insights
- Challenges
- Next Step

Disclaimer: All images (other than my own analysis tables and graphs) included herein are from Google Search. I do not own any of them.

First, to bring everyone to the same page



Problem Statement

All NBA teams want to maximize the value of each possession, a.k.a maximize the chance of making each shot (apparently I guess). So can we predict the results of shots? Or in other words, can we evaluate the quality of NBA shots?

Data Source

kaggle

Dataset contains shots taken during the 2014-2015 season.

Dataset is scraped from NBA's REST API.

Link for the dataset: https://www.kaggle.com/dansbecker/nba-shot-logs

Scope

- Only two-pointers
  - Time constraint
  - Two pointers are fundamentally different than three pointers
  - Analyzing both point types together might distort modeling performance
  - Three pointers analysis will be my next step
- Shooters and Defenders
  - Dataset includes the shooter's and closest defender's name
  - For purpose of this project, shooters' and defenders' abilities will not be considered
  - Position of the player and mismatching will not be considered
  - Will be my next step!

Dataset Overview

	# of NaN[a]	# Shot Distance Mis-class[b]	# of Obs (After dropping a & b)	# of Features
Whole set	5,567	513	90,339	15
Training set	NaN are dropped	Obs are dropped	54,205 60/40 split	As above
Testing set	NaN are dropped	Obs are dropped	36,134 60/40 split	As above

b. Some shots have distance greater than 23.75 fts. These shots are either mistakenly classified as two point shots or their distance are measured incorrectly. For shots that are greater than 22 feets (baseline 3pts line), I assume that all these shots are 2pts from the wing or the arc. There is no way I can make sure since the dataset does not include shot locations.

Data Cleaning & Variable Transformation

Matchup

- String format: ‘MAR 04, 2015 - CHA @ BKN’
- Transformed into month and date in *panda* format
- Better for understanding of whether shots are from regular season or playoffs

Location

- String format: ‘Home’ or ‘Away
- Turn into binary variable
- 1 = ‘Away’; 0=’Home’

Game Clock

- String format: ‘1:09’
- Transformed into the secondth of the game
- Will not be able to use the feature without the transformation

Dribbles

- Integer format
- Transformed into binary
  - Dribbles = 0 → ‘Catch & Shoot’ = 1
  - Dribbles > 0 → ‘Catch & Shoot’ = 0
- This transformation helped increase the explaining power of the feature (higher correlation with response variable after transformation)

EDA - Correlations

	FINAL_MARGIN	SHOT_NUMBER	PERIOD	TOUCH_TIME	SHOT_DIST	CLOSE_DEF_DIST	FGM	location_t	GAME_CLOCK_t	SHOT_CLOCK_t	Catch&Shot
FINAL_MARGIN		0.0069	0.0103	0.0094	-0.0150	0.0153	0.0480	-0.1647	0.0109	-0.0038	0.0121
SHOT_NUMBER	0.0069		0.6566	0.1529	0.0080	-0.0200	-0.0045	0.0011	0.5838	-0.0402	-0.0812
PERIOD	0.0103	0.6566		0.0659	-0.0099	-0.0223	-0.0077	-0.0033	0.9701	-0.0525	-0.0315
TOUCH_TIME	0.0094	0.1529	0.0659		0.0640	-0.0935	-0.0844	0.0138	0.0377	-0.1618	-0.5533
SHOT_DIST	-0.0150	0.0080	-0.0099	0.0640		0.4614	-0.1688	0.0038	-0.0005	-0.2265	-0.0312
CLOSE_DEF_DIST	0.0153	-0.0200	-0.0223	-0.0935	0.4614		0.0514	-0.0047	-0.0166	0.0222	0.1542
FGM	0.0480	-0.0045	-0.0077	-0.0844	-0.1688	0.0514		-0.0070	-0.0081	0.1129	0.0946
location_t	-0.1647	0.0011	-0.0033	0.0138	0.0038	-0.0047	-0.0070		-0.0027	-0.0097	-0.0065
GAME_CLOCK_t	0.0109	0.5838	0.9701	0.0377	-0.0005	-0.0166	-0.0081	-0.0027		-0.0468	-0.0221
SHOT_CLOCK_t	-0.0038	-0.0402	-0.0525	-0.1618	-0.2265	0.0222	0.1129	-0.0097	-0.0468		0.1446
Catch&Shot	0.0121	-0.0812	-0.0315	-0.5533	-0.0312	0.1542	0.0946	-0.0065	-0.0221	0.1446	

