BASKETBALL SCORE PREDICTION

Andrew Niedringhaus (amn3wn)
B.S. Computer Science
University of Virginia
Charlottesville, VA 22903
amn3wn@virginia.edu

Jui Tao Tsai (jt2fx)
B.S. Computer Science
University of Virginia
Charlottesville, VA 22903
jt2fx@virginia.edu

Skylor Matsuda (smm7rv)
B.S. Computer Science
University of Virginia
Charlottesville, VA 22903
smm7rv@virginia.edu

September 21, 2022

ABSTRACT

Sports betting is a very popular activity in America, with companies like Fanduel and Draftkings bringing in increasing revenue each year. If sports betting continues to remain popular, Morgan Stanley predicts it will generate $7 billion U.S. dollars per year by 2025. In this experiment we analyze what factors contribute to the difference in scores(spread) and over/under that are assigned to individual events. We focus exclusively on the National Basketball Association (NBA), using statistics from players and teams during a given season to predict the spread of upcoming games. We tested how close our algorithm can predict the difference in score of a given game. After testing and training our model, we have found a solution to accurately predict a reasonable spread for a given basketball game.

1 Introduction

Every game in an NBA season has betting going on behind the scenes. People from all over the world are speculating and gambling on the outcomes of games. However, it is more than just predicting the win or loss between two teams because there is always one underdog team between two teams. If people in sports only bet on the win or loss of a game, there is not much variety within betting events. Therefore, in the sport industry, there is a notion of spread betting, where rather than predicting the win or loss of a team, people need to predict the point differential as different match ups will have varying degrees of skill spread, and as a result adds another dimension of complexity to the bet. As a result, we want to have a metric with features that we gather about the statistics such as scores, rebounds, assists, turnovers, etc to predict the exact difference in scores between two teams matching against each other. We want to train different machine learning models to find out which is the more appropriate model for predicting the difference in score match up in this context. Furthermore, we want to be able to offer sport betting centers the applications that have such model to utilize such that they are able to give a set of fair betting options where certain less probabilistic outcomes to be more valuable whereas the more probabilistic outcomes to be less valuable.

2 Method

Dataset Discussion:

We found plenty of datasets on different sport websites that offer insights into the performances of each team, individual states, etc. We decided to lean more towards using the team statistics because basketball games are team sports and usually predicting the outcome of a match up would have a higher degree of accuracy if we discuss in terms of the performances of teams.

For our supervised learning, we want to be able to accurately predict the spread of matches. As we discussed earlier, the spread is the differential between the scores of two matching teams. As a result, our y axis label for each data point is the difference between team points and opponent points since that is the definition of spread. Despite that it was not given to us on the original dataset, we created an additional variable labeled "difference". In terms of our independent

variables, there are several options of features that we can include since there are various datasets out there from the sport websites which account for quantitative and qualitative attributes of each team. For simplicity, we chose several quantitative datasets that are truly relevant within a match up. We will be discussing the datasets that we included for the features below.

First, we included the most relevant variables for a team performance such as the Field-Goals, Field-Goals percentage, Three point shot attempts and its percentage, and free throw shot attempts and its percentage, reflected in our data exploration. In order to account for more variance attributable to our model, thus increasing the power of our tests and modeling, we also included team rebounds, assists, blocks and turnovers. Since we are comparing teams performances, we also included the stats of the match-up team that a team plays against with similar stats such as the opposite team Field-Goals and its percentage, Three point shot attempts and its percentage, free throw shot attempts and its percentage, and team rebounds, assists, blocks, and turnovers. In doing so, we account for the other side of the coin as point differential will be reliant on metrics of both teams.

Machine Learning Model Discussion:

1. Linear Regression: We thought linear regression would be a good starting point as a model. Linear Regression is a simple model that will likely underfit the dataset, but can be useful when trying to predict something like scores in a basketball game because usually there is a high degree of direct correlation between the metrics of our teams. Our prediction was relatively accurate, which is logical considering linear regression is good at modelling data with linear trends, such as field goal percentage, to the number of overall points scored by each team.

