Table of Contents

[Data Loading: 2](#_Toc59787396)

[Preprocessing: 2](#_Toc59787397)

[First Innings run prediction Model 3](#_Toc59787398)

[First Innings Feature Engineering 3](#_Toc59787399)

[First Innings Models 7](#_Toc59787400)

[Second Innings run prediction Model 14](#_Toc59787401)

[Second Innings Feature Engineering 14](#_Toc59787402)

[Second Innings Models 15](#_Toc59787403)

[Final Approach: 18](#_Toc59787404)

[First Innings final feature engineering: 18](#_Toc59787405)

[First Innings Final Model: 18](#_Toc59787406)

[Second Innings final feature engineering: 19](#_Toc59787407)

[Second Innings Final Model: 20](#_Toc59787408)

[Overall Accuracy: 20](#_Toc59787409)

# Data Loading:

1. **Notebook – Aggregate matches:**

Outputs:

MatchList.csv – a single csv with all match summary

<match\_id.csv> : 1 csv for each match , ball by ball details

1. **Notebook - Create Extended match summary**

Outputs

match\_stats.csv – innings statistics of batsman wise score and bowler wise wicket with total runs and total wickets

batsman\_list.pkl

bowler\_list.pkl

# Preprocessing:

1. **Notebook – Team Ranking:**

Outputs:

Country\_rank\_<year>.csv – year wise country ranking

Details:

country, effective\_win\_by\_runs, effective\_win\_by\_wickets, matches\_played,win\_count,score (summing after minmax normalization)

1. **Notebook – Batsman Ranking:**

Outputs:

batsman\_performance\_<year>.csv – year wise batsman score

Details:

batsman, country, total\_runs, average\_score, opponent\_mean, player\_of\_the\_match, winning\_contribution, run\_rate\_effectiveness, consistency(ignored), batsman\_score (summing after minmax normalization)

winning\_contribution – runs scored in winningmatch/team runs scored in winning match

run\_rate\_effectiveness – ratio of runrate in winning match/ team run rate in winning match

1. **Notebook – Bowler Rank:**

Outputs:

Bowler\_performance\_<year>.csv – year wise bowler score

Details:

Bowler, country, negative\_rate(-runrate given), no\_of\_wickets, wickets\_per\_match,, wickets\_per\_run, no\_of\_wins, team\_score, opponent\_mean, winning\_contribution, winning\_wicket\_rate\_contribution, bowler\_score (summing after minmax normalization)

winning\_wicket\_rate\_contribution- bowler wicket per match in winning matches/ team wicket per match in winning matches

# First Innings run prediction Model

## First Innings Feature Engineering

1. **Notebook – 1st innings model feature engineering**

Outputs:

feature\_first\_innings.csv

Details:

Match\_id, match\_date,team, opponent, location, is\_train, runs\_scored(first\_innings)

**Previous year features** (prior to 2019)

team\_score,opponent\_score, batsman\_mean, batsman\_max,run\_scored

**Trend calculated by last 5 matches against same opponent**- opponent\_base, opponent\_trend, opponent\_trend\_predict,opponent\_mean

**Trend calculated by last 5 matches in same location**-location\_base,location\_trend,location\_trend\_predict,location\_mean

**Trend calculated by last 5 matches-**current\_base, current\_trend,current\_trend\_predict,current\_mean

1. **Notebook – 1st innings model feature engineering expanded**

Outputs:

first\_innings\_feature\_expanded.csv

first\_innings\_team.csv

first\_innings\_opponent.csv

first\_innings\_batsman.csv

first\_innings\_batsman\_max.csv

first\_innings\_opponent\_bowler.csv

Details:

**first\_innings\_feature\_expanded.csv**

match\_id,match\_date, team, opponent, location, runs\_scored, opponent\_base, opponent\_trend, opponent\_trend\_predict, opponent\_mean, location\_base, loction\_trend, location\_trend\_predict, location\_mean, current\_base, current\_trend, current\_trend\_predict, current\_mean, is\_train, noise

**first\_innings\_team.csv**

Last 1 year team data from the date of the current match:

Match\_id,Country,win\_ratio,effective\_win\_by\_runs,effective\_win\_by\_wickets,matches\_played,win\_count

**first\_innings\_opponent.csv**

Last 1 year opponent data from the date of the current match:

match\_d,Country,win\_ratio,effective\_win\_by\_runs,effective\_win\_by\_wickets,matches\_played,win\_count

**first\_innings\_batsman.csv**

Last 1 year batsman data (top 5 mean) from the date of the current match:

Batsman,country,total\_runs,run\_rate,average\_score,opponent\_variability,player\_of\_the\_match,winning\_contribution,run\_rate\_effectiveness

**first\_innings\_batsman\_max.csv**

Last 1 year batsman data (max of each feature) from the date of current match:

Match\_id,Batsman,country,total\_runs,run\_rate,average\_score,opponent\_variability,player\_of\_the\_match,winning\_contribution,run\_rate\_effectiveness

**first\_innings\_opponent\_bowler.csv**

Last 1 year opponent bowler data(mean) from the date of current match:

Match\_id,negative\_rate,no\_of\_wickets,,wickets\_per\_match,wickets\_per\_run,no\_of\_wins,opponent\_variability,winning\_wicket\_rate\_contribution

1. **Notebook – 1st innings embedding features (V1\V2)**

Learn and embedding of Team, Location, Opponent using neural network

Outputs:

country\_location\_enc\_map.pkl

country\_location\_dec\_map.pkl

run\_model\_embedding.h5/json

first\_inn\_model\_embedding.h5/json

second\_inn\_model\_embedding.h5/json

group\_encode\_model.h5/json

**V2:**

country\_enc\_map.pkl

enc\_country\_map.pkl

loc\_enc\_map.pkl

enc\_loc\_map.pkl

run\_model\_embedding\_V2.h5/json

first\_inn\_model\_embedding\_V2.h5/json

second\_inn\_model\_embedding\_V2.h5/json

group\_encode\_model\_V2.h5/json

location\_model\_model\_embedding\_V2.h5/json

**Additional Output:**

location\_similarity.xlsx - A location similarity excel which shows which stadium is similar to it. This is done using the location\_model\_model\_embedding\_V2.h5/json

Details:

country\_location\_enc\_map.pkl – One hot encoding map of teams and location

country\_location\_dec\_map.pkl – One hot decoding map of teams and location

run\_model\_embedding – the model trained on one Hot team , opponent and location to run

first\_inn\_model\_embedding-child model with one hot team input

second\_inn\_model\_embedding – child model with one hot opponent input

group\_encode\_model-child model with team, opponent, location and intermediate concatenated output.

Model train date :2014/Jan-2018/Dec

Model validation data -2019/jan-2020/Nov

Model performance : loss: 4005.3777 - mean\_absolute\_percentage\_error: **22.7719** - mean\_absolute\_error: 49.9671 - val\_loss: 4062.1545 - val\_mean\_absolute\_percentage\_error: **23.7581** - val\_mean\_absolute\_error: 49.0957

V2:

loss: 4110.1011 - mean\_absolute\_percentage\_error: 22.9532 - mean\_absolute\_error: 50.2060 - val\_loss: 4132.1484 - val\_mean\_absolute\_percentage\_error: 23.9446 - val\_mean\_absolute\_error: 49.9264

NB:in V1 country and location are under same one hot encoding. In V2 they have separate One hot encoding (V1 results are slightly better in Linear regression)

Diagram

Description automatically generated

1. **Notebook - 1st innings batsman embedding features**

Learn an embedding of batsman, position, location, opposition, using neural network

Inputs:

country\_enc\_map.pkl

Preprocessing CSVs

Outputs:

'batsman\_enc\_map.pkl'

'enc\_batsman\_map.pkl'

'loc\_enc\_map\_for\_batsman.pkl’

'enc\_loc\_map\_for\_batsman.pkl'

batsman\_model.h5\json

batsman\_position\_model.h5\json

batsman\_location\_model.h5\json

batsman\_group\_encode\_model.h5\json

batsman\_encode\_runs\_model.h5\json

Additional Output:

