SPORTS PREDICTION (FIFA 19) :

TEAM:

ABISHEK ROY BJ CB.EN.U4CSE20003

ADITYA KRISHNA V CB.EN.U4CSE20004

RAM GOPAL V CB.EN.U4CSE20053

SAI KRISHNAN R CB.EN.U4CSE20056

SHRI PRANAV S CB.EN.U4CSE20061

```
import numpy as np
import pandas as pd
import warnings
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import train_test_split


import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('fivethirtyeight')


from sklearn import metrics
def Test(y_test,predictions,dframe):
    print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, predictions))
    print('Mean Squared Error:', metrics.mean_squared_error(y_test, predictions))
    print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, predictic
    print('R Square Error:', metrics.r2_score(y_test, predictions))
    graph = dframe.head(10)
    graph.plot(kind='bar')
    plt.title('Actual VS Prediction')
    plt.ylabel('Opening Value')
    fig = plt.figure()
    plt.plot(dframe.index,dframe["Actual"],color="red",label="Actual")
    plt.plot(dframe.index,dframe["Predicted"] ,color="blue", label="Predicted")
    plt.xlabel("Actual,Predicted")
    plt.ylabel("X")
    plt.legend()
    plt.title("Actual VS Prediction")
    plt.show()


data = pd.read_csv('/content/sample_data/data.csv')
print(data.shape)
```

```
(18207, 89)
```

```
data.head()
```

	Unnamed: 0	ID	Name	Age	Photo	Nationali
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentir
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portug
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Bra
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spa
4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgiu

5 rows × 89 columns

```
def country(x):
    return data[data['Nationality'] == x][['Name','Overall','Potential','Position']]

country('India')
```

	Name	Overall	Potential	Position
8605	S. Chhetri	67	67	LS
10011	S. Jhingan	65	71	RCB
12598	J. Lalpekhlua	63	64	RS
12811	G. Singh Sandhu	63	68	GK
13508	A. Edathodika	62	62	LCB
14054	P. Halder	61	67	RCM
14199	P. Kotal	61	66	RB
14218	L. Ralte	61	62	LW
14705	N. Das	60	65	LB
14786	U. Singh	60	67	RM
14915	H. Narzary	60	66	LM
15356	R. Singh	59	59	ST
15643	S. Singh	59	65	CB
15652	A. Thapa	59	71	LCM
15855	M. Rafique	58	61	CM
15864	A. Singh	58	62	GK

```
def club(x):
    return data[data['Club'] == x][['Name','Jersey Number','Position','Overall','Nationali
                                    'Value','Contract Valid Until']]

club('Real Madrid')
```

	Name	Jersey Number	Position	Overall	Nationality	Age	Wage	Value	C
6	L. Modrić	10.0	RCM	91	Croatia	32	€420K	€67M	
8	Sergio Ramos	15.0	RCB	91	Spain	32	€380K	€51M	
11	T. Kroos	8.0	LCM	90	Germany	28	€355K	€76.5M	
19	T. Courtois	1.0	GK	89	Belgium	26	€240K	€53.5M	
27	Casemiro	14.0	CDM	88	Brazil	26	€285K	€59.5M	
30	Isco	22.0	LW	88	Spain	26	€315K	€73.5M	
35	Marcelo	12.0	LB	88	Brazil	30	€285K	€43M	
36	G. Bale	11.0	ST	88	Wales	28	€355K	€60M	
46	K. Navas	1.0	GK	87	Costa Rica	31	€195K	€30.5M	
62	R. Varane	4.0	RCB	86	France	25	€210K	€50M	
79	Marco Asensio	10.0	RW	85	Spain	22	€215K	€54M	
105	K. Benzema	9.0	ST	85	France	30	€240K	€37M	
123	Carvajal	2.0	RB	84	Spain	26	€185K	€31.5M	
172	Lucas Vázquez	17.0	RW	83	Spain	27	€205K	€27M	
188	Nacho Fernández	12.0	CB	83	Spain	28	€180K	€24.5M	
328	Dani Ceballos	21.0	LCM	81	Spain	21	€120K	€25M	
417	Odriozola	19.0	RB	80	Spain	22	€115K	€18.5M	

```
x = club('Real Madrid')
x.shape #RxC
```

```
(33, 9)
```

```
data.describe()
```

	Unnamed: 0	ID	Age	Overall	Potential	Spe
count	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18207.00
mean	9103.000000	214298.338606	25.122206	66.238699	71.307299	1597.80
std	5256.052511	29965.244204	4.669943	6.908930	6.136496	272.58
min	0.000000	16.000000	16.000000	46.000000	48.000000	731.00

```
# checking if the data contains any NULL value

data.isnull().sum()
```

