BRAC UNIVERSITY

Inspiring Excellence

CSE422: Artificial Intelligence

Project Report

Project topic:

T20 Cricket Match Score Prediction

Submitted by

Group no: 20

Section: 19

Student Name	Student ID
Sakib Raihan	22299180

Table of Contents

1. Introduction

This project explores the factors influencing T20 Match Score by leveraging machine learning techniques. This dataset contains information from T20 cricket matches, including features like overs played, wickets lost, run rate, pitch and weather conditions, and opponent strength. The objective of this dataset is to help analysts, coaches, or automated systems forecast total runs based on early match performance (e.g., overs played, wickets lost) and also to support decision-making during live matches — such as adjusting strategies based on predicted outcomes

The project aims to address a foundation for machine learning models in sports analytics, particularly for fast-paced formats like T20 where rapid changes impact outcomes significantly. By studying these patterns and relationships, it seeks to provide insights that could help the team to post a good score. The motivation stems from the desire to better understand how modifiable factors can improve the performance and the score of the team.

2. Dataset Description

● **Features:** There are 7 features. 8 columns (7 Features, 1 Target)

● **Classification/Regression:** Regression problem (T20 Match Score is measured in numbers (e.g., 150 runs) It's a continuous numerical value. There are no discrete classes or categories to predict. As we know Regression problems predict continuous numerical values so we are considering this as a regression problem)

