IST 718 – BIG DATA ANALYTICS

NBA Players Analysis

Group 9

Group Members

Ayappa Sumanth Bhuma

Khushboo Saxena

Kirat Saran

Shreya Balan Nambiar

Table of Contents

[1] Abstract

As the popularity of the NBA has increased, so has the need to understand the behaviors of players on such platforms. To do this, researchers have collected data from ESPN's NBA section, one of the most popular sports databases which include features such as points per game, assists, rebounds, and more. After examining these unique features, we have tried to view the impact of height and weight of a player when it comes to their performance, predict if the player would get drafted in the upcoming round, determine if the player is offensive and understand specific performance characteristics of a player that helped their gameplay in a particular season. The dataset had been pulled together using NBA stats API and the initial missing values have been manually filled using data from Basketball Reference. We utilized a data collection containing 21 Player related attributes for each of the 12305 records. After that, the project goes on to fit multiple models with different hyperparameters and score their performance. Basketball is an amazing sport which contains a lot of data. The NBA has been using statistics so heavily, that they might surpass that of Major League Baseball (MLB) in using data, which was one of the first US (United States) sports league to use data to find hidden insights and patterns to benefit their team. In this study, we used a data collection of 12305 records of performance of players from 1996 to 2001 to investigate the link between a player's attributes and its performance. As a result of this investigation, we have trained and altered a slew of machine-learning models to forecast their performance. We implement a wide range of machine learning approaches to tackle these problems.

Link to this dataset: https://www.kaggle.com/datasets/justinas/nba-players-data?select=all_seasons.csv+%E2%80%8B

Overview and description of the data:

- The dataset has about 12k records of player performance from 1996-2021
- There are about 21 attributes for each player, out of them 17 being numerical.
- There are several dependent variables since we are tackling multiple issues.
- The in- detailed description of the features can be found in the appendix.

Specific Goals

Through machine learning models, we predict multiple events. We estimate whether a player is offensive, see the impact of height and weight of a player when it comes to their performance, understand specific performance characteristics of a player that helped their gameplay in the particular season. We have implemented classification and regression models here. The tuned hyperparameters for classification models are elasticNetParam, regParam (clustering and logistic regression) and the tunned hyperparameters for regression models are regParam, fitIntercept, elasticNetParam (for linear regression), maxDepth, maxBins, numTrees (for random forest regression). The next step was to hardcode these parameters. We found out Mean Squared Error values and … to determine the accuracy of the models.

[2] Data Collection / Cleaning

In this section, we will see a brief description of our features followed by data cleaning and wrangling.

[2.1] Exploring Data Analysis

1. **Checking the datatypes of our data:**

- First, we must check the datatype of our 22 features.
- For our analysis, we have not used all the columns of the dataset. Columns such as “college”, “draft_number” have not been used.
- We have created various visualizations to determine the correlation between the features.
- We have converted features into right datatype for model training

2. **Checking for NULL/NA Values:**

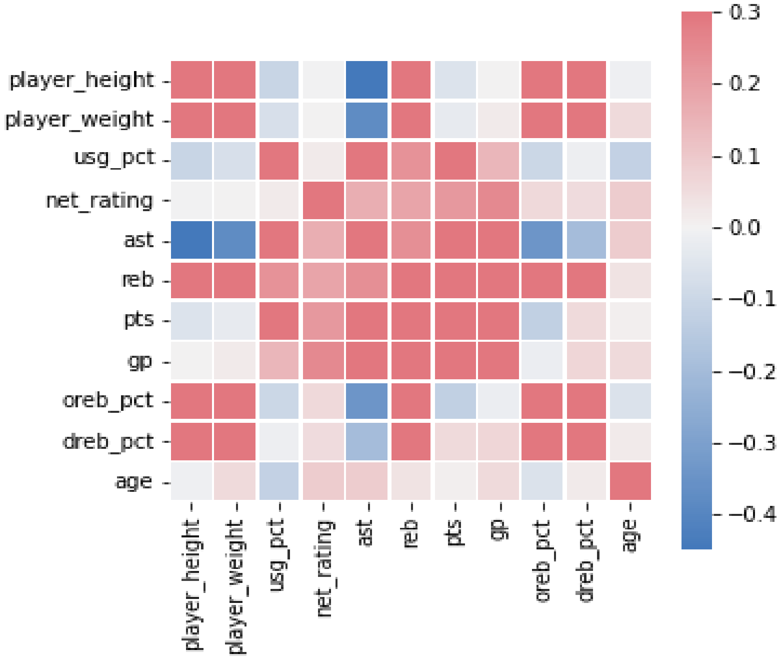
- We found out that the dataset did not contain null values.

