Classification Write Up

All-NBA Player Classification

Derek Call

Abstract

Being one of the 15 players chosen to be a part of an All-NBA team each year is a great honor. Being considered top 3 at their position in the NBA denotes they are an elite player in the league. It also comes with the possibility of increased pay. Once selected to an All-NBA team a player will become eligible to sign a super-max contract on their next extension or signing. This allows a player to receive 5-10% more of a team’s salary cap each year, with annual 8% increases, which is worth 10’s of millions of dollars. Being able to classify which players will and will not be selected to All-NBA teams is of utmost importance to teams trying to budget and plan for the future, while also notifying players of their options in Free-Agency.

Design

Using data gathered through a free NBA API, I was able to calculate seasonal player averages for more than 20 statistics. This data was used to feed into many different classification models.

Data

Data was previously gathered by use of a free NBA API found at (<https://rapidapi.com/theapiguy/api/free-nba/>). Historical All-NBA selection data was scraped using Selenium and Beautiful soup from (<https://www.nba.com/news/history-all-nba-teams> ) . Using calculated season player averages we were able to partition the data for model training to classify which players are considered All-NBA.

Algorithms and Feature Engineering

Using Selenium through Python and BeautifulSoup I was able to grab all the All-NBA selection data to combine with the previous collected NBA statistics data. Python was used for EDA and model testing. Created season averages of player statistics, along with a winning percentage feature.

Models

I split the data into training + validation data (80%) and test data (20%). I used SMOTE to introduce additional minority observations as the data was very imbalanced (the positive target class was only 3.3% of the observations). Random Forest was able to show the feature importance’s while the Logistic Regression provided the feature coefficients. These two models performed best, with the Gaussian Naïve Bayes model being the least predictive of the three.

Findings

Training the models and the validating on the holdout set provided high accuracy scores, though that is not a good measurement metric due to the imbalances in the data. The better metric was found to be the F1 Score, which is a good balance between precision and recall. The probability threshold with the highest F1 score was 0.925, so high. The precision score at this threshold is 0.648 with the recall score being 0.776. In this instance recall is more important as there will always be players not selected to the All-NBA teams, but the classification of the most likely is important.

The general takeaway is that playing and scoring efficiently are the best indications of an All-NBA classification. All shooting metrics are negatively correlated, while points and free throws are positively correlated. As long as you score well without taking too many shots your odds of being selected to an All-NBA team go up.

The most import features and their coefficients in the final model are:

Blocks: 1.2032,

Turnovers: -0.8474,

Fouls: -0.3011,

Steals: 0.567,

Minutes: -0.0956,

Assists: 0.6852,

Rebounds: 0.2297,

Field Goal Attempts: -0.3931,

Field Goal Makes: -0.5109,

Free Throw Attempts: -0.2163,

Winning Percentage: 10.5354,

Free Throw Makes: 0.5176,

Points: 0.9659

How to interpret coefficients for Log Odds:

For every 1 increase in a season steal average ≈ 0.567 ⟹e0.567 ⟹ 1.76 times the odds of being named to an All-NBA team.

For every 1 increase in a season turnover average ≈ -0.8474⟹e-0.8474 ⟹ 0.429 times the odds of being named to an All-NBA team.

Tools

* Selenium and BeautifulSoup Python packages for html retrieval and parsing
* Pandas, Numpy, and SciKitLearn Python packages for data cleaning and modeling
* Matplotlib and Seaborn Python packages for data visualization