Methods of predicting basketball match outcomes and identifying player outliers

Jason Botha (218083949)

University of KwaZulu-Natal

Abstract. Given a dataset of previous basketball results and statistics, this data can be used to form a prediction on a match outcome. Many methods exist to process this information through a machine learning model to form a prediction that is as accurate as possible. By looking at a linear ranking model and decision tree ensemble to predict a match outcome, it was shown how established metrics are the best basis to use as input parameters and how introducing a factor of uncertainty may provide a better prediction that with just a traditional win or lose label. Further, in finding player outliers it is likely better to predefine a range where a player can be regarded as outstanding, as compared to relying on a supervised learning approach which is less efficient and controllable.

1 Introduction

It is a well-established piece of knowledge that ancient peoples like the Romans and Greeks could bet on the outcome of sport matches. Given the devotion of sports fans and the appeal of betting, it is not surprising in today's world that technology can be used to enhance the guess made for a match outcome. Such a utilization can give an upper hand in a bet by utilizing the previous results and data of matches. However, processing this data in a meaningful way differs from sport to sport, and many approaches could be taken under a single sport.

Basketball is no different. Given a summarised dataset of a team's performance for each year, there are nearly two dozen statistics that are shown in a professional level dataset. Points scored & conceded, matches won & lost, rebounds, and turnovers are just some of these statistics. Given this information, how can an accurate predictor be modelled? There are seemingly countless approaches to this - statistics can be used together, omitted, or weighted differently, not to mention that the predictive models that can be produced to process this information are also countless. It is also possible to model either a regressive or classification model for such sport predictors.

Assuming that a dataset exists that records team data and individual player data, some of these filtering and prediction techniques can be attempted. This is what will be undertaken in this paper: firstly, predicting a winning team given two team choices and their past data, and secondly, identifying outlier players with outstanding performance records.

2 Related Works

In terms of identifying a better basketball team, ex-basketball player and coach Dean Oliver [6] introduced his "Four Factors" based specifically on the statistics of individual basketball teams. Each factor has an associated weight and is derived from the atomic statistics of a team; these are a mix of offensive and defensive statistics.

What if the predication was simply made off which team had a higher win percentage in the past? Lin, Short, and Sundaresen [4] found that this approach was correct 63% of the time in estimating a winning basketball team. Using logistic regression returned a 64% test accuracy. It was also noted how more recent data is more relevant and could be weighted more than distant samples.

Filtering and processing the atomic attributes effectively is paramount to ensuring the model knows what to look for, as pointed out by Shi et. al [10]. It is mentioned how using unnecessary statistics will lead to worse results and how the authors "...found it surprising how stark the influence of choosing the right attributes was on achieving best results." [10]. It would be an oversight to simply use all the statistics as they are and not to transform them somehow, as

intuitively, some metrics are significantly more important, and some may even present a negative record.

With regards to player rankings, Dean Oliver [6] also establishes an approach of getting relevant information out of the dataset, in particular with the Approximate Value (VA) metric for player statistics. This metric is designed to capture the longer-term performance and consistency of a player over a season.

A related metric from Hollinger [3], the Player Efficiency Rating (PER), also assigns a score to players based off their season performance. Hollinger summarizes the metric as taking into account both the positive and negative accomplishments of a player throughout the season, into a per-minute rating [1]. This metric is more tolerant of high variance throughout the season of a player [2].

Assuming a reliable and accurate multivariate feature space is constructed, the identification of outliers can be done. Visually, it is often trivial for a human to identify outliers on a graph, but this does not lend itself to computing. Ostheimer [7] utilizes the Density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm as described in [9] to identify outliers. This method may be appropriate for the identification of basketball player outliers.

3 Methods and Techniques

From the results of our related works research, we cannot see a single type of approach work best, which is to be expected given the so-called 'No free lunch' theorem. It should be noted from our related works research that most sports prediction models are classifiers and not regressors – in such models there are typically only two possible classes which means we should already expect a 50% accuracy rate if a prediction is done at random.

3.1 Team Comparison

A notable aspect of this project is that the type of data we have in our dataset appears very restrictive in terms of what approaches can be used. This is because we have unlabelled, summarized data; there are no instances of single matches containing the teams that played them and the result. This leaves a frustrating precedent – how are the models from this dataset tested for accuracy and implemented? The related works all appear to use individual match data in one way or some existing model as a basis, which allows accuracy testing and classification on their models.

If the data set was constructed as a labelled source as in Fig. 1, it would more easily lend itself to supervised learning techniques. It could be argued that the team's name is the label in the unlabelled data, but this appears like an incorrect assumption to model the team as a class - we are not trying to predict a team. The issue with unlabelled data as we are given (and predicting a class which is inherently unknown until the match takes place), is that we cannot truly test the accuracy of a model that we make. We can find the relative performance of each team and conclude that Team LAL has much better performance than Team SDS, but it would be an oversight to conclude that Team LAL will win.

