Dream11 Midprep

Team 86

The Challenge

THE CHALLENGE

Cluttered Data:

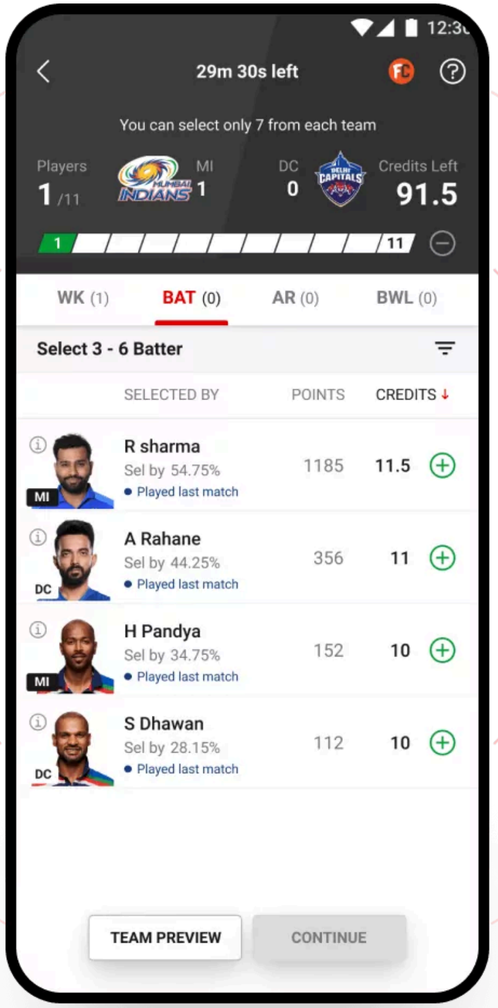
Overwhelming player data makes it difficult to extract actionable insights.

Personal Bias & Complexity:

Personal biases and complex data analysis often overshadow logical, data-driven decisions.

Entry Barrier:

These factors hinder users from making informed team selections, reducing engagement.



*Fantasy Sports Market size was valued at **USD 27.20 billion** in 2022 and is poised to grow from **USD 30.95 billion** in 2023 to USD 87.07 billion by 2031, growing at a **CAGR of 13.80%***

Our Solution

THE DEMO

Explainability

The predictions by our ML model has a significant understandable relation from the features used, but providing these features directly to the user is not a very good option for the user experience.

We turned to a better approach for the user experience by turning all features and the result to **natural language explanations.**

We used Gemini1.5-Flash to provide explainable responses while being faster at the same time.

We further aimed to enhance the user experience by adding verbal explanations using **Google Cloud’s Translation and TTS** services to provide audio cues in **15 Indian languages.**

Feature Engineering

Explicit features

Features that were directly used to calculate fantasy points and have a well-defined impact on player performance metrics

All the explicit features were derivable from the cricsheet dataset.

- *Runs Scored*
- *Boundaries (4s, 6s)*
- *Fifties and Hundreds.*
- *Ducks*
- *30-run Innings*
- *Caught (Catches)*
- *Run Out (Fielding)*
- *Direct Hits (Fielding)*
- *Stumping (Wicketkeepers)*
- *3+ Catches in a Match*
- *Wickets Taken (Bowling)*
- *3/5-Wicket Hauls*
- *Maiden Overs*
- *Wickets by LBW or Bowled*
- *Strike Rate*
- *Economy*

Implicit features

Features that indirectly affect the players' performance in a match and add contextual depth to the model

All the explicit features were derivable from the cricsheet dataset.

Cricsheet Derivable

- *Dot Balls*
- *Balls per Boundary*
- *Runs Conceded*
- *Bowling Economy*
- *Cumulative Matches Played*
- *Venue*
- *Toss Decisions*
- *Match Type*
- *Gender of Players*
- *Player-to-Player Matchups*
- *Batting Position*

We tried to scrape some data but due to inconsistent and insufficient data, the exact information of some implicit features were not considered for our final database.

Example

- *Pitch Types*
- *Weather Conditions*
- *Injury History*
- *Bowling Style*

FEATURE ENGINEERING

Category-1: Sparse Features

New features derived by taking their **Historical Cumulative Moving Average** for window size of 10, 30, 50. **HCMA with window size 30** were taken into account in the final dataset.

