Deloitte.

Predicting MLB player performance. A product of the Statcast era.

Trevor Flanagan

MAKING AN IMPACT THAT MATTERS since 1845

Presentation Overview

1

Business Importance

What is the Statcast Era?

How does this generate useful insights?

2

Modeling Process

What is our source of data?

How are we measuring performance?

How do different models perform?

3

Next Steps

How will the model perform on new data?

What could improve model reliability?

What is the Statcast era?

Overview

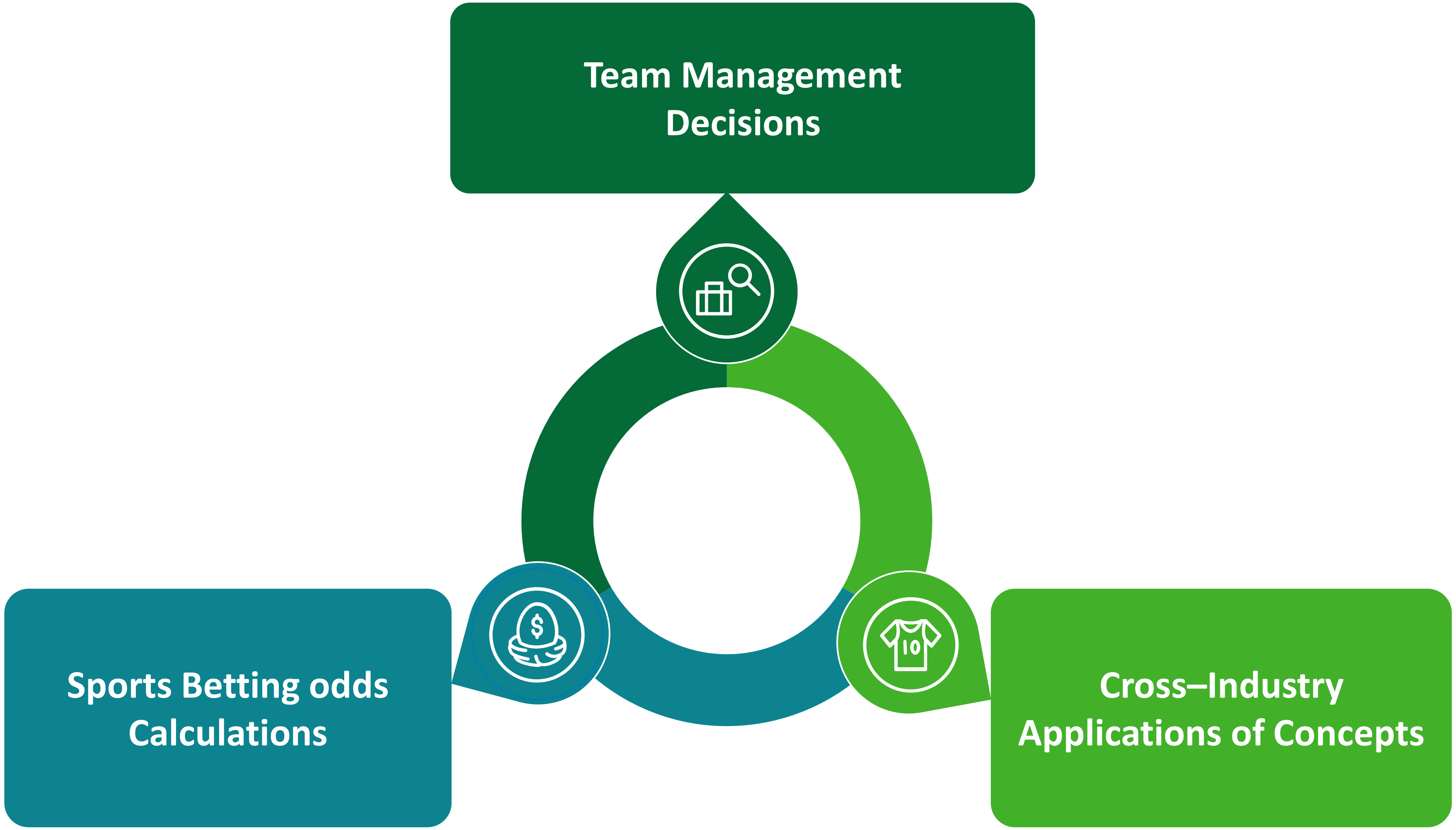
Statcast = State of the art tracking technology

- Installed in all 30 parks in **2015**
- Industry has adapted to run off this data

Statcast Influences

- **Front Office Decision Making**
- **Manager Decision Making**
- **Training and Conditioning**

How does can we translate predicting MLB player performance to business insights?