Batsman\_similarity.xlsx

Diagram

Description automatically generated

## First Innings Models

1. **Notebook-1st innings model**

Different types of regression model using the basic feature engineering from ‘1st innings model feature engineering’

Input:

feature\_first\_innings.csv

Output:

first\_innings\_linear\_regression.pkl

first\_innings\_linear\_regression\_scaler.pkl

Other outputs:

result\_df\_tran.csv

result\_df\_test.csv

Models tried:

linear regression(sklearn),lasso,xgboot,RandomForestRegressor,statsmodel.ols, linear regresson with PCA,SVR(svm),polynomial regression, comparison with trends

Simple comparison with trends produce 28-39% Mape

Best result:

Linear Regression with Column selection from statsmodel ols based on P value

Selected columns :

'team\_score', 'opponent\_score', 'location\_base', 'location\_mean','batsman\_mean', 'batsman\_max', 'bowler\_mean'

Best metrics:

MSE :

3441 (training)

MAE:

42.06 (training)

46.29 (testing)

MAPE:

19.92(training)

22.43(testing)

1. Notebook-**1st innings Model Classification**

Different types of classification model using the basic feature engineering from ‘1st innings model feature engineering’. The scored\_runs in output is categorized

Inputs:

feature\_first\_innings.csv

Techniques tried:

Logistic Regression, XGBoost, KNN, Custom Naive Bayes(custom developed), Gaussian naive bayes

Max test accuracy (with low variance)in Gaussian Naïve Bayes:30%

1. Notebook-**1st innings model-expanded**

Regression using expanded features from **1st innings model feature engineering expanded**

Inputs : All outputs from **1st innings model feature engineering expanded**

Results:

No improvement from basic 1st innings models (Regression)

1. Notebook- **1st innings model-expanded-Neural-Network**

A hierarchical neural network model with features from **1st innings model feature engineering expanded.** Produces many child models which enables to get performance measures team an batsman.

Inputs: All outputs from **1st innings model feature engineering expanded**

Ouptut:

run\_model.h5/json

team\_model.h5/json

batsman\_model.h5/json

opponent\_model.h5/json

bowler\_model.h5/json

history\_model.h5/json

batsman\_scaler.pkl

opponent\_scaler.pkl

opponent\_bowler\_scaler.pkl

Results:

Train- mse loss: 3368.9673 - mean\_absolute\_percentage\_error: 21.2354 - mean\_absolute\_error: 45.1116

Test- mse loss: 3935.7935 - mean\_absolute\_percentage\_error: 22.8567 - mean\_absolute\_error: 48.1334

Additional benefits: team\_model and batsman\_model can be sued to score teams and batsman given the feature vectors.

**Batsman\_ranking** notebook can be sued to create batsman vecotrs

Diagram

Description automatically generated

1. Notebook - **1st innings model with embedding (V1\V2)**

Use basic features from “**1st innings model feature engineering”** (applying only the highly significant ones) along with creating group embedding features from one hot vectors and models produced by “**1st innings embedding features”**

Inputs :

feature\_first\_innings.csv

country\_location\_enc\_map.pkl

group\_encode\_model.h5/json

V2:

country\_enc\_map.pkl

loc\_enc\_map.pkl

group\_encode\_model\_V2.h5/json

Ouputs:

first\_innings\_linear\_regression\_enc.pkl

first\_innings\_linear\_regression\_scaler\_enc.pkl

V2:

first\_innings\_linear\_regression\_enc\_V2.pkl

first\_innings\_linear\_regression\_scaler\_enc\_V2.pkl

features:

'team\_score', 'opponent\_score', 'location\_base', 'location\_mean','batsman\_mean', 'batsman\_max', 'bowler\_mean'

+

30 embedding features produced by feeding one hot vectors from country\_location\_enc\_map to group\_encode\_model

Models tried:

linear regression(sklearn),lasso,xgboot,RandomForestRegressor,statsmodel.ols, linear regresson with PCA,SVR(svm),polynomial regression, comparison with trends

Best results:

Linear regression:

MAE:

train-39.85

test-43.51

MAPE:

Train-18.76

**Test-19.85 (Best so far -3% improvement)**

V2:

MAE:

Train-40.28

Test-44.9

MAPE:

Train-19.02

Test-20.83

N.B.: V2 uses separate one hot encoding for country and location

1. Notebook **- 1st innings model-expanded-Neural-Network-embedding**

Same features as “**1st innings model with embedding”** but with hierarchial Neural Network.

