A Data Engineering Lab Project Report on

IPL Data Analysis

Submitted by-

Chitresh Kulhade

Group ‘1’

Team members-

Ajay Kumar Pal, Chitresh Kulhade, Navya N

Under the guidance of

Dr. Mrinal Kanti Das

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them. We are highly indebted to Dr. Mrinal Kanti Das for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our special gratitude and thanks to Teaching Assistance of Data Science Department for giving us such attention and time.

Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

Table of Content

ABSTRACT

The Indian Premier League is a professional Twenty20 cricket league, contested by ten teams based out of ten Indian cities. The league was founded by the Board of Control for Cricket in India in 2007. It is usually held between March and May of every year and has an exclusive window in the ICC Future Tours Programme. There have been fourteen seasons of the IPL tournament. The current IPL title holders are the Chennai Super Kings, winning the 2021 season. The venue for the 2020 season was moved due to the COVID-19 pandemic and games were played in the United Arab Emirates.

Our project consists of two different datasets, first one contains details of all matches that have been played till 2019 and second dataset contains ball by ball details of each delivery during each match.

We have extracted information from the datasets based on **Extract-Transform-Load** model and summarised the insightful information gained after performing data analysis. In this way raw data from different sources has been transformed into an organised set of information.

We have created a web application based dashboard for representation of all required insights of IPL in the form of informative graphs and charts along with user based login for personnel experience.

Individual - Contribution

I have been associated with tasks such as :-

1. Connecting to maria-db database management system through jupyter notebook using python. This task is achieved by using sql-alchemy in which I had created engines which can connect to database for respective datasets. Ball by Ball dataset exceeded the limit of database capacity therefore changed default database capacity.
2. Pre-process the dataset for analytics. This stage includes cleaning the datasets, along with removing unnecessary features and imputing missing values within datasets. I have used pandas python library for achieving above tasks along with creating relevant data-frames for each plot required.
3. I plotted representation of data associated with question 4 to 7 given in the requirement analysis. For achieving above task I have used seaborn python library for plotting different pie-charts and bar-graphs.
    - Most win in IPL matches
    - Whether winning toss results into winning match
    - Top 10 Batsmen of 2019 IPL season
    - Top 10 Bowlers of 2019 IPL season
    - Best bowlers against any 10 batsmen

    Plots regarding above mentioned points can be seen in pre-processing section.

Requirement Analysis

Website must have these functionalities

1. User registration
2. Login/Logout Functionality
3. After successful login, dashboard should be visible and an option for logout.
4. The following only plots should be visible in the dashboard

A dropdown to select the team stats /batsman's stats/bowler's stats.

Team stats:

1. A dropdown to select the team and for each team,
   An animated labelled bar plot with labels on each bar with y axis as stadium names and x axis as number of wins for top 5 stadiums where the team won maximum number of wins.
2. A pie-chart showing the percentages of either winning toss increases choice of victory or losing the toss.
3. A bar chart representing the most wins in all ipl matches for each team with count of wins in y axis.
4. A bar plot to show most wins in eliminator with x-axis as team names and y-axis as wins count.

Batsman's stats:

1. A bar plot with x-axis as player names and y-axis represents the score of each player in some match .
2. A dropdown to select the season .
   A bar plot with x-axis as top 10 batsman names and y-axis are the runs scored by the batsman in selected year.

Bowler's stats:

1. A bar plot with x-axis as bowler names and y-axis are the wickets taken by the bowlers in given match.
2. A dropdown to select the season
   A bar plot with x-axis as bowler names and y-axis are the wickets taken by the bowlers in selected year.

System Analysis

MariaDB MySQL Database :-

MariaDB is a popular fork of MySQL created by MySQL's original developers. It offers support for both small data processing tasks and enterprise needs. MariaDB includes a wide selection of storage engines, including high-performance storage engines, for working with other RDBMS data sources. It runs on a number of operating systems and supports a wide variety of programming languages.

SQL Alchemy :-

SQL-Alchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL. It provides a full suite of well known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language.