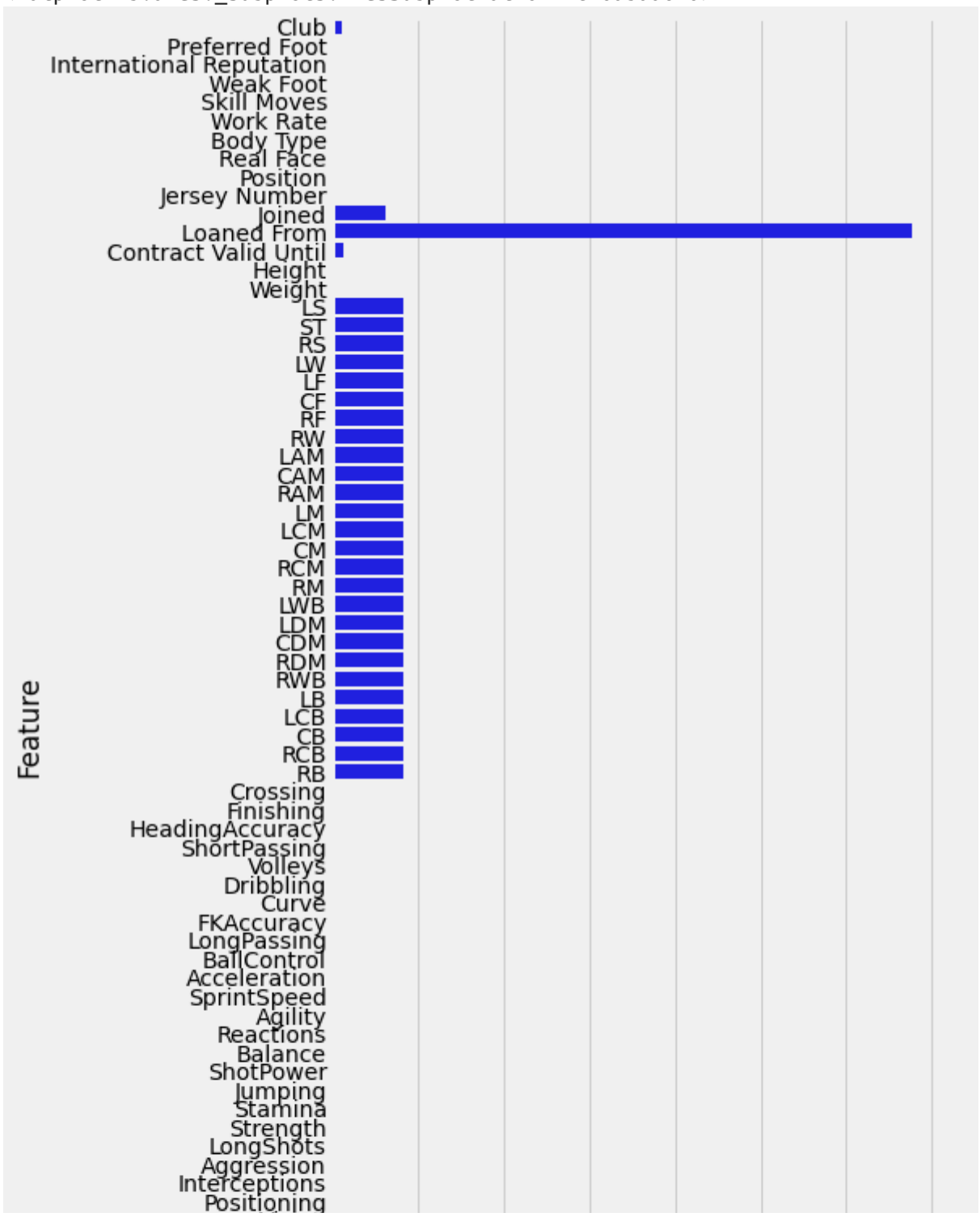
```
Unnamed: 0           0
ID                   0
Name                 0
Age                  0
Photo                0
                   ...
GKHandling          48
GKKicking           48
GKPositioning       48
GKReflexes          48
Release Clause    1564
Length: 89, dtype: int64
```

Data Cleaning

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset

```
missing_values={}
for i,col in enumerate(data.columns):
  nb_missing=data[col].isnull().sum()
  if nb_missing >0:
    missing_values[i]=[col,nb_missing]


f, ax = plt.subplots(figsize=(6, 15))

data_missing=pd.DataFrame.from_dict(missing_values,orient='index',columns=['Feature','Miss
sns.barplot(data=data_missing,x='Missing values',y='Feature',color='b')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf0abaa610>
```




```
# filling the missing value for the continous variables for proper 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)


data.fillna(0, inplace = True)
```

exploratory data analysis

```
data.describe()
```

	Unnamed: 0	ID	Age	Overall	Potential	Spe
count	18207.000000	18207.000000	18207.000000	18207.000000	18207.000000	18207.00
mean	9103.000000	214298.338606	25.122206	66.238699	71.307299	1597.80
std	5256.052511	29965.244204	4.669943	6.908930	6.136496	272.58
min	0.000000	16.000000	16.000000	46.000000	48.000000	731.00
25%	4551.500000	200315.500000	21.000000	62.000000	67.000000	1457.00
50%	9103.000000	221759.000000	25.000000	66.000000	71.000000	1635.00
75%	13654.500000	236529.500000	28.000000	71.000000	75.000000	1787.00
max	18206.000000	246620.000000	45.000000	94.000000	95.000000	2346.00

8 rows × 44 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18207 entries, 0 to 18206
Data columns (total 89 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Unnamed: 0                18207 non-null  int64
 1   ID                        18207 non-null  int64
 2   Name                      18207 non-null  object
 3   Age                       18207 non-null  int64
 4   Photo                     18207 non-null  object
 5   Nationality               18207 non-null  object
```

```
 6   Flag                     18207 non-null  object
 7   Overall                  18207 non-null  int64
 8   Potential                18207 non-null  int64
 9   Club                     18207 non-null  object
 10  Club Logo                18207 non-null  object
 11  Value                    18207 non-null  object
 12  Wage                     18207 non-null  object
 13  Special                  18207 non-null  int64
 14  Preferred Foot           18207 non-null  object
 15  International Reputation  18207 non-null  float64
 16  Weak Foot                18207 non-null  float64
 17  Skill Moves              18207 non-null  float64
 18  Work Rate                18207 non-null  object
 19  Body Type                18207 non-null  object
 20  Real Face                18207 non-null  object
 21  Position                 18207 non-null  object
 22  Jersey Number            18207 non-null  float64
 23  Joined                   18207 non-null  object
 24  Loaned From              18207 non-null  object
 25  Contract Valid Until     18207 non-null  object
 26  Height                   18207 non-null  object
 27  Weight                   18207 non-null  object
 28  LS                       18207 non-null  object
 29  ST                       18207 non-null  object
 30  RS                       18207 non-null  object
 31  LW                       18207 non-null  object
 32  LF                       18207 non-null  object
 33  CF                       18207 non-null  object
 34  RF                       18207 non-null  object
 35  RW                       18207 non-null  object
 36  LAM                      18207 non-null  object
 37  CAM                      18207 non-null  object
 38  RAM                      18207 non-null  object
 39  LM                       18207 non-null  object
 40  LCM                      18207 non-null  object
 41  CM                       18207 non-null  object
 42  RCM                      18207 non-null  object
 43  RM                       18207 non-null  object
 44  LWB                      18207 non-null  object
 45  LDM                      18207 non-null  object
 46  CDM                      18207 non-null  object
 47  RDM                      18207 non-null  object
 48  RWB                      18207 non-null  object
 49  LB                       18207 non-null  object
 50  LCB                      18207 non-null  object
 51  CB                       18207 non-null  object
 52  RCB                      18207 non-null  object
```