● **Data points:** 1500 rows

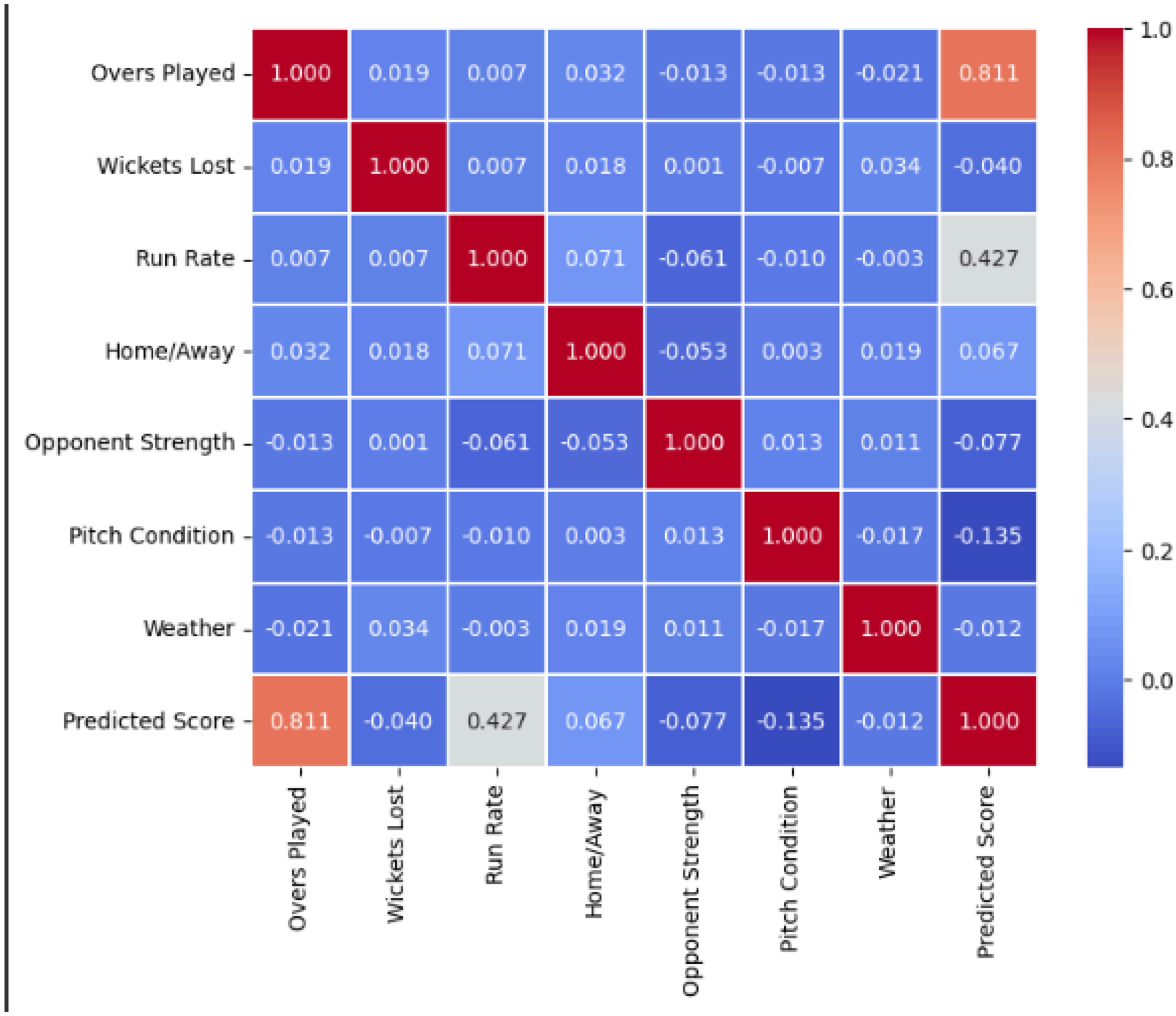
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Match ID           1500 non-null   int64
 1   Overs Played       1500 non-null   int64
 2   Wickets Lost       1500 non-null   int64
 3   Run Rate           1500 non-null   float64
 4   Home/Away          1500 non-null   object
 5   Opponent Strength  1500 non-null   int64
 6   Pitch Condition    1500 non-null   object
 7   Weather            1500 non-null   object
 8   Predicted Score    1500 non-null   int64
dtypes: float64(1), int64(5), object(3)
memory usage: 105.6+ KB
```

● **Feature types:**

o **Quantitative:** Match ID, Overs Played, Wickets Lost, Run Rate, Opponent Strength

o **Categorical:** Home/Away, Pitch Condition, Weather

