BasketBall IQ White Paper

1 Introduction

In this white paper, we will provide a more detailed overview of the dataset that we are using, the methods we are applying (learning algorithms, permutation feature importance, bootstrap prediction intervals) and the corresponding results. We will also provide visual support material to illustrate the results. We aim to give a detailed account of our work while justifying our choices in methodology and explaining the overall process of going from raw data to actionable insights. We will also interpret our results in terms of basketball insights and highlight the limitations of using a mathematical model to make decisions. In this white paper, we leave the mathematics out on purpose, but make a conscious effort to reference the methods we are using in order to make the methodological details more accessible to the reader.

2 The Data

Our dataset consists of 299 college players who got drafted between and including the years 2003 and 2014, giving us 12 years worth of data. This only corresponds to 25 players per draft year as we only kept players coming out of college (i.e, we excluded international players and the few players drafted out of high school). Furthermore, we only kept players who had played at least 60 total games in their 3^{rd} and 4^{th} NBA seasons combined, with at least 30 games played in both years. Finally, we had to carefully remove outliers, which were skewing our model and giving worse predictions/generalization. As an example, Isaiah Thomas and Tyrus Thomas are removed, but Hasheem Thabeet, O.J. Mayo or Kawhi Leonard are not. For each player, we gathered and constructed 30 variables, which can be seen in the correlation matrix below from Figure 1. Since we have 299 players, this gives us about 10 players per variable. One can see in Figure 1 that many variables are highly correlated.

Highly colinear variables can lead to worse predictions and parameters to switch sign (in a linear regression context), rendering them uninterpretable [1]. We thus want to remove some of the more colinear variables in order to measure and interpret the effect of our variables on the response we are trying to predict (see Section 3.3). To do so, we use the well-known variance-inflation-factor (VIF) [2] with a threshold of 10. By doing so, we remove 8 variables, thus leaving 22 variables that were used for our subsequent predictions and analysis. Therefore, there 14 observations (i.e. players) per variable, where as a rule of thumb, one would want a minimum of 10 observations per variable. The new correlation matrix is shown in Figure 2.

The only highly correlated variables left as a result are the offensive rebound % (OR%) and the defensive rebound % (DR%). The other 4 highly correlated variables at the bottom right of the correlation matrix are the average PER/48, OWS, DWS and WS/48 that players obtained across their 3^{rd} and 4^{th} NBA seasons. These statistics are used as performance metrics, since they are well known and widely used. We also thought of using the contracts (i.e. salaries) players get after having played 4 years of NBA as a performance/talent metric. However, this measure suffers from the

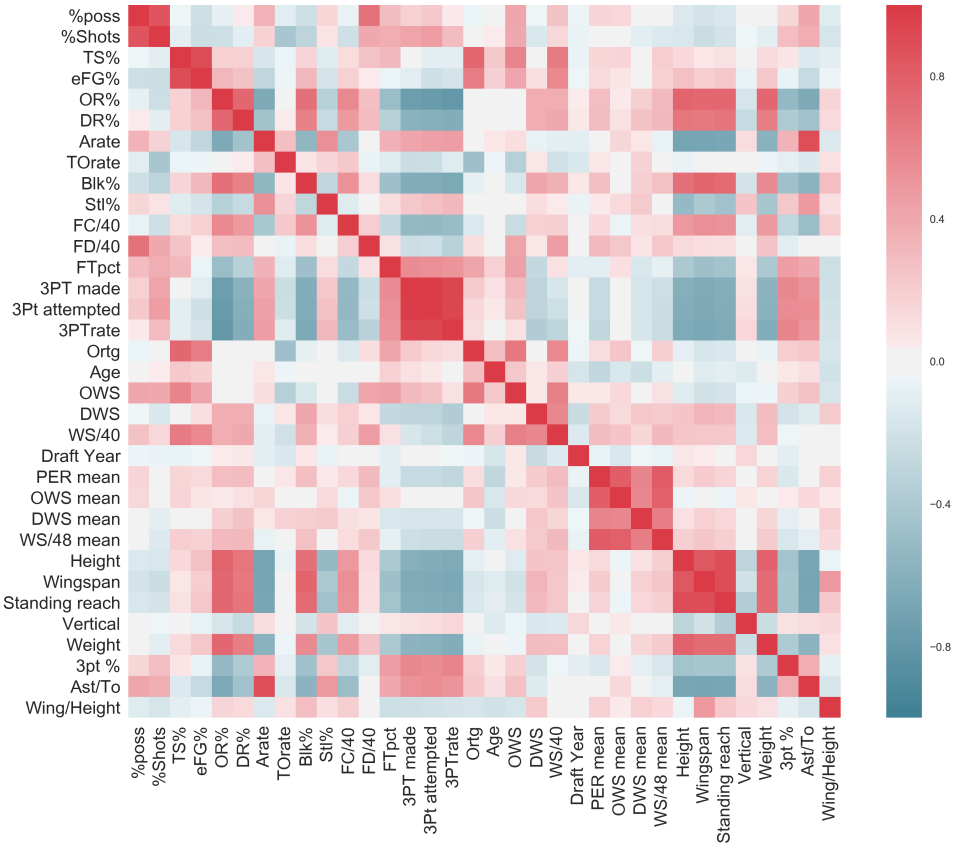


Figure 1: Correlation matrix before removing the highly colinear variables.

fact that even players with similar contracts often vary greatly in skillsets. The two seasons are taken to be equally important and we weight the average by the number of games played in each season. If we let G_{Y3}, G_{Y4} be the number of games played in seasons 3 and 4 respectively, and similarly, P_{Y3}, P_{Y4} be the average performance metrics for seasons 3 and 4, then our final computed metric P is given by

$$P = \frac{G_{Y3} \times P_{Y3} + G_{Y4} \times P_{Y4}}{G_{Y4} + G_{Y3}} \tag{1}$$

By taking a weighted average across two seasons, we more accurately capture the performance of players, by having more games to judge them by and by averaging out the effects on individual performance that teammates may have, as the teams can have a high turnover over the period of a year. We have 4 different metrics which all measure different aspects of performance, with for instance, defensive win shares (DWS) obviously focusing more on defensive skills than offensive win shares (OWS). We arbitrarily chose to use the PER/48 mean, but, as all the metrics are highly correlated with each other, we could use them interchangeably. Furthermore, as we will see in Section 3, we make use of the other three metrics (OWS, DWS and WS/48) when predicting the PER/48, by using them as additional inputs to the initial 22 variables.

