

# FIFA 21

DATA ANALYSIS FINDINGS

Group 2 : Carlos Ruiz / Maria Bollain / Ying Wang

The FIFA logo is displayed in white, bold, sans-serif capital letters within a dark blue square. The background of the slide features a geometric pattern of overlapping triangles in various shades of blue in the top right corner.

**FIFA**

# Content

1. Executive summary
2. Problem definition
3. Data reading
4. Data cleaning
5. Exploratory data analysis
6. Data processing
7. Data modeling
8. Machine learning
9. Conclusions
10. Methodology

# FIFA



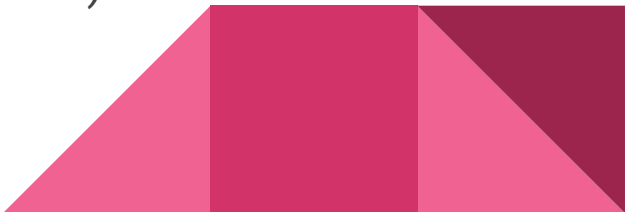
# 1. Executive summary

In this project, we perform an analysis of the FIFA 21 dataset that we have been provided with; more specifically, we work with different scenarios where we use both with “release clause” and “wage” to compare the market value across players in different selections.

Here we provide a summary of the key findings that we provide further information over the presentation.



# 1. Executive summary

- The players with higher value are not necessarily the ones with higher wages (that cost more money to the club).
  - If we keep the “release clause” and the outliers, we have a higher  $r^2$  coefficient, compared where we get rid of the “release clause” and remove the outliers.
  - Overall scores and potential scores resemble more a “normal distribution”, whereas “wage” behaves more like a left-skewed and “total stats” like a right-skewed distribution
  - From the multicollinearity graph, we find that market value of the player is highly correlated to wage (0.82) and release clause (0.98).
- 

## 2. Problem definition

With this dataset, we have focused on providing the answer to these two questions:

- Rank players by market value and compare with wages.
- Predict the market value using linear regression.



### 3. Data reading

- We get a total of 107 columns and we spent some time doing research on the abbreviations:
  - OVA = overall scores
  - BOV = best overall
  - POT = potential scores
  - GK = goalkeeping
  - W/F = weak foot
  - SM = skilled moves
  - IR = international reputation
  - CF = center forward



## 4. Data cleaning

"attacking" = sum ("crossing", "finishing", "heading\_accuracy", "short\_passing", "volleys")

"skill" = sum ("dribbling", "curve", "fk\_accuracy", "long\_passing", "ball\_control")

"movement" = sum ("acceleration", "sprint\_speed", "agility", "reactions", "balance")

"power" = sum ("shot\_power", "jumping", "stamina", "strength", "long\_shots")


"mentality" = sum ("aggression", "interceptions", "positioning", "vision", "penalties", "composure")

"defending" = sum ("marking", "standing\_tackle", "sliding\_tackle")

"goalkeeping" = sum ("gk\_diving", "gk\_handling", "gk\_kicking", "gk\_positioning", "gk\_reflexes")

"total\_stats" = sum ("attacking", "skill", "movement", "power", "mentality", "defending", "goalkeeping")

"base\_stats" = sum ("pac", "sho", "pas", "dri", "def", "phy")



## 4. Data cleaning

```
drop_columns = ["position","player_photo","club_logo","flag_photo","team_&_contract","joined","contract",  
  
"crossing","finishing","heading_accuracy","short_passing","volleys",  
  
"dribbling","curve","fk_accuracy","long_passing","ball_control",  
  
"acceleration","sprint_speed","agility","reactions","balance",  
  
"shot_power","jumping","stamina","strength","long_shots",  
  
"aggression","interceptions","positioning","vision","penalties","composure",  
  
"marking","standing_tackle","sliding_tackle",  
  
"gk_diving","gk_handling","gk_kicking","gk_positioning","gk_reflexes",  
  
"base_stats","w/f","sm","pac","sho","pas","dri","def","phy","hits",  
  
"ls","st","rs","lw","lf","cf","rf","rw","lam","cam","ram","lm","lcm",  
  
"cm","rcm","rm","lwb","ldm","cdm","rdm","rwb","lb","lcb","cb","rcb","rb","gk","gender"]
```



## 4. Data cleaning

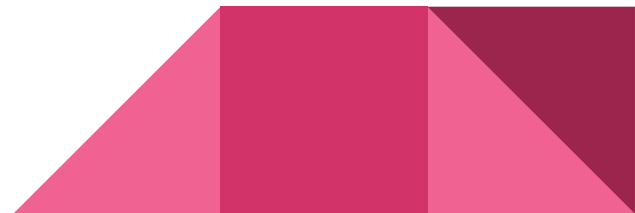
### NaN values:

club 23 → fill with “unknown”

loan\_date\_end 16215 → drop column

attacking\_work\_rate 89 → drop rows

defensive\_work\_rate 89 → drop rows



## 4. Data cleaning

**Columns that are objects and should be numerical:**

height, weight, value, wage, release\_clause, weak\_foot, skilled\_moves,  
international\_reputation, hits

```
def clean_value(x):  
    x = x.replace("€", "")  
    if "M" in x:  
        x = float(x.replace("M", "")) * 1000000  
    elif "K" in x:  
        x = float(x.replace("K", "")) * 1000  
    else:  
        x = float(x)  
    x = int(x)  
    return x  
  
data["value"] = list(map(clean_value, data["value"]))
```

