Data Visualization for FIFA 20 Players Data

Venkata Ravi Teja Takkella
113219890
venkataravite.takkella@stonybrook.edu

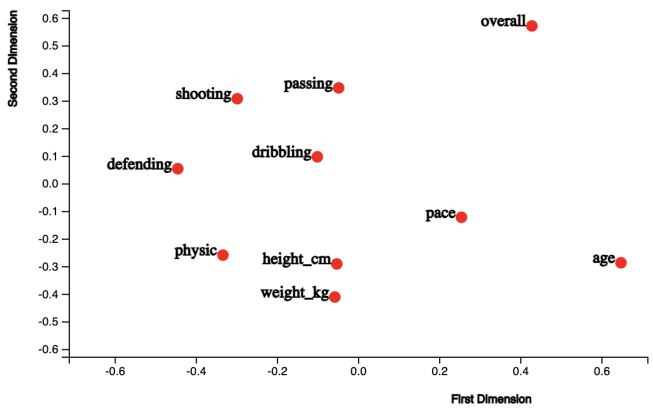
Video Link : https://youtu.be/6y763-3T5b8

As mentioned in the assignment, we were asked to do the data processing in python and visualise the data through d3. I've used Flask for python server and built the front end using d3.

I've used the same data taken for Lab 1, i,e **FIFA 20 Players Data** and proceeded for this assignment. Out of this dataset I've taken 10 numerical data features to go ahead. The numerical data is in the fifa_processed_numerical.csv whereas the entire dataset used for Lab 2 is in fifa_processed_second.csv.

Observations:

1) Based on the Scree plot under multiple dimensions, I can conclude that the features "pace" and "shooting" are most common among the top 4. These of course provide a huge impact on the rating of a player. If you can see, there are a lot of strikers in the top 500 compared to defenders or midfielders.
2) And also, from the 6th dimension; we can see the top 4 features are almost similar and just differ in ordering. "Pace", " physic", " defending", " shooting" can be taken as the major contribution information of the dataset.
3) Even the scatter matrix shows the same, these four features provide distinct clusters on the matrix plot compared to others.
4) The data is well clustered and the Euclidean distance plot also provides similar information when plotted onto 2 dimensions.



5)

From the above correlation plot, we can show the dissimilar features. For example: shooting and defending are not much correlated and hence the plot shows the same.

Visualisation Learnings:

1) The Bi-plot provides a bit less information compared to others as most of the eigenvectors don't pass through the centre of the plotted data points to show that it can be taken as the highest feature.
2) Whereas the PCP plot also shows a lot of information without much generalisation of information. We should be able to predict or come up with a summary of values of the features, which isn't happening with the data picked.
3) The PCP plot also looks so clustered if there are a lot of values of categorical features.
4) The precomputed MDS plot looks much informative providing the relationship between the features.

