CRICKET MATCH RESULT PREDICTION

Team Members

Abhishek Soni-153050043

Akash Kumar-153050071

Sankalp Rangare-153050087

Swaresh Sankpal-153050085

Introduction

- Goal of project is to predict the outcome of a One Day international Cricket match.
- Game winner depends on various parameters like toss, venue, players statistics etc.
- Machine Learning algorithms can be used to predict the outcome of the game.
- Although cricket is game of uncertainties, achieving 100% accuracy is difficult but it depends upon above described parameters.
- Objective: Build a model which will predict the output of a cricket match as Winner(either team1 or team2) or Tie.

Dataset

- Available Datasets
  - Kaggle: datasets related to T20 matches only - our model is for ODI
  - Cricsheet.com: No information on playing XI
- Crickinfo.com maintains year wise match information
- Crawled data using a Python's BeautifulSoup from **cricinfo.com.**
- It consists of 50 different features related to game of cricket.
- Outcome of the game is either win, lose or tie.
- Currently our dataset holds information of all ODI matches held in between **1990-2016** which is **3050 matches**.

Dataset

Approach 1 (Pre-Midterm)

- team1
- team2
- city
- Date
- tosswinner
- first to bat
- Playing 11 names for team1
- Playing 11 names for team2

Classification Algorithms	Accuracy (%)
Naive Bayes	83
K-Neighbour Classifier	66

Dataset

Approach 2 (Post-Midterm)

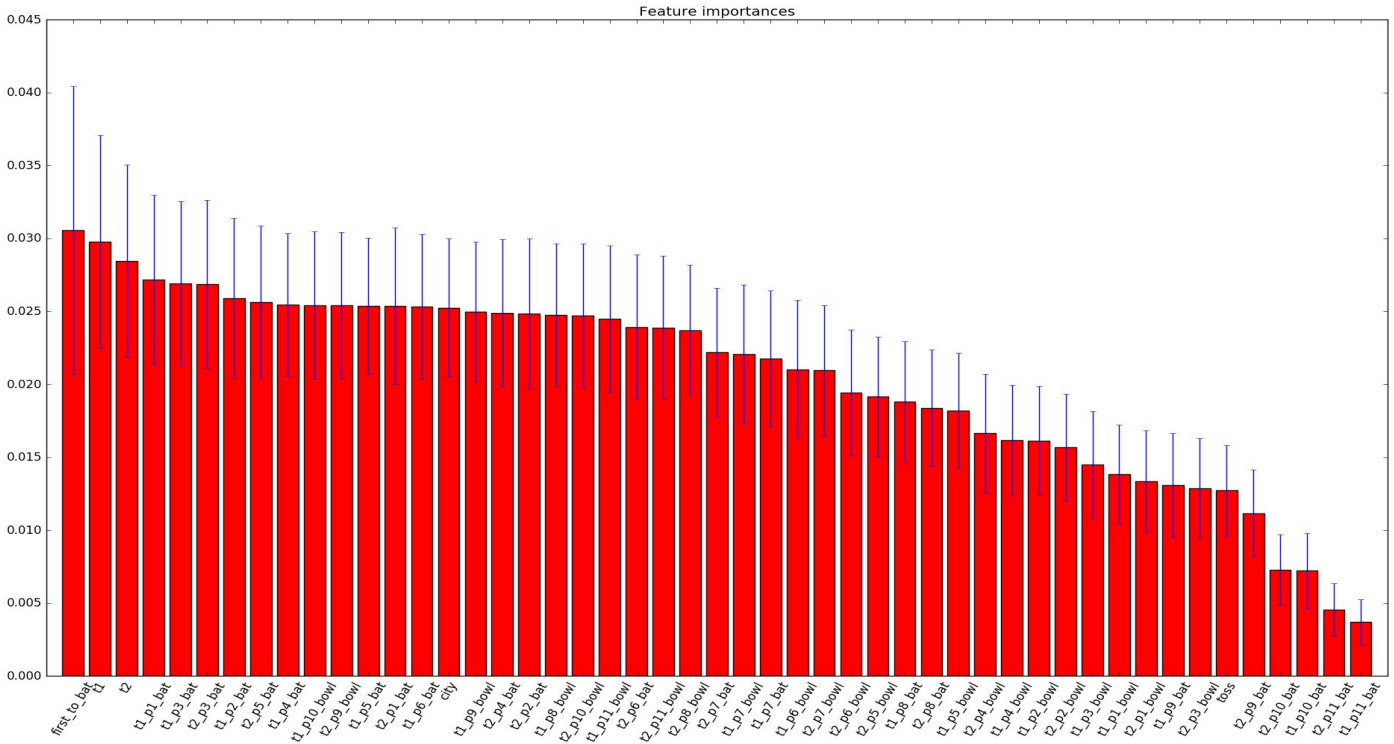
- team1
- team2
- city
- Date
- tosswinner
- first to bat
- { Batting rank, Bowling Rank } for each player in Playing 11 for team1
- { Batting rank, Bowling Rank } for each player in Playing 11 for team2

Dataset

Approach 3

- Instead of giving rank (bowling, batting) to individual players, we thought it would be better if we given an overall batting and bowling rank to team so that the number of features in our classification model are reduced.
- This approach didn’t show any improvement compared to initial approach.