## 4. Data cleaning

Better position column: a lot of abbreviations.

```
defense_positions = ["CB", "RB", "LB", "LWB", "RWB"]
midfield_positions = ["CDM", "CM", "RM", "LM", "CAM"]
attack_positions = ["LW", "RW", "CF", "ST"]

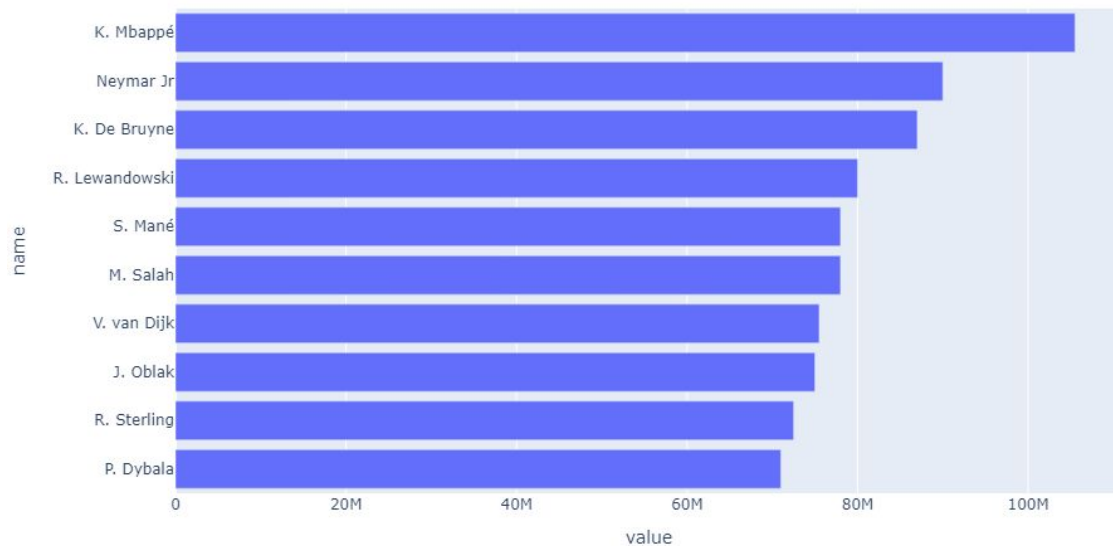
def clean_position(x):
    if x in defense_positions:
        x = "Defense"
    elif x in midfield_positions:
        x = "Midfield"
    elif x in attack_positions:
        x = "Attack"
    elif x == "GK":
        x = "Goalkeeper"
    return x

data["better_position"] = list(map(clean_position, data["better_position"]))
data["better_position"].value_counts()
```

```
Midfield      6672
Defense       5550
Attack        3247
Goalkeeper    1567
Name: better_position, dtype: int64
```

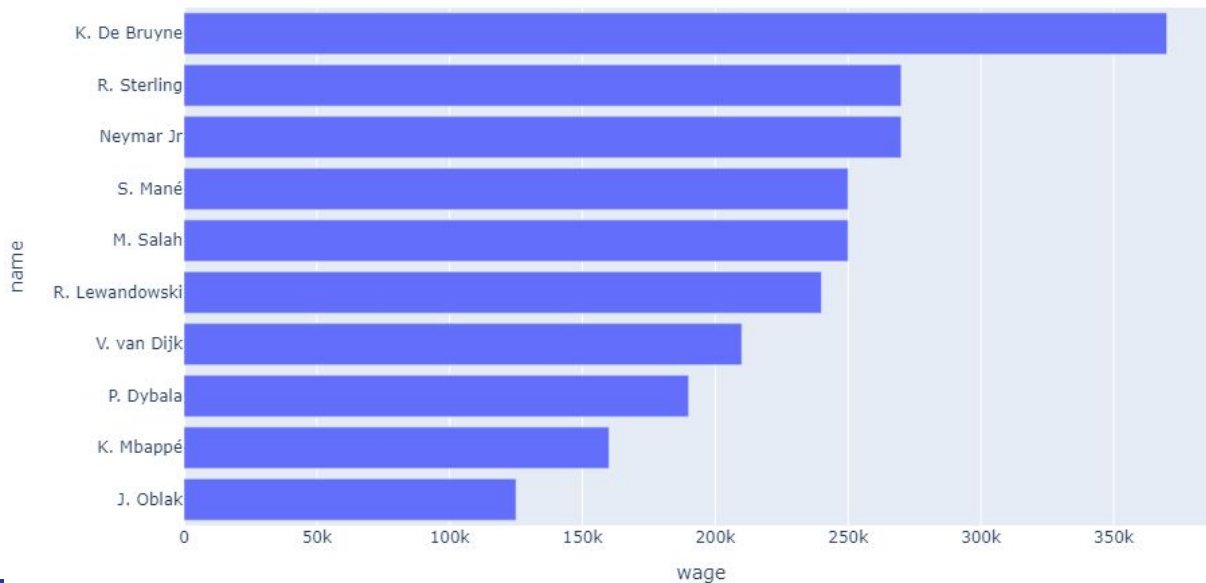
## 5. EDA: Key findings - Golden players

