Course: IST 718 Big Data Analysis
Title: NBA Players Analysis

Project Description:

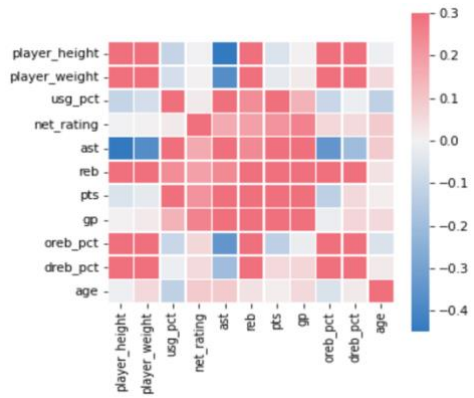
Aimed to analyze NBA players performance and investigate link between a player's physical attributes – height and weight to their performance. Also aimed to determine if a player is offensive and predict net rating of players for the next draft round.

Dataset Description:

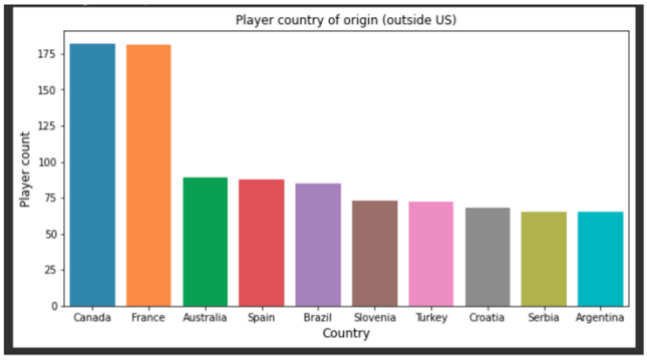
Dataset was taken from Kaggle in the form of a csv file. Data was originally gathered by NBA. The dataset consisted of 12305 rows and 21 columns ranging from 1996 - 2021. Features consisted of details of players, their performance and physical attributes like player age, height, weight, games per point, draft year, assists, rebounds, net rating, etc.

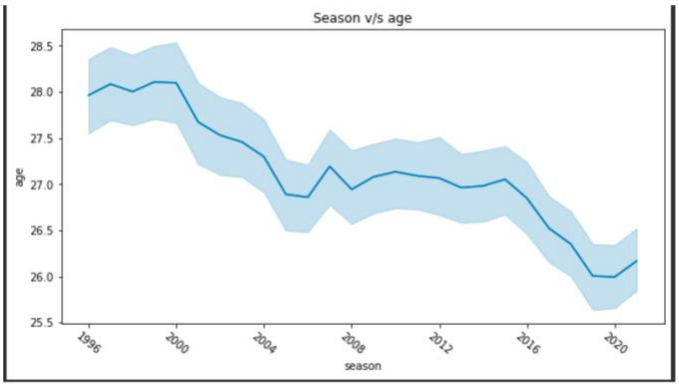
Data Exploration:

We realized that our data contained features such as college and draft number that were not relevant to our analysis. There were no null or duplicate values in our dataset. We changed the datatypes of our 17 numerical features to the right datatype for further analysis.
We also fixed schema schema for columns draft_year, draft_round and draft_number from string value 'Undrafted' to 0 so that our machine learning models would interpret the data better.