3. **Fixing Schema**

- Fixed the schema for columns ‘draft_year’, ‘draft_round’ and ‘draft_number’ by replacing string value ‘Undrafted’ with 0.

[2.2] Data Exploration Insights using Statistical Techniques

Heat Map



Screenshot 1: A heatmap to see how strongly correlated a variable is with ither variables.

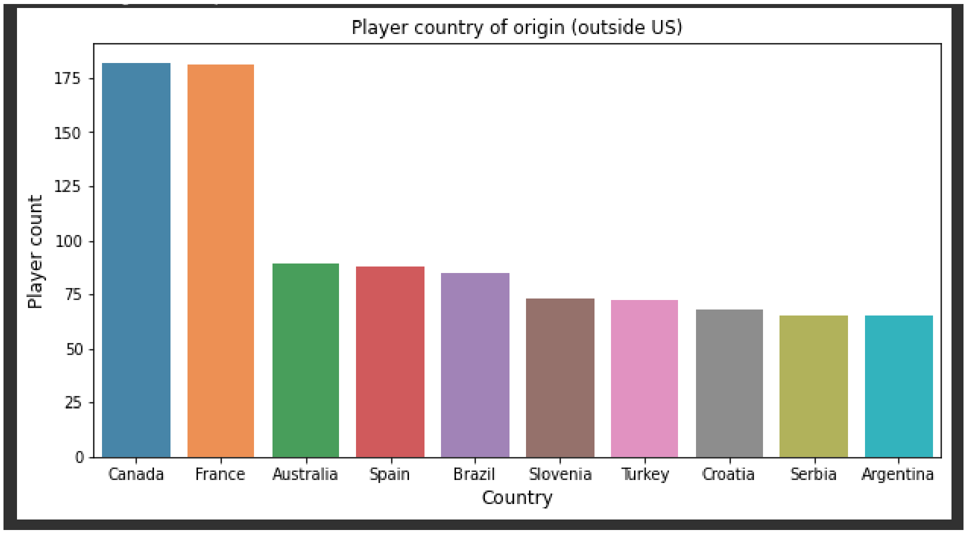
[2.3] EDA Takeaways

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

[3] Methodology

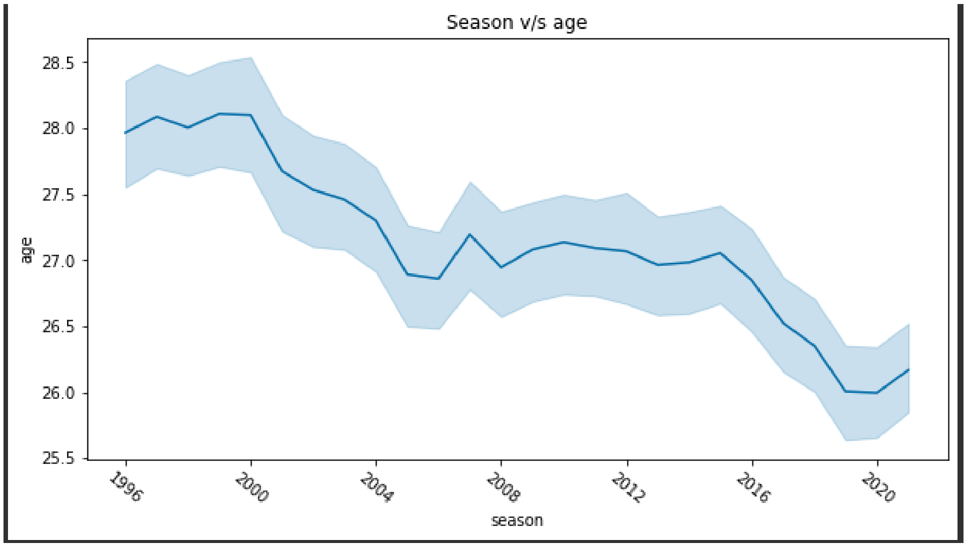
- To analyze the data of the NBA player, we used the CRISP-DM approach.
- We checked for missing data as part of the data preparation process, and we updated the variable structure.
- **Fixing Schema:** Fixed the schema for columns 'draft_year', 'draft_round' and 'draft_number' by replacing string value 'Undrafted' with 0.
- There is no particular "target variable" in the data, so, we decided to include a variety of strategies covered in class in our prediction models.
- We used grouping approaches to select the data required for the model from all seasons' data for all the prediction models we created.