- boundaries
- fifties
- hundreds
- ducks
- thirty_run_innings
- caught
- maiden_overs
- run out
- direct
- stumped
- 3+catches
- wickets_taken
- 3wickets_haul
- 5wickets_haul
- wickets_lbw_bowled

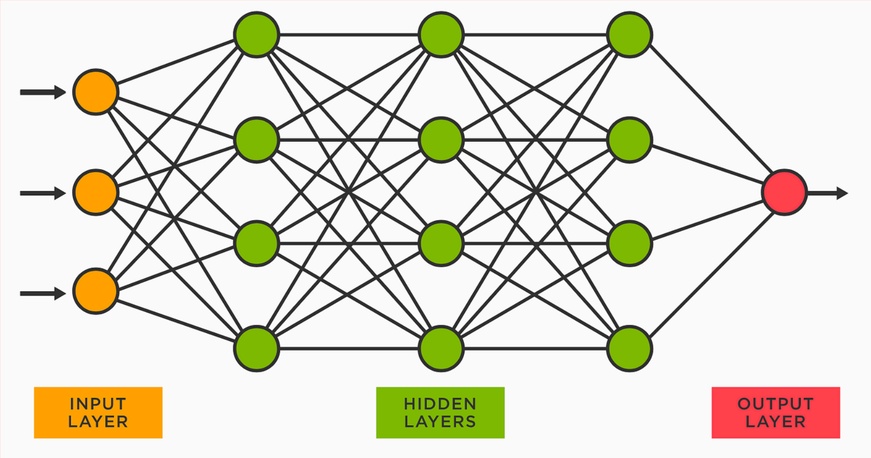
Category-2: Dense Features

Their **Exponential Moving Average** was taken to form new numerical features, for a **window size of 5, 7, 9 and alpha value of 0.5, 0.7, 0.9**

- dot_balls
- total_runs
- balls_faced
- strike_rate
- runs_conceded
- balls_bowled
- economy_rate
- dots
- bowling_average

Model Selection and Inference

Getting a baseline



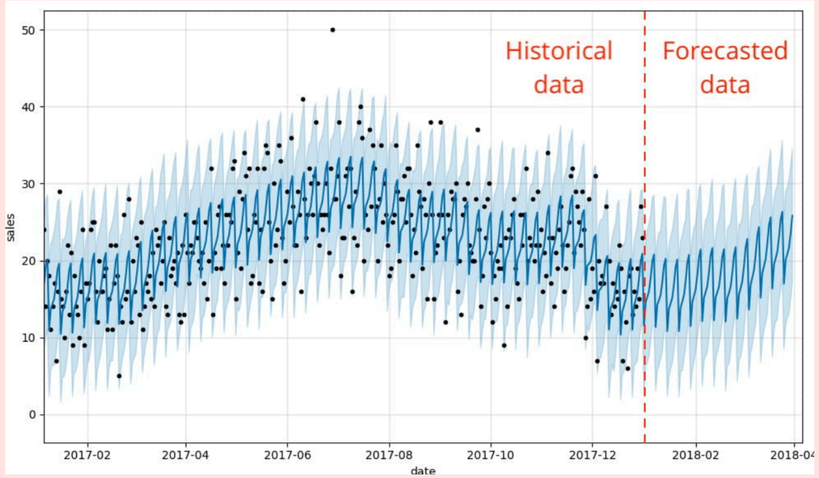
Initial benchmarking for feature contribution analysis using, XGBoost and Random Forest as baseline models

Neural Network architectures with the Historical Cumulative Moving Average (HCMA) and Exponential Moving Average (EMA) Features.

Architectures tried out were:

- **LargeNet** with 5 hidden layers with BatchNorm.
- **ImprovedNet** with dropout and residual connections.
- **OptimisedNet** with 2-layer residual connections and Xavier Initialization.

