A
Project Report
On

T20 World Cup Winner Prediction

By

Anupam Tiwari (2020510065)

Under the guidance of

Internal Supervisor

Prof. Pallavi Thakur

Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University
2022-23

CERTIFICATE OF APPROVAL

This is to certify that the following students

Anupam Tiwari (2020510065)

Have satisfactorily carried out work on the project entitled

“T20 World Cup Winner Prediction”

Towards the fulfilment of project, as laid down
by
Sardar Patel Institute of Technology
during year
2022-23.

Project Guide:
Prof. Pallavi Thakur

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Anupam Tiwari (2020510065)

Have successfully completed the Project report on

“T20 World Cup Winner Prediction”,

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI

INTERNAL EXAMINER EXTERNAL EXAMINER

HEAD OF DEPARTMENT PRINCIPAL

Contents

Abstract

Cricket is one of the most popular team games in the world. We embark on predicting the outcome of the ICC T20 World Cup cricket match using a supervised learning approach from a team composition perspective. Our work suggests that the relative team strength between the competing teams forms a distinctive feature for predicting the Winner. Modeling the team's strength boils down to modeling individual players' batting and bowling performances, forming the basis of our approach.

I have used past data sets as well as recent performances of a player to model him. Player-independent factors have also been considered to predict the outcome of a match. We will show that the Random Forest Algorithms yields better results as compared to other classifiers like Decision Tree, Naive Bayes classification algorithm, Support Vector Machine(SVM), RGB Regressor Algorithm, etc. The Performance is affected by the type, size, and quality of the data.

Objectives

The T20 World Cup Prediction is used

- The main objective of developing this project is to create a training model which learns the past features of the T20 world cup data set using a machine learning algorithm and predicts the final winning of the upcoming T20 World Cup as an output.
- Provide the accuracy of the percentage of the different models of the T20 world cup data set.

List of Figures

List of Tables

1 Introduction

1.1 Problem Definition

Machine Learning has been used in sports for decades and has contributed significantly to the success of the field.Cricket is one of the most popular sports in the world,second only to soccer.Various natural factors affecting the game,enormous media coverage,and a huge betting market have been given from various perspectives.However, the complex rules governing the game, the ability of players and their performances on a given day,and various other natural parameters play an integral role in affecting the outcome of a cricket match.This presents significant challenges in predicting the accurate results of a game.

The game of cricket is played in three formats – Test Matches, ODIs, and T20s. We focus our research on T20, the most popular format of the game.To Predict the outcome of T20 cricket Matches, We will propose an approach where we first estimate the batting and bowling potentials of the 22 players in the match using their career statistics and active participation in recent games.We will use these player potentials to render the relative dominance one team has over the other. Taking two other base features into account, namely, the toss decision and the venue of the match, along with the relative team strength, we adopt supervised learning algorithms to predict the winner of the match.

1.2 Objectives and Scope

- The main objective of developing this project is to create a training model which learns the past features of the T20 world cup dataset using a machine learning algorithm and predicts the final winning of the upcoming T20 World Cup as an output.
- Provide the accuracy of percentage of the different model of T20 world cup dataset.

1.3 Feasibility Study

1.3.1 Technical Feasibility

Requirements of the Libraries:
The requirements of Libraries are as follows :

- tensorflow
- keras
- imutils
- numpy
- opencv-python
- matplotlib
- sckipy

1.3.2 Economic Feasibility

Development Cost: Rs 35000
Total Budget: Rs 40,000

1.4 System Requirements

- Hardware Requirements

Table 1.5.1: Hardware Requirements

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 10 GB Hard Disk Space for smooth run

- Software Requirements

Table 1.5.3: Software Requirements

Operating System	IOS,Window,Linux
Software	Python,Anaconda,Jupyter Lab

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

The main objective of developing this project is to create a training model which learns the past features of the T20 world cup dataset using a machine learning algorithm and predicts the final winning of the upcoming T20 World Cup as an output.

2.2 Scope

Provide the accuracy of the percentage of the different models of T20 world cup datasets. The game of cricket is played in three formats – Test Matches, ODIs, and T20s. We focus our research on T20, the most popular format of the game.

2.3 References

1. https://ieeexplore.ieee.org/document/9642558
2. https://www.researchgate.net/publication/353718778
3. https://www.researchgate.net/publication/342335546

2.4 Product Perspective

Cricket is one of the most popular sports in the world, second only to soccer. Various natural factors affecting the game, enormous media coverage, and a huge betting market have been given from various perspectives. However, the complex rules governing the game, the ability of players and their performances on a given day, and various other natural parameters play an integral role in affecting the outcome of a cricket match. This presents significant challenges in predicting the accurate results of a game.

2.5 Product Functions

The Product Functions follow the following steps:

-
- Datasets Collection
- Pre-processing
- Training the datasets
- Testing the datasets
- Model Evaluation
- Predicts Output

2.6 Assumptions and Dependencies

2.6.1 Assumptions

The datasets which we will be used for the initial entries of the training of the system is assumed to be the correct input for the system. The training dataset is taken from the online repositories such as Kaggle.

2.6.2 Dependencies

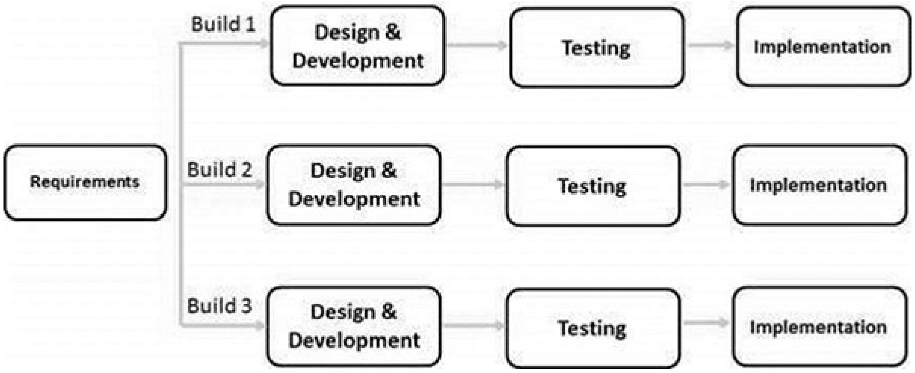
- Python
- Tensorflow – keras
- Juypter
- Pandas

3 Project Analysis and Design

3.1 Methodologies Adapted

The best suitable model for this project is the Iterative model.
Iterative Model:

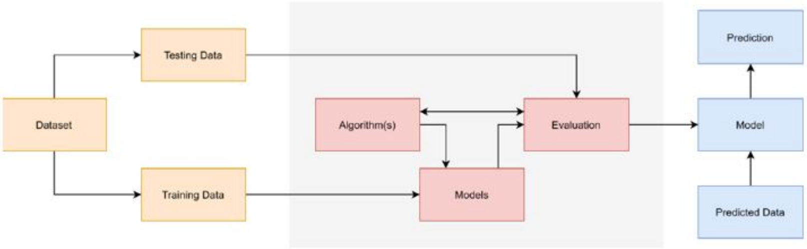
In this project, we iterate through the training and the testing phase during the development.With Iterative development, there are changes on each iteration, evolves and grows. As each iteration builds on the previous one, software design remains consistent.



