Getting Useful Information From Cricket Data Set Using Data Analytics Techniques Odi Cricket Team Performance Analysis Using Data Mining Classification Techniques

Dr Udayabhanu N P G Raju [a], Dr Pardeep Kumar [b], Dr Nara Sreekanth [c], Dr V Anbarasu[d] and Dr.K.Vengatesan[e]

[a] *Assistant Professor,Department of Computer Science and Engineering, Aditya College of Engineering, Andhra Prdesh. India.unpgraju_cse@acoe.edu.in*

[b] *Assistant Professor,Department of Artificial Intelligence, Anurag University, Hyderabad, Telangana State, India pardeepsep2@gmail.com*

[c]*Associate Professor,Department of Computer Science and Engineering,BVRIT HYDERABAD College of Engineering for Women,Telangana State, India sreekanth.n@bvrithyderabad.edu.in*

[d]*Associate Professor,Department of CSE, SOC,SRM institute of Science and Technology (SRMIST), Tamilnadu, Indiaanbarasukv@gmail.com*

[e]*Dr.K.Vengatesan,Professor, Department of Computer Engineering,Sanjivani College of Engineering, Kopargaon kvengatesancomp@sanjivani.org.in*

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: Data analytics is an important part of data science. It deals with analysing a data set and generating useful information from it. It is the basis for applying machine learning algorithms, prediction techniques, etc. In this work, ODI Cricket data set of the year 2018 is considered. From this data, useful information such as teams playing most of the matches, teams winning or losing most of the matches, monthly frequency of the matches, groundwise distribution is visualised in the graphical format. Also, a user can enter the name of a particular team and check its performance based on total matches played, total matches lost, total matches won in terms of runs or wickets, etc. This analysis is useful in predicting the performance of a team in future matches. This can also be used to improve the results in future matches.

Keywords: cricket analysis, ODI analysis, targeted searching, data analytics, cricket data

I Introduction

We are living in the 21st century, and the greatest invention has been the rise of Computer technology. With the increasing computer technology, the options that are available in front of the people have increased tremendously. The result of all this is: The data that is generated is of very large quantity. Each and every second people generate lots of data. This is known as big data. Most of the data is available in JSON and CSV formats on the Internet. From common people's point of view, the data present in CSV format or JSON format is not very easy to understand. Hence, data analysis is very important.

During the process of data analysis, the first step is always data wrangling or data cleaning. During this process, the unwanted or incomplete data is cleaned using various processes and methods. For example, consider cricket data set. In this data there can be some fields like winner of the match. It is not always important that one of the teams should always win the match. There can be certain scenarios in which none of the teams is the winner of the match. That is, there can be a tie or no result. In the CSV or JSON format this is mostly left as blank fields. Now this may create problems during the data analysis process. Hence it is important to go for data wrangling techniques and methodologies. in such cases, we replace blank fields with NULL values and then perform data analysis. If we do not replace the blank fields with null values, this may create problems while coding hence data wrangling or data cleaning is an important aspect of the data analysis process. There are many languages that can be used for data science. In this work we will be dealing with our given data set using Python 3.x. Python has got many inbuilt libraries for helping in data science. For example, the “csv” module. This module has got several functions that can be used to deal with CSV files. This makes the job of data analysis easier.

In this work, we have taken the ODI Cricket data set. It has got various fields like Team 1, Team 2, Winner, Margin, Ground, Match Date, and Scorecard. Team 1 and Team 2 respectively show the two teams that are participating in the match. The winner attribute shows the name of the winner team. The winner's name is shown precisely if there was no tie or any other problem for a particular match. The margin attributes shows by how much margin a particular team won the match. To put it simply, margin attribute shows by how many runs or by how many wickets a particular team won the match. The ground attributes shows the name of the ground on which the match was played. The match date attribute shows the dates on which matches were played during the year 2018. This column includes all the values from 1st January upto the end of the year. The scorecard attribute

shows the cardinality of an ODI match that is from #3946 to #4073 that is a total of 128 matches were played during the year 2018.
Using this data we can get useful information like the number of times a particular team won the matches in total. Through this data we can easily predict what could be the future results. And hence we can predict the performance of a particular team in the future matches. As we are having the winner's column, we can also do some manipulation and get the list of the teams who lost the matches. We can also know the team's performance, that is, whether a team performs well on home ground or foreign ground. This analysis can be useful to have a better result in the future. Through the winner's field, we can come to know the strengths and weaknesses of a particular team and hence results can be enhanced in the future by working on these strengths and weaknesses.
Finally, we can plot this data using various graphs and hence we can see at a glance the performance of various teams. The teams which won most of the ODI matches, the teams which lost most of the ODI matches, ground wise distribution, month wise distribution of matches, etc. Also, we can add functionality to check the performance of a particular team. The user needs to input the name of a team for searching. The program goes through all the records of our data and returns the list of various parameters such as the total number of matches won by the team, the total number of matches lost by the team, total number of matches that were a tie, etc. This analysis can be useful to predict the future performance of a particular team in the matches.

