## *Introduction*

- Simulating the match, given 2 teams, batting order , bowling order and team that bats first.
- 1)Using Probability

  Simulating a match ball by ball using Probability Statistics
- 2)Using Decision Tree

  Simulating a match over by over with the help of Decision Tree

## Related work

1) Hadoop with Definitive Guide by ***Tom-White***

2) big-data-analytics-beyond-hadoop

3)Learning Spark, Lightning-Fast Big Data Analysis By ***Matei Zaharia, Holden Karau, Andy Konwinski, Patrick Wendell .***

## ALGORITHM/DESIGN

### ***Phase 1***:

K-Means clustering is applied to batsmen statistics .

10 Clusters for batsmen were created to resemble 10 batting positions .

5 Clusters for bowlers were created .

```
Players of class  6
Number of players under class 6 =  14


['AR Patel', 'DJ Bravo', 'EJG Morgan', 'GJ Bailey', 'Gurkeerat Singh', 'Harbhajan Singh', 'JP Faulkner', 'K
P Pietersen', 'KS Williamson', 'N Rana', 'SW Billings', 'Sachin Baby', 'Shakib Al Hasan', 'UT Khawaja']


Players of class  1
Number of players under class 1 =  15


['AR Patel', 'B Kumar', 'BB Sran', 'DJ Bravo', 'DS Kulkarni', 'Harbhajan Singh', 'JJ Bumrah', 'MC Henriques'
, 'MJ McClenaghan', 'MM Sharma', 'Mustafizur Rahman', 'P Kumar', 'SR Watson', 'Sandeep Sharma', 'YS Chahal']
```

## ***Phase 2***:

From the player vs player statistics a probability list of possible runs is created .

Cumulative frequency is computed for the probability list and a run is obtained by checking the interval that a random number generated between 0 and 1 falls into .

Input : Batting Lineup , Bowling Order.

Pre work – The cluster statistics from the previous phase are used to sort the players as per their skill . Average runs scored by a cluster is used for evaluating batsmen and average wickets taken by a cluster is used for evaluating bowlers.

Each batsman is given a certain confidence score between 0 and 100 . Initial confidence score per batsman is dependent on his skill (decided by the index of the cluster number in the sorted list as per mentioned criteria) . Same method is used to assign bowler confidence as well .

The confidence score of a batsman implies the probability of him not getting out . This score increases or decreases based on the outcome of the delivery being a run or a dot or a wicket . The probability of a batsman scoring higher magnitude of runs is directly proportional his confidence score. The confidence score of a bowler implies the probability of him bowling a wicket taking ball .

Algorithm :

For each delivery , a the batsman vs bowler statistics is retrieved and runs and wickets for that delivery are calculated as described earlier (based on probability statistics and confidence score) .

Challenges : Dealing with debutants

From the player vs player statistics we compute group vs group statistics. Using these we deal with 4 cases of debutant pair .

1) Existing bowler , debut batsman : An average statistic of existing bowler class against all batsmen classes is computed .

2) Existing batsman , debut bowler : An average statistic of existing batsman class against all bowler classes is computed .

3) Existing batsman , existing bowler (but new combination) : An average statistic of existing batsman class against existing bowler class is computed .

4) Debut batsman , debut bowler : An average statistic of all batsmen classes against all bowler classes is computed .

```
For batsman : [runs,dots,4s,6s,balls,strike rate , out_by]

{'SR Watson': [12, 3, 1, 0, 7, 171.42857142857142, 'Sandeep Sharma'], 'F du Plessis': [48, 5, 1, 3, 23, 208
.69565217391303, ''], 'SK Raina': [75, 13, 7, 2, 42, 178.57142857142858, ''], 'AT Rayudu': [0, 0, 0, 0, 0,
0, ''], 'MS Dhoni': [0, 0, 0, 0, 0, 0, ''], 'DJ Bravo': [0, 0, 0, 0, 0, 0, ''], 'RA Jadeja': [0, 0, 0, 0, 0
, 0, ''], 'DL Chahar': [0, 0, 0, 0, 0, 0, ''], 'KV Sharma': [0, 0, 0, 0, 0, 0, ''], 'SN Thakur': [0, 0, 0,
0, 0, 0, ''], 'L Ngidi': [0, 0, 0, 0, 0, 0, '']}

For bowler : [balls , runs , wickets , economy]

{'B Kumar': [18, 33, 0, 11.0], 'S Kaul': [12, 18, 0, 9.0], 'Rashid Khan': [12, 24, 0, 12.0], 'Sandeep Sharm
a': [18, 37, 1, 12.333333333333334], 'Shakib Al Hasan': [6, 17, 0, 17.0], 'CR Brathwaite': [6, 6, 0, 6.0]}
135
1
Match Statistics :
1st Innings :
Score : 133      Wickets : 9
2nd Innings :
Score : 135      Wickets : 1
```

## *Phase 3:*

- Used pyspark.ml.classification to import the DecisionTreeClassifier.