● **Correlation:** We have applied joint plot, Pair plot, histogram, and boxplot to understand the relation between Predicted Score for every single feature, and we applied heatmap to understand the correlations between the features

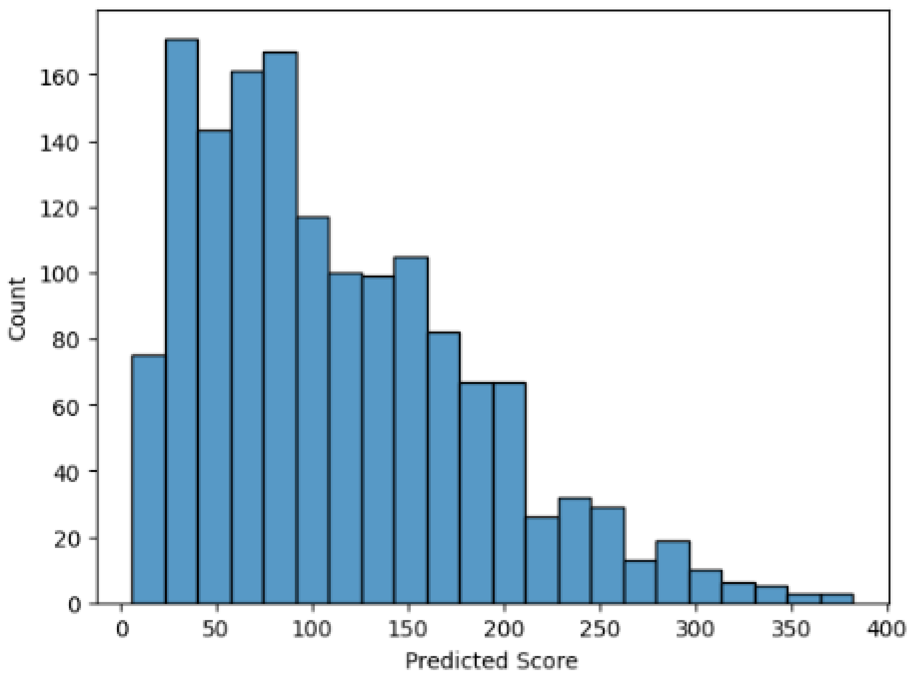


<u>Correlation between the features</u>

The color scale helps interpret the relationships:

- Light/Dark Red color (towards 1.0): Strong positive correlation
- Blue color (towards -1.0): Strong negative correlation
- Neutral/White colors (around 0): Little to no correlation

● **Imbalance Dataset:** This visual representation helps us understand data distribution and identify unusual patterns, allowing us to understand the data better. As this is a regression problem we don't have any output features. Here we are attaching the Predicted Score histogram. The graph is not normally distributed.



3. Dataset Preprocessing:

● **Duplicate values:** No duplicate values in our dataset

