Time Series Forecasting of NBA Players' Statistics

Ayush Oza, Janelle Samar-Brizuela

1 Introduction

Deep learning models in sports analytics generally explore specific player actions to predict the outcome of games. There is a lack of research into predictive models for longer term player performance and outcomes. For instance, player careers in the NBA have long-term temporal dependencies that can be extracted in terms of statistical features through neural networks.

In this report, we aim to generate monthly projections for a chosen set of statistics for the next 2 seasons based on player performance in the previous 2 seasons they have played in. The statistics we use and predict are unaffected by team statistics and based solely on player performance. Although it is possible to use varied length input by entering the player's full career, the lack of data justifies data augmentation and windowing with fixed length inputs, as explored further in the data processing section.

Transformers have shown great promise recently over RNN-LSTMs in sequence-to-sequence prediction models through the introduction of self-attention. We use transformers for monthly time-series forecasting for each player with a set of 12 statistics. Our model makes use of the complex nature of the data in terms of feature dimensions to achieve realistic predictions of player growth and career potential.

2 Illustration

Fig 2.1 roughly illustrates how the Transformer model works: the input goes through Positional Encoding to retain information about the order of the sequence (i.e. the sequence of player statistics throughout 2 seasons), then it is taken through an encoder layer which has its own self-attention layer that allows inputs to interact with each other while encoding the sequence, then it is taken through a feed forward neural network and then passed to the decoder which contains the same two layers.

It is very similar to a seq2seq model in the sense that both use encoders and decoders, however Transformers rely on self-attention to model dependencies between different parts of the sequence instead of attention.

Fig 2.1 Transformer Model

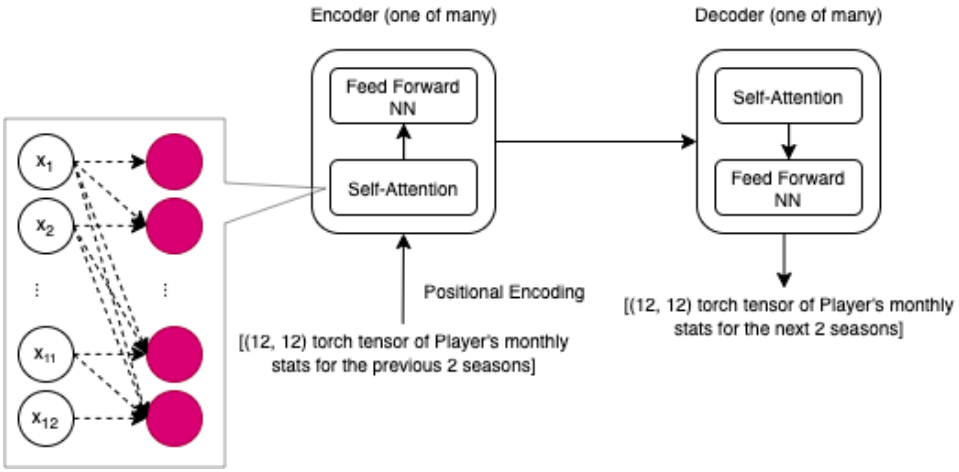
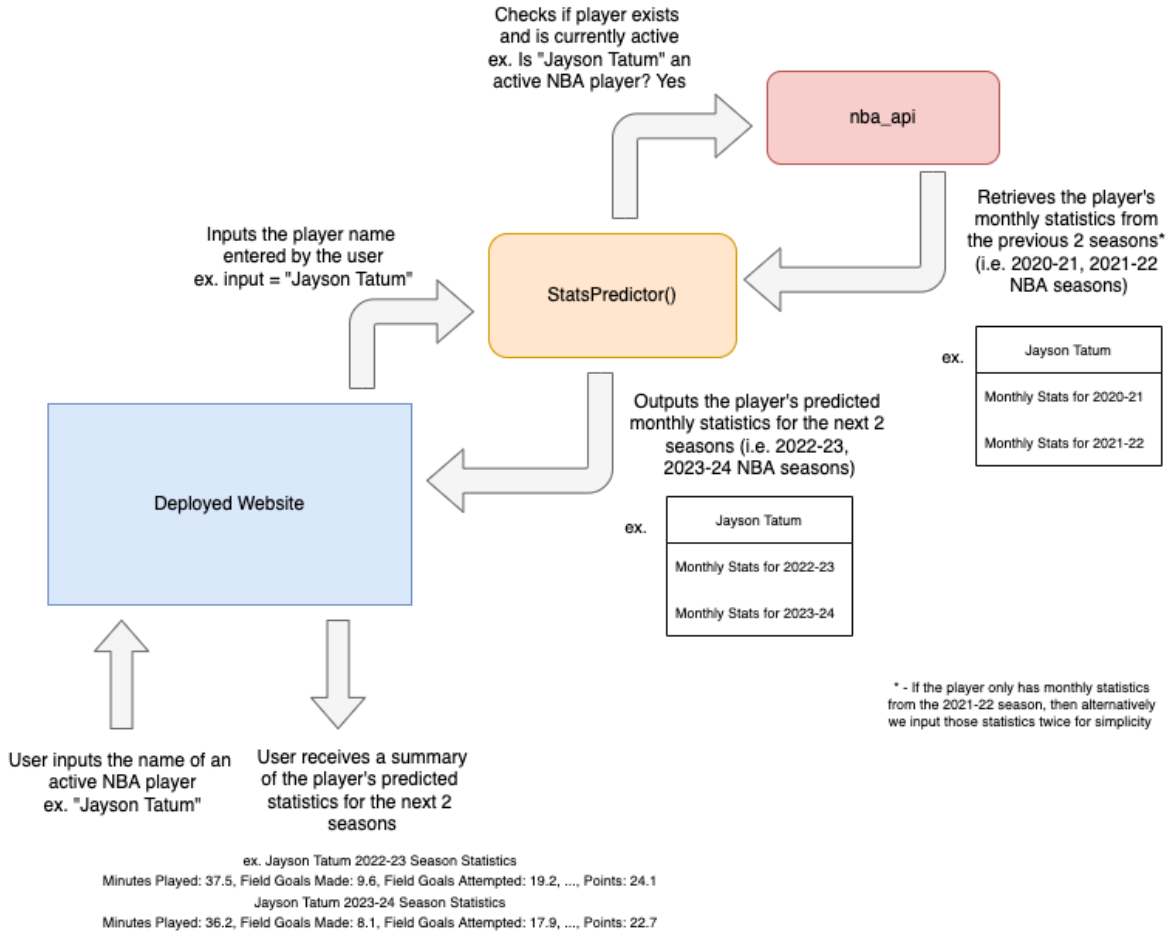


