# The American Express Campus Challenge 2024

## Data Titans

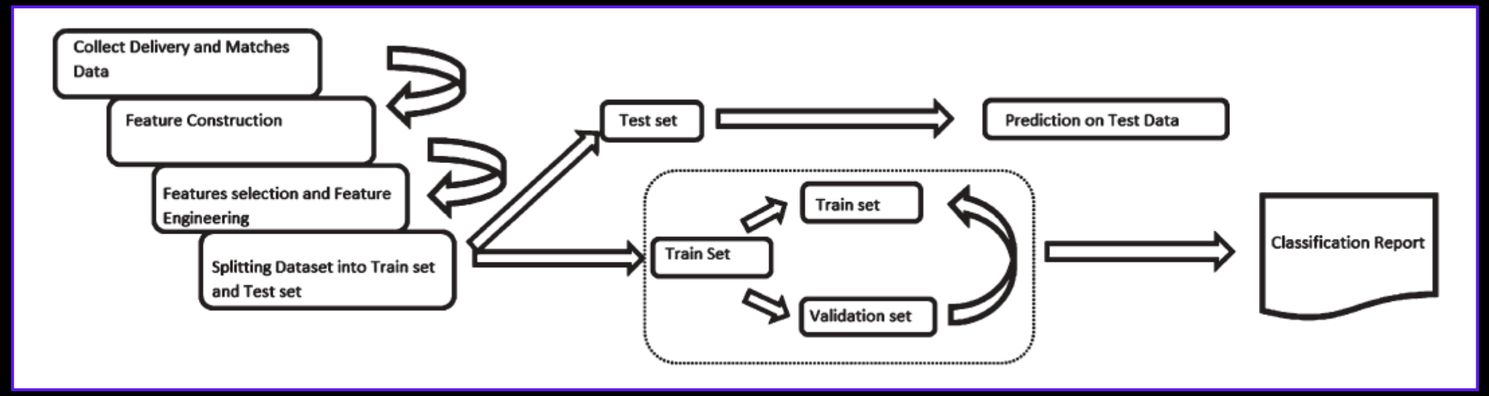
**Pritish Saha, Adrij Das, Rounak Nath**