Top 10 players ordered by value



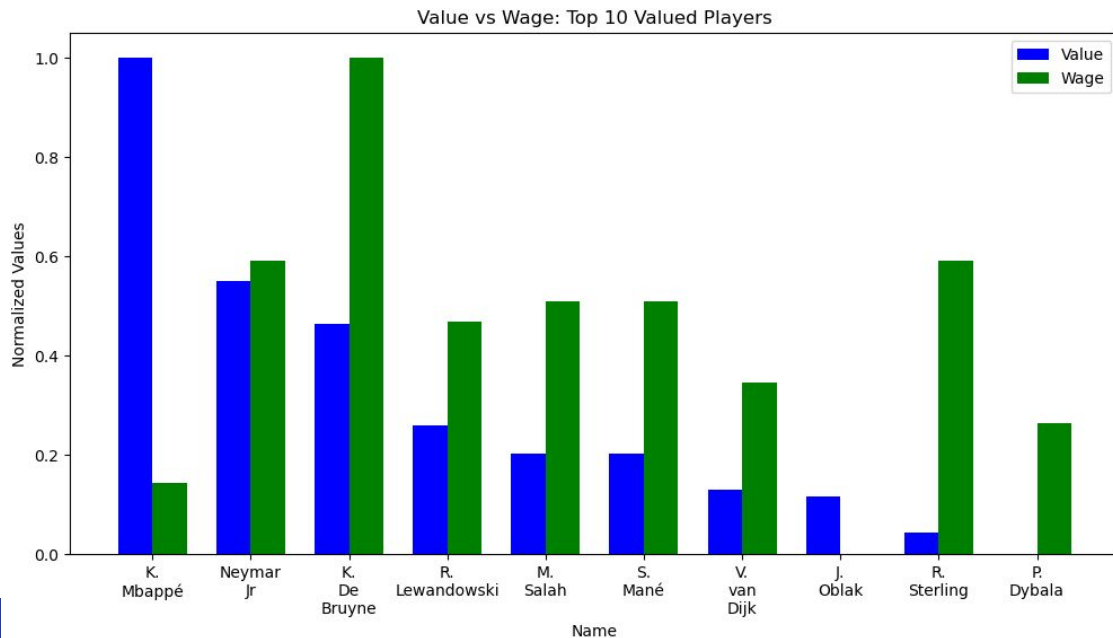
## 5. EDA: Key findings - Golden players

Top 10 players ordered by wages



## 5. EDA: Key findings - Golden players

The players with higher value are not necessarily the ones with higher wages (that cost more money to the club).



