MICHAEL JORDAN VS LEBRON JAMES

Utilizing regression analysis to compare both players and determine who is the best NBA star

Marc Martín Ortega (1564274)
Marina Vázquez Guerrero (1563735)
Mireia Fernández Fernández (1562636)
Aprenentatge Computacional, III MatCAD

December, 2021

Contents

1 Introduction

Competitiveness has always been a prominent trait in people's lives. In the words of Sigmund Freud, "Humans are born screaming for attention and full of organic drives for fulfillment in various areas" [1]. This is why it feels natural to compare and compete with each other, being athletics the star field for the matter.

More precisely, the purpose of this project is to compare two of the most recognized basketball players in the world: Michael Jordan, the greatest player of all time, and LeBron James, a star that is also considered one of the best players in the world.

Will LeBron James be able to catch up, or even surpass, Michael Jordan before his career ends or not? That is the question that we will try to answer in the following pages, utilizing different types of regressions and analysis methods on the datasets made for each player [2]. These will include linear regressions, polynomial fits, as well as the application of an automatic PCA process and the implementation of a gradient descent.

[1] Quote extracted form: Bonnie R. Strickland, *The Gale Encyclopedia of Psychology, 2nd Edition* (United States of America: Gale Group, 2007).

[2] The two datasets used have been extracted from: Kaggle. "Jordan vs Lebron". https://www.kaggle.com/edgarhuichen/nba-players-career-game-log (last accessed December 6th, 2021).

2 Data Analysis

2.1 Data preparation

Once the *csv*s are downloaded, we observe that there are two different datasets with the same column types and similar number of rows, one with the data for Michael Jordan and one for LeBron James[3].

With just a glance it is noticeable that some columns have a worrying amount of NULL entries, therefore the first step for a clear visualization is to remove or change the values of those variables with that characteristic: the column *min_plus* is completely removed since Jordan's does not have a single value, and the necessary amount of zeros is added to the columns *threep* and *ftp* of each player[4].

After that, we decided to modify some given variables to be able to work with them more efficiently. This is the case for the *result*, where we removed the "W" or "L" etiquette and left the numerical value; the *age*, where we changed it to a decimal number representing the years lived; and lastly, in the *mp* column we chose to transform it to total number of minutes played.

Finally, we determined that the values of *game*, which corresponds to the number of the row, and *date*, exact date of each match played, were not relevant enough and eliminated them.

2.2 Data visualization

Even though the data gathered has 21 different attributes for each player, some of them display quite unstable distributions while some appear to have clearer Gaussian distributions [5] such as the points gotten in a match by the player alone or the final result. At first this may not be considered important enough, but it will be demonstrated later on that these attributes must be considered when applying regression methods to predict their game scores.

[3]The table with the information of all the variables and their data types, as well as their brief explanation of their meaning, can be found in the subsection 6.1 of the appendix.

[4]This decision was made because we noticed that *nan* entries were caused by a division by zero in their formulae.

[5]Their respective graphs are shown in the figures 10 and 11, in the appendix.

2.3 Target attribute

After all considerations were taken into account, we chose to predict the game score of the player for each dataset. This was due to the fact that the main goal of this project is to understand and to contrast LeBron James' and Michael Jordan's power in the basketball court.

For that purpose, we have decided to create histograms of the game scores for a better understanding of what we are going to predict:

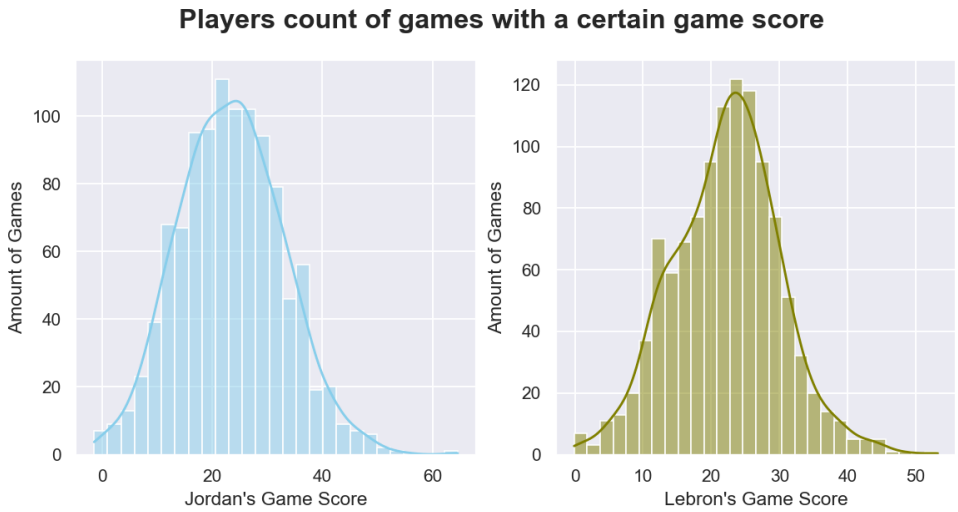


Figure 1: Visualization of Jordan's and LeBron's target attribute

We can clearly see that both players have similar amounts of data but two completely different cases. On one hand, Michael Jordan showcases a larger range with a more defined Gaussian distribution; that is important when comparing both player's overall performance since it is observable that he reached better scores than the other, in particular, he got a 64.9 in his best game versus the 53.2 of LeBron's. On the other hand, it is LeBron who's counts go over 100 matches with a similar game_score and has a slimmer curve, which could indicate a more stable performance over the years.

With all of this in mind, we will proceed to carry out different regressions to obtain the right tools to compare both players [6].

[6] In the following regressions an standardization of the target attribute will also be used, their respective histograms can be found in the subsection 6.2 of the appendix.

3 First Regressions

The following section contains all the data gathered for and obtained in all the different models we used to predict the target attributes later on used to compare both players and draw our conclusions.

