

# Applying Data Mining to Scouting

Markus Wehr  
Nazih Kalo  
Stephen Stark  
Tam Nguyen  
WooJong Choi





# AGENDA

- 01 Business Use Case
- 02 Analytics in the Marketplace
- 03 Data Overview, EDA, Engineering
- 04 Client Pipeline
- 05 Model Engineering
- 06 Anomaly Detection
- 07 Bid Prediction
- 08 Next Steps

A soccer ball is depicted with intense, swirling flames and wisps of smoke, suggesting it is on fire. The ball is white with black pentagonal panels. The background is a dark, smoky environment with a jagged, torn paper effect along the right edge.

# Business Use Case

# Business Use Case



## Scouting meets Advanced Analytics

### Scouting yesterday

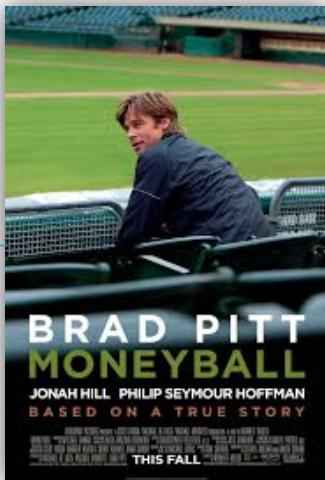
- Observation
- Rudimentary data
- Intuition

VS

### Scouting today

- Advanced analytics
  - availability of massive data
  - ability to process & capture insight

### The hit movie: Moneyball (2011)



- The potential of unconventional sabermetrics in sport
- Scouting in soccer is a global challenge.

- Poor performing clubs face relegation which has an immediate impact on the club's bottom line
- Important for small \$\$\$ teams to use analytics to compete with larger clubs

### Business Objectives

We are positioning ourselves as a scouting agency that:

- uses the **FIFA 2018 dataset** and
- apply various **data mining methods** to:

1

Enhance the **discovery of talents**

2

Help soccer clubs better understand the **dynamics (features)** that come into play when determining the **value** of a player

### Key Assumptions

Our dataset reflects information up to Summer 2018.

Market values are not biased and reflect the true intrinsic value of the player. We understand that may not be the case, but for the purpose of our models, we assume that it is.

All feature scores, which are developed by an independent third party, are accurate and reflective of the true player style. These features are reflective of historical performance



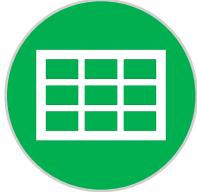
# Data Overview, EDA, Engineering



# Data - Overview

## Features

### Dataset



- CSV
- 18207(R) x 89(C)

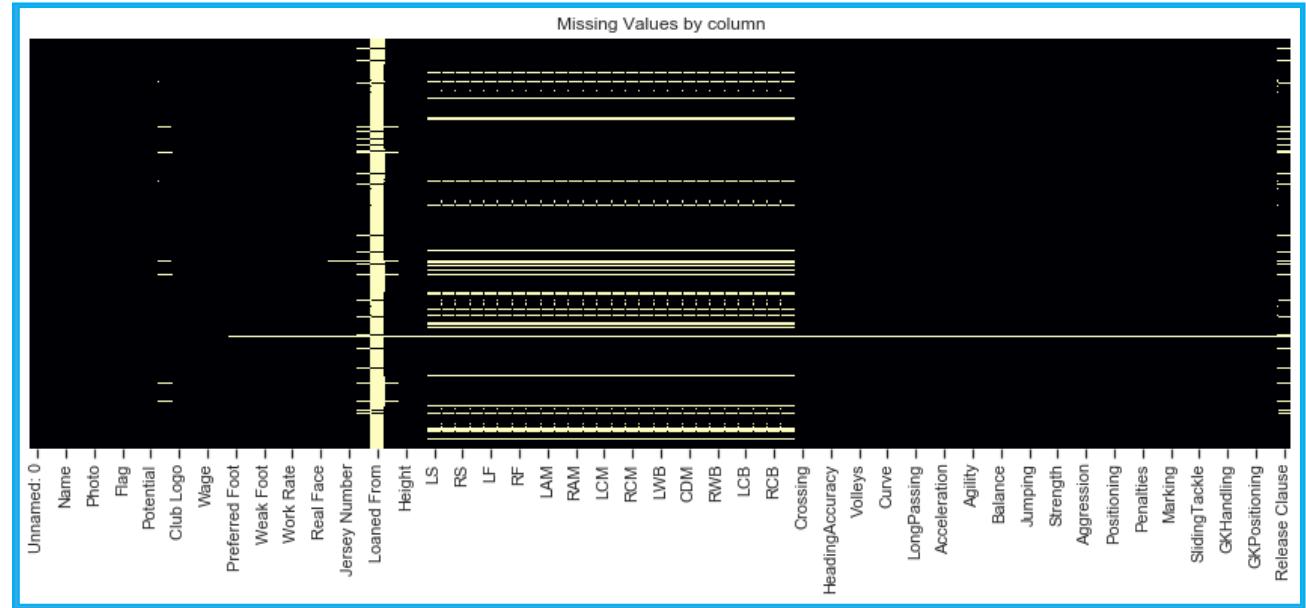
Profile		
• ID	• Club	• Joined
• Name	• Club Logo	• Loaned From
• Age	• Preferred Foot	• Contract Valid Until
• Height	• Weak Foot	• Int. Reputation
• Weight	• Body Type	• Photo
• Nationality	• Real Face	
• Flag	• Jersey Number	

Position Related		
• Position	• LS	• LAM
	• ST	• CAM
	• RS	• RAM
	• LW	• LM
	• LF	• LCM
	• CF	• CM
	• RF	• RCM
	• RW	• RM
		• LDM
		• CDM
		• RDM
		• RWB
		• LB
		• LCB
		• CB
		• RCB
		• RB

Attributes/Skills								
• Overall	• Crossing	• Dribbling	• Acceleration	• ShotPower	• Aggression	• Marking	• GKDiving	
• Potential	• Finishing	• Curve	• SprintSpeed	• Jumping	• Interceptions	• StandingTackle	• GKHandling	
• Special	• HeadingAccuracy	• FKAccuracy	• Agility	• Stamina	• Positioning	• SlidingTackle	• GKKicking	
• Skill Moves	• ShortPassing	• LongPassing	• Reactions	• Strength	• Vision		• GKPositioning	
• Work Rate	• Volleys	• BallControl	• Balance	• LongShots	• Penalties		• GKReflexes	
					• Composure			

\$\$\$		
• Value	• Wage	• Release Clause

## MISSING VALUES



# Data Processing & Feature Engineering



## Original Data

Feature	Data Type	Missing Values
ID	Categorical	-
Name	Text	-
Age	Numerical	-
Height	Text	48
Weight	Text	48
Nationality	Categorical	-
Flag	Categorical	-
Club	Categorical	241
Club Logo	Text	
Preferred Foot	Categorical	48
Weak Foot	Numerical	48
Body Type	Categorical	48
Real Face	Categorical	48
Jersey Number	Categorical	60
Joined	Date	1553
Loaned From	Categorical	16943
Contract Valid Until	Date	289

## After Data Processing & Feature Engineering

Processing/Feature Engineering	Imputation / Drop	Data Type
Dropped	-	-
Dropped	-	-
-	-	Numerical
Converted inches to centimeters	48 missing rows dropped	Numerical
Removed the text "lbs" and converted to integer	48 missing rows dropped	Numerical
Dropped and new column "Continent" created to assign continent instead	0	Dummy
Dropped	-	-
Dropped and new column "Club Reputation" created by taking the mean of 'International Reputation' for players for each club	Filled in missing values with "No_club"	Numerical
Dropped	-	-
Converted to Binary: 0 = left, 1 = right	48 missing rows dropped	Categorical
No change	48 missing rows dropped	Numerical
Removed one-off body types and replaced them with either "lean", "stocky" and "normal" based on domain knowledge	48 missing rows dropped	Numerical
Converted to Binary: 0 = No, 1 = Yes	48 missing rows dropped	Categorical
No change	48 missing rows dropped. 12 remaining missing values were filled in using the mode Jersey Number of the player's position	Categorical
Converted to int: 2019/1/1 - Joined Date	Filled in missing values with 0	Numerical
Converted to Binary: 0 = Not on loan, 1 = On loan	Missing value means the player is not on loan. These missing values are assigned 0	Categorical
Converted to int: years of contract left from 2018	Filled in missing values with 0 (expired)	Numerical



# Data Processing & Feature Engineering



## Original Data

Feature	Data Type	Missing Values
Position	Categorical	60
LS	Text	2085
	Text	2085
RB	Text	2085

## After Data Processing & Feature Engineering

Processing/Feature Engineering	Imputation / Drop	Data Type
Position_Group column created that assigns one of the following to the player: Forward, Midfielder, Defender, GoalKeeper, Other (no position)	Players assigned Other originally did not have a position, but later imputed based on the players' max ability from Attacking, Defending, GoalKeeping	Dummy
"+ int" removed and a new column created to capture just the int. Column converted to integer.	2025 missing values are Goalkeepers, who do not have a value for this column	Dummy
"+ int" removed and a new column created to capture just the int. Column converted to integer.	2025 missing values are Goalkeepers, who do not have a value for this column	Dummy
"+ int" removed and a new column created to capture just the int. Column converted to integer.	2025 missing values are Goalkeepers, who do not have a value for this column	Dummy
+ int" removed and a new column created to capture just the int. Column converted to integer.	2025 missing values are Goalkeepers, who do not have a value for this column	Dummy
+ int" removed and a new column created to capture just the int. Column converted to integer.	2025 missing values are Goalkeepers, who do not have a value for this column	Dummy
+ int" removed and a new column created to capture just the int. Column converted to integer.	2025 missing values are Goalkeepers, who do not have a value for this column	Dummy
+ int" removed and a new column created to capture just the int. Column converted to integer.	2025 missing values are Goalkeepers, who do not have a value for this column	Dummy
+ int" removed and a new column created to capture just the int. Column converted to integer.	2025 missing values are Goalkeepers, who do not have a value for this column	Dummy



# Data Processing & Feature Engineering



## Original Data

Feature	Data Type	Missing Values
Overall	Numerical	-
Potential	Numerical	-
Special	Numerical	-
Skill Moves	Numerical	48
Work Rate	Categorical	48
* Attributes x 34	Numerical	48
Value	Text	
Wage	Text	
Release Clause	Text	1564