**(Indian Institute of Technology Kharagpur)**

# SUMMARY OF THE FINAL SOLUTION

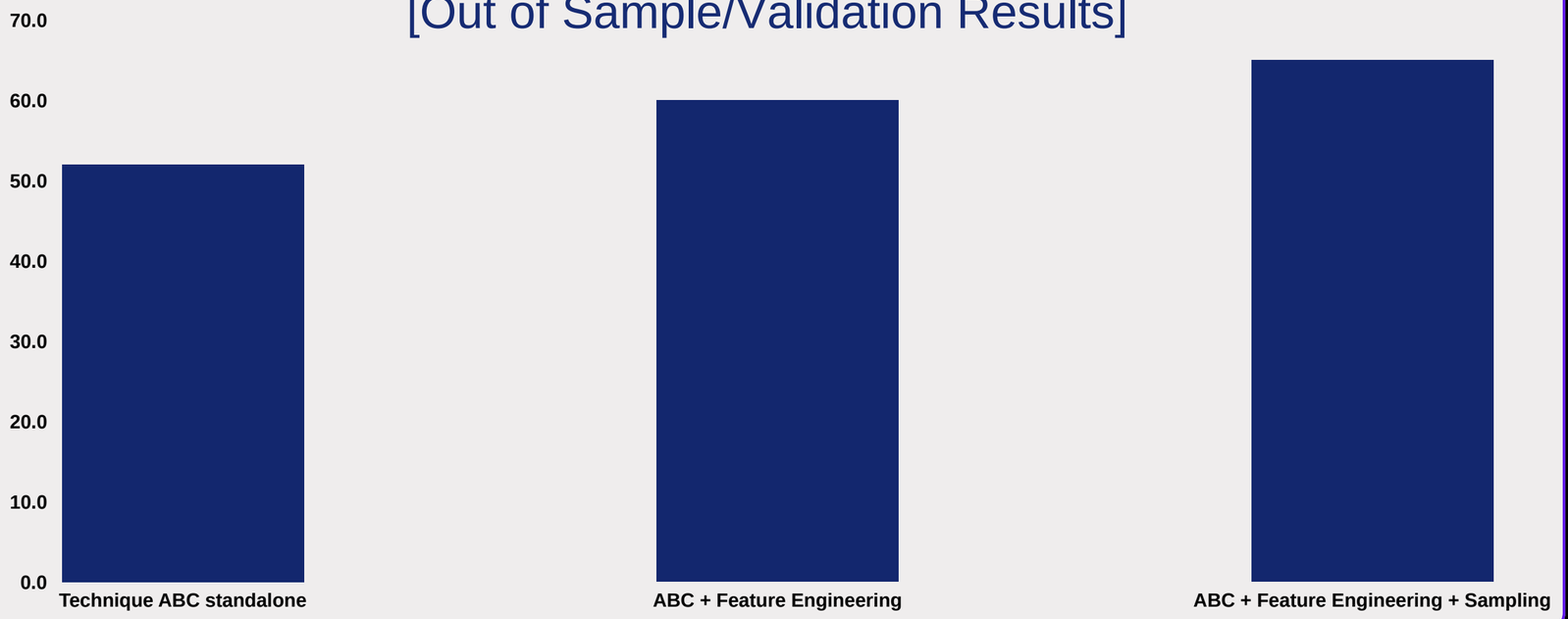
| Objective Function/Dependent Variable | Sampling | Feature Engineering | Modeling Technique | Any other Dimension? |
|---|---|---|---|---|
| **Objective:**<br>To predict the winning team in T20 matches.<br><br>**Dependent Variable:**<br>Created the **'winner_01'** variable to indicate the winning team (0 for team1 and 1 for team2). | **Stratified Sampling** was used to ensure balanced representation of matches won by different teams. | Performed the following major feature engineering steps:<br><br>• Created features like team_avg_Econ_last10, team_srrate_ratio_last10, team_runs_ratio_last10, percentage_dot_balls_bowled_last_5, etc.<br>• Used **mutual information** and **VIF** for feature selection to identify key features. | We reviewed the following modeling techniques:<br><br>1. XGBoost<br>2. CatBoost<br>3. Gradient Boosting Machine (GBM)<br>4. LightGBM<br><br>**XGBoost** was selected as the final solution based on achieving the highest accuracy metric of **65%** after hyperparameter optimization using **Bayesian Optimization** and **Grid Search CV**. | • Detailed preprocessing and feature engineering steps.<br>• Robust model evaluation using cross-validation and hyperparameter tuning.<br>• Applied rigorous parameter tuning using Bayesian optimization to achieve optimal model performance. |

# MODEL PERFORMANCE – ALL ITERATIONS




## Accuracy

[Out of Sample/Validation Results]




- **Iteration 3** led to a significant jump in accuracy from **52%** to **65%** due to optimized **feature engineering and hyperparameter tuning**.
- **XGBoost** outperformed other models consistently.

- XGBoost gave best R1 accuracy score and was used as final modeling technique in Round 2

## Detailed overview of the Modeling Technique

Chosen Model: **XGBoost**

Why XGBoost:

- High predictive accuracy and robustness to overfitting.
- Efficient handling of large datasets and missing values.

**Objective Function:** Binary cross-entropy loss for binary classification as it measures the performance of a classification model whose output is a probability value between 0 and 1.

Created the **dependent variable** as the probability of winning based on match outcomes.

