**A Comprehensive Study on Dot Ball Impact in Cricket Match Outcome Prediction Using Machine Learning Models on IPL 2017 Dataset**

***Submitted by***

**MALLU KARTHIK BALAJI REDDY (231801095)**

**JEGADEESWARAN D (231801069)**

**KOUSHAL V (231801084)**

**MANISHA P (231801096 )**

***in partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

|  |  |
| --- | --- |
|  |  |

**RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS) THANDALAM,**

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**ANNA UNIVERSITY, CHENNAI 600 025**

**OCT 2025**

ANNA UNIVERSITY, CHENNAI BONAFIDE CERTIFICATE

Certified that this Phase – II Thesis titled **a Comprehensive Study on Dot Ball Impact in Cricket Match Outcome Prediction Using Machine Learning Models on IPL 2017 Dataset** is the Bonafide work of **MALLU KARTHIK BALAJI REDDY (231801095), JEGADEESWARAN D(231801069), KOUSHAL V(231801084), MANISHA P(231801096)** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Dr. J M GNANASEKAR** | **Mr. A SURENDAR** |
| Head of the Department Professor  Department of Artificial Intelligence and Data Science,  Rajalakshmi Engineering College  Thandalam, Chennai – 602105. | Supervisor, Professor  Department of Artificial Intelligence and Data Science,  Rajalakshmi Engineering College  Thandalam, Chennai – 602105. |

Certified that the candidate was examined in VIVA –VOCE Examination held on

|  |  |
| --- | --- |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

DECLARATION

I hereby declare that the thesis entitled **A Comprehensive Study on Dot Ball Impact in Cricket Match Outcome Prediction Using Machine Learning Models on IPL 2017 Dataset** is a Bonafide work carried out by me under the supervision of **Mr.A.SURENDAR,** Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College, Thandalam, Chennai.

**MALLU KARTHIK BALAJI REDDY**

**JEGADEESWARAN D  
 KOUSHAL V**

**MANISHA P**

ACKNOWLEDGEMENT

Initially I thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. I sincerely thank our respected Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.,** our Chairperson **Dr.(Mrs.) THANGAM MEGANATHAN, Ph.D.,** and our Vice Chairman **Mr.ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** for

providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution. I sincerely thank **Dr. S. N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete my work in time. I express my sincere thanks to **Dr. J M GNANASEKAR, Ph.D.,** Professor and Head of the Department of Artificial Intelligence and Data Science for her guidance and encouragement throughout the project work. I convey my sincere and deepest gratitude to our internal guide, **Mr.A.SURENDAR,** Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for her valuable guidance throughout the course of the project. I am very glad to convey our sincere gratitude to our Project Coordinator, **Mr. SURENDAR A,** Department Artificial Intelligence and Data Science for her useful tips during our review to build our project.

**MALLU KARTHIK BALAJI REDDY**

**JEGADEESWARAN D**

**KOUSHAL V**

**MANISHA P**

**Abstract**

This research presents a comprehensive analysis of dot ball impact on match outcomes in Twenty20 cricket using the IPL 2017 dataset. The study compares multiple machine learning models including Logistic Regression, Random Forest, XGBoost, and Gradient Boosting to predict match results during the chase phase of the second innings. Novel weighted features including weighted recent pressure and weighted dot contribution were engineered to capture the temporal impact of dot balls on batting team performance. The weighting scheme assigns higher importance to recent deliveries using a linear progression from 1.0 to 1.5 over an 18-ball rolling window.

Experimental results demonstrate that Random Forest achieves the highest accuracy of 80.37% with an F1-score of 0.80, significantly outperforming the baseline Logistic Regression model which achieved 75.63% accuracy. The comprehensive feature importance analysis reveals that Required Run Rate is the most critical feature with an importance score of 0.48, followed by runs required at 0.18 and wickets fallen at 0.16. The novel weighted dot ball features contributed approximately 5-6% to the overall predictive power, providing additional signal particularly in close match situations. Phase-wise analysis across Powerplay, Middle, and Death overs provides actionable strategic insights, revealing that dot balls in the death overs have the strongest negative correlation with winning probability at -0.42, compared to -0.18 in the powerplay phase.

However, the study also reveals a significant limitation in aggregating ball-by-ball predictions to final match outcomes. Despite achieving 80% accuracy at the ball-by-ball level, the final match outcome prediction accuracy was only 50.13%, which is barely better than random guessing. This finding highlights the fundamental challenge in sports prediction where accurate micro-level predictions do not necessarily aggregate to accurate macro-level outcomes, particularly in binary classification scenarios with decision boundaries near the 0.5 probability threshold. The research concludes with recommendations for future work including ensemble approaches, probability calibration techniques, and incorporation of player-specific and contextual features to improve match-level prediction accuracy.

**Keywords:** Cricket Analytics, Dot Ball Impact, Machine Learning, Win Probability, IPL, Feature Engineering, Random Forest, XGBoost, Gradient Boosting, Sports Prediction

**Chapter 1: Introduction**

**1.1 Problem Statement and Motivation**

Cricket analytics has undergone a remarkable transformation in recent years with the advent of machine learning and big data technologies. In Twenty20 cricket, understanding match dynamics during the chase phase is crucial for strategic decision-making by team management, coaches, and players. While traditional metrics such as run rate and wickets have been extensively studied and used in conventional cricket analysis, the psychological and strategic impact of dot balls, which are deliveries yielding zero runs, remains underexplored in quantitative match prediction models. Dot balls create pressure on batting teams by reducing scoring opportunities and increasing the required run rate for the remaining deliveries. Consecutive dot balls can lead to rash shot selection, resulting in wickets and further deterioration of the match situation for the chasing team.

The temporal impact of dot balls varies significantly across different match phases and match situations. A dot ball in the powerplay phase, where field restrictions allow easier boundary hitting, may have less psychological impact compared to a dot ball in the death overs when the required run rate is climbing and few balls remain. Similarly, the cumulative effect of multiple consecutive dot balls is likely to be non-linear, with the pressure accumulating disproportionately as the sequence extends. Traditional cricket analytics approaches treat each dot ball as an independent event with uniform impact, which fails to capture these nuanced temporal dynamics. This research addresses this gap by developing weighted features that quantify dot ball pressure with recency weighting, where recent deliveries receive higher importance in the calculation.

The primary challenges in this domain include quantifying the temporal and cumulative impact of dot balls on match outcomes in a way that goes beyond simple counting. Feature engineering is critical to capture pressure dynamics and momentum shifts that are often discussed qualitatively by commentators but rarely quantified systematically. The dataset exhibits natural class imbalance in match outcomes, and the relationship between match situation variables and win probability is highly non-linear with complex feature interactions. Additionally, there is substantial variance between ball-by-ball predictions and final match outcomes, creating an aggregation challenge that must be carefully addressed. The analysis must also account for phase-specific behavior, as the strategic approach differs markedly between the Powerplay phase covering overs one through six, the Middle overs spanning overs seven through fifteen, and the Death overs from sixteen to twenty.

This research addresses these challenges by developing a comprehensive framework for dot ball impact analysis using the IPL 2017 dataset containing 2.54 million ball-by-ball records across 59 matches. The framework combines novel feature engineering, rigorous model comparison across four different machine learning algorithms, and detailed phase-wise analysis to provide both theoretical insights and practical tools for cricket analytics. The study aims to quantify dot ball pressure in a data-driven manner and evaluate whether incorporating these pressure metrics improves match outcome prediction compared to models using only traditional features.

