

Springboard Capstone 2 Milestone Report 1

Aaron Kochman

Scouting Potential Premier League Signings with Machine Learning

Problem Statement:

Elite soccer teams around the world are constantly on the lookout for players to give their squad the advantage they need to win more games, tournaments, and ultimately increase profits. Machine learning can be utilized with data surrounding player value, game stats, and compiled metrics of players' abilities based on previous seasons in video games like FIFA 20.

Building a scouting report for a premier league team based on player attributes and value could be a valuable tool for scouting replacement players or potential signings that can increase a club's performance.

My project utilizes web scraping and machine learning to analyze player performance, transfer value, and game statistics to predict signings and potential values for my favorite Premier League club Arsenal.

Data Sources

In order to explore different datasets to build prediction models, different techniques were used to wrangle, clean, and combine datasets from selected sources. The three main datasets produced in this project stemmed from SoFIFA, transfermarkt, and Fantasy Premier League.

transfermarkt

Transfermarkt tracks and compiles transfer data from multiple soccer leagues globally as well as compiles statistics including goals, assists, and player accolades.

#14 Pierre-Emerick Aubameyang





Date of birth/Age: Jun 18, 1989 (30)

Place of birth:  Laval

Citizenship:  Gabon

Height: 1,87 m

Position: Centre-Forward

Agent: Relatives

Current international:  Gabon

Caps/Goals: 51/20



Arsenal

Premier League

League level:  First Tier

Joined: Jan 31, 2018

Contract until: 30.06.2021

\$79.80m

Last update: Jun 13, 2019

Fantasy Premier League

Fantasy Premier League contains attributes as well as real-world statistics. The main component on FPL that is useful for determining a player's value are components like Influence, Creativity, and Threat.

Pierre-Emerick Aubameyang

Forward

Arsenal




Form 3.5	GW 12 2pts	Total 69pts	Price £11.0	TSB 24.9%
-------------	---------------	----------------	----------------	--------------

Influence 337.0	Creativity 177.6	Threat 464.0	ICT Index 97.7
--------------------	---------------------	-----------------	-------------------


SoFIFA

SoFIFA compiles player data from the video game franchise FIFA, produced by EA Sports. Player data on sofifa includes statistics like Speed, Agility, Aggression, Ball Control, Shooting, as well as transfer values.



P. Aubameyang (ID: 188567)

FIFA 20 NOV 13, 2019

Pierre-Emerick Aubameyang  ST LM Age 30 (Jun 18, 1989) 6'2" 176lbs

88 Overall Rating	88 Potential	Value €57M	Wage €205K
-------------------	--------------	------------	------------

Data Wrangling

Once our data sources were determined, the methods behind wrangling data from each data source needed to be developed.

transfermarkt Data Wrangling

Transfermarkt is HTML based thus the Python package BeautifulSoup was used for parsing out transfer data as well as player statistics. In essence each of the players

profiles contained a unique URL and each statistic or metric are encased in HTML classes including 'responsive-table', 'td', and 'tbody'.

```
if __name__ == "__main__":
    scraper = PageScraper()
    soup = scraper(LEAGUES_URL)
    LeagueTables = soup.find("table", class_="items").find("tbody")
    Leagues = LeagueTables.find_all("a", href=re.compile("wettbewerb/[A-Z]{2}1"), title=re.compile("\w"))
    Leagues = Leagues[:N_LEAGUES]
    LeagueUrlDic = { league.text : BASE_URL + league["href"] for league in Leagues }
    LeaguesData = []
    for leagueName, leagueUrl in LeagueUrlDic.items():
        print( "Scraping the %s..." %leagueName)
        LeaguesData.append( League( leagueName, leagueUrl, scraper))

    #flattening all players information to pandas.DataFrame and exporting to csv
    PlayerProfiles = [player.PlayerData for league in LeaguesData for team in league.TeamsData for player in team.PlayersData]
    df = pd.DataFrame( PlayerProfiles)
    df.to_csv("transfer.csv", index=False)
```

```
Scraping the Premier League...
['17/18', 'Jul 1, 2017', 'Benfica', 'Man City', '22,00 mil. €', '40,00 mil. €']
['15/16', 'Jul 1, 2015', 'Rio Ave FC', 'Benfica', '1,20 mil. €', '500 K €']
['12/13', 'Jul 1, 2012', 'GD Ribeirão', 'Rio Ave FC', 0, 'Free transfer']
['11/12', 'Jul 1, 2011', 'Benfica U19', 'GD Ribeirão', 0, 'Free transfer']
['10/11', 'Jul 1, 2010', 'Benfica U17', 'Benfica U19', 0, 0]
['09/10', 'Jan 1, 2010', 'São Paulo U17', 'Benfica U17', 0, '?']
Ederson done
```

