SC1015 Project:
Beautiful Data for the Beautiful Game

DSF 1 Group 3
Syed Mohammed Mosayeeb Al Hady Zaheen
Jayden Yeo He

Our dataset

For our project, we wanted to use data from the English Premier League

We obtained data from 8 seasons, so our dataset should contain 380 * 8 = 3040 rows

MATCHES NEWS TABLE STATS PLAYERS

Season
2016-17

	Club	MP	W	D	L	GF	GA	GD	Pts	Last 5
1	Chelsea	38	30	3	5	85	33	52	93	✓ ✓ ✓ ✓ ✓
2	Tottenham	38	26	8	4	86	26	60	86	✓ ✓ ✓ ✗ ✓
3	Man City	38	23	9	6	80	39	41	78	✓ ✓ ✓ ✓ –
4	Liverpool	38	22	10	6	78	42	36	76	✓ ✓ – ✓ ✗
5	Arsenal	38	23	6	9	77	44	33	75	✓ ✓ ✓ ✓ ✓
6	Man United	38	18	15	5	54	29	25	69	✓ – ✗ ✗ –
7	Everton	38	17	10	11	62	44	18	61	✗ ✓ ✗ ✗ –
8	Southampton	38	12	10	16	41	48	-7	46	✗ – ✓ ✗ –

Our dataset

- We got our initial dataset from Tara Nguyen's Premier League dataset from Kaggle
- We scraped remaining data from the Premier League's official website using Python Selenium Webscraper and Chrome Webdriver, then merged the two datasets together
- We obtained 35 columns in the final data set

```python
# Now, for every match number
for match_number in range(SEASON_FIRST_MATCHES[season], SEASON_FIRST_MATCHES[season]+380):

    # open the webpage dedicated to that match
    url = f'https://www.premierleague.com/match/{match_number}'
    driver.get(url)

    # This is the stats button XPATH
    stats_button_xpath = "//li[@data-tab-index='2']"

    # wait for the stats button to appear
    wait_till_element_appears(By.XPATH, stats_button_xpath, time=4)

    # click on the stats button
    try:
        stats_button = driver.find_element_by_xpath(stats_button_xpath)
        stats_button.click()
    except NoSuchElementException:
        print(
            "The element you were looking for could not be found. Check your XPATH")
```

The final dataset – after cleaning

	Season_x	HomeTeam_x	AwayTeam_x	HomePossession	AwayPossession	HomeTouches	AwayTouches	HomePasses	AwayPasses	HomeTackles	...
0	2010/11	Aston Villa	West Ham United	56.8	43.2	636	529	395	313	27	...
1	2010/11	Blackburn Rovers	Everton	30.4	69.6	450	729	208	469	15	...
2	2010/11	Bolton Wanderers	Fulham	46.5	53.5	592	636	336	394	26	...
3	2010/11	Chelsea	West Bromwich Albion	59.5	40.5	782	571	592	394	16	...
4	2010/11	Sunderland	Birmingham City	44.1	55.9	514	581	304	386	9	...
...	...	...	...	...	...	...	...	...	...	...	...
3035	2017/18	Newcastle United	Chelsea	41.9	58.1	585	764	406	569	19	...
3036	2017/18	Southampton	Manchester City	30.3	69.7	441	782	259	583	24	...
3037	2017/18	Swansea City	Stoke City	57.6	42.4	744	612	544	414	16	...
3038	2017/18	Tottenham Hotspur	Leicester City	64.0	36.0	672	453	480	265	20	...
3039	2017/18	West Ham United	Everton	56.6	43.4	652	528	479	365	11	...

3040 rows × 35 columns

```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3040 entries, 0 to 3039
Data columns (total 35 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Season_x             3040 non-null   object
 1   HomeTeam_x           3040 non-null   object
 2   AwayTeam_x           3040 non-null   object
 3   HomePossession       3040 non-null   float64
 4   AwayPossession       3040 non-null   float64
 5   HomeTouches          3040 non-null   int64
 6   AwayTouches          3040 non-null   int64
 7   HomePasses           3040 non-null   int64
 8   AwayPasses           3040 non-null   int64
 9   HomeTackles          3040 non-null   int64
 10  AwayTackles          3040 non-null   int64
 11  HomeClearances       3040 non-null   int64
 12  AwayClearances       3040 non-null   int64
 13  HomeOffsides         3040 non-null   int64
 14  AwayOffsides         3040 non-null   int64
 15  Date                 3040 non-null   object
 16  Referee              3040 non-null   object
 17  FullTime             3040 non-null   object
 18  Halftime             3040 non-null   object
 19  HomeGoals            3040 non-null   int64
 20  HomeGoalsHalftime    3040 non-null   int64
 21  HomeShots            3040 non-null   int64
 22  HomeShotsOnTarget    3040 non-null   int64
 23  HomeCorners          3040 non-null   int64
 24  HomeFouls            3040 non-null   int64
 25  HomeYellowCards      3040 non-null   int64
 26  HomeRedCards         3040 non-null   int64
 27  AwayGoals            3040 non-null   int64
 28  AwayGoalsHalftime    3040 non-null   int64
 29  AwayShots            3040 non-null   int64
 30  AwayShotsOnTarget    3040 non-null   int64
 31  AwayCorners          3040 non-null   int64
 32  AwayFouls            3040 non-null   int64
 33  AwayYellowCards      3040 non-null   int64
 34  AwayRedCards         3040 non-null   int64
dtypes: float64(2), int64(26), object(7)
```

The Game Plan

Project aim:

Generate **actionable recommendations** for Premier League teams to score more goals and win more football matches!

Problem formulation

The classic problem of football: *How do we score more goals?*

Data Science problems:

1. Can we predict **the number of goals scored *(response)*** using the **other relevant variables in the dataset *(predictors)***?
2. Can we use **feature importance** to determine which of these predictor variables is the most important to **maximise/minimise** for a team?

Data cleaning and preparation

- During the scraping, each season's data was put into 8 individual .csv files
- We combined the 8 .csv files and the Tara Nguyen data for the final dataset
- To clean the data for machine learning, we removed metadata such as the referee name, date of the match, season, etc.