Most important features seem to be:

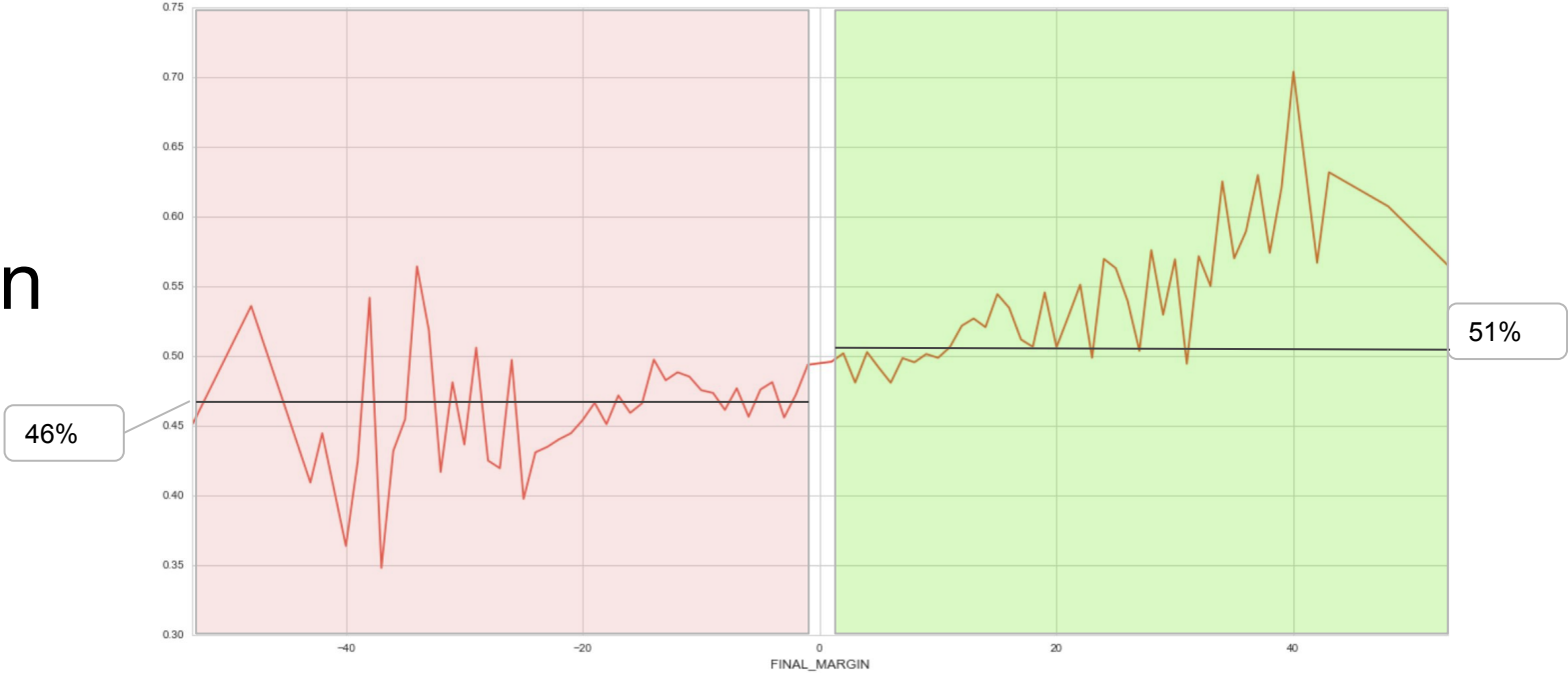
- Final_Margin: Point difference at the end of game.
- Touch_Time: Number of touches by the shooters before the shots.
- **Shot_DIST**: Shot distance from the basket.
- **CLOSE_DEF_DIST**: Distance from the closest defender.
- **Catch&Shoot**: Whether the shot is catch & shoot or off dribble.
- Shot_Clock: Seconds left in the shot clock when the shots are made.