- To develop pipelines for training and testing the models, we utilized pyspark's standard pipeline methods.
- We used vector assembler to make the features vector and standard scalar to standardize the features in the models.
- To conduct the grid search for hyper parameter tuning, we employed regression and binary classification evaluators on the validation set.
- To test the classification model, we utilized the AUC score, and to test the regression models, we used the mean square error as a metric.
- Since data analysis makes up the majority of our project, the feature engineering varied with each prediction model. In order to organize and feature-engineer the variables, we used standard PySpark methods. In the model prediction section, specifics on feature engineering for each model are provided.

[4] Data Exploration Insights



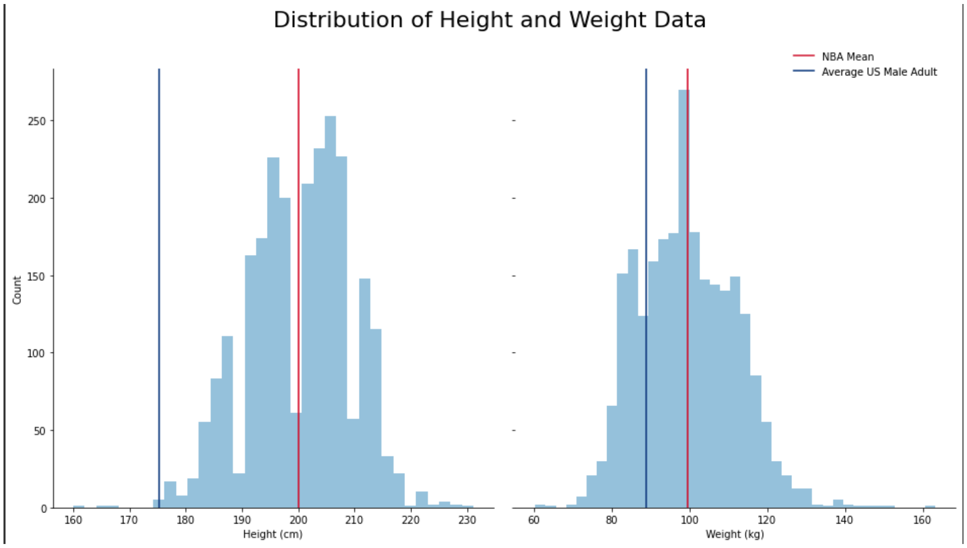
Screenshot 2: A bar plot displaying countries having most players outside of the US

We can observe that outside of the United States, most of the players come from Canada and France.



