Cricket Score Prediction
Project Proposal

Group Members

Muhammad Aqeel

21-NTU-CS-1209

Faran Hassan

21-NTU-CS-1220

Abdul Rehman

21-NTU-CS-1234

Submitted to

Ma'am Kainat Abdullah

Department of Computer Science

National Textile University Faisalabad

Table of Content

Contents

Overview

The Cricket Score Predictor is a web-based application designed to predict the projected score of a T20 cricket match using machine learning. The application allows users to input match data, including teams, city, current score, overs played, wickets lost, and the number of runs scored in the last five overs. The model uses historical match data to provide a predicted final score for the match. The project utilizes Flask for the web interface and XG Boost for machine learning model prediction. The backend processes the inputs, preprocesses the data, and applies a trained model to generate predictions.

Features

- **Web Interface**: Extremely user-friendly and clean interface allowing users to insert the match specifics.
- **Data Prediction**: To predict the final score of the cricket match, we use historical data.
- **Dynamic City and Team Selection:** The selections of cities and teams through drop-down lists will ensure that the data on different pages is consistent.
- **Model Training:** High-quality training anu given machine learning score for the historical players
- **Scalability**: The model is designed to be scalable, thus allowing it to process data of various sizes.

Dependencies

The project depends on the following Python packages:

- Flask (for web development)
- pandas (for data manipulation)
- scikit-learn (for preprocessing and model evaluation)
- xg boost (for the machine learning model)

You can install these dependencies using `requirements.txt`:

- **`pip install -r requirements.txt`**

How to Run

1. Ensure all dependencies are installed using requirements.txt.
2. Place the DataSet_Cricket.csv dataset in the data/ folder.

3. Run the Flask application:

   python run.py

4. Open a web browser and navigate to http://127.0.0.1:5000/.

5. Use the form to input match details and get the predicted final score.

Usage

1. **Home Page**
   a. Input match details including **Batting Team**, **Bowling Team**, **City**, **Current Score**, **Overs Played**, **Wickets Lost** and **Runs in Last 5 Overs**.
   b. Submit the form to get prediction
2. **Prediction**
   a. Once the data has been submitted and processed, the program will automatically train the model to make predictions on the final score of the game.

Technical Details

1. Data Processing

a. Script: models/ preprocess.ipynb
b. City data is completed by using the data obtained from the venue.
c. The cities that meet the given criterion based on the count are chosen for the stay of the data.
d. One-hot encoding is used for categorical columns, whereas scaling is done for numerical columns.

2. Model

a. Model Type: XG Boost Regressor
b. Parameters:
n_estimators=1000,learning_rate=0.2,max_depth=12,random_state=1
c. Pipeline:
i. Column Transformer: Applies one-hot encoding to categorical columns.
ii. Standard Scaler: Scales the numeric columns.
iii. XGB Regressor: Predicts the score.