Results are not better than Linear Regression

Results:

Train: loss: 2875.8057 - mean\_absolute\_percentage\_error: 18.7146 - mean\_absolute\_error: 40.0145 - val\_loss: 4073.9697 - val\_mean\_absolute\_percentage\_error: 22.0898 - val\_mean\_absolute\_error: 47.7797

Test: loss: 4174.1196 - mean\_absolute\_percentage\_error: 22.6665 - mean\_absolute\_error: 49.8182

1. Notebook- **1st innings model with batsman embedding**

A linear regression model using batman embedding which is learnt in “**1st innings batsman embedding features**”

Inputs:

country\_enc\_map.pkl

batsman\_enc\_map.pkl

loc\_enc\_map\_for\_batsman.pkl

batsman\_group\_encode\_model

Additional Inputs to combine with previous features:

loc\_enc\_map.pkl (**from 1st innings embedding features**)

group\_encode\_model\_V2 ( **from 1st innings embedding features**)

feature\_first\_innings.csv (**from 1st innings model feature engineering**)

columns ['team\_score', 'opponent\_score', 'location\_base', 'location\_mean','batsman\_mean', 'batsman\_max', 'bowler\_mean']

Ouputs:

(batsman embedding+team embedding+selected features)

scaler\_combined\_embedding\_first\_innings\_regression.pkl

combined\_embedding\_first\_innings\_regression.pkl

prediction\_results:

first\_innings\_embedding\_prediction\_train.csv

first\_innings\_embedding\_prediction\_test.csv

Results:

With only batsman embedding:

MAPE-

Train:15.95

Test:20.56

MAE-

Train:33

Test:43

With batsman embedding and team embedding:

MAPE-

Train:15.18

Test:20.15

With batsman embedding , team embedding and selected engineered features (Data reduced: Test data 165->109

MAPE-

Train:14.88

Test:18.43 (Best)

MAE:

Train:32.11

Test:39.05

1. Notebook - **1st innings model LSTM with batsman embedding**

Using the team and batsman embedding to predict overall runs as well as Individual runs

Inputs:

country\_enc\_map.pkl

batsman\_enc\_map.pkl

loc\_enc\_map\_for\_batsman.pkl

loc\_enc\_map.pkl

group\_encode\_model\_V2.h5\json

batsman\_group\_encode\_model.h5\json

Approach 1 (Total runs only):

Timeline

Description automatically generated

Results:

loss: 2934.8440

mean\_absolute\_error: 42.2753

mean\_absolute\_percentage\_error: 19.7362

val\_loss: 4425.2393 –

val\_mean\_absolute\_error: 50.7724

val\_mean\_absolute\_percentage\_error: 24.5317

Approach 1 (Individual runs with Total runs):

Timeline

Description automatically generated

Results:

loss: 3901.1565

individual\_output\_loss: 978.7510

total\_output\_loss: 2922.4060

individual\_output\_mean\_absolute\_error: 20.4457

total\_output\_mean\_absolute\_error: 42.4627

val\_loss: 5432.1118

val\_individual\_output\_loss: 953.9545

val\_total\_output\_loss: 4478.1572

val\_individual\_output\_mean\_absolute\_error: 20.6076

val\_total\_output\_mean\_absolute\_error: 51.4404

# Second Innings run prediction Model

## Second Innings Feature Engineering

1. Notebook - **2nd innings feature engineering**

Output:

feature\_second\_innings.csv

'match\_id','match\_date','team','opponent','location','team\_score','opponent\_score', , 'is\_train', 'noise', 'target\_score', 'runs\_scored', 'win'

Trend analyzed features:

'opponent\_base','opponent\_trend','opponent\_trend\_predict','opponent\_mean','location\_base','location\_trend','location\_trend\_predict','location\_mean'current\_base','current\_trend', 'current\_trend\_predict','current\_mean'

Features from previous year :

'batsman\_mean','batsman\_max','bowler\_mean', 'bowler\_max'

1. Notebook - **2nd innings feature engineering-alternate**

Historical trend features are replaced with historical win ratios.

