Using Linear and Huber regression to predict individual basketball player's points scored in a game

1. Introduction

Data has been collected from National Basketball Association (NBA) games throughout the 80-year history of the league. As the amount of different data collected from NBA games continues to increase, the role of data analysis in teams' and players' success becomes more crucial. Teams have been hiring data scientists for the last two decades to analyze this data and to help them gain competitive advantage. Data science and machine learning can be used to make players make more effective plays on the court, predict the outcome of games and even determine the salaries of professional players. [1]

The goal of this project is to predict the number of points scored in an NBA-game by a single player, Stephen Curry. Stephen Curry is regarded by many as the greatest shooter in the history of basketball and one of the best players in the world today. He has played his whole professional career (2009 - present) for the Golden state Warriors. The project outcomes could by used for example by a basketball player, coach, bettor, fantasy league-player or just a fan of basketball.

The sections of the paper are as follows: **Problem formulation**: data, features, label, outliers, **Methods**: regression models, loss function, **Results:** test accuracy, **Conclusions:** assessment, future, **References, Appendix.**

2. Problem formulation

A data point is a single basketball game. More closely, a game contains information on individual player's performance like for example points (PTS), rebounds (REB) and assists (AST) by a player. Below is an example of some of Stephen Curry's statistics in a single game:

Season_ye	Season_di	Date	OPP	Result	T Score	O Score	MIN	FGM	FGA	3PTM	3PTA	3P%	REB	AST	PTS
2009-2010	Regular	Wed 4/14	POR	W	122	116	48	13	25	4	6	66.7	9	8	42

The dataset is collected from Kaggle [2].

Features

For features there are multiple candidates. I am a basketball player myself and have been following the NBA for many years and therefore used my domain knowledge to determine some of the features:

1. **Opponent:** Some opponents might have players playing in the same position as Stephen Curry who are particularly difficult or easy to score against. Also, the arenas of certain teams might have some emotional factors in increasing or decreasing player's performance.
2. **Point average**: The average number of points scored per game in the ongoing season so far. This is perhaps the most obvious one, as in principle points per game tells if the player scores a lot or not.
3. **Last n games' average points and made 3-point shots**: These features try to take into account the player's current form. Players are humans and might get hot or cold on occasion. My intuition is that a streak of extremely well or badly played games can last somewhere between 3 to 10 games. The average number of 3-point shots are an important factor in Stephen Curry's point production since throughout his career, he's gotten 47% of his points from behind the three-point line [3].

Features: 1. LAST_N_3PTM (float), 2. LAST_N_PTS (float), 3. SEASON_AVG (float) and 4. Opponent (OPP) (string).

Label: Points scored in a game (PTS) (integer).

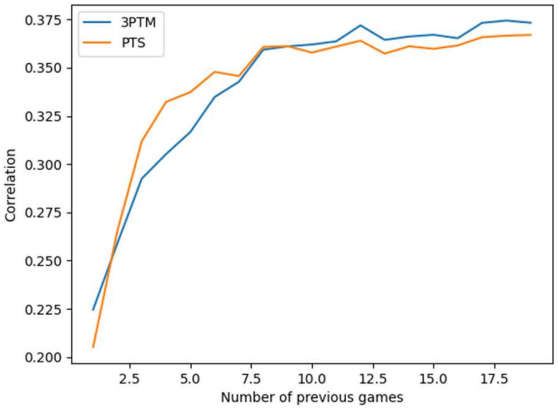

Figure 1: Correlation of n previous games

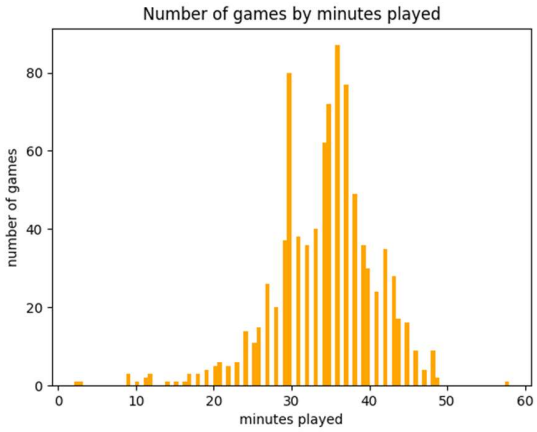

Figure 2: Games by minutes played

To determine the number of games a hot/cold streak lasts, a diagram is plotted in figure 1. The diagram presents the correlation of 3-point shots made and points scored in the last n games to the label. The correlation seems to increase the fastest when n -> 9. For this reason, nine is used as the number of previous games considered in creating features: LAST_N_3PTM and LAST_N_PTS.