3. Model Training

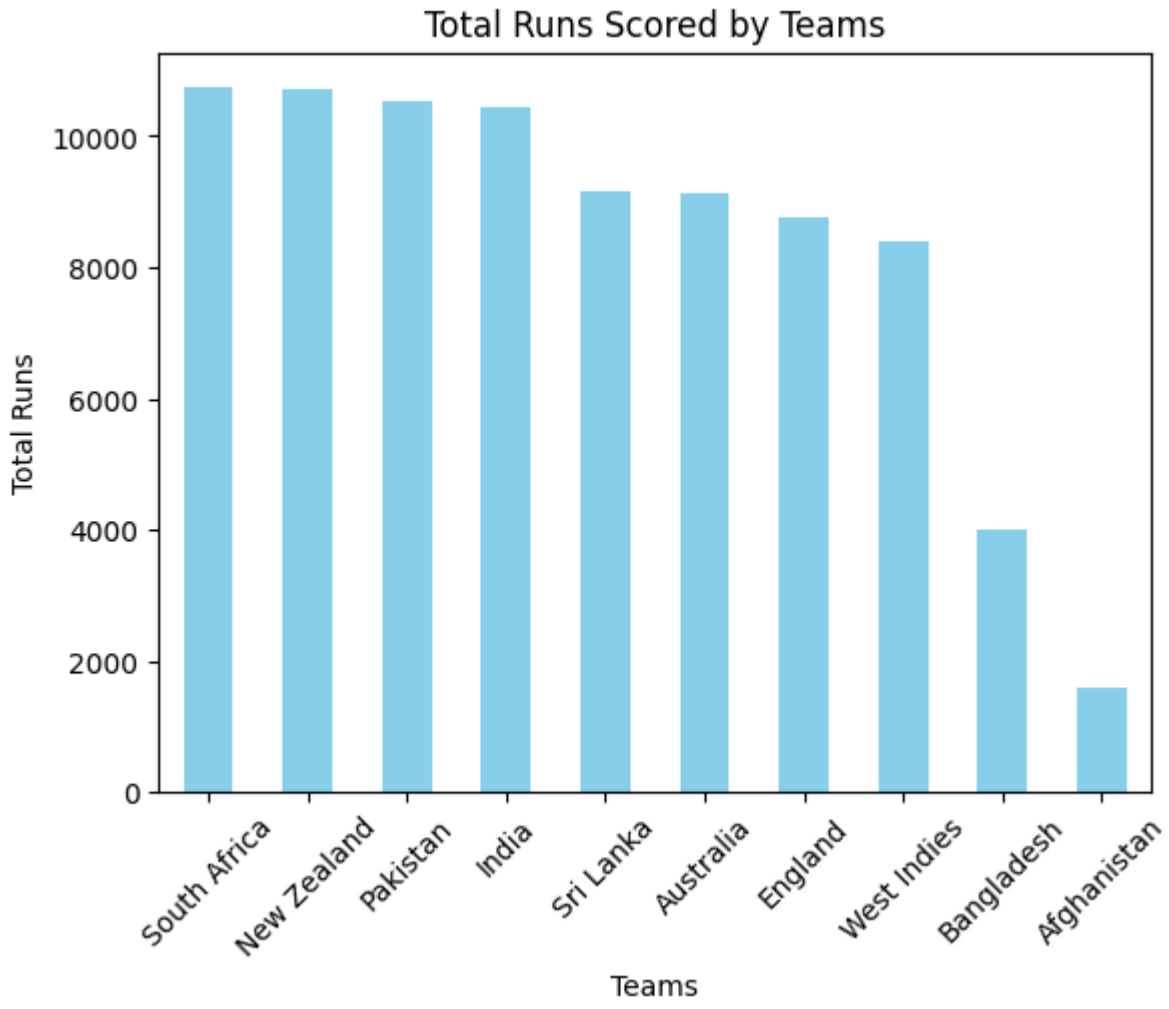
a. Script: models/ train_model.ipynb
b. Load and preprocess the data.
c. Define the features (X) and target (y).
d. Train-test split with 60% for training and 40% for testing**.**

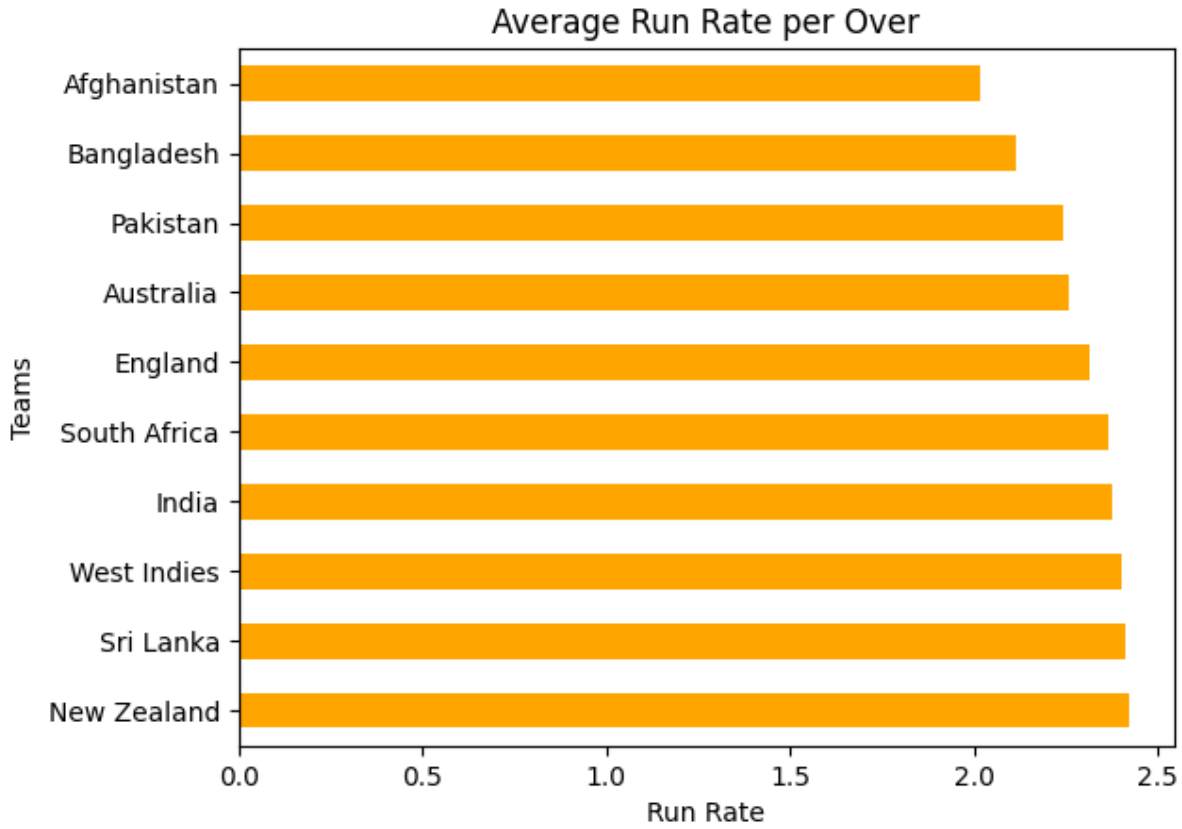
e. Build and train the model pipeline using XG Boost.
f. Save the trained model as cricket_predition.pkl.