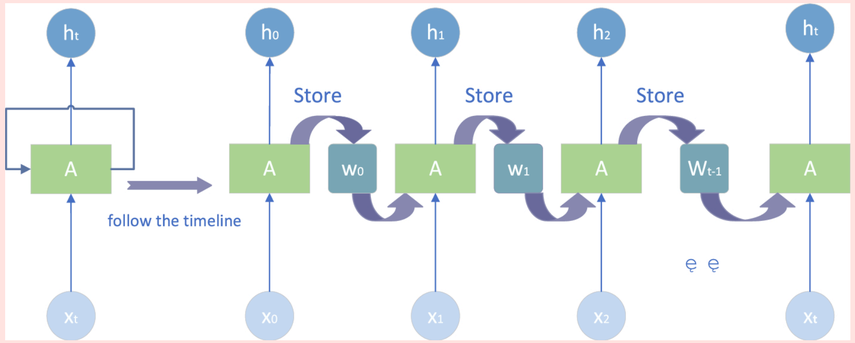
Time Series Model



Next, we tried to approach the problem using time series models. The reasoning behind it was as follows:

- A player's **performance** in consecutive games is often **correlated**. For example, form, fitness, and confidence can carry over from previous matches.
- Time series models can ***identify anomalies or outliers***, such as an unusually high-scoring game, and adjust predictions accordingly to avoid overestimating future performance.
- They can also ***identify long-term trends in performance***, such as gradual improvement or decline over a season.

RNN-Based Time Series Models

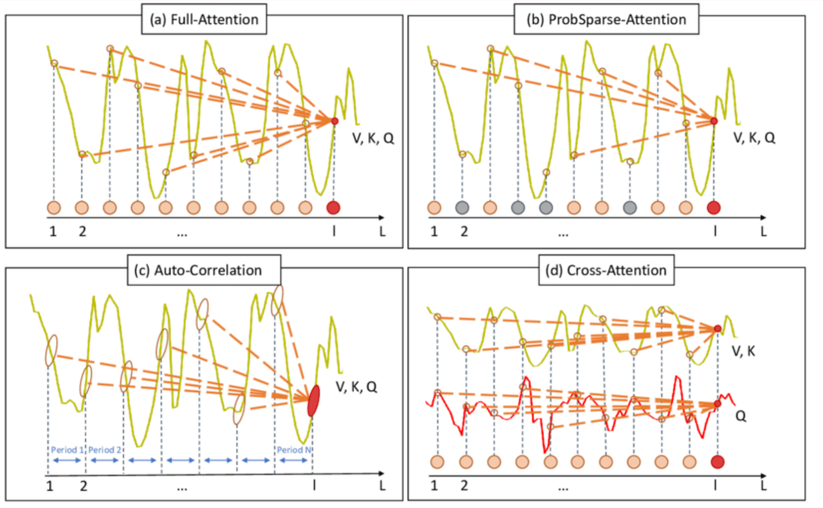


We experimented with several models such as LSTM, BiLSTM, LSTM-attention, GRU, BiGRU, GRU-attention.

These models are advantageous because:

- **Dynamic Dependencies:** They capture the evolving nature of player performance over time via short-term and long-term memories.
- **Sequence Modeling:** They are designed to work on ordered, time-dependent data.
- **Feature Integration:** Combine historical performance with match-specific contextual data effectively.

Transformer-Based Time Series Models



Autoformer replaces self-attention with an auto-correlation mechanism, which identifies periodic patterns by computing correlations across different time steps.

It also decomposes the input data into:

- *Trend Component:* Captures overall direction or growth.
- *Seasonal Component:* Captures recurring patterns.

Hence, it captures long-term dependencies directly by focusing on periodic relationships rather than individual time-step interactions.

Informers employ ProbSparse Self-attention, which allows them to focus on the most important connections in the data, dramatically reducing computational complexity.

By using self-attention distilling, it refines the input sequence and can extract the essence of the data without getting bogged down in details.

Results

RESULTS OF OUR EXPERIMENTS

Model	MAE	MSE	RMSE
Random Forest	32.173	1662.395	40.772
XGBoost	30.303	1311.129	36.209
ANN (LargeNet)	**24.669**	**1195.053**	**34.567**
LSTM (Long Short-Term Memory)	32.173	1278.518	35.756
GRU (Gated Recurrent Unit)	26.191	1286.699	35.871
BiGRU	26.187	1295.174	35.988
LSTM with attention	26.182	1295.232	35.989
GRU with attention	26.186	1292.063	35.945
Time Series Transformer	27.114	1,303.896	36.109

Conclusion

After extensive experimentation, we concluded that Time Series might not be the best approach to solve the problem

Non-Time-Dependent Features:

- *Opponent strength:* How well does the player perform against specific teams?
- *Venue impact:* Performance variations at different grounds.
- *Match conditions:* Pitch type, injury, etc.

