Quantifying uncertainty in NFL game outcomes

Luther Landry

Summary

- Motivation and data science goals
- How the game works
- The baseline model
- Future models

Motivation

- Sports initiated my interest in statistics

- Fascinating questions in probability, statistics, and uncertainty

Do Firms Maximize? Evidence from Professional Football

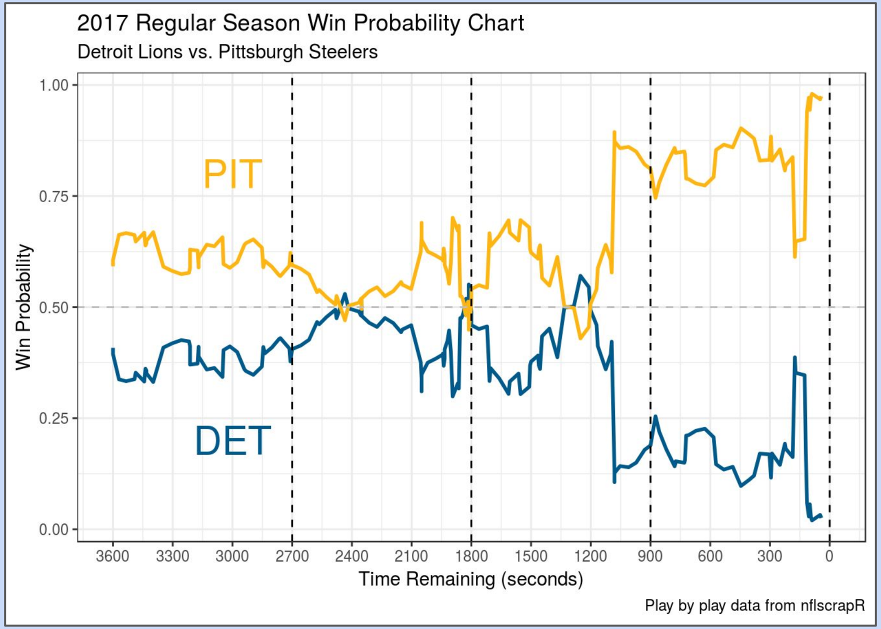
David Romer

University of California, Berkeley and National Bureau of Economic Research

This paper examines a single, narrow decision—the choice on fourth down in the National Football League between kicking and trying for a first down—as a case study of the standard view that competition in the goods, capital, and labor markets leads firms to make maximizing choices. Play-by-play data and dynamic programming are used to estimate the average payoffs to kicking and trying for a first down under different circumstances. Examination of actual decisions shows systematic, clear-cut, and overwhelmingly statistically significant departures from the decisions that would maximize teams' chances of winning. Possible reasons for the departures are considered.

Motivation

- NFL analysts never tell you how uncertain their predictions are.



Motivation

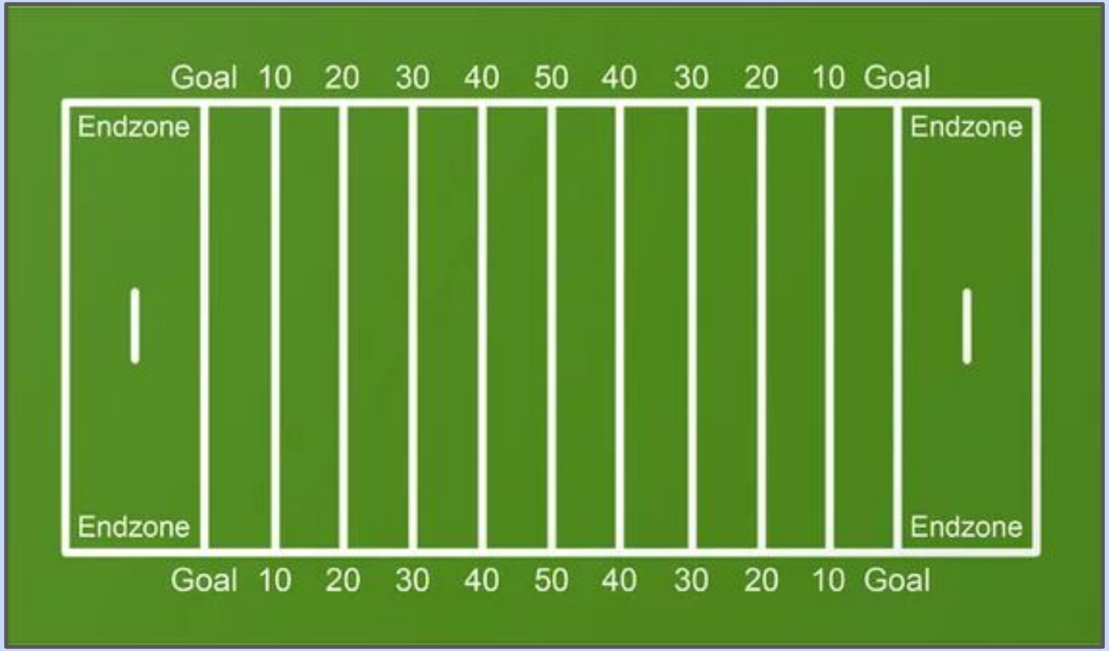
- NFL analysts never tell you how uncertain their predictions are.

