Data-Driven Decision Making for Clutch-Time Shot Selection: Analyzing Player Performance Using Predictive Modeling

Wei-Chieh Chen Peiya Qi Yash Gaikwad

2024-11-13

Abstract

This project leverages NCAA Men's Basketball data to identify the best Purdue player for taking critical shots in high-pressure moments. Focusing on the final five minutes of close games, the analysis considers factors like score difference, time remaining, and win probability to predict shot success. The Random Forest model emerged as the most effective predictor, accurately capturing the conditions linked to successful shots. Based on the model's predictions, Zach Edey was identified as the optimal player for game-deciding moments, offering a data-driven approach to enhance strategic decision-making in pivotal basketball scenarios.

Introduction

In NCAA basketball, the final moments of close games are pivotal, and deciding which player should take the last shot can significantly impact the outcome. This study aims to offer a data-driven recommendation for the optimal player on the Purdue Men's Basketball team to take a winning shot in high-pressure moments. By analyzing player performance under specific game conditions—such as time remaining, score differential, and possession—the project seeks to develop a reliable model to predict shot success during clutch situations. This model will ultimately provide actionable insights for in-game decision-making.

Data Pre-Processing & Exploratory Data Analysis

Data Preparation

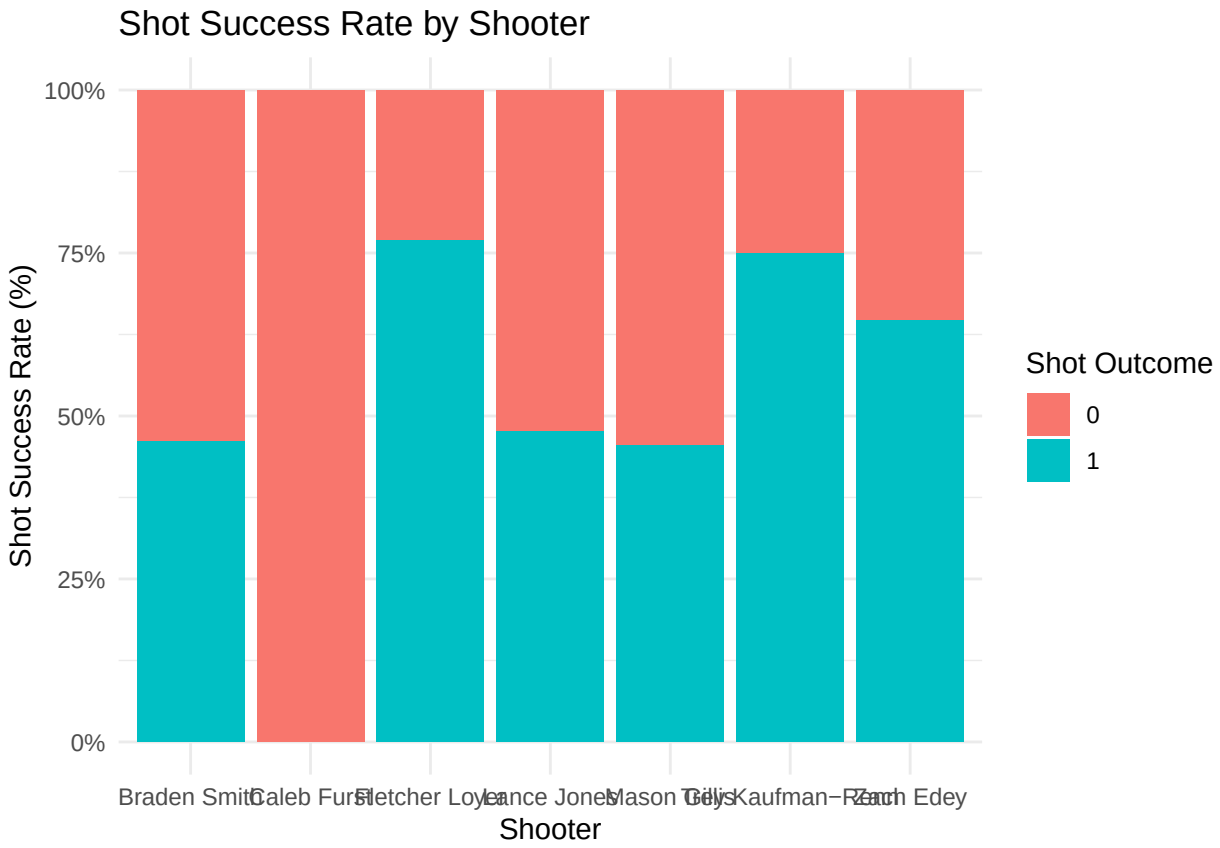
Clutch-time data from the NCAA Men's Basketball Division 1 2023-2024 season was filtered for analysis, focusing on situations in the final five minutes of the game where the score difference was within five points. Variables included score_diff, secs_remaining, win_prob, shooter, and others related to game conditions and player actions. The outcome variable, shot_success, was binary, marking each shot as "Success" or "Fail".