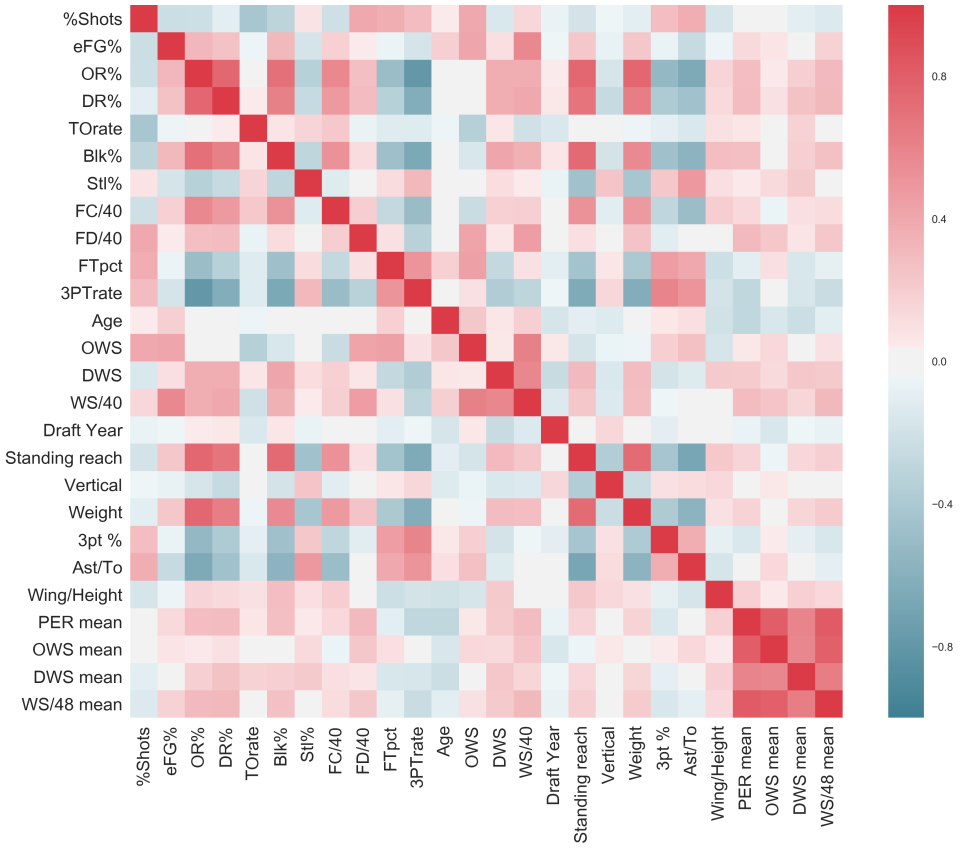


Figure 2: Correlation matrix after removing the highly colinear variables.

3 The Methodology

We divide the Methodology section into three subsections covering the following topics: the PER mean predictions and the validation of our results, the computation of the bootstrap samples, and finally, the method used to compute the feature importance to highlight the strengths and weaknesses of players.

3.1 Predictions and Validation

As mentioned previously, our dataset consists of 299 players from 12 different draft classes. In order to compute how good our predictions are on players unseen by the model, we adopt a 12-fold testing scheme, where each time the testing set is taken to be all the players drafted in a given draft year. In doing so, we get as close as possible to how predictions are made in the real world, where we predict an entire draft class all at once. For instance, when the test set is taken to be players drafted in 2003, the remaining players are used to perform a (random) 10-fold cross-validation in order to tune the hyperparameters in our regression algorithms. Once the values of these parameters are found, we retrain the model on the all the players drafted from and including 2003 to 2014 and make predictions for players in the test set. All our

visualisations and predictions are reported on the test set, so there is no possibility of invalid results and potential overfitting.

An important contribution of our method is that we recognised that this is in fact a multi-target regression problem. This is because we have 4 different metrics, which can all be predicted using the same 22 variables mentioned earlier. While we saw in Figures 1 and 2 that these metrics are highly correlated, they do not contain the same information. In other words, they are complementary and we would like to make use of the OWS, DWS and WS/48 when predicting the PER/48. The method we use is called target stacking [3]. In general, multi-target regression leads to a better performance when compared to a single target regression as we use extra information by exploiting the relationship that the 4 variables/metrics share. Our features are now a superset of the features we would use in single regression. In Section 4, we will report our results for various learning algorithms and for the single and multi-target regression approaches.