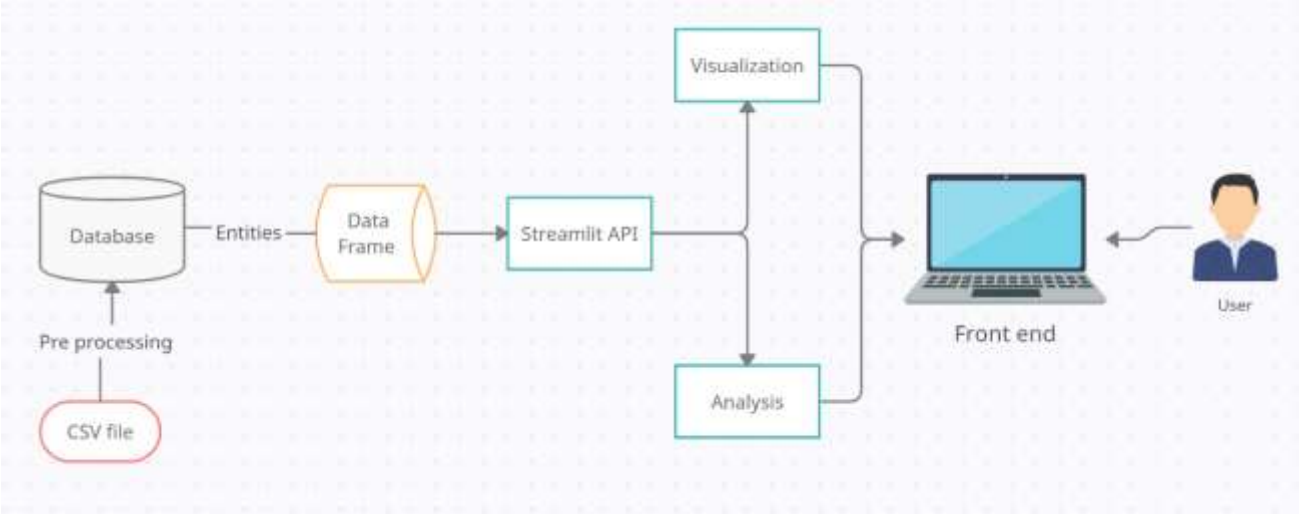
Python Libraries :-

1. **Seaborn** :- Seaborn is a library mostly used for statistical plotting in Python. It is built on top of Matplotlib and provides beautiful default styles and colour palettes to make statistical plots more attractive
2. **Pandas** :- Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data

Stream-lit :-

It is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. It is compatible with major python libraries such as scikit-learn, keras, py-torch, latex, numpy, pandas, matplotlib, seaborn etc.

System Design



Entity Relationship Diagram

Stages of Pre-Processing

DBMS :-

Instead of using CSV files directly, we have stored data in mariadb database for effective retrieval of datasets namely matches and ball by ball for further processing. SQL alchemy has been used to establish a connection between database management system and jupyter notebook.

