

BIG DATA PROGRAMMING

Coursework

Yuwarajalingam Dinesh

20221733

2237630

2022 September Batch

Table of Contents

[Table of figures 2](#_Toc122778582)

[1. System Architectural Design 4](#_Toc122778583)

[2. Data Analysis 7](#_Toc122778584)

[2.1 MapReduce 7](#_Toc122778585)

[2.2 Hive 11](#_Toc122778586)

[2.3 Spark 14](#_Toc122778587)

[3. Spark MLlib 16](#_Toc122778588)

[4. Presentation of the analysis 22](#_Toc122778589)

[References 23](#_Toc122778590)

# **Table of figures**

[Figure 1:Map Reduce log for first question 7](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778704)

[Figure 2:Answer Calculated for MapReduce first question 7](/Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG DATA PROGRAMMING.docx" \l "_Toc122778705)

[Figure 3:Code for MapReduce1 8](#_Toc122778706)

[Figure 4:MapReduce Log for second question 9](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778707)

[Figure 5:Answer for MapReduce second question 10](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778708)

[Figure 6:Code for MapReduce2 10](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778709)

[Figure 7:Hive Logs for question 1 11](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778710)

[Figure 8:Answer for hive question 1 11](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778711)

[Figure 9:hive query for first question 12](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778712)

[Figure 10:Hive Logs for second question 12](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778713)

[Figure 11:Answer for hive second question 13](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778714)

[Figure 12:Query for hive second question 13](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778715)

[Figure 13:Spark first question query and answer 14](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778716)

[Figure 14:Result of the transformed data 14](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778717)

[Figure 15:SQL query to transform data to calculate the result 14](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778718)

[Figure 16:Answer for the spark second question 15](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778719)

[Figure 17:Create spark MLlib session 16](#_Toc122778720)

[Figure 18:Data imported screen shot 16](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778721)

[Figure 19:Importing Data code 16](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778722)

[Figure 20:Creating temporary view ipl 17](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778723)

[Figure 21:Temporary view selected data 17](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778724)

[Figure 22:Sql query to get the power play score of each team 17](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778725)

[Figure 23:Power play each team against their opponent 18](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778726)

[Figure 24:Making data frame using relevant sql query 18](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778727)

[Figure 25:Casting powerplay score to float 18](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778728)

[Figure 26:Selecting data where when all the columns in dataset isnull 19](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778729)

[Figure 27:Mapping string values into numbers 19](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778730)

[Figure 28:transformed dataset 20](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778731)

[Figure 29:Assemble all the features with vector assembler 20](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778732)

[Figure 30:Splitting data for test and train 21](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778733)

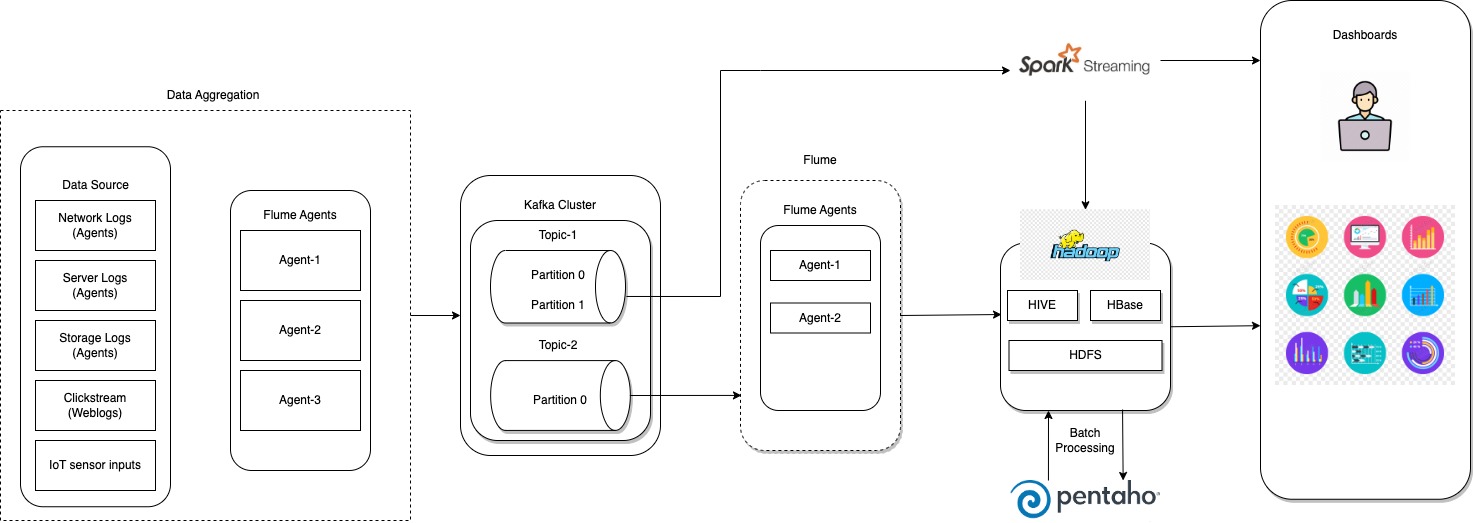
[Figure 31:Training data with Gradient Booster Tree Regression model with iteration of 70 21](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778734)

[Figure 32:Showing predictions and actual score of teams against their relevant opponent teams 21](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778735)

[Figure 33:PowerBi fourth question 22](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778736)

[Figure 34:PowerBi Data Analysis for first three questions 22](file:////Users/yuwarajalingamdinesh/Desktop/Msc_BigData/Semester1/BigDataProgramming/Assessments/BIG%20DATA%20PROGRAMMING.docx#_Toc122778737)

# **System Architectural Design**



Flume agents are used to collect the data from the Data sources and passing them to kafka cluster to store them in the que. when the flume consumers are ready data will be consumed and will be pushed to HDFS to store the data for the historical data processing. In the meantime spark streaming will also consume these data and based on the end users requirements data will be modeled. this is based on the real time analytics. for the historical data, pentaho tool is used to process the data periodically and from the Hadoop, endusers can visualize the data.

Below comes the description of the each component, its duties and the interaction with each other.

**Flume agent on servers:**

Flume is a lightweight java program which runs on JVM and collects log messages from servers. This agent is configured to read the log file and push the log messages ('event' as per Flume's language) into the Data collector i.e. Kafka broker on configured batch size or time interval.

**Data Collector/Kafka Broker:**

Apache Kafka is chosen as the data collector because of its scalability, durability and low-latency advantages. This module collects all the log messages pushed by the flume agents and keeps it safe for consumers to consume.

**Cluster and Load Balancing of Kafka broker:**

For high availability and durability of the messages, I choose to have multi node-multi broker setup. In this diagram, all the log messages are sent to a single topic T and all the flume agents will logically push them to that topic.

**Flume Consumer:**

This layer consists of group of flume consumer agents connected to Kafka broker and subscribed to topic T. These agents together, will read the data from the subscribed topic and syncs into an intermediate storage area which will be of HDFS files.

