Status Report:
T20 Win Predictor

•••

CS 403

Overview

Problem Statement- Motivation
Formulation
Approach
Literature Survey
Future Plans for completion

Problem Statement

Motivation

- Cricket is full of uncertainties, especially in the shortest format like IPL
- We are passionate about cricket
- We will learn about the different classification models , their performance and behaviour
- We will learn about various hyper-parameter optimization techniques

Problem Statement

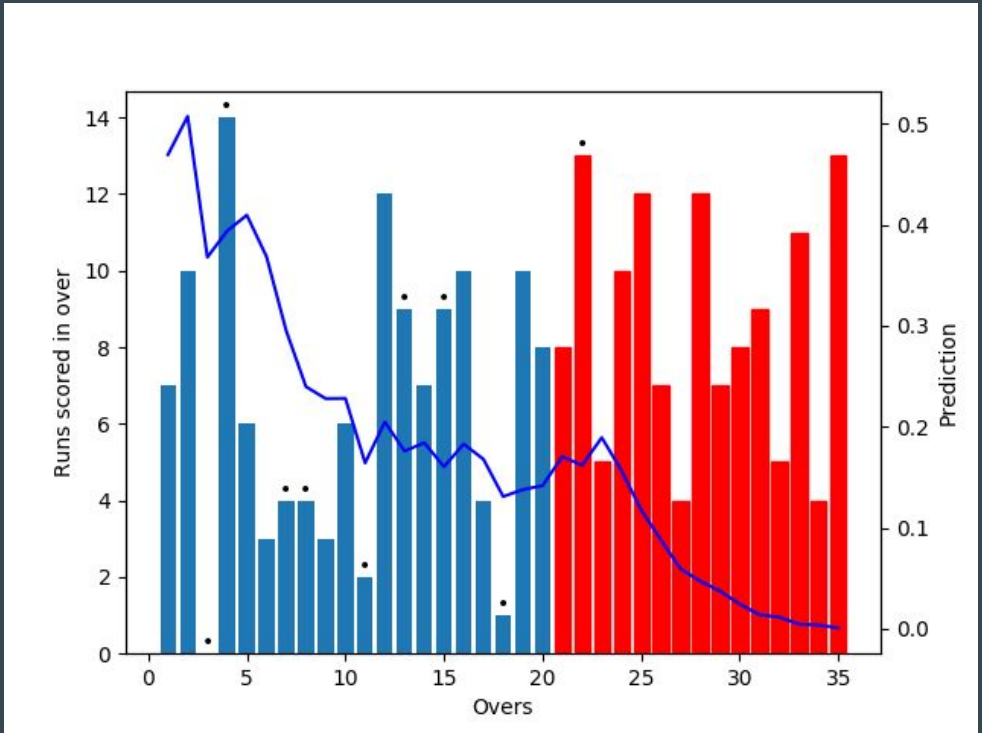
Formulation

- Given the scoreboard of the match we try to predict the outcome.
- We look at match scoreboard after every over, instead of predicting at each ball of an over.
- We extract the following features from the ball by ball data to use in our classifier.
    - For each over, we calculate runs scored and number of dismissals in that over.
    - Using information of current over we update the innings score, innings wickets, current run rate, required run rate, remaining target etc.

