Project Report Format

1. INTRODUCTION:

1.1 Project overview:

The T20 score prediction model aims to forecast cricket match outcomes by analysing historical data, player statistics, pitch conditions, and other relevant factors. Leveraging machine learning algorithms, it generates predictions for team scores in T20 matches, providing insights into potential match outcomes based on various parameters. The model continuously refines its predictions as new data becomes available, enhancing its accuracy over time.

1.2 Purpose:

The purpose of the T20 score prediction model is to assist cricket enthusiasts, analysts, and sports betting professionals by providing data-driven forecasts for T20 matches. By leveraging advanced analytics and historical performance data, the model aims to offer valuable insights into potential team scores, aiding in informed decision-making and enhancing the overall viewing and engagement experience for cricket fans.

2. LITERATURE REVIEW:

2.1 Existing problem:

One existing problem with T20 score predictors is the inherent unpredictability of the format due to its fast-paced and dynamic nature. Factors such as unexpected player performances, rapidly changing game situations, and unique pitch conditions pose challenges for accurate predictions. Additionally, external variables like weather conditions and player injuries further complicate prediction models. Balancing these uncertainties while maintaining a high level of accuracy remains a challenge for T20 score predictors.

2.2 References:

- https://www.researchgate.net/publication/319335442_Dynamic_Winner_Prediction_in_Twenty20_Cricket_Based_on_Relative_Team_Strengths.
- https://ieeexplore.ieee.org/document/9807929
- https://www.researchgate.net/publication/353718778_An_Analytical_Model_for_Prediction_of_Upcoming_ICC_T20_World_Cup_2021_Using_Classification_Algorithms
- https://ieeexplore.ieee.org/abstract/document/9318077
- https://ieeexplore.ieee.org/document/8820235
- https://www.irjet.net/archives/V4/i3/IRJET-V4I3269.pdf

- https://ieeexplore.ieee.org/document/8966691
- https://ieeexplore.ieee.org/document/9642558

2.3 Problem Statement Definition:

The problem in T20 score prediction lies in developing a model that effectively navigates the inherent volatility of the format, accounting for rapid changes in game dynamics, unpredictable player performances, and external factors like pitch conditions and weather. Balancing the need for accuracy with the dynamic nature of T20 cricket poses a challenge for creating reliable and robust score prediction systems.

3. IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map Canvas:

Cricket enthusiasts seeking T20 score predictions desire accuracy, real-time updates, and adaptability to the format's dynamic nature. They value insights into player performance and match dynamics, with frustration arising from inaccurate predictions. Trust is essential, necessitating transparency in models that consider various variables and account for unexpected events during T20 matches, enhancing overall engagement and enjoyment.

3.2 Ideation & Brainstorming:

Ideation and brainstorming for T20 score prediction involve generating innovative concepts and strategies. Collaborative sessions focus on incorporating machine learning algorithms, considering player statistics, pitch conditions, and dynamic game elements. The goal is to create a predictive model that adapts to the fast-paced nature of T20 cricket, emphasizing real-time updates, transparency, and a holistic approach to enhance accuracy and user engagement.

4. REQUIREMENT ANALYSIS:

4.1 Functional requirement:

Functional requirements for T20 score prediction include real-time data integration, machine learning algorithms for predictive analysis, consideration of player statistics, pitch conditions, and dynamic match situations. The system should offer accurate score predictions, continuous updates during matches, and adaptability to unforeseen events. User-friendly interfaces for accessing predictions and comprehensive reporting are essential, ensuring a seamless and informative experience for users.

4.2 Non-Functional requirement:

Non-functional requirements for T20 score prediction encompass aspects such as system reliability, scalability to handle varying user loads, responsiveness for real-time updates, and adaptability to dynamic match conditions. Security measures to protect sensitive data, user-friendly interfaces, and adherence to industry standards are crucial. Additionally, the system should exhibit high accuracy and transparency in predictions, fostering user trust and satisfaction.

5. PROJECT EDSIGN:

5.1 Data Flow Diagrams & User Stories:

Data Flow Diagrams (DFDs) for T20 score prediction illustrate the flow of information, showing how data moves between entities like databases, prediction algorithms, and user interfaces. User stories articulate end-user requirements, such as "As a cricket fan, I want real-time score predictions during T20 matches to enhance my viewing experience," guiding the development of features that align with user needs and system functionality.

5.2 Solution Architecture:

The solution architecture for T20 score prediction involves a multi-layered approach, incorporating data acquisition from diverse sources such as player statistics and match conditions. Machine learning algorithms analyse this data, generating real-time predictions. The system integrates with user interfaces for accessibility, ensuring scalability, reliability, and adherence to security protocols. A modular and flexible architecture allows for continuous improvement and adaptation to the dynamic nature of T20 cricket.

6. PROJECT PLANNING & SCHEDULING:

6.1 Technical Architecture:

The technical architecture of T20 score prediction includes data sources integration, a machine learning module for predictive analysis, a database for storing historical and real-time data, and user interfaces for result presentation. It may leverage cloud infrastructure for scalability and flexibility, with APIs facilitating data communication. Robust security measures ensure the protection of sensitive information, while continuous monitoring and updates maintain system performance and accuracy.

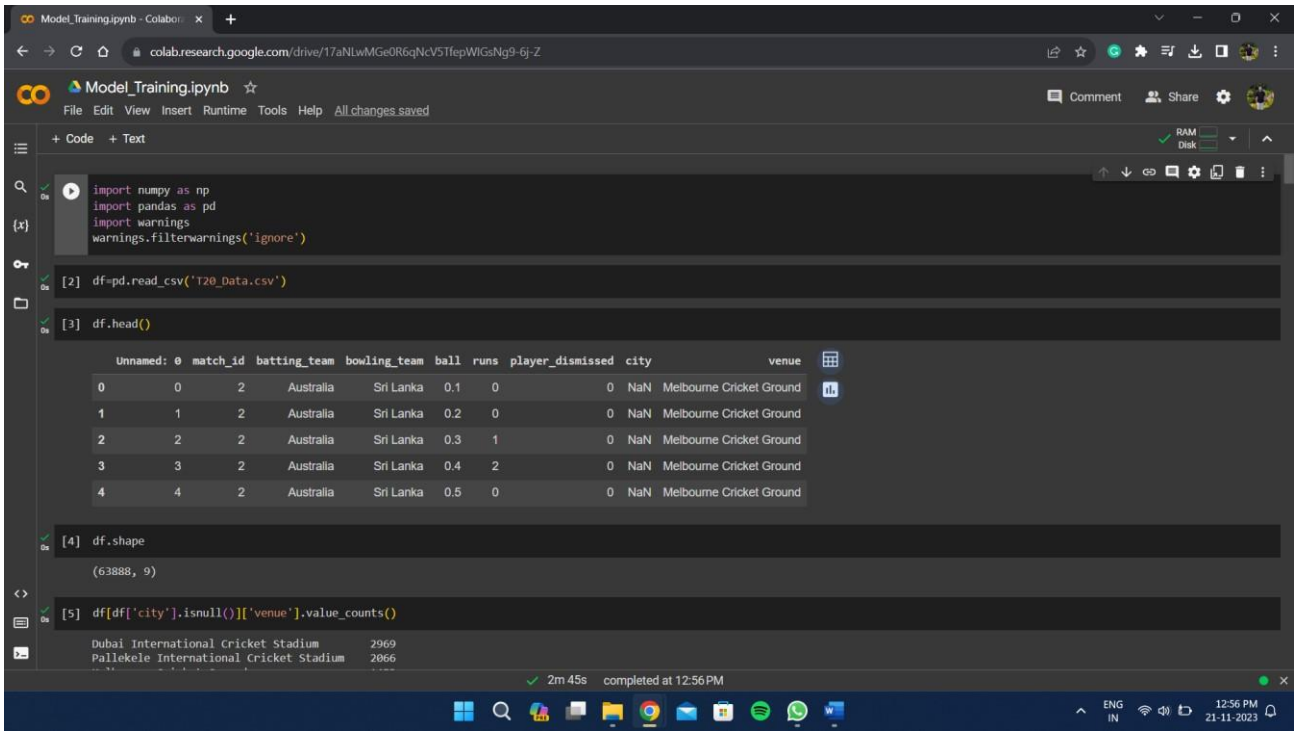
6.2 Sprint Planning & Estimation:

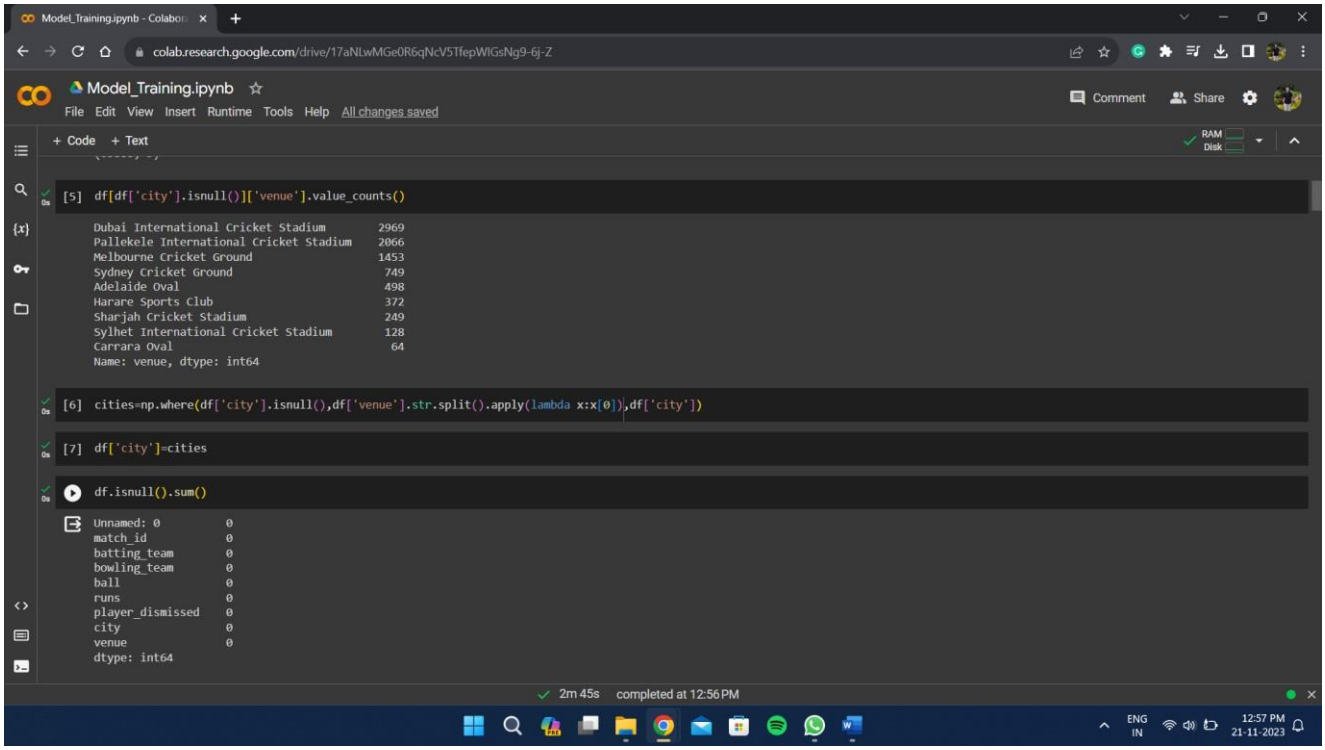
Sprint planning for T20 score prediction involves collaborative sessions to prioritize features and tasks, breaking them into manageable units. Estimation involves assigning story points to tasks based on complexity and effort. The agile methodology ensures iterative development, allowing the team to adapt and refine the T20 score prediction model incrementally. Regular sprint reviews and retrospectives further enhance the development process.

