# DS - MINI PROJECT

**CRICKET SCORE PREDICTION**

**Problem Statement:** To predict the total score of a cricket team in a match based on real-time and historical data, such as player performance, match conditions, and contextual factors.

**Select suitable dataset**:

1. **Kaggle Cricket Datasets**

* **IPL Ball-by-Ball Data:** Provides detailed ball-by-ball data for IPL matches, including runs scored, wickets taken, and match details. It can be aggregated to create features like overs, wickets, run\_rate, and total\_runs.

2. **Cricinfo StatsGuru**

* Extract historical match data from platforms like Cricinfo's StatsGuru. Use Python libraries like BeautifulSoup to scrape data if allowed.

3. **Open Cricket Archives**

* Some open cricket datasets, such as **CricSheet**, provide CSV files of match events. CricSheet Download.

4. **Synthetic Data**

* If no suitable dataset is available, simulate data similar to the structure:
* overs: Overs completed in the innings.
* wickets: Wickets fallen by that point.
* runs: Runs scored until that point.
* total\_runs: Final score at the end of the innings.

**Implement Project using Python:**

# Import necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Simulate a dataset

np.random.seed(42)

data = {

'overs': np.random.randint(1, 21, 100), # Random overs between 1 and 20

'wickets': np.random.randint(0, 11, 100), # Random wickets between 0 and 10

'runs': np.random.randint(0, 200, 100), # Random runs between 0 and 200

}

df = pd.DataFrame(data)

df['total\_runs'] = df['runs'] + np.random.randint(0, 100, 100) # Simulate total runs

# Calculate run rate for feature engineering

df['run\_rate'] = df['runs'] / df['overs']

# Generate Box Plot

plt.figure(figsize=(8, 6))

sns.boxplot(data=df[['overs', 'wickets', 'run\_rate']])

plt.title('Box Plot for Overs, Wickets, and Run Rate')

plt.show()

# Generate Histogram

plt.figure(figsize=(8, 6))

sns.histplot(df['total\_runs'], kde=True, bins=15, color='skyblue')

plt.title('Histogram of Total Runs')

plt.xlabel('Total Runs')

plt.ylabel('Frequency')

plt.show()

# Generate Scatter Plot

plt.figure(figsize=(8, 6))

sns.scatterplot(x=df['overs'], y=df['total\_runs'], hue=df['wickets'], palette='viridis', size=df['run\_rate'], sizes=(20, 200))

plt.title('Scatter Plot: Overs vs Total Runs')

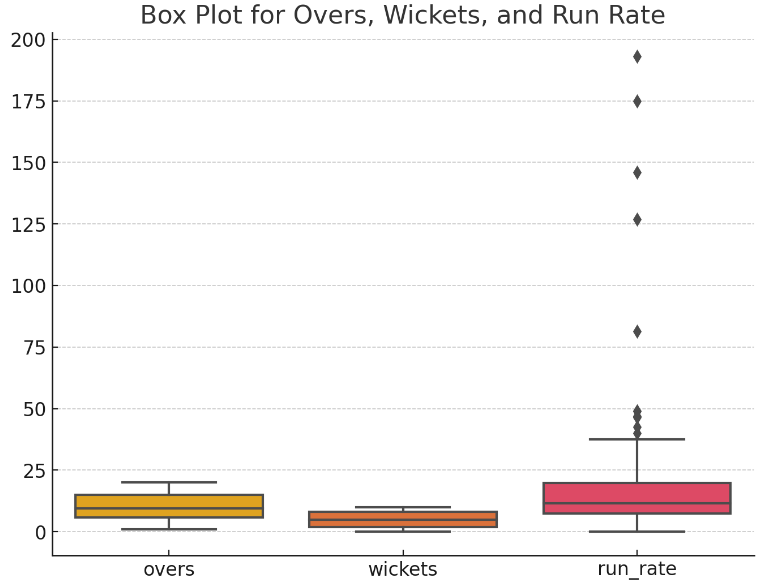
plt.xlabel('Overs')

plt.ylabel('Total Runs')

