Analyzing IPL dataset with MongoDB

Kopal Chaudhary
Department of Computer Science
Jaypee Institute of Information Technology
Noida, India
chaudharykopal1994@gmail.com

Mohini Gupta
Department of Computer Science
Jaypee Institute of Information Technology
Noida, India
mohinigupta125@gmail.com

Dr. Parmeet kaur
Department of Computer science
Jaypee Institute Of Information Technology
Noida, India
parmeet.kaur@jiit.ac.in

Abstract— **Development has emerged use of NoSQL databases for storage and retrieval of big data. A highly popular NoSQL database is the document-oriented database, MongoDB which has superseded the relational databases in numerous applications. This paper investigates the querying performance of MongoDB using a dataset from the popular Indian cricket league, IPL (Indian Premier League). The present work analyzes the IPL dataset using MongoDB queries for determining the attributes that are important for a team in order to win a match. The experimental results have shown that for considered dataset, it was observed from the performed analysis that the parameters of winning a toss, playing at home grounds and application of Duckworth Lewis rule plays an important role in determining the outcome of a match**.

Keywords— MongoDB; NoSQL; Document-oriented; scalable; BSON format.

I. Introduction

NoSQL databases have been called the future of data economy due to their intrinsic features suitable for big data applications. Since these databases provide scalability and flexibility, they are already in use by many enterprises such as Facebook, eBay, Twitter etc. Existing data can be migrated from relational databases into NoSQL products as MongoDB with quite an ease and therefore many companies are shifting their projects to use MongoDB instead of traditional SQL database.NoSQL databases are classified into four major types, namely, document-based (MongoDB, Couch DB etc.), Column-based (like Cassandra, HBase etc.), key-value pair storage (like Voldemorte, Aerospike etc.) and graph databases (likeOrientDB, HypergraphDB).[1,2] This variety of databases gives a wide choice in data storage and processing models. A developer can select the type of database that is most appropriate for his use case. This results in a higher ease and flexibility of development as compared to the traditional relational database systems.

MongoDB is one of the most popular NoSQL databases [5]. It is an open-source document-oriented database that provides a format of storage called BSON (Binary JSON) format. A MongoDB collection contains a set of documents, each of which is further comprised of key-value pairs. A document can have any fundamental data type, such as dates, arrays, numbers, strings, or an embedded sub-document [4]. Although MongoDB does not provide joins like relational databases, it allows embedding of documents to provide similar functionality with lower latency. Its distributed nature is due to its features of sharding and replication. Sharding is the horizontal partitioning of documents across multiple nodes while replication is maintaining multiple copies of data for high availability and fault tolerance. Since MongoDB provides flexibility, scalability, and availability for data models, applications can be continuously enhanced and delivered at different scale.

The present work utilizes MongoDB for an empirical analysis of data related to the popular Indian cricket league, IPL (Indian Premier League) [6]. The objective is to identify the factors on which the winner of a league match depends upon. Results of this work can be used to base winner prediction algorithms in future.

The remainder of paper is structured as follows: Section II discusses the related work in the field of analysis using MongoDB. The methodology of the empirical study is presented in Section III. Results are put forth and discussed in Section IV. The last section concludes the paper.

II. Related Work

MongoDB have gained considerable interest in a short span as they have been developed specifically for unstructured and huge volumes of data. As a result[15], they involve faster read and write operations as compared to relational databases. The work in [1] evaluates different NoSQL databases based on their features and data model. Key-Value stores, as Riak, document-oriented databases, as MongoDB, and column-based families, as Cassandra, have been compared to each other. The NoSQL databases do not hold ACID properties and as per the CAP theorem [3], MongoDB is consistent and partition tolerant (CP). NoSQL databases do not have common query language. Each database behaves differently and so users can choose anyone of these according to their application's requirements [2].

Differences between MongoDB and the Oracle database system have been illustrated in [4]. The NoSQL based system has been compared with the SQL based Oracle on basis of their architecture, features, techniques for data distribution and query fetch times.

With the objective of describing the MongoDB system, the authors of [5] have presented a formal abstraction of MongoDB's query language. Several research efforts have utilized MongoDB for data analysis. The work analyzes Twitter data with R and MongoDB operations like aggregation. Another application is discussed in [6] which is an online water resources monitoring system. This system had requirements for storage and processing of volumes of data and hence, MongoDB was deemed appropriate. The entire data can be

978-1-5386-5933-5/19/$31.00 ©2019 IEEE

Authorized licensed use limited to: MEHRAN UNIV OF ENGINEERING AND TECHNOLOGY. Downloaded on December 21,2023 at 03:48:38 UTC from IEEE Xplore. Restrictions apply.

stored in a single document or it may be stored in different documents which can be subsequently related using their fields as reference.

