INDIA'S CHANCES OF WINNING ICC WORLD CHAMPIONSHIP IN 2017

BACKGROUND

ICC World championship is an ICC tournament played between top eight teams in ODI ranking. The teams are divided into 2 groups and within each group the four teams play in a league fashion and the top 2 teams of each groups qualify for semi-finals.

There are many methods of analysing cricket matches but it has been found that in certain cases the methods prove to be biased and hence a new concept should be developed.

PROBABILISTIC MODEL USED

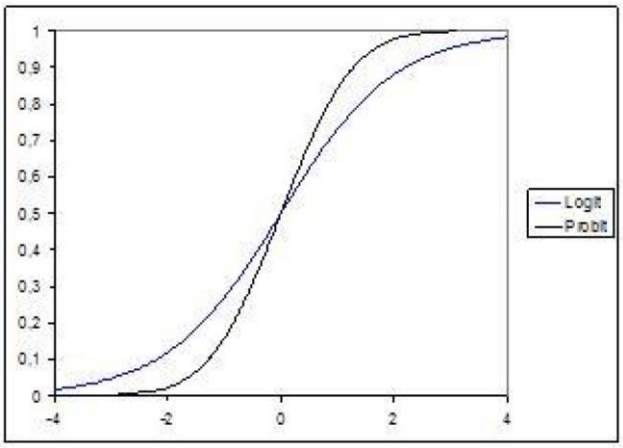
LOGISTIC REGRESSION MODEL-

It is basically a modified version of probabilistic model. It provides output between 0 and 1 and never goes out of range as in the case of linear probabilistic model.

It is helpful when the dependent variable is binary i.e. when its value is either 1 or 0. It is also one of the predictive analysis. This model is basically used to explain the relationship between one dependent binary variable with one or more independent variable.

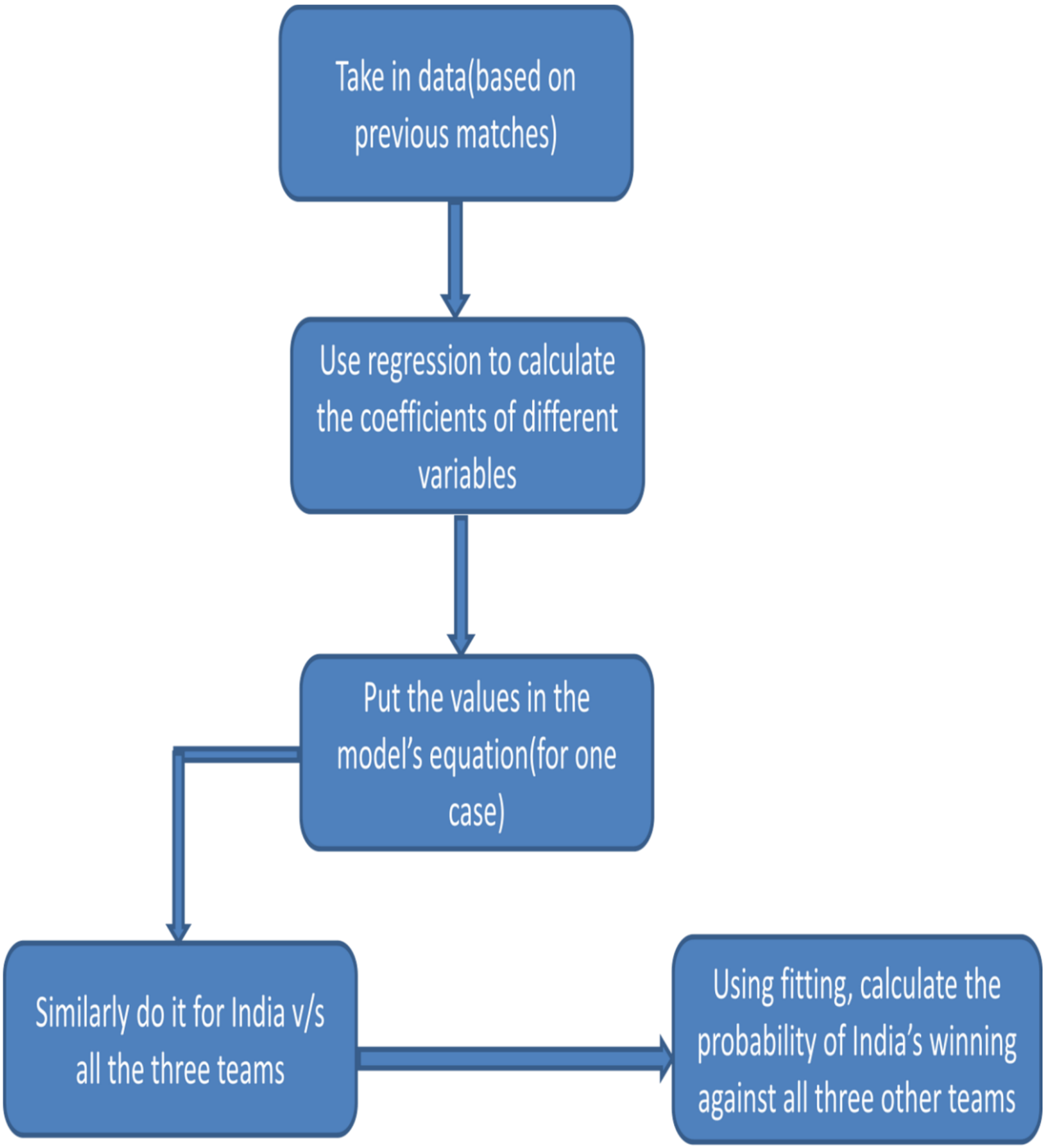
In this model it is assumed that if the value comes out to be less than 0.5, it is equal to 0 otherwise it is equals to 1.