Feature opponent

There are 30 teams in the NBA. This means that Stephen Curry has had 29 opponents to play against. The OPP-feature is currently presented as a string indicating the team's abbreviation. Two approaches to encode the categorical string-data into numerical are tested:

1. One-hot encoding:
   This is an approach used to encode nominal categorical data [4]. Nominal data means that there is no ordering between values. In one-hot encoding a binary value is added for each unique categorical value in the feature. This means that from the feature Opponent, becomes 29 new features (one for each opposing team), each including one 1 and 28 0's.
2. Numerical encoding:
   In numerical encoding each unique category gets an integer value. 29 teams -> 29 values (i.e., from 1 to 29). Numerical encoding assumes that the feature has ordering between values. For this approach I ordered the opposing teams based on the average number of points Stephen Curry has scored against them. Numerical encoding doesn't increase the number of features.

The feature Opponent can be considered as ordinal or nominal depending on the point of view. Some of the teams can be difficult to rank throughout a span of over Curry's career, but on the other hand, ranking them based on the points scored against them by Curry is rather simple. By comparing the regression results using these two approaches, approach 2 proves to give a lower error rate and therefore numerical encoding will be used.

Outliers:

Games where there is an unusual number of minutes played, which usually occur due to an injury or multiple overtimes, are considered as outliers in the data. Minutes played heavily affects points scored. The outliers can easily be found from a histogram presented in figure 2, indicating the number of games when a certain number of minutes have been played. From the original data, games with less than 15 - or more than 50 minutes played are filtered out. This results in removal of 25 datapoints. After data cleaning, the dataset contains **856 data points**.

3. Methods

A player's performance depends on multiple factors such as the player's social life, how well rested the player is, the player's mental and physical health and also some unknown factors. The fact that all players are humans already makes predicting their performance extremely difficult. Some of these are impossible to measure and only very few of the measurable factors are publicly available.

For handling the data I used Pandas-library [5], for the visualizations Matplotlib.pyplot [6] and for some mathematical operations I used Nympy [7]. Machine learning models, train-test splitting, loss-function and PCA are from Sklearn [8]. Otherwise, the methods used are normal Python methods.

I'm using 4 features to determine the number of points scored. To visualize the relationship between features and label, I used principal component analysis to reduce the number of features from four to just one, which however, retains most of the data's variance.

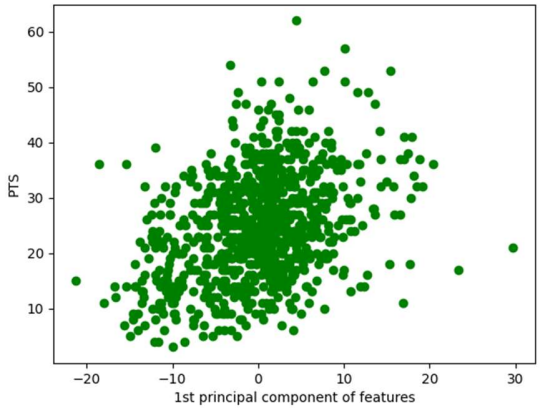

Figure 3: Relationship between features and labels

Model	Linear	Huber
Training error (MAE)	6.787	6.775
Validation error (MAE)	7.144	7.169

Table 1: Training and validation Results

As can be seen from figure 3, although, there is a lot of variance, a linear relationship exists. The first approach is to use SKlearn's **Linear regression** to capture that linear correlation.

The selection of the other machine learning model is also based on the shape of the data. Even though the largest outliers have been filtered out from the data in the beginning, in figure 3 can still be noticed multiple points outside of the cluster. These are not considered as outliers from a basketball player's point of view since it is completely natural for a player to have even vastly inconsistent performances. But a model could still "see" these points as outliers. To reduce the effect of the outliers, **Huber regression** is used. Huber regression is a method robust towards outliers and it takes as a parameter a value that controls the number of samples that should be classified as outliers [9]. Multiple parameter values were tried, and the one producing the best results was chosen.

The dataset is split into training, validation and testing sets with respective ratios of 0.6/0.2/0.2. This is a quite basic split [10], but I also experimented with splits of different size and found this to be close to optimal.