Exploratory Data Analysis

The exploratory data analysis (EDA) provides visual insights into the factors influencing shot success under clutch conditions. The following visualizations were used to understand player performance and contextual factors:

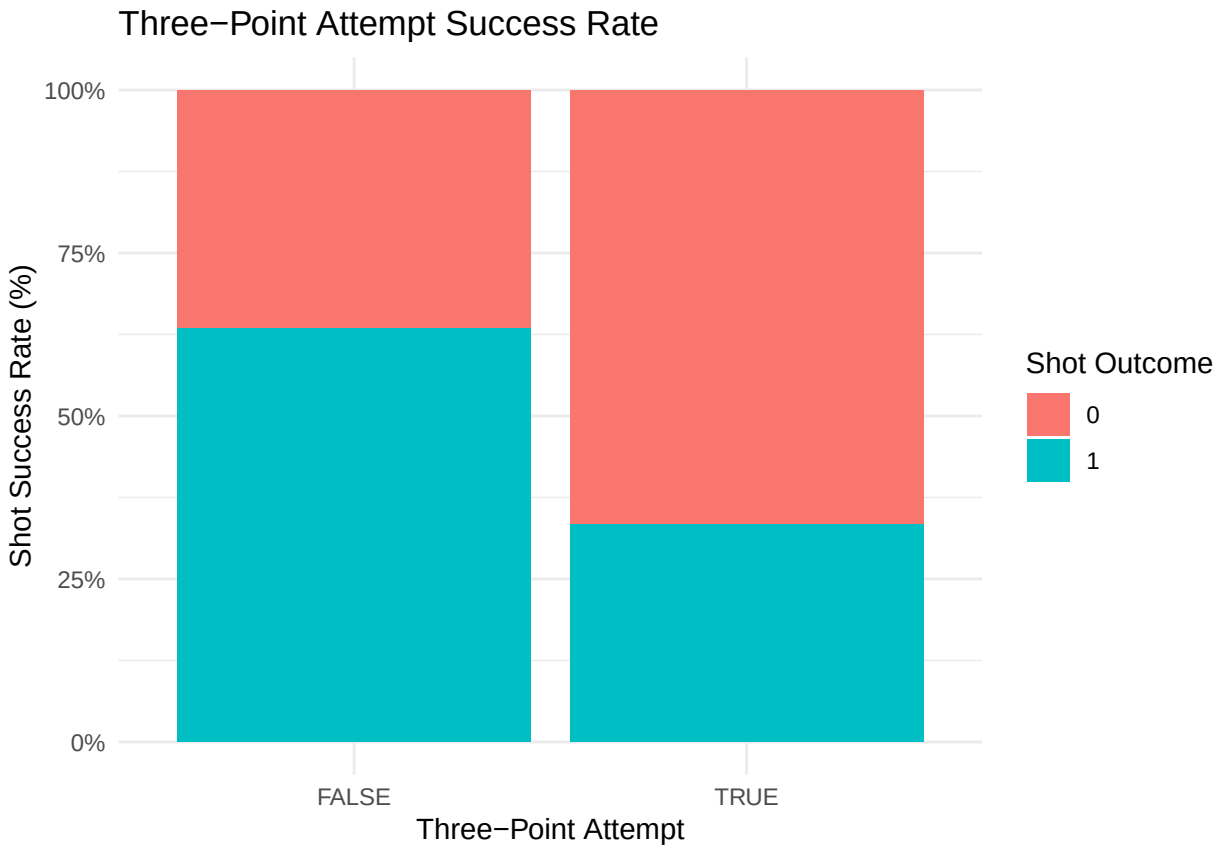
1. Distribution of Shot Success by Shooter:

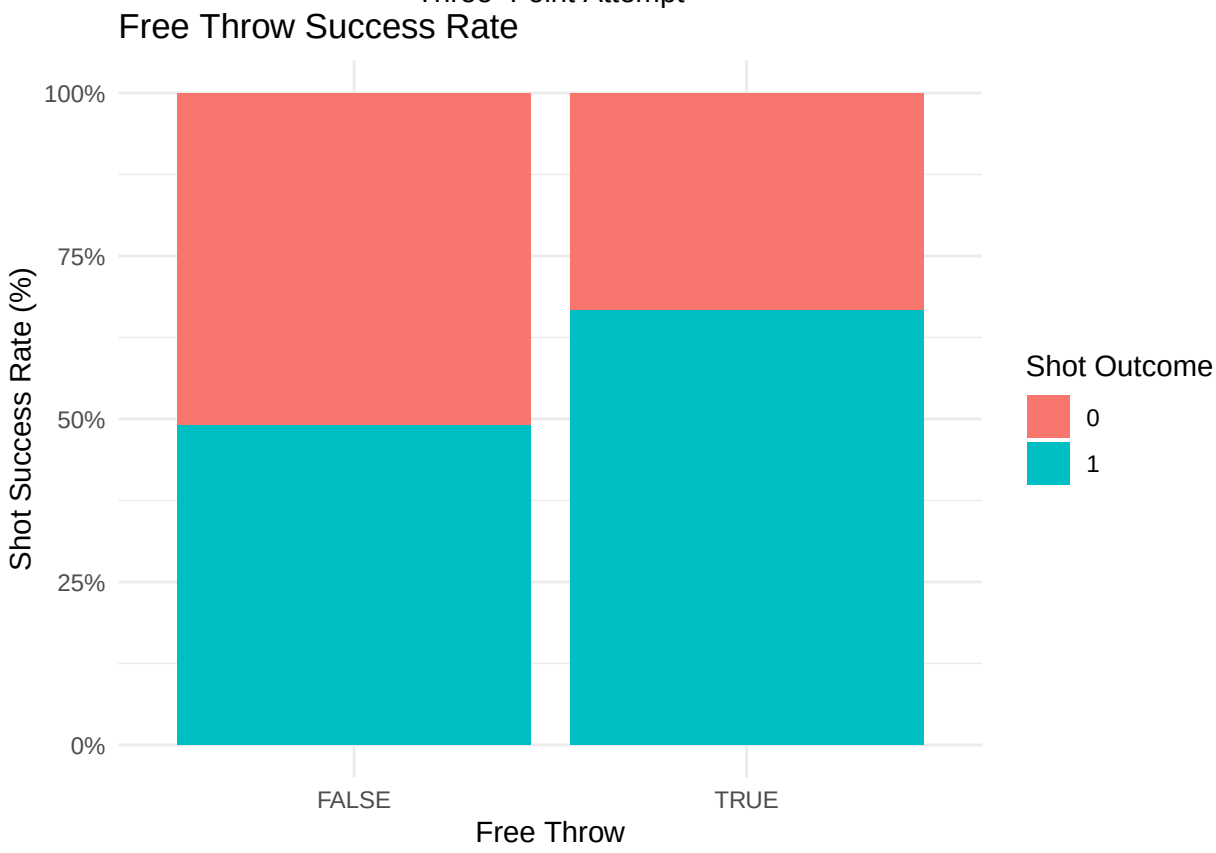
This bar plot shows the percentage of successful versus failed shots for each player. A higher proportion of success in any player's bar indicates a greater reliability in clutch situations. This plot helps to quickly assess individual players' shot success rates, which can guide decisions on which players are most effective in scoring under pressure.