Instead of logit model, one can also use probit model. Both use the same concept, the only difference are in the equations they use. Below is a graph for logit model and probit model for some case.



http://www.statisticssolutions.com/what-is-logistic-regression/

BLOCK DIAGRAM



CODE

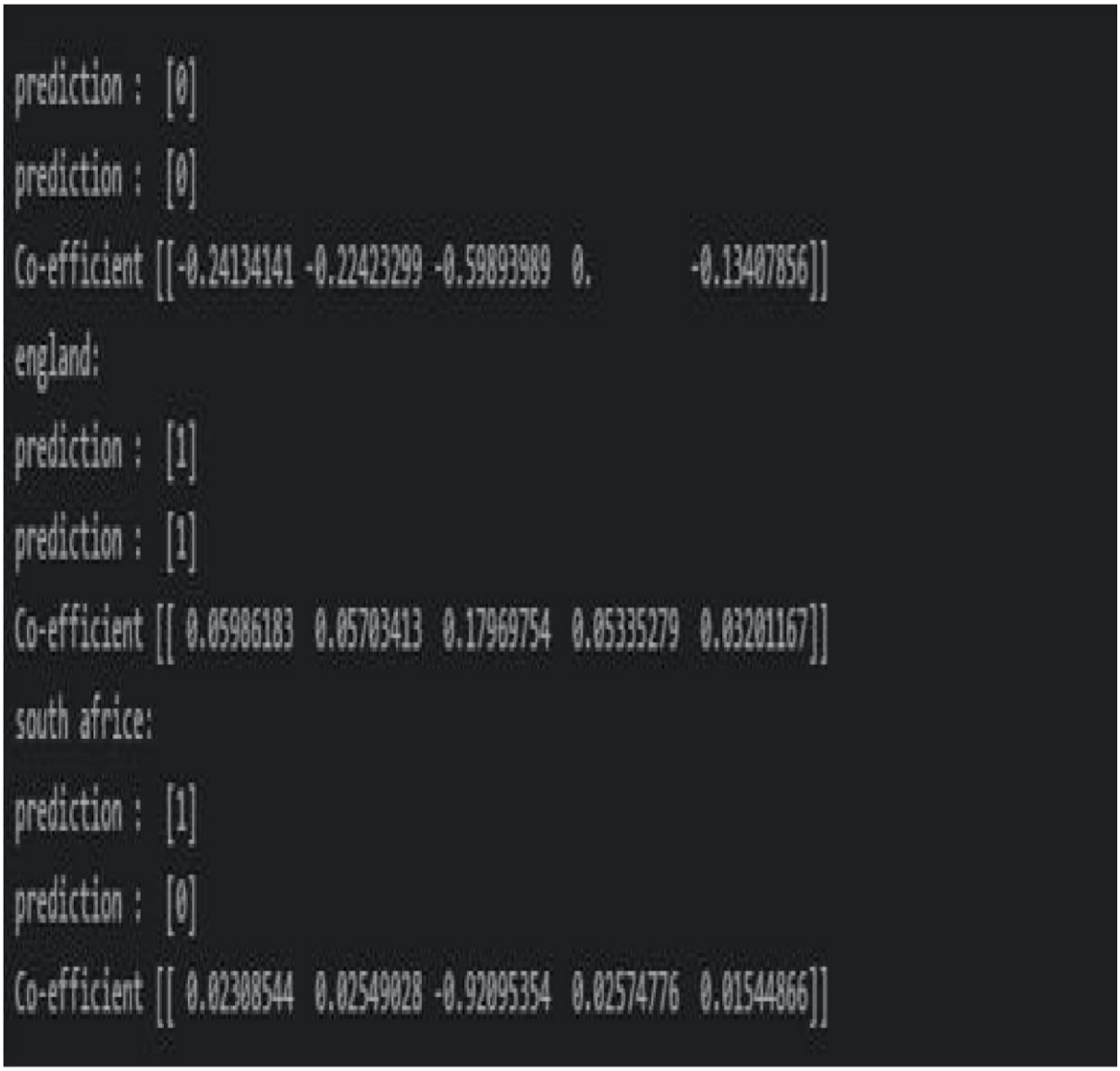
```
from sklearn import linear_model import
numpy as np

#australia:
x =
np.array([[1.08,1.00344,1,0,0.6],[1.08,1.00344,1,0,0.6],[1.08,1.00344,0,0,0.6],[
1.08,1.00344,1,0,0.6],[1.08,1.00344,0,0,0.6]]) Y
= np.array([0,0,0,0,1])
model = linear_model.LogisticRegression()
model.fit(x,Y) print ('australia:') a =
[1.08,1.00344, 0, 0, 0.6] print ('prediction :
',model.predict(a)) b = [1.08, 1.00344, 1, 0,
0.6] print ('prediction : ',model.predict(b))
print ('Co-efficient',model.coef_)

#england:
print ('england:') x
=
np.array([[1.122,1.069,1,1,0.6],[1.122,1.069,0 ,1,0.6],[1.122,1.069, 0,1,0.6],[1.12
2,1.069,1,1,0.6],[1.122,1.069,1,1,0.6]])
Y = np.array([1,1,0,0,1])
model = linear_model.LogisticRegression()
model.fit(x,Y) a = [1.122 ,1.069, 0, 0 ,0.6]
print ('prediction : ',model.predict(a)) b =
[1.122 ,1.069 ,1, 0, 0.6] print ('prediction :
',model.predict(b)) print ('Co-
efficient',model.coef_)

#sa: print ('south
africa:') x =
np.array([[0.8966,0.99,1,1,0.6],[0.8966,0.99,0,1,0.6],[0.8966,0.99,1,1,0.6],[0.89
66,0.99,0,1,0.6],[0.8966,0.99,1,1,0.6]])
Y = np.array([0,1,0,1,0])