After scraping transfermarkt for performance statistics (goals, assists, games played, etc.) as well as transfer valuation, each data set was parsed out by season (17/18, 18/19, 19/20, etc.) and cleaned to produce formatting that would be standardized between datasets. Data from each scrape was joined based on player name and season year, but not every player had transfer data for certain years.

Fantasy Premier League Data Wrangling

Fantasy Premier League contains an API that can be called using a JSON request. Once connected to the API, a JSON file was downloaded containing the relevant data.

```
▼ root: {} 8 keys
  ► events: [] 38 items
  ► game_settings: {} 22 keys
  ► phases: [] 11 items
  ► teams: [] 20 items
    total_players: 6801617
  ▼ elements: [] 541 items
```

The JSON file was then normalized with the json_normalize package from pandas.io.json and exported as a CSV.

```
df = json_normalize(d['elements'])
print('Columns:\n', list(df), '\n')
print('Dataframe Head:\n', df.head())
```

SoFIFA (Kaggle) Data Acquisition

EA Sport's FIFA 20 was released September 24, 2019 so at the time of the data wrangling phase I was not hopeful to utilize FIFA data since transfermarkt and FPL seemed to be more up to date. I stumbled upon a very clean and useful dataset on Kaggle by user stefanoleone992 who had scraped SoFIFA and compiled a CSV of every player from FIFA 15-20 (6 FIFA video games and 15,458 players throughout multiple years). I decided after struggling through my own web scraping process that this dataset was going to be the most useful for machine learning, since it included player skill metrics as well as transfer values in a clean format from one source.

Data Compiling and Matching Player Names with FuzzyWuzzy

Once each data source was cleaned and processed, I was ready to combine datasets. The major problem during this step was matching names that were not formatted in a standard method to explore holistically.

```
# List for dicts for easy dataframe creation
dict_list = []
# iterating over our players without salaries found above
for name in df_fifa.short_name:
    # Use our method to find best match, we can set a threshold here
    match = match_name(name, df_prem_field_players.name, 60)

    # New dict for storing data
    dict_ = {}
    dict_.update({"fifa_name" : name})
    dict_.update({"transfermarkt_name" : match[0]})
    dict_.update({"score" : match[1]})
    dict_list.append(dict_)

merge_table = pd.DataFrame(dict_list)
# Display results
merge_table.head()
```

	fifa_name	transfermarkt_name	score
0	K. De Bruyne	Kevin De Bruyne	81
1	V. van Dijk	Virgil van Dijk	77
2	M. Salah	Mohamed Salah	67
3	H. Kane	Harry Kane	71
4	Alisson		-1

Once each player was fuzzy matched between datasets, the matching data was joined together and data that did not match was dropped. This process was repeated to join transfermarkt data and FPL data to the FIFA dataset.