```
#Duplicate values
t20_score.duplicated().sum()
```

```
np.int64(0)
```

● **Categorical values:** For fitting the data into a model we need all the columns in numerical form as some machine learning models cannot properly learn from categorical values. converting object columns into numerical is called encoding. We had 3 categorical columns(Home/Away, Pitch Condition and Weather) and applied label encoder.

● **Feature selection:** Using heatmap we have found out that many columns are correlated among themselves by much margin so we didn't drop any of the features.

4. Feature scaling:

Our dataset has features with very different ranges:

● Overs Played: Between 0-20
● Wickets Lost: Between 0-10

● Some algorithms might not work properly without scaling. So, we have used the Standard Scaler to avoid this

5. Dataset splitting:

To train the model data splitting was done randomly as this is a regression problem. We have split the dataset into Train set (70%) and Test set (30%).

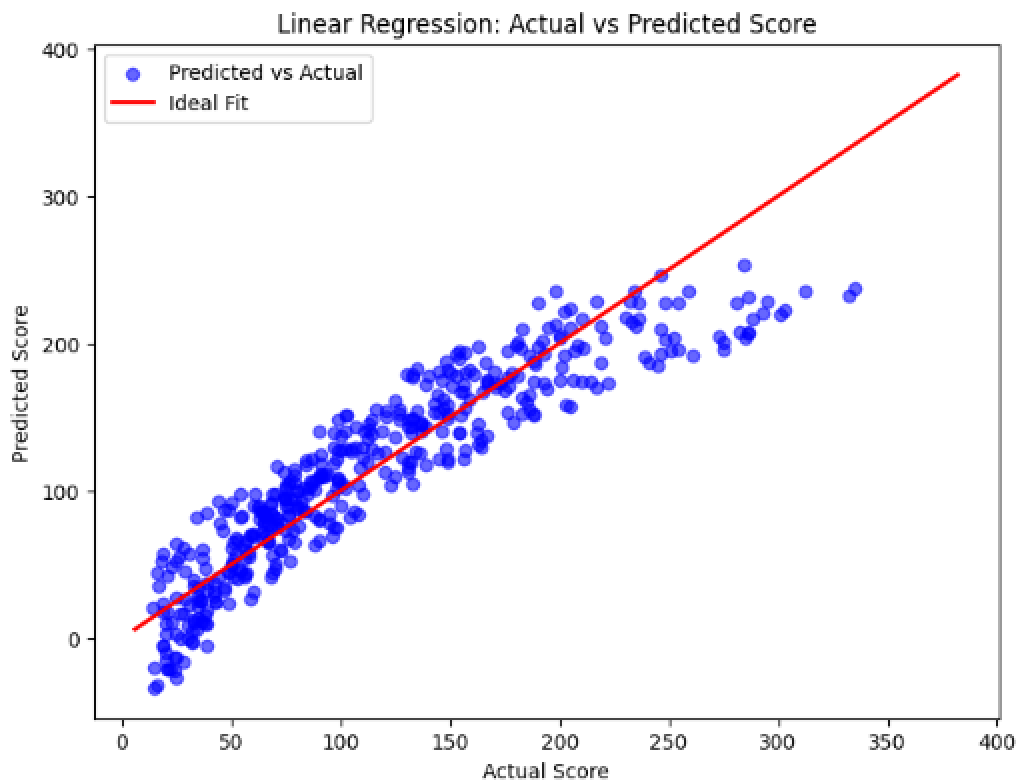
```
print(xTrain.shape)
print(xTest.shape)
print(yTrain.shape)
print(yTest.shape)
```