Modeling Process

01 Load Values

02 Assign X & y

03 Build & Evaluate First Simple Model

04 Reduce Underfitting

05 Reduce Overfitting

06 Evaluate Final Model

Data Source

Pybaseball – Python module

```
from pybaseball import batting_stats
```

Pulls Data from:
- FanGraphs
- Baseball Reference
- Baseball Savant

Filtering Data:
- Data pulled within **last 20 seasons**
- Players need to have at **least 200 plate appearances**
- Exclude players with only 1 season of data

Final Table:
- **7,114 Rows** (Player Seasons)
- **320 Columns** (Players Season Statistics)

	IDfg	Season	Name	Team	Age	G	AB	PA	H	1B	...	maxEV	HardHit	HardHit%	Events	CStr%	CSW%	xBA	xSLG	xwOBA	L-WAR
0	1109	2002	Barry Bonds	SFG	37	143	403	612	149	70	...	NaN	NaN	NaN	0	0.127	0.191	NaN	NaN	NaN	12.7
1	1109	2004	Barry Bonds	SFG	39	147	373	617	135	60	...	NaN	NaN	NaN	0	0.124	0.164	NaN	NaN	NaN	11.9
8	15640	2022	Aaron Judge	NYY	30	157	570	696	177	87	...	118.4	246.0	0.609	404	0.169	0.287	NaN	NaN	NaN	11.2
15	13611	2018	Mookie Betts	BOS	25	136	520	614	180	96	...	110.6	217.0	0.500	434	0.220	0.270	NaN	NaN	NaN	10.4
2	1109	2003	Barry Bonds	SFG	38	130	390	550	133	65	...	NaN	NaN	NaN	0	0.135	0.223	NaN	NaN	NaN	10.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Modeling Process

01 Load Values

02 Assign X & y

03 Build & Evaluate First Simple Model

04 Reduce Underfitting

05 Reduce Overfitting

06 Evaluate Final Model

Target Variable Selection

Target Variable = Next season **WAR** (Wins above replacement)
Add **new variable: Next_WAR**

	IDfg	Name	Season	WAR	Next_WAR
5562	1	Alfredo Amezaga	2006	1.1	2.0
5006	1	Alfredo Amezaga	2007	2.0	1.2
~~5252~~	~~1~~	~~Alfredo Amezaga~~	~~2008~~	~~1.2~~	~~NaN~~
1169	2	Garret Anderson	2002	3.7	5.1
864	2	Garret Anderson	2003	5.1	0.8
2569	2	Garret Anderson	2004	0.8	-0.2
4187	2	Garret Anderson	2005	-0.2	0.1
3964	2	Garret Anderson	2006	0.1	1.4
1925	2	Garret Anderson	2007	1.4	1.4
3346	2	Garret Anderson	2008	1.4	-1.1
~~4937~~	~~2~~	~~Garret Anderson~~	~~2009~~	~~1.1~~	~~NaN~~

Predictor Variables Selection

Sequential Feature Selector

```
from sklearn.feature_selection import SequentialFeatureSelector
```