Data cleaning and preparation

- We also removed some data that is *incidental*
- This means that the teams playing the match are not actively incorporating maximising/minimising these variables into their playstyles.
- These variables include cards, corners, number of fouls, etc.
- Notice how every team wants to minimise fouls and maximise corners, because they are clearly optimal
- Finally, we limited the data to just the home team as it would be easier to perform exploratory analysis and recognize patterns in the data by eye

```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3040 entries, 0 to 3039
Data columns (total 35 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Season_x            3040 non-null   object
 1   HomeTeam_x          3040 non-null   object
 2   AwayTeam_x          3040 non-null   object
 3   HomePossession      3040 non-null   float64
 4   AwayPossession      3040 non-null   float64
 5   HomeTouches         3040 non-null   int64
 6   AwayTouches         3040 non-null   int64
 7   HomePasses          3040 non-null   int64
 8   AwayPasses          3040 non-null   int64
 9   HomeTackles         3040 non-null   int64
 10  AwayTackles         3040 non-null   int64
 11  HomeClearances      3040 non-null   int64
 12  AwayClearances      3040 non-null   int64
 13  HomeOffsides        3040 non-null   int64
 14  AwayOffsides        3040 non-null   int64
 15  Date                3040 non-null   object
 16  Referee             3040 non-null   object
 17  FullTime            3040 non-null   object
 18  Halftime            3040 non-null   object
 19  HomeGoals           3040 non-null   int64
 20  HomeGoalsHalftime   3040 non-null   int64
 21  HomeShots           3040 non-null   int64
 22  HomeShotsOnTarget   3040 non-null   int64
 23  HomeCorners         3040 non-null   int64
 24  HomeFouls           3040 non-null   int64
 25  HomeYellowCards     3040 non-null   int64
 26  HomeRedCards        3040 non-null   int64
 27  AwayGoals           3040 non-null   int64
 28  AwayGoalsHalftime   3040 non-null   int64
 29  AwayShots           3040 non-null   int64
 30  AwayShotsOnTarget   3040 non-null   int64
 31  AwayCorners         3040 non-null   int64
 32  AwayFouls           3040 non-null   int64
 33  AwayYellowCards     3040 non-null   int64
 34  AwayRedCards        3040 non-null   int64
dtypes: float64(2), int64(26), object(7)
```

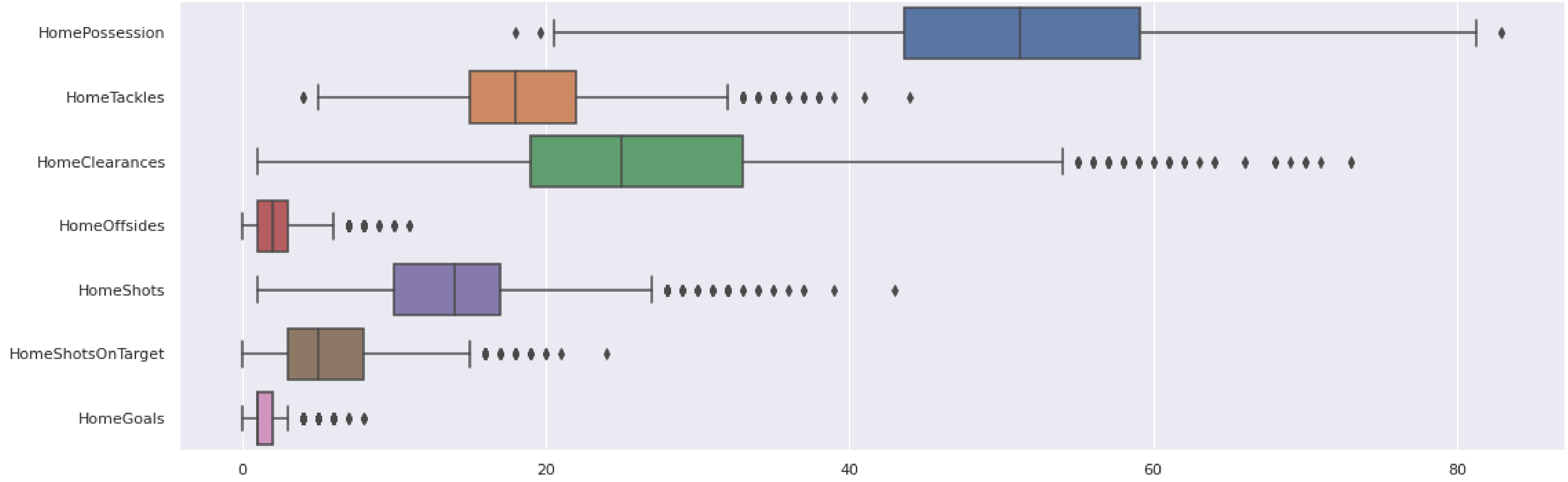
```
HomeData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3040 entries, 0 to 3039
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   HomePossession      3040 non-null   float64
 1   HomeTouches         3040 non-null   int64
 2   HomePasses          3040 non-null   int64
 3   HomeTackles         3040 non-null   int64
 4   HomeClearances      3040 non-null   int64
 5   HomeOffsides        3040 non-null   int64
 6   HomeShots           3040 non-null   int64
 7   HomeShotsOnTarget   3040 non-null   int64
 8   HomeGoals           3040 non-null   int64
dtypes: float64(1), int64(8)
```