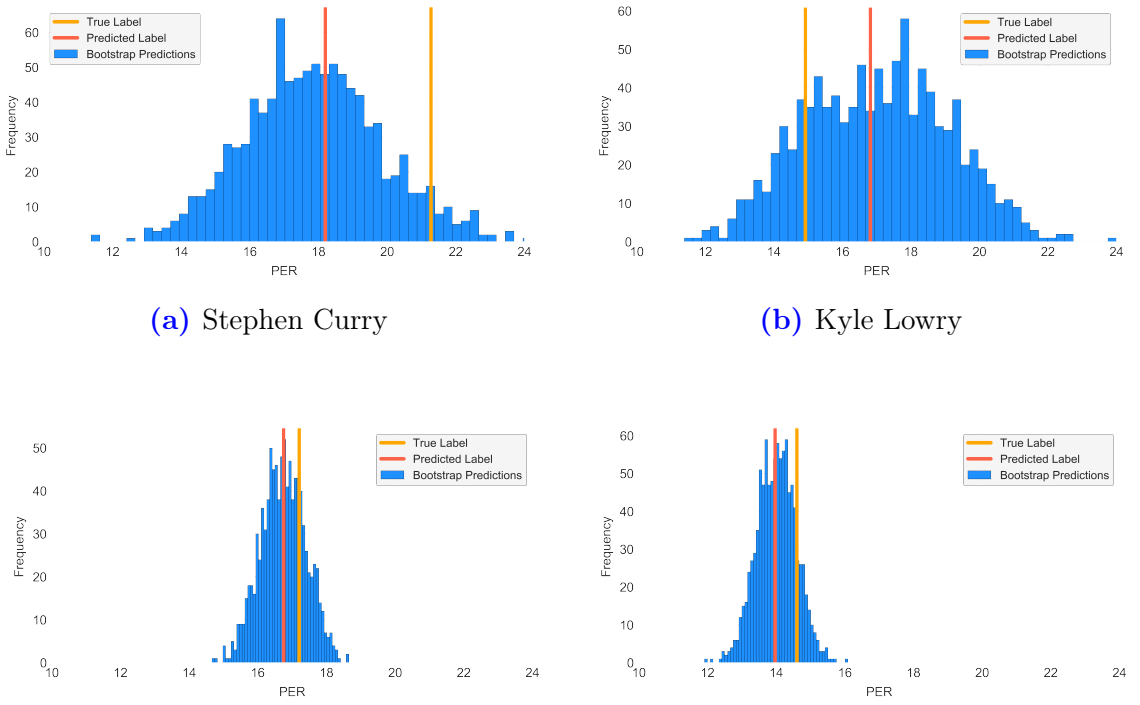
Finally, on top of stacking targets, we also stack several learning algorithms and use their predictions as meta features. The stacking of models/algorithms allows us to combine the predictions from our different learning algorithms. As an example, one model may perform better for offensive players and another model - for defensive players. Stacking would allow us to combine the strength of our two models and perform well for both defensive and offensive players. A great and simple illustration of how stacking models work in practise can be found in [4]. We implemented strategies A and B from [5]. Strategy A did not work as it introduced too much information leakage due to the small size of the dataset. Strategy B improved the results slightly over stacking targets on its own, but introduced a lot of complexity and made the model harder to maintain. Our conclusion is that stacking targets without stacking models offer the best trade-off between complexity and performance and is thus our preferred method. We tried the following algorithms: ridge regression, elastic-net regression, kernel ridge regression (KRR), support vector regression (SVR) and gradient boosting with trees (XGboost), for which we report the results in Section 4.

3.2 Prediction Bootstrap Intervals

In this section, we explain how the bootstrap intervals are computed and how they should be interpreted in the basketball context. We also show a few examples from our analysis in order to illustrate and strengthen our argumentation. In our analysis, we decided to use a nonparametric paired bootstrap for regression [6]. In other words, we sample (with replacement) players at random, and construct a new training set. We then train our regression model on the bootstrapped training set and make predictions for the players in our validation set. We repeat the process 1000 times and obtain a distribution for the predicted metric P from Eq. 1, from which we can compute a number of statistics such as the mean, and any quantiles in order to build prediction intervals. For a given player, a prediction interval can be interpreted in the following way: if a prediction interval is narrow, it means that our model is robust to variations in the training set and that we are fairly confident in our prediction. On the other

hand, if our interval is wide, it means that we are unsure about our prediction. This may happen when we are making predictions for a player with unique characteristics (i.e. only a few players in the training set are similar to this player). Thus, it is likely that these rare players do not get sampled many times in our bootstrapped training sets, rendering inference difficult in such cases.

Figure 3 below displays the prediction bootstrap intervals for Stephen Curry, Kyle Lowry, Jared Sullinger and Jared Dudley. We can see that Lowry and Curry have much wider intervals, spanning across 10 PER points, while intervals for Dudley and Sullinger span across 4 PER points. In turn, we can see that this uncertainty is reflected by the error between the true values and our predictions. This shows us how prediction intervals can be used as a proxy for how uncertain we are in our predictions. Furthermore, one can look at the skewness of the distributions in order to assess the nature of our uncertainty. Intuitively, if a player has a right skewed histogram (e.g. Kyle Lowry), it means that most of our uncertainty comes from the fact that we are rating this player more highly than we have predicted (i.e. the red line on the graphs). Note that these examples were picked to illustrate how prediction intervals and prediction errors (i.e. true label vs. predicted label) relate, but there will always be cases where the interval is narrow and the error is large.



(a) Stephen Curry

(b) Kyle Lowry

(c) Jared Sullinger

(d) Jared Dudley

Figure 3: 95% prediction bootstrap intervals for Stephen Curry, Kyle Lowry, Jared Sullinger and Jared Dudley. Each bootstrap sample contains 1000 samples.

3.3 Feature Importance and Skills Evaluation

3.3.1 Background

As it will be illustrated in Section 4, using a simple Ridge regression or Elastic-Net regression does not return the most accurate performance. This is due to the fact that it does not capture the dependencies and the interactions between the features. For example, we know that height interacts with 3pt%, making linear models mostly inappropriate for the task at hand. Kernel methods are popular and effective methods to capture interactions by mapping the input space (i.e. the features) to higher and potentially infinite dimensional space (e.g. for the RBF kernel) [7].