2. Random Forest Regressor: We wanted to test if random forest would be effective for the problem we were trying to solve. Since linear regression performed so well on its own, we figured running an ensemble algorithm using multiple decision trees during training would outperform linear regression. After running it, it ended up being our worst performing algorithm of the three that we tried. This can likely be optimized through additional tuning.

3. Support Vector Regressor: Using our experience from codeathon 2, we figured it would be a good idea to try running a SVR on our dataset. We know that whilst using the algorithm of SVR, maximizing the margin and minimizing the cost function, which in turn segregated our data into similar components, which evolved into a powerful way to predict magnitude of the differential. We were pleased to find that it did end up being our best performing algorithm, undercutting the RMSE of our linear model by a half and our RFR model by a third.

4. Deep Neural Network: After learning about DNNs we wanted to test it out on our dataset. We figured that a DNN could be optimal due to the high degree of complexity and variability presented by the challenge of human actor prediction. DNNs can adapt well to both linear and non-linear relationships in data, and as such, it turns out that DNNs were the best performing algorithm of all, cutting our RMSE of our previous low in half yet again.

3 Experiments

For the experiments, we ran both supervised and unsupervised learning algorithms to test out both labeled and unlabeled data.

We first started with the simplest algorithm to run the dataset to predict the spread (difference in game score). Because of the nature of Linear Regression algorithm, it has a relatively higher RMSE due to overfitting but it was not that inaccurate. The RMSE in Linear Regression is 2.37.

Secondly, we used Support Vector Regressor, which is often used in classification problems often, and it does turn out to be the best among the supervised learning algorithms and we think that it is because it adjust the dataset into the appropriate dimision to compare the labeled data. After trying different kernels and tuning C parameters, We used rbf as our radial basis function kernel with C being 10 being the best parameter that yields the best result. The RMSE of Support Vector Regressor is 1.29.

Third, we used RandomForestRegressor which is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset. We used this approach in corresponding to the overfitting problem that we encounter in the Linear Regression algorithm that we ran earlier because random forest groups the dataset into different subgroups and then uses the average of the outcome to improve the predictive accuracy and control over-fitting.

However, RandomForestRegressor turns out be the least accurate outcome yet acceptable because as the model might be underfitting and the features that are being grouped together might not have worked so well. The Random Forest Regression RMSE is 4.90941. This is attributable to our model clustering our data instead of curve fitting.

One of the biggest advantages that is offered by deep learning over the aforementioned algorithms is its capacity to learn our features and fit itself closer to the true model than any of our previous predictive models could provide. Colloquially, this model is a mash-up of all of our previous models, and combines them in a manner that takes some of the best attributes of each one. This model provided a RMSE half of what our most significant model provided thus far.

The best performing experiment we ran was the Deep Neural Network.

4 Results

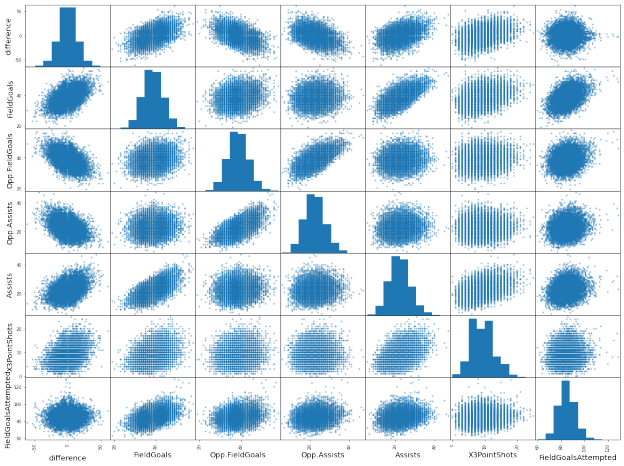
Results:

1. Deep Neural Network: RMSE = 0.483977
2. Support Vector Regressor: RMSE = 1.29587
3. Linear Regression: RMSE = 2.371628
4. Random Forest Regressor: RMSE = 4.90941