Correlations within features themselves:

- Game Clock and Period
- Catch&Shoot and Touch_Time
- Game Clock and Shot_Number….
- More discussion will be followed

Final Margin

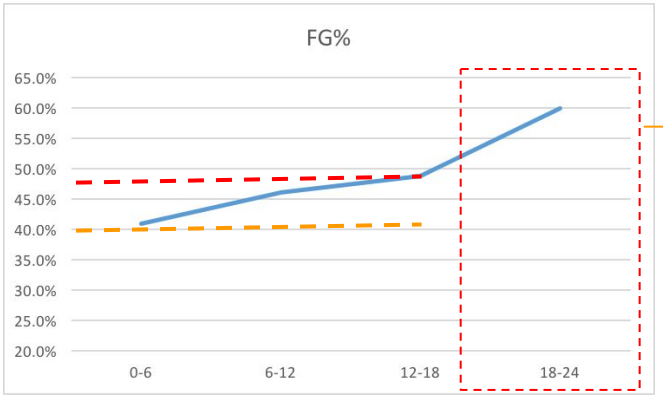
FG% by Final Margin - Egg First? Chicken First? What do you think which one cause the other?

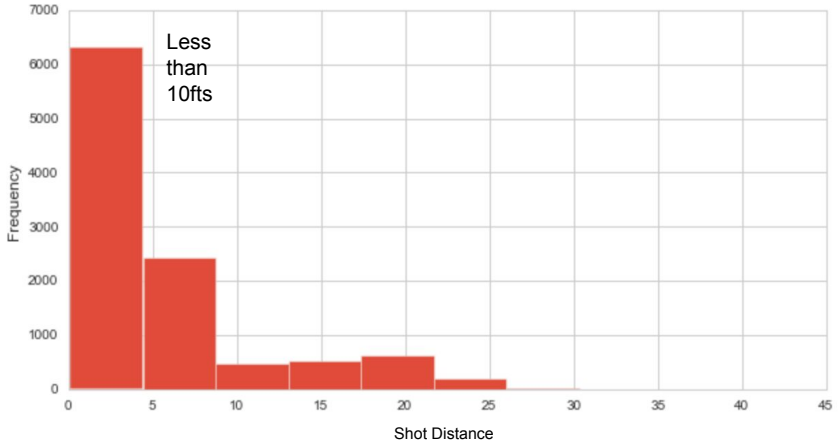


Touch Time & Shot Clock

Touch Time - 0.9 correlation with Dribbles. Basically Touch Time and Dribbles measure the same thing - whether the shooter make the shot by a quick catch and shoot or not. Including it in the machine learning model will not help.

Shot Clock - correlation with shot distance. Most shots taken at the beginning of the shot clock are close shots! It does look like much of the correlation between shot clock and FG% comes from the correlation between shot distance and shot clock.





The Big Three?

Shot Distance

Closest Defender Distance

Catch & Shoot

Shot Distance and Closest Defender

Closet Defender Distance	Shot Distance											
	2	4	6	8	10	12	14	16	18	20	20+	Total
1	50.6%	50.3%	44.1%	38.2%	36.1%	34.5%	34.7%	26.3%	32.1%	25.0%	26.3%	45.5%
2	57.5%	51.6%	43.0%	38.6%	35.3%	37.1%	36.3%	41.1%	38.5%	33.3%	28.6%	45.4%
3	65.8%	61.1%	52.3%	42.6%	36.2%	38.0%	39.7%	37.2%	36.9%	34.0%	34.8%	49.2%
4	75.4%	73.7%	60.8%	46.5%	40.7%	38.6%	41.5%	39.6%	41.8%	38.1%	34.3%	50.8%
5	78.1%	82.6%	67.6%	56.2%	47.2%	43.8%	44.8%	39.5%	43.7%	38.6%	37.7%	49.3%
6	79.6%	87.2%	75.3%	64.1%	49.3%	50.6%	45.5%	43.9%	43.7%	40.4%	38.6%	48.9%
7	78.5%	88.4%	85.1%	53.5%	57.7%	51.3%	50.6%	45.2%	44.1%	45.5%	43.8%	51.5%
8	91.9%	93.3%	91.3%	46.7%	53.3%	34.8%	60.6%	46.2%	49.2%	44.0%	36.8%	50.8%
9	100.0%	96.0%	84.2%	100.0%	40.0%	54.5%	48.1%	42.9%	46.3%	40.8%	42.3%	50.0%
10	100.0%	100.0%	91.7%	100.0%	50.0%	60.0%	78.6%	60.7%	43.1%	58.7%	42.1%	58.4%
10+	100.0%	99.3%	97.5%	100.0%	66.7%	50.0%	84.6%	71.2%	46.9%	49.5%	39.7%	61.5%
Total	64.5%	62.9%	52.0%	43.4%	39.1%	40.2%	42.8%	41.2%	42.6%	40.3%	37.9%	48.9%