According to [13], MongoDB can handle dynamic schemas. Data with varying formats can be inserted into this database in the form of one field. This database can be used in circumstances when data is very large or when database structure is changed repeatedly. [10] mainly focuses on comparison between MongoDB and MySQL and at last justifies why MongoDB is preferred over MySQL.

The present work presents an interesting application for analysis of IPL data using MongoDB.

III. Empirical study

The objective of the empirical study is to analyze the factors that determine the winner of an IPL match. It is aimed to identify the attributes of the data set that influence the match winner.

A. Data Set

IPL dataset is taken from kaggle in CSV format. The dataset includes 17 columns and 636 rows of data collected from IPL seasons from 2008 to 2017. The link to the dataset is https://www.kaggle.com/manasgarg/ipl/

The various attributes present in the dataset are:

- Date: on which match was held.
- City: venue city
- Team1: name of first team
- Team2: name of second teamToss_winner: name of team which had won the toss
- Toss_decision: decision taken by toss_winner field that is either field or bat
- DL-applied: A method which is applied due to rain between the match value are either 0 or 1
- Result: this attribute tells whether its normal or tie
- Winner: name of team which won the match
- Win_by_runs: runs by which a team won
- Win_by_wicktes: wicktes by which a team won
- Player of the match: name of the player who won the match
- Venue: name of stadium
- Umpire1: name of first umpire
- Umpire2:name of second umpire

B. Methodology

Firstly the input data, i.e., IPL dataset was imported into a MongoDB collection. Subsequently, queries were designed and executed to analyze the stored data. The queries were designed to evaluate the effect of the parameters of winning a toss, batting or fielding first, application of Duckworth Lewis method and playing at the home ground

The queries are listed as follows:

1) Objective:

Count the number of times toss winner team wins the match.

MongoDB Query:

```
db.things.find({$where:"this.toss_winner==this.winner"})
```

2) Objective:

Count the number of times the toss winner teams wins after choosing to field first.

MongoDB Query:

```
db.things.find({$and:[{$where:"this.toss_winner==this.win
ner"},{"toss_decision":"field"}]}).
```

3) Objective:

Count the number of times toss winner teams wins after choosing to bat first.

MongoDB Query:

```
db.things.find({$and:[{$where:"this.toss_winner==this.win
ner "},{"toss_decision":"bat"}]})
```

4) Objective:

Count the number of times toss winner teams wins and chooses to field first in case Duckworth–Lewis method is applied (dl_applied)

MongoDB Query:

```
db.things.find({$and:[{$where:"this.toss_winner==this.win
ner"},{"toss_decision"="field"},{"dl_applied":1},]})
```

5) Objective:

Count the number of times toss winner teams wins and chooses bat to first when dl_applied

MongoDB Query:

```
db.things.find({$and:[{$where:"this.toss_winner==this.win
ner "},{"toss_decision"="bat"},{"dl_applied":1},]})
```

6) Objective:

Count no of times Mumbai Indians team have won the match in their home ground, i.e., Wankhede Stadium.

MongoDB Query:

```
db.things.find({$and:[{'venue":"Wankhade
Stadium"},{"winner":"Mumbai Indians"}]})
```

7) Objective:

Count no of times Chennai Super Kings have won the match in their home ground, i.e., MA Chidambaram Stadium Chepauk.

MongoDB Query:

Authorized licensed use limited to: MEHRAN UNIV OF ENGINEERING AND TECHNOLOGY. Downloaded on December 21,2023 at 03:48:38 UTC from IEEE Xplore. Restrictions apply.

db.things.find($and:[“venue”:” MA Chidambaram Stadium Chepauk”},{“winner”:”Chennai Super Kings”}]})

8) Objective:

Count no of times Sunrisers Hyderabad have won the match in their home ground, i.e., Rajiv Gandhi International Stadium Uppal.

MongoDB Query:

db.things.find($and:[“venue”:”Rajiv Gandhi International Stadium Uppal”},{“winner”:”Sunrisers Hyderabad”}]})

9) Objective:

Count no of times Kolkata Knight Riders have won the match in their home ground, i.e., Eden Gardens.

MongoDB Query:

db.things.find($and:[“venue”:”EdenGardens”},{“winner”:” Kolkata Knight Riders”}]})

10) Objective:

Count no of times Delhi Daredevils have won the match in their home ground, i.e., Feroz Shah Kotla.