3.1 Covariance matrices

In order to get the most accurate results, it is compulsory to get their covariance matrices to see which attributes are more correlated to each other. The idea here is to take a look at their last rows and to start making some hypothesis about which are the best candidates for our linear regressions.

In those matrices, shown in the figures 12 and 13 in the appendix, we observe the lack of correlation in some areas. Whereas this may cause some confusion at first, it makes sense if we consider the physical events they describe, for how many times the player has stolen the ball from the opponent team has no direct link with the percentage of three point field goals the player got in said game.

Nevertheless, we are mostly interested in the last row of the heatmaps as it indicates the correlation between the different columns and our target attribute, the *game_score*. The data exhibits that the most important attributes in order to achieve a good prediction, which are the same for both players, will be *fg*, *fgp*, *ft*, *fta* and *pts*. Those are the ones that have greater correlation, and all of them on or above the 0.5 mark. Moreover, if we take a closer look, we can see that there is also great correlation, on and over the 0.8 mark, between *fg* and *fgp*, *ft* and *fta*, and between the *ft* and *pts* attributes. Lastly, we observe that there is no relevant negative correlation in our data.

3.2 Linear Regressions

Starting from the base, the next step is trying to get a linear model with just one of the attributes to describe the game score. Seen that there is almost absolute linear correlation between *pts* and *game_score* it is to be expected that the best fit occurs when the attribute used is this one.

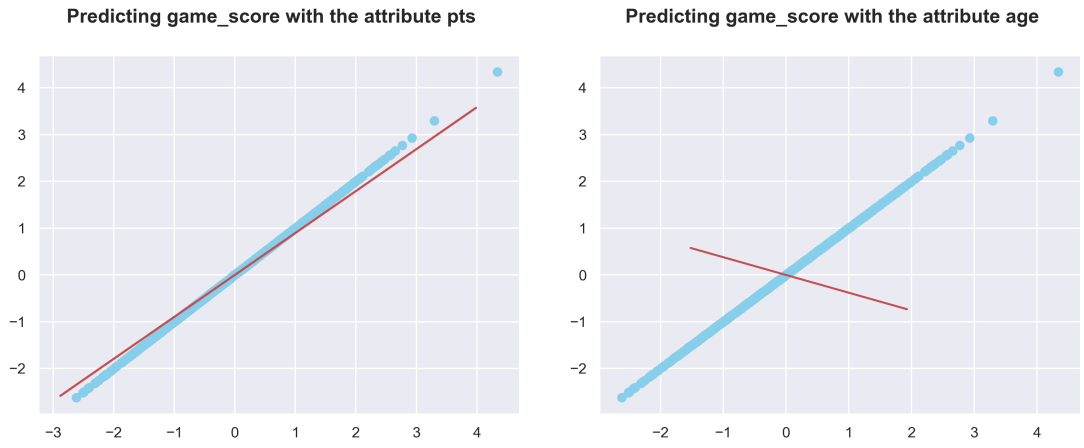
In any case, we considered a simple linear model with just each attribute separately, and an intercept, using the standarized datasets and got the results shown below.

3.2.1 Simple linear regressors for M. Jordan's dataset

	age	**result**	**mp**	**fg**	**fga**	**fgp**	**three**
MSE	0.8551	0.9493	0.8793	0.3445	0.7432	0.5638	0.9552
R^2	0.1441	0.0499	0.1199	0.6552	0.2561	0.4357	0.0439

	threeatt	**threep**	**ft**	**fta**	**ftp**	**orb**
MSE	0.9725	0.9687	0.6696	0.7113	0.8971	0.9566
R^2	0.0266	0.0304	0.3297	0.2880	0.1020	0.0425

	drb	**trb**	**ast**	**stl**	**blk**	**tov**	**pts**
MSE	0.9715	0.9415	0.9376	0.8415	0.9450	0.9957	0.1971
R^2	0.0276	0.0576	0.0615	0.1578	0.0541	0.0034	0.8027



(a) Regression made with *pts* (b) Regression made with *age*

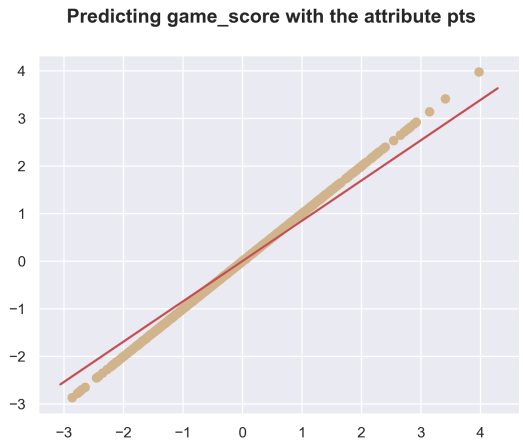
Figure 2: Graphics of Jordan's *game_score* best and worst linear predictions using just one attribute.

3.2.2 Simple linear regressors for L. James' dataset

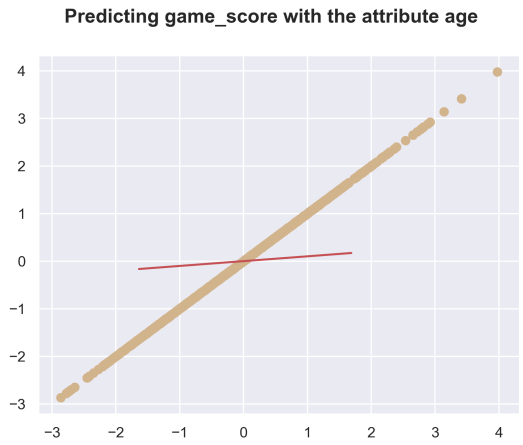
	age	**result**	**mp**	**fg**	**fga**	**fgp**	**three**
MSE	0.9889	0.9260	0.9196	0.4484	0.8310	0.6261	0.8237
R^2	0.0102	0.0732	0.0796	0.5512	0.1682	0.3734	0.1756

	threeatt	**threep**	**ft**	**fta**	**ftp**	**orb**
MSE	0.9071	0.9012	0.8131	0.8647	0.9449	0.9741
R^2	0.0921	0.0980	0.1862	0.1345	0.0542	0.0251

	drb	**trb**	**ast**	**stl**	**blk**	**tov**	**pts**
MSE	0.9532	0.9347	0.9247	0.9419	0.9730	0.9674	0.2833
R^2	0.0459	0.0645	0.0745	0.0572	0.0262	0.0317	0.7164



(a) Regression made with *pts*



(b) Regression made with *age*

Figure 3: Graphics of LeBron's *game_score* best and worst linear predictions using just one attribute.