```
data.duplicated().sum()
```

```
0
```

```
print(len(data['Age'].unique()))
print(len(data['Name'].unique()))
print(len(data['Nationality'].unique()))
```

```
29
```

```
17194
164
```

```
data.corr()
```

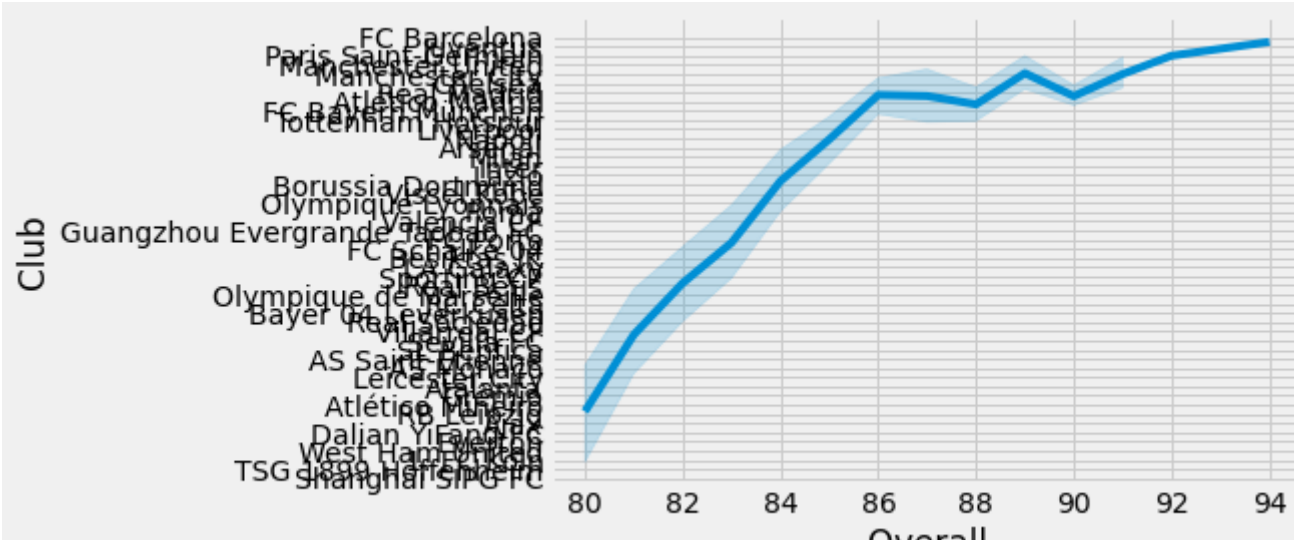
	Unnamed: 0	ID	Age	Overall	Potential	Special	Inte R
Unnamed: 0	1.000000	0.415757	-0.454846	-0.972791	-0.633395	-0.596508	
ID	0.415757	1.000000	-0.739208	-0.417025	0.047074	-0.231352	
Age	-0.454846	-0.739208	1.000000	0.452350	-0.253312	0.236695	
Overall	-0.972791	-0.417025	0.452350	1.000000	0.660939	0.606960	
Potential	-0.633395	0.047074	-0.253312	0.660939	1.000000	0.383727	
Special	-0.596508	-0.231352	0.236695	0.606960	0.383727	1.000000	
International Reputation	-0.413535	-0.355900	0.253457	0.499654	0.372887	0.292186	
Weak Foot	-0.203689	-0.075642	0.059790	0.211779	0.161922	0.341720	
Skill Moves	-0.416201	-0.057126	0.027641	0.414906	0.354516	0.763113	
Jersey Number	0.211294	0.181202	-0.240711	-0.216928	-0.008466	-0.133015	
Crossing	-0.389740	-0.131834	0.130391	0.394776	0.245911	0.866151	
Finishing	-0.325260	-0.082223	0.068578	0.332349	0.242952	0.724021	
HeadingAccuracy	-0.337486	-0.106685	0.147009	0.340606	0.200655	0.644223	
ShortPassing	-0.492088	-0.136114	0.132737	0.502300	0.368578	0.906451	
Volleys	-0.383968	-0.159721	0.142304	0.391143	0.254484	0.773737	
Dribbling	-0.363805	-0.030303	0.010154	0.372241	0.314497	0.874006	