MongoDB Query:

db.things.find($and:[“venue”:”FerozShahKotla”},{“winner ”:”Delhi Daredevils”}]})

IV. Experimental Outcomes

The results of execution of the queries are described as follows:

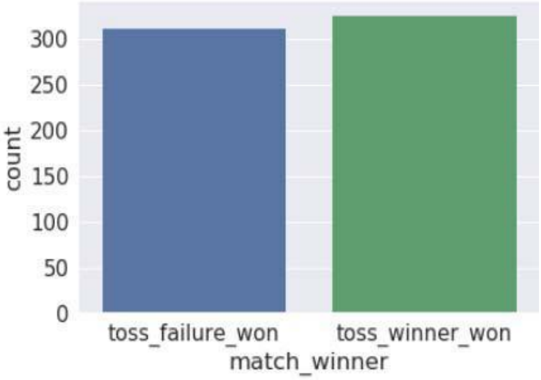

Fig.1 Influence of toss on match outcome

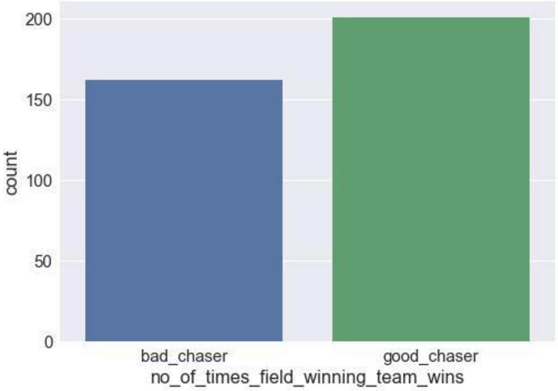

Fig.2 Count of times when team fielding first wins the match

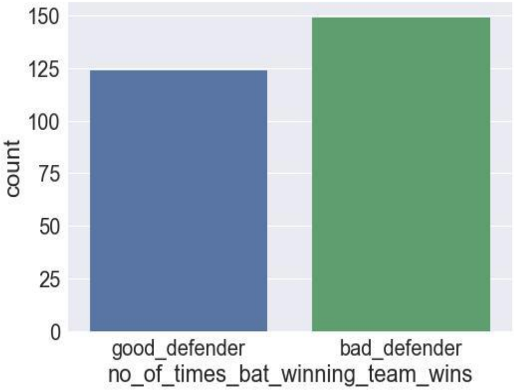

Fig.3 Count of times when team batting first wins the match

Fig.1 represents the effect of winning toss on the outcome of a match while Fig. 2, 3 shows whether the teams batting first or fielding first won more matches. It can be observed that that more teams won when they were chasing totals.

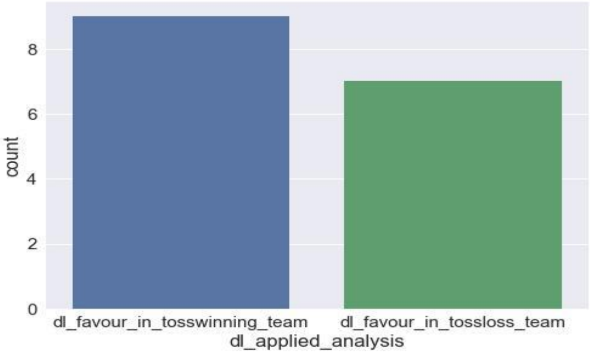

Fig.4 Influence of DL rule on match result

Authorized licensed use limited to: MEHRAN UNIV OF ENGINEERING AND TECHNOLOGY. Downloaded on December 21,2023 at 03:48:38 UTC from IEEE Xplore. Restrictions apply.

The result of the graph in Fig. 4 shows that when the Duckworth Lewis Rule is applied, it has favored the team fielding first more than the team batting first.