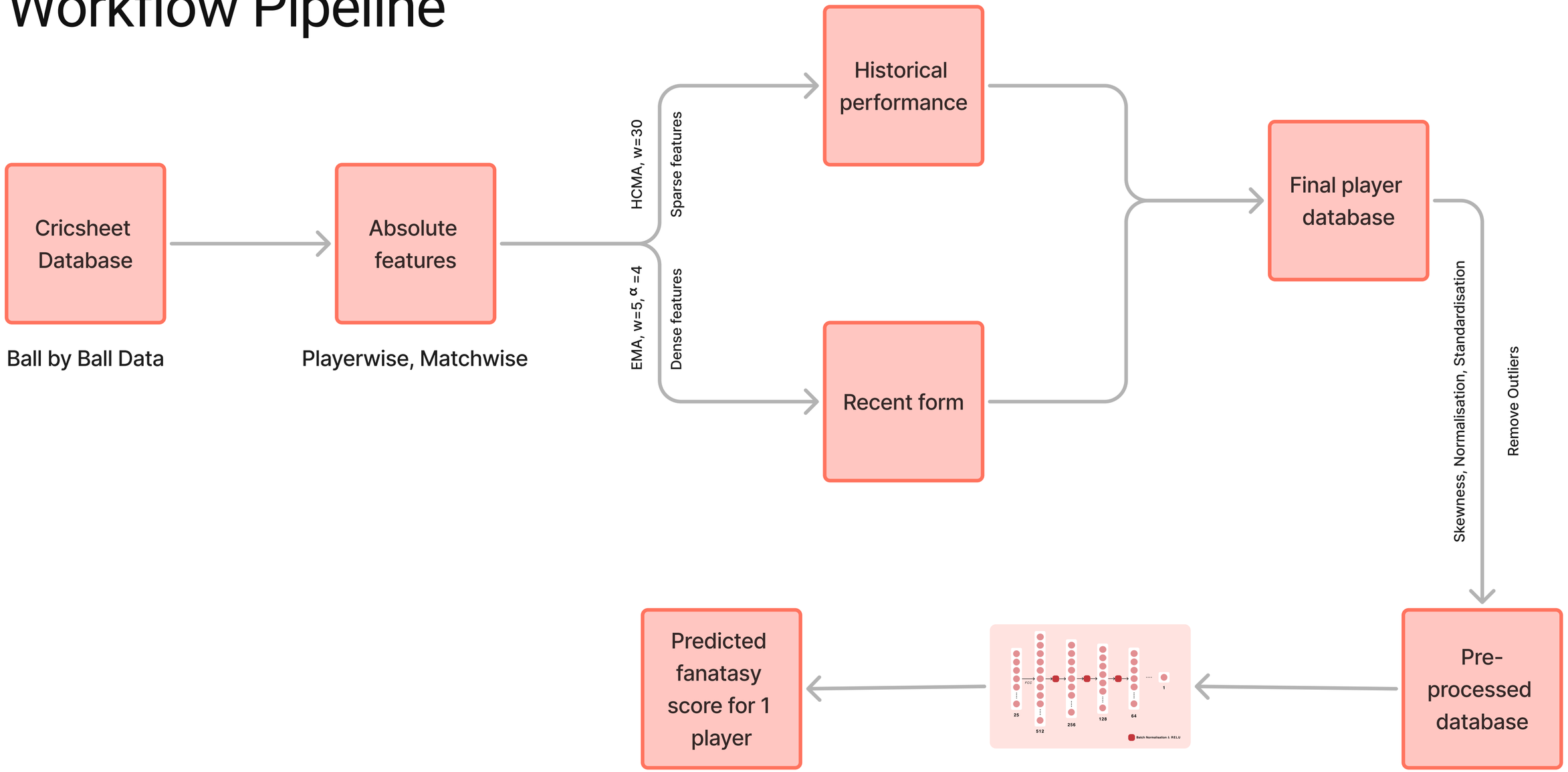
Short-term dependencies

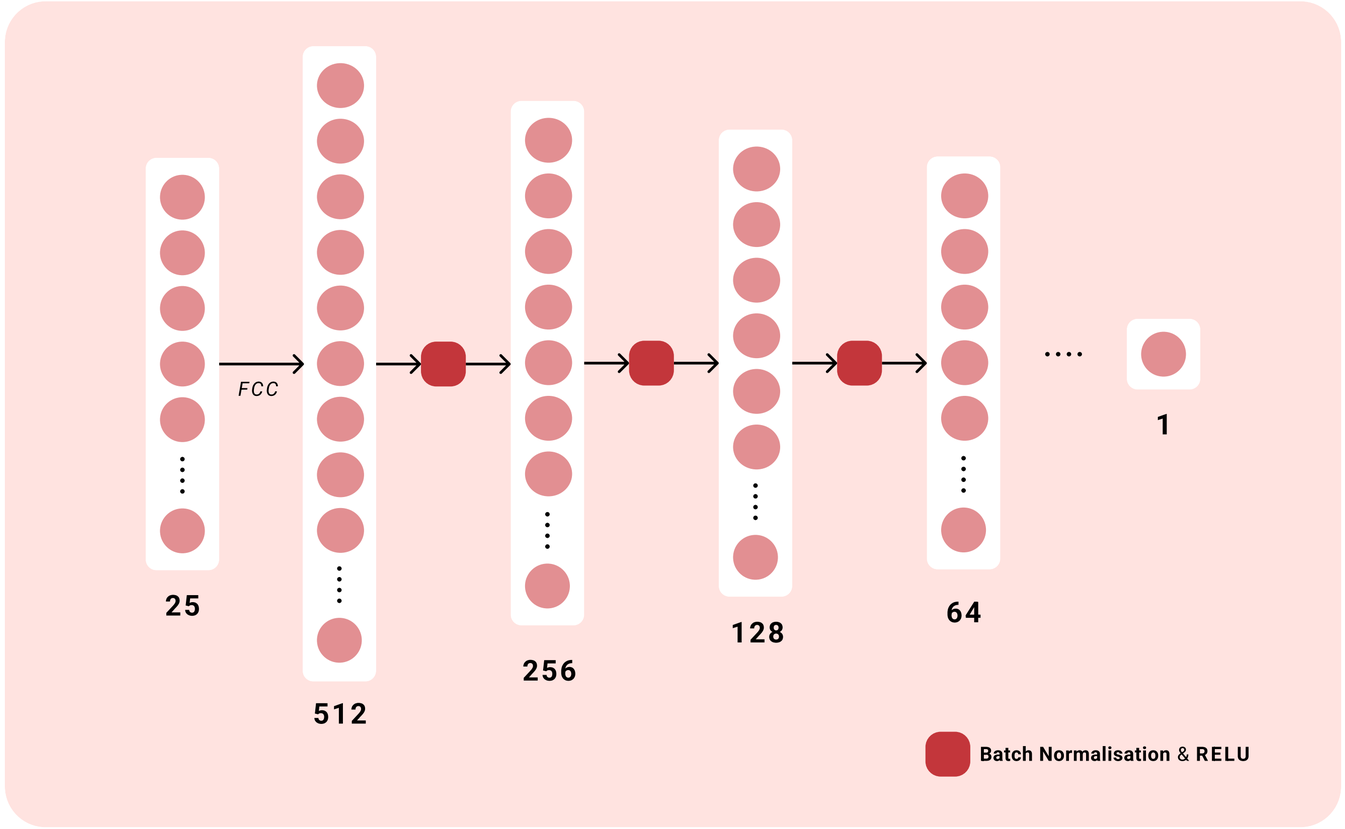
Time Series models are designed to learn long-term dependencies, but short-term dependencies are often more relevant than performances from several seasons ago.