```
(1050, 7)
(450, 7)
(1050,)
(450,)
```

6. Model Training and Testing:

● Linear regression:

Created a scatter plot that visually compares the model's predictions (`yPrediction_lr`) with the true values (`yTest`). This is a way to quickly assess how well the model's predictions align with reality.

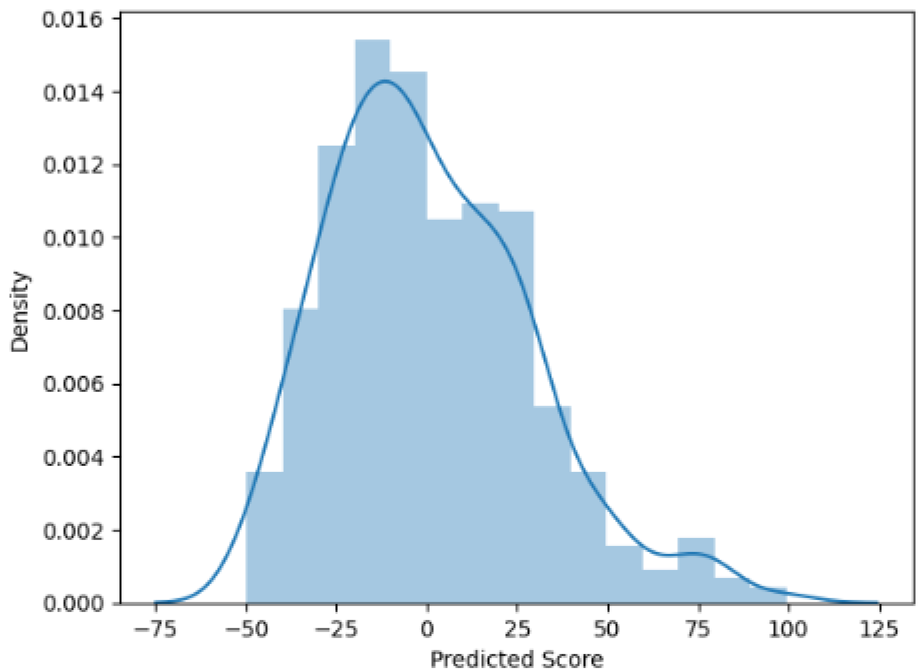


Then we calculated the Evaluation metrics:
MAE: Average absolute difference between predicted and actual values
MSE: Average squared difference (penalizes larger errors more)
RMSE: The square root of MSE
R-squared: How well the model explains the variance in data (1 is perfect)