2. Distribution of Three-Point Shots and Free Throws:

The plots for Three-Point Attempt Success Rate and Free Throw Success Rate provide insights into players' strengths with specific types of shots. A high success rate in the three-point plot indicates a player who is effective from beyond the arc, while a high success rate in the free throw plot highlights players who are reliable at the line. This breakdown is useful for understanding which shot types maximize each player's scoring potential.

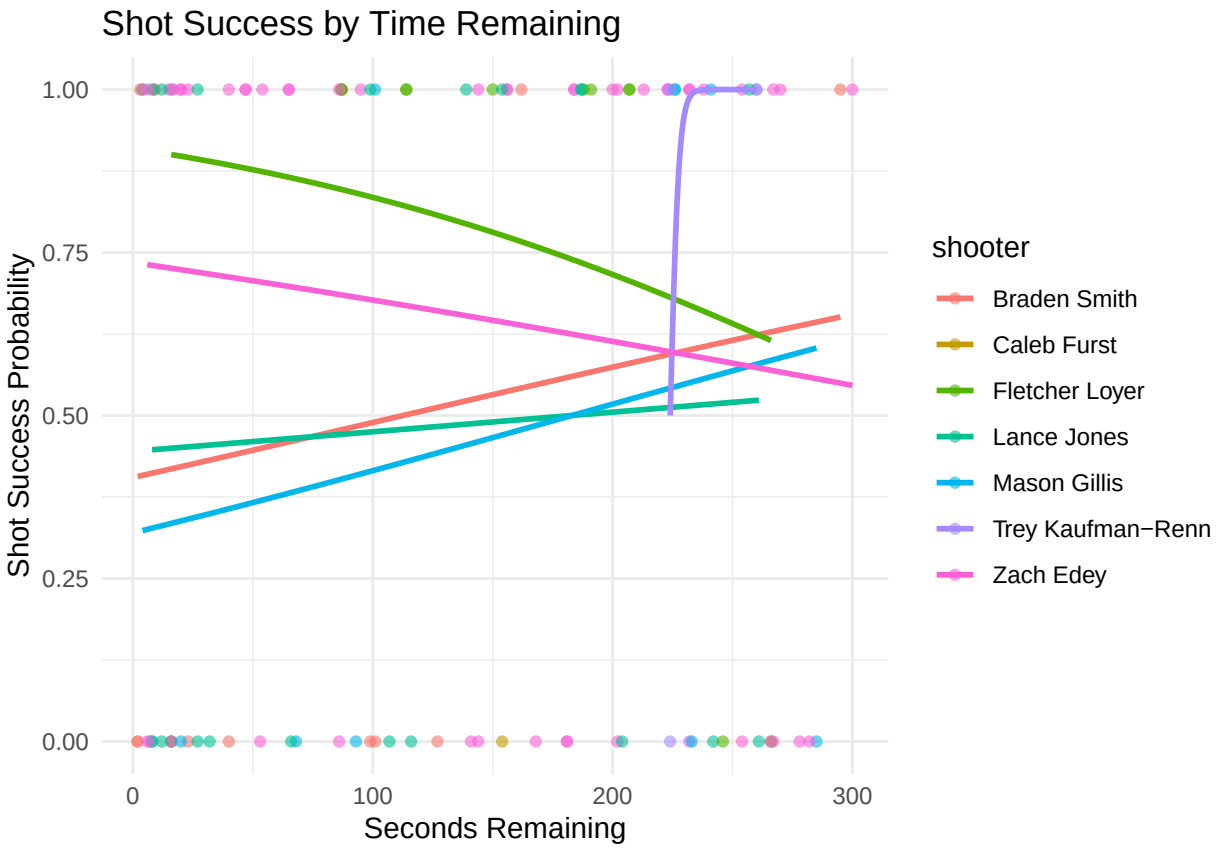




3. Time Remaining and Shot Success:

This scatter plot, with a trend line, shows how shot success probability changes as time counts down in the last five minutes of the game. If the trend line remains steady or increases, it indicates that players maintain or improve shot success as time decreases, showcasing resilience under time pressure. Conversely, a downward trend could indicate decreasing accuracy as pressure builds, suggesting time sensitivity in shot effectiveness.

```
## `geom_smooth()` using formula = 'y ~ x'
```