## 5. EDA: Key findings - value distribution

Mean: 1,450,565

Min: 0

25%: 350,000

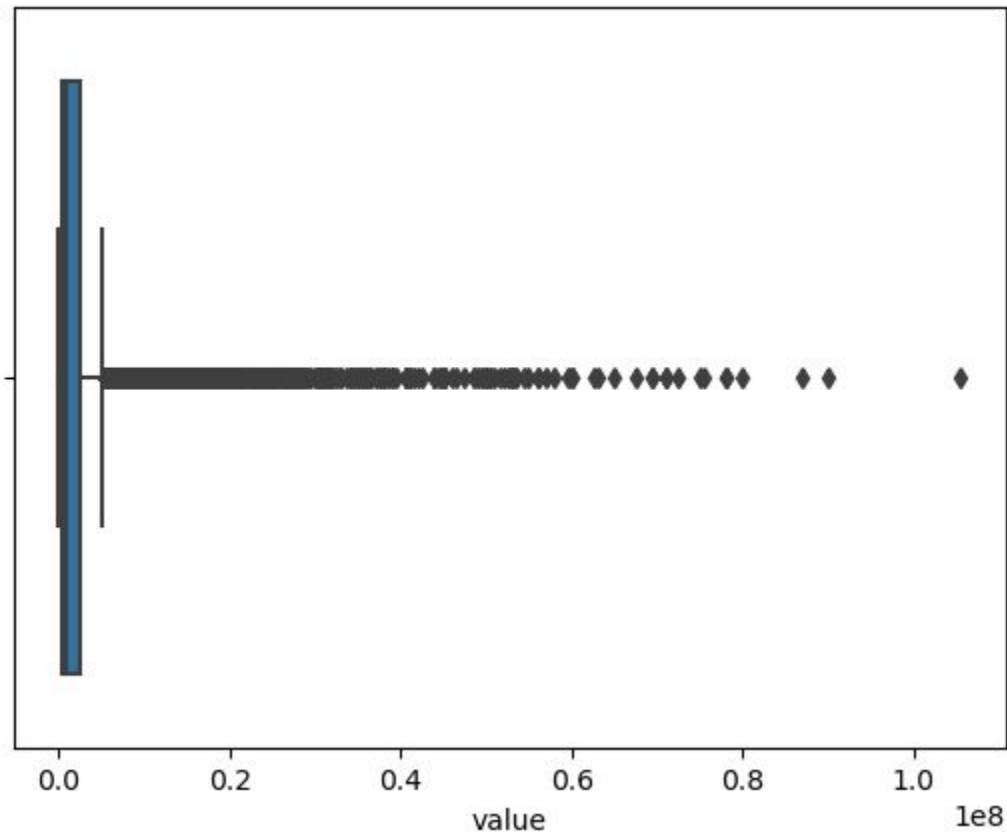
50%: 725,000

75%: 1,500,000

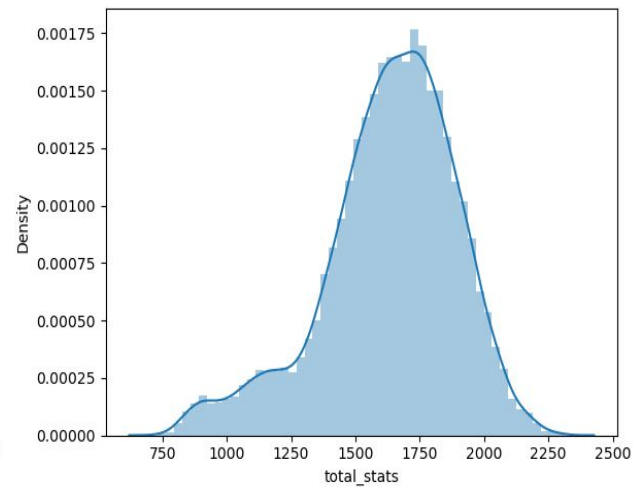
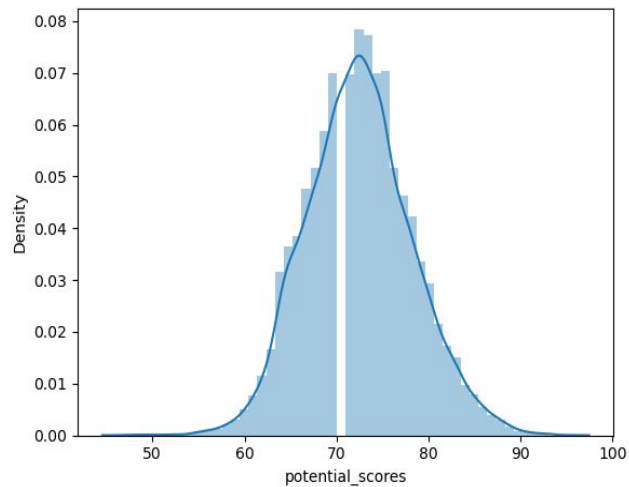
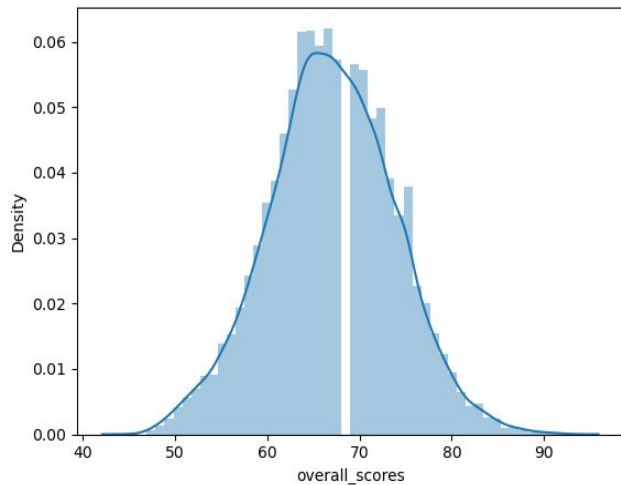
Max: 27,000,000

Huge number of outliers!!!

Also a lot of outliers in wage and release clause



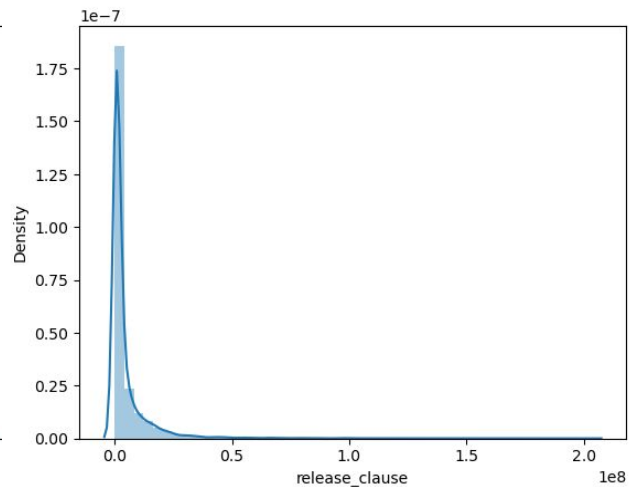
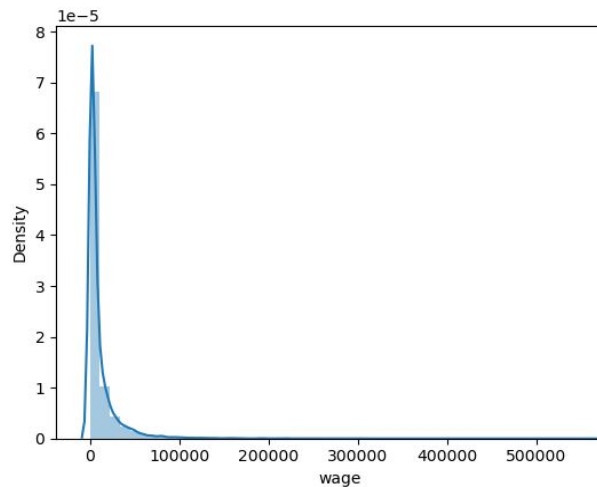
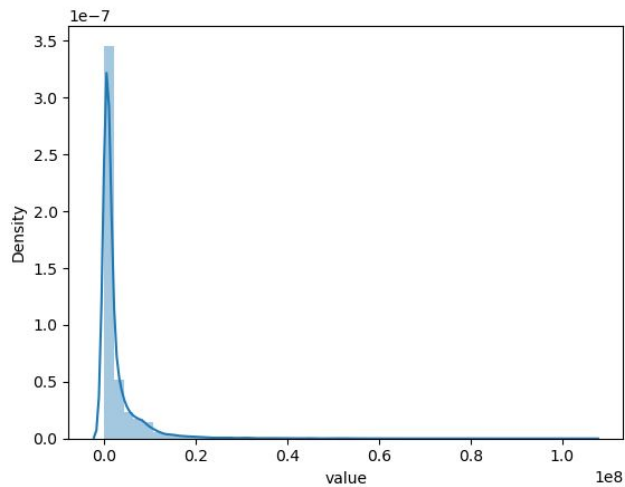
## 5. EDA: Density graphs



+ normal distribution



## 5. EDA: Density graphs



Highly left skewed

## 5. EDA: Correlation between variables: multicollinearity

We find that the features that are most correlated with the target are "release\_clause" (0.98) and "wage" (0.82).