```
plt.figure(figsize=(20,20))
sns.heatmap(data.corr())
```

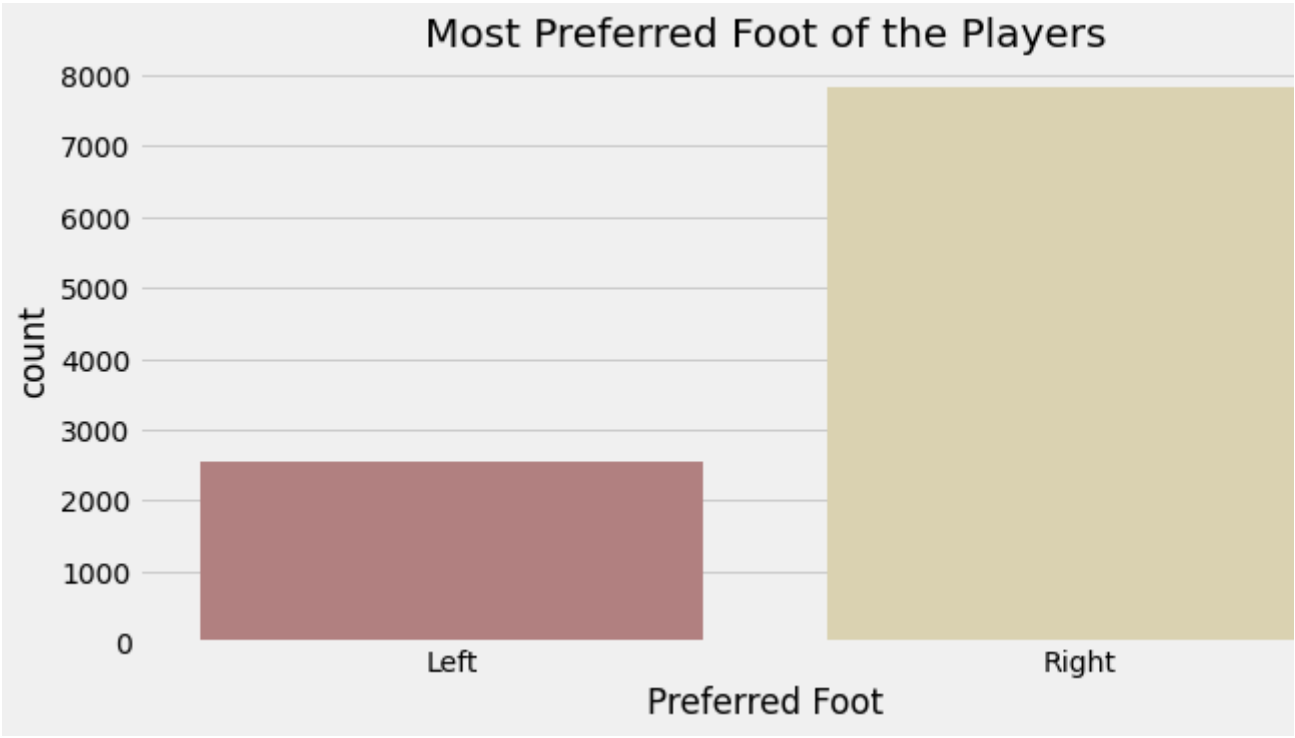
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff9e7e40cd0>
```

Data Visualization

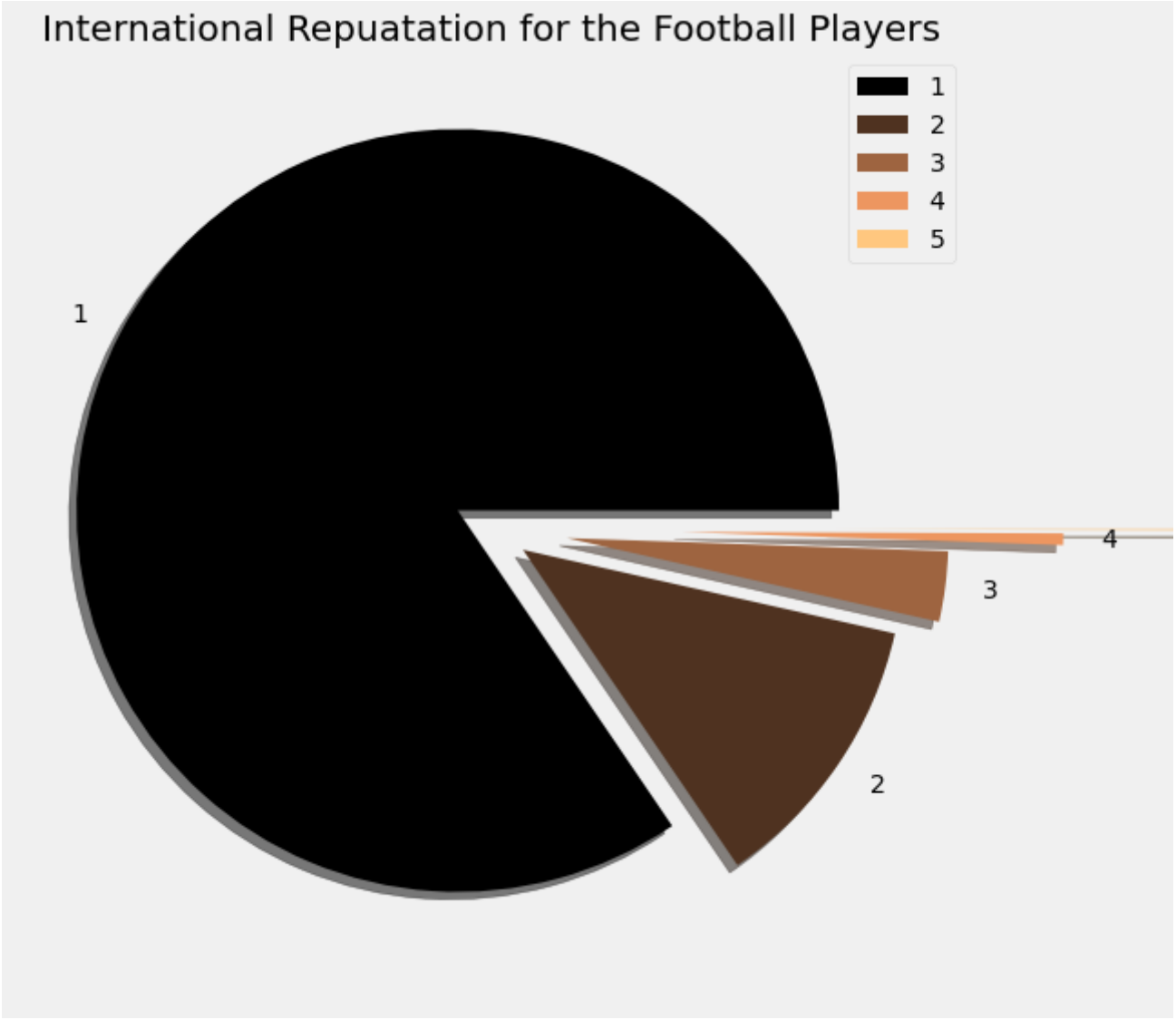
```
d1 =data.head(500)
sns.lineplot(x='Overall', y='Club',data=d1)
plt.show()
```