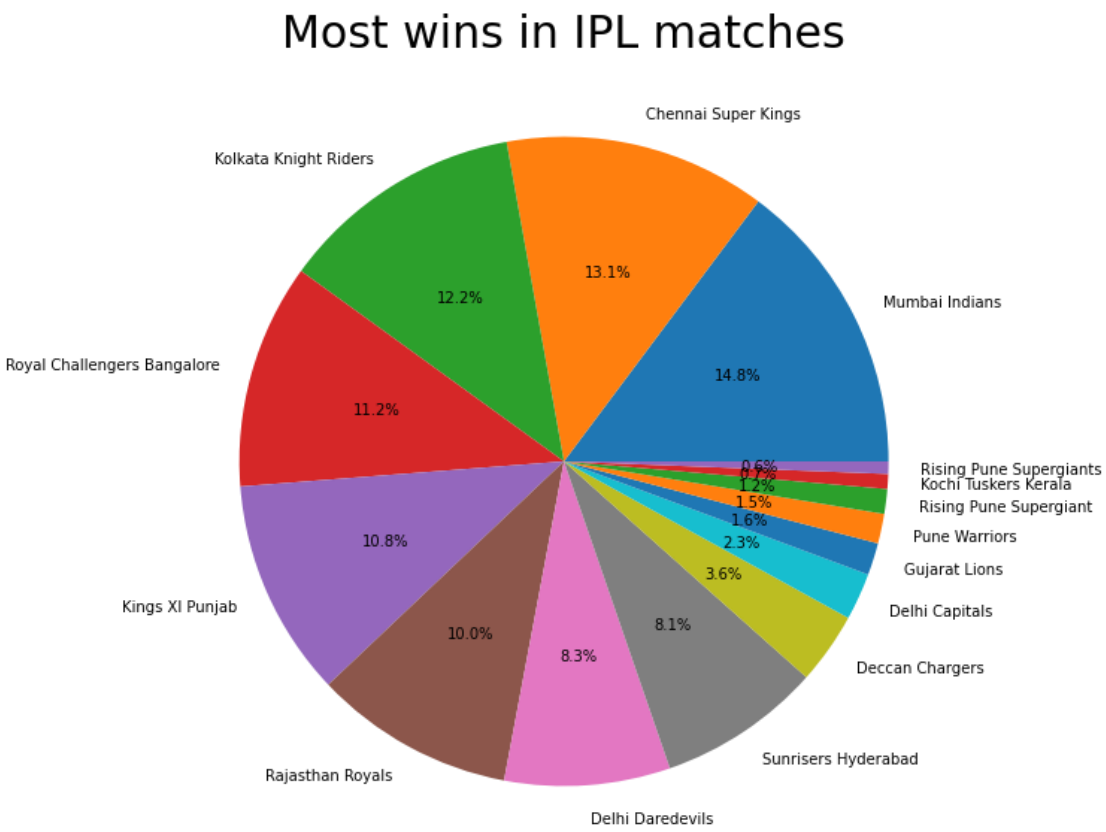
Data Cleaning :-

Pandas has been excessively used to visualize tables from database as dataframes. Some columns with excessive null values like ‘method’ had been removed from the dataframe, also imputed missing values of some important columns with suitable values. Modified dataframe had been used for further analysis.