Work Flow Diagram:

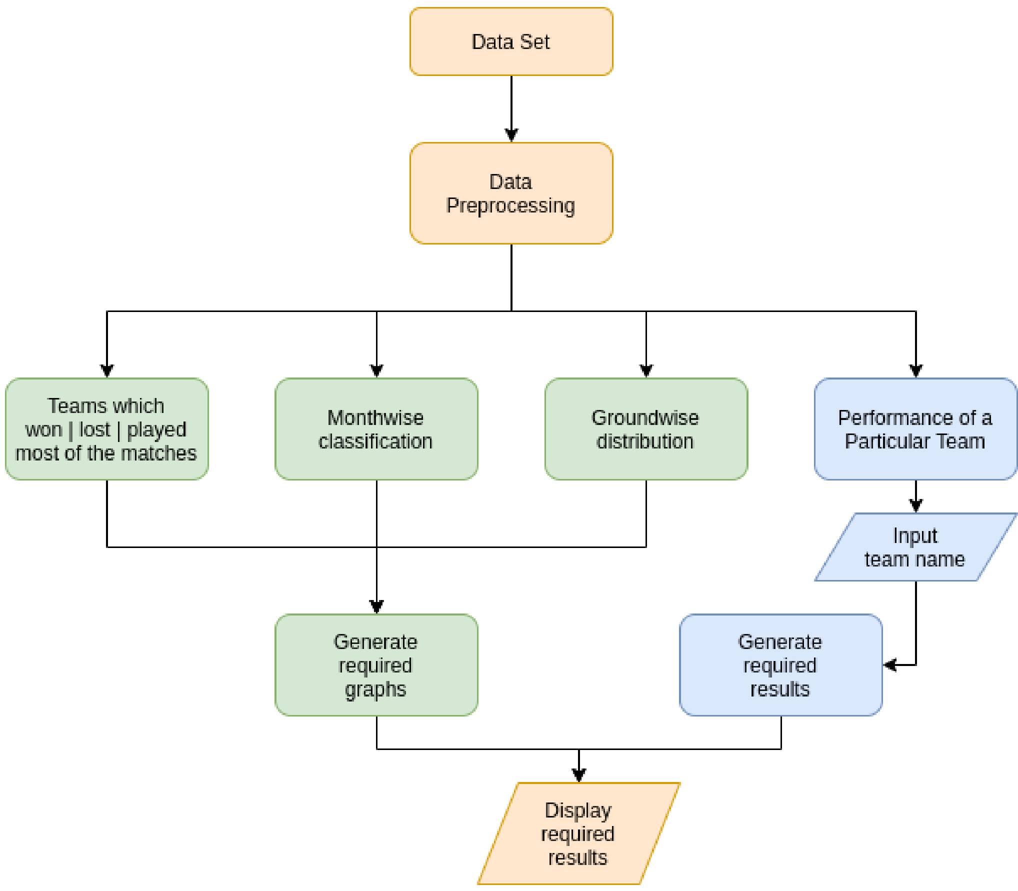


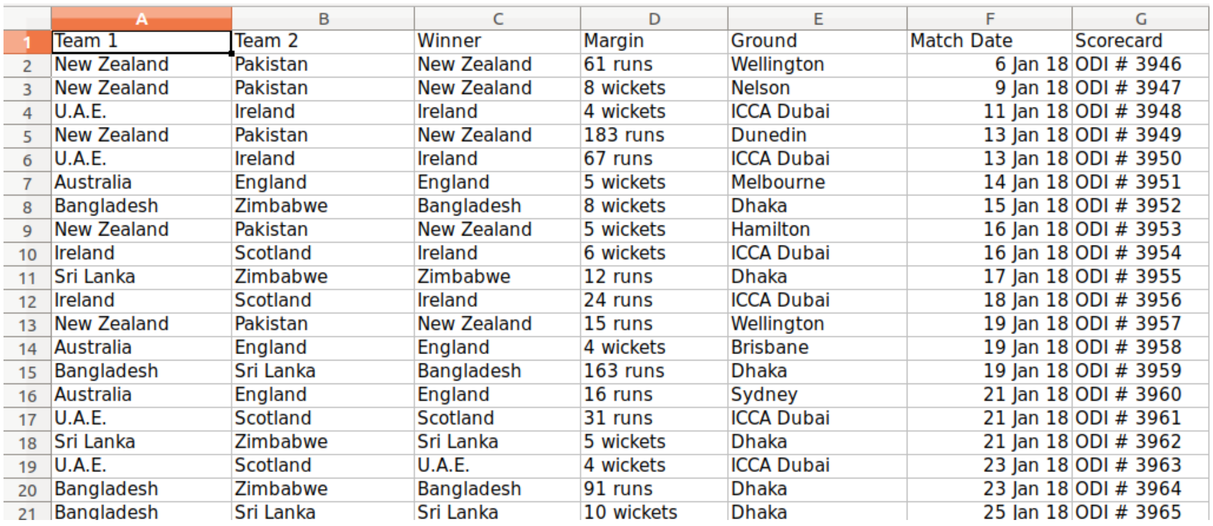
Figure 1: Workflow Diagram

[Fig. 1] represents the workflow diagram of this work. The first step in any data analytics work is data preprocessing. This involves removing unwanted (or not-so-useful) fields (or rows) from the dataset. In the considered ODI Cricket data set, there are some blank fields. Using Python language, we have handled them using NULL string. This gets the job done. Further, various graphs are generated using matplotlib library in Python. These include: Teams which played most of the matches, teams which won most of the matches, teams which lost most of the matches, ground wise distribution of matches, monthly frequency of matches, etc. These are useful in getting useful information from the considered data set at a glance. Also, an additional feature is provided to the user for checking the performance of a particular team. The user just needs to enter the name of a team and the program outputs various parameters such as total number of matches played, total matches lost, total matches won, matches won by runs or wickets, total numbers of matches that resulted in a tie or no result, etc.

This analysis is useful to predict the performance in the future. Also, studies can be done to improve a team's performance.

Results and Discussions:

In the considered data set, there are various fields such as Team 1, Team 2, Winner, Margin, Ground, Match Date, Scorecard. This is a CSV file.The Team 1 and Team 2 fields represent the teams that are playing the match. The winner field represents the winner of the match. There can also be a tie or no result. This also has to be taken care. The Margin field represents the winning criteria (runs or wickets). In case of a tie or no result, this is left blank (NULL string). The ground field represents the name of the ground on which the match was played. Match date and scorecard represent the date of the match and the cardinality of the match from the point ODIs were started since 1971. This data set has a total of 7 columns and 128 rows. [Fig. 2]