3.1.1: Diagrammatic Representation of Iterative Model

3.2 Algorithms

The major contribution of this system is creating a training model which learns the T20 dataset using Machine Learning algorithms. The following flowcharts can give the process information:



3.2.1: Steps in building the model

Algorithms used in T20 World Cup Datasets:

3.2.1 Random Forest Algorithm:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining concept of ensemble learning , which is a process of combing multiple classifiers to solve a complex problem and improve the performance of the model.

3.2.2 Decision Tree Algorithm:

A decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a Tree-structured classifier, where internal nodes represents the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

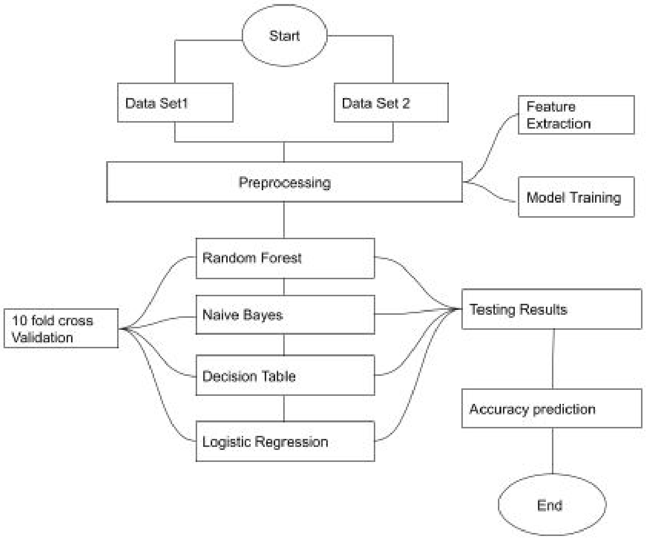
3.2.3 RGB Regressor Algorithm:

XGBoost is a powerful approach for building supervised regression models. The validity of this statement can be inferred by knowing about its (XGBoost) objective function and base learners. The objective function contains loss function and a regularization term. It tells about the difference between actual values and predicted values, i.e how far the model results are from the real values.

3.2.4 SVM Algorithm:

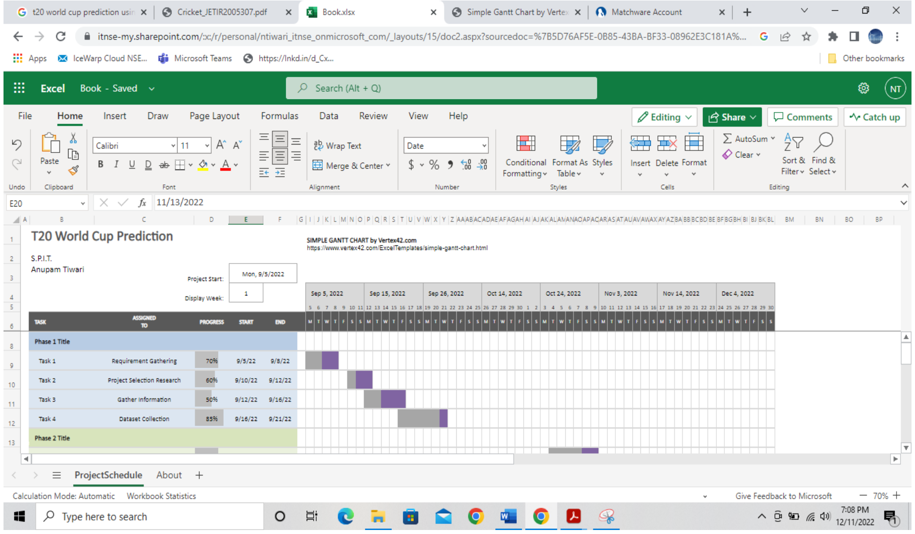
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

3.3 Flowchart Diagram



3.3.1: Experimental Flow Chart

3.4 Gantt Table



E20 11/13/2022

T20 World Cup Prediction

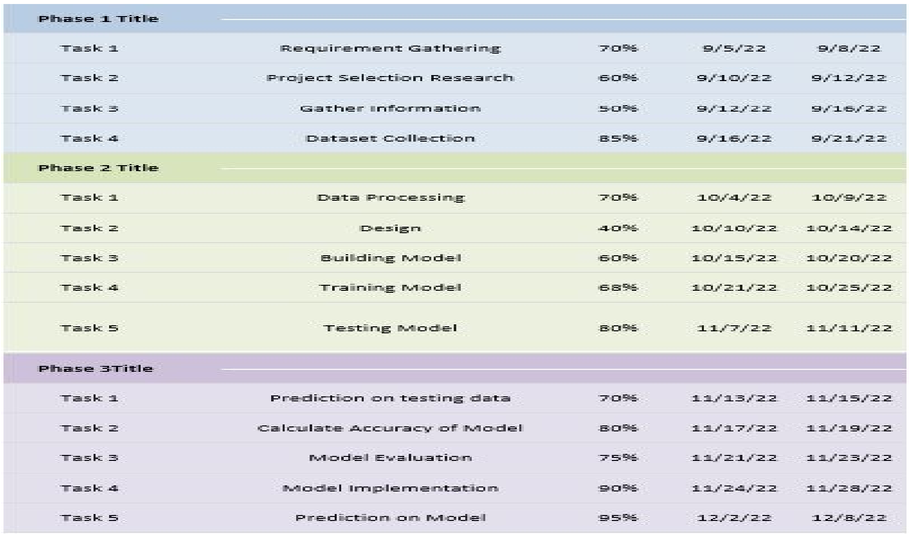
S.P.I.T.
Anupam Tiwari

Project Start: Mon, 9/5/2022

Display Week: 1

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1 Title				
Task 1	Requirement Gathering	70%	9/5/22	9/8/22
Task 2	Project Selection Research	60%	9/10/22	9/12/22
Task 3	Gather Information	50%	9/12/22	9/16/22
Task 4	Dataset Collection	85%	9/16/22	9/21/22
Phase 2 Title				

