**IPL Powerplay score prediction**

**-By Anhad Shrivastava and Sumiran Bhatnagar**

**Factors:**

In our project, we have implemented 3 different algorithms to predict the score at the end of powerplay based on 3 key factors:

1. Pitch conditions: A pitch greatly affects the batting and bowling performances during the match, with the given data and some web scraping, we are implementing a non-linear regression to predict the scores of a match based on stadium/pitch, temperature, innings and humidity
2. Batsman capabilities: An analytical prediction of the expected score based on batsman performances alone will give us our second output of prediction
3. Bowler Capabilities: An analytical prediction of the wicket taking capacity and the run conceding tendency of the bowler will give us the final predicted score that we need for our algorithm

Once we have the predicted scores due to all these three separate factors, we have implemented a multivariate linear regression which takes these three predicted scores as the input and gives and the actual score during that match as the output. These inputs will give us coefficients to add to the regression which will have the output as the actual score of that match and these triple-predictions help us to minimize the R2 value

Once we know how many wickets he takes and approx. at which over/ball and we also know the batting line up of the team, we can create a model that tells us how many balls does each batsman play in the first six overs. A probability model can also work here

Depending on their scoring rate or strike rate of a batsman in the powerplay, the number of balls he gets to play at the given strike rate and the bowler’s wicket taking tendency and the pitch’s scoring capability, we make a shrewd prediction of the score of the first 6 overs

We calculate the trike rate of a batsman in powerplay, the wicket taking rate of a bowler in powerplay, then we twerk these numbers according to the pitch conditions, and then we predict how many balls each batsman plays and the total score depending on the strike rate

Wicket taking capability = number of wickets in pp/number of overs bowled in pp

Score = (strike rate in pp \* number of balls played in pp)/100

Pitch economy: Lower economy -> rms of 1/3 of the lowest scores in pp

Higher economy -> rms of 1/3 of the highest scores in pp

Depending on weather/time of year/ teams playing (batting bowling avgs) we will decide if we use lower, general or higher economy of the pitch

**Csv file algorithms**

Create a csv file where sheet 1 contains all batsmen avg data

SNo, batsman\_name, year, pp\_score, pp\_balls, pp\_dismissal, death\_score, death\_balls, death\_dismissal, total\_score, total\_balls, total\_dismissal,

For every ball

1. if the name of the striker is not part of our batsmen list, then add him to the list
2. for the striker batsman
   1. increase total\_score value by runs\_off\_bat value
   2. increase the total\_balls by 1
3. if the integral part of over is between 0-5
   1. add the runs\_off\_bat value to the pp\_score of the batsman
   2. increase the balls played in pp value by 1
4. if the integral part of the over is between 15-19
   1. add the runs\_off\_bat value to death\_score of the batsman
   2. increase death\_balls by 1
5. if wicket\_type is not empty then for the player\_dismissed
   1. add total\_dismissal by 1
   2. if the integral part of over is between 0-5 add pp\_dismissal by 1
   3. if the integral part of the over is between 15-19 add death\_dismissal by 1

in the same csv file where sheet 2 contains all bowlers avg data

SNo, bowler\_name, year, pp\_score, pp\_balls, pp\_wickets, death\_score, death\_balls, death\_wickets, total\_score, total\_balls, total\_wickets

For every ball

1. if the decimal part of the ball is 0 and the bowler is not part of our bowler list, then add him to the list
2. for the bowler
   1. increase total\_score value by (runs\_off\_bat+extras) value
   2. total\_balls by 1
   3. if the wicket\_type column is not empty, and not runout, add 1 to total\_wickets
3. if the integral part of over is between 0-5
   1. add the (runs\_off\_bat+extras) value to the pp\_score of the bowler
   2. increase the balls played in pp value by 1
   3. if the wicket\_type column is not empty, and not runout, add 1 to pp\_wickets
4. if the integral part of the over is between 15-19
   1. add the runs\_off\_bat+extras value to death\_score of the bowler
   2. increase death\_balls by 1
   3. if the wicket\_type column is not empty, and not runout, add 1 to death\_wickets

**Batsmen Model**

1. Calculate runs scored by batsman per over even if he was off-strike for the entire over
2. Calculate probability of the batsman getting out in some over
3. Create the model:
   1. Let probability of batsman B1 getting out in over O1 be p1.
      1. Then calculate the runs scored by batsman B2 and B3 in the remaining overs using the probability distribution table
      2. If the successive batsman also gets out by probability, create a team probability in general of scoring by the other batsmen of the team at loss of 2 or 3 wickets. Team probability data will be stored
      3. Limit the wickets lost to 3 in power play while calculating probability.
   2. Let probability of batsman B2 getting out in over O1 be p2
      1. Then calculate the runs scored by batsman B1 and B3 in the remaining overs using the probability distribution table
      2. If the successive batsman also gets out by probability, create a team probability in general of scoring by the other batsmen of the team with loss of 2 or 3 wickets. Team probability data will be stored
      3. Limit the wickets lost to 3 in power play while calculating probability.
4. The sum of the runs scored by the batsmen capabilities alone is the batsmen model
5. For better results, we can calculate batsmen capabilities by home ground/away ground or by home ground, away batting pitch, away bowling pitch