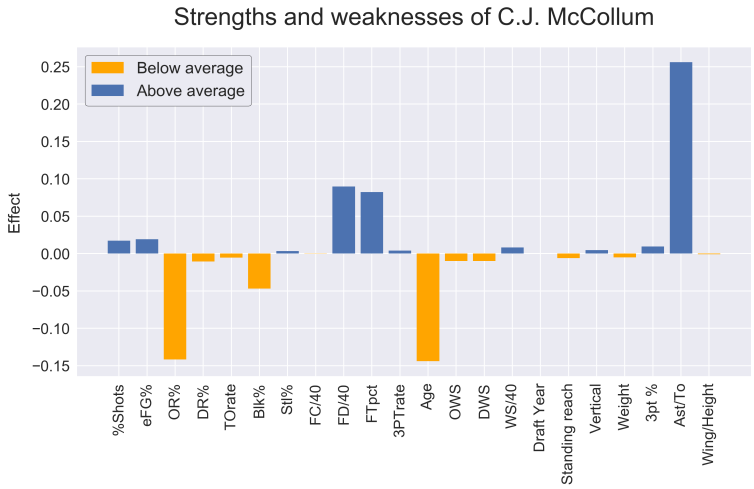
However, using kernel methods gives rise to a common issue in machine learning: the complexity vs. accuracy trade-off. In other words, as complexity increases, so does accuracy (i.e. the generalization power) but we usually lose interpretability. By interpretability we mean: "the model predicts x because of the following factors, and this is how much each factor matters". For so called black-box models, as opposed to a simple linear regression (e.g. Ridge regression), one cannot directly interpret the feature importance by reading the values of the parameters. However, even for the simplest linear models, interpretability is often questionable due to a potentially large number of features, which are all inter-dependent. When the relationship between the variables and the response is not linear, the interpretation only makes sense when the model is well suited and generalises well for the task at hand. Next, we expose the method that we used to make our model interpretable by computing the importance of each variable in predicting the response (i.e. P in Eq. 1).

3.3.2 Permutation Feature Importance

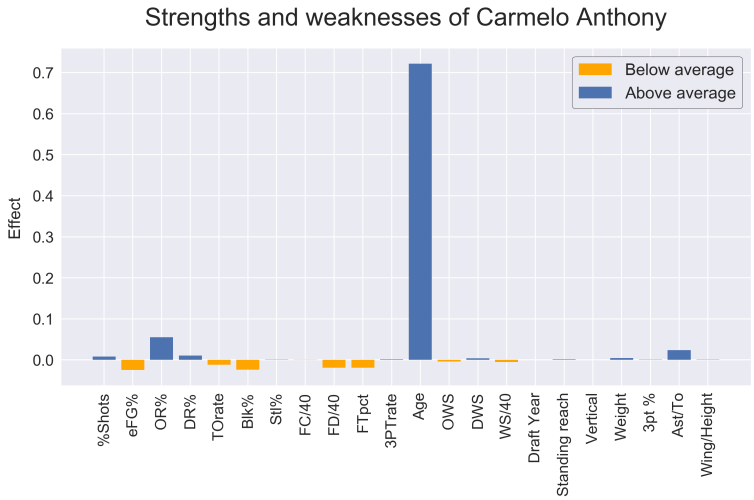
In this section, we present the method we used in order to quantify the importance of individual variables, which will then allow us to build a personalized skills map of the players. The Permutation Feature Importance (PFI) [8] is a flexible method, which allows us to compute feature importance on any black-box models with complicated relationships between the input and the output. The PFI looks at how much each feature influences predictions from a trained model by randomly shuffling each feature at a time and making predictions on the validation set. Permuting an irrelevant feature would give the same predictions, leading to a feature importance close to zero. On the other hand, if we permute a predictive feature, the predictions would become significantly worse, leading to a high feature importance score for that given feature. Note that it is very important to assess the colinearity between the variables before attempting to measure the importance of each feature. This is because highly colinear variables will share their importance with the other variables, making them appear less important than they actually are. A detailed discussion of this phenomena [9] looks at the effect of colinearity/correlation and also compares the PFI method to the feature importance of the random forest. Once the feature importance is computed for each variable, we standardize them such that they sum to 1 (the plot is shown in Section 4.

3.3.3 Skills Evaluation: Strengths and Weaknesses

We can use the previously obtained feature importance in order to assess the strengths and weaknesses of a player. This in turn allows us to compare each player with all the remaining drafted players from 2003 to 2014. For each player, we normalize their college statistics with a mean of zero and a standard deviation of 1, such that any negative number indicates that the player is below average with regards to that particular variable/skill. Finally, to obtain the entire skillset of a player, we simply multiply the normalized statistics with the feature importance found from the PFI. As an example, we show the strengths and weaknesses of C.J. McCollum and Carmelo Anthony in Figure 4.



(a)



(b)

Figure 4: Skillsets of C.J. McCollum and Carmelo Anthony in their last college seasons.

To interpret the figures, the orange color signifies that a player is below average relative to the rest of the drafted players, while blue indicates higher than average (n.b. for age, below average means older). For Anthony and McCollum, most variables are centered around zero, which is either because the player is average or our model has found these variables not to matter in having a successful transition from college to the NBA. If a player is above average in a skill that our model deems unimportant, the effect of this skill on his NBA performance will be zero. The importance graph of each skill, in order to have a successful transition from college to NBA, is not shown in this whitepaper, but is provided as part of the development tool. For C.J. McCollum, teams should be very pleased about his ability to draw fouls and his assist to turnover ratio, but should be aware of his advanced age and his lack of defensive skills (i.e. low offensive rebound % and block %). For Anthony, we see that he does not stand out in any of the statistics (i.e. fluctuates around the mean), but is much younger than his drafted peers (2003-2014 drafts) and our model finds it to be a positive attribute. Another way to interpret the skillset of Anthony is that, despite being a freshman, he had attributes at the same level as the average of all the draftees, thus being compared to much more experienced college players.

4 Results

4.1 Background

In this section, we present our predictive algorithm from which the bootstrap prediction intervals and the permutation feature importance follow. Having the best predictions possible is thus not only very important in itself, but is also the backbone of the analysis. Firstly, we will present our results, for different learning algorithms, without using multi-target stacking. Then, we will implement the multi-target stacking strategy [3] (see Section 3.1) on the three best algorithms from the first step.

4.2 Preliminary Results

In this subsection, we present and compare the results of several learning algorithms (Ridge regression, Elastic-net regression, XGboost, Kernel ridge regression and SVM). Again, we emphasize that all the results presented in this section are for players unseen during the training process. In order to evaluate the performance of our approach, we compute 2 metrics - the mean absolute error (MAE) and the Pearson's correlation coefficient (CC) - between the true labels and our predictions and the results are shown in Table 1. As mentioned in Section 3.1, the reported results are the average across the 12 validation folds, where each fold corresponds to a draft class from 2003 to 2014.

Table 1: Average MAE and CC across the 12 validation folds. These results are obtained without augmenting the input features by using multi-target stacking.

