Lens Desrouleaux

Computer Science 320

Lu

6 May 2020

Using Kmeans Clustering to Analyze FIFA Cards

Abstract:

I researched machine learning algorithms and their implementations. By learning Python and using the libraries it contains I implemented a unsupervised clustering algorithm, Kmeans clustering, to search for hidden value amongst different FIFA players. By analyzing the clusters formed I would be able to gain insights into players and their attributes and underlying play. After analyzing the clusters; and using another machine learning algorithm principal component analysis to reducing dimensionality, I was able to gain insights into player’s underlying statistics that drive their playstyle. I also gained information on how to best utilize those players to have them at their most effective position.

*Introduction*

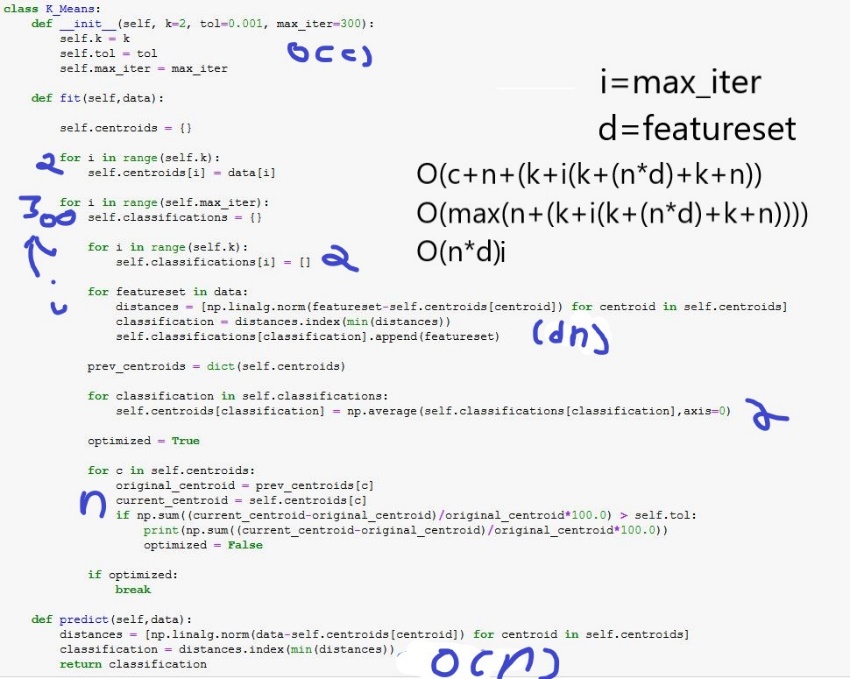
Clustering and grouping data are very important. Through clustering algorithms, you can advertise more efficiently, better identify and meet the needs of your consumers, direct aid to the people most in need, and more. As humans we are dispositioned to belong in groups and have group identity. Clustering allows you to identify those groups on a massive scale. A business that get thousands of shoppers a day or hour could never effectively advertise to their wide base without grouping. Clustering can help provide better insight into sports and athletes, by showing which athletes are statistically similar. The problem in sports games is that while overall ratings may be similar some players are not because of underlying statistics.

Clustering algorithms are very refined and increasingly complex. With the rise of big data computer scientists could collect substantial amounts of data to test their algorithms and refine like never before. Amazon, Walmart and Target depend on algorithms like these to use effective advertisements and sell relevant products to people. They also use them to group shoppers and their habits together, as well as detect spam. Their utility is vast and can be implemented in a wide array of tasks. I wanted to implement these algorithms in order to make my life easier and to learn new skills and get familiar with a new language.

FIFA and other sporting games have far too many statistics without explanation. What this leads to is players who may have similar top layer stats (like pace overall etc.) stats playing vastly different because the more underlying statistics are completely different. You could compare players by hand but that would take far too much time you would also have to compare the fluctuating prices of these players. A big problem with this is there are players with underlying statistics that are a great value because their overall is low. A lot of those players play like a cheaper version of a top star. But because of the sheer number of players it is far too hard to comb through manually. There needed to be a better way to compare these players to find the ones with better value.