Total stats are the sum of "Attacking", "Skill", "Movement", "Power", "Mentality", "Defending" and "Goalkeeping", so we decide to get rid of them as they are variables that show high multicollinearity themselves.

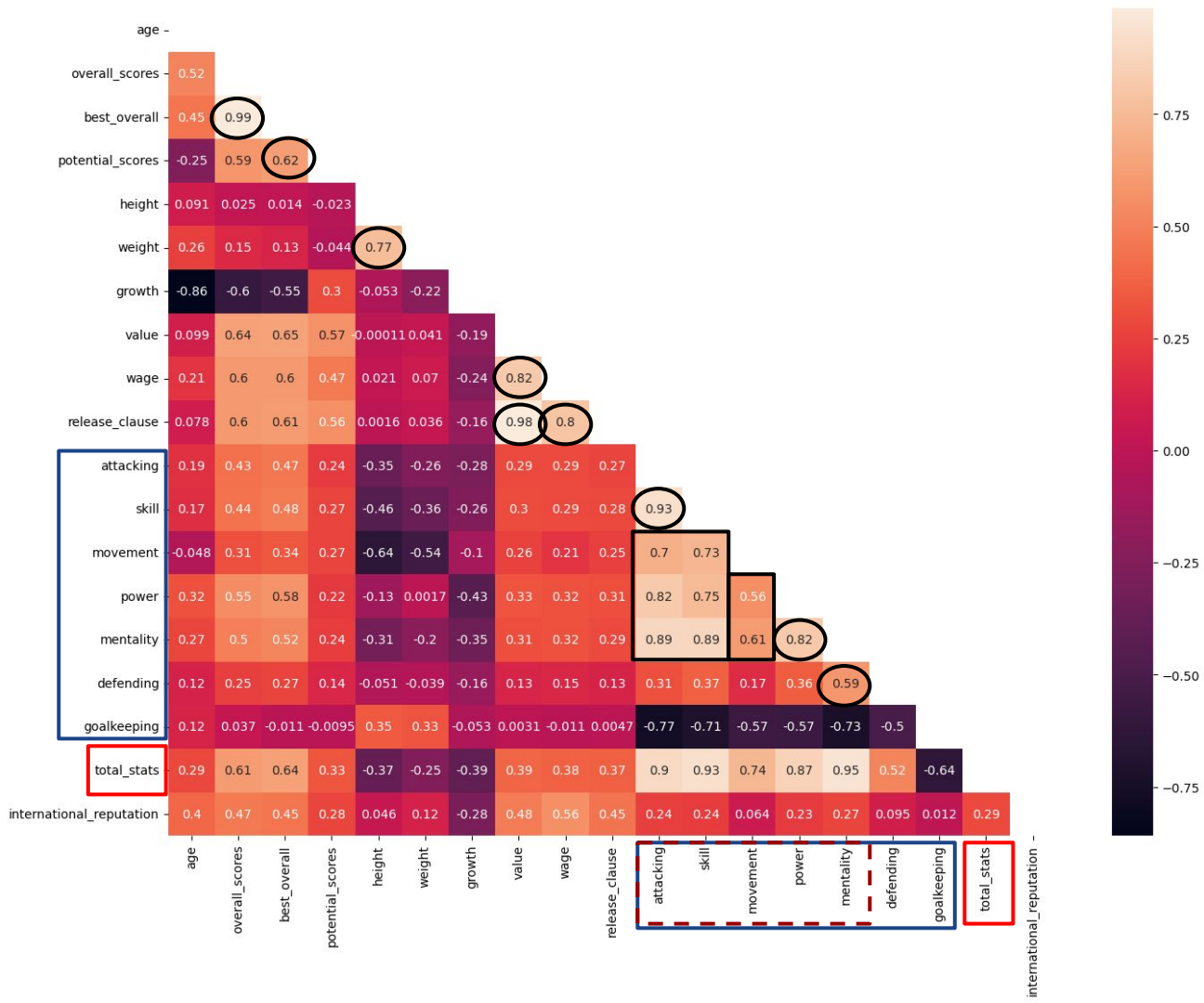
"Best overall" and "overall" scores also show a really high correlation, 0.99.



## 6. Data processing

- Here is the initial multicollinearity analysis that we get...





