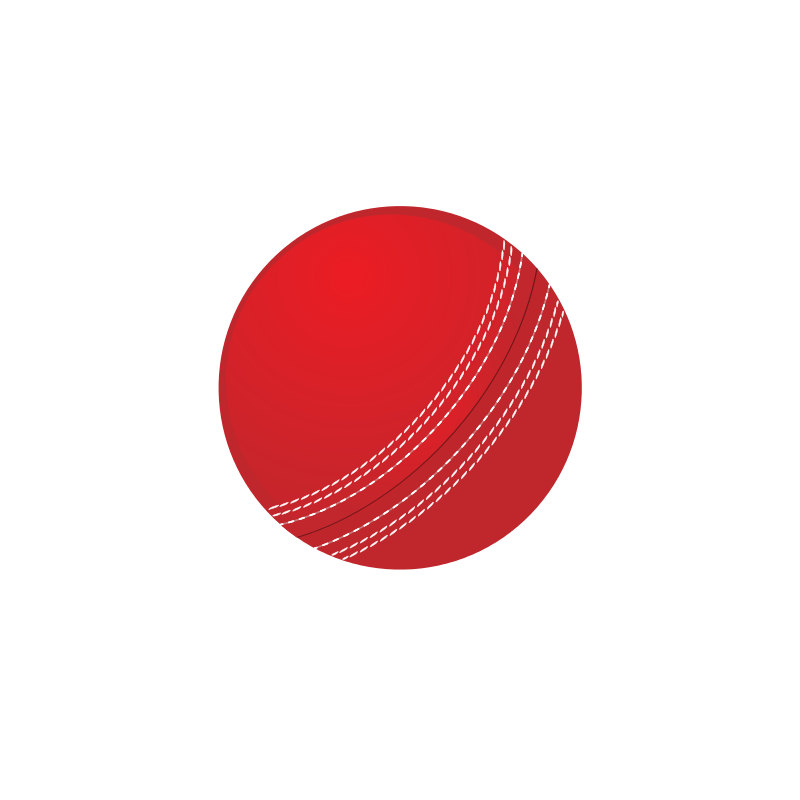


Data Mining Project



***Cricket Sco re Prediction***

*Made By* :Aryaman Kumar Chandan (2019UCO1668)

Vijay Pal (2019UCO1676)

Aaryan Raj Sarda (2019UCO1684)

*Branch/Section* :COE/3

*Semester* :V

Subject Code : COCSC16

*Teacher* : Ms. Rashmi Chaudhary

*Table of Contents*

[**Abstract**](#_f8tazk5o2n8f) **3**

[Introduction](#_chp54ibey1ou) **3**

[Overview](#_x5rgu4e9i48n) 3

[Motivation](#_n6x2imnpy164) 3

[**Methodology**](#_gs7lyttt70vm) **4**

[**Performance Metrics**](#_3yva3ygc9p9f) **9**

[Results & Analysis](#_pbaat8o9faif) **10**

[**Conclusion & Scope**](#_atbor9gc8v4g) **13**

[References](#_ge9zhgj9siw0) **13**

# 

# 

# Abstract

Cricket is a very popular sport in many countries, especially India. The outcome of a match depends on various factors like a home ground advantage, performances in the past, experience, performance at the specific venue, performance against the specific team, and the current form of the team and the player. In this work, a model has been proposed that predicts the score of the team based on various factors such as wickets fallen, runs, overs, etc. This method has been implemented using various Machine Learning algorithms and the conclusion is made on the best algorithm.

# Introduction

### Overview

Cricket as a professional sport is being played in many countries around the globe with a vast number of domestic and international tournaments taking place daily. Cricket comprises two teams having 11 players each. It is played in various formats- T20, ODI, and Test.

The result is either a win, loss, or a tie. This game is extremely unpredictable because at every stage of the game the momentum shifts to one of the teams between the two. Considering all these scenarios of this unpredictable game, there is a huge interest among the spectators to do some predictions either at the start of the game or during the game. Many apps have also come up like Dream 11 which allows spectators to bet on their teams and make money.