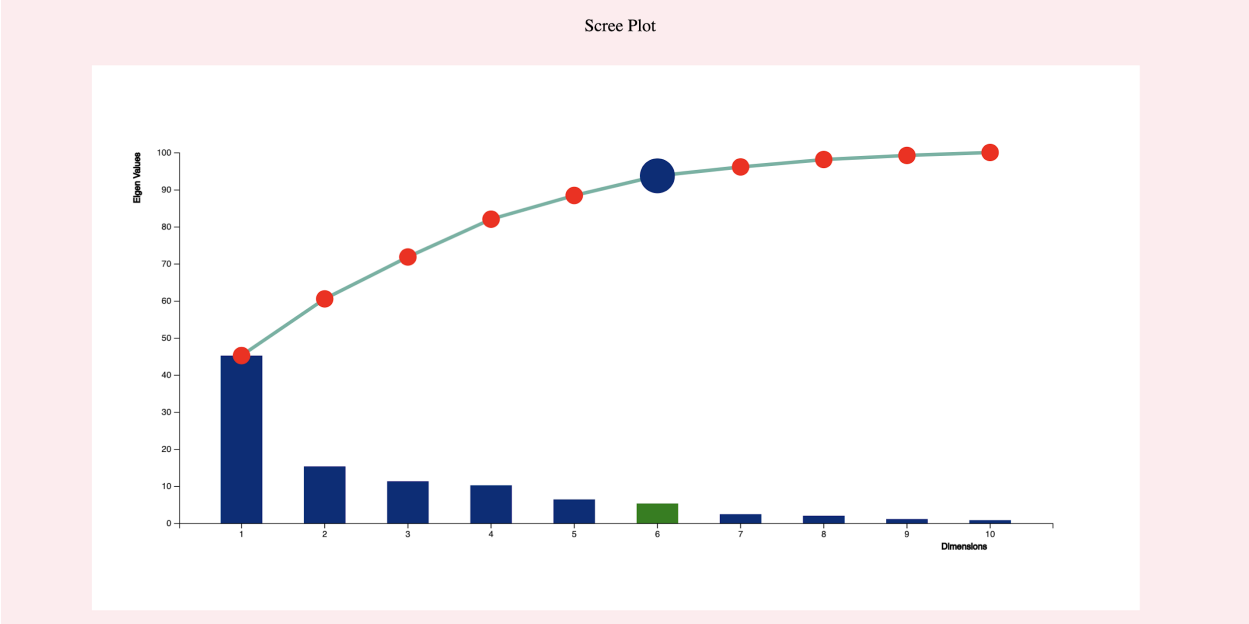
Interesting Features:

1) The entire code is generic and works all fine for any dataset by just categorizing the features into categorical and numerical.
2) Instead of various online templates, every plot is built using base knowledge of scatter plot/bar chart done in Lab 1.

Task Explanations:

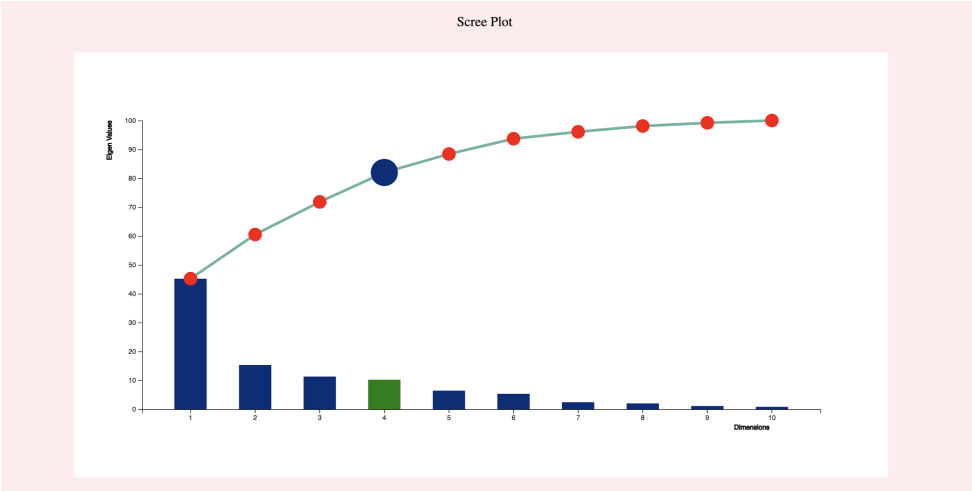
Task 1

a) Coming to the requirements, we were asked to build a scree plot using all the dimensions and project the eigenvalues.

