DATA MINING II

BANA 7047-001

FINAL PROJECT REPORT

A study of Unsupervised Learning techniques on the FIFA 20 dataset.

By:

Sankirna Joshi

Group – 15

Master's in Business Analytics

M13263600

University of CINCINNATI
CARL H. LINDNER COLLEGE OF BUSINESS

Contents

Abstract

FIFA 20 is a football simulation video game published by Electronic Arts as part of the FIFA series. It is one of the most popular games for Football and has a very rich data on the players and team. This dataset is available freely on Kaggle and has substantial potential for studying many different data mining techniques and especially unsupervised learning techniques. In this project we explore the data through the domain of unsupervised learning performing principal component analysis and clustering analysis. One goal of this project is to best describe the variation in the different types of players. Doing so would equip us with insight into how to best choose players in a team. In a high-dimensional data, it is often difficult to develop an intuition of the features and our goal in this project is to reduce the dimensionality of the dataset so that we can visualize the relationships between the features and clusters in our dataset. We start with 104 features and bring down the dimensionality to 28 features by selecting key features using our domain knowledge, removing highly correlated features using regression techniques, and then further to just two principal components using PCA. We visualize the data using these principal components, perform clustering analysis and visualize the clusters and develop an inference for the same.

Introduction

1. Background

FIFA 20 is the 27th installment in the FIFA series, and was released on 27 September 2019 for Microsoft Windows, PlayStation 4, Xbox One, and Nintendo Switch. The game will feature more than 30 official leagues, over 700 clubs and over 17,000 players. Included for the first time is the Romanian Liga I and its 14 teams, as well as Emirati club Al Ain, who were added following extensive requests from the fans in the region. With the amount of player and team data available on the game, this makes for an interesting dataset with a rich in-depth breakdown of every possible recorded attribute a player can have. In addition, football lends a detailed structure to the player data due to the dynamic nature of the game and various positions a player may command. This results in a very complex and interesting structure in the dataset which we will look to explore in this project through unsupervised learning techniques

2. Problem domain

Unsupervised learning is one of the three main domains of machine learning, along with supervised learning and reinforcement learning. In unsupervised learning, the objective is to look for previously undetected patterns in a dataset without any pre-existing labels and with a minimum of human supervision. In contrast to supervised learning that usually makes use of human-labelled data, unsupervised learning allows for the modeling of probability densities over inputs. The FIFA 20 is a dataset consisting of the entire roster of players available on the EA sports' FIFA 20 video game. It includes various physical

attributes of the players, their market value, teams and nationalities; and is an ideal candidate for studying unsupervised learning techniques due to the richness of the data such as the vast variety of different clusters that can be formed among the players and the available labels that can withheld to compare the performance.

Problem Description

Our goal is to study and apply unsupervised learning to the FIFA dataset to mine for interesting patterns. While the labels are not usually available for unsupervised learning approaches, domain knowledge can prove to be useful for inferring the patterns that we uncover. In this project, we will explore the clusters the players form based on their attributes to see if we have discovered any patterns inherent in the data, compare it with withheld labels and evaluate the clustering performance. Visualization helps us build an intuition about our data, and we will reduce the data to just two dimensions to be able to view the clusters on a plot and use Principal Component Analysis for this purpose.–

Project Description

1. Getting Started

In this project, we will analyze a dataset containing data on various football players available in the FIFA 20 dataset on Kaggle. The dataset for this project can be found on the Kaggle website.

2. Data Exploration

This dataset provides the complete statistics available at the player level in the FIFA 20 game. This dataset was scraped from the sofifa.com and is very clean in terms of the structure and expected data types. We will just transform the data to the format we need and keep only selected features which might be useful in our analysis. Our data has a lot of features that are explained or have the same information captured in them as other variables. Some stats are also not available based on the player's positions. Some key observations in the dataset:

- *potential* and *overall* are highly correlated. *potential* is basically an integer greater than or equal to *overall*.
- overall is a computation based on all the other skill ratings of a player such as shooting, passing, etc.
- Unless a player plays at a Goalkeeper position (GW), all his goalkeeper statistics are NaNs.
- The columns ls, st, rs, lw etc. are playing positions in the game and the data in these columns is basically the max potential of a player if he were to play in that

position. We will assume a player only plays in his preferred position and we will drop all these columns.