3.2.3 Comparison of the simple linear regressors

Given the linearity of the *game_score*, it is plausible that an attribute that has such a high correlation, as *pts* has, is able to predict with a model this simple. Jordan's linear model seen in 2(a) is the most astonishing result, for its R^2 even surpasses the 0.8 mark.

3.3 Multivariate Regression

It has been seen that a simple linear regressor could do a decent job at predicting *game_score* with just one variable. Now, we will take another step and try to predict more precisely adding more variables to create a linear multivariate regressor.

In the following figures, we can observe that if we plot the real values against the predictions, the result is an almost perfect line that coincides with the bisection of the quadrant, meaning that this regressor is almost perfect.

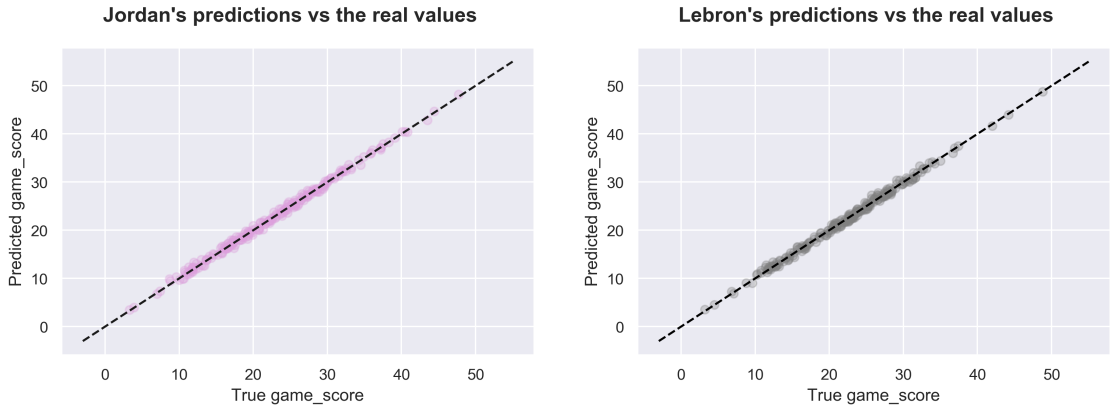


Figure 4: Multiple linear regression for each players using all the attrbiutes.

In this case, the data used has not gone through the standardization process because we wanted to see if it was able of predicting well with the original values. Initially, the idea was to show how much it could change from non standardized to standardized data, but since the results are much more precise than we originally thought, we decided it was worth showing these results.

3.3.1 Table of comparison between multivariate regressors

Nevertheless, it is important to compare how the models behave when standardising the attributes. Below there are two tables that display the different scores for both R^2 and MSE.

- R^2 score

		Training	Cross Validation	Test
Non Standardized Data	**L. James**	0.9962	0.9949	0.9954
	M. Jordan	0.9968	0.9951	0.9963
Standardized Data	**L. James**	0.9962	0.9949	0.9954
	M. Jordan	0.9968	0.9951	0.9963

- MSE score

		Training	Cross Validation	Test
Non Standarized Data	**L. James**	0.2572	0.2425	0.2334
	M. Jordan	0.2545	0.3487	0.2424
Standarized Data	**L. James**	0.0042	0.0040	0.0038
	M. Jordan	0.0028	0.0038	0.0027

The most noticeable trait it presents is the fact that the R^2 values are the same for both types of data [7]. It could have been expected to return similar numbers, since the linearity is evident at this point, yet the *MSE* score shows obvious differences between the two groups: the standardized data returns values two magnitudes smaller than the non standardized data.

Given that there is most likely to be a linear formula to calculate the *game_score* of the players, we concluded that the normalization of the datasets does not influence in the predictions made.

[7] In the tables above, there are only four decimal digits presented for reading comfort and neatness. In the appendix's section 6.4 can be seen that all the digits coincide, except maybe the last one.

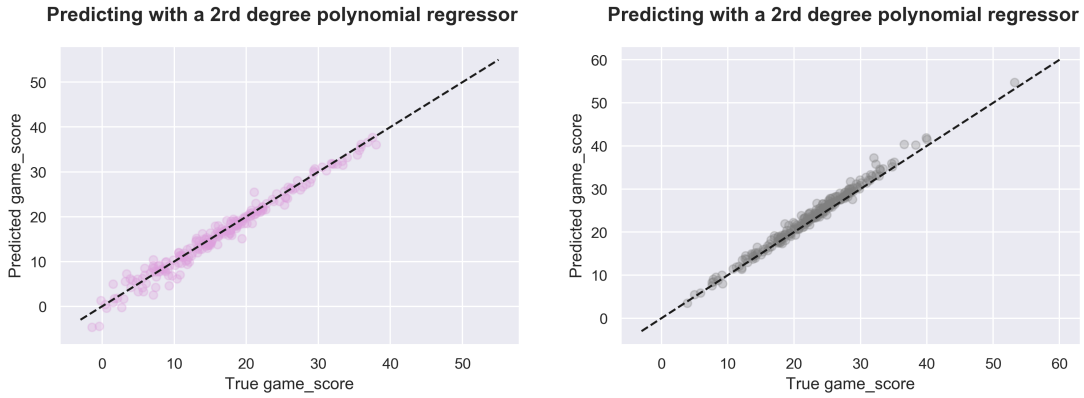
3.4 Polynomial Regressions

In this final type of regression analysis, we have chosen to cover the most used polynomial fits: the second and third degree. The goal is to observe whether the results that we had gathered up to this point could be improved or not, using higher degrees in the model.