```
MAE : 22.466432584412882
MSE : 798.042378226782
RMSE : 28.249643860176043
R-squared : 0.8435441294615332
```

The numerical metrics help us to get a more precise understanding of how the model is performing, which lets us know how well this regression model is performing.

Analyzing model errors: Residual distribution



Then we measured the coefficients of the linear regression model to see how each feature impacts T20 Match Score. The bigger the coefficient, the more that feature impacts the T20 Match Score.

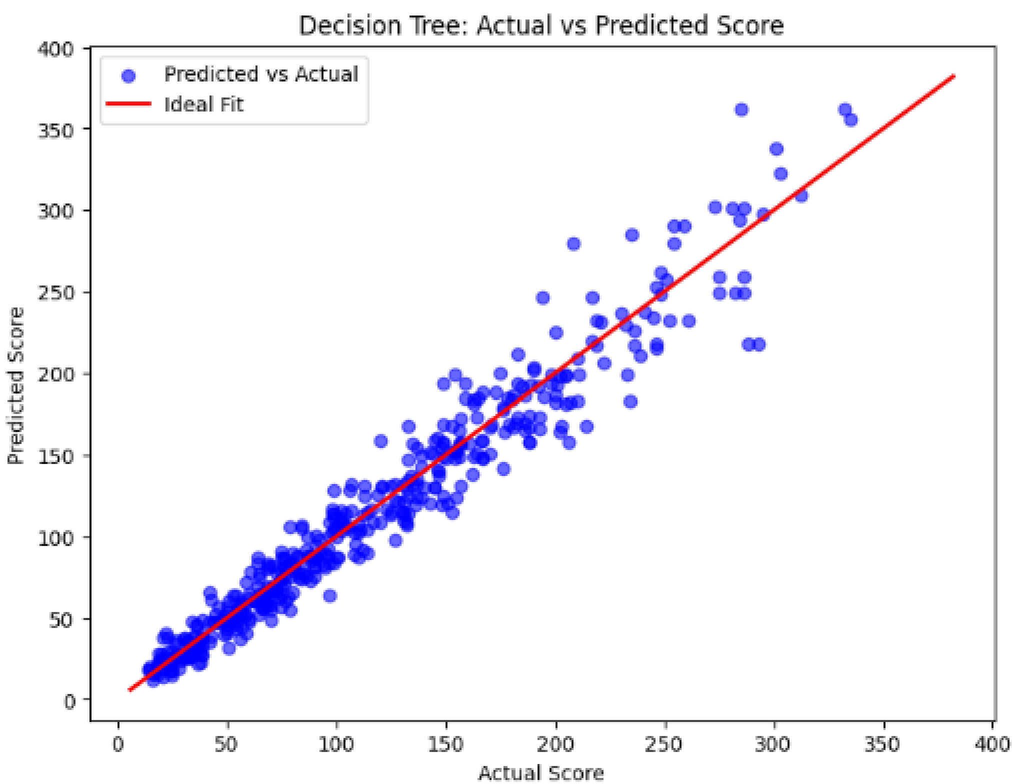
	Coeffecient
Overs Played	57.810463
Wickets Lost	-4.840925
Run Rate	29.966983
Home/Away	0.321218
Opponent Strength	-3.218139
Pitch Condition	-8.794662
Weather	0.550479

● Decision Tree Regressor Model:

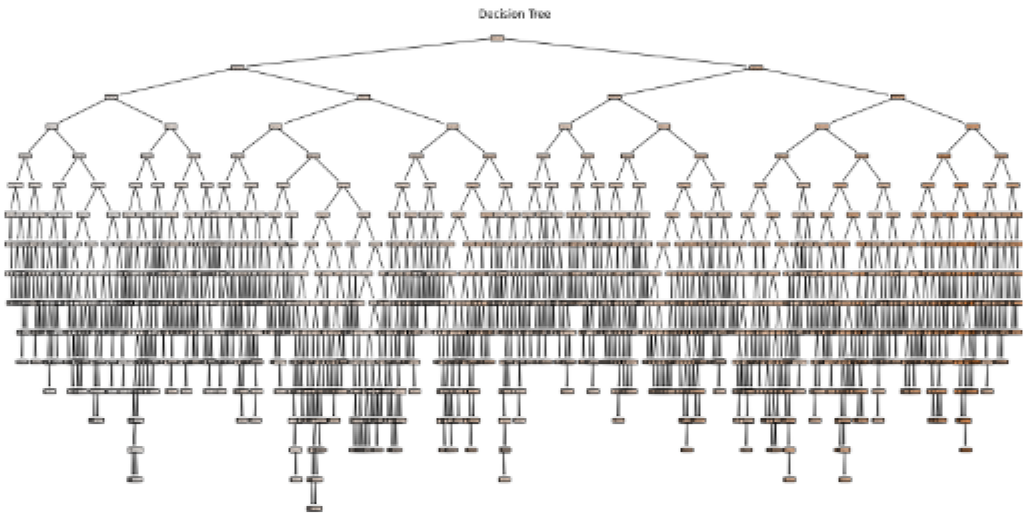
We calculated the common error metrics (MAE, MSE, RMSE, R-squared) to measure how accurate the model's predictions are compared to actual values