	A	B	C	D	E	F	G
1	Team 1	Team 2	Winner	Margin	Ground	Match Date	Scorecard
2	New Zealand	Pakistan	New Zealand	61 runs	Wellington	6 Jan 18	ODI # 3946
3	New Zealand	Pakistan	New Zealand	8 wickets	Nelson	9 Jan 18	ODI # 3947
4	U.A.E.	Ireland	Ireland	4 wickets	ICCA Dubai	11 Jan 18	ODI # 3948
5	New Zealand	Pakistan	New Zealand	183 runs	Dunedin	13 Jan 18	ODI # 3949
6	U.A.E.	Ireland	Ireland	67 runs	ICCA Dubai	13 Jan 18	ODI # 3950
7	Australia	England	England	5 wickets	Melbourne	14 Jan 18	ODI # 3951
8	Bangladesh	Zimbabwe	Bangladesh	8 wickets	Dhaka	15 Jan 18	ODI # 3952
9	New Zealand	Pakistan	New Zealand	5 wickets	Hamilton	16 Jan 18	ODI # 3953
10	Ireland	Scotland	Ireland	6 wickets	ICCA Dubai	16 Jan 18	ODI # 3954
11	Sri Lanka	Zimbabwe	Zimbabwe	12 runs	Dhaka	17 Jan 18	ODI # 3955
12	Ireland	Scotland	Ireland	24 runs	ICCA Dubai	18 Jan 18	ODI # 3956
13	New Zealand	Pakistan	New Zealand	15 runs	Wellington	19 Jan 18	ODI # 3957
14	Australia	England	England	4 wickets	Brisbane	19 Jan 18	ODI # 3958
15	Bangladesh	Sri Lanka	Bangladesh	163 runs	Dhaka	19 Jan 18	ODI # 3959
16	Australia	England	England	16 runs	Sydney	21 Jan 18	ODI # 3960
17	U.A.E.	Scotland	Scotland	31 runs	ICCA Dubai	21 Jan 18	ODI # 3961
18	Sri Lanka	Zimbabwe	Sri Lanka	5 wickets	Dhaka	21 Jan 18	ODI # 3962
19	U.A.E.	Scotland	U.A.E.	4 wickets	ICCA Dubai	23 Jan 18	ODI # 3963
20	Bangladesh	Zimbabwe	Bangladesh	91 runs	Dhaka	23 Jan 18	ODI # 3964
21	Bangladesh	Sri Lanka	Sri Lanka	10 wickets	Dhaka	25 Jan 18	ODI # 3965

Figure 2: Considered Cricket Data Set

We can visualise this data in the form of graphs for a better understanding of the data. In python, we first use **"csv" module** to load our dataset into the memory of the system. Now, we need to skip the first line (header) as it represents the titles of the various fields. This can be achieved as follows:

headers=next(data)

