SPORTS ANALYTICS

PROJECT FOR DSE 309

Manas Dubey

Roll no.: 19184

Department of Data Science and Engineering

IISERB

December 2021

Contents

ABSTRACT

As a spectator, the world of sports is that of adrenaline, mesmerisation and inspiration. When we see those immaculately coordinated passes, strategic advances and the passionate coaches shouting hysterically ,one would only wonder just how much preparation and planning has gone into it. Mustn't it be extremely pressurising for the coaches and players to rely on just raw talent, experience and training as the stakes go higher? The players have to be screened efficiently for better utilisation of talent.The sheer complexity of the number of variables under consideration is intimidating.What if there was a way to efficiently utilise and analyse this humongous data generated in sports, thus aiding coaches, managers and players in on-field and off-field analytics. Enter: Sports Analytics.

Keywords Regressions · NBA · Soccer · Pythagorean · Data Visualisations

1 Introduction

Sports analytics are a collection of relevant, historical, statistics that can provide a competitive advantage to a team or individual. Through the collection and analyzation of these data, sports analytics inform players, coaches and other staff in order to facilitate decision making both during and prior to sporting events. The term "sports analytics" was popularized in mainstream sports culture following the release of the 2011 film, Moneyball, in which Oakland Athletics General Manager Billy Beane (played by Brad Pitt) relies heavily on the use of analytics to build a competitive team on a minimal budget.

There are two key aspects of sports analytics — on-field and off-field analytics. On-field analytics deals with improving the on-field performance of teams and players. It digs deep into aspects such as game tactics and player fitness. Off-field analytics deals with the business side of sports. Off-field analytics focuses on helping a sport organization or body surface patterns and insights through data that would help increase ticket and merchandise sales, improve fan engagement, etc. Off-field analytics essentially uses data to help rightsholders take decisions that would lead to higher growth and increased profitability.I shall be demonstrating some instances of both On-field Analytics and Off-Field Analytics which are used extensively by sports organisations.

The first part is On-Field Anaytics. Initially, I demonstrate the implementation of Pythagorean Expectation which can be used as a tool for match win prediction in Soccer. The Second part is on Off-Field Analytics, in which i demonstrate an instance of Salary Prediction Model (Classification in ML)based on certain evaluative variables and using machine learning models ,gives a prediction of the salary that should be imbursed to a player.

2 Data

- The data that i used in the first sub-part of the On-Field Analytics section was that of the English premier league 2018-2019 season[1].
- The data used for the Salary Prediction Model in Off-Field Analytics part was from a database of approximately 50 statistics and salary information of every player dating back to 1950. But i chose to go with data from the 1995 season to avoid discrepancies.Also, I used the information for the total salary dissemination information from another source.[3]
- I used another source for getting the total no. of wins of the different teams.[4]

3 On-Field Analytics

3.1 Pythagorean Expectation

The Pythagorean expectation is an idea devised by the famous baseball analyst, Bill James[5], but it can in fact be applied to any sport. In any sports league, teams win games by accumulating a higher total than opponent. In baseball and cricket the relevant totals are runs, in basketball it is points, and in soccer and hockey it is goals (by “hockey” we mean here what the world outside of the US and Canada usually calls ice hockey, but in fact the same is true in field hockey). This is a concept which can help to explain not only why teams are successful, but also can be used as the basis for predicting results in the future.

3.2 Pythagorean Expectation in Soccer

The Pythagorean expectation can be described thus: in any season, the percentage of games won will be proportional to the square of total runs/points/goals scored by the team squared divided by the sum of total runs/points/goals scored by the team squared plus total runs/points/goals conceded by the team squared.In soccer, teams score goals, and we can calculate Pythagorean Expectations based on goals scored and goals conceded.

$$wpc = TF^2/(TF^2 + TA^2) \tag{1}$$

Where TF is the goals scored and TA is the goals conceded.

3.3 Implementation on real data

The procedure that I followed is threefold:-

1. Data re-structuring
2. Calculating the Pythagorean Expectation
3. Regressional Analysis

3.3.1 Libraries

The libraries that i used in this subsection were Numpy and Pandas for basic arithmetic operations and data manipulations through CSV files. I used statsmodels for providing statistical summaries of my regression. Finally I used Seaborn,a python library built on top of matplotlib for running smooth and concise linear regressions.