Labelled historical basketball data

team	opponent	year	win
SDS	LAL	1998	n
SDQ	NJN	1999	n
BOS	KEN	2001	y

Unlabelled historical basketball data

team	points_scored	points_conceeded	wins	losses
SDS	2726	3388	12	65
BOS	2823	3142	29	49
LAL	3291	2712	41	22

Fig. 1.

With the aforementioned in mind, it seems logical to approach the prediction as a regression problem, or introduce a level of uncertainty in a classifier. To form a basis of the relative team

rankings, the Four Factors method is used. These metrics from Oliver [6], their weightings, and their attributes as described further by Migliorati [5] are:

$$Field\ Goals\ Percentage = \frac{P2M + 1.5 * P3M}{P2A + P3A}\ (0.4) \tag{1}$$

$$Turnovers\ Ratio = \frac{TOV}{POSS}\ (0.25) \tag{2}$$

$$Rebounding\ \% = \frac{OREB}{OREB + DREB}\ (0.2) \tag{3}$$

$$Free\ Throws\ Rate = \frac{FTM}{P2A + P3A}\ (0.15) \tag{4}$$

The dataset from [1] is processed according to the Four Factors calculations and each team is assigned a score between 0 and 1. A higher score suggests a better team. From this scoring model a linear ranking is created to provide a basis for future models.

A comparison classifier can be modelled to provide a confidence of a team having a "Win", "Lose" and "Either way" outcome against another given team. A possible manner to approach this is to utilize ensembling learning in the form of bagging while considering the most pressing statistics from the teams. As pointed out by Shi, Moorthy, Zimmerman, diversifying the statistic pool too much can lead to worse results, as wins and points are the most relevant statistics.

With this in mind, the following decision trees (or a primitive CART trees as detailed by Migliorati [5]) with bagging in Fig. 2 is proposed to provide some sense of uncertainty in cases where the teams have a similar ranking in some aspects of the game. An arbitrary 5% margin of difference between teams is checked for which will provide weighting for the "Uncertain" class. Notably, the win and point ratio are also slightly lower weighted than in the Four Factors ranking. A ratio for a team is simply the gained/conceded result e.g., win ratio = won/lost, steal ratio = steals/turnover.

This decision tree is not based off the GINI coefficients nor another formal method of calculating where and when a node should be split. Such an approach would again require a labelled dataset.

3.2 Player Outliers

The player data is different from the team data in the sense that one cannot directly compare two players in a meaningful basketball context and declare "Player A will beat Player B". This means that fundamentally, we are forced to relatively rank players, but not purely as a result of the dataset format. With players specifically, a relevant analysis is to identify outliers from the dataset which is also the task at hand. Like with the team analysis, some atomic statistics are of particular use while others are not considered. Taking into consideration the consistency of a player is necessary to filter out players who may have only one or two matches wherein they perform very well; a player should perform well in most matches to standout and be considered an outlier.

Oliver [6] and Hollinger [3] have different approaches in determining player performance as seen in the existing works review. The AV metric by Oliver looks to reward more consistent performance, while the PER metric by Hollinger is more giving of varying performance but also punishes players for negative statistics. These metrics by themselves are not generally a good indicator – most outlier players would have a high AV, but not all, and a high efficiency rating does not immediately suggest that the player could be an outlier. By Beckler [2] using these metrics together, it was found that a significantly better outlier analysis could be made. Although, it is noted that the PER metric was likely calculated differently due to differing data sources; however, the overall distribution and shape that was calculated remains about the same and comparable to the source example. The scatterplot with this distribution is shown in Fig. 3.

Regardless of the kind of graph that is deduced and metrics that are considered, it is most important to correctly identify outliers. This can be a subjective matter, as the less extreme outliers are a grey-area as to whether they lie in the outliers or not. Subsequently, it would be ideal to have some level of flexibility or visual representation to find the ideal parameters. If a DBSCAN approach is used, the epsilon parameter is most critical in determining the clusters and