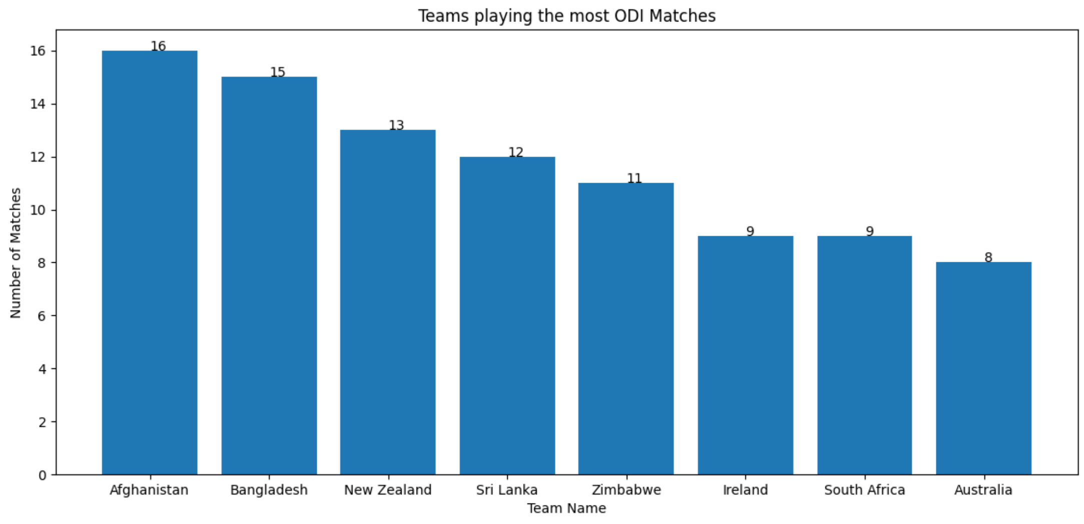
Now, we can manipuate the data and perform necessary operations to get useful information like which team won the maximum number of matches, which team lost maximum number of matches, which teams played the most number of matches, etc. This information can be used to predict the future results of the matches based on the historic preformance. Data can be visualized in the following ways:

I. Teams playing maximum number of ODI matches:

This analysis helps us to know which are the teams that have played the most number of ODI matches. For plotting this graphically, we have used **"matplotlib"library** in Python. The generated graph is shown in[Fig. 3].

Figure 3: Teams playing the most ODI matches

[Fig. 3] From the graph, we can conclude that Afghanistan Team played the most number of matches, ie. 16. This data has been sorted in descending number of matches played for a better understanding of the user.



II. Teams which won the most ODI Matches

This analysis is plotted using ***plot()*** function available in the matplotlib library. This is done for a better visibility to the user.

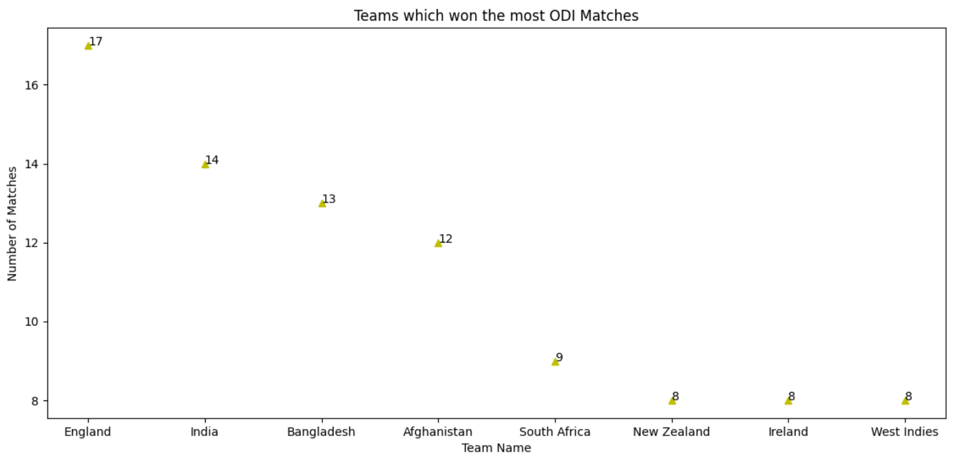


Figure 4:

Teams which won the most ODI Matches

[Fig. 4] From the graph, we can conclude that the team which won the most number of matches is none other than England with India ranking at the second position. England won a total of 17 matches. Here, care has been taken to exclude the matches whose result was either **tie** or **no result.** Hence, only the precise teams which actually won the match regardless of tie and no result are displayed.

III. Month wise Distribution of the Matches

This analysis helps in understanding the monthly frequency of matches played. Various factors such as seasons, climatic conditions, festivals, political scenarios, elections, etc. have an impact on this analysis. Hence, this analysis can also be used in future to predict the frequency of the future matches. We have used dashed line graph for this analysis.