Inter-Player Interactions:

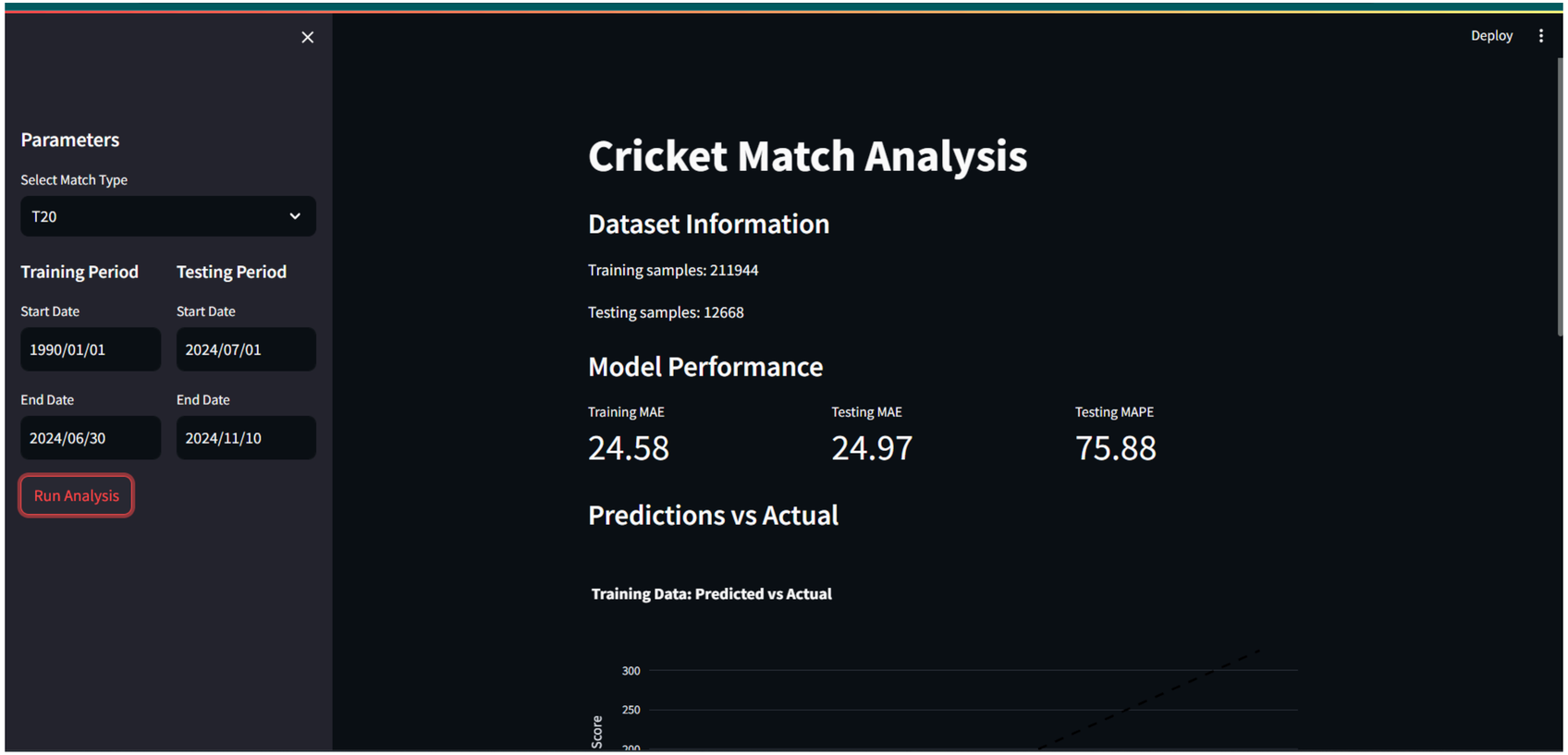
Fantasy points depend on team dynamics, and these dependencies are naturally not captured in a pure time series framework.

