A PROJECT REPORT ON

"IPL WINNING TEAM PREDICTION"

PRAMOD D

TLS21A1777

INDEX

ABSTRACT

The IPL (Indian Premier League) is one of the most viewed cricket league in the world. With a perpetual increase in the popularity and advertising associated with it, forecasting the IPL matches is becoming a need for the advertisers and the sponsors. This paper is centered on the implementation of machine learning to foretell the winner of an IPL match. The cricket in the T-20 format is highly unpredictable - many features contribute to the result of a cricket match, and each attribute feature has a weighted impact on the outcome of a game. In this paper, first, a meaningful dataset through data mining was defined; next, essential features using various methods like feature engineering and Analytic Hierarchy Process were derived. Besides, a key issue on data symmetry and the inability of models to handle it was identified, which extends to all types of classification models that compare two or more classes using similar features for both the classes. This concept in the paper is termed as model ambiguity that occurs due to the model's asymmetric nature. Alongside, different machine learning classification algorithms Linear Regression ,k- Nearest Neighbor, Random Forest, Logistic Regression were adopted to train the models for predicting the winner.

INTRODUCTION

The aim of this project is to predict the winning teams of IPL match 2021 on a given a set of features as inputs. Input variables are team 1 , team 2 ,city And the output variable is winner. We are dealing only with winner. We have winner between two team. The higher the value the better is winner. In this project we will treat each team separately and their aim is to be able and find the winner of the match that work well for new unseen data. These are the classifiers.

Our IPL dataset contains match by match records from the first match played in the 2008 season till the complete 2020 season.

In this project we are explaining the steps we followed to predict winner of IPL 2021, the study of data to extract knowledge and insights from the data and apply knowledge and actionable insights. In this project, we will work on IPL Data Analysis and Visualization Project using Python where we will explore interesting insights from the data of IPL matches like most run by a player, most wicket taken by a player, and much more from IPL season 2008-2020. We will initially perform simple statistical analysis and then slowly build to more advanced analysis.

TASKS

- DATA ACQUISITION AND CLEANING
- DATA VISUALIZATION
- DATA MODELLING
- TESTING
- COMPARISON AND MEASUREMENT

DATA ACQUISITION AND CLEANING

We are using Two datasets in this project, One dataset containing 2008 -2020 IPL Matches used for training the model and another dataset containing 2021 IPL Matches. The inputs are team names information, such as team1, team2 , city and the output is based on the team which has high winning probability.
The dataset provides the team names and , Season ,City ,Date ,Team1 ,Team2 ,Toss_winner ,Toss_decision ,Result, dl applied Win_by_wicket, Win_by_runs, Player_of_match, Umpire1, Umpire2 etc.

Input variables based on given data set:

- Team1
- Team2
- City

Output variable based on sensory data:

- Winner

DATA CLEANING

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset. But it is crucial to establish a template for data cleaning process. We applied Data cleaning on some unwanted columns. But before data cleaning some explorations and data visualization were applied on the data set using this it gave us an idea on how to deal with missing values and also extreme values.

We analysed the dataset and decided to remove some of the unwanted columns which are not useful for our prediction. Such columns are :

- date
- toss_winner
- toss_decision
- result
- dl_applied
- win_by_wickets
- win_by_runs
- player_of match
- umpire1
- umpire2
- umpire3

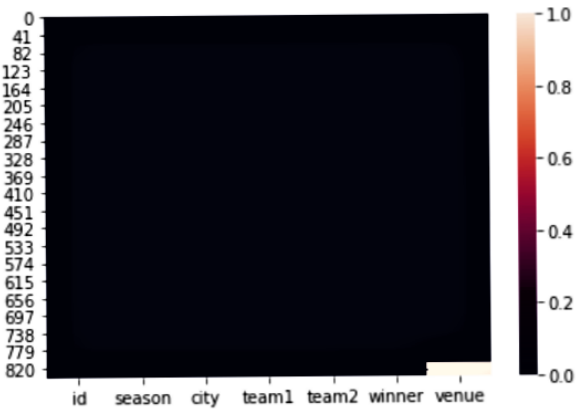
OUTPUT

In [312]:
```
train_data = train_data.drop(columns=['date','umpire1','umpire2','toss_winner','win_by_runs','win_by_wickets','toss_decision','to
print(train_data)
sns.heatmap(train_data.isnull())
plt.show()
```

```
          id  season         city                 team1  \
0         60    2008    Bangalore  Kolkata Knight Riders
1         61    2008   Chandigarh    Chennai Super Kings
2         62    2008        Delhi       Rajasthan Royals
3         63    2008       Mumbai         Mumbai Indians
4         64    2008      Kolkata    Sunrisers Hyderabad
..       ...     ...          ...                    ...
838  1216502    2021    Abu Dhabi                    RCB
839  1216506    2021  Dubai (DSC)                    CSK
840  1216530    2021      Sharjah                    KKR
841  1216495    2021    Abu Dhabi                    SRH
842  1216505    2021  Dubai (DSC)                     DC

                         team2                    winner  \
0    Royal Challengers Bangalore       Kolkata Knight Riders
1                Kings XI Punjab         Chennai Super Kings
2                 Delhi Capitals              Delhi Capitals
3    Royal Challengers Bangalore  Royal Challengers Bangalore
4          Kolkata Knight Riders       Kolkata Knight Riders
..                           ...                        ...
838                         SRH                       Draw
839                        KXIP                       Draw
840                          RR                       Draw
841                          MI                       Draw
842                         RCB                       Draw

                                    venue
0                     M Chinnaswamy Stadium
1    Punjab Cricket Association Stadium, Mohali
2                          Feroz Shah Kotla
3                          Wankhede Stadium
4                              Eden Gardens
..                                      ...
838                                    NaN
839                                    NaN
840                                    NaN
841                                    NaN
842                                    NaN
```



DATA VISUALIZATION

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

Data Visualization provides a clear understanding of the data. The information acquired in this section may be used during the modelling phase.