		Record			Super Bowl Winner
Team	Conference	W	L	T	
Eagles	NFC	8	0	0	26%
Vikings	NFC	8	1	0	18%
Chiefs	AFC	7	2	0	11%
Ravens	AFC	6	3	0	7%
Bills	AFC	6	3	0	7%
Dolphins	AFC	7	3	0	6%
Titans	AFC	6	3	0	5%

Goals

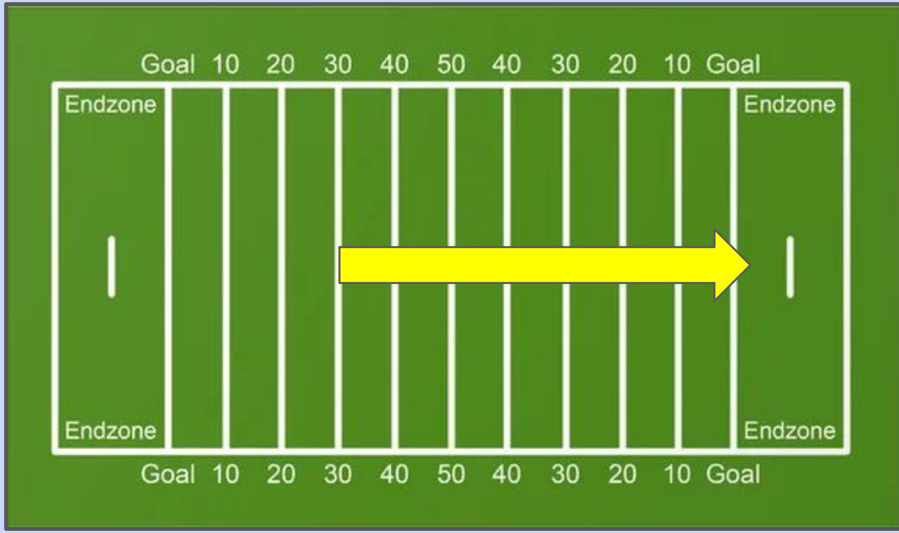
- Quantify the true probability of NFL outcomes with **robust uncertainty** estimates.
- Generate robust predictions for two types of outcomes: individual game outcomes, and team season outcomes.
- More generally, treat this as an MLOps problem.

How NFL football works



How NFL football works

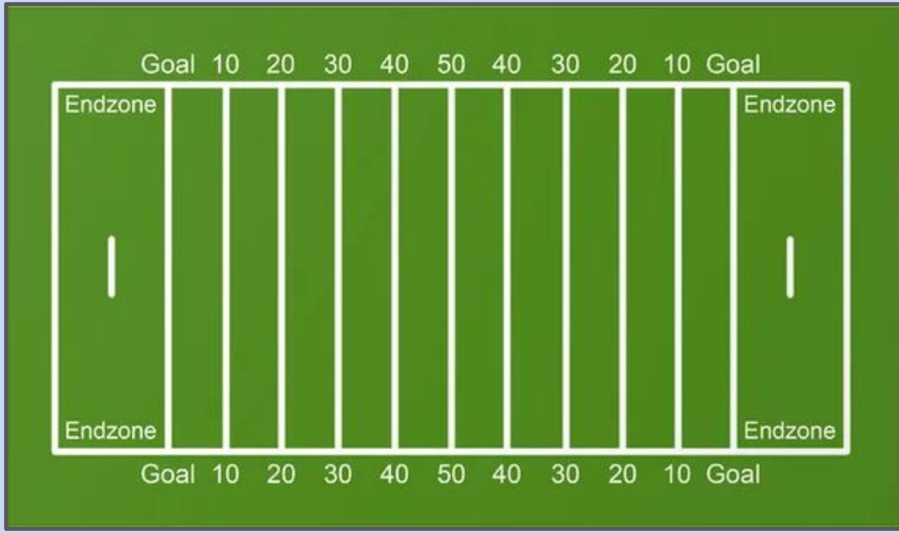
- The object is to get the football into your opponent's endzone and keep it out of your own.
- The team possessing the ball gets a finite number of attempts to score.
- Four tries to advance the ball ten yards. If you succeed, you get another four tries.



How NFL football works

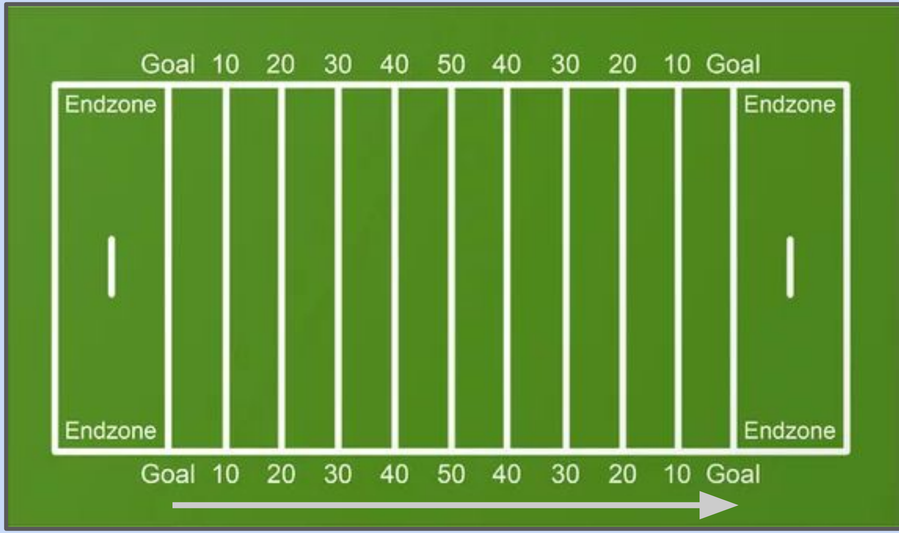
- Football has a finite state space of discrete plays — and each play is a stepwise movement in the state space.