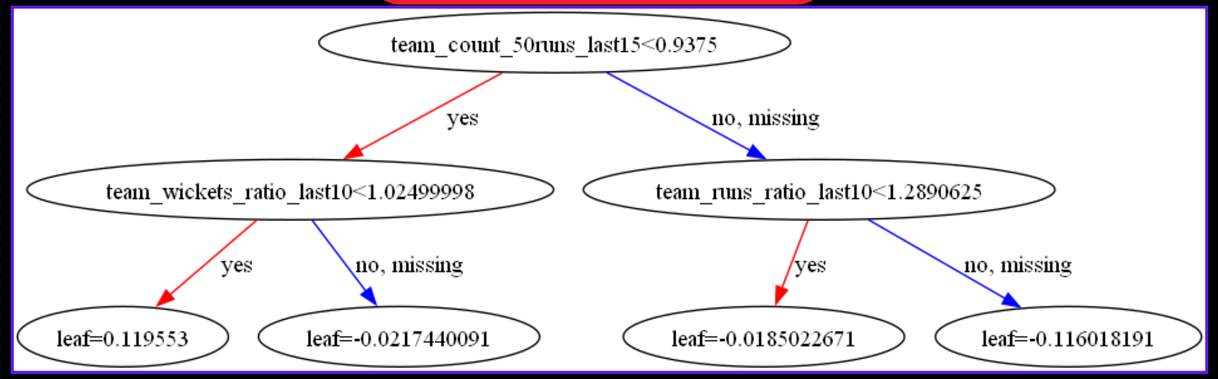
**Inner Workings:**

- **Model Architecture:** Gradient boosting framework with decision trees as base learners.
- **Model Equation:** Aggregation of weighted predictions from individual trees.

**Real-World Example:** XGBoost has been used in various Kaggle competitions, often outperforming other algorithms due to its efficiency and performance.

**Academic Literature:** Referenced in numerous research papers for its efficiency and performance.

XGBoost Inner Workings




$$\text{Similarity Score} = \frac{(\sum \text{Residual}_i)^2}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)] + \lambda}$$

$$\text{Gain} = \text{Left}_{\text{Similarity}} + \text{Right}_{\text{Similarity}} - \text{Root}_{\text{Similarity}}$$

$$\text{Output Value} = \frac{(\sum \text{Residual}_i)}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)] + \lambda}$$

$$\log\left(\frac{p}{1-p}\right) = \log(\text{odds})$$

$$\text{Probability} = \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}}$$

# MODEL TECHNIQUE/ALGORITHM DETAILS

**Techniques Tried:** We experimented with both CatBoost and XGBoost to identify the best-performing model for our dataset.

**Hyperparameter Tuning:**

- XGBoost: Utilized Bayesian Optimization and GridSearchCV to find the optimal hyperparameters, ensuring the best possible model performance.
- CatBoost: Employed RandomizedSearchCV to tune hyperparameters, allowing for a thorough exploration of the parameter space in a computationally efficient manner.

**Performance and Selection:** After extensive experimentation, XGBoost provided the best R1 accuracy score and was selected as the final modeling technique in Round 2 due to its consistent performance and robustness.

**Bayesian Optimization for XGBoost:**

- Focused on optimizing key hyperparameters such as learning rate, max depth, and subsample ratio.
- This iterative approach helped in efficiently navigating the hyperparameter space, leading to improved model accuracy and generalization.