**Load balancing on consumer side:**

Each of these agents will decide among themselves who will read from which partition and do load-balancing by themselves. These agents will be configured to sync the consumed log messages into HDFS files. To achieve this, all these agents are being grouped into one consumer group, so that they can work in parallel.

**Spark Streaming:**

Spark Streaming is an extension of the core Spark API that allows data engineers and data scientists to process real-time data from various sources including Kafka, Flume, and Amazon Kinesis. This processed data can be pushed out to file systems, databases, and live dashboards.

**Pentaho:**

Pentaho is an extensively used Business Intelligence tool set (suite) across industries for data management. Analysts, data managers, software developers, and even students find the applicability of this tool. Pentaho suite enhances the overall performance of the business by generating informative reports in varied formats like text, XML, HTML, CSV, Excel, PDF, etc.

# **2. Data Analysis**

## **2.1 MapReduce**

**The code is given in the attached folder itself. Please refer for map reduce part.**



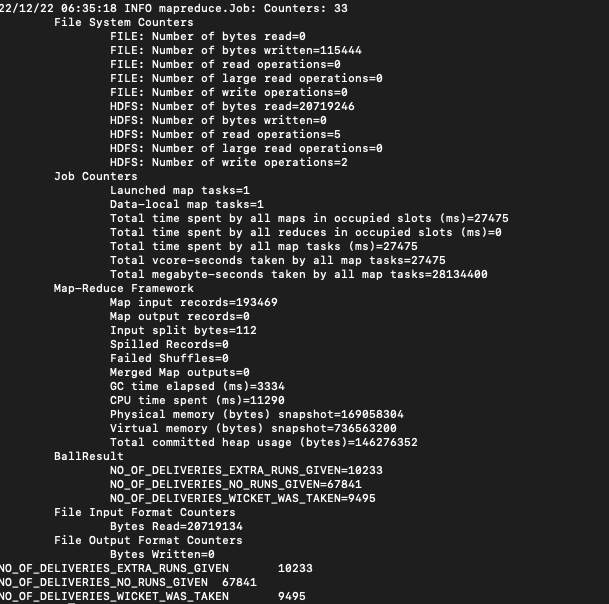


Figure 1:Map Reduce log for first question

Text

Description automatically generated

Figure 2:Answer Calculated for MapReduce first question

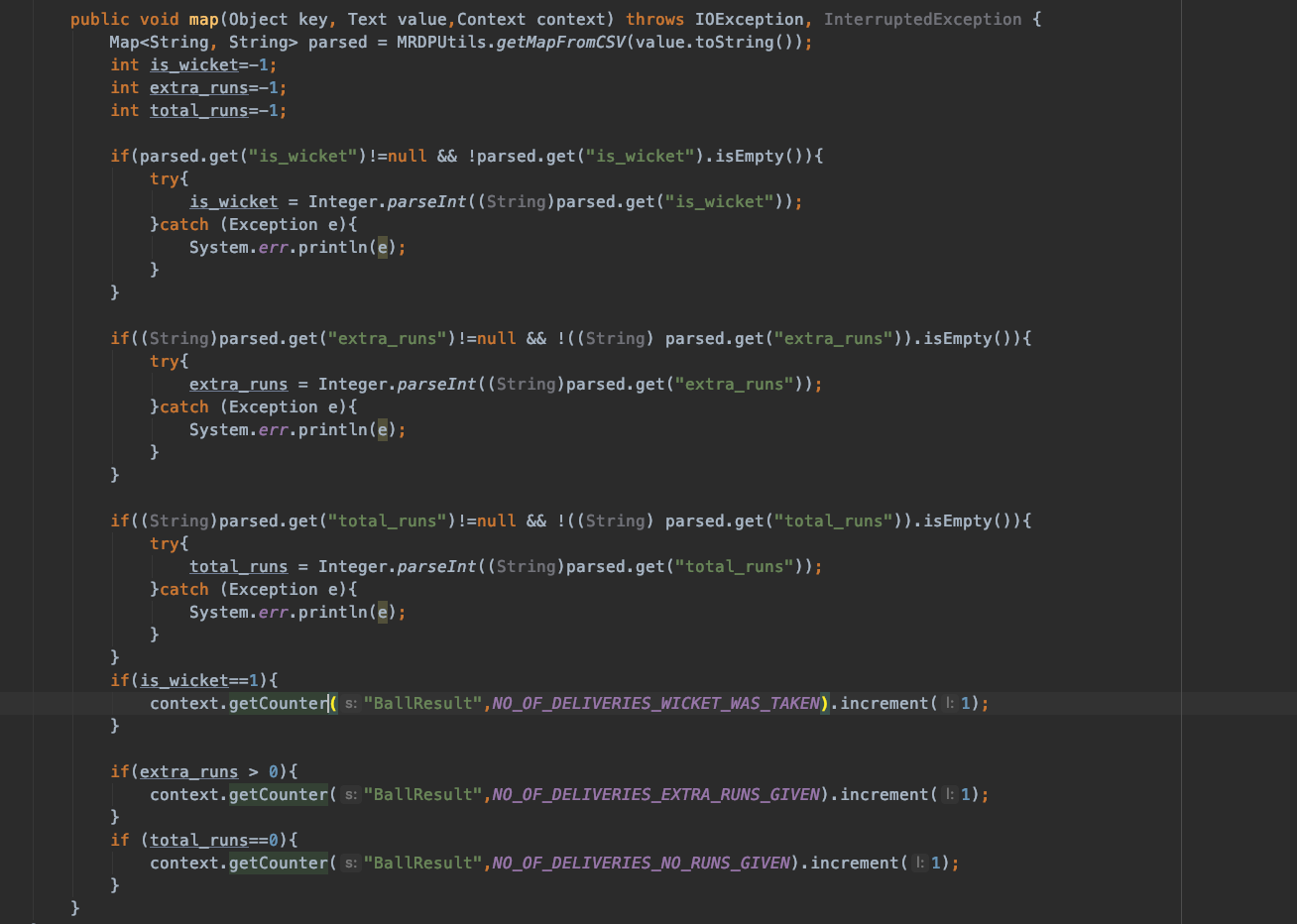


Figure 3:Code for MapReduce1



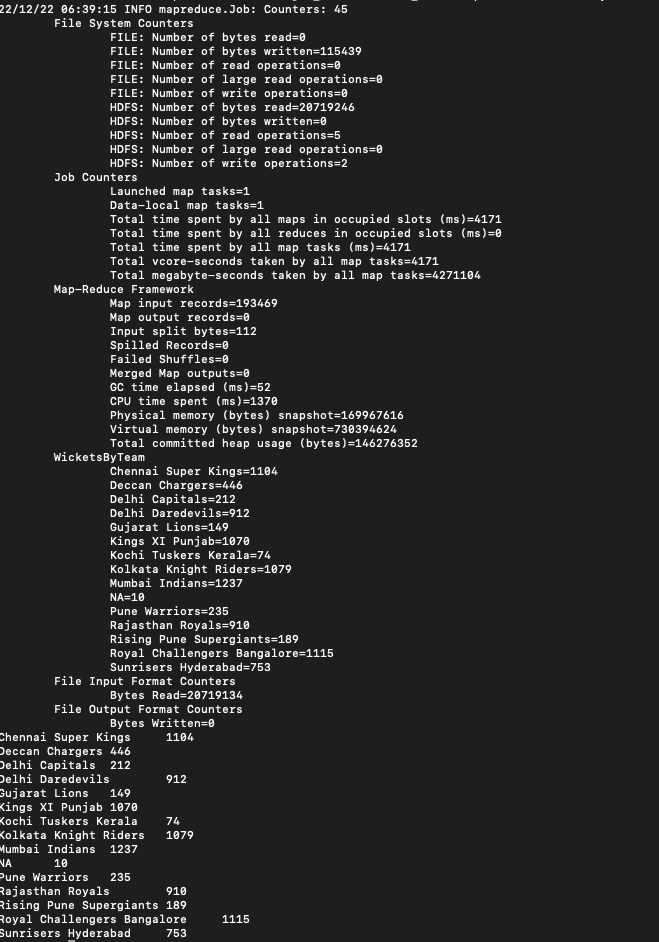


Figure 4:MapReduce Log for second question



Figure 5:Answer for MapReduce second question

Text

Description automatically generated

Figure 6:Code for MapReduce2

## **2.2 Hive**





Figure 7:Hive Logs for question 1

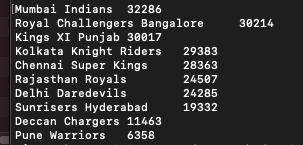


Figure 8:Answer for hive question 1

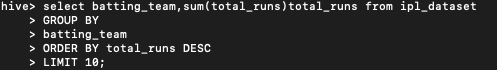


Figure 9:hive query for first question

1. 

Figure 10:Hive Logs for second question

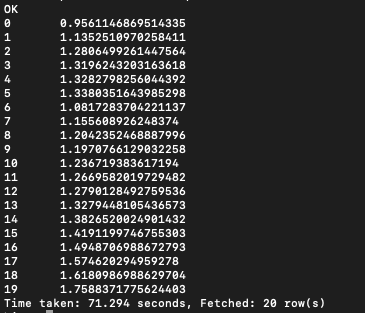


Figure 11:Answer for hive second question



Figure 12:Query for hive second question

## **2.3 Spark**

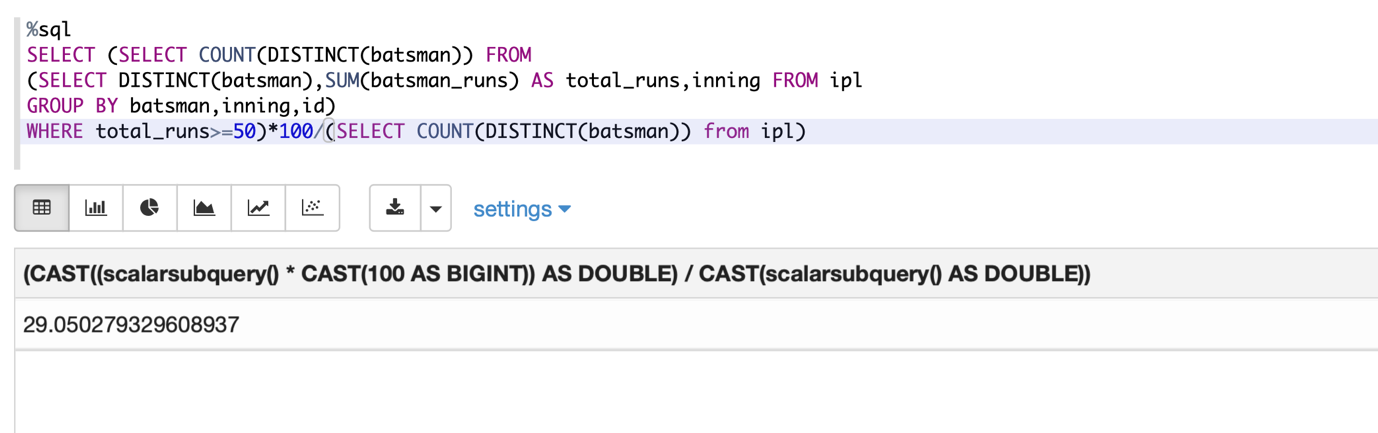
1. 

Figure 13:Spark first question query and answer



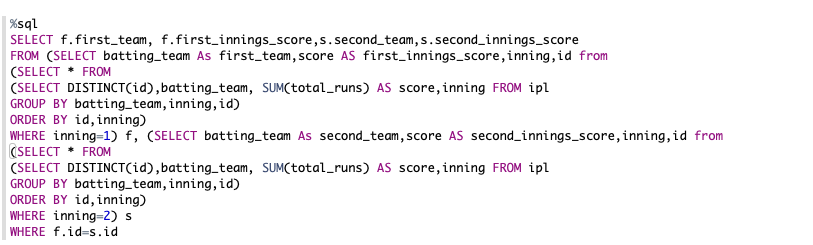
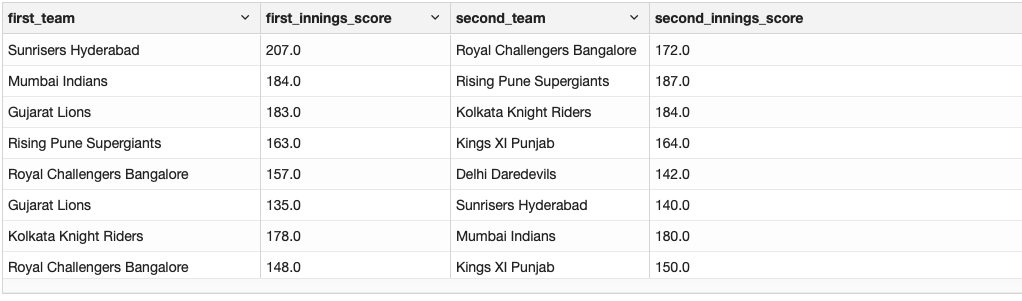