Further exploratory analysis

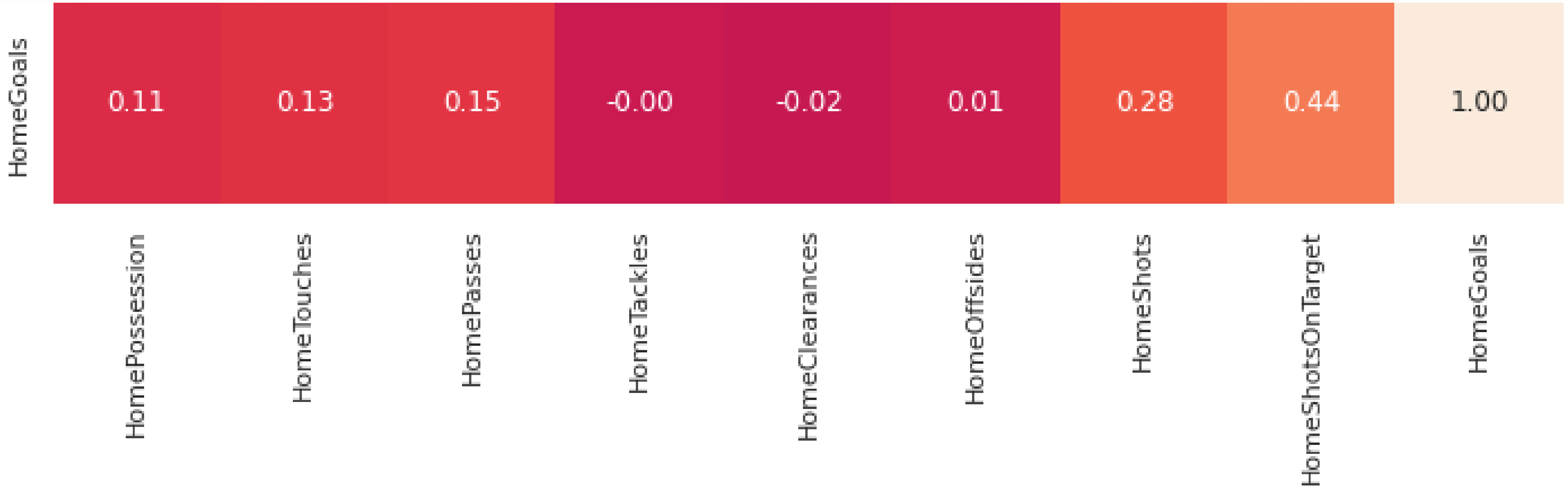
Our data at a glance

<matplotlib.axes._subplots.AxesSubplot at 0x7f348b005e90>



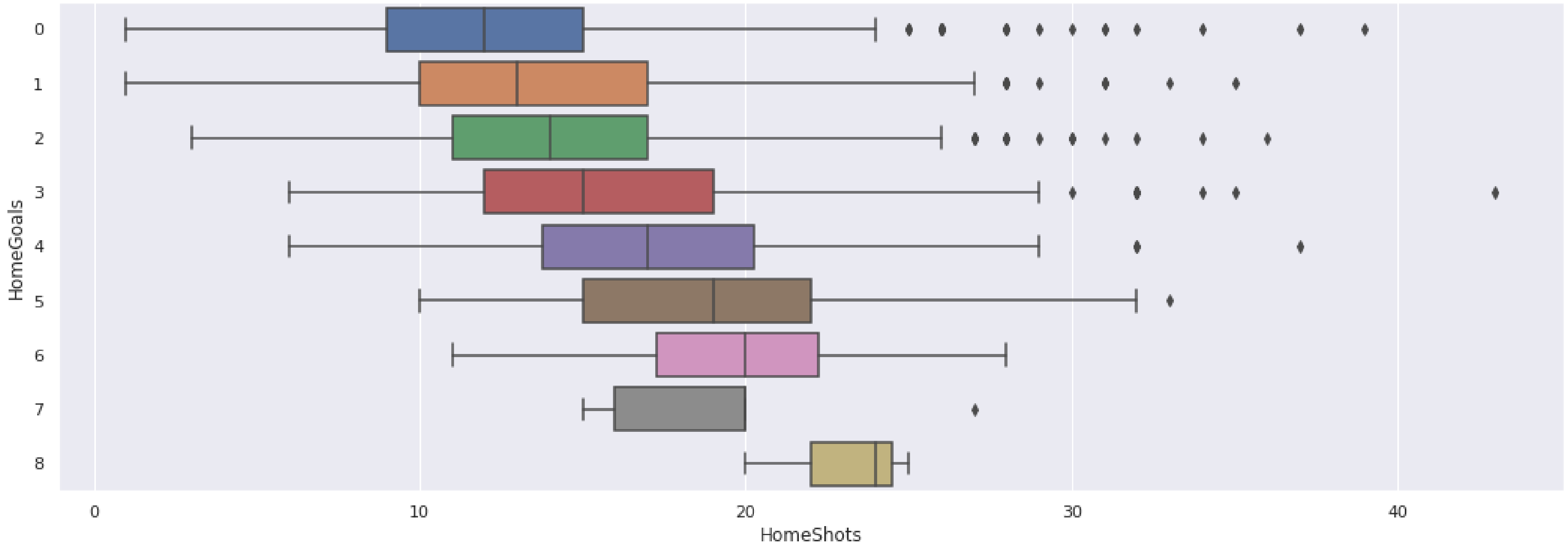
Further exploratory analysis

Correlation with HomeGoals



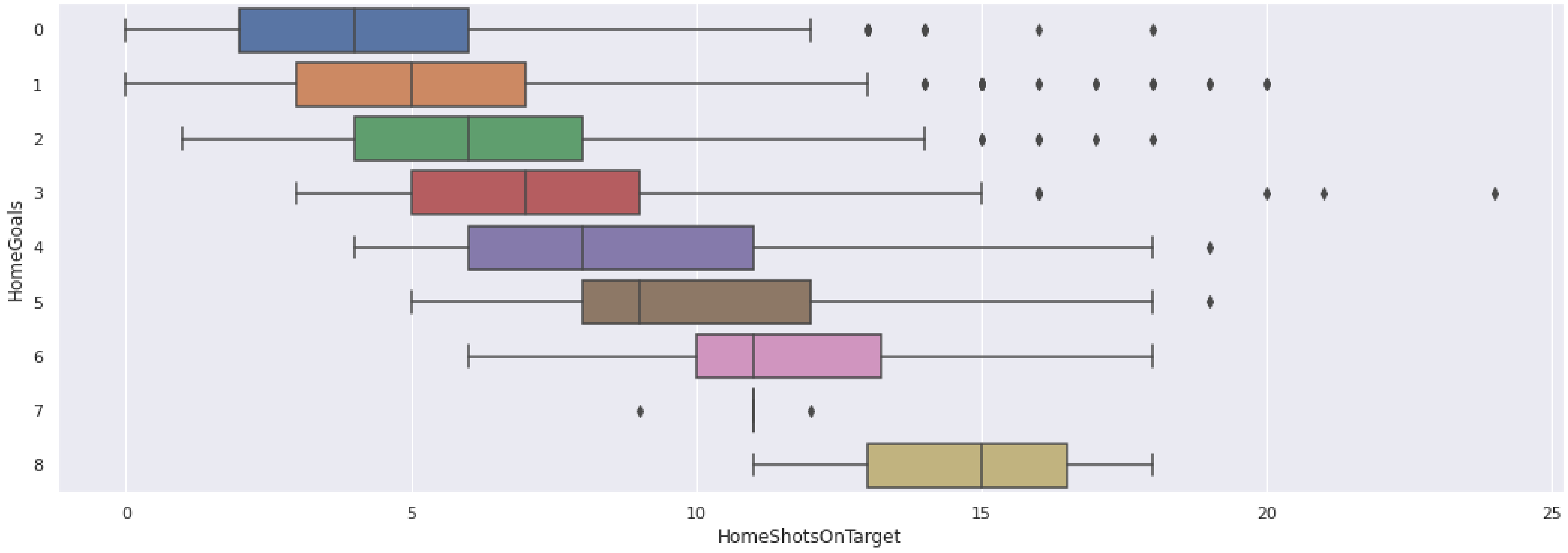
Further exploratory analysis

	HomeShots	HomeGoals
HomeShots	1.000000	0.275334
HomeGoals	0.275334	1.000000



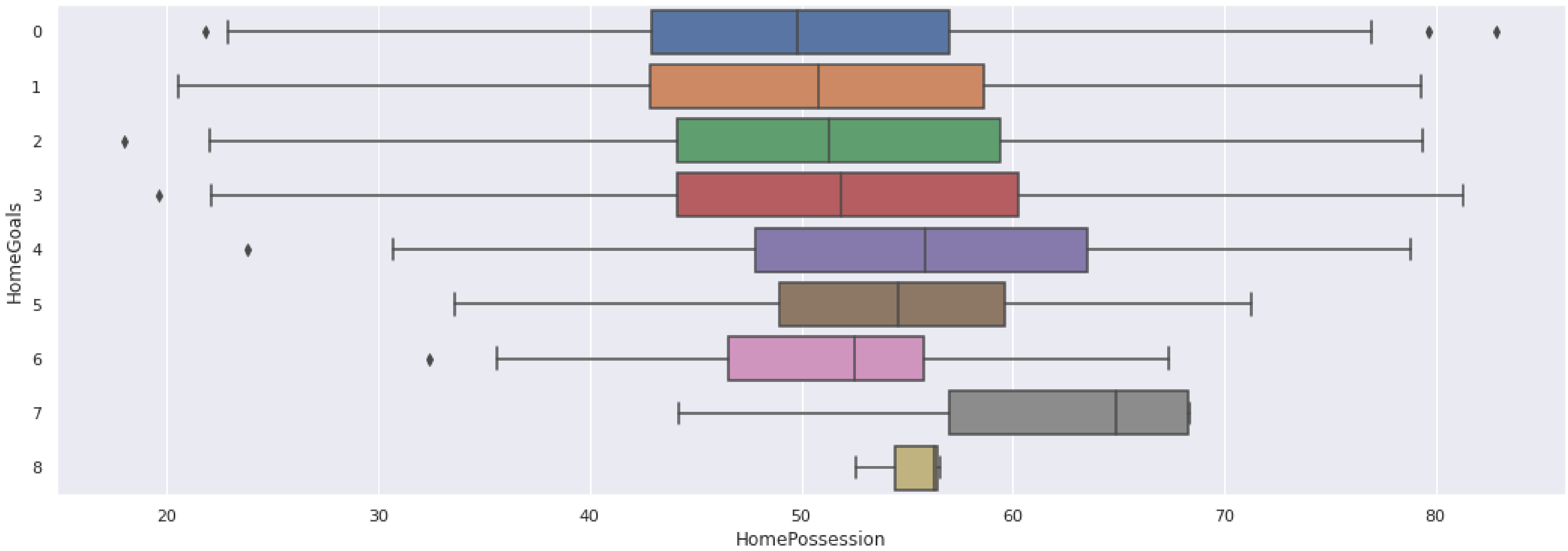
Further exploratory analysis

	HomeShotsOnTarget	HomeGoals
HomeShotsOnTarget	1.00000	0.44286
HomeGoals	0.44286	1.00000



Further exploratory analysis

	HomePossession	HomeGoals
HomePossession	1.000000	0.105717
HomeGoals	0.105717	1.000000



MODEL 1: PREDICTING NUMBER OF GOALS SCORED

Model used: Multivariate Linear Regression

Response: HomeGoals

Predictors: HomePasses, HomePossession, HomeTouches, HomeClearances, HomeTackles, HomeOffsides, HomeShots, HomeShotsOnTarget

RESULTS

```
Goodness of Fit of Model        Train Dataset
Explained Variance (R^2)        : 0.24585267882156603
Mean Squared Error (MSE)        : 1.3240264390971843

Goodness of Fit of Model        Test Dataset
Explained Variance (R^2)        : 0.18300473277553309
Mean Squared Error (MSE)        : 1.250544499136237