Feature Importance



Classification Algorithms

1. Naive Bayes
2. Decision Tree
3. Gradient Boosting
4. Support Vector
5. Logistic Regression
6. K-Nearest Neighbors
7. Extra-Tree
8. Multilayer Perceptron

Naive Bayes Approach

- Easy and fast method to predict class of test data set.
- It also performs multi-class prediction.
- Classification technique based on Bayes Theorem.
- Used MultinomialNB module of sklearn library

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Decision Tree Approach

- It maps observation of an item to conclusion about its target value.
- These models are highly accurate,stable and easy to interpret.
- Types of decision trees:
  - Categorical Variable Decision Tree
  - Continuous Variable Decision Tree
- Used DecisionTreeClassifier module of sklearn library.

Gradient Boosting classification

- It starts with rough prediction and building series of decision trees.
- Its gives weights to every model based on their accuracy.
- Consolidated result is generated at the end.
- Used GradientBoostingClassifier module of sklearn library

Support Vector Classifier

- Very effective method in higher dimensional space
- This method tries to find the best possible separating hyperplane.
- Different kernel functions can be used as decision function.
- Used SVC module of sklearn library.

Logistic Regression

- This method can be used to solve multi-classification problem.
- Given input data point, this technique tries to predict the probability of class of data point.
- Cost function in Logistic regression is logarithm of the sigmoid function.
- Used Logistic Regression module of sklearn library.

K-Neighbour Classifier

- This non-parametric method is used for regression and classification.
- Data set is plotted on n-dimensional space.
- For an unclassified data point, its k nearest neighbours class labels are noted and the label which occurs maximum number of times is given to unclassified data point.
- Used KNeighborsClassifier module of sklearn library with parameter neighbours=2.

Extra-Tree Classifier

- This method implements meta estimator that fits many random decision trees.
- It works on various sub-samples of datasets and uses averaging to improve accuracy and control over-fitting.
- Used ExtraTreesClassifier module of sklearn library.

Neural Network (Multilayer Perceptron)

- This model optimizes the log-loss function using LBFGS, ADAM or stochastic gradient descent.
- MLP Classifier trains iteratively since at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters.
- This implementation works with data represented as dense numpy arrays or sparse scipy arrays of floating point values.
- We have used MLP classifier module of sklearn neural network library.
- Parameters :
  - Learning rate : 0.01
  - Solver : Adam
  - Hidden layer size : 2 layers with 10 neurons each

Results

Classification Algorithms	Accuracy (%)
Naive Bayes	69.12
K-Neighbour Classifier	73.91
Decision Tree	74.12
Neural Network (MLP)	74.95
SVC	75.51
Tree Classifier	78.09
Gradient Boosting	79.09
Logistic Regression	80.17

Work Distribution

- Selection of features - Brainstorming by all members
- Akash - Search and finalisation for data sources , Data Crawling of year wise data from cricinfo
- Abhishek - Data Crawling of batting and bowling rankings from reliance icc ranking and Feature Engineering, Feature importance
- Sankalp Rangare : Implemented different classification techniques like NaiveBayes, SVM, Logistic regression and Gradient boosting.
- Swaresh Sankpal : Implemented different classification techniques like logistic decision tree, KNN, tree classifier, neural network (MLP).

References

[1] **Cricinfo** webpage available at http://www.http://www.espncricinfo.com/, retrieved March 2017.

[2] **Classifiers (NB, KNN, SVC, etc..)** http://scikit-learn.org/stable/supervised_learning.html

[3] **Beautiful Soup Documentation** webpage available at https://www.crummy.com/software/BeautifulSoup/bs4/doc/, retrieved March 2017.

[4] **Cricsheet** webpage available at http://cricsheet.org/, retrieved March 2017.

[5] **Reliance ICC rankings** webpage available at http://www.relianceiccrankings.com, retrieved March 2017.

[6] Multi Layer Perceptron http://scikit-learn.org/stable/modules/neural_networks_supervised.html