3.4.1 Second degree polynomial fit

For this first approach, the data shows that this kind of model estimates well enough for both players, returning an R^2 score of 0.989971 for Jordan and a 0.988814 for LeBron.

Despite this, the cross validation section of the samples returned R^2 scores lower than the seen until now, as seen in 5, with 0.972059 for Jordan's data and 0.965862 for LeBron's. If we compare these with the scores obtained until now, we start to see the loss of precision for each player. In 5(a), the lower predictions are more scattered; while in 5(b), the predictions tend to be higher than the real values.

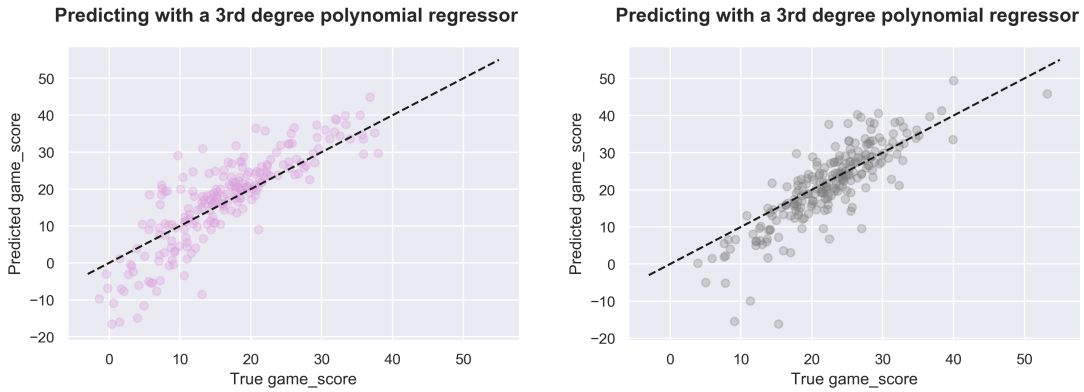


(a) Results for the Jordan's dataset (b) Results for the LeBron's dataset

Figure 5: Graphics of the prediction results for their cross validation subsets.

3.4.2 Third degree polynomial fit

For this other approach, it has been made clear that this is a regressor that does not describe well the data given. Just by looking at the graphics it is clear that the information that this polynomial model does not meet the expectations.



(a) Results for the Jordan's dataset (b) Results for the LeBron's dataset

Figure 6: Graphics of the prediction results for their cross validation subsets.

In these cases, the predictions for the cross validation samples returned R^2 scores of 0.229665 for LeBron James and 0.365974 for Michael Jordan. This alone was enough to decide that it was not worth running more tests for those models.

3.5 Principal Component Analysis

For the final point of this section, we used the PCA process to monitor the performance of the prediction models with different reductions of components in its attributes.

In order to do so, we performed different PCA reductions where we explicitly stated the number of components that the space should get reduced from dimensions 1 to 20, which corresponds to the number of attributes that datasets contain, as seen in the graphs displayed in the figure 7.

For both datasets, the automatic PCA chose to not reduce the space, and decided it had to stay in a 20 dimensional space. The results of the R^2 and the MSE scores implied that, for both players, the best multivariate regressors were the ones shown in 3.3, since the ones returned by the code coincided exactly with the data exhibited in that section.

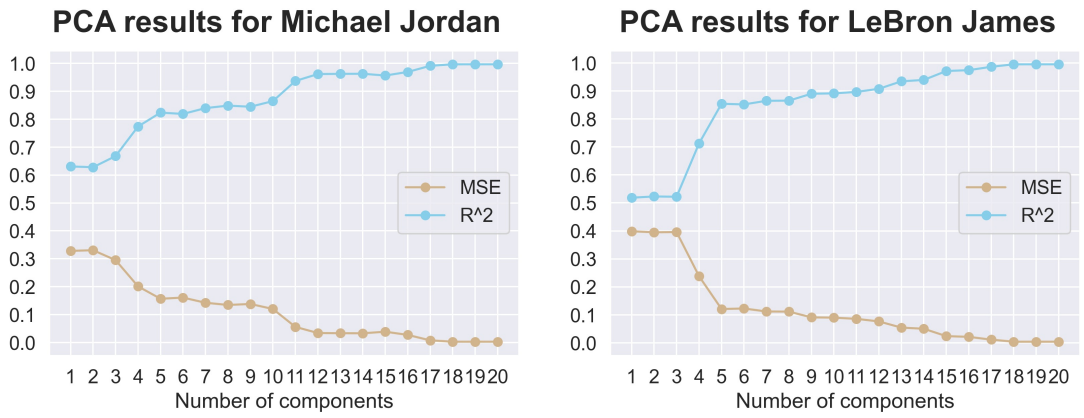


Figure 7: PCA's MSE and R^2 for each dimensionality reduction.

The behaviour of Jordan's PCA procedure through the number of components used has a stairs-like pattern, ascending in the case of the R^2 score and almost exactly mirrored, but descending, for the MSE score. It is observable that at four, eleven and seventeen as dimensions, the scores change more notably and then keep similar values until the next number of those mentioned.

Surprisingly, the behaviour of James' PCA procedure shows quite a different evolution. In this case, its R^2 increment, and MSE descendent, is much more uniform; with the only big growths in the dimension reductions four and five, and then subtler growths in the dimension reductions nine, thirteen and fifteen.

Looking at the overall performance, particularly from the fifth dimension onward, both cases show a similar growth and conclude what had been said previously, the more the number of components is the better the scores get, leading to a 20 dimension space that is actually the initial space we were working with.

4 Gradient Descent

On this section, we implemented a linear regressor that takes advantage of the gradient descent method. The implementation takes the form shown below when given the following variables:

- m is the number of rows of the data used.
- y is our predicted variable.
- x is the row of attributes from the samples that corresponds to the y.
- α is the learning rate (hyperparameter).
- λ is the regularizator (hyperparameter).