```

- The accuracy of the model seems quite low
- We will try another model next in order to better predict goals

RESULTS

```
Intercept of Regression          : b =  1.6562849791701781

Coefficient of HomePossession (times 1000):     :  -14.692344924709069
Coefficient of HomeTouches (times 1000):        :  -9.648298296960684
Coefficient of HomePasses (times 1000): :  11.717850497555034
Coefficient of HomeTackles (times 1000):        :  14.521979100089407
Coefficient of HomeClearances (times 1000):     :  11.741204486041056
Coefficient of HomeOffsides (times 1000):       :  15.137778678294772
Coefficient of HomeShots (times 1000):   :  5.822368316389428
Coefficient of HomeShotsOnTarget (times 1000):  :  175.72291998552026
```

- The coefficient of HomeShotsOnTarget is 0.175, much higher than the coefficient of all other variables
- Thus, to maximise goals scored, teams need to maximise shots on target

MODEL 2: PREDICTING NUMBER OF GOALS SCORED

Model used: Random Forest Classifier (using Grid Search Cross-Validation)
Response: HomeGoals (Same as Model 1)

Predictors: HomePasses, HomePossession, HomeTouches, HomeClearances, HomeTackles, HomeOffsides, HomeShots, HomeShotsOnTarget (Same as Model 1)