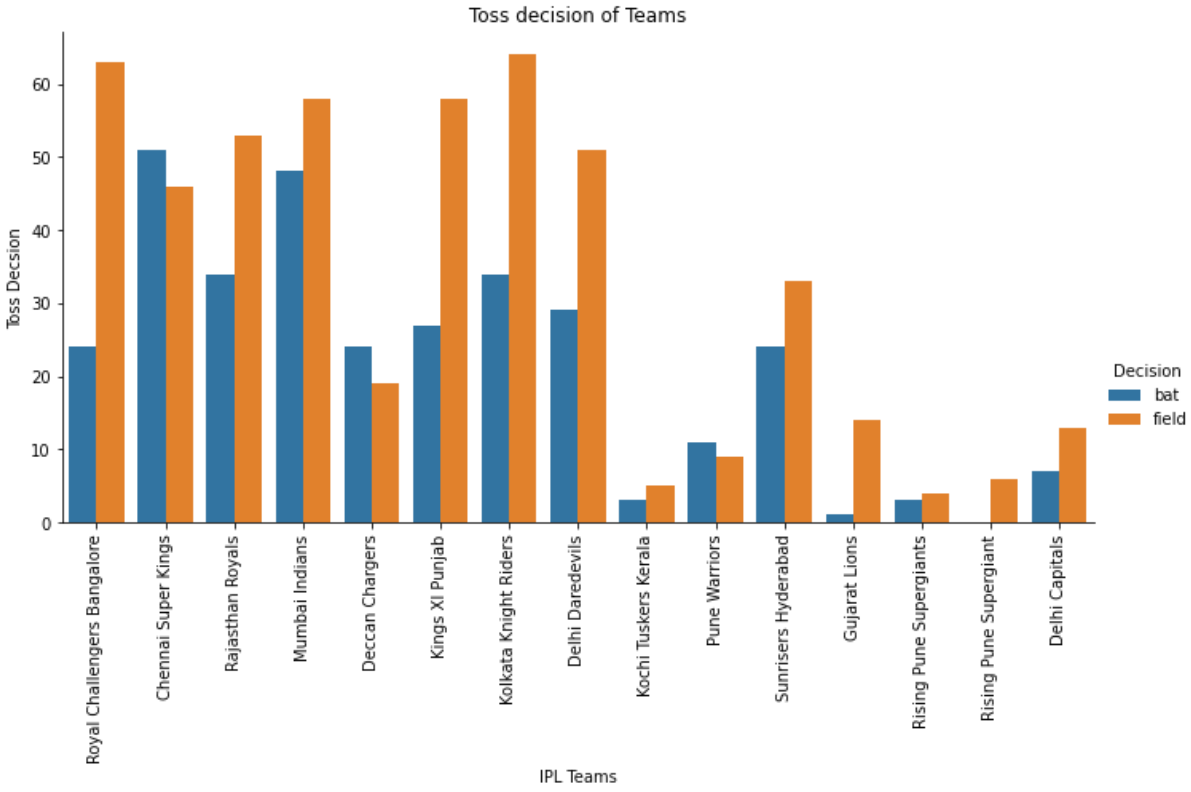
Data Visualization :-

It includes representation of informative data obtained after manipulating datasets according to need of project. We have used seaborn library of python for plotting various visual represention like, bar graph, pie chart etc.