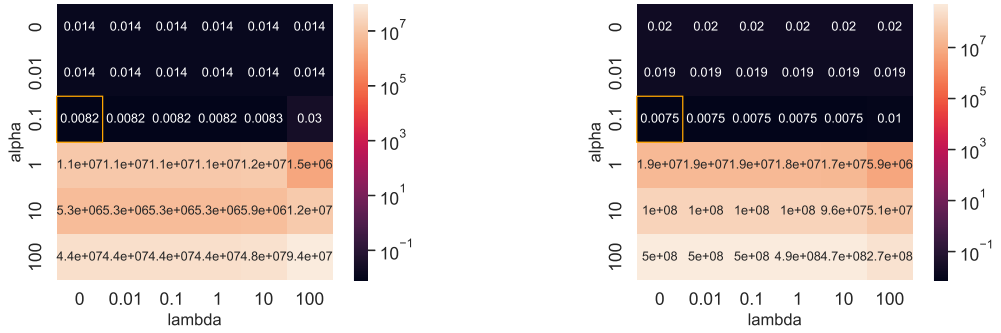
$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(f(x^i;w) - y^i\right)^2 + \lambda\sum_{j=1}^{n}\left(w_j^2\right)\right]$$

$$w_0 = w_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(f(x^i;w) - y^i)\cdot 1$$

$$w_j = w_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(f(x^i;w) - y^i\right)\cdot x_j^i - \frac{\lambda}{m}w_j\right]$$

Given the high dimensionality of our dataset, it has 21 attributes, it is crucial to finely tune our initialization, since if we were to start at any random point, we could fall into a non-wanted optimum. Taking that into account, we decided to start the initial weights of our regressions at the one-digit truncated optimum found already in 3.3 in order for the gradient descent to be efficient and well-behaved.

We have found that for both regressors the best combination of hyperparameters is $(\alpha, \lambda) = (0.1, 0)$. Nevertheless, we believe that the regularizator does not improve our model for any value due to its natural linearity, since the MSE gets its best value at 0.008151 for Jordan's dataset and 0.007516 for LeBron's, taking about 1.5 seconds for both datasets. Here 8, we provide the different MSEs scores found for different hyperparameters.



(a) Results for the Jordan's dataset (b) Results for the LeBron's dataset

Figure 8: MSEs for both players given a set of hyperparameters.

Now, since we have established that polynomial models don't perform well enough with our purely linear data, we decided to train our gradient descent with only the two most important attributes, which are the same for both players, those being the *pts* and *fg*. We took advantage of this dimensionality and plotted our predictor plane with all the data, see 9.

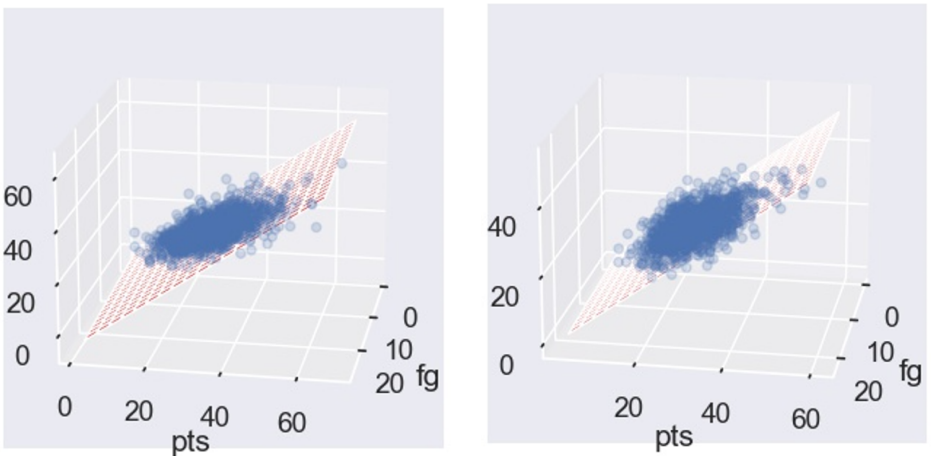


Figure 9: Predictor planes for Jordan and LeBron respectively, with *game_score* on the Z axis.

Lastly, if we were to visualize the predictor hyperplane over the data, it wouldn't really help to categorize those badly predicted since they are basically non-existant, again, by the natural linearity of our datasets.

5 Conclusions

Given the fact that we were working with two different datasets at the same time, both with the same attributes for two different people, we ought to look for a type of model that could work on both or find two different models that we could use to compare the behaviour of the data.

Over the course of the work that has been done, two results rose to the surface.

Firstly, the models implemented for Michael Jordan would always return better fits, in terms of the R^2 and the MSE scores, from a hundredth of the result to increasing LeBron's score almost by a half in 3.4.2.[8]

Secondly, almost all of the implemented regressors achieved a high R^2 score and resulted in decent models; from the linear regressors using only one attribute to the polynomial fits, in exception of the third degree. Therefore, after taking into consideration the results, as well as the amount of time invested in each model to adjust and compare, we concluded that the best model for the data given is the multivariate models found in 3.3, shown in 6.4.

Moreover, the later analysis that were made using the PCA procedure and then the Gradient descent, where consistent with what has been said.

And as a final insight, when we realized than in this multivariate regressor, half of the coefficients returned similar values. This behaviour, and the outstanding results for the linear regressors gave us the intuition that there was some sort of formula to obtain a player's *game_score*. Indeed, there os an exact method [9].

Upon further investigation, we realized that the R^2 obtained until now would not reach a perfect score due to the fact that there is a variable used in the formula that is missing in the datasets: the count of personal fouls by game, yet the model will be perfectly able to make predictions in the future.

[8]The complete R^2 score for LeBron James was 0.22966478111415867, where we can notice that $0.22966478111415867 \cdot 1.5 = 0.344497171671238$, that is almost 0.3659743391035688, the score for Jordan's regressor.