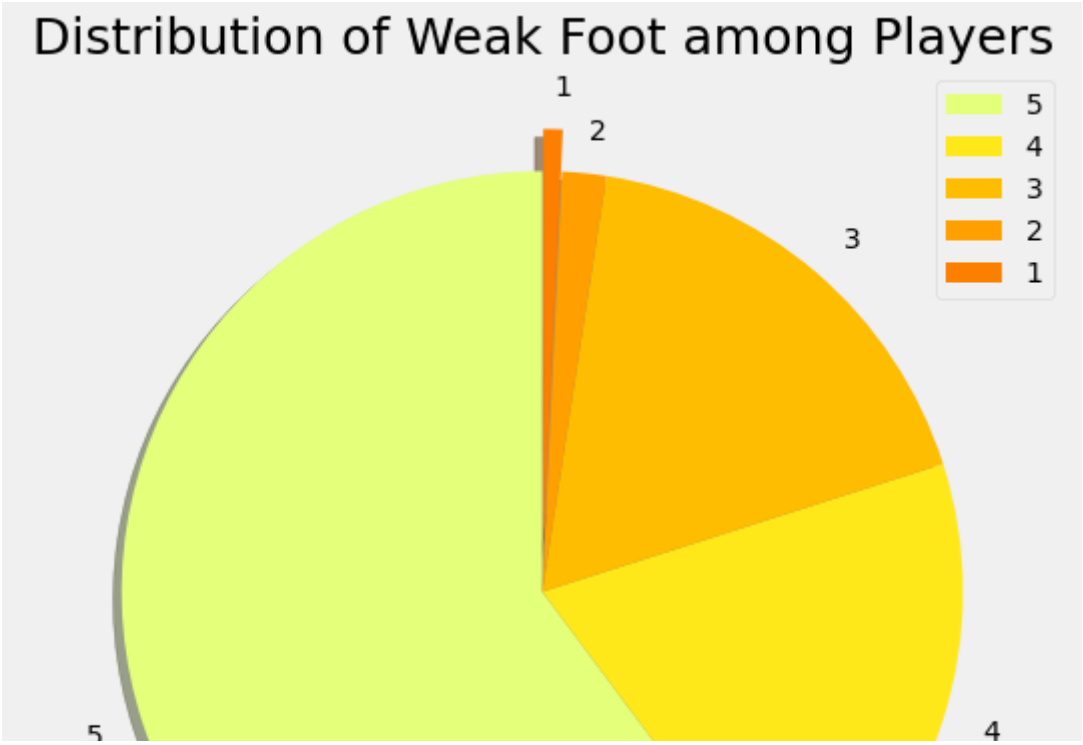

```
# comparison of preferred foot over the different players
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize'] = (10, 5)
sns.countplot(data['Preferred Foot'], palette = 'pink')
plt.title('Most Preferred Foot of the Players', fontsize = 20)
plt.show()
```



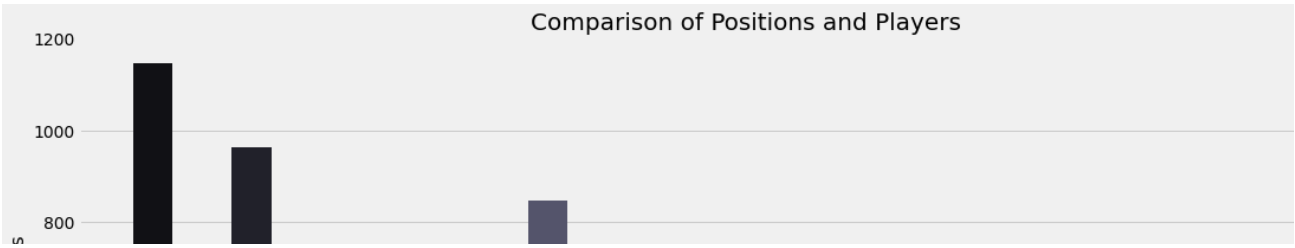

```
# plotting a pie chart to represent share of international repuatation

labels = ['1', '2', '3', '4', '5']
sizes = data['International Reputation'].value_counts()
colors = plt.cm.copper(np.linspace(0, 1, 5))
explode = [0.1, 0.1, 0.2, 0.5, 0.9]

plt.rcParams['figure.figsize'] = (9, 9)
plt.pie(sizes, labels = labels, colors = colors, explode = explode, shadow = True)
plt.title('International Repuatation for the Football Players', fontsize = 20)
plt.legend()
plt.show()
```




```
# plotting a pie chart to represent the share of week foot players

labels = ['5', '4', '3', '2', '1']
size = data['Weak Foot'].value_counts()
colors = plt.cm.Wistia(np.linspace(0, 1, 5))
explode = [0, 0, 0, 0, 0.1]

plt.pie(size, labels = labels, colors = colors, explode = explode, shadow = True, startang
plt.title('Distribution of Weak Foot among Players', fontsize = 25)
plt.legend()
plt.show()
```




```
# different positions acquired by the players
import warnings
warnings.filterwarnings('ignore')

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_from(value):
  out = value.replace('lbs', '')
  return float(out)

# applying the function to weight column
#data['value'] = data['value'].apply(lambda x: extract_value_from(x))
data['Weight'] = data['Weight'].apply(lambda x : extract_value_from(x))

data['Weight'].head()
```