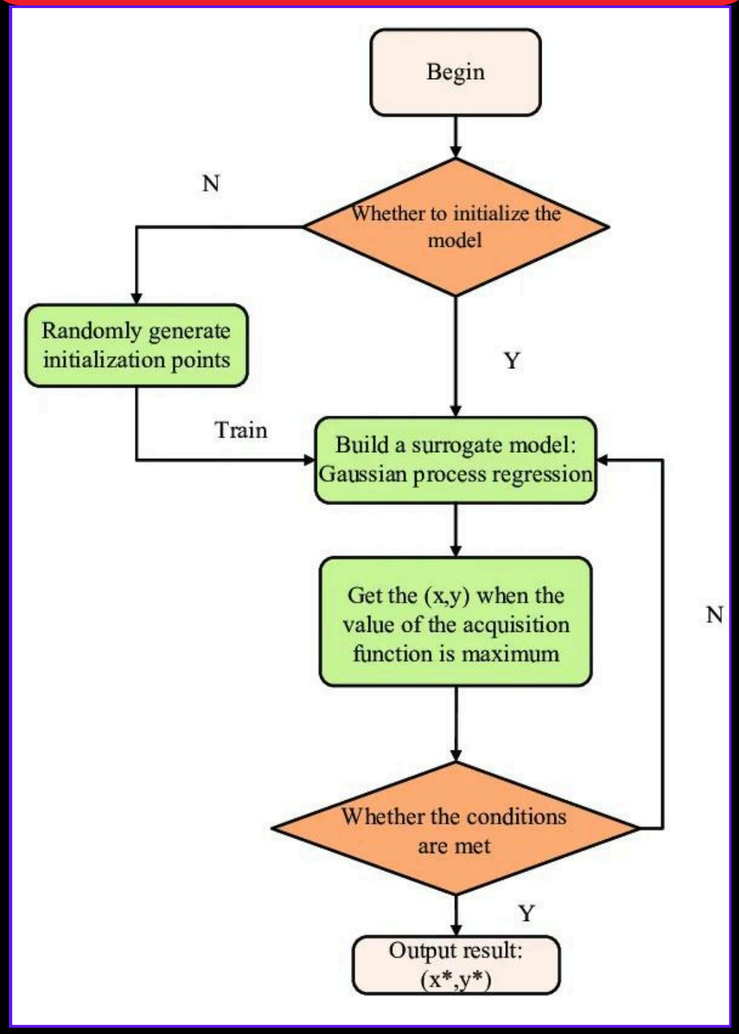
**GridSearchCV for XGBoost:**

- Conducted an exhaustive search over specified parameter values for the XGBoost model.
- Enabled the identification of the best combination of hyperparameters by evaluating the model's performance on the validation set.

**RandomizedSearchCV for CatBoost:**

- Implemented a randomized search strategy over hyperparameters to quickly identify promising configurations.
- Balanced exploration and exploitation by sampling a wide range of hyperparameter values.

Basic flowchart of Bayesian Optimization



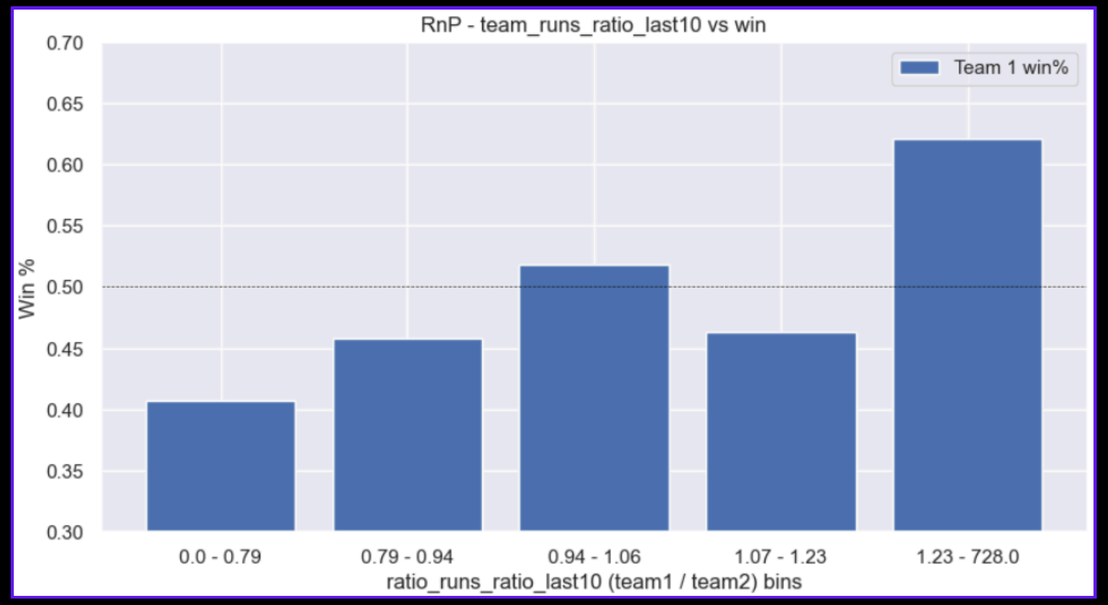
- The following features were created/engineered …

| S. No. | Feature Description |
|---|---|
| 1 | team_avg_Econ_last10 |
| 2 | .team_srrate_ratio_last10 |
| 3 | team_runs_ratio_last10 |
| 4 | percentage_dot_balls_bowled_last_5 |
| 5 | pitch condition |
| 6 | team_wickets_ratio_last10 |
| 7 | percentage_runs_through_boundaries_last_5_ratio |
| 8 | team_avg_econ_ratio_last10 |
| 9 | runs_through_extras_last_5_ratio |
| 10 | bowling_srrate_ratio_last_10 |
| 11 | weighted wins |
| 12 | day_night_match |
| 13 | batting_bowling_win_ratio |
| 14 | ratio_avg_runs_last15 |

We used a combination of **Mutual Information (MI)** and **Variance Inflation Factor (VIF)** for selecting relevant features. This approach helped us identify the most informative and non-collinear features for our model.