- For our analysis, we will drop all columns unnecessary for our analysis as and when we reach that conclusion. For now, all descriptive columns like *sofifa_id*, *player_url*, *nationality* etc. will be dropped.
- *player_positions* are the preferred positions of the player. We will keep only the first playing position for our analysis.

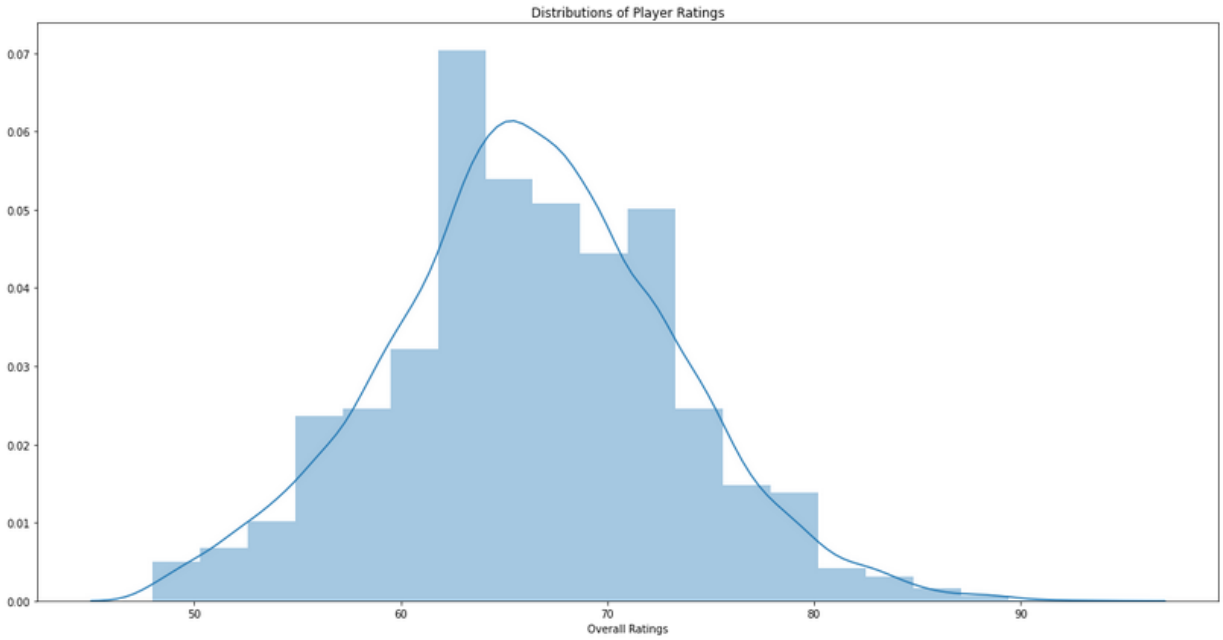

Figure 1

Since, *overall* decides the overall quality of a player, we can plot its histogram to visualize how players are distributed. Figure 1. shows the distribution of player overall score. As we are interested in the top-rated players, we remove players with overall rating lower than 70 just as a soft criterion. In Figure 2, we can see an almost normal distribution of player age with their rankings.