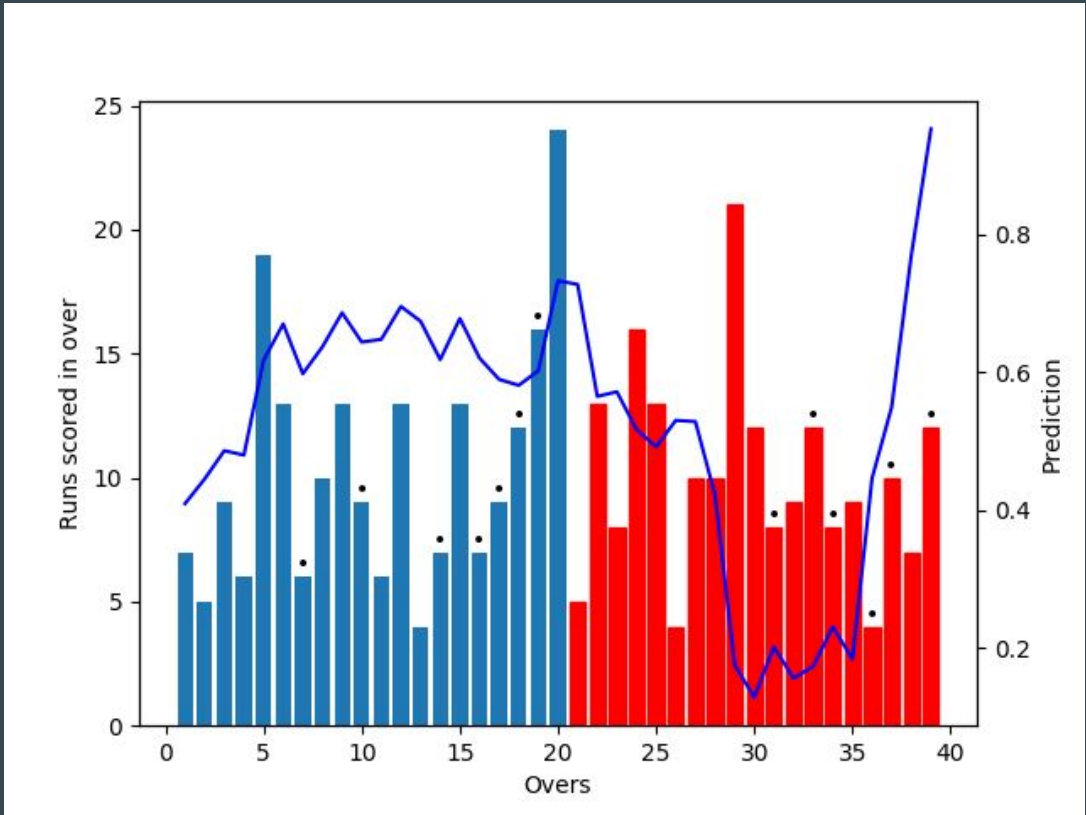
Approach

- We have used the multi-layer perceptron (MLP) classifier to get expected classification
- We have used scikit-learn library's implementation of the MLP classifier
- The neural network has 2 hidden layers with 12 nodes each.

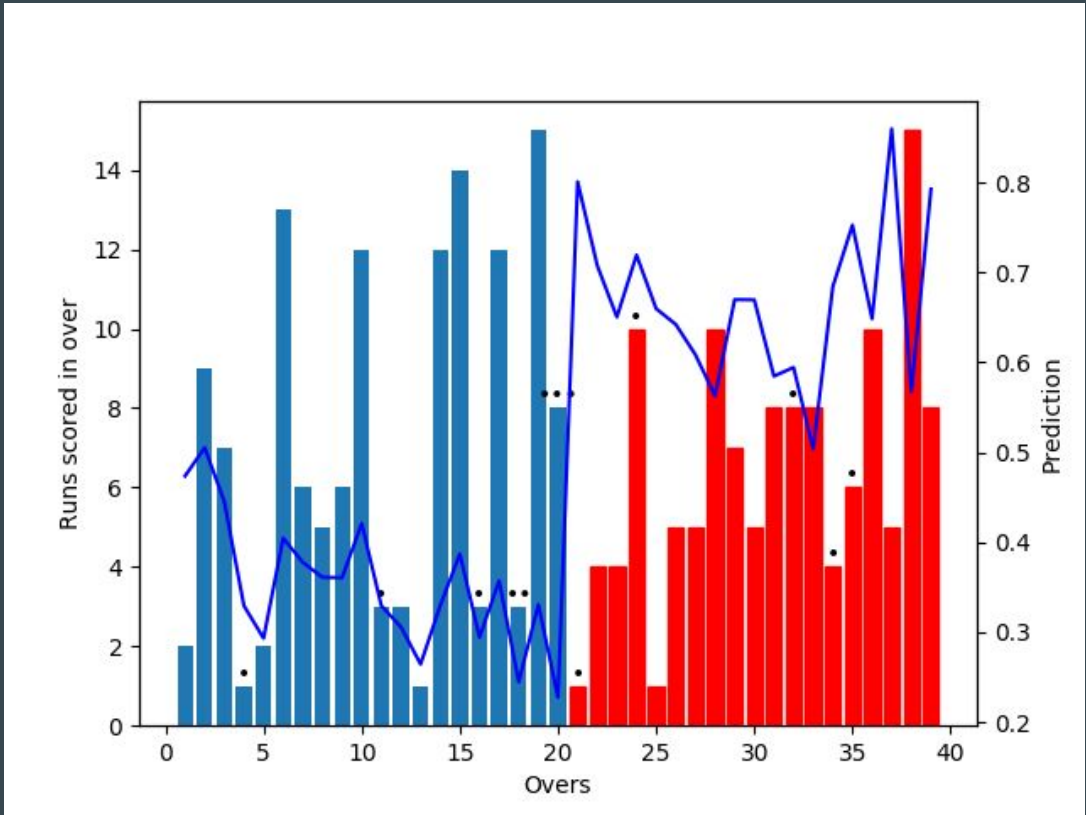
Sample Output



Actual match stats at http://www.iplt20.com/archive/2008/3



Actual match stats at http://www.iplt20.com/match/2016/60



Actual match stats at http://www.iplt20.com/match/2016/40

Literature Survey

- https://arxiv.org/ftp/arxiv/papers/1511/1511.05837.pdf
  - Problem Statement : Predicting the outcome of a English County T20 match.
  - 2 Models were created one using only team features and other using both team and player features
  - Features of both models were then selected using different correlation scores.
  - Classification techniques like Naive Bayes and Random Forests were used to predict
- http://pages.cs.wisc.edu/~shruthir/Documents/MachineLearning_Final_Report.pdf
  - Problem Statement : Predicting the outcome of a T20 match
  - Features for each player: Average Runs, Strike Rate, etc for batsmen and Average wickets per match, Economy rate, Strike Rate etc. for bowlers
  - Classifiers used : Decision Trees, Naive Bayes, Random Forests, SVM, Adaboost
  - Evaluation : 10 fold cross validation on IPL data set
  - Another approach using player ratings was employed, which considered bowler-batsmen interaction and time scaling

Future Plans

- Following are the approaches described in the literature that we will try to implement in the future
    - Generate the features of each team based on their past performances, like win percentage, batting average, bowling average, batting run rate, bowling economy etc.
    - Generate the features of each player for batting : SR, average, no. of 30s, no. of 4s, no of 6s, and for bowling : SR, economy, average, 3 wicket hauls.
    - Use different techniques SVM, Elastic nets, Adaboost, ensemble learning