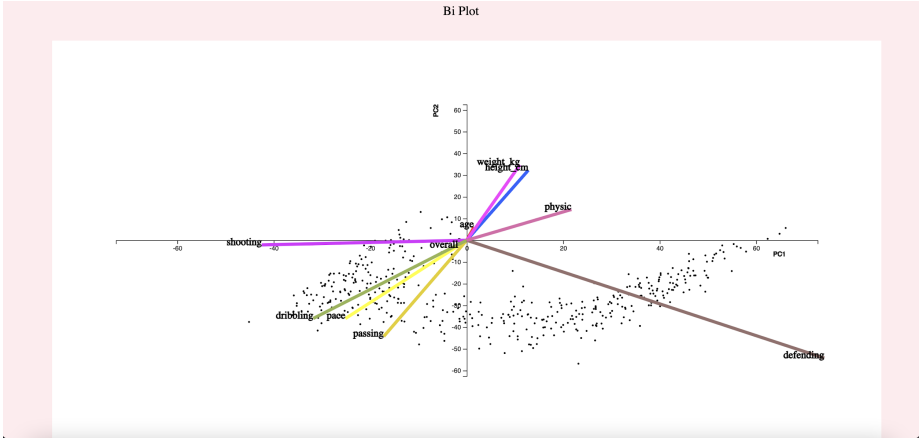


The bar chart describes the individual eigenvalues, whereas the path shows the cumulative sum and it converges to 1 as we reach the last dimension. I've used the **sklearn PCA** libraries for finding the eigenvalues and eigenvectors.

b) I've added the interaction element over the circles and the bars, such that the selected dimension (d_i) gets highlighted by projecting the next tasks below. This colour gets changed to the selected dimension on the go. An example is shown below.



c) Next we were asked to plot the data onto a PCA based biplot. I've used 2 components for the PCA fitting and generated the eigenvectors along with the transformed data and finally showed them together onto a biplot.

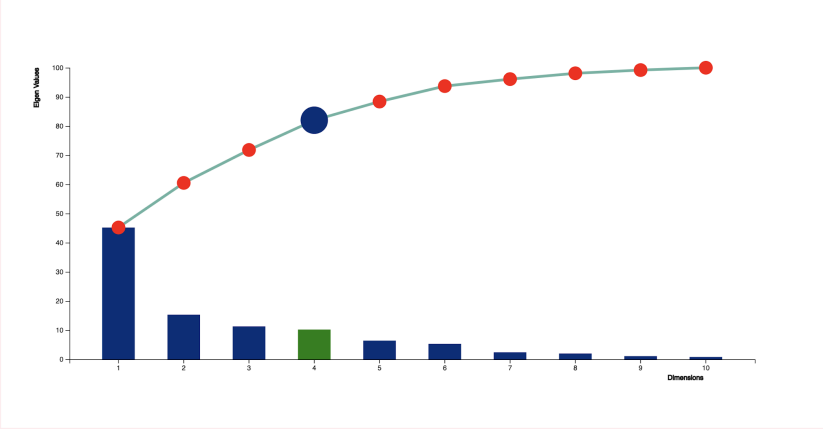


Since I've used 10 features, There are 10 eigenvectors being shown above.

Task 2

a) Using the d_i selected from the scree plot, we were asked to fit the PCA with d_i components and generate the PCA loadings of all the dimensions and pick the top 4. We were asked to print the above data into a tabular form too.

Scree Plot



So when the d_i is selected as the 4rth dimension, we get four PCs and hence the below table displays the same too.

PCA Loadings

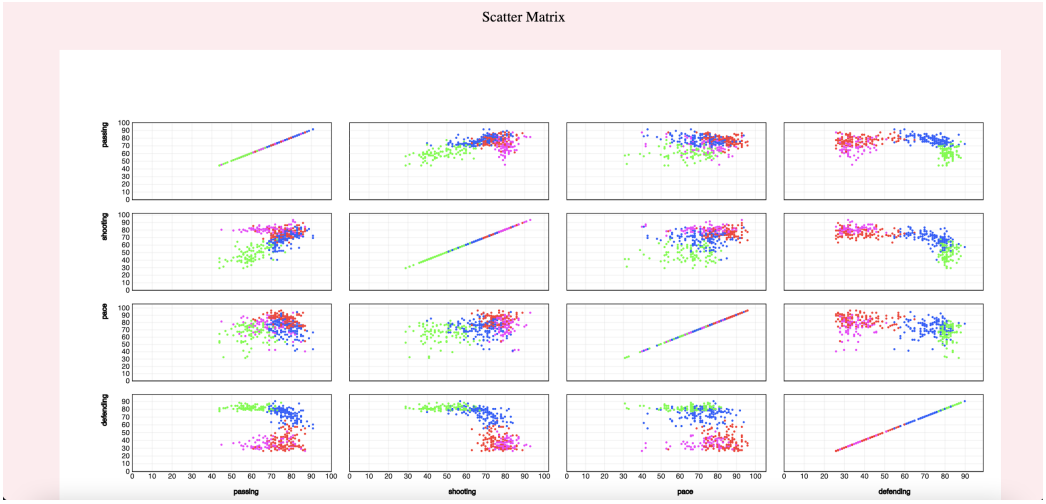
Attribute	PC1	PC2	PC3	PC4
passing	0.7412244455942064	-0.542591254911941	0.17285636985464417	-0.13776846102313767
shooting	-0.251626525611673	-0.36059240892468053	-0.571815832126096	-0.5998722562931701
pace	-0.4247986614454592	-0.02139906991410 16	0.5474847663542488	-0.34387156492068616
defending	-0.17205852524276644	-0.4428796778647749	0.44840077005941675	0.16284130691793927