Figure 14:Result of the transformed data

Figure 15:SQL query to transform data to calculate the result

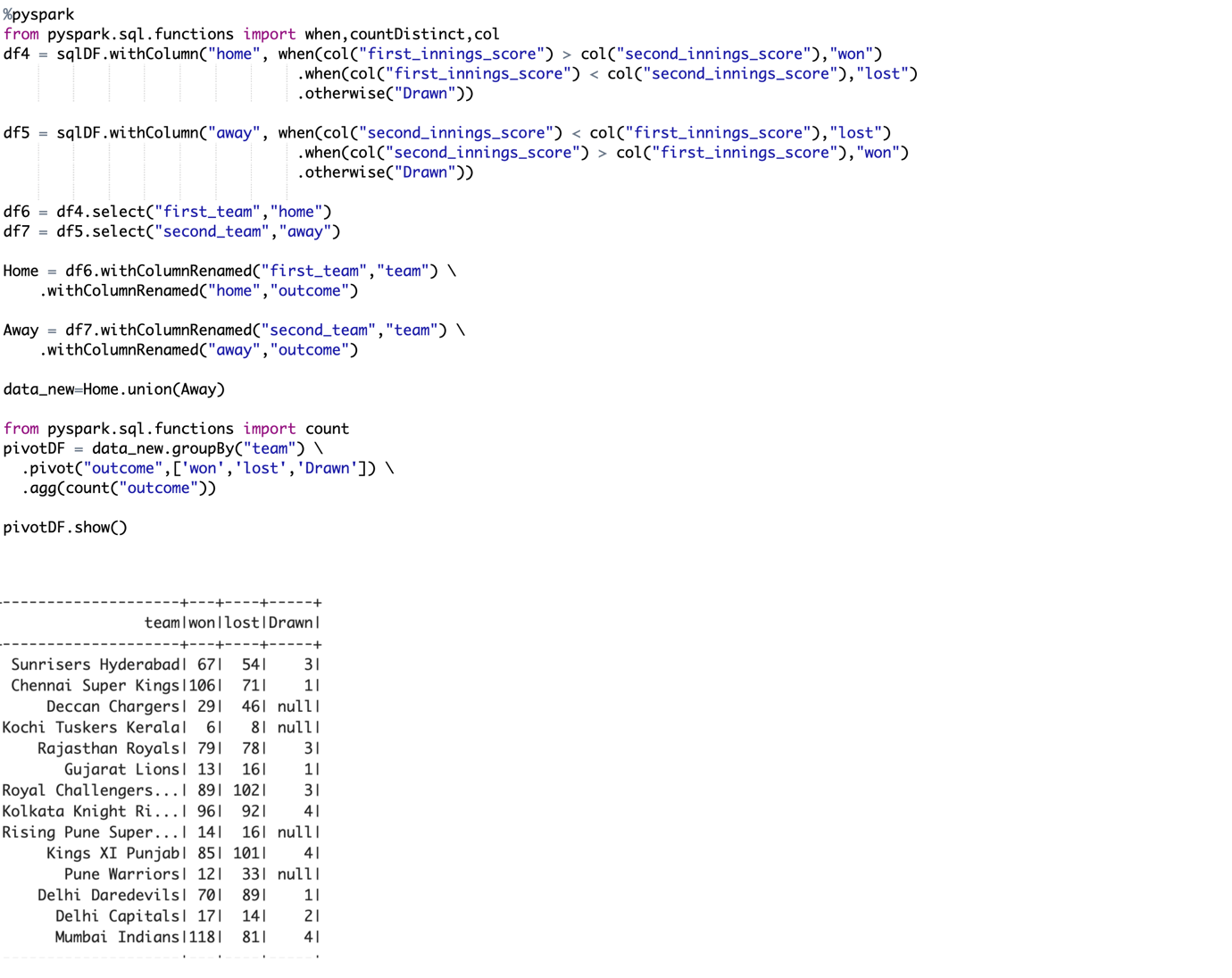


Figure 16:Answer for the spark second question

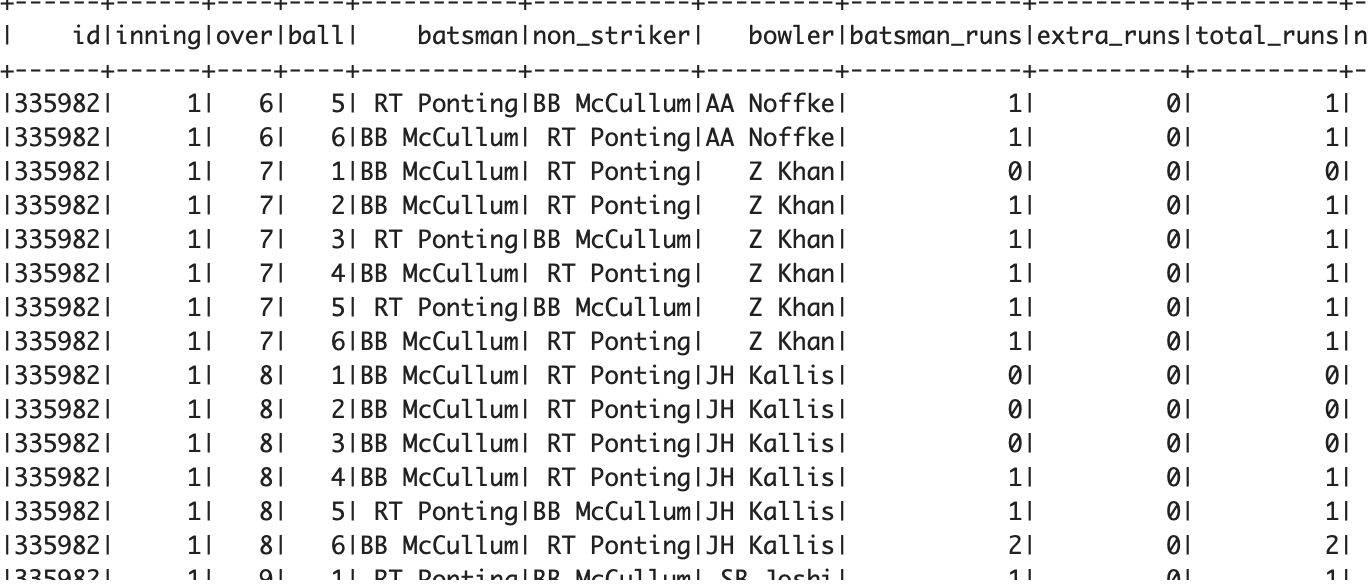
# **3. Spark MLlib**

**Import Spark MLlib session from spark**



Figure 17:Create spark MLlib session

**Importing data and show to confirm data has imported**

A picture containing graphical user interface

Description automatically generated

Figure 18:Data imported screen shot

Figure 19:Importing Data code