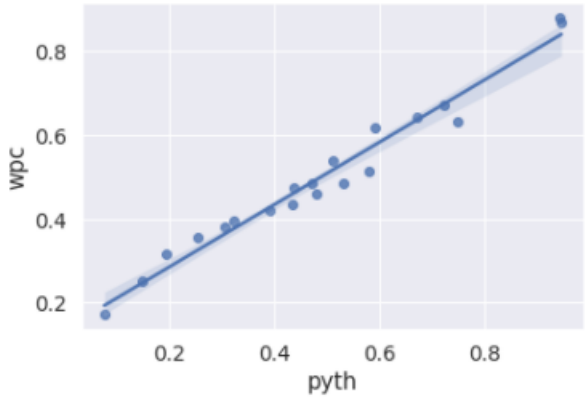
3.3.2 Data re-structuring

After loading the data-set ,i needed to organise it such that I have a metric for the number of wins and losses by a team. Therefore, i set up a count based on number of wins scored by a team as a home team and as an away team and store it into variables *awinvalue,hwinvalue* respectively, assigning 1 for win and 0.5 draw. Now, I created separate columns for these two values, I will now add up the two values for each team ,thus fetching me a total wining points.

3.3.3 Calculating the Pythagorean Expectation

Moving forward, I created separate columns for total goal scored and total goals conceded, merging the datasets for the goals as an away and home team.After which, I proceeded to calculate the wining-percentage for each team and stored it in a separate column.Finally, I used the formula described in section 3.2 to calculate the Pythagorean expectation(*pyth*) and stored it in a separate column.

3.3.4 Regression Analysis



Now that i have everything, i can now run a regression between *wpc* and *pyth* .The regression output tells you many things about the fitted relationship between win percentage and the Pythagorean Expectation. Regression is a method for identifying an equation which best fits the data. In this case that relationship is:-

wpc = Intercept + coef * *pyth*

You can see the value of Intercept is 0.1354 and coef is 0.7472. It's this latter value were interested in. It means that for every one unit increase in *pyth*, the value of *wpc* goes up by 0.7472.

Two other points to note:-

(i) The standard error (std err) gives us an idea of the precision of the estimate. The ratio of the coefficient (coef) to the standard error is called the t statistic (t) and its value informs us about statistical significance. This is illustrated by the p-value (P > |t|) - this is the probability that we would observe the value 0.7472 by chance, if the true value were really zero. This probability here is 0.000 - (this is not exactly zero, but the table doesn't include enough decimal places to show this) which means we can confident interval is not zero. By convention, it is usual to conclude that we cannot be confident that the value of the coefficient is not zero if the p-value is greater than 0.05.

(ii) The R squared statistic tells you the percentage of variation in the y-variable (wpc) which can be accounted for by the variation in the x variables (pyth). R-squared can be thought of as a percentage - here the Pythagorean Expectation can account for 74.72 percent of the variation in win percentage.

4 Off-Field Analytics

4.1 Player salary prediction model in NBA

4.1.1 Procedure:-

The following was the procedure I used to develop a generic model:-

1. Discover which statistics are the best predictors of an NBA player's salary.

2. Use a machine learning model to predict NBA salaries.

3. Determine which players have been overvalued and undervalued according to their given vs. predicted salary.

4. Evaluate the dependence of Salary on team performance.

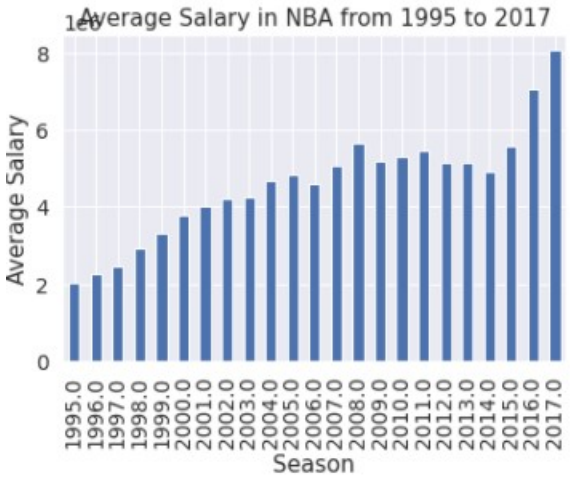
4.1.2 Libraries

Initially, the libraries used in this section are numpy and pandas for basic CSV file and data manipulations. Apart from that i used matplotlib and seaborn for using data visualisation tools such as heat-maps and histograms. For the classification part, we use a plethora of regression, cross validation, train-test functions from scikit-learn library.

