Team -06: DATA ENCODERS

Version-01

Dataset from the Website

https://m.cricbuzz.com/

Excel Sheet

(For importing the data)

Unstructured

daa

Python for further conversions in the dataset to visualize

Power BI for visualization

Structured data

SQL or Excel (yet to be decided)

**Work Division:**

1. Implementation of Python : Raghunandan
2. Data prep : Deepthi and Nikitha Kethireddy
3. Visualization : Jeevan , Dheeraj

Version-02

Using a package called (pycricbuzz) for generating the live data

A new file with data is generated in the form of JSON file.

We are using python to append the data to the generated file every time the data is updated in the website

Whenever we update the file in power BI, the graphs/bars etc will be updated dynamically.

We use that generated file in Power BI and we have an option to update that file by setting time

**Work Division:**

1. Implementation of Python : Raghunandan
2. Data prep : Deepthi and Nikitha Kethireddy
3. Visualization : Jeevan , Dheeraj

Version-03

Using a package called (pycricbuzz) for generating the live data

We are using python to get the data onto the local system every time the data is updated in the website.

Data is appended to MongoDB as we get live score from the package.

Reports in Power BI will be refreshed as we update the data in MongoDB

We connect to MongoDB to fetch the data and will use Power BI to generate the graphs

**Work Division:**

1. Implementation of Python: Raghunandan
2. Data prep: Deepthi and Nikitha Kethireddy
3. Visualization: Jeevan
4. Commentary Sentiment Analysis: Dheeraj

Version-04

Using a package called (pycricbuzz) for generating the live data

We are using python to get the data onto the local system every time the data is updated in the website.

Data will be in format of JSON and this will be sent to KAFKA

JSON

Fetch from KAFKA

Reports in Power BI will be refreshed as we update the data in KAFKA

We connect to KAFKA to Power BI

Reporting

**Work Division:**

1. Implementation of Python: Raghunandan
2. Data prep: Deepthi and Nikitha Kethireddy
3. Visualization: Jeevan
4. Commentary Sentiment Analysis: Dheeraj

**Version-05**

**Score Estimation:**

Fetch the data from API in JSON Format

JSON Data

Convert JSON data to Bytes using JSONSerializer

Data in Byte format

Stream the bytes data into Kafka

Data in Byte format

Convert Byte format data into JSON using JSONDeserializer

JSON Data

Consuming JSON Data into MongoDB

Fetch Data from MongoDB

Data Transformation

Visualize the data using PowerBI

**Data Source:** We are fetching the data from the API provided by [www.crickbuzz.com](http://www.crickbuzz.com) to perform data visualization.

**Steps for Data Extraction:**

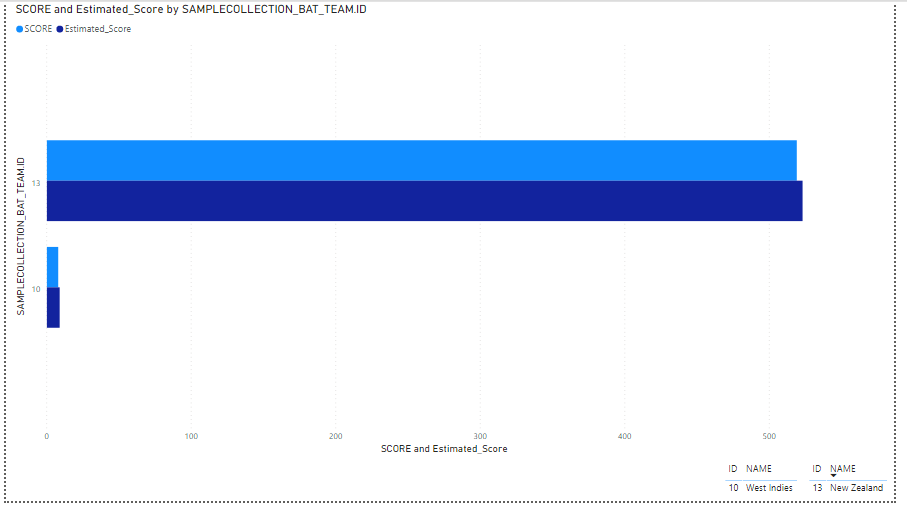
* Run zookeeper and kafka by using the following commands:
  + .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
  + .\bin\windows\kafka-server-start.bat .\config\server.properties
* Run local MongoDB server and connect it PowerBI using JDBC connections.
* Build the project using *mvn clean compile assembly:single* command
* Run the producer and consumer by using any Java IDE using the following commands:
  + java -cp target/KafkaLiveScoreStream-1.0-SNAPSHOT-jar-with-dependencies.jar edu/nwmissouri/KafkaLiveScoreStream/LiveScoreKafkaProducer
  + java -cp target/KafkaLiveScoreStream-1.0-SNAPSHOT-jar-with-dependencies.jar edu/nwmissouri/KafkaLiveScoreStream/LiveScoreKafkaConsumer
* Here the producer gets the data from the API of the data source as mentioned above by using Java.
* The reason for using java is because of Serialization and Deserialization of the JSON data is easy.
* Consumer consumes the data by using Java and stores the end-result using MongoDB.
* Reason for using MongoDB is that, it’s used to store the streamed data and also it acts as a middleware between Kafka and PowerBI.

**Data Analyzing and transformation:**

* Establishment connection from MongoDB to Power BI
* Click on transform data option. A new tab will be opened wherein we will perform the following operations:
  + We need to perform group-by operation upon the unique id which is generated for every individual teams.
  + We should be finding most recent overs and recent score by calculating the maximum of overs and maximum score so as to obtain the most recent overs and runs scored by each team at that point of time.
  + Using score and total number of overs played, we will be run rate of the team. This is done by dividing score with overs. The value thus obtained will be rounded-up.
* By adding the run rate to total score we can obtain the estimated score.
* The estimated score thus obtained will be used to visualize the data accordingly.

**Data Visualizing:**

We will be visualizing the data using Power BI.



The image shown above is the final visualized output for the streaming data for a cricket match.\

**Work Division:**

* Implementation of Kafka and Data Extraction: Raghunandan Kumar Naishadam
* Connecting MongoDB to PowerBI : Nikitha Kethireddy
* Data Transformations: Deepthi Chokka
* Data Visualization: Jeevan

**Commentary Sentiment Analysis:**

Fetch Commentary in JSON format using Pycricbuzz package.

Sentiment Analysis with NLTK Package and generate score

Generate emoji using Sentiment score

**Work Division:**

* Data Extraction: Raghunandan Kumar Naishadam
* Analysis of Commentary: Dheeraj

**Data Source:** We are fetching the data from the API provided by [www.crickbuzz.com](http://www.crickbuzz.com) to perform data visualization.

**Data Extraction:**

Commentary data will be extracted using a python package called “pycricbuzz”

**Data Analyzing:**

* Data Analysis is done by using a python package called as NLTK which results in score.
* Based on the score emojis will be generated.

**Data Visualization:**

The data thus obtained will be visualized by using python programming.