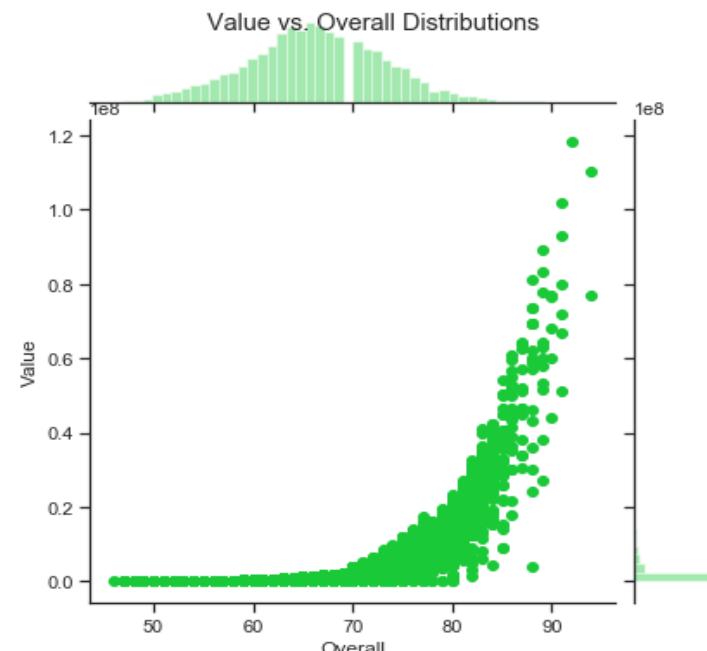
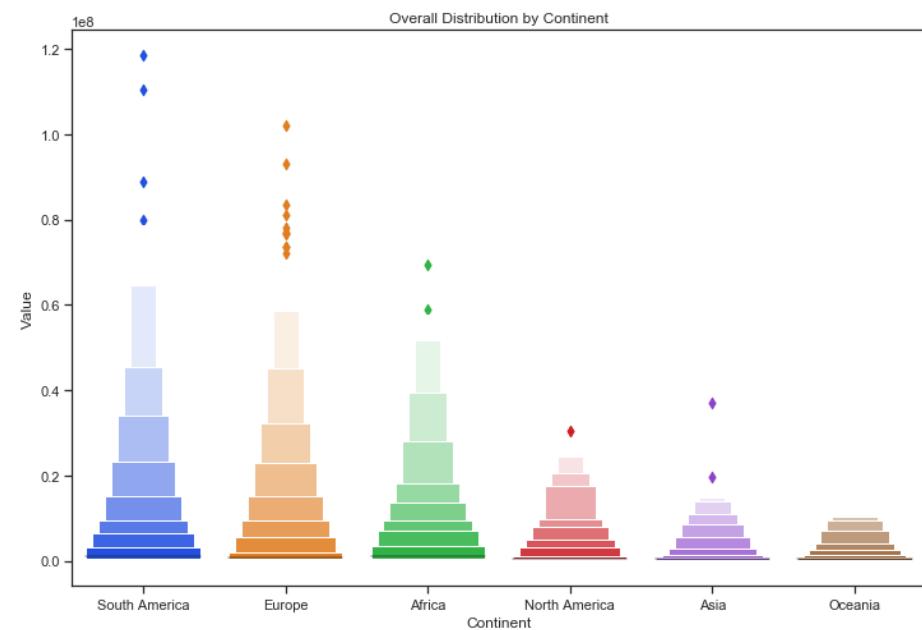
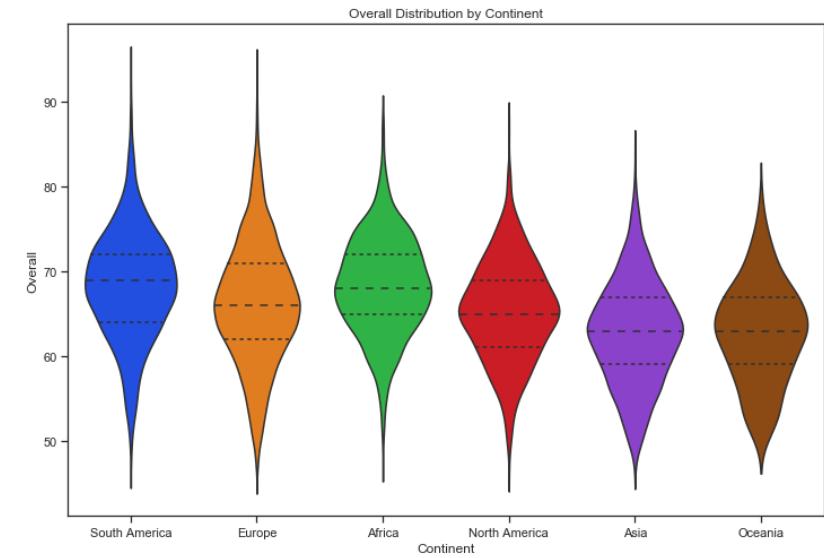
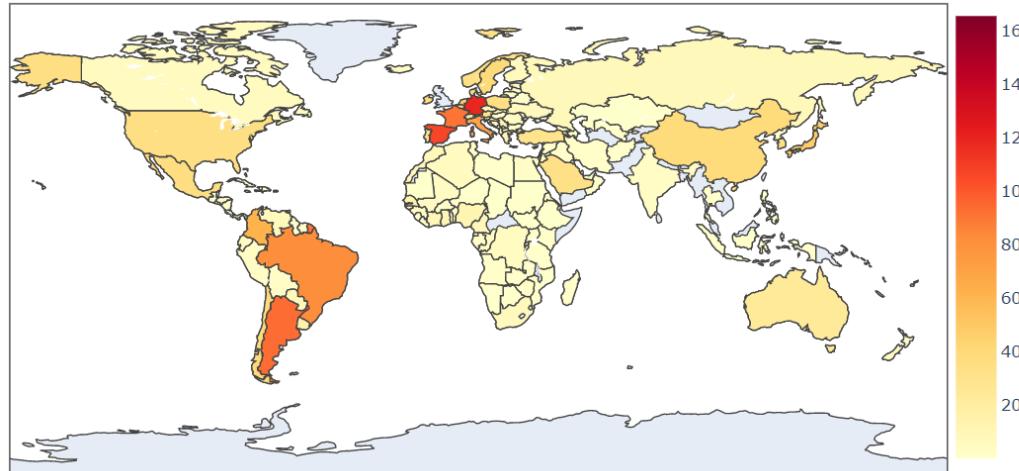
## After Data Processing & Feature Engineering

Processing/Feature Engineering	Imputation / Drop	Data Type
-	-	Numerical
-	-	Numerical
-	-	Numerical
-	48 missing rows dropped	Numerical
Dropped and created new columns "Attack_WR" and "Defense_WR"	48 missing rows dropped	Numerical
7 New columns created "Attack", "Skill", "Movement", "Power", "Mentality", "Defending", "GoalKeeping" and assigned with means of attributes that belong to the group	48 missing rows dropped	Numerical
Removed currency signs and converted to integer.		Numerical
Removed currency signs and converted to integer.		Numerical
Removed currency signs and converted to integer	Missing values filled in with 0	Numerical