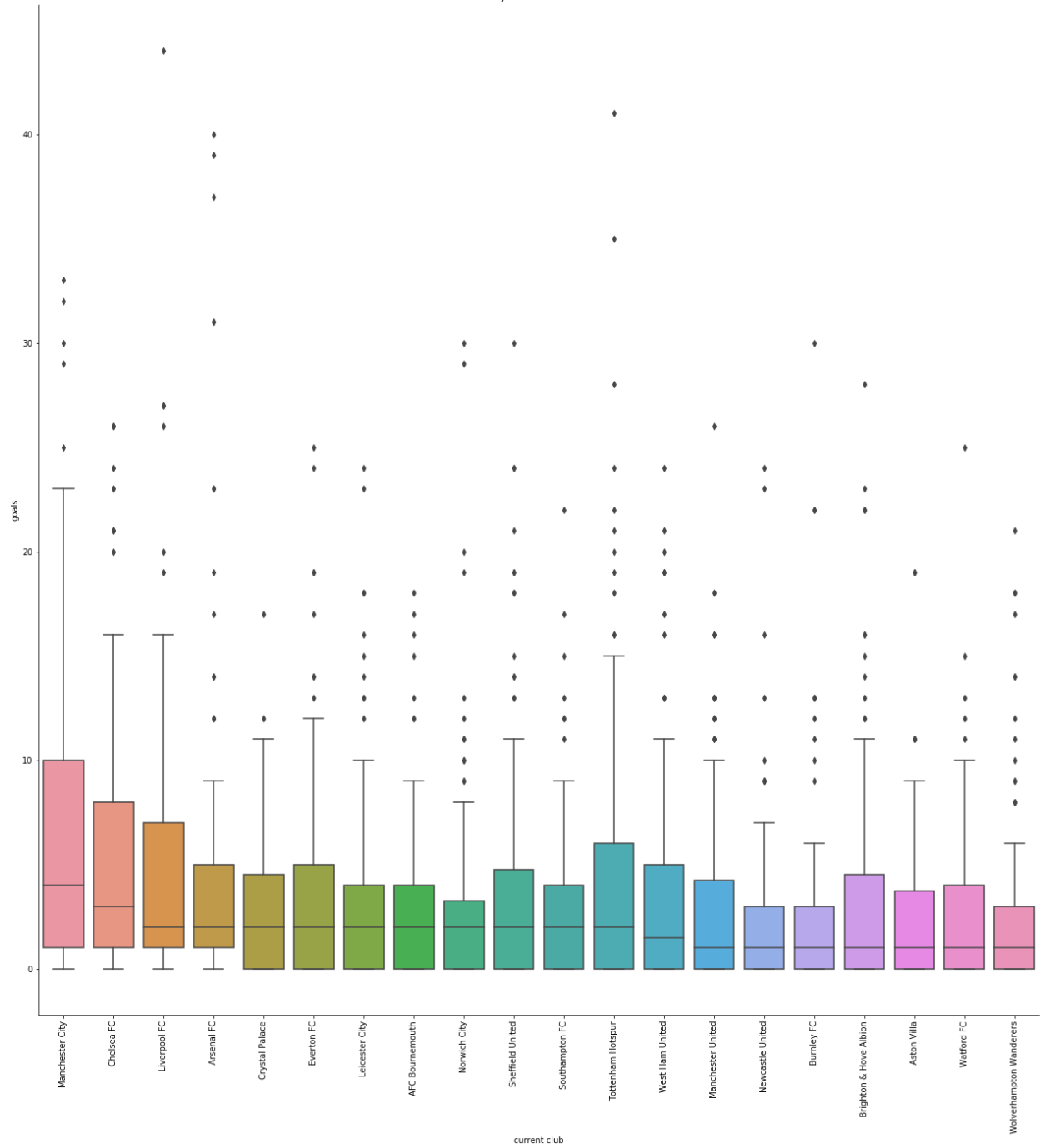
The resulting data was analyzed and explored to determine use but consequently, it was determined that too much data was being lost in the fuzzy matching phase to provide enough training data to use during machine learning.