Fig 2.2 is an illustration of the expected flow of the model. The user will input an active NBA player's name into the deployed website (i.e. A player that is currently playing in the league), and the website will send that name into the StatsPredictor object (class used to predict player statistics using the trained Transformer model) which will then use nba_api to check if that name is the name of a current NBA player. If it is confirmed, the StatsPredictor object will retrieve the player's monthly statistics from the previous 2 NBA seasons as input and compute the predicted 2 following NBA seasons. The user will then receive a summary of those predicted statistics for the desired player.

Fig 2.2 Expected Model Flow with an example



3 Background and Related Work

Most previous work concerned with player performance in sports has frequently looked at short-term outcomes such as game results. However, predictive models that project statistics for a longer period of time have generally applied RNN-LSTM approaches. In comparison, the transformer model has shown significantly better performance for time-series forecasting problems through mitigating long dependency issues with self-attention where each time step has direct access to all previous time steps.

Benavidez et al., 2019 [1], researchers at Stanford, look at a similar player performance projection model in baseball as opposed to basketball. The multivariate LSTM model, the best performing model in the paper, learns yearly offensive "hitting" metrics for baseball players to predict the OPS (on-base plus

slugging) for the next year. Other approaches are linear regression and SVM models that fail to consider dependencies from one time step to another. The loss metric used is MSE, similar to the model we implement. The paper concludes that the poor performance of the models is due to insufficient features and discusses that improvements could build on different player career trends. Based on this, we incorporate statistics that are more closely related with each other and remove redundant features. Additionally, the nature of the problem involves predicting the future, which is technically impossible but developments on existing methods can have more accurate player projections.
The paper by Devarapalli et al., 2018 [2] explores game-to-game NBA player performance using deep learning and more closely relates to our problem. The report explores a RNN-LSTM model that takes in 18 player statistics for 10 games in the 2016-17 season to predict a score reflecting a player's value in fantasy NBA. The results are enhanced by using a neural network on top of the LSTM model to provide context to predictions. The paper recommends consideration of more training data including previous season data in future work, which we accommodate in our model.