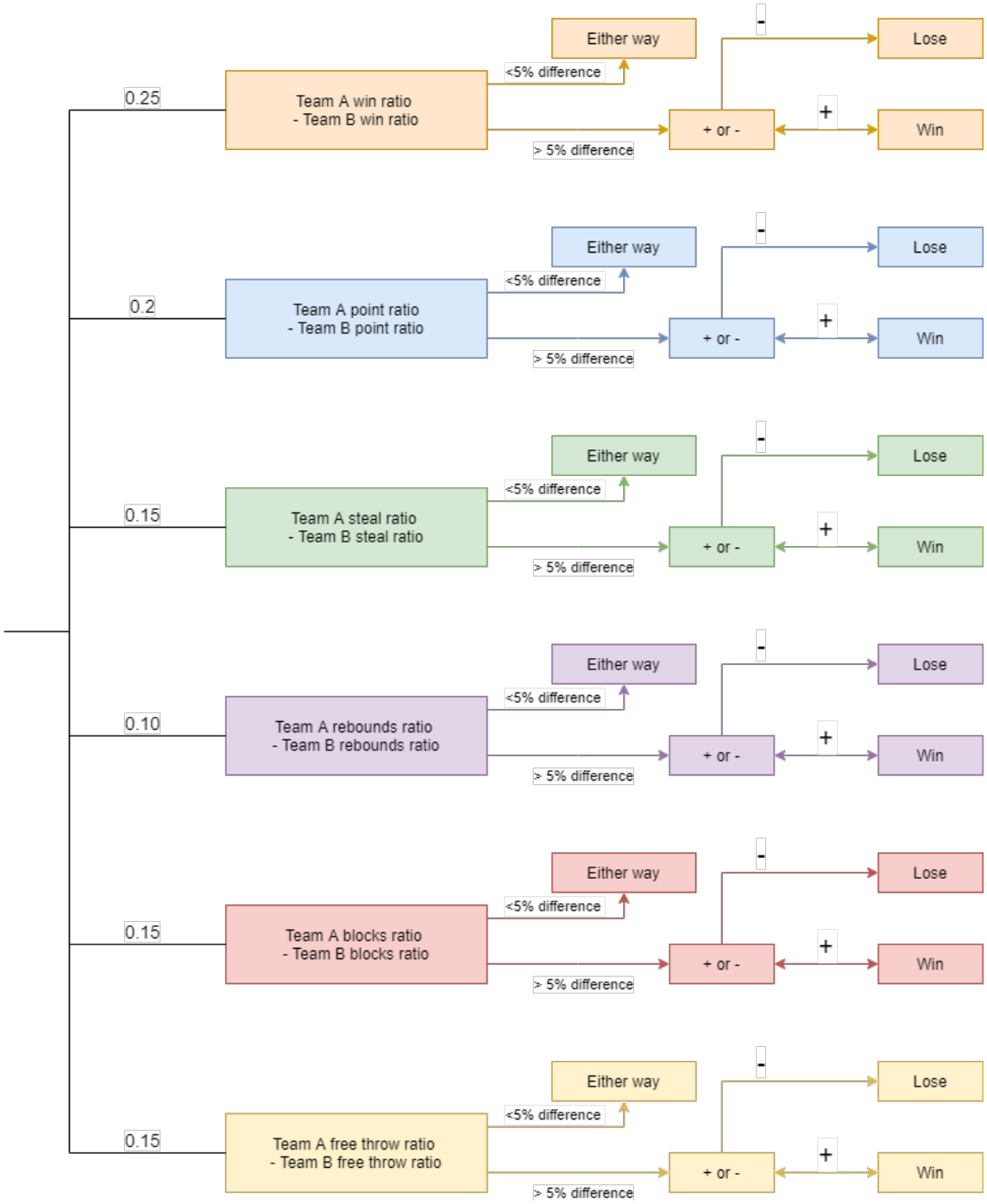
0.25
Team A win ratio
- Team B win ratio
Either way
<5% difference
> 5% difference
+ or -
-
+
Lose
Win
0.2
Team A point ratio
- Team B point ratio
Either way
<5% difference
> 5% difference
+ or -
-
+
Lose
Win
0.15
Team A steal ratio
- Team B steal ratio
Either way
<5% difference
> 5% difference
+ or -
-
+
Lose
Win
0.10
Team A rebounds ratio
- Team B rebounds ratio
Either way
<5% difference
> 5% difference
+ or -
-
+
Lose
Win
0.15
Team A blocks ratio
- Team B blocks ratio
Either way
<5% difference
> 5% difference
+ or -
-
+
Lose
Win
0.15
Team A free throw ratio
- Team B free throw ratio
Either way
<5% difference
> 5% difference
+ or -
-
+
Lose
Win

Fig. 2.