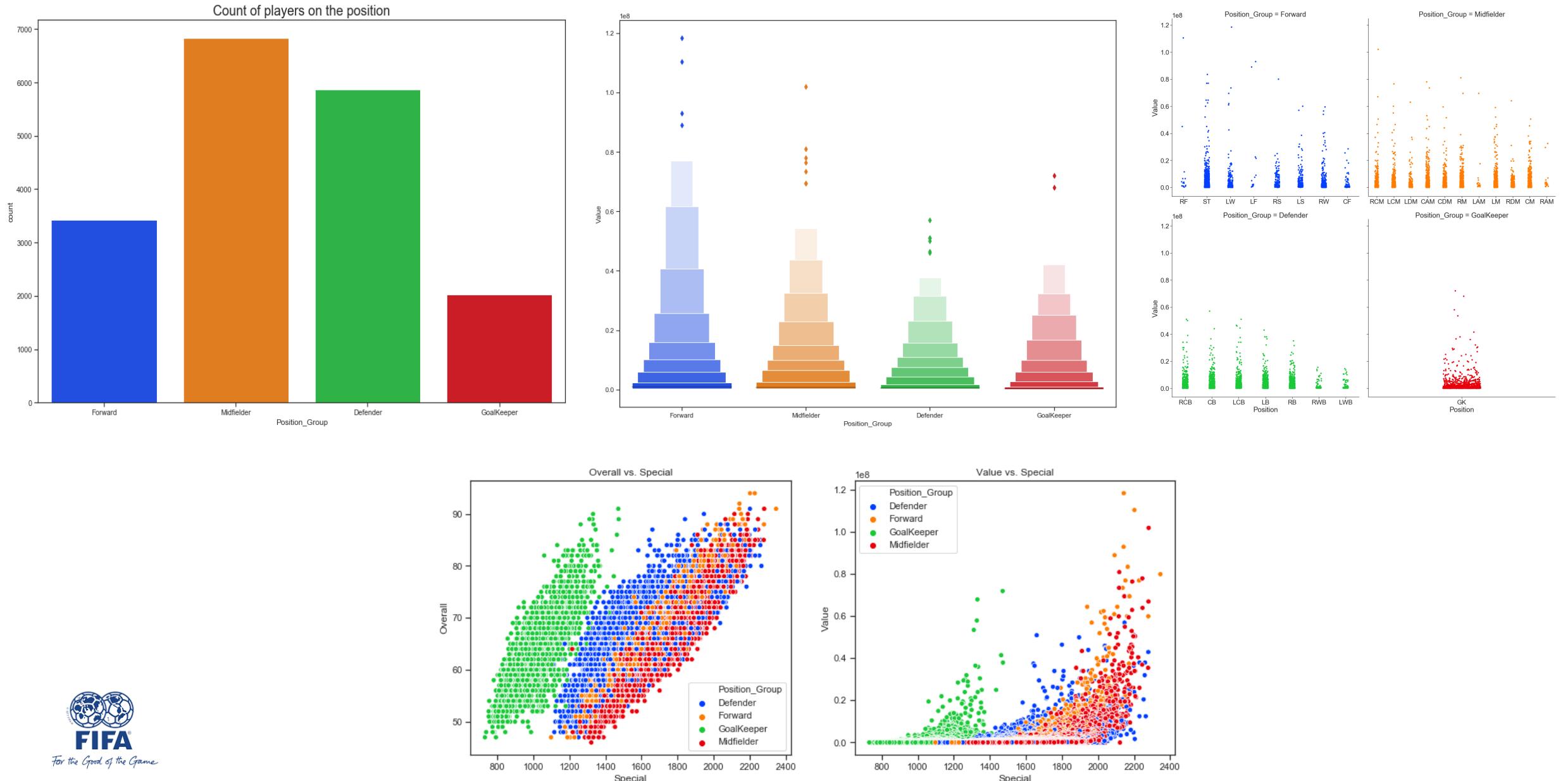
**Summary:**  
18159 rows x 125 columns



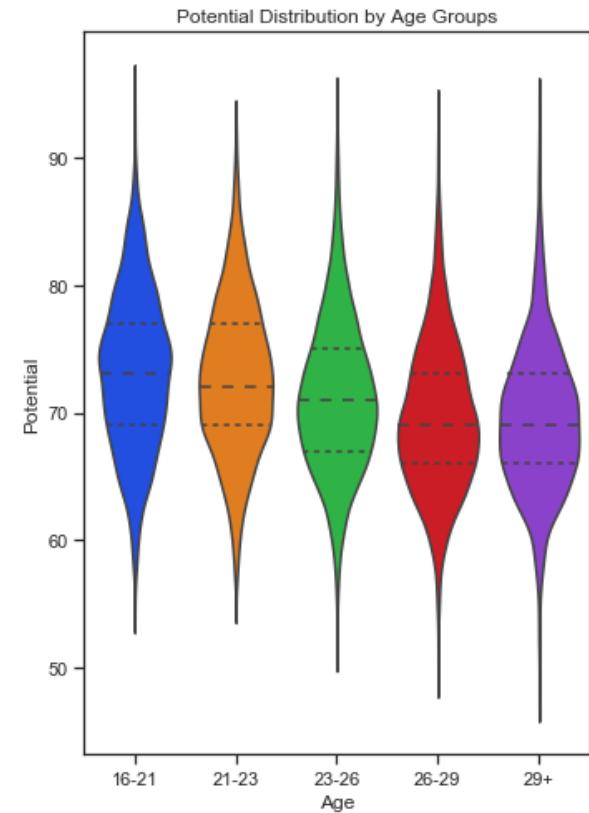
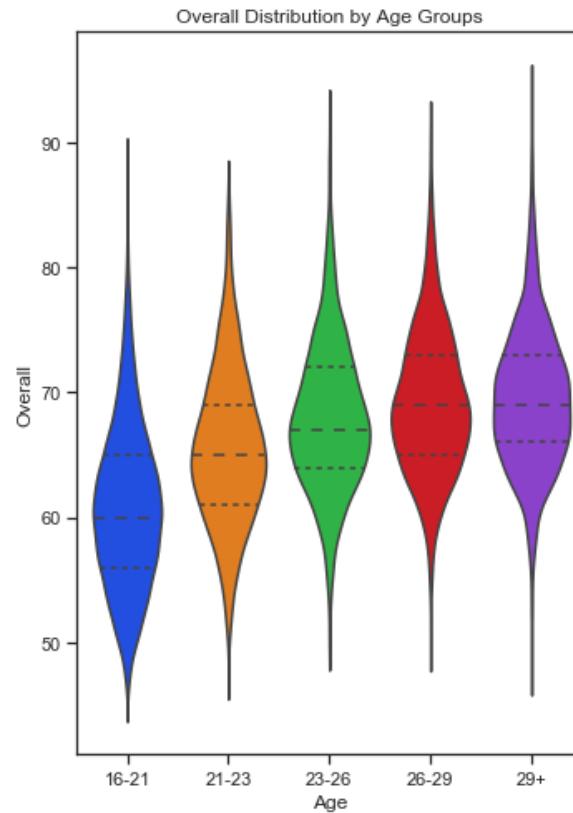
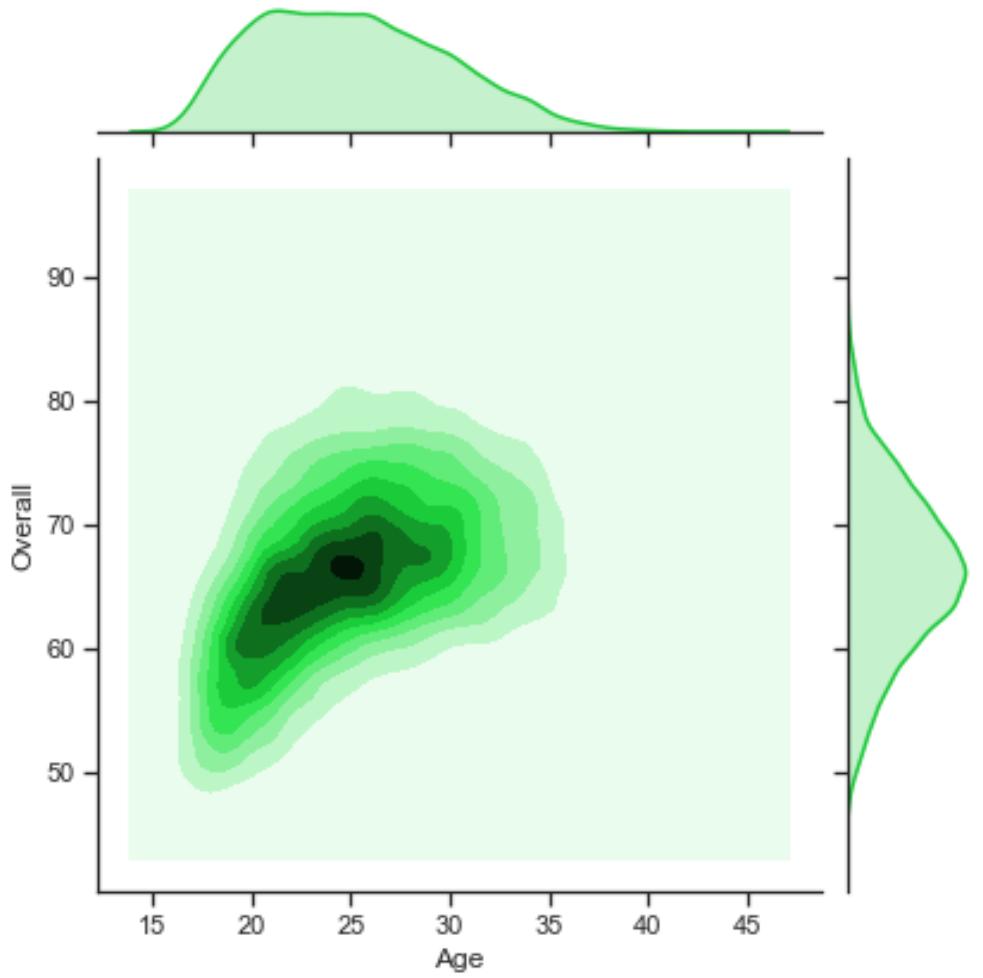
# EDA – Visualization (1/3)

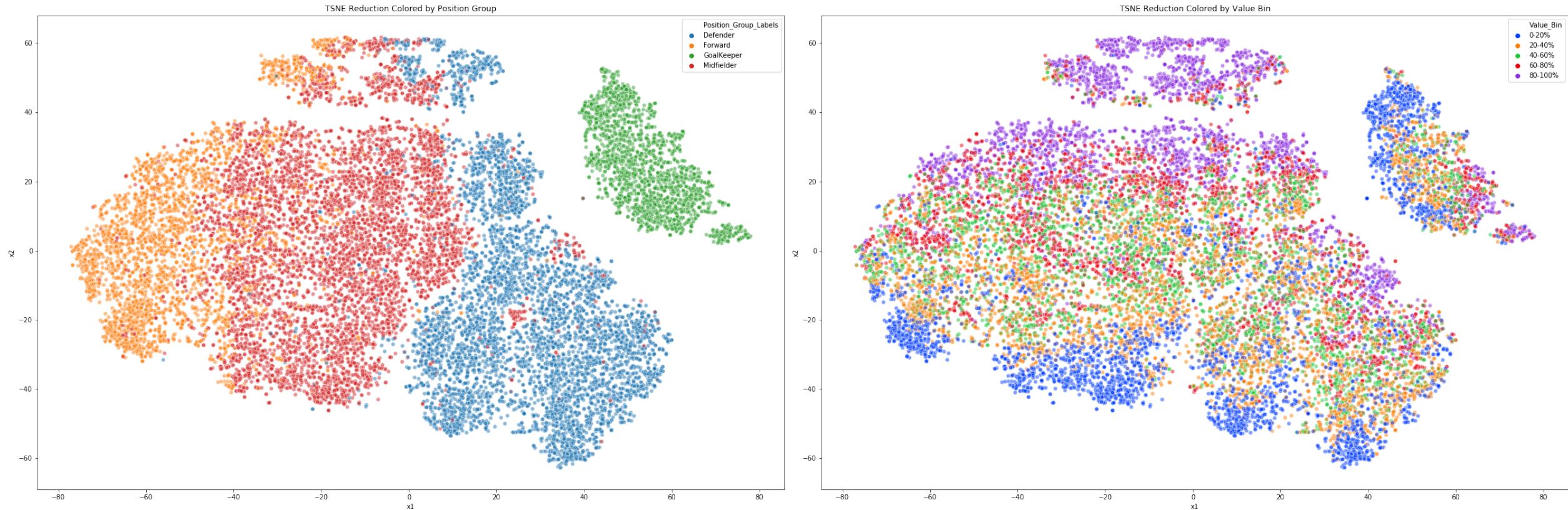


# EDA – Visualization (2/3)



# EDA – Visualization (3/3)





TSNE reduction shows clustering of position groups...

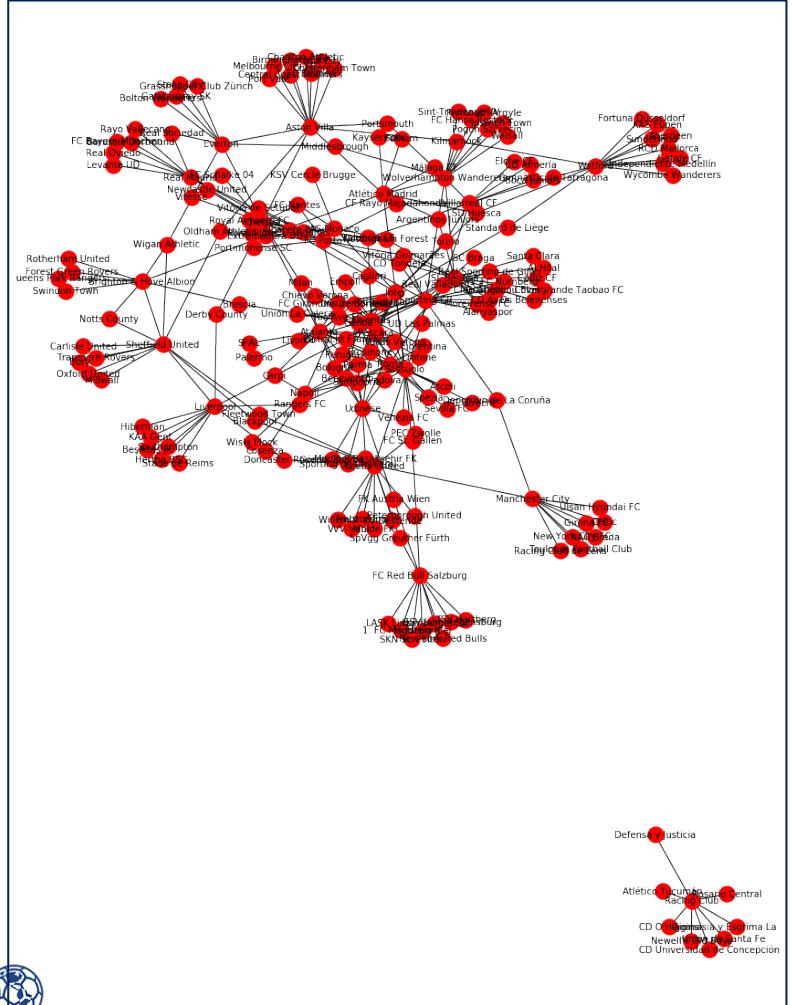
and within these position group clusters there is additional clustering of players by valuation (\$) level.