4.1.3 Data cleaning and pre-processing

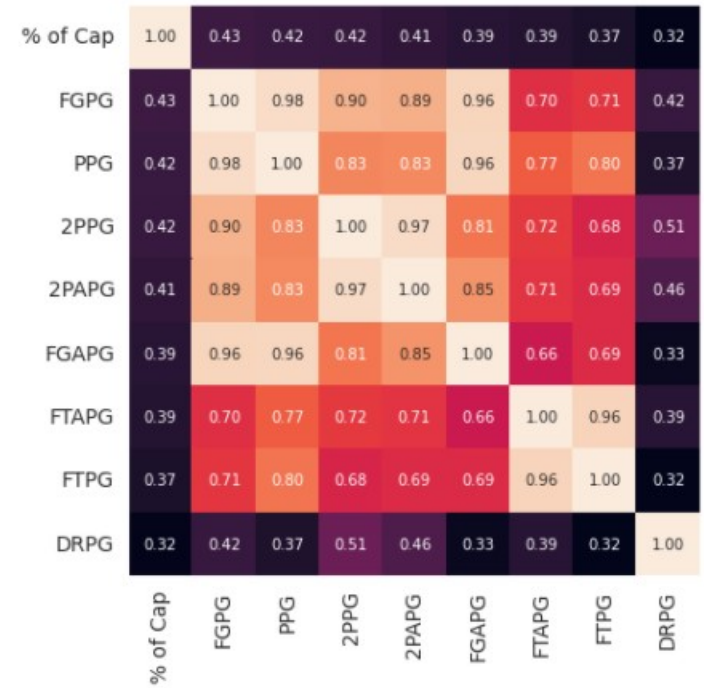
Initially, the total statistics like minutes played or total points scored were replaced by their per game equivalents to normalize the statistics, assuming that the player played more than a certain number of games that season (if not, he was removed from the dataset). Additionally, there were a number of players who did not have salary information listed for that season and were removed from the data set.

Obviously ,the average NBA salary has drastically increased over the past 20 years. Therefore,The solution was to normalize salary data by putting it as a percentage of the league's salary cap, the total salary limit that a team can spend on its players in a given season. The salary cap has risen as player salaries have.



4.1.4 Feature Selection

The next question that emerges is to select the best features to run a regression upon, which can predict the players salary efficiently. For that, I used a correlation heat map for the 8 statistics that had the highest Pearson r^2 values with salary.



4.1.5 Creating the prediction model using ML

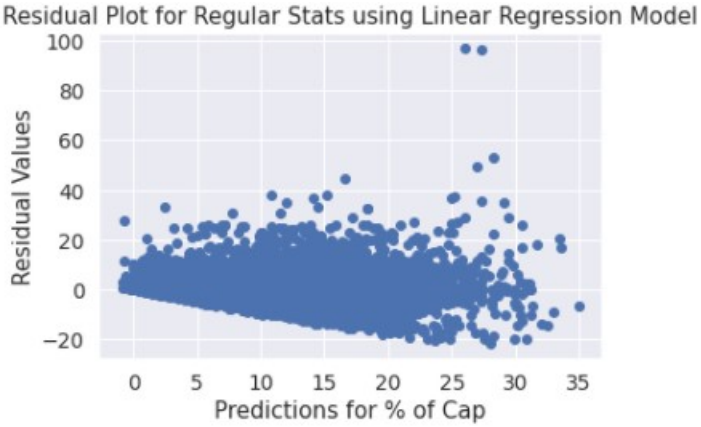
After I had identified the 8 most important variables based on correlation.I proceeded with creating an annotated data-set for the target array which was the percentage capitalisation row vector and the regressors which were *PPG, 2PAPG,FTAPG,DRPG* through the training of which i would be predicting the pecentage cap salaries of the individual players.