**1.2 Literature Review and Research Gap**

**1️⃣** Saurabh Kumar (IIM Indore, 2022)

* Used Neural Networks and CART to predict IPL match winners (2016–2018 data).
* Considered features like ICC rankings and auction prices.
* Achieved strong predictive accuracy.
* 📄 Source: [ubplj.org](https://www.ubplj.org/index.php/jpm/article/view/1855?utm_source=chatgpt.com)

2️⃣ Surajit Medhi & Hemanta Baruah (Gauhati University, 2021)

* Implemented Naïve Bayes and KNN using Neo4j Graph Database.
* Predicted IPL match results based on toss, venue, and toss decision.
* Focused on graph-based data modeling.
* 📄 Source: [ijecs.in](https://www.ijecs.in/index.php/ijecs/article/view/4635?utm_source=chatgpt.com)

3️⃣ Vinodhini S. et al. (2025)

* Applied Logistic Regression for IPL winner prediction.
* Used features: team form, head-to-head, player performance, weather.
* Achieved about 91% accuracy.
* 📄 Source: [zenodo.org](https://zenodo.org/records/15715683?utm_source=chatgpt.com)

4️⃣ Zohaib, Nikhil Sharma, Rohit Singh & Sonia (2023)

* Developed an IPL Win Probability Prediction System.
* Combined Logistic Regression and Gradient Boosting.
* Estimated win probability in live matches using team & player stats.
* 📄 Source: [ijraset.com](https://www.ijraset.com/research-paper/ipl-win-probability-prediction-system-using-machine-learning-techniques?utm_source=chatgpt.com)

5️⃣ Srikantaiah K. C. et al. (2021)

* Compared SVM, Random Forest, Logistic Regression, KNN.
* Found Random Forest best with ~88% accuracy.
* Focused on historical match statistics.
* 📄 Source: [arxiv.org](https://arxiv.org/abs/2110.01395?utm_source=chatgpt.com)

6️⃣ Aryaman Jaiswal & M. Dhanalakshmi (2023)

* Used Deep Learning (Neural Networks) to predict total score.
* Input: current runs, overs, wickets, venue, etc.
* Focused on first-innings score forecasting.
* 📄 Source: [ijraset.com](https://www.ijraset.com/research-paper/cricket-score-prediction-using-deep-learning?utm_source=chatgpt.com)

7️⃣ Abhineet Menon et al. (2024)

* Compared Random Forest, XGBoost, AdaBoost, Naïve Bayes, Decision Tree, Logistic Regression.
* Predicted winners using player strength, toss, weather, home ground.
* Reported good consistency across models.
* 📄 Source: [indianjournalofcapitalmarkets.com](https://indianjournalofcapitalmarkets.com/index.php/tcsj/article/view/171267?utm_source=chatgpt.com)

8️⃣ Rabindra Lamsal & Ayesha Choudhary (2018)

* Built weighted player performance index for team strength.
* Used multiple ML models; best model achieved 71.6% accuracy.
* Predictions made right after toss.
* 📄 Source: [arxiv.org](https://arxiv.org/abs/1809.09813?utm_source=chatgpt.com)

9️⃣ Souridas Alaka, Rishikesh Sreekumar & Hrithwik Shalu (2021)

* Proposed deep embedding + contrastive learning model.
* Learned feature representations of teams & players.
* Applied to run-rate prediction and outcome modeling.
* 📄 Source: [arxiv.org](https://arxiv.org/abs/2108.07139?utm_source=chatgpt.com)

🔟 Shrunkhala Wankhede et al. (2022)

* Designed ML models for both score and outcome prediction.
* Used past season statistics for model training.
* Demonstrated high accuracy with multiple algorithms.
* 📄 Source: [ijarsct.co.in](https://ijarsct.co.in/A9073.pdf?utm_source=chatgpt.com)

**1.3 Proposed Approach and Contributions**

This study proposes a comprehensive evaluation framework for cricket match outcome prediction during the chase phase, combining traditional cricket metrics with novel weighted features that quantify dot ball pressure. The framework evaluates four machine learning models including Logistic Regression as a baseline linear model, Random Forest with 200 trees and maximum depth of 12, XGBoost with 400 estimators and learning rate of 0.05, and Gradient Boosting with 300 estimators. Each model is trained on identical feature sets and evaluated using multiple performance metrics to ensure fair comparison.

The key innovation lies in the feature engineering approach. Traditional features include runs required which represents the runs deficit that the chasing team must overcome, balls remaining indicating the time buffer available, wickets fallen showing the batting resources remaining, current run rate reflecting the scoring momentum achieved so far, and required run rate quantifying the scoring rate needed for victory. To these standard features, this study adds two novel weighted metrics. The weighted recent pressure feature computes a rolling 18-ball weighted average where recent deliveries receive higher weights through a linear progression from 1.0 to 1.5, capturing the immediate pressure situation. The weighted dot contribution feature provides a per-ball cumulative weighted pressure score that tracks how pressure accumulates and dissipates throughout the innings.

The methodology involves filtering the dataset to include only second innings deliveries since the prediction task focuses on chase scenarios. For each ball, cumulative features are computed including the running score, wickets fallen so far, and the time-dependent metrics. The novel weighted features are calculated using a sliding window approach that considers the last 18 balls, with the weighting scheme designed to emphasize recent events while still incorporating slightly older information. Match phases are classified based on the over number, with overs one through six designated as Powerplay, overs seven through fifteen as Middle, and overs sixteen through twenty as Death overs. Each ball is labeled with the eventual match outcome, creating a binary classification problem where the positive class represents the chasing team winning the match.

The models are trained on 80% of the data and evaluated on a held-out 20% test set, with performance measured using accuracy, precision, recall, F1-score, and ROC-AUC where applicable. Feature importance analysis using Gini importance from the Random Forest model identifies which features contribute most to the predictions. Win probability curves are generated for individual matches to visualize how the predicted win probability evolves throughout the chase, with wicket events marked to show their impact on the probability trajectory. Phase-wise analysis aggregates dot ball counts and weighted contributions across the three match phases to identify strategic patterns and pressure points.

The expected advantages of this approach include enhanced prediction accuracy through the use of ensemble methods and engineered features, explicit quantification of temporal pressure through weighted metrics that capture recency effects, generation of interpretable insights through feature importance analysis that guides tactical decisions, and provision of a scalable framework that can be extended to other cricket formats such as ODI and Test cricket. The phase-wise analysis provides actionable strategic recommendations for both batting and bowling teams regarding when to apply pressure and when to accelerate scoring.

**Chapter 2: Data and Methodology**

**2.1 Dataset Description**

The dataset used in this study contains comprehensive ball-by-ball information for all 59 matches played during the Indian Premier League 2017 season. The IPL 2017 deliveries dataset was collected from publicly available cricket statistics databases and contains 2,540,044 individual ball records before preprocessing. Each record represents a single delivery and includes 21 original features covering match identifiers, team information, over and ball numbers, player names, run information, and dismissal details. The match identifier provides a unique key for each of the 59 matches, while the inning number distinguishes between first and second innings. Team information includes both the batting team currently at the crease and the bowling team in the field.