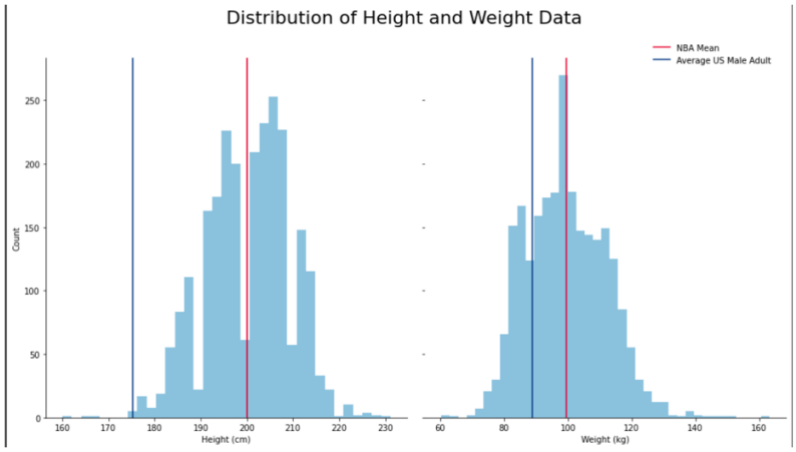


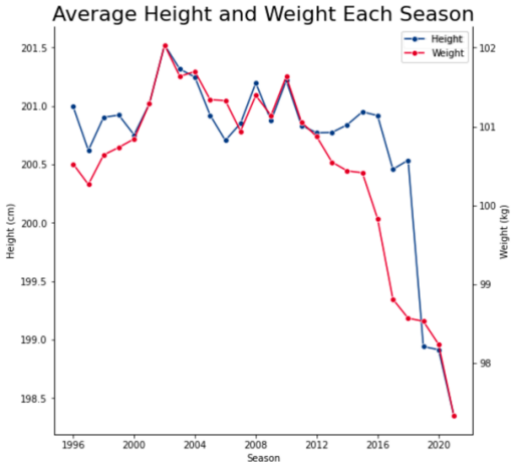
- We found that player height and weight are positively correlated with each other which meant that taller and heftier players get more rebounds.
- Points and assists are positively correlated which meant that assists are a key factor for scoring points.
- Height and assists are negatively correlated which meant that shorter players tend to be better assists.





We observed that outside of the United States, most of the players come from Canada and France. We also observed that as seasons pass, age of players being drafted had reduced considerably.





We thought that average height and weight of an NBA player is way more than an average US male. But as per the data we observed that it was not the case always. We also observed that the average height and weight of NBA players over the seasons had reduced.

Methodologies:

There was no target variable in the data, so we included variety of strategies covered in class for our prediction models.

- We used grouping approaches to select features required for different prediction models for all seasons' data.
- Feature engineering varied with each prediction model. Organized and feature engineered the variables using standard PySpark methods.
- We used vector assembler to make the features vector and standard scalar to standardize the features for our models.
- We developed pipelines to train and test the models using PySpark's standard pipeline methods.
- Conducted grid search for hyper parameter tuning using regression and binary classification evaluators on the validation set.
- To test classification model, we utilized AUC score, and to test regression models, we used mean square error as an evaluation metric.

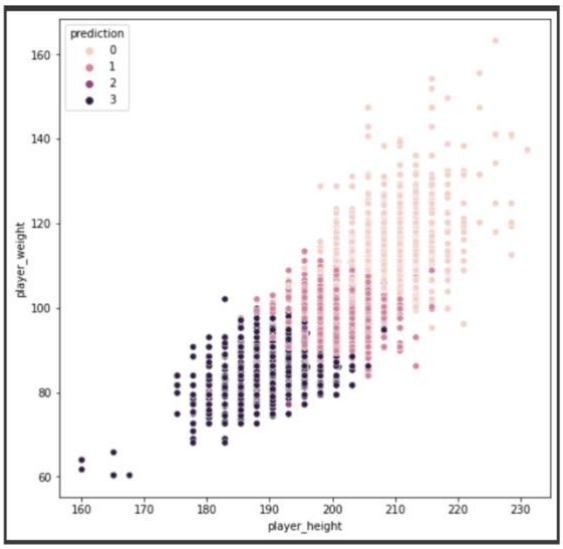
Data Modelling:

1. Clustering

A positive correlation between a player's physical attributes and the rebounds he received, and a negative correlation between a player's physical attributes and the assists he received was observed, so we investigated this relationship further. Filtered data to only show columns that describe a player's performance and physical characteristics. Columns used for clustering were ‘player_height’, ‘player_weight’, ‘pts’ (points), ‘reb’ (rebounds), ‘ast’ (assists) and ‘net_rating’. We used KMeans algorithm to build our clustering model. Clustering occurred based on physical characteristics by comparing the heights and weights of observations in clusters. We then compared the average values of each variable in the clusters to analyze them.