	Algorithms					
Metrics	Mean	Ridge	Elastic-net	KRR (poly)	SVM (rbf)	XGboost
MAE	2.84	2.45	2.49	2.41	2.47	2.49
CC	NA	0.29	0.30	0.31	0.31	0.28

As it can be seen in Table 1, all the algorithms perform significantly better than random guess abbreviated as 'Mean'. For the random guess, we simply make a constant prediction, which corresponds to the average PER rating of all the players in our training set. The best performing algorithm is the Kernel ridge regression with a polynomial kernel of degree 2, followed by Ridge regression and SVM with RBF kernel. To access the hyperparameter values and reproduce these results, the code and the data will be given as part of the All-Star Package.

4.3 Final Results

Table 2 below displays the results for the three best algorithms from Table 1, but this time we implement multi-target stacking. We see that we do not gain accuracy from multi-target stacking on the linear Ridge regression but we get improvements on both the KRR (poly) and the SVM (rbf).

Table 2: Average MAE and CC across the 12 validation folds. These results are obtained with augmented inputs using multi-target stacking.

	Algorithms		
Metrics	Ridge	KRR (poly)	SVM (rbf)
MAE	2.45	2.39	2.38
CC	0.30	0.30	0.31

In Appendix A, in Figures 9 and 10, we show the prediction graphs from the KRR (poly) algorithm for all the draft classes from 2003 to 2014, where each draft class is taken as the validation fold. We can look at our results and immediately gain some insights from the predictions. For instance, in the 2003 draft, we find that the top four picks are on a very similar level, but that David West and Josh Howard could have perhaps been drafted at a higher position. We also notice that Mo Williams and Kyle Korver may be overlooked. The 2005 draft is a perfect example, which shows that our model should be used alongside human experts. Indeed, while our algorithm agrees with drafting Andrew Bogut at the top spot, it also rates Danny Granger much higher than Chris Paul. On the other hand, it suggests that Chris Paul could have gone ahead of both Marvin Williams and Deron Williams. We also find that David Lee and Brandon Bass could have been picked earlier in the draft. In 2006, we find that Paul Millsap, Rajon Rondo and Kyle Lowry were all overlooked by the

scouts and that Adam Morrison should not have gone second. Overall, our algorithm makes accurate predictions and is good at finding overrated players (e.g. Adam Morrison, Marvin Williams, Jeff Green, Corey Brewer, O.J. Mayo, Hasheem Thabeet, Wesley Johnson, Brandon Rush, Michael Kidd-Gilchrist, Harrison Barnes, Terrence Ross, Carter Williams) and underrated players (e.g. David West, Josh Howard, Mo Williams, Kyle Korver, Danny Granger, David Lee, Rajon Rondo, Kyle Lowry, Paul Millsap, Joakim Noah, Thaddeus Young, Rodney Stuckey, Brook Lopez, DeAndre Jordan, Stephen Curry, DeMarcus Cousins, Hassan Whiteside, Kawhi Leonard, Kenneth Faried, Damian Lillard, Andre Drummond, Steven Adams). On the other hand, like any models, it is imperfect and make mistakes, the most notable being: Kirk Hinrich, Chris Paul, Mike Conley, Russel Westbrook, George Hill, Evan Turner, Paul George, Eric Bledsoe, Kemba Walker and Nikola Vucevic.

Another advantage of quantifying the importance of the variables is that we can query our model in order to get some insights on why it behaved a certain way. For instance, we can investigate why our model did not rate Paul George and Russel Westbrook high enough. Note that the scouts managed to assess Westbrook's potential but also overlooked Paul George.

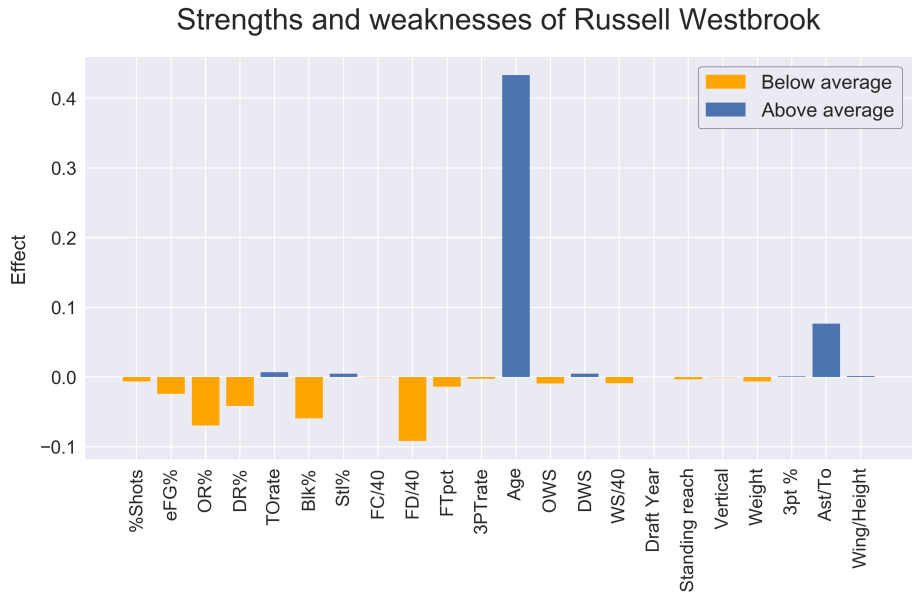


Figure 5: Skillset of Russel Westbrook in his last (sophomore) college season.