4 Data Processing

Our data is primarily collected through the nba_api [3], which leverages statistics from nba.com that can be accessed through different endpoints of the API. The list of statistics we use can be found in Fig A.1 in the Appendix. We use 2 endpoints: PlayerCareerStats and PlayerGameLog to retrieve player statistics for each game they've played in.

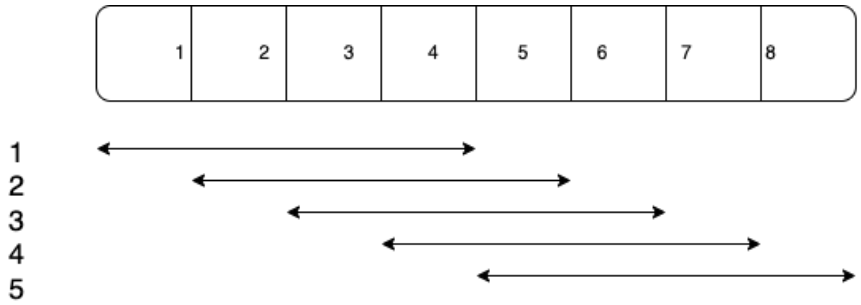
The steps are listed as follows with the relevant Python scripts identified:

1. Retrieve player information (player_id, player_name, seasons_played) from the PlayerCareerStats endpoint. Included in data_collection/season_data.py.
2. Remove all seasons before 1973-74 and any careers less than 4 seasons long. Also included in data_collection/season_data.py.
3. Collect game-to-game statistics using player_id from PlayerGameLog endpoint. Included in data_collection/monthly_data.py.
4. Compute monthly averages for each player and all statistics. Also included in data_collection/monthly_data.py.
5. Standardize all values by subtracting the mean and then dividing the standard deviation. Included in data_collection/standardizing.py.

*Results of all Python scripts are in csv format.

Given there are approximately 4374 all-time NBA players, the dataset of player careers is limited. Therefore, we augment the data by turning player careers into windows of 4 seasons with a stride of 1 season (or 6 months). Fig 4.1 demonstrates the windowing concept on an example career of 8 seasons.

Fig 4.1 Windowing of 8 season career



We split the data with a simple 60% training, 20% validation and 20% testing proportion. An important note here is to avoid cross-contamination of data; the data is split in such a way that two windows of a specific player's career do not appear in two different sets.

After data augmentation and data split, the statistics for the data are as follows:

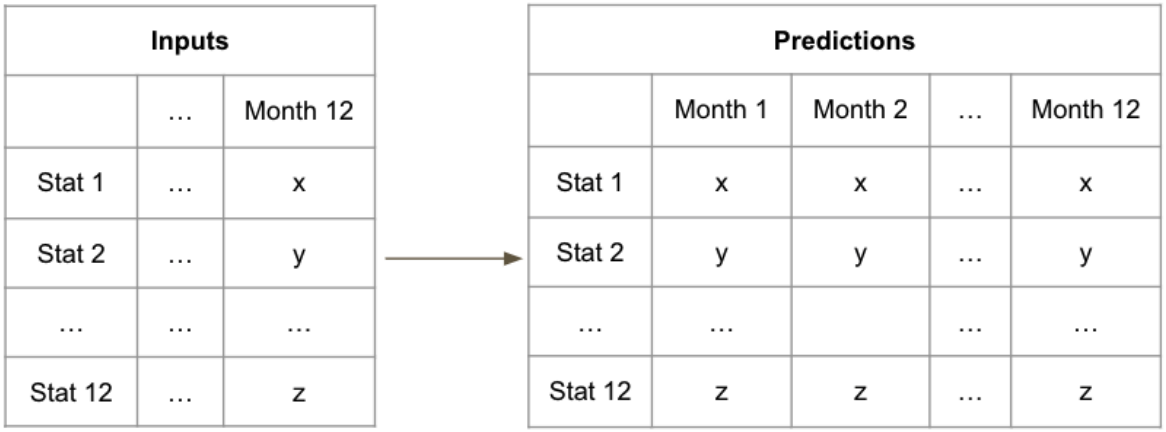
- 9439 total data points (2718432 statistics, 226536 months, 3667 players)
    - Average career length: 6.574 seasons (39.4 months)
- Training set: 5795 data points (11.8MB)
- Validation set: 1826 data points (3.7MB)
- Test set: 1818 data points (3.7MB)
- Standard data point size: 4 seasons (24 months)
    - Input size: 2 seasons (12 months)
    - Label size: 2 seasons (12 months)

Each data point is a 24 (months) by 12 (statistics) matrix where the first 12 (months) by 12 (statistics) is the input to the model and the next 12 (months) by 12 (statistics) are the labels. For a clear insight, we provide an example data point in data_processing/datapoint_example.txt.