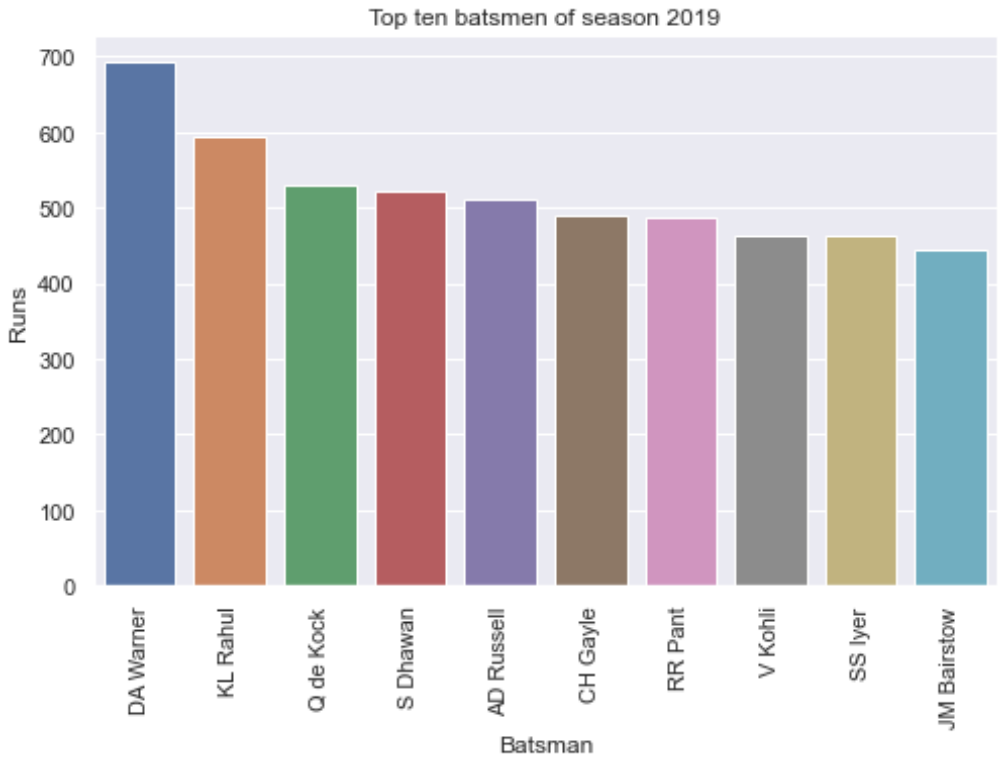
1. Most win in IPL matches



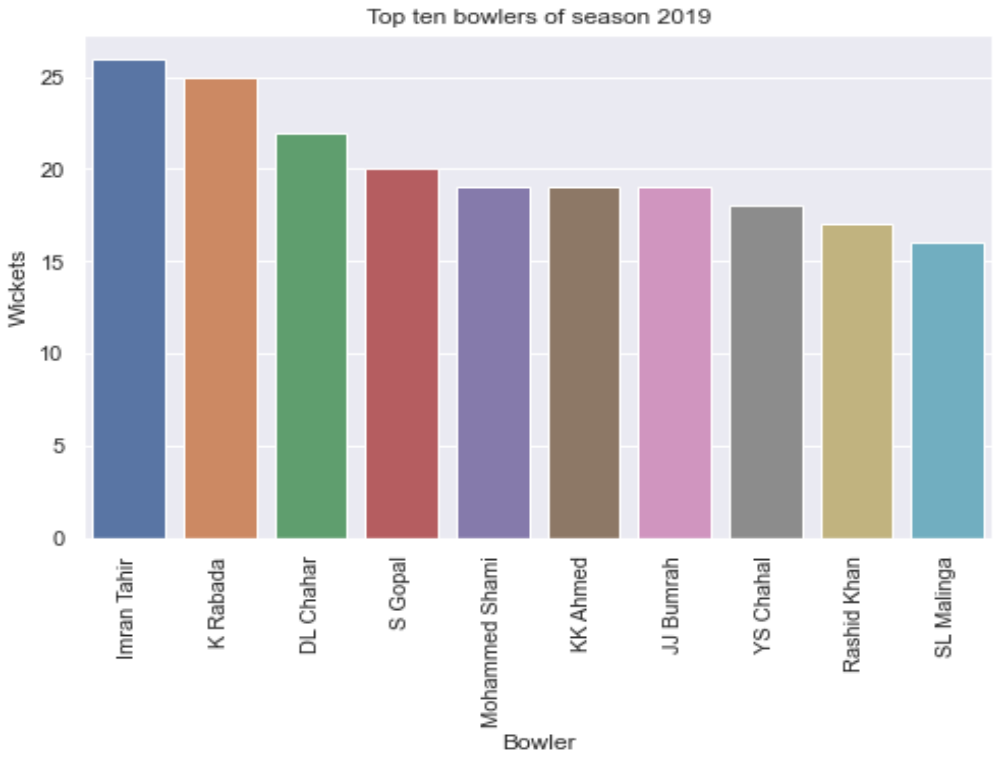
2. Toss decisions vs wins by opting bat/field



