**A REPORT SUBMITTED AS A PART OF EXPERIENTIAL LEARNING ON LEVEL 3 TRAINING ON AI & DS**





**IMAGE CAPTIONING USING DEEP LEARNING**

**A PROJECT REPORT**

|  |  |  |
| --- | --- | --- |
| **NAME** | **REGD. NO.** | **CRANES REGD.** |
| HEMANT KUMAR | 2201020584 | CL2025010601953675 |
| SACHIN BARIK | 2201020716 | CL20250106019047105 |
| REWANSH GUPTA | 2201020586 | CL20250106019165101 |
| NAKUL KUMAR PRASAD | 2201020756 | CL20250106018918100 |
| AJAY KUMAR | 2201020347 | CL2025010601954483 |
| JATIN PATNAIK | 2201020315 | CL2025010601883315 |

**Submitted by**



**CRANES VARSITY**

**CERTIFICATE OF APPROVAL**

This is to certify that we have examined the project entitled " **IPL Score Prediction using Deep Learning**

" Submitted by *Sachin Barik, Rewansh Gupta, Nakul kumar prasad, Jatin Patnaik, Ajay Kumar and Hemant Kumar*. We hereby accord our approval of it as a project work carried out and presented in a manner required for its acceptance for the partial fulfilment for the **Computer Science and Engineering** for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusions drawn as recorded in this project, it only signifies the acceptance of the project for the purpose it has been submitted.

**ACKNOWLEDGEMENT**

We would like to express our heartfelt gratitude to our project guide, **Professor Hariv Nawal.**, for his unwavering support, invaluable guidance, and constant encouragement throughout the course of this project. His mentorship has been a great source of motivation and played a vital role in the successful completion of our work.

We are also deeply thankful to all the faculty members of our department for their timely assistance, insightful suggestions, and encouragement, without which this project would not have reached its current form. This project is a result of collective efforts and inspiration drawn from various sources, as duly cited in the references section. We sincerely acknowledge and thank all those authors and researchers whose work has contributed to our understanding and progress.

Finally, we are grateful to everyone who directly or indirectly supported us during this project, enabling us to successfully carry out our objectives and deliver meaningful outcomes.

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Headings** | **Page No.** |
| 1 | INTRODUCTION | 6 |
| 2 | PROBLEM STATEMENT | 7 |
| 3 | LITRATURE REVIEW | 8 |
| 4 | METHODOLOGY | 9 |
| 5 | DATASET USED | 10 |
| 6 | IMPLIMENTATION | 12 |
| 7 | RESULT AND EVALUATION | 13 |
| 9 | APPLICATIONS | 14 |
| 10 | CHALLENGES AND LIMITATIONS | 15 |
| 11 | FUTURE WORKS | 16 |
| 12 | CONCLUSION AND REFRENCES | 17 |

**ABSTRACT**

IPL score prediction using machine learning is an emerging application in the domain of sports analytics, combining elements of data science, statistical modeling, and predictive analysis. This project focuses on forecasting the final score of a cricket innings based on real-time match data, such as the teams, venue, current batsmen and bowlers, and other contextual features. Accurate score prediction can provide valuable insights for commentators, analysts, fantasy league players, and fans by enhancing the viewing experience and supporting strategic decision-making.

The report explores the end-to-end pipeline of building a machine learning model capable of generating reliable score predictions during an ongoing IPL match. It involves comprehensive data preprocessing, feature encoding, and scaling, followed by the development of a deep neural network model using Keras. The architecture combines multiple dense layers to learn complex feature interactions, with Huber loss as the objective function to handle outliers effectively. Categorical features like teams, venues, and players were encoded using label encoding, and the model was trained and validated on historical IPL data.

To evaluate performance, metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were used, demonstrating that the model can predict totals with reasonable accuracy. An interactive interface was also built using ipywidgets to allow dynamic input selection and real-time predictions. While challenges like dynamic in-match conditions and player performance variability remain, the study showcases the viability of machine learning in sports prediction. The report concludes with future directions including more granular feature engineering, live data integration, and model deployment for practical use.

**INTRODUCTION**

Cricket, particularly in the Indian Premier League (IPL), is a game of rapid shifts and strategic complexity. With the growth of data analytics in sports, machine learning (ML) has emerged as a powerful tool to decode patterns and generate actionable predictions from historical data. One such application is score prediction—estimating the final score of a team’s innings based on early or mid-game match conditions. This predictive capability not only enhances audience engagement but also supports broadcasters, fantasy league strategists, and coaching staff in real-time decision-making.