[9]NBAstuffer. "Analytics 101, Game Score in Basketball Explained". https://www.nbastuffer.com/analytics101/game-score/ (last accessed December 4th, 2021)

6 Appendix

6.1 Table with the information about each attribute

Variable	Type	Description
game	int	Number of the game
date	date	The date of the game
age	str	Age of the player
team	str	Team he is playing with
opp	str	Name of the opponent team
result	str	Difference of points in the result
mp	time	Minutes played in the game
fg	int	Field goals in the game
fga	int	Field goals attempted in the game
fgp	float	Field goal percentage
three	int	3 points field goals in the game
threeatt	int	3 points field goals attempted in the game
threep	int	3 points field goals percentage in the game
ft	int	Free throws in the game
fta	int	Free throws attempted in the game
ftp	float	Free throws percentage in the game
orb	int	Offensive rebounds in the game
drb	int	Defensive rebounds in the game
trb	int	Total rebounds in the game
ast	int	Assists in the game
stl	int	Steals in the game
blk	int	Blocks in the game
tov	int	Turnovers in the game
pts	int	Points in the game
game_score	float	Game score of the player
minus_plus	int	Contribution to team points

6.2 Visualization of the data

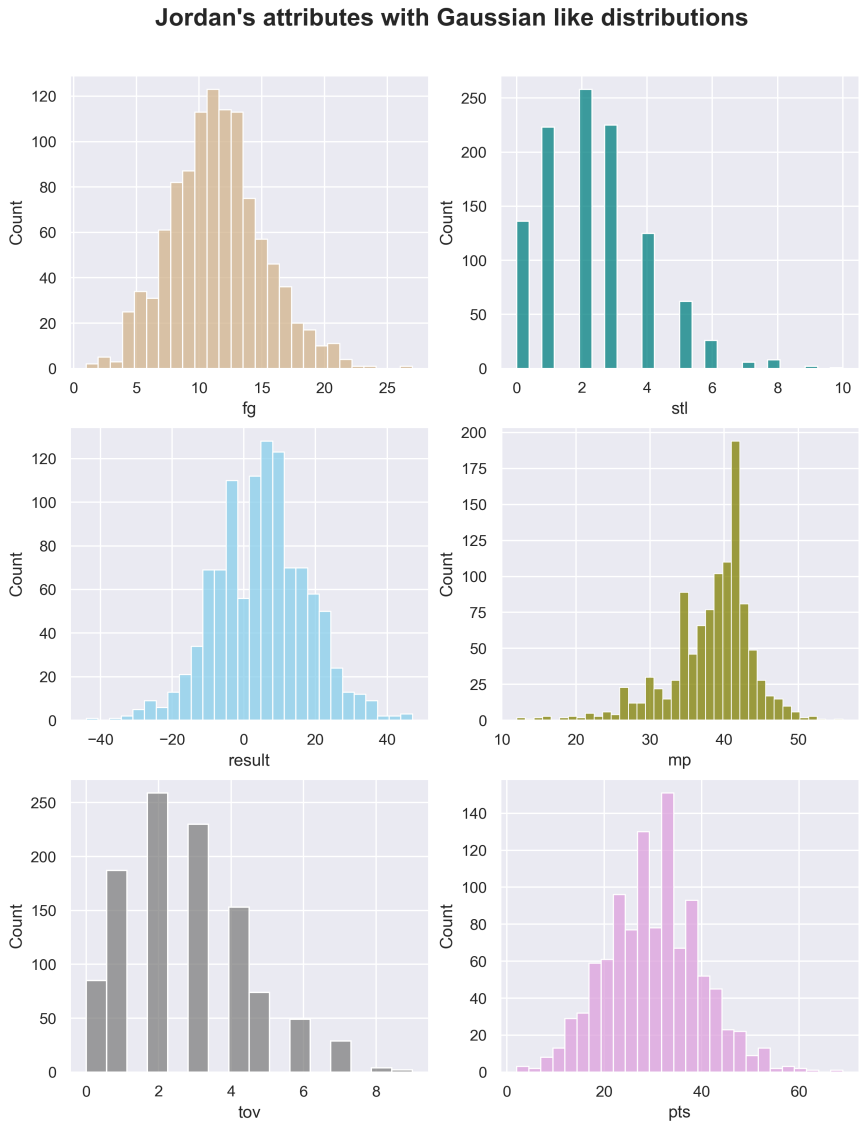


Figure 10: Gaussian like distributed attributes of Jordan's matches

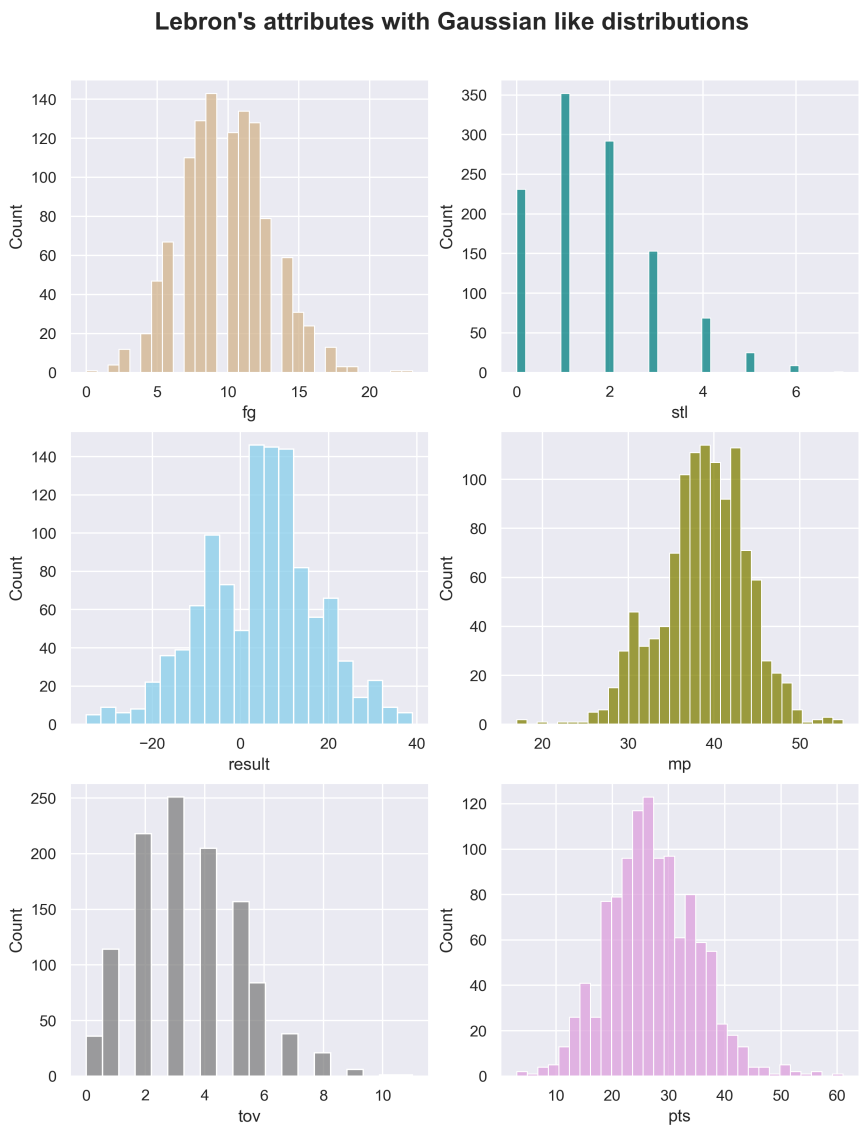


Figure 11: Gaussian like distributed attributes of LeBron's matches

6.3 Correlation *Heatmaps*

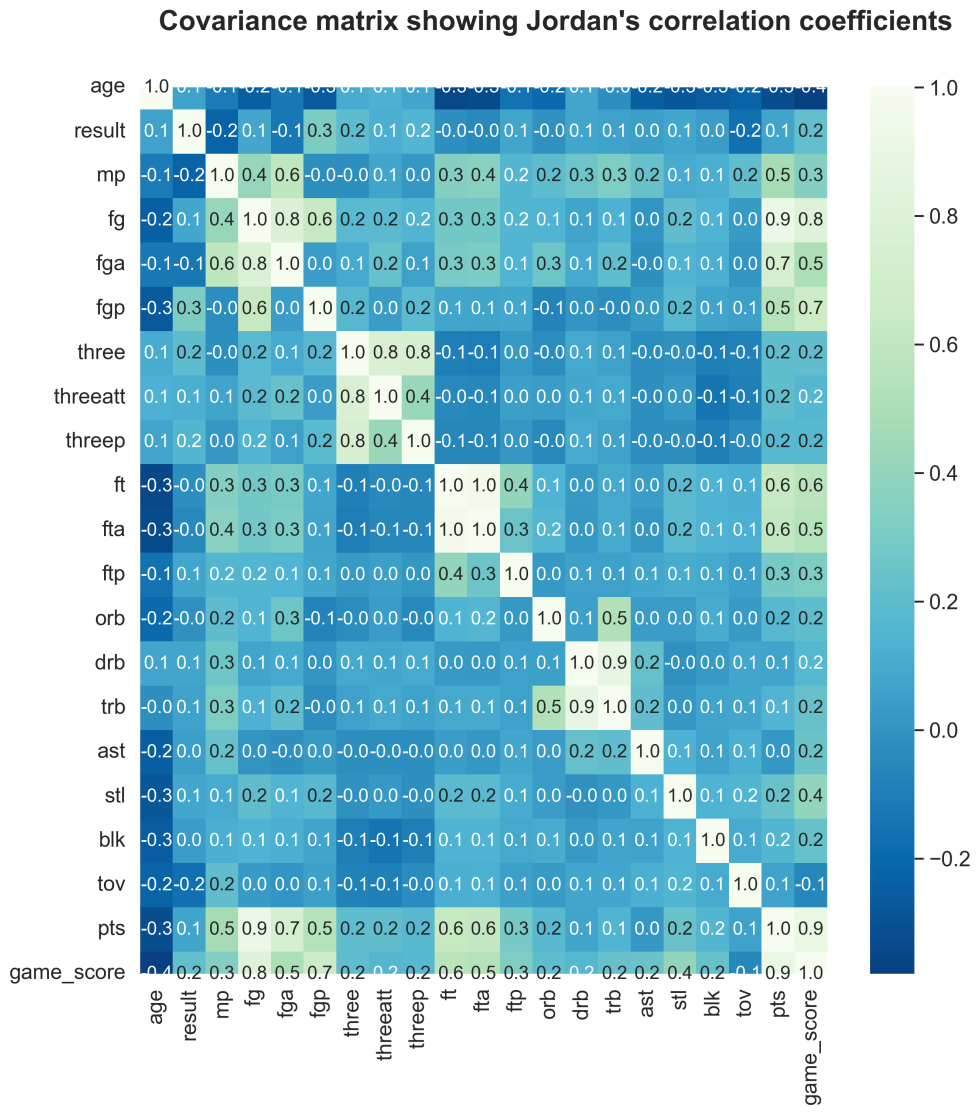


Figure 12: Visualization of Jordan's *heatmap* to observe the data correlation.