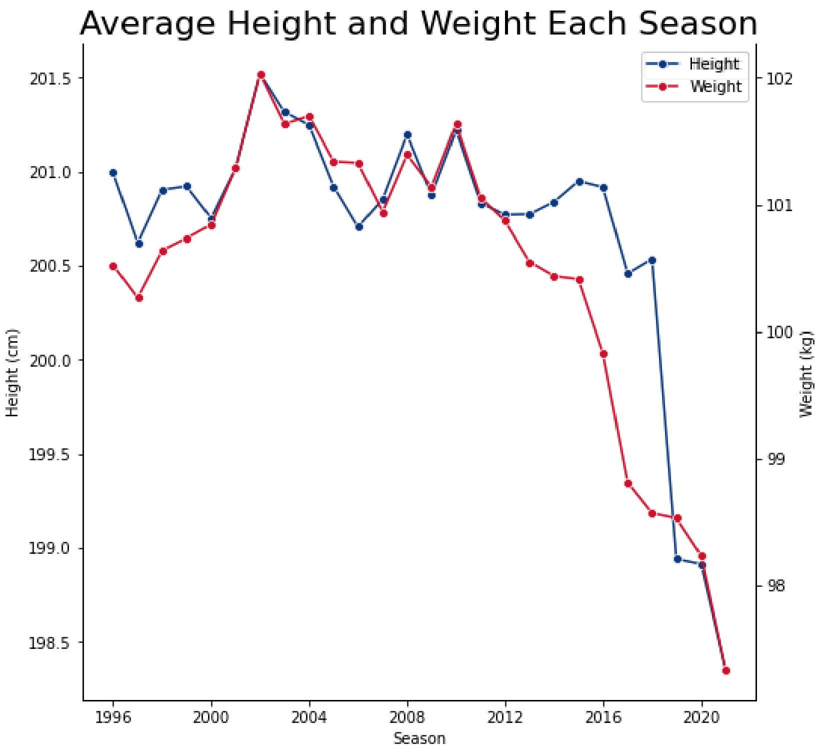
Screenshot 3: A line plot displaying a trend of age of players over the years

We can observe that as seasons pass, age of players being drafted has reduced considerably.



Screenshot 4: A distribution height and weight of players

We can see that average height and weight of an NBA player is way more than an average US male. But as per the graph we can say that it is not the case always.



Screenshot 5: A distribution average height and weight of players over the seasons

We can see that the average height and weight of NBA players over the seasons has reduced.

[5] Data Modelling / Model Prediction

[5.1] Clustering

As we noticed 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, we decided to investigate this relationship further.

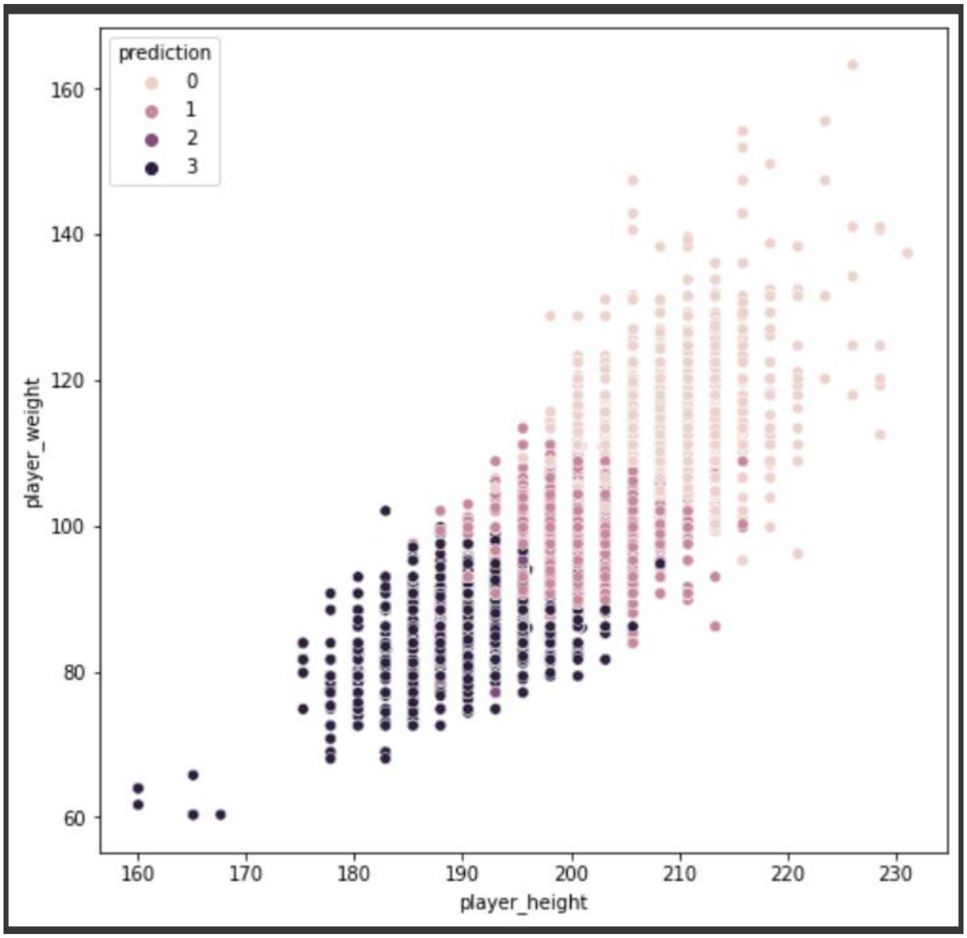
We have filtered the data to only show columns that describe a player's performance and physical characteristics. Using clustering, we grouped the data in this set according to their physical properties.