- The state of a game is completely described by a few variables — score, time, down, distance, and yardline.



How do we quantify a team's skill?

- Three key metrics



How do we quantify a team's skill?

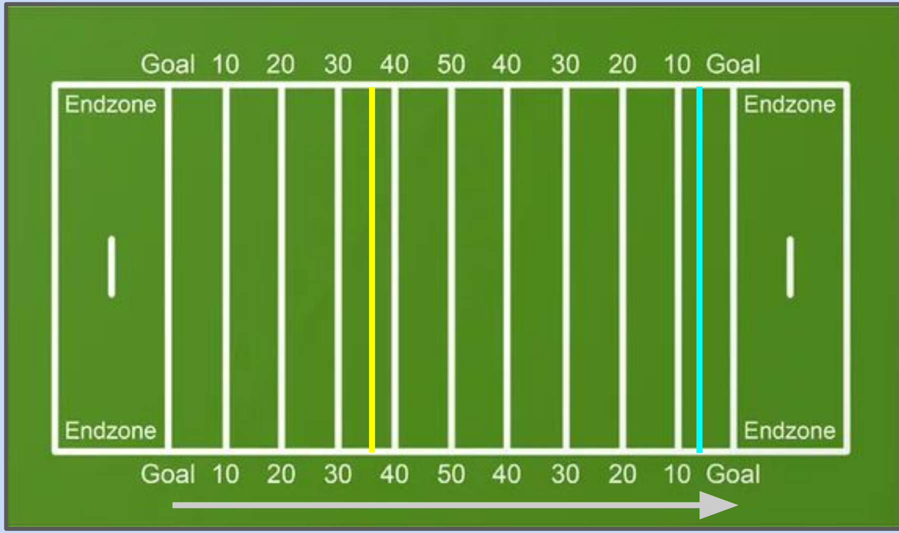
- Three key metrics

- **Pythagorean expectation**: where do points come from?

$$\text{Pythagorean wins} = \frac{\text{points for}^{2.37}}{\text{points for}^{2.37} + \text{points against}^{2.37}}$$

How do we quantify a team’s skill?

- Three key metrics
- **Pythagorean expectation**: where do points come from?
- **Expected points** (EP): what is the expected value of the current game state?



How do we quantify a team's skill?

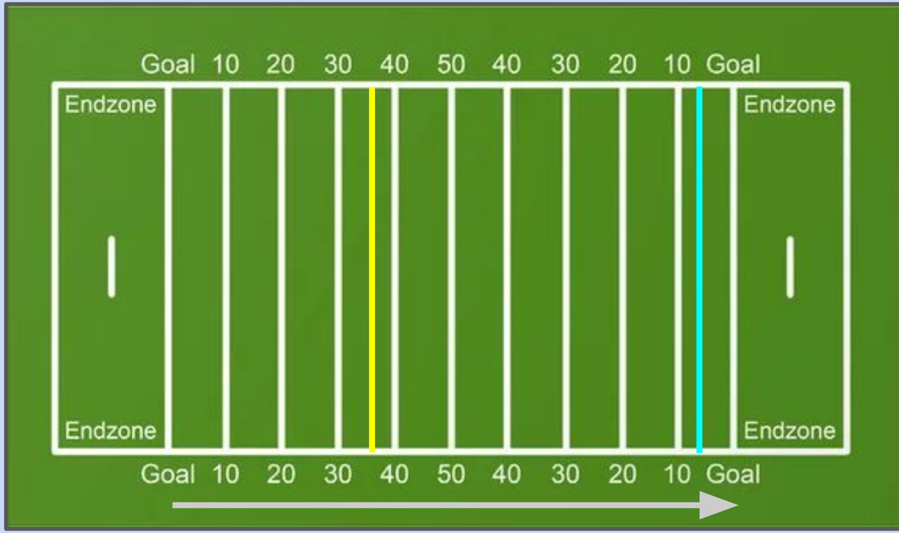
- Three key metrics

- **Pythagorean expectation**: where do points come from?

- **Expected points** (EP): what is the expected value of the current game state?

- **Win probability** (WP): what is the probability of winning given the current game state?