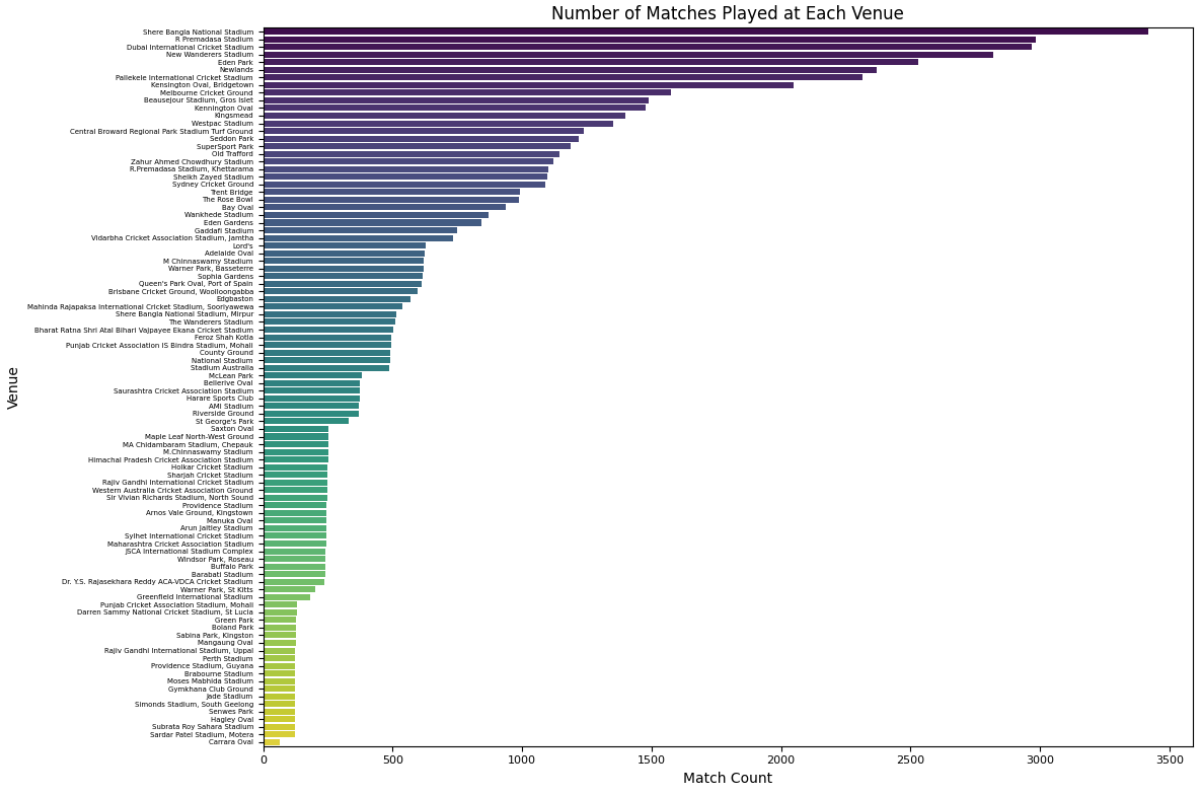
Deployment with Flask

- main.py: Main Flask application.
- index.html: HTML template for user interface
- Ensure all dependencies are installed.
- Place the model file cricket_predition.pkl. in the models/ folder.
- Run the Flask server:
- Navigate to http://127.0.0.1:5000/ in your browser to access the application.
- Users input match details in the form fields.
- The Flask backend processes these inputs and predicts the final score using the pre-trained model.
- The result is displayed on the same page.

Visualization




Average Run Rate per Over
Afghanistan
Bangladesh
Pakistan
Australia
England
South Africa
India
West Indies
Sri Lanka
New Zealand
Teams
0.0
0.5
1.0
1.5
2.0
2.5
Run Rate


Number of Matches Played at Each Venue
Venue
0
500
1000
1500
2000
2500
3000
3500
Match Count