The columns in the clustering data are “player_height”, “player_weight”, “pts” (points), “reb” (rebounds), “ast” (assists) and “net_rating”

To create a clustering model, we used PySpark ML's KMeans machine learning technique.

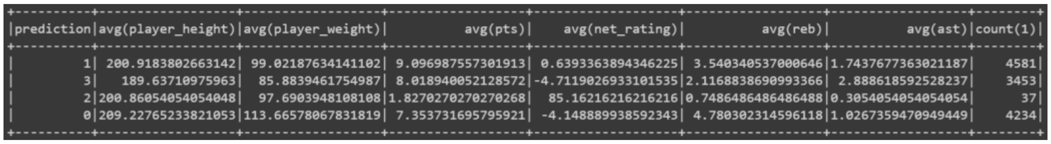
We determined that 4 was the ideal k using the elbow method and silhouette score, then we applied the KMeans clustering algorithm to divide the data into 4 clusters.

We noticed that the clustering occurred based on physical characteristics by comparing the heights and weights of observations in clusters.



Screenshot 6: A distribution showing clustering of height and weight of players

Then, we 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|1.0267359470949449|    4234|
+----------+------------------+------------------+------------------+-------------------+------------------+------------------+--------+
```

Screenshot 7: A table showing the average values

There are 36 players in one cluster who have low stats. We observe a decrease in the average assist count and an increase in the average rebound count for other clusters as average height and weight increase. This is consistent with what the correlation diagram showed.

[5.2] Net Rating Prediction

The team's point difference per 100 possessions while the player is on the floor is provided by the "net rating" variable. Since this is a true indicator of a player's performance, we built regression models to estimate a player's "net rating."

The following columns were used as features in the models we built: 'team_abbreviation', 'age', 'player_height', 'player_weight', 'gp', 'pts', 'reb', 'ast', 'oreb_pct', 'dreb_pct', 'usg_pct', 'ts_pct', 'ast_pct'

In PySpark, we constructed two regression models: one with linear regression and the other with a random forest regressor. By running a gridsearch of the parameters, we chose the hyperparameters for both models.

Prior to creating the models, we created the feature vector using vector assembler and obtained standardized features using the standard scalar.

Next, we constructed pipelines and ran a grid search using the regression methods of Spark ML.

To compare the models in the grid, we have performed cross validation using regression evaluator.

The grid search for linear regression produced an optimal regression parameter (alpha) of 0.1, an elastic net parameter (lambda) of 0.5, and the optimal model's validation mean square error of 129.45.

With the best hyperparameters, the linear regression model produced a train MSE of 143.88 and a test MSE of 142.35.

The grid search for random forest regression produced the ideal number of trees as 10, the maximum Depth as 5, and the maximum number of bins as 5.

With the best hyperparameters, the Random Forest regression model produced a train MSE of 136.61 and a test MSE of 136.59.

On test data, the random forest model produced a marginally better MSE.

The original dataset's net rating variable has a range of -250 to 300. The test MSE reveals the models did well in calculating a player's net rating when compared to the range of the "net rating" variable.

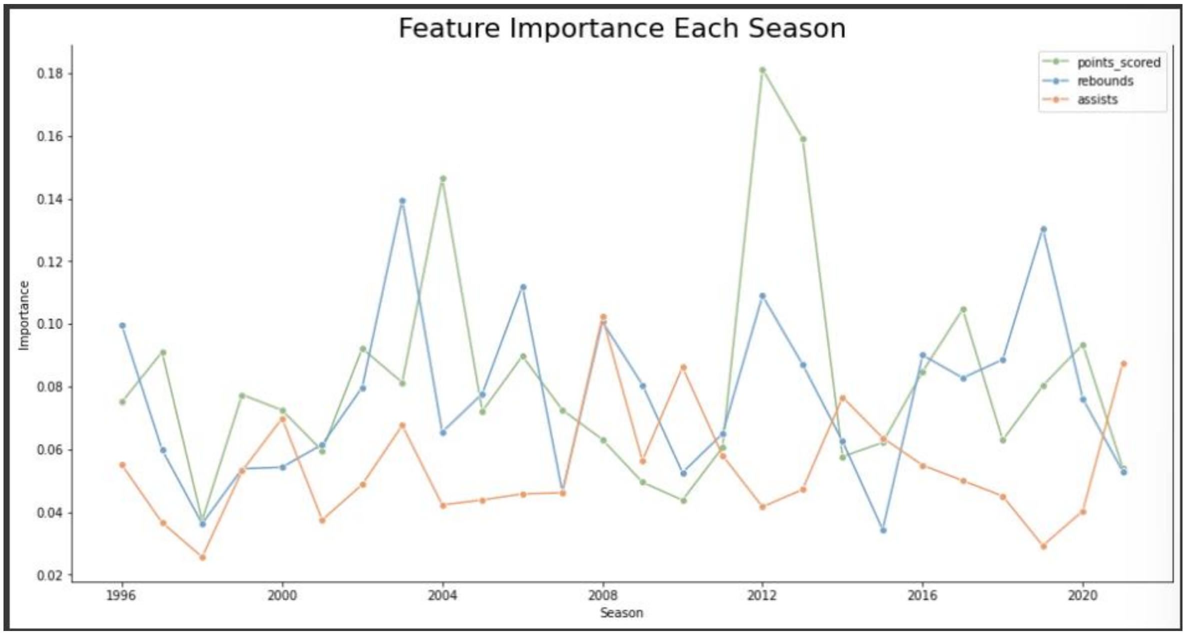