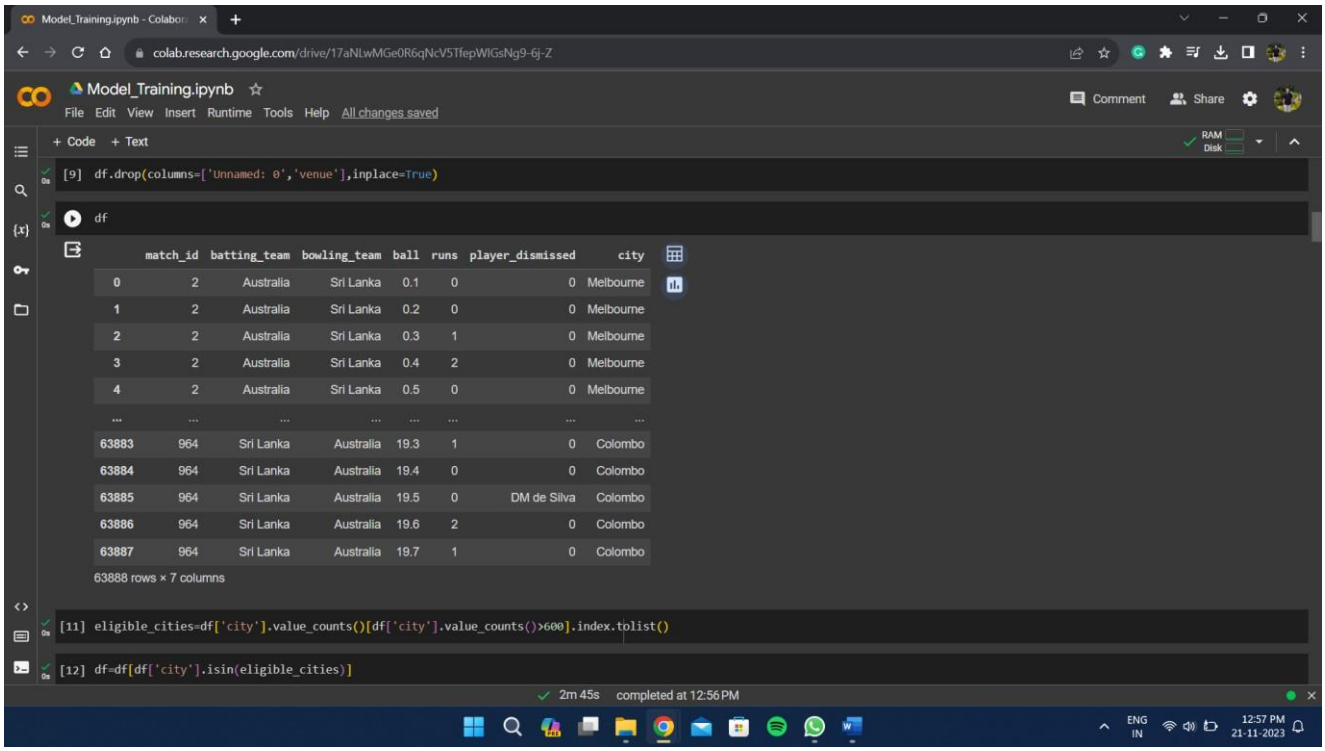
6.3 Sprint Delivery Schedule:

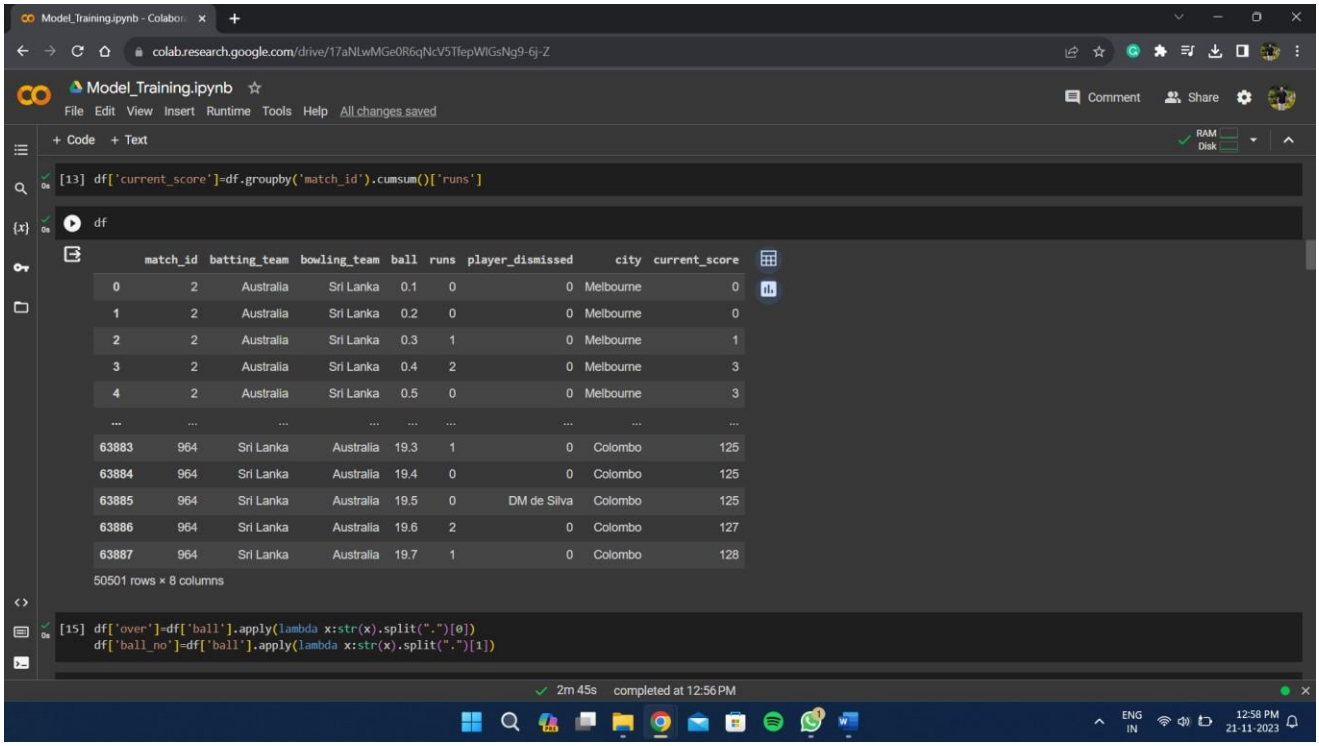
Sprint delivery for T20 score prediction involves releasing increments of the prediction model at the end of each sprint cycle. This iterative approach allows for continuous improvements based on user feedback and changing requirements. The delivered increments provide tangible value, such as enhanced prediction accuracy or additional features, ensuring that the T20 score prediction system evolves and meets user expectations over time.

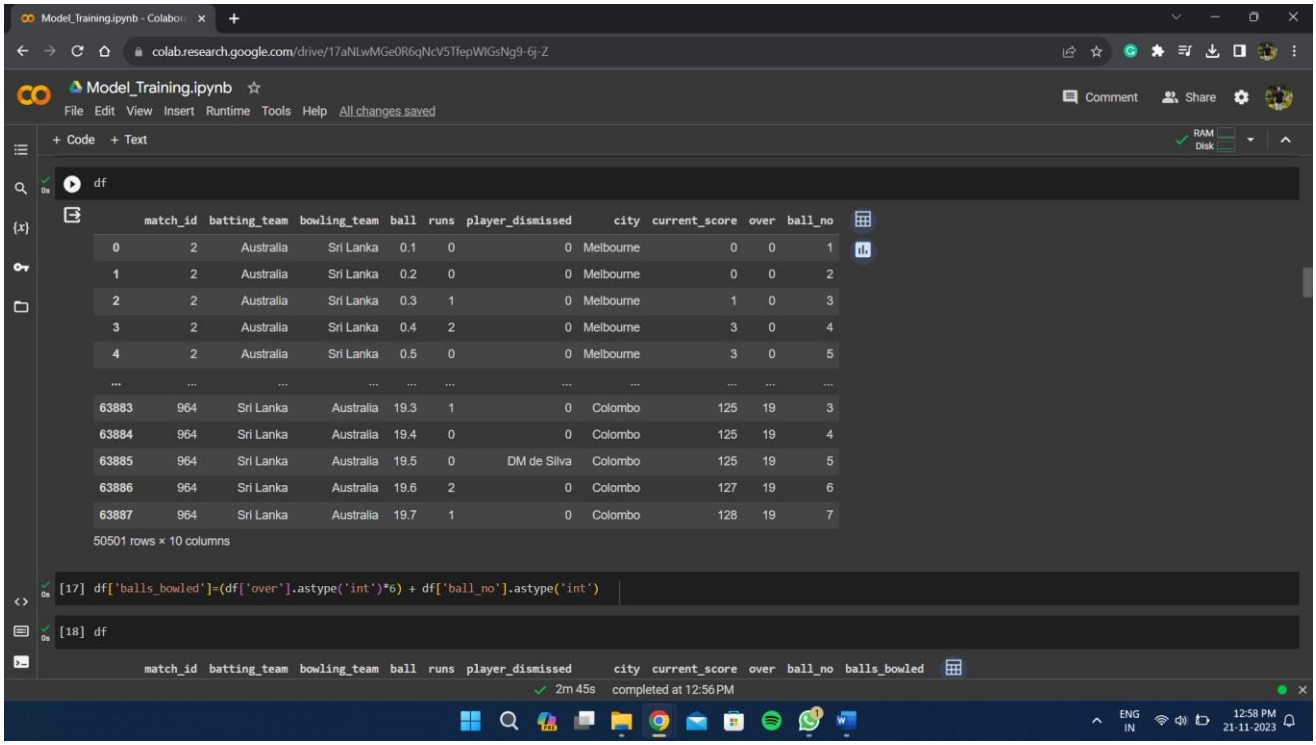
7. CODING & SOLUTIONING:

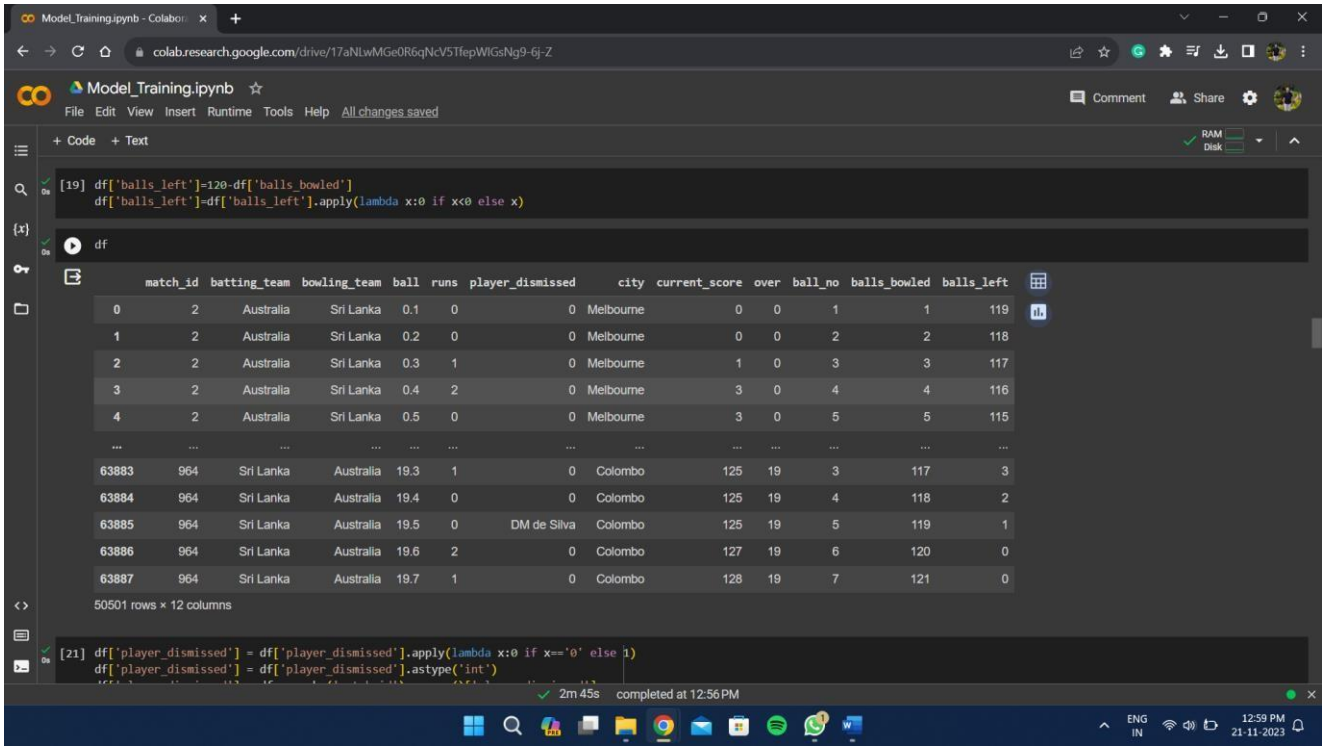


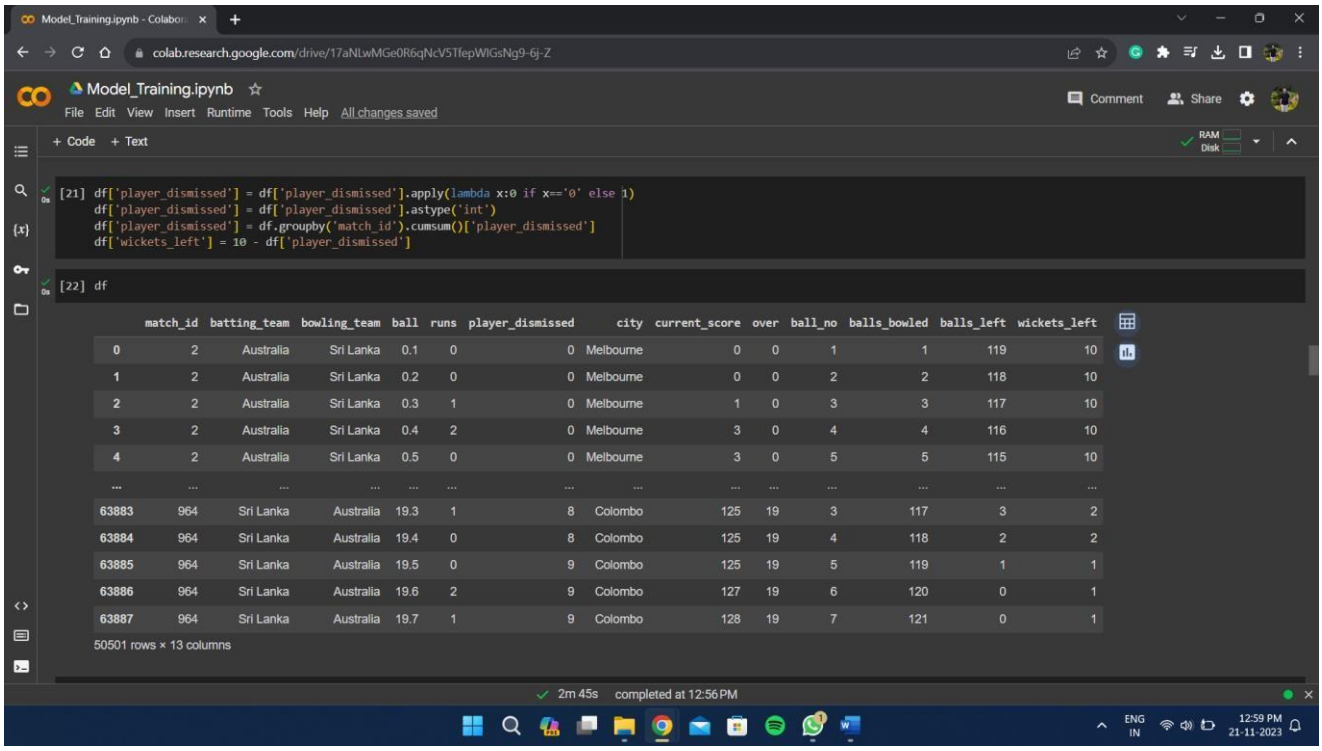

Model_Training.ipynb - Colabor
colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z
Model_Training.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[5] df[df['city'].isnull()]['venue'].value_counts()
Dubai International Cricket Stadium 2969
Pallekele International Cricket Stadium 2066
Melbourne Cricket Ground 1453
Sydney Cricket Ground 749
Adelaide Oval 498
Harare Sports Club 372
Sharjah Cricket Stadium 249
Sylhet International Cricket Stadium 128
Carrara Oval 64
Name: venue, dtype: int64
[6] cities=np.where(df['city'].isnull(),df['venue'].str.split().apply(lambda x:x[0]),df['city'])
[7] df['city']=cities
df.isnull().sum()
Unnamed: 0 0
match_id 0
batting_team 0
bowling_team 0
ball 0
runs 0
player_dismissed 0
city 0
venue 0
dtype: int64
2m 45s completed at 12:56 PM
ENG
IN
12:57 PM
21-11-2023