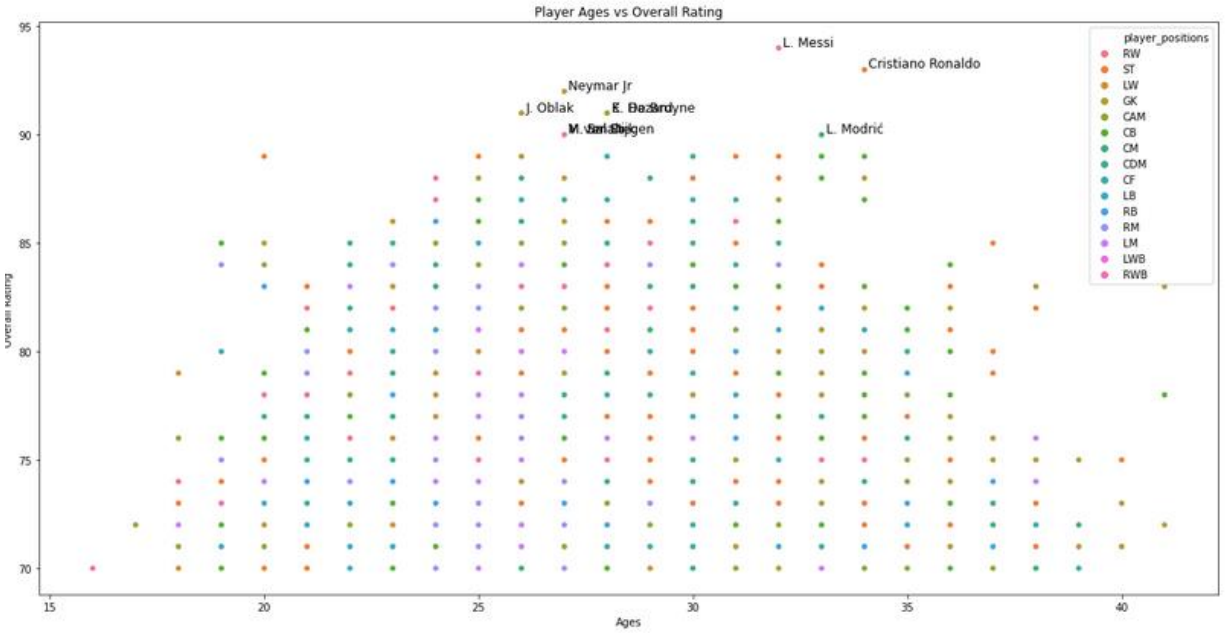

Figure 2

We see an obvious yet interesting insight here that a player reaches his maximum potential during the middle of his career, usually in between 25-32 years of age. Since we suspect the data to be highly correlated, we will look at a heatmap of the data.