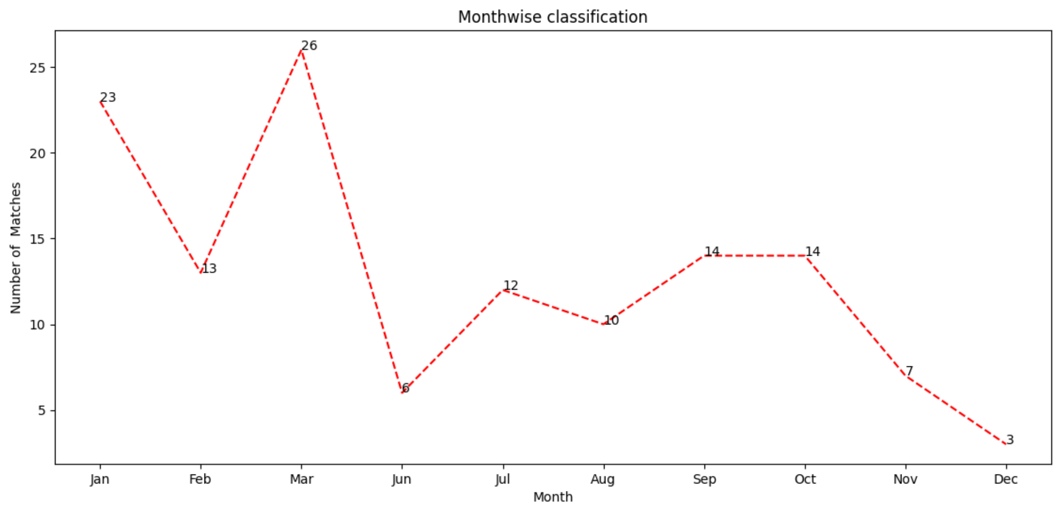
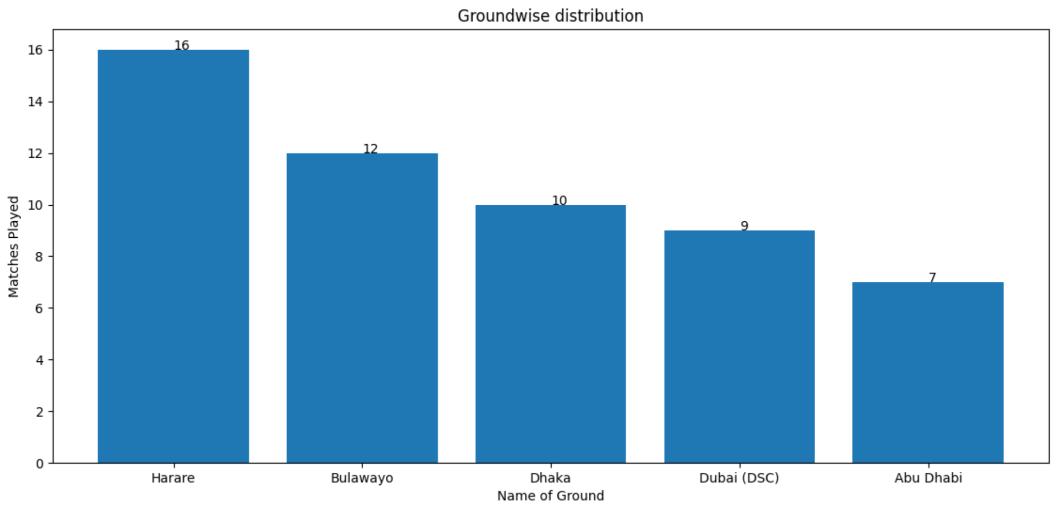


Figure 5: Month wise Distribution of the Matches

[Fig. 5] From the graph we can conclude that the maximum matches were played in the month of March (ie. 26 matches). The frequency of matches gradually decreased in the months of June to August as it is rainy season in most of the countries.

IV. Ground wise Distribution

This analysis is done to know which the most preferred grounds are for ODI matches. Various studies are possible on this analysis so as to know which grounds need to be worked on, why a particular ground is preferred



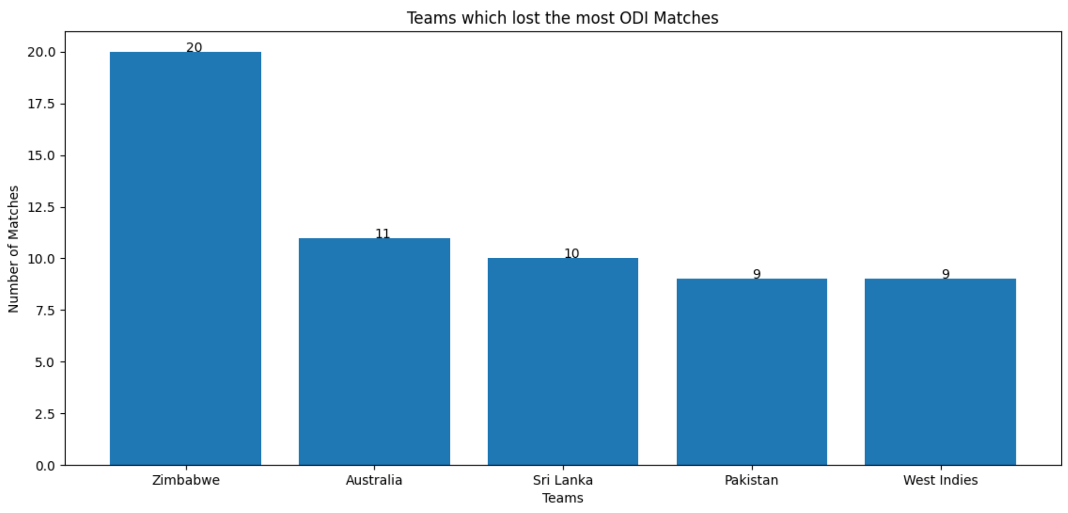
more, etc.