[5.2.1] Comparision of Feature Importance across seasons

Since we have season-by-season player data, we have compared each feature's significance in predicting a player's net rating as the seasons go by. The idea is to identify the shift in factors determining the net rating of a player.

In order to perform the analysis, we divided the data into seasons (year numbers) and fitted the seasons into the random forest regressor to determine the importance of the features.

As the data is used only to get the feature importance, we used entire data of each group instead of taking train and validation sets

Then, to see how the features' relative importance changed, we created line plots of the features. The comparison of the feature importance of these variables has been provided because points, assists, and rebounds are the primary focus of player statistics.



Screenshot 8: Comparison of feature importance across season

Although the importance of any variable did not continuously increase or decrease, plot suggests that no variable is always important for all seasons. This implies that if getting more points was crucial for one season, getting more rebounds was crucial for season. According to the analysis, players and teams should prioritize overall development for improved team performance.

[5.3] Classifying if a player is offensive

We wanted to classify if a player is offensive by using the data that we had. We built a logistic regression model to achieve this using PySpark.

To get our target variable, we used column 'ts_pct'. 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 our player is over this average value, then the player will be classified as offensive (1) otherwise player will be classified as non-offensive (0).

The following columns were used as features to build this model: 'oreb_pct' , 'ast_pct', 'pts', 'ast'. We created a feature using vector assembler.

Next, we constructed pipelines and ran a grid search using the logistic regression methods of Spark ML.

To compare the models in the grid, we have performed cross validation using Binary Classification evaluator.

Gridsearch gave us the best hyperparameters to build our final model with an alpha value of 0.1 and lambda value of 0.001.

With the best hyperparameters, the Logistic regression model produced a test AUC of 0.7121.

[6] Model Inference

S.No.	Model	Features	Techniuque used	Scoring Metric	Goal
1	Clustering	“player_height”, “player_weight”, “pts” , “reb”, “ast” , “net_rating”	KMeans machine learning	-	Analyse in depth the positive correlation between a player's physical attributes and the rebounds he received ; a negative correlation between a player's physical attributes and the assists he received
2	Net rating prediction	'team_abbreviation', 'age', 'player_height', 'player_weight', 'gp', 'pts', 'reb', 'ast', 'oreb_pct', 'dreb_pct', 'usg_pct', 'ts_pct', 'ast_pct'	Linear Regression and Random Forest Regression	MSE	Predict net rating of a player based on the player stats that we have.
3	Player is offensive	‘oreb_pct’ , ‘ast_pct’, ‘pts’, ‘ast’	Logistic Regression	AUC	Predict if a player is offensive based on the data we have.