5 Baseline Model

We design a baseline model as a basis for the performance of the final model. The baseline approach taken is previous month's statistics. It is simple and does not require machine learning. The model uses a player's last month's statistics in input data (month 12) as predictions for the statistics in the upcoming 2 seasons or 12 months of the player's career. In sports, especially basketball, consistency in player performance is visible through their career and most frequently, their statistics for the "next" season are similar to the statistics for the previous season they played in (not the case for injury-prone players and shortened seasons).

Fig 5.1 Baseline model interpretation



With the MSE loss as the metric used to evaluate the performance of the model, we find the loss for each statistic independently and compute the mean MSE loss across the standardized test dataset. Fig 5.2 illustrates the results of the baseline model.

Fig 5.2 MSE loss for each statistic

Statistic	MSE
MP	0.7061250296953756
FGM	0.5258116721053107
FGA	0.472193456693931
FTM	0.6031856182503763
FTA	0.5816796654267539
REB	0.5170435450337901
ASS	0.3715254132220216
STL	0.9520204722885331
BLK	0.5799924219484872
TOV	0.7827062129906852
PF	0.9592876724068834
PTS	0.49608033024651843
Mean MSE (for all statistics)	**0.628970959192389**

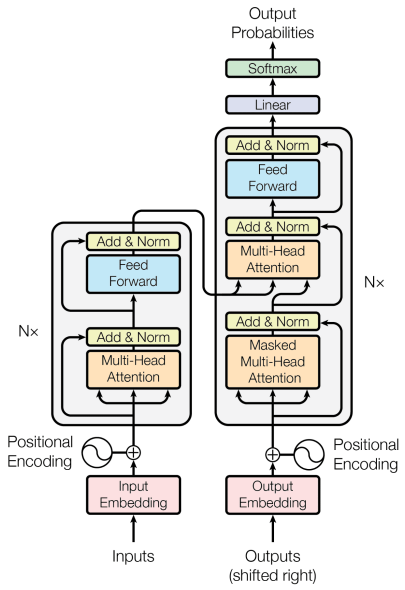
The final MSE for the baseline, ~0.63, acts as a minimal threshold for the final model as explored in the following sections.

Baseline model can be found in data_processing/baseline.py.

6 Architecture

As mentioned, we implement a transformer model for the sequence to sequence prediction. The model architecture is based around the initial proposal of the Transformer model by Vaswani et al [4]. The inputs to our model are the 12-month sequence and the attention mask (a matrix representing the parts of the input sequence that can be learned from). We use a simple fully-connected linear layer as the first encoder from the number of input features to the embedding dimension. Then, we use the predefined positional encoding to equip each month's data with information about its position in relevance to the entire input. Sinusoidal functions including sin and cos, as opposed to binary bits, represent the positions of each specific monthly input. After the positional encoding, we pass our data through the transformer encoder, which includes a series of transformer encoder layers that use self-attention in a feedforward network. We specify that the input data has been normalized prior to the operations in the transformer encoder. Each layer has multi-headed attention that learns from different subspaces at multiple positions. Finally, we decode the output with a fully-connected layer to produce a resulting 12-month sequence.

Fig 6.1 Proposed Transformer Model



The dimensions and parameters of this model are first adjusted according to the training loss in order to overfit to specific data points. They are consequently tuned along with other hyperparameters to generalize to the validation set so that the model performs ideally on unseen data in the test set. We calculate loss with MSE and utilize the Adam optimizer (preferred over SGD).

The parameters and hyperparameters are as follows:

1. d_model (embedding dimension - encoder output dimension and decoder input dimension): 12
2. n_head (number of heads in multi-attention model - used in Transformer Encoder Layer): 6
3. d_hid (hidden dimensions in feedforward network - used in Transformer Encoder Layer): 12
4. n_layers (number of Transformer Encoder Layers): 3
5. dropout (used in Positional Encoding): 0.1
6. lr (learning rate): 0.0005
7. weight_decay: 0.0
8. batch_size: 64
9. epochs: 500

Justification for the slight lack in model complexity can be found in the next section.

