

 DATASET INPUT FORMAT FILE STRUCTURE REQUIREMENTS.TXT PLAYERS LIST SAMPLE INPUT FILES

## Problem Statement:

Given certain input parameters regarding an innings of a T20 cricket match, predict the total runs scored by the batting team at the end of 6 overs.

- Programming Language to be used: only Python 3.8
- Maximum size of zip file allowed: 50MB
- Maximum permitted execution time: 10secs.
- We will be executing your code on 8GB RAM machine - so please ensure that the code submitted will execute within these constraints for us to get the output.

## Recommended packages

- The python environment that will be used for evaluation is given in the requirements.txt file - [click here to download](#)
- Please note that we will not support the use of any other packages (or versions) that are not present in this file. You may use different packages for your training phase. However, kindly ensure that your test code during submission does not depend on packages other than the mentioned ones in the [requirements.txt](#) file
- In case of errors such as packages not found, the submission will be disqualified.
- It is also to be noted that we will not be using GPUs during our tests. Therefore, please refrain from using packages that depend on GPU computing (NOTE: tensorflow is supported; tensorflow-gpu is not).

## Input Data:

We are providing the link to the Dataset (Source: cricsheet.org) which contains historic data of T20 matches that have occurred in the past. This dataset may be used by the candidate teams to train an ML model or come up with a data analytics based algorithm that can perform the required prediction as mentioned above.

## Dataset download:

CSV Files for IPL Matches: [Click here to download](#)

## Data Description (quoted directly from Cricsheet page:

<https://cricsheet.org/downloads/>):

The CSV File provided follows the "NEW" format.

The "new" format consists of a single row for each delivery in a match (or group of matches).

The first row of each CSV file contains the headers for the file, with each subsequent row providing details on a single delivery. The headers in the file are:

- |                |                |
|----------------|----------------|
| • match_id     | • bowling_team |
| • season       | • striker      |
| • start_date   | • non_striker  |
| • venue        | • bowler       |
| • innings      | • runs_off_bat |
| • ball         | • extras       |
| • batting_team | • wides        |

• noballs

- penalty
- wicket\_type
- player\_dismissed
- other\_wicket\_type
- Other\_player\_dismissed

Most of the fields above should, hopefully, be self-explanatory, but some may benefit from clarification

- **"innings"** contains the number of the innings within the match. If a match is one that would normally have 2 innings, such as a T20 or ODI, then any innings of more than 2 can be regarded as a super over.
- **"ball"** is a combination of the over and delivery. For example, "0.3" represents the 3rd ball of the 1st over.
- If a wicket occurred on a delivery then **"wicket\_type"** will contain the method of dismissal, while "player\_dismissed" will indicate who was dismissed.
- There is also the, admittedly remote, possibility that a second dismissal can be recorded on the delivery (such as when a player retires on the same delivery as another dismissal occurs). In this case **"other\_wicket\_type"** will record the reason, while **"other\_player\_dismissed"** will show who was dismissed.

## Test Case Input:

The following will be provided as input test case data:

venue, innings, batting\_team, bowling\_team, batsmen who batted during the first 6 overs, bowlers who bowled during the first 6 overs.

The candidates may consume the data and preprocess or convert them in any manner for making it work as per their developed model.

## Input format

Input will be fed from a CSV file testInput.csv. [Click here to download the file.](#)

## Evaluation:

Points scored by each team = R2 error value (sum of square of error) between the actual score and predicted score

(The lower the points, better is the prediction outcome).

## Submission:

### File structure

[SubmissionFormat.zip](#) contains two python source files main.py and predictor.py.

(The lower the points, better is the prediction outcome).

**predictor.py:** In the sample format, predictor.py contains a definition predictRuns().

**main.py:** This is a test file. It is recommended to NOT edit this file. Once you implement the predictor.py, you can test the functionality of your implementation.

## How to implement:

You can customize the implementation of predictRuns() in predictor.py. A list of recommended python based ML packages can be found below. If you are making a model, then create your training script separately and train it against the training data given on the contest page. At the end of the training, save your model as an appropriate file. In predictRuns, load your model and do the necessary computation by feeding the input. You can customize your implementation by adding more classes and definitions to support predictRuns. The expected return value is an integer

Let model.xyz be the saved ML model, then save it as a sibling document to predictor.py.


Whenever the model is loaded, use the relative path to read the file.

CRICKET AND CODING - CRACK THE CHALLENGE

PARTICIPATE NOW

 HELP VIDEOS

IIT Madras, Chennai - 600036

 7850-999966 (Mon-Fri 9am-6pm)  
t20-contest@onlinedegree.iitm.ac.in



Scan QR code  
to chat with our  
WhatsApp bot  
+91  
9840776800

[View Site Map](#)  
© IIT Madras. All rights reserved.



[Privacy Policy](#) | [Terms of Service](#)