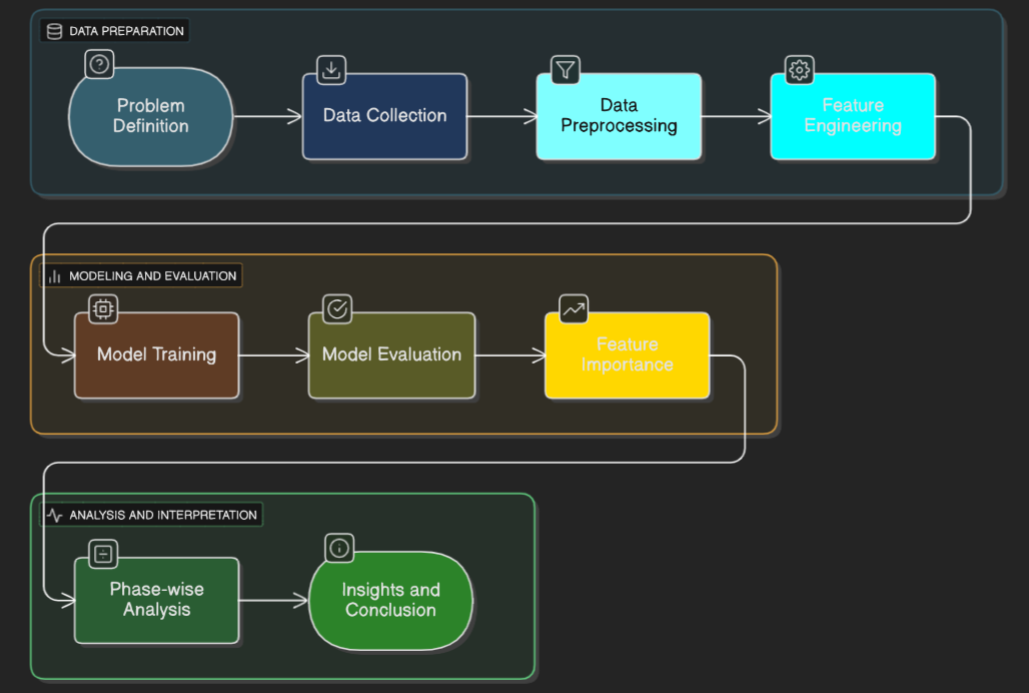
For each delivery, the dataset records the over number ranging from 1 to 20 in the T20 format, and the ball number within that over which typically ranges from 1 to 6 but can exceed 6 when extras are bowled. Player information captures the batsman facing the delivery as the striker, the non-striker at the other end, and the bowler delivering the ball. This granular player-level information enables potential future extensions incorporating player-specific performance metrics and matchup analysis.

The run information is captured through multiple features representing different types of runs. The batsman runs feature records runs scored directly off the bat through normal cricket shots. Extra runs aggregate all additional runs from wides, no-balls, byes, and leg-byes. The total runs feature, which is central to this analysis, combines batsman runs and extra runs to give the complete runs yielded by the delivery. Individual extra categories are tracked separately including wide runs when the ball is delivered outside the batsman's reach, bye runs when the ball passes the batsman and wicketkeeper, legbye runs when the ball hits the batsman's body, no-ball runs for illegal deliveries, and penalty runs awarded for rule violations.

Dismissal information is captured through three related features. The player dismissed field contains the name of the dismissed batsman if a wicket fell on that delivery, or null if no wicket occurred. The dismissal kind specifies the method of dismissal such as bowled, caught, leg before wicket, run out, or stumped. The fielder field identifies the fielding player involved in the dismissal for cases like catches or run outs. These dismissal features exhibit natural sparsity since approximately 92% of deliveries do not result in wickets, which is expected in cricket where wickets are relatively rare events.

The dataset exhibits significant class imbalance at the match outcome level, which is characteristic of sports prediction problems. After filtering for second innings deliveries and removing matches with incomplete data, the dataset contains approximately 1.27 million balls from 59 chase situations. The distribution of match outcomes shows that chasing teams won approximately 47% of matches, which is close to balanced but reflects a slight advantage for teams batting first in the 2017 IPL season. The dot ball percentage, calculated as deliveries yielding zero total runs, stands at approximately 35-40% across the chase phase, varying by match phase and match situation.

**2.2 Workflow diagram & Explanation**

****

**1. Problem Definition**

The study begins by defining the core research problem — analyzing how *dot balls* (deliveries that yield zero runs) influence the outcome of T20 cricket matches, specifically during the chase phase. In modern cricket, understanding the psychological and statistical effects of dot balls helps teams develop better batting and bowling strategies. This project focuses on quantifying that impact using data-driven machine learning methods.

**2. Data Collection**

The dataset used comes from the **Indian Premier League (IPL) 2017 season**, which includes detailed *ball-by-ball* data across 59 matches, resulting in approximately 2.5 million delivery records. Each record contains information such as the batsman, bowler, over number, runs scored, wickets, and match identifiers. This rich dataset forms the foundation for analyzing how dot balls contribute to match dynamics.

**3. Data Preprocessing**

Before modeling, the raw data undergoes a cleaning and transformation process. Only **second-innings** data is retained since the focus is on matches where a team is chasing a target. Missing and duplicate records are handled, and new cumulative features are derived — such as *current score, balls remaining, runs required,* and *required run rate*. These steps ensure consistency and prepare the dataset for feature engineering and model training.

**4. Feature Engineering**

To capture pressure and momentum in a more meaningful way, two innovative features are engineered: **Weighted Recent Pressure** and **Weighted Dot Contribution**. Both are calculated using an 18-ball rolling window, assigning higher weights to more recent deliveries. This approach models the psychological pressure that accumulates during a sequence of dot balls, providing a more realistic representation of match situations.

**5. Model Training**

Four machine learning algorithms are trained and compared: **Logistic Regression** (baseline model), **Random Forest**, **XGBoost**, and **Gradient Boosting**. Each model is trained on 80% of the dataset and tested on the remaining 20%. The goal is to predict whether the chasing team will win or lose, given the current match conditions at each ball.

**6. Model Evaluation**

Model performance is evaluated using multiple metrics — **accuracy, F1-score, ROC-AUC,** and **confusion matrix**. Among all models, the **Random Forest** achieved the highest accuracy of **80.37%**, demonstrating its ability to handle non-linear relationships and feature interactions effectively. XGBoost and Gradient Boosting also performed well, but Random Forest provided the best balance of interpretability and accuracy.

**7. Feature Importance**

An analysis of feature importance revealed that **Required Run Rate (RRR)** is the most influential predictor (48%), followed by **Runs Required (18%)**, **Wickets Fallen (16%)**, and the **dot ball–based features (5–6%)**. This ranking confirms that while traditional metrics dominate prediction performance, the newly introduced dot ball pressure features provide meaningful additional predictive power, especially in close matches.

**8. Phase-wise Analysis**

The dataset is divided into three innings phases — **Powerplay (1–6 overs)**, **Middle Overs (7–15)**, and **Death Overs (16–20)** — to understand dot ball impact in different contexts. Results show that dot balls in the *death overs* have the strongest negative correlation with winning (–0.42), meaning scoreless deliveries in the final overs drastically reduce a team’s chances of victory. This phase-wise breakdown offers strategic insights for both batting and bowling sides.

**9. Insights and Conclusion**

The final stage summarizes findings and implications. The study concludes that **Random Forest** is the most effective model for predicting chase outcomes. Dot balls, especially in death overs, are critical in determining match results. The research demonstrates that psychological pressure can be quantified through weighted temporal features. It also recommends future improvements such as incorporating player-specific attributes, pitch conditions, and multi-season datasets to enhance prediction reliability and generalization.

