ClassicXI- Optimal IPL Team selection based on Player Rating and Venue Based Analysis

Ritik Hariani
PES1201801558
5 J

Lavitra Kshitij Madan
PES1201800137
5 A

Team

Aditya G Burli
PES1201800034
5 J

Arjun Chengappa
PES1201800119
5 A

What problem have you selected and what dataset have you used to solve the problem?

Cricket Statistics Analysis

This is a vast domain and we will be focussing on Predicting the playing XI of a team.
Indian Premier League (IPL) is one of the most prestigious T20 cricket tournaments across the world. This paper focuses on suggesting the playing XI given a squad of 15 players. We propose a model that rates all the players in the IPL and based on their predicted rating suggests the best playing XI combination the team can select for a given match at a given venue. Our model experiments with various distance similarity measures and other statistics to provide the best possible playing XI. It also provides insights on toss decision at a given venue based on history.

DATASET - IPL

- One of the datasets used in this project is taken from the Kaggle source. This consists of two csv files deliveries.csv and matches.csv
- The file deliveries.csv contains the ball by ball data of every match since 2008. Data includes columns such as bowling team, batting team, over, ball, batsman etc. This dataset has a total of 21 columns and 179078 rows.
- The file matches.csv contains details about the match as a whole. Data includes columns such as season, city, teams, toss_winner, toss_decision, etc.
- The second dataset used was obtained from the official IPL website. It has scraped using a Scraper that can be found here scraper.js This dataset consists of the playing XI details for every match that has been played in the IPL from 2008 to 2020 and also includes the respective player stats such as the player's full and short name, date of birth, nationality and their skillset. The winner, score of both teams and date of every match was also obtained. This dataset consisting of 167422 lines of data was extracted in JSON format and was converted into a Dataframe.

Why is what you have done important/ useful?

IPL is a tournament that involves big bucks and big players. It has caused a frenzy all over the world. Every team management spends a large amount of time and money to decide the playing XI players from the 15 in the bag.

Usefulness

- Player ratings for optimal playing XI for a given venue against a given team to provide them the best opportunity to win the match.
- Quantitative assessment of players at the end of the season
- Strategic bidding can be made in the auction along with analyzing the strengths of scouted players to be bought in the auction

What is the approach you have taken?

Modelling Batsmen algorithm

The ***input*** features to this model are:

1. Matches: Number of innings player batted in
2. Batting Average: Total Runs divided by (Number of Innings - Not Outs)
3. Strike Rate: (Total Runs / Balls faced) * 100
4. Number of Fifties
5. Number of Hundreds
6. Number of 4s and 6s

Output: Batsman rating

Algorithm 1: Modelling Batsmen

Input: Players data from IPL 2008 to 2019 seasons, IPL Career Stats
Output: Batsmen Rating
for *all players in the dataset* **do**
    /* All time Form (IPL 2008 to 2019) */
    matches = Number of innings batted in (Min Max Normalization);
    achievementWeight = (20 * Hundreds) + (10 * Fifties) + (3 * Sixes) + Fours;
    careerStatScore = (0.3 * achievementWeight) + (0.55 * Batsman Average) + (0.15 * StrikeRate);
    OverallCareerScore = (matches * careerStatScore);
end
for *all players in the recent dataset* **do**
    /* Recent Form (IPL 2018 to 2019) */
    matches = Number of innings batted in (MinMax Normalization);
    achievementWeight = (20 * Hundreds) + (10 * Fifties) + (3 * Sixes) + Fours;
    StatScore = (0.3 * achievementWeight) + (0.55 * Batsman Average) + (0.15 * StrikeRate);
    RecentScore = (matches * StatScore);
end
for *all players in the merged dataset* **do**
    /* Calculating Batsman Rating */
    OverallCareerScore=OverallCareerScore/max(OverallCareerScore);
    RecentScore = RecentScore / max(RecentScore);
    BatsmanRating = (0.4 * OverallCareerScore) + (0.6 * RecentScore);
    Perform MinMax Normalization(Batsman Rating)
end

What is the approach you have taken?

Modelling Bowler algorithm

The ***input*** features to this model are:

1. Matches: Number of Innings player bowled in
2. Bowling Economy: Number of Runs conceded per over bowled
3. Bowling Average: Number of Runs conceded per wicket taken
4. 4 wicket Hauls: Number of matches where 4+ wickets were taken
5. 5 wicket Hauls: Number of matches where 5+ wickets were taken
6. Wickets Taken: Total wickets taken by the bowlers
7. Maiden overs: Number of overs in which no run was conceded
8. Number of dot balls: Totals balls on which no run was scored