# Analysis on players on loan: Graph

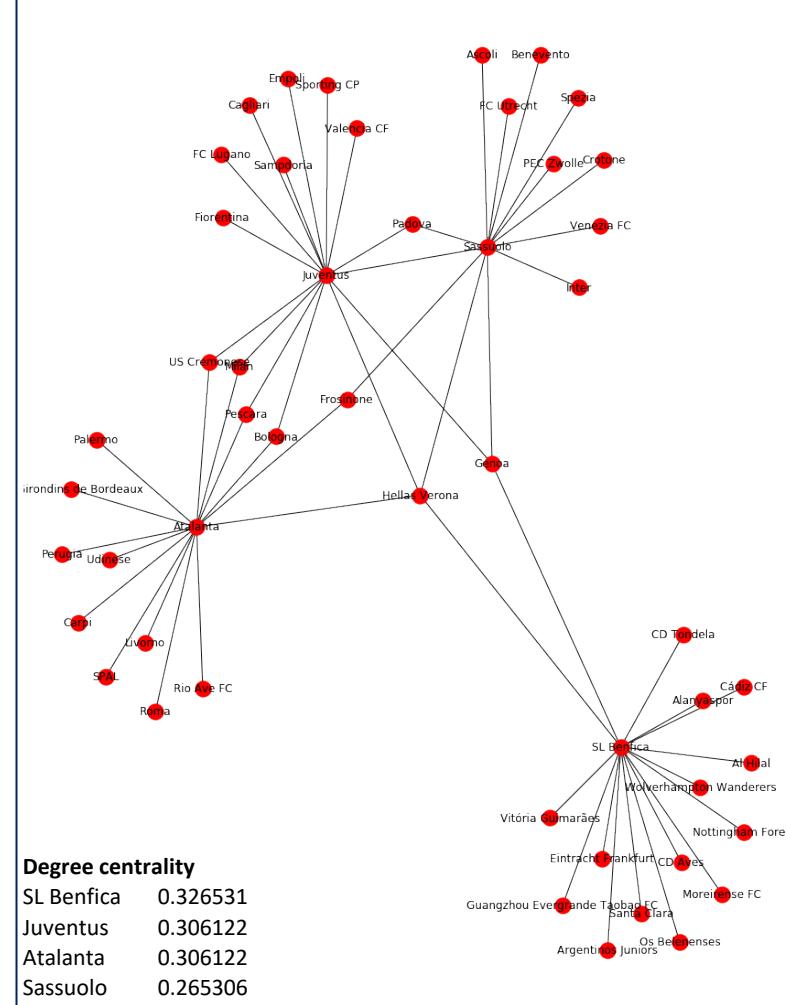
Players on loan: 1265

Clubs that loaned out **10 or more** players



England  
Italy  
Portugal  
Spain  
France  
Austria

Clubs that loaned out **15 or more** players



England 10  
Italy 8  
Portugal 3  
Spain 3  
France 1  
Austria 1

Italy 3  
Portugal 1

**Why?** Italy doesn't have B teams, so they send young players out on loan to give them playing time.  
(B teams allowed in Italy from 2019 so this pattern may change)





01

Create Restricted Set of Recommended Players



- K-Nearest Neighbors

02

Isolate Outlier Players

Anomaly Detection:

- SVM-One Class
- Local Outlier factor
- Isolation Forest
- DBSCAN

03

Predict Bid Price for Isolate Outlier Players



- Linear Regression
- Decision Tree
- Random Forest
- XGBoost
- SVR





# Model Engineering

# Client Pipeline Process



Give me 300 hundred  
players similar to  
“M. Salah”



01

Create Restricted  
Set of Recommended  
Players



- K-Nearest Neighbors

02

Isolate Outlier  
Players

Anomaly Detection:  

- SVM-One Class
- Local Outlier factor
- Isolation Forest
- DBSCAN

03

Predict Bid Price for  
Isolate Outlier  
Players

- Linear Regression
- Decision Tree
- Random Forest
- XGBoost
- SVR





# Filtering Functions

## Option #1:

**filter\_players(position, ovr\_min = 0, ovr\_max= 100)**

Accepts a position name and overall range and returns a filtered list & dataframe of the players that meet those criteria

**Step 1:** Enter the position looking for:

CM

**Step 2:** What is the min overall?:

74

**Step 3:** What is the max overall?:

86

## Output

Here are the filtered players based on your criteriea:

```
['Thiago',
'S. Milinković-Savić',
'Jorginho',
'I. Gündoğan',
'N. Keita',
'C. Tolisso',
'A. Rabiot',
'L. Goretzka',
'J. Draxler',
'Cesc Fàbregas',
'M. Dembélé',
'Rodri',
```

Here are the filtered players' features based on your criteriea:

Age	Overall	Potential	Special	Preferred Foot	International Reputation	Weak Foot	Skill Moves	Real Face	Height	Weight	LS	ST
67	27	86	86	2190	1	3.0	3.0	5.0	1	175	154	75 75
78	23	85	90	2206	1	2.0	4.0	4.0	1	190	168	81 81
121	26	84	87	2136	1	2.0	3.0	3.0	0	180	148	70 70
136	27	84	84	2138	1	3.0	4.0	4.0	1	180	176	75 75
161	23	83	88	2082	1	2.0	4.0	4.0	1	173	141	73 73
162	23	83	88	2207	1	2.0	3.0	3.0	1	180	179	78 78
168	23	83	87	2184	0	2.0	3.0	3.0	1	193	176	77 77
169	23	83	88	2203	1	3.0	4.0	3.0	1	188	174	77 77
184	24	83	86	2112	1	3.0	5.0	4.0	1	188	170	79 79



For the Good of the Game

## Option #2:

**recommended\_k\_players\_df(player, k\_players = 100)**

Accepts a player's name and number of players to recommend and returns a dataframe of the recommended players and a list of their names. The recommendations are limited to players from the same position group.

**Step 1:** Enter the player you are looking for:

M. Salah

**Step 2:** Enter the number of similar players you are looking for:

300

## Output

Here are 300 players similar to M. Salah:

0	L. Messi
1	Cristiano Ronaldo
2	Neymar Jr
4	K. De Bruyne
5	E. Hazard
6	L. Modrić
7	L. Suárez
10	R. Lewandowski
11	T. Kroos
13	David Silva
15	P. Dybala
16	H. Kane
17	A. Griezmann
20	Sergio Busquets
21	F. Cavani



Here are the players' features

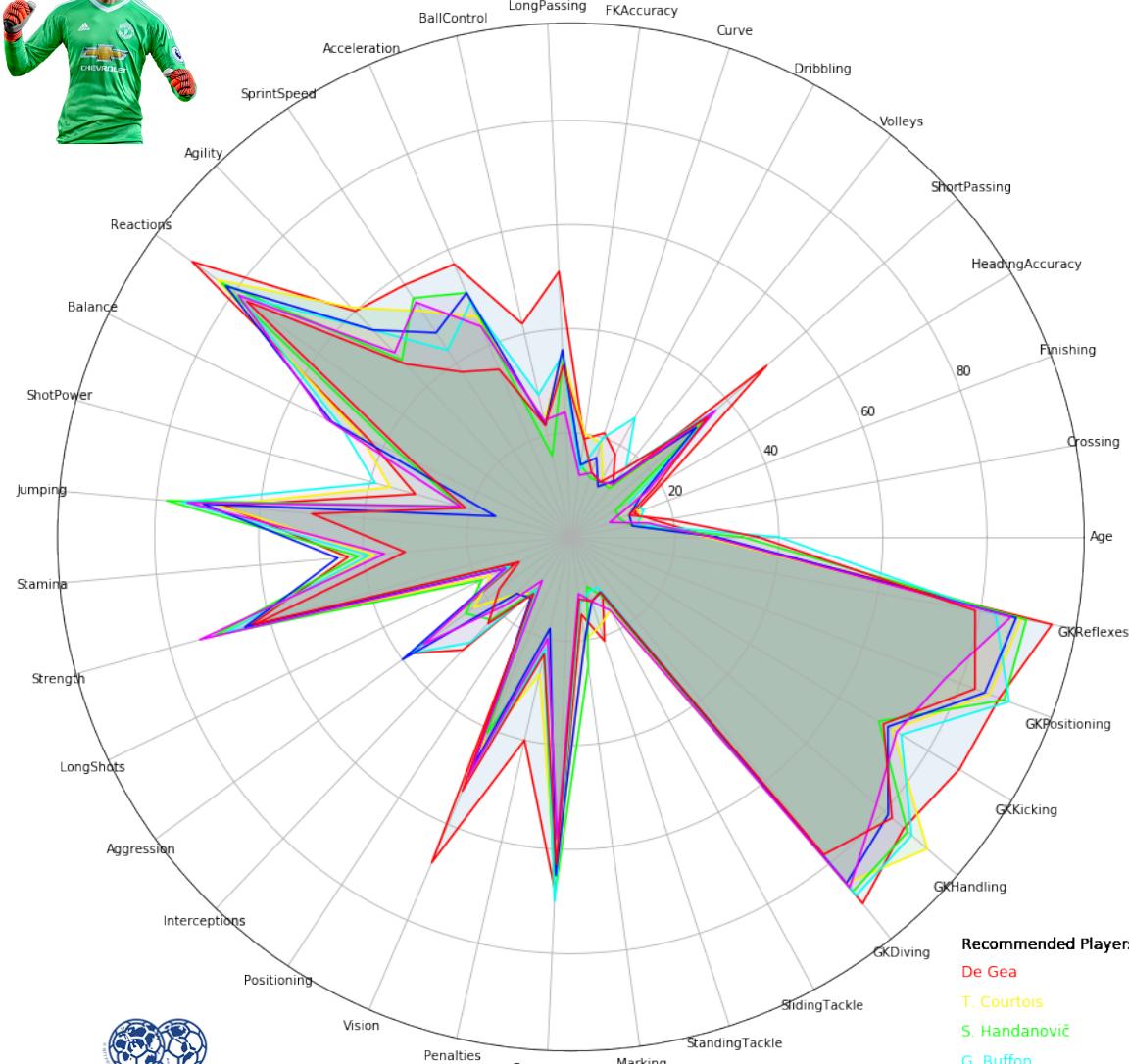
	Age	Overall	Potential	Special	Preferred Foot	International Reputation	Weak Foot	Skill Moves	Real Face	Height	Weight	LS	ST
G. Bale	28	88	88	2279	0	4.0	3.0	4.0	1	185	181	86	86
A. Griezmann	27	89	90	2246	0	4.0	3.0	4.0	1	175	161	86	86
M. Reus	29	86	86	2172	1	4.0	4.0	4.0	1	180	157	82	82
R. Lewandowski	29	90	90	2152	1	4.0	4.0	4.0	1	183	176	87	87
A. Sánchez	29	85	85	2172	1	4.0	3.0	4.0	1	170	163	81	81
P. Pogba	25	87	91	2247	1	4.0	4.0	5.0	1	193	185	81	81
I. Perišić	29	85	85	2199	1	3.0	5.0	4.0	1	185	176	82	82
E. Cavani	31	89	89	2161	1	4.0	4.0	3.0	1	185	170	85	85