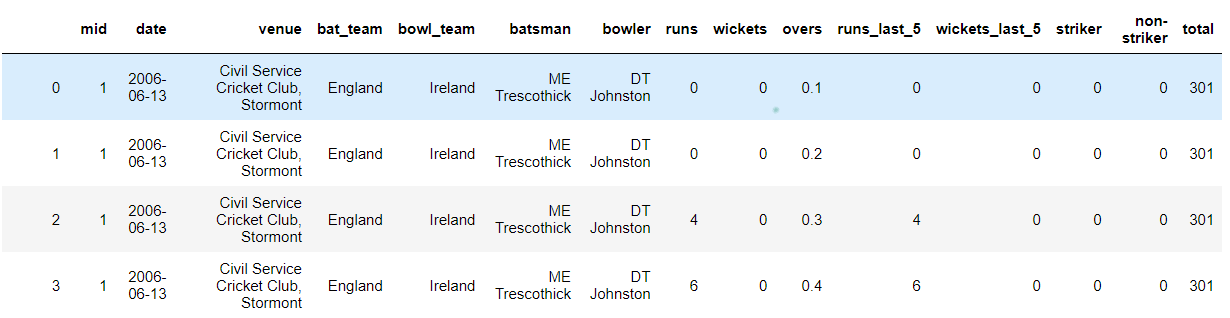
### Motivation

Taking into account the increasing enthusiasm, money, and effort involved in this sport, this project aims at studying the problem of predicting the game results at every point in the game based on statistics and data available from the data set. The game prediction problem takes into consideration the runs scored by the team ball by ball, wickets fallen, runs scored in the last 5 overs, wickets fallen in the last 5 overs, batsman on the striker, and non-striker’s end.

# 

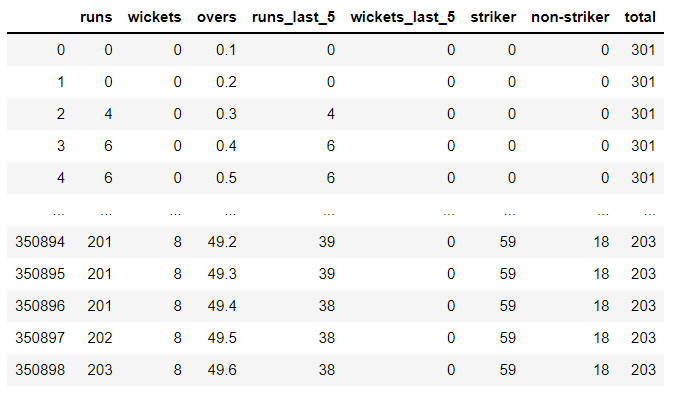
# Methodology

1. *Data Collection:*   
   We have collected the ball by ball dataset from the datasets available on [cricsheet](http://cricsheet.org/downloads) in YAML form and converted it into CSV so that we can use the pandas library to read the data.



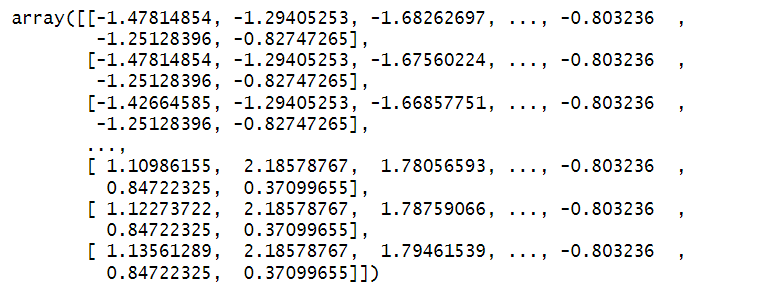
1. *Data Cleaning:*   
   In this step, we will be considering only those attributes which will have an impact on the runs scored by a team. We removed attributes mid, date, ‘Venue’, ‘Team Names’, ‘Player Names’.

After doing this our dataset looks like



1. *Data Preprocessing:*

We have applied Feature Scaling of the attributes using standard Scaler available from sklearn library.



We get the normalized values of the attributes.

*STANDARDIZATION*

We try to make parameters consistent so that we get accurate results.

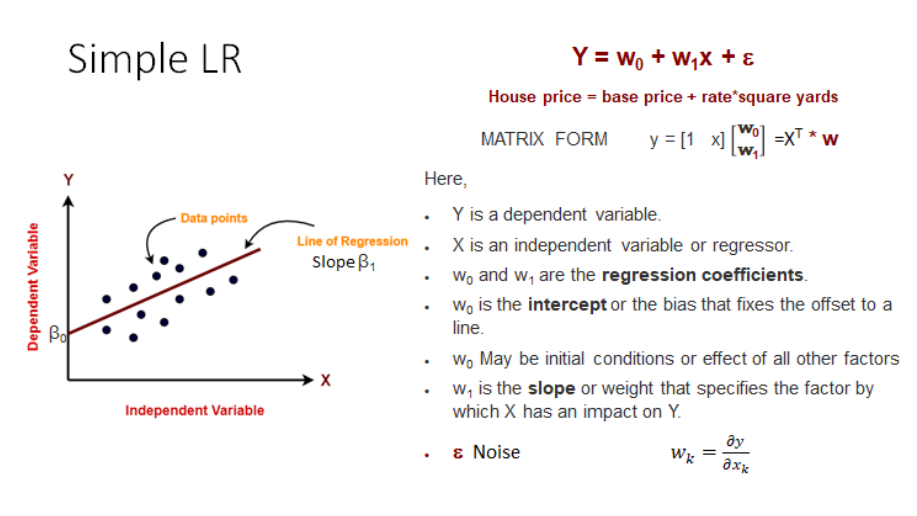
It is useful when data has varying scales. So we rescaled the attributes so that they have a mean value of 0 and a standard deviation of 1.