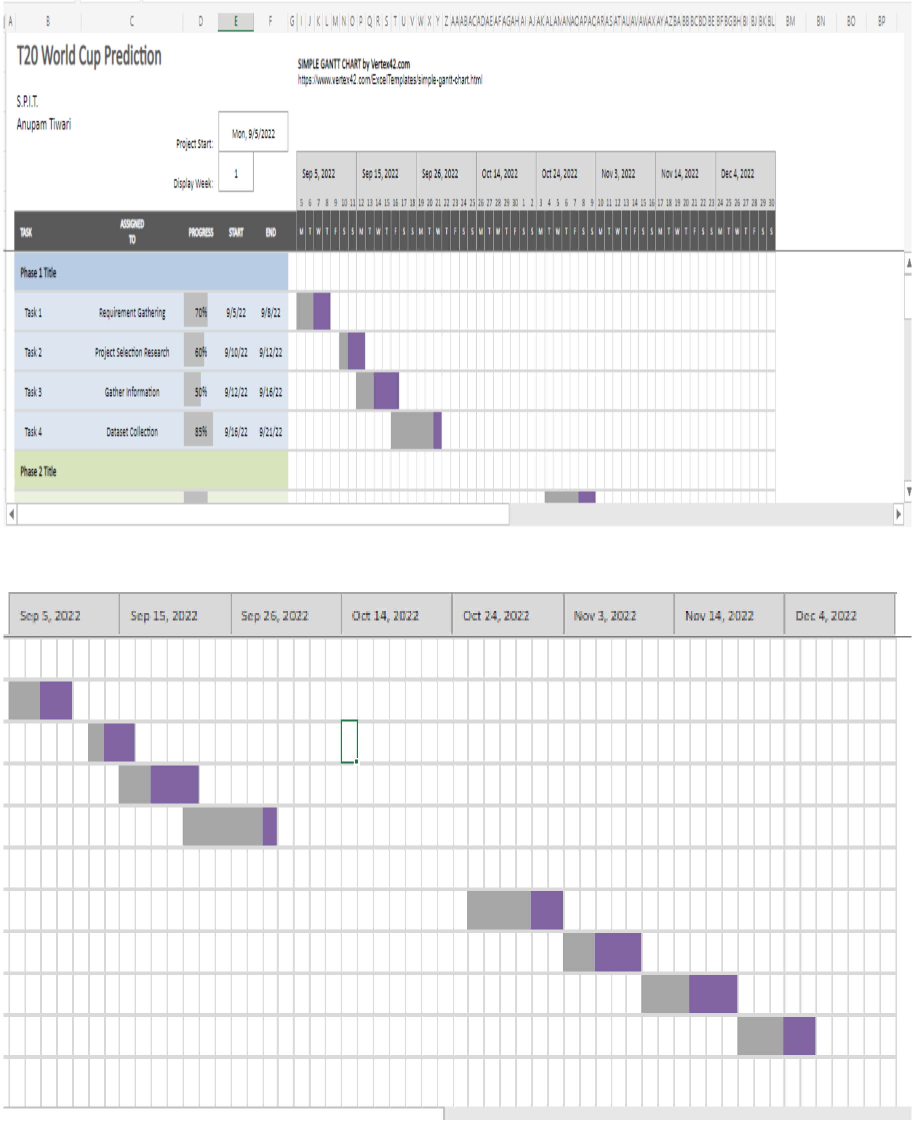


Phase 1 Title				
Task 1	Requirement Gathering	70%	9/5/22	9/8/22
Task 2	Project Selection Research	60%	9/10/22	9/12/22
Task 3	Gather Information	50%	9/12/22	9/16/22
Task 4	Dataset Collection	85%	9/16/22	9/21/22
Phase 2 Title				
Task 1	Data Processing	70%	10/4/22	10/9/22
Task 2	Design	40%	10/10/22	10/14/22
Task 3	Building Model	60%	10/15/22	10/20/22
Task 4	Training Model	68%	10/21/22	10/25/22
Task 5	Testing Model	80%	11/7/22	11/11/22
Phase 3Title				
Task 1	Prediction on testing data	70%	11/13/22	11/15/22
Task 2	Calculate Accuracy of Model	80%	11/17/22	11/19/22
Task 3	Model Evaluation	75%	11/21/22	11/23/22
Task 4	Model Implementation	90%	11/24/22	11/28/22
Task 5	Prediction on Model	95%	12/2/22	12/8/22

3.4.1: Gantt Table

3.5 Gantt Chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time.



T20 World Cup Prediction

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

S.P.I.T.
Anupam Tiwari

Project Start: Mon, 9/5/2022

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1 Title				
Task 1	Requirement Gathering	70%	9/5/22	9/8/22
Task 2	Project Selection Research	60%	9/10/22	9/12/22
Task 3	Gather Information	50%	9/12/22	9/16/22
Task 4	Dataset Collection	85%	9/16/22	9/21/22
Phase 2 Title				

3.5.1: Gantt Chart in Weeks

4 System Design

4.1 Users of the System

User: The user who handles the System. Collects datasets on problem definition, Processes the data applies model algorithm on the datasets, and predicts the correct output.

4.2 Modularity criteria

The proposed system has the following modules:

- data collection
- pre-processing
- split the data
- building the model
- testing the model
- implement the model

4.3 Design Methodologies

XGBoost Algorithm:

- XGBoost is a powerful approach for building supervised regression models.
- The validity of this statement can be inferred by knowing about its (XGBoost) objective function and base learners.
- The objective function contains a loss function and a regularization term. It tells about the difference between actual values and predicted values, i.e how far the model results are from the real values.
- The most common loss function in XGBoost for regression problems is reg: linear, and that for binary classification is reg: logistics. Ensemble learning involves training and combining individual models (known as base learners) to get a single prediction, and XGBoost is one of the ensemble learning methods.
- XGBoost expects to have the base learners which are uniformly bad at the remainder so that when all the predictions are combined, bad predictions cancel out and better one sums up to form final good predictions.

5 Implementation and Testing

5.1 Tools/Scripts for Implementation

- Anaconda

  Anaconda is a free open-source software and it supports free GPU. we can improve your Python programming language coding skills. Develop machine learning and deep learning applications using popular libraries such as Keras, TensorFlow and OpenCV. We use Anaconda for developing the training model.

- TensorFlow

  TensorFlow is an end-to-end open source platform for machine learning. It's a comprehensive and flexible ecosystem of tools, libraries and other resources that provide workflows with high- level APIs. The framework offers various levels of concepts for you to choose the one you need to build and deploy machine learning models. For instance, if you need to do some large machine learning tasks, you can use the Distribution Strategy API in order to perform distributed hardware configurations and if you need a full production machine learning pipeline, you can simply use TensorFlow Extended (TFX).

