IPL Win Probability:- Victory is the ultimate goal in any sport.

This app predicts the win probability of both teams which is basically an attempt to predict the win probability of the teams in a given match at the end of each over and to look at the important factors affecting the match output.

There are studies that analyse the magnitude of the victory. It is found that most of these studies describe the factors affecting winning **but do not focus on the analysis of the factors with the goal of predicting the probability of victory**. In the real-world scenario, however, there are cases where the magnitude of the victory is important especially when betting is involved.

Sports analytics is the process of collecting past matches data and analyzing them to extract the essential knowledge out of it, with a hope that it facilitates in effective decision making. Decision making may be anything including which player to buy during an auction, which player to set on the field for tomorrow's match, or something more strategic task like, building the tactics for forthcoming matches based on players' previous performances.

The various factors that affect the outcome of a cricket match were analysed , and it was observed that home team, away team, venue, target, runs completed, over completed, batting/bowling team influence the win probability of a team.

Dataset:-

Used from Kaggle which had record of each and every bowl in all stadiums from 2008-2020.For Player retention dataset used was of 2021 as we will only need players who have played recent IPL season.

Data Preprocessing:-

Indian Premier League has been 11 years old, which is why only 634 matches data were available after the pre-processing. But the total balls bowled were around 72-73K which is massive.

Due to certain difficulties with some team franchises, in some seasons the league has seen the participation of new teams, and some teams have discontinued. Presence of those inactive teams in the dataset was not really necessary, but if the matches data were omitted where the inactive teams appeared, the chances were that the valuable knowledge about the teams which were still active in the league would deteriorate.

Also need to change name of certain teams in dataset before pre-processing. For e.g., Delhi daredevils name was changed to Delhi Capitals, Deccan Chargers to Sunrisers Hyderabad

As win probability will only be applicable during 2nd innings so we had to consider 2nd innings.

Also need to calculate Current Runrate and Required Run rate.

The final dataset considered only parameters which we need.

Algorithm:-

For predicting continuous values, Linear Regression appeared to be quite effective, and for classification problems like predicting the outcome of matches or classifying players, learning algorithms like **Logistic Regression, Random Forests** were found being used.

However in this model, I have used Logistic Regression as it generates a considerably correct probability. It does not directly gives 100% win probability or vice-versa. Even for team which is about to loose it still shows 1% winning probability.

In random forest, it showed the result as 100% or 0% in most of cases even if less overs were bowled.As we all know cricket or any sport is uncertain. We cannot predict winner based on start of game so this model cannot be considered in real-world.

Accuracy:-80.44% using logistic regression & 99.8% using random forest classifier.Test size was 20%.

Player Retention:-

The **Naive Bayesian** classifier was used to classify the performance of all-rounder players (bowler plus batsman) into four various non-overlapping categories, viz., a performer, a batting all-rounder, a bowling allrounder or an underperformer by being based on their strike rate and economy rate. When validated, the Naive Bayesian model was able to classify 66.7% of the all-rounders correctly

Logistic Regression:-

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.

Real World Usage:-ESPN cricinfo

Webapp:-

Streamlit was used. Stream lit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps.

Actually, you can write a whole app using python and markdown. Under the hood, Streamlit is ran by React. Besides the functionalities it comes with, you can install components to give you extra functionalities such as embedding video, code snippets, or animations. All from the comfort of python. You can go to their home page here to learn more.

Advantages of Streamlit

- No need to worry about routing.
- No need to worry about front end development.
- Extremely easy to learn.
- Super-fast development to deployment time. Literally minutes.

Advantage:- Every time you want to update your app, save the source file. When you do that, Streamlit detects if there is a change and asks you whether you want to rerun your app. Choose "Always rerun" at the top-right of your screen to automatically update your app every time you change its source code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app.

Disadvantages of Streamlit

- Will not scale.
- Cannot easily customize any of the frontend components.
- Relatively new, so there are features that are still beta.
- Since it’s relatively new, sometimes it’s hard to find answers to your questions.

Streamlit is a solution to rapidly develop apps with minimal code, and Flask is a solution to create backends and APIs for apps.

Streamlit vs. Flask

Streamlit is a data dashboarding tool, while Flask is a web framework. Serving pages to users is an important but small component of data dashboards. Flask doesn't have any data visualization, manipulation, or analytical capabilities (though since it's a general Python library, it can work well with other libraries that perform these tasks). Streamlit is an all-in-one tool that encompases web serving as well as data analysis.

- **Use Streamlit** if you want a structured data dashboard with many of the components you'll need already included. Use Streamlit if you want to build a data dashboard with common components and don't want to reinvent the wheel.
- **Use Flask** if you want to build a highly customized solution from the ground up and you have the engineering capacity.