Similarly if we select d_i = 8, the table looks like.

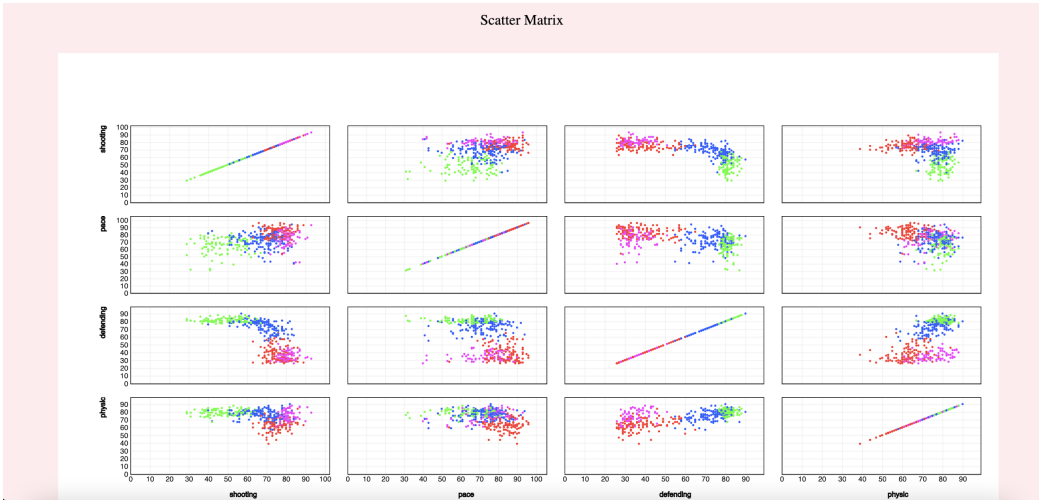
PCA Loadings

Attribute	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
shooting	0.21698008499644633	0.14165017788335477	0.1973029182413381	-0.5803254386269061	0.1417640723127412	0.7030354778547644	-0.02822181875460738	-0.20425906861997564
pace	0.7412244455942064	-0.5425912549119941	0.17285636985464417	-0.13776846102313767	0.11482288093229959	-0.26536423941995124	-0.027183411959611 52	0.14276586471496316
defending	-0.251626525611673	-0.36059240892468053	-0.571815832126096	-0.5998722562931701	-0.13298573737732272	-0.20026408010653346	0.17237647083057792	-0.17062490300342972
physic	-0.4247986614454592	-0.02139906991410 16	0.5474847663542488	-0.34387156492068616	0.4737112267878844	-0.3894789007942778	-0.116305753247070 18	0.038087438375506204

b) Using these four attributes, we were asked to generate the scatter matrix plot.
So when d_i = 4, the scalar matrix looks like:



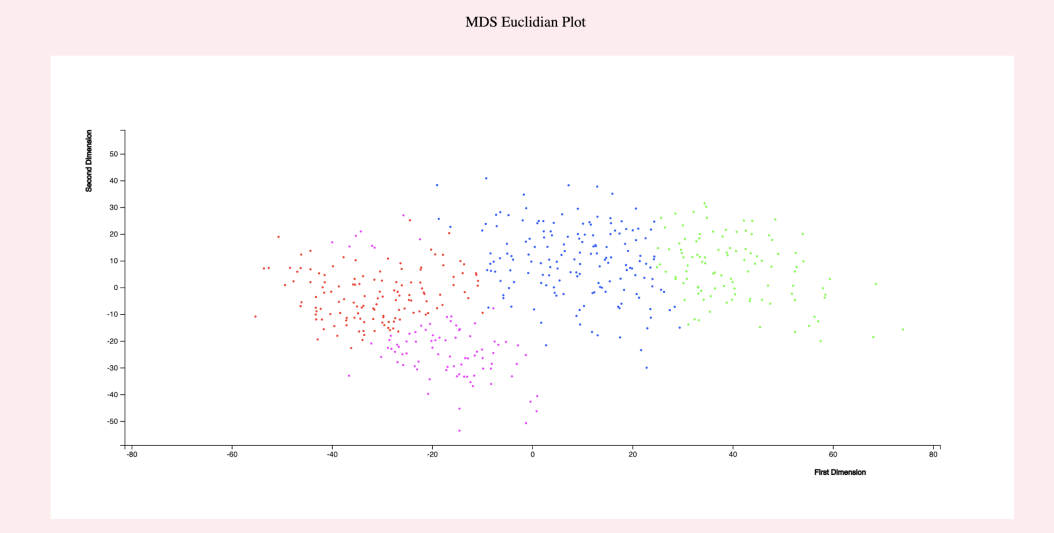
Where as when d_i = 8, it looks like



c) We were asked to use K-means clustering to colour our data points too, I've used the sklearn library of K-means to cluster the initial data. After checking for a few values of clusters, I preferred to pick 4. The above mentioned scatter matrix plots show the same for proof.

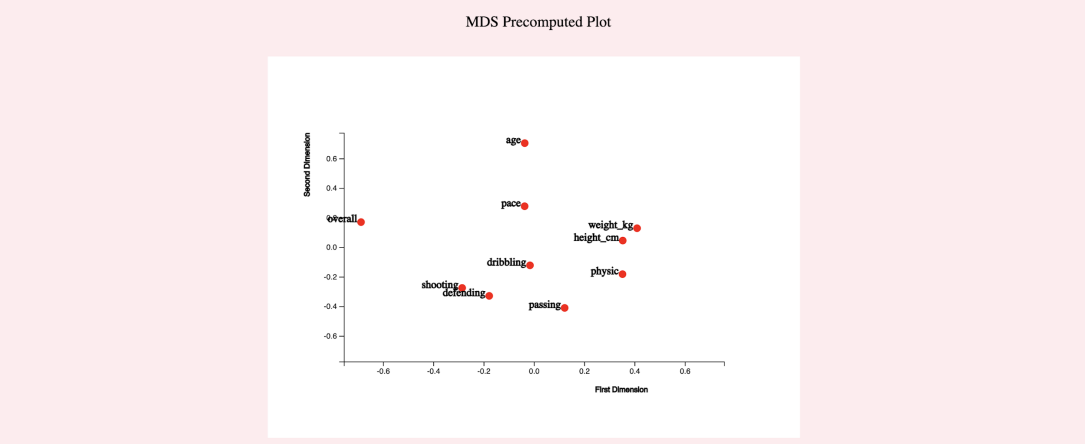
Task 3

a) We were asked to plot MDS using Euclidean distance and scatter the data points in 2 dimensions.