7 Quantitative Results

The model's accuracy cannot be tested directly as it is impossible to predict the exact value of each statistic for the next few months. Therefore, we use the MSE loss graphs as representations for the performance of our model. Fig 7.1 illustrates the training and validation losses with the chosen hyperparameter settings. As seen, the model is able to surpass the baseline model (for training set) within the first 10 epochs and steadily continues to improve at a small rate and plateaus around 500 epochs at a training loss of 0.49 and a validation loss of 0.52.

Fig 7.1 Training and Validation Losses

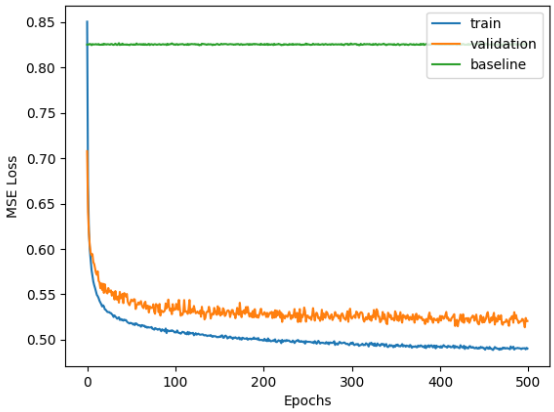
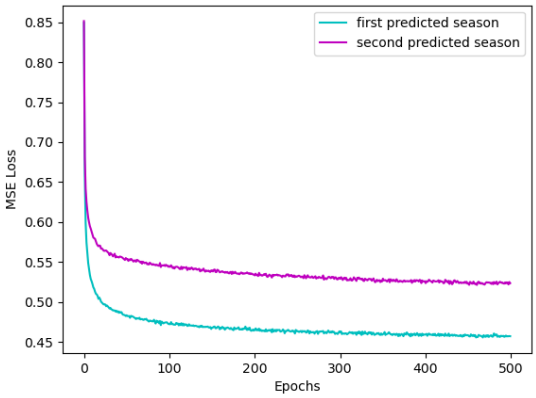


Fig 7.2 1st vs. 2nd season predictions



On further analysis, we find that the first season predictions are much better than second season predictions as shown in Fig 7.2. The loss for the first season reaches 0.45 whereas the second season loss remains around the 0.55 mark. This is reasonable since player projections are generally more accurate for the near future where fewer variables change as opposed to 2 years in advance where more factors are likely to change and affect player performance statistics.

Note that it may seem that the model can be made more complex through adding more encoder and decoder layers with non-linear activation functions. However, we experiment with these modifications only to find that the model overfits heavily to the training set (0.4 MSE) whereas validation loss flatlines at 0.6. Therefore, the model complexity is justified at its current level and the project complexity is primarily centered around the use of the transformer model with positional encoding to perform sequence-to-sequence predictions on augmented data.

The model can be found as research/training_final1.py. Included are the loss plots as well as the model parameters and optimized hyperparameter settings.

8 Qualitative Results

There are clear distinctions between data on which the model performs well and data on which the model doesn't perform up to standard; however, it is not based on player ability but rather on the stage and length of the player's career. Generally, the model performs well with longer careers as they post consistent numbers (most frequently, "good" statistics). As well as that, during the peak years of a player's career (year 3 to approximately 3 years before they retire if they have a long career), they perform consistently and therefore, their monthly statistics are within the same small range leading to better predictions by the model. Towards the twilight of their careers though, players' numbers drop off as they play fewer minutes and therefore, the model is not able to perform because the monthly statistics are far lower than previous seasons. Seasons where players are injured also suffer in the same manner.

To gain a better insight and understanding of our model's performance, we provide two examples. Our model's best predictions with a MSE loss of 0.065 are for the peak years and the midway point (2001-02 and 2002-03 predictions with inputs of 1999-00 and 2000-01 monthly statistics) of Michael Curry's 13-year long career (well above the average career length of 6.57). It is important to note that during this time, Curry remained on the same team and played in the same defensive position. The model's worst predictions with a MSE loss of 2.98 are for Ronnie Lester's last 2 seasons (1984-85 and 1985-86). Lester's stay in the NBA only lasted 6 years as he suffered injuries and unlike Curry, it is important to note that Lester joined the Los Angeles Lakers for the final 2 years of his career from the Chicago Bulls.

