ML Project

IPL Win Probability Prediction [Road Map]

Design a ML model that can predict the winning probability of an IPL team & deploy this model to Heroku. (using Streamlit for GUI creation).

Steps:

1. Imports
2. Read both datasets → matches, deliveries
3. Inspect dfs → shape, describe, info. (no cleaning, just confirm)
4. Groupby (match_id, inning) .sum() → total runs cols.
   ↳ Store res. in total_run_df
5. Capture only 1st innings, as we will be predicting for the 2nd inning.

```
totalrundf = totalrundf[totalrundf['inning'] == 1]
```

6. Merge the total first innings score df with matches df,

match_df = matches.. merge(totalruns_df [[matchid, totalruns]]

left_on = id , right_on = match_id)

7. In match_df, check for unique teams in col. "team1".

8. In match_df, replace 'Delhi Daredevils' with 'Delhi Capitals'.
'Deccan Chargers' with 'Sunrisers Hyderabad'.

9. All possible Univariate Analysis on this data. } Write
10. —"— Bivariate —"— } down

your insights in Markdown cell.

11. delivery_df = match_df.merge(deliveries, on = 'match_id')

12. Consider only 2nd inning
deliv_df = delivery_df [df [inning] == 2]

13. Create a new col. 'Current Score' of a particular match.
df ['Curr Score'] = df.groupby(matchid).cumsum()

14. Runs left

ddf['runsleft'] = total_runs_x − Current_score

15. Balls left = 126 − (over * 6 + current_ball)

16. Check if 'player_dismissed' has nan's.

if yes, fill them with 0.

17. Convert player_dismissed col. to boolean. (lambda)
1 = Dismissed
0 = Not dismissed

18. Current Run_rate — new col.

$$\frac{(ddf[\text{curr\_score}] * 6)}{(120 - ddf[\text{balls\_left}])}$$

19. Reqd. run rate — new col.

= (ddf[runs_left] * 6) / (ddf[balls_left])

20. Create new col. → result

if batting_team == winner ⇒ 1
else 0

21. Create a df called 'final_df' consisting of following cols.

- batting_team
- bowling_team
- city
- runs_left
- balls_left
- wickets
- total_runs_x
- curr_run_rate
- req_run_rate
- result.

Shape: (72413, 10)

22. Check for na & drop if any.

23. Filter.

```
final_df = final_df[final_df['balls_left'] != 0]
```

24. train ← every thing other than result col.

test ← result col.

25. Train Test Split

Note: Research on Column Transformer, Sklearn Pipeline } Use of this is mandatory

26. Betting team, Bowling team & city are Categorical cols. Convert it to numeric using "OneHotEncoder" ——— research on This.

27. Create a Pipeline
Two steps: (1) 26
(2) ML algo.

28. Predict using Pipeline.

29. Create Multiple models using different pipelines. (Logistic Reg, RFC)
predict ^ using predict_proba()

30. Create a pickle file & dump it in local drive.

app.py ⇒ GUI using Streamlit

(v.v.easy)