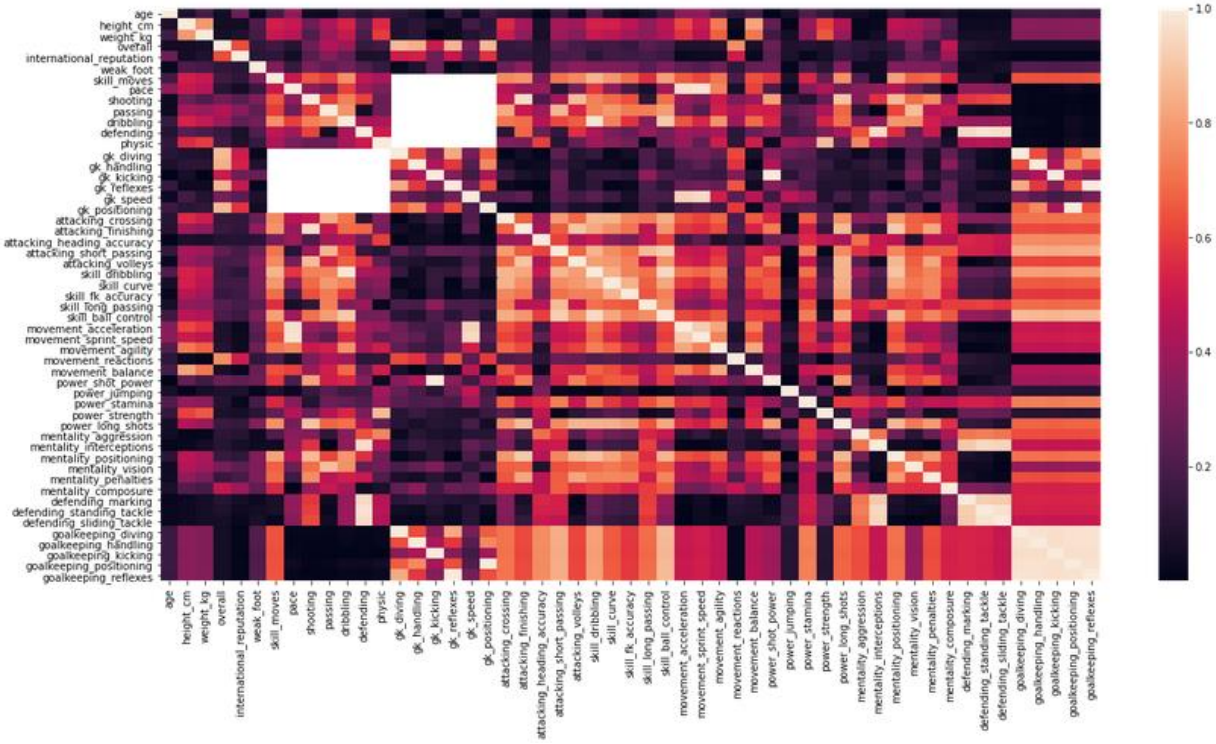

Figure 3

As we can see, goalkeeper related features are perfectly correlated as seen by the white squares in the data. Goalkeepers are a separate group and none of the main player skills apply to goalkeepers. We will assume this as a separate cluster and remove all goalkeepers from the dataset. Now we will have to also drop the rows with *player_position* with the value GK. That is, we will also drop all the goalkeepers from the dataset. PCA requires continuous features only and hence we will also drop all features that are categorical. The reason for this is that PCA looks to capture the maximum variance in the data in the principal components and categorical features are discrete in nature with zero variance.

3. Feature Relevance

Now we are left with over 39 variables and we still need to check if our initial assumptions that overall and other summary skills can be explained by the other variables. A simple way to check this is to run a regression model on these features are the response and all other features as the predictors. Let us build a Decision Tree model to check this. We will create a function to perform this regression. The function will run regression with some feature as response and all other features as predictors. The R2 scores for response greater than 0.95 is shown below. We will remove these features as the variance in these features can be explained by the remaining variables and they do not add a lot of further information to our analysis.

```
R2 Score for feature pace is 0.997
R2 Score for feature shooting is 0.984
R2 Score for feature passing is 0.957
R2 Score for feature dribbling is 0.982
R2 Score for feature defending is 0.99
R2 Score for feature physic is 0.956
R2 Score for feature attacking_finishing is 0.968
R2 Score for feature skill_dribbling is 0.966
R2 Score for feature movement_acceleration is 0.985
R2 Score for feature movement_sprint_speed is 0.99
R2 Score for feature defending_standing_tackle is 0.954
```

Figure 4

4. Principal Component Analysis

Since, for PCA we need scaled data, we will transform the dataset by trying various scaling techniques. The original distribution of the dataset can be seen in the figure below.