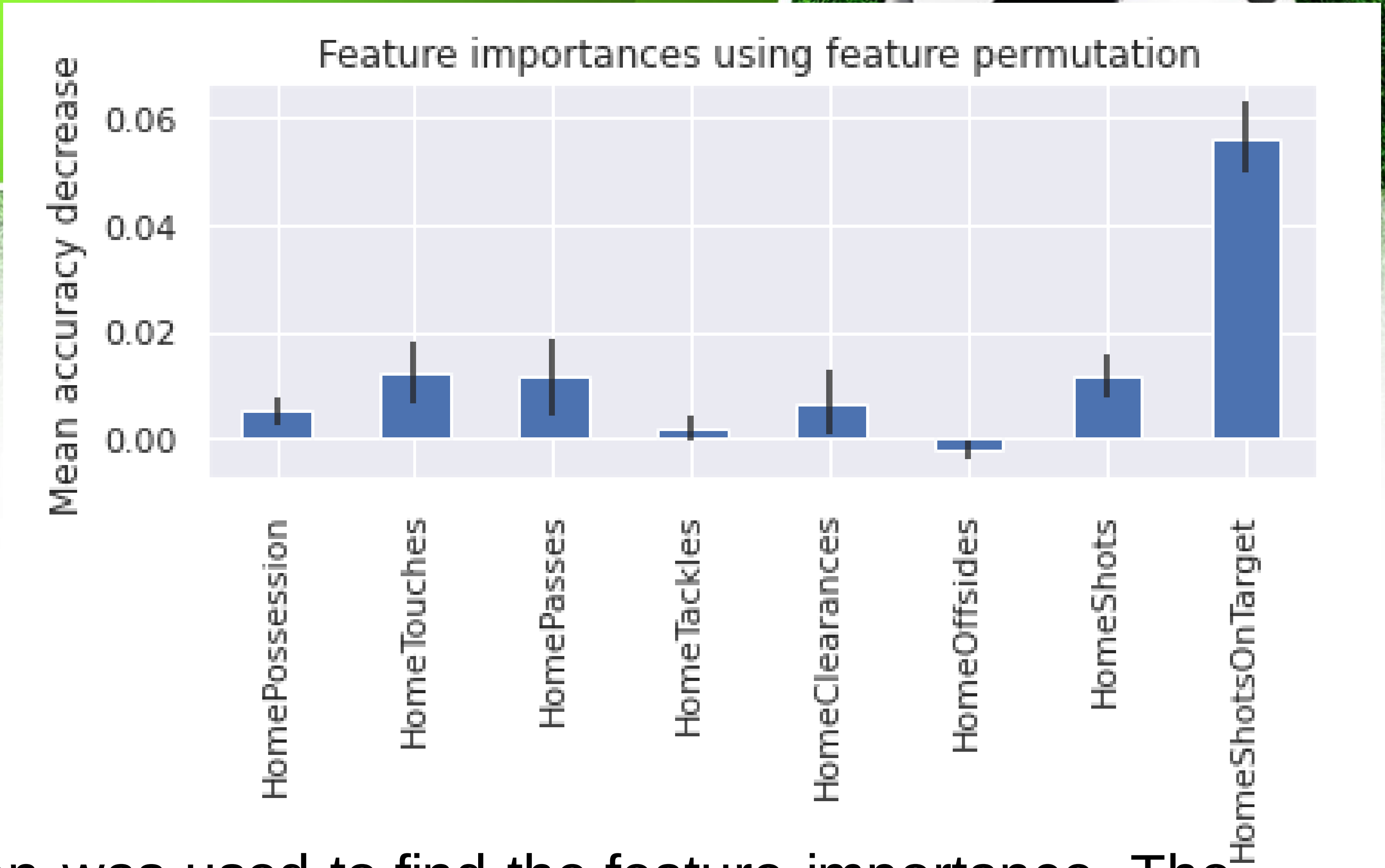
RESULTS

```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 0.41140350877192298

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 0.3526315789473684
```

- R^2 value is not the best at 0.35 for the testing set, but it is better than the linear regression

RESULTS



- Feature Permutation was used to find the feature importance. The feature importance of HomeShotsOnTarget again dwarfs all other variables

Overall insights

- We were definitely able to predict number of goals with a greater accuracy with Model 2
- Our feature importance calculations show that HomeShotsOnTarget was the most important feature for *both* models
- It is quite obvious that taking more shots on target will lead to more goals.
- So maybe we need to re-formulate our problem. Let’s try that

Problem (re-)formulation

- Given that it is quite obvious that taking more shots leads to more goals, we are more interested in the other non-shot variables and how they may relate to goals scored.
- How teams maximise and minimise other variables can be considered their 'style of play'
- Our ultimate objective is to relate style of play to number of goals scored, and then recommend that teams follow a certain style of play.
- So now, our Data Science problem is:

  Is there **a pattern** to any of the remaining variables, from which we can derive style of play? Which style of play **relates most to goals scored?**