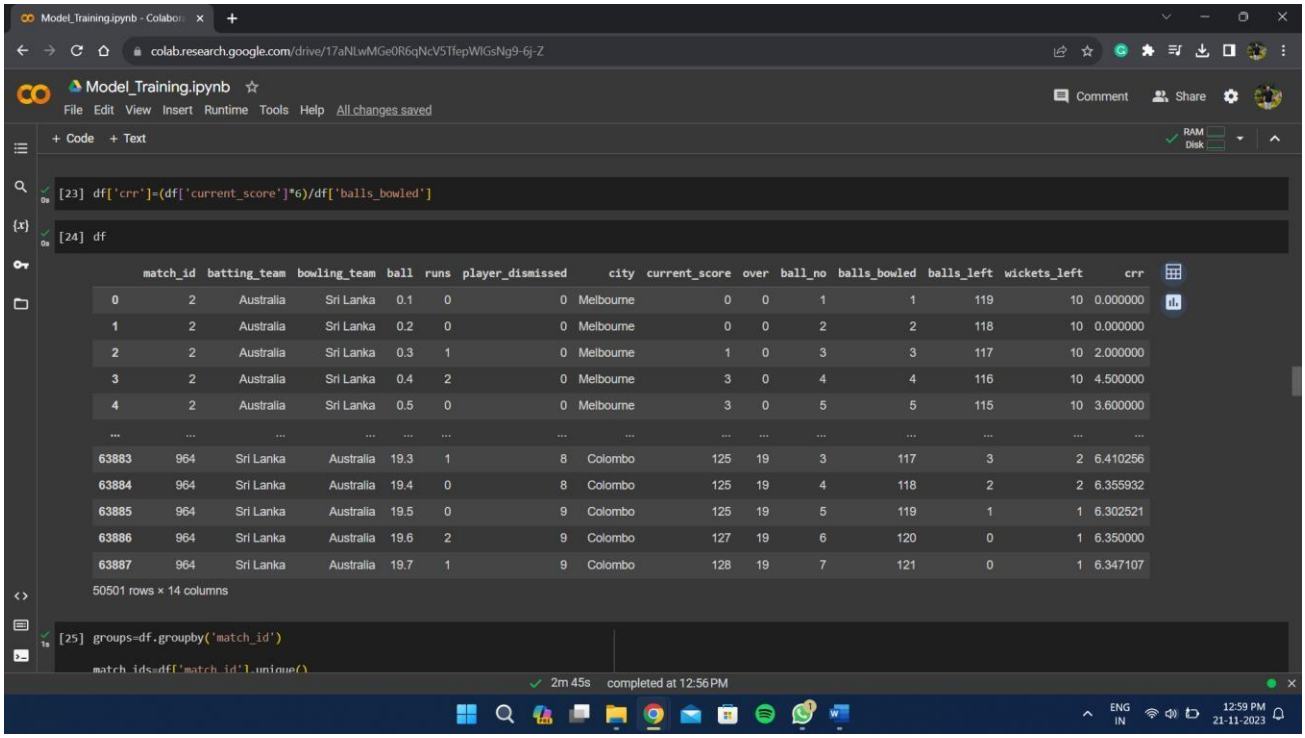

Model_Training.ipynb - Colabor
colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z
Model_Training.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[9] df.drop(columns=['Unnamed: 0','venue'],inplace=True)
df
match_id batting_team bowling_team ball runs player_dismissed city
0 2 Australia Sri Lanka 0.1 0 0 Melbourne
1 2 Australia Sri Lanka 0.2 0 0 Melbourne
2 2 Australia Sri Lanka 0.3 1 0 Melbourne
3 2 Australia Sri Lanka 0.4 2 0 Melbourne
4 2 Australia Sri Lanka 0.5 0 0 Melbourne
63883 964 Sri Lanka Australia 19.3 1 0 Colombo
63884 964 Sri Lanka Australia 19.4 0 0 Colombo
63885 964 Sri Lanka Australia 19.5 0 DM de Silva Colombo
63886 964 Sri Lanka Australia 19.6 2 0 Colombo
63887 964 Sri Lanka Australia 19.7 1 0 Colombo
63888 rows × 7 columns
[11] eligible_cities=df['city'].value_counts()[df['city'].value_counts()>600].index.tolist()
[12] df=df[df['city'].isin(eligible_cities)]
2m 45s completed at 12:56 PM
ENG
IN
12:57 PM
21-11-2023


Model_Training.ipynb - Colabor
colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z
Model_Training.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[13] df['current_score']=df.groupby('match_id').cumsum()['runs']
df
match_id batting_team bowling_team ball runs player_dismissed city current_score
0 2 Australia Sri Lanka 0.1 0 0 Melbourne 0
1 2 Australia Sri Lanka 0.2 0 0 Melbourne 0
2 2 Australia Sri Lanka 0.3 1 0 Melbourne 1
3 2 Australia Sri Lanka 0.4 2 0 Melbourne 3
4 2 Australia Sri Lanka 0.5 0 0 Melbourne 3
... ...
63883 964 Sri Lanka Australia 19.3 1 0 Colombo 125
63884 964 Sri Lanka Australia 19.4 0 0 Colombo 125
63885 964 Sri Lanka Australia 19.5 0 DM de Silva Colombo 125
63886 964 Sri Lanka Australia 19.6 2 0 Colombo 127
63887 964 Sri Lanka Australia 19.7 1 0 Colombo 128
50501 rows × 8 columns
[15] df['over']=df['ball'].apply(lambda x:str(x).split(".")[0])
df['ball_no']=df['ball'].apply(lambda x:str(x).split(".")[1])
2m 45s completed at 12:56 PM
ENG
IN
12:58 PM
21-11-2023


Model_Training.ipynb - Colabor
colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z
Model_Training.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
df
match_id batting_team bowling_team ball runs player_dismissed city current_score over ball_no
0 2 Australia Sri Lanka 0.1 0 0 Melbourne 0 0 1
1 2 Australia Sri Lanka 0.2 0 0 Melbourne 0 0 2
2 2 Australia Sri Lanka 0.3 1 0 Melbourne 1 0 3
3 2 Australia Sri Lanka 0.4 2 0 Melbourne 3 0 4
4 2 Australia Sri Lanka 0.5 0 0 Melbourne 3 0 5
... ...
63883 964 Sri Lanka Australia 19.3 1 0 Colombo 125 19 3
63884 964 Sri Lanka Australia 19.4 0 0 Colombo 125 19 4
63885 964 Sri Lanka Australia 19.5 0 DM de Silva Colombo 125 19 5
63886 964 Sri Lanka Australia 19.6 2 0 Colombo 127 19 6
63887 964 Sri Lanka Australia 19.7 1 0 Colombo 128 19 7
50501 rows × 10 columns
[17] df['balls_bowled']=(df['over'].astype('int')*6) + df['ball_no'].astype('int')
[18] df
match_id batting_team bowling_team ball runs player_dismissed city current_score over ball_no balls_bowled
2m 45s completed at 12:56 PM
ENG
IN
12:58 PM
21-11-2023

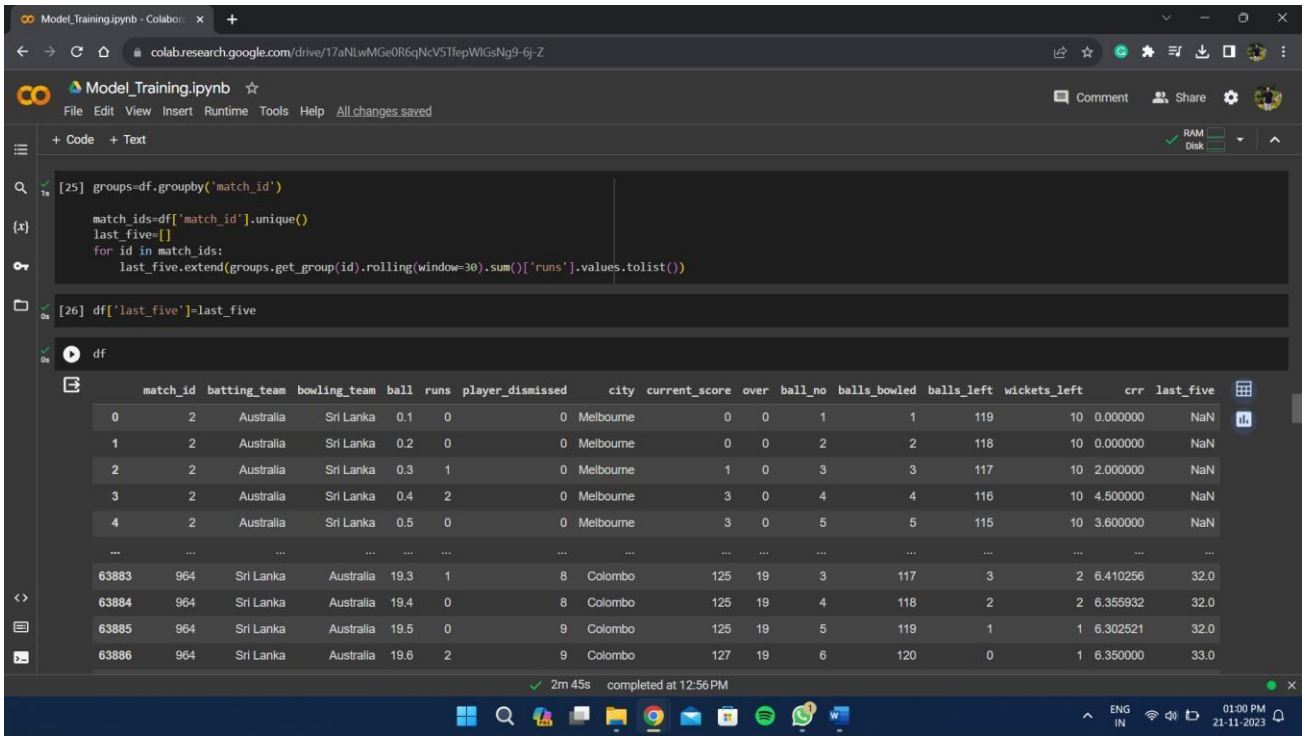

Model_Training.ipynb - Colabor
colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z
Model_Training.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[19] df['balls_left']=120-df['balls_bowled']
df['balls_left']=df['balls_left'].apply(lambda x:0 if x<0 else x)
df
match_id batting_team bowling_team ball runs player_dismissed city current_score over ball_no balls_bowled balls_left
0 2 Australia Sri Lanka 0.1 0 0 Melbourne 0 0 1 1 119
1 2 Australia Sri Lanka 0.2 0 0 Melbourne 0 0 2 2 118
2 2 Australia Sri Lanka 0.3 1 0 Melbourne 1 0 3 3 117
3 2 Australia Sri Lanka 0.4 2 0 Melbourne 3 0 4 4 116
4 2 Australia Sri Lanka 0.5 0 0 Melbourne 3 0 5 5 115
63883 964 Sri Lanka Australia 19.3 1 0 Colombo 125 19 3 117 3
63884 964 Sri Lanka Australia 19.4 0 0 Colombo 125 19 4 118 2
63885 964 Sri Lanka Australia 19.5 0 DM de Silva Colombo 125 19 5 119 1
63886 964 Sri Lanka Australia 19.6 2 0 Colombo 127 19 6 120 0
63887 964 Sri Lanka Australia 19.7 1 0 Colombo 128 19 7 121 0
50501 rows × 12 columns
[21] df['player_dismissed'] = df['player_dismissed'].apply(lambda x:0 if x=='0' else 1)
df['player_dismissed'] = df['player_dismissed'].astype('int')
2m 45s completed at 12:56 PM
ENG
IN
12:59 PM
21-11-2023