For simplicity purposes, only the season averages of both players are included in the A.2 and A.3 tables in the Appendix. Given the large size of our input and output, we refrain from including it in this report. It may instead be found as research/example.txt.

9 Discussion

The final test loss for our model is 0.5008. In direct comparison, the transformer's performance exceeds the performance of the baseline (test loss: 0.6289). Given the number of features, the model does perform well but the improvement in performance (roughly 20%) is not as significant as one would expect. As discussed in previous sections, the model's predictions are realistic and reasonable given a player's form is consistent. However, the caliber of players who can carry a high level of consistency throughout their career is very high and thus, these players are very limited.

We believe the model's performance can be drastically enhanced through variable length input where full careers can be entered into the transformer. The caveat to this is a lack of data present as explored in the data processing section. Additionally, a feature input of age could also suffice in place and be experimented with. With these changes, the model could learn the stage of the career that it is trying to predict for in order to address the issue highlighted in the qualitative results section.

Furthermore, related work in the field suggests that a player's statistics cannot solely be based on their performance in previous years; there are many external factors that our model does not and cannot consider such as physical health, mental health, play-by-play logs, team performance and management. Although players' talent and skills may grow over time and be easier to predict, the external factors may have a great impact on how the player performs in a game setting. For instance, a change in management can lead to the new coach placing an offensive player at a defensive position where the player's defensive statistics and offensive statistics would then be affected drastically.

Further research can be conducted in these aforementioned areas. The research could also expand to other leagues of teams sports such as the MLB, Premier League, NHL and NFL but the real application of the model could perhaps be exploited in individual sports such as tennis, golf and boxing (where team performance and management does not have a drastic effect in statistics).

10 Ethical Considerations

Applications of machine learning models in the sports industry are not without their ethical implications. Our model is intended to be used for recreational purposes for fantasy basketball drafts or as a tool to guide scouts for selecting best-fit talents. However, it can as easily be used for sports betting and lead to gambling addiction. Additionally, it is not meant to act as a replacement for scouts; rather, it is meant for reinforcing their decisions. Determining player value, signing fee and salary on the model predictions is also not suitable.

In terms of data collection, the statistics used in this model are tracked by the same companies that violate player privacy and consent to track personal and sensitive data such as sleep patterns, brain waves, diet and bodily fluids. This concern is present throughout other sports, too.

The model has been made readily available to anyone to mitigate any unfair coaching risks that exploit player projections so that teams may not gain advantages over others through using the model.

The ethics model card can be found as model_card.pdf.

References

[1] S. Benavidez, S. Brito, D. McCreight and P. McEvoy, “Prediction of Future Offensive Performance of MLB Position Players,” Dec, 2019. [Online] Available: https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf. [Accessed Jan. 24, 2022].

[2] G. Xu, “10707 Final Project: Report,” 2018. [Online] Available: https://grantxu.com/assets/files/10707-final-project%20.pdf. [Accessed Jan. 24, 2022].

[3] S. Patel, “nba_api (Version 1.1.11)”, 2021. [Online] Available: https://github.com/swar/nba_api. [Accessed Jan. 24, 2022]

[4] A. Vaswani et al., “Attention is all you need,” Jun, 2017. [Online] Available: https://arxiv.org/abs/1706.03762. [Accessed Feb. 7, 2022].

Appendix

Fig A.1 Statistics and definitions

1. Minutes Played (MP): Number of minutes the player is in the game
2. Field Goals Made (FGM): Number of field goals made (2 and 3 pointers)
3. Field Goal Attempts (FGA): Number of field goals attempted (2 and 3 pointers)
4. Free Throws Made (FTM): Number of free throws made
5. Free Throw Attempts (FTA): Number of free throws attempted
6. Rebounds (REB): Number of rebounds (offensive rebounds + defensive rebounds)
7. Assists (AST): Number of assists
8. Steals (STL): Number of steals
9. Blocks (BLK): Number of blocked shots
10. Turnovers (TOV): Number of turnovers committed
11. Personal Fouls (PF): Number of personal fouls committed
12. Points (PTS): Number of points scored by field goals and free throws