MODEL 3: CLUSTERING

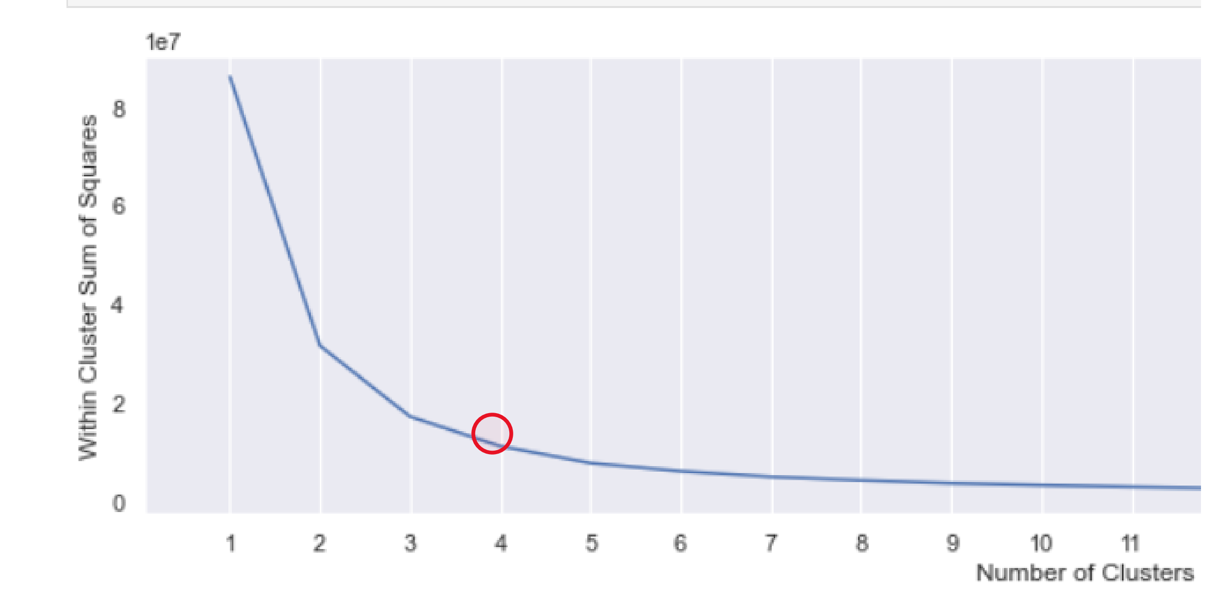
Model used: K-means clustering

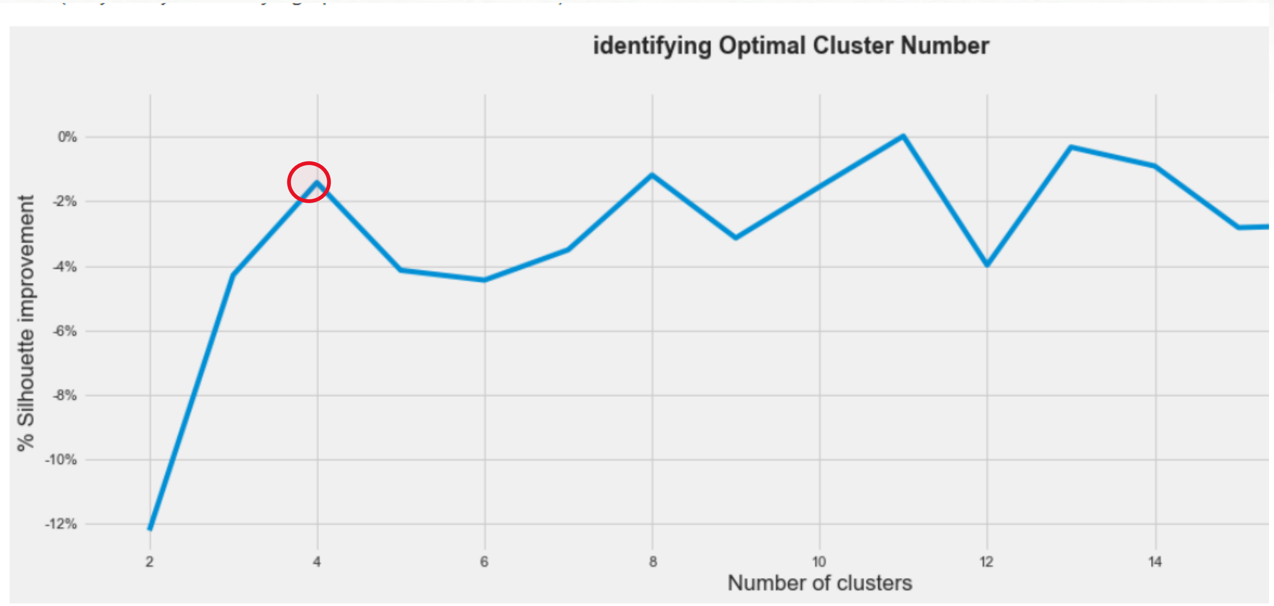
Variables: HomePasses, HomePossession, HomeTouches, HomeClearances, HomeTackles, HomeOffsides

(NO HomeGoals, HomeShots, HomeShotsOnTarget!)

Finding optimum cluster number

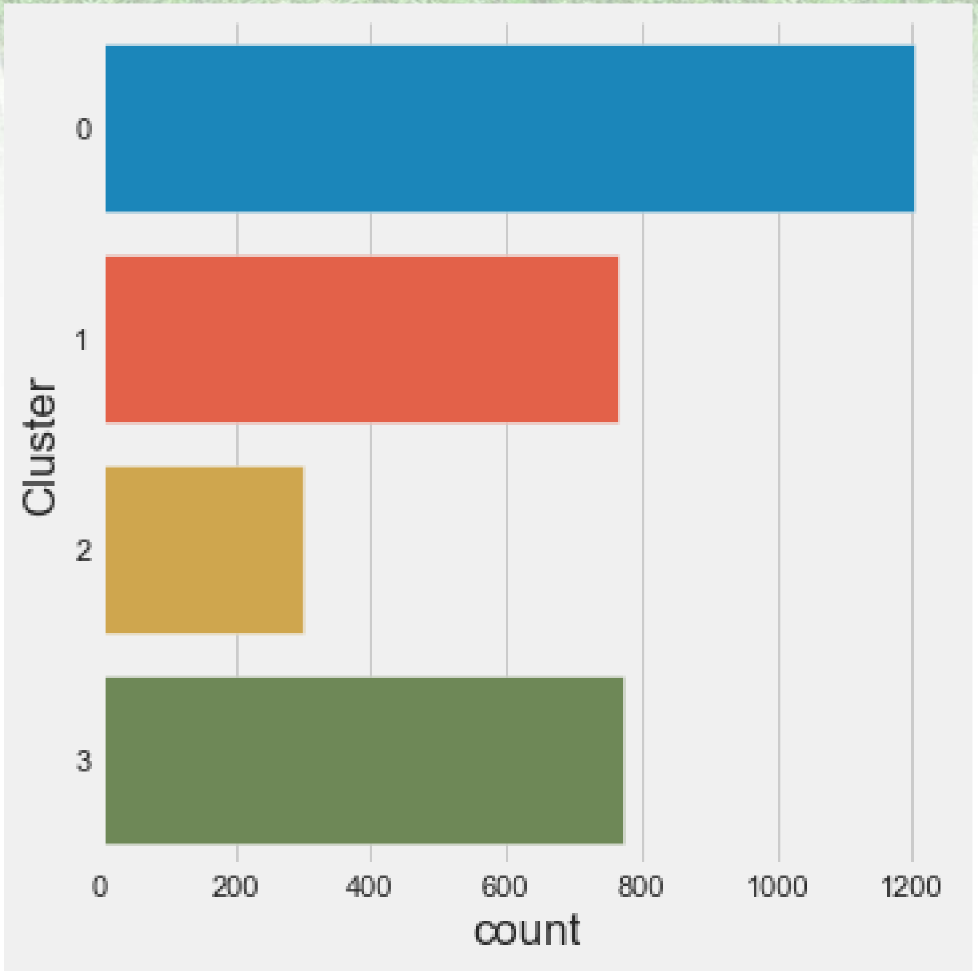
- In order to find the optimum cluster number, we used the **cluster sum of squares** and **silhouette score**



