

## ipl winning predictor

```
#import necessary libraries
```

```
import pandas as pd
```

```
from matplotlib import pyplot as plt
```

```
import numpy as np
```

```
import seaborn as sns
```

```
#read csv file
```

```
iplmatches = pd.read_csv('/content/IPL Matches 2008-2020.csv')
```

```
ipldelivery = pd.read_csv('/content/IPL Ball-by-Ball 2008-2020.csv')
```

```
#perform head and tail operation
```

```
iplmatches.head(5)
```

```
ipldelivery.head(5)
```

output:

	id	inning	over	ball	batsman	non_striker	bowler	batsman_runs	extra_runs	total_runs	non_boundary	is_wicket	dismissal_kind	player_dismissed
0	335982	1	6	5	RT Ponting	BB McCullum	AA Noffke	1	0	1	0	0	NaN	NaN
1	335982	1	6	6	BB McCullum	RT Ponting	AA Noffke	1	0	1	0	0	NaN	NaN
2	335982	1	7	1	BB McCullum	RT Ponting	Z Khan	0	0	0	0	0	NaN	NaN
3	335982	1	7	2	BB McCullum	RT Ponting	Z Khan	1	0	1	0	0	NaN	NaN
4	335982	1	7	3	RT Ponting	BB McCullum	Z Khan	1	0	1	0	0	NaN	NaN

```
#took the first five entries
```

```
iplmatches.describe()
```

output;

	id	neutral_venue	result_margin
count	8.160000e+02	816.000000	799.000000
mean	7.563496e+05	0.094363	17.321652
std	3.058943e+05	0.292512	22.068427
min	3.359820e+05	0.000000	1.000000
25%	5.012278e+05	0.000000	6.000000
50%	7.292980e+05	0.000000	8.000000
75%	1.082626e+06	0.000000	19.500000
max	1.237181e+06	1.000000	146.000000

#describe the constraints

ipldelivery.describe()

output:

	id	inning	over	ball	batsman_runs	extra_runs	total_runs	non_boundary	is_wicket
count	1.934680e+05	193468.000000	193468.000000	193468.000000	193468.000000	193468.000000	193468.000000	193468.000000	193468.000000
mean	7.567688e+05	1.482131	9.177027	3.615967	1.240231	0.066414	1.306645	0.000083	0.049078
std	3.060971e+05	0.499682	5.676848	1.807128	1.610867	0.339991	1.598802	0.009094	0.216031
min	3.359820e+05	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	5.012270e+05	1.000000	4.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	7.292970e+05	1.000000	9.000000	4.000000	1.000000	0.000000	1.000000	0.000000	0.000000
75%	1.082628e+06	2.000000	14.000000	5.000000	1.000000	0.000000	1.000000	0.000000	0.000000
max	1.237181e+06	2.000000	19.000000	9.000000	6.000000	7.000000	7.000000	1.000000	1.000000

#checking any datas isnull

iplmatches.isnull()

output:

id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin	eliminator	method	umpire1	umpire2
ie	False	False		False	False	False	False	False	False	False	False	False	False	True	False	False
ie	False	False		False	False	False	False	False	False	False	False	False	False	True	False	False
ie	False	False		False	False	False	False	False	False	False	False	False	False	True	False	False
ie	False	False		False	False	False	False	False	False	False	False	False	False	True	False	False
ie	False	False		False	False	False	False	False	False	False	False	False	False	True	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
ie	False	False		False	False	False	False	False	False	False	False	True	False	True	False	False
ie	False	False		False	False	False	False	False	False	False	False	False	False	True	False	False
ie	False	False		False	False	False	False	False	False	False	False	False	False	True	False	False
ie	False	False		False	False	False	False	False	False	False	False	False	False	True	False	False
ie	False	False		False	False	False	False	False	False	False	False	False	False	True	False	False

#first five entries of ipl matches

iplmatches.head(10)

output:

id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_margin
35982	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	Kolkata Knight Riders	runs	140.0
35983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	0	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Chennai Super Kings	runs	33.0
35984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	Delhi Daredevils	wickets	9.0
35985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore	wickets	5.0
35986	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	0	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	Kolkata Knight Riders	wickets	5.0
35987	Jaipur	2008-04-21	SR Watson	Sawai Mansingh Stadium	0	Rajasthan Royals	Kings XI Punjab	Kings XI Punjab	bat	Rajasthan Royals	wickets	6.0

VISUALIZATION:

Number of IPL matches won by each team:

```
plt.figure(figsize = (10,6))
```

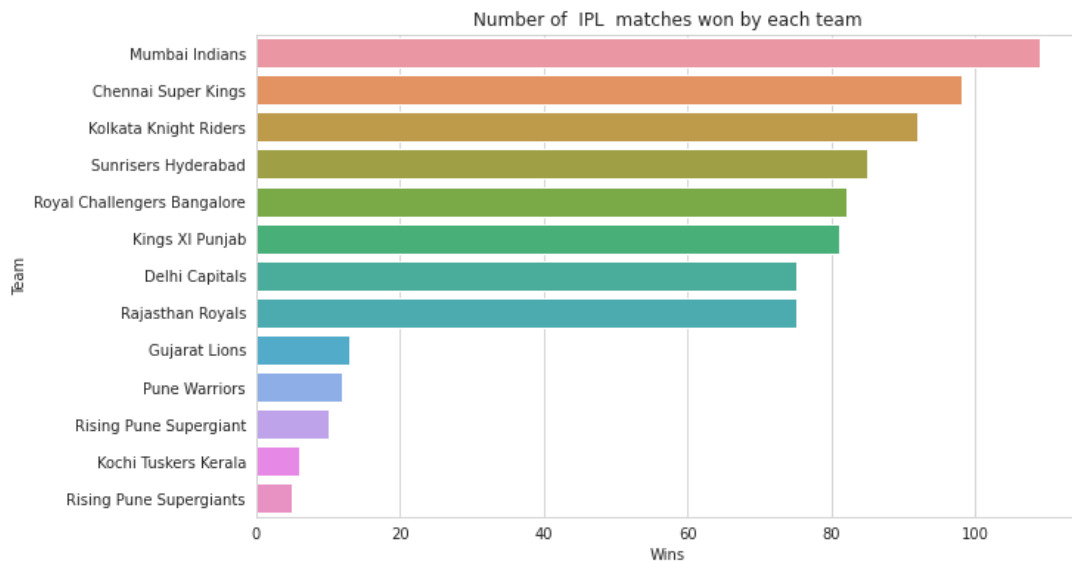
```
sns.countplot(y = 'winner',iplmatches = iplmatches,order= data['winner'].value_counts().index)
```

```
plt.xlabel('Wins')
```

```
plt.ylabel('Team')
```

```
plt.title('Number of IPL matches won by each team')
```

OUTPUT:



Total number of matches played in a different stadium

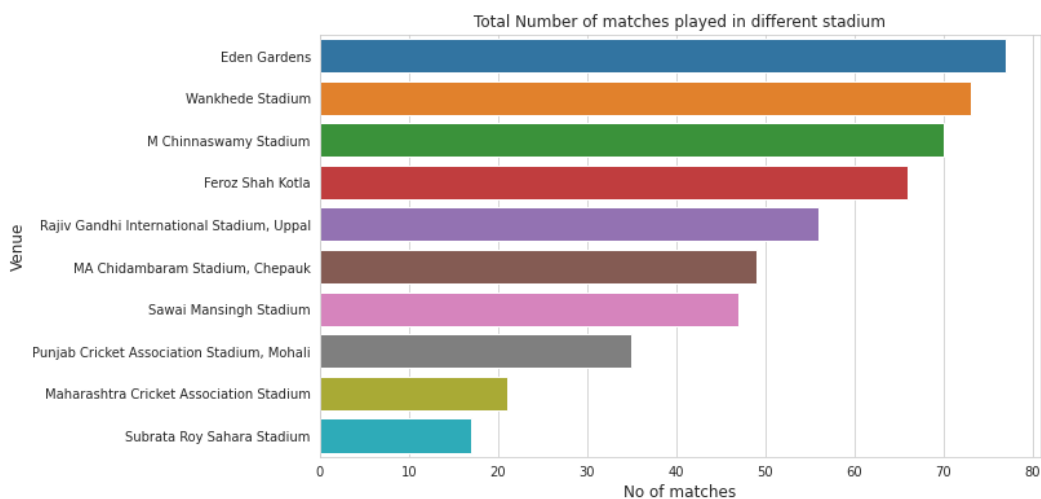
```
plt.figure(figsize = (10,6))
```

```
sns.countplot(y = 'venue',iplmatches = iplmatches,order = data['venue'].value_counts().iloc[:10].index)
```

```
plt.xlabel('No of matches',fontsize=12)
```

```
plt.ylabel('Venue',fontsize=12)
```

```
plt.title('Total Number of matches played in different stadium')
```



The decision was taken by the toss winning team.

```
plt.figure(figsize = (10,6))

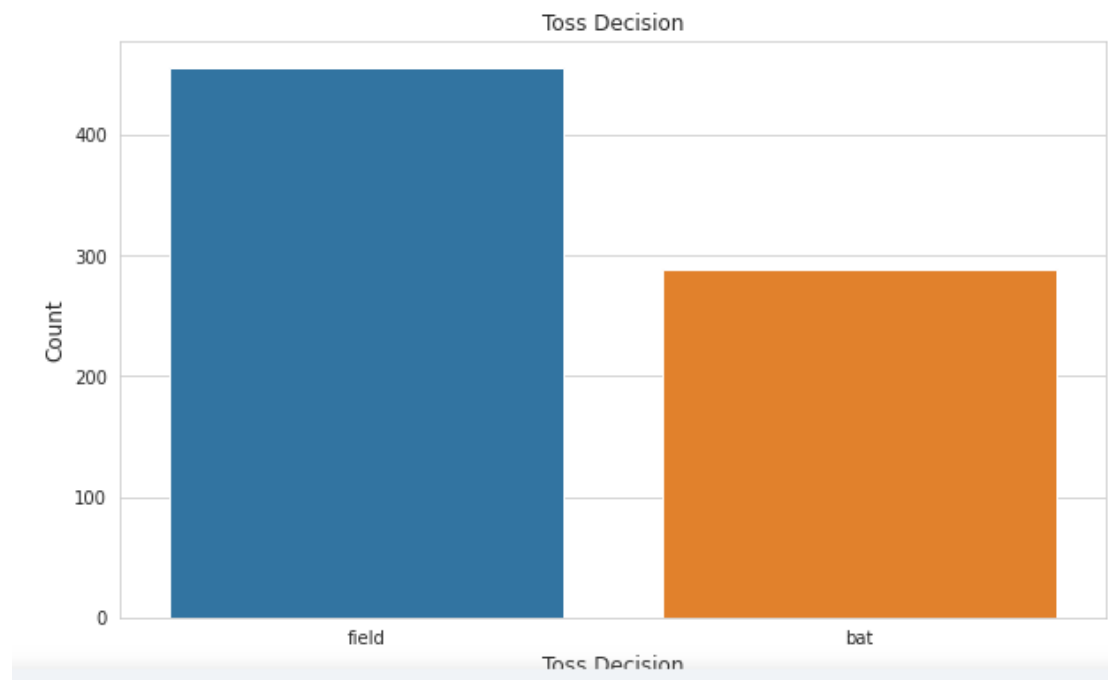
sns.countplot(x = "toss_decision", iplmatches=iplmatches)

plt.xlabel('Toss Decision',fontsize=12)

plt.ylabel('Count',fontsize=12)

plt.title('Toss Decision')
```

OUTPUT:



Now let's check the unique values presented in each feature

```
x = ["city", "toss_decision", "result", "dl_applied"]
```

```
for i in x:
```

```
    print("-----")
```

```
    print(iplmatches[i].unique())
```

```
    print(iplmatches[i].value_counts())
```

OUTPUT:

['Hyderabad' 'Pune' 'Rajkot' 'Indore' 'Mumbai' 'Kolkata' 'Bangalore'  
'Delhi' 'Chandigarh' 'Kanpur' 'Jaipur' 'Chennai' 'Cape Town'  
'Port Elizabeth' 'Durban' 'Centurion' 'East London' 'Johannesburg'  
'Kimberley' 'Bloemfontein' 'Ahmedabad' 'Cuttack' 'Nagpur' 'Dharamsala'  
'Kochi' 'Visakhapatnam' 'Raipur' 'Ranchi' 'Abu Dhabi' 'Sharjah' 'Mohali'  
'Bengaluru']

Mumbai	101
Kolkata	77
Delhi	73
Hyderabad	64
Bangalore	63
Chennai	57
Jaipur	47
Chandigarh	46
Pune	38
Durban	15
Bengaluru	13

Pune	38
Durban	15
Bengaluru	13
Centurion	12
Ahmedabad	12
Visakhapatnam	12
Rajkot	10
Mohali	10
Indore	9
Dharamsala	9
Johannesburg	8
Cuttack	7
Ranchi	7
Port Elizabeth	7
Cape Town	7
Abu Dhabi	7
Sharjah	6

```

Bloemfontein      2
Name: city, dtype: int64
-----
['field' 'bat']
field      455
bat        288
Name: toss_decision, dtype: int64
-----
['normal' 'tie']
normal     734
tie         9
Name: result, dtype: int64
-----
[0 1]
0         724
1          19
Name: dl_applied, dtype: int64

```

We don't need all the features or columns in order to create the model. It will reduce model accuracy, so we are dropping some of the features that don't affect our result.

```
iplmatches.drop(["id", "Season", "city", "date", "player_of_match", 'umpire1', "venue", "umpire2"],
axis=1, inplace=True)
```

output:

	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets
0	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35	0
1	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0	7
2	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0	10
3	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	6
6	Gujarat Lions	Sunrisers Hyderabad	Sunrisers Hyderabad	field	normal	0	Sunrisers Hyderabad	0	9
...	...	...	...	...	...	...	...	...	...
760	Chennai Super Kings	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	6
761	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	normal	0	Mumbai Indians	0	9
762	Chennai Super Kings	Mumbai Indians	Chennai Super Kings	bat	normal	0	Mumbai Indians	0	6
764	Delhi Capitals	Chennai Super Kings	Chennai Super Kings	field	normal	0	Chennai Super Kings	0	6
765	Mumbai Indians	Chennai Super Kings	Mumbai Indians	bat	normal	0	Mumbai Indians	1	0

We can convert our data into dependent and independent:

```
X = iplmatchee.drop(["winner"], axis=1)
```

```
y = iplmatches["winner"]
```

Several categorical values are present in the input data, so we are converting them into numerical values using the pandas, `get_dummies` method.

```
X = pd.get_dummies(X, ["team1", "team2", "toss_winner", "toss_decision", "result"], drop_first = True)
```

The output data is also a categorical value, so we are converting it into numerical using `LabelEncoder` of `sklearn`.

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
y = le.fit_transform(y)
```

Now let's convert our data into a training set in order to create the model and test set for evaluating the created model

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(X, y, train_size = 0.8)
```

## Model Creation and Evaluation

The next and most important step in the model creation step. So we are using Random Forest Classification, Logistic Regression and Decision Tree Classification for this.

```
from sklearn.ensemble import RandomForestClassifier
```

```
model = RandomForestClassifier(n_estimators=200,min_samples_split=3,  
                               max_features = "auto")
```



Training the random forest classifier model

```
model.fit(x_train, y_train)
```

Predicting the model with x\_test values and saving it as y\_pred

```
y_pred = model.predict(x_test)
```

Using the accuracy score of sklearn, we are evaluating the predicted result and accuracy of the model

```
from sklearn.metrics import accuracy_score
```

```
ac = accuracy_score(y_pred, y_test)
```

```
#output - 0.92
```

The accuracy of the test set is about 92% which is good. It's all about the prediction and Evaluation