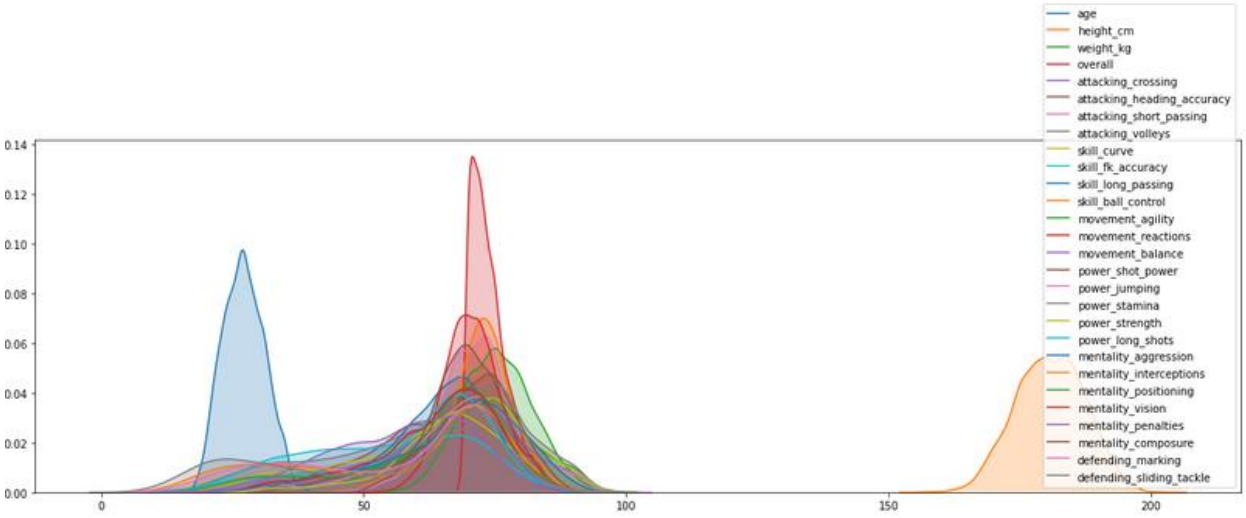

Figure 5

We can notice that the features are on different scales and most of the features are slightly left-skewed. We try the following scaling techniques on the data and chose the one that

results in the best explained variance by the first two principal components. The scaling techniques and variance explained by the first two principal components are summarized in the table below:

Scaling Technique	Variance explained by PC1 and PC2
Log Scaling	68.4%
Standard Scaling	52.5%
Min Max Scaling	57.2%
Log normal Scaling	50.8%

Table 1

The explained variances and the original features contributing to them are shown below.

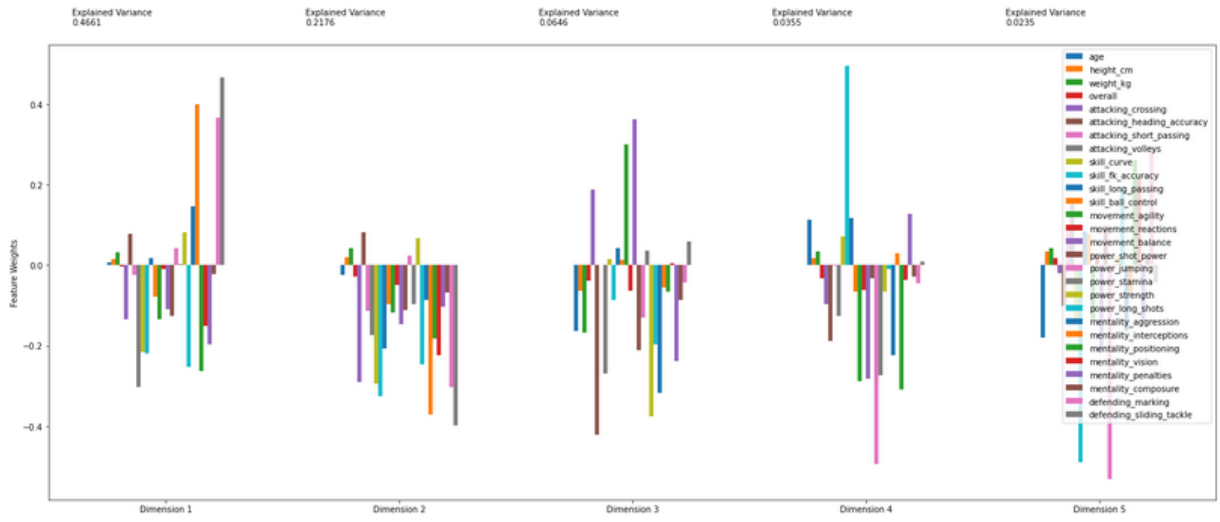

Figure 6

1. Visualizing original features on the principal components

When using principal component analysis, one of the main goals is to reduce the dimensionality of the data — in effect, reducing the complexity of the problem. However, dimensionality reduction comes at a cost as fewer dimensions used implies less of the

total variance in the data is being explained. Because of this, the *cumulative explained variance ratio* is extremely important for knowing how many dimensions are necessary for the problem. Additionally, if a significant amount of variance is explained by only two or three dimensions, the reduced data can be visualized afterwards.