- We have used the following parameters to predict the outcome of runs and wickets:

- OVER NUMBER
- STRIKER
- NON STRIKER
- BOWLER
- HOME TEAM

- AWAY TEAM
- VENUE
- TOSS WON (NAME OF THE TEAM)
- TOSS DECISION (BAT OR FIELD)
- BATTING TEAM

- We have used get_dummies(from pandas) and OneHotEncoder(from sklearn) to convert the categorical .

features into numerical data

- We replace the names of the STRIKER and NON-STRIKER with their cluster numbers that we found out during PHASE1 .

- We replace the names of the BOWLER with their cluster numbers that we found out during PHASE1 .

- We convert the above data into SPARK DataFrames and then train the DecisionTreeClassifier by fitting the data consisting of 2016 IPL over by over statistics.

- For the predictions we pass the Batting Lineup,Bowling Order and the corresponding values of the features specified above.

- We are predicting the scores over by over.

- Our predictions return (RUNS,WICKETS) in that OVER.

```
7 0 P Negi SW Billings YS Chahal 24
6 0 SW Billings P Negi SR Watson 18
11 0 P Negi SW Billings Iqbal Abdulla 12
10 0 SW Billings P Negi SR Watson 6
136 4
--------
6 0 CH Gayle V Kohli  Z Khan 120
7 0 V Kohli  CH Gayle CH Morris 114
9 0 CH Gayle V Kohli  Z Khan 108
6 0 V Kohli  CH Gayle CH Morris 102
11 0 CH Gayle V Kohli  Z Khan 96
5 0 V Kohli  CH Gayle CH Morris 90
5 0 CH Gayle V Kohli  P Negi 84
9 0 V Kohli  CH Gayle A Mishra 78
11 0 CH Gayle V Kohli  P Negi 72
10 0 V Kohli  CH Gayle J Yadav 66
6 0 CH Gayle V Kohli  CR Brathwaite 60
7 1 V Kohli  CH Gayle A Mishra 54
11 0 CH Gayle AB de Villiers CR Brathwaite 48
9 0 AB de Villiers CH Gayle A Mishra 42
10 1 CH Gayle AB de Villiers P Negi 36
7 2 AB de Villiers KL Rahul  A Mishra 30
5 0 KL Rahul  SR Watson CR Brathwaite 24
11 0 SR Watson KL Rahul  Z Khan 18
9 0 KL Rahul  SR Watson CR Brathwaite 12
9 0 SR Watson KL Rahul  CH Morris 6
139 4
--------
Chasing Team Wins
```

## *EXPERIMENTAL RESULTS*

Phase 1 : Batsman RMSE – `4476.613557267382`

Bowler RMSE - `3158.079092522345`

Phase 2 : Accuracy – 60% (sample size = 10 matches)

Phase 3 : Accuracy – 73% (sample size = 11 matches)

## *FUTURE ENHANCEMENTS*

- First class cricket statistics of debutants will help in increasing the accuracy .
- Only 2016 statistics were considered for simulating 2018 matches . If we could consider 2008-2016 statistics it would increase accuracy while simulating 2018 matches .
- Decision tree model can be improved by considering more factors like weather , pitch condition , recent performance of the team etc.
- Making a Interactive web based application for the same project

## *REFERENCES*


1) https://spark.apache.org/docs/latest/ml-classification-regression.html#decision-trees

2) http://spark.apache.org/docs/latest/mllib-clustering.html#k-means

3) https://blog.scalac.io/scala-spark-ml.html


EVALUATIONS (Leave this for the faculty)

| Date | Evaluator | Comments | Score |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## *CHECKLIST*

| SNo | Item | Status |
|---|---|---|
| 1. | Source code documented | |
| 2 | Source code uploaded to CCBD server | |
| 4 | Instructions for building and running the code. Your code must be usable out of the box. Link to your gitlab account | |
| 5 | Dataset used for project uploaded. Please include a description of the dataset format. This includes input file format. | |
| 6 | Poster of your project | |
| | | |