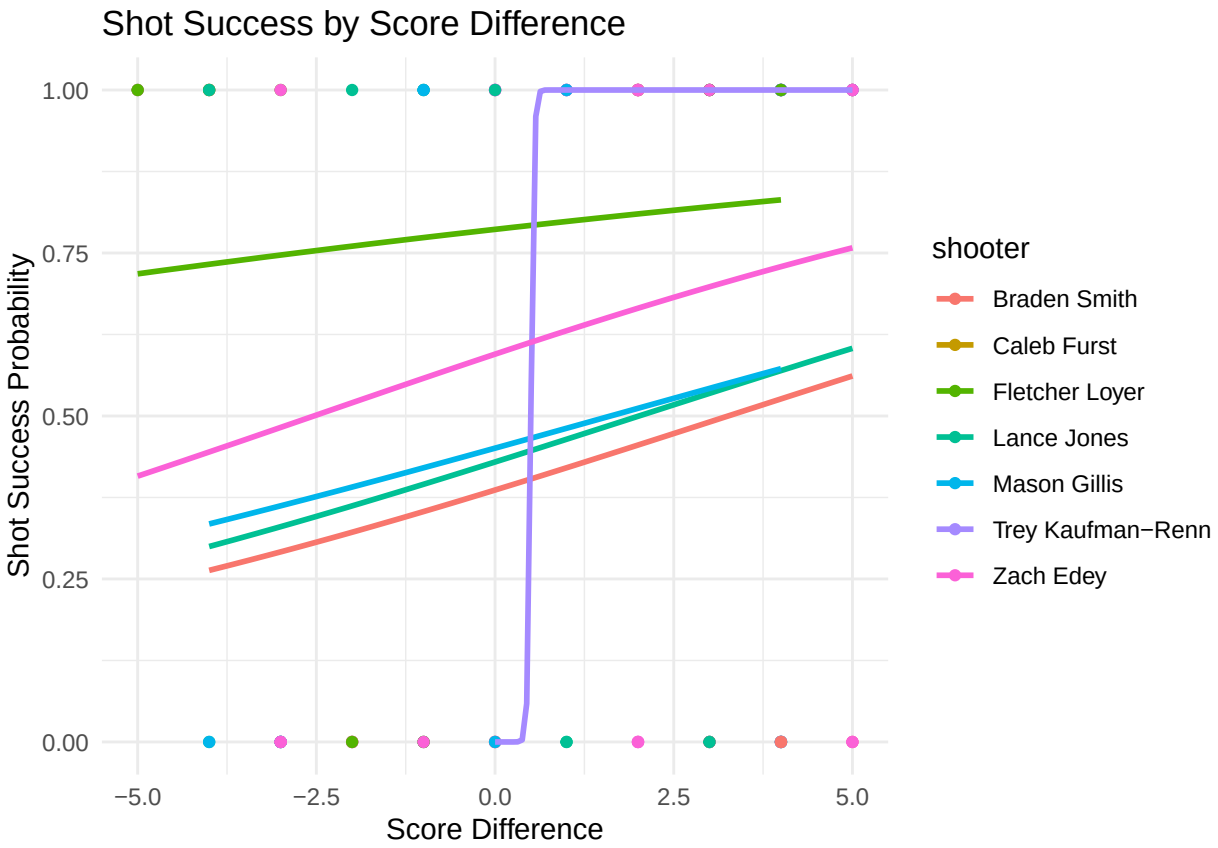


4. Shot Success by Score Difference:

In this plot, shot success is analyzed against the score difference between teams. A consistent trend line across different score differences suggests that players' shooting accuracy is stable regardless of the game's competitiveness. A steep slope could imply that players perform differently based on the closeness of the game, either thriving under pressure or becoming less effective as games get tighter.

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```



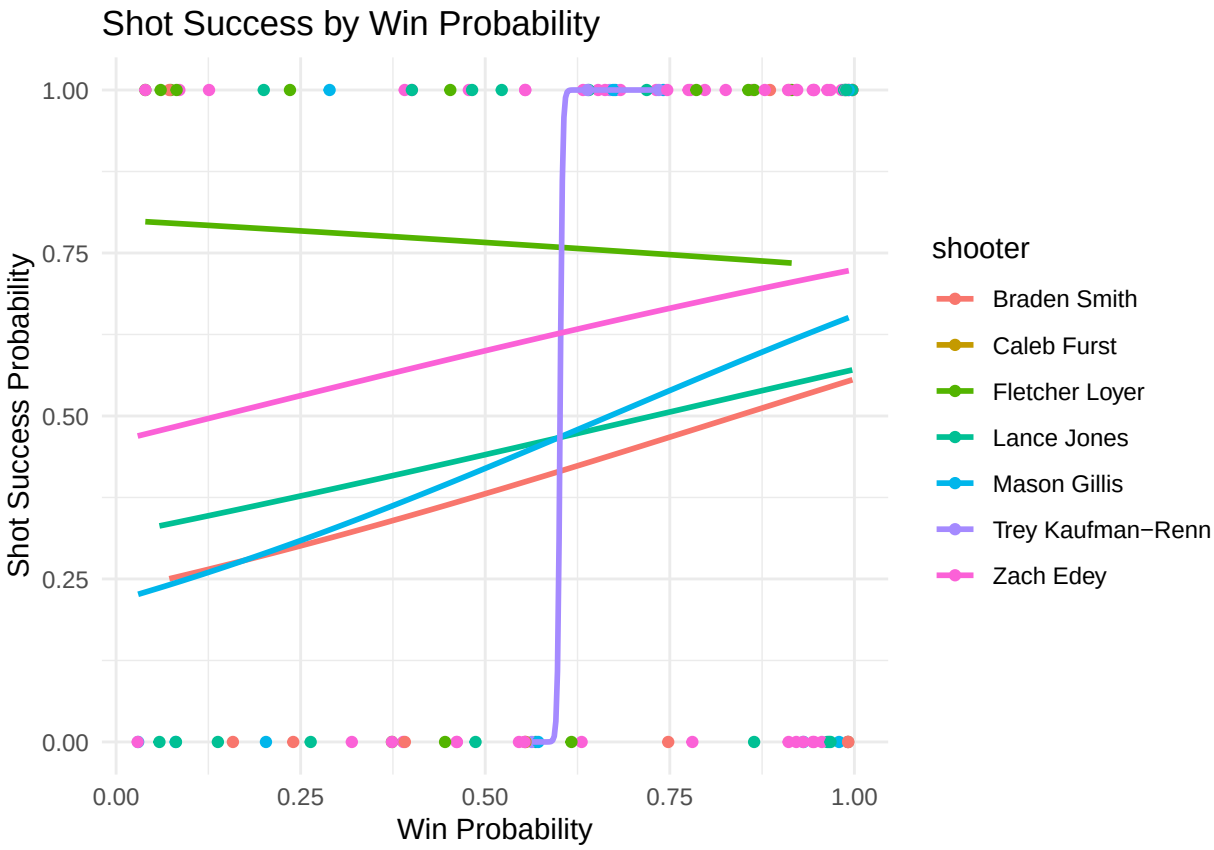
5. Shot Success Rate by Game Situation:

The Shot Success by Win Probability plot reveals how players perform relative to the predicted chances of their team winning. A stable trend here suggests that players' effectiveness is independent of the game's expected outcome, while variability could indicate that certain players perform better when the odds are either in or against their favor.

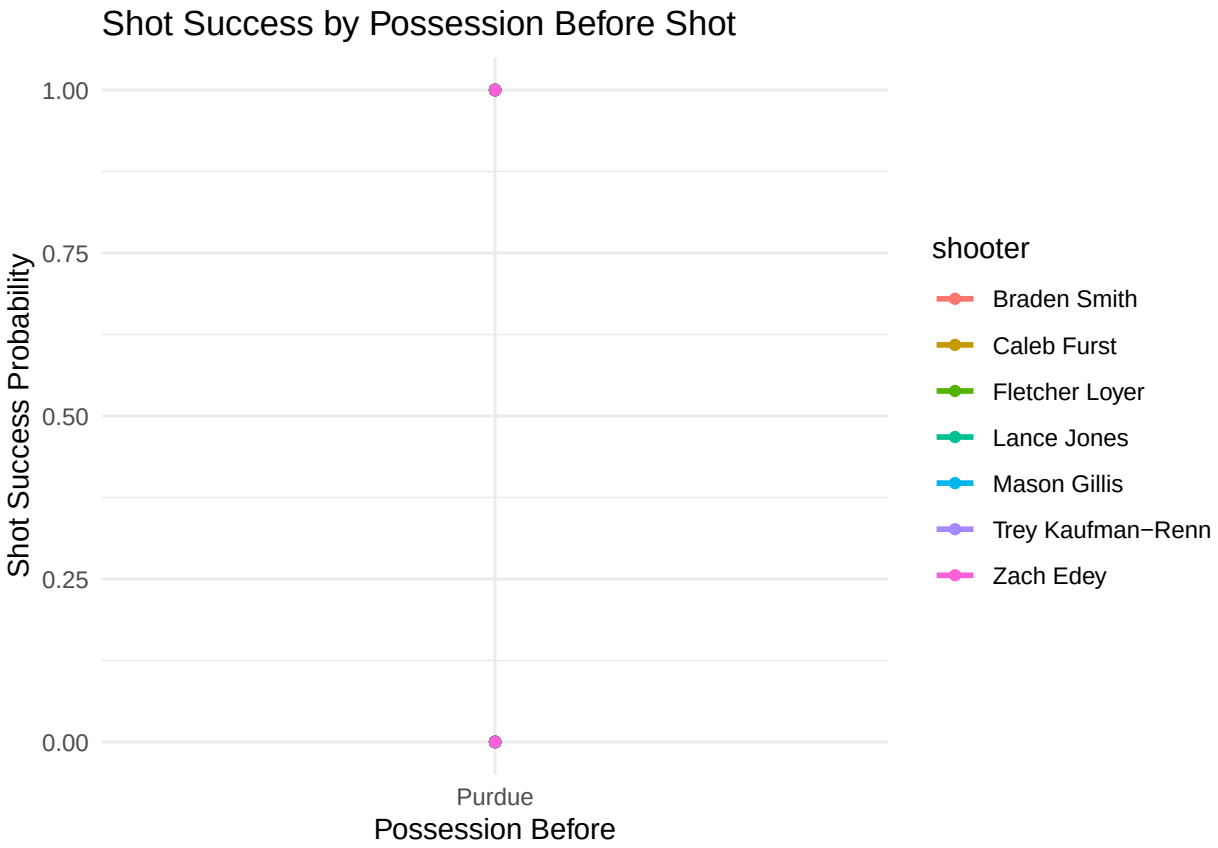
The Shot Success by Possession Before Shot plot examines how possession changes prior to the shot influence success. It may show whether a player is more likely to score following specific possession scenarios, such as a defensive rebound or steal, offering insights into the flow and rhythm that set up successful shots.