- Keras

  Keras on the other hand is a high-level neural networks library that is running on top of TensorFlow, CNTK, and Theano. Using Keras in deep learning allows for easy and fast prototyping as well as running seamlessly on CPU and GPU. This framework is written in Python code which is easy to debug and allows ease of extensibility.

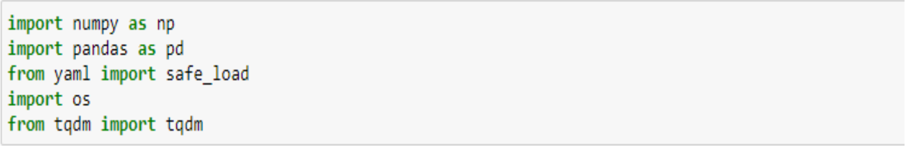
- OpenCV

  OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products.

5.2 Module Hierarchy

- Dataset Collection

  For T20 World Cup prediction, the dataset has been collected From the Kaggle website i.e. Cricsheet Dataset.

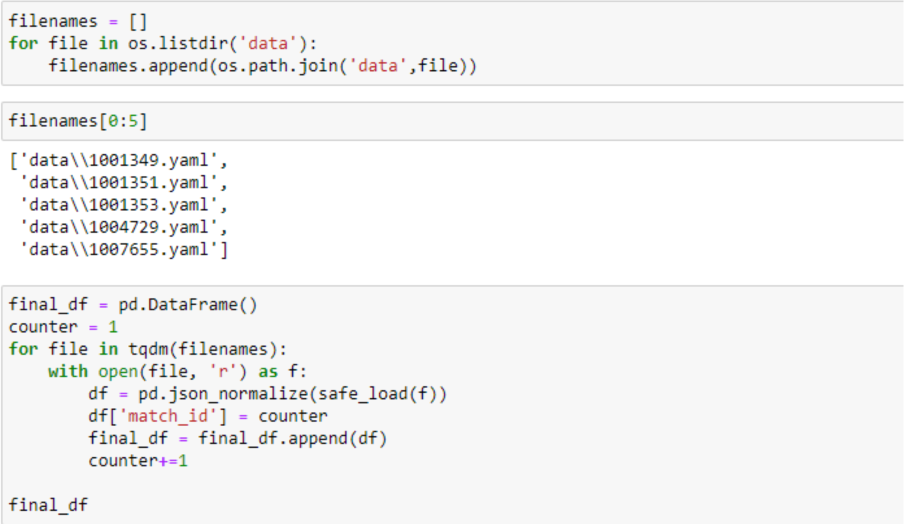


```
import numpy as np
import pandas as pd
from yaml import safe_load
import os
from tqdm import tqdm
```

5.2.1: Dataset Collection

- Preprocessing

  Preprocessing on datasets is done to convert .yaml data into tqdm. Preprocessing in the caption is performed to make sentences that were previously in the form of the word into a sequence of tokens based on a unique word index in the dictionary.



```
filenames = []
for file in os.listdir('data'):
    filenames.append(os.path.join('data',file))
```

```
filenames[0:5]
```

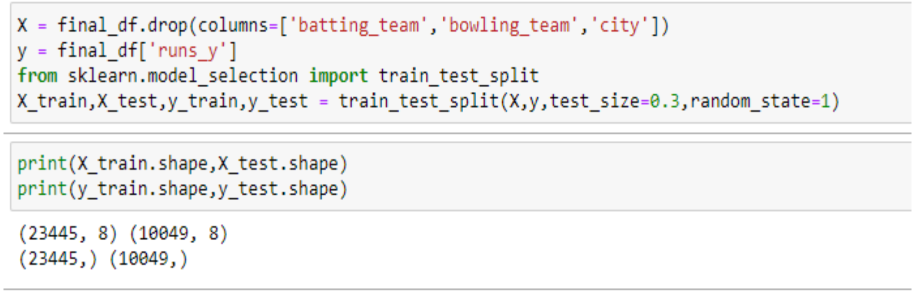
```
['data\\1001349.yaml',
 'data\\1001351.yaml',
 'data\\1001353.yaml',
 'data\\1004729.yaml',
 'data\\1007655.yaml']
```

```
final_df = pd.DataFrame()
counter = 1
for file in tqdm(filenames):
    with open(file, 'r') as f:
        df = pd.json_normalize(safe_load(f))
        df['match_id'] = counter
        final_df = final_df.append(df)
        counter+=1

final_df
```

5.2.2: Processing on Datasets

- Training the datasets

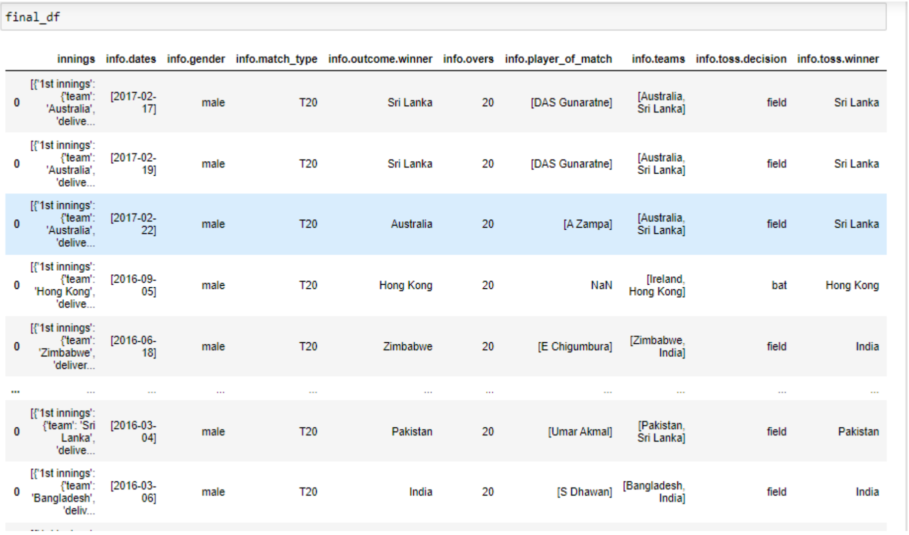
  Training data (or a training dataset) is the initial data used to train machine Learning models. Training datasets are fed to machine learning algorithms to teach them how to make predictions or perform a desired task.



```
X = final_df.drop(columns=['batting_team','bowling_team','city'])
y = final_df['runs_y']
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=1)
```

```
print(X_train.shape,X_test.shape)
print(y_train.shape,y_test.shape)
```

```
(23445, 8) (10049, 8)
(23445,) (10049,)
```



```
final_df
```