**Create temporary view to execute sql queries**

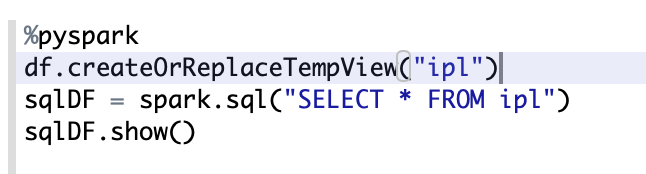


Figure 20:Creating temporary view ipl

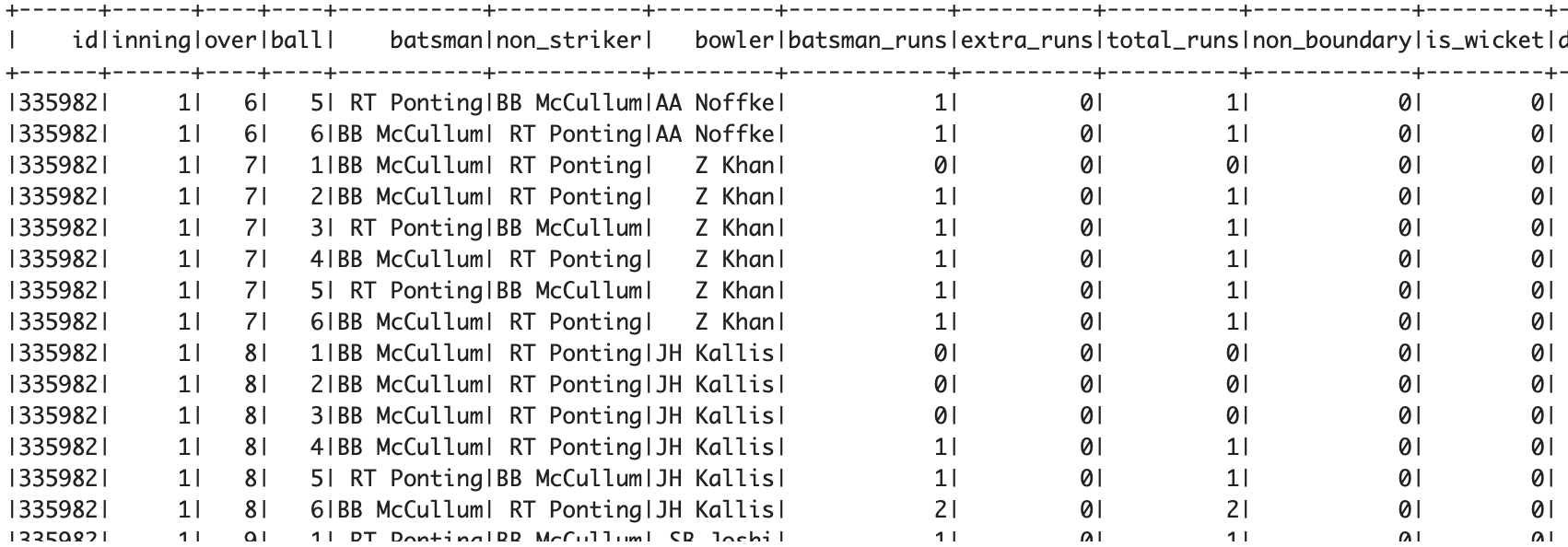


Figure 21:Temporary view selected data

**Sql query to get the powerplay runs of each team against their relevant opponent team**

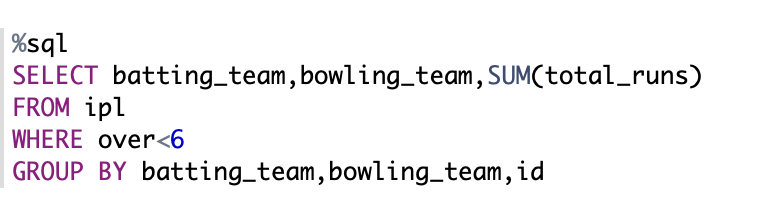


Figure 22:Sql query to get the power play score of each team

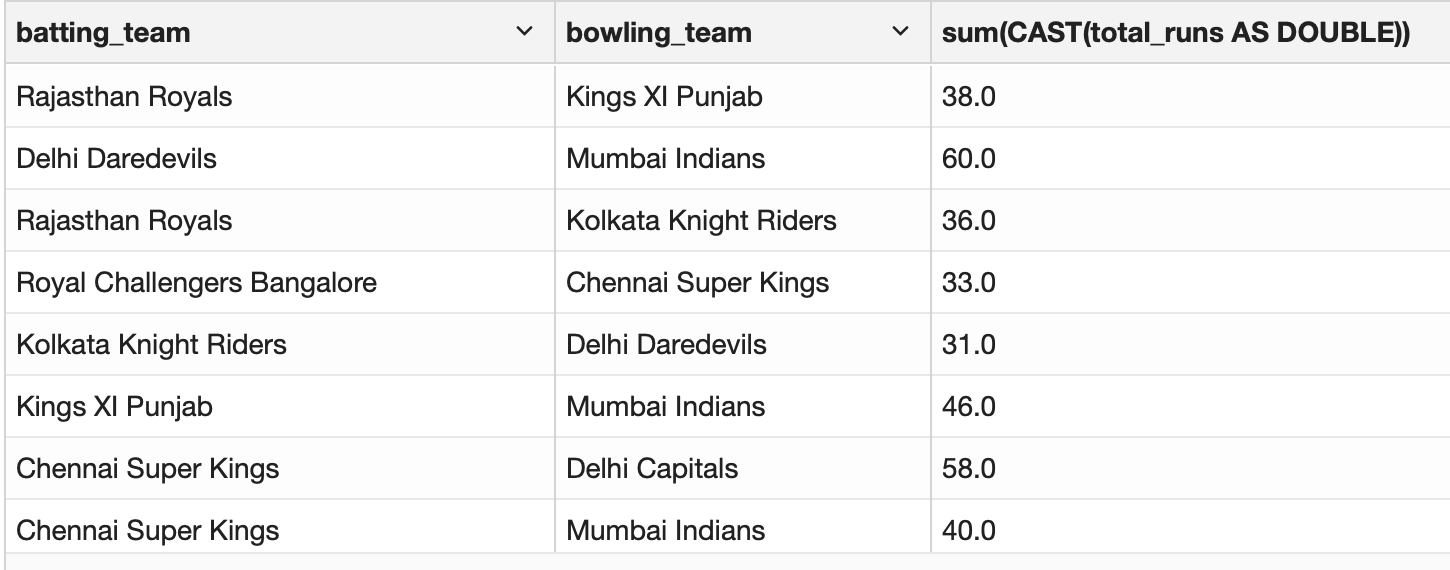


Figure 23:Power play each team against their opponent

**Making sql data frame using the above sql query**

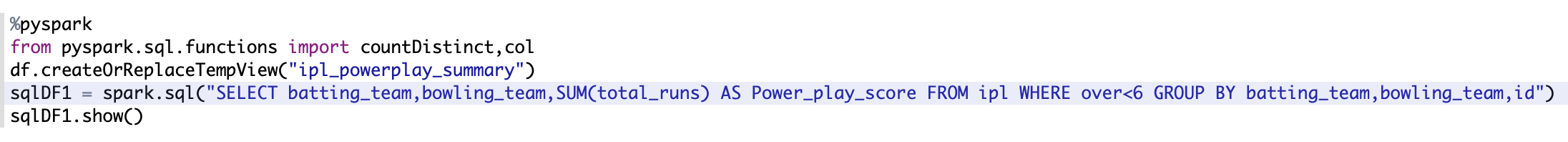