*Method:*

Machine learning was the natural answer to this problem. The ability for a computer to process that sheer amount of information in such a short time was necessary. After drawing up the problem on paper I began researching machine learning algorithms that would be good for this problem. Clustering was the obvious answer. Clustering algorithms are used in a wide breadth of applications, from targeting advertising and couponing to allowing for easier and faster data analysis. K means clustering seeks to create subgroups based on centroids. It is an unsupervised non parametric algorithm. Computation time depends on the number of datapoints and outside of setting initial parameters the programmer does not have much to do compared to something like linear regression.

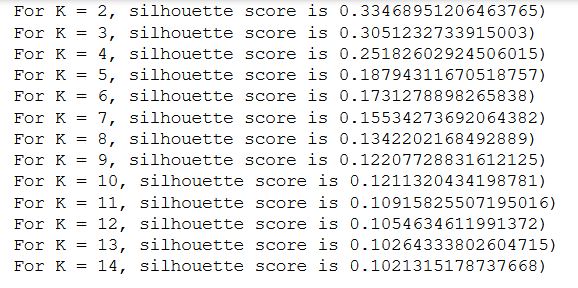
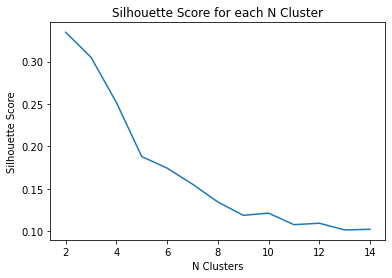
 The time complexity of kmeans clustering is hard to determine. Since there are variations of kmeans and different implementations. So, I wrote the code for the kmeans implementation I used. After manually tracing the code I determined that it had a time complexity of O(i(nd)). Initialization takes a constant time. The predict function loop takes n time for each time it is called. Where the largest time comes from is the fit function. It has a for loop that will run k times then a iteration loop that will run I times. This iteration loop has 4 nested for loops and one if statement. The first for loop will run k times the second will run d times, but has another for loop nested that will run n times causing it to have a time complexity of n\*d. The third for loop will run K times. The fourth for lop will run n times and determine if it is optimized, the final if will break if it is optimized. After adding all these times together, the loop with n\*d time dominates by big-o’s addition theorem.

Research was a large part of this project. I wanted to work with Python because of its libraries in mathematics, machine learning and finance. For machine learning it is far superior than java or C++ while being easier to learn due to the some of the overlap with C++. Before gathering the data, I had to research implement and test the algorithm. I spent a lot of time familiarizing myself with Python and using Jupyter notebooks. I learned about the libraries and what I’d have to do before my presentation. I learned the syntax of Python and the libraries I would be using.

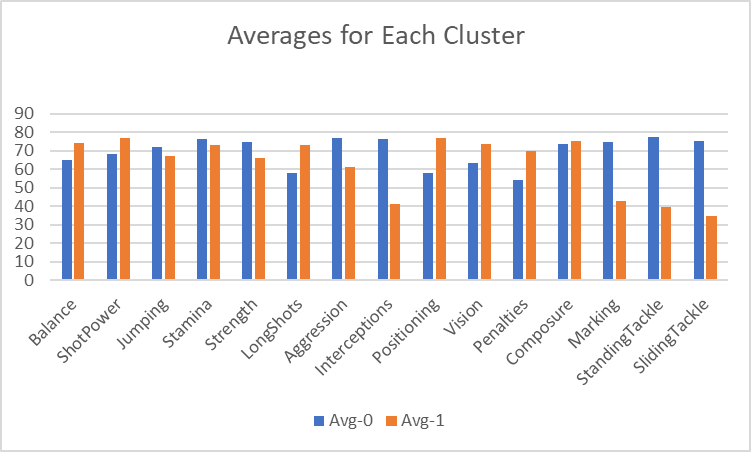
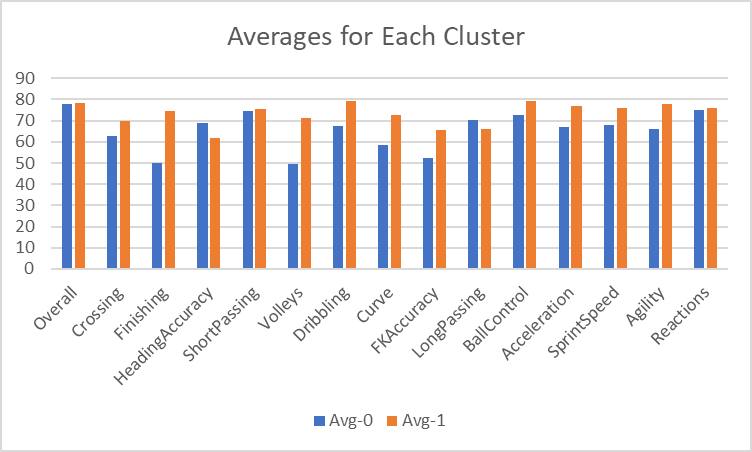
After this a lot of time was spent searching arranging and scrubbing data to have a good set of players to analyze and run through the algorithm. I got datasets for every player in FIFA 20 and whittled that down to only players having above a 75-overall rating. While there are a lot of good players and value to be had from players below that rating, they are just too few and far between to include in this dataset. I also wanted to exclude goalies because while all players have stats in every category, including the goalie specific ones, goalkeepers have a wider variance and stranger statistics. This is where I first used the Kmeans cluster Algorithm as a separation method to get rid of outliers and goalies in the data.