# Analyzing Recommendation Feature Similarities



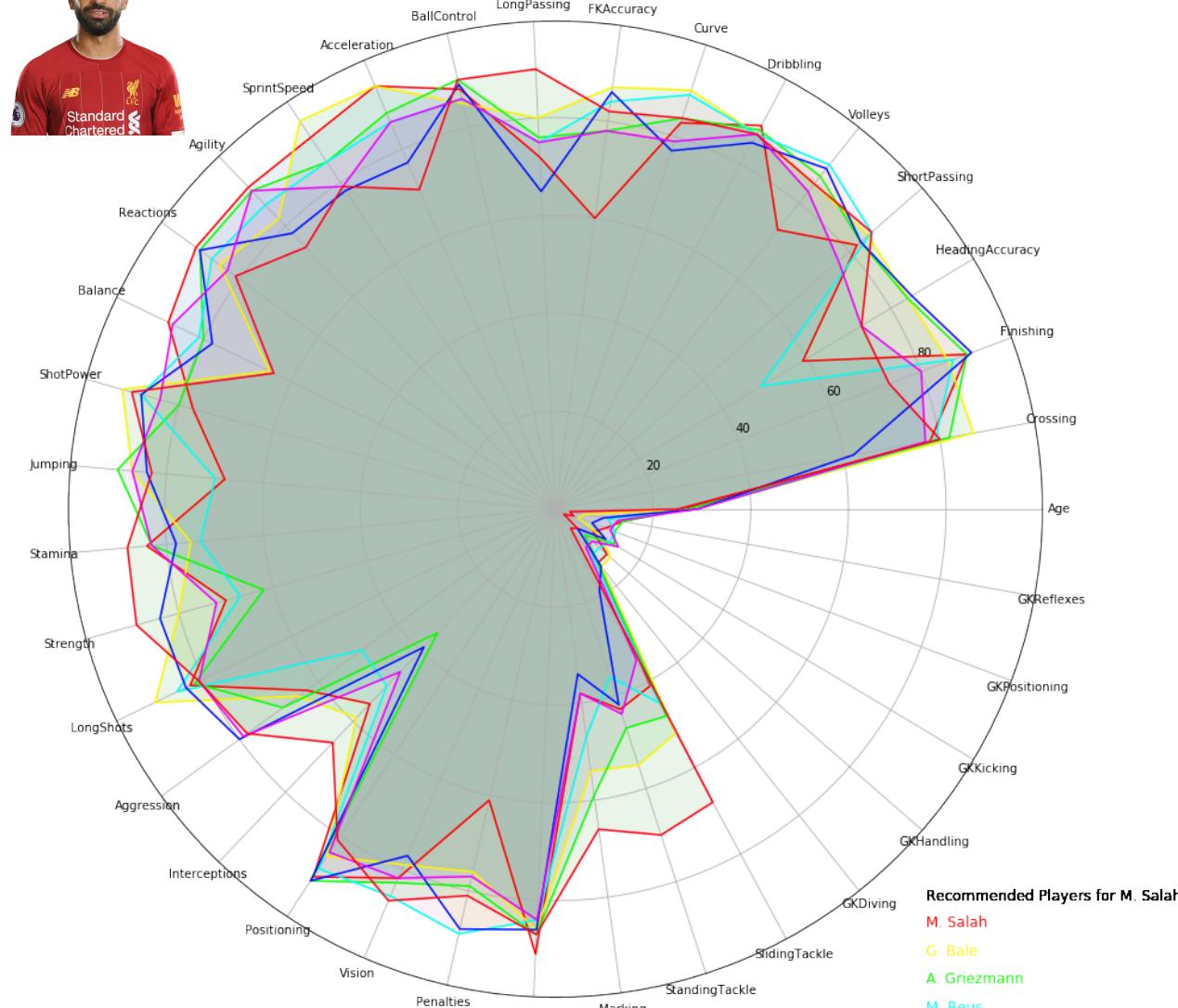
Recommendation Works for Goalkeepers...



For the Good of the Game



...and for Forwards





# Anomaly Detection

# Client Pipeline Process



01

Create Restricted  
Set of Recommended  
Players



- K-Nearest Neighbors

02

Isolate Outlier  
Players

Anomaly Detection:  

- SVM-One Class
- Local Outlier factor
- Isolation Forest
- DBSCAN

03

Predict Bid Price for  
Isolate Outlier  
Players

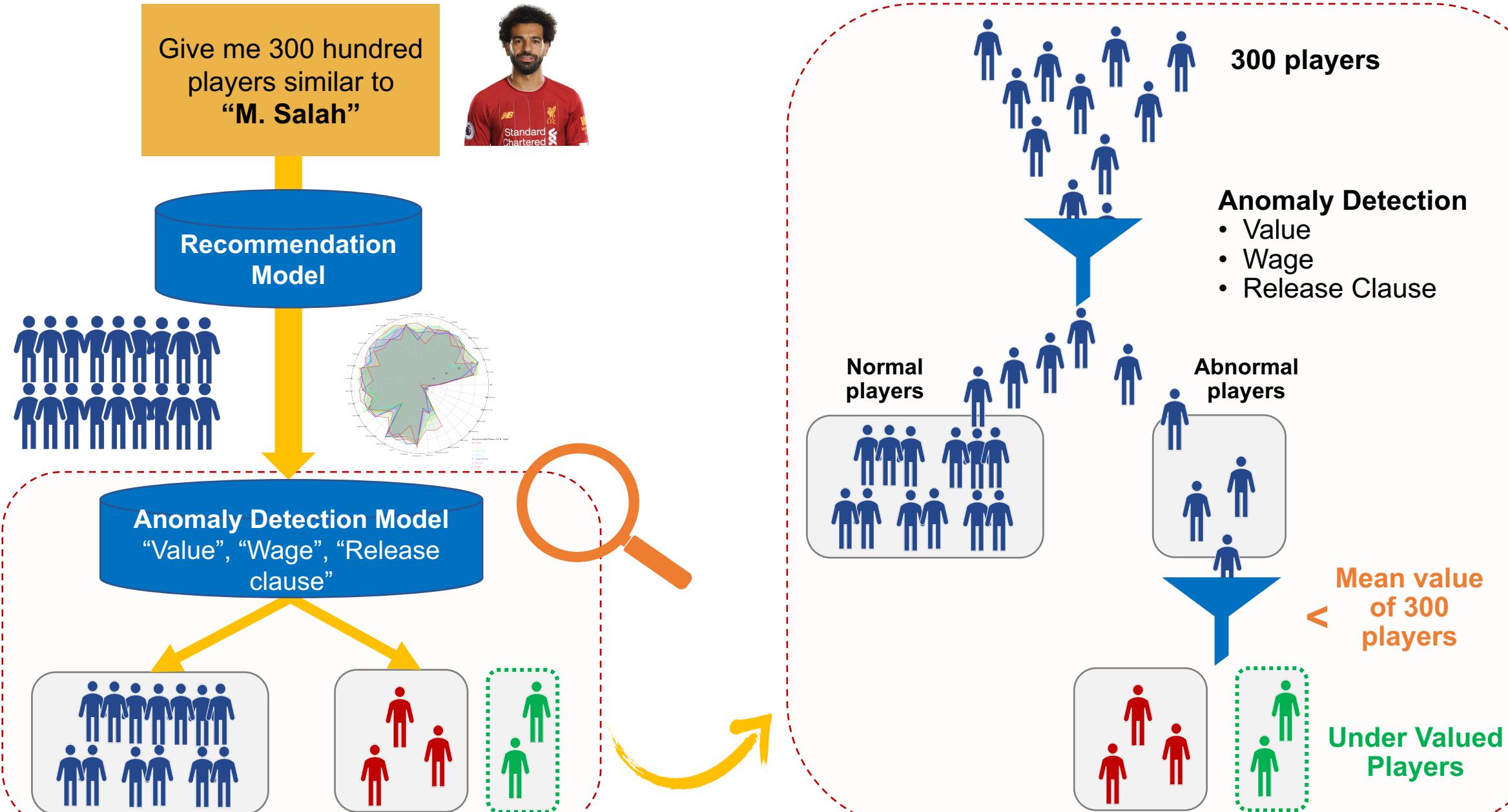


- Linear Regression
- Decision Tree
- Random Forest
- XGBoost
- SVR





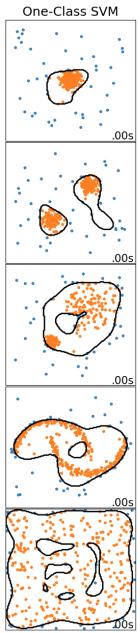
# Anomaly Detection Process





# Anomaly Detection Methods

## Parametric: OneClass SVM



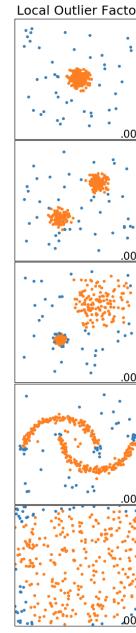
**Suitable with novelty detection**

### Pros:

- Scales well to high dimensional data
- Difficult to understand and interpret the final model
- Difficult to tune hyperparameters gamma & nu
- One-class SVM approach does not control over the false alarm rate (class imbalance)

### Cons:

## Density Based: LOF

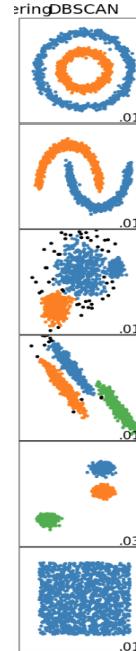


**Interpret k ratio depends on business knowledge and experience**

### How they work ?

1. Provide normal training data
2. Algorithm creates a representational model of this data (boundary).
3. If newly encountered data is too different it is labeled as out-of-class.

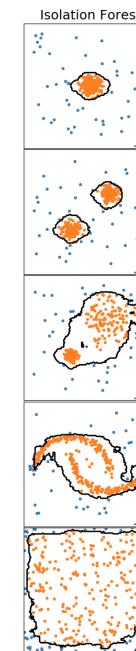
## Density Based: DBSCAN



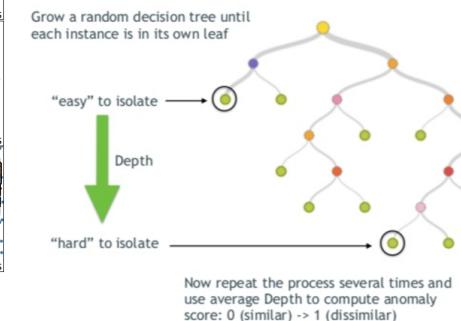
**Depends on how we choose eps and min\_samples**

1. Define eps and min.samples
2. Core point if a minimum number of points are within a given distance
3. A point is reachable if there is a path consisting of core points from start to end
4. Any point that is not reachable is considered an outlier

## Ensemble: Isolation Forest



1. Build forest of decision trees
2. For each tree, select a random feature and a random split point.
3. Outliers should be identified closer to the roots of the trees on average >> score
4. S = 1: anomaly, S<0.5 normal
5. If all scores close to 0.5, then no clear anomalies.