****

Figure 24:Making data frame using relevant sql query

**Casting power play score to float**

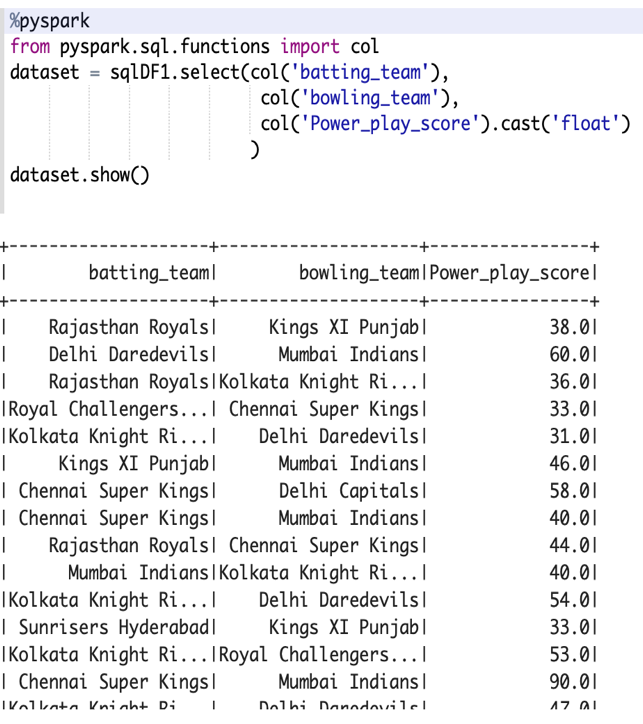
****

Figure 25:Casting powerplay score to float

**Selecting data where when all the columns in dataset isnull**

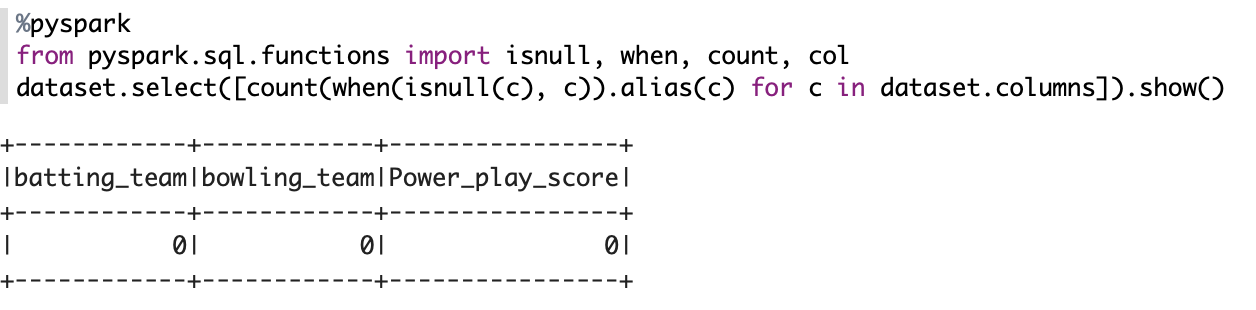
****

Figure 26:Selecting data where when all the columns in dataset isnull

**Mapping batting\_team and bowling\_team to numbers where machine learning model can understand as input**

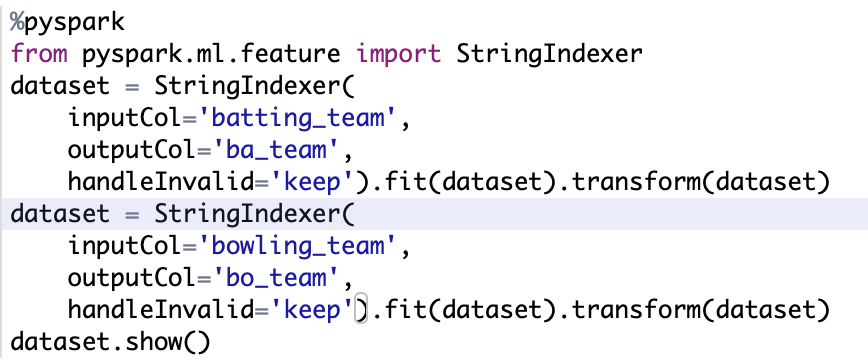


Figure 27:Mapping string values into numbers