Model_Training.ipynb - Colabor
colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z
Model_Training.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[21] df['player_dismissed'] = df['player_dismissed'].apply(lambda x:0 if x=='0' else 1)
df['player_dismissed'] = df['player_dismissed'].astype('int')
df['player_dismissed'] = df.groupby('match_id').cumsum()['player_dismissed']
df['wickets_left'] = 10 - df['player_dismissed']
[22] df
match_id batting_team bowling_team ball runs player_dismissed city current_score over ball_no balls_bowled balls_left wickets_left
0 2 Australia Sri Lanka 0.1 0 0 Melbourne 0 0 1 1 119 10
1 2 Australia Sri Lanka 0.2 0 0 Melbourne 0 0 2 2 118 10
2 2 Australia Sri Lanka 0.3 1 0 Melbourne 1 0 3 3 117 10
3 2 Australia Sri Lanka 0.4 2 0 Melbourne 3 0 4 4 116 10
4 2 Australia Sri Lanka 0.5 0 0 Melbourne 3 0 5 5 115 10
63883 964 Sri Lanka Australia 19.3 1 8 Colombo 125 19 3 117 3 2
63884 964 Sri Lanka Australia 19.4 0 8 Colombo 125 19 4 118 2 2
63885 964 Sri Lanka Australia 19.5 0 9 Colombo 125 19 5 119 1 1
63886 964 Sri Lanka Australia 19.6 2 9 Colombo 127 19 6 120 0 1
63887 964 Sri Lanka Australia 19.7 1 9 Colombo 128 19 7 121 0 1
50501 rows × 13 columns
2m 45s completed at 12:56 PM
ENG
IN
12:59 PM
21-11-2023

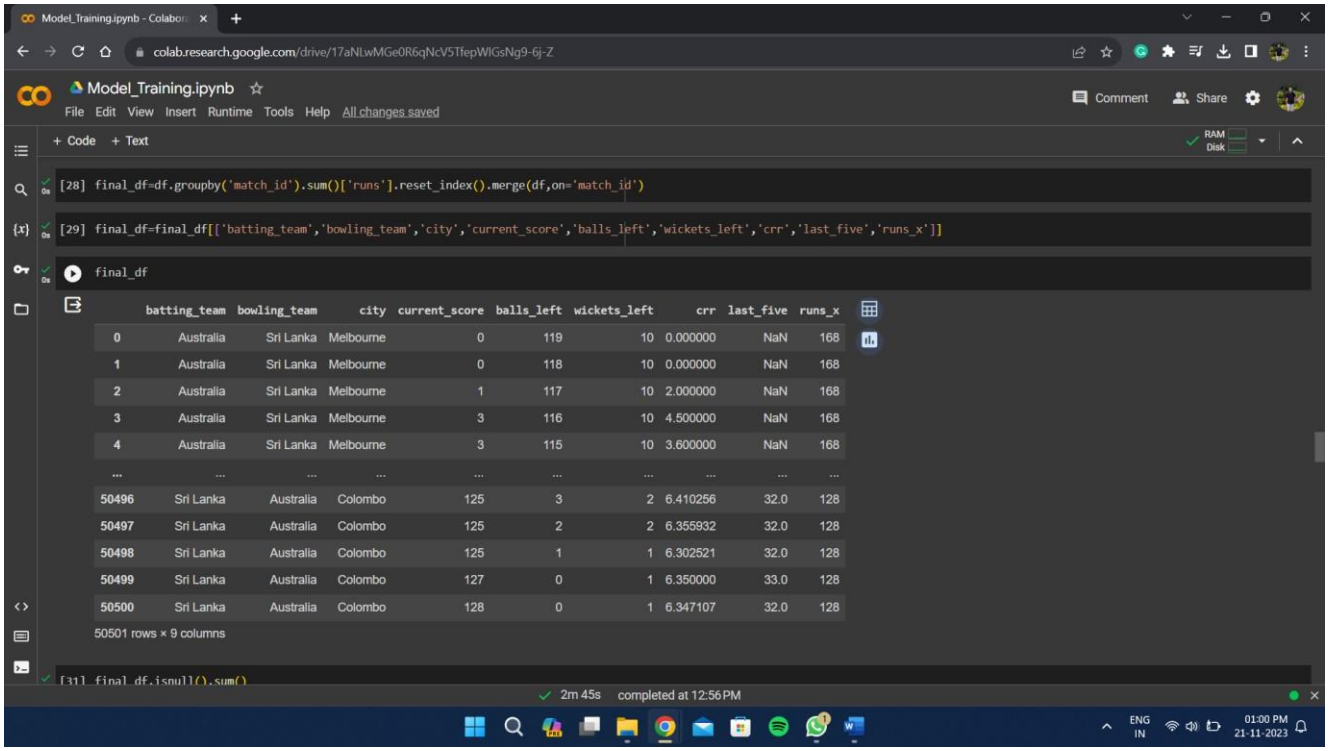


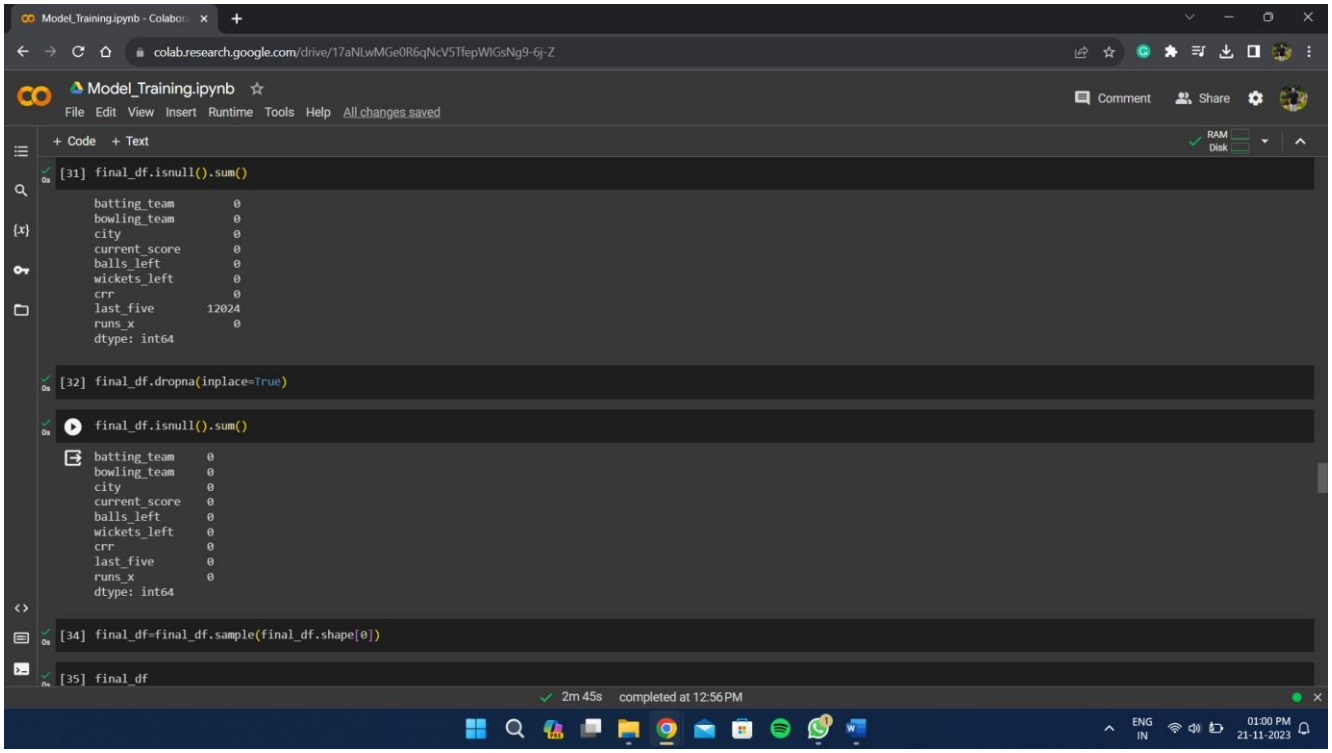
	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	current_score	over	ball_no	balls_bowled	balls_left	wickets_left	crr
0	2	Australia	Sri Lanka	0.1	0	0	Melbourne	0	0	1	1	119	10	0.000000
1	2	Australia	Sri Lanka	0.2	0	0	Melbourne	0	0	2	2	118	10	0.000000
2	2	Australia	Sri Lanka	0.3	1	0	Melbourne	1	0	3	3	117	10	2.000000
3	2	Australia	Sri Lanka	0.4	2	0	Melbourne	3	0	4	4	116	10	4.500000
4	2	Australia	Sri Lanka	0.5	0	0	Melbourne	3	0	5	5	115	10	3.600000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
63883	964	Sri Lanka	Australia	19.3	1	8	Colombo	125	19	3	117	3	2	6.410256
63884	964	Sri Lanka	Australia	19.4	0	8	Colombo	125	19	4	118	2	2	6.355932
63885	964	Sri Lanka	Australia	19.5	0	9	Colombo	125	19	5	119	1	1	6.302521
63886	964	Sri Lanka	Australia	19.6	2	9	Colombo	127	19	6	120	0	1	6.350000
63887	964	Sri Lanka	Australia	19.7	1	9	Colombo	128	19	7	121	0	1	6.347107

50501 rows × 14 columns



	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	current_score	over	ball_no	balls_bowled	balls_left	wickets_left	crr	last_five
0	2	Australia	Sri Lanka	0.1	0	0	Melbourne	0	0	1	1	119	10	0.000000	NaN
1	2	Australia	Sri Lanka	0.2	0	0	Melbourne	0	0	2	2	118	10	0.000000	NaN
2	2	Australia	Sri Lanka	0.3	1	0	Melbourne	1	0	3	3	117	10	2.000000	NaN
3	2	Australia	Sri Lanka	0.4	2	0	Melbourne	3	0	4	4	116	10	4.500000	NaN
4	2	Australia	Sri Lanka	0.5	0	0	Melbourne	3	0	5	5	115	10	3.600000	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
63883	964	Sri Lanka	Australia	19.3	1	8	Colombo	125	19	3	117	3	2	6.410256	32.0
63884	964	Sri Lanka	Australia	19.4	0	8	Colombo	125	19	4	118	2	2	6.355932	32.0
63885	964	Sri Lanka	Australia	19.5	0	9	Colombo	125	19	5	119	1	1	6.302521	32.0
63886	964	Sri Lanka	Australia	19.6	2	9	Colombo	127	19	6	120	0	1	6.350000	33.0