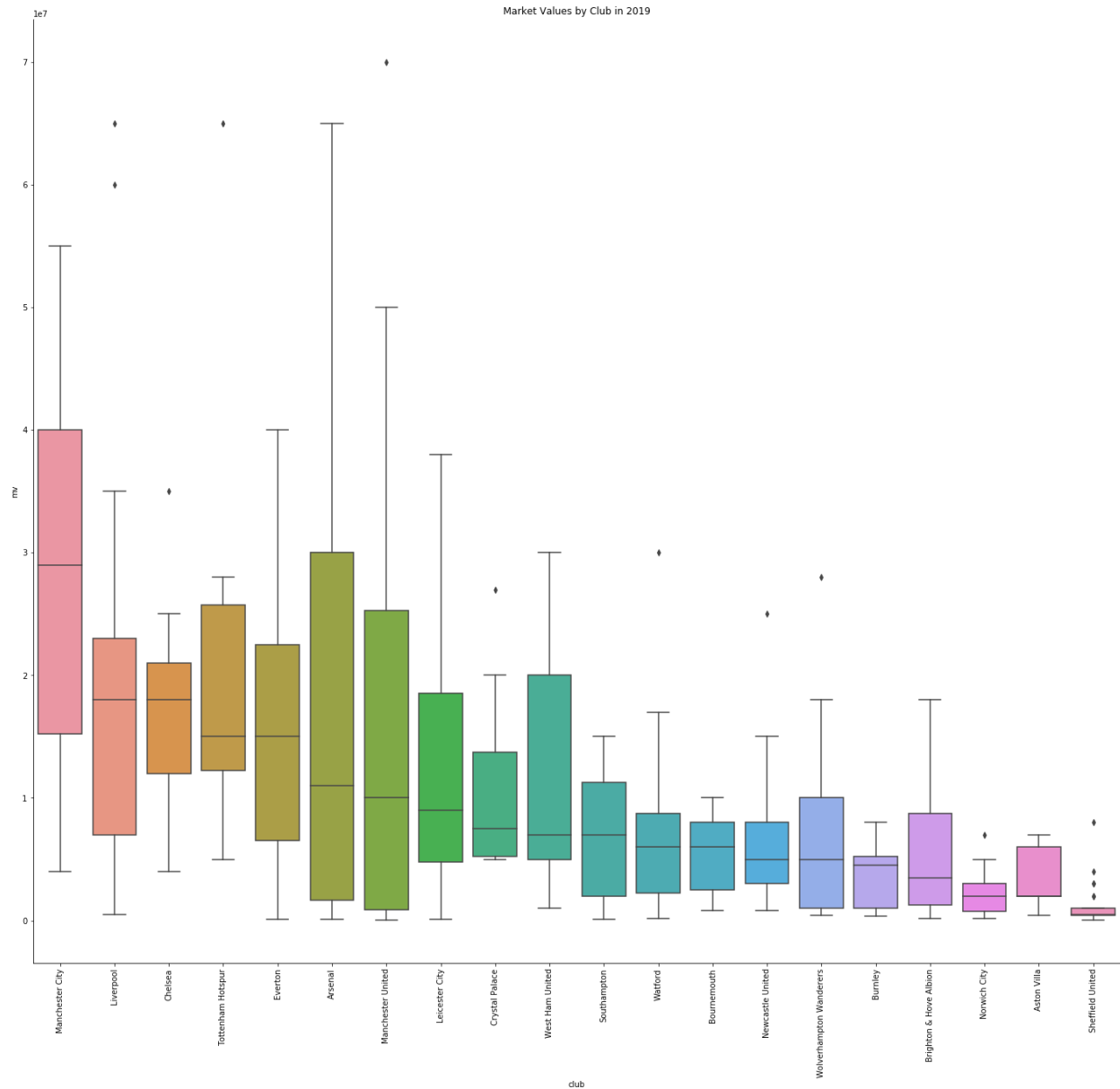
Exploratory Analysis and Exploratory Machine Learning

From the combined data and individual data sources, we could analyze a variety of questions including, what is the team in the English Premier League with the highest market value, which team scored the most goals, and how could the data be prepared to create an accurate machine learning model.

We can see from the data that higher value teams tend to score more goals and have larger budgets for players. We can also determine that players who are currently on higher tiered teams are probably higher valued players.