Since PCs describe variation and account for the varied influences of the original characteristics, we can plot the PCs to find out which feature produces the differences among clusters. To do this we plot the loadings, or vectors representing each feature of the PC plot centered at (0, 0) with the direction and length of these vectors showing how much significance each feature has on the PCs. Also, the angle between these vectors let us know correlation between the features with a small angle denoting high correlation. A plot that visualizes the above information is called a Biplot. The Biplot for our dataset is provided below. As we can see, the features *mentality_interceptions*, *defending_sliding_tackle* and *defending_marking* is close together. Also, these strongly influence both PC1 and PC2.

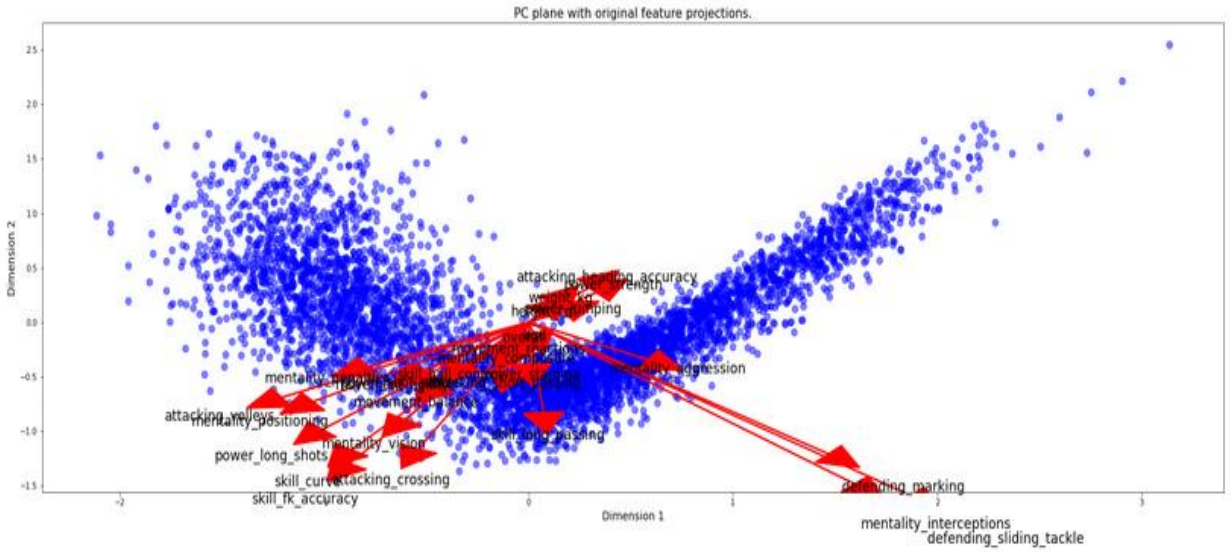

Figure 7

5. Clustering

In this section, we choose to use a K-Means clustering algorithm to identify the various player segments hidden in the data. Advantages of KMeans clustering algorithm are: Kmeans is very fast. This is because Kmeans only needs to fit data to cluster centers. This makes KMeans faster in training. However, one drawback is that KMeans only assigns hard clusters and does not give the probability score of the cluster. Based on the data, it seems KMeans would do a good job assuming that the players are well segmented, and each player assumes a special role.

Depending on the problem, the number of clusters in the data may not be known in advance. As a result, we do not know for sure if a certain number of clusters are the best choice for our data. Since we do not know the structure present in the data, in order to measure the "goodness" of our clustering, we calculate each point's *silhouette coefficient*. The silhouette coefficient for a data point measures how similar it is to its assigned cluster from -1 (dissimilar) to 1 (similar). Calculating the *mean* silhouette coefficient provides for a simple scoring method of a given clustering. The Silhouette Coefficient is defined for each sample and is composed of two scores (a and b):

a. The mean distance between a sample and all other points in the same class.
b. The mean distance between a sample and all other points in the next nearest cluster.

The Silhouette Coefficient *s* for a single sample is then given as:

$$s = \frac{b - a}{max(a, b)}$$

The Silhouette Coefficient for a set of samples is given as the mean of the Silhouette Coefficient for each sample. In our analysis, we receive the highest Silhouette Score of about 0.53 for three clusters. Another popular method to guess the appropriate number of clusters is the Elbow Method. In this method, we choose that value of K, which lies at the elbow of the curve plotted between the number of clusters and sum of distances between each point and its centroid. As we can see from the image below, the elbow of the curve appears at 3 clusters thus concurring with the Silhouette score.

