CS 578 Project: Effect of Team Selection on Performance

Parameswaran Desigavinayagam Pradeep Kumar Srinivasan

Introduction

We explore the possibility of predicting the winner of a T20 cricket match based on the team composition and the history of player performances from the past.

Problem

Given the composition of two cricket teams along with past statistics of the teams and players, how well can we predict their match performance?

Dataset

The dataset consists of YAML files with information about the T20 cricket matches (from https://cricsheet.org/). Each file has ball-by-ball data (batsman, bowler, non striker, runs scored etc) for matches played within the Indian Premier League.

Experimental Setup:

- We extracted individual statistics like batting average, bowling strike rate, etc. for each player on either side and used them as input features. We also used team-level features like win rate and net run rate.
- The target variable we wanted to predict was the match outcome - win or loss.
- Therefore, we used different classifiers such as linear SVM, kernel SVM, logistic regression, and ensemble classifiers.
- We further used feature selection to narrow down the features. The algorithms we used were greedy selection and forward fitting algorithms.
- We used slack variables in order to possibly lessen the effect of outliers.
- To set the hyper-parameters for our model, we used k-fold cross-validation.
- To test our models, we plotted precision vs recall curves. We also plotted accuracy against number of samples and accuracy against the hyperparameters.
- We trained and tested our classifiers against different subsets of the data - such as one season at a time, all 10 seasons together, and league matches vs final matches. We also ran it on training sets with different combinations of features. For example, we tested batting averages alone against the match outcome, then bowling strike rate alone, and so on.
- Finally, we validated our model against data from two other T20 datasets - Big Bash League (BBL) and International T20s.
- We implemented our project in Matlab and Python. We extracted features from the raw YAML dataset using Python and implement different classifiers in Matlab. Finally, we plotted graphs using Matlab.

Approach

- We split the data into (Xtraincv & ytraincv) and (Xtest & ytest)
  (where number of rows in traincv = training_percent * number_of_matches)
- Whenever past data for a player was missing, we used a default value for the feature.
- We performed k-fold cross-validation on the training set without looking at the test set. This allowed us to choose the hyper-parameters C and gamma.

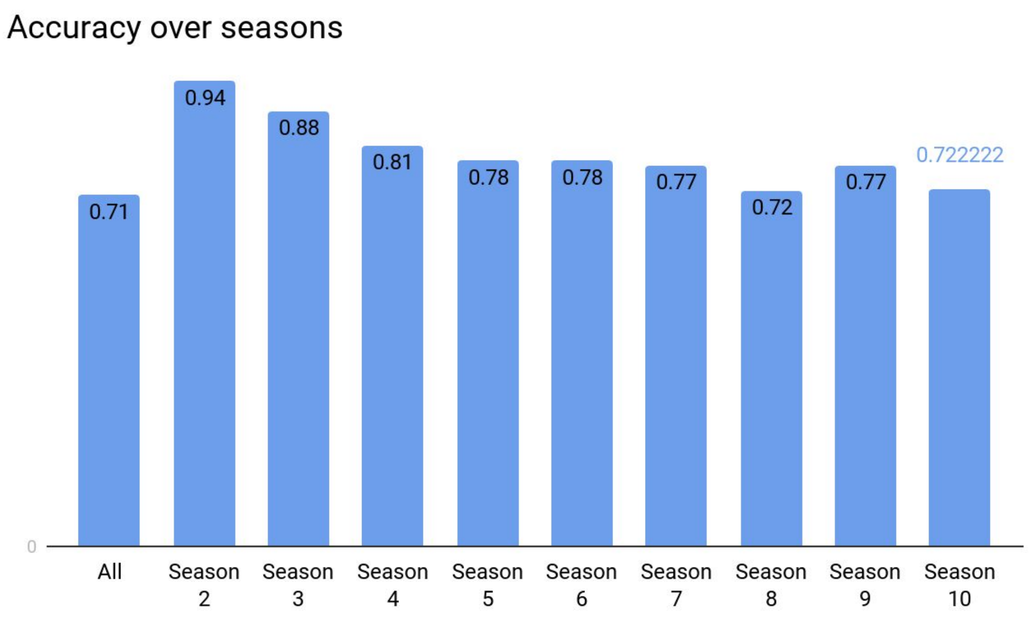
Feature Description

Team features	**Description**
Win rate	Games won / Games played
Net run rate	Runs given / overs bowled - runs scored / overs batted