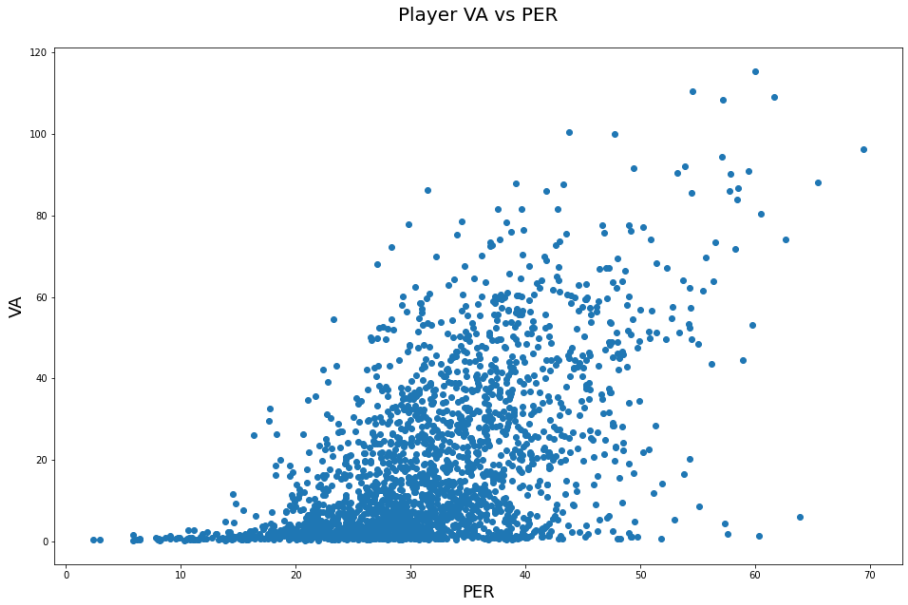

Fig. 3.

the resulting outliers, and can be tried with different values to land at an optimal point. The epsilon value represents the maximum distance between points so that they may be considered as neighbours. A brief graphic representation of this value and the DBSCAN algorithm is outlined in Fig. 4.

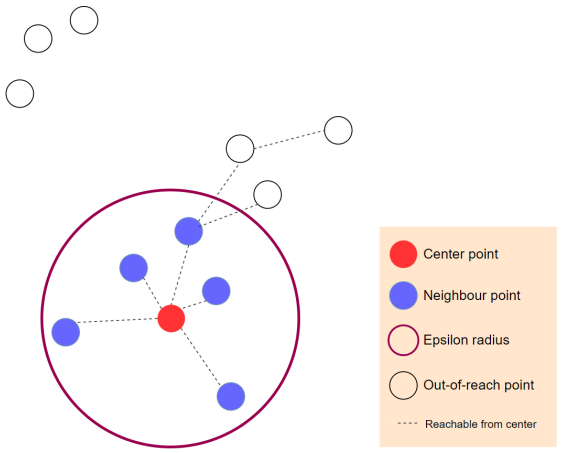

Fig. 4.

An alternative approach would be to define a kernel to split the points into two classes, with the outliers and normal players. This enables us to only define outliers as having high PER and VA values, and not introduce potential low value outliers. In the context of our scatterplot, a polynomial kernel seems appropriate to split the points into two classes. This polynomial kernel can be produced in the scikit-learn library [8], which would require labelled points to be fed into the training set.

A simpler implementation to the aforementioned, which also gives more control over the definition of an outlier, would be to manually define the separator line. This approach differs in that we are not relying on supervised learning to deduce a separator, but compare each point against our predefined idea of what an outstanding player is. This is possible in this context because the

data points are relatively predictable in terms of their range – if we construct a fixed-shape curved separator like the one in Fig. 5, it does not undergo constant reshaping to fit certain datapoints, and can be shifted along the x and y axis in the event the model should be stricter or more lenient in choosing outlier players.

