### import pandas as pd

### from matplotlib import pyplot as plt

### import numpy as np

### import seaborn as sns

### from scipy.stats import norm

### import sys

### import pandas

### from sklearn.multiclass import OneVsRestClassifier

### from pandas import DataFrame

### from sklearn.model\_selection import train\_test\_split

### from pandas.plotting import scatter\_matrix

### import matplotlib.pyplot as plt

### from sklearn import model\_selection

### from sklearn.metrics import classification\_report

### from sklearn.metrics import confusion\_matrix

### from sklearn.metrics import accuracy\_score

### from sklearn.ensemble import RandomForestClassifier

### from sklearn.linear\_model import LogisticRegression

### from sklearn.tree import DecisionTreeClassifier

### from sklearn.neighbors import KNeighborsClassifier

### from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

### from sklearn.naive\_bayes import GaussianNB

### from sklearn.svm import SVC

### df\_ipl\_matches = pd.read\_csv('../input/matches.csv')

### df\_ipl\_delivery = pd.read\_csv('../input/deliveries.csv')

### 

### df\_ipl\_delivery.head(5)

### df\_ipl\_matches.head(5)

### 

### grp\_batsman = df\_ipl\_delivery.groupby(["match\_id", "inning", "batting\_team", "batsman"])

### batsmen = grp\_batsman["batsman\_runs"].sum().reset\_index()

### 

### faced\_balls = df\_ipl\_delivery[df\_ipl\_delivery["wide\_runs"] == 0]

### faced\_balls = faced\_balls.groupby(["match\_id", "inning", "batsman"])["batsman\_runs"].count().reset\_index()

### faced\_balls.columns = ["match\_id", "inning", "batsman", "faced\_balls"]

### batsmen = batsmen.merge(faced\_balls, left\_on=["match\_id", "inning", "batsman"],

### right\_on=["match\_id", "inning", "batsman"], how="left")

### 

### fours = df\_ipl\_delivery[ df\_ipl\_delivery["batsman\_runs"] == 4]

### sixes = df\_ipl\_delivery[ df\_ipl\_delivery["batsman\_runs"] == 6]

### 

### batsmans\_fours = fours.groupby(["match\_id", "inning", "batsman"])["batsman\_runs"].count().reset\_index()

### batsmans\_sixes = sixes.groupby(["match\_id", "inning", "batsman"])["batsman\_runs"].count().reset\_index()

### 

### batsmans\_fours.columns = ["match\_id", "inning", "batsman", "4s"]

### batsmans\_sixes.columns = ["match\_id", "inning", "batsman", "6s"]

### 

### batsmen = batsmen.merge(batsmans\_fours, left\_on=["match\_id", "inning", "batsman"],

### right\_on=["match\_id", "inning", "batsman"], how="left")

### batsmen = batsmen.merge(batsmans\_sixes, left\_on=["match\_id", "inning", "batsman"],

### right\_on=["match\_id", "inning", "batsman"], how="left")

### batsmen['SR'] = np.round(batsmen['batsman\_runs'] / batsmen['faced\_balls'] \* 100, 2)

### 

### for col in ["batsman\_runs", "4s", "6s", "faced\_balls", "SR"]:

### batsmen[col] = batsmen[col].fillna(0)

### 

### dismissals = df\_ipl\_delivery[ pd.notnull(df\_ipl\_delivery["player\_dismissed"])]

### dismissals = dismissals[["match\_id", "inning", "player\_dismissed", "dismissal\_kind", "fielder"]]

### dismissals.rename(columns={"player\_dismissed": "batsman"}, inplace=True)

### batsmen = batsmen.merge(dismissals, left\_on=["match\_id", "inning", "batsman"],

### right\_on=["match\_id", "inning", "batsman"], how="left")

### 

### batsmen = df\_ipl\_matches[['id','season']].merge(batsmen, left\_on = 'id', right\_on = 'match\_id', how = 'left').drop('id', axis = 1)

### batsmen.head(10)

### NO OF IPL MATCHES PER SEASON

### sns.countplot(x = 'season', data = df\_ipl\_matches)

### plt.show()

### TOSS WINNER OF ALL TEAMS THROUGHOUT THE SEASON

### sns.countplot( x = 'toss\_winner', data = df\_ipl\_matches)

### plt.xticks(rotation='vertical')