The graph of the features and the statistics of the features we used:

	Game	TeamPoints	OpponentPoints	FieldGoals	FieldGoalsAttempted	FieldGoals.	X3PointShots	X3PointShotsAttempted	X3PointShots.	FreeThrows	FreeThrowsAttempted	FreeThrows.	OffRebounds	TotalRebounds	Assists	Steals	Blocks	Turnovers	TotalFouls	Opp.FieldGoals	Opp.FieldGoalsAttempte
Game	1.000000	0.059389	0.061245	0.078181	0.058137	0.044019	0.044623	0.052515	0.011959	-0.037187	-0.036595	-0.000459	0.002806	0.017693	0.082710	-0.009256	-0.017147	-0.080715	-0.089735	0.079539	0.057872
TeamPoints	0.059389	1.000000	0.360526	0.836203	0.284428	0.710187	0.518022	0.265727	0.477539	0.318523	0.267833	0.175700	-0.011585	0.093312	0.574607	0.102418	0.056991	-0.115901	0.151177	0.269846	0.278018
OpponentPoints	0.061245	0.360526	1.000000	0.269846	0.278018	0.098300	0.138392	0.158751	0.034548	0.196832	0.192497	0.029329	0.004876	-0.036973	0.116507	-0.091221	-0.153516	0.038266	0.217472	0.836203	0.284428
FieldGoals	0.078181	0.836203	0.269846	1.000000	0.441729	0.777024	0.329092	0.109583	0.363237	-0.175325	-0.185503	0.004876	-0.000923	0.080150	0.637564	0.097681	0.062316	-0.151886	0.051371	0.219627	0.215258
FieldGoalsAttempted	0.058137	0.284428	0.278018	0.441729	1.000000	-0.216425	0.091678	0.263961	-0.136795	-0.217743	-0.217224	-0.036973	0.509138	0.424753	0.196857	0.121444	0.037162	-0.276904	0.074338	0.215258	0.220973
FieldGoals.	0.044019	0.710187	0.098300	0.777024	-0.216425	1.000000	0.685878	-0.068197	0.492585	-0.035912	-0.047040	0.030727	-0.353501	-0.206529	0.554583	0.020817	0.041642	-0.027613	0.006092	0.087331	0.080453
X3PointShots	0.044623	0.518022	0.138392	0.329092	0.091678	0.293077	1.000000	0.747013	0.685878	-0.099134	-0.115349	0.027792	-0.123974	-0.013439	0.419289	0.008089	-0.002103	-0.005752	0.033058	0.111014	0.112263
X3PointShotsAttempted	0.052515	0.265727	0.158751	0.109583	0.263961	-0.066197	0.747013	1.000000	0.068394	-0.091881	-0.099366	0.007242	0.016708	0.090278	0.226847	0.046612	-0.029048	0.039400	0.047430	0.119367	0.120457
X3PointShots.	0.011959	0.477539	0.034548	0.363237	-0.136795	0.493585	0.685878	0.068394	1.000000	-0.050268	-0.066249	0.032668	-0.201343	-0.115364	0.374811	-0.029048	0.000760	0.005081	0.005355	0.032436	0.036698
FreeThrows	-0.037187	0.318523	0.196832	-0.175325	-0.217743	-0.035912	-0.099134	-0.091881	-0.050268	1.000000	0.923855	0.331921	0.052336	0.083195	-0.153130	0.039400	0.012540	0.022667	0.201047	0.113287	0.136452
FreeThrowsAttempted	-0.036595	0.267833	0.192497	-0.185503	-0.217224	-0.047040	-0.115349	-0.099366	-0.066249	0.923855	1.000000	-0.029283	0.092457	0.091154	-0.166925	0.051204	0.015715	0.027632	0.213832	0.105879	0.138041
FreeThrows.	-0.000459	0.175700	0.029329	0.004876	-0.036973	0.030727	0.027792	0.007242	0.032668	0.331921	-0.029283	1.000000	-0.089685	-0.049788	0.015414	-0.017714	0.003262	-0.010282	0.000268	0.024164	0.017233
OffRebounds	0.002806	-0.011585	0.017356	-0.000923	0.509138	-0.353501	-0.123974	0.016708	-0.201343	0.052336	0.092457	-0.089685	1.000000	0.584383	-0.115555	0.035432	0.007817	0.044670	0.046376	0.005993	-0.044799
TotalRebounds	0.017693	0.093312	-0.277404	0.080150	0.424753	-0.206529	-0.013439	0.090278	-0.115364	0.083195	0.091154	-0.049788	0.584383	1.000000	0.014261	-0.109771	0.167239	0.125667	0.005050	-0.261143	0.320054
