In [294]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

In [295]:

```
df = pd.read_csv("ipl.csv")
df.head()
```

Out[295]:

	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	resu
0	335982	Bangalore	18-04-2008	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	Kolkata Knight Riders	runs	
1	335983	Chandigarh	19-04-2008	MEK Hussey	Punjab Cricket Association Stadium, Mohali	0	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Chennai Super Kings	runs	
2	335984	Delhi	19-04-2008	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	Delhi Daredevils	wickets	
3	335985	Mumbai	20-04-2008	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore	wickets	
4	335986	Kolkata	20-04-2008	DJ Hussey	Eden Gardens	0	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	Kolkata Knight Riders	wickets	

In [296]:

```
df.columns
```

Out[296]:

```
Index(['id', 'city', 'date', 'player_of_match', 'venue', 'neutral_venue',
       'team1', 'team2', 'toss_winner', 'toss_decision', 'winner', 'result',
       'result_margin', 'eliminator', 'method', 'umpire1', 'umpire2'],
      dtype='object')
```

In [297]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 816 entries, 0 to 815
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   id               816 non-null    int64  
 1   city             803 non-null    object 
 2   date             816 non-null    object 
 3   player_of_match  812 non-null    object 
 4   venue            816 non-null    object 
 5   neutral_venue    816 non-null    int64  
 6   team1            816 non-null    object 
 7   team2            816 non-null    object 
 8   toss_winner      816 non-null    object 
 9   toss_decision    816 non-null    object 
 10  winner           812 non-null    object 
 11  result           812 non-null    object 
 12  result_margin    799 non-null    float64
 13  eliminator       812 non-null    object 
 14  method           19 non-null     object 
 15  umpire1          816 non-null    object 
 16  umpire2          816 non-null    object 
dtypes: float64(1), int64(2), object(14)
memory usage: 108.5+ KB
```

In [298]:

```
ipl = df.drop(['date','neutral_venue','eliminator','method','umpire1','umpire2','player_of_match','result','result_margin'],axis=1)
ipl.head()
```

Out[298]:

	id	city	venue	team1	team2	toss_winner	toss_decision	winner
0	335982	Bangalore	M Chinnaswamy Stadium	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	Kolkata Knight Riders
1	335983	Chandigarh	Punjab Cricket Association Stadium, Mohali	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Chennai Super Kings
2	335984	Delhi	Feroz Shah Kotla	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	Delhi Daredevils
3	335985	Mumbai	Wankhede Stadium	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore
4	335986	Kolkata	Eden Gardens	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	Kolkata Knight Riders

In [299]:

```
ipl.dropna(inplace=True)
```

In [300]:

```
ipl.isnull().sum()
```

Out[300]:

```
id              0
city            0
venue           0
team1           0
team2           0
toss_winner     0
toss_decision   0
winner          0
dtype: int64
```

In [301]:

```
#for Delhi Capitals
ipl['team1']=ipl['team1'].str.replace('Delhi Daredevils','Delhi Capitals')
ipl['team2']=ipl['team2'].str.replace('Delhi Daredevils','Delhi Capitals')
ipl['toss_winner']=ipl['toss_winner'].str.replace('Delhi Daredevils','Delhi Capitals')
ipl['winner']=ipl['winner'].str.replace('Delhi Daredevils','Delhi Capitals')
```

In [302]:

```
#for sunrisers Hyderabad
ipl['team1']=ipl['team1'].str.replace('Deccan Chargers','Sunrisers Hyderabad')
ipl['team2']=ipl['team2'].str.replace('Deccan Chargers','Sunrisers Hyderabad')
ipl['toss_winner']=ipl['toss_winner'].str.replace('Deccan Chargers','Sunrisers Hyderabad')
ipl['winner']=ipl['winner'].str.replace('Deccan Chargers','Sunrisers Hyderabad')
```

In [303]:

```
# only keep current team which are present
consistent_team = ['Kolkata Knight Riders', 'Chennai Super Kings', 'Rajasthan Royals',
       'Mumbai Indians','Kings XI Punjab',
       'Royal Challengers Bangalore', 'Delhi Capitals','Sunrisers Hyderabad']
```

In [304]:

```
# filtering based on consistency
ipl = ipl[(ipl['team1'].isin(consistent_team)) & (ipl['team2'].isin(consistent_team)) & (ipl['toss_winner'].isin(consistent_team)) & (ipl
```

In [305]:

```
ipl['team1'].unique()
```

Out[305]:

```
array(['Royal Challengers Bangalore', 'Kings XI Punjab', 'Delhi Capitals',
       'Mumbai Indians', 'Kolkata Knight Riders', 'Rajasthan Royals',
       'Sunrisers Hyderabad', 'Chennai Super Kings'], dtype=object)
```

In [306]:

```
ipl.shape
```

Out[306]:

```
(685, 8)
```

In [307]:

```
ipl.head()
```

Out[307]:

	id	city	venue	team1	team2	toss_winner	toss_decision	winner
0	335982	Bangalore	M Chinnaswamy Stadium	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	Kolkata Knight Riders
1	335983	Chandigarh	Punjab Cricket Association Stadium, Mohali	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Chennai Super Kings
2	335984	Delhi	Feroz Shah Kotla	Delhi Capitals	Rajasthan Royals	Rajasthan Royals	bat	Delhi Capitals
3	335985	Mumbai	Wankhede Stadium	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore
4	335986	Kolkata	Eden Gardens	Kolkata Knight Riders	Sunrisers Hyderabad	Sunrisers Hyderabad	bat	Kolkata Knight Riders

In [308]:

```
# Convert categorical variables to numerical using LabelEncoder
label_encoder = LabelEncoder()
categorical_columns = ['city','venue','team1', 'team2', 'toss_winner', 'toss_decision', 'winner']
for col in categorical_columns:
    ipl[col] = label_encoder.fit_transform(ipl[col])
print(categorical_columns)
```

```
['city', 'venue', 'team1', 'team2', 'toss_winner', 'toss_decision', 'winner']
```

In [288]:

```
# Separate features (X) and target (y)
X = ipl.drop(columns=['winner'])
y = ipl['winner']

# Split the data into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15, random_state=42)

# Step 2: Model selection (Random Forest Classifier)
model = RandomForestClassifier(random_state=42)

# Step 3: Model training
model.fit(X_train, y_train)

# Step 4: Model evaluation
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy of the model:", accuracy)
```

```
Accuracy of the model: 0.823076923076923
```

In [292]:

```
import pickle
file_name = 'ipl_winner_prediction_model.pkl'
pickle.dump(model , open(file_name,'wb'))
```

In [293]:

```
# Save the trained model and label encoder as a dictionary
import joblib

model_and_encoder = {
    'model': model,
    'label_encoder': label_encoder
}

joblib.dump(model_and_encoder, 'ipl_winner_prediction_model.pkl')
```

Out[293]:

```
['ipl_winner_prediction_model.pkl']
```

In [ ]:

```

```