Fig A.2 Michael Curry season statistics

CAREER REGULAR SEASON STATS

SEASON	TEAM	AGE	GP	GS	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	STL	BLK	TOV	PF
2004-05	IND	36	18	7	13.8	1.7	0.7	1.6	44.8	0.0	0.0	0.0	0.2	0.4	50.0	0.4	1.1	1.5	0.8	0.3	0.2	0.3	2.2
2003-04	TOR	35	70	15	17.5	2.9	1.1	2.8	38.8	0.0	0.2	20.0	0.7	0.8	84.5	0.3	0.9	1.2	0.8	0.3	0.1	0.7	2.2
2002-03	DET	34	78	77	19.9	3.0	1.2	2.9	40.2	0.2	0.7	29.6	0.5	0.6	80.0	0.2	1.4	1.6	1.3	0.6	0.1	0.6	2.1
2001-02	DET	33	82	75	23.2	4.0	1.5	3.4	45.3	0.1	0.3	26.9	0.9	1.1	79.1	0.2	1.9	2.0	1.5	0.6	0.1	0.7	2.5
2000-01	DET	32	68	58	21.9	5.2	2.1	4.7	45.5	0.1	0.1	44.4	0.9	1.1	84.9	0.3	1.5	1.8	1.9	0.4	0.0	0.9	2.5
1999-00	DET	31	82	3	19.6	6.2	2.2	4.6	48.0	0.0	0.1	20.0	1.7	2.0	83.9	0.3	1.0	1.3	1.1	0.4	0.1	0.9	2.5
1998-99	MIL	30	50	4	22.9	4.9	1.8	4.1	43.7	0.0	0.3	6.7	1.3	1.6	79.7	0.4	1.8	2.2	1.6	0.8	0.1	0.7	2.7
1997-98	MIL	29	82	27	24.1	6.6	2.4	5.1	46.9	0.0	0.1	44.4	1.8	2.1	83.5	0.3	0.9	1.2	1.7	0.7	0.2	0.9	2.7
1996-97	DET	28	81	2	15.0	3.9	1.2	2.7	44.8	0.3	1.0	29.9	1.2	1.3	89.8	0.3	1.2	1.5	0.5	0.4	0.1	0.3	1.6
1995-96	WAS	27	5	0	6.8	2.0	0.6	2.0	30.0	0.0	0.6	0.0	0.8	0.8	100	0.4	0.6	1.0	0.2	0.2	0.0	0.2	0.6
1995-96	DET	27	41	1	18.3	4.9	1.7	3.7	46.4	0.5	1.2	40.0	1.0	1.4	70.7	0.6	1.3	2.0	0.6	0.6	0.0	0.6	2.2
1995-96	TOT	27	46	1	17.0	4.6	1.6	3.5	45.3	0.4	1.2	37.7	1.0	1.3	72.6	0.6	1.3	1.8	0.6	0.5	0.0	0.5	2.0
1993-94	PHL	25	10	0	4.3	0.9	0.3	1.4	21.4	0.0	0.2	0.0	0.3	0.4	75.0	0.0	0.1	0.1	0.1	0.1	0.0	0.3	0.6

Fig A.3 Ronnie Lester season statistics

CAREER REGULAR SEASON STATS

SEASON	TEAM	AGE	GP	GS	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	STL	BLK	TOV	PF
1985-86	LAL	27	27	0	8.2	2.5	1.0	1.9	50.0	0.0	0.1	0.0	0.6	0.7	78.9	0.0	0.4	0.4	2.0	0.3	0.1	1.6	1.0
1984-85	LAL	26	32	1	8.7	2.8	1.1	2.6	41.5	0.0	0.0	0.0	0.7	1.0	67.7	0.1	0.7	0.8	2.5	0.5	0.1	1.0	0.8
1983-84	CHI	25	43	3	16.0	5.4	1.8	4.4	41.5	0.0	0.1	20.0	1.7	2.0	86.2	0.5	0.6	1.1	3.9	0.7	0.1	1.7	1.4
1982-83	CHI	24	65	38	22.1	8.1	3.1	6.9	45.3	0.0	0.1	0.0	1.9	2.6	72.5	0.7	1.9	2.6	5.1	0.8	0.1	2.1	1.9
1981-82	CHI	23	75	74	30.0	11.6	4.4	8.8	50.1	0.1	0.1	50.0	2.8	3.4	81.3	1.0	1.8	2.8	4.8	1.1	0.2	2.5	2.1
1980-81	CHI	22	8		10.4	3.8	1.3	3.0	41.7	0.0	0.0	-	1.3	1.4	90.9	0.4	0.4	0.8	0.9	0.3	0.0	1.1	0.6