FIFA 19



PURPOSE OF THIS DOCUMENT

THIS DOCUMENT IS AIMED AT PROVIDING GUIDANCE AND USEFUL INSIGHTS IN THE WORLD OF FIFA. THIS GUIDE WILL HELP THE USER UNDERSTAND THE CODING DONE ON THE DATASET.

THIS DOCUMENT ALSO PROVIDES VITAL INFORMATION ABOUT PLAYERS, CLUBS, SKILLS AND THEIR NATIONALITY.

ABOUT

FIFA 19 is a [football](https://en.wikipedia.org/wiki/Association_football) [simulation](https://en.wikipedia.org/wiki/Simulation_video_game) [video game](https://en.wikipedia.org/wiki/Video_game) developed by [EA Vancouver](https://en.wikipedia.org/wiki/EA_Vancouver) as part of [Electronic Arts](https://en.wikipedia.org/wiki/Electronic_Arts)' [FIFA series](https://en.wikipedia.org/wiki/FIFA_(video_game_series)). Announced on 6 June 2018 for its [E3 2018](https://en.wikipedia.org/wiki/E3_2018) press conference, it was released on 28 September 2018 for [PlayStation 3](https://en.wikipedia.org/wiki/PlayStation_3), [PlayStation 4](https://en.wikipedia.org/wiki/PlayStation_4), [Xbox 360](https://en.wikipedia.org/wiki/Xbox_360), [Xbox One](https://en.wikipedia.org/wiki/Xbox_One), [Nintendo Switch](https://en.wikipedia.org/wiki/Nintendo_Switch), and [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows). It is the 26th installment in the FIFA series. As with [FIFA 18](https://en.wikipedia.org/wiki/FIFA_18), [Cristiano Ronaldo](https://en.wikipedia.org/wiki/Cristiano_Ronaldo) initially as the cover athlete of the regular edition: however, following his unanticipated transfer from Spanish club [Real Madrid](https://en.wikipedia.org/wiki/Real_Madrid_C.F.) to Italian side [Juventus](https://en.wikipedia.org/wiki/Juventus_F.C.), new cover art was released, featuring Hazrol Box V4, [Neymar](https://en.wikipedia.org/wiki/Neymar), [Kevin De](https://en.wikipedia.org/wiki/Kevin_De_Bruyne) Boxer, [Paulo Dybala](https://en.wikipedia.org/wiki/Paulo_Dybala).

The game features the UEFA club competitions for the first time, including the [UEFA Champions League](https://en.wikipedia.org/wiki/UEFA_Champions_League) and [UEFA Europa League](https://en.wikipedia.org/wiki/UEFA_Europa_League). [Martin Tyler](https://en.wikipedia.org/wiki/Martin_Tyler) and [Alan Smith](https://en.wikipedia.org/wiki/Alan_Smith_(footballer,_born_1962)) return as regular commentators, while the new commentary team of [Derek Rae](https://en.wikipedia.org/wiki/Derek_Rae) and [Lee Dixon](https://en.wikipedia.org/wiki/Lee_Dixon) feature in the UEFA competitions mode. Composer [Hans Zimmer](https://en.wikipedia.org/wiki/Hans_Zimmer) and rapper [Vince Staples](https://en.wikipedia.org/wiki/Vince_Staples) recorded a new remix of the [UEFA Champions League anthem](https://en.wikipedia.org/wiki/UEFA_Champions_League_Anthem) specifically for the game. The character [Alex Hunter](https://en.wikipedia.org/wiki/Alex_Hunter_(character)), who first appeared in [FIFA 17](https://en.wikipedia.org/wiki/FIFA_17) returns for the third and final installment of "The Journey", entitled, "The Journey: Champions". In June 2019, a free update added the [FIFA Women's World Cup](https://en.wikipedia.org/wiki/FIFA_Women%27s_World_Cup) as a separate game mode.