Drop columns:

- "Best\_overall"
- "Growth"
- "Attacking"
- "Skill"
- "Movement"
- "Power"
- "Mentality"
- "Defending"
- "Goalkeeping"

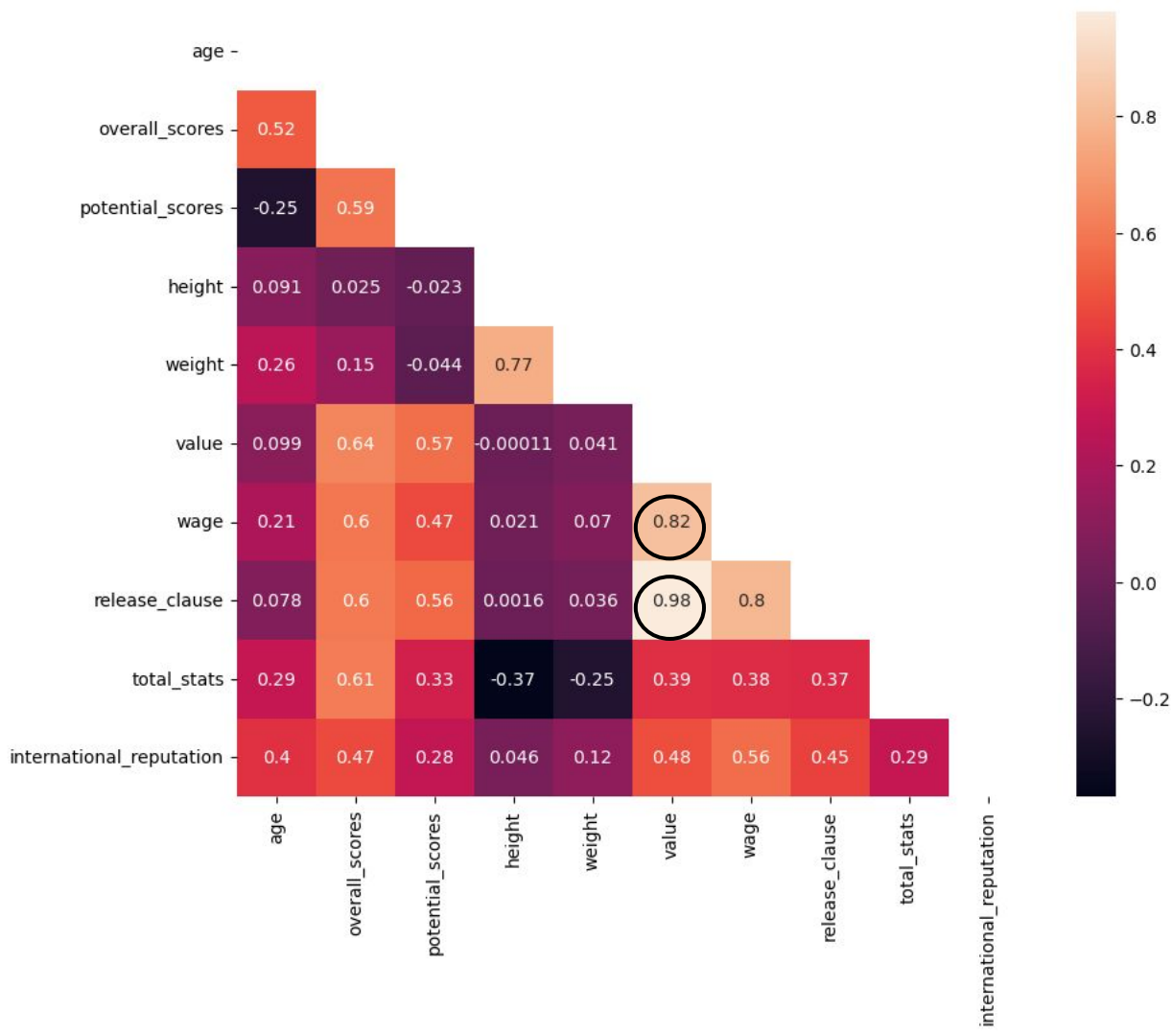
## 6. Data processing

- And this is the new one that we get...



# Multicollinearity

From the graph, we find that market value of the player is highly correlated to wage (0.82) and release clause (0.98).



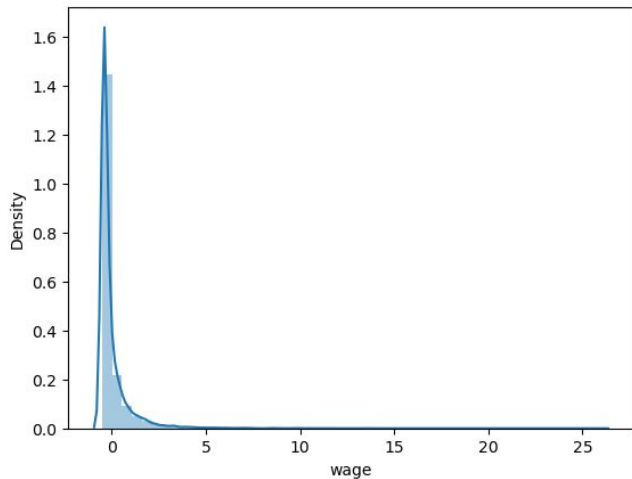
## 7. Modeling - Linear Regression

- X-y split
- Separate numerical and categorical data and more processing
- Concat into final dataset
- Train-test split
- Apply linear regression
- Model validation
- Try different models
- Final decision with model



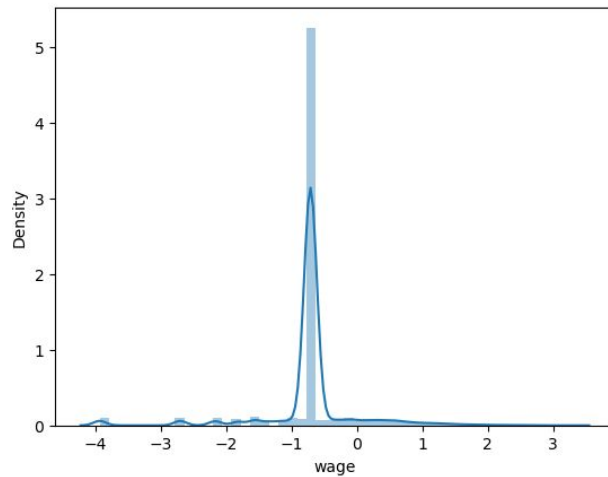
# Machine Learning: transformations - numericals

