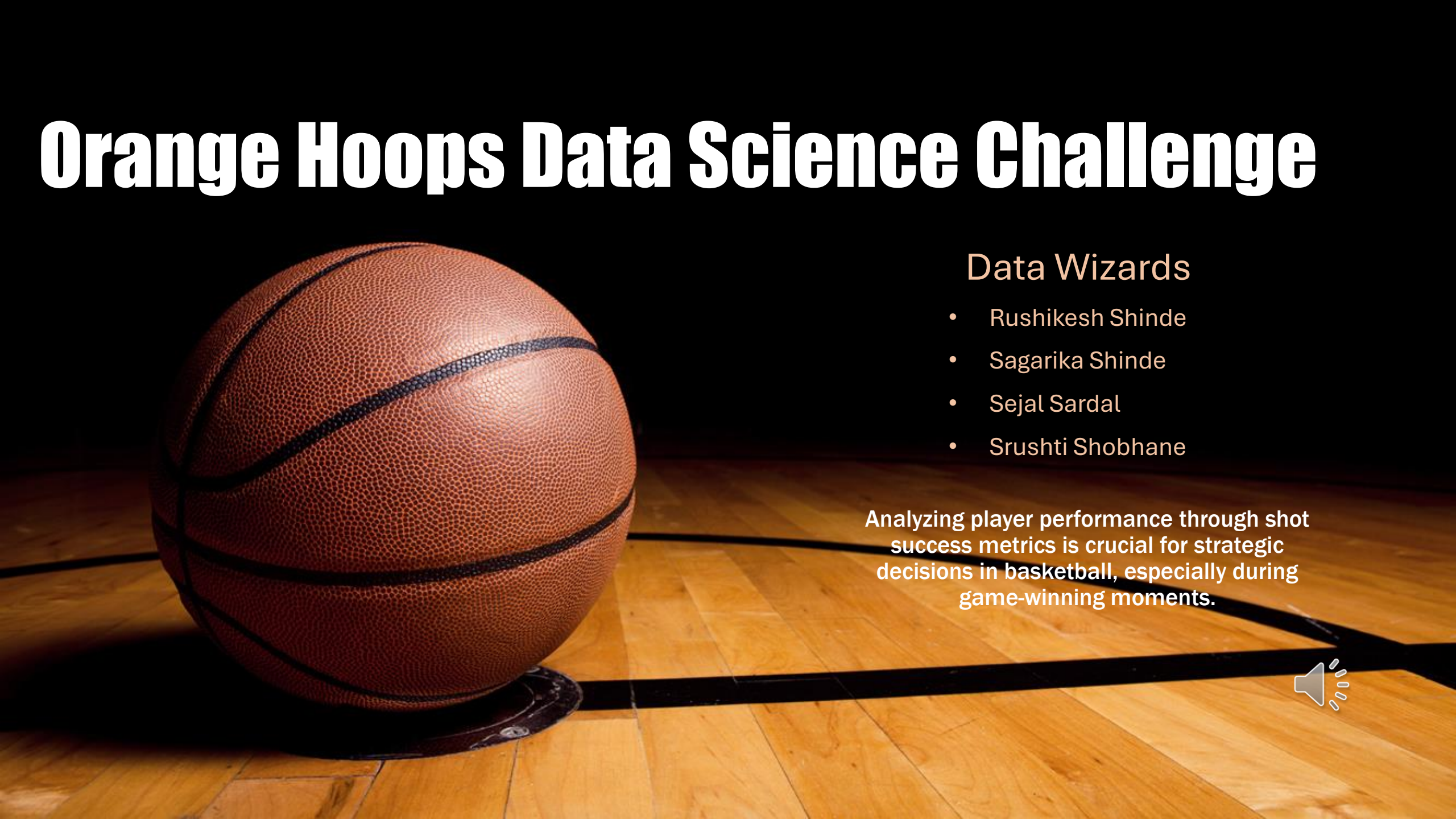
Orange Hoops Data Science Challenge



Data Wizards

- Rushikesh Shinde
- Sagarika Shinde
- Sejal Sardal
- Srushti Shobhane

Analyzing player performance through shot success metrics is crucial for strategic decisions in basketball, especially during game-winning moments.

Goal: Predicting Player who takes winning shot

The aim of this project is to predict shot success in basketball games, facilitating recommendations for the optimal player to take game-winning shots. By analyzing data from the Boston College basketball team, the project leverages machine learning to enhance decision-making in high-pressure situations.