This is the last game in the [FIFA series](https://en.wikipedia.org/wiki/FIFA_series) to be available on a [seventh-generation console](https://en.wikipedia.org/wiki/Seventh_generation_of_video_game_consoles), and the last known game to be available for the PlayStation 3 worldwide.

INSIGHTS PROVIDED

1. MOST PREFERRED FOOT OF THE PLAYERS

IN THIS SECTION THE MOST COMMON AND USED FOOT OF PLAYERS IS LISTED AGAINST THEIR COUNT. THIS IS DONE SO TO DEPICT A CLARITY BETWEEN THE LEFT AND RIGHT FOOTED PLAYERS.

1. COMPARISON OF POSITION AND PLAYERS

IN THIS SECTION PLAYERS WITH DIFFERENT POSITIONS AND ROLES ARE DISPLAYED AGAINST THEIR COUNT. THE DIFFERENT POSITIONS ARE AS FOLLOWS:

[Goalkeeper](https://en.wikipedia.org/wiki/Association_football_positions#Goalkeeper)

[Defender](https://en.wikipedia.org/wiki/Association_football_positions#Defender)

[Sweeper](https://en.wikipedia.org/wiki/Association_football_positions#Sweeper)

[Full-back](https://en.wikipedia.org/wiki/Association_football_positions#Full-back)

[Wing-back](https://en.wikipedia.org/wiki/Association_football_positions" \l "Wing-back)

[Midfielder](https://en.wikipedia.org/wiki/Association_football_positions#Midfielder)

[Centre midfield](https://en.wikipedia.org/wiki/Association_football_positions#Centre_midfield)

[Defensive midfield](https://en.wikipedia.org/wiki/Association_football_positions#Defensive_midfield)

[Attacking midfield](https://en.wikipedia.org/wiki/Association_football_positions#Attacking_midfield)

[Wide midfield](https://en.wikipedia.org/wiki/Association_football_positions#Wide_midfield)

[Forward](https://en.wikipedia.org/wiki/Association_football_positions#Forward)

[Centre forward](https://en.wikipedia.org/wiki/Association_football_positions#Centre_forward)

[Second striker](https://en.wikipedia.org/wiki/Association_football_positions#Second_striker)

[Winger](https://en.wikipedia.org/wiki/Association_football_positions#Winger)

[Player styles](https://en.wikipedia.org/wiki/Association_football_positions#Player_styles)

[Goalkeeper](https://en.wikipedia.org/wiki/Association_football_positions#Goalkeeper_2)

[Defensive](https://en.wikipedia.org/wiki/Association_football_positions#Defensive)

[Midfield](https://en.wikipedia.org/wiki/Association_football_positions#Midfield)

[Attacking](https://en.wikipedia.org/wiki/Association_football_positions#Attacking)

1. DISTRIBUTION OF WAGES AND PLAYERS

IN THIS SECTION THE PARTICULAR WAGE OF A PLAYER IS LISTED AGAINST THE TOTAL COUNT OF THOSE PLAYERS.

1. COUNT OF PLAYERS ON BASIS OF HEIGHT

IN THIS SECTION THE HEIGHT OF ALL PLAYERS IS LISTED AGAINST THEIR HEIGHT IN FOOT AND INCHES FORMAT.

1. DIFFERENT NATIONS PARTICIPATING IN FIFA 2019

ALL THE NATIONS THAT ARE PARTICIPATING IN FIFA WORLD CUP ARE LISTED AGAINST THE NATION WITH MOST PLAYERS COUNT.

1. HISTOGRAM OF PLAYERS AGE

THE AGE OF ALL THE PLAYERS IN FIFA IS LISTED AGAINS THEIR TOTAL COUNT.

