1 TITLE OF THE PROJECT- IPL EXTRACT

2 PROJECT IDEA

The goal of our project is to find the various teams' matches data based on historical data like toss winner, toss decision, result of the match, won by runs, won by wickets, player of the match, umpires. We are going to work on two datasets- matches and deliveries of the IPL from 2008 to 2019 data. The outcome of this project is buildingmodels to predict match results and find out which factors are influential in winning the match. We are going to write goals and stories for this data which works as predictive analysis.

3 TOOLS AND TECHNOLOGIES

We are going with pyspark for data processing, and we may use tableau for visualization. PySpark is the Python API for Apache Spark, an open source, distributed computing framework and set of libraries for real-time, large-scale data processing. If you're already familiar with Python and libraries such as Pandas, then PySpark is a good language to learn to create more scalable analyses and pipelines. PySpark can significantly accelerate analysis by making it easy to combine local and distributed data transformation operations while keeping control of computing costs. In addition, the language helps data scientists to avoid always having to down sample large sets of data.

4 HIGH-LEVEL ARCHITECTURE OR METHODOLOGY OF PROJECT

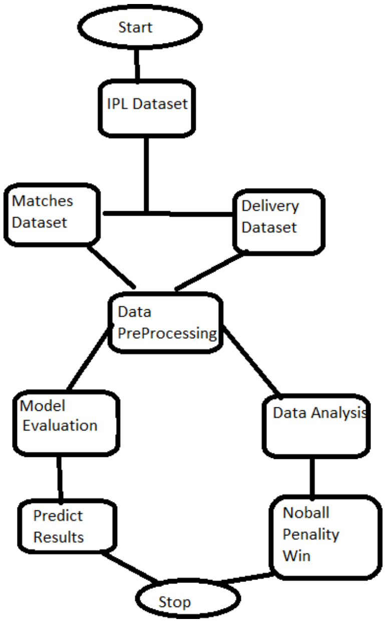

Fig. 1. High-Level Architecture Diagram

5 ARCHITECTURE DIAGRAM EXPLAINATION

- We are working on IPL Dataset from seasons 2008 to 2019. It consists of two separate datasets as matches and deliveries.

- The project used to predict the IPL matches using these two datasets with fields like season, city, date, team1, team2, toss winner, toss decision, result of match, dl applied, win by runs, win by wickets, player of match, venue, umpire1, umpire 2 and umpire 3.
- Data is processed using Pyspark queries.
- Model evaluation is done with conclusion drawn are written in the project documentation report.
- Data Analysis is achieved by formulating the goals to predict the performance of the players and IPL team's management use this data analysis for better games.
- Prediction analysis is achieved by conclusions drawn by the Stories with 5V's metric on the data which is helpful not only for team's management but also there are several betting and fantasy cricket platform, which are highly rely upon analytics for his or her success. Prediction Analytics can help all of them for his or her success.
- Resources are maintained through GitHub repo through URL where more conclusions and citationsused for this data which can addition a help for further research on this data.

6 GOALS

(1) Goal to find player of the match from 2008 to 2019 based on season, city, and venue where player of the match is Virat Kohli.

```
spark.sql(' select season, player_of_match, venue, city from matches where player_of_match="V Kohli"').show()
```

(2) To find the matches which are tie from 2008 to 2019.

```
spark.sql('select season, team1, team2, result from matches where result = "tie"').show()
```

(3) To find the highest number of wins by each team from 2008 to 2019 based on result.

```
spark.sql('select winner, count(result) as result from matches where group by winner order by winner').show()
```

(4) To find the winning team with 100+ margin of runs based on season, toss winner and player of the match.

```
spark.sql('select season, winner, toss_winner, win_by_runs, player_of_match from matches where win_by_runs > 100').show()
```

(5) To find the team which wins by 10 wickets margin based on season, winner, toss_winner, toss_decision, venue and player of the match.

```
spark.sql('select season, winner, toss_winner, toss_decision, venue, player_of_match from matches where win_by_wickets >= 10').show()
```

(6) Query to find the count of player_of_matches won by the individual players.

```
spark.sql('select player_of_match, count(player_of_match) from matches group by player_of_match').show()
```

(7) Goal to combine two tables using Join where winner and player of the match from matches table and batsman, non-striker and is super over from deliveries table.

```
spark.sql('select matches.winner, matches.player_of_match, deliveries.batsman, deliveries.non_striker, deliveries.is_super_over = 1 FROM matches JOIN deliveries where matches.id = deliveries.id').show()
```

(8) The goal is to find the no.of runs conceded by each bowler in all through seasons (from deliveries table).

```
spark.sql('SELECT bowler, count(total_runs) FROM deliveries group by bowler order by count(total_runs) desc').show()
```