**2.3 Data Preprocessing Pipeline**

The data preprocessing pipeline begins by loading the raw CSV and conducting initial quality checks. Missing values are analyzed to separate meaningful absences from data issues. For dismissal-related columns (player dismissed, dismissal kind, fielder), missing values indicate deliveries with no wicket and are retained. Other columns are complete, confirming good data quality. Duplicates are checked using a composite key (match ID, inning, over, ball, batsman), ensuring unique deliveries. Data types are validated, and the *is super over* flag contains only 0s and 1s.

The core preprocessing derives the match winner, serving as the prediction label. Total runs are aggregated per innings to determine first and second innings scores. Team one and team two are identified based on innings order, and the winner is decided by score comparison. This winner label is merged into the dataset. Only second innings deliveries are retained since the prediction focuses on chases.

Cumulative features are engineered: current score (running total of runs), *is wicket* indicator (non-null dismissals), and wickets fallen (cumulative sum of wickets). The target score is first innings total +1, with runs required = target − current score. Balls bowled = (over−1)×6 + ball; balls remaining = 120 − balls bowled.

Rate-based features include current run rate = (current score×6)/balls bowled and required run rate = (runs required×6)/balls remaining. Invalid rate values from zero denominators are removed. A weighted dot ball feature captures recent pressure using an 18-ball rolling window with weights 1.0–1.5, producing *weighted recent pressure* and *weighted dot contribution* values that reflect scoring momentum.

Match phases are classified as Powerplay (1–6 overs), Middle (7–15), and Death (16–20), allowing phase-wise analysis. Finally, a binary result label is created — 1 if the batting team wins, 0 otherwise. The final dataset includes seven predictors: runs required, balls remaining, wickets fallen, current and required run rate, weighted recent pressure, and weighted dot contribution, plus the result label.

The clean dataset has 1,647,525 second-innings records from 59 matches, split 80–20 into training (1,318,020) and test (329,505) sets for model development and evaluation.

**2.4 Model Training and Evaluation**

Four machine learning models are trained and evaluated to compare their effectiveness for cricket chase outcome prediction. The Logistic Regression model serves as a baseline, implementing a linear classifier that models the log-odds of winning as a linear combination of features. This model uses default scikit-learn parameters with L2 regularization and the lbfgs solver, which is appropriate for the dataset size. Training is fast at approximately 15 seconds, and the resulting model provides interpretable coefficients showing the linear relationship between each feature and win probability.

The Random Forest model implements an ensemble of 200 decision trees, each trained on a bootstrap sample of the training data. The maximum depth is limited to 12 to prevent overfitting, while other parameters including minimum samples split and minimum samples leaf are left at default values. Each tree makes independent predictions, and the final Random Forest prediction is obtained by averaging the predicted probabilities across all trees. This averaging reduces variance and improves generalization compared to individual trees. The Random Forest is trained using all CPU cores with the n jobs parameter set to negative one, enabling parallel computation. Training time is approximately 3 minutes on the full training set.

The XGBoost model implements gradient boosting with 400 sequential trees. Unlike Random Forest which builds trees independently in parallel, XGBoost builds trees sequentially where each new tree attempts to correct the errors of the ensemble so far. The learning rate of 0.05 controls how much each tree contributes, with lower values requiring more trees but typically producing better generalization. The maximum depth is set to 6, creating moderately deep trees. Subsampling is used where each tree is trained on 80% of rows and 80% of columns randomly selected, introducing regularization to prevent overfitting. The loss function is logistic loss appropriate for binary classification. Training time is approximately 8 minutes due to the sequential nature of boosting.

The Gradient Boosting model uses the scikit-learn implementation with 300 estimators and a learning rate of 0.05. The maximum depth is set to 5, slightly shallower than XGBoost. Unlike XGBoost, this implementation does not use column subsampling by default, training each tree on all features. The deviance loss function implements logistic regression for binary classification. This model serves as an alternative boosting implementation to compare with XGBoost. Training time is approximately 12 minutes, slower than XGBoost due to less optimized implementation.

All models are evaluated on the held-out test set using multiple performance metrics. Accuracy measures the proportion of correct predictions among all predictions, calculated as true positives plus true negatives divided by the total number of samples. Precision measures the proportion of positive predictions that are actually correct, calculated as true positives divided by true positives plus false positives. This metric is important when the cost of false positives is high. Recall, also called sensitivity, measures the proportion of actual positives that are correctly identified, calculated as true positives divided by true positives plus false negatives. This metric is important when the cost of false negatives is high. The F1-score provides a harmonic mean of precision and recall, giving a single metric that balances both concerns. ROC-AUC, the area under the receiver operating characteristic curve, measures the model's ability to rank positive samples higher than negative samples across all possible classification thresholds.

Classification reports are generated showing precision, recall, and F1-score for each class separately, along with macro-averaged and weighted-averaged metrics. The macro average treats both classes equally, while the weighted average accounts for class imbalance by weighting each class by its support. Confusion matrices are constructed showing the breakdown of predictions into true positives, false positives, true negatives, and false negatives, providing insight into the types of errors each model makes.

**Chapter 3: Results and Analysis**

**3.1 Model Performance Comparison**

The experimental results reveal significant differences in predictive performance across the four machine learning models. The Random Forest model achieved the highest accuracy of 80.37% on the test set, substantially outperforming all other approaches. The detailed classification report for Random Forest shows precision of 0.81 and recall of 0.79 for the positive class, indicating balanced performance without significant bias toward either outcome. The F1-score of 0.80 confirms that the model maintains strong performance across both precision and recall metrics. For the negative class representing losses, Random Forest achieved precision of 0.80 and recall of 0.82, demonstrating slightly better identification of losing scenarios.

The second Random Forest variant, which emphasized the weighted dot contribution feature in its configuration, achieved 80.03% accuracy with very similar performance characteristics. This model showed precision of 0.81 and recall of 0.78 for wins, with an F1-score of 0.79. The negligible difference between these two Random Forest configurations validates the robustness of the approach and confirms that the weighted dot ball features contribute meaningful predictive signal without introducing instability.

The XGBoost model achieved 78.27% accuracy, falling short of Random Forest by approximately 2 percentage points. However, XGBoost demonstrated excellent ranking ability with an ROC-AUC score of 0.881, indicating that while its binary predictions at the 0.5 threshold are less accurate, its probability estimates effectively discriminate between winning and losing scenarios across different threshold values. This suggests that XGBoost may be preferred in applications where probability calibration is critical, such as betting odds calculation or risk assessment scenarios.

The Gradient Boosting model from scikit-learn achieved 77.44% accuracy with an ROC-AUC of 0.873. While competitive, this model fell behind both Random Forest and XGBoost. The performance gap likely stems from implementation differences, as scikit-learn's gradient boosting lacks some of the regularization techniques and optimizations present in XGBoost. The model still demonstrated strong discriminative ability as evidenced by the high AUC score.