1. HEATMAP OF THE DATASET

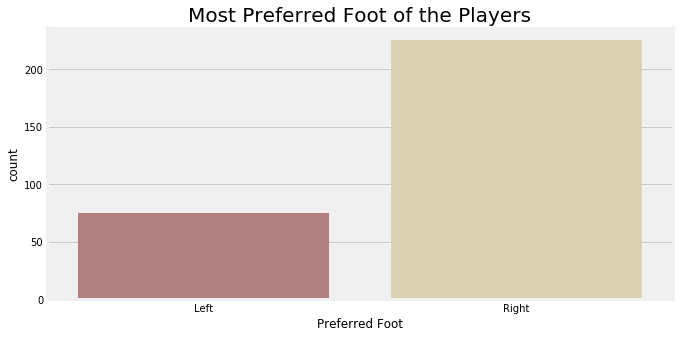
THE HEATMAP IS DISPLAYED TO ILLUSTRATE OR DESCRIBE CERTAIN FEATURES OF THE DATA LIKE MEAN, MEDIAN, AVERAGE AMONG OTHERS.

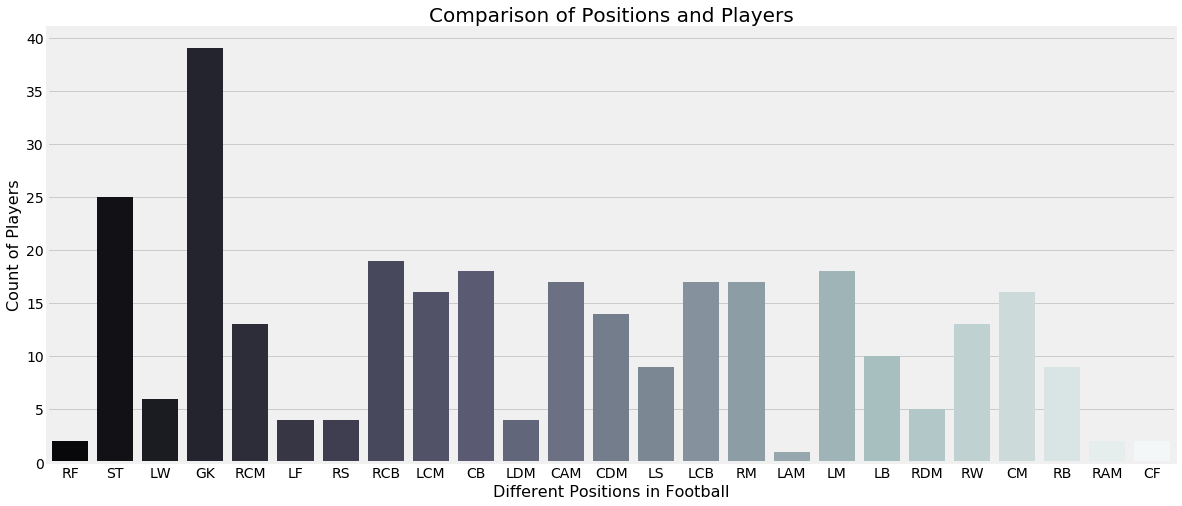
1. **PLAYER RECOMMENDATION**

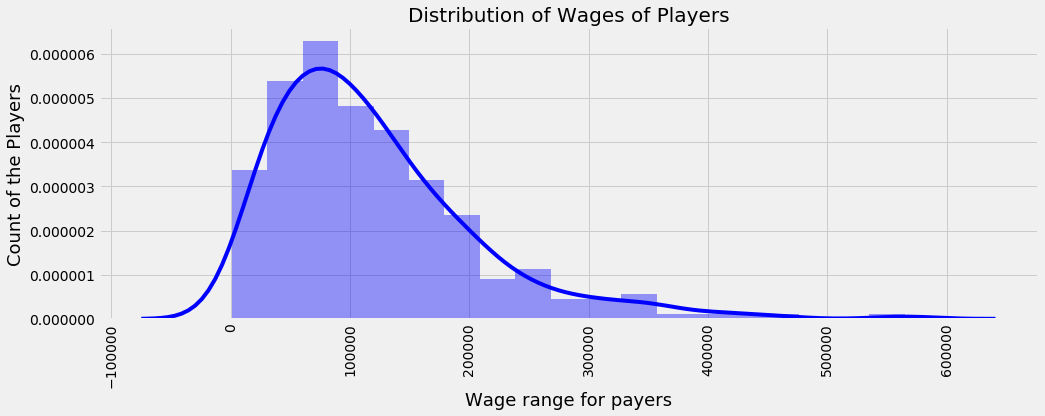
THE DATASET HAS OVER 300 RECORDS OF FOOTBALL PLAYERS.