a. Shot distance alone - close up shots have better accuracy, but after 10 fts, shot distance doesn't seem to make a difference.
b. Closest defender distance
    i. For close-up shots, defender definitely needs to keep up with the shooters - the margin decrease in FG% are much higher for every foot defender gain.
    ii. For shots that are further away from the basket, the marginal gain of one foot is not as high. The difference in FG% between an 'open' shot (4 - 6 fts) and a 'wide open' shot are not significant.
c. Mid-range 'open' shots: surprisingly these shots have a lower accuracy rate than my expectation. How are the closest defender distance measured?

Catch and Shoot Impact

FG% Difference Between Catch & Shoot and Off Dribble

Closet Defender Distance	Shot Distance											
	2	4	6	8	10	12	14	16	18	20	20+	Total
1	2.5%	5.8%	4.3%	4.0%	-1.2%	-15.4%	28.2%	26.5%	9.6%	60.0%	-31.3%	8.3%
2	6.2%	10.1%	9.3%	8.5%	1.9%	2.6%	-13.9%	10.8%	-7.7%	-11.6%	8.2%	12.6%
3	11.0%	5.3%	6.4%	8.3%	11.1%	-8.1%	-6.9%	-3.4%	7.0%	3.7%	-5.5%	13.3%
4	9.0%	12.1%	11.6%	4.1%	-3.3%	-2.1%	2.3%	8.4%	1.5%	-5.0%	0.1%	13.3%
5	14.3%	6.1%	3.0%	5.3%	6.4%	4.2%	2.1%	2.2%	1.5%	2.3%	3.0%	7.6%
6	11.3%	10.7%	-7.2%	6.1%	13.0%	-21.3%	4.9%	1.4%	1.1%	-2.4%	-1.5%	1.1%
7	14.2%	5.9%	-10.1%	-10.6%	-24.9%	-27.2%	-23.6%	0.4%	-1.1%	4.3%	14.9%	1.2%
8	7.6%	7.4%	2.3%	10.0%	60.7%	9.2%	-4.5%	5.1%	10.0%	11.1%	-0.7%	5.0%
9	0.0%	2.0%	8.9%	0.0%	38.1%	-7.1%	19.9%	15.0%	8.6%	6.0%	-10.2%	2.0%
10	0.0%	0.0%	-16.7%	NA	NA	-16.7%	-2.2%	42.9%	0.3%	4.0%	-1.8%	-4.2%
10+	0.0%	-2.6%	3.2%	0.0%	50.0%	NA	40.9%	35.4%	5.1%	-0.6%	7.4%	-21.7%
Total	8.5%	9.1%	9.1%	8.3%	6.6%	-2.4%	3.3%	7.2%	3.5%	3.7%	3.3%	9.7%

a. Overall, catch and shoot improve FG% for almost all shot distances and closest defender distance. The improvement is most significant for closely defended shots - (<4 fts).

b. Interestingly, for shots from 12 fts, catch and shoot actually make FG% worse. Unfortunately, shot location information is not included in the dataset.

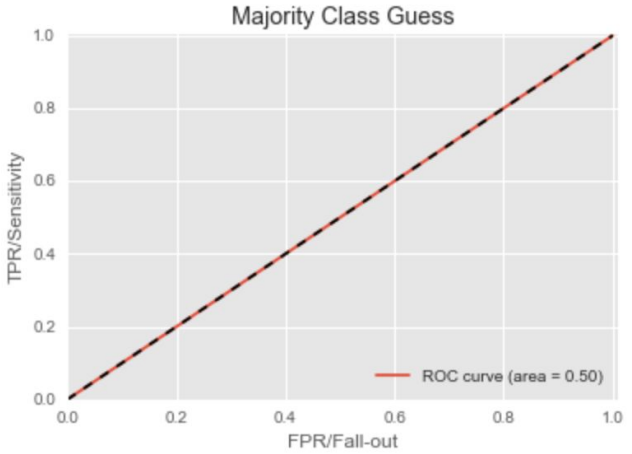
Machine Learning

Majority Class Guess

Predict ‘missed’ for all shots.