Model_Training.ipynb - Colabor
colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z
Model_Training.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[28] final_df=df.groupby('match_id').sum()['runs'].reset_index().merge(df,on='match_id')
[29] final_df=final_df[['batting_team','bowling_team','city','current_score','balls_left','wickets_left','crr','last_five','runs_x']]
final_df
batting_team bowling_team city current_score balls_left wickets_left crr last_five runs_x
0 Australia Sri Lanka Melbourne 0 119 10 0.000000 NaN 168
1 Australia Sri Lanka Melbourne 0 118 10 0.000000 NaN 168
2 Australia Sri Lanka Melbourne 1 117 10 2.000000 NaN 168
3 Australia Sri Lanka Melbourne 3 116 10 4.500000 NaN 168
4 Australia Sri Lanka Melbourne 3 115 10 3.600000 NaN 168
... ...
50496 Sri Lanka Australia Colombo 125 3 2 6.410256 32.0 128
50497 Sri Lanka Australia Colombo 125 2 2 6.355932 32.0 128
50498 Sri Lanka Australia Colombo 125 1 1 6.302521 32.0 128
50499 Sri Lanka Australia Colombo 127 0 1 6.350000 33.0 128
50500 Sri Lanka Australia Colombo 128 0 1 6.347107 32.0 128
50501 rows × 9 columns
[31] final_df.isnull().sum()
2m 45s completed at 12:56 PM
ENG
IN
01:00 PM
21-11-2023


Model_Training.ipynb - Colabor
colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z
Model_Training.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[31] final_df.isnull().sum()
batting_team 0
bowling_team 0
city 0
current_score 0
balls_left 0
wickets_left 0
crr 0
last_five 12024
runs_x 0
dtype: int64
[32] final_df.dropna(inplace=True)
final_df.isnull().sum()
batting_team 0
bowling_team 0
city 0
current_score 0
balls_left 0
wickets_left 0
crr 0
last_five 0
runs_x 0
dtype: int64
[34] final_df=final_df.sample(final_df.shape[0])
[35] final_df
2m 45s completed at 12:56 PM
ENG
IN
01:00 PM
21-11-2023

Model_Training.ipynb - Colabor

colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z

Model_Training.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

[36] #Train-Test-Split

```
X=final_df.drop(columns=['runs_x'])
y=final_df['runs_x']
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,y,test_size=0.2,random_state=1)
```

X_train

	batting_team	bowling_team	city	current_score	balls_left	wickets_left	crr	last_five
28689	India	England	Manchester	68	68	9	7.846154	35.0
30721	Sri Lanka	Pakistan	Abu Dhabi	128	20	6	7.680000	29.0
44882	India	Australia	Melbourne	96	56	10	9.000000	47.0
1098	India	England	Bangalore	177	12	7	9.833333	58.0
24531	Pakistan	England	Barbados	87	45	6	6.960000	29.0
...	...	...	...	...	...	...	...	...
30302	India	England	Kolkata	60	67	7	6.792453	40.0
38583	West Indies	New Zealand	Wellington	153	5	5	7.982609	54.0
33167	West Indies	New Zealand	Pallekele	96	48	6	8.000000	32.0
9289	Australia	Sri Lanka	Adelaide	178	29	9	11.736264	61.0
1908	England	Australia	Melbourne	50	74	7	6.521739	32.0

30781 rows × 8 columns

2m 45s completed at 12:56 PM

ENG IN 01:01 PM 21-11-2023

Model_Training.ipynb - Colabor

colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z

Model_Training.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

[38]
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import r2_score,mean_absolute_error
```

[39]
```
trf = ColumnTransformer([
    ('trf',OneHotEncoder(sparse=False,drop='first'),['batting_team','bowling_team','city'])
]
,remainder='passthrough')
```

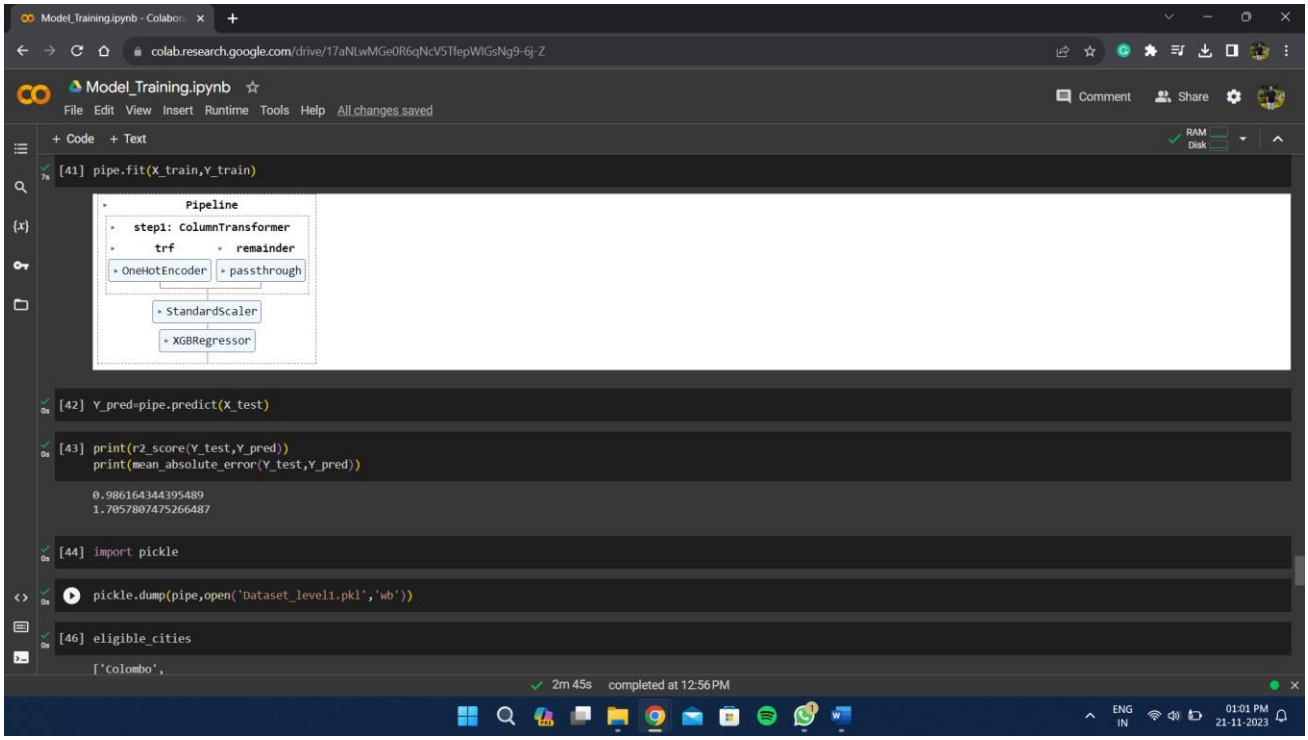
[40]
```
pipe=Pipeline(steps=[
      ('step1',trf),
      ('step2',StandardScaler()),
      ('step3',XGBRegressor(n_estimators=300,learning_rate=0.2,max_depth=12,random_state=1))

])
```

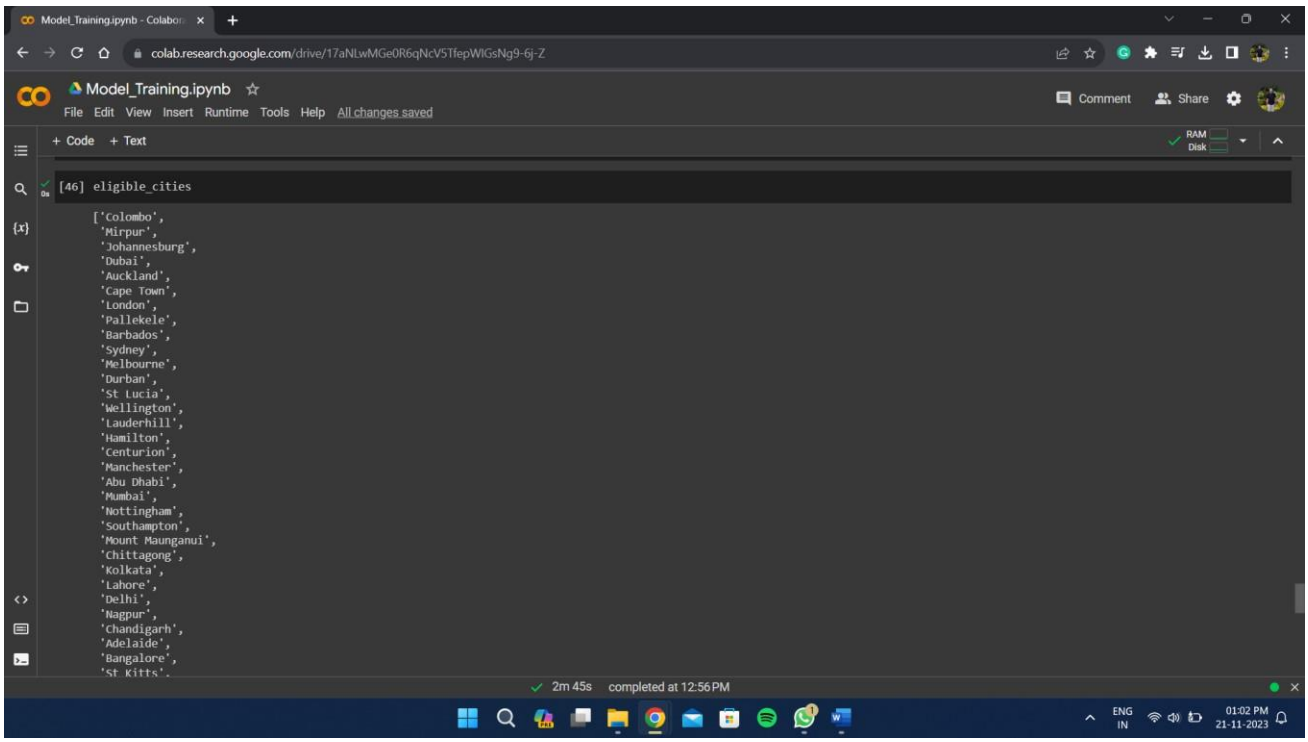
[41] pipe.fit(X_train,Y_train)

Pipeline
step1: ColumnTransformer
trf
OneHotEncoder
remainder
passthrough
StandardScaler

2m 45s completed at 12:56 PM

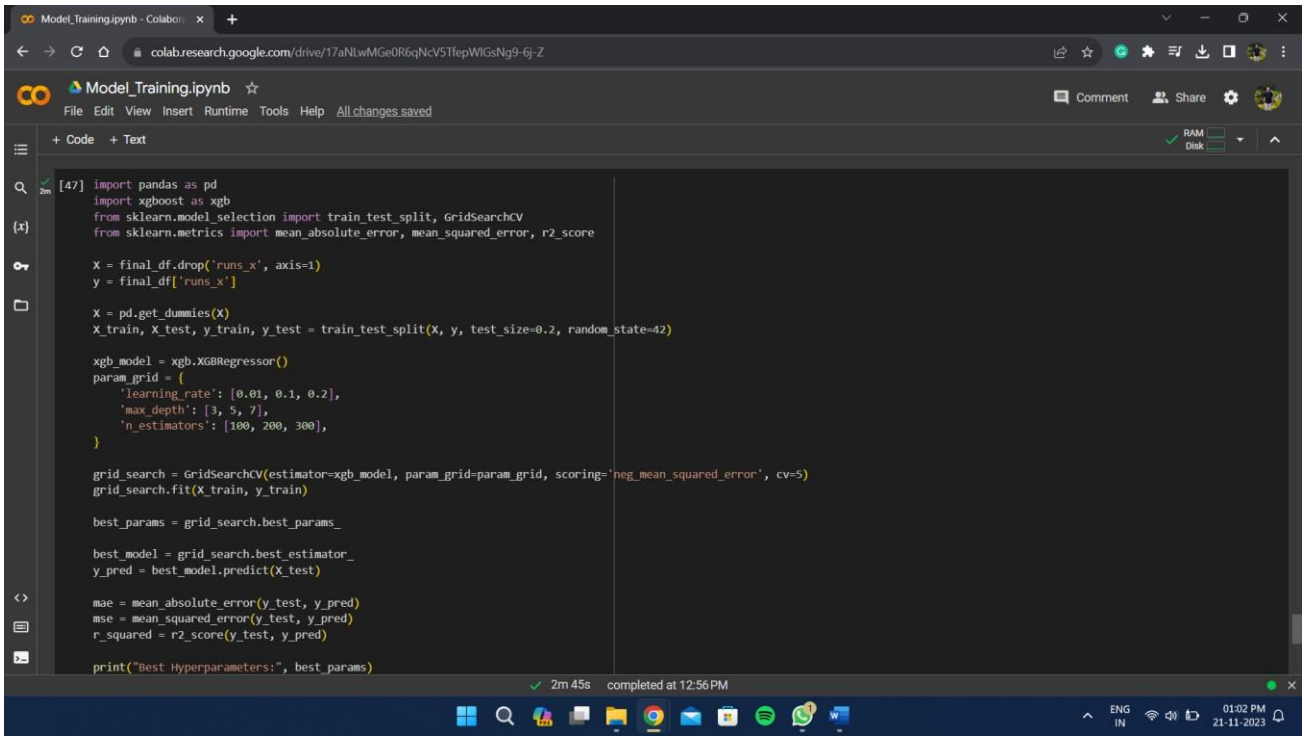
ENG IN 01:01 PM 21-11-2023


Model_Training.ipynb - Colabor
colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z
Model_Training.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[41] pipe.fit(X_train,Y_train)
Pipeline
step1: ColumnTransformer
trf
remainder
OneHotEncoder
passthrough
StandardScaler
XGBRegressor
[42] Y_pred=pipe.predict(X_test)
[43] print(r2_score(Y_test,Y_pred))
print(mean_absolute_error(Y_test,Y_pred))
0.986164344395489
1.705780475266487
[44] import pickle
pickle.dump(pipe,open('Dataset_level1.pkl','wb'))
[46] eligible_cities
['Colombo',
2m 45s completed at 12:56 PM
ENG
IN
01:01 PM
21-11-2023

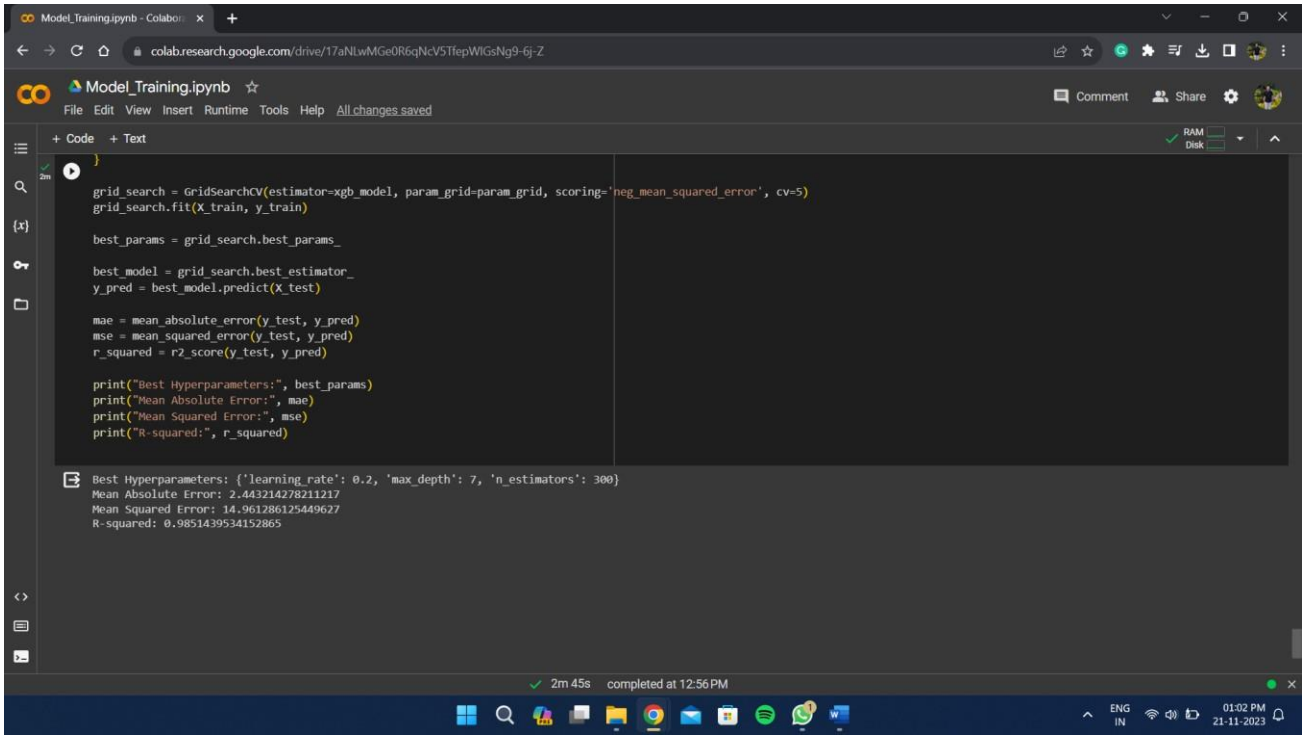

Model_Training.ipynb - Colabor
colab.research.google.com/drive/17aNLwMGe0R6qNcV5TfepWlGsNg9-6j-Z
Model_Training.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[46] eligible_cities
['Colombo',
'Mirpur',
'Johannesburg',
'Dubai',
'Auckland',
'Cape Town',
'London',
'Pallekele',
'Barbados',
'Sydney',
'Melbourne',
'Durban',
'St Lucia',
'Wellington',
'Lauderhill',
'Hamilton',
'Centurion',
'Manchester',
'Abu Dhabi',
'Mumbai',
'Nottingham',
'Southampton',
'Mount Maunganui',
'Chittagong',
'Kolkata',
'Lahore',
'Delhi',
'Nagpur',
'Chandigarh',
'Adelaide',
'Bangalore',
'St Kitts',
2m 45s completed at 12:56 PM
ENG
IN
01:02 PM
21-11-2023

8.

PERFORMANCE TESTING:



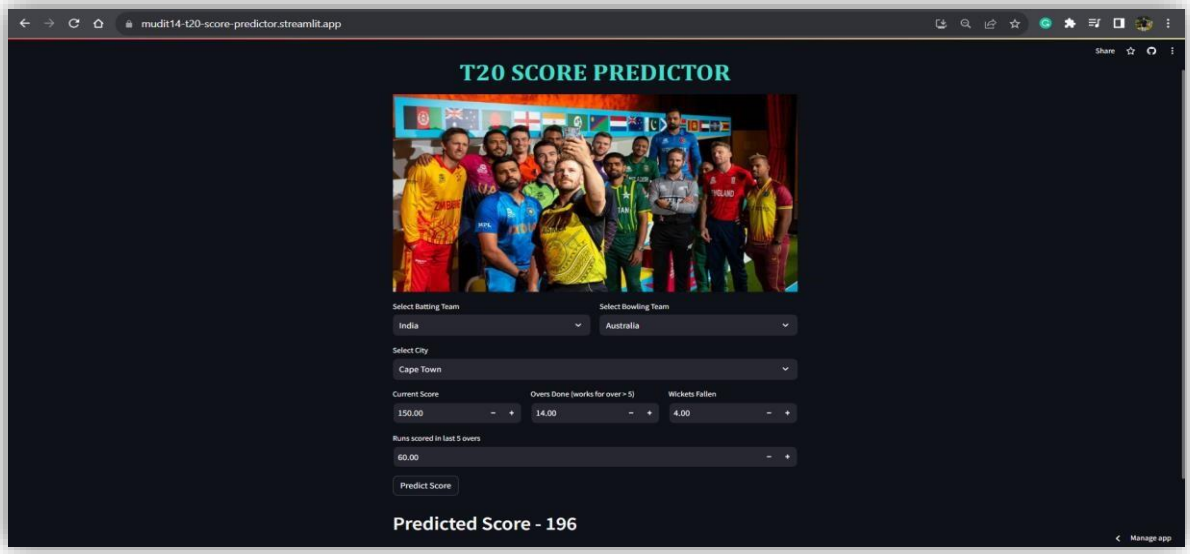
```
[47] import pandas as pd
     import xgboost as xgb
     from sklearn.model_selection import train_test_split, GridSearchCV
     from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

     X = final_df.drop('runs_x', axis=1)
     y = final_df['runs_x']

     X = pd.get_dummies(X)
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

     xgb_model = xgb.XGBRegressor()
     param_grid = {
         'learning_rate': [0.01, 0.1, 0.2],
         'max_depth': [3, 5, 7],
         'n_estimators': [100, 200, 300],
     }

     grid_search = GridSearchCV(estimator=xgb_model, param_grid=param_grid, scoring='neg_mean_squared_error', cv=5)
     grid_search.fit(X_train, y_train)

     best_params = grid_search.best_params_

     best_model = grid_search.best_estimator_
     y_pred = best_model.predict(X_test)

     mae = mean_absolute_error(y_test, y_pred)
     mse = mean_squared_error(y_test, y_pred)
     r_squared = r2_score(y_test, y_pred)

     print("Best Hyperparameters:", best_params)
```

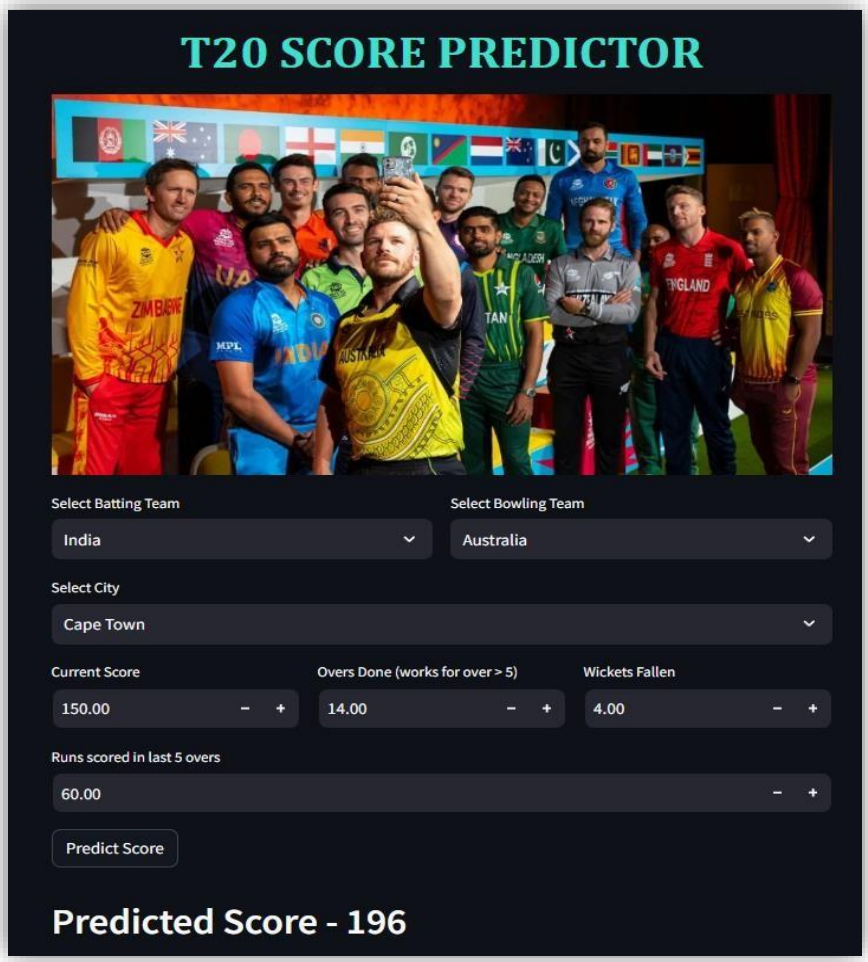


```
     }
     grid_search = GridSearchCV(estimator=xgb_model, param_grid=param_grid, scoring='neg_mean_squared_error', cv=5)
     grid_search.fit(X_train, y_train)

     best_params = grid_search.best_params_

     best_model = grid_search.best_estimator_
     y_pred = best_model.predict(X_test)

     mae = mean_absolute_error(y_test, y_pred)
     mse = mean_squared_error(y_test, y_pred)
     r_squared = r2_score(y_test, y_pred)

     print("Best Hyperparameters:", best_params)
     print("Mean Absolute Error:", mae)
     print("Mean Squared Error:", mse)
     print("R-squared:", r_squared)

Best Hyperparameters: {'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 300}
Mean Absolute Error: 2.443214278211217
Mean Squared Error: 14.961286125449627
R-squared: 0.9851439534152865
```

9.

RESULTS:


mudit14-t20-score-predictor.streamlit.app
Share
T20 SCORE PREDICTOR
Select Batting Team
India
Select Bowling Team
Australia
Select City
Cape Town
Current Score
150.00
Overs Done (works for over > 5)
14.00
Wickets Fallen
4.00
Runs scored in last 5 overs
60.00
Predict Score
Predicted Score - 196
Manage app


T20 SCORE PREDICTOR
Select Batting Team
India
Select Bowling Team
Australia
Select City
Cape Town
Current Score
150.00
Overs Done (works for over > 5)
14.00
Wickets Fallen
4.00
Runs scored in last 5 overs
60.00
Predict Score
Predicted Score - 196

10.

ADVANTAGES & DISADVANTAGES:

Advantages of T20 Score Prediction:

- Users can make informed decisions in fantasy cricket, sports betting, or simply enjoy matches with a deeper understanding of likely outcomes.
- T20 score prediction adds an extra layer of excitement and engagement for cricket enthusiasts by providing real-time insights and forecasts.
- Coaches and teams can use predictions to strategically plan their gameplay, considering opponent strengths and weaknesses.
- The model facilitates comprehensive statistical analysis, contributing to a better understanding of player and team performance in T20 matches.

Disadvantages of T20 Score Prediction:

- The fast-paced and unpredictable nature of T20 cricket poses a challenge for accurate predictions, as unexpected events can significantly influence match outcomes.
- Users may over rely on predictions, potentially diminishing the spontaneous and unpredictable aspects that make cricket exciting.
- Factors like weather conditions, player injuries, and last-minute changes are challenging to incorporate into prediction models, affecting the accuracy of forecasts.
- Developing and maintaining a sophisticated prediction model requires continuous effort, and the complexity increases with the dynamic nature of T20 cricket.

11. CONCLUSION:

T20 score prediction models offer valuable benefits such as informed decision-making, enhanced engagement, and strategic planning for cricket enthusiasts, teams, and coaches. However, the unpredictable nature of T20 cricket, external factors, and the potential for overreliance on predictions pose challenges. Striking a balance between accuracy and adaptability is crucial for the success of such models. While T20 score prediction adds a data-driven dimension to the sport, it is essential to appreciate cricket's inherent unpredictability and maintain a holistic perspective on the game. Continuous refinement and consideration of user feedback are key to improving the effectiveness and reliability of T20 score prediction systems.

12. FUTURE SCOPE:

The future scope of T20 score prediction lies in advancements in machine learning algorithms, incorporation of more granular player and team data, and integration of real-

time contextual information. Enhanced predictive analytics, improved model adaptability to dynamic match scenarios, and potential collaborations with cricketing bodies for more comprehensive data access could further refine and broaden the applicability of T20 score prediction models. Additionally, advancements in technology, such as the use of artificial intelligence and advanced statistical models, may contribute to increased accuracy and reliability in predicting T20 cricket match outcomes.

13. APPENDIX:

1. Data Source:

The T20 cricket match data used in this project was sourced from Kaggle. The dataset is available at [Kaggle T20 Cricket Dataset](https://www.kaggle.com/example/dataset).

2. Data Preprocessing:

2.1 Data Cleaning:

- Removed duplicate entries.

- Handled missing values in relevant columns.

- Corrected inconsistent data formats.

2.2 Feature Engineering:

- Extracted relevant features such as team performance, ground stats, and match conditions.

3. XGBoost Model:

3.1 Model Selection:

XGBoost (Extreme Gradient Boosting) was chosen as the predictive modelling algorithm due to its efficiency and effectiveness in handling structured data.

3.2 Hyperparameter Tuning:

- Performed hyperparameter tuning using techniques such as grid search.

- Optimized parameters for better prediction performance.

4. Model Evaluation:

4.1 Performance Metrics:

- Evaluated the model using the following metrics:

- R-squared (R^2)

- Mean Squared Error (MSE)

- Mean Absolute Error (MAE)

- Confusion matrix and ROC curves were also analysed.

4.2 Results:

- The detailed results of the model evaluation, including tables and visualizations, are mentioned in the project document.

5. Streamlit Web Application:

5.1 Application Development:

- Developed a Streamlit web application to showcase the T20 score prediction model.

- Integrated user-friendly interfaces for input and result display.

5.2 Deployment:

- Deployed the Streamlit app on a web server for public access.

- Provided the app's URL: [https://mudit14-t20-score-predictor.streamlit.app/]

6. Repository Information:

- All project documents, including data preprocessing steps, model training details, and Streamlit app development, are available in the Git repository.

- Repository URL: [https://github.com/smartinternz02/SI-GuidedProject-6127431698932186]

7. Dependencies:

- Listed the key Python libraries and their versions used in the project.

8. Conclusion and Future Work:

- Summarized the key findings and conclusions of the T20 score prediction project.

- Discussed potential enhancements or future directions for the project.

9. References:

- https://www.researchgate.net/publication/319335442_Dynamic_Winner_Prediction_in_Twenty20_Cricket_Based_on_Relative_Team_Strengths.
- https://ieeexplore.ieee.org/document/9807929
- https://www.researchgate.net/publication/353718778_An_Analytical_Model_for_Prediction_of_Upcoming_ICC_T20_World_Cup_2021_Using_Classification_Algorithms
- https://ieeexplore.ieee.org/abstract/document/9318077
- https://ieeexplore.ieee.org/document/8820235
- https://www.irjet.net/archives/V4/i3/IRJET-V4I3269.pdf
- https://ieeexplore.ieee.org/document/8966691
- https://ieeexplore.ieee.org/document/9642558

10. Acknowledgments:

- Our sincere appreciation goes to SmartBridge for giving us the opportunity to work on this engaging project.

- We would like to express our gratitude to our dedicated team members:

- Pratip Sahoo

- Sounak Maity

- A special acknowledgment to our team leader, Mudit, for his valuable guidance and leadership throughout the project.