

CRICKET AND CODING - CRACK THE CHALLENGE

-  TRAINING DATASET
-  INPUT FORMAT
-  REQUIREMENTS.TXT
-  PLAYERS, TEAMS AND VENUES
-  SAMPLE PREDICTION

Problem Statement:

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

1. Programming Language to be used: only Python 3.9
2. Maximum size of zip file allowed: 5MB.
3. Maximum permitted execution time: 20secs.
4. We will be executing your code on a VM with the same docker image having a single threaded processor.

Data:

1. **Training data:** You will be provided two files for training data, they are as following:
 - a. **IPL_Ball_by_Ball_2008_2022.csv:** This file contains a ball by ball record of each IPL match since 2008 to 2022. The column names are self explanatory. This file is already present in the docker image.
 - b. **IPL_Matches_Result_2008_2022.csv:** This file contains results of each IPL match since 2008 to 2022. The column names are self explanatory. This file is already present in the docker image.
2. **Sample test data:** Each row represents a test sample.
 - a. **test_file.csv:** The column names are self explanatory.
3. **Sample submission file:** This file contains predicted runs by your model corresponding to each sample in test data.
 - a. **sample_submission.csv:** The column `id` contains the innings for which the prediction is made and the column `predicted_run` contains your prediction for the respective innings

IITM BS IPL Contest 2023 Shared files - [Google Drive](#) contains above files for your reference

Note: The test file for an innings may have discrepancies that your model/code will have to take care of, e.g. new players making their debut in IPL 2023, change in team name, change in stadium name etc.

Setting up your training environment

1. Install docker desktop from [link](#).
2. You will be using **this docker image**.
3. Run the following command to download the docker Image

```
> docker pull swarnimsoni/iitmbsiplcontest2023:latest
```

Above image contains all the necessary libraries and files to run your code.
4. Create a project directory with following files:
 - a. **mymodelfile.py:** Write your model's code in this file. **Do not change the structure/signature of the "MyModel" class.** You may create as many private variables and methods as you need. It has three important methods:
 - i. **__init__():** You can use this method to define your model and include any preprocessing steps.
 - ii. **fit():** This method trains the parameters of the model. It receives training data as a pair of pandas dataframes, trains the model and returns reference to the MyModel object itself.
 - iii. **predict():** This method receives test data as pandas dataframe and returns the predictions in specified format.
 - iv. **Note:** Your code should produce the match score as an integer. If it is float, it will be truncated to integer.
Example: a match score of 26.9 will be truncated to 26 from our side before the calculation of error.
5. The docker image uses Python 3.9.16. Please do not use packages other then the ones mentioned in the **requirements.txt** file.

How to run your model

1. Create a "project_dir" directory with following files:
 - a. **test_file.csv:** which will contain the test samples without the labels.
 - b. **submission.csv:** which contains the predictions by your model if there is no execution error.
 - c. **mymodelfile.py:** which contains your "MyModel" class.
 - d. **logs.txt:** which contains error messages in case the code throws any runtime exceptions, to help you debug your code.
2. Open a terminal and change directory to "project_dir", then:
 - a. On Linux (Bash)

```
docker run \
--mount type=bind,src=$(pwd)/test_file.csv,dst=/var/test_file.csv \
--mount type=bind,src=$(pwd)/submission.csv,dst=/var/submission.csv \
--mount type=bind,src=$(pwd)/mymodelfile.py,dst=/var/mymodelfile.py \
--mount type=bind,src=$(pwd)/logs.txt,dst=/var/logs.txt \
swarnimsoni/iitmbsiplcontest2023:latest
```

- b. On Windows (command line):

```
docker run ^
--mount type=bind,src="%cd%\test_file.csv",dst=/var/test_file.csv ^
--mount type=bind,src="%cd%\submission.csv",dst=/var/submission.csv ^
--mount type=bind,src="%cd%\mymodelfile.py",dst=/var/mymodelfile.py ^
--mount type=bind,src="%cd%\logs.txt",dst=/var/logs.txt ^
swarnimsoni/iitmbsiplcontest2023:latest
```

- c. On Windows (power shell):

```
docker run `
--mount type=bind,src=$(PWD)/test_file.csv,dst=/var/test_file.csv `
--mount type=bind,src=$(PWD)/submission.csv,dst=/var/submission.csv `
--mount type=bind,src=$(PWD)/mymodelfile.py,dst=/var/mymodelfile.py `
--mount type=bind,src=$(PWD)/logs.txt,dst=/var/logs.txt `
swarnimsoni/iitmbsiplcontest2023:latest
```

How to make your submissions?

1. Sign in to your Dashboard using the registered email address that you used for signing up for the contest
2. You will see a submission button in the page that will redirect you to a Google form
3. Select the Match Number for which you are doing prediction
4. If you have selected the coding competition, submit your prediction code in zip format. For Guess the score category, submit only the prediction in the form.


How to submit your model?

1. Help video [link](#)
2. **Zip your “mymodelfile.py” file. The name of the zipped file should be exactly the same as your team id. E.g. if your team id is TEAM0104123, then your submission file name should be be “TEAM0104123.zip”**
3. Only submission files with ".zip" extension are considered valid. Submissions with any other extension (e.g. ".gzip", ".rar", ".lz", ".7z", ".jar" etc) will be discarded.
4. The size of the zipped file should not exceed 5 MB.
5. Do not rename your “mymodelfile.py” file to anything else, otherwise your submission will be discarded.
6. In your zipped folder there should be only one file named “mymodelfile.py”, if there are extra files, your submission will be discarded.

How will your code be evaluated?

1. Your model will be evaluated in a very similar process as the one you used to train your model and make predictions.
2. The same docker image will be used for evaluation.
3. Your model will be executed by a single threaded processor.
4. Your predictions will be evaluated against actual labels. The evaluation metric used will be “mean_absolute_error”
5. **Your model will have exactly 20 seconds to run and produce predictions. If your model takes longer than 20 seconds, the evaluation process will be stopped and your submission will be considered invalid and discarded.**

Contact Us

- 
- Online Degree Office, 3rd Floor, ICSR Building,
IIT Madras, Chennai - 600036
ipl-contest-23@study.iitm.ac.in

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

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