Figure 6: Groundwise Distribution

[Fig. 6] From the graph, we can clearly see that the ground of Harare was preferred the most for 2018 ODI matches. Further studies can be done on this analysis.

V. Teams which lost most of the ODI matches

This is a very important analysis. This field is not present in our data set. We need to do manipulation to get this



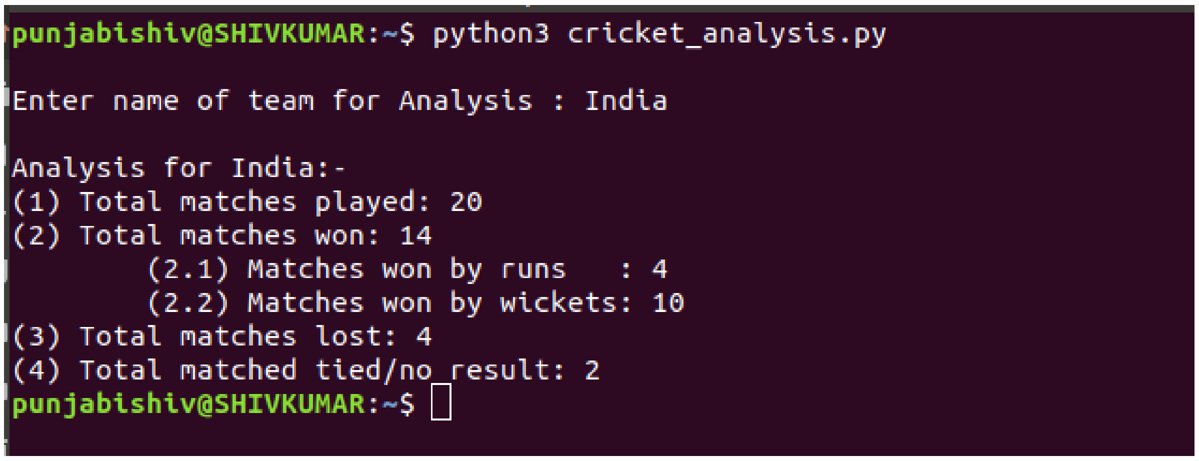
information. Here, care has been taken to not include the matches that resulted in a **tie** or **no result.** Because, this may give wrong impression about a particular team.

Figure 7: Teams which lost most of the ODI matches

[Fig. 7] This analysis can be used to know the weaknesses of a particular team which can further help to have better results in the future matches.

VI. Performance of a Particular Team

This is a part of information gathering. The user can enter the name of a particular team to search for the performance of that team. This includes: Total number of matches played, total number of matches won, matches



```
punjabishiv@SHIVKUMAR:~$ python3 cricket_analysis.py

Enter name of team for Analysis : India

Analysis for India:-
(1) Total matches played: 20
(2) Total matches won: 14
        (2.1) Matches won by runs    : 4
        (2.2) Matches won by wickets: 10
(3) Total matches lost: 4
(4) Total matched tied/no_result: 2
punjabishiv@SHIVKUMAR:~$
```

won by runs/wickets, total number of matches lost, total matches which resulted in a tie or no result. This code is written in Python.

Figure 8: Performance of a Particular Team

[Fig. 8] There is also error handling in case a user enters the wrong name of a team. Appropriate message is shown to the user. This analysis is useful to predict the performance of a particular team in the future matches.