1. *Data Splitting:*   
   We used the scikit-learn library to split the data into train and test data. Test data is 30% of the total dataset.
2. *Model Generation:*   
   We will be using the Linear Regression model, Lasso Regression, Ridge Regression, Random Forest model for the prediction. The model with the highest accuracy will be selected for the prediction.
3. *Final Prediction:*   
   Finally, the data will be passed through the model and then the user inputs will be taken. After getting the user inputs and matching them with the historical data we will be predicting a range of the score i.e., from lower bound to the upper bound.

***Regression Models Used***

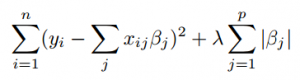
*Linear regression*

Linear regression is the simplest algorithm to train the model based on a set of independent attributes giving an outcome (dependent attribute).

****

*Lasso Regression*

Lasso solutions are quadratic programming problems, which uses a cost function to calculate the error between predicted and actual values. Then a hyperparameter is introduced which is called a learning rate and the algorithm tries to minimize cost function by tuning the learning rate. The equation is :



Which is the same as minimizing the [sum of squares](https://www.statisticshowto.com/residual-sum-squares/) with constraint

Σ |Bj≤ s (Σ = [summation notation](https://www.calculushowto.com/calculus-definitions/summation-notation-sigma-function/)).

Some of the βs are shrunk to exactly zero, resulting in a regression model that’s easier to interpret.

*Ridge Regression*

Ridge [regression](https://www.mygreatlearning.com/blog/what-is-regression/) is used to solve the problem of multicollinearity. This method performs L2 regularization. It is used when there are higher predictor variables the number of observations. Its cost function is given by:

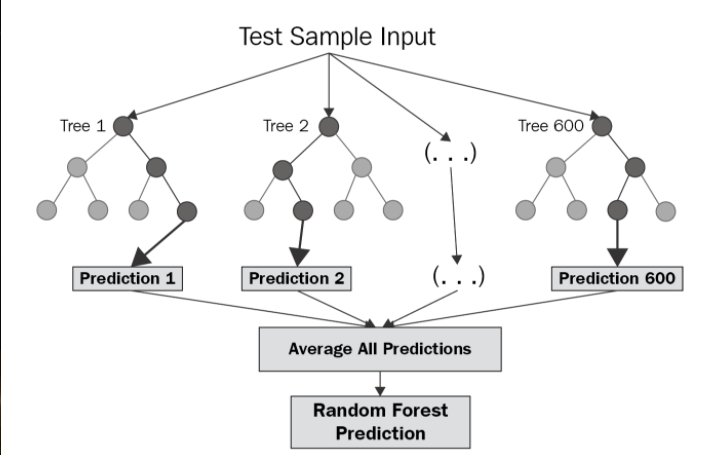
***Min(||Y – X(theta)||^2 + λ||theta||^2)***

Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function. So, changing the values of alpha can control the penalty term. The higher the values of alpha, the bigger is the penalty and therefore the magnitude of coefficients is reduced.

* It shrinks the parameters. Therefore, it is used to prevent multicollinearity.
* It reduces the model complexity by coefficient shrinkage.

*Random Forest Regression (Most accurate for our model)*

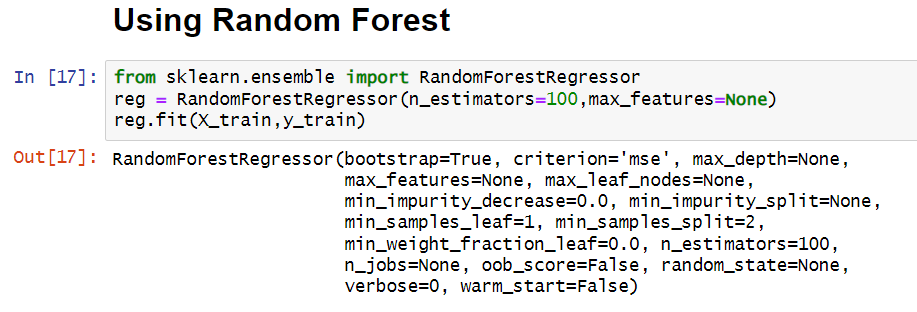
Random Forest Regression is a supervised learning algorithm that creates multiple decision trees from the dataset and predicts the outcomes based on the average of all the combinations. The representation is as follows:

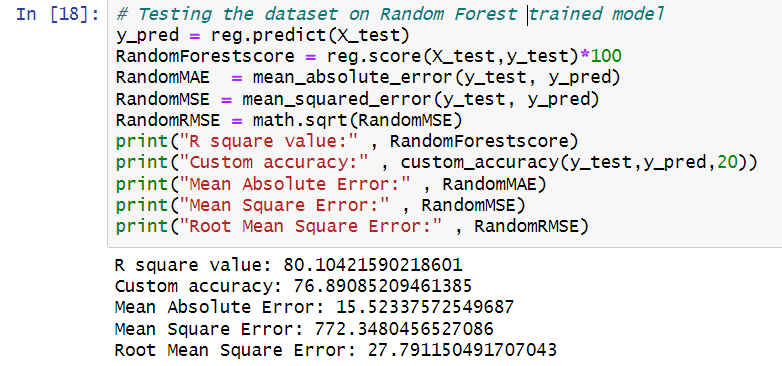


The algorithm works in the following way:

1. It randomly picks k data points from the training set and builds a decision tree with these k data points.
2. Choose the number N of trees you want to build and repeat steps 1.
3. For a new data point, make each one of your N-tree trees predict the value of y for the data point in question and assign the new data point to the average across all of the predicted y values.

Code :





# 

# Performance Metrics

For evaluation of the models, we have used evaluation metrics. We are using MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error), and R squared value as evaluation metrics.

* MAE: It is the average of the difference between the original and the predicted values.

where : size of dataset

: absolute errors

* MSE: It is the measure of the distance between the regression line and the original values and squares them to remove the negative signs.

where : size of dataset

: observed value

: predicted value

* RMSE: It is the square root of the MSE value.

If the MAE, MSE, RMSE values are closer to zero then we can say that the model is well fitted to the data.

* R squared value: It determines how close the regression line is to the original value. R squared value closer to 1 indicates that the model is well fitted to the data. In evaluation metrics, the R squared value is used to measure the accuracy of the model.

Where,   
 = coefficient of determination.   
 = sum of squares of residuals.   
 = total sum of squares

Where,   
 = mean value of a sample

# Results & Analysis

We calculated the evaluation metrics for Linear Regression, Random Forest Regression, Lasso Regression, and Ridge Regression. Following are the graphs and scores of evaluation metrics for the models used.