prediction	avg(player_height)	avg(player_weight)	avg(pts)	avg(net_rating)	avg(reb)	avg(ast)	count(1)
1	200.9183802663142	99.02187634141102	9.096987557301913	0.6393363894346225	3.540340537000646	1.7437677363021187	4581
3	189.63710975963	85.8839461754987	8.018940052128572	-4.7119026933101535	2.1168838690993366	2.888618592528237	3453
2	200.86054054054048	97.6903948108108	1.8270270270270268	85.16216216216216	0.7486486486486488	0.3054054054054054	37
0	209.22765233821053	113.66578067831819	7.353731695795921	-4.148889938592343	4.780302314596118	1.026735947094944	4234

We observed that there were 36 players in one cluster who had low stats. Also, a decrease in the average assist count and an increase in the average rebound count for other clusters as average height and weight increased. This was consistent with what we had analyzed from the heatmap.

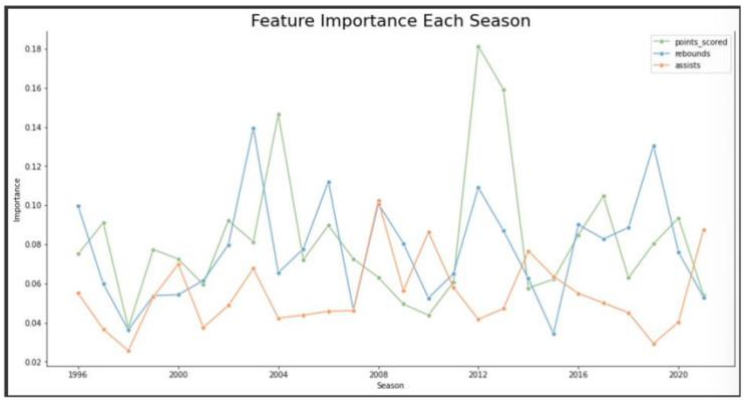


2. Net Rating Prediction

We used regression models – Linear Regression and Random Forest to estimate a player’s net rating. Columns used for regression models were 'team_abbreviation', 'age', 'player_height', 'player_weight', 'gp', 'pts', 'reb', 'ast', 'oreb_pct', 'dreb_pct', 'usg_pct', 'ts_pct', 'ast_pct'. With the best hyperparameters, on the test data, Random Forest model produced a marginally better MSE, even though both the models did well in calculating a player's net rating when compared to the range of the "net rating" variable.

- Comparison of feature importance across seasons

We divided the data into seasons and fitted the seasons into the random forest regressor to determine the importance of the features – points, assists and rebounds as they were the primary factors in determining net rating. We used entire data of each group for this analysis.



The importance of any variable did not continuously increase or decrease, which suggested that no variable was important for all seasons. This implied that if getting more points was crucial for one season, getting more rebounds was crucial for another season. According to this analysis we inferred that players and teams should prioritize overall development for improved team performance.

3. Classifying if a player is Offensive

We built logistic regression model to classify if a player is offensive. To get our target variable, we used column 'ts_pct' which was a measure of a player's shooting efficiency. We set a threshold for this classification by using the average of column 'ts_pct' which came out to be 0.51. If the value of this column for a player is over this average value, then the player was classified as offensive (1) otherwise player was classified as non-offensive (0). Column used for this model were 'oreb_pct' - a percentage measure of offensive rebounds , 'ast_pct' – a percentage measure of assisted goals, 'pts' – average number of points scored, 'ast' – average number of assists distributed. With the best hyper-parameters, the model produced a test AUC of 0.7121.

Future Scope:

By gathering more data, we can significantly improve the performance of our models since our data does not take into consideration other factors that lead to a player's success. Other factors such as body fat percentage, standing reach, hand width, shuttle run and many more could help improve the performance of our models.

Specific contribution to the project:

- Assisted in data cleansing, fixing schema for data.
- Assisted in standardizing features.
- Assisted in data analysis and building plots.
- Built model for classifying if a player is offensive.

Learning outcomes from the project:

- Learnt implementation of an entire project using PySpark. I was not well-versed with PySpark before this course.
- Got well versed with the concept of vector assembler for feature selection for models.
- Learnt to develop pipelines for training, testing and validation using PySpark.