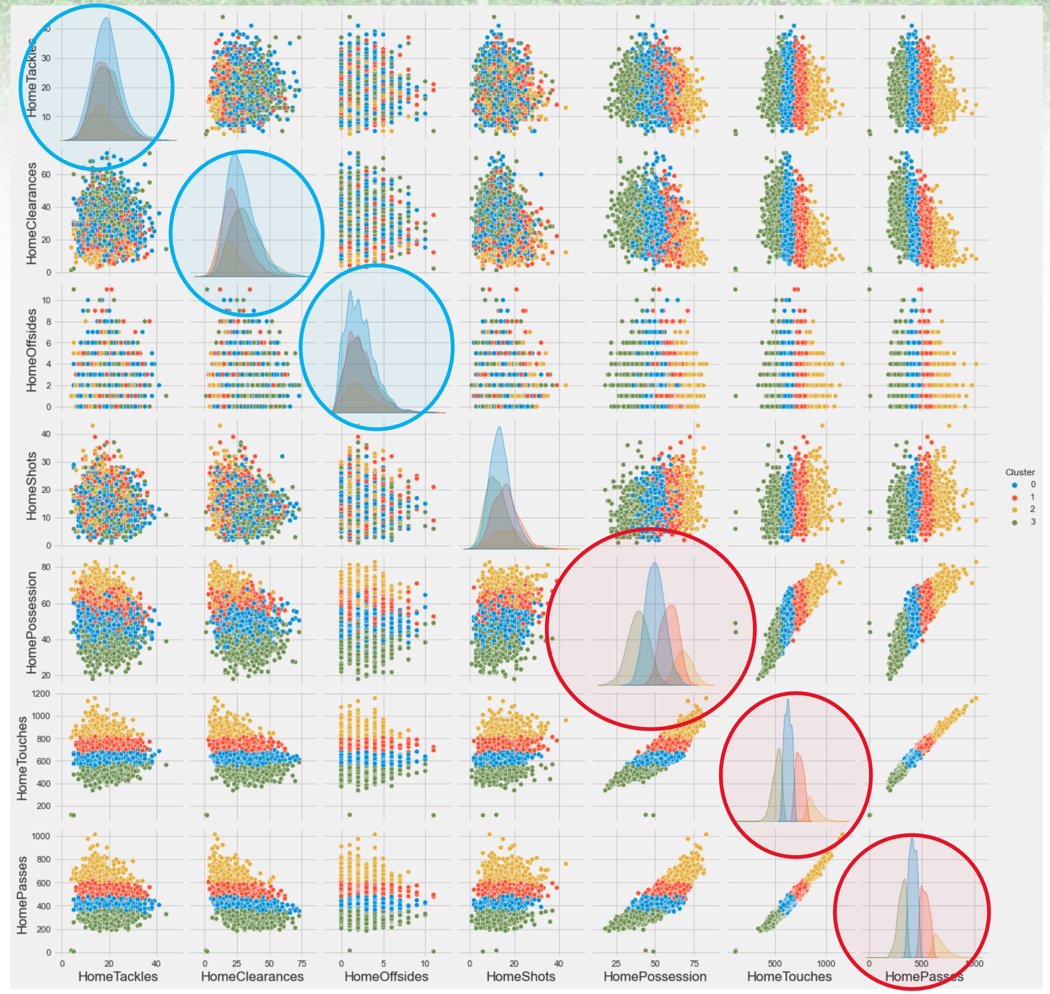


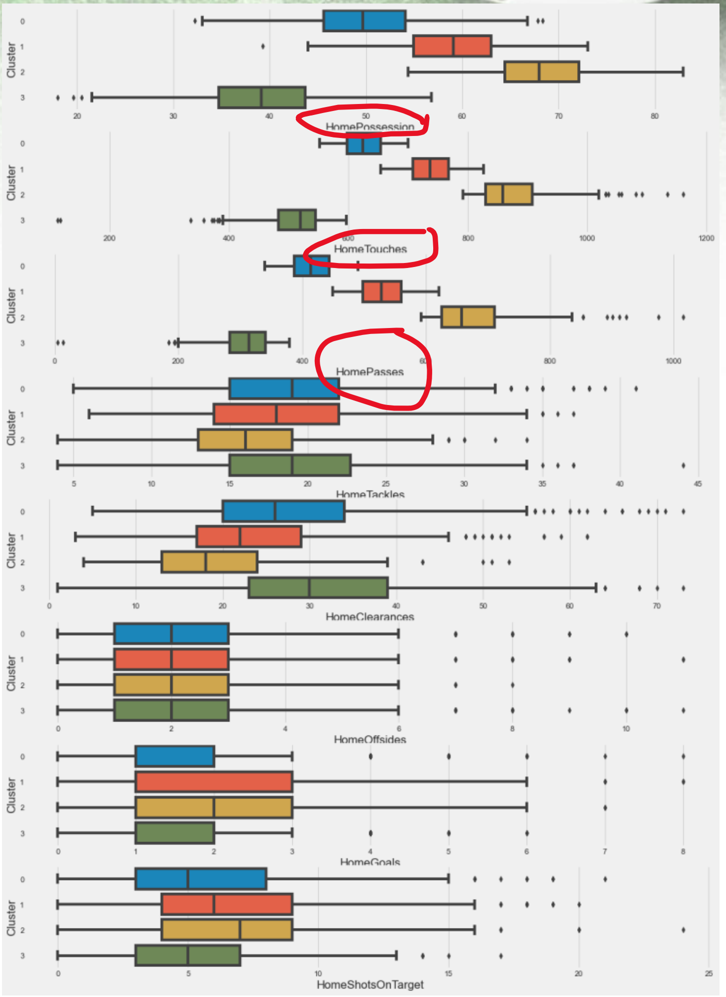
- We found that **4** is the ideal number to perform our clustering