***Linear Regression***

*R square value: 53.02440531360795*

*Custom accuracy: 59.269497482663624*

*Mean Absolute Error: 31.816873508146074*

*Mean Square Error: 1823.5777273736612*

*Root Mean Square Error: 42.70336904008466*

***Lasso Regularization***

*R square value: 51.95992074294465*

*Custom accuracy: 70.46451980621259*

*Mean Absolute Error: 32.302955509869946*

*Mean Square Error: 1864.9006817109873*

*Root Mean Square Error: 43.18449584875326*

***Ridge Regularization***

*R square value: 53.02440441750236*

*Custom accuracy: 70.91384060036098*

*Mean Absolute Error: 31.81688295211993*

*Mean Square Error: 1823.5777621601962*

*Root Mean Square Error: 42.703369447389*

***Random Forest***

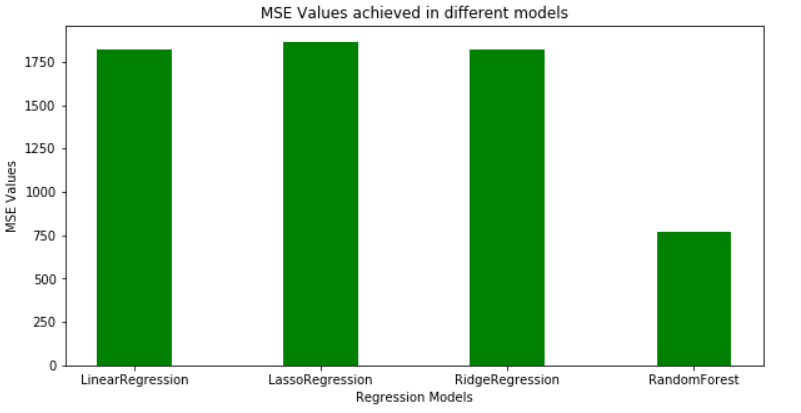
*R square value: 80.08610065874623*

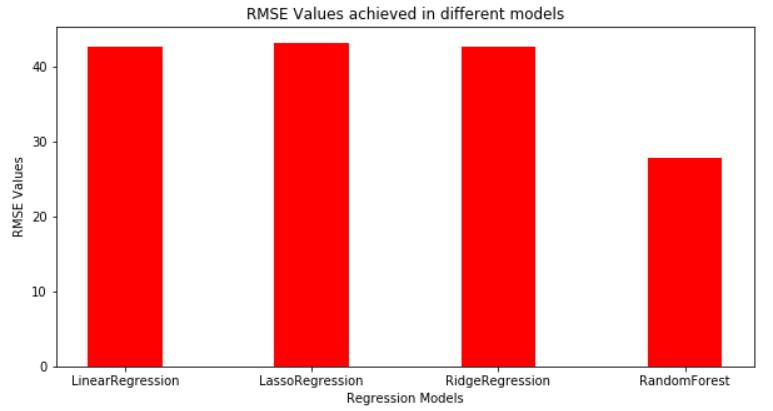
*Custom accuracy: 77.08463949843261*

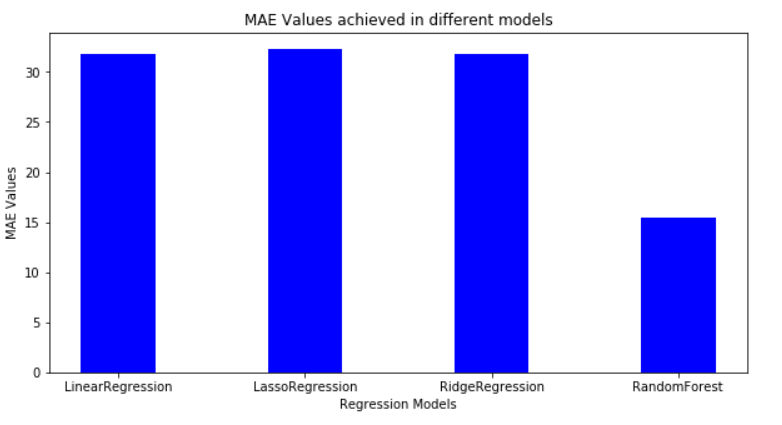
*Mean Absolute Error: 15.50745263502313*

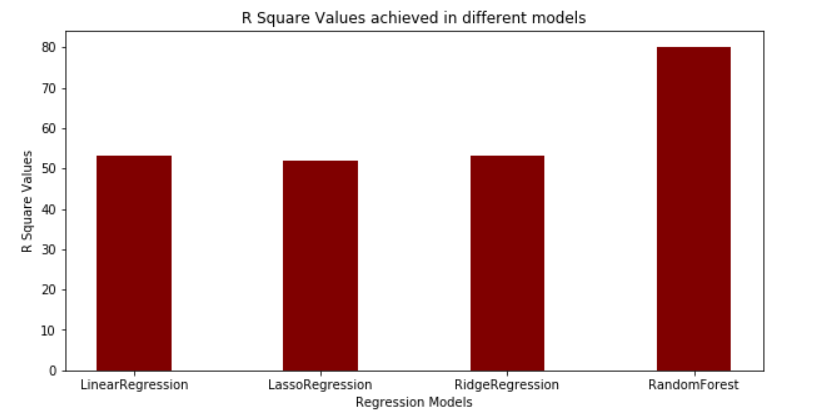
*Mean Square Error: 773.0512736732*

*Root Mean Square Error: 27.80379962654745*









*--------------------------------------------------------------------------------------------------*

***Tools Used***

*Coding - Python Jupyter Notebook*

*Libraries Used: Scikit-Learn, pandas*

*Data Cleaning: Using Python codes*

# Conclusion & Scope

Random Forest works best among the given algorithms with an accuracy of about 80%. So, it will be used to predict the scores of cricket games.

Reasons for other models being less accurate may be overfitting of data, inappropriate learning rate etc.

In future, this model can be used for various purposes like analysis of a team, specific player etc. and can be developed for cricket teams. The accuracy of the model can be further enhanced with the collection of data such as wind speeds, stats of players in official matches etc.

# References

Dataset from: [Cricsheet](https://cricsheet.org/downloads)

ML Algo: <https://scikit-learn.org/stable/>

Info: <https://en.wikipedia.org/wiki/Cricket>