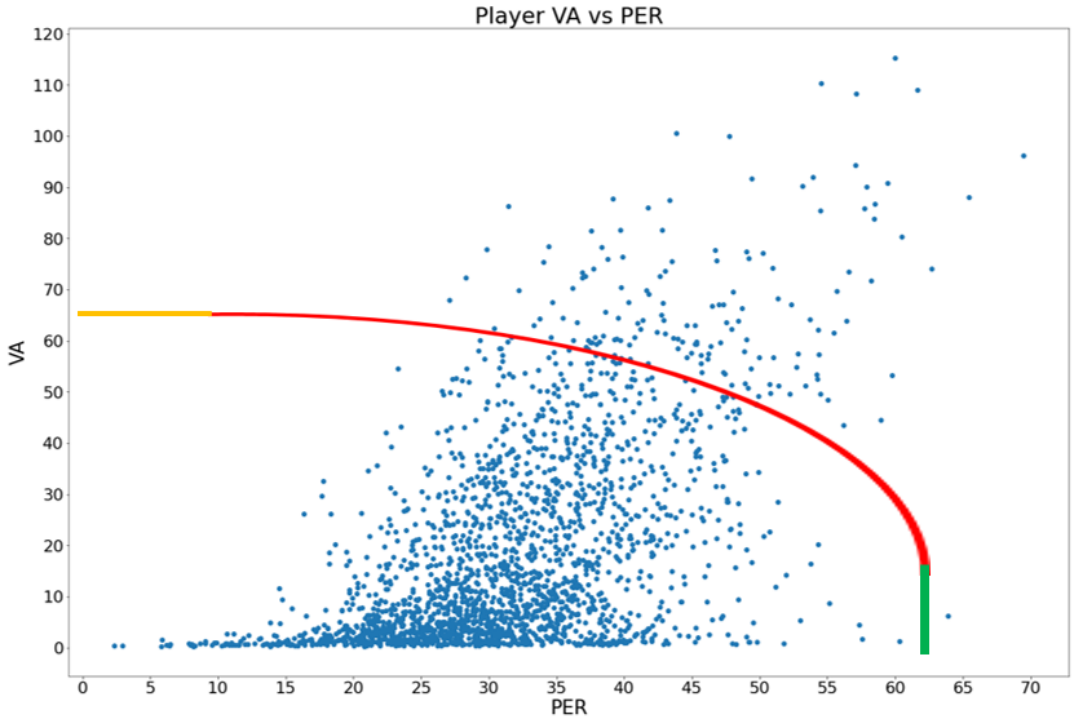

Fig. 5.

In this particular example, the red quarter circle line with radius 60 units was chosen as it appears to portray a reasonable gradient and adequately covers the datapoints; a radius that is too small will not cover enough points, but a radius too large would have the same effect as just using a straight line separator. This red line can be represented by

$$(x - x_{shift})^2 + (y - y_{shift})^2 = 60^2, \quad x \geq x_{shift} \ \ \& \ \ y \geq y_{shift}, \tag{5}$$

with the green and orange segments represented as

$$x = 60 + x_{shift}, \quad 0 \leq y \leq y_{shift} \tag{6}$$

$$y = 60 + y_{shift}, \quad 0 \leq x \leq x_{shift} \tag{7}$$

respectively. The logic in using a curved separator as opposed to a linear one is that the curved line punishes players who are good in one area but lag behind in another. The extent to which this happens is contingent on the curve radius. Granted that a player can sufficiently make up for one of their poor metrics with the other metric, they will be able to qualify as an outlier. We can also define a strictness level that dictates how far the separator line is shifted to allow more or fewer outliers. It may also be desirable to implement a hyperbola formula as a separator instead of using a quarter circle. This would further lend itself to manipulating the shape and gradient of the separator.

4 Results and Discussion

4.1 Team Comparison

Using the weighted Four Factors as described by Oliver [6] and replicated by Migliorati [5] resulted in a predictable linear scale of ratings. Excluding records with missing metrics and obtained before 1974, the data was first filtered and only the ten most recent records for each team were kept. This was to remove interference cause by null metrics and to discard outdated results. A visual representation of the distribution of ratings with labels of select teams is shown in Fig. 6.

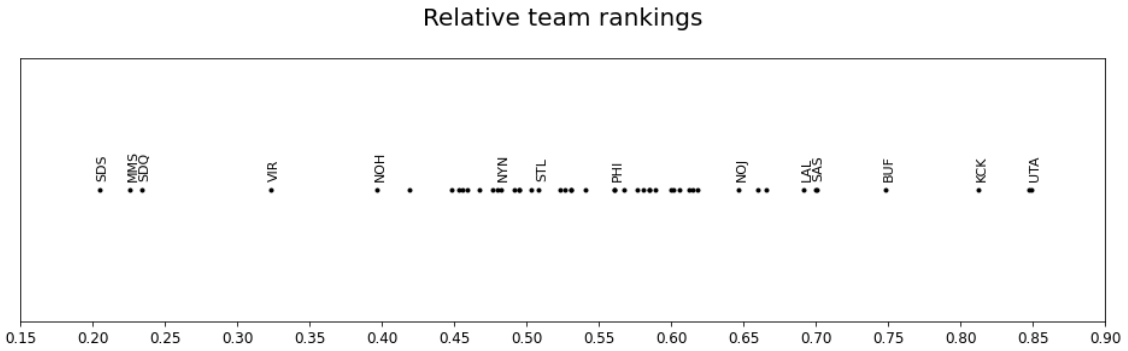

Fig. 6.

This provides a baseline that can be used to gauge the performance of teams. It is visually inspectable by someone and intuitive to compare the teams of interest. However, it becomes unclear on how to compensate for the distance between two teams – teams STL and PHI are comparable in performance, but to which extent should team PHI be given an advantage as it has the higher score? It is difficult to establish this, an arbitrary margin would have to be selected.

The approach with the decision tree to allow some degree of uncertainty did accommodate this issue to an extent. Fig. 7 compares the relative ranking scores against the decision tree metrics (Win, Uncertainty, Lose). A negative relative ranking implies that a team is ranked lower.