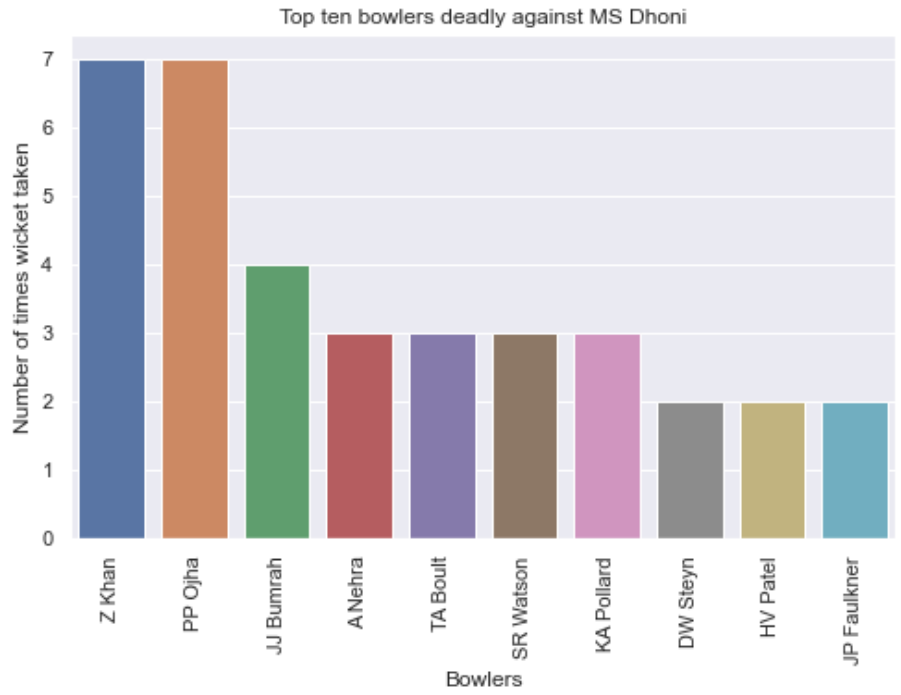
3. Top 10 batsman of 2019 IPL season



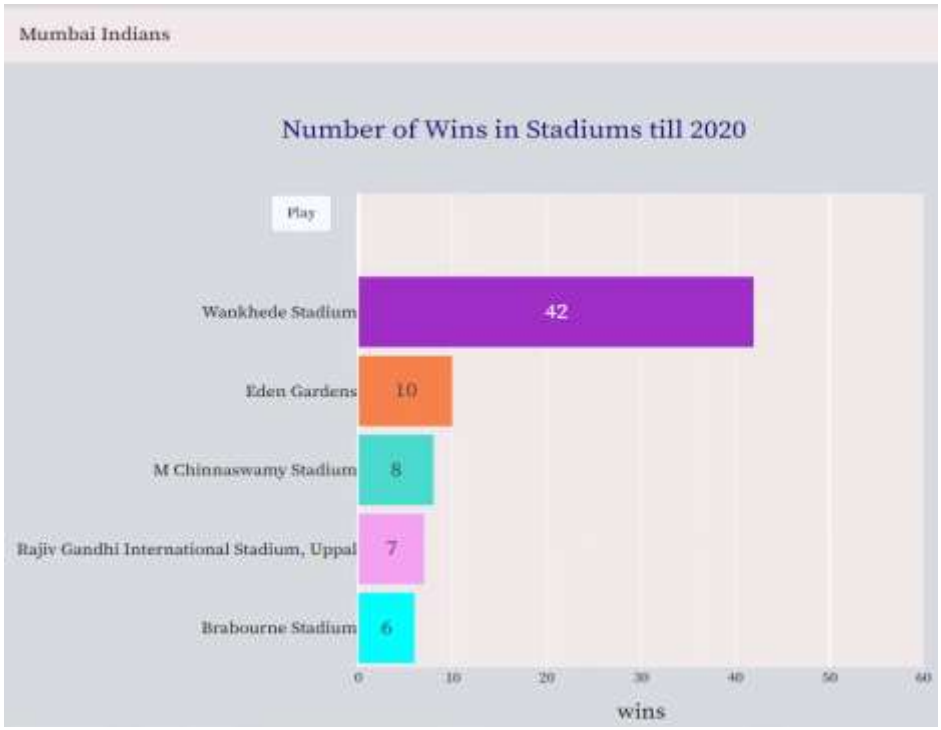
4. Top 10 bowlers of 2019 IPL season



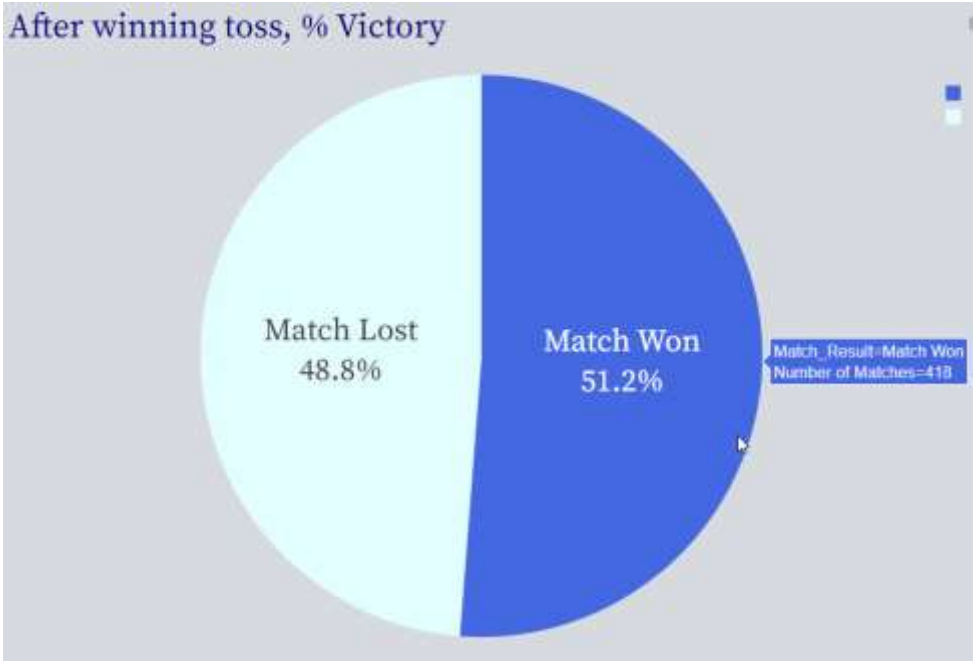
5. Best bowlers against Mahindra Singh Dhoni



