Predicting NBA team wins using Machine Learning

Andy Hughes

North Carolina State University

DSC 412: Exploring Machine Learning

Instructor Trenton Embry

November 29, 2024

Abstract

In this project, I built several regression models to predict a nba team's winning percentage over a single season given 19 team statistics recorded over the course of a season. A linear regression model had a top correlation score of 0.80 while a linear support vector regressor had a top correlation score of 0.77. A multilayer did not perform well but had a top correlation score of 0.30. One limiting factor in this project was the number of data points was only 180.

Background

The national basketball association generates billions of dollars of annual revenue. As technology advances, teams increasingly rely on data analytics to find ways of improving their winning percentage and team revenues. This project seeks uses publicly available team season data to predict a team's winning percentage in a given season. The features used to train models are team, win_percentage, minutes_per_game, points_per_game, two_pointers_made_per_game, two_pointers_attempted_per_game, three_pointers_made_per_game, three_pointers_attempted_per_game, free_throws_made_per_game, free_throws_attempted_per_game, offensive_rebounds_per_game, defensive_rebounds_per_game, assists_per_game, turnovers_per_game, steals_per_game, blocks_per_game, blocks_attempted_per_game, personal_fouls_per_game, and season. I applied statistical methods such as stratified random sampling to only sample a given team once during every 4 seasons. This helps overcome bias present when winning teams stick together and losing teams disintegrate.

Data Analysis

I used exploratory data analysis techniques to look at characteristics of winning and losing teams. Certain columns such as point differential are tightly related to winning percentage. Looking at a correlation matrix helped determine which variables were highly correlated and could be flattened or condensed. Finding an average of the correlation values with the output variable was useful in developing a multi-layer perceptron later in this project. The features over the average correlation of 19.5 were points per game, defensive rebounds per game, assists per game, turnovers per game, blocks per game, blocked attempts per game, and personal fouls per game. One key point discovered is that the scoring statistics (besides points per game) were not heavily correlated with winning percentage. Also, points per game is positively correlated with higher three-point attempts/makes. Three point attempts/makes and two point attempts/makes are inversely correlated since a team only has three choices of scoring and a team's average possessions per game is roughly constant for all teams (not in the data set).

Data Augmentation

I decided to use stratified random sampling to only sample each team once every 4 years. It makes sense that good teams would stick together but it also introduces bias. The best teams rely on superstars who may help create statistical anomalies and skew the average data. A few teams had to relocate cities. In most cases I chose to model a team using its city instead of the franchise. Also, for the multi-layer perceptron model's data, I removed all of the features which had weaker than average correlation with winning_percentage.

Model Selection

The first model I chose was a linear regression model. It is most familiar to me since it is used in statistics and is centuries old. I also tried a linear support vector machine regression model. We completed homework assignments using both of these models, so it was easy to justify their use. On the contrary, I used a multi-layer perceptron to be able to see if a complex model could also predict well.

Training Methodology

I trained the data using hyperparameter tuning and a 80/20 test training data split. Two columns "season" and "team" were transformed into categorical variables in the data frame. Then the models would predict the output for the test inputs. And calculate the r^2 scores and error between the predicted test data and the actual test data. For the linear regression and linear support vector regression model, using different scalars such as RobustScalar, StandardScalar, MinMaxScalar, and Normalizer in Scikit-learn seemed to affect performance. In all models, the robust scalar lead to having the best r^2 score. I used a GridSearch with the multi-layer perceptron with different hidden layer sizes, activation functions, alpha values, and learning rates. It is interesting to think about how the hyperparameters connect with the data model.

Results

The linear regression and linear support vector regression models work great and have around a 0.8 correlation coefficient r^2 score. In statistics, this is a medium strong correlation. Many features were linearly independent (little collinearity), yet helped explain a portion of the winning percentage. Furthermore, simple model types generalize well on datasets with a low

datapoint to feature ratio. Theoretically a multi-layer perceptron should detect linear relationships, but 0.4 was the best score that my model could produce. By increasing the dataset size, the neural network-based model could hopefully achieve similar results to the linear models.

Future Work

The model performed well considering the imposed constraints of the data. It is tough to improve the current project without breaking fair data use laws. Mentions of running statistical models and/or creating a database from various statistics were strictly prohibited. I could try other linear machine learning models such as k means clustering, logistic regression, decision tree regressor, etc. Or I could eliminate the stratified sampling to introduce more bias and train using the neural network model. In the future, I would like to experiment more with larger datasets, participate in Kaggle competitions, and engage in learning more theory about machine learning.

Stakeholder Acknowledgements

Stakeholders might be an average nba enthusiast, sports analyst, or possibly gambling bookie. If one truly believes in the data models and that feature weights impact winning percentage, then the results could be useful. There is nba futures betting which can be used to gamble on a team's ending regular season standing. However, the statistical models used by analysts or gambling bookies are larger and more sophisticated than mine. In practice, the model does not predict the future in any way. It uses the features as explanatory variables to explain the winning percentage outcome. To use as a predictor for a given winning percentage, one would need to predict the

feature variables to use as inputs. On another note, the models will be more volatile during the beginning of the season since the averages are more influenced.

Citations

"API Reference". Scikit learn

https://scikit-learn.org/1.5/api/index.html

"Michael H". (2024). *NBA Team Stats*. Kaggle https://www.kaggle.com/datasets/mharvnek/nba-team-stats-00-to-18