Based on the distribution plots, we used log transformation to `wage` and `release\_clause`



Before log transformation

VS



After log transformation



# Machine Learning: Standardization - numericals

```
# Normalize (numerical)
X_scaled = StandardScaler().fit_transform(X_num)

def normalize(X):
    X_mean = X.mean(axis=0)
    X_std = X.std(axis=0)
    X_std[X_std==0] = 1.0
    X = (X-X_mean)/X_std
    return X

X_num = normalize(X_num)
X_num.head()
```

	age	overall_scores	potential_scores	height	weight	wage	release_clause	total_stats	international_reputation
id									
2	1.575619	0.299917	-0.603693	0.241231	0.972601	-0.169349	-0.443850	1.141676	1.912181
16	2.388466	0.592291	-0.255018	-0.496382	-1.432582	-0.169349	-0.337674	1.053208	-0.326867
27	1.575619	0.592291	-0.255018	-0.865189	-0.292771	0.208245	-0.443850	0.530091	1.912181
41	2.185254	2.054161	1.488356	-1.602802	-0.988935	0.066647	0.251122	1.468624	6.390278
61	0.356348	-0.138644	-0.429355	-1.233995	-0.165815	-0.263747	-0.443850	0.064671	1.912181

# Machine Learning: transformations - categoricals

COLUMN	UNIQUE VALUES	ACTION
name	16092	Drop column
nationality	167	Keep the top nationalities, group the rest into other
club	907	Drop column
position	4	Keep
foot	2	Keep
attacking_work_rate	3	Drop column
defensive_work_rate	3	Drop column

# Machine Learning: transformations - categoricals

```
top_countries = ["England", "Germany", "Spain", "France", "Brazil", "Argentina"]
```

```
def clean_nation(x):  
    if x not in top_countries:  
        x = "Other"  
    else: x  
    return x
```

```
X_cat["nationality"] = list(map(clean_nation, X_cat["nationality"]))  
X_cat["nationality"].value_counts()
```

```
Other          10461  
England         1704  
Germany         1150  
Spain           1120  
France           983  
Brazil           843  
Argentina        775  
Name: nationality, dtype: int64
```

+ Apply get dummies

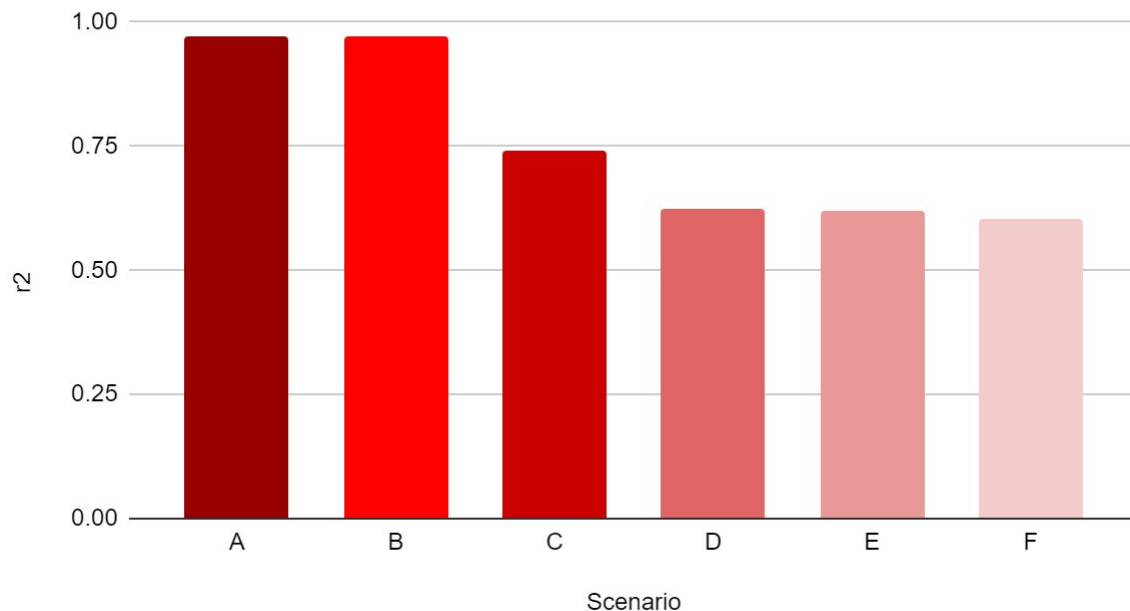
# Different models - metrics

Description	Scenario	r2	adjusted r2	mse	rmse	mae
<b>With wage and release_clause and no removing outliers</b>	<b>A</b>	<b>0.9725337218</b>	<b>0.9725030531</b>	<b>805836061008</b>	<b>897683.7199</b>	<b>447639.9012</b>
With release_clause, without wage and no removing outliers	B	0.9716063963	0.9715763625	833042963945	912711.8735	465650.1525
With wage, without release_clause and no removing outliers	C	0.7400586937	0.7397837367	7626445670117	2,761,602.01	1,444,970.17
<b>With wage and release_clause and removing outliers</b>	<b>D</b>	<b>0.623302951</b>	<b>0.6227224771</b>	<b>261289735368</b>	<b>511,165.08</b>	<b>265,334.22</b>
With release_clause, without wage and removing outliers	E	0.6184826613	0.6179869339	674,059,705,433.26	821,011.39	413,704.37
With wage, without release_clause and removing outliers	F	0.6045195049	0.6040182975	1,606,708,888,227.77	1,267,560.21	829,364.98