Goals by Club from 2015-2019

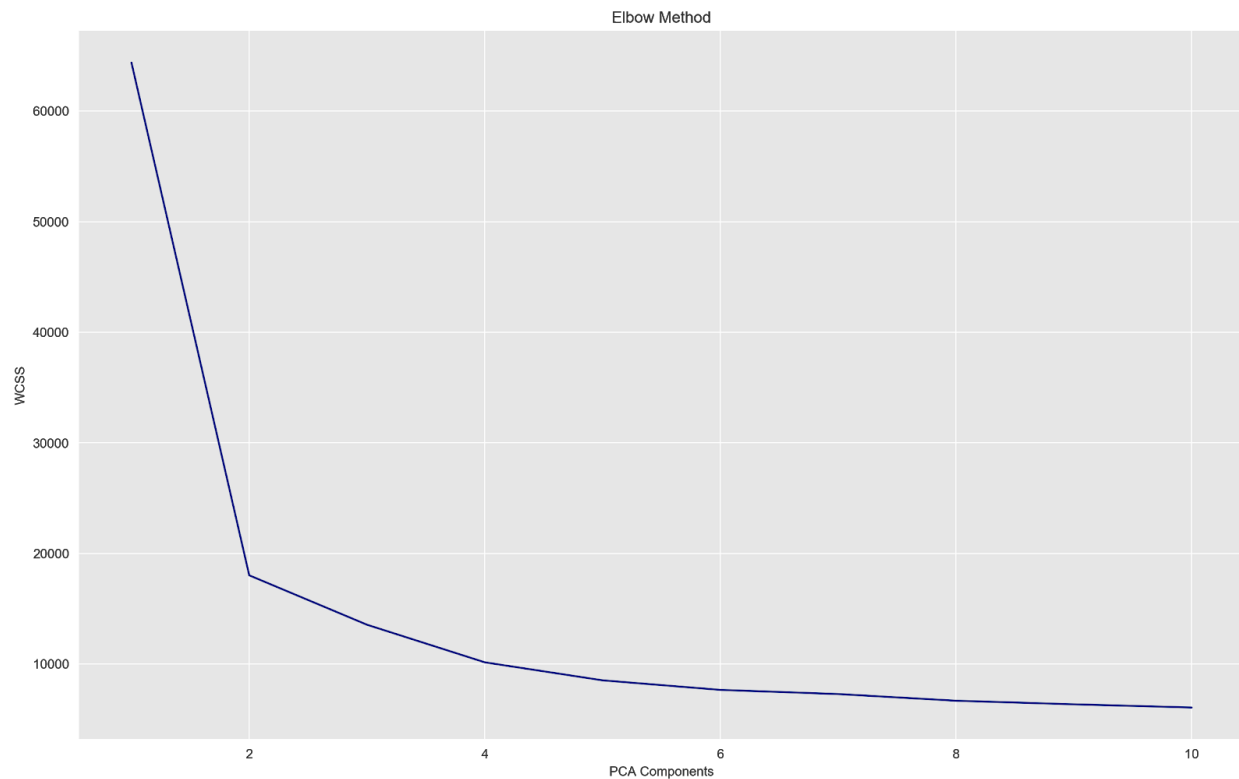




The combined data was split into a 2-Component Principal Component Analysis based on features that stemmed mostly from the FIFA data set.

```
features = ['attacking_crossing',  
'attacking_finishing',  
'attacking_heading_accuracy',  
'attacking_short_passing',  
'attacking_volleys',  
'skill_dribbling',  
'skill_curve',  
'skill_fk_accuracy',  
'skill_long_passing',  
'skill_ball_control',  
'movement_acceleration',  
'movement_sprint_speed',  
'movement_agility',  
'movement_reactions',  
'movement_balance',  
'power_shot_power',  
'power_jumping',  
'power_stamina',  
'power_strength',  
'power_long_shots',  
'mentality_aggression',  
'mentality_interceptions',  
'mentality_positioning',  
'mentality_vision',  
'mentality_penalties',  
'mentality_composure',  
'defending_marking',  
'defending_standing_tackle',  
'defending_sliding_tackle',  
'overall','goals', 'assists']
```


We can see from the Elbow plot and PCA variance of the combined data that 2 components would be sufficient to analyze.