Assists	0.082710	0.574607	0.116507	0.637564	0.196857	0.554583	0.419289	0.226847	0.374811	-0.153130	-0.166925	0.015414	-0.115555	0.014261	1.000000	0.093768	0.087635	-0.032052	-0.004874	0.184091	0.155268
Steals	-0.009256	0.102418	0.091221	0.097681	0.121444	0.020817	0.008089	0.046612	-0.029048	0.039400	0.051204	-0.017714	0.035432	-0.109771	0.093768	1.000000	0.005441	0.125748	0.027803	-0.103220	-0.175084
Blocks	-0.017147	0.056991	-0.153516	0.062316	0.037162	0.041642	-0.002103	-0.005608	0.000760	0.012540	0.015715	0.003262	0.007817	0.167239	0.087635	0.005441	1.000000	0.033009	-0.011439	-0.142266	0.258940
Turnovers	-0.080715	-0.115901	0.038266	-0.151886	-0.276904	0.027613	-0.005752	-0.012005	0.005081	0.022667	0.027632	-0.010282	0.044670	0.125667	-0.032052	0.125748	0.033009	1.000000	0.166297	0.036841	0.134820
TotalFouls	-0.089735	0.151177	0.217472	0.051371	0.074338	0.006092	0.033058	0.047430	0.005355	0.201047	0.213832	0.000268	0.046376	0.005050	-0.004874	0.027803	-0.011439	0.166297	1.000000	-0.133546	-0.136791
Opp.FieldGoals	0.079539	0.269846	0.836203	0.219627	0.215258	0.087331	0.111014	0.119367	0.032436	0.113287	0.105879	0.024164	0.005993	-0.261143	0.184091	-0.103220	-0.142266	0.036841	-0.133546	1.000000	0.441729
Opp.FieldGoalsAttempted	0.057872	0.278018	0.284428	0.215258	0.220973	0.080453	0.112263	0.120457	0.035698	0.136452	0.138041	0.017233	-0.044799	0.320054	0.155268	-0.175984	0.258940	0.134820	-0.136791	0.441729	1.000000
Opp.FieldGoals.	0.045632	0.098300	0.710187	0.087331	0.080453	0.037220	0.040118	0.044611	0.008392	0.029185	0.020658	0.013527	0.038295	-0.506287	0.003510	0.009620	-0.333857	-0.054332	-0.047338	0.777024	-0.216425
Opp.3PointShots	0.045094	0.138392	0.518022	0.111014	0.112263	0.040118	0.086748	0.115195	0.004273	0.042949	0.040671	0.004444	-0.032134	-0.096056	0.043603	-0.008035	-0.128773	-0.035222	-0.101793	0.329092	0.091678
Opp.3PointShotsAttempted	0.052239	0.168751	0.265727	0.119367	0.120457	0.044611	0.115195	0.142953	0.021016	0.053237	0.051995	0.008472	-0.054321	0.123637	0.075047	0.001993	-0.083209	0.029536	-0.071186	0.199583	0.263961
Opp.3PointShots.	0.012826	0.034548	0.477539	0.032436	0.035698	0.008392	0.004273	0.021016	-0.020231	0.013225	0.011773	-0.002838	0.011654	-0.262272	-0.020319	-0.011480	-0.107660	-0.073106	-0.074101	0.363237	-0.136795
Opp.FreeThrows	-0.035976	0.196832	0.318523	0.113287	0.136452	0.029185	0.042949	0.053237	0.013225	0.184051	0.189033	0.016390	0.044460	-0.068021	0.036127	-0.007415	0.003910	0.037067	0.726398	-0.175325	-0.217743
Opp.FreeThrowsAttempted	-0.035012	0.192497	0.267833	0.105879	0.138041	0.020658	0.040671	0.051995	0.011773	0.189033	0.196992	0.009400	0.050168	-0.012068	0.036588	-0.005369	0.010110	0.051572	0.796039	-0.185503	-0.217224
Opp.FreeThrows.	-0.001151	0.029329	0.175700	0.024164	0.017233	0.013527	0.004444	0.008472	-0.002838	0.016390	0.009400	0.024936	-0.010278	-0.148038	-0.000819	-0.013070	-0.023272	-0.024940	-0.042603	0.004876	-0.036973
Opp.OffRebounds	0.002942	0.017356	-0.011585	0.005993	-0.044799	0.038295	-0.032134	-0.054321	0.011654	0.044460	0.050168	-0.010278	-0.019325	-0.057418	0.017860	0.037814	0.230188	0.043966	0.078206	-0.000923	0.509138
Opp.TotalRebounds	0.019189	-0.277404	0.093312	-0.261143	0.320054	-0.506287	-0.096056	0.123637	-0.262272	-0.068021	-0.012068	-0.148038	-0.057418	0.025957	-0.176137	0.074825	0.125134	-0.184277	0.047793	0.080150	0.424753