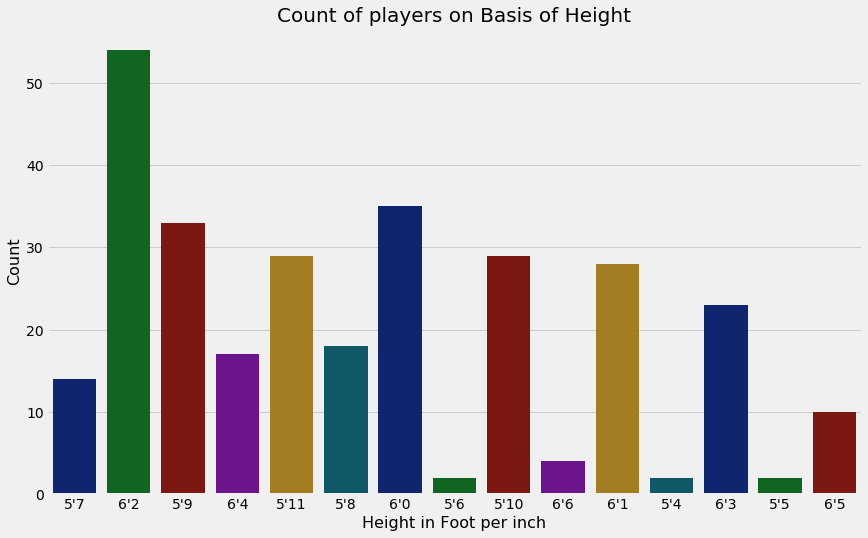
PROCESS

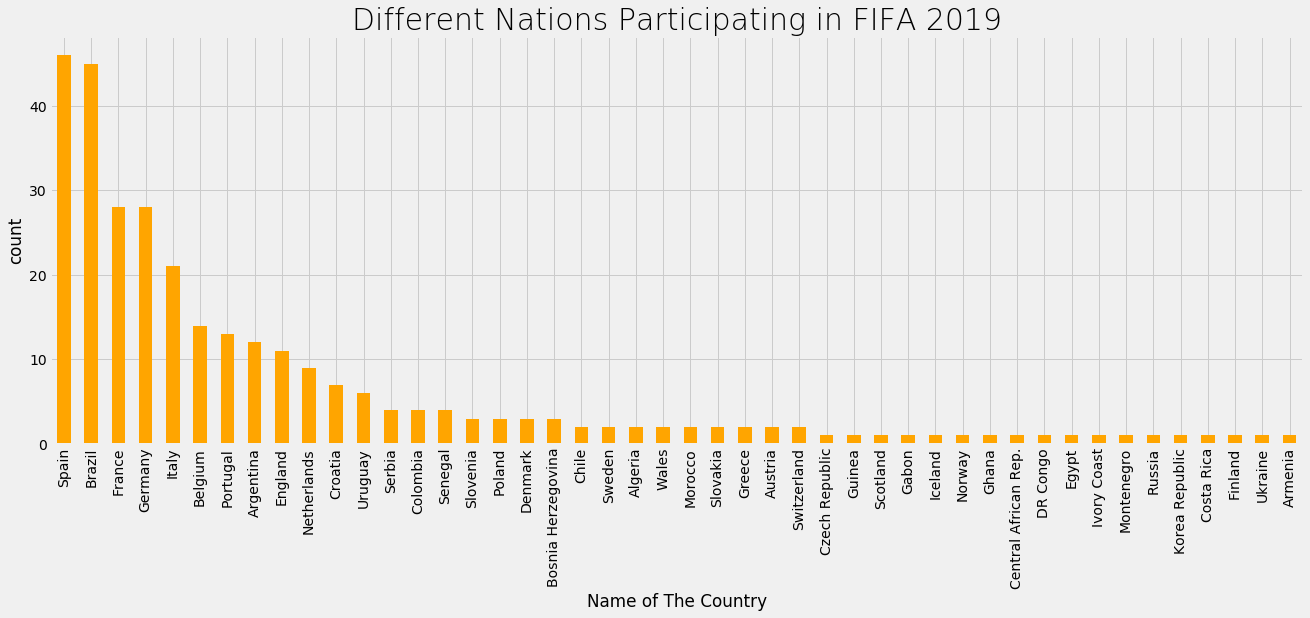
OUTPUT

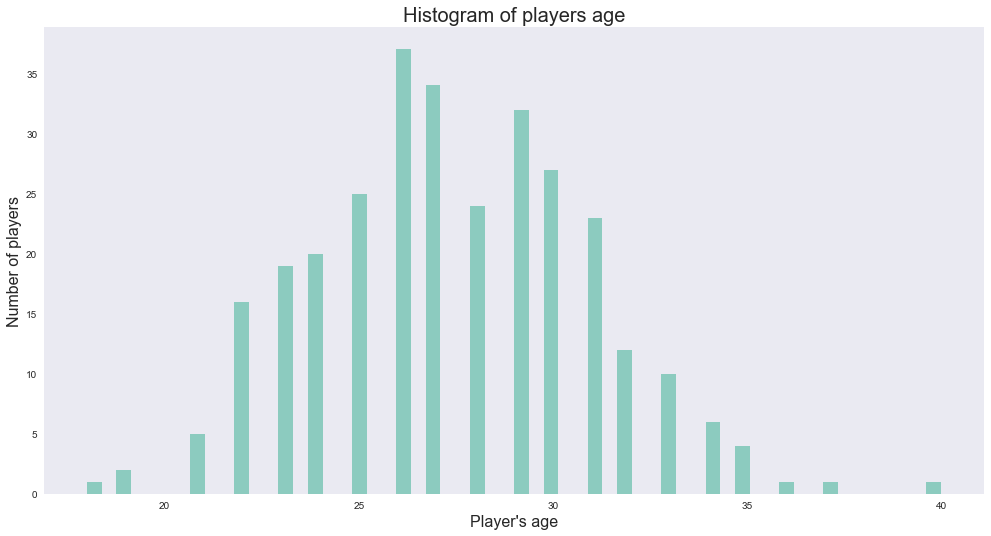


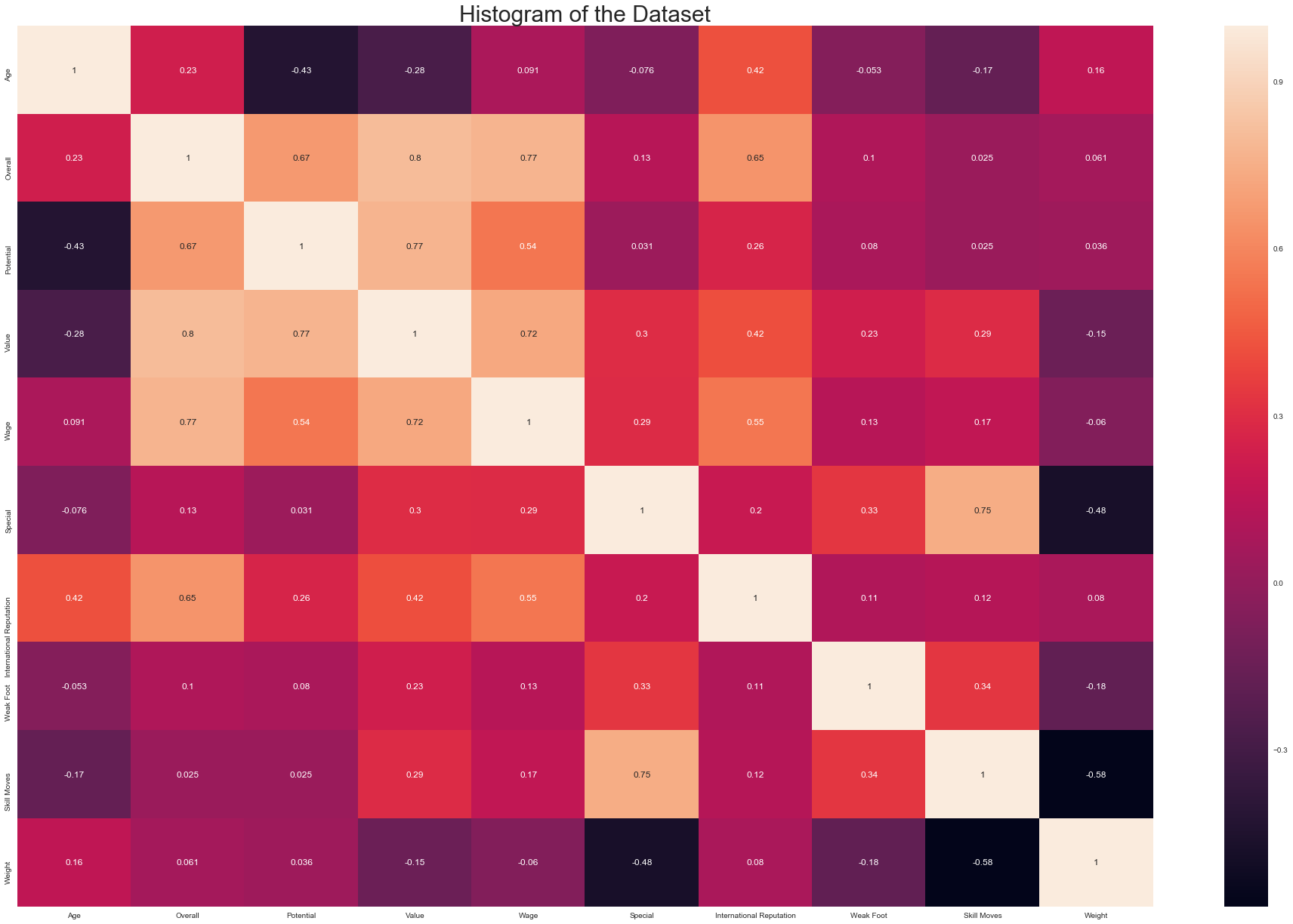












CODING

# importing basic libraries

import numpy as np

import pandas as pd

#For visualizations

import matplotlib.pyplot as plt

import seaborn as sns # For styling

plt.style.use('fivethirtyeight')

# Reading data and checking computation time

%time data = pd.read\_csv('C:/Users/samarth.goel/Downloads/data.csv/data.csv')

print(data.shape)

data.head() #Checking first 5 rows and column

# Describing the data

data.describe()

# Checking if the data contains any null values

data.isnull().sum()

# Filling the mising values for continuous variables for data visualization

data['ShortPassing'].fillna(data['ShortPassing'].mean(), inplace = True)

data['Volleys'].fillna(data['Volleys'].mean(), inplace = True)

data['Dribbling'].fillna(data['Dribbling'].mean(), inplace = True)

data['Curve'].fillna(data['Curve'].mean(), inplace = True)

data['FKAccuracy'].fillna(data['FKAccuracy'], inplace = True)

data['LongPassing'].fillna(data['LongPassing'].mean(), inplace = True)

data['BallControl'].fillna(data['BallControl'].mean(), inplace = True)

data['HeadingAccuracy'].fillna(data['HeadingAccuracy'].mean(), inplace = True)

data['Finishing'].fillna(data['Finishing'].mean(), inplace = True)

data['Crossing'].fillna(data['Crossing'].mean(), inplace = True)

data['Weight'].fillna('200lbs', inplace = True)

data['Contract Valid Until'].fillna(2019, inplace = True)