Player-level features	**Description**
Batting average	Runs scored / matches batted
Batting strike rate	100 * Runs scored / balls faced
Bowling economy	Runs given / overs bowled
Bowling strike rate	Overs bowled / wickets taken
Bowling average	Runs given / wickets taken

Higher-level features	**Description**
Batting combo	Batting average + strike rate
Bowling combo	Bowling average + strike rate + economy

Results and Analysis



1. Accuracy in each season

(Feature selection - Greedy, 12 feature subset)

S.No	Season(s)	Primal SVM Accuracy
1	all (2-10)	0.71
2	2	0.94
3	3	0.88
4	4	0.81
5	5	0.78
6	6	0.78
7	7	0.77
8	8	0.72
9	9	0.77
10	10	0.72

We note that per-season accuracy is higher than accuracy when training over all seasons. This is to be expected given the high turnover within teams, especially every three seasons, when there is a nearly-complete reshuffle in IPL.

2. Best accuracy by classifier

(Feature selection - Greedy, 24 features selected)

Classifier	Best Accuracy
Dual SVM	0.71
Ensemble	0.70
Logistic	0.72
Primal SVM	0.78

We found Primal SVM to have the highest accuracy. In general, the SVM models outperformed the Ensemble classifier.

3. Comparison of Feature Selection algorithms

S.No	Percent of	Feature selection	Accuracy for Feature subset size

	data trained on	algorithm	(20)	(10)	(6)
1	70%	Greedy	0.72	0.73	0.72
2	70%	Forward Fitting	0.73	0.72	0.73
3	75%	Greedy	0.75	0.75	0.74
4	75%	Forward Fitting	0.73	0.75	0.74

We did not see an increase in accuracy when moving from low number of features to high number of features, suggesting that a few players have a disproportionate impact on the match outcome. The features most commonly picked by the feature selection algorithms were: the bowlers on both teams (players 9, 11, 21, and 22 in the overall order. Next in order of importance were batsman at positions 3, 4, and 5, which suggests that they have more impact than the openers at position 1 and 2

4. Accuracy vs Percent of Training data used

Features used - `22 * bowling combo <bowling average + economy + strike rate>`
Feature selection - Greedy, 10 feature subset

S.No	Percent of data trained on	Primal SVM Accuracy
1	0.6	0.74
2	0.7	0.73
3	0.75	0.75
4	0.8	0.72
5	0.9	0.75

We did not see a significant trend when changing the percentage of training data, suggesting that knowing the results of a few more matches does not make much of a difference as compared to past performance data.

5. Comparison of Classifiers in BBL & T20 Data

S.No	Classifier	Big Bash League			T20 Internationals		
		Accuracy for Feature subset size			Accuracy for Feature subset size		
		(46)	(68)	(90)	(46)	(68)	(90)
1	Dual SVM	0.57	0.65	0.55	0.58	0.52	0.58
2	Primal SVM	0.51	0.61	0.57	0.64	0.69	0.63
3	Logistic Regression	0.63	0.63	0.63	0.61	0.61	0.61

4	Ensemble	0.71	0.71	0.71	0.69	0.69	0.68

Surprisingly, the Ensemble classifier outperforms the other classifiers when it comes to other leagues. It may be less susceptible to overtraining compared to the others.

6. Feature combination vs Accuracy - (with Primal SVM)

S.No	Percent of data trained on	Accuracy for Feature combination			
		batting average	batting strike rate	bowling economy	bowling strike rate
1	0.6	0.67	0.66	0.67	0.73
2	0.7	0.68	0.68	0.67	0.74
3	0.75	0.66	0.68	0.65	0.77
4	0.8	0.68	0.66	0.67	0.74