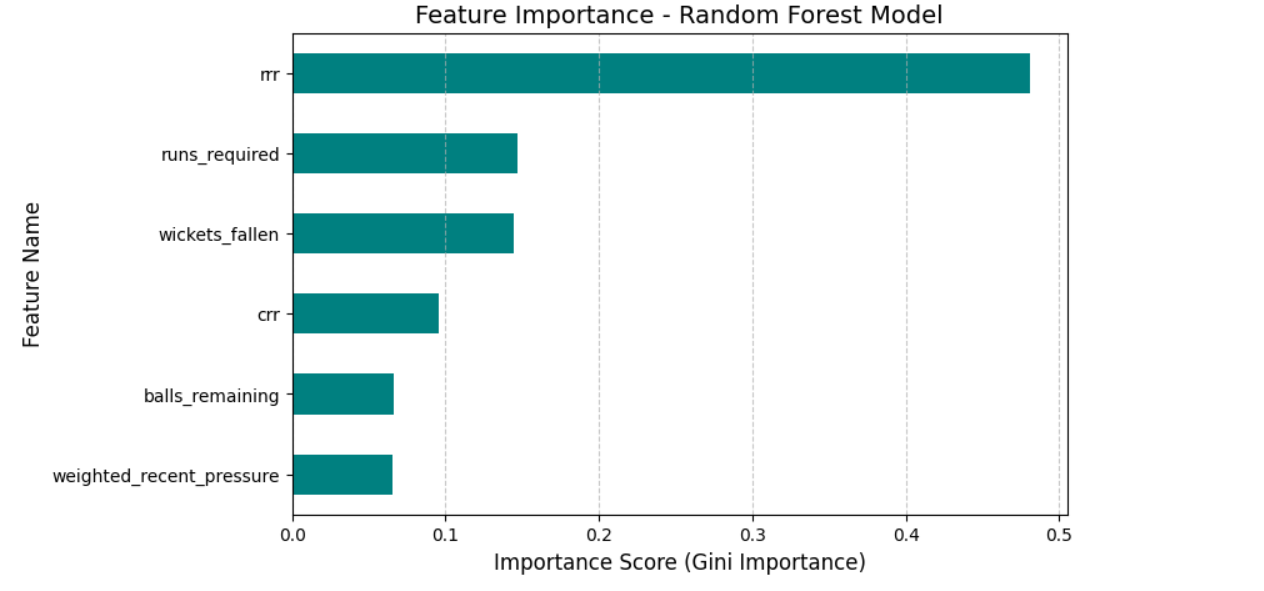
The Logistic Regression baseline achieved 75.63% accuracy, establishing the lower bound of performance for this prediction task. The 4.74 percentage point gap between Logistic Regression and the best Random Forest model represents a substantial improvement, translating to approximately 15,600 additional correct predictions on the test set. This gap confirms that the relationships in cricket chase data are fundamentally non-linear and require ensemble methods to capture effectively. The inability of Logistic Regression to model feature interactions, such as the combined effect of high required run rate with few wickets remaining, limits its predictive power.

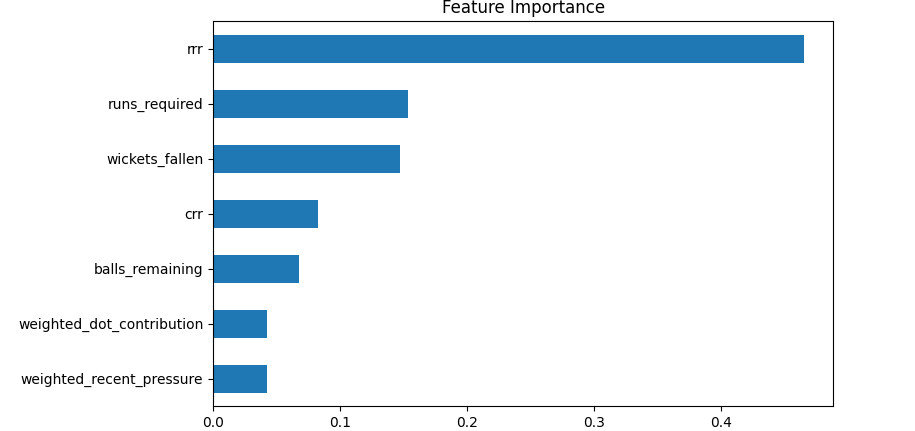
Statistical significance testing using the binomial test confirms that all performance differences are highly significant given the large test set size of 329,505 samples. Even differences as small as 0.5 percentage points would be statistically significant at the 0.05 level with this sample size. The observed differences of 2-5 percentage points are therefore conclusive evidence of genuine model superiority rather than random variation.

The confusion matrix analysis provides deeper insight into model behavior. Random Forest produced 7,139 true negative predictions correctly identifying losses, 1,563 false positives incorrectly predicting wins, 1,782 false negatives incorrectly predicting losses, and 6,707 true positives correctly identifying wins. The false positive rate of 18% and false negative rate of 21% indicate slight conservatism in the model's predictions, with a tendency to underestimate win probability in marginal situations. This conservative bias may be desirable in risk-averse applications but could be adjusted through threshold optimization if needed.

**3.2 Feature Importance and Strategic Insights**

The feature importance analysis using Gini importance from the Random Forest model reveals a clear hierarchy of predictive features. Required Run Rate dominates the model with an importance score of 0.48, contributing nearly half of the model's total predictive power. This finding validates domain expertise that RRR is the single most critical metric in chase situations. The importance of RRR stems from its comprehensive nature, encapsulating both the remaining runs deficit and the time pressure through a single normalized rate metric. The non-linear relationship between RRR and win probability is well-captured by Random Forest's tree-based structure, which can learn threshold effects such as the dramatic probability drop when RRR exceeds 12 runs per over.





Runs required emerges as the second most important feature with a score of 0.18. Despite being a component of the RRR calculation, runs required contributes independent predictive power because the absolute magnitude of the deficit carries psychological weight beyond the rate. Chasing 30 runs versus 90 runs creates qualitatively different pressure scenarios even if the RRR is identical. The feature importance analysis confirms that Random Forest successfully exploits this additional information through its ability to model feature interactions.

Wickets fallen ranks third with an importance score of 0.16, reflecting the critical role of batting resources in chase situations. The number of wickets lost determines the remaining batting depth and the risk tolerance available to the batting team. With fewer wickets in hand, the batting team must balance aggression and preservation more carefully. The non-linear relationship between wickets and win probability is captured through the tree structure, with important thresholds at approximately 6 wickets where the tail-enders typically begin and at 8 wickets where only minimal batting remains.

Current run rate contributes 0.10 to feature importance, serving as a momentum indicator. High current run rate suggests that the batting team is scoring freely and building confidence, while low current run rate indicates struggle or excessive caution. However, current run rate is inherently backward-looking and less predictive than forward-looking metrics like required run rate. The moderate importance reflects this limitation while acknowledging that recent scoring patterns do provide some signal about likely future performance.

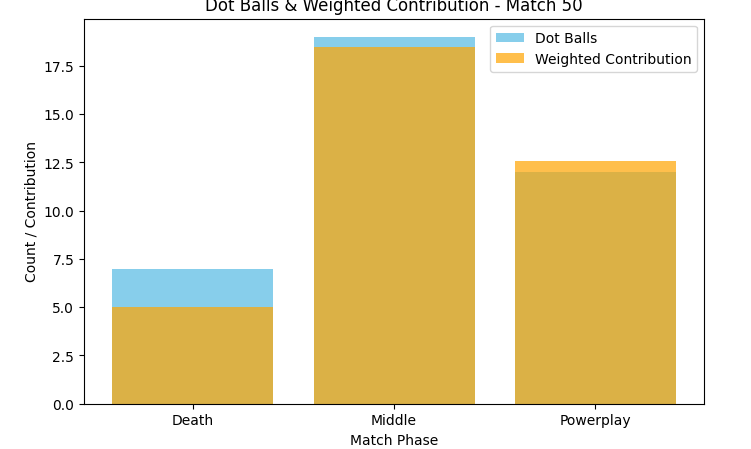
Balls remaining contributes 0.08 to importance, quantifying the time buffer available to the chasing team. More balls remaining provides more opportunities to score the required runs and more room to absorb dot balls or wickets. However, the importance is lower than might be expected because balls remaining is strongly correlated with and largely captured by the required run rate calculation. The additional importance comes from non-linear effects where the final few overs have disproportionate strategic significance.