- **Mutual Information:** Assessed the mutual dependence between features and the target variable, helping to identify features with the most predictive power.
- **Variance Inflation Factor (VIF):** Used to detect and mitigate multicollinearity among features, ensuring that the features selected are not highly correlated with each other.

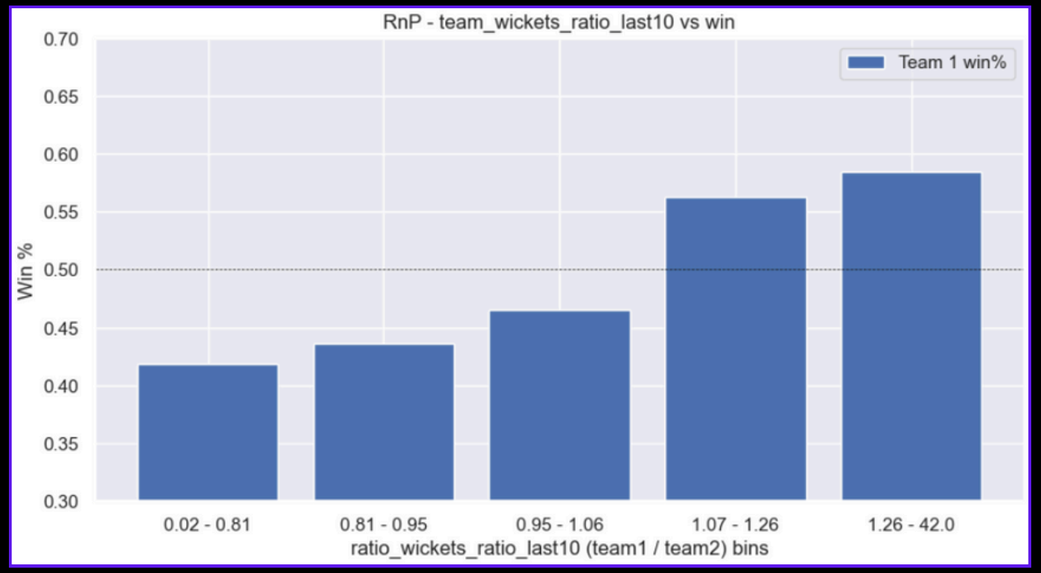
## Some good Feature Trends

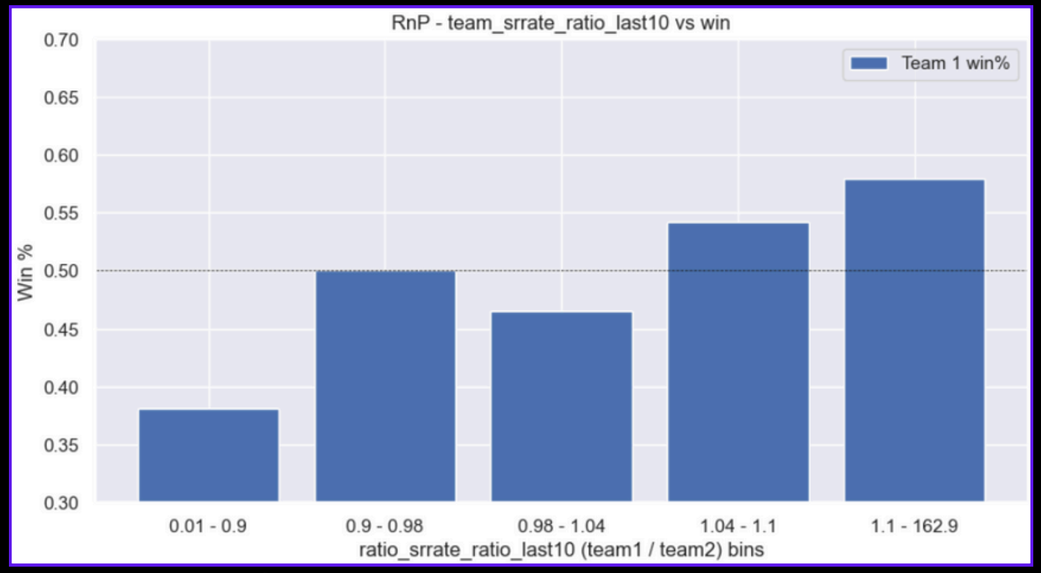


## Top 10 Features in the Final Solution

| Rank | Feature | Imp |
|---|---|---|
| 1 | team_count_50runs_last15 | 18.4% |
| 2 | team_runs_ratio_last10 | 14.7% |
| 3 | team_wickets_ratio_last10 | 12.5% |
| 4 | team_srrate_ratio_last10 | 12.0% |
| 5 | bowling_srrate_ratio_last_10 | 10.8% |
| 6 | ratio_avg_runs_last15 | 9.1% |
| 7 | team_winp_last5 | 9.0% |
| 8 | weighted wins | 6.7% |
| 9 | ground_avg_runs_last15 | 6.6% |
| 10 | team1_winp_team2_last15 | 0.0% |

## Some good Feature Trends



# Detailed overview of the Sampling Technique

**Stratified Sampling** was employed to maintain the distribution of match outcomes across training, feature selection, and testing sets. This ensured that each subset accurately represented the overall dataset, leading to reliable model evaluation and improved performance stability.

Why Stratified Sampling Was Used:

- **Maintaining Class Distribution:** Preserved original distribution of match outcomes (win/loss).
- **Reliable Model Evaluation:** Ensured evaluation metrics reflected true performance.
- **Improved Model Generalization:** Enhanced model's robustness to unseen data.