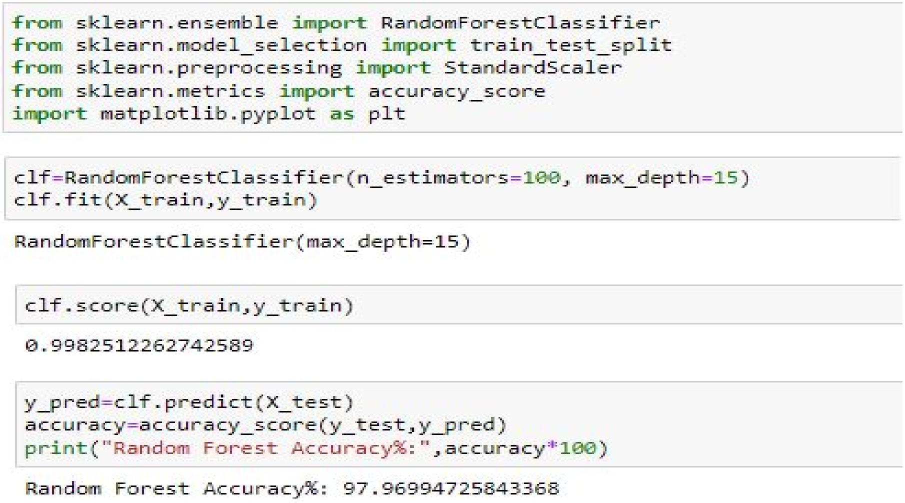
	innings	info.dates	info.gender	info.match_type	info.outcome.winner	info.overs	info.player_of_match	info.teams	info.toss.decision	info.toss.winner
0	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-17]	male	T20	Sri Lanka	20	[DAS Gunaratne]	[Australia, Sri Lanka]	field	Sri Lanka
0	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-19]	male	T20	Sri Lanka	20	[DAS Gunaratne]	[Australia, Sri Lanka]	field	Sri Lanka
0	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-22]	male	T20	Australia	20	[A Zampa]	[Australia, Sri Lanka]	field	Sri Lanka
0	[{'1st innings': {'team': 'Hong Kong', 'delive...	[2016-09-05]	male	T20	Hong Kong	20	NaN	[Ireland, Hong Kong]	bat	Hong Kong
0	[{'1st innings': {'team': 'Zimbabwe', 'deliver...	[2016-06-18]	male	T20	Zimbabwe	20	[E Chigumbura]	[Zimbabwe, India]	field	India
...	...	...	...	...	...	...	...	...	...	...
0	[{'1st innings': {'team': 'Sri Lanka', 'delive...	[2016-03-04]	male	T20	Pakistan	20	[Umar Akmal]	[Pakistan, Sri Lanka]	field	Pakistan
0	[{'1st innings': {'team': 'Bangladesh', 'deliv...	[2016-03-06]	male	T20	India	20	[S Dhawan]	[Bangladesh, India]	field	India

5.2.3: Training the datasets

5.3 Testing the Model

- Random Forest Algorithm

  Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on concept of ensemble learning, which is a processs of combining concept of ensemble learning , which is a process of combing multiple classifiers to solve a complex problem and to improve the performance of the model.



```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
```

```
clf=RandomForestClassifier(n_estimators=100, max_depth=15)
clf.fit(X_train,y_train)
```

```
RandomForestClassifier(max_depth=15)
```

```
clf.score(X_train,y_train)
```

```
0.9982512262742589
```

```
y_pred=clf.predict(X_test)
accuracy=accuracy_score(y_test,y_pred)
print("Random Forest Accuracy%:",accuracy*100)
```

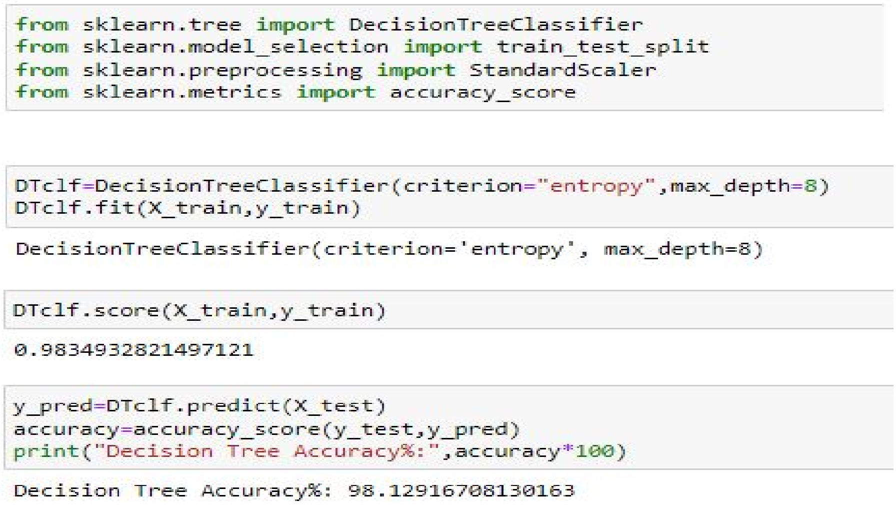
```
Random Forest Accuracy%: 97.96994725843368
```

5.3.1: Random Forest Algorithm

Random Forest Algorithm has predicated accuracy approx.97.96 percent by using T20 World Cup datasets.

- Decision Tree Algorithm

  Decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a Tree-structured classifier, where internal nodes represents the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.



```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
```

```
DTclf=DecisionTreeClassifier(criterion="entropy",max_depth=8)
DTclf.fit(X_train,y_train)
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=8)
```

```
DTclf.score(X_train,y_train)
```

```
0.9834932821497121
```

```
y_pred=DTclf.predict(X_test)
accuracy=accuracy_score(y_test,y_pred)
print("Decision Tree Accuracy%:",accuracy*100)
```

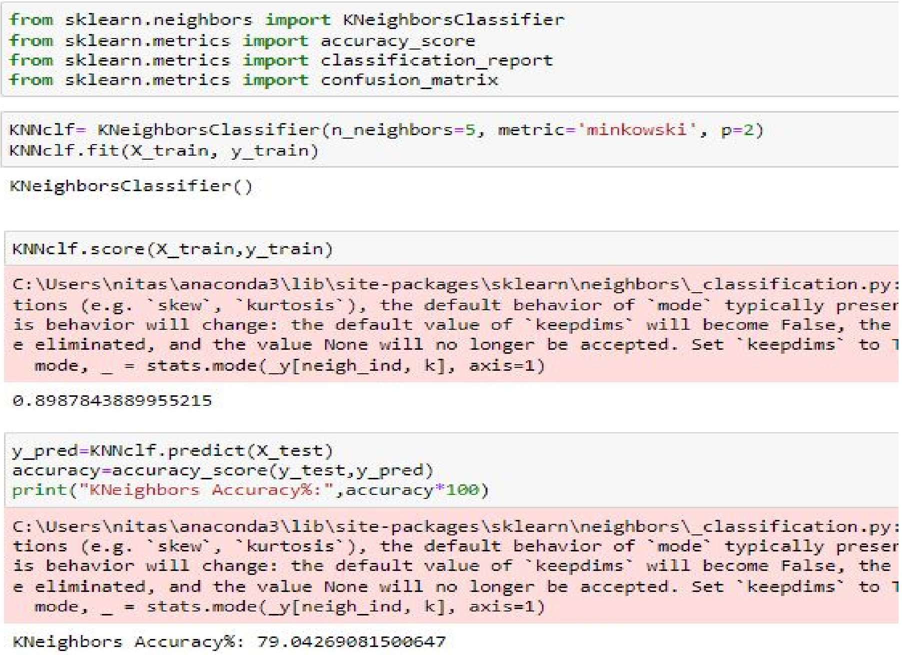
```
Decision Tree Accuracy%: 98.12916708130163
```