Although UCI provides the dataset ready for analysis, it's good practice to check it for possible issues we may encounter in the future.

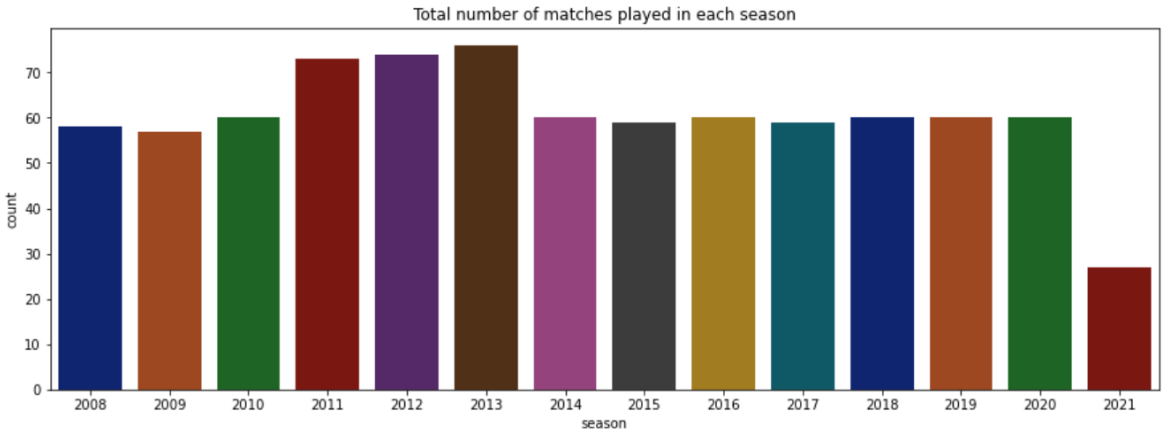
Now, with a basic understanding of the attributes let us now start our project of data analysis and visualization of the IPL dataset with Python. We will initially perform simple statistical analysis and then slowly build to more advanced analysis..

1.Bar graph :

A bar graph is a chart that plots data using rectangular bars or columns(called bins) that represents total amount of observations in data for the category .

- Vertical bar graph:
  1. Season vs count

```
In [307]: plt.subplots(figsize = (15,5))
          sns.countplot(x = 'season' , data = train_data, palette='dark')
          plt.title('Total number of matches played in each season')
          plt.show()
```



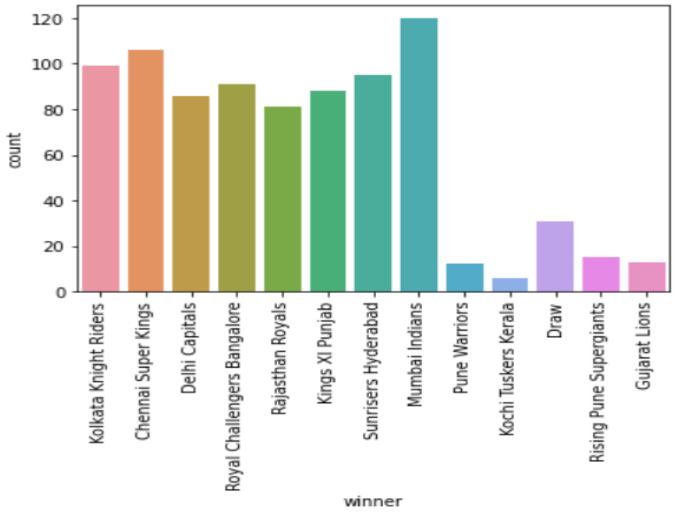
In this graph ,X axis represents seasons and y axis represents count i.e total number of matches played in each season. In 2008 approximately 58% matches played .In 2011,the count increases to 72%. There is no much variation in the number of matches played from 2014 to till 2021, Its in constant rate .

2. Winner vs count:

```
In [311]: sns.countplot(train_data['winner'])
          plt.xticks(rotation = 90)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
rg: x. From version 0.12, the only valid positional argument will be `data`
yword will result in an error or misinterpretation.
  warnings.warn(
```

```
Out[311]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]),
 [Text(0, 0, 'Kolkata Knight Riders'),
  Text(1, 0, 'Chennai Super Kings'),
  Text(2, 0, 'Delhi Capitals'),
  Text(3, 0, 'Royal Challengers Bangalore'),
  Text(4, 0, 'Rajasthan Royals'),
  Text(5, 0, 'Kings XI Punjab'),
  Text(6, 0, 'Sunrisers Hyderabad'),
  Text(7, 0, 'Mumbai Indians'),
  Text(8, 0, 'Pune Warriors'),
  Text(9, 0, 'Kochi Tuskers Kerala'),
  Text(10, 0, 'Draw'),
  Text(11, 0, 'Rising Pune Supergiants'),
  Text(12, 0, 'Gujarat Lions')])
```