Pseudo Code:

```
import matplotlib.pyplot as plt
import csv, operator
fo1=open("data.csv")
data=csv.reader(fo1,delimiter="|")
most_odi=dict()
most_odi_winner=dict()
most_odi_loser=dict()
dates=dict()
grounds=dict()
countries=set()

fo1.seek(0)
headers=next(data)
for row in data:
    team_1=row[0]
    team_2=row[1]
    winner=row[2]
    date=row[5]
    divided_date=date.split()
    month=divided_date[1]
    if winner==team_1:
        loser=team_2
        most_odi_loser[loser]=most_odi_loser.get(loser,0)+1
        try:
            most_odi_winner[winner]+=1
        except:
            most_odi_winner[winner]=1
    elif winner==team_2:
        loser=team_1
        most_odi_loser[loser]=most_odi_loser.get(loser,0)+1
        try:
            most_odi_winner[winner]+=1
        except:
            most_odi_winner[winner]=1

    most_odi[team_1]=most_odi.get(team_1,0)+1
    dates[month]=dates.get(month,0)+1
    grounds[row[4]]=grounds.get(row[4],0)+1

#Graph 1: Teams playing the most ODI Matches
sorted_d = dict(sorted(most_odi.items(), key=operator.itemgetter(1), reverse=True))
x=list(sorted_d.keys())[:8]
y=list(sorted_d.values())[:8]
plt.bar(x,y)
plt.xlabel("Team Name")
plt.ylabel("Number of Matches")
plt.title("Teams playing the most ODI Matches")
for a,b in zip(x, y):
plt.text(a, b, str(b))
plt.show()

#Graph 2: Teams which won the most ODI Matches
sorted_d = dict(sorted(most_odi_winner.items(), key=operator.itemgetter(1), reverse=True))
x=list(sorted_d.keys())[:8]
```

```
y=list(sorted_d.values())[:8]
plt.plot(x,y,"y^")
plt.xlabel("Team Name")
plt.ylabel("Number of Matches")
plt.title("Teams which won the most ODI Matches")
for a,b in zip(x, y):
plt.text(a, b, str(b))
plt.show()

#Graph 3: Monthwise classification
x=list(dates.keys())
y=list(dates.values())
plt.plot(x, y, "r--")
plt.xlabel("Month")
plt.ylabel("Number of  Matches")
plt.title("Monthwise classification")
for a,b in zip(x, y):
plt.text(a, b, str(b))
plt.show()

#Graph 4: Groundwise distribution
sorted_d = dict(sorted(grounds.items(), key=operator.itemgetter(1), reverse=True))
x=list(sorted_d.keys())[:5]
y=list(sorted_d.values())[:5]
plt.bar(x,y)
plt.xlabel("Name of Ground")
plt.ylabel("Matches Played")
plt.title("Groundwise distribution")
for a,b in zip(x, y):
plt.text(a, b, str(b))
plt.show()

#Graph 5: Teams which lost the most ODI Matches
sorted_d = dict(sorted(most_odi_loser.items(), key=operator.itemgetter(1), reverse=True))
x=list(sorted_d.keys())[:5]
y=list(sorted_d.values())[:5]
plt.bar(x,y)
plt.xlabel("Teams")
plt.ylabel("Number of Matches")
plt.title("Teams which lost the most ODI Matches")
for a,b in zip(x, y):
plt.text(a, b, str(b))
plt.show()

#Performance of a Particular Team
team_name=input("Enter name of team for Analysis : ")
fo1.seek(0)
headers=next(data)
total_matches=0
matches_won=0
matches_lost=0
matches_won_by_runs=0
matches_won_by_wickets=0
tied_matches=0
flag=0
for row in data:
        if row[0]==team_name or row[1]==team_name:
                flag=1
                total_matches+=1
                if row[2]==team_name:
                        matches_won+=1
                        winning_criteria=row[3]
                        judgement=winning_criteria.split()
                        if judgement[1]=="runs":
                                matches_won_by_runs+=1
                        elif judgement[1]=="wickets":
                                matches_won_by_wickets+=1
                elif row[2]==row[0] or row[2]==row[1]:
                        matches_lost+=1
                else:
                        tied_matches+=1
if flag==0:
        print("No match found.")
else:
        print("\nAnalysis for "+team_name+":-")
        print("(1) Total matches played:",total_matches)
        print("(2) Total matches won:",matches_won)
        print("\t(2.1) Matches won by runs   :",matches_won_by_runs)
        print("\t(2.2) Matches won by wickets:",matches_won_by_wickets)
        print("(3) Total matches lost:",matches_lost)
        print("(4) Total matched tied/no result:",tied_matches)
```

Conclusion:
Data analytics is an important aspect in today’s world where big data is generated by the users. To know some useful information from the data, it is important to perform data analysis. This is a part of data science. In this work, data set of ODI Cricket matches is considered. This data is in the form of CSV file. Python has been used for analysing the data in this file. The data is also visualized in the form of the graphs. Various graphs such as the teams which won most of the ODI matches, the teams which lost most of the ODI matches, which teams played most of the matches, monthly frequency of matches, etc. are represented. Also, there is a functionality to check the performance of a particular team. The user needs to input the name of a team for searching. The program goes through all the records our data and returns the list of various parameters such as the total number of matches won by the team, the total number of matches lost by the team, total number of matches that resulted in a tie, etc. This analysis can be used to predict the performance of a particular team in the future matches.