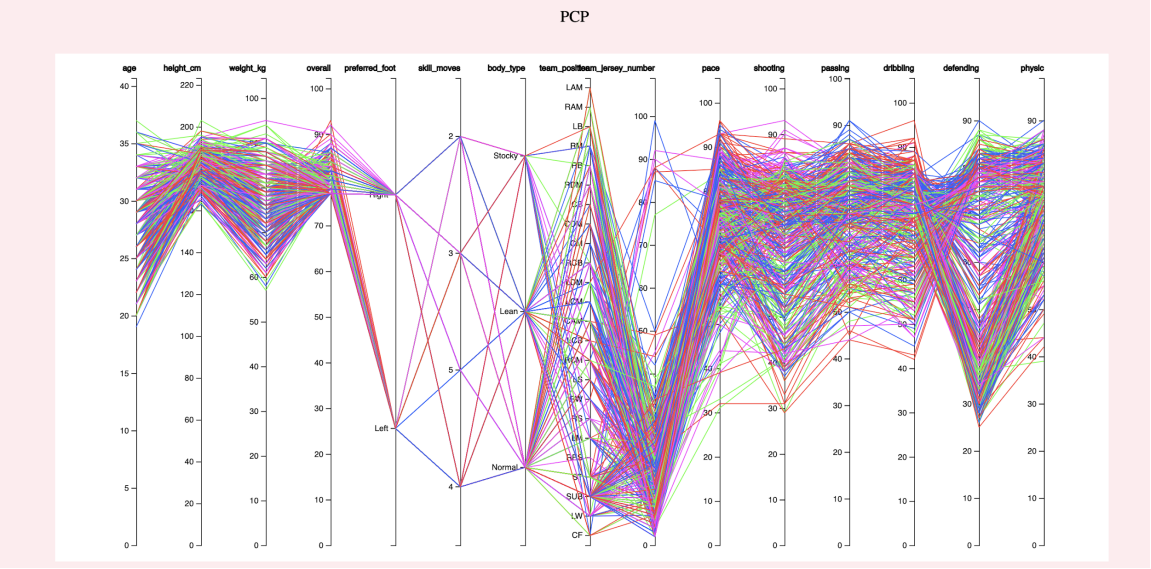
b) Using K Means clusters, the coloring of each data point is taken based on the original data.

c) Similarly I've used 1- |correlation| distance to plot the variables in 2 dimensions.

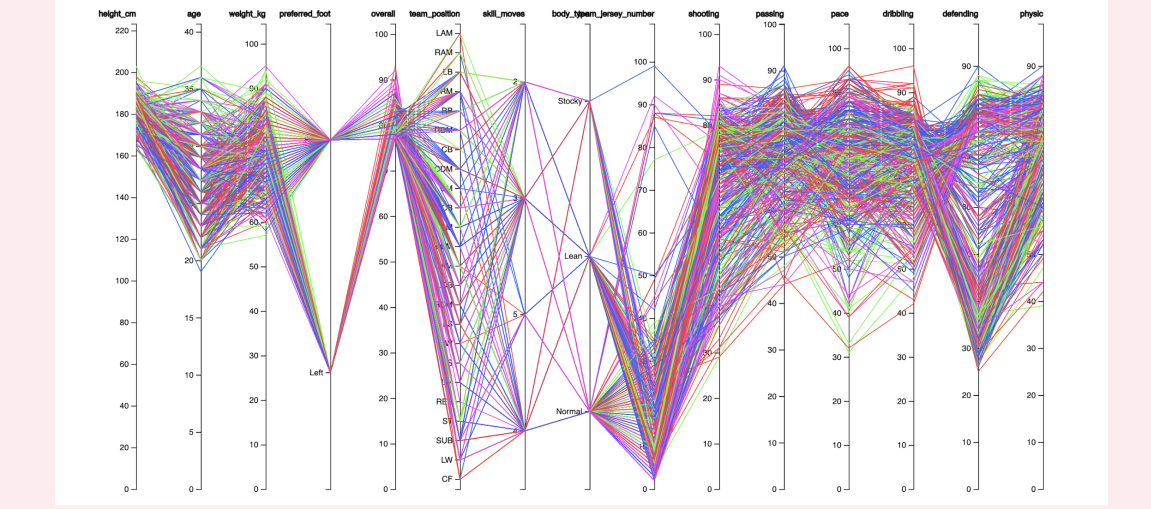


Task 4

a) To show all the available features as parallel coordinates and visualise the data. The below shown plot contains both categorical and numerical features.



b) Using d3 drag features, I've added user interaction so that the feature scales can be moved horizontally to get a more meaningful experience.