Several trends can be seen:

1. The Win probability increases as the ranking difference increases.
2. The Lose probability increases as the ranking difference decreases.
3. The Uncertainty factor peaks as the ranking difference approaches 0.

These trends indicate that the bagging approach does have a correlation to the established ratings. It is noted that some discrepancies exist between the two models, where the decision tree bagging may suggest that a team will win despite the linear model indicating otherwise. This can likely be attributed to the fact that the data processing is different, and it does not use the Four Factors as input. These Four Factors are probably more reliable as a metric of performance than the alternative ones used in the decision trees.

It would be ideal to further investigate a refined model of our ensemble classifier with adjusted inputs for testing, as the decision tree parameters like the weights and margin of uncertainty were not created with scrutinised values, but were rather approximated and estimated.

4.2 Player Outliers

Given the correct parameters, the DBSCAN algorithm is capable of approximating outliers in the scatterplot of PER and VA metrics. The visual representation of points split into classes is shown in Fig. 8. When the epsilon and minimum class size are too small, the dataset is split into more than two classes and the split is too localised. As the values increase, the split is clearer, but the outlier detection may still be too wide. This can be seen particularly when epsilon = 3 and

Team	Opponent	Linear Model	Decision Trees		
		Ranking difference	Win	Uncertainty	Lose
SDS	UTA	-0,644	0,15	0,00	0,85
SDQ	UTA	-0,6153	0,40	0,00	0,60
NOH	UTA	-0,452	0,10	0,15	0,75
VIR	LAL	-0,3682	0,40	0,15	0,45
STL	UTA	-0,3405	0,40	0,00	0,60
LAL	UTA	-0,1573	0,55	0,45	0,00
BUF	UTA	-0,101	0,30	0,20	0,50
CAP	CLE	-0,0819	0,55	0,30	0,15
SAS	LAL	0,0082	0,15	0,60	0,25
NYK	MIL	0,0511	0,25	0,45	0,30
ORL	CHR	0,1042	0,60	0,15	0,25
POR	PHI	0,14	0,40	0,30	0,30
LAL	NYN	0,2095	0,70	0,30	0,00
BUF	STL	0,2395	0,80	0,20	0,00
UTA	WAS	0,3225	0,60	0,40	0,00
UTA	DAL	0,393	0,35	0,35	0,30
KCK	SDS	0,6076	0,65	0,00	0,35

Fig. 7.

min_samples = 7, where the outliers on the poorer performing points are included. With epsilon = 6 and min_samples = 10, we manage to isolate only the better performing outliers.

The issue with using the DBSCAN algorithm is that the shape of the distribution may not lend itself to the optimal outlier detection. In the case of the VA vs PER metrics, the scatterplot has a very uneven distribution and is most dense at the very bottom edge. This explains why some points on the upper left are also identified as outliers if the parameter values are too low, despite most players outperforming them. We are looking for outliers and those points are still outliers if the parameters are set accordingly, just not the good performing outliers. Although the DBSCAN algorithm seems to return a suitable prediction of well performing players with the correct parameters, it is not designed to identify outliers as much as it is designed to perform clustering, so this approach is inherently limited.

The alternative approach with the shifting separator line was naturally more predictable. In cases where the strictness level is low, the term “outliers” is a misnomer as many more points exist on the external convex side of the separator which defines the outliers. The decreasing class size of outliers as the strictness level increases is visualised in Fig. 9. The curve and shift of the split is also visible.

Given that the datapoints are relatively predictable and fall within a range, it may be a better approach to hard code a model or separator line as above as opposed to using the more complicated approach of training and testing a model. In this case, such an implementation did provide desired results, and it could be fed new player points and be able to adapt if new players sufficiently shift the distribution. This is much simpler and more efficient than continuously adapting a DBSCAN model.