5.3.2: Decision Tree Algorithm

Decision Tree Algorithm has predicated accuracy approx.98.13 percent by using T20 World Cup datasets.

- KNN Algorithm

  The k-nearest neighbors classifier (KNN) is a non-parametric supervised machine learning algorithm. It's distance-based: it classifies objects based on their approximate neighbors' classes. KNN is most often used for classification but can be applied to regression problems as well.



```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

```
KNNclf= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)
KNNclf.fit(X_train, y_train)
```

```
KNeighborsClassifier()
```

```
KNNclf.score(X_train,y_train)
```

```
C:\Users\nitas\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:
tions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preser
is behavior will change: the default value of `keepdims` will become False, the
e eliminated, and the value None will no longer be accepted. Set `keepdims` to T
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

```
0.8987843889955215
```

```
y_pred=KNNclf.predict(X_test)
accuracy=accuracy_score(y_test,y_pred)
print("KNeighbors Accuracy%:",accuracy*100)
```

```
C:\Users\nitas\anaconda3\lib\site-packages\sklearn\neighbors\_classification.py:
tions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preser
is behavior will change: the default value of `keepdims` will become False, the
e eliminated, and the value None will no longer be accepted. Set `keepdims` to T
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

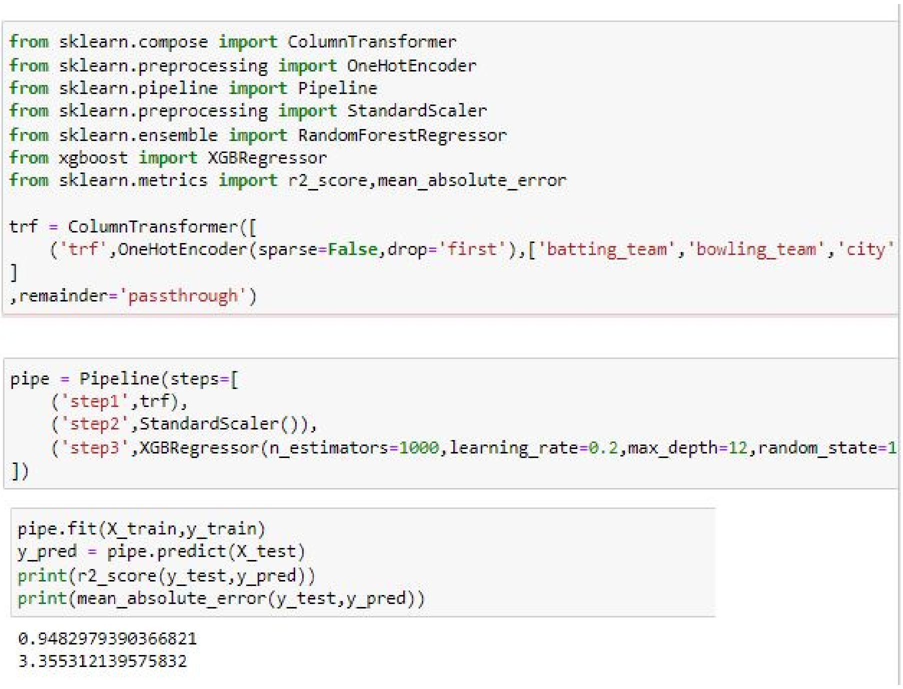
```
KNeighbors Accuracy%: 79.04269081500647
```

5.3.3: KNN Algorithm

KNeighbors Classifier Algorithm has predicated Accuracy approx. 79.04 percent by using T20 World Cup datasets.

- XGBRegressor Algorithm

  XGBoost is a powerful approach for building supervised regression models. The objective function contains loss function and a regularization term. It tells about the difference between actual values and predicted values, i.e how far the model results are from the real values.



```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import r2_score,mean_absolute_error

trf = ColumnTransformer([
    ('trf',OneHotEncoder(sparse=False,drop='first'),['batting_team','bowling_team','city'
]
,remainder='passthrough')
```

```
pipe = Pipeline(steps=[
    ('step1',trf),
    ('step2',StandardScaler()),
    ('step3',XGBRegressor(n_estimators=1000,learning_rate=0.2,max_depth=12,random_state=1
])
```

```
pipe.fit(X_train,y_train)
y_pred = pipe.predict(X_test)
print(r2_score(y_test,y_pred))
print(mean_absolute_error(y_test,y_pred))
```

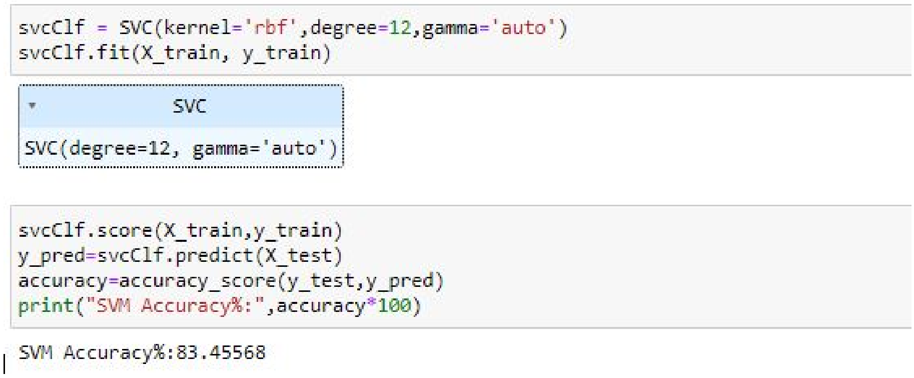
```
0.9482979390366821
3.355312139575832
```

5.3.4: XGBRegressor Algorithm

XGBRegressor Algorithm has predicated Accuracy approx. 94.82 percent by using T20 World Cup datasets.

- SVM Algorithm

  Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.