### Pros:

- Can handle high dimensional data
- Low linear time-complexity and a small memory-requirement
- Does not employ distance/density and only considers isolation

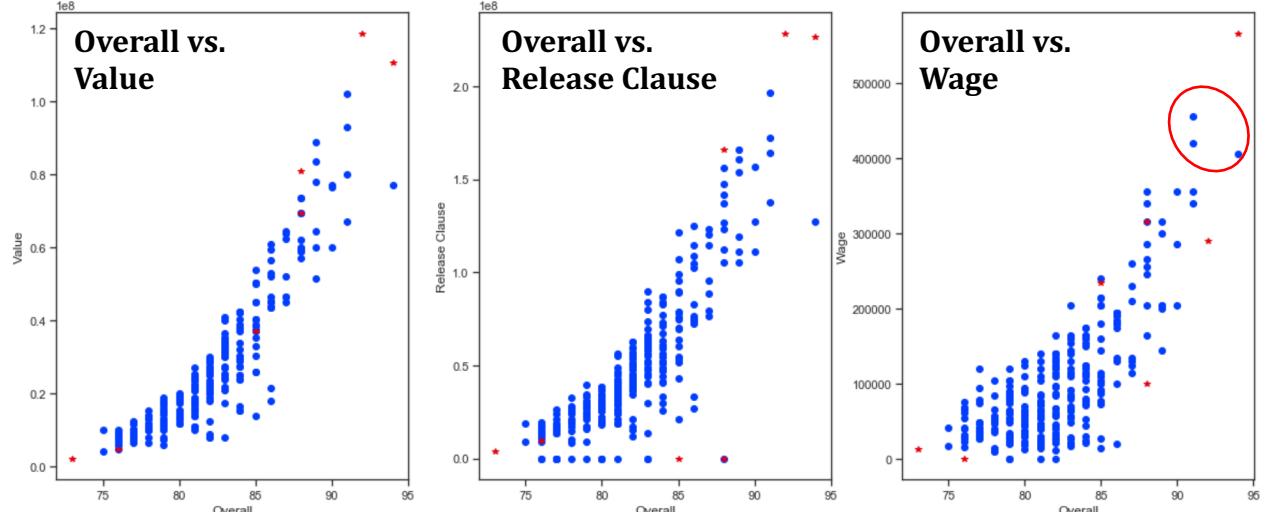
### Cons:

- Not ideal if we have a model or good understanding of outliers (I.e. if there is training data)

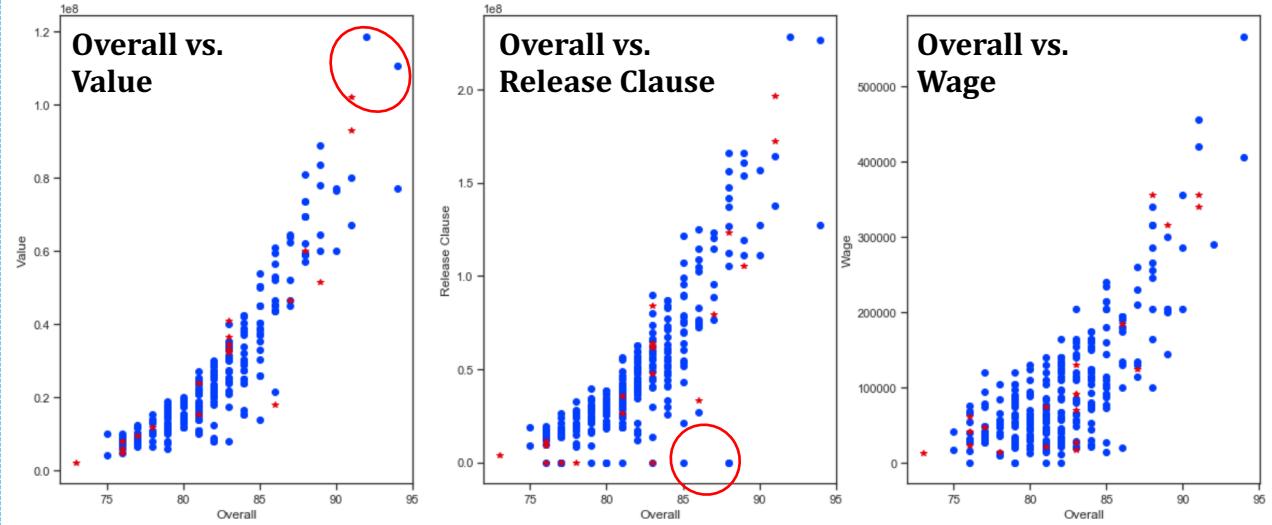
# Anomaly Detection - Compare across methods



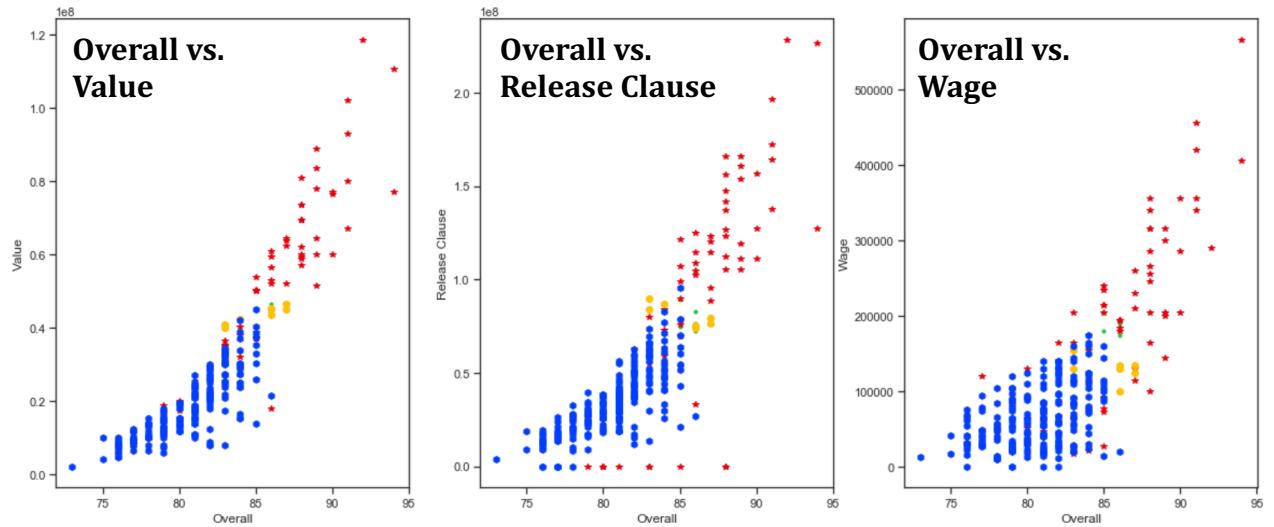
## SVM ONE CLASS



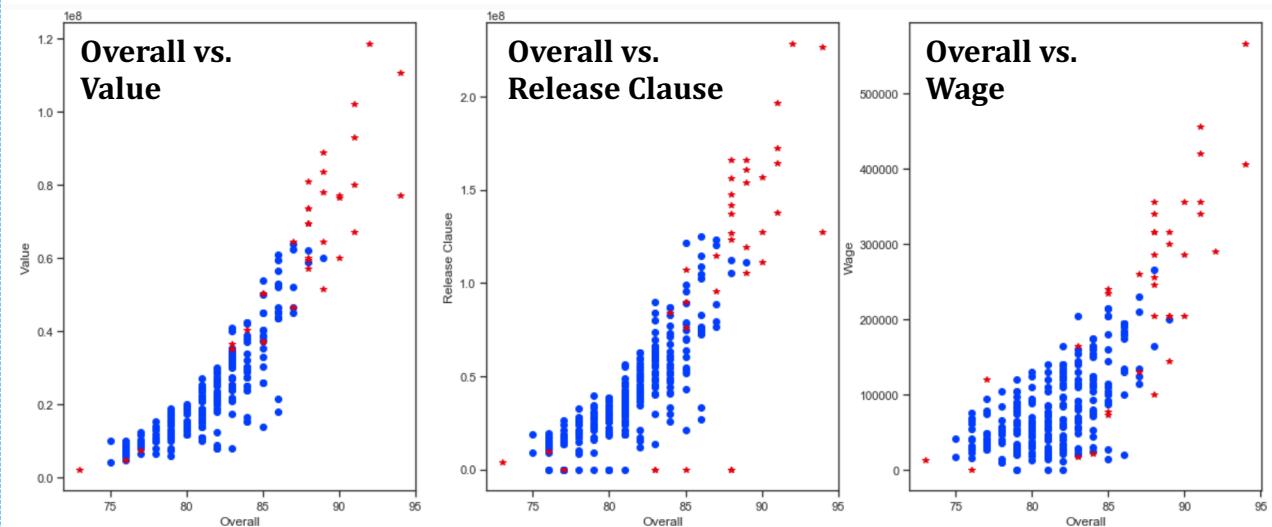
## LOCAL OUTLIER FACTOR



## DBSCAN



## ISOLATION FOREST





# Bid Prediction



# Client Pipeline Process

01

Create Restricted Set of Recommended Players



- K-Nearest Neighbors

02

Isolate Outlier Players

Anomaly Detection:

- SVM-One Class
- Local Outlier factor
- Isolation Forest
- DBSCAN

03

Predict Bid Price for Player



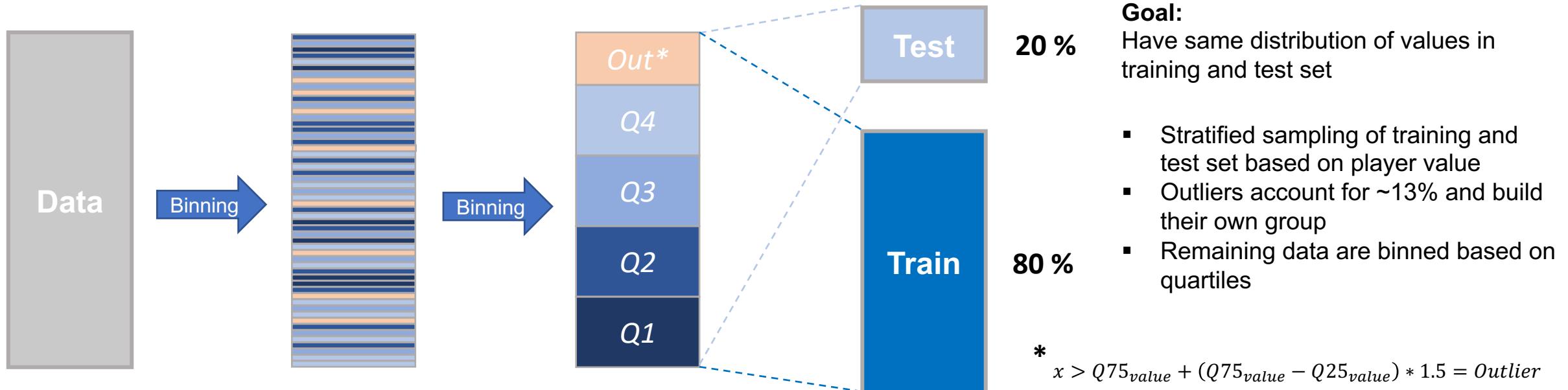
- Linear Regression
- Decision Tree
- Random Forest
- XGBoost
- SVR



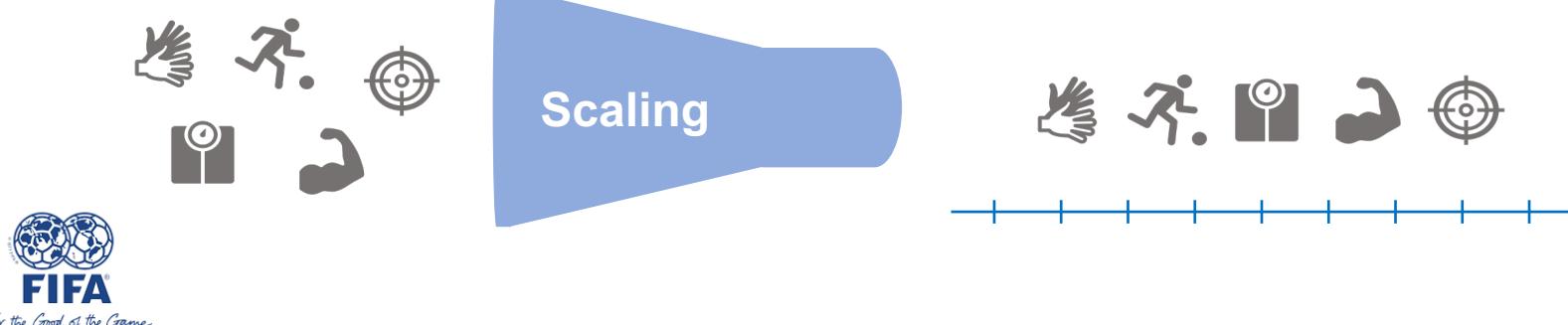


# Bid Prediction – Data Pre-Processing

## 1 Stratified train and test sampling



## 2 Scaling



**Goal:**  
Normalizing range of independent features

- Scaling all numerical features that are not categorical
- After scaling, each feature has *mean* = 0 and *standard deviation* = 1