Data Loading and Inspection

```
#printing the first few rows of the dataset

print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5991 entries, 0 to 5990
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   game_id                   5991 non-null   int64
 1   date                      5991 non-null   object
 2   home                      5991 non-null   object
 3   away                      5991 non-null   object
 4   play_id                   5991 non-null   int64
 5   half                      5991 non-null   int64
 6   time_remaining_half       5991 non-null   object
 7   secs_remaining            5991 non-null   int64
 8   secs_remaining_absolute   5991 non-null   int64
 9   description               5990 non-null   object
 10  action_team               5796 non-null   object
 11  home_score                5991 non-null   int64
 12  away_score                5991 non-null   int64
 13  score_diff                5991 non-null   int64
 14  play_length               5991 non-null   int64
 15  scoring_play              5991 non-null   bool
 16  foul                      5991 non-null   bool
 17  win_prob                  5991 non-null   float64
 18  naive_win_prob            5991 non-null   float64
 19  home_time_out_remaining   5991 non-null   int64
 20  away_time_out_remaining   5991 non-null   int64
 21  home_favored_by           5991 non-null   float64
 22  total_line                5991 non-null   float64
 23  referees                  5991 non-null   object
 24  arena_location            5991 non-null   object
 25  arena                     5991 non-null   object
 26  attendance                5991 non-null   int64
 27  shot_team                 2920 non-null   object
 28  shot_outcome              2920 non-null   object
 29  shooter                   2920 non-null   object
 30  three_pt                  2920 non-null   object
 31  free_throw                2920 non-null   object
 32  possession_before         5989 non-null   object
 33  possession_after          5933 non-null   object
 34  shooter_encoded           5991 non-null   int32
dtypes: bool(2), float64(4), int32(1), int64(12), object(16)
memory usage: 1.5+ MB
None
```

Data utilized in this analysis encompasses game statistics from the Boston College basketball team, focusing on shot outcomes and player performance metrics. The dataset, thoroughly cleaned and structured, lays the foundation for extracting valuable insights essential for determining the most effective shooter's performance.

Data Cleaning and Preprocessing

Initial cleaning involved dropping rows with missing essential data such as shooter, shot outcome, and shot team.

- Dropped rows missing essential data in shooter, shot_team, and shot_outcome to maintain data reliability.
- Converted secs_remaining to numeric, handling any non-numeric values as NaN.
- Transformed three_pt into a Boolean type (True for three-point attempts), aiding in logical filtering.

```
27]: data_cleaned.describe()
```

27]:

	game_id	play_id	half	secs_remaining	secs_remaining_absolute	home_score	away_score	score_diff	play_length	win_prob	...	home_t
count	2.920000e+03	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2.920000e+03	...	
mean	4.015979e+08	158.507534	1.529452	1145.329795	1160.843493	39.030137	36.786644	2.243493	10.393151	5.931869e-01	...	
std	1.136446e+04	92.962103	0.517414	697.755897	701.710556	23.474501	21.739418	10.020498	9.120875	3.237856e-01	...	
min	4.015762e+08	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	-32.000000	0.000000	2.450000e-12	...	
25%	4.015921e+08	77.750000	1.000000	529.000000	545.000000	18.000000	19.000000	-5.000000	1.000000	3.138073e-01	...	
50%	4.016041e+08	157.000000	2.000000	1140.500000	1156.500000	38.000000	36.000000	1.000000	9.000000	5.963663e-01	...	
75%	4.016042e+08	237.250000	2.000000	1747.000000	1766.000000	58.000000	54.000000	8.000000	17.000000	9.534236e-01	...	
max	4.016254e+08	361.000000	3.000000	2392.000000	2681.000000	95.000000	90.000000	33.000000	40.000000	1.000000e+00	...	

8 rows × 21 columns

```
[27]: data_cleaned.describe()
```

[27]:

remaining	away_time_out_remaining	home_favored_by	total_line	attendance	shooter_encoded	shooter_made	shooter_rolling_accuracy	score_diff_lag	shot_success
)20.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000	2920.000000
5.220548	1.766438	4.446233	144.818493	4986.690411	86.686644	0.514041	0.510103	2.238014	0.514041
0.862997	3.114615	8.341802	7.596582	2343.762362	47.292376	0.499888	0.267182	10.017347	0.499888
2.000000	-8.000000	-7.500000	125.500000	0.000000	0.000000	0.000000	0.000000	-32.000000	0.000000
5.000000	0.000000	-1.500000	141.500000	3886.000000	44.000000	0.000000	0.400000	-5.000000	0.000000
5.000000	2.000000	3.500000	148.500000	4866.000000	74.000000	1.000000	0.600000	1.000000	1.000000
6.000000	4.000000	9.500000	149.500000	6611.000000	137.000000	1.000000	0.666667	8.000000	1.000000
6.000000	6.000000	22.500000	154.500000	8606.000000	168.000000	1.000000	1.000000	33.000000	1.000000

Feature Engineering

To enhance predictive modeling, key features were engineered from raw data, focusing on player performance, game context, and shot characteristics.