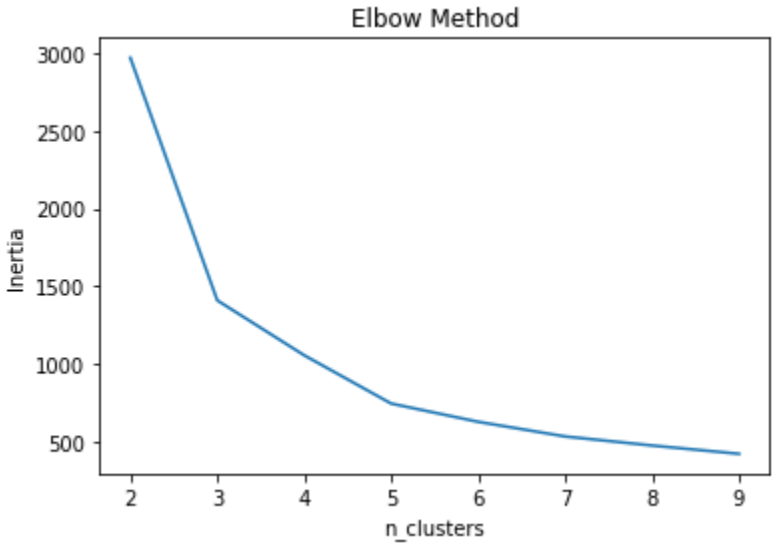

Figure 8

6. Visualizing Optimum number of clusters

Since we have 3 clusters, we plot the three clusters along with their cluster centers. In addition, since this is football data, it is interesting to visualize the different sample players from different playing positions on the clusters. The map of playing positions in the game is shown in Figure 9 below.

Figure 9

1. Midfielders:

   Midfielders are players that play in the center of the pitch and often perform a role of controlling the game. Let us look at a random sample of 10 midfielders in our dataset plotted on the cluster.

   As we can see, 7 out of the 10 defenders are plated in cluster 2 or the pink cluster.

Figure 10

If we go back to figure 6 and 7 to look at the PCs and the loadings affecting cluster 2 we notice that the most important features in along cluster 2 are *skill_long_passing, short_passing, power_stamina, mental_composure* etc. Thus, we can infer that midfielders are the players that possess these traits and we can identify the strongest players in this cluster for a midfielder role.

2. Forwards

The most important role of a forward player in football is to score goals. Forwards are made up of wingers (*RW, LW*), strikers (*ST*) and center forwards (*CF*) with the most common goal scorers being the *ST* and *CF*. We again plot 10 random players in *ST* and *CF* role in the below figure.