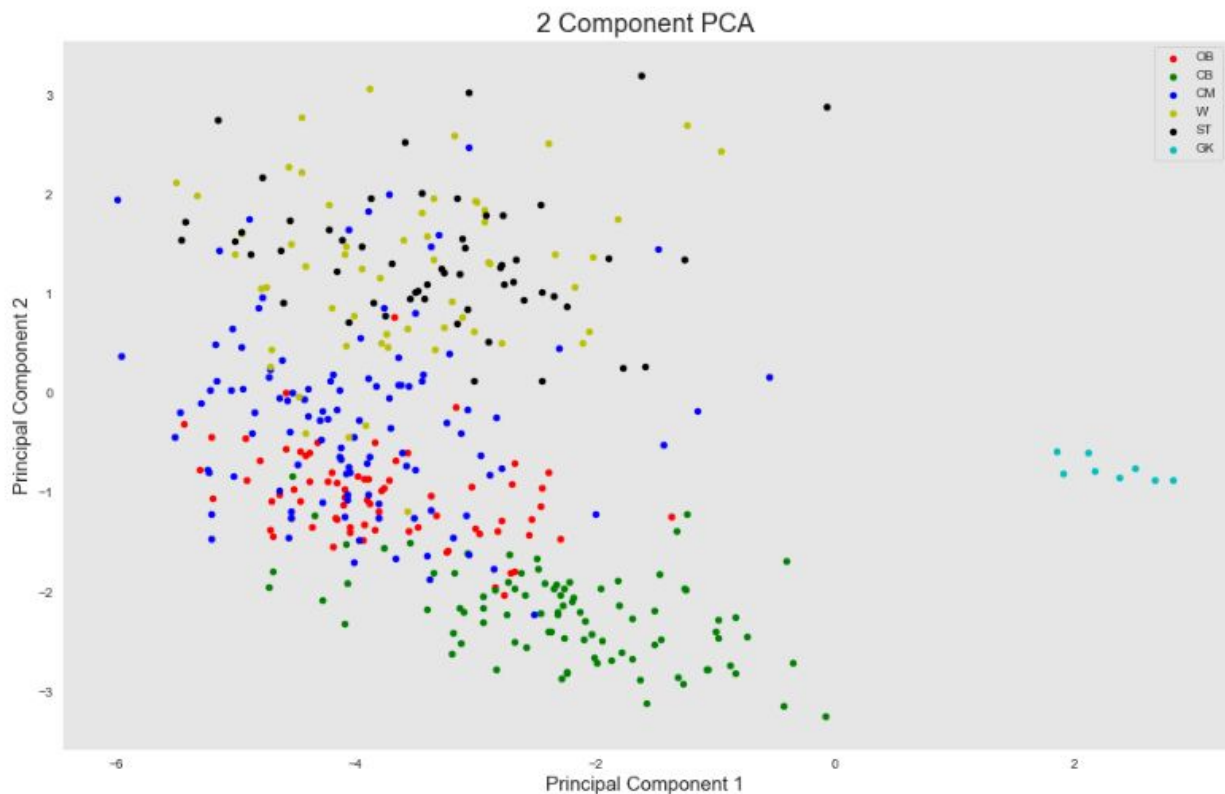


```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents
                           , columns = ['principal component 1', 'principal component 2'])
principalDf.head()
```