To evaluate the performance of the models on the training and validation sets, mean absolute error (MAE) is used due to it being unbiased towards both models. For example, mean squared error can be seen as more biased towards linear regression as that is what linear regression is trying to minimize [11]. Mean absolute error as the name states, calculates the mean of the absolute values of each point in the dataset subtracted by a corresponding point in the predicted set. Using MAE also gives a good idea of how accurate the model is in terms of basketball points.

4. Results

Table 1 shows the two machine learning models and their training - and validation error on the same datasets using the same split of 0.6/0.2/0.2. Huber regression didn't improve or decrease the results significantly. My intuition is that the outliers are spread almost evenly -> as many exceptionally low - as high scoring numbers, and that is why the results didn't change much. In both cases, both the margin between training and validation error is small so almost no overfitting happens.

Linear regression is selected as the final model as the validation and training errors are closer to each other, although the difference to Huber regression is negligible. Mean absolute error is once again used to evaluate the final model's performance on the unseen test dataset. The result is: MAE = **6.645.** This is a lower error than even training error, which sounds rather strange as the model is fitted particularly to predict the training data. However, I think this can be explained by the small number of datapoints. In other words, in the test dataset there were most likely not as varying values as in the other two sets. When there are only 856 datapoints, differences in the different datasets don't get completely evened out.

Stephen Curry's point average throughout his career is 24.3 points. I observed that values closer to that get predicted quite accurately, but when he scores a particularly high number of points, the error becomes larger. Below is a table of the predicted versus true labels.

Truth	15	26	23	29	43	41	25	29	27	19	21	35	24	10	13
Predicted	17.77	27.01	19.3	22.14	25.22	24.68	20.85	31.76	25.64	27.57	27.11	20.27	26.51	17.49	16.63
Error	2.77	1.01	3.7	6.86	17.78	16.32	4.15	2.76	1.36	8.57	6.11	14.73	2.51	7.49	3.63

5. Conclusions

The results are not extremely accurate. However, considering the margins of human error, variance and randomness which are huge in case of data that is dependent on human performance, I would say that the result is not bad by any means. The result could be improved in the future by using more features and adding more data points by inspecting a more experienced player, multiple players or by upsampling. Also, the first couple of seasons of Stephen Curry's career he was not as consistent as he is today, so filtering out the first seasons could improve accuracy. From the publicly available data, new features to use could be for example the form of opposing team or teammates. An NBA teams' staff however have access to a larger number and far more advanced data than the public so they can make more accurate predictions [12].

The scope of this project could be changed for example to predict whether a player reaches their scoring average in a game or not, which is a common target for betting on basketball games [13]. Another option would be to predict team's performance instead of just individual player's.

The number of different ways to use machine learning in basketball is unlimited. Data science is already an important area of the game, but the surface of its true potential has just been scratched. [1]

References

1. randerson112358. How The NBA Uses Data & Analytics. Medium, 2018. https://randerson112358.medium.com/how-the-nba-uses-data-analytics-6eac3c43a096
2. Mujin Jo. Stephen Curry stats 2009-2021 in NBA. Kaggle, 2022. https://www.kaggle.com/datasets/mujinjo/stephen-curry-stats-20092021-in-nba
3. Basketballreference, 2022. https://www.basketball-reference.com/players/c/curryst01.html
4. Jason Brownlee. Ordinal and One-Hot Encodings for Categorical Data. Machine Learning Mastery, 2020. https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/
5. Pandas. https://pandas.pydata.org/docs/index.html
6. Matplotlib. https://matplotlib.org/3.5.1/tutorials/introductory/pyplot.html
7. Numpy. https://numpy.org/doc/stable/reference/index.html
8. Sklearn. https://scikit-learn.org/stable/modules/classes.html
9. Jason Brownlee. Robust Regression for Machine Learning in Python. Machine Learning Mastery, 2020. https://machinelearningmastery.com/robust-regression-for-machine-learning-in-python/
10. Rachel Draelos. Best Use of Train/Val/Test Splits, with Tips for Medical. Glass Box, 2019. Datahttps://glassboxmedicine.com/2019/09/15/best-use-of-train-val-test-splits-with-tips-for-medical-data/
11. Introduction to linear regression. Top coder, 2021. https://www.topcoder.com/thrive/articles/introduction-to-linear-regression
12. The NBA Data Scientist. Bloomberg, 2019. https://www.youtube.com/watch?v=MpLHMKTolVw
13. Zovak et al. Game-to-Game Prediction of NBA Players' Points in Relation to Their Season Average. IEEE, 2019. https://ieeexplore.ieee.org/document/8756733

Appendix

The code of this project can be found from: https://github.com/nikulam/Machine-learning-project2