References
1. Sohail Akhtar, Philip Scarf, “Forecasting test cricket match outcomes in play”, International Journal of Forecasting, Volume 28, Issue 3, July–September 2012, Pages 632-643.
2. Uday Damodaran, “Stochastic Dominance and Analysis of ODI Batting Performance: the Indian Cricket Team, 1989-2005”, The 8th Australasian Conference on Mathematics and Computers in Sport, 3-5 July 2006, Queensland, Australia.
3. Amal Kaluarachchi, S. Varde Aparna, “CricAI: A classification-based tool to predict the outcome in ODI cricket”, 2010 Fifth International Conference on Information and Automation for Sustainability.
4. Madan Gopal Jhawar, Vikram Pudi, “Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach”, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2016).
5. Michael Hynds, Ian Smith, “The demand for test match cricket”, Applied Economics Letters,Volume 1, 1994 - Issue 7.
6. Ambeth Kumar, V.D. et al. "Active volume control in smart phones based on user activity and ambient noise". Sensors (Switzerland) 20. 15(2020): 1-17.
7. Vengatesan, K. et al. "An approach of sales prediction system of customers using data analytics techniques". Advances in Mathematics: Scientific Journal 9. 7(2020): 5049-5056.
8. Srinivas, K. et al. "An implementation of subsidy prediction system using machine learning logistical regression algorithm". Advances in Mathematics: Scientific Journal 9. 6(2020): 3407-3415.
9. Vengatesan, K. et al. "Analysis of Mirai Botnet Malware Issues and Its Prediction Methods in Internet of Things". Lecture Notes on Data Engineering and Communications Technologies 31. (2020): 120-126.
10. Vimal, V. et al. "Artificial intelligence-based novel scheme for location area planning in cellular networks". Computational Intelligence. (2020).

11. Kumar, A. et al. "Black hole attack detection in vehicular ad-hoc network using secure AODV routing algorithm". Microprocessors and Microsystems. (2020).
12. Kumar, A. et al. "Comparative Analysis of Data Mining Techniques to Predict Heart Disease for Diabetic Patients". Communications in Computer and Information Science 1244 CCIS. (2020): 507-518.
13. Vengatesan, K. et al. "Credit card fraud detection using data analytic techniques". Advances in Mathematics: Scientific Journal 9. 3(2020): 1185-1196.
14. Preetha, J. et al. "Data mining technique based critical disease prediction in medical field". Advances in Parallel Computing 37. (2020): 104-108.
15. Kumar, V.D.A. et al. "Exploration of an innovative geometric parameter based on performance enhancement for foot print recognition". Journal of Intelligent and Fuzzy Systems 38. 2(2020): 2181-2196.
16. Vengatesan, K. et al. "Intrusion detection framework using efficient spectral clustering technique". Advances in Parallel Computing 37. (2020): 98-103.
17. Vikrant, S. et al. "M-learning for promoting advancements in agriculture: An innovative educational model for ethiopian farmers". Advances in Mathematics: Scientific Journal 9. 7(2020): 5057-5064.
18. Vengatesan, K. et al. "Secure Data Transmission Through Steganography with Blowfish Algorithm". Lecture Notes on Data Engineering and Communications Technologies 35. (2020): 568-575.
19. Lone, T.A. et al. "Securing communication by attribute-based authentication in HetNet used for medical applications". Eurasip Journal on Wireless Communications and Networking 2020. 1(2020).
20. Vengatesan, K. et al. "Simple Task Implementation of Swarm Robotics in Underwater". Lecture Notes on Data Engineering and Communications Technologies 35. (2020): 1138-1145.
21. Mallikalava, V. et al. "Theft vehicle detection using image processing integrated digital signature-based ECU". Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020. (2020): 913-918.
22. Kesavan, S. et al. "An investigation on adaptive HTTP media streaming Quality-of-Experience (QoE) and agility using cloud media services". International Journal of Computers and Applications. (2019).
23. Kumar, A. et al. "Teaching literacy through animation & multimedia". International Journal of Innovative Technology and Exploring Engineering 8. 5(2019): 73-76.
24. Kumar, A. et al. "3D Lighting Courseware development for 3D Motion Picture Science". 2018 International Conference on Recent Innovations in Electrical, Electronics and Communication Engineering, ICRIEECE 2018. (2018): 2621-2623.
25. Kumar, A. et al. "Study and Research of 3D Animation Courseware Development". 2018 International Conference on Recent Innovations in Electrical, Electronics and Communication Engineering, ICRIEECE 2018. (2018): 2514-2516.
26. Carter M., Guthrie G. (2004) Cricket interruptus: fairness and incentive in limited overs cricket matches. Journal of the Operational Research Society 55, 822-829.
27. Clarke S.R. (1988) Dynamic Programming in one-day cricket- optimal scoring rates. Journal of the Operational Research Society 39, 331-337.
28. Kimber A.C., Hansford A.R. (1993) A statistical analysis of batting in cricket. Journal of the Royal Statistical Society: Series A (Statistics in Society) 156, 443-455.
29. Lewis A.J. (2005) Towards fairer measures of player performance in one-day cricket. Journal of the Operational Research Society 56, 804-815.
30. Preston I., Thomas J. (2002) Batting strategy in limited overs cricket. Journal of the Royal Statistical Society: Series D (The Statistician) 51, 189-202.
31. Swartz T.B., Gill P.S., Beaudoin D., de Silva B.M. (2006) Optimal batting orders in one-day cricket. Computers and Operations Research 33, 1939-1950.