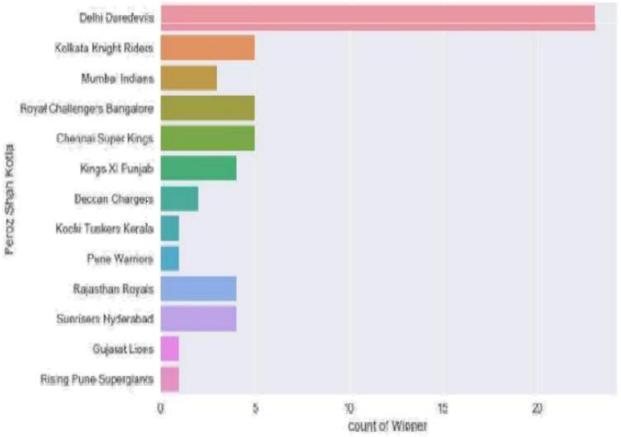

Fig 5 (a) Winning Teams at Feroz Shah Kotla Stadium

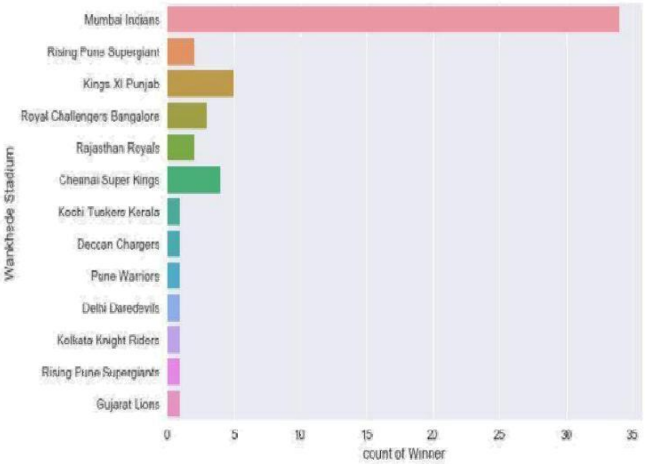

Fig 5 (b) Winning Teams at Wankhede Delhi

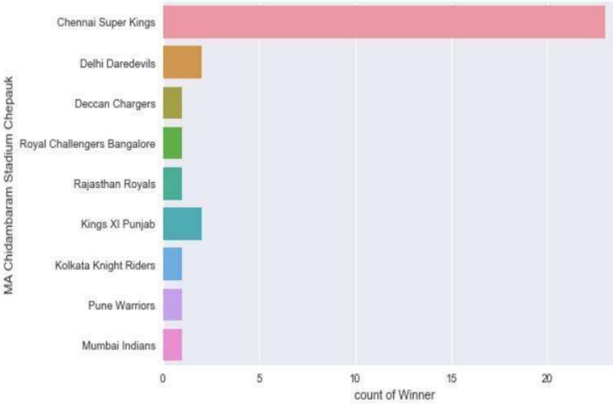

Fig 5(c) Winning Teams at MA Chidambaram Stadium Chepauk

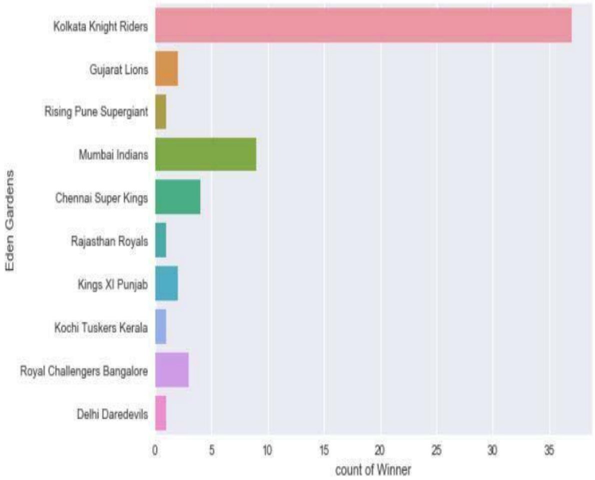

Fig 5 (d) Winning Teams at Eden Gardens

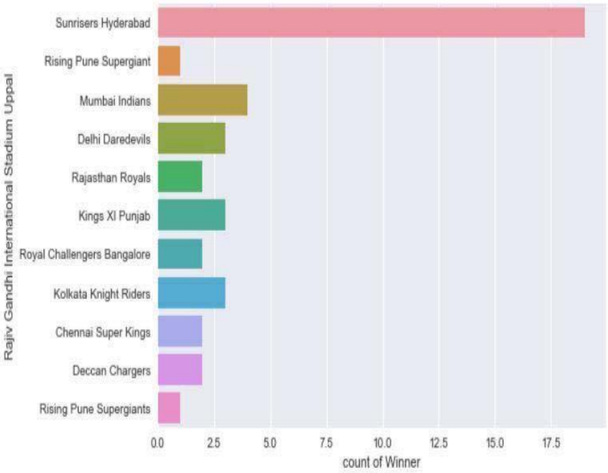

Fig 5(e) Winning Teams at Rajiv Gandhi International Stadium Uppal

Fig 5 (a)-(e) show the influence of playing on their home ground on a team's performance. It can be observed that home ground is a major influence as Mumbai Indians team have won maximum of matches played in their home ground, i.e., Wankhede Stadium. Similar results are seen for Delhi Daredevils team at Feroz Shah Kotla (Fig 5b), Chennai Super Kings at MA Chidambaram Stadium Chepauk (Fig 5c), Kolkata Knight Riders at Eden Gardens(Fig 5d) and Sunrisers Hyderabad team at Rajiv Gandhi International Stadium Uppal (Fig 5e).