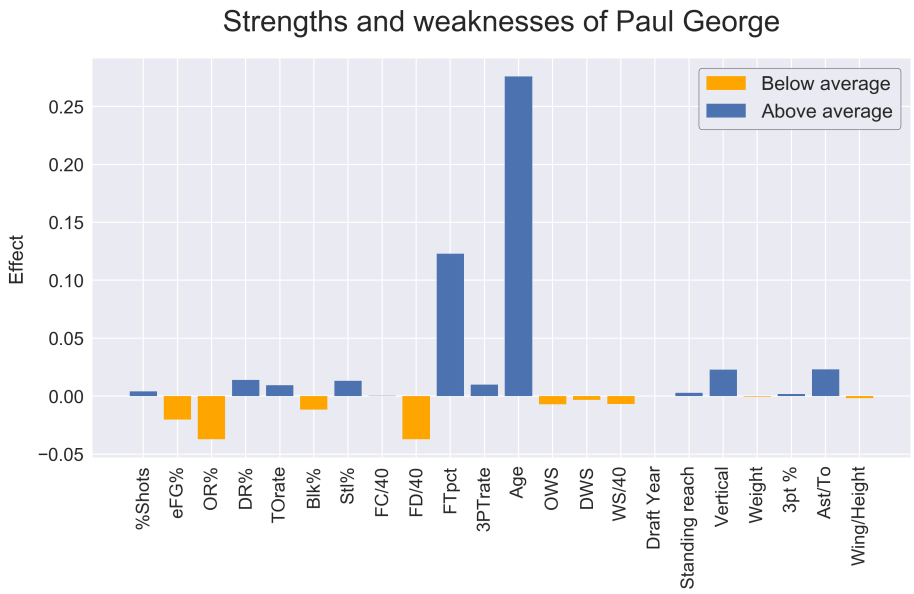


Figure 6: Skillset of Paul George in his last (sophomore) college season.

In Figure 5, we can see that the main upside of Russell Westbrook is his young age and his assist to turnover ratio, which can be somewhat compared to Carmelo Anthony in Figure 4. However, Anthony was even younger and was much closer to the average in many of the statistics, making him a more promising prospect. Paul George's skillset in Figure 6 is also similar to Westbrook's with a much higher free throw percentage. In conclusion, our algorithm does not rate them highly as they are below average in many important categories (e.g. rebounding, shooting efficiency, drawing fouls) and do not stand out significantly in more than two categories, one of which being age. Much like Carmelo, and unlike McCollum, Westbrook and George are young promising players, but they are not NBA ready and a lot of their NBA success was dependent on their development and adaptation once they reached the league. This is where human judgment and scouting are very important in order to assess qualities correlated to development (e.g. work ethic, basketball IQ).

4.4 Model Vs. Scouts

So far, we have compared our predictions to the true values, and highlighted where our model got it right or wrong. In this subsection, we evaluate how our model fares against scouts and front offices over the years (2003-2014). While we do not have PER predictions available from the scouts, we can use the draft position as a proxy for their assessment. We come up with two ways to compare the performance of our model against the scouts, using the PER as the comparison metric.

The first way is to compare the performance within drafts: for each draft, we compute the average true PER of the top 10 picks chosen by the scouts, and the average true PER of the top 10 picks predicted by our model (i.e. the 10 players with the highest predicted PER). These two averages are shown in Figure 7 below,

for each draft class. We can see that our model is consistently better except for the year 2008. Looking at Figures 9 and 10 in Appendix A, we can inspect which players are responsible for such disparities (e.g. Adam Morrison is not part of the top 10 in our model for the 2006 draft class).

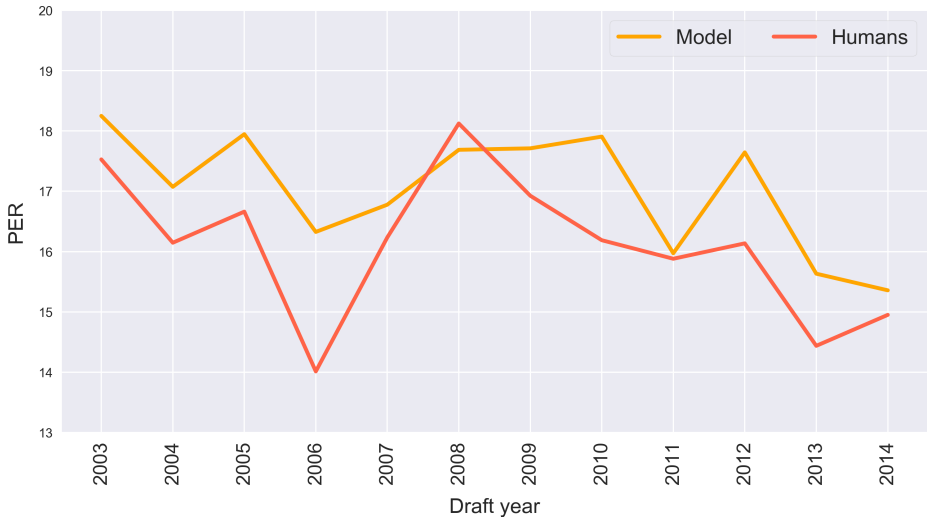


Figure 7: Average PER of the top 10 players for each draft class. 'Model' refers to the top 10 players chosen by our model, while 'Humans' refers to the top 10 players chosen by the teams.

We also compare the performance for each draft position across the drafts. This allows us to check whether our model is better for each draft position by averaging the PER across the 12 draft classes and the results are shown in Figure 8 below.

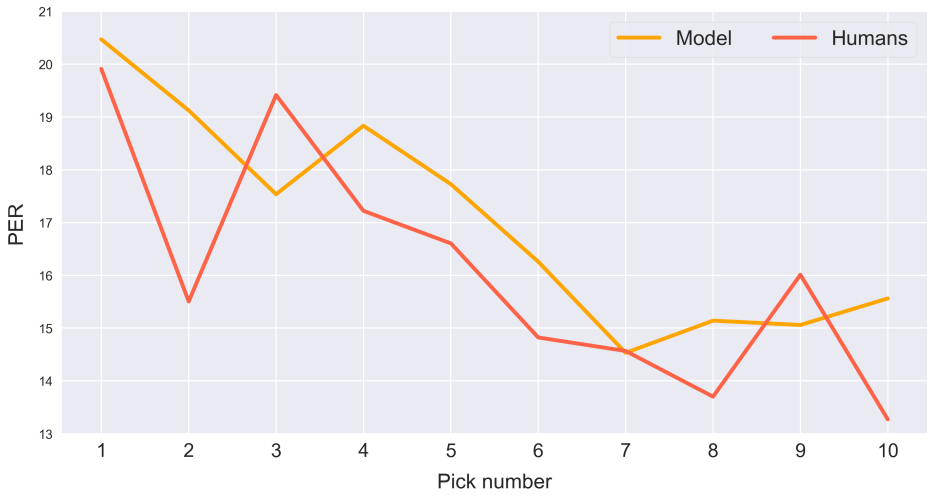


Figure 8: Average PER of each draft class, conditioned on the draft position, from the 1st pick to the 10th. 'Model' refers to the top 10 players chosen by our model, while 'Humans' refers to the top 10 players chosen by the teams.