Formula:

P(k, n)-> probability the batsman k will get out in over n

R(k, n)-> Runs scored by batsman k in over n

S(k1, k2, o)-> Runs scored by batsman k1 and k2 if they start playing from over n+1

kc = the next batsman that comes if one of the previous get out

α = The score reduction factor due to loss of wicket

S(ka, kb, o) =

while o<6

if kc<=k5 (5th batsman)

P(ka, o)\*[ S(kb, kc, o+1) + αR(ka, o) +R(kb, o)] + P(kb, o)\*[S(ka, kc, o+1) + R(ka, o) + αR(kb, o)] + P(ka, o)\*P(kb, o)[S(kc, kd, o+1)+ α(R(ka, o)+R(kb, o))] +{1-P(ka, o)-P(kb, o)-P(ka, o)\*P(kb, o)}\*[S(ka, kb, o+1) + R(ka, o) + R(kb, o)]

o++

S(b1, b2, 0) = P(b1, o)\*[ S(b2, b3, o+1) + α(R(b1, o) +R(b2, o))] + P(b2, o)\*[S(b1, b3, o+1) + α(R(b1, o) + R(b2, o))] +{1-P(b1, o)-P(b2, o)}[S(b1, b2, o+1) + R(b1, o) + R(b2, o)]

Predicted score = S(b1, b2,0)

**Bowler Model**

1. Calculate runs conceded by bowler in each over (1-6) per over
2. Calculate the probability of the bowler to take a wicket in some over
3. Calculate the economy of the bowler after taking a wicket
4. Calculate the probability of the bowler to take 2 or more wickets in pp
5. Calculate the economy of the bowler when he takes 2+ wickets in pp

Let us take the bowling line up of the bowlers as 1,2,1,3,2,3 or 1,2,1,2,3,2 or 1,2,1,2,1,3

We can either team-wise establish the bowling line up of the powerplay or take these general line ups to create our code

Bn is the bowler bowling in the nth over

Let P(b, n) be the probability that a bowler takes a wicket in over n

Since probability to take 1 wicket in 6 balls is P(b, n), the probability to take the second should be P(b, n)\*5/6

Overall probability of 2 wickets in an over is P2(b, n) \*5/6 and for 3 wickets it is P3(b, n) \*20/36 and so on

Assume 1 wicket per over (will do 2 after algo for 1 is created)

W= number of wickets lost

C(bn, n, w) = number of runs conceded by bowler b in over n when w wickets are already taken

Lookup that bowler b1 and for that bowler find r(|n||w|) for that bowler if w>2, put w =2

T(n, w) = total score conceded by bowlers from over n to over 6 (n-6) when w wickets are taken

T(0, 0) = Starting state

T(1, 0) = P(b1, 1)\*[C(b1, 1, 0) +T(2,1)] + P(b1,2)\*[C(b1,1,0)+T(2,2)] + (1-P(b1,1)-P(b1,2))\*[C(b1,1,0 + T(2,0)]

=>T(1, 0) = C(b1,1,0) + P(b1,1)\*T(2,1) + P(b1,2)\*T(2,2) + (1-P(b1,1)-P(b1,2))\*T(2,0)

In variables:

T(n, w) = C(bn, n ,w) + P(bn, 1)\*T(n+1,w+1) + P(bn,2)\*T(n+1,w+2) + (1-P(bn,1)-P(bn,2))\*T(n+1,w)

Calling function,

PrdeictedRunsByBowlerPerformance (input variables)

If n<=6

Set an order for bowlers as they bowl through the overs based on the team’s bowling preferences

T(0, 0)

Else return 0

**Pitch Model**

1. Pitch economy constant (Pec) = economy of bowler on that pitch /economy of the bowler
2. => economy of a bowler on a pitch = Pec \* economy of bowler
3. Pitch economy wicket constant (Pewc) = wickets of a bowler per match on that pitch/wickets of a bowler per match in general
4. =>

Data Transfer Algorithm for batsman and bowler statistics: (**Almost done)**

bs= variable for Striker, bns = variable for non-striker, bw = variable for bowler

k -> integer part of the over, d-> decimal part of the over

Begin with bs=””, bns=”” and bw=””

blist -> list with bw and bns

i->innings

wickets=0

Set i=0, w=0

For every ball:

1. if 2015<= Year<=2019
2. k=integral part of the over
3. if k <6
4. bs=striker, bns=non-striker, bw=bowler
5. innings change data
   1. if i != innings => innings changed
   2. w=0
   3. i=innings
6. over change data:
   1. if the decimal part of the over is 1 => over changed
   2. if bw in bowler sheet
   3. m(|k+1||w|) +=1
   4. if bs in batsman sheet
   5. m(|k+1|) +=1
   6. if bns in batsman sheet
   7. m(|k+1|) +=1
   8. blist = {bs, bns}
7. Player change data
   1. If decimal part of over is not 1
   2. If bw != bowler => bowler changed
      1. if bw in bowler data sheet
      2. m(|k+1||w|) +=1
      3. bw = bowler
   3. Change in batsman
      1. if bs not in blist=> striker changed
      2. if bs in batsman sheet
      3. m(|k+1|) +=1
      4. if bns not in list=>non-striker changed
      5. if bns in batsman sheet
      6. m(|k+1|) +=1
      7. blist = {bs, bns}
8. Update batsman data
   1. If bs in batsman sheet
   2. t(|k+1|) += runs\_off\_bat
9. Update bowler data
   1. If bw in bowler sheet
   2. t(|k+1||w|) += runs\_off\_bat + extras
10. if wicket\_type is not empty
    1. w+=1
    2. if player\_dismissed in batsman sheet
    3. w(|k+1|) += 1
    4. if bw in bowler sheet
    5. w(|k+1|) += 1

ToDo (22/04/2021)

1. Transfer the data (**almost done)**
2. Find a website to get information about weather and humidity (done)
   1. Use web scraping on that sire to get relevant data (**Sumi)**
3. Execute non-linear multivariate regression to find score prediction x3 based on pitch u1, innings u2, temperature u3 and humidity u4 ie for each stadium, there is a different prediction (**Sumi+Ann)**
4. Execute Multivariate linear regression to calculate predicted score y based on predicted score due to batsmen performance x1, bowler performance x2 and stadium conditions x3 (**Sumi+Ann)**
5. Test on training data (**Sumi+Ann)**
6. Test on test data (**Sumi+Ann)**
7. Use various %ages of training and test data to make suitable algorithms (**Ann)**

Dikkat:

1. Need to transfer more data than coded
   1. Batsmen and bowler data (2015-2019) | 2020-21 ---22
   2. Pitch/Stadium data (2010-2017) | (2018-19,2021) ---22
      1. Pitch-wise predicted score model for the 6 given pitches based on innings, weather, humidity
      2. [u1=innings, u2= weather, u3=humidity]
      3. [x3=predicted score]
      4. non-linear regression

**To create input & output test file**

Beginning values -> Ven=””; inn=0; batting\_team=””; bowling\_team=””; bat\_list=[]; bowl\_list=[]; score=0

For every ball:

if 2019<= year <= 2021 and innings<3:

1. If inn!=innings:
   1. #Store previous data
      1. Input sheet: C1->venue, C2->innings, C3->batting\_team, C4->bowling\_team, C5->space separated bat\_list, C6->space separated bowl\_list
      2. Output sheet: score
   2. #reset data
      1. Ven=venue; inn=innings; batting\_team=batting\_team; bowling\_team=bowling\_team; bat\_list=[]; bowl\_list=[]; score=0
2. elif int(ball)<6:
   1. #keep updating data
   2. score+=runs\_off\_bat + extras
   3. if striker not in bat\_list, add striker to bat\_list
   4. if non\_striker not in bat\_list, add non\_striker to bat\_list
   5. if bowler not in bowl\_list, add bowler to bowl\_list

**To store pitch-wise data for a score prediction**

First, create a sheet with 6 stadium names

For each stadium we need innings, temperature, humidity and score during that match

For each row:

1. Take innings as inn
2. Take the date as date and use that date to find temperature and humidity values of the day and store these values in the temperature and humidity counter
3. Take sum of scores for first 6 overs as score (template available from other codes)

If batsman in dictionary -> proceed

Else add batsman to the empty row and to a dictionary -> then proceed

Decision making for data usage:

1. i/p o/p -> 2019,2021 (test)
2. player data ->2014-2018(train), 2019, 2021(test 23%)
3. pitch data 2012-2018 (train), 2019,2021(test 17%)

New tasks:

1. Predictions based on location, innings, temperature and weather (Sumiii)
   1. Assign an average pp score for each stadium since 2018
   2. Innings will affect as required, try with and without innings
   3. Let regressor decide coefficients for temp and humidity
2. Predictions based on recent form (3)
   1. Take the recent form (ipl 2021) only of each player to predict their performance
3. Bowler predictions based on economy and bowling order (Ann)
   1. Save pp economy of bowler and use a bowling order to define overs
   2. We know the number of wickets. We can use that to tweak the expected score depending on wicket loss
   3. Use simple pp economy sum and a regressor to work through values
4. In test\_data create column: no of wickets lost and use this column to

Factors:

1. Batsman performance
2. Bowler performance
3. Wicket loss affect
4. Pitch/stadium conditions -> temp, humidity, innings, avg pp score of that stadium
5. Team batting -> all scores in a list
6. Team bowling -> all scores in a list

🡪 **a(batting)** + b(bowling) + [c(stadium avg) + d(temperature) + e(humidity)] + f(batting team) + g(bowling team) + h(PvP history runs) + i(form of batsman) + j(form of bowler)

Stadium\_avg= avg pp scores in a stadium iin 2018,19,21 ->1avg value

Batting team= avg runs scored by batting team in pp

Bowling team = avg runs conceded by bowling team in pp