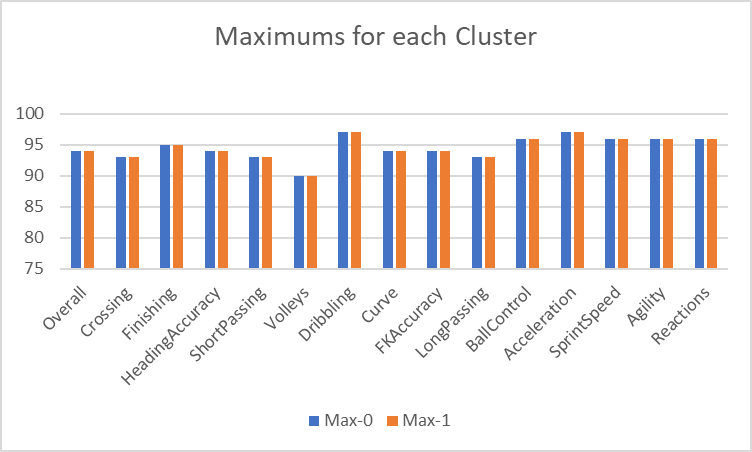
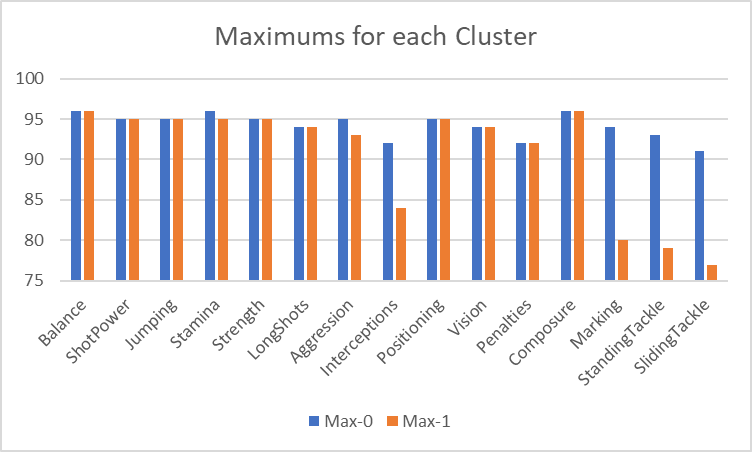
*Discussion and Results*

