Diana Taurasi’s Sweet Spot

By Melanie Ackerman

Abstract

Diana Taurasi is the best player in the WNBA. She is buried in accolades and broken records and was just named the WNBA GOAT (greatest of all time). The Phoenix Mercury coaching staff is looking to capitalize on her scoring abilities. They want to know if her scoring productivity is related to where on the court she shoots from in a given game. The coaches will use this information to inform new plays intended to set Taurasi up to score. This project is a continuation of the Metis linear regression project in which I regressed the percentage of shots made from different zones on the court on the number of points Taurasi made in a game. Building on that project, I now look at how different spots on the court relate to whether she makes a shot.

Design

This project answers hypothetical client Phoenix Mercury’s question of where DT is most successful shooting from. The shot-by-shot data were scraped from www.stats/wnba.com. I created dummy variables for categories from the shot area variable and shot range variable and included a discrete variable for period of the game as features. The target is the made shot flag. I test three classification models: k-nearest neighbors using only court coordinates of each shot, logistic regression using the aforementioned features and targets, and a random forest classifier.

Data

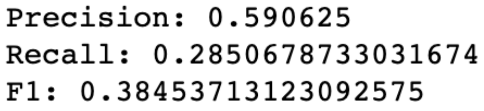
I use shot tracking data that I scraped from www.stats/wnba.com. In the raw data, each observation is one shot attempt made by Taurasi and includes information such as the pre-designated court zone, shot type, time remaining in the game, game date, and x- and y-coordinates of the shot.

Algorithms

Models: I test k-nearest neighbors, logistic regression, and random forest classifiers before settling on the random forest as the best model. Though it performed similarly to the logistic regression, the interpretability of the decision tree makes it best suited in this context.

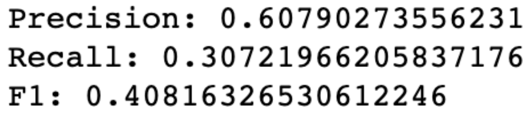
Model evaluation: I partitioned the entire dataset into 80/20 train vs. holdout and further partitioned the training set into 75/25 train vs. validation.

The following model performance metrics for the random forest classifier were calculated on the training and validation sets only:



```
Precision: 0.590625
Recall: 0.2850678733031674
F1: 0.38453713123092575
```

The scores on the final random forest model were only run at the end on the holdout set:



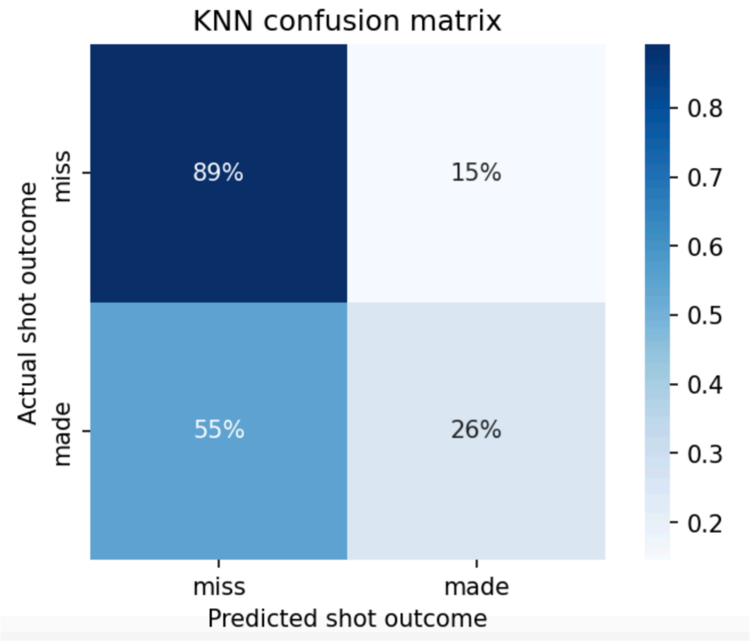
```
Precision: 0.60790273556231
Recall: 0.30721966205837176
F1: 0.40816326530612246
```