Output:

feature\_second\_innings\_alt.csv

'match\_id','match\_date','team','opponent','location','team\_score','opponent\_score', , 'is\_train', 'noise', 'target\_score', 'runs\_scored', 'win'

Features from previous year :

'batsman\_mean','batsman\_max','bowler\_mean', 'bowler\_max'

Historical Trends:

'opponent\_win\_ratio', 'location\_win\_ratio', 'last\_5\_match\_win\_ratio'

1. Notebook - **2nd innings feature engineering expanded**

Output:

second\_innings\_feature\_expanded.csv

second\_innings\_team.csv

second\_innings\_opponent.csv

second\_innings\_batsman.csv

second\_innings\_batsman\_max.csv

second\_innings\_opponent\_bowler.csv

Details:

Same as 1sr innings feature engineering expanded (only win is added as 1/0)

## Second Innings Models

1. Notebook- win\_loss\_with\_first\_innings\_model

Use first innings model (basic without encoding)to predict second innings score and predict win loss based on score comparison

Input:

first\_innings\_linear\_regression.pkl

first\_innings\_linear\_regression\_scaler.pkl

Accuracy :

75% (train)

73% (test)

**To do**: Check after best first Innings model is finalized

1. Notebook- second\_innings\_model

Simple classification model to verify if team will be able to successfully chase the first innings score

Input:

feature\_second\_innings.csv

Output:

'second\_innings\_model\_lrg.pkl'

'second\_innings\_scaler.pkl'

Description:

Classifications tried with Logistic Regression,XGBoost, Random Forest,SVM, statsmodel.Logit, Gaussian Naïve Bayes.

Best results from Logistic Regression

Accuracy:

Train: 85%

Test: 83%

Overall accuracy if first innings prediction is used

Train: 77%

Test: 77%

Overall accuracy if first innings prediction is used with first innings embedding model:

Train:81%

Test:79%

1% improvement of test accuracy, if less than 10% difference in probability of win\loss is eliminated as unpredictable.

2% improvement of test accuracy, if less than 15% difference in probability of win\loss is eliminated as unpredictable.

1. **Notebook - second\_innings\_model-alt**

In this classification historical win ratio is used from 2nd Innings feature Engineering Alternate

Input :

feature\_second\_innings\_alt.csv

Accuracy:

Train:84%

Test:81%

Overall with first innings prediction:

Train:77%

Test:73%

1. Notebook - **2nd innings model-expanded-Neural-Network**

Use Extended features with hierarchial neural network

Inputs:

second\_innings\_feature\_expanded.csv

second\_innings\_team.csv

second\_innings\_opponent.csv

second\_innings\_batsman.csv

second\_innings\_batsman\_max.csv

second\_innings\_opponent\_bowler.csv

Accuracy:

Train:80%

Test:70%

1. Notebook - **second\_innings\_model\_with\_embedding**

Inputs :

feature\_second\_innings.csv

country\_enc\_map.pkl

loc\_enc\_map.pkl

group\_encode\_model\_V2.h5\json

batsman\_enc\_map.pkl

loc\_enc\_map\_for\_batsman.pkl

batsman\_group\_encode\_model.h5\json

Additional inputs to combine with first innings model prediction:

first\_innings\_embedding\_prediction\_train.csv

first\_innings\_embedding\_prediction\_test.csv

Outputs:

second\_innings\_model\_with\_embedding\_lrg.pkl

second\_innings\_model\_with\_embedding\_svm.pkl

second\_innings\_scaler\_with\_embedding.pkl

Features:

Batsman embedding sum + country embedding + Engineered features from feature\_second\_innings.csv

Model type : Logistic regression\SVM

Results:

Train Accuracy -92%

Test Accuracy -86%

87%(SVM)

When results combined with First innings prediction

Train Accuracy -91%

Test Accuracy - 84%

# Final Approach:

## First Innings final feature engineering:

1. **Batsman embedding sum**

Sum of all the embeddings for the team batsman. Each batsman has an embedding for batsman name, position, location opposition

Use :

batsman\_enc\_map.pkl

country\_enc\_map.pkl

loc\_enc\_map\_for\_batsman.pkl

batsman\_group\_encode\_model.h5/json

ref: Notebook : **1st innings batsman embedding features**

1. **Team embedding**

Each team has an embedding for team, opposition, location

Use:

country\_enc\_map.pkl

loc\_enc\_map.pkl

group\_encode\_model\_V2.h5/json

ref: Notebook : **1st innings embedding features**

1. **Engineered features selected based on P value:**

'team\_score', 'opponent\_score', 'location\_base', 'location\_mean','batsman\_mean', 'batsman\_max', 'bowler\_mean'

However subset of the following features needs to be considered while model rebuilding

**Previous year features** (prior to 2019)

team\_score,opponent\_score, batsman\_mean, batsman\_max,run\_scored

**Trend calculated by last 5 matches against same opponent**- opponent\_base, opponent\_trend, opponent\_trend\_predict,opponent\_mean

**Trend calculated by last 5 matches in same location**-location\_base,location\_trend,location\_trend\_predict,location\_mean

**Trend calculated by last 5 matches-**current\_base, current\_trend,current\_trend\_predict,current\_mean

Ref: Notebook-**1st innings model feature engineering**

## First Innings Final Model:

Linear regression on the features

scaler\_combined\_embedding\_first\_innings\_regression.pkl

combined\_embedding\_first\_innings\_regression.pkl

Final results:

MAPE-

Train:14.88

Test:18.43 (Best)

MAE:

Train:32.11

Test:39.05

ref: Notebook - **1st innings model with batsman embedding**

## Second Innings final feature engineering:

1. **Batsman embedding sum**

Sum of all the embeddings for the team batsman. Each batsman has an embedding for batsman name, position, location opposition

Use :

batsman\_enc\_map.pkl

country\_enc\_map.pkl

loc\_enc\_map\_for\_batsman.pkl

batsman\_group\_encode\_model.h5/json

ref: Notebook : **1st innings batsman embedding features**

1. **Team embedding (from first innings)**

Each team has an embedding for team, opposition, location

Use:

country\_enc\_map.pkl

loc\_enc\_map.pkl

group\_encode\_model\_V2.h5/json

ref: Notebook : **1st innings embedding features**

Note: this is done on first innings, because it is believed that the potential of a team is based on first innings score. Moreover second innings is based on a target and is classification.

1. **Engineered features:**

**Historic(last year performance based)**

'team\_score','opponent\_score', 'batsman\_mean', 'batsman\_max', 'bowler\_mean',

'bowler\_max',

**Opponent Trend based:**

'opponent\_base', 'opponent\_trend','opponent\_trend\_predict', 'opponent\_mean',

**Location Trend based:**

'location\_base','location\_trend', 'location\_trend\_predict', 'location\_mean',

**Rexent Trend based**

'current\_base', 'current\_trend', 'current\_trend\_predict','current\_mean'

**Current match based**:

'target\_score',

Ref Notebook: **2nd innings feature engineering**

## Second Innings Final Model:

Logistic regression\SVM on the features

second\_innings\_model\_with\_embedding\_lrg.pkl

second\_innings\_model\_with\_embedding\_svm.pkl

second\_innings\_scaler\_with\_embedding.pkl

Final results:

Results:

Train Accuracy -92%

Test Accuracy -86%

87%(SVM)

When results combined with First innings prediction

Train Accuracy -91%

Test Accuracy - 84%

## Overall Accuracy:

Train Accuracy -91%

Test Accuracy - 84%

Other benefits:

1. Optimize team and batting order
2. Predict sufficient target for first innings while batting first
3. Predict achievable target by second innings team