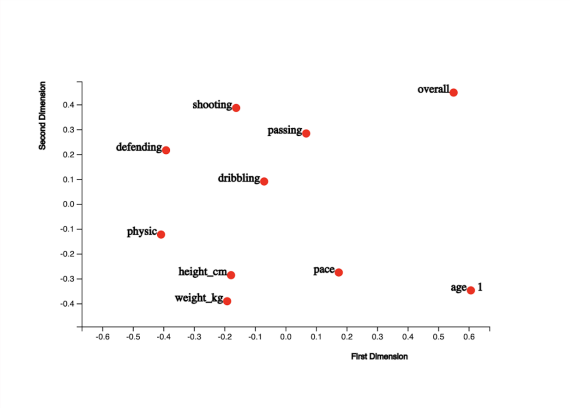


c) The above polylines have been colored based on the cluster of K Means again.

Extra Credit:

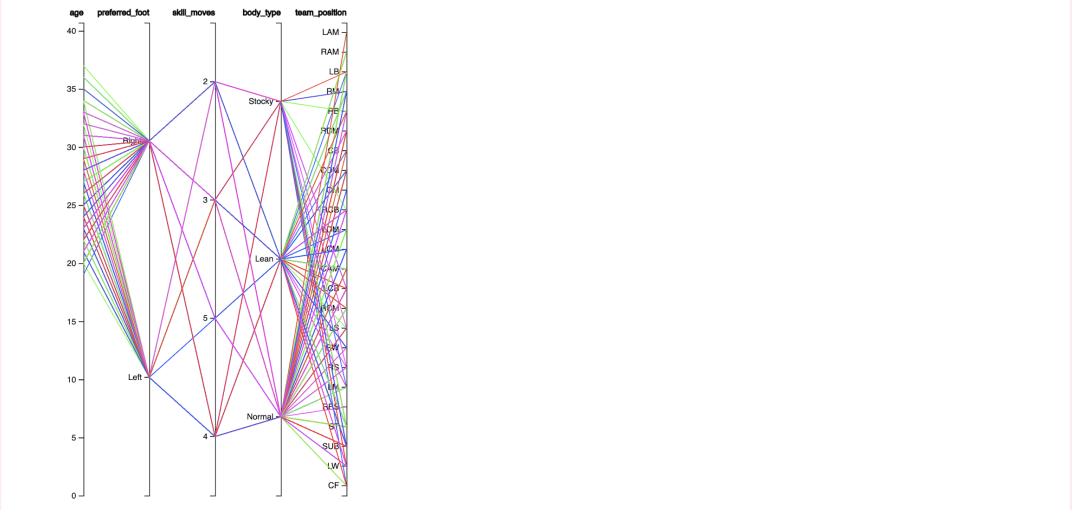
a) Based on the precomputed MDS plot, I've given the user interaction to select the ordering of the numerical features and plot the PCP plot based on that.

MDS Precomputed Plot



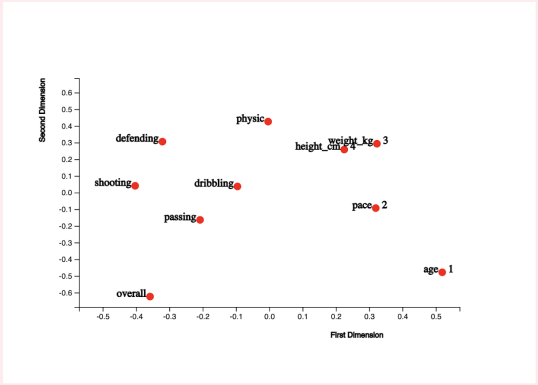
Above here age is selected first, similarly the PCP plot looks like

PCP



And if more features are selected on the Precomputed MDS plot, the PCP plot looks like

MDS Precomputed Plot



PCP