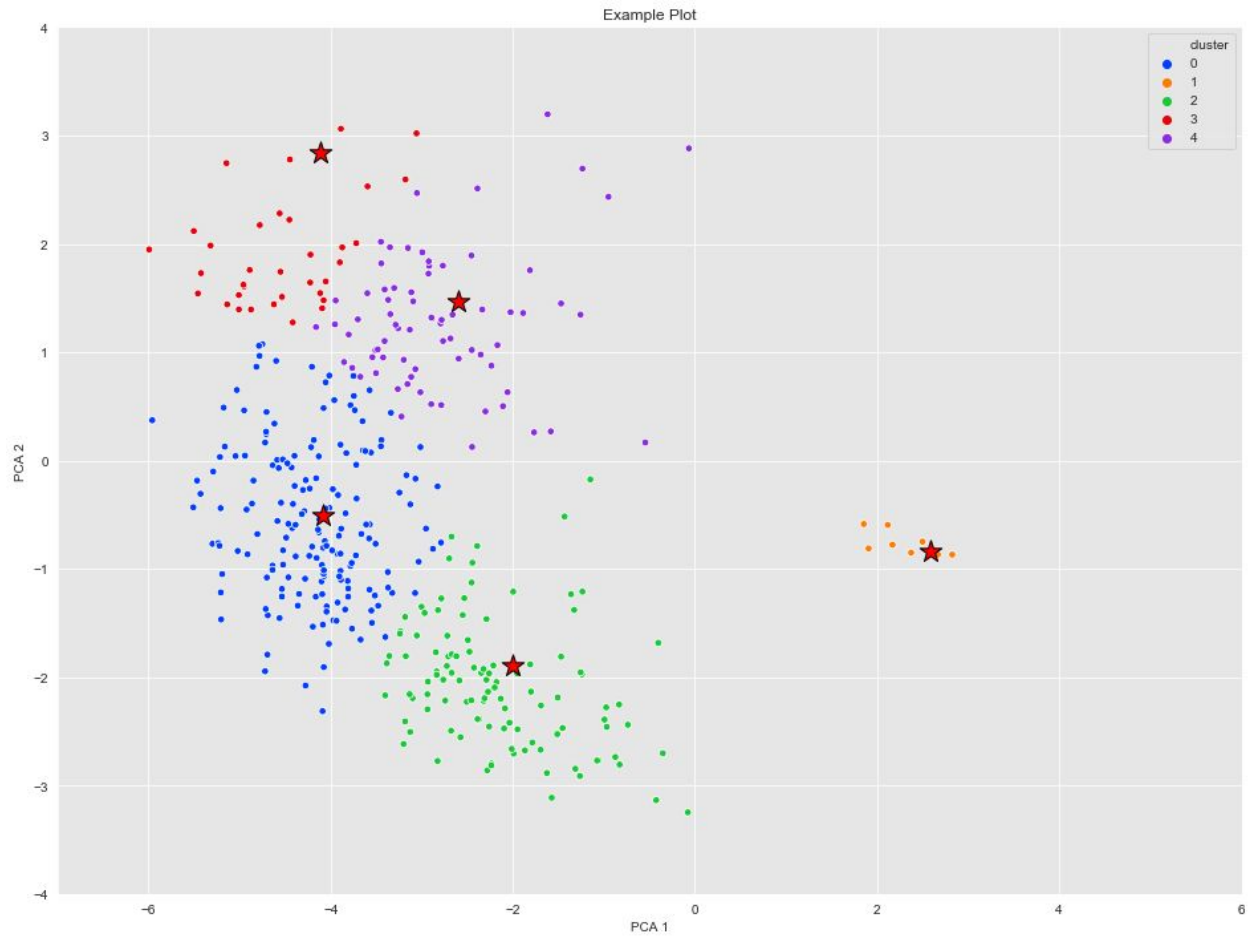
	principal component 1	principal component 2
0	-3.270765	-2.164194
1	-3.517926	-1.864947
2	-3.411034	-1.868319
3	-3.663187	-2.139183
4	-3.400502	-2.167475

```
pca.explained_variance_ratio_
array([0.80055365, 0.08004547])
```

The first trial of the 2-Component PCA on the combined dataset showed that ultimately, goalkeepers were outliers as expected. We can see that defenders are fairly different from midfielders or strikers, but midfielders seem to overlap with strikers quite often.