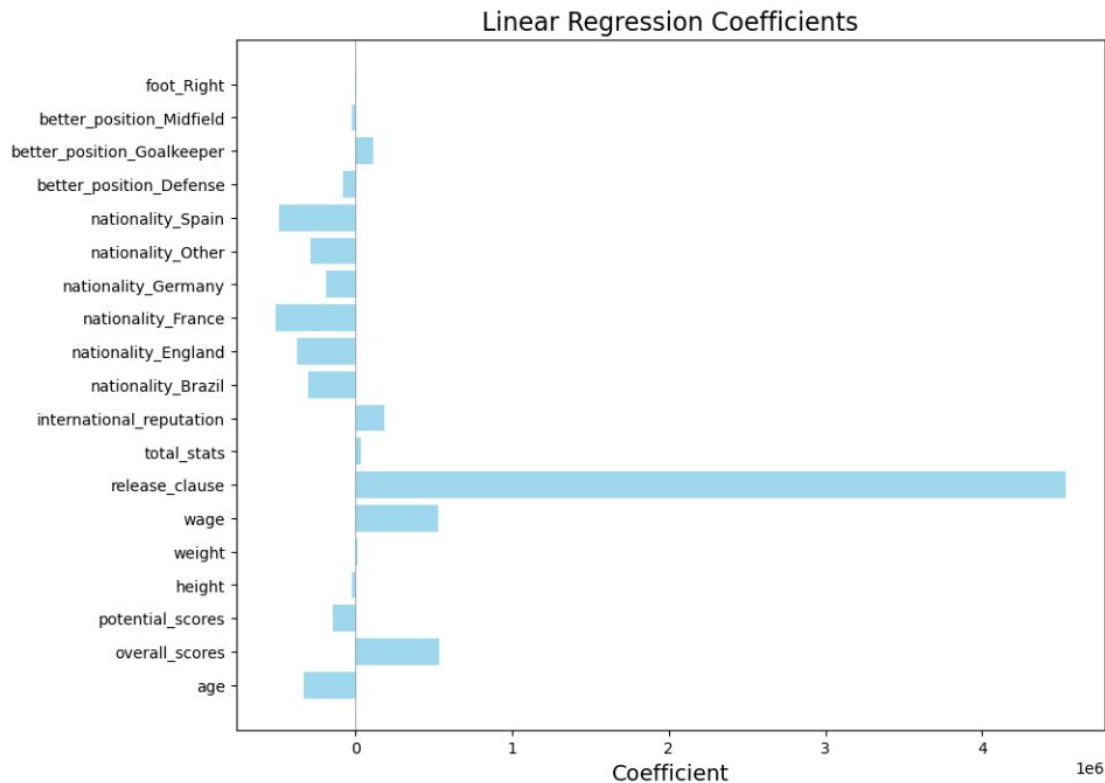
## R2 coefficient, release clause and outliers

From our findings, we see that if we keep the “release clause” the outliers, we have a higher  $r^2$  coefficient ( $r^2 = 0.9725$ ), compared where we get rid of the “release clause” and keep the outliers ( $r^2 = 0.7400$ ).

$r^2$  vs. Scenario



# Interpretation for model results



**Intercept: 2.906.693**

## **Coefficients:**

age : -333.696

overall\_scores : 532.967

potential\_scores : -144.272

height : -21.010

weight : 8.866

wage : 529.293

release\_clause : 4.531.966

total\_stats : 35.330

international\_reputation : 184.612

nationality\_Brazil : -305.620

nationality\_England : -377.411

nationality\_France : -510.246

nationality\_Germany : -190.868

nationality\_Other : -287.273

nationality\_Spain : -489.241

better\_position\_Defense : -78.842

better\_position\_Goalkeeper : 111.112

better\_position\_Midfield : -26.354

foot\_Right : 7.606

# Conclusions and limitations

Takeovers for future FIFA data analysis:

- Highlight the top players for their outstanding performances over a discrete season.
- Build a top football team with conditions as such budget, location, etc.

Due to time boundaries, we did not enter into these two applications:

- Decide when to transfer a player.
- Decide the best replacement for a transferred player.

Had to spend some time finding out the meaning of some abbreviations and finding that total stats were already a sum of variables like “attacking”, “skills”, and “power”, as well as finding out the meaningful variables to our model.

Also, we encountered very high errors and model overfitting issues.



# Methodology

Sample size: 17.125 observations

Source: Kaggle

Link to the data source:

[https://www.kaggle.com/datasets/ekrembayar/fifa-21-complete-player-dataset?select=fifa21\\_male2.csv](https://www.kaggle.com/datasets/ekrembayar/fifa-21-complete-player-dataset?select=fifa21_male2.csv)

Link to the github repository

[https://github.com/Yinguin/data\\_mid\\_bootcamp\\_project\\_FIFA\\_MoneyBall/blob/master/FIFA.ipynb](https://github.com/Yinguin/data_mid_bootcamp_project_FIFA_MoneyBall/blob/master/FIFA.ipynb)