Output: Bowler Rating

Algorithm 2: Modelling Bowler

Input: Players data from IPL 2008 to 2019 seasons, IPL Career Stats
Output: Bowler Rating
for *all players in the dataset* **do**
  /* All time Form (IPL 2008 to 2019) */
  matches = Number of innings bowled in (Min Max Normalization);
  wicketWeight = (30 * 5WicktHaul) + (20 * 4WicketHaul) + (10 * WicketsTaken);
  achievementWeight = (5 * MaidenOvers) + dotBalls;
  careerStatScore = (Bowling Average) + (Bowling Economy) + (1 / 1 + achievementWeight);
  OverallCareerScore = (matches * wicketWeight) / (careerStatScore)
end
for *all players in the recent dataset* **do**
  /* Recent Form (IPL 2018 to 2019) */
  matches = Number of innings bowled in (Min Max Normalization);
  wicketWeight = (30 * 5WicktHaul) + (20 * 4WicketHaul) + (10 * WicketsTaken);
  achievementWeight = (5 * MaidenOvers) + dotBalls;
  StatScore = (Bowling Average) + (Bowling Economy) + (1 / 1 + achievementWeight);
  RecentScore = (matches * wicketWeight) / (StatScore)
end
for *all players in the merged dataset* **do**
  /* Calculating Bowler Rating */
  OverallCareerScore=OverallCareerScore/max(OverallCareerScore);
  RecentScore = RecentScore / max(RecentScore);
  Bowler Rating = (0.35 * OverallCareerScore) + (0.65 * RecentScore);
  Perform MinMax Normalization(Bowler Rating)
end

How did you evaluate your solution/the algorithm you implemented?

Input: 15 players squad given
SP Narine, A Mishra, DJ Bravo, RA Jadeja, SL Malinga, CH Gayle, AB de Villiers, MS Dhoni, SK Raina, V Kohli, S Gopal, Yuvraj Singh, PP Ojha, MP Stoinis, DA Miller.
Venue: M Chinnaswamy Stadium
Distance measures used:

- Euclidean distance
- Manhattan distance
- Cosine distance
- Chebyshev distance
- Canberra distance

Top 3 Batsmen: SK Raina, V Kohli, CH Gayle
Top 3 Bowlers: RA Jadeja, A Mishra, SP Narine
Middle 5: MS Dhoni, AB de Villiers are 2 batsmen (batting inclined), DJ Bravo, S Gopal and SL Malinga are the 3 bowlers (bowling inclined).
Combination: 2 batting, 3 bowling

	Players	Nationality	batsman_rating	bowler_rating
0	SK Raina	Indian	100.000000	1.987405
1	V Kohli	Indian	95.546684	0.060746
2	CH Gayle	West Indian	85.805574	0.821980
3	MS Dhoni	Indian	83.395706	0.000000
4	AB de Villiers	South African	80.892032	0.000000
5	SP Narine	West Indian	20.635550	74.252936
6	RA Jadeja	Indian	19.106095	81.674093
7	DJ Bravo	West Indian	18.009862	71.519727
8	S Gopal	Indian	2.872895	70.881631
9	A Mishra	Indian	2.241545	79.323012
10	SL Malinga	Sri Lankan	0.380898	64.601210

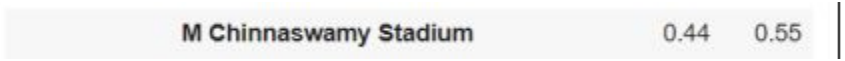
How did you evaluate your solution/the algorithm you implemented?

- The predicted Playing XI obtained from Euclidian, Manhattan and Cosine as the closeness measures provided the same result.
- Whereas, while using Chebyshev distance we find that, MS Dhoni was replaced by Yuvraj Singh and AB de Villiers was replaced by MP Stoinis.
- Although while using Canberra distance measure, along with MS Dhoni being replaced by Yuvraj Singh and AB de Villiers being replaced by MP Stoinis, SL Malinga was replaced by PP Ojha.
- On the basis of domain knowledge, we understand that the results provided by Euclidean, Manhattan and Cosine distance are similar to the real-world statistics and hence, are more accurate for the purpose of distance measure in our work

For a match between 'Royal Challengers Bangalore' and 'Sunrisers Hyderabad' at 'M Chinnaswamy Stadium' if 'Royal Challengers Bangalore' wins the toss, it would be preferable to choose to field first. This data concurs with Figure below which indicates that there is a 55% chance of winning a match at 'M Chinnaswamy Stadium' when the team chose to 'field' first against 45% when the team chose to 'bat' first.

```
team1 = "Royal Challengers Bangalore"
team2 = "Sunrisers Hyderabad"
venue = "M Chinnaswamy Stadium"
bat_or_field(df2, team1, team2, venue)
```

```
Field
```



Conclusion and Any Learnings from this project

We conclude that our model can be used as a suggestion for the team management in the real-world.
The novelty in our work is that we have included more features than pre-existing player rating models. As our model considers the all-time performance as well as the recent performances of the players (IPL season 2018 and 2019) we can expect sufficient and satisfactory prediction results to aid the team. The suggestion for toss decision is also a key factor that needs to be taken into consideration for constructing a match winning team.

Inferences about the data: The datasets had different formats for date, No balls and Super Over. They had to be cleaned and corrected.

Learnings:

- Applying knowledge gained from Data Analytics UE18CS312 course
- Scraping of data
- Learnt the importance of commitment in a project
- Team work
- Research work

Contribution by each member

The whole project was divided into 3 major sections and each section was led by different team members.

1. Pre-processing Visualization/EDA
Section head: Lavitra Kshitij Madan
Other contributors: Ritik Hariani, Arjun Chengappa, Aditya Burli

2. Modelling Algorithm and Design
Section head: Ritik Hariani
Other contributors: Arjun Chengappa, Lavitra Kshitij Madan

3. Playing XI Selection and Toss Decision model
Section head: Arjun Chengappa
Other contributors: Lavitra Kshitij Madan, Ritik Hariani