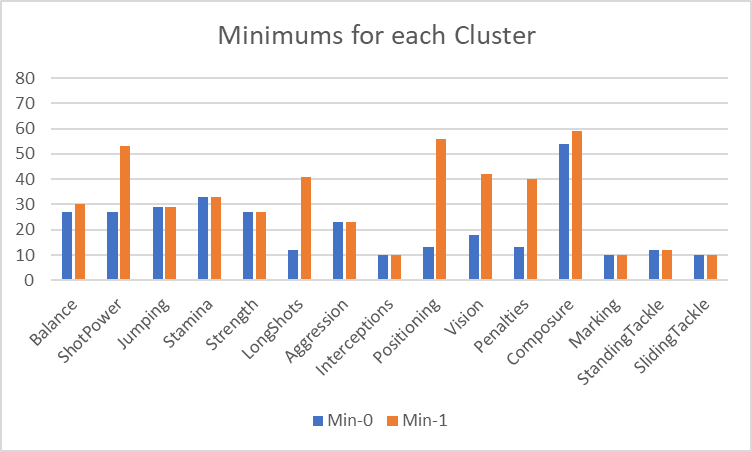
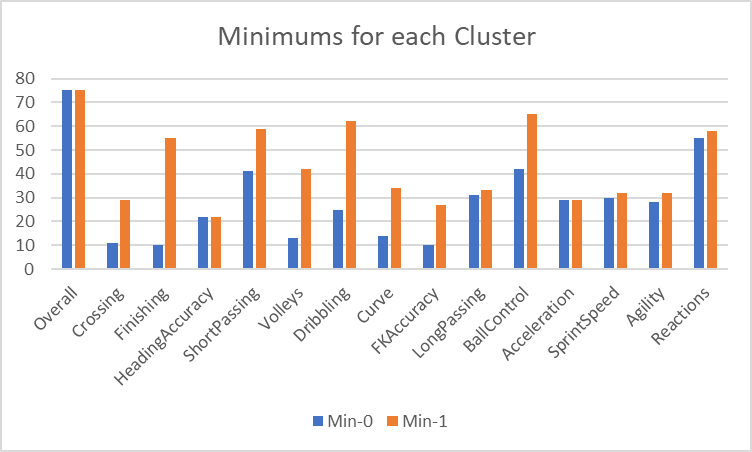
After writing the data to a new file I could run the algorithm on the new filtered data in order to gather insights. In order to visualize my results into a graph I had to use another machine learning algorithm called principal component analysis. This algorithm works by analyzing trends in data in order to compress dimensionality while maintaining information. Since so many different attributes were grouped together to be analyzed the dimensionality of the data would have been to high without using principal component analysis. After running the algorithm, I was able to analyze the data. To determine K, I used silhouette score and found that it maximizes at k=2.











Players in Cluster 0 are better at man-marking, interceptions and tackling. They are far better at defensive aspects of the game. They are also stronger and have better stamina. Players in Cluster 1 excel at the more skilled aspects of the. There higher shooting, passing and freekick ability allows them to shine in the offensive parts of the game. Their superior acceleration and sprint-speed makes them faster over a sprint distance.

The clusters did provide better insights into specific players; however, to specifically find value in lower rated cards throughout the game would require a different algorithm than K means clustering. Well it does help into grouping players As I build on this project over the summer, I will test affinity propagation and other algorithms to help me better find hidden value players.

Appendix

In a Python environment install Jupyter after installing call “jupyter notebook fifacards.ipynb” in terminal and click on the forwarding link provided by the terminal. You can then run the code in Jupyter by the blocks. Make sure the files “scrubbedfifa.csv” and “predata.csv” are in the folder shared with the jupyter notebook file.

References

Arthur, David, and Sergei Vassilvitskii. How Slow is the K-Means Method?, Stanford. <https://theory.stanford.edu/~sergei/papers/kMeans-socg.pdf>

Dagli, Rishit. "Build K-Means from Scratch in Python." Medium, Oct 29, 2019. <https://medium.com/@rishit.dagli/build-k-means-from-scratch-in-python-e46bf68aa875>

Galarnyk, Michael. "PCA using Python (Scikit-Learn)." Towards Data Science, Dec 7, 2017.

<https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60>

Halthor, Ajay. K-Means Clustering - EXPLAINED!. 2018, <https://www.youtube.com/watch?v=O6b2L_lYH9k>

Leone, Stefano. "FIFA 20 Complete Player Dataset.", Sep 26, 2019, <https://www.kaggle.com/stefanoleone992/fifa-20-complete-player-dataset#players_20.csv>.

The base file for all the data used in the analysis.

Moffitt, Chris. "Using WSL to Build a Python Development Environment on Windows " Practical Business Python,

<https://pbpython.com/wsl-python.html>

"Python Tutorial for Programmers - Python Crash Course." , directed by Mosh Hamedi. , produced by Mosh Hamedi. , performance by Mosh Hamedi. , Programming with Mosh, 2018.

<https://www.youtube.com/watch?v=f79MRyMsjrQ>

"Sklearn.Cluster.KMeans — Scikit-Learn 0.22.2 Documentation.", 2020,

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>.