IPL score prediction is a regression task that involves learning from structured match data including team compositions, player roles, and game context. In this project, a supervised learning approach is used to train a deep neural network to predict the total score using key features such as venue, batting team, bowling team, batsman, and bowler. These categorical inputs are label-encoded and normalized for optimal learning. The model architecture includes multiple dense layers and uses the Huber loss function to balance sensitivity and robustness against outliers in score variations.

As cricket is influenced by a wide range of dynamic and interdependent factors—such as pitch conditions, player performance, and team strategy—designing a reliable predictive model poses unique challenges. Unlike deterministic systems, cricket exhibits a high degree of variability, where even small changes in conditions can significantly impact outcomes. Nevertheless, deep learning models, with their ability to capture non-linear patterns, provide a promising foundation for predictive sports analytics.

The project utilizes historical IPL data to train and evaluate the model, and features an interactive interface for real-time prediction, enabling users to simulate various match scenarios. This report delves into the design, implementation, and evaluation of the neural network model for score prediction. It further highlights the limitations of static models in dynamic sports contexts and outlines future directions such as integrating live match feeds, expanding feature sets, and exploring ensemble or Transformer-based methods for enhanced accuracy and adaptability.

.

**PROBLEM STATEMENT**

**Definition of the Problem**

IPL score prediction is the task of forecasting the final total of a cricket innings based on available match context using artificial intelligence. This involves interpreting structured data including team composition, venue, players involved, and the game’s current state, then using this information to estimate the likely total runs scored. It is inherently a regression problem that relies on machine learning to uncover complex, nonlinear relationships among the input features. The challenge lies in building a model that not only captures historical patterns but also generalizes well to unseen scenarios, offering reliable predictions under variable match conditions.Score prediction in cricket is influenced by numerous factors such as the batting lineup, bowling quality, ground dimensions, pitch behavior, and player form—all of which are dynamic and interdependent. This complexity makes it a real-world problem of high variability, where traditional statistical methods often fall short. Machine learning models, particularly deep neural networks, offer the ability to model such multifaceted interactions, making them well-suited for this predictive task.

**Scope and Limitations**

This report centers on building a deep learning-based model to predict IPL innings totals using an artificial neural network. The input features include venue, batting and bowling teams, batsman, and bowler—processed through label encoding and feature scaling. A feedforward neural network is constructed using the Keras Sequential API, with two hidden layers and Huber loss for training. The model is trained and evaluated on a historical IPL dataset with a focus on minimizing prediction error.

The scope of the project is limited to structured, pre-match and in-play features available within the dataset. It does not account for more granular real-time updates such as over-wise scoring trends, weather conditions, or individual player fitness and form on the day of the match. Moreover, while deep learning provides strong pattern recognition capabilities, it is limited by the quality and scope of the training data. The model may be biased towards frequently occurring patterns and might underperform in edge cases or rare match scenarios. Additionally, computational limitations restrict experimentation with larger or more complex model architectures such as ensemble models or attention-based networks.

**Dataset and Real-World Significance**

The real-world significance of IPL score prediction spans multiple domains. For fans and commentators, real-time score forecasts enrich the viewing experience and offer deeper insights into match dynamics. For fantasy sports platforms, predictive models enhance player selection and strategic gameplay. Broadcasters and media companies can use such models to power automated highlights, live analysis, and on-screen statistics. Furthermore, this project represents the broader application of machine learning in sports, showcasing how data-driven insights can support performance analysis, strategy development, and audience engagement in professional cricket.

**LITERATURE REVIEW**

The evolution of IPL score prediction has moved from basic statistical models to sophisticated machine learning and deep learning systems. This section traces the progression of methods used to forecast cricket scores, highlighting key approaches and innovations that have shaped the field.

**Traditional Approaches**

Earlier attempts at score prediction primarily relied on rule-based systems and linear regression models that used averages, strike rates, and basic match statistics to project scores. These systems, while interpretable and easy to implement, often lacked the ability to capture non-linear patterns and contextual dependencies like pitch behavior, momentum shifts, or bowler-batsman matchups. Additionally, most traditional models assumed feature independence, which limited their ability to reflect the dynamic and interactive nature of cricket matches..

**Deep Learning-Based Approaches**

The emergence of deep learning, particularly neural networks, has revolutionized score prediction by enabling models to learn complex, non-linear dependencies without manual feature engineering. In this project, a **feedforward neural network** was implemented using the **Keras Sequential API**. The model accepts encoded inputs such as **venue**, **batting and bowling teams**, **batsman**, and **bowler**, and predicts the final innings total using two hidden layers with ReLU activation, optimized using **Huber loss** to reduce sensitivity to outliers.