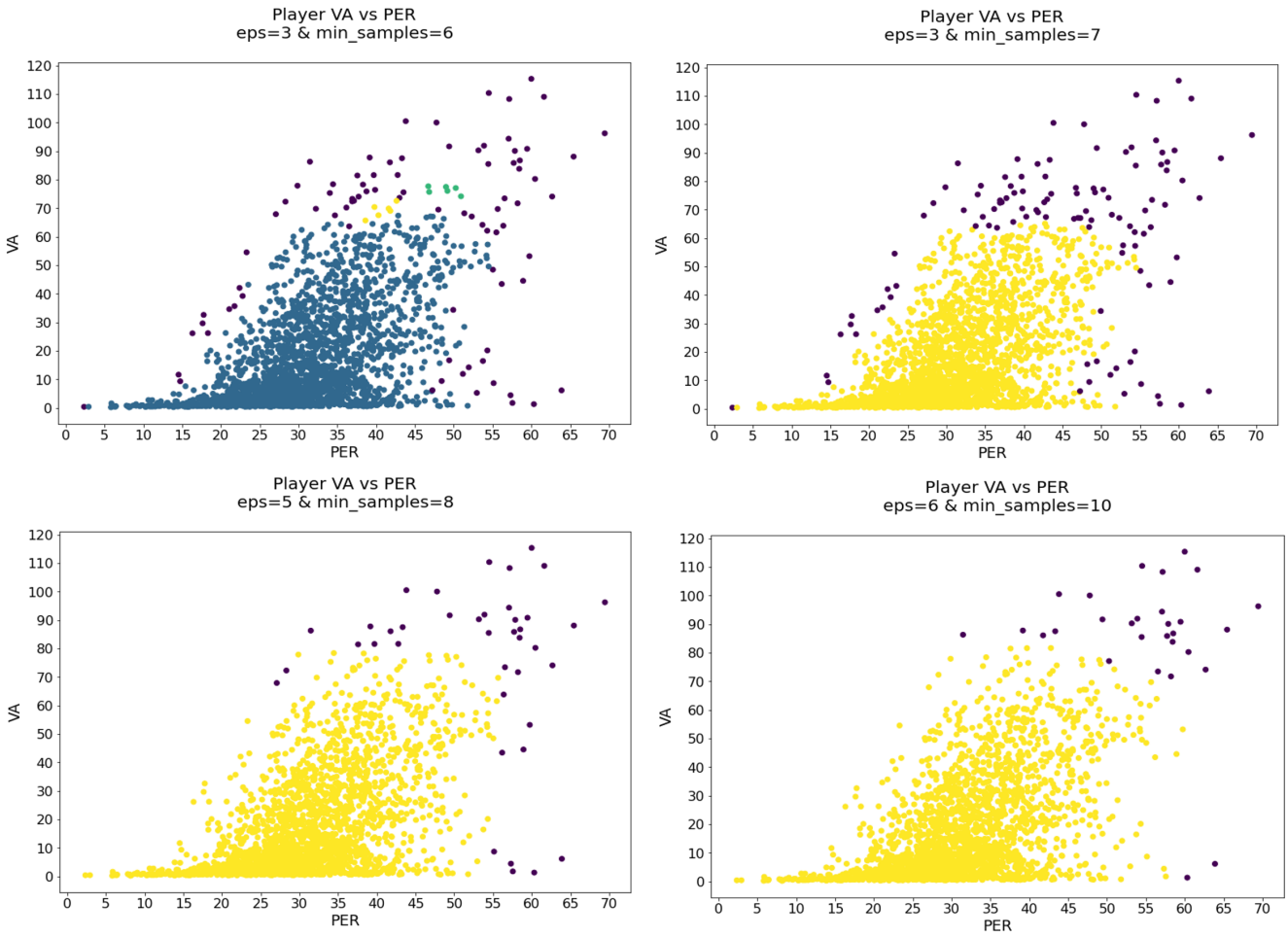

Fig. 8. Visualisation of outlier players using the DBSCAN clustering approach

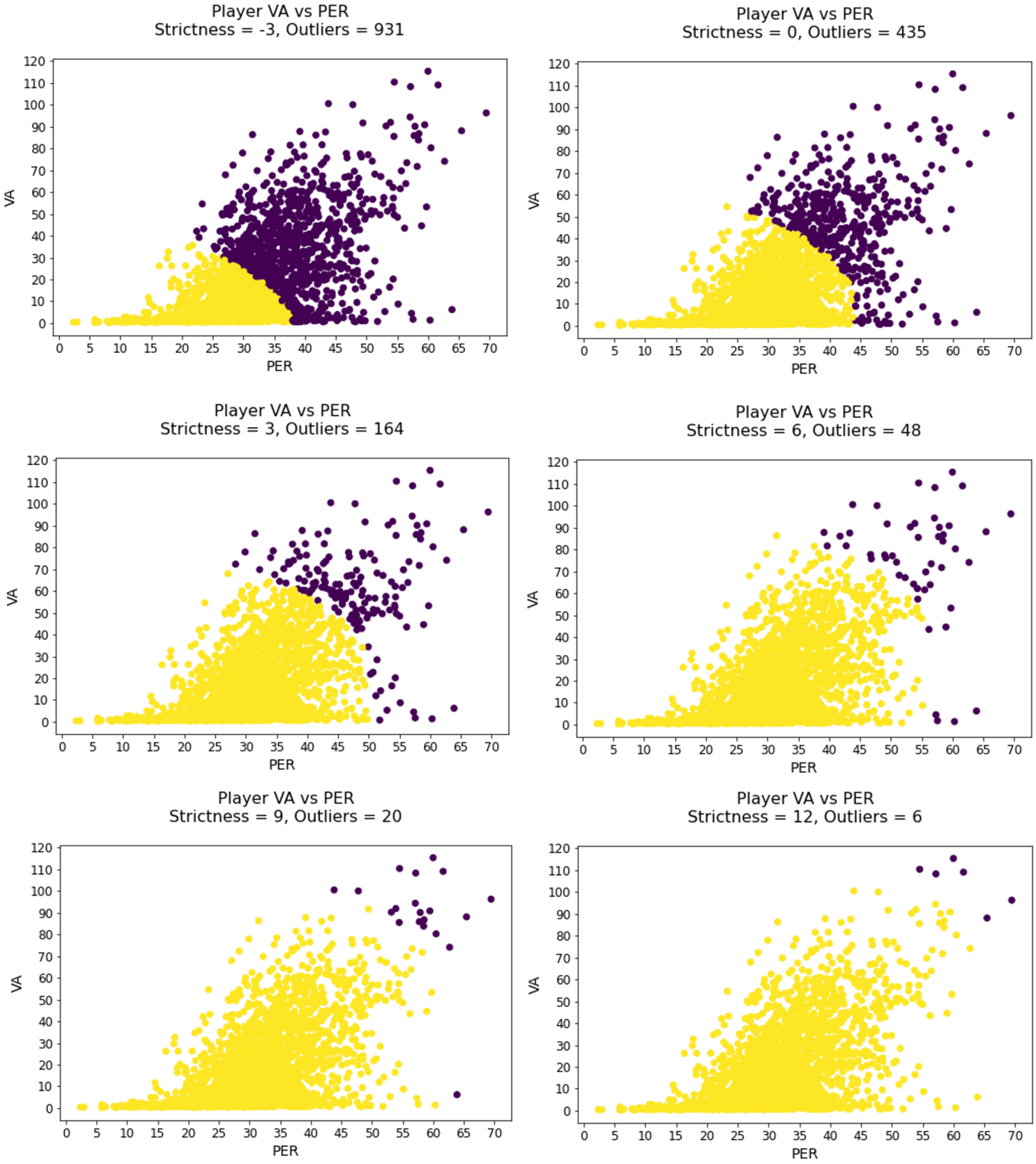

Fig. 9. Visualisation of outlier players using the shifting separator approach

5 Conclusion

Several conclusions can be reached after our task of ranking basketball players and predicting match outcomes. Firstly, it is necessary to ensure that the modelling approach is suitable for a given dataset. If the dataset is unlabelled, one is forced to relatively rank the performance of teams and cannot conclude with certainty that a team will either win or lose. Further, regardless of the type of data, it would be best to use an element of uncertainty in predictions, as the very nature of match prediction will always have uncertainty in it. This uncertainty factor should increase as the apparent difference in performance between two teams decreases.

Secondly, it is pivotal to use only relevant statistics in processing. The Four Factors used as input parameters provided more consistent results than the custom parameters. It would be best to stick to established metrics in future prediction. This is consistent with previous attempts that were researched.

Thirdly, using a manually-defined separator in identifying outliers can provide simpler and better results. The DBSCAN algorithm has to be precisely adjusted to return only desired outliers, whereas the shifting line curve has better control and predictability. It is especially useful to take this approach when the dataset is unlabelled, and the definition of an outlier player is not established.