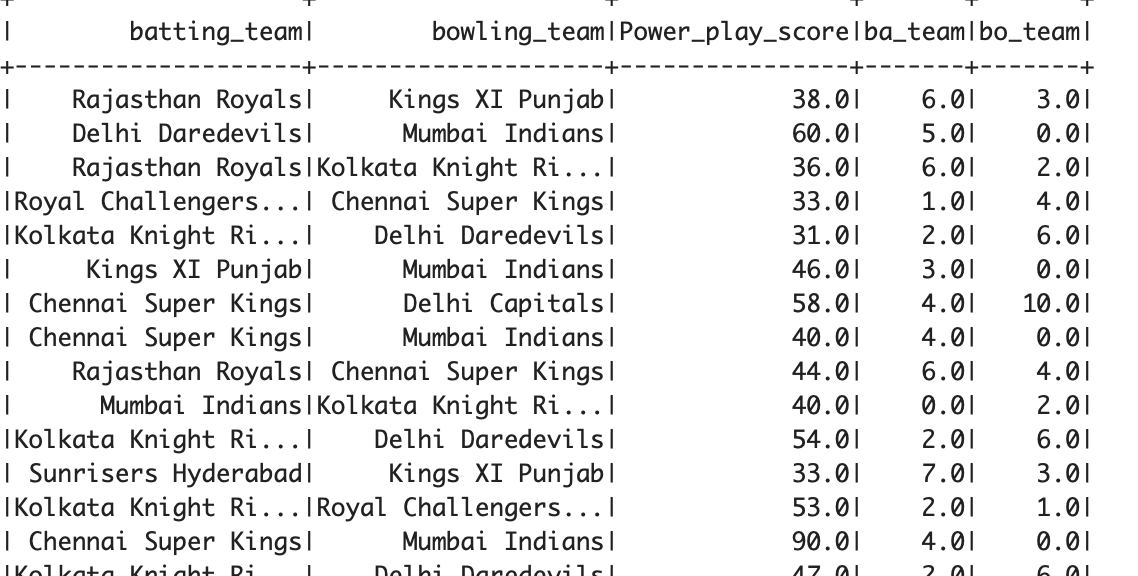


Figure 28:transformed dataset

**Assemble all the features with vector assembler**

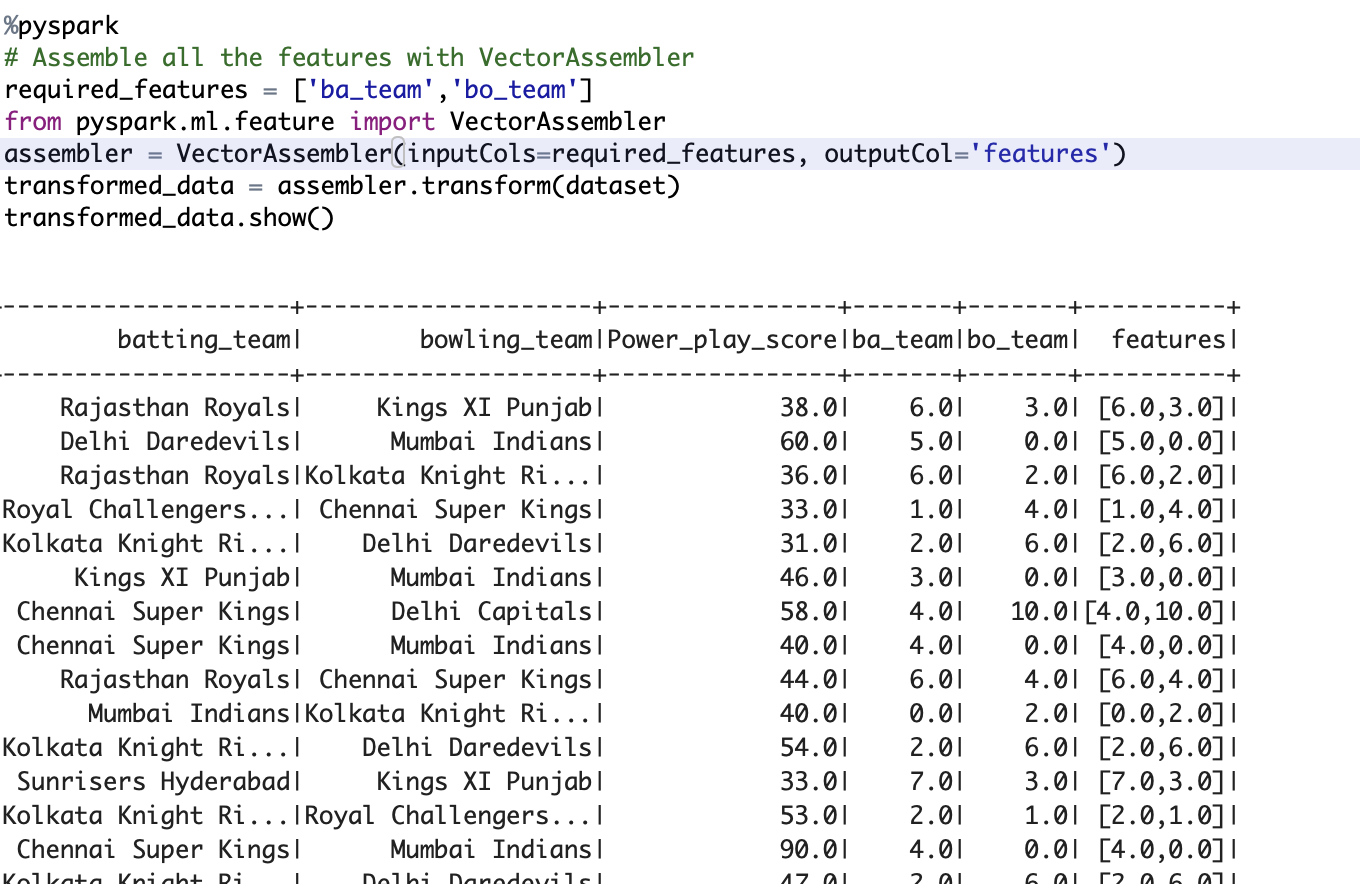
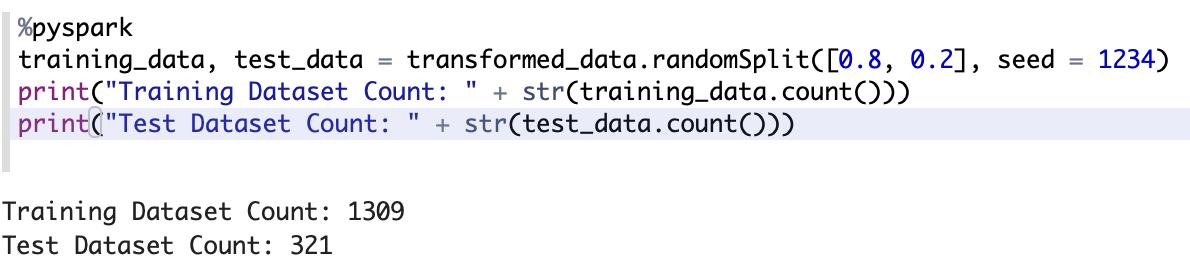


Figure 29:Assemble all the features with vector assembler

**Splitting training data and test data**

Figure 30:Splitting data for test and train



**Training data with Gradient Booster Tree Regression model with iteration of 70**

**Word

Description automatically generated**

Figure 31:Training data with Gradient Booster Tree Regression model with iteration of 70