Score: 40-3



Raw data

- Raw data comes in the form of play-by-play outcomes from NFLverse

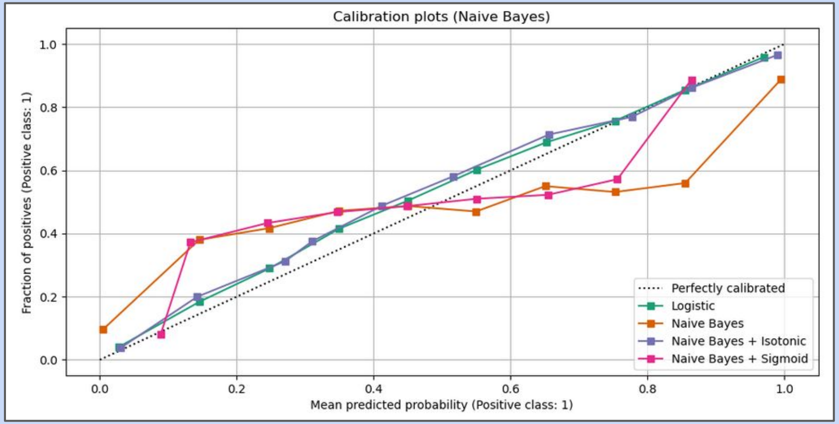
```
RangeIndex: 26353 entries, 0 to 26352
Columns: 372 entries, play_id to pass_oe
dtypes: float64(200), int32(7), object(165)
memory usage: 74.1+ MB
None
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		play_id	game_id	old_game_id	home_team	away_team	season_type	week	posteam	posteam_type	defteam	side_of_field	yardline_100	game_date	quarter_seconds	half_seconds_re	game_seconds_	game_half
2	0	1	2020_01_ARI_S	2020091311	SF	ARI	REG	1						2020-09-13	900	1800	3600	Half1
3	1	39	2020_01_ARI_S	2020091311	SF	ARI	REG	1	SF	home	ARI	ARI	35	2020-09-13	900	1800	3600	Half1
4	2	54	2020_01_ARI_S	2020091311	SF	ARI	REG	1	SF	home	ARI	SF	75	2020-09-13	900	1800	3600	Half1
5	3	93	2020_01_ARI_S	2020091311	SF	ARI	REG	1	SF	home	ARI	SF	55	2020-09-13	882	1782	3582	Half1
6	4	118	2020_01_ARI_S	2020091311	SF	ARI	REG	1	SF	home	ARI	ARI	41	2020-09-13	839	1739	3539	Half1
7	5	143	2020_01_ARI_S	2020091311	SF	ARI	REG	1	SF	home	ARI	ARI	39	2020-09-13	801	1701	3501	Half1
8	6	165	2020_01_ARI_S	2020091311	SF	ARI	REG	1	SF	home	ARI	ARI	45	2020-09-13	759	1659	3459	Half1
9	7	197	2020_01_ARI_S	2020091311	SF	ARI	REG	1	SF	home	ARI	ARI	34	2020-09-13	716	1616	3416	Half1
10	8	226	2020_01_ARI_S	2020091311	SF	ARI	REG	1	ARI	away	SF	SF	35	2020-09-13	710	1610	3410	Half1