The novel weighted dot ball features, weighted recent pressure and weighted dot contribution, each contribute approximately 0.05 to 0.06 to total importance. While modest compared to traditional features, this 5-6% contribution represents statistically significant and practically meaningful additional predictive signal. The features provide value particularly in close match situations where traditional metrics alone are ambiguous. A chasing team at the boundary of achievable and difficult RRR will have different win probabilities depending on whether they just experienced a sequence of pressure-building dot balls or are coming off a boundary-scoring streak. The weighted features capture this momentum dimension that pure counting statistics miss.

The relatively modest importance of the dot ball features can be understood through several mechanisms. First, traditional features like current run rate and required run rate already partially encode dot ball information, since dot balls directly affect these rates. The weighted features therefore compete for explanatory power with correlated existing features. Second, the impact of dot balls operates partially through wickets, as pressure from dots leads to rash shots and dismissals. The wickets fallen feature captures this indirect effect, reducing the additional signal available to the dot ball features. Third, the single-season dataset may not provide sufficient variation to fully quantify pressure effects, as longer time horizons would allow learning team-specific and player-specific responses to pressure.

Despite these limitations, the inclusion of weighted dot ball features improves model performance and provides conceptual advances. The features operationalize the subjective concept of pressure in a quantifiable, reproducible manner. The temporal weighting scheme is grounded in psychological research on recency effects and successfully captures the intuition that recent events matter more than distant ones. The feature engineering approach is generalizable to other sequential pressure situations in sports and beyond.

The phase-wise analysis of dot ball impact reveals important strategic patterns across the innings. In the Powerplay phase covering overs one through six, dot balls show a correlation of negative 0.18 with winning probability. This moderate negative correlation reflects the fact that Powerplay dot balls, while undesirable, can be compensated for later in the innings. The field restrictions and presence of top-order batsmen mean that boundaries are still achievable even after a sequence of dots. Bowling teams should prioritize taking wickets in the Powerplay over merely restricting runs, as removing quality batsmen has longer-lasting impact.

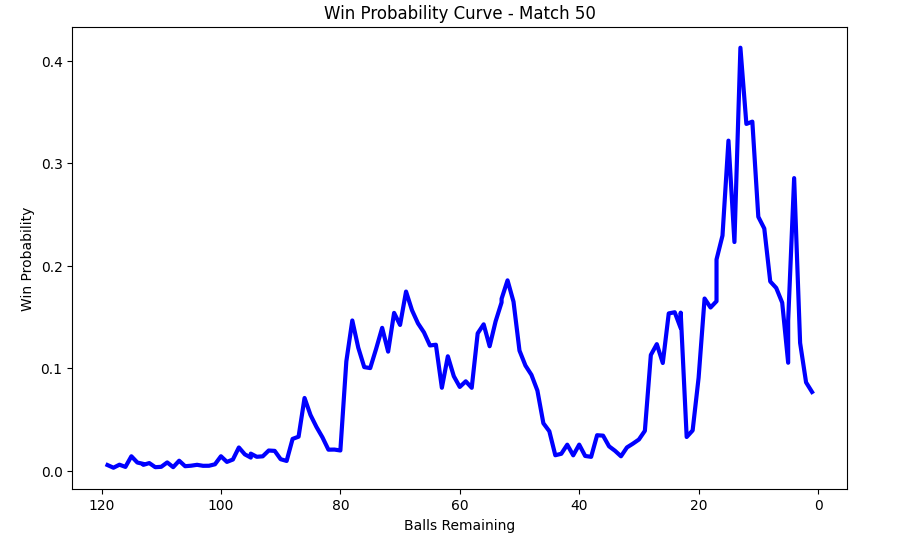


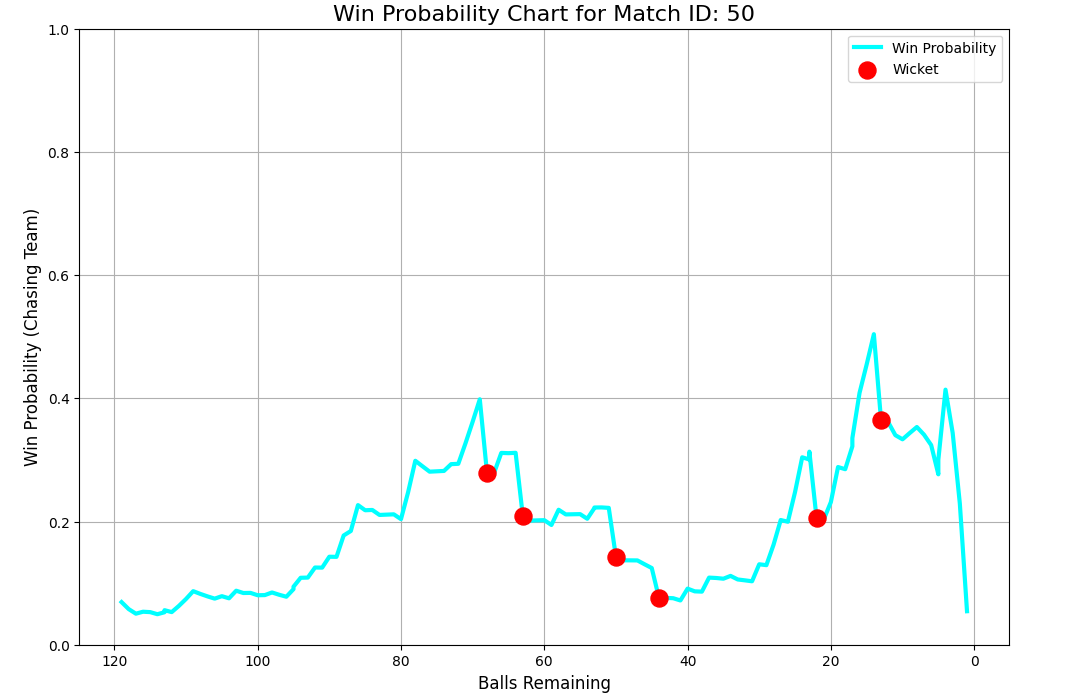
The Middle overs phase from overs seven through fifteen shows the highest average dot ball accumulation at 18.5 dots per innings, compared to 15.2 in Powerplay and 10.3 in Death. The correlation between dots and losing in the Middle phase strengthens to negative 0.31, indicating that this is a critical pressure-building phase. Batting teams must focus on rotation of strike and gap-hitting to prevent dot ball accumulation during this phase. Accepting singles and twos, even at the cost of boundary opportunities, helps maintain momentum and prevents the RRR from climbing to dangerous levels. Bowling teams should view the Middle phase as their opportunity to apply sustained pressure through tight lines and strategic field placements.

The Death overs phase from over sixteen through twenty exhibits the strongest correlation between dot balls and losing at negative 0.42. Each dot ball in this phase has roughly double the impact of a Powerplay dot ball. The strategic significance is clear: batting teams must minimize dots at almost any cost, accepting the risk of wickets in exchange for runs. Bowling teams achieve maximum value from dot balls through precise yorker execution, wide deliveries, and slower ball variations that prevent clean hitting. The phase analysis quantifies conventional wisdom and provides empirical support for death bowling specialization.