The graph of correlations between features



5 Conclusion

Our hypothesis was to find if there was a way to accurately predict a basketball games outcome, and it appears that we were able to find an effective solution to this problem. We have tried different machine learning algorithms with both supervised and unsupervised approaches to build models on top of the dataset we found in sports center of team performance including the statistics for both teams matching against each other. We found that Support Vector Regressor performed the best among the supervised learning algorithms because it adjusts the dataset to the appropriate dimension while Linear Regression had a over-fitting problem and Random Forest Regressor had a grouping problem.

While our focus when training our model was on the NBA as a whole, our findings are super useful for college basketball teams in Virginia and the Washington Wizards. This will be very useful for people who put wagers on teams in Virginia, and know if what they are betting on has a high or low probability on being correct. This model could also be used for little league competitive teams, to ensure that when teams are made at the beginning of the season that they are made as equal and fair as possible. Our model fits a generic case where we have much relevant data for an event that is unpredictable, and we hope our use of basketball data can be extrapolated to other similar instances.

A possible shortcoming would be data aggregation as there are a finite number of games to be played in a season. We were able to find useful quantitative data on team and player stats, but it would be even more useful if there was a way to get qualitative data as well. For instance, if there were a way to interview players before games and see their mood, if they had any nagging injuries/bruises, team chemistry, etc, this could further increase the accuracy of our predictions. Even if a team is playing extremely well, we all know fatigue and off-court activities can have an immense effect on what happens on the court. If this experiment were to continue, it would be super interesting to find out the effect subjective data has on the outcome of games.

6 References

Kelepouris, I. (2018). NBA Team Game Stats from 2014 to 2018. https://www.kaggle.com/ionaskel/nba-games-stats-from-2014-to-2018
Sbr. (2019, November 15). Morgan Stanley Boosts US Sports Betting Market Forecasts To $7 Billion by 2025. https://www.sportsbookreview.com/sports-betting-news/morgan-stanley-boosts-us-sports-betting-market-forecasts-7-billion-2025-85463

7 Contribution

Andrew Niedringhaus (amn3wn):
Introduction, Results, Conclusion, Models/Algorithms

Jui Tao Tsai (jt2fx):
Abstract, Method, Experiments, Data Visualization

Skylor Matsuda (smm7rv):
Coding, Model/Algorithms. Cleanup