- **Forward Selection** for **15 most useful predictors**

Modeling Process

01 Load Values

02 Assign X & y

03 Build & Evaluate First Simple Model

04 Reduce Underfitting

05 Reduce Overfitting

06 Evaluate Final Model

Linear Regression Model

$$y = m_1x_1 + m_2x_2 + \ldots + mx_{15} + b$$

Train-Test-Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 100)
```

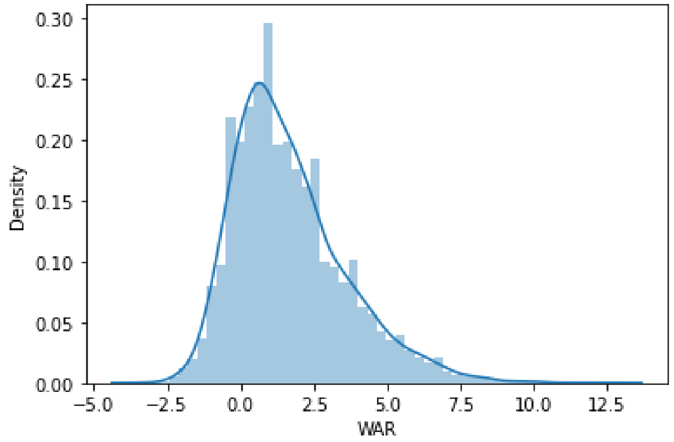
Training data = 75% of original set
Testing data = 25% of original set

Evaluating Model Performance

RMSE

- Training Data = 1.649
- Validation = 1.658

WAR STD = 1.933

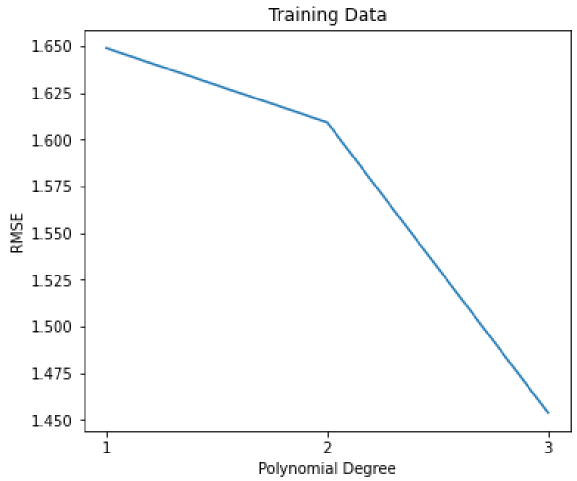


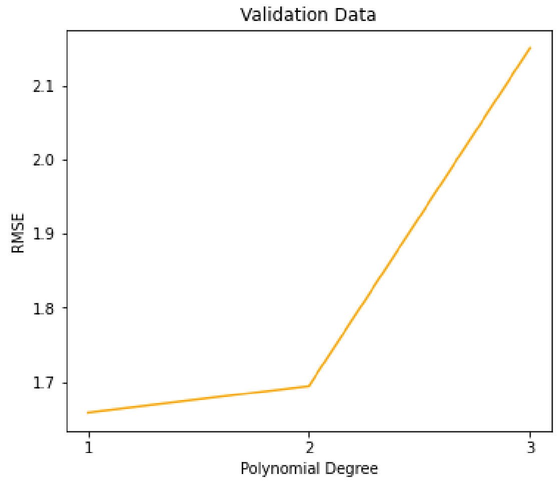
Modeling Process

Adding Polynomial Features

$$y = a_0 + a_1x_1 + a_2x^2 + \ldots + a_nx^n + b$$

Choosing # of Polynomial Features





Evaluating Model Performance

Simple Linear Regression Model:

- Training Data = 1.649
- Validation = 1.658

Models with Polynomial Transformations:

Degree (1)

- Training Data = 1.649
- Validation = 1.658

Degree (2)

- Training Data = 1.609
- Validation = 1.694

Degree (3)

- Training Data = 1.454
- Validation = 2.15

Modeling Process

01 Load Values

02 Assign X & y

03 Build & Evaluate First Simple Model

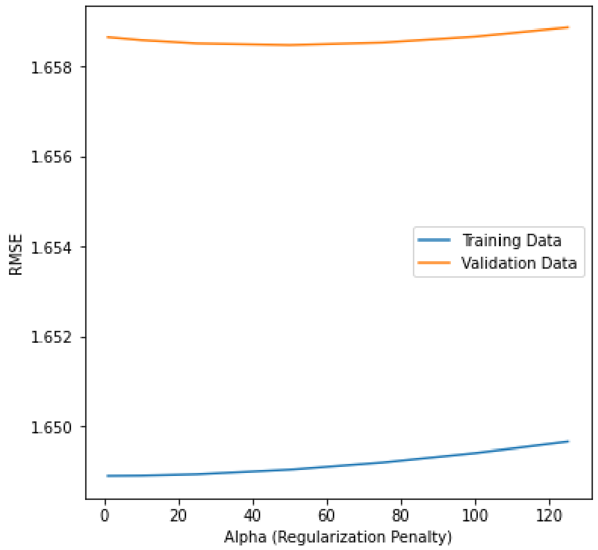
04 Reduce Underfitting

05 Reduce Overfitting

06 Evaluate Final Model

Regularization using Ridge Model

Adding a regularization penalty α



Evaluating Model Performance