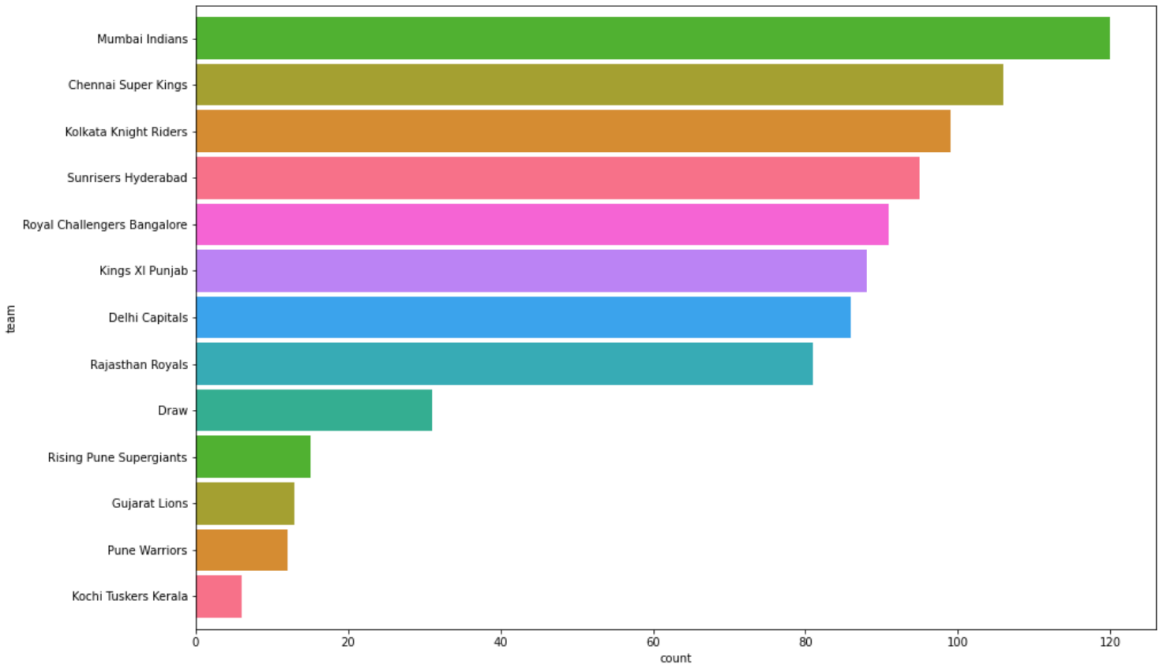


In this graph, X axis represents winners and y axis represents count.Team Mumbai Indians won maximum matches about count 120. Kolkata knight riders won matches about count approximately 100. Least number of winning count about 15 by Team Gujarat Lions.

- Horizontal bar graph:

A horizontal bar chart is a type of bar graph that represents data variables using proportional horizontal bars.Here,the winning count (winners) are placed on the vertical axis of graph while the team names is placed on horizontal axis of graph.

```
In [308]: plt.subplots(figsize=(15,10))
          ax = train_data['winner'].value_counts().sort_values(ascending=True).plot.barh(width=.9,color=sns.color_palette("husl", 9))
          ax.set_xlabel('count')
          ax.set_ylabel('team')
          plt.show()
```



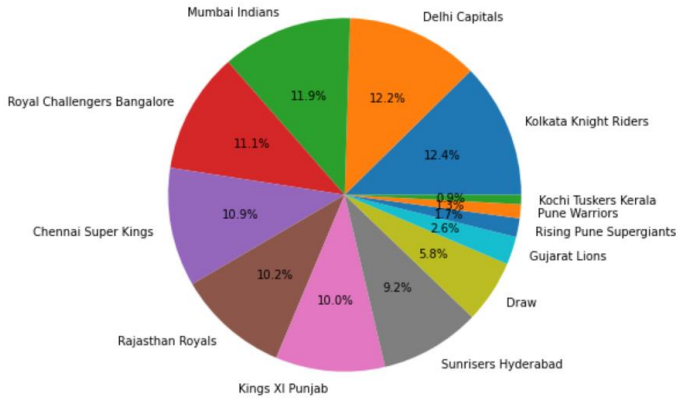
In this graph, there are 13 teams in which Team Mumbai Indians has won maximum matches and the count is almost 120 .After Mumbai Indians next comes Team Chennai Super Kings has won approximately 105 matches. Least matches won by Team Kochi Tuskers kerala about of count 5-6 .

2. Pie Chart :

A pie chart (or a circle chart) is a circular statistical graphic,which is divided into slices to illustrate numerical proportion.In a pie chart,the arc length of each slice (and consequently its central angle and area),is proportional to number of matches won by batting second.

- Matches won by batting second:

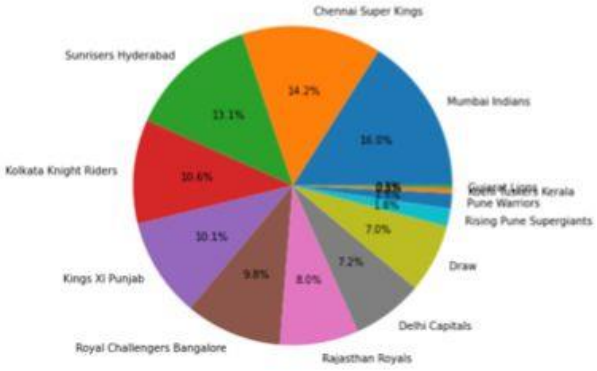
```
In [310]: #extracting those records where a team has won after batting second
batting_second=train_data[train_data['win_by_wickets']!=0]
#Making a pie chart for distribution of most wins after batting second
plt.figure(figsize=(7,7))
plt.pie(list(batting_second['winner'].value_counts()),labels=list(batting_second['winner'].value_counts().keys()),autopct='%0.1f%
plt.show()
```



This graph shows the results of a predictions(survey) in which number of teams won matches by batting second.Here,12.4% matches won by kolkata knight riders,and next Delhi capitals are of percentage 12.2 .Least prediction to win matches by second is kochi tuskers kerala is of 0.9%.