[7] Conclusion

- The NBA players dataset contains a variety of intriguing variables that provide useful information about the game's various statistics. Having the player's season-by-season data, we turned our attention to comparing the statistics as the seasons went by.
- We choose our EDA queries and modeling strategies with a variety of distributed computing and Pyspark methodologies on a dataframe, to become familiar with Pyspark.

- We evaluated the change in attributes as the NBA season progressed using EDA on the original dataframe and physical attributes vs. performance dataframe. After that, we used clustering to comprehend this change better.
- We used regression techniques after tuning their hyper parameters to predict the net_rating of a player based on his statistics. Then, as the season progressed, we compared the shift in feature importance of variables used to determine a player's net rating.
- We performed logistic regression to determine if a player is offensive or not. To construct the features vector and predict the class, we employed feature engineering techniques in pyspark

[8] Appendix

[8.1] Detailed Description of Data

- player_name : Name of the player
- team_abbreviation : Abbreviated name of the team the player played for
- age : age of the player
- player_height : Height of the player (in cm)
- player_weight : Weight of the player (in kg)
- college : Name of the college attended by the player
- country: Name of the country the player was born in
- draft_year : The year in which the player was drafted
- draft_round : The draft round the player was picked
- draft_number : The number at which the player was picked in his draft round
- gp : No. of games played throughout the season
- pts : Average number of points scored
- reb : Average number of rebounds grabbed
- ast : Average number of assists distributed
- net_rating : Team's point differential per 100 possessions while the player is on the court
- oreb_pct : Percentage of available offensive rebounds the player grabbed while he was on the floor

- dreb_pct : Percentage of available defensive rebounds the player grabbed while he was on the floor
- usg_pct : Percentage of team plays used by the player while he was on the floor (FGA + Possession Ending FTA + TO) / POSS)
- ts_pct : Measure of the player's shooting efficiency that takes into account free throws, 2 and 3 point shots (PTS / (2*(FGA + 0.44 * FTA)))
- ast_pct: Percentage of teammate field goals the player assisted while he was on the floor
- season : NBA season played

[8.2] References

- https://spark.apache.org/docs/latest/api/python/
- https://www.google.com/
- Class notes

[8.3] Future Work

- Gather more data to improve our models

There are some limitations since it does not take into consideration all the other factors that contribute to the success of the player. Since basketball is a team sport, we have the data of the individual players and not the team's performance. There are other factors such as body fat percentage, standing reach, hand width, shuttle run that would help improve the accuracy of the model.