RESULTS



RESULTS

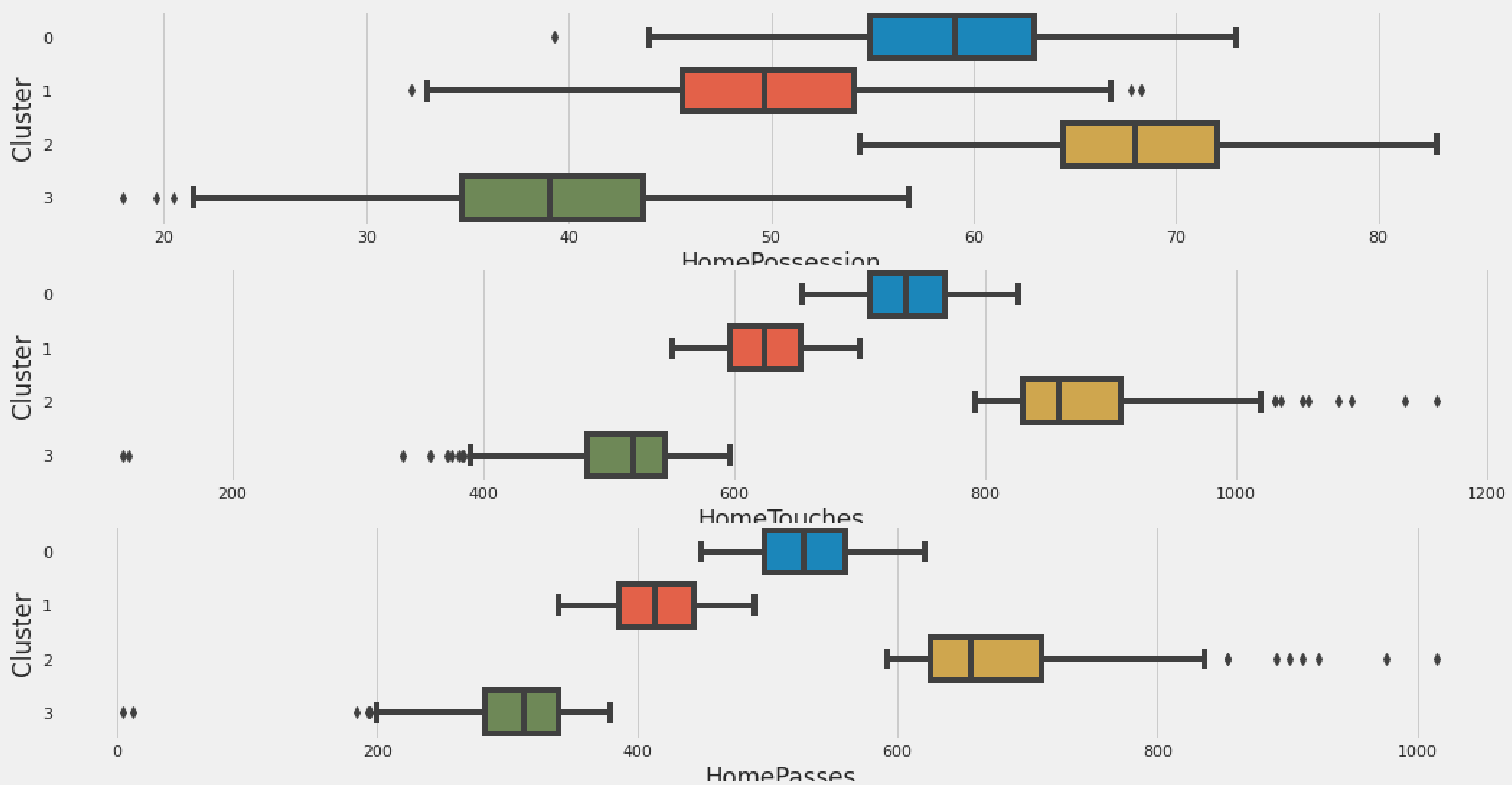

HomeTackles
HomeClearances
HomeOffsides
HomeShots
HomePossession
HomeTouches
HomePasses
Cluster
0
1
2
3


Cluster
HomePossession
HomeTouches
HomePasses
HomeTackles
HomeClearances
HomeOffsides
HomeGoals
HomeShotsOnTarget

RESULTS

Cluster rankings, by most passes, possession and touches:

1. Cluster 2
2. Cluster 0
3. Cluster 1
4. Cluster 3



RESULTS

Average number of goals scored by each cluster

```
Cluster 0 has scored 1.434637801831807 per game
Cluster 1 has scored 1.7238219895287958 per game
Cluster 2 has scored 2.0398671096345513 per game
Cluster 3 has scored 1.3863049095607236 per game
```

Cluster rankings,
by average goals per game:

1. Cluster 2
2. Cluster 0
3. Cluster 1
4. Cluster 3

Same!

Cluster rankings,
by possession, passes and touches:

1. Cluster 2
2. Cluster 0
3. Cluster 1
4. Cluster 3

Insights

- From this analysis, we can clearly see that **teams should choose to maximise <u>possession, passing and touches</u> in order to likely score more goals** throughout the game

- Is this always the case?

Limitations

- There have clearly been many famous wins where the winning team had much lower possession. We know that some teams, like Mourinho's Chelsea in 2013/14, willingly chose to give up possession to the opposition as a playstyle.
- Further data analysis is needed to check our recommendation

Feature Engineering

- The reason why Chelsea willingly gave up possession was so that they can play more "counter-attacks"
- Counter-attacks are when the opposition team has possession and suddenly lose it, and their goal is not well-defended because all their players were trying to attack
- This results in shots that have a higher probability of being a goal, which we shall call **"shot quality"**

Feature Engineering

The shot taken here by the Chelsea player will have a higher chance of being a goal, as he is 1v1 with the goalie

This shot will have higher “shot quality”

Feature engineering

“Shot quality" =
(goals actually scored) – (goals predicted to be scored, by bi-variate linear regression based on shots taken)

(For bi-variate linreg:
Predictor: HomeShotsOnTarget,
Response: HomeGoals)

Feature engineering

- We expect shot quality to **increase** as teams play with **less** possession, passes and touches
- Thus, the **reverse** rankings for average shot quality of the clusters should be the **same** as those ranked by possession, passes and touches!

RESULTS

Average shot quality by each cluster

```
Shot quality of cluster 0 (times 100) :  6.778059441242723
Shot quality of cluster 1 (times 100) :  -11.00987797960349
Shot quality of cluster 2 (times 100) :  28.919301389227574
Shot quality of cluster 3 (times 100) :  -0.8784884730264999
```

Cluster REVERSED rankings, by average **“shot quality”** per game:

4. Cluster 1
3. Cluster 3
2. Cluster 0
1. Cluster 2

NOT Same!

Cluster rankings,
by possession, passes and touches:

1. Cluster 2
2. Cluster 0
3. Cluster 1
4. Cluster 3

Conclusion

In conclusion, Premier League teams will be more successful in scoring goals if they:

1. **Take more shots on target;**
2. **Maximise possession, passes and touches**

Thank you!