**Showing predictions and actual score of teams against their relevant opponent teams**

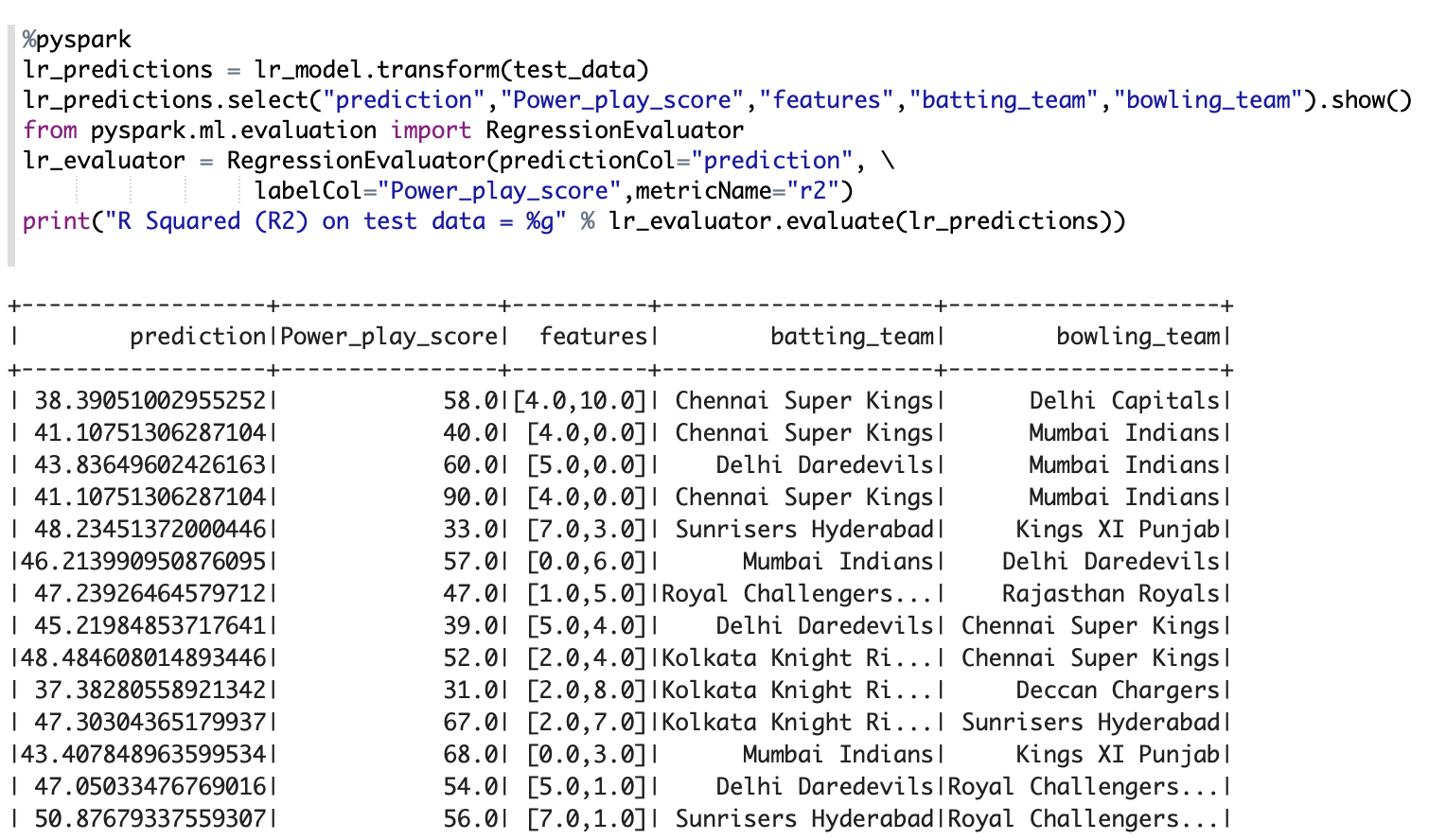
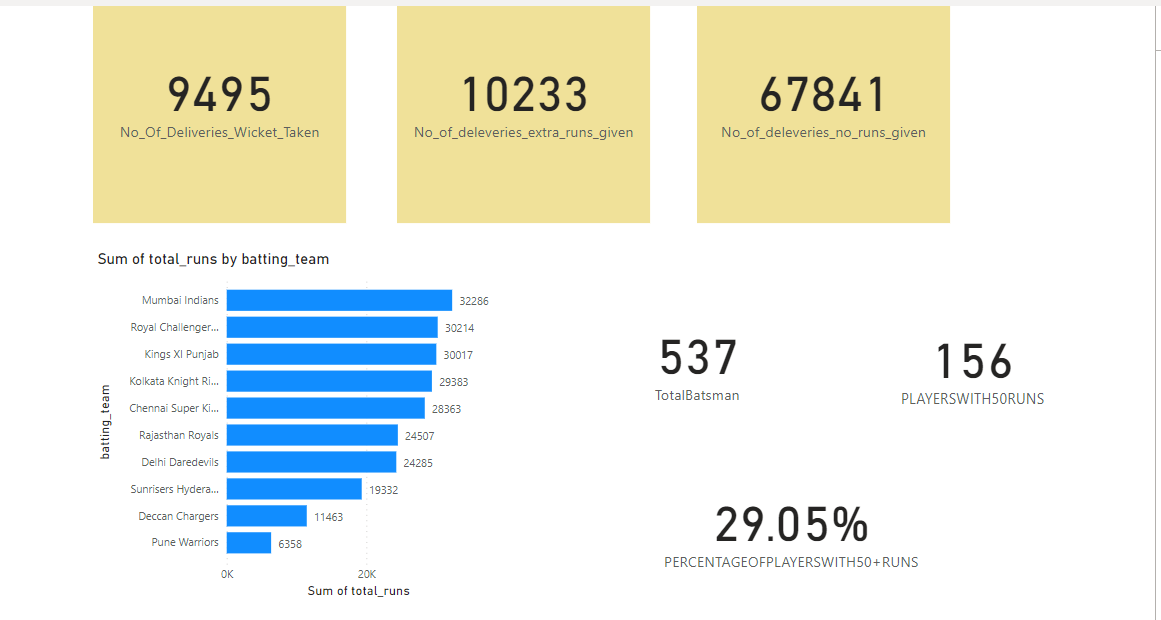


Figure 32:Showing predictions and actual score of teams against their relevant opponent teams

# **4. Presentation of the analysis**



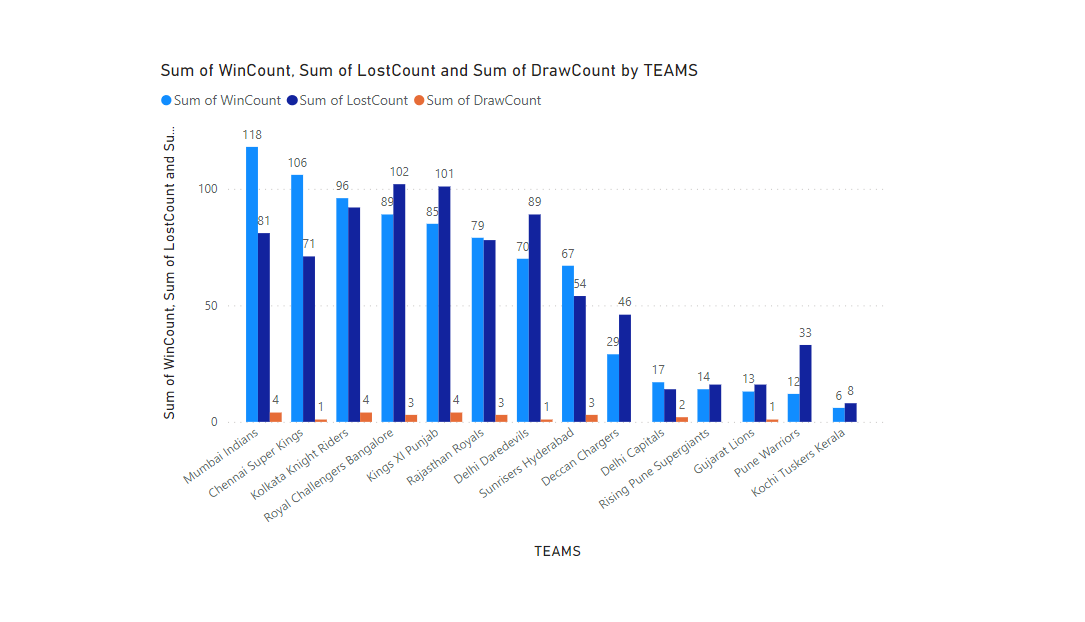


Figure 33:PowerBi fourth question

Figure 34:PowerBi Data Analysis for first three questions

# **References**

hokkaido university, (2019), containerbased-sizing-framework-for-apache-hadoopspark-clusters [ONLINE]. Available at: https://image.slidesharecdn.com/1110akiyoshisugikiv1-161031185742/95/a-containerbased-sizing-framework-for-apache-hadoopspark-clusters-10-638.jpg?cb=1477940273 [Accessed 10 January 2019].