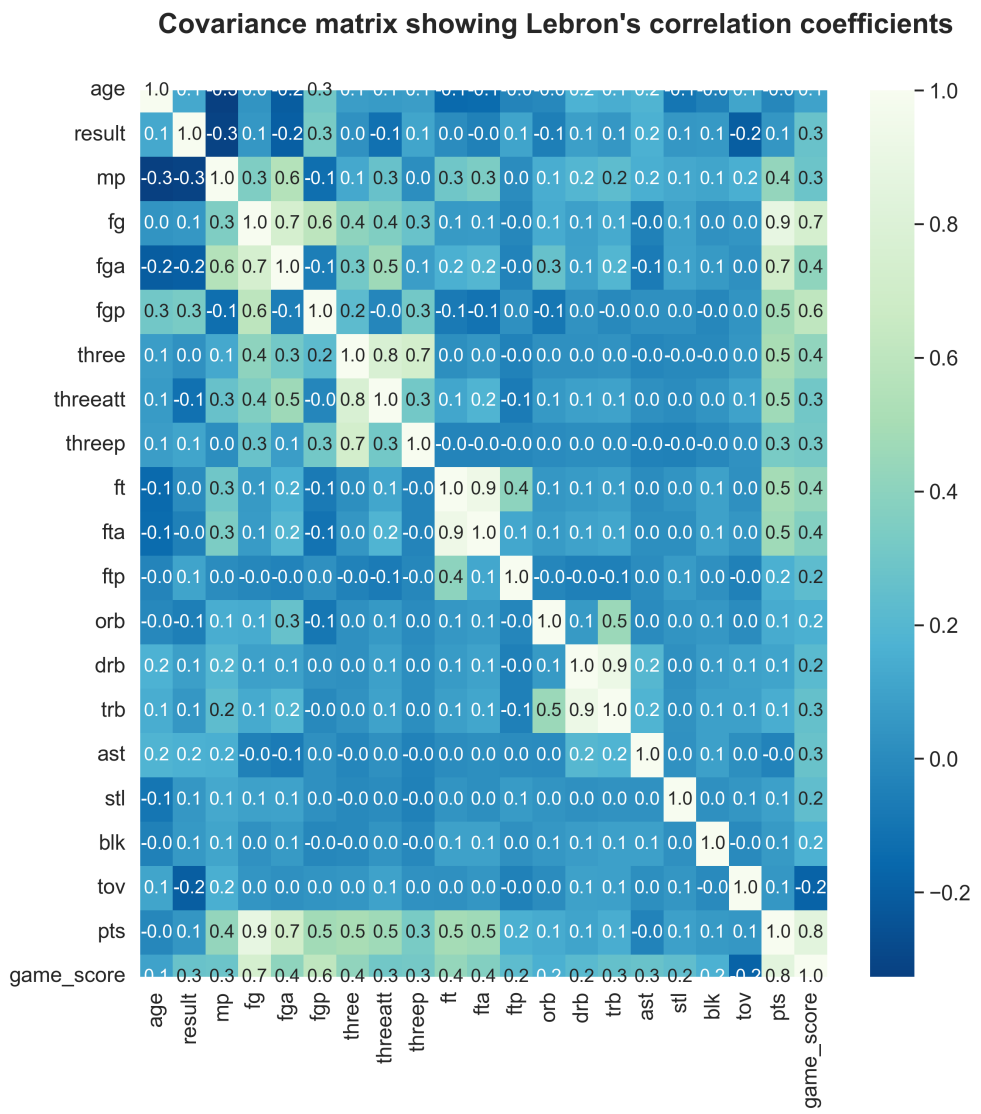


Figure 13: Visualization of LeBron's *heatmap* to observe the data correlation.

6.4 Multivariate Regressors

6.4.1 Model and data collected for LeBron James

- Multivariate model

$$gs_{pred} = -0.5401998904075498 + 0.0117163081 \cdot age + 0.000368018072 \cdot result$$
$$-0.00749322108 \cdot mp + 0.348127029 \cdot fg - 0.703409263 \cdot fga - 0.0779448289 \cdot fgp$$
$$-0.0584432123 \cdot three + 0.00690145526 \cdot threeatt + 0.0919804063 \cdot threep$$
$$+0.388994305 \cdot ft - 0.421185397 \cdot fta - 0.00400334723 \cdot ftp + 0.375885498 \cdot orb$$
$$-0.0361065719 \cdot drb + 0.339778926 \cdot trb + 0.706484830 \cdot ast + 0.993857592 \cdot stl$$
$$+0.712659727 \cdot blk - 1.03748551 \cdot tov + 1.02680515 \cdot pts$$

- Measurements and scores of the model

	Standardized Data	**Non Standardized Data**
Testing R2 Score	0.9954132723346405	0.9954132723346405
Testing MSE	0.0038455583174864636	0.23341290258702227
Training R2 Score	0.996233397280678	0.9962333972806781
Training MSE	0.004237097026837373	0.25717803084140395
Cross validation R2 Score	0.9948837067317874	0.9948837067317874
Cross validation MSE	0.0039954061658180415	0.24250818039530642

6.4.2 Model and data collected for Michael Jordan

- Multivariate model

$$gs_{pred} = -0.561635467854682 + 0.04581011 \cdot age + 0.00316263 \cdot result$$

$$-0.02762215 \cdot mp + 0.34371023 \cdot fg - 0.73175504 \cdot fga - 1.86309188 \cdot fgp$$

$$+0.07300259 \cdot three - 0.06799796 \cdot threeatt - 0.22033934 \cdot threep$$

$$+0.30467866 \cdot ft - 0.38266728 \cdot fta + 0.18582104 \cdot ftp + 0.37499931 \cdot orb$$

$$-0.03488572 \cdot drb + 0.34011358 \cdot trb + 0.69922455 \cdot ast + 1.01041767 \cdot stl$$

$$+0.72456767 \cdot blk - 1.01206531 \cdot tov + 1.06510171 \cdot pts$$

- Measurements and scores of the model

	Standardized Data	**Non Standardized Data**
Testing R2 Score	0.9963052176912673	0.9963052176912673
Testing MSE	0.0026931527806955623	0.24239841607495954
Training R2 Score	0.9967889565452204	0.9967889565452203
Training MSE	0.002827159144216972	0.2544597185359222
Cross validation R2 Score	0.9951397943608496	0.9951397943608495
Cross validation MSE	0.0038736985944607862	0.3486539680851687