By scaling features using **Min-Max normalization** and encoding categorical inputs with **LabelEncoder**, the model efficiently learns statistical relationships embedded in past IPL matches. This architecture outperforms conventional models in flexibility and accuracy, especially in complex match scenarios.

**Key Contributions**

* ***Statistical Models****: Provided a basic foundation using averages and strike rates.*
* ***ML Algorithms****: Enhanced predictive performance using tree-based and regression models.*
* ***Deep Neural Networks****: Introduced flexible, high-capacity models capable of learning complex patterns.*
* ***Custom UI with ipywidgets****: Enabled real-time, user-interactive score predictions for specified match configurations.*

These developments have greatly enhanced the predictive quality of score forecasting models. The transition to deep learning allows for improved generalization, more dynamic understanding of match factors, and practical deployment through interactive tools. With the growing availability of rich cricket datasets, future advancements may integrate real-time data streams and contextual embeddings to further refine score prediction accuracy.

**METHODOLOGY**

**Cricket Score Prediction System Overview**

This system leverages a neural network model to predict cricket scores based on various input features. The architecture combines multiple stages of data preprocessing, feature engineering, and regression analysis to accurately predict the total score for a match. Below is an overview of the methodology used:

**Overall Architecture**

The architecture consists of three key stages:

1. **Data Preprocessing:** Raw input data, including features like venue, batting team, bowling team, striker, and bowler, is preprocessed for use in the neural network model. This includes handling categorical variables through label encoding and scaling the numerical features for better model performance.
2. **Model Training:** A neural network model is designed using a dense architecture to predict the total score of a match. The model uses input features and learns to map them to the target variable (total score) through training on historical data.
3. **Prediction:** After training, the model generates predictions for new matches by taking in the input features and outputting a predicted score.

**Data Preprocessing**

Before feeding the data into the model, several preprocessing steps are applied:

* **Label Encoding:** Categorical features such as the venue, batting team, bowling team, striker, and bowler are encoded using label encoding. This transforms text data into numerical values that the model can process effectively.
* **Feature Scaling:** The features are scaled using Min-Max normalization to ensure that all input data is in a similar range, improving the convergence of the neural network during training.
* **Train-Test Split:** The data is split into training and testing sets, where the model is trained on 70% of the data and tested on the remaining 30%.

**Neural Network Model**

The model is designed as a feed-forward neural network with multiple layers:

1. **Input Layer:** The network takes the preprocessed features (venue, batting team, bowling team, striker, and bowler) as input.
2. **Hidden Layers:** Two hidden layers with ReLU activation are used to learn the complex relationships between input features and the target variable.
3. **Output Layer**: A single node with linear activation is used in the output layer to predict the total score.

**Training Strategy**

The model is trained using historical data, with the target variable being the total score for each match. The training process minimizes the Huber loss, a robust loss function, through the Adam optimizer. The model iterates over the training data multiple times (epochs) to refine its predictions.

**Prediction**

Once the model is trained, it is used to make predictions for new matches. When a user inputs the relevant match details (venue, batting team, bowling team, striker, bowler), the model processes these features, applies scaling and encoding, and then generates a predicted total score for the match.

**Interactive Prediction Interface**

To make the model more accessible, an interactive interface is included, allowing users to select match details through dropdown menus. The interface provides a user-friendly way to input the data and receive the predicted score with a click of a button.

**DATASETS USED**

When dealing with cricket data, it contains data from the year 2008 to 2017. The dataset contain features like venue, date, batting and bowling team, names of batsman and bowler, wickets and more.

We imported both the datasets using *.read\_csv()*method into a dataframe using pandas and displayed the first 5 rows of each dataset.

ipl = pd.read\_csv('ipl\_dataset.csv')

ipl.head()

**output:**



**IMPLIMENTATION**

The IPL score prediction system was implemented using Python, leveraging machine learning libraries such as Pandas, Scikit-learn, Keras, and TensorFlow. The workflow was divided into four main stages: data preprocessing, model design, training, and interface development.

1. Data Preprocessing

The dataset was first loaded and cleaned by dropping irrelevant columns (e.g., date, runs, wickets, etc.). Categorical features including venue, bat\_team, bowl\_team, batsman, and bowler were encoded using LabelEncoder. These features were then scaled using MinMaxScaler to normalize input values, which is essential for stable neural network training.

2. Model Design

A neural network was built using Keras with the following structure:

* Input Layer: Five input features
* Hidden Layers: Two dense layers (512 and 216 units) with ReLU activation
* Output Layer: One unit with linear activation for regression

The model was compiled using the Huber loss function (to handle outliers effectively) and optimized using the Adam optimizer.