```
MAE : 11.71111111111111
MSE : 269.6355555555556
RMSE : 16.42058328913914
R-squared : 0.9471380634368023
```

Created a visualization comparing actual vs predicted score, with the red line showing perfect predictions and scattered points showing actual model performance



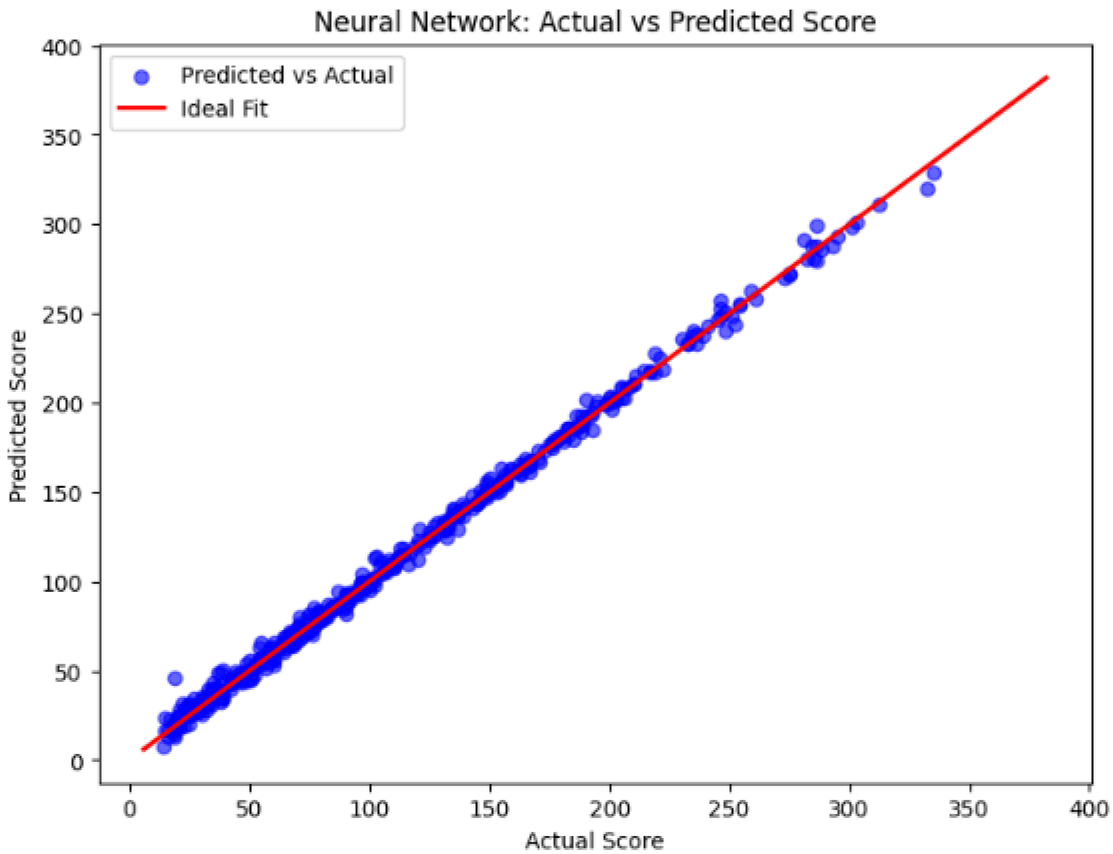
This is a visualization of the decision tree, illustrating the hierarchy of decision nodes and splits based on feature thresholds that guide predictions.



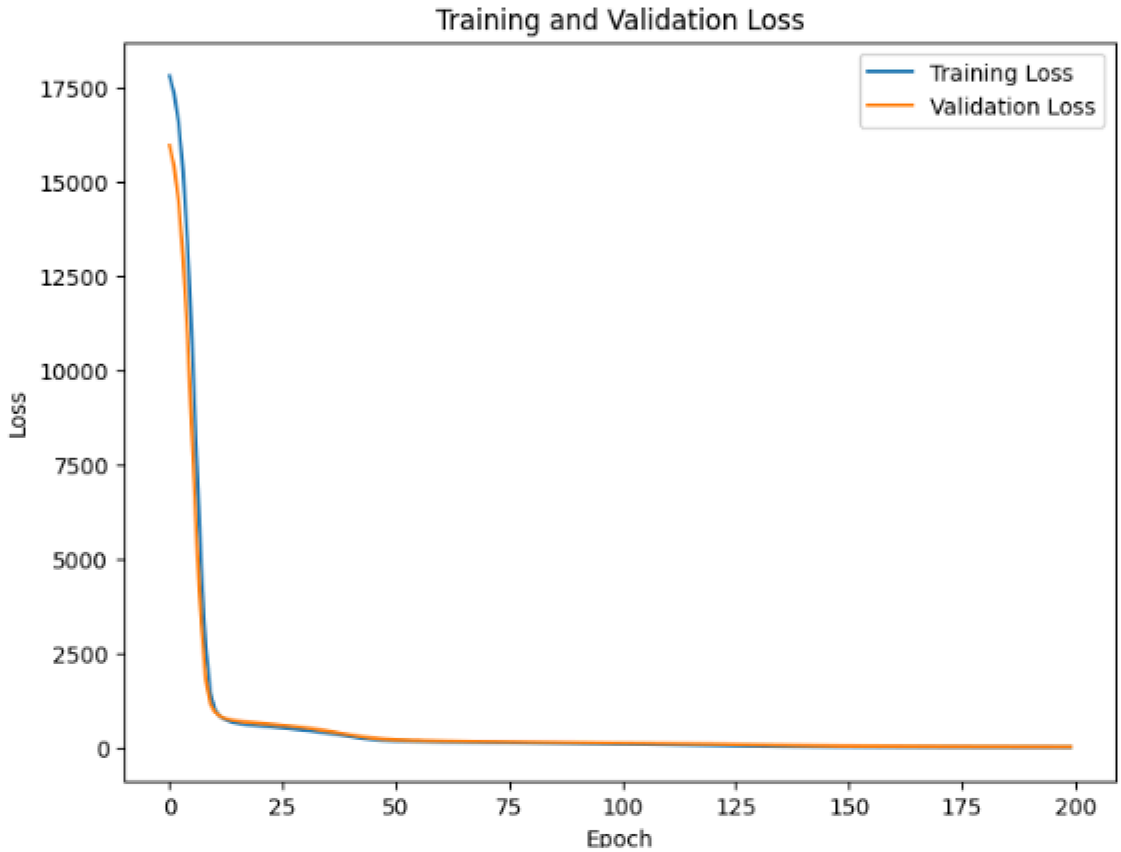
● Neural Networks:

We created a Neural Network with 3 layers (64 neurons, 32 neurons, and 1 output neuron) for predicting T20 Match Score Trained the model for 200 epochs using the Adam optimizer and mean squared error as the loss function Evaluated model performance using standard metrics (MAE, MSE, RMSE, R-squared) and prints the results. And then Created a visualization comparing actual vs predicted T20 Match Score values.