Creation of New Features:

- score_diff_category: Categorizes game situations as 'Losing,' 'Close,' or 'Winning' for contextual understanding.
- clutch_time: Identifies high-pressure shots in the last 2 minutes of the game.
- shooter_encoded: Assigns numerical IDs to players for individual analysis.
- shooter_rolling_accuracy: Reflects recent shooting performance over the last 5 attempts.
- score_diff_lag: Tracks recent score trends for momentum analysis.One-Hot Encoded score_diff_category: Converts game situations into machine-readable format.
- three_pt: Distinguishes three-point attempts from other shot types.
- shot_success: Binary target variable defining shot outcome.

```
# Final prepared dataset with only necessary features
data_prepared = data_cleaned[final_features + ['shot_success']]
print(data_prepared.head())
```

	secs_remaining	score_diff	three_pt	shooter_encoded	clutch_time	\
0	2382	2	False	146	False	
1	2364	-1	True	84	False	
4	2308	-1	False	148	False	
6	2304	1	False	148	False	
9	2285	1	False	74	False	

	shooter_rolling_accuracy	score_diff_lag	score_diff_category_Close	\
0	1.0	0.0	True	
1	1.0	2.0	True	
4	0.0	-1.0	True	
6	0.5	-1.0	True	
9	0.0	1.0	True	

	score_diff_category_Winning	shot_success
0	False	1
1	False	1
4	False	0
6	False	1
9	False	0

Model Training

Model Selection Based on Recall and F1 Score

Chose XGBoost Classifier for its effectiveness with imbalanced data and ability to capture complex patterns, prioritizing high Recall and F1 Score.

- **Key Metrics Used**:
  - **Recall**: Recall: Identifies successful shots, crucial for prioritizing high-confidence shots.
  - F1 Score: Balances precision and recall, ideal for evaluating prediction accuracy.
  - ROC AUC: Measures model's ability to distinguish between successful and unsuccessful shots.
- **Approach Taken**:
  - Compared multiple classifiers (Random Forest, SVM) before selecting Gradient Boosting.
  - Data Split: 70% training, 30% testing.
  - Feature Scaling: Used StandardScaler for consistency.
  - Cross-Validation: Applied 5-fold CV to ensure stability.
  - Model Evaluation: Achieved 89% accuracy, 0.90 F1, and 0.96 ROC AUC.Feature Importance: Analyzed top features impacting shot success prediction.

Model Evaluation

Classification Report: Provides detailed metrics, such as precision, recall, and F1-score, to assess the model's performance in predicting shot success.

- **Measures the accuracy of positive predictions, indicating the proportion of correctly identified successful shots out of all predicted successful attempts. This metric helps evaluate the model's confidence in identifying high-quality shots**.

- **Recall** focuses on the model's ability to capture all actual successful shots, ensuring that high-probability opportunities aren't missed during critical moments.

- **F1-Score** Balances precision and recall, providing an overall measure of model performance in predicting shot outcomes accurately.

Accuracy Score: Measures the overall percentage of correct predictions made by the model..

- Represents the overall percentage of correct predictions for shot success across all attempts. While accuracy offers a general view of performance, it is supplemented by precision and recall for a deeper understanding of model reliability under various game situations.

Model Results and Insights

- The model successfully identifies high-probability shots, especially under specific game conditions, like close scores or final moments, highlighting its effectiveness in strategic decision-making.
- High accuracy reflects the model's effectiveness in correctly predicting injury occurrences based on the training data.