```
svcClf = SVC(kernel='rbf',degree=12,gamma='auto')
svcClf.fit(X_train, y_train)
```

```
SVC(degree=12, gamma='auto')
```

```
svcClf.score(X_train,y_train)
y_pred=svcClf.predict(X_test)
accuracy=accuracy_score(y_test,y_pred)
print("SVM Accuracy%:",accuracy*100)
```

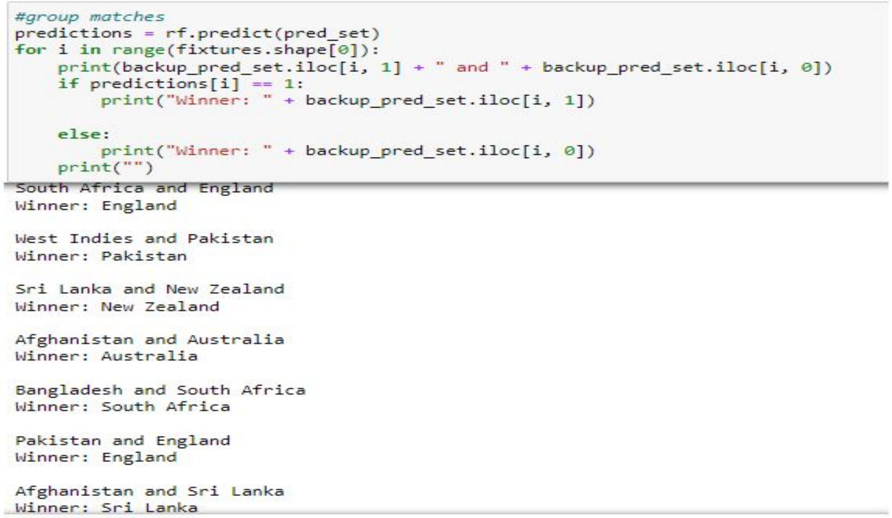
```
SVM Accuracy%:83.45568
```

5.3.5: SVM Algorithm

SVM Algorithm has predicated Accuracy approx. 83.45 percent by using T20 World Cup datasets.

5.4 Model Evaluation

- Predicated results between Team A and team B

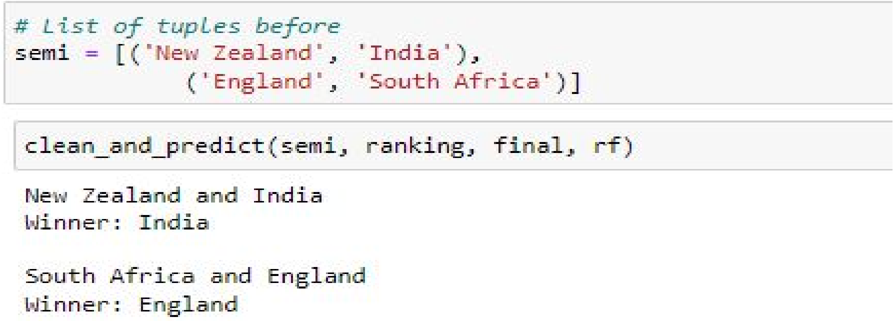


```
#group matches
predictions = rf.predict(pred_set)
for i in range(fixtures.shape[0]):
    print(backup_pred_set.iloc[i, 1] + " and " + backup_pred_set.iloc[i, 0])
    if predictions[i] == 1:
        print("Winner: " + backup_pred_set.iloc[i, 1])

    else:
        print("Winner: " + backup_pred_set.iloc[i, 0])
    print("")
```

```
South Africa and England
Winner: England

West Indies and Pakistan
Winner: Pakistan

Sri Lanka and New Zealand
Winner: New Zealand

Afghanistan and Australia
Winner: Australia

Bangladesh and South Africa
Winner: South Africa

Pakistan and England
Winner: England

Afghanistan and Sri Lanka
Winner: Sri Lanka
```

5.4.1: Predicated results between Team A and team B

- Semifinal result between Team A and Team B



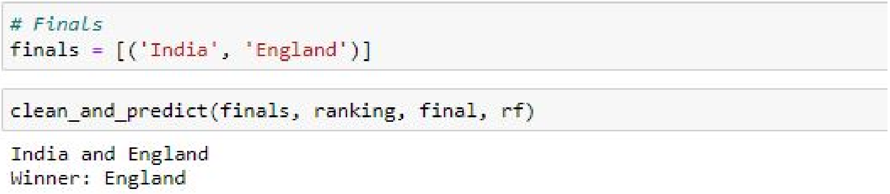
```
# List of tuples before
semi = [('New Zealand', 'India'),
            ('England', 'South Africa')]
```

```
clean_and_predict(semi, ranking, final, rf)
```

```
New Zealand and India
Winner: India

South Africa and England
Winner: England
```

5.4.2: Semifinal results between Team A and team B

- Final Results between Team A and Team B



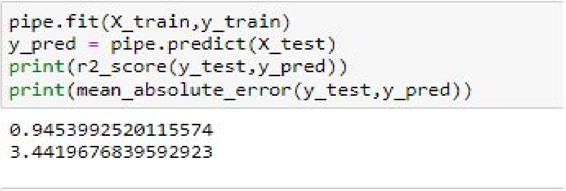
```
# Finals
finals = [('India', 'England')]
```

```
clean_and_predict(finals, ranking, final, rf)
```

```
India and England
Winner: England
```

5.4.3: Final results between Team A and team B

- Model evaluation of predicated datasets



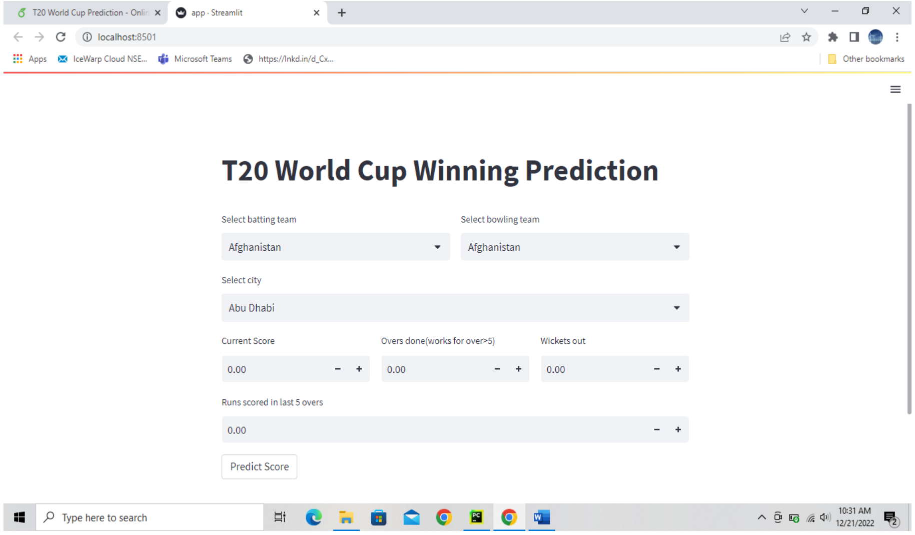
```
pipe.fit(X_train,y_train)
y_pred = pipe.predict(X_test)
print(r2_score(y_test,y_pred))
print(mean_absolute_error(y_test,y_pred))
```