11	9	245	2020_01_ARI_S	2020091311	SF	ARI	REG	1	ARI	away	SF	ARI	75	2020-09-13	710	1610	3410	Half1
12	10	274	2020_01_ARI_S	2020091311	SF	ARI	REG	1	ARI	away	SF	ARI	72	2020-09-13	684	1584	3384	Half1

Steps to deploying a prediction model

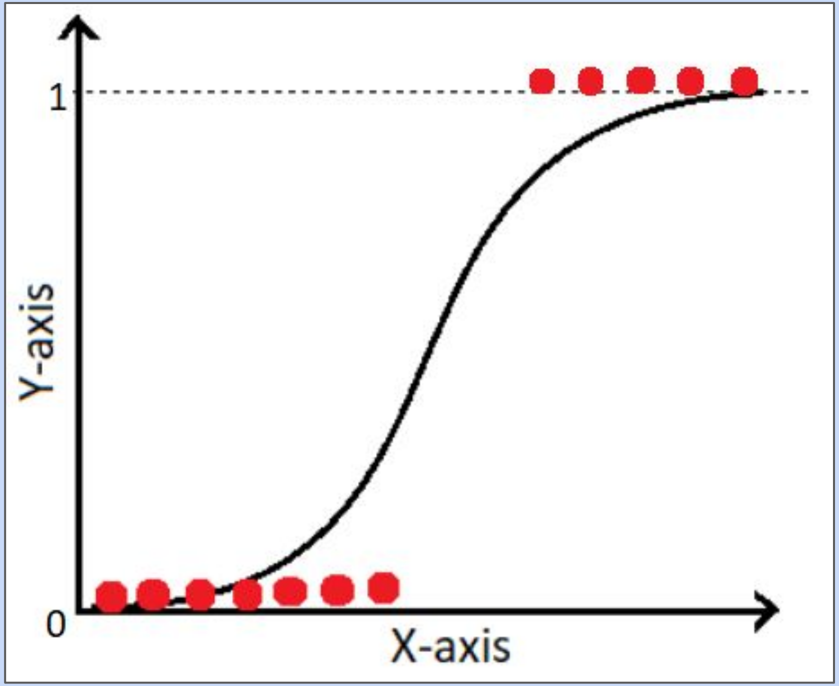
- Select metrics
  - Brier score
  - Calibration curves for validation data

$$BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2$$



Steps to deploying a prediction model

- Select metrics: Brier score and calibration curves
- Train a suitable baseline model
  - Logistic regression on Pythagorean expectation and home/away



Steps to deploying a prediction model

- Select metrics: Brier score and calibration curves

- Train a suitable baseline model: logistic regression

- Train and compare new models
  - Xgboost with EPA and WPA features

Steps to deploying a prediction model

- Select metrics: Brier score and calibration curves

- Train a suitable baseline model: logistic regression

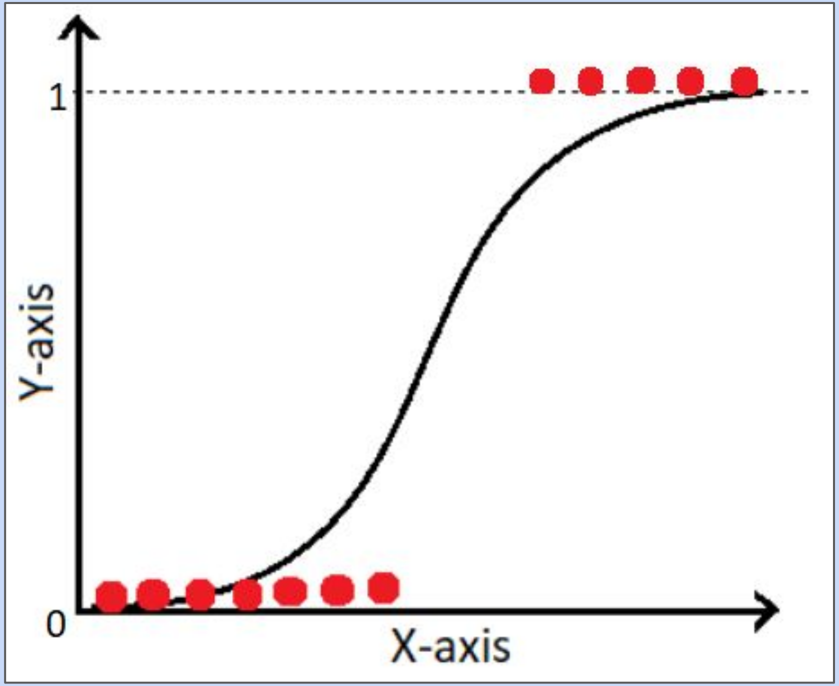
- Train and compare new models

- Deploy the most well-calibrated model or ensemble

Baseline model: logistic regression

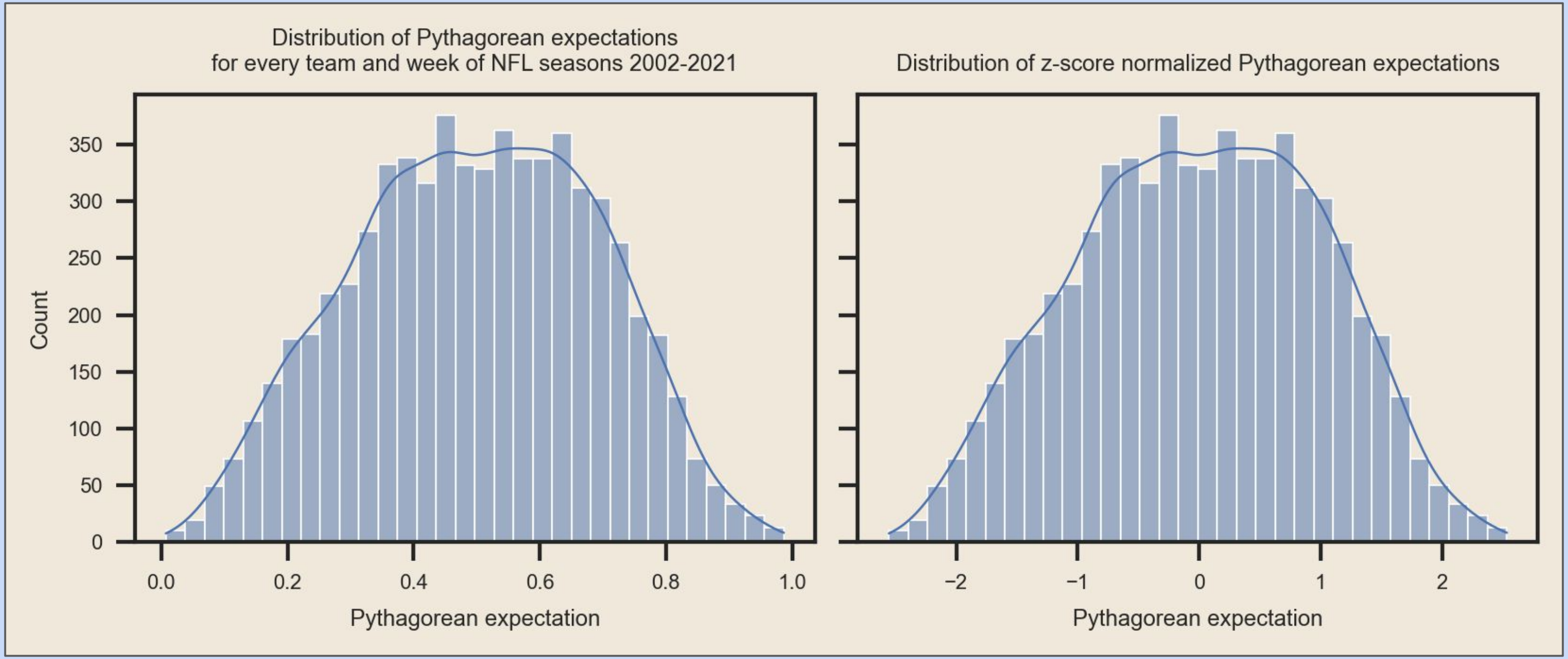
- Trained logistic regression on three features:
  - Object team Pythagorean expectation
  - Adversary Pythagorean expectation
  - Is object team home or away?