References

1. Basketball statistics and history (2000), https://www.basketball-reference.com/
2. Beckler, M., Wang, H., Papamichael, M.: Nba oracle (2008)
3. Hollinger, J.: Pro basketball forecast, 2005-06. Potomac Books (2005)
4. Lin, J., Short, L., Sundaresan, V.: Predicting national basketball association winners (2014)
5. Migliorati, M.: Detecting drivers of basketball successful games: an exploratory study with machine learning algorithms. Electronic Journal of Applied Statistical Analysis EJASA, Electron. J. App. Stat. Anal. Electronic Journal of Applied Statistical Analysis **13**, 454–473 (10 2020). https://doi.org/10.1285/i20705948v13n2p454
6. Oliver, D.: Basketball on paper : rules and tools for performance analysis. Potomac Books (2004)
7. Ostheimer, J.: How to detect outliers in a 2d feature space (06 2020), https://towardsdatascience.com/outlier-detection-python-cd22e6a12098
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
9. Sander, J., Ester, M., Kriegel, H.P., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. Data Mining and Knowledge Discovery **2**, 169–194 (1998). https://doi.org/10.1023/a:1009745219419, https://link.springer.com/article/10.1023/A%3A1009745219419
10. Shi, Z., Moorthy, S., Zimmermann, A.: Predicting ncaab match outcomes using ml techniques -some results and lessons learned (2013)