A picture containing text, sky, outdoor, person

Description automatically generated

FIFA 22 PLAYERS

DS 630 MACHINE LEARNING FINAL PROJECT

Mir Ahmed

# Overview

This project performed a deeper dive into the FIFA 22 players' dataset that has variables that attribute to a players preferred playing position, growth, overall rating and values.

The dataset was downloaded from: <https://www.kaggle.com/cashncarry/fifa-22-complete-player-dataset>

The dataset has 89 attributes and 19260 rows (One row for each FIFA registered player). So, we can consider this dataset as the whole universe of data for this domain.

The following questions were answered through this project:

1. What is the best categorization model to predict best position for each player?
2. What is best regression model for predicting growth of a player?

First, we preformed feature selection using f\_classif method and identify outliers using DBSCAN.

Then, we created K-means clusters using our selected features.

For categorization, we tried exploring prediction/ categorization models such as decision tree classifier, random forest classifier and neural network.

For regression, we explored linear regression, ridge expression, random forest regressor and SVR models.

# Dataset

* We used pandas package to import the csv file.
* The dataset contains attributes related to different soccer related abilities as well as some other features, i.e., value, wage, nationality, club name, playing position in club team, national team name, playing position in national team, jersey number in national team and club, release clause for clubs, contract info with clubs etc.
* The scores to the soccer related abilities are given out by FIFA ratings division on a scale of 0 to 100. TotalStats is sum of scores from all soccer abilities related attributes and BaseStats is sum of scores from position related attributes.
* We can see the columns in the screenshot below.

Text

Description automatically generated

# Transformation

* In this step, we identified the object categorical variables and converted them to numerical (integer) categorical variables.
* We can see on the screenshot to the left how each position was converted to an integer value
* Then we dropped the unnecessary columns, such as Name, FullName, PhotoUrl, Nationality, Positions, Club, ReleaseClause, ClubPosition, ContractUntil, ClubNumber, ClubJoined, OnLoad, NationalTeam, NationalPosition, NationalNumber, PreferredFoot, AttackingWorkRate, DefensiveWorkRate

Graphical user interface, text, application

Description automatically generated

# Exploratory Data Analysis

## Players’ frequency by age:

Chart, bar chart

Description automatically generated

* Most of the players are from 20-29 of age bucket.
* Age 21 has the highest frequency.

## Players’ frequency by growth:

Chart, histogram

Description automatically generated

Chart, bar chart

Description automatically generated

* Usually, growth is calculated by the difference in potential after 3 years and current rating.
* Even though most amount have 0 growth, we do see a good number of players with exceptional growth (>10). The future of soccer looks good!

## Players’ growth by age:

Chart, bar chart

Description automatically generated

* Age and growth have a strong negative correlation (-0.86421).
* That is why we see growth declining with increase in age.
* A player has the most growth when he is young which is why buying young players or developing youths is very beneficial as a manager.

## Players’ growth and frequency by position:

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

* We have a lot of CBs (Centre Back) in our dataset.
* GK (Goalkeeper) has the highest growth, but CAM (Center attacking midfielder) has highest average growth.

## Players’ wage and value by position:

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

* CF (Center Forward) & LW (Left Winger) tend to have higher wages as well
* CF (Center Forward) & LW (Left Winger) tend to have higher values (Euro in millions) in the transfer market.

# Problem Statements

## Does position have an impact on the wage and the valuation of a player?

|  |  |  |
| --- | --- | --- |
| **Key1** | **Key2** | **correlation** |
| BestPosition\_code | ValueEUR | 0.018958 |
| BestPosition\_code | WageEUR | 0.008704 |

So, position has little to no impact on wage or valuation of a player.

## Does growth have an impact on the wage and the valuation of a player?

|  |  |  |
| --- | --- | --- |
| **Key1** | **Key2** | **correlation** |
| Growth | WageEUR | 0.194634 |
| Growth | ValueEUR | 0.104524 |

So, Growth might have a very little impact on wage or valuation of a player. But it is strongly & negatively correlated with Age.

# Feature Selection

Graphical user interface, text, application, email

Description automatically generated

* After dropping the unnecessary columns, we were left with 73 attributes.
* We used f\_classif method which computes the ANOVA F-value for the provided sample and drop the variables those have very low variances or too similar.
* Thus, we were able to get 30 attributes.
* Still, we saw a couple of attributes which have nearly the same definitions as other attributes.
* So, we dropped the following variables manually: 'GKDiving’, 'GKReflexes’, 'GKHandling’, 'GKKicking’, 'GKPositioning’
* We added ‘BestPosition\_code’ as it will be the response variable.
* Finally, we got a dataset with 26 attributes.

# Correlation Heatmap

A picture containing text, electronics

Description automatically generated

As 'BestPosition\_code' is a categorical variable, we do not see any strong correlations with other ratio variables.