```
MAE : 2.898503303527832
MSE : 15.979104995727539
RMSE : 3.9973872711719514
R-squared : 0.996867299079895
```

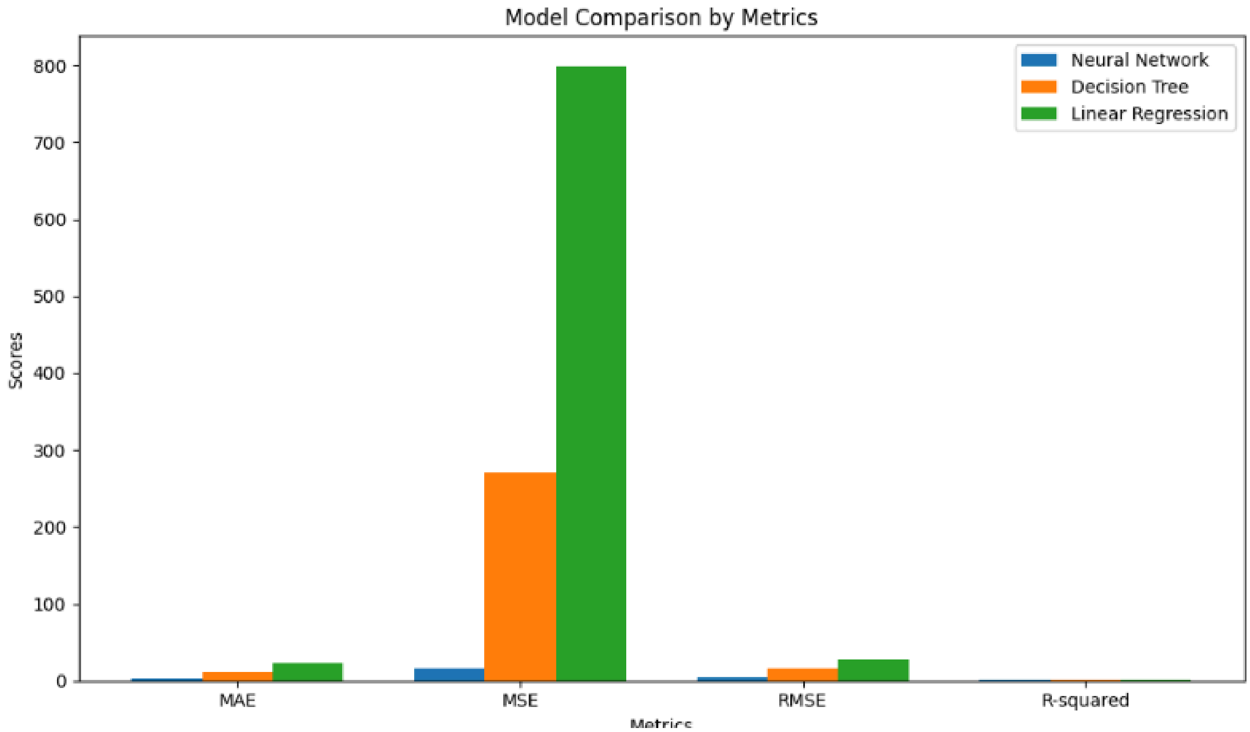


Created a plot showing how the model's loss (error) changes during training over time. Which helps us to identify if the model is overfitting.



7. Comparison Analysis:

We compared three machine learning models (Neural Network, Decision Tree, and Linear Regression) across four metrics: MAE, MSE, RMSE, and R-squared. Which gives us:



Based on the metrics shown:

- Linear Regression has higher error rates (MAE, MSE, RMSE) compared to others
- All models have R-squared values greater than 0.84
- Neural Network and Decision Tree show comparable performance with slightly lower error rates than Linear Regression.

Model Selection:

We have to go with Neural Networks since it:

- Has the lowest error rates (MAE, MSE, RMSE)
- Has the highest R-squared value
- Is simpler to implement and interpret
- Would require less computational resources

8. Conclusion:

In this project, we utilized the “T20 Cricket Match Score” dataset to develop a predictive model for T20 Match Score. The process involved several key steps: data preprocessing, feature scaling, and dataset splitting. Subsequently, we applied three Machine Learning models—Linear Regression, Decision Tree Regressor, and Neural Networks—to predict T20 Match Score.

The performance of these models was evaluated and compared based on relevant metrics. Following the analysis, Neural Networks was identified as the most suitable model for this problem, demonstrating superior performance relative to the other approaches.