Similarly to Figure 7, we can see that the average PER across the drafts for the second pick is much higher for our model partly due to the fact that it did not pick Adam Morrison. Again, we can see that our model 'outperforms' the humans in all the draft positions except for the third and ninth picks. On average (i.e. for the 120 players: 10 players per draft for 12 drafts), our model yields 0.92 points of PER more than the players chosen by the humans. Note that we chose the top 10 picks arbitrarily in order to focus more on the best players. We would expect our model to have a smaller PER average for the lower draft picks, since this is a zero-sum game.

5 Limitations

We have seen that our model is able to accurately predict NBA potential based on college statistics. Furthermore, using the model, we were able to evaluate the uncertainty in our predictions and to quantify the importance of college statistics. However, it is important to understand the limitations of our method and how to use this tool before making a decision.

The first thing to consider is that all models are biased in the way they learn from data and make decisions. There is also subjectivity and bias introduced by the practitioners themselves, starting from the choice of the learning algorithm and the dataset collected. In addition, while our model make reasonably good predictions on average, it does make mistakes and fail to make accurate predictions for some players. It is also not sensible to draw any conclusions for players whose predictions are very close, as it is not statistically significant.

Finally, the data contains incomplete information since we are only using quantifiable 'on-the-court' variables. For instance, many characteristics such as work-ethic, team work, leadership, lifestyle, shooting mechanics, medical history or culture fit in a particular team are not taken into account, but are of primary importance. On the other hand, scouts and front offices are well suited to judge the aforementioned characteristics. For all these reasons, our results should be used as a supplementary tool alongside human judgment during the draft process.

References

[1] C. F. Mela and P. K. Kopalle, "The impact of collinearity on regression analysis: the asymmetric effect of negative and positive correlations," *Applied Economics*, vol. 34, no. 6, pp. 667–677, 2002.

[2] R. M. O'Brien, "A caution regarding rules of thumb for variance inflation factors," *Quality & quantity*, vol. 41, no. 5, pp. 673–690, 2007.

[3] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas, "Multi-target regression via input space expansion: treating targets as inputs," *Machine Learning*, vol. 104, no. 1, pp. 55–98, 2016.

[4] "A Kaggler's guide to model stacking in practice." http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice/. Accessed: 10-01-2019.

[5] "Cross validation strategy when blending/stacking." https://www.kaggle.com/general/18793. Accessed: 10-01-2019.

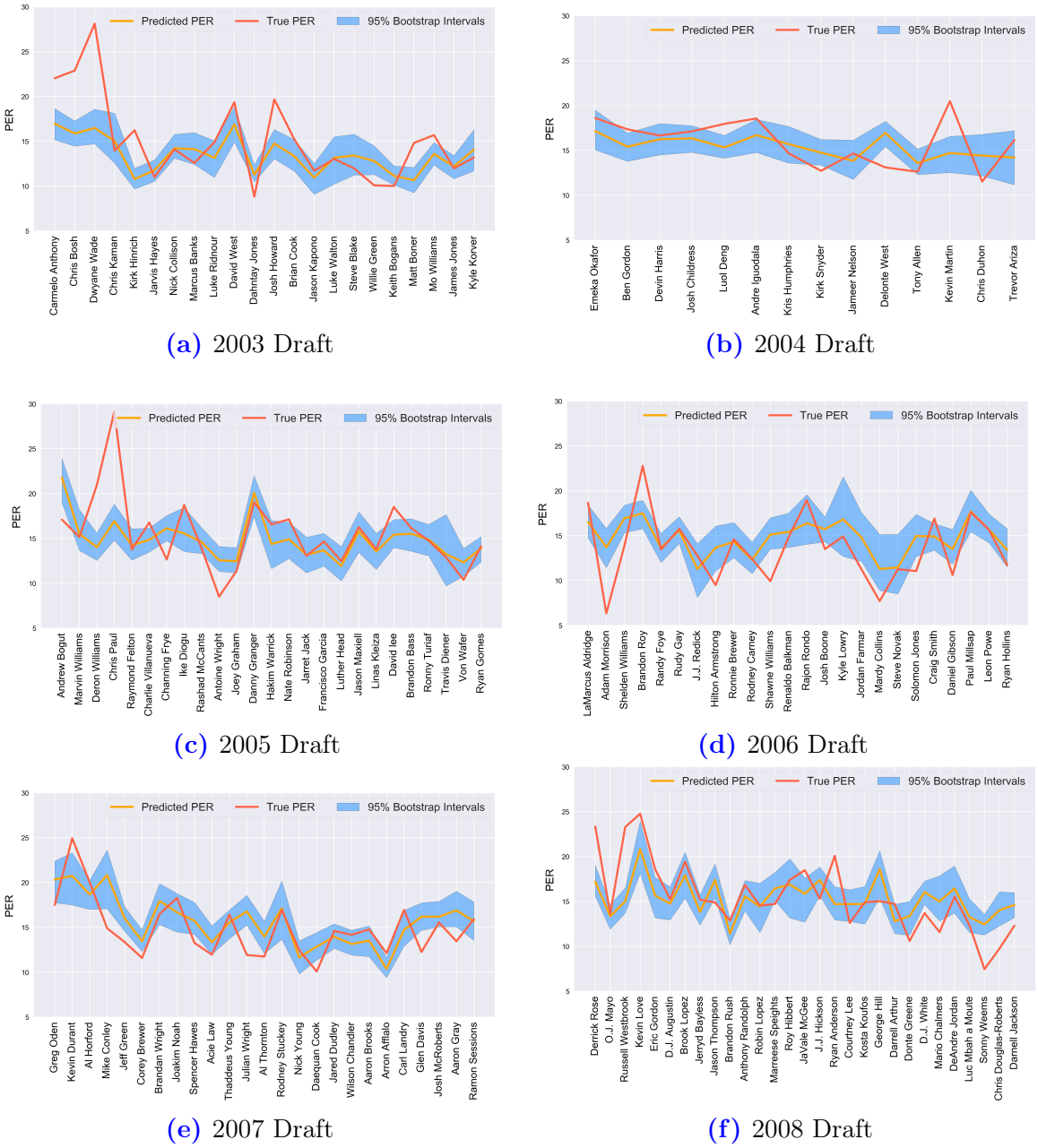
[6] "Computational statistics: The bootstrap." http://www.stats.ox.ac.uk/~caron/teaching/sb1b/slidesbootstrap.pdf. Accessed: 08-04-2019.