		Hypothesized	
		Missed	Made
True Class	Missed	27,865	0
	Made	26,646	0

Training Accuracy is **51%**.

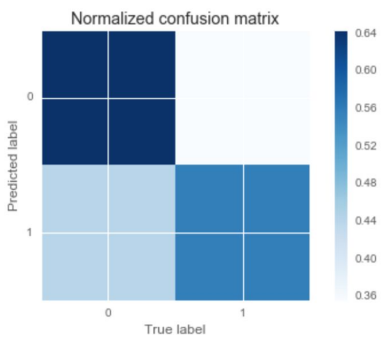


Logistic Regression

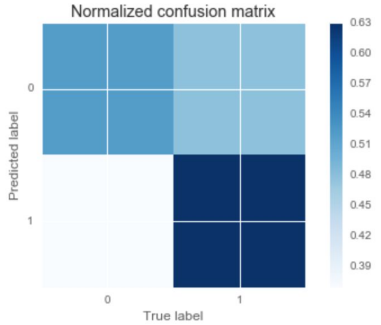
1. All variables
2. Shot Distance, Closest Defender Distance & Catch&Shoot
3. Shot Distance
4. Shot Distance & Catch&Shoot
5. Shot Distance & Closest Defender Distance

Logistic Regression

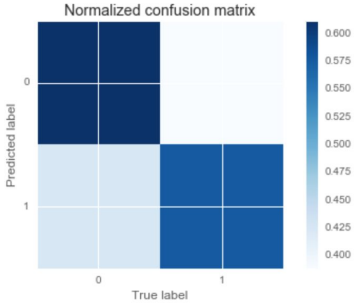
Model 1



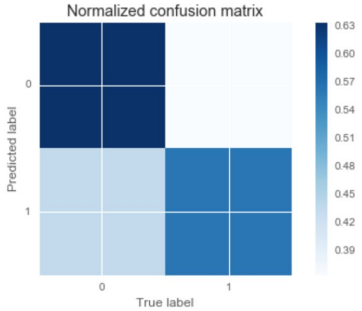
Model 3



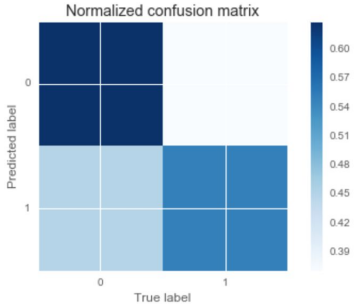
Model 5



Model 2



Model 4



1. All variables
2. Shot Distance, Closest Defender Distance & Catch&Shoot
3. Shot Distance
4. Shot Distance & Catch&Shoot
5. Shot Distance & Closest Defender Distance

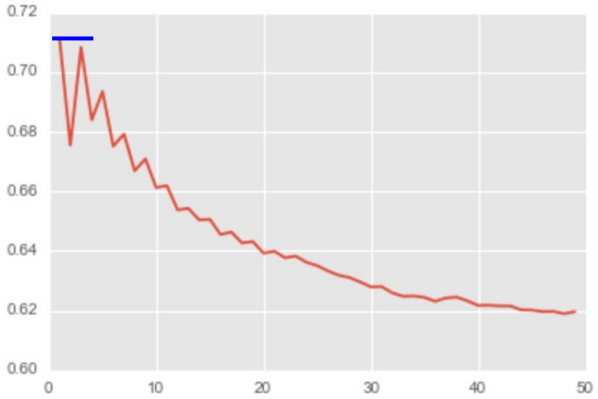
Logistic Regression

- Logistic model 2 are preferred
- Accuracy rate is almost the same as model 1 (all variables included)
- Only three features are used - simpler than model 1

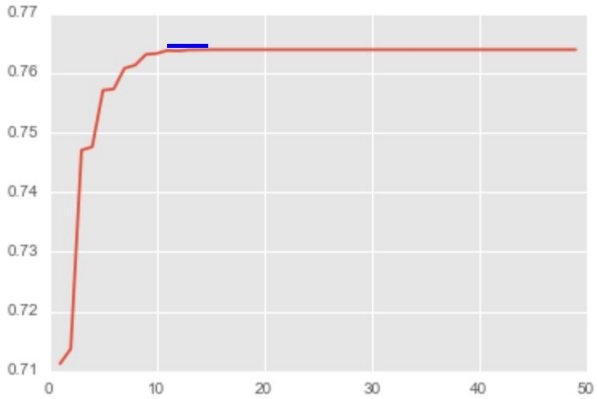
KNN

Three variables KNN model - Shot Distance, Closest Defender Distance and Catch&Shoot.
All variables are scaled to between 0 and 1.