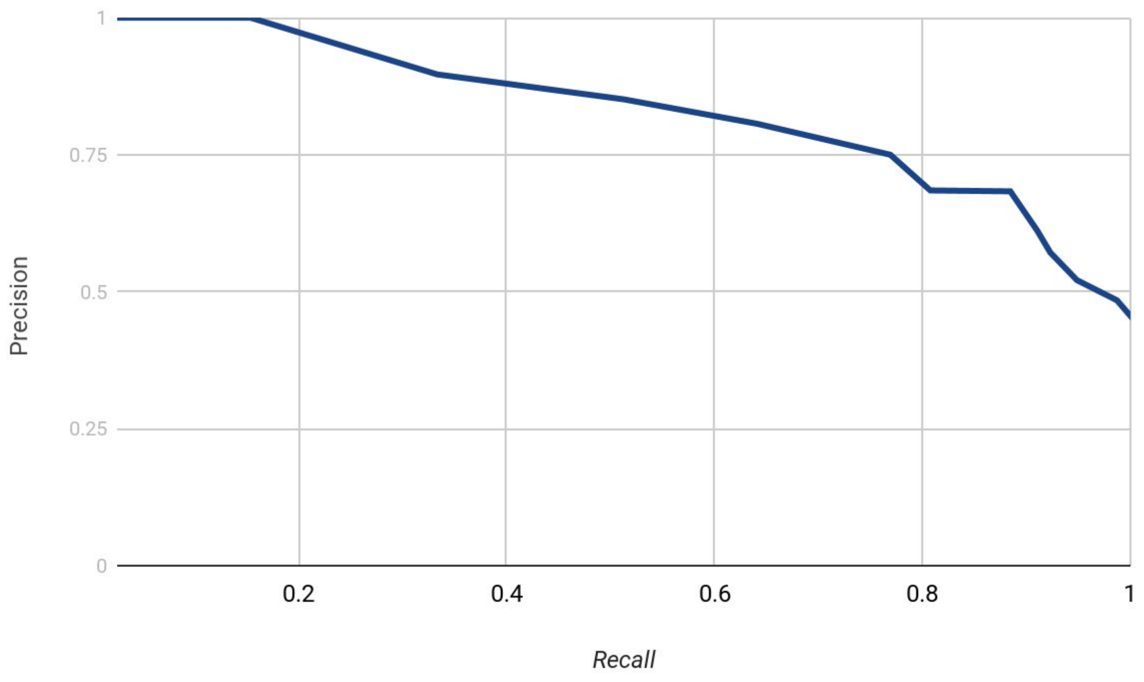
Bowling strike rate is clearly the most predictive feature type. The next best is batting average, followed by bowling economy and batting strike rate, which is usually the feature that people pay attention to.

7. Predicting Finals after training on data from League matches

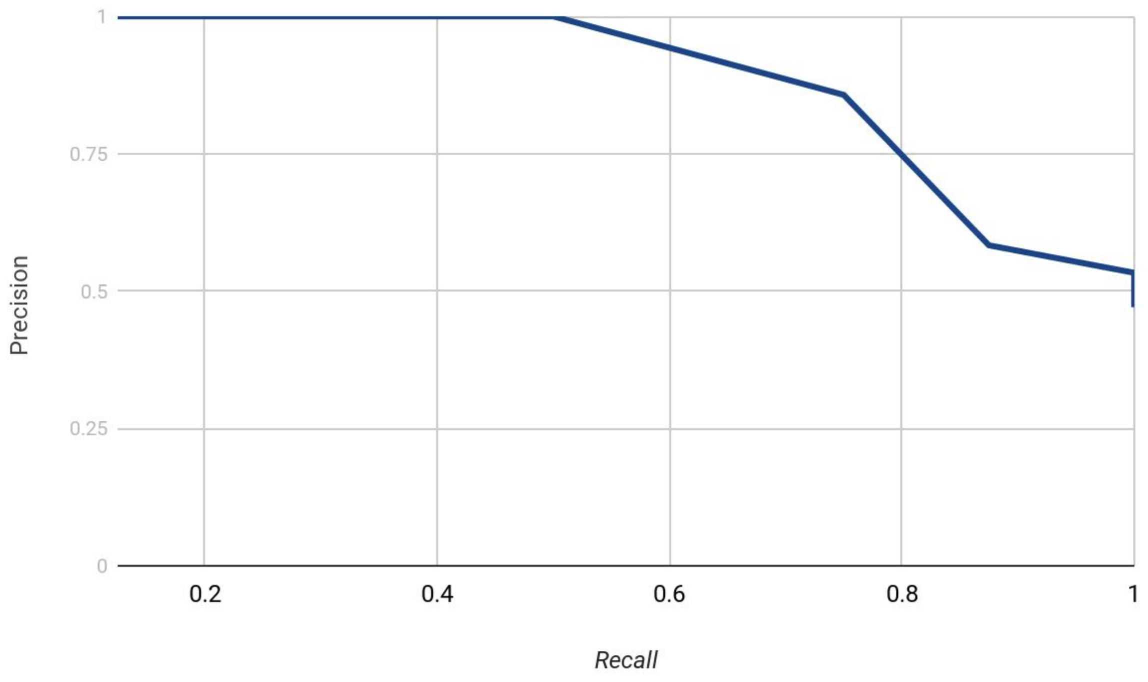
S.No	Season	Average Accuracy
1	2	0.84
2	3	0.56
3	4	0.53
4	5	0.34
5	6	0.81
6	7	0.69
7	8	0.50
8	9	0.72
9	10	0.56