```
Index: 3241 entries, 2002_05_ARI_CAR to 2021_15_WAS_PHI
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   obj_pyexp   3241 non-null   float64
 1   adv_pyexp   3241 non-null   float64
 2   is_home     3241 non-null   int64
dtypes: float64(2), int64(1)
memory usage: 101.3+ KB
None
```



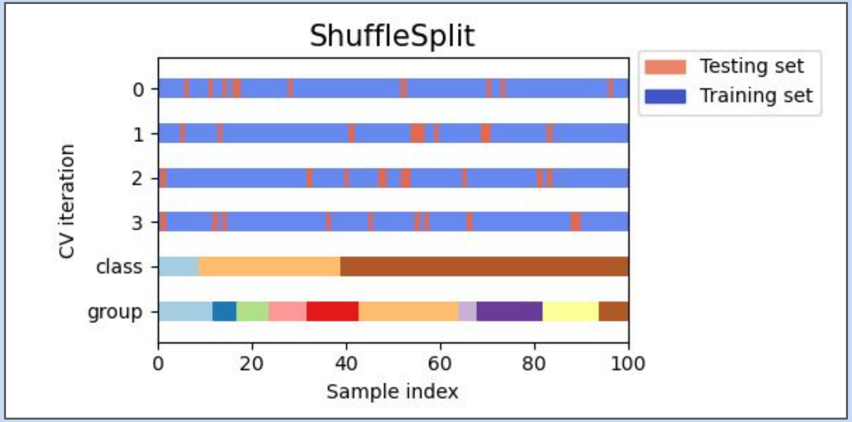
Baseline model: logistic regression

- Pythagorean expectation is z-score normalized before training



Baseline model: logistic regression

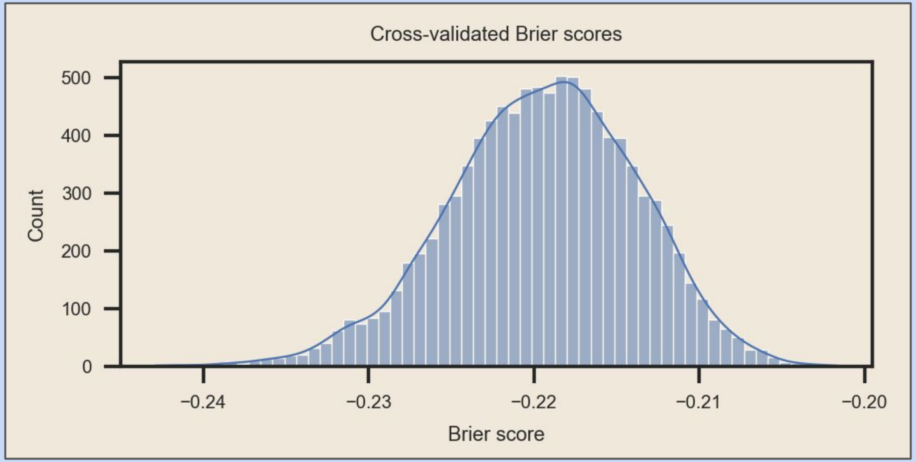
- Held out the latest 1,000 games from the dataset for validation
- Training was cross-validated with 10,000 shuffled and split training/test sets from the remaining rows



Baseline model: logistic regression

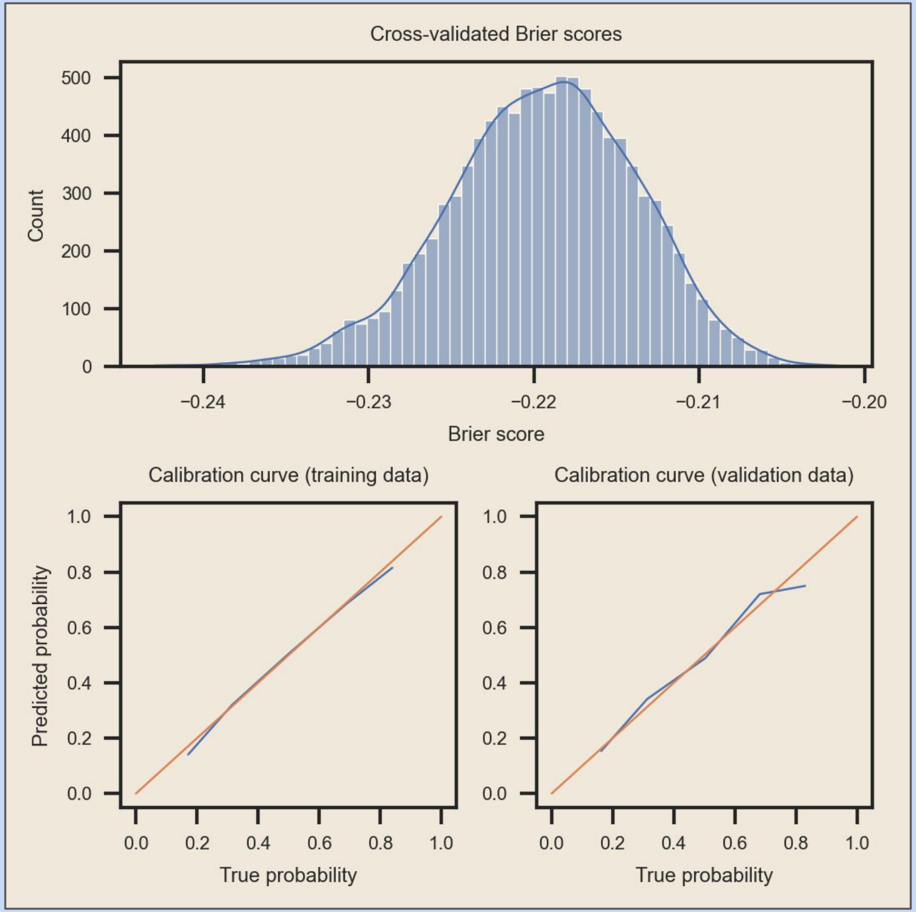