The above analysis can be summarized as follows:

- When the team wins the toss they win the match 50% times
- When the toss winner takes field first then the team win 61.84% times

Authorized licensed use limited to: MEHRAN UNIV OF ENGINEERING AND TECHNOLOGY. Downloaded on December 21,2023 at 03:48:38 UTC from IEEE Xplore. Restrictions apply.

- When dl is applied the one who wins toss wins 56.25% times
- When stadium is Wankhede then 63.15% times the team that fields first wins the match
- The winner teams have mostly won the matches in their home ground Stadiums.

V. CONCLUSION

The paper has presented an analysis of IPL data set using MongoDB queries. The presented empirical study aimed to illustrate the strength of MongoDB in data modeling as well as query design. For the considered dataset, it was observed from the performed analysis that the parameters of winning a toss, playing at home grounds and application of Duckworth Lewis rule play an important role in determining the outcome of a match. The work can be further extended to more domains with larger data sets for an effective data analysis.

REFERENCES

[1] Abdullah Talha Kabakus, Resul Kara, "A performance evaluation of in-memory databases", Journal of King Saud University - Computer and Information Sciences, Volume 29, Issue 4, Pages 520-525, 2017.

[2] Abramova, V., Bernardino, J., Furtado, P. ," Which NoSQL database? A performance overview", Open Journal of Databases, Volume 1, Issue 2, pp. 17–24, 2014.

[3] Aboutorabi, S.H., Rezapour, M., Moradi, M. and Ghadiri N., "Performance evaluation of SQL and MongoDB databases for big e-commerce data", 2015.

[4] Boicea, Alexandru, Florin Radulescu, and Laura Ioana Agapin. "MongoDB vs Oracle--database comparison", 2012 third international conference on emerging intelligent data and web technologies. IEEE, 2012.

[5] Botoeva, Elena, et al., "Expressivity and Complexity of MongoDB Queries", LIPIcs-Leibniz International Proceedings in Informatics. Vol. 98. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[6] Saini, Sonia, and Shruti Kohli".Healthcare Data Analysis Using R and MongoDB", Big Data Analytics. Springer, Singapore, 2018. 709-715.

[7] Yan. H ," Application of MongoDB in processing of water resources on-line monitoring data",Guangxi Water Resources & Hydropower Engineering 3 (2017): 019.

[8] Eth Gilbert and Nancy Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services", ACM SIGACT News, Volume 33 Issue 2 (2002), pg. 51–59

[9] Liu, Yimeng, Yizhi Wang, and Yi Jin. "Research on the improvement of MongoDB Auto-Sharding in cloud environment." In Computer Science & Education (ICCSE), 2012 7th International Conference on, pp. 851-854. IEEE, 2012.

[10] Zhao, Gansen, Weichai Huang, Shunlin Liang, and Yong Tang. "Modeling MongoDB with relational model." In Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on, pp. 115-121. IEEE, 2013.

[11] Kanade, Anuradha, Arpita Gopal, and Shantanu Kanade. "A study of normalization and embedding in MongoDB." In Advance Computing Conference (IACC), 2014 IEEE International, pp. 416-421. IEEE, 2014.

[12] Abramova, Veronika, and Jorge Bernardino. "NoSQL databases: MongoDB vs cassandra." In Proceedings of the international C* conference on computer science and software engineering, pp. 14-22. ACM, 2013.

[13] Yunhua Gu, Shu Shen, Jin Wang, Jeong-Uk Kim on"Application of NoSQL Database MongoDB" 2015 International Conference on Consumer Electronics-Taiwan (ICCE-TW).

[14] Arora, Rupali, and Rinkle Rani Aggarwal. "Modeling and querying data in mongodb." International Journal of Scientific and Engineering Research 4, no. 7 (2013): 141-144.

[15] Biswajeet Sethi,Samaresh Mishra,Prasant ku. Patnaik on"A study of NoSQL database", International Conference of Engineering Research and technology(IJERT),2014 .

Authorized licensed use limited to: MEHRAN UNIV OF ENGINEERING AND TECHNOLOGY. Downloaded on December 21,2023 at 03:48:38 UTC from IEEE Xplore. Restrictions apply.