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

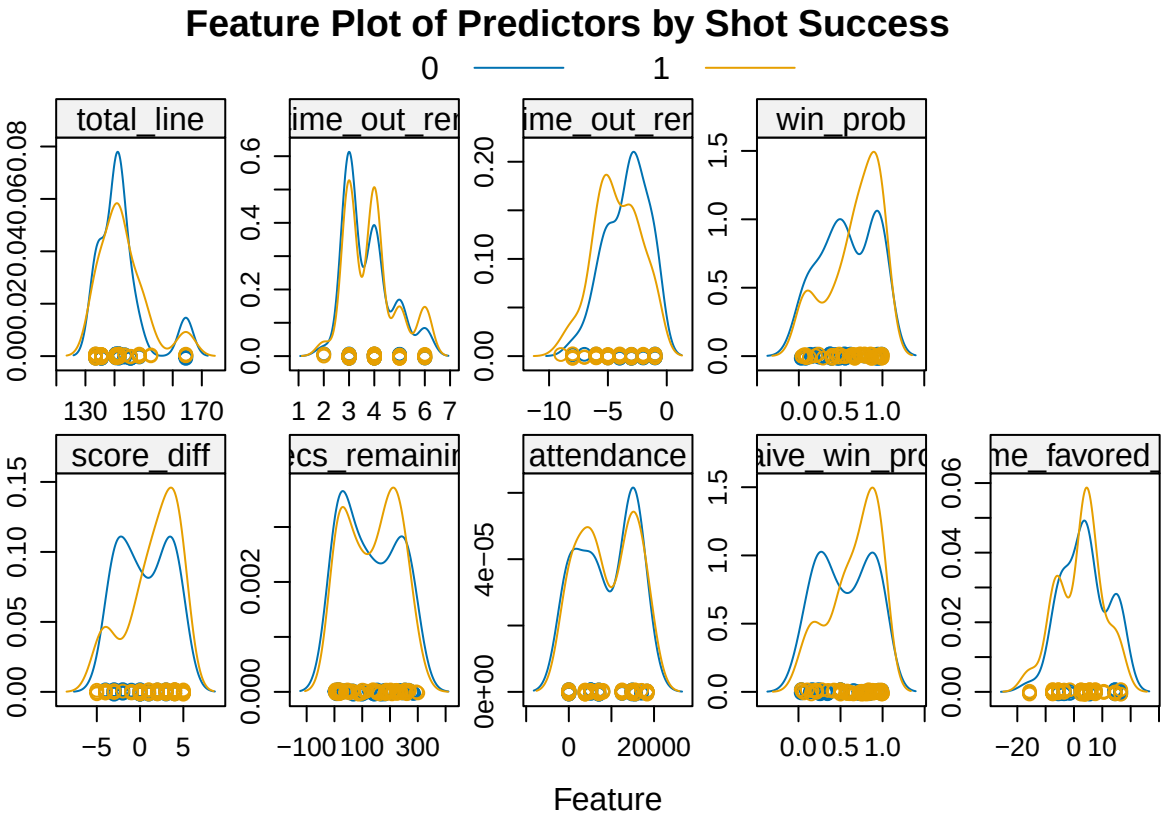



```
## `geom_smooth()` using formula = 'y ~ x'
```



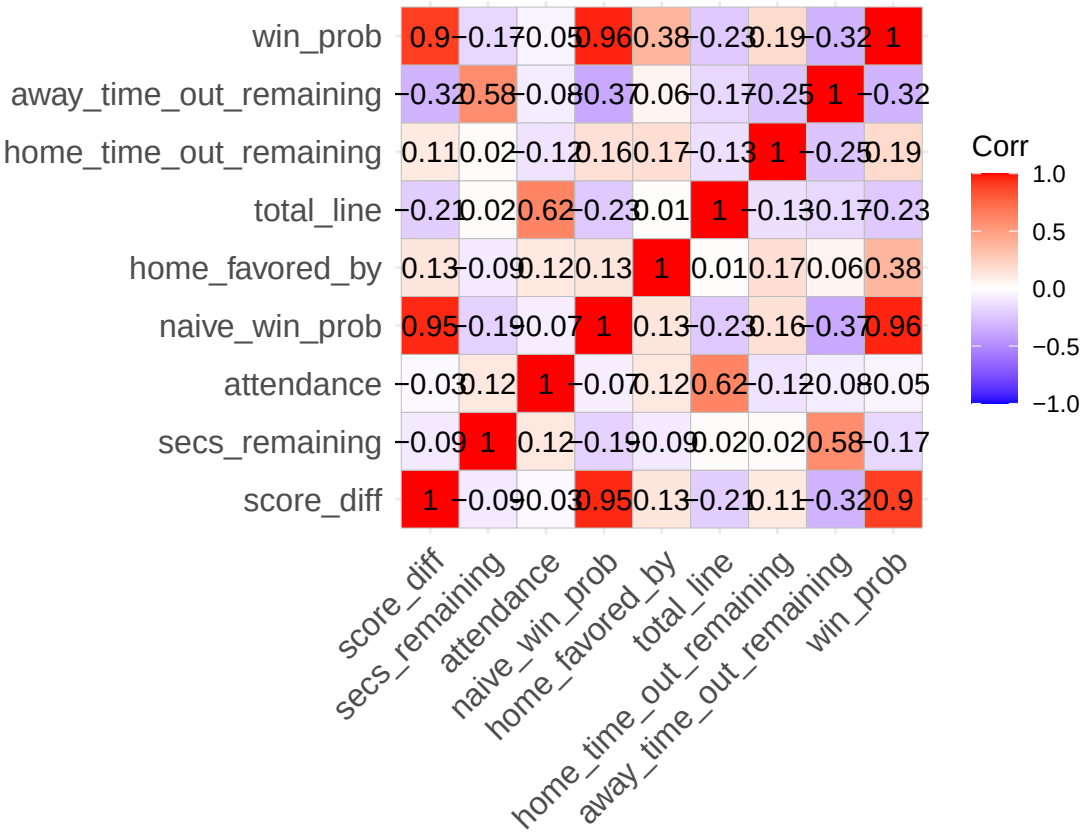
6. Feature Plot:

The density plots for each predictor variable allow for comparison between successful and failed shots across features like score difference, time remaining, and win probability. Peaks in the density plots for successful shots can reveal favorable conditions that correlate with higher shot success rates. This plot enables a nuanced understanding of the conditions under which players are most effective, helping to inform situational strategies based on these factors.