**Impact:**

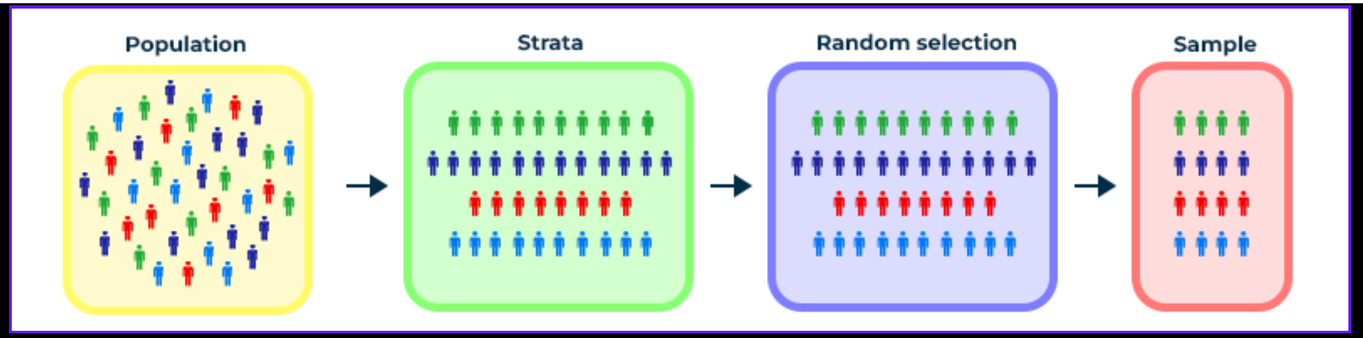
- **Row Count:** Total rows unchanged; balanced class representation in subsets.
- **Event Rate:** Consistent win/loss proportion across all subsets.

**Benefits:**

- **Improved Generalization:** Better performance on both training and validation data.
- **Performance Stability:** Reduced overfitting/underfitting risks.
- **Reliable Metrics:** Accurate reflection of model’s accuracy and performance.

**Conclusion:**

- Stratified sampling was crucial for maintaining dataset integrity, ensuring reliable evaluation, and enhancing model stability and accuracy.



## 1 ML Enhancements

1. Incorporate advanced ensemble techniques like Stacked Generalization and Voting Classifiers.
2. Experiment with neural network architectures, including various LSTM and GRU configurations.
3. Explore temporal models for capturing sequential patterns.
4. Advanced hyperparameter tuning using Bayesian Optimization and Tree-structured Parzen Estimator (TPE).

## 2 Feature Engineering

1. Introduce advanced features such as player-specific performance metrics, weather, and pitch conditions.
2. Perform deeper statistical analysis to uncover hidden patterns.
3. Utilize external datasets for enriched feature set.
4. Implement SMOTE and Tomek Links for balancing class distribution and improving model robustness.

## 3 Any other dimension?

1. Implement real-time data processing for live match prediction.
2. Intelligent feature engineering based on live data and ongoing match conditions.

Thank You!

# Data Titans