KKN 1: Uniform - accuracy is maximized at k = 1

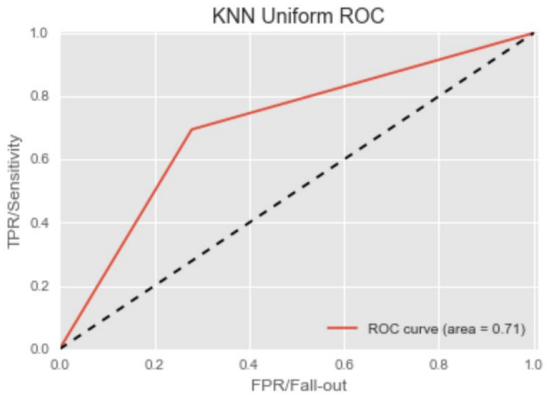


KKN 2: Weighted - accuracy is maximized at k = 15



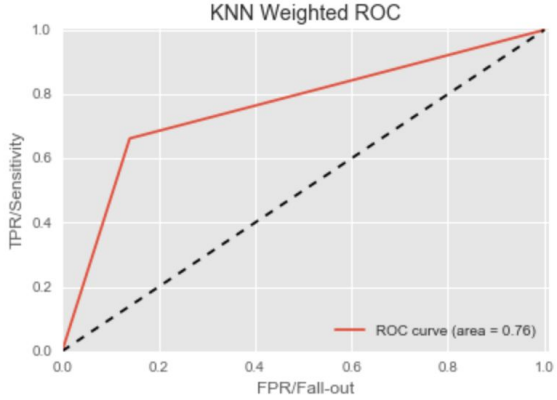
KNN

KKN 1: Training result



KKN 1: Cross validation accuracy = 0.53

KKN 2: Training result



KKN 2: Cross validation accuracy = 0.55

Random Forest

- Hyper Parameter:
  - Tree: 100
  - Maximum feature per tree: 3
  - Minimum leaf: 5
- Training result: 0.89
- Cross validation: 0.59
- Overfit
- Tried increasing to 1,000 and 10,000 - no significant change on cross validation result
- Also tried different combinations of maximum features and minimum leaf, but did not really improve the cross validation result

Grid Search

Random Forest:

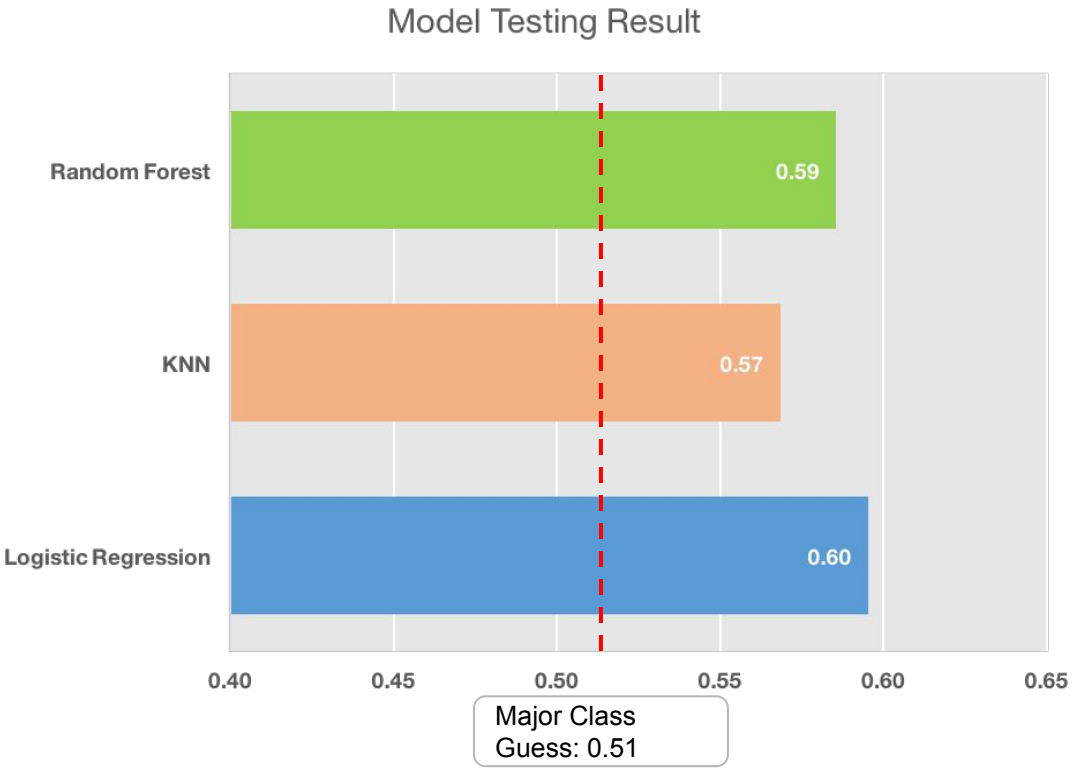
- Still loading…..