7. Correlation Plot:

The correlation plot shows the relationships among the numeric predictors used in the regression model for predicting clutch shot success.



Key observations from the correlation plot include:

High Correlation:

win_prob and naive_win_prob: These variables have a high positive correlation (around 0.95), indicating potential redundancy, as both are likely capturing similar aspects of game state. score_diff and win_prob: There is a strong positive correlation (approximately 0.9), suggesting that as the score difference increases, the probability of winning also rises, which is expected in close-game situations.

Given the high correlations among some predictors, specifically win_prob and naive_win_prob, and win_prob with score_diff, multicollinearity could be an issue. High multicollinearity may inflate standard errors in regression models, potentially making it challenging to determine the individual effect of each predictor.

To address this: Consider Removing Redundant Variables: Since win_prob and naive_win_prob are highly correlated, it may be beneficial to drop "win_prob" to simplify the model and reduce multicollinearity.

To sum up, these EDA plots provided a foundation for selecting variables for model training, helping to capture essential aspects of player performance in clutch situations.

Method & Result

Model Training and Model Comparison

Multiple machine learning models were trained to predict shot success in clutch moments, specifically in the last five minutes of a close game. These models included Logistic Regression, Ridge Regression, Support Vector Machine (SVM), Random Forest, Gradient Boosting Machine (GBM), and k-Nearest Neighbors (kNN). Each model was evaluated on performance metrics such as Accuracy, Kappa, Sensitivity (true positive rate), Specificity (true negative rate), AUC (Area Under the Curve), and F1 Score to assess its ability to distinguish between successful and unsuccessful shots under high-pressure scenarios.

```
## Setting levels: control = Fail, case = Success
```

```
## Setting direction: controls < cases

## Setting levels: control = Fail, case = Success

## Setting direction: controls < cases

## Setting levels: control = Fail, case = Success

## Setting direction: controls > cases

## Setting levels: control = Fail, case = Success

## Setting direction: controls < cases

## Setting levels: control = Fail, case = Success

## Setting direction: controls < cases

## Setting levels: control = Fail, case = Success

## Setting direction: controls > cases

##                 Model  Accuracy       Kappa Sensitivity Specificity     AUC
## 1 Logistic Regression 0.5000000  0.02409639       0.625         0.4 0.53750
## 2    Ridge Regression 0.6111111  0.22222222       0.625         0.6 0.53750
## 3                 SVM 0.4444444 -0.12500000       0.375         0.5 0.70000
## 4       Random Forest 0.6666667  0.34146341       0.750         0.6 0.53750
## 5                 GBM 0.6111111  0.22222222       0.625         0.6 0.52500
## 6                 KNN 0.5000000 -0.05194805       0.250         0.7 0.68125
##    F1_Score
## 1 0.5263158
## 2 0.5882353
## 3 0.3750000
## 4 0.6666667
## 5 0.5882353
## 6 0.3076923
```

Accuracy: Random Forest has the highest accuracy (0.6667), followed by Ridge Regression and GBM (0.6111 each). SVM has the lowest accuracy (0.4444).

Kappa: Cohen's Kappa indicates the consistency of the model predictions with the actual outcomes. Random Forest has the highest Kappa (0.3415), while SVM has a negative Kappa (-0.125), suggesting it performs worse than random guessing.

Sensitivity (Recall): Random Forest has the highest sensitivity (0.750), indicating it performs well in identifying positive cases, while KNN has the lowest sensitivity (0.250).

Specificity: KNN shows the highest specificity (0.7), meaning it better identifies negative cases, though its sensitivity is low. Logistic Regression and Random Forest have lower specificity values (0.4 and 0.6, respectively).

AUC (Area Under the Curve): SVM has the highest AUC (0.700), suggesting it has a relatively good performance in distinguishing between classes. KNN follows with an AUC of 0.6813.

F1 Score: Random Forest has the highest F1 Score (0.6667), indicating a good balance between precision and recall. SVM and KNN have lower F1 Scores (0.375 and 0.3077, respectively).

Summary:

- Best Overall Model: Based on the metrics, Random Forest is the top-performing model with high accuracy, Kappa, sensitivity, and F1 Score.
- Highest AUC: SVM has the highest AUC, making it suitable for tasks that prioritize distinguishing between classes rather than pure accuracy.

- Trade-offs: KNN has high specificity and AUC but lower sensitivity, indicating it is effective at avoiding false positives but may miss actual positives.

For predicting NCAA men's basketball players' performance in clutch situations, Random Forest might be the most reliable choice based on overall performance, though SVM's high AUC could be useful for nuanced classification tasks.

Selecting the Best Model

Metrics were normalized and weighted to calculate a Composite Score, providing an overall performance measure. Weights were assigned to Accuracy (25%), AUC (25%), F1 Score (20%), Sensitivity (15%), Specificity (10%), and Cohen's Kappa (10%) to balance precision and predictive power.