Simple Linear Regression Model:

- Training Data = 1.649
- Validation = 1.6586

Ridge Regression Model

- Training Data = 1.649
- Validation = 1.6585

Data Process

01 Load Values

02 Assign X & y

03 Build & Evaluate First Simple Model

04 Reduce Underfitting

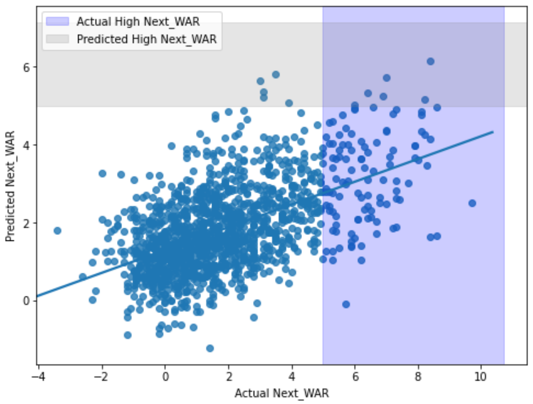
05 Reduce Overfitting

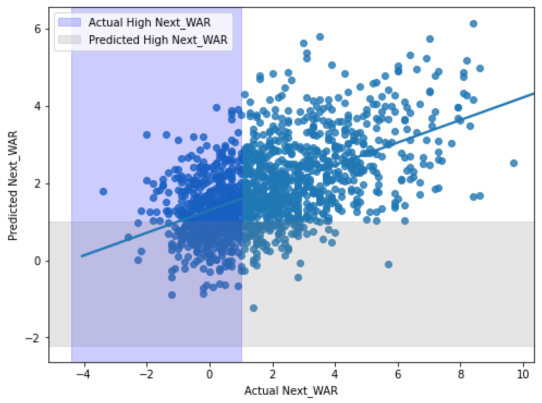
06 Evaluate Final Model

Address level of irreducible error present

Factors that can lead to increased model error:

- **Breakout performances**
- **Statcast era effects on performance**
- **Off season changes**

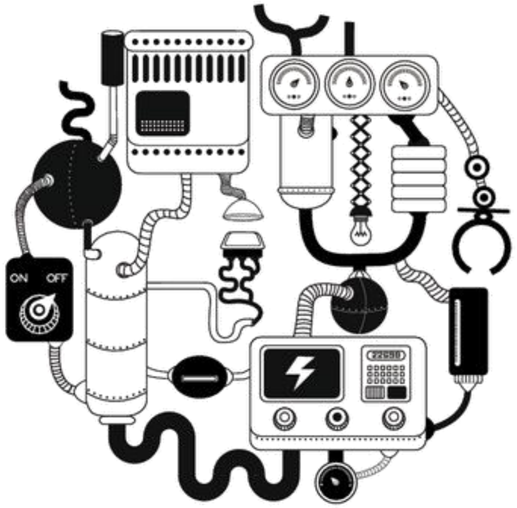




Next Steps

What Could Bolster Model Performance?

- Add **data from Minor Leagues**
- Account for **WAR effect on later seasons** than just the next one
- **Principal Component Analysis** (PCA)



Keep an eye on Model Performance next season

Table showing top 10 short stops in the 2022 season and their predicted WAR in 2022

	Name	Team	Season	WAR	Next_WAR
4	Francisco Lindor	NYM	2022	6.8	6.917300
5	Dansby Swanson	ATL	2022	6.4	1.667297
2	Trea Turner	LAD	2022	6.3	5.832474
0	Xander Bogaerts	BOS	2022	6.1	1.604064
12	Tommy Edman	STL	2022	5.6	-2.443622
7	Willy Adames	MIL	2022	4.7	14.560846
3	Bo Bichette	TOR	2022	4.5	2.100935
6	Corey Seager	TEX	2022	4.5	8.921839
1	Carlos Correa	MIN	2022	4.4	3.902566
11	Nico Hoerner	CHC	2022	4.0	-0.616988

Thank you for your time

Any questions?

Trevor Flanagan

tflanagan@deloitte.com