# Bid Prediction – Model Consideration

Linear

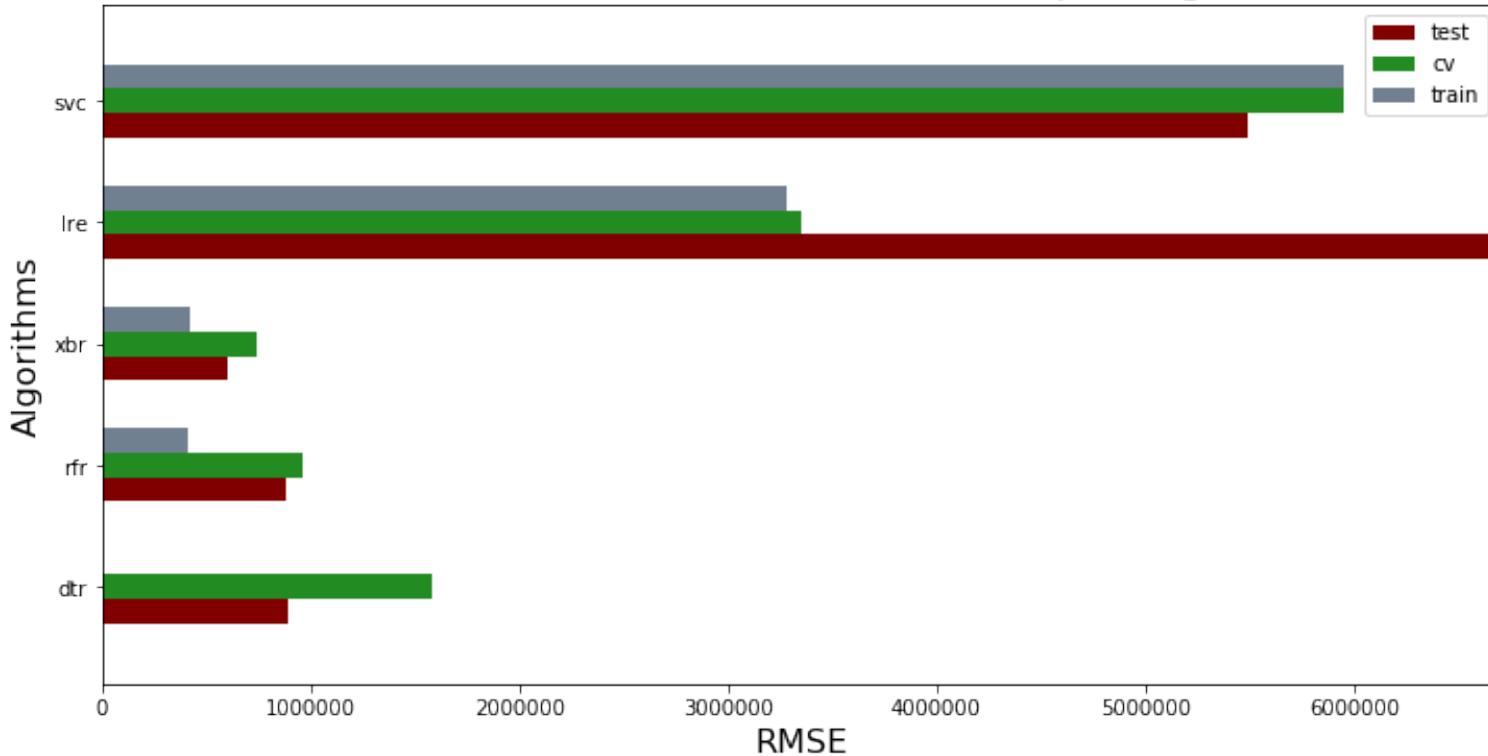
Non-linear

Model Type	Strengths	Weaknesses
<b>Linear Regression</b>	<ul style="list-style-type: none"> <li>Simple</li> <li>Easy to understand relationships (Interpretable coefficients)</li> <li>Inference focused</li> </ul>	<ul style="list-style-type: none"> <li>Poor performance with non-linear data relationships between dependent and independent variables</li> <li>Not naturally flexible enough to capture more complex patterns, and adding the right interaction terms &amp; polynomials difficult.</li> </ul>
<b>Support Vector Regression</b>	<ul style="list-style-type: none"> <li>Can handle non-linear relationships without changing the explanatory variables through "kernel trick"</li> <li>Effective in the higher dimension</li> </ul>	<ul style="list-style-type: none"> <li>Difficult to tune hyperparameters</li> <li>Difficulty specifying the 'right' kernel function</li> </ul>
<b>Decision Tree</b>	<ul style="list-style-type: none"> <li>Capable of understanding non-linear relationships</li> <li>Handles collinearity efficiently.</li> <li>No assumptions on distribution of data</li> </ul>	<ul style="list-style-type: none"> <li>Greedy algorithm</li> <li>Prone to overfit when complexity not controlled</li> </ul>
<b>Random Forest</b>	<ul style="list-style-type: none"> <li>Same as DT +</li> <li>More resistant to over-fitting</li> <li>RF is much easier to tune than GBM.</li> <li>Biased in favor of categorical variables with attributes with more levels</li> </ul>	<ul style="list-style-type: none"> <li>Computationally expensive</li> <li>Not a well descriptive model over the prediction.</li> </ul>
<b>Gradient Boosting</b>	<ul style="list-style-type: none"> <li>Same as DT +</li> <li>Learns sequentially</li> <li>Deals with unbalanced datasets better than RF</li> </ul>	<ul style="list-style-type: none"> <li>Prone to overfit to noisy data</li> <li>Slower than RF because trees are built sequentially</li> <li>Harder to tune than RF</li> </ul>

# Bid Prediction – Baseline Model Results



Test, Cross Validation and Train Error per Algorithm



- *Using RMSE as evaluation metric\**
- Support Vector Regression most stable model
- Linear Regression with extremely high test error
- Decision Tree with virtually no training value
- Random Forest shows some variance, but has a relatively low bias overall
- *XGBoost with the best result, weighing variance and bias*

\* 
$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y})^2}$$

# Bid Prediction – Feature Selection



1

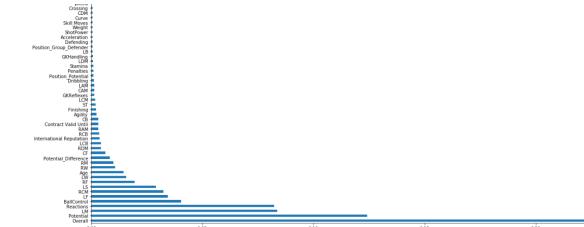
		Specs	Score	Expl_percent
5	International Reputation	10926.667128	1.020489e+01	
1	Overall	9247.932429	8.637050e+00	
75	Club_Reputation	8635.202466	8.064794e+00	
2	Potential	7144.411856	6.672479e+00	
54	Reactions	5942.582653	5.550038e+00	
66	Composure	3632.566134	3.392613e+00	
8	Real Face	3565.014584	3.329523e+00	
3	Special	2416.453120	2.256832e+00	
64	Vision	2126.810904	1.986322e+00	
81	Mentality	1937.306594	1.809336e+00	
44	ShortPassing	1773.834349	1.656662e+00	
7	CHILL	1620.277212	1.520210e+00	

## F-Value:

- Start with constant model  $M_0$
  - Try all models  $M_1$  consisting of just one feature and pick the best according to the F statistic
  - Try all models  $M_2$  consisting of  $M_1$  plus one other feature and pick the best

$$\frac{F_{score}}{\sum_{n=1}^N F_{score,n}} > 0.01$$

2

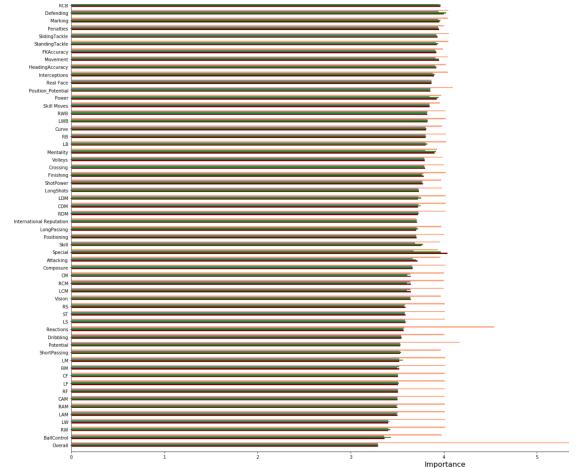


## *Top ten features*

## Tree Regressor

- Based on Extra Tree Regressor (Decision Tree with random splits)
  - Total reduction of the criterion brought by that feature (Gini importance)
  - Rank by total reduction

3



## RMSE-based:

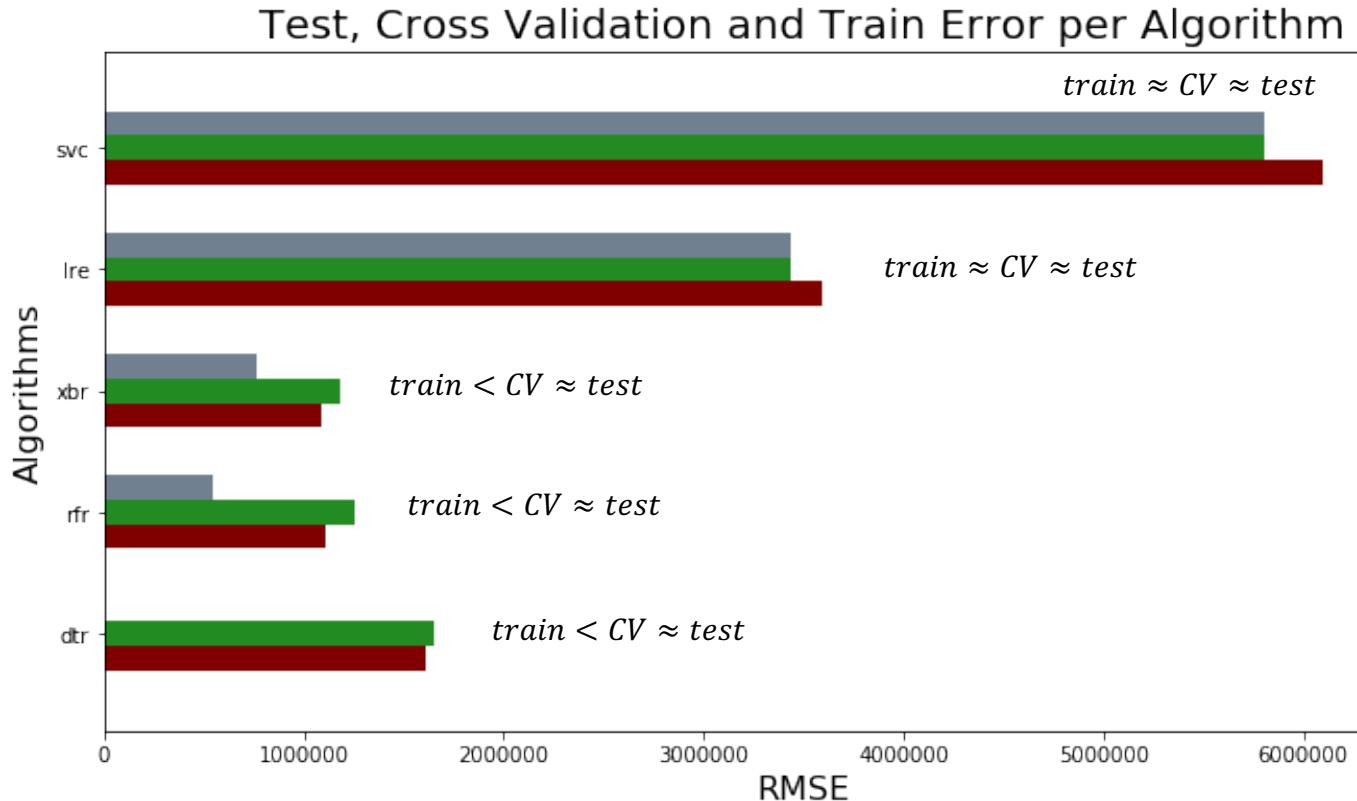
- Try all models  $M_1$  consisting of just one feature and calculate the RMSE for each of the baseline models
  - Rank by lowest RMSE