For each model, a composite score is computed by multiplying each normalized metric by its weight and summing the results. This formula integrates both normalized and original metric values to balance model strengths and weaknesses across several dimensions. Models are then sorted by their composite scores, from highest to lowest, to determine the overall ranking. The model with the highest composite score is chosen as the "Best Model."

```
## [1] "Ranked Models by New Composite Score:"
```

```
##                 Model  Accuracy       Kappa Sensitivity Specificity     AUC
## 4       Random Forest 0.6666667  0.34146341       0.750         0.6 0.53750
## 2    Ridge Regression 0.6111111  0.22222222       0.625         0.6 0.53750
## 5                 GBM 0.6111111  0.22222222       0.625         0.6 0.52500
## 1 Logistic Regression 0.5000000  0.02409639       0.625         0.4 0.53750
## 3                 SVM 0.4444444 -0.12500000       0.375         0.5 0.70000
## 6                 KNN 0.5000000 -0.05194805       0.250         0.7 0.68125
##    F1_Score Accuracy_norm  AUC_norm F1_Score_norm Composite_Score
## 4 0.6666667     1.0000000 0.7678571     1.0000000       0.8315375
## 2 0.5882353     0.9166667 0.7678571     0.8823529       0.7624627
## 5 0.5882353     0.9166667 0.7500000     0.8823529       0.7579984
## 1 0.5263158     0.7500000 0.7678571     0.7894737       0.6723138
## 3 0.3750000     0.6666667 1.0000000     0.5625000       0.6291667
## 6 0.3076923     0.7500000 0.9732143     0.4615385       0.6280139
```

```
## [1] "Best Model:"
```

```
##           Model  Accuracy     Kappa Sensitivity Specificity    AUC  F1_Score
## 4 Random Forest 0.6666667 0.3414634        0.75         0.6 0.5375 0.6666667
##   Accuracy_norm  AUC_norm F1_Score_norm Composite_Score
## 4             1 0.7678571             1       0.8315375
```

The Random Forest model ranks highest with a composite score of approximately 0.8315, indicating it performs well across the various metrics when the weights are applied. This suggests Random Forest is the most balanced model, achieving strong performance in terms of accuracy, AUC, F1 Score, sensitivity, and other factors.

Ridge Regression and GBM follow with scores of around 0.76, indicating they are also strong performers but slightly behind Random Forest based on this scoring method. They still balance well across the metrics, especially in accuracy and sensitivity.

According to the composite score, the Random Forest model is the best choice overall, as it demonstrates strong, balanced performance across all key metrics, making it suitable for situations where a robust and versatile model is needed. However, Ridge Regression and GBM also show promise as alternative choices, especially if a particular aspect of their performance aligns with specific needs in the application.

Selecting the Optimal Player

The Optimal Player section analyzes each player's performance under clutch-time conditions to identify the best candidate for taking the final shot in high-stakes scenarios. This analysis utilizes the Random Forest (RF) model, chosen for its high accuracy and capability to handle complex, non-linear patterns in the data.

The RF model evaluated each player's likelihood of scoring based on situational factors, such as score difference, time remaining, win probability, and other game-specific variables. By assessing these conditions, the model calculated the probability of a successful shot for each player, identifying the most reliable performers under clutch scenarios.

The table below illustrates the predicted success probabilities for each player given different score differences:

```
## Setting levels: control = Fail, case = Success
```

```
## Setting direction: controls < cases
```

```
## Adding missing grouping variables: `score_diff`
```

```
## # A tibble: 9 x 3
##   score_diff shooter        predicted_prob
##        <int> <fct>                   <dbl>
## 1         -2 Zach Edey               0.366
## 2          0 Zach Edey               0.748
## 3         -3 Mason Gillis            0.424
## 4          3 Fletcher Loyer          0.644
## 5          4 Zach Edey               0.83
## 6          5 Zach Edey               0.456
## 7         -1 Mason Gillis            0.258
## 8          1 Zach Edey               0.69
## 9          2 Zach Edey               0.846
```

Based on the RF model's predictions, Zach Edey consistently achieves high probabilities of successful shots across various score differentials, indicating his reliability in clutch situations:

High Probabilities Under Key Conditions: Edey's success probabilities are strong both when his team is tied or leading, with his highest probability (0.846) occurring when the team leads by 2 points. This suggests he performs well in scenarios where maintaining or extending a lead is critical.

Performance When Tied or Trailing: When the score is tied (score difference of 0), Edey still shows competitive probabilities (0.748), demonstrating his ability to remain effective under pressure.

Comparison with Other Players: Mason Gillis, with lower probabilities (e.g., 0.258 when trailing by 1), and Fletcher Loyer, who appears only once, lack the consistency that Edey shows. This makes Edey the more dependable choice for high-stakes shots.

In summary, the selection of Zach Edey as the preferred shooter in clutch-time scenarios is grounded in data-driven analysis. The RF model's assessment shows that Edey's shot success probabilities align well with conditions typical of clutch situations (e.g., close score differences, limited time remaining). His consistency and high success rates make him the optimal choice, providing statistical backing for decisions that could otherwise be based solely on intuition.

Conclusion

The Random Forest model, with its high composite score and effective handling of non-linear patterns, is well-suited for predicting shot success in Purdue Men's Basketball's clutch situations. Zach Edey's selection as the optimal player for critical shots is grounded in data-driven evidence, supported by his consistent shot success across high-stakes scenarios. This analysis demonstrates the value of machine learning in sports decision-making, showing how data-backed insights can enhance strategic choices in critical game moments.