- Held out the latest 1,000 games from the dataset for validation

- Training was cross-validated with 10,000 shuffled and split training/test sets from the remaining rows



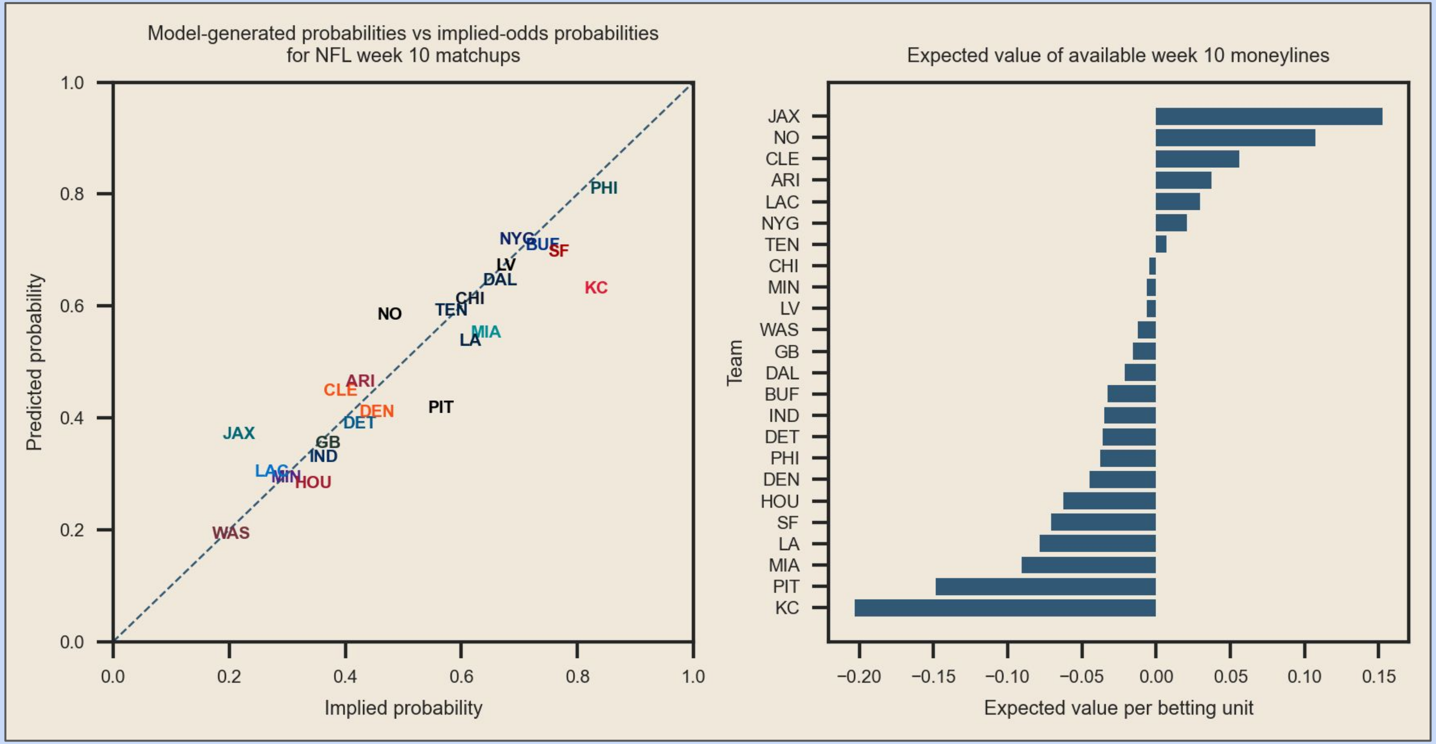
Baseline model: logistic regression

- Held out the latest 1,000 games from the dataset for validation
- Training was cross-validated with 10,000 shuffled and split training/test sets from the remaining rows
- Retrained on training data, then tested on hold-out validation data



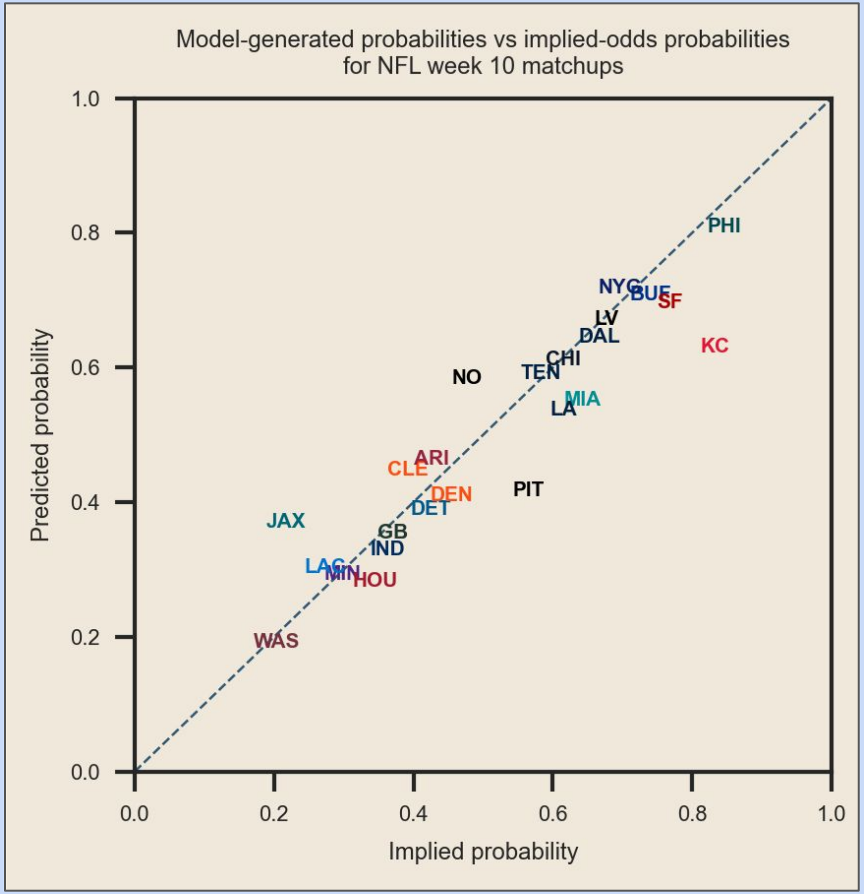
Baseline model: logistic regression

- Predicted probabilities for last week's games



Next Steps

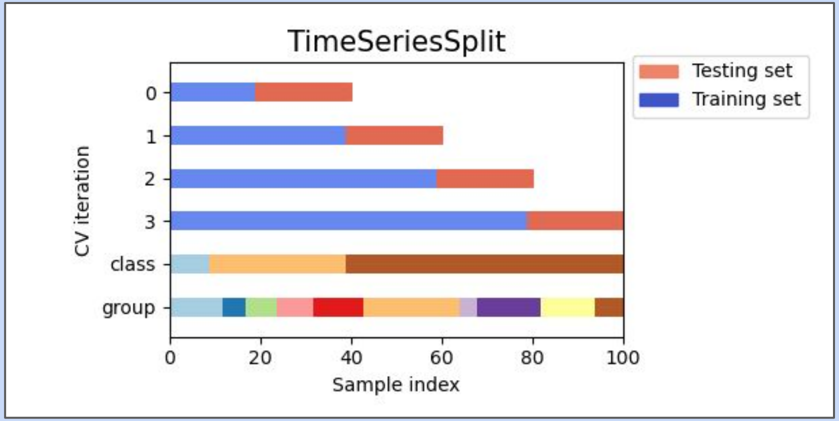
- Error bars for the predicted probability outputs of the baseline model



Next Steps

- Error bars for the predicted probability outputs of the baseline model