```
0.9453992520115574
3.4419676839592923
```

5.4.4: Model evaluation of predicated datasets

5.5 User Interface Layouts

User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).



T20 World Cup Winning Prediction

Select batting team: Afghanistan

Select bowling team: Afghanistan

Select city: Abu Dhabi

Current Score: 0.00

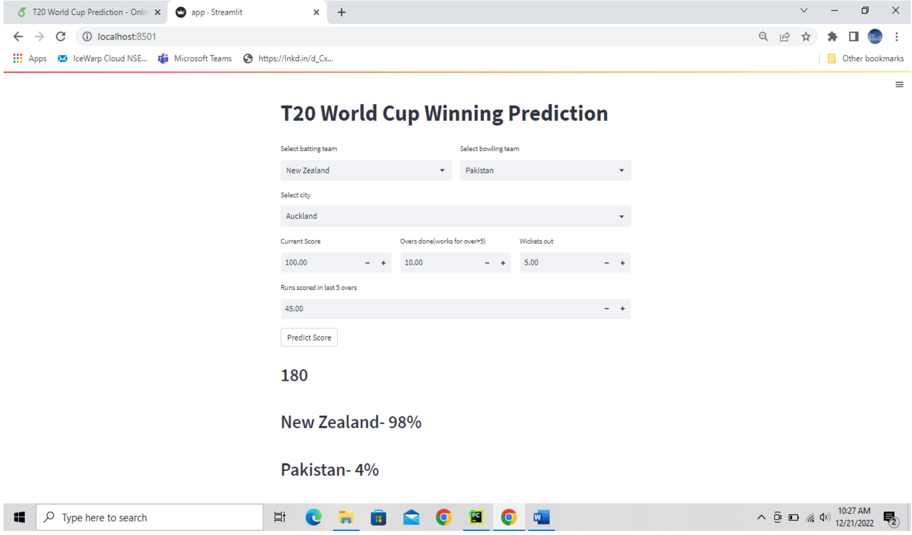
Overs done(works for over>5): 0.00

Wickets out: 0.00

Runs scored in last 5 overs: 0.00

Predict Score

5.5.1: User Interface of T20 World Cup



T20 World Cup Winning Prediction

Select batting team: New Zealand

Select bowling team: Pakistan

Select city: Auckland

Current Score: 100.00

Overs done(works for over>5): 10.00

Wickets out: 5.00

Runs scored in last 5 overs: 45.00

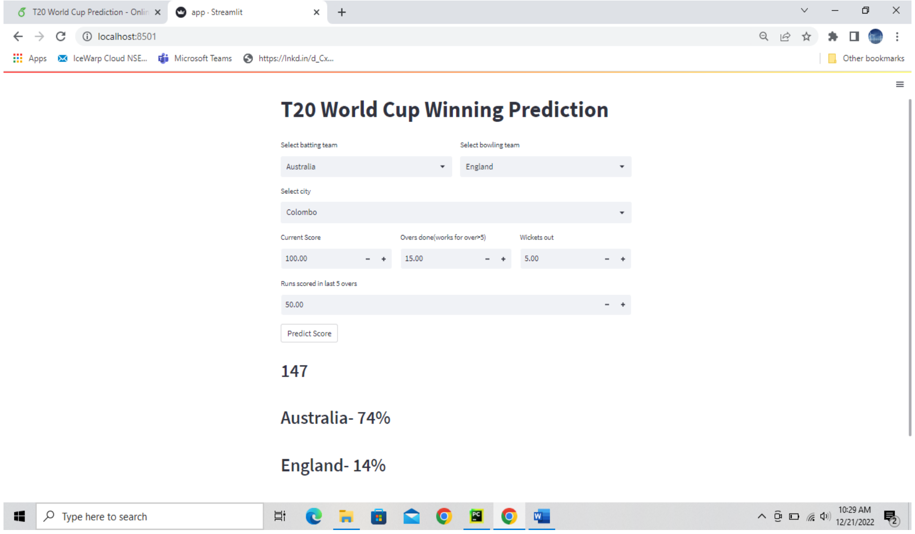
Predict Score

180

New Zealand- 98%

Pakistan- 4%

5.5.2: Winning Prediction Between New Zealand and Pakistan



T20 World Cup Winning Prediction

Select batting team: Australia

Select bowling team: England

Select city: Colombo

Current Score: 100.00

Overs done(works for over>5): 15.00

Wickets out: 5.00

Runs scored in last 5 overs: 50.00

Predict Score

147

Australia- 74%

England- 14%

5.5.3: Winning Prediction Between Australia and England

6 Future Enhancement

- I will use ReactJS framework as well as Streamlit API for developing user interface, predicts live score value as well as create a bar graph for two different teams that will display winning probability in terms of percentage for upcoming T20 World Cup.
- Enhance the accuracy of this model by applying efficient algorithms on training datasets and calculate accuracy level.
- And Finally deploy this system on AWS platform or Heroku.

7 Limitations

- When datasets are too large, then it is difficult to process all the datasets.
- While applying classified algorithms on the T20 World Cup datasets, some classified algorithms may not give accurate results.
- This Model predicts 80-90 percent accurate value using ICC T20 World Cup datasets. of 80-90 percent.

8 Conclusion

- In conclusion, This study is an exploratory study aimed to predict the winner of the upcoming ICC T20 World Cup by utilizing a few prominent parameters associated with T20 previous matches. A feature score was computed from the selected parameters to test the formulated hypothesis of predicting the probable winner.
- It is apparent from the accuracy comparison models that the Decision Tree algorithm gives the highest prediction as it is able to find the most important feature from the selected feature set and is less prone to overfitting.

9 Bibliography

9.1 Web References

[1.] https://ieeexplore.ieee.org/document/9642558
[2.] https://www.pyimagesearch.com/
[3.] https://scikit-learn.org/
[4.] https://stackoverflow.com/
[5.]https://keras.io/api/layers/pooling_layers/average_pooling2d/
[6.] https://keras.io/api/layers/reshaping_layers/flatten/
[7.] https://keras.io/api/layers/core_layers/dense/
[8.] https://keras.io/api/layers/activations/
[9.] https://keras.io/api/applications/mobilenet/
[10.] https://www.geeksforgeeks.org/