- Matches won by batting first:

```
In [309]: #Extracting the records where a team won batting first
          batting_first=train_data[train_data['win_by_runs']!=0]
          #Making a pie chart
          plt.figure(figsize=(7,7))
          plt.pie(list(batting_first['winner'].value_counts()),labels=list(batting_first['winner'].value_counts().keys()),autopct='%0.1f%%'
          plt.show()
```



This graph shows the results of survey in which number of matches won by Teams

by batting first .Here about 16% of matches won by Team Mumbai Indians, next comes to the race is Team Chennai Super Kings about 14.2% .Least number of matches won by Team Gujarat Lions by 0.5%.

DATA MODELLING

The process of modelling means training a machine learning algorithm to predict the labels from the features, tuning it for the business need, and validating it on holdout data.

Linear Model : We Firstly use Linear Regression to predict the winning team for provided two different teams. This model establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a *X +b. After applying the Linear Model the Prediction Accuracy we got was 29.628%. It seems the model is less accurate for prediction.

Logistic Model: Logistic Regression is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variables. In simple words, it predicts the probability of occurrence of an event by fitting data to a logistic function. Since, it predicts the probability, its output values lies between 0 and 1. We use this model to predict the winning team for the provided two different teams. After applying the Logistic Model the Prediction Accuracy we got was 23.48%. It seems the model is less accurate than Logistic regression for prediction.

KNN Model: It can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function. We use this model to predict the winning team for the provided two different teams. After applying the KNN Model the Prediction Accuracy we got was 61.56%. It seems the model is more accurate for prediction than Linear and Logistic regression models.

Random Forest:

Random forests improve prediction performance over classification trees by averaging multiple decision trees. The algorithm creates several random subsets of the original data, in this case the training set, and calculates the classification trees, then the final result is the average of all trees. The name random forest derives from the random process of splitting the data and creating many trees, or a forest. After applying the Random Forest Model the Prediction Accuracy we got was 80.66%. It seems the model is more accurate for prediction than Linear, Logistic regression and KNN models that we applied previously.

TESTING

We are using two datasets in this project, One dataset containing 2008 - 2020 IPL Matches used for training the model and another dataset containing only 2021 IPL Matches schedule. 2008-2020 IPL match dataset consists more than 800 matches which is used for training. 2021 2nd phase IPL match dataset nearly 27 matches which is used for testing using different classification and regression algorithms.

PYTHON CODE

We have imported the CSV dataset below with the help of pandas read_csv functions We can see the content of the dataset by using head() function.

```
In [302]: import numpy as np # linear algebra
          import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
          import os
          from sklearn.model_selection import train_test_split
          import seaborn as sns
          from sklearn.datasets import make_classification
          import matplotlib.pyplot as plt
          from sklearn import preprocessing
          from sklearn.model_selection import GridSearchCV
          from sklearn.preprocessing import StandardScaler
          from sklearn.linear_model import LinearRegression
          from sklearn.linear_model import LogisticRegression
          from sklearn.neighbors import KNeighborsClassifier
          from sklearn.ensemble import RandomForestClassifier

In [303]: train_data = pd.read_csv('E:/BACKUP/users-admin files/Downloads/Training Matches IPL 2008-2020.csv')
          train_data.head()

Out[303]:
```

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets	player_
0	60	2008	Bangalore	18/04/2008	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0.0	Kolkata Knight Riders	140.0	0.0	BB
1	61	2008	Chandigarh	19/04/2008	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	normal	0.0	Chennai Super Kings	33.0	0.0	ME
2	62	2008	Delhi	19/04/2008	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	normal	0.0	Delhi Daredevils	0.0	9.0	MF
3	63	2008	Mumbai	20/04/2008	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	normal	0.0	Royal Challengers Bangalore	0.0	5.0	M
4	64	2008	Kolkata	20/04/2008	Deccan Chargers	Kolkata Knight Riders	Deccan Chargers	bat	normal	0.0	Kolkata Knight Riders	0.0	5.0	I

```
In [304]: train_data.isnull().sum()
```

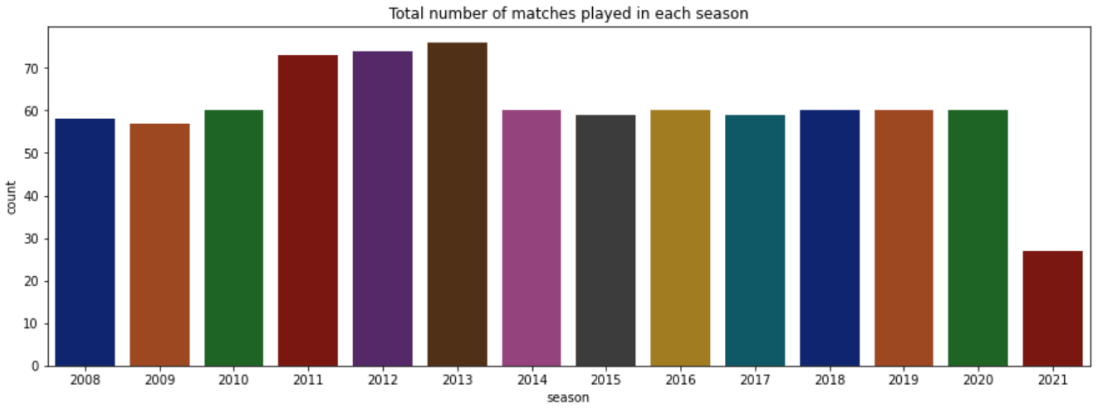
```
Out[304]: id                 0
          season             0
          city               7
          date               0
          team1              0
          team2              0
          toss_winner       27
          toss_decision     27
          result            27
          dl_applied        27
          winner            31
          win_by_runs       27
          win_by_wickets    27
          player_of_match   31
          venue             27
          umpire1           29
          umpire2           29
          dtype: int64
```

```
In [305]: train_data['city'].fillna('Abu Dhabi',inplace=True)
          train_data['winner'].fillna('Draw', inplace = True)
```

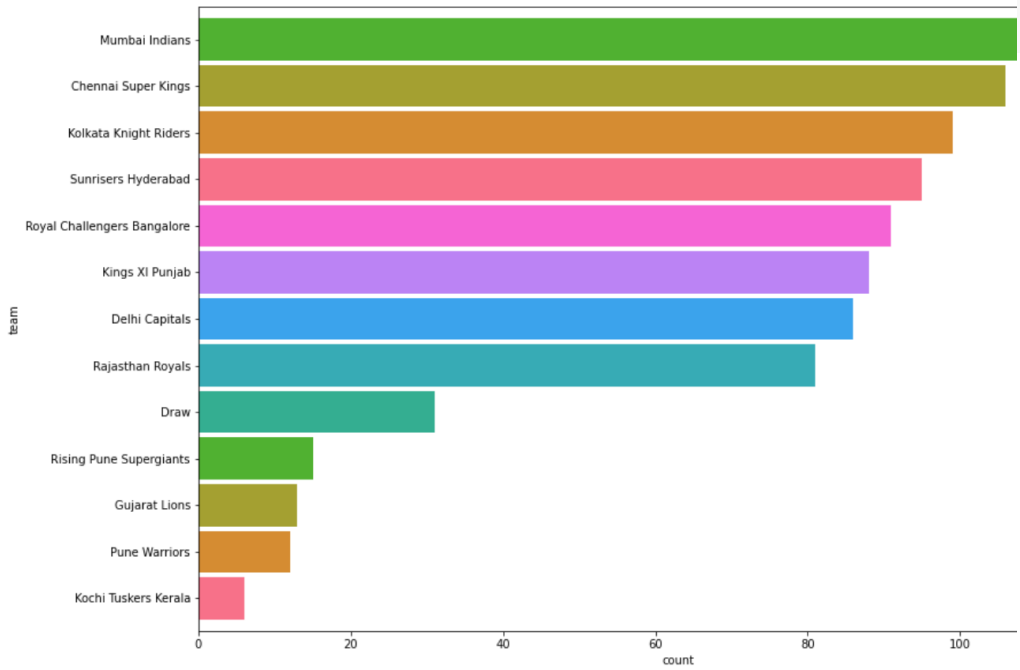
```
In [306]: #Both Rising Pue Supergiant and Rising Pune Supergiants represents same team similarly
          #Deccan Chargers and Sunrisers Hyderabad
          train_data.replace("Rising Pune Supergiant","Rising Pune Supergiants", inplace=True)
          train_data.replace('Deccan Chargers', 'Sunrisers Hyderabad', inplace=True)
          train_data.replace('Delhi Daredevils', 'Delhi Capitals', inplace=True)
```

OUTPUT:

```
In [307]: plt.subplots(figsize = (15,5))
          sns.countplot(x = 'season' , data = train_data, palette='dark')
          plt.title('Total number of matches played in each season')
          plt.show()
```

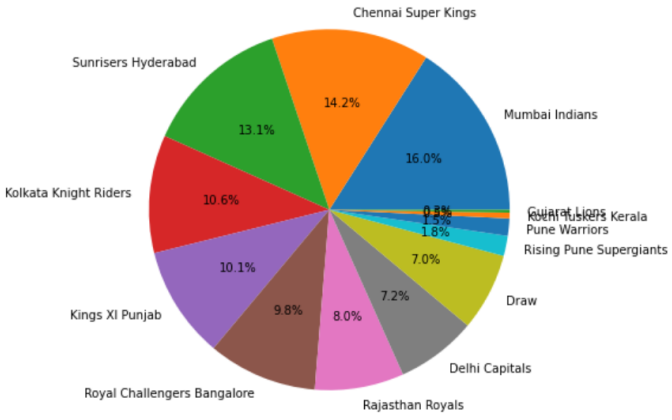


In [308]:
```
plt.subplots(figsize=(15,10))
ax = train_data['winner'].value_counts().sort_values(ascending=True).plot.barh(width=.9,color=sns.color_palette(
ax.set_xlabel('count')
ax.set_ylabel('team')
plt.show()
```

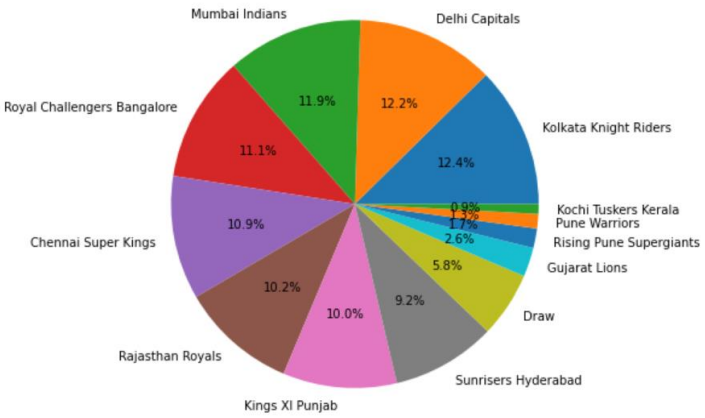


count

In [309]:
```
#Extracting the records where a team won batting first
batting_first=train_data[train_data['win_by_runs']!=0]
#Making a pie chart
plt.figure(figsize=(7,7))
plt.pie(list(batting_first['winner'].value_counts()),labels=list(batting_first['winner'].value_counts().keys()),autopct='%0.1f%%'
plt.show()
```



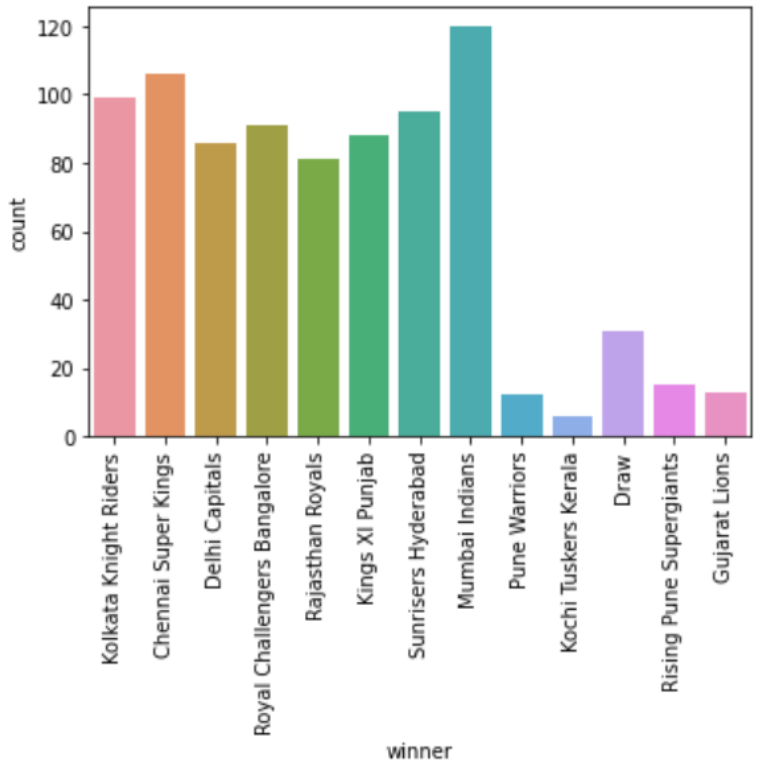
In [310]:
```
#extracting those records where a team has won after batting second
batting_second=train_data[train_data['win_by_wickets']!=0]
#Making a pie chart for distribution of most wins after batting second
plt.figure(figsize=(7,7))
plt.pie(list(batting_second['winner'].value_counts()),labels=list(batting_second['winner'].value_counts().keys()),autopct='%0.1f%
plt.show()
```



In [311]:
```
sns.countplot(train_data['winner'])
plt.xticks(rotation = 90)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
rg: x. From version 0.12, the only valid positional argument will be `dat
yword will result in an error or misinterpretation.
  warnings.warn(
```

Out[311]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]),
 [Text(0, 0, 'Kolkata Knight Riders'),
  Text(1, 0, 'Chennai Super Kings'),
  Text(2, 0, 'Delhi Capitals'),
  Text(3, 0, 'Royal Challengers Bangalore'),
  Text(4, 0, 'Rajasthan Royals'),
  Text(5, 0, 'Kings XI Punjab'),
  Text(6, 0, 'Sunrisers Hyderabad'),
  Text(7, 0, 'Mumbai Indians'),
  Text(8, 0, 'Pune Warriors'),
  Text(9, 0, 'Kochi Tuskers Kerala'),
  Text(10, 0, 'Draw'),
  Text(11, 0, 'Rising Pune Supergiants'),
  Text(12, 0, 'Gujarat Lions')])
```

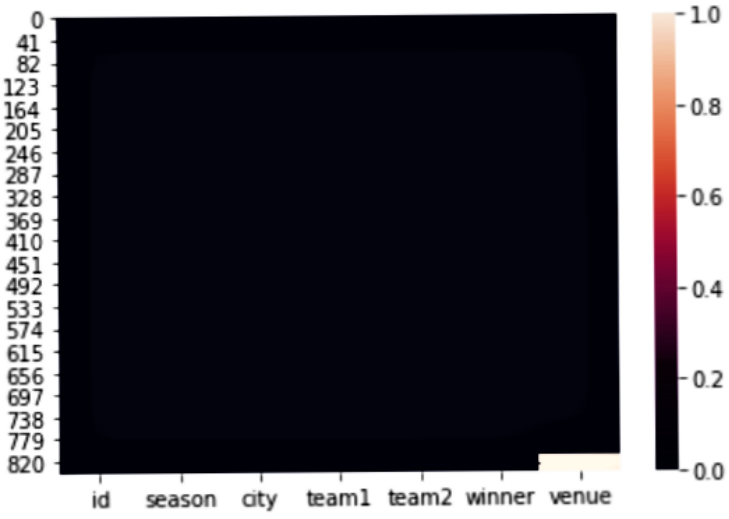


In [312]:
```
train_data = train_data.drop(columns=['date','umpire1','umpire2','toss_winner','win_by_runs','win_by_wickets','toss_decision','to
print(train_data)
sns.heatmap(train_data.isnull())
plt.show()
```

```
          id  season         city                  team1  \
0         60    2008    Bangalore  Kolkata Knight Riders
1         61    2008   Chandigarh    Chennai Super Kings
2         62    2008        Delhi       Rajasthan Royals
3         63    2008       Mumbai         Mumbai Indians
4         64    2008      Kolkata    Sunrisers Hyderabad
..       ...     ...          ...                    ...
838  1216502    2021    Abu Dhabi                    RCB
839  1216506    2021  Dubai (DSC)                    CSK
840  1216530    2021      Sharjah                    KKR
841  1216495    2021    Abu Dhabi                    SRH
842  1216505    2021  Dubai (DSC)                     DC

                         team2                     winner  \
0    Royal Challengers Bangalore       Kolkata Knight Riders
1                Kings XI Punjab         Chennai Super Kings
2                 Delhi Capitals               Delhi Capitals
3    Royal Challengers Bangalore  Royal Challengers Bangalore
4          Kolkata Knight Riders       Kolkata Knight Riders
..                           ...                        ...
838                         SRH                       Draw
839                        KXIP                       Draw
840                          RR                       Draw
841                          MI                       Draw
842                         RCB                       Draw

                                  venue
0                  M Chinnaswamy Stadium
1    Punjab Cricket Association Stadium, Mohali
2                        Feroz Shah Kotla
3                        Wankhede Stadium
4                            Eden Gardens
..                                    ...
838                                  NaN
839                                  NaN
840                                  NaN
841                                  NaN
842                                  NaN
```



In [313]:
```
train_data.replace({"Mumbai Indians":"MI", "Delhi Capitals":"DC",
                "Sunrisers Hyderabad":"SRH", "Rajasthan Royals":"RR",
                "Kolkata Knight Riders":"KKR", "Kings XI Punjab":"KXIP",
                "Chennai Super Kings":"CSK", "Royal Challengers Bangalore":"RCB",
               "Kochi Tuskers Kerala":"KTK", "Rising Pune Supergiants":"RPS",
               "Gujarat Lions":"GL", "Pune Warriors":"PW"}, inplace=True)


encode = {'team1': {'KKR':1,'CSK':2,'RR':3,'MI':4,'SRH':5,'KXIP':6,'RCB':7,'DC':8,'KTK':9,'RPS':10,'GL':11,'PW':12},
          'team2': {'KKR':1,'CSK':2,'RR':3,'MI':4,'SRH':5,'KXIP':6,'RCB':7,'DC':8,'KTK':9,'RPS':10,'GL':11,'PW':12},
           'winner': {'KKR':1,'CSK':2,'RR':3,'MI':4,'SRH':5,'KXIP':6,'RCB':7,'DC':8,'KTK':9,'RPS':10,'GL':11,'PW':12,'Draw':13}}
train_data.replace(encode, inplace=True)
train_data.head(5)
```

Out[313]:

	id	season	city	team1	team2	winner	venue
0	60	2008	Bangalore	1	7	1	M Chinnaswamy Stadium
1	61	2008	Chandigarh	2	6	2	Punjab Cricket Association Stadium, Mohali
2	62	2008	Delhi	3	8	8	Feroz Shah Kotla
3	63	2008	Mumbai	4	7	7	Wankhede Stadium
4	64	2008	Kolkata	5	1	1	Eden Gardens

In [314]:
```
dicVal = encode['winner']
train = train_data[['team1','team2','city','winner']]
train.head(6)
```

Out[314]:

	team1	team2	city	winner
0	1	7	Bangalore	1
1	2	6	Chandigarh	2
2	3	8	Delhi	8
3	4	7	Mumbai	7
4	5	1	Kolkata	1
5	6	3	Jaipur	3

In [315]:
```
df = pd.DataFrame(train)
var_mod = ['city']
le = preprocessing.LabelEncoder()
for i in var_mod:
    df[i] = le.fit_transform(df[i])
df.dtypes
```

Out[315]:
```
team1     int64
team2     int64
city      int32
winner    int64
dtype: object
```

Applying Linear, Logistic ,KNN and Random Forest algorithms and measuring accuracy.

In [316]:
```
X = df[['team1', 'team2','city']]
y = df[['winner']]
sc = preprocessing.StandardScaler()
X = sc.fit_transform(X)
```

In [317]:
```
linear_model= LinearRegression()
linear_model.fit(X,y)
print("Linear Regression accuracy: ",(linear_model.score(X,y))*100)
logistic_model = LogisticRegression()
logistic_model.fit(X,y)
print("Logistic Regression accuracy: ",(logistic_model.score(X,y))*100)
Random_model = RandomForestClassifier()
Random_model.fit(X,y)
print("Random Forest accuracy: ", (Random_model.score(X,y))*100)
knn_model = KNeighborsClassifier()
knn_model.fit(X,y)
print("KNeighbor Classifier accuracy", (knn_model.score(X,y))*100)
```

```
Linear Regression accuracy:  29.62801158731574
Logistic Regression accuracy:  23.487544483985765
```

```
Random Forest accuracy:  80.66429418742585
KNeighbor Classifier accuracy 61.56583629893239
```

Testing the model with 2021 IPL matches Dataset

```
test_data = pd.read_csv('E:/BACKUP/users-admin files/OneDrive/Desktop/Testset Matches IPL 2021.csv')

encode = {'team1': {'KKR':1,'CSK':2,'RR':3,'MI':4,'SRH':5,'KXIP':6,'RCB':7,'DC':8,'KTK':9,'RPS':10,'GL':11,'PW':12},
          'team2': {'KKR':1,'CSK':2,'RR':3,'MI':4,'SRH':5,'KXIP':6,'RCB':7,'DC':8,'KTK':9,'RPS':10,'GL':11,'PW':12}}
test_data.replace(encode,inplace=True)
var_mod = ['city']
le = preprocessing.LabelEncoder()
for i in var_mod:
    test_data[i] = le.fit_transform(test_data[i].astype(str))
```

```
test_X = test_data[['team1','team2','city']]
test_X = sc.fit_transform(test_X)
y_predict = Random_model.predict(test_X)
newlist = list()
for i in y_predict:
    newlist.append(list(dicVal.keys())[list(dicVal.values()).index(i)])
test_data['winner'] = newlist

#Decoding Team Names
for i in range(27):
    test_data['team1'][i]=(list(dicVal.keys())[list(dicVal.values()).index(test_data['team1'][i])])
    test_data['team2'][i]=(list(dicVal.keys())[list(dicVal.values()).index(test_data['team2'][i])])
test_data.head()
```

PREDICTION :

IPL 2021 2nd phase Matches Winner team prediction:

Out[22]:

	id	season	city	date	team1	team2	venue	winner
0	1216492	2021	1	9/19/2021	MI	CSK	Sheikh Zayed Stadium, Abu Dhabi	MI
1	1216494	2021	0	9/20/2021	KKR	RCB	Sheikh Zayed Stadium, Abu Dhabi	KKR
2	1216537	2021	1	9/21/2021	KXIP	RR	Sheikh Zayed Stadium, Abu Dhabi	KXIP
3	1216532	2021	1	9/22/2021	DC	SRH	Sheikh Zayed Stadium, Abu Dhabi	KXIP
4	1216508	2021	0	9/23/2021	KKR	MI	Sheikh Zayed Stadium, Abu Dhabi	SRH
5	1216544	2021	2	9/24/2021	RCB	CSK	Dubai International Cricket Stadium	CSK
6	1216543	2021	0	9/25/2021	DC	RR	Dubai International Cricket Stadium	RCB
7	1216542	2021	2	9/25/2021	SRH	KXIP	Dubai International Cricket Stadium	SRH
8	1216536	2021	0	9/26/2021	CSK	KKR	Dubai International Cricket Stadium	Draw
9	1216547	2021	1	9/26/2021	RCB	MI	Dubai International Cricket Stadium	SRH
10	1216507	2021	1	9/27/2021	SRH	RR	Dubai International Cricket Stadium	MI
11	1216497	2021	2	9/28/2021	KKR	DC	Sheikh Zayed Stadium, Abu Dhabi	KKR
12	1216503	2021	0	9/28/2021	KXIP	MI	Sheikh Zayed Stadium, Abu Dhabi	RCB
13	1216514	2021	1	9/29/2021	RCB	RR	Sheikh Zayed Stadium, Abu Dhabi	Draw
14	1216528	2021	2	9/30/2021	SRH	CSK	Dubai International Cricket Stadium	CSK
15	1216523	2021	1	10/01/2021	KXIP	KKR	Sheikh Zayed Stadium, Abu Dhabi	KKR
16	1216529	2021	2	10/02/2021	MI	DC	Sheikh Zayed Stadium, Abu Dhabi	RPS
17	1216533	2021	0	10/02/2021	CSK	RR	Sheikh Zayed Stadium, Abu Dhabi	CSK
18	1216531	2021	2	10/03/2021	RCB	KXIP	Sharjah Cricket Stadium	DC
19	1216512	2021	1	10/03/2021	SRH	KKR	Sheikh Zayed Stadium, Abu Dhabi	KKR
20	1216509	2021	1	10/04/2021	DC	CSK	Sharjah Cricket Stadium	DC
21	1216541	2021	2	10/05/2021	RR	MI	Sheikh Zayed Stadium, Abu Dhabi	RR
22	1216502	2021	0	10/06/2021	RCB	SRH	Sharjah Cricket Stadium	RCB
23	1216506	2021	1	10/07/2021	CSK	KXIP	Sheikh Zayed Stadium, Abu Dhabi	CSK
24	1216530	2021	2	10/07/2021	KKR	RR	Dubai International Cricket Stadium	MI
25	1216495	2021	0	10/08/2021	SRH	MI	Sharjah Cricket Stadium	RCB
26	1216505	2021	1	10/08/2021	DC	RCB	Sheikh Zayed Stadium, Abu Dhabi	DC

Conclusion

This report uses two datasets of IPL Matches to predict the winning team based on performance of the team in the previous matches . Before starting the predictions, the report makes a brief summary of model evaluation, explaining the most common metrics used in categorical problems in machine learning. In data preparation, the training and testing sets are created and they will be used during the model building. In data exploration and visualization, we look for features that may provide good prediction results. The best predictors have low distribution overlapping area and low correlation among them. Modelling starts explaining very simple models and gradually moves to more complex ones. There's a brief explanation on some of the models used in this report. The results section presents the modelling results and discusses the model performance. The algorithms are used to predict winning team. However, Since IPL is a fixed event we can't manipulate more data for training or testing. Therefore the availability of the data is very limited and the predictions made by different models are not exactly accurate and true. We can see the many deviations in the resulted prediction.