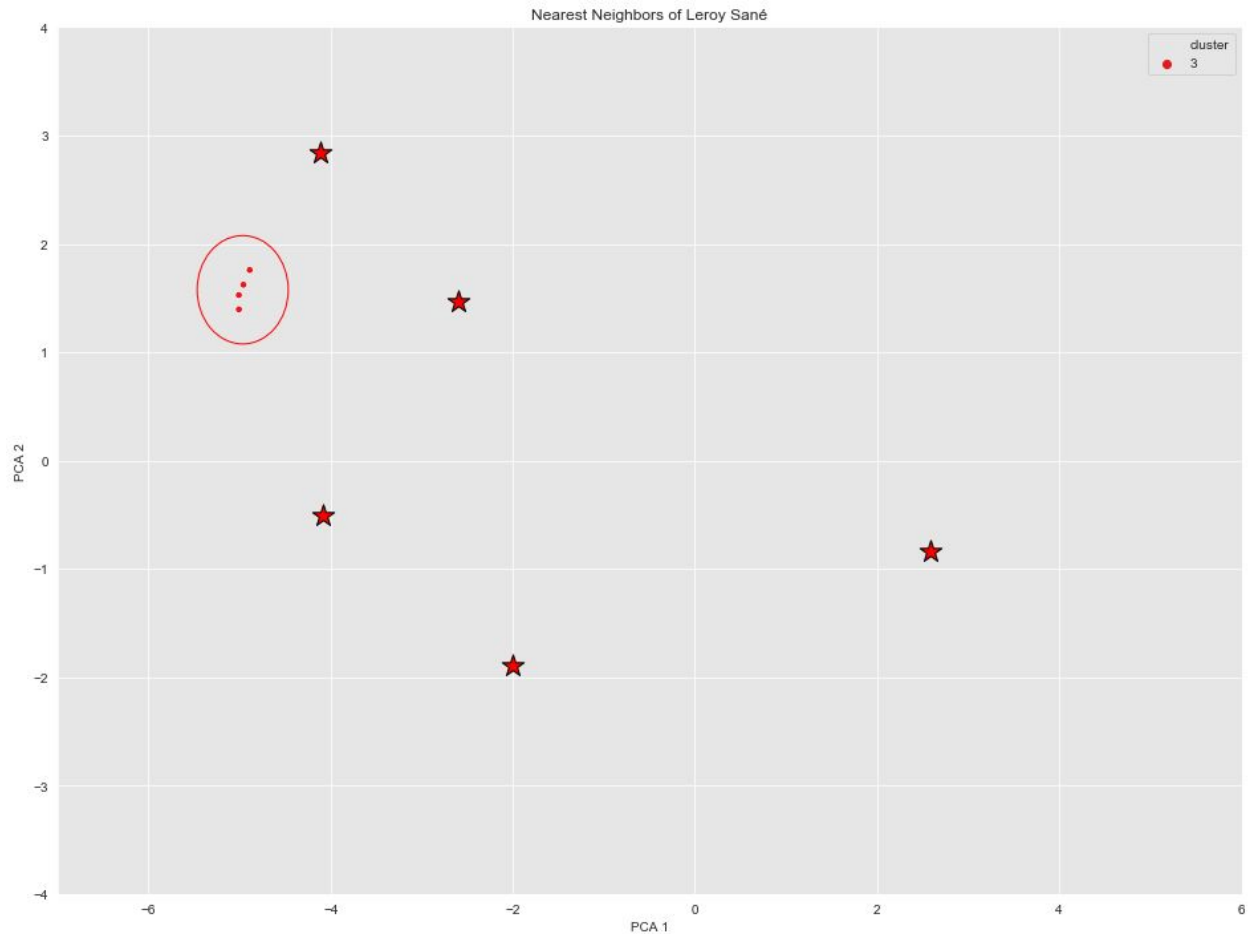


Kmeans clustering was used to determine clusters for each player based on position to verify that nearest neighbors could be extrapolated accurately. We can see centroids for each position group that could be utilized to predict player positions based on the PCA components.



We can also see that a player like Leroy Sané can be matched with players of similar features and positions.

	cluster	name	player_position	principal component 1	principal component 2
98	3	Marcus Rashford	ST	-4.956602	1.623060
886	3	Alexandre Lacazette	ST	-5.006974	1.527708
1281	3	David Silva	CM	-4.887999	1.759096
818	3	Bernardo Silva	W	-5.006415	1.395515



Once we determined that the data could be used to cluster players together, we decided to run machine learning on the entire FIFA dataset, excluding the SoFIFA and transfermarkt datasets.

Hypotheses and Prediction Models

From the exploratory analyses we can formulate a business case as well as develop hypotheses to be applied to the business case. From exploring the data it seems as though we can utilize the FIFA statistics to predict transfer values for players.

The hypothesis testing could then be outlined as:

Null Hypothesis (H0): There is no relationship between player values and skill level.

Alternative Hypothesis (Ha): There is some relationship between player values and skill level.

The business case for this project was determined to be to develop a scouting report for an EPL team Arsenal, to predict replacement transfer players as well as values to current players on the squad.

Methodology for Producing a Transfer Scouting Report Based on FIFA Metrics

Clustering Method

1. Sklearn k-means Clustering
 - a. Utilized for clustering player positions

Nearest Neighbors Method

1. Sklearn neighbors KDtree
 - a. Utilized to determine nearest neighbors for players if transferred

Models for Transfer Value Predictions based on PCA grouping

1. XGBoost Linear Regression
2. Sklearn Linear Regression