model = linear_model.LogisticRegression()
model.fit(x,Y) a = [0.8966, 0.99, 0, 0, 0.6]
print ('prediction : ',model.predict(a))
b = [0.8966, 0.99 ,1 ,0 ,0.6] print
('prediction : ',model.predict(b)) print
('Co-efficient',model.coef_)
```

RESULT



INFERENCE

From our result we could derive the inference that:

- If India plays against Australia in the finals then it will lose irrespective of the result of toss .

- If India plays against England in the finals of champions trophy then it will win irrespective of the result of toss.
- If India plays against South Africa in the finals of champions trophy then it will win if India wins the toss otherwise India will lose.

FUTURE SCOPE

The approach which we have used doesn't take into consideration the factors such as rain , pitch , home ground, etc. Though some of these factors doesn't always affect the game but many of the factors play a crucial part in the analysis of the outcome.

There are many other methods such as Duck-worth Lewis method which proves to be biased at times and hence there should be an approach to analyse the match based on the past performance of the team or a method which effectively analyses each factor and doesn't prove to be biased.

REFERENCES

- http://data.princeton.edu/wws509/notes/c3.pdf
- http://www.icmis.net/icmis15/icmis15cd/pdf/S5044-final.pdf
- https://pdfs.semanticscholar.org/2e30/e4b8321718f69933b9b29993e3df95c12cd9.pdf