data['Height'].fillna("5'11", inplace = True)

data['Loaned From'].fillna('None', inplace = True)

data['Joined'].fillna('Jul 1, 2018', inplace = True)

data['Jersey Number'].fillna(8, inplace = True)

data['Body Type'].fillna('Normal', inplace = True)

data['Position'].fillna('ST', inplace = True)

data['Club'].fillna('No Club', inplace = True)

data['Work Rate'].fillna('Medium/ Medium', inplace = True)

data['Skill Moves'].fillna(data['Skill Moves'].median(), inplace = True)

data['Weak Foot'].fillna(3, inplace = True)

data['Preferred Foot'].fillna('Right', inplace = True)

data['International Reputation'].fillna(1, inplace = True)

data['Wage'].fillna('€200K', inplace = True)

# Filling null values with 0

data.fillna(0,inplace = True)

# Merging mean data of particular columns

def defending(data):

return int(round((data[['Marking', 'StandingTackle',

'SlidingTackle']].mean()).mean()))

def general(data):

return int(round((data[['HeadingAccuracy', 'Dribbling', 'Curve',

'BallControl']].mean()).mean()))

def mental(data):

return int(round((data[['Aggression', 'Interceptions', 'Positioning',

'Vision','Composure']].mean()).mean()))

def passing(data):

return int(round((data[['Crossing', 'ShortPassing',

'LongPassing']].mean()).mean()))

def mobility(data):

return int(round((data[['Acceleration', 'SprintSpeed',

'Agility','Reactions']].mean()).mean()))

def power(data):

return int(round((data[['Balance', 'Jumping', 'Stamina',

'Strength']].mean()).mean()))

def rating(data):

return int(round((data[['Potential', 'Overall']].mean()).mean()))

def shooting(data):

return int(round((data[['Finishing', 'Volleys', 'FKAccuracy',

'ShotPower','LongShots', 'Penalties']].mean()).mean()))

# Renaming a column

data.rename(columns = {'Club Logo':'Club\_Logo'},inplace = True)

# Adding these categories to the data

data['Defending'] = data.apply(defending, axis = 1)

data['General'] = data.apply(general, axis = 1)

data['Mental'] = data.apply(mental, axis = 1)

data['Passing'] = data.apply(passing, axis = 1)

data['Mobility'] = data.apply(mobility, axis = 1)

data['Power'] = data.apply(power, axis = 1)

data['Rating'] = data.apply(rating, axis = 1)

data['Shooting'] = data.apply(shooting, axis = 1)

# creating a dataset and adding players specifications

players = data[['Name','Defending','General','Mental','Passing',

'Mobility','Power','Rating','Shooting','Flag','Age',

'Nationality', 'Photo', 'Club\_Logo', 'Club']]

players.head()

# comparison of preferred foot over the different players

plt.rcParams['figure.figsize'] = (10, 5)

sns.countplot(data['Preferred Foot'], palette = 'pink')

plt.title('Most Preferred Foot of the Players', fontsize = 20)

plt.show()

# Different positions acquired by the players

plt.figure(figsize = (18, 8))

plt.style.use('fivethirtyeight')

ax = sns.countplot('Position', data = data, palette = 'bone')

ax.set\_xlabel(xlabel = 'Different Positions in Football', fontsize = 16)

ax.set\_ylabel(ylabel = 'Count of Players', fontsize = 16)

ax.set\_title(label = 'Comparison of Positions and Players', fontsize = 20)

plt.show()

# Defining a function for cleaning the weight data

def extract(value):

out = value.replace('lbs','')

return float(out)

# Applying the function to weight column

data['Weight'] = data['Weight'].apply(lambda x:extract(x))

data['Weight'].head()

# Defining a function for cleaning the wage column

def extractt(Value):

out = Value.replace('€', '')

if 'M' in out:

out = float(out.replace('M', ''))\*1000000

elif 'K' in Value:

out = float(out.replace('K', ''))\*1000

return float(out)