[7] "How to intuitively explain what a kernel is?." https://stats.stackexchange.com/questions/152897/how-to-intuitively-explain-what-a-kernel-is. Accessed: 10-01-2019.

[8] "Permutation feature importance." https://blogs.technet.microsoft.com/machinelearning/2015/04/14/permutation-feature-importance/. Accessed: 10-01-2019.

[9] "Colinearity in permutation feature importance." https://explained.ai/rf-importance/index.html. Accessed: 10-01-2019.

Appendix A

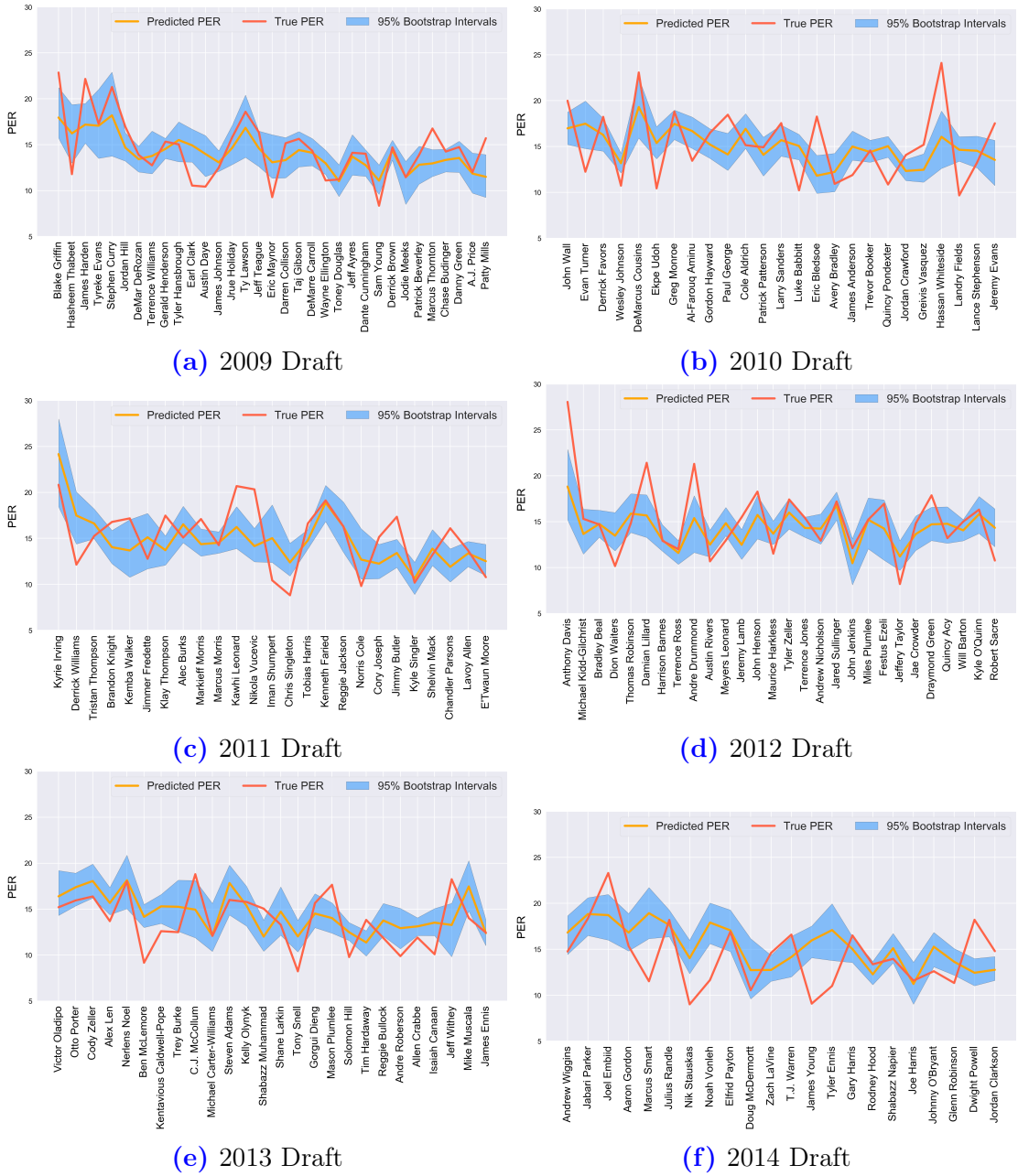


(a) 2003 Draft **(b)** 2004 Draft

(c) 2005 Draft **(d)** 2006 Draft

(e) 2007 Draft **(f)** 2008 Draft

Figure 9: Predictions from the KRR (poly) algorithm for each draft class from 2003 to 2008. Predictions are made on unseen data, where each draft class in turn serves as the validation fold.



(a) 2009 Draft **(b)** 2010 Draft

(c) 2011 Draft **(d)** 2012 Draft

(e) 2013 Draft **(f)** 2014 Draft

Figure 10: Predictions from the KRR (poly) algorithm for each draft class from 2009 to 2014. Predictions are made on unseen data, where each draft class in turn serves as the validation fold.