6. Top five stadiums of a team in a particular year

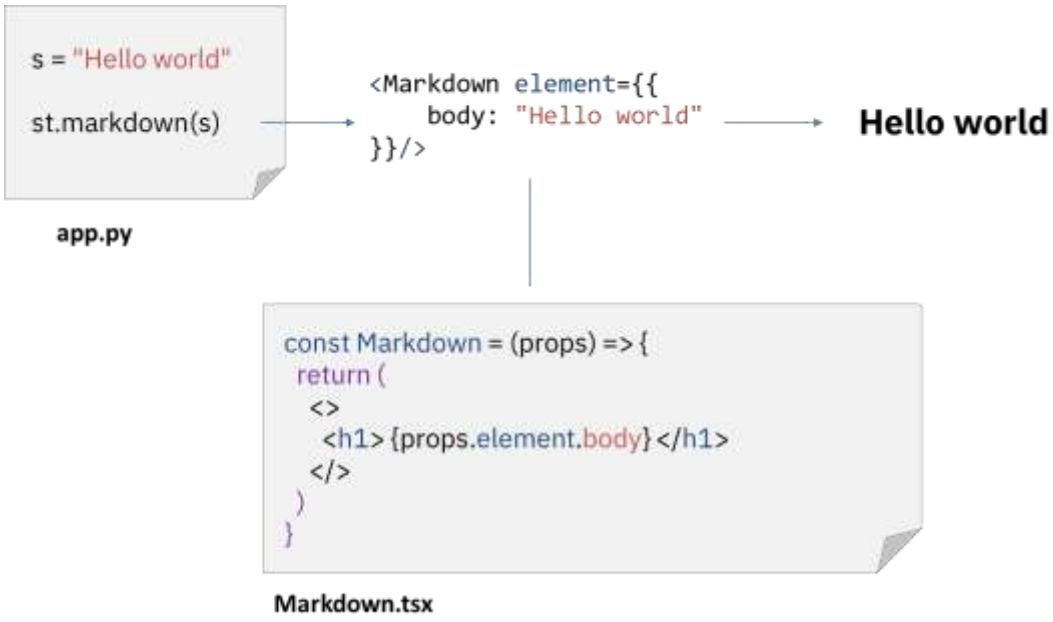


7. Percentage of victory according to toss decision



Dashboard

For Dashboard creation we have exclusively used Stream-lit API. Each Stream-lit call on the Python side loads up a React component from the running Stream-lit server, which is then rendered onto default web browser. If we go under the hood of a ‘st.markdown’ call:



The Stream-lit call ‘st.markdown’ is mapped to a ‘Markdown’ React component. This ‘Markdown’ component is a Java-script class or function that is rendered as a bunch of HTML/CSS/JS code. The component is then implemented inside a ‘Markdown.tsx’ file and accessed from a running browser via a path to the filesystem. For this example, it defines a functional component which returns a single ‘<h1>’ block for React to render in the browser.

Conclusion And Future Work

We have successfully fulfilled all the requirements of the project. We have used SQL alchemy for establishing a connection between system and database. All plots showing visual representation of data had been achieved by using seaborn library along with pandas. Stream-lit API had been used for creating dash board for providing a user friendly experience. As data used by the API is static hence for further improvement a dynamic model can be created which gets modified according to new addition in the datasets.

Bibliography :-

1. An introduction to seaborn — seaborn 0.11.2 documentation (pydata.org)
2. pandas documentation — pandas 1.3.4 documentation (pydata.org)
3. Streamlit • The fastest way to build and share data apps
4. A Beginners Guide To Streamlit - GeeksforGeeks
5. Streamlit bridges Python and React :: Streamlit Components Tutorial (streamlit-components-tutorial.netlify.app)