## *Top ten features*

The FIFA logo consists of two interlocking soccer balls forming a stylized 'F' shape, with the word 'FIFA' in bold capital letters below it, and the tagline 'the Good of the Game' in script below that.

## Final feature selection

# Bid Prediction – Prediction on Reduced Features



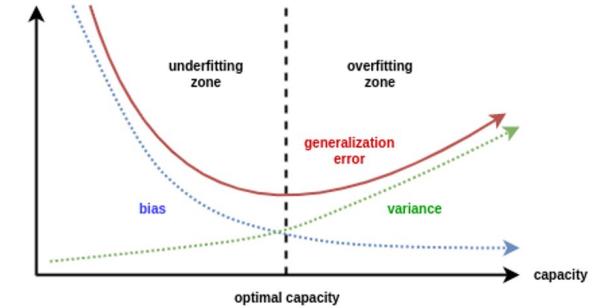
- Errors became more stable for most of the models, as compared to baseline model
- Especially Linear Regression improved significantly
- Bias similar to baseline models, therefore, we did not lose much information by reducing number of features
- XGBoost, Random Forest and Decision Tree show signs of overfitting
- Parameter tuning needed



# Bid Prediction – Parameter Tuning

## 1 Setting the goal

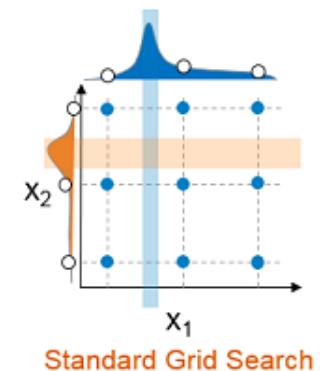
- *Problem:* Setting the optimal parameters for each model to find the sweet spot between variance and bias
- Decrease complexity for XGBoost, Random Forest and Decision Tree
- If possible, decrease bias without significantly increasing variance for all models



## 2 GridSearch

- GridSearch is an exhaustive method to find optimal hyperparameters

Model	# of parameters	# of fits
Decision Tree	4	8,000
Random Forest	4	243
XGBoost	5	324
Support Vector Reg.	2	60



## 3 Manual adjustments

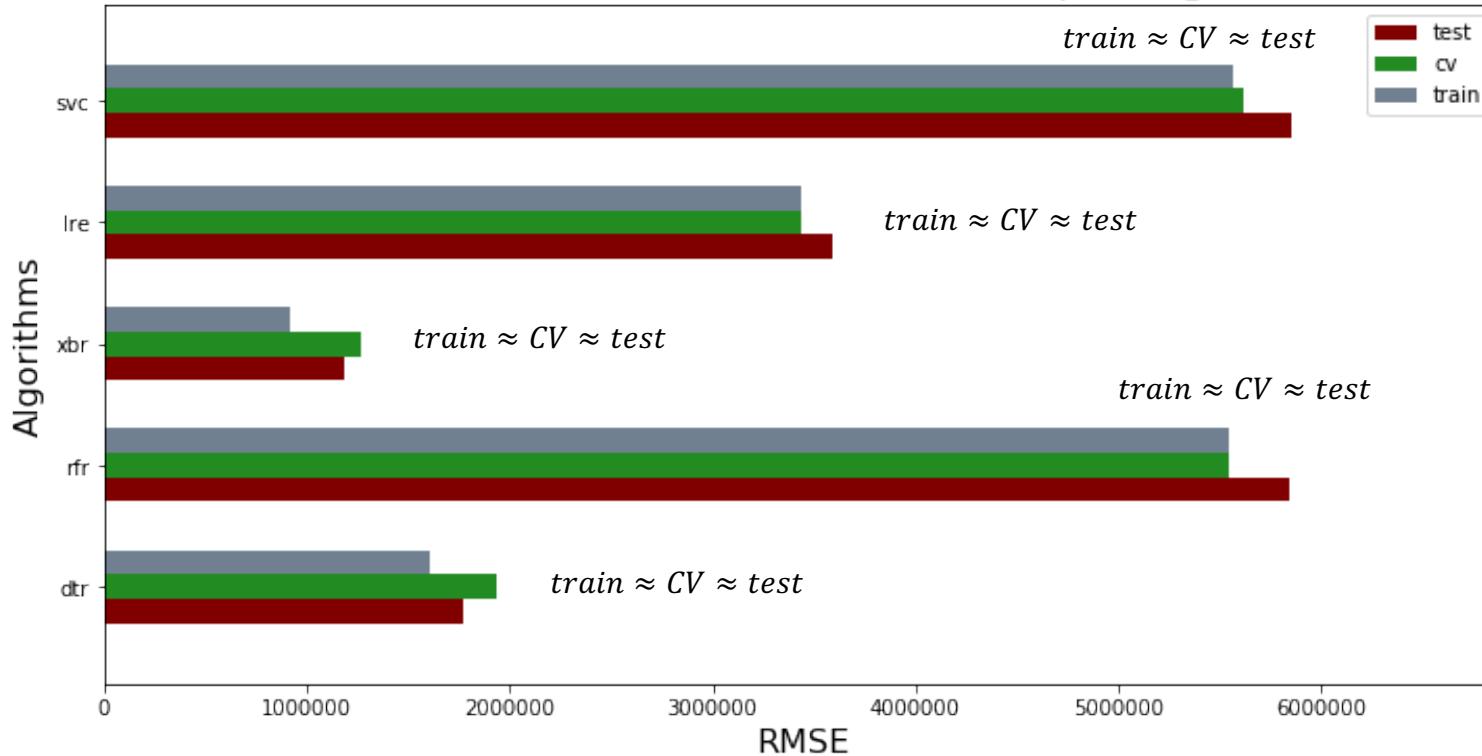
- GridSearch is optimizing MSE, but not considering variance-bias tradeoff
- To balance variance and bias, manually adjustment is needed  
(Trial and Error process)



# Bid Prediction – Final Evaluation



Test, Cross Validation and Train Error per Algorithm



- In terms of variance, all models are more or less stable
- XGBoost and Decision Tree show somewhat more variance than other models
- Lowest RMSE by far for XGBoost and maybe Decision Tree
- Even though XGBoost show a little more variance, we accept this in turn for a lower bias



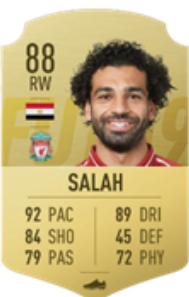
# Results



Please choose between the 2 options below:

## Option 1: PLAYER

Select a player



## Option 2: POSITION AND SCORE

Select a position

CM  
CDM  
RW  
...

Min Overall score

76

Max Overall score

88

Here is your first bid players suggestion



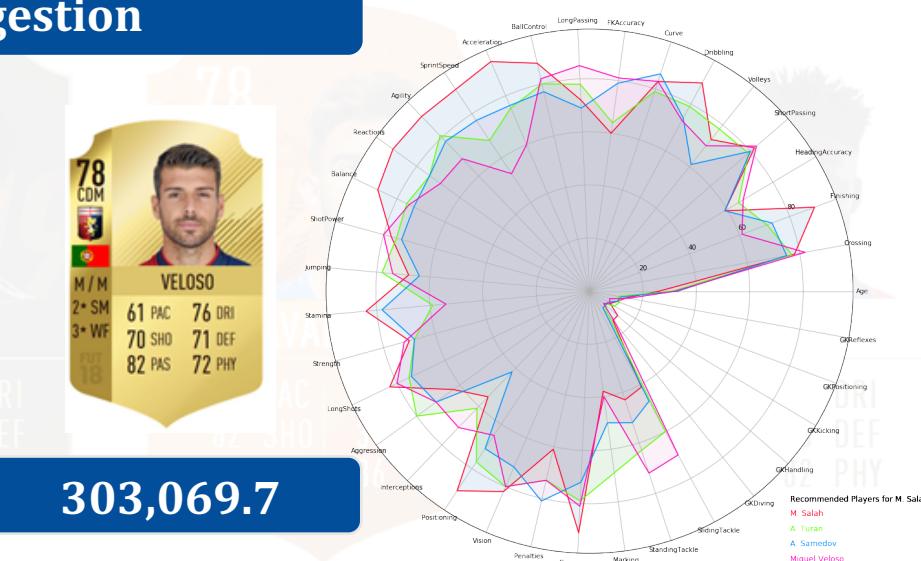
Suggested value:

3,612,241.2

894,736.06

303,069.7

Recommended Players for M. Salah  
M. Salah  
A. Turan  
A. Samedov  
Miguel Veloso

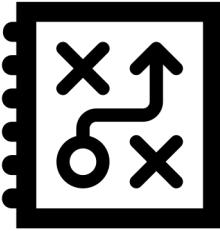
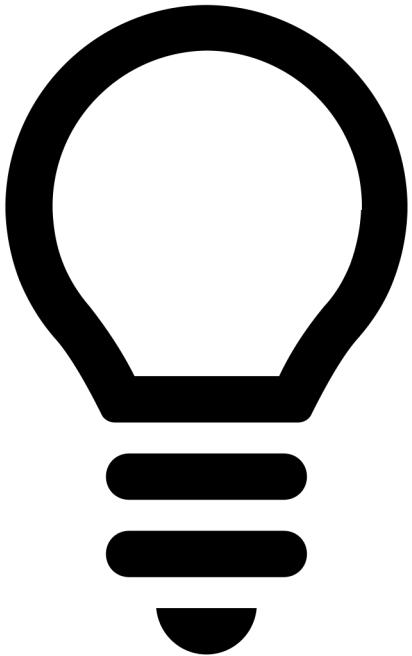




## Next Steps



# Next Steps



- Expand dataset to include historical data
- Incorporate intra-match statistics, including geospatial data as well as personal health data such as heart-rate monitoring
- Develop analytics to assess coaching style and style of play



# Thank you!