KNN:

- Still loading…..

So who is the MVP?

Testing Result

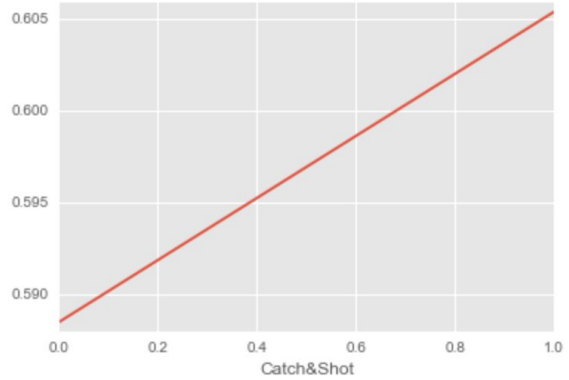


The MVP is logistic regression

1. It does not overfit…
2. It performs consistently in CV and testing
3. Less features are used
4. It is way less time consuming

A Deeper Look

Logistic Regression Accuracy Rate by Features

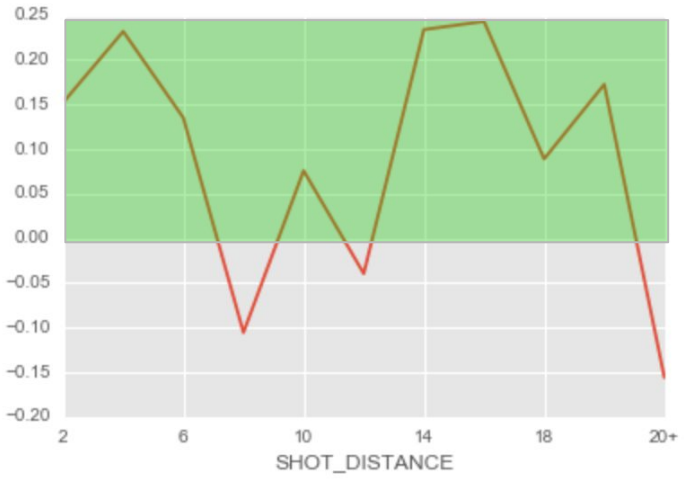


Insights and Implications

- The logistic model is 60% correct as compared to 51% of all ‘missed guess’
- Might not seem significant but think…
  - I am assigning probability to shots, in other words, evaluating shot quality as compared to just assigning ‘0’ or ‘1’
  - This could help understand the different game strategy efficiency
  - The final margin feature shows that 75% of the game were decided within 9 points, which is an average of 3-4 shots attempts.
  - Shots attempts per game average more than 170 for both teams - so increasing accuracy rate by 1% can somehow decide the game

Let’s have some fun… Curry taking lower quality shots but better at shooting?

Curry: Actual FG% minus predicted FG probability



Westbrook: Actual FG% minus predicted FG probability

Challenges and Next Steps

Challenges

- Visualization in Python is not easy
- Overfitting and grid search is time consuming
- Evaluate difference models' performance
- How to better feature engineer

Next Steps

- Model the three pointers
- More features can be included - players, locations, fatigue and etc.
- Grid search
- Model improvement - Boosting and Regularization
- Simulate a game

Question/FeedBack

Thank you!

https://github.com/gyzhang328/Predicting-NBA-Shot-Results