Now, I split the data set in Train and Test sets. The training set was 80 percent and the test set was 20 percent of the data-set.After which, I ran the regression and the consequent predictions were stored in a variable *predictions*.

4.1.6 Performance Analysis

I calculated the accuracy of my classification. Both by using the .score() function and by K-fold Cross Validation. The accuracy for the latter was around 47 percent. Which is a relatively poor classification performance. I also calculated the performance metrics for the linear regression and checked the RMSE and R-squared values.

It seemed that there was a huge error between the real salaries of the players and the salaries I predicted. I demonstrated this by creating a separate variable called the Residual .i.e the true salary - the predicted salary.And then i plotted the residuals against the predictions for percentage caps.



4.1.7 Why the low accuracy?

Now the reason for the under-prediction could be the following:-

1. Multi-collinearity in the regressors I chose which might have diminished the individual contribution of the variables towards the regression.

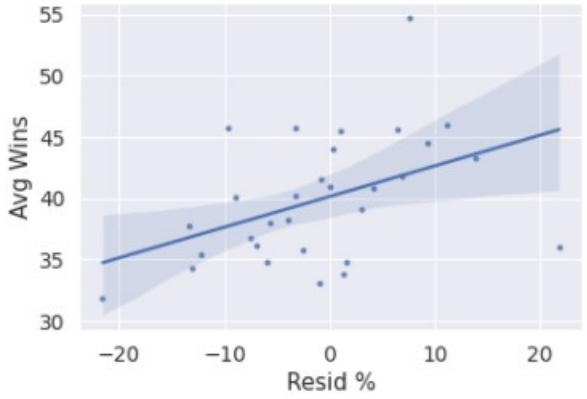
2. Existence of over-valued and under-valued players. As the fame and experience and approval ratings of a player accretes, the players start to be paid a lot more(even if their performance remains the same).Whereas the more inexperienced rookies settle for a lesser pay.

This is the reason for the high deviations in my data and thereby, fetching me lesser accurate predictions. I have demonstrated this by creating a list of the top 25 overvalued and undervalued players.

4.1.8 Valuation of players vs. Team performance

Finally, I proceeded to asses if there was any correlation between the valuation of players to the teams they played for with the teams performance in a season(based on the number of wins). I grouped the players by team, calculated the average percentage per team based on average actual percentage of cap spent per player vs. average predicted percentage, and added the average amount of wins over the past 22 years as an additional category.

From there, I created a scatter plot of residual percentage vs. average number of wins per team. From the scatter plot,one can observe that the teams that overvalued its players generally had more success than the teams that undervalued them. Therefore, *the coaches oughta pay em ballers more!*



5 References

1. https://salcorpenterprise.com/calculating-pythagorean-wins-for-nfl-teams-using-python/
2. https://github.com/joshrosson/NBASalaryPredictions/blob/master/FinalProject.ipynb
3. https://www.coursera.org/programs/iiser-bhopal-on-coursera-wi792?currentTab=CATALOG

6 Citations

[1] Githubusercontent.com. [Online]. Available: https://raw.githubusercontent.com/Dubeman/Python-project-CSV-files/main/Engsoccer2018-19.csv. [Accessed: 05-Dec-2021].

[2] Githubusercontent.com. [Online]. Available: https://raw.githubusercontent.com/joshrosson/NBASalaryPredictions/master/NBAdata.csv. [Accessed: 05-Dec-2021].

[3] "NBA salary cap history," Basketball-reference.com. [Online]. Available: https://www.basketball-reference.com/contracts/salary-cap-history.html. [Accessed: 05-Dec-2021].

[4] "NBA League Averages - franchise win totals," Basketball-reference.com. [Online]. Available: https://www.basketball-reference.com/leagues/NBA$_w ins.html.[Accessed: 05-Dec-2021]$.

[5] Wikipedia contributors, "Pythagorean expectation," Wikipedia, The Free Encyclopedia, 06-Nov-2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Pythagorean$_e xpectationoldid = 1053789332.[Accessed: 05-Dec-2021]$.