- Use a KFold cross-validation strategy over shuffle and split to identify ideal training size



Next Steps

- Error bars for the predicted probability outputs of the baseline model

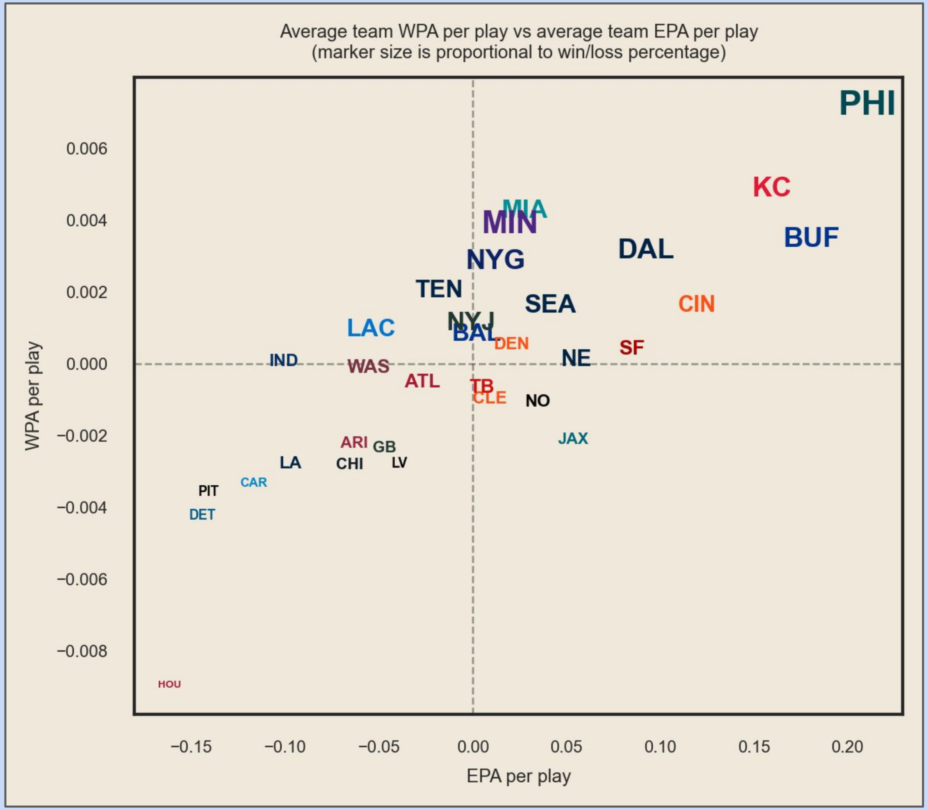
- Use a KFold cross-validation strategy over shuffle and split to identify ideal training size

- Search for a better exponent to the Pythagorean expectation equation

$$\text{Pythagorean wins} = \frac{\text{points for}^{2.37}}{\text{points for}^{2.37} + \text{points against}^{2.37}}$$

Next Model

- Can gradient-boosted decision trees beat the baseline?
- How “lucky” are points? Is there a better feature to consider?



Summary

- Motivation and data science goals
- How the game works
- The baseline model
- Future models