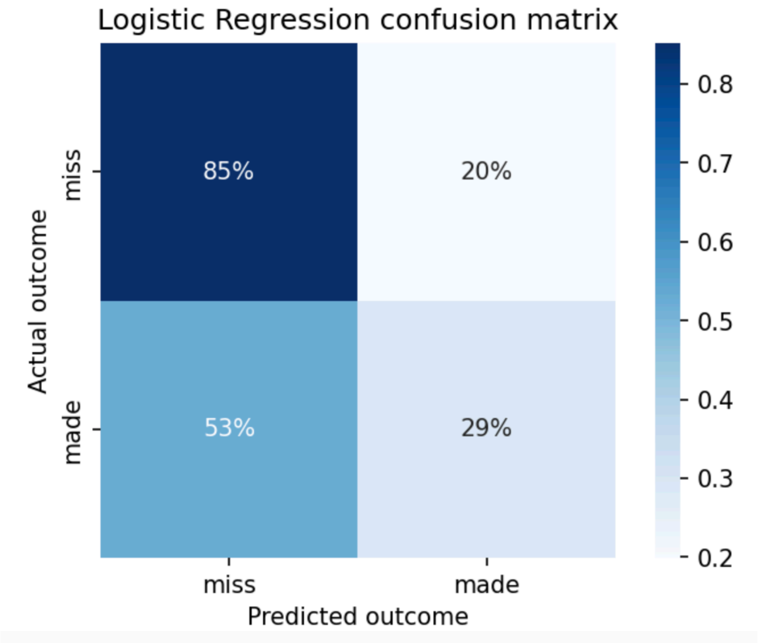
Tools

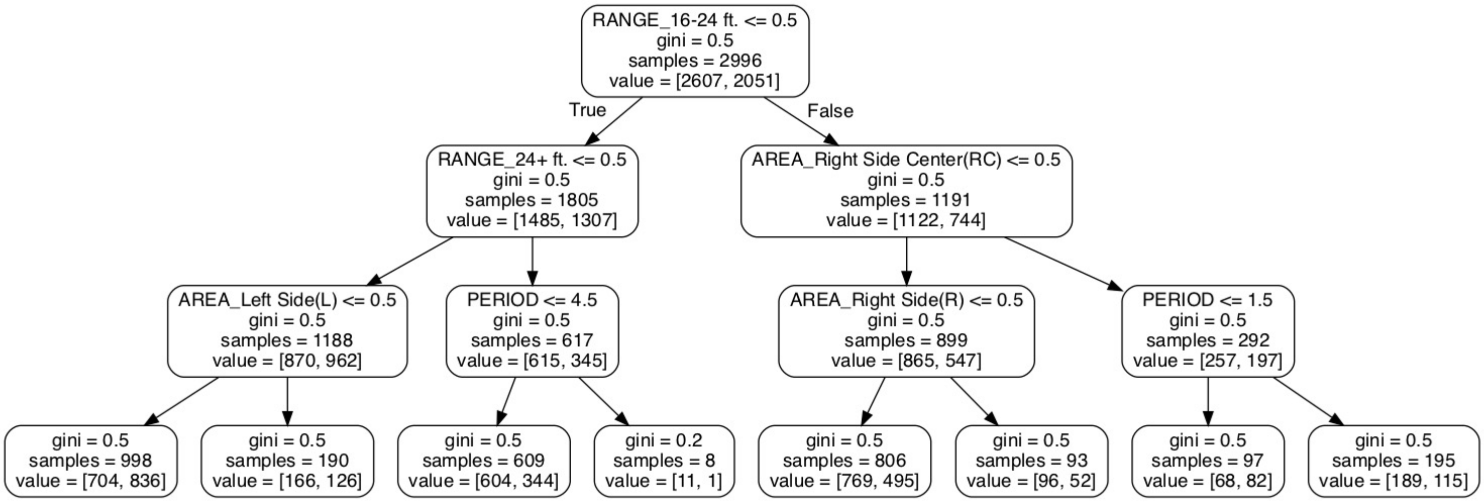
Requests and JSON for web-scraping
NumPy and Pandas for data manipulation
Seaborn and GraphViz from scikit-learn for visualization
Scikit-learn for models

Communication

I produced slides containing the following confusion matrices and random forest decision tree visualization:




Logistic Regression confusion matrix
Actual outcome
miss
made
85%
20%
53%
29%
miss
made
Predicted outcome
0.8
0.7
0.6
0.5
0.4
0.3
0.2


RANGE_16-24 ft. <= 0.5
gini = 0.5
samples = 2996
value = [2607, 2051]
True
False
RANGE_24+ ft. <= 0.5
gini = 0.5
samples = 1805
value = [1485, 1307]
AREA_Right Side Center(RC) <= 0.5
gini = 0.5
samples = 1191
value = [1122, 744]
AREA_Left Side(L) <= 0.5
gini = 0.5
samples = 1188
value = [870, 962]
PERIOD <= 4.5
gini = 0.5
samples = 617
value = [615, 345]
AREA_Right Side(R) <= 0.5
gini = 0.5
samples = 899
value = [865, 547]
PERIOD <= 1.5
gini = 0.5
samples = 292
value = [257, 197]
gini = 0.5
samples = 998
value = [704, 836]
gini = 0.5
samples = 190
value = [166, 126]
gini = 0.5
samples = 609
value = [604, 344]
gini = 0.2
samples = 8
value = [11, 1]
gini = 0.5
samples = 806
value = [769, 495]
gini = 0.5
samples = 93
value = [96, 52]
gini = 0.5
samples = 97
value = [68, 82]
gini = 0.5
samples = 195
value = [189, 115]