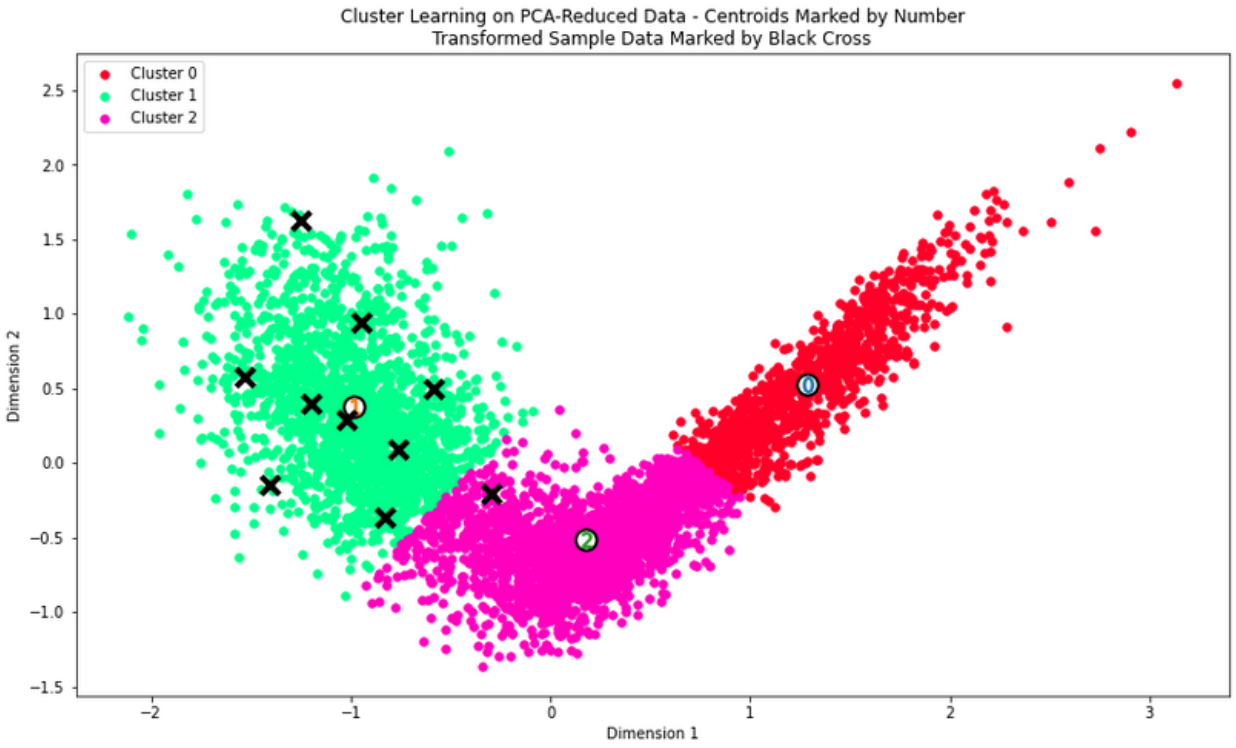

Figure 11

We notice that the forwards are spread across the green cluster or cluster 1. The most prominent features that influence cluster 1 can again be seen in Figure 6 and figure 7. Some of the key ones are *attacking, volleys, mentality_positioning, attaacking_header_accuracy*.

3. Defenders.

   The main role of defenders is to prevent the opposite team from scoring a goal. When random points are plotted in the dataset, as expected, we see most of them in the red cluster or cluster 0. Some of the key features in this cluster are *defending_marking, mentality_interceptions, defending_sliding_tackle* etc

Conclusions

By studying the structure of the data in the FIFA 20 dataset, we can reach some key insights about the game and the player base. We understand that the key skills of players in different positions differ by the role that they are performing. We develop a method to systematically reduce the dimensionality of the data by combining our domain knowledge, plotting correlations variables and using the predictive power of the variables to eliminate features. After performing principal component analysis, we can capture about 70% of the variability of the data in the first two PCs. By performing Kmeans on the transformed data, we identify 3 clusters in our dataset. The clustering helps us understand that the cluster of forwards and midfielders are connected as the positions in these two roles have a fair bit of overlap. Similarly, midfielders and defenders have a connected cluster. There are also some issues with our clusters as we do not see a clean separation in our data. However, with only 70% variability we are nonetheless able to reach some great insights into how the original data points map to the reduced data.

Bibliography

1. FIFA 20 Information - Wikipedia." https://en.wikipedia.org/wiki/FIFA_20.
2. FIFA 20 complete player dataset | Kaggle." 26 Sep. 2019, https://www.kaggle.com/stefanoleone992/fifa-20-complete-player-dataset
3. Sci-kit learn – Dimensionality Reduction - https://scikit-learn.org/stable/modules/decomposition.html
4. Silhouette Score — scikit-learn 0.22.2 ...." http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html.
5. Elbow method (clustering) - Wikipedia." https://en.wikipedia.org/wiki/Elbow_method_(clustering).
6. Udacity – Clustering visualization https://github.com/udacity/mlnd
7. How to read PCA biplots and scree plots – Linh Ngo https://blog.bioturing.com/2018/06/18/how-to-read-pca-biplots-and-scree-plots/
8. Sofifa.com – Data Scraped to Kaggle from sofifa - sofifa.com