Workflow Pipeline

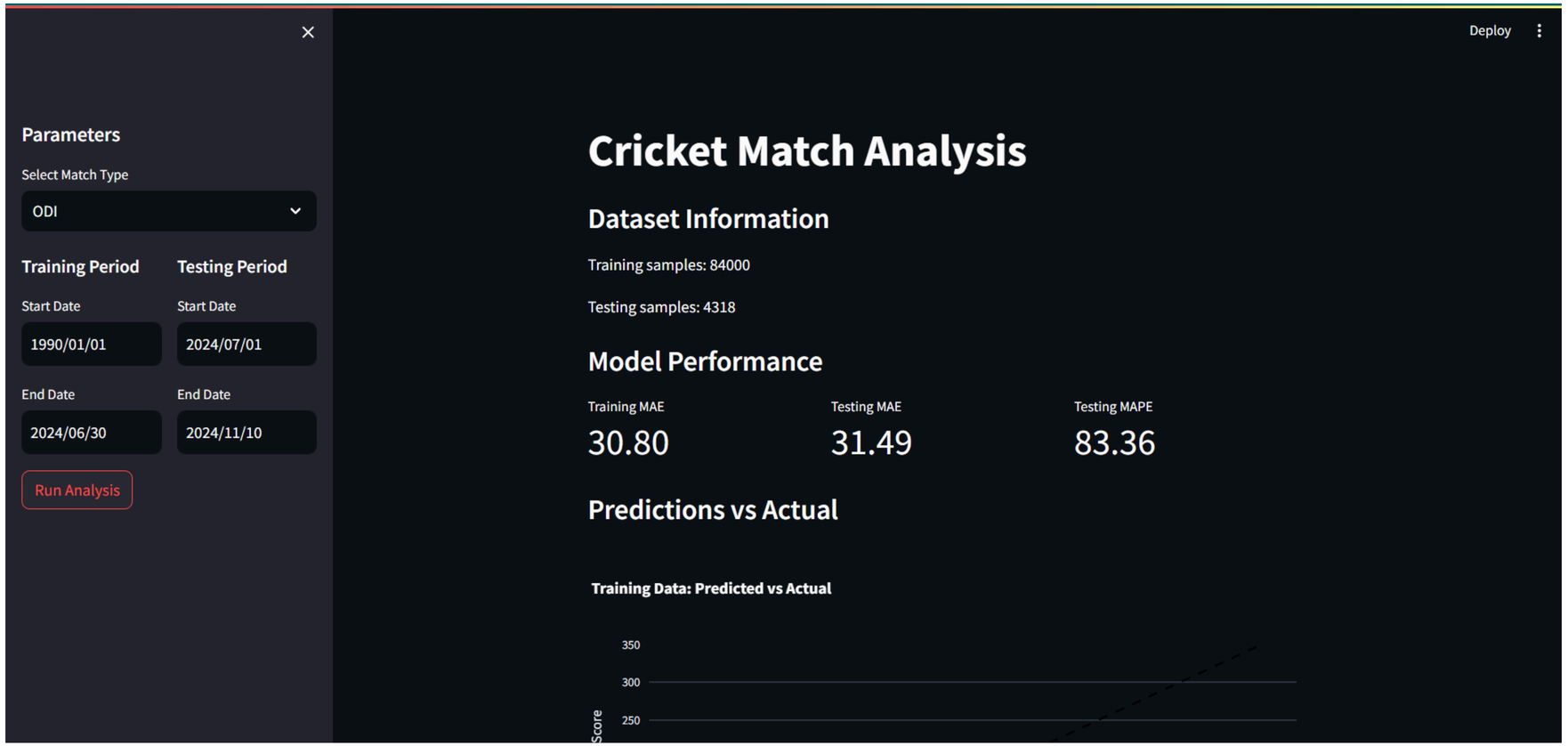



FCC
25
512
256
128
64
1
Batch Normalisation & RELU

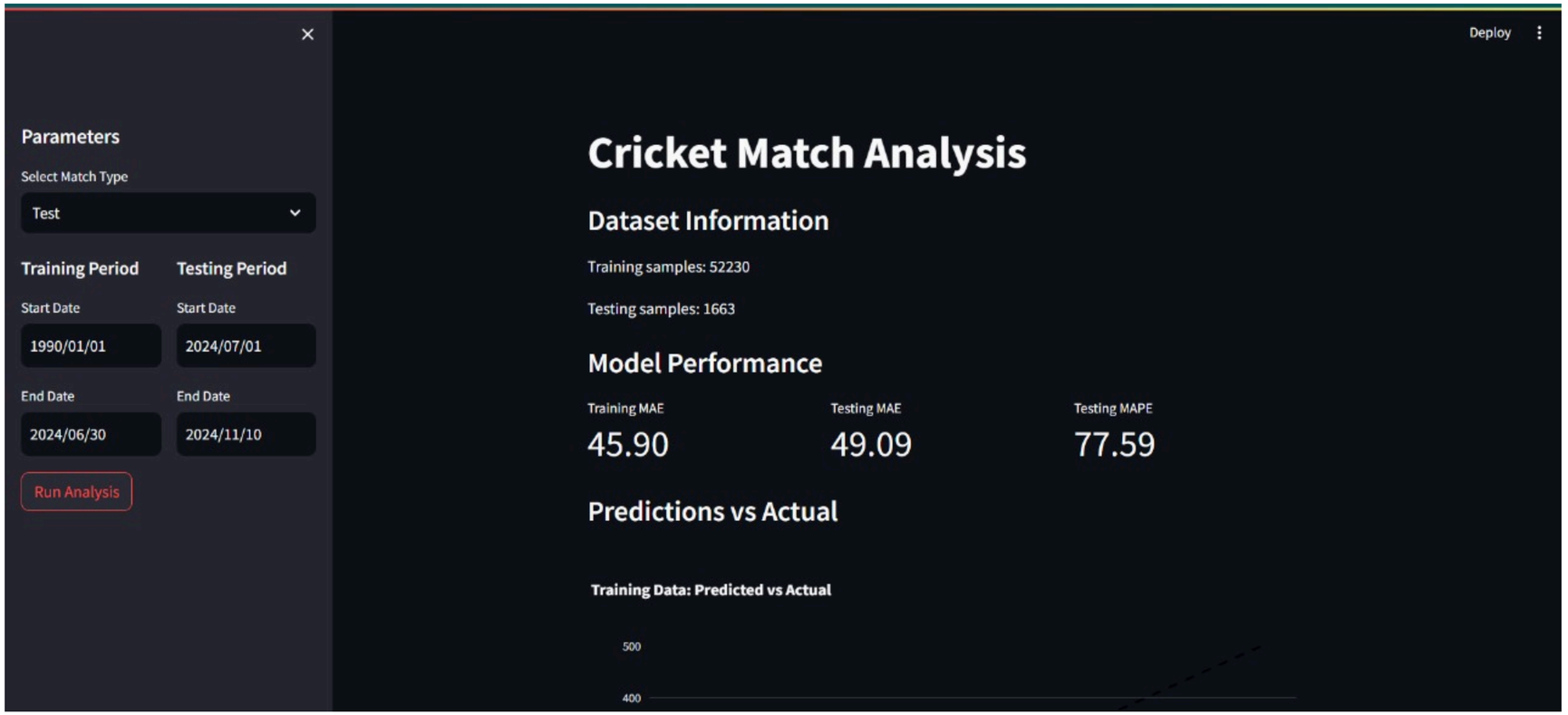
RESULTS


Deploy
Parameters
Select Match Type
T20
Training Period
Start Date
1990/01/01
End Date
2024/06/30
Testing Period
Start Date
2024/07/01
End Date
2024/11/10
Run Analysis
Cricket Match Analysis
Dataset Information
Training samples: 211944
Testing samples: 12668
Model Performance
Training MAE
24.58
Testing MAE
24.97
Testing MAPE
75.88
Predictions vs Actual
Training Data: Predicted vs Actual
300
250
Score

RESULTS


Deploy
Parameters
Select Match Type
ODI
Training Period
Start Date
1990/01/01
End Date
2024/06/30
Testing Period
Start Date
2024/07/01
End Date
2024/11/10
Run Analysis
Cricket Match Analysis
Dataset Information
Training samples: 84000
Testing samples: 4318
Model Performance
Training MAE
30.80
Testing MAE
31.49
Testing MAPE
83.36
Predictions vs Actual
Training Data: Predicted vs Actual
350
300
250
Score

RESULTS


Deploy
Parameters
Select Match Type
Test
Training Period
Start Date
1990/01/01
End Date
2024/06/30
Testing Period
Start Date
2024/07/01
End Date
2024/11/10
Run Analysis
Cricket Match Analysis
Dataset Information
Training samples: 52230
Testing samples: 1663
Model Performance
Training MAE
45.90
Testing MAE
49.09
Testing MAPE
77.59
Predictions vs Actual
Training Data: Predicted vs Actual
500
400