T20 Cricket Six Predictor: Project Documentation

1. Project Overview

The T20 Cricket Six Predictor project aims to predict the probability of a batter hitting a six on the next ball during a T20 cricket match using Bayesian Logistic Regression. The project also includes a Streamlit-based web application to allow users to interactively predict sixes based on certain game conditions.

---

2. Dataset Description

- **Source**: The dataset used is IPL 2022 deliveries data (`ipl_2022_deliveries.csv.zip`).
- **Structure**: Contains match-level and delivery-level details, such as `striker`, `bowler`, `runs_of_bat`, and other contextual features.
- **Target Variable**:
  - `is_six`: A binary column created for prediction purposes, where `1` indicates the batter hit a six, and `0` indicates otherwise.

---

3. Key Steps in the Project

3.1 Data Preprocessing

- Extracted the target variable `is_six` by checking if `runs_of_bat` equals 6.
- Performed one-hot encoding for categorical features like `striker` and `bowler` to convert them into numeric form.
- Dropped irrelevant or non-numeric columns (e.g., `match_id`, `date`, `venue`) to focus on features relevant for prediction.
- Ensured only numeric features were used for model training.

3.2 Splitting the Dataset

- Divided the data into training (80%) and testing (20%) sets using `train_test_split` to evaluate model performance effectively.

3.3 Bayesian Logistic Regression

- Implemented Bayesian Logistic Regression using PyMC.
  - **Priors**:
    - Coefficients (`betas`) and intercept modeled as normal distributions with mean 0 and standard deviation 10.
  - **Likelihood**:
    - Modeled the probability of hitting a six using the logistic regression equation.
  - Used the No-U-Turn Sampler (NUTS) to sample from the posterior distribution.
- Parameters such as `tune`, `cores`, and `target_accept` were set to ensure convergence and computational efficiency.

3.4 Prediction and Evaluation

- Made predictions on the test dataset using posterior predictive sampling.
- Evaluated the model using metrics such as:
  - **Confusion Matrix**: Provides the breakdown of true positives, false positives, true negatives, and false negatives.

  - **Classification Report**: Includes precision, recall, F1-score, and accuracy.
- Saved evaluation results in a file (`evaluation_results.txt`).

---

4. Streamlit Application

A simple Streamlit-based web application was created to allow users to make predictions interactively.

Features:

- Users can input:
  - Runs scored by the batter in the current over.
  - Current over number.
- Outputs the likelihood of the batter hitting a six based on dummy predictions (placeholder logic in the current implementation).

Code: The Streamlit app code is saved in `six_predictor.py` for easy deployment.

---

5. Instructions to Run the Project

Step 1: Set Up the Environment

- Install Anaconda and create a virtual environment:

  ```
  conda create --name six_predictor_env python=3.9
  conda activate six_predictor_env
  ```

- Install required libraries:

  ```
  conda install numpy pandas matplotlib
  conda install pymc3 -c conda-forge
  pip install scikit-learn streamlit
  ```

Step 2: Run the Prediction Script

- Navigate to the project directory:

  ```
  cd path\to\SixPredictorProject
  ```

- Execute the prediction script:

  ```
  python six_predictor.py
  ```

Step 3: Launch the Streamlit App

- Run the Streamlit application:

  ```
  streamlit run six_predictor.py
  ```

- Open the app in your browser (default: `http://localhost:8501`).

---

6. Files and Artifacts

- **Code Files**:

    - `six_predictor.py`: Streamlit app code.
    - `evaluation_results.txt`: Model evaluation metrics.
- **Dataset**: `ipl_2022_deliveries.csv.zip`

---

7 Key Python Libraries Used

- **Pandas**: For data manipulation and preprocessing.
- **NumPy**: For numerical operations.
- **Scikit-learn**: For train-test split and evaluation metrics.
- **PyMC**: For Bayesian Logistic Regression modeling.
- **Streamlit**: For building the interactive web application.

---

8. Conclusion

This project demonstrates the use of Bayesian methods to predict cricket outcomes. The integration of a predictive model with a web interface makes it user-friendly and scalable. With additional data and feature engineering, the model can be enhanced to provide even more accurate predictions.