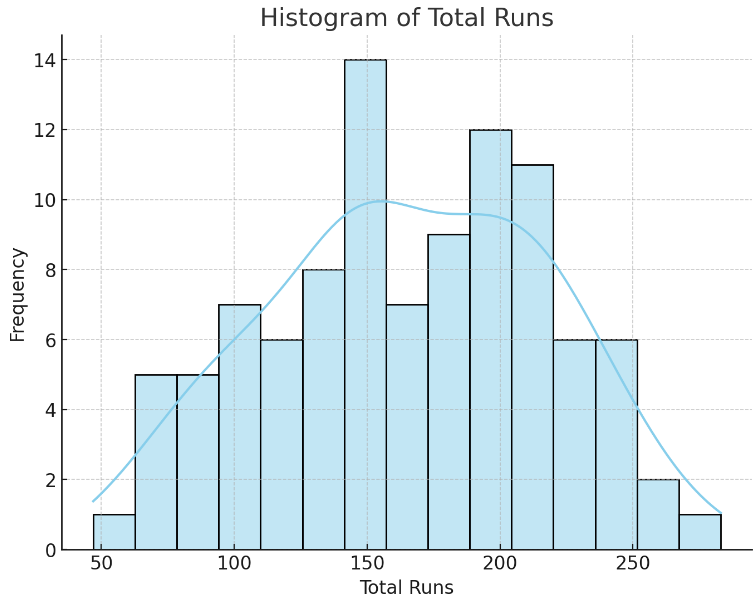
plt.show()

**Visualize data with box plot, Histogram, Scatter Plot:**

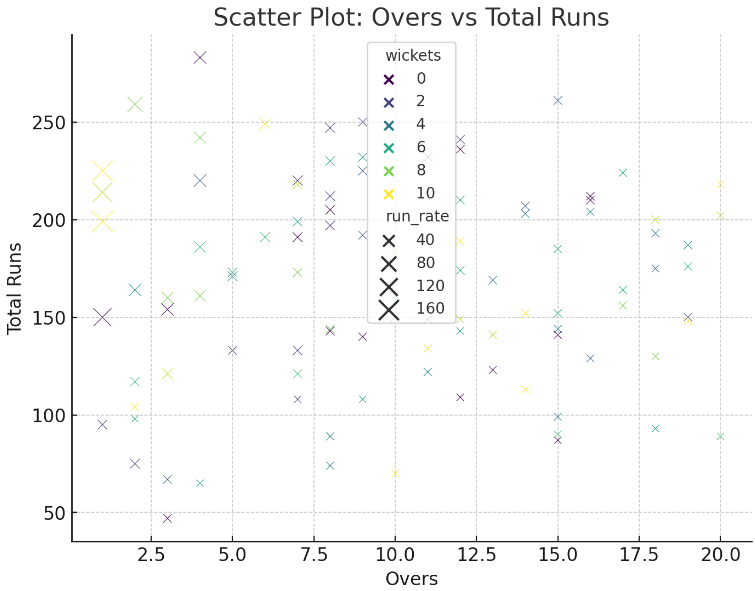
Box Plot :-



Histogram :-



Scatter Plot :-



**Plot Pearson correlation and explain about relation :-**

The Pearson correlation coefficient is a statistical measure that quantifies the strength and direction of the linear relationship between two variables. It ranges from -1 to +1:

* +1 indicates a perfect positive linear relationship.
* -1 indicates a perfect negative linear relationship.
* 0 indicates no linear relationship.

In the context of cricket score prediction, the goal is to predict a team's score based on various factors, such as the players' performance, pitch conditions, weather, and historical data. To understand how these variables relate to the total score, we can use Pearson correlation to analyze the strength of relationships between different variables and the final score.

**Identify Dependent and Independent features :-**

In the context of cricket score prediction, we typically want to predict the total score of a team based on various independent factors.

* Dependent Variable (Target): This is the variable you're trying to predict. In this case, it's the total score of the team in a match.
* Total\_Score: This is the target variable because you are trying to predict the total score of the cricket team.
* Independent Variables (Predictors): These are the features (or attributes) that will help you predict the dependent variable. These can include player performance metrics, environmental factors, and match-related data.
* Batting\_Avg
* Boundaries\_Hit
* Overs\_Played
* Weather\_Conditions
* Pitch\_Condition
* Opposition\_Strength
* Wickets\_Lost

**Analyse /Predict as per problem statement :-**

To effectively analyze and predict the total score of a cricket team, we need to follow a structured approach that combines both exploratory data analysis (EDA) and predictive modeling.