Precision vs Recall

Seasons 2-10 - Primal SVM
Greedy, 20 feature subset

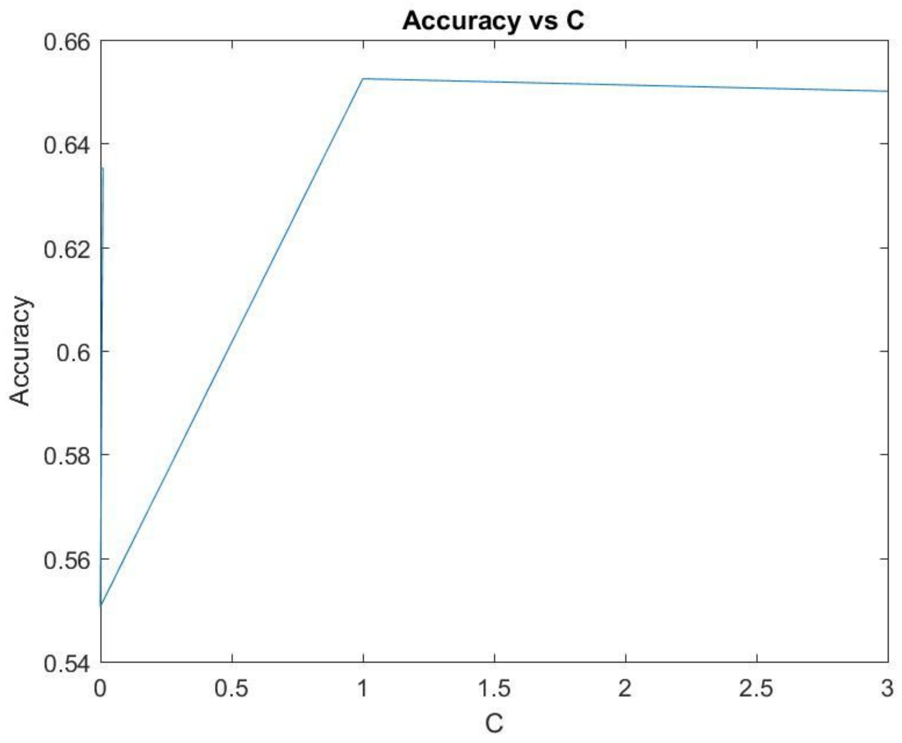


Season 8 - Dual SVM
Forward Fitting, 24 feature subset

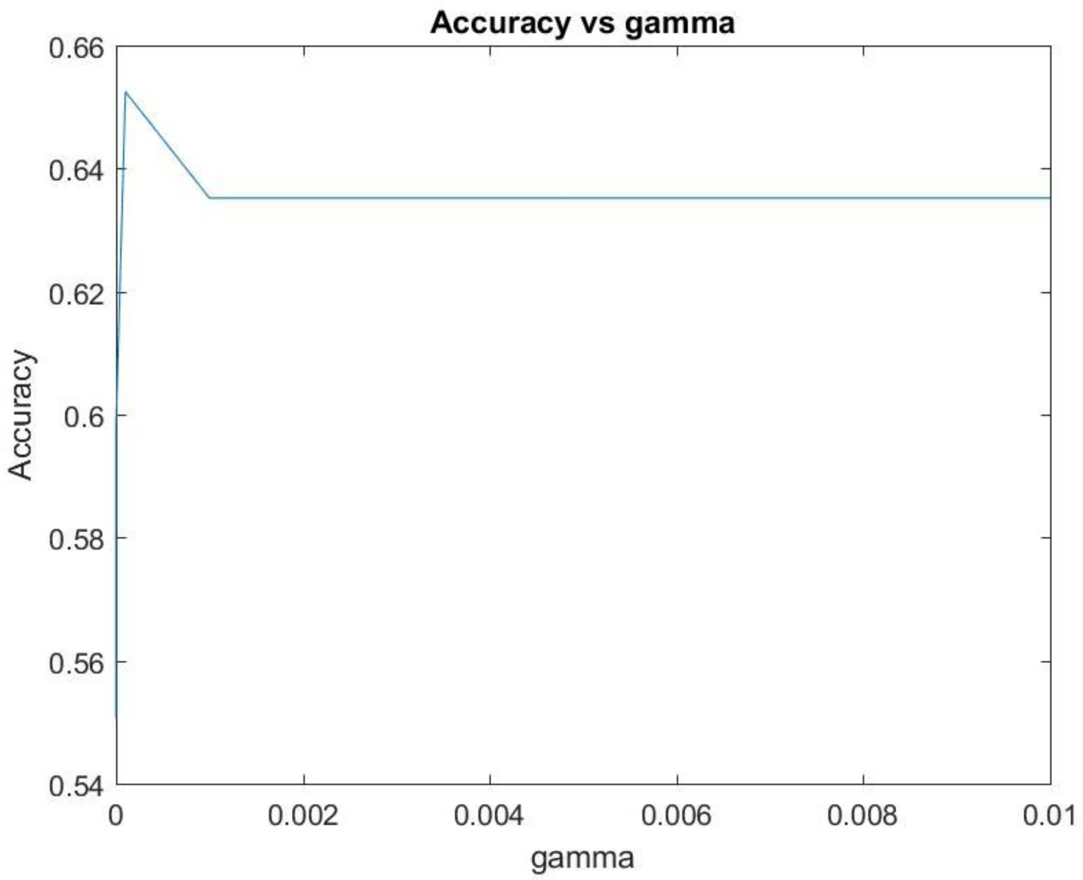


We can see that as we try to increase recall, the precision decreases.

Accuracy vs Hyperparameter C



Accuracy vs Hyperparameter gamma (for RBF Kernel)



We did not notice much variation in accuracy beyond some values for the hyperparameters.

Discussion and Conclusion

We found that is possible to forecast the winner of a T20 match with around 71% accuracy when training over all seasons, and an average of 78% when training on one season at a time. This seems to be a higher rate of accuracy than existing T20 prediction levels and even standard betting odds[1]. As per our experiments, the most predictive features were bowling strike rates and averages, along with batting averages. Of the four classifiers we considered - linear SVM, dual SVM, logistic regression, and ensemble classifiers - the one that performed best on average was Linear SVM. Overall, we found that SVMs outperformed logistic regression and ensemble classifiers on our datasets.

One curious thing we found in another paper was the possibility of exploiting this predictive power in betting markets:

> "In many similar academic studies, e.g. (Warner, 2010), at this point it is usual to find that any accuracy level achievable with machine learning methods is lower than that found in the gambling industry. The relatively consistent over performance by the naive Bayes learner displayed in Figure 16 is therefore surprising and represents a potential financial opportunity. Annually the annual cricket gambling market is thought to be worth $10 billion legally and between $40-50 billion illegally, driven by the Asian markets. Given the scale of the betting industry worldwide, there is a lot of potential for monetary gains for anyone with access to superior prediction techniques."[2]

We compared our models to the models used by the above paper, which dealt with English County T20 matches. They achieved an average performance of around 64.5% using a Naive Bayes model over multi-level team and player features and 62.4% on simple features, and had worse performance using logistic regression, random forests, and gradient-boosted trees. Our model achieves accuracy of TODO on combined features and an accuracy of TODO on simple features and seems to outperform the above paper.

We also note that our model performed well even on foreign datasets like BBL and international T20 matches, which suggests that the features we chose strongly predict the outcome of a T20 match regardless of the league or even country where it is being played.

In the future, we could extend this approach to other formats of cricket as well: ODI and Test matches. We could also try different classifiers. With respect to the data, one curious irregularity was that a team changed their name slightly from one season to the next, "Rising Pune Supergiants" to "Rising Pune Supergiant". We would undoubtedly have better performance if we had considered them to be the same team. Regarding features, we could explore features not present in our dataset like day or night conditions and pitch details. Most importantly, given the predictive power of the features we tried, it would be worth it to test other feature combinations, giving special weight to bowler statistics.

---

[1] Using Machine Learning to Predict the Outcome of English County twenty over Cricket Matches, Stylianos Kampakis, William Thomas, Arxiv (https://arxiv.org/abs/1511.05837)

[2] ibid.