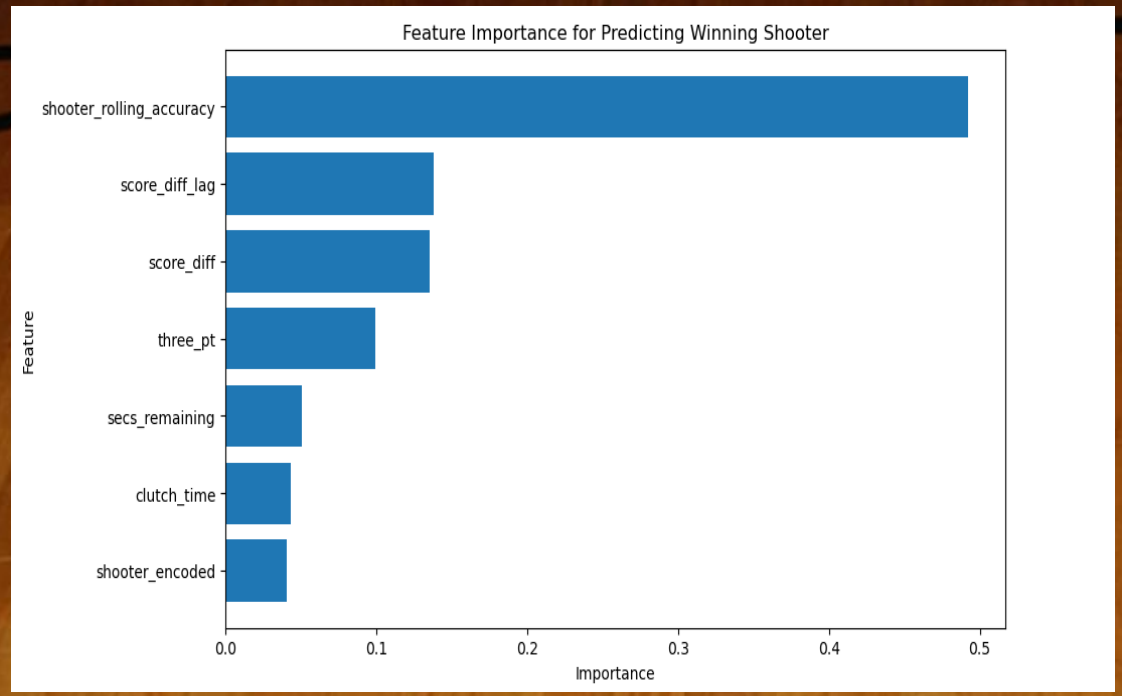
Focus Areas for Improvement

Feature Expansion: Incorporate additional game context (e.g., defensive pressure) to enhance prediction accuracy.

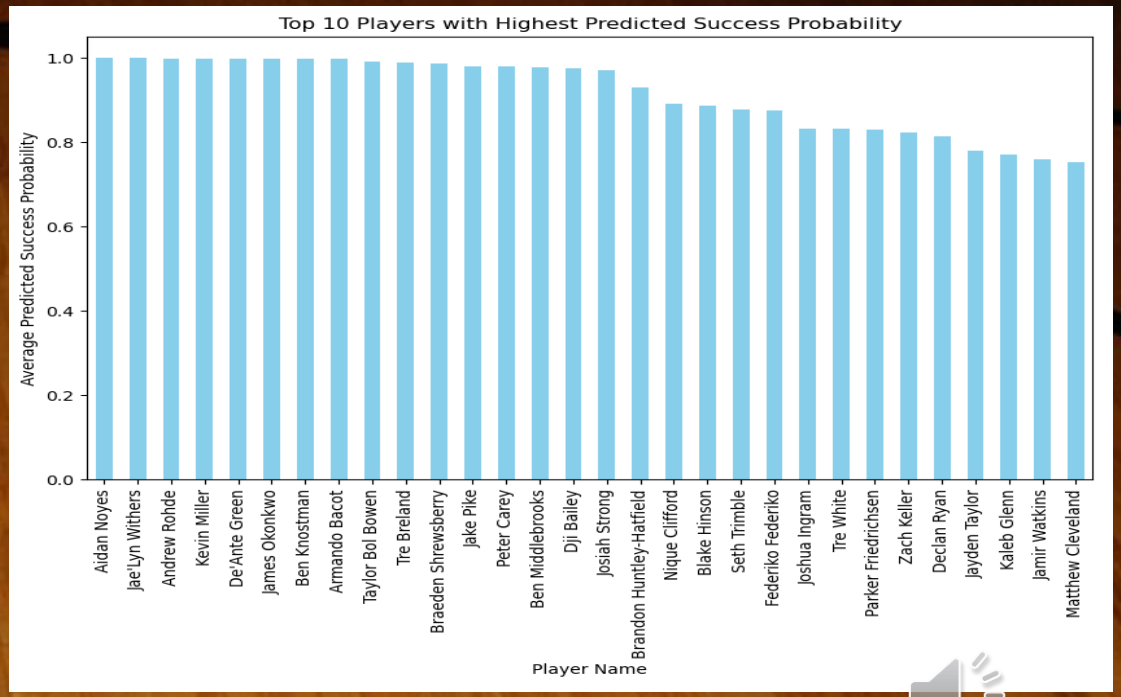
Model Tuning: Fine-tune hyperparameters further, especially for XGBoost, to maximize Recall without compromising F1 Score.

Visualization

Feature Importance for Predicting Winning Shooter



Top 10 Players with Highest Predicted Success Probability




Thankyou