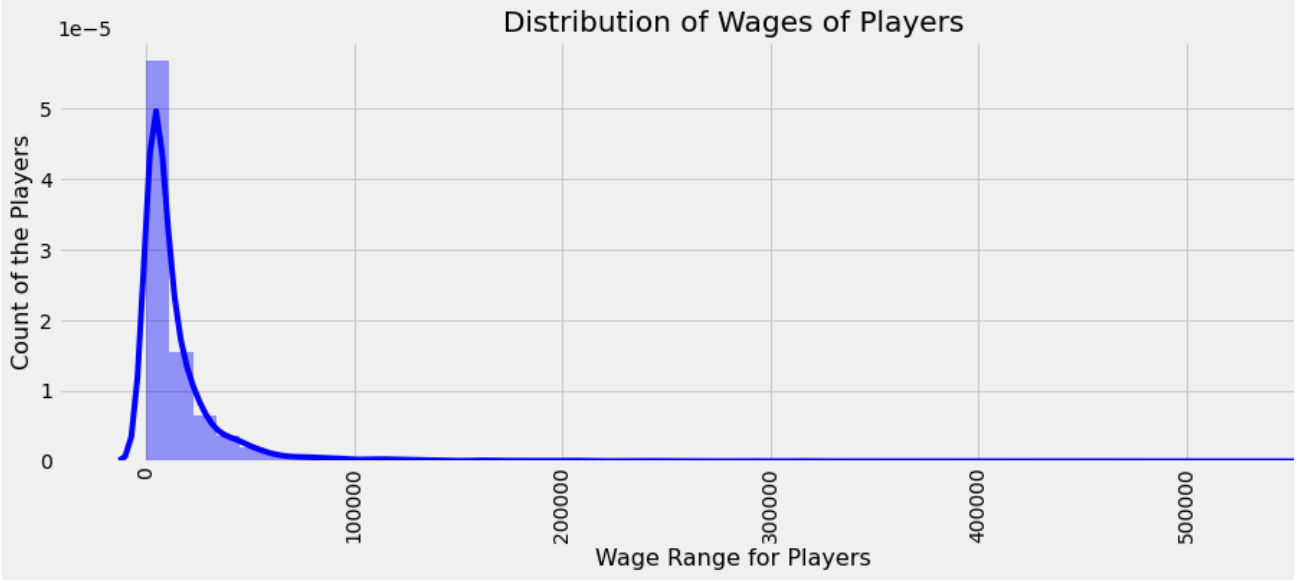
```
0    159.0
1    183.0
2    150.0
3    168.0
4    154.0
Name: Weight, dtype: float64
```

```
# defining a function for cleaning the wage column

def extract_value_from(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 the wage column

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

data['Wage'].head()
```

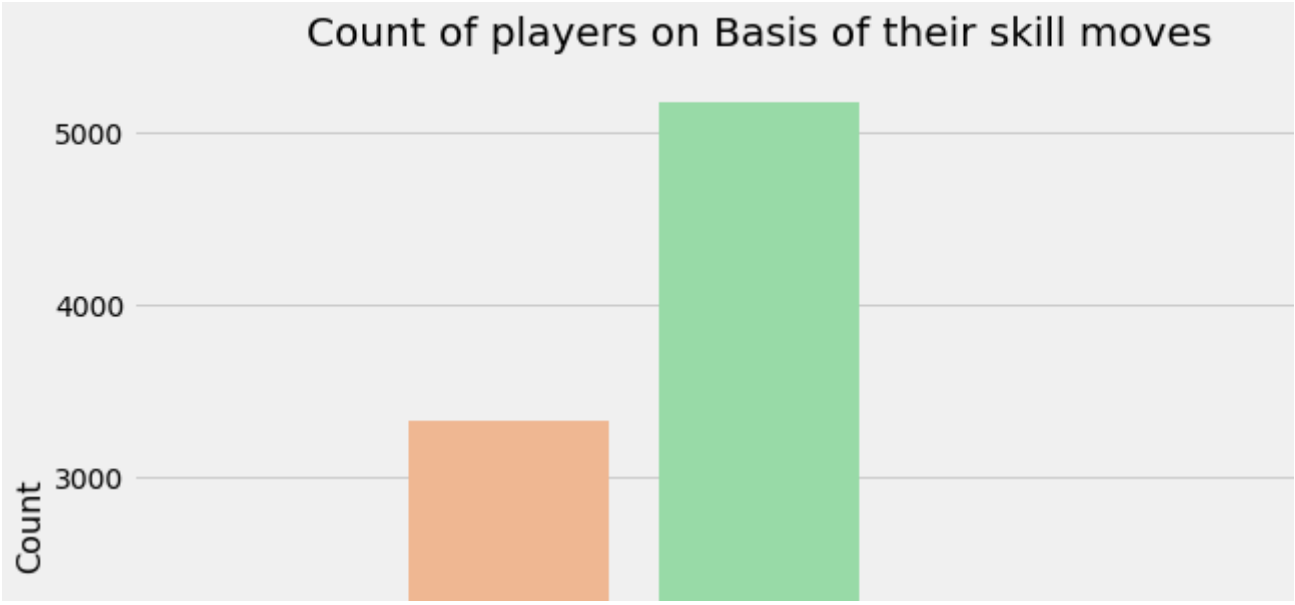
```
0    565000.0
1    405000.0
2    290000.0
3    260000.0
4    355000.0
Name: Wage, dtype: float64
```

```
# Comparing the players' Wages

import warnings
```