# Outlier Detection - DBSCAN

Chart

Description automatically generated

* We must provide a value for epsilon, a parameter for DBSCAN, which defines the maximum distance between two points.
* we find a suitable value for epsilon by calculating the distance to the nearest n points for each point, sorting and plotting the results. Then we look to see where the change is most pronounced and select that as epsilon.
* We can calculate the distance from each point to its closest neighbor using the NearestNeighbors. The point itself is included in n\_neighbors. The kneighbors method returns two arrays, one which contains the distance to the closest n\_neighbors points and the other which contains the index for each of those points.
* The optimal value for epsilon will be found at the point of maximum curvature.

Chart, line chart

Description automatically generated

We have used the knee method to locate the optimum value of eps (maximum curvature for nearest neighbor method).

According to the graph on the left, value of eps should be 1.4080565595824033.

Chart, scatter chart

Description automatically generated

* We have used the eps value created through the nearest neighbor method for DBSCAN.
* We have used 27 as our minimum sample as the rule of thumb is to add 1 to the total number of columns for this parameter.
* According to DBSCAN method, we have 19 outliers in the dataset. We will leave them for now as they are not that drastic.
* So, DBSCAN basically identified the 16 best players at present, 1 youth player and 2 not so good players for their age as outliers. Below is the count distribution for DBSCAN output.

|  |  |
| --- | --- |
| DBSCAN output | Count |
| 1 | 17118 |
| 0 | 2123 |
| -1 | 19 |

# K-Means Clustering

Chart, line chart

Description automatically generated

* We have used the elbow method to get the optimal value of K
* We see the last elbow bend at n = 4 at the graph above.
* Therefore, we will tell python to create 4 clusters in the next phase.

Chart, scatter chart

Description automatically generated

We see the clusters properly distributed around their respective centroids. There are some overlaps between cluster 2,3 & 4.

We had dived deeper into each cluster in next section to have better insight.

## K-Means Clustering – Cluster 1

Graphical user interface, application, table

Description automatically generated

Shape

Description automatically generated

Only Goalkeepers were selected in this cluster.

## K-Means Clustering – Cluster 2

Table

Description automatically generated

Chart, bar chart

Description automatically generated

Players with good playmaking abilities were selected in this cluster.

## K-Means Clustering – Cluster 3

Graphical user interface, application

Description automatically generated

Chart, bar chart

Description automatically generated

Players with good defending abilities were selected in this cluster.

## K-Means Clustering – Cluster 4

Graphical user interface, text, application

Description automatically generated

Chart, bar chart

Description automatically generated

Players with finishing (better goal scoring abilities) were selected in this cluster.

# Decision Tree Classifier

So, we calculated accuracy for max depth from using just 1 feature to all the way to using all 26 features.

Table

Description automatically generated

We found that at max\_depth = 12 we get the highest average accuracy.

Diagram

Description automatically generated with low confidence

Our decision tree model has ~62% accuracy. Also, the tree diagram is huge and hard to decipher.

# Random Forest Classifier

* For this model, we used all the integer attributes from our dataset.
* As random forest can predict string categories, we used the original best position column as our response variable.

Text

Description automatically generated

Below is the comparison of actual labels and predicted labels from this model:



Using the GridSearchCV parameters we created our final random forest classifier model.

Table

Description automatically generated

We can see that the hyper tuned model or the final model has a better accuracy than the base random forest classifier model. The final model also has way better accuracy than decision tree classifier model.

Using PCA to reduce dimensions (99% variance) did not give better accuracy (~76.3%).

# Neural Network

* First, we created a base neural network model with default parameters. Below is the comparison of actual labels and predicted labels from this model as well as the loss and accuracy of the model:





* Using the GridSearchCV parameters we created another neural network model. We can see that the hyper tuned model has a better accuracy than the base neural network model.



* Finally, we created another model after using PCA to reduce dimensions (99% variance) which gave us the highest accuracy with the lowest loss. This will be our preferred model for classification/ categorization. Below is the loss and accuracy of the final model:

# Regression

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Parameter | Hyper Tuned | Training Score | Test Score | RMSE |
| Linear regression | NA | No | 79.46445373 | 79.49090244 | 2.457451785 |
| Ridge regression | alpha=.1 | No | 79.46673515 | 79.49022246 | 2.457492524 |
| Ridge regression | alpha=2 | Yes | 79.46494424 | 79.49228732 | 2.457368814 |
| Random Forest Regressor | n\_estimators=200 | No | 98.974 | 92.724 | 1.463741355 |
| Random Forest Regressor | n\_estimators=300 | Yes | 98.98423877 | 92.77775627 | 1.458303105 |
| SVR | kernel="linear" | No | 78.93478475 | 78.93478475 | 2.481827582 |
| SVR | kernel="rbf" | No | 88.61223627 | 88.61223627 | 1.870760401 |
| SVR | kernel="rbf", C = 32 | Yes | 94.84316955 | 91.88830311 | 1.545494939 |

* According to the table above, we can say that Random Forest regressor after a GridSearchCV gave us the highest train and test accuracy and the lowest root mean squared error.
* For higher accuracy, Random Forest regressor model is preferred for this dataset.
* In case low overfitting is preferred, then SVR model should be used for this dataset.