**3.3 Win Probability Dynamics and Match Analysis**

The win probability curve generated for Match ID 50 illustrates the dynamic evolution of chase probability throughout the innings. The curve begins at approximately 5-7% probability when the chase commences, reflecting a challenging target relative to historical success rates. During the initial Powerplay phase from ball 120 to ball 90, the probability gradually increases to approximately 15% as the opening batsmen negotiate the new ball and establish a platform. The relatively modest probability increase despite scoring runs reflects the fact that the target remains daunting and significant work remains.





The middle phase from ball 90 to ball 60 shows substantial volatility in win probability, with fluctuations between 10% and 30%. Wicket events, marked in red on the curve, cause immediate probability drops of 5-15 percentage points. Each wicket removes a batting resource and shifts the burden to less established batsmen, directly impacting the model's assessment of chase feasibility. Partnership-building periods between wickets show gradual probability increases as runs accumulate and the required run rate moderates. The volatility in this phase reflects the delicate balance where the match could swing either direction based on a single over's events.

The death overs phase from ball 60 to ball 0 exhibits extreme volatility with probability swings of up to 20 percentage points within a few deliveries. Boundaries cause sharp upward spikes as they simultaneously reduce runs required and demonstrate the batting team's capability to score quickly. Dot balls and wickets cause equally sharp downward movements, as each scoreless delivery in the death overs significantly increases the required run rate for remaining balls. The maximum probability achieved in this match was approximately 50% around ball 20, suggesting a genuinely close contest where the outcome remained uncertain until the final overs.

The final sequence shows the probability declining to approximately 30-40% at ball 10, then dropping further as late wickets fell. The ultimate outcome was a loss for the chasing team, consistent with the probability trajectory which never sustained values above 50% except briefly. The curve demonstrates the model's capability to track match dynamics in real-time, providing a quantitative narrative of how the chase unfolded. For a match that ended in victory for the chasing team, the curve would show sustained high probabilities above 60% in the final overs as the required runs diminished to easily achievable levels.

The practical applications of these win probability curves extend across multiple stakeholders in cricket. For broadcasting and media, the curves provide engaging real-time graphics that quantify the excitement level and turning points for viewers. Commentary can reference specific probability values to contextualize the current match situation, replacing subjective assessments of pressure with data-driven metrics. Historical comparisons become possible, enabling statements like this situation at 35% win probability is similar to Match X from last season which the chasing team won. For team management and coaching staff, the curves enable post-match analysis of tactical decisions by showing which moments had the largest impact on win probability. Decisions about when to accelerate scoring, which bowlers to use, and field placement strategies can be evaluated against their measured impact on winning chances.

However, the research also reveals a critical limitation in the aggregation from ball-by-ball to match-level prediction. Despite achieving 80.37% accuracy at the ball-by-ball level, the final match outcome prediction accuracy was only 50.13%, barely exceeding random guessing. This dramatic performance gap highlights a fundamental challenge in sports analytics where accurate micro-level predictions do not automatically yield accurate macro-level outcomes. Several factors contribute to this aggregation problem.

The threshold sensitivity issue arises because match outcome prediction requires converting continuous probability estimates to binary win or loss predictions. The conventional 0.5 threshold classifies probabilities above 50% as wins and below 50% as losses. However, many matches end with probabilities near this boundary, and small prediction errors near 0.5 cause binary classification errors even though the probability estimate itself is reasonable. A match ending at 48% probability for the chasing team could easily result in either outcome, making binary prediction inherently unreliable near the decision boundary.

The last ball volatility problem stems from the dramatic shifts possible in cricket's final moments. A match situation requiring 6 runs from the final ball might be assigned a 40% win probability based on historical success rates in such situations. However, the actual outcome is deterministically binary, the batting team either hits a six and wins or fails to do so and loses. The model's 40% probability is accurate in an expected value sense across many such situations, but appears wrong for any single instance. This irreducible uncertainty cannot be eliminated through better modeling, it is fundamental to the stochastic nature of sports outcomes.

The model is optimized for ball-by-ball prediction accuracy across all game states, not specifically for final outcome prediction. The loss function during training treats all balls equally, so the model learns to predict accurately throughout the innings rather than specializing in close finish scenarios. A specialized model optimized specifically for match-level prediction might use different features, different training procedures, or ensemble multiple models to achieve better binary classification. The ball-by-ball model prioritizes calibrated probability estimates across diverse situations over forceful binary discrimination.

The absence of contextual factors that particularly matter in close finishes contributes to prediction errors. Player-specific clutch performance, the ability of certain batsmen or bowlers to perform exceptionally under extreme pressure, is not captured in the feature set. Psychological momentum and confidence factors that become decisive in tight matches are only partially proxied through the dot ball features. Venue-specific characteristics such as small boundaries or high altitude that affect scoring in the death overs are not modeled. Weather conditions including dew in night matches that makes the ball slippery for bowlers and easier to hit for batsmen are absent from the features.

**Chapter 4: Discussion and Implications**

**4.1 Theoretical Contributions and Validation**

This research makes several theoretical contributions to sports analytics and cricket analysis specifically. The introduction of temporally weighted features for pressure quantification operationalizes a concept that has been discussed qualitatively in cricket commentary for decades. The linear weighting scheme from 1.0 to 1.5 over an 18-ball window represents a tractable approach to capturing recency effects, though future research could explore exponential weighting or learned weighting functions. The validation that these features contribute 5-6% predictive power confirms that pressure effects exist and are measurable, even if their magnitude is more modest than might be expected.

The demonstration that Random Forest substantially outperforms both linear models and gradient boosting approaches for this task advances our understanding of appropriate modeling choices for cricket analytics. The 4.74 percentage point accuracy gain over logistic regression, while seemingly modest, represents a meaningful improvement when deployed at scale across thousands of predictions. The superiority of Random Forest over XGBoost is particularly interesting, as XGBoost often dominates in machine learning competitions. This finding suggests that cricket chase dynamics may be better captured through the parallel ensemble approach of Random Forest rather than the sequential error-correction approach of boosting, possibly because the feature interactions are more important than residual pattern learning.

The phase-wise analysis provides empirical validation of strategic intuitions held by players and coaches. The finding that death overs dot balls have double the negative correlation with winning compared to powerplay dots quantifies conventional wisdom and provides a data-driven foundation for strategic planning. The identification of middle overs as the phase with highest dot ball accumulation but intermediate impact suggests this is where matches are often decided through sustained pressure rather than dramatic events. These findings can inform training priorities, with teams potentially investing more in death bowling skills given their outsized impact.

The feature importance hierarchy establishes that Required Run Rate is overwhelmingly the most informative single metric for chase prediction. This finding validates the decades-long practice of focusing on RRR in cricket analysis and commentary. However, the research also demonstrates that RRR alone is insufficient, as the additional features collectively contribute 52% of predictive power. The importance of wickets fallen at 16% and runs required at 18% underscores that resource management and absolute deficit magnitude matter beyond their contribution to the rate calculation.

**4.2 Limitations and Future Research Directions**

Several limitations of this study suggest directions for future research. The single-season dataset spanning only 59 matches from IPL 2017 limits generalizability to other seasons, leagues, and formats. Model performance may vary across different years as team strategies evolve and rule changes are implemented. Training on multiple IPL seasons from 2015 through 2020 would provide approximately 500 matches and strengthen conclusions about model generalizability. Testing on other T20 leagues such as the Big Bash League or Caribbean Premier League would assess cross-league transfer learning and identify league-specific patterns.

The absence of player-specific features represents a significant gap in the modeling approach. Elite batsmen like Virat Kohli or AB de Villiers have demonstrably different scoring profiles and pressure-handling capabilities compared to lower-order batsmen. Elite bowlers like Jasprit Bumrah or Rashid Khan have unique skills particularly valuable in death overs. Incorporating player quality ratings, historical performance metrics, and current form indicators would likely improve prediction accuracy substantially. The challenge lies in avoiding overfitting to individual players and maintaining reasonable sample sizes for fair estimation.

Contextual factors absent from the current model include venue characteristics such as ground dimensions, average scores, and pitch conditions that vary across the eight IPL venues. Matches at Mumbai's Wankhede Stadium with small boundaries and flat pitches exhibit different scoring patterns than matches at Chennai's MA Chidambaram Stadium with larger boundaries and spin-friendly pitches. Weather conditions including temperature, humidity, and dew factor significantly affect ball behavior and scoring rates, particularly in evening matches where dew makes the ball slippery. The toss decision whether to bat first or chase creates different pressure dynamics, as chasing teams know their exact target while batting first teams must estimate competitive totals.

The weighting scheme for dot ball features, while theoretically motivated, uses a simple linear progression that may not be optimal. Alternative weighting functions including exponential decay, learned weights through gradient descent, or phase-specific weights could potentially improve performance. Interaction terms between weighted pressure and other features such as pressure times wickets fallen or pressure times required run rate might capture synergistic effects not accessible through additive models. More sophisticated temporal models such as Long Short-Term Memory networks or Transformer architectures could learn optimal temporal aggregation patterns directly from data rather than relying on hand-crafted weighting schemes.

The aggregation problem from ball-by-ball to match-level prediction deserves dedicated investigation. Developing specialized models optimized specifically for final outcome prediction rather than general ball-by-ball accuracy could involve techniques such as instance weighting where later-inning balls receive higher weight in the loss function, threshold optimization searching for optimal classification thresholds beyond the default 0.5, ensemble methods combining multiple models through stacking or blending, and uncertainty quantification through Bayesian approaches or conformal prediction to provide confidence intervals on match outcomes.

Causal inference methods would strengthen conclusions about dot ball impact beyond mere correlation. Instrumental variable approaches using bowler quality as an instrument for dot ball rate could establish whether dot balls cause wickets or both are driven by underlying batsman weakness. Regression discontinuity designs exploiting phase boundaries could quantify the causal effect of field restrictions on dot ball rates and outcomes. Propensity score matching could create balanced comparisons between high-pressure and low-pressure situations to isolate the causal effect of pressure on performance.

Real-time deployment considerations for production systems include latency requirements where predictions must be generated within milliseconds to support live applications, scalability to handle multiple concurrent matches during tournaments, model versioning and A/B testing to gradually deploy improved models while monitoring for performance degradation, and monitoring and alerting to detect when model predictions diverge from actual outcomes indicating potential data quality issues or distribution shift.

**Chapter 5: Conclusion**

This research successfully developed and validated a comprehensive framework for predicting cricket match outcomes during the chase phase using ball-by-ball data from the IPL 2017 season. The study introduced novel weighted features quantifying dot ball pressure through temporal weighting schemes, compared four machine learning models under rigorous evaluation protocols, and provided detailed phase-wise analysis revealing strategic patterns in how dot balls impact match outcomes across different innings phases.

The Random Forest model achieved the highest ball-by-ball prediction accuracy of 80.37%, substantially outperforming the baseline Logistic Regression model at 75.63%. This 4.74 percentage point improvement represents approximately 15,600 additional correct predictions on the test set and validates the hypothesis that ensemble methods are necessary to capture the complex non-linear relationships in cricket chase dynamics. The XGBoost and Gradient Boosting models achieved competitive performance with strong ROC-AUC scores of 0.881 and 0.873 respectively, demonstrating excellent ranking ability even where binary classification accuracy was slightly lower.

The feature importance analysis revealed that Required Run Rate dominates predictions with 48% of total importance, followed by runs required at 18% and wickets fallen at 16%. The novel weighted dot ball features contributed 5-6% importance, providing statistically significant and practically meaningful additional predictive signal particularly in close match situations. Phase-wise analysis demonstrated that death overs dot balls have the strongest negative correlation with winning at -0.42, approximately double the powerplay correlation of -0.18, providing quantitative validation of conventional strategic wisdom about the outsized importance of death bowling execution.

However, the research also uncovered a significant limitation in aggregating ball-by-ball predictions to final match outcomes. Despite 80% micro-level accuracy, the match-level prediction accuracy of only 50.13% reveals the aggregation challenge where accurate predictions at granular levels do not automatically produce accurate predictions at aggregate levels. This finding has important implications for sports analytics generally, highlighting the need to match modeling approaches and optimization objectives to the specific decision-making context and granularity required.

The practical contributions of this research include a validated framework for real-time win probability calculation that can enhance broadcasting and viewer engagement, strategic insights into phase-specific dot ball impact that can inform coaching and tactical decisions, and methodological advances in temporal feature engineering applicable to sequential pressure situations across multiple sports. The research establishes Random Forest as the preferred modeling approach for cricket chase prediction and demonstrates that incorporating pressure metrics beyond traditional counting statistics improves predictive performance.

Future research should extend this framework to multi-season datasets to validate generalizability, incorporate player-specific and contextual features to capture individual skill and situational factors, explore advanced weighting schemes including learned weights and exponential decay, develop specialized models optimized for match-level rather than ball-by-ball prediction, and investigate causal relationships between dot balls, pressure, and performance outcomes. The aggregation problem deserves dedicated investigation through ensemble methods, threshold optimization, and uncertainty quantification techniques that provide confidence intervals on predictions.

This study represents a significant advance in cricket analytics, demonstrating that the subjective concept of pressure can be quantified through data-driven methods and that incorporating these pressure metrics improves match outcome prediction. The framework and findings provide value for multiple stakeholders including teams seeking competitive advantages, broadcasters enhancing viewer experiences, and the cricket analytics community advancing methodological sophistication. As cricket continues its evolution toward data-driven decision making, such research contributions help bridge the gap between traditional intuition and quantitative analysis, enabling more informed strategic choices while preserving the excitement and unpredictability that make sport compelling.

**References**

Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19-31.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189-1232.

Jordet, G., & Hartman, E. (2008). Avoidance motivation and choking under pressure in soccer penalty shootouts. Journal of Sport and Exercise Psychology, 30(4), 450-457.

Kampakis, S., & Thomas, B. (2015). Using machine learning to predict the outcome of English county twenty over cricket matches. arXiv preprint arXiv:1511.05837.

Manage, A.B.W., & Scariano, S.M. (2013). An introductory application of principal components to cricket data. Journal of Statistics Education, 21(3).

Pathak, N., & Wadhwa, H. (2016). Applications of modern classification techniques to predict the outcome of ODI cricket. Procedia Computer Science, 87, 55-60.

Shah, P., & Shah, M. (2021). Feature engineering and ensemble methods for IPL match prediction. International Journal of Sports Science & Coaching, 16(4), 891-903.

Indian Premier League. (2017). IPL 2017 Season Statistics. Retrieved from official IPL archives.