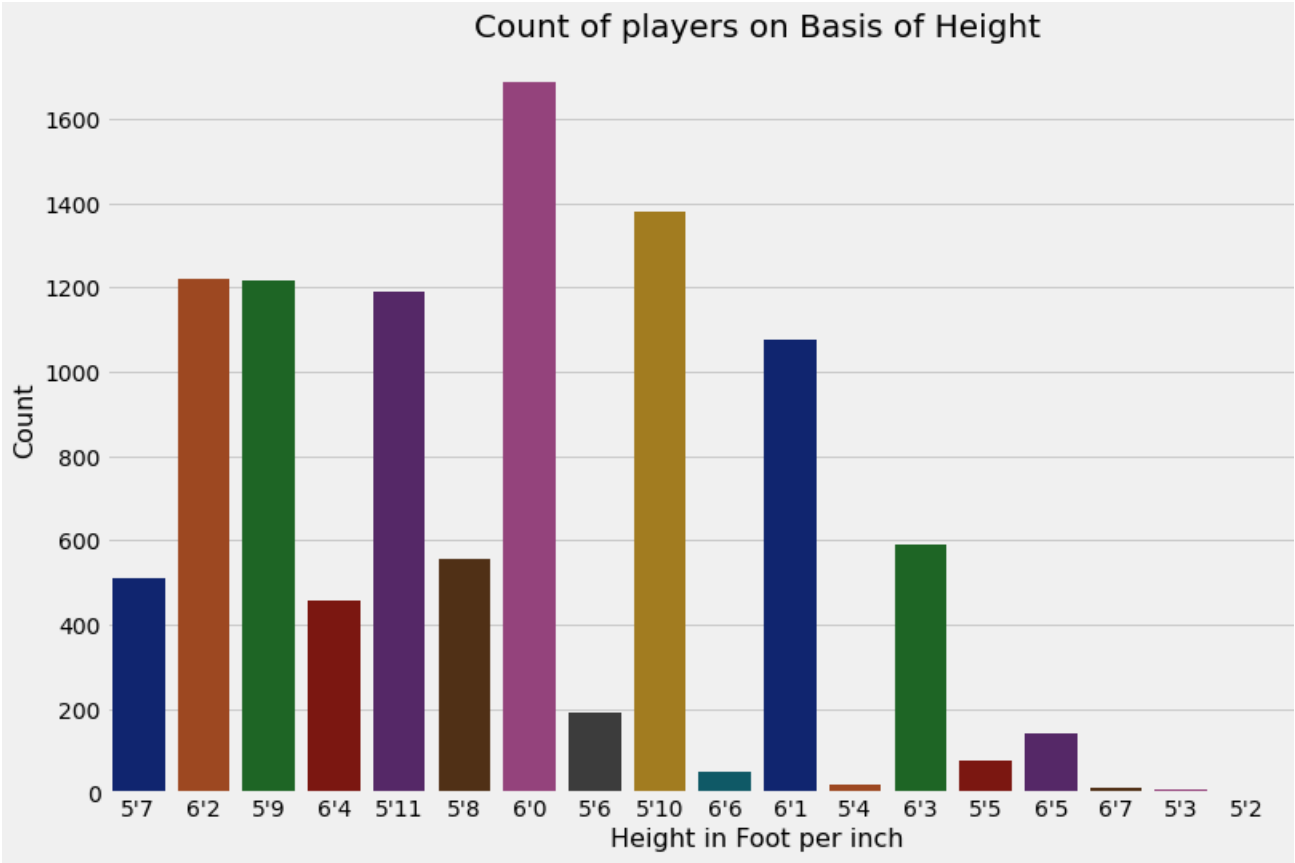
```
warnings.filterwarnings('ignore')

plt.rcParams['figure.figsize'] = (15, 5)
sns.distplot(data['Wage'], color = 'blue')
plt.xlabel('Wage Range for Players', fontsize = 16)
plt.ylabel('Count of the Players', fontsize = 16)
plt.title('Distribution of Wages of Players', fontsize = 20)
plt.xticks(rotation = 90)
plt.show()
```



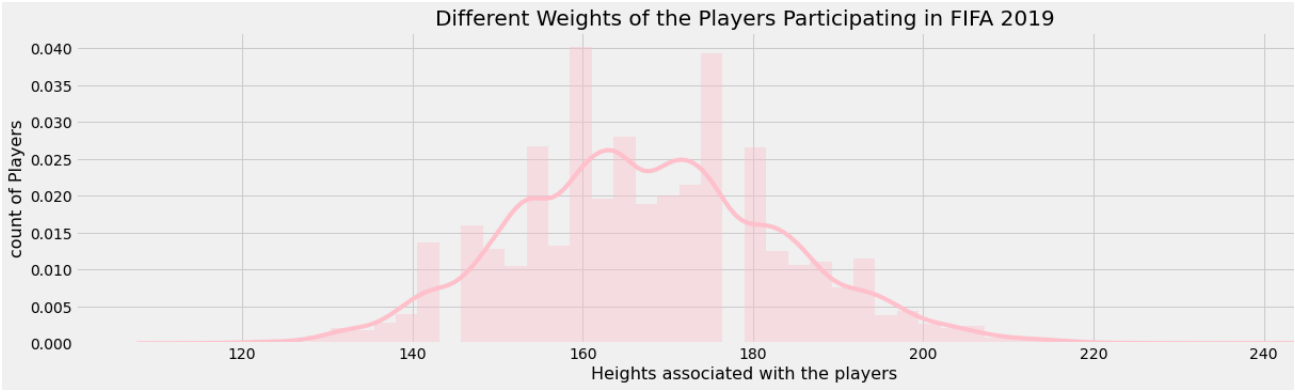

```
plt.figure(figsize = (10, 8))
ax = sns.countplot(x = 'Skill Moves', data = data, palette = 'pastel')
ax.set_title(label = 'Count of players on Basis of their skill moves', fontsize = 20)
ax.set_xlabel(xlabel = 'Number of Skill Moves', fontsize = 16)
ax.set_ylabel(ylabel = 'Count', fontsize = 16)
plt.show()
```



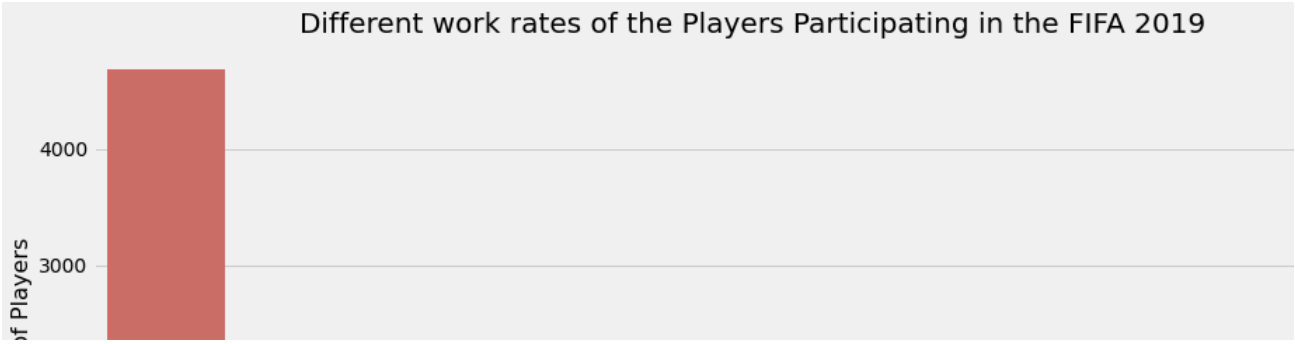

```
plt.figure(figsize = (13, 8))
ax = sns.countplot(x = 'Height', data = data, palette = 'dark')
ax.set_title(label = 'Count of players on Basis of Height', fontsize = 20)
ax.set_xlabel(xlabel = 'Height in Foot per inch', fontsize = 16)
ax.set_ylabel(ylabel = 'Count', fontsize = 16)
plt.show()
```



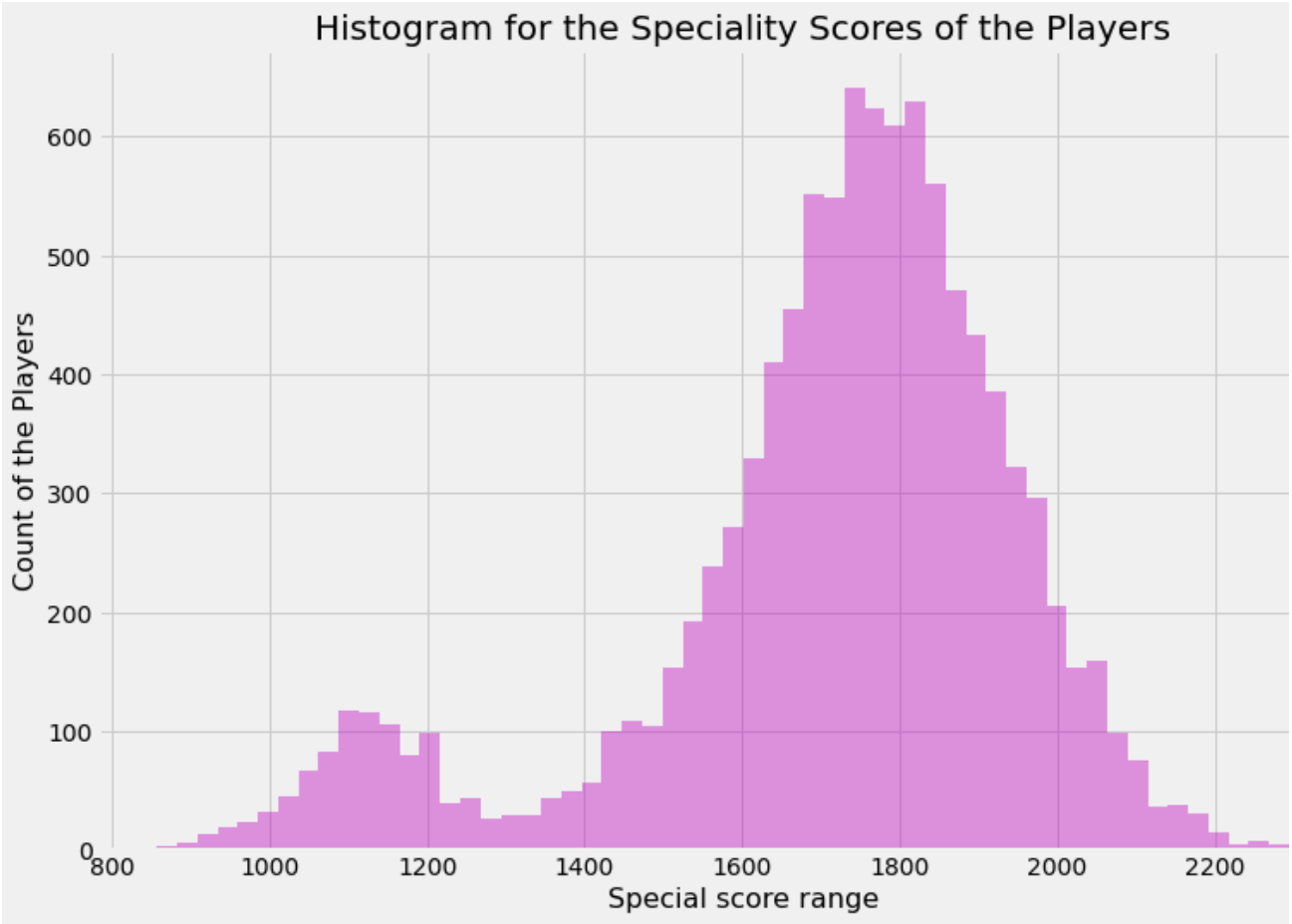
```
plt.figure(figsize = (20, 5))
sns.distplot(data['Weight'], color = 'pink')
plt.title('Different Weights of the Players Participating in FIFA 2019', fontsize = 20)
plt.xlabel('Heights associated with the players', fontsize = 16)
plt.ylabel('count of Players', fontsize = 16)
plt.show()
```




```
plt.figure(figsize = (15, 7))

sns.countplot(x = 'Work Rate', data = data, palette = 'hls')
plt.title('Different work rates of the Players Participating in the FIFA 2019', fontsize =
plt.xlabel('Work rates associated with the players', fontsize = 16)
plt.ylabel('count of Players', fontsize = 16)
plt.show()
```