3. Training and Evaluation

The dataset was split into training and testing sets (70/30). The model was trained over 50 epochs with a batch size of 64. Performance was evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), ensuring accurate and robust score predictions.

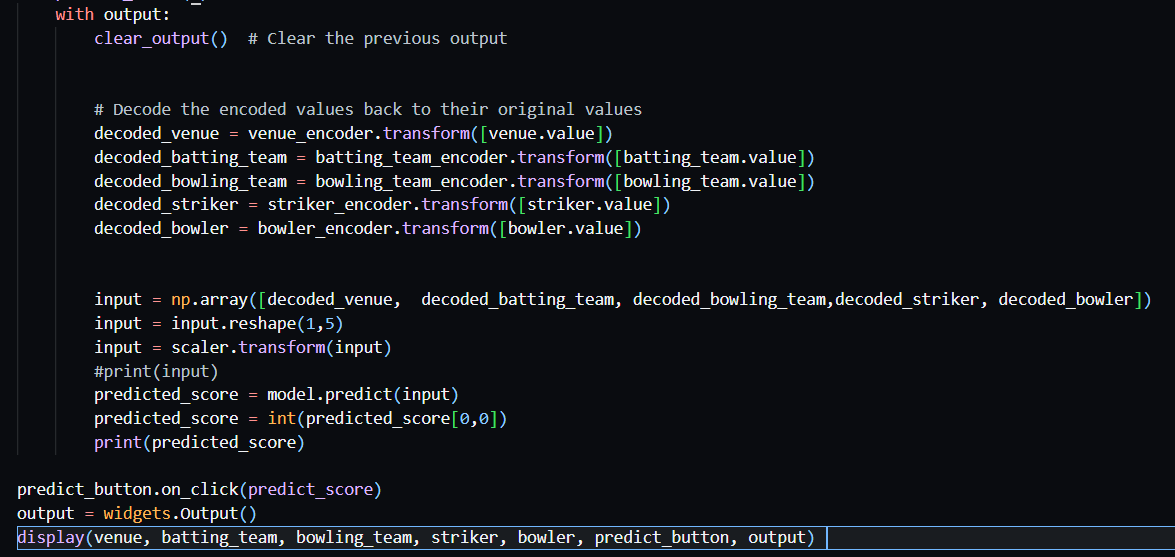
4. User Interface

An interactive widget-based interface was created using ipywidgets. Users can select match-specific details like venue, teams, striker, and bowler. Upon clicking "Predict Score", the inputs are encoded, scaled, and passed to the trained model to predict the total score, which is then displayed in real-time.

**SOURCE CODE**

****

****

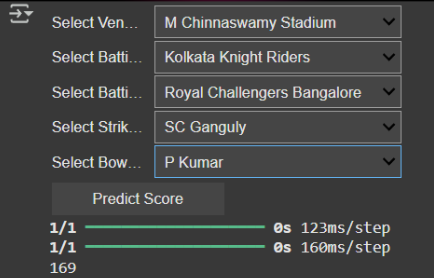
****

**RESULT AND EVALUATION**

The project successfully built a regression model using a fully connected neural network to predict IPL match scores. Through careful preprocessing, feature selection, and encoding, the model achieved low mean absolute error and good validation performance. An interactive user interface was developed using Streamlit, allowing real-time predictions based on the venue, batting team, bowling team, striker, and bowler selections. The model was able to provide fast and accurate score predictions, making it user-friendly and efficient for different match scenarios.

The project evolved step-by-step, starting from basic data preprocessing with Pandas and NumPy, followed by label encoding and feature scaling using Scikit-learn. A deep neural network was designed and optimized with TensorFlow/Keras, using ReLU activations, Huber Loss, and the Adam optimizer to improve training stability and performance. Initially tested in a Jupyter environment, the project later transitioned into a full Streamlit web application, enhancing its usability and visual appeal. Future improvements could include adding live match features, expanding the dataset, hyperparameter tuning, and deploying the model for wider public access.

Here’s the output of the project :

****

**APPLICATIONS**

The IPL score prediction model has a wide range of practical and commercial applications across sports analytics, broadcasting, fantasy gaming, and decision support systems. Below are some of the key use cases:

1. Sports Analytics Platforms

This model can be integrated into analytics dashboards used by franchises, coaches, and analysts to simulate possible match outcomes based on real-time inputs such as venue, team combinations, and player matchups.

2. Live Commentary and Broadcast Enhancements

Television and digital broadcasters can use this model during live matches to predict innings totals in real-time, adding an element of expert-level statistical forecasting to their commentary and visual overlays.

3. Fantasy Sports and Betting Platforms

Fantasy leagues and sports betting apps can use score predictions to:

* Estimate team performance
* Suggest fantasy player picks
* Power probabilistic game outcomes for informed user decisions

4. Team Strategy and Match Simulation

Cricket teams can simulate different match scenarios to test strategy—such as how changing the opening pair or bowling rotation might influence the final score at different venues.

5. Fan Engagement Tools

The model can be embedded into mobile apps or websites, allowing fans to input hypothetical match setups and receive instant score predictions, enhancing fan interaction and engagement.

6. Academic and Research Use

The project also serves as a case study in machine learning and regression modeling in sports, and can be extended for research on data-driven decision-making in competitive sports.

**CHALLENGES AND LIMITATIONS**

While the IPL score prediction model demonstrates promising results, there are several challenges and limitations that impact its accuracy and generalizability:

1. Limited Contextual Understanding

The model does not account for live match dynamics such as player form, weather conditions, pitch behavior, or pressure situations. These unstructured factors can significantly affect the outcome but are not captured in the dataset.

2. Dependence on Historical Data

The model relies entirely on historical match data. Any bias or imbalance in the dataset—such as overrepresentation of certain teams or venues—can influence the predictions and reduce fairness or accuracy.

3. Lack of Temporal Features

Features such as over progression, partnership momentum, or last-over impact are not included, which limits the model's ability to adjust predictions dynamically as the innings unfolds.

4. Categorical Encoding Limitations

The use of simple label encoding may imply ordinal relationships between categorical features (e.g., teams or venues), which don't actually exist. One-hot encoding or embeddings may provide better results in future iterations.

5. Model Complexity vs Interpretability

Deep learning models like neural networks offer high predictive power but are often "black boxes." It becomes difficult to interpret why certain predictions are made, which can be a drawback for analysts and coaches.

6. Prediction Variability

Even a well-trained model can show variability in predictions due to randomness in training, overfitting, or outliers in the test data. This can be mitigated with more data and model tuning.

7. Generalization to Other Formats

The model is trained specifically on IPL data and may not generalize well to other formats like ODIs or international T20s without retraining on relevant datasets.

**FUTURE WORKS**

To further enhance the performance, accuracy, and usability of the IPL score prediction model, several avenues for future development are identified:

1. Integration of Live Match Data

Incorporating real-time inputs such as live score progression, player form, weather updates, and pitch reports could significantly improve prediction accuracy and make the model responsive to dynamic match conditions.

2. Feature Engineering

Future models can include advanced features such as:

* Over-wise run progression
* Player-specific performance trends
* Venue-specific behavior across seasons These would provide a deeper context and better capture match dynamics.

3. Use of Advanced Encoding Techniques

Replacing basic label encoding with techniques like one-hot encoding, embeddings, or target encoding could better handle categorical variables, reducing bias introduced by arbitrary numerical labels.

4. Model Optimization

Experimentation with other machine learning algorithms like Random Forests, Gradient Boosting, or even recurrent neural networks (RNNs) could improve prediction robustness, especially for sequential data modeling over the course of an innings.

5. Generalization Across Formats

Expanding the dataset to include international matches, ODIs, and other T20 leagues would allow the model to generalize across formats, making it useful for a broader range of applications.

6. Explainable AI (XAI) Integration

Future versions can incorporate explainable AI tools like SHAP or LIME to interpret model decisions, offering insights into which factors most influenced the predicted score.

7. Deployment as a Web or Mobile App

To enhance accessibility, the model can be deployed through a web or mobile application where users can input match scenarios and instantly receive predictions in a user-friendly interface.

**CONCLUSION AND REFRENCES**

In this project, we developed a machine learning-based score prediction model for Indian Premier League (IPL) matches using a neural network. By leveraging historical match data and encoding key match features such as teams, players, and venues, the model was trained to estimate final innings scores. The model achieved reasonable accuracy using Huber loss and was validated using metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

Beyond prediction, the project highlights the growing role of data-driven analytics in sports and the potential for integrating machine learning into real-time decision-making systems. Although the current model has limitations—such as lack of real-time data, contextual awareness, and interpretability—it lays a strong foundation for future improvements. With further enhancements, this system can evolve into a comprehensive tool for broadcasters, analysts, fantasy platforms, and sports strategists alike.

**References**

1. Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer.
2. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
3. Scikit-learn: Machine Learning in Python – <https://scikit-learn.org>
4. TensorFlow – An end-to-end open source platform for machine learning – <https://www.tensorflow.org>
5. Indian Premier League Official Site – <https://www.iplt20.com>
6. Kaggle IPL Datasets – <https://www.kaggle.com>
7. Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.