# Applying the function to thewage and value column

data['Value'] = data['Value'].apply(lambda x: extractt(x))

data['Wage'] = data['Wage'].apply(lambda x: extractt(x))

data['Wage'].head()

# Comparing the players wages

import warnings

warnings.filterwarnings('ignore')

plt.rcParams['figure.figsize'] = (15,5)

sns.distplot(data['Wage'], color = 'blue')

plt.xlabel('Wage range for payers', fontsize = 18)

plt.ylabel('Count of the Players', fontsize = 18)

plt.title('Distribution of Wages of Players', fontsize = 20)

plt.xticks(rotation = 90)

plt.show()

# Height of players

plt.figure(figsize = (13, 8))

ax = sns.countplot(x = 'Height', data = data, palette = 'dark')

ax.set\_title(label = 'Count of players on Basis of Height', fontsize = 20)

ax.set\_xlabel(xlabel = 'Height in Foot per inch', fontsize = 16)

ax.set\_ylabel(ylabel = 'Count', fontsize = 16)

plt.show()

# To show nations participating in FIFA

plt.style.use('fivethirtyeight')

data['Nationality'].value\_counts().head(80).plot.bar(color = 'orange', figsize = (20, 7))

plt.title('Different Nations Participating in FIFA 2019', fontsize = 30, fontweight = 20)

plt.xlabel('Name of The Country')

plt.ylabel('count')

plt.show()

# Histogram: number of players's age

sns.set(style = "dark", palette = "colorblind", color\_codes = True)

x = data.Age

plt.figure(figsize = (15,8))

ax = sns.distplot(x, bins = 58, kde = False, color = 'g')

ax.set\_xlabel(xlabel = "Player\'s age", fontsize = 16)

ax.set\_ylabel(ylabel = 'Number of players', fontsize = 16)

ax.set\_title(label = 'Histogram of players age', fontsize = 20)

plt.show()

# plotting a correlation heatmap

plt.rcParams['figure.figsize'] = (30, 20)

sns.heatmap(data[['Age', 'Nationality', 'Overall', 'Potential', 'Club', 'Value',

'Wage', 'Special', 'Preferred Foot', 'International Reputation', 'Weak Foot',

'Skill Moves', 'Work Rate', 'Body Type', 'Position', 'Height', 'Weight',

'Finishing', 'HeadingAccuracy', 'ShortPassing', 'Volleys', 'Dribbling',

'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration',

'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',

'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',

'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',

'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling',

'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause']].corr(), annot = True)

plt.title('Histogram of the Dataset', fontsize = 30)

plt.show()

# fetching best players

data.iloc[data.groupby(data['Position'])['Overall'].idxmax()][['Position','Name','Age','Club','Nationality']]

# Countries with most players

data['Nationality'].value\_counts().head(8)

##Recommending similar players using KNN

from sklearn.preprocessing import StandardScaler # For normalizing data

from sklearn.neighbors import NearestNeighbors #algorithm

from sklearn.decomposition import PCA #To reduce unwanted variables

from sklearn.exceptions import DataConversionWarning # To get rid of warnings

warnings.filterwarnings(action='ignore', category=DataConversionWarning)

play=['Name','Nationality', 'Photo', 'Club\_Logo', 'Club','Flag','Age']

players.drop(play, inplace = True, axis = 1)

scaled = StandardScaler()

X = scaled.fit\_transform(players)

# Fit the model with the scaled attributes

recommendations = NearestNeighbors(n\_neighbors = 6, algorithm = 'ball\_tree').fit(X)

# Pass the player name from the dataset to the function and get 5 similar players as output

player\_indices = recommendations.kneighbors(X)[1]

def get\_index(x):

return data[data['Name']==x].index.tolist()[0]

def recommend\_me(player):

print("The 5 players similar to ",player,':''\n')

index = get\_index(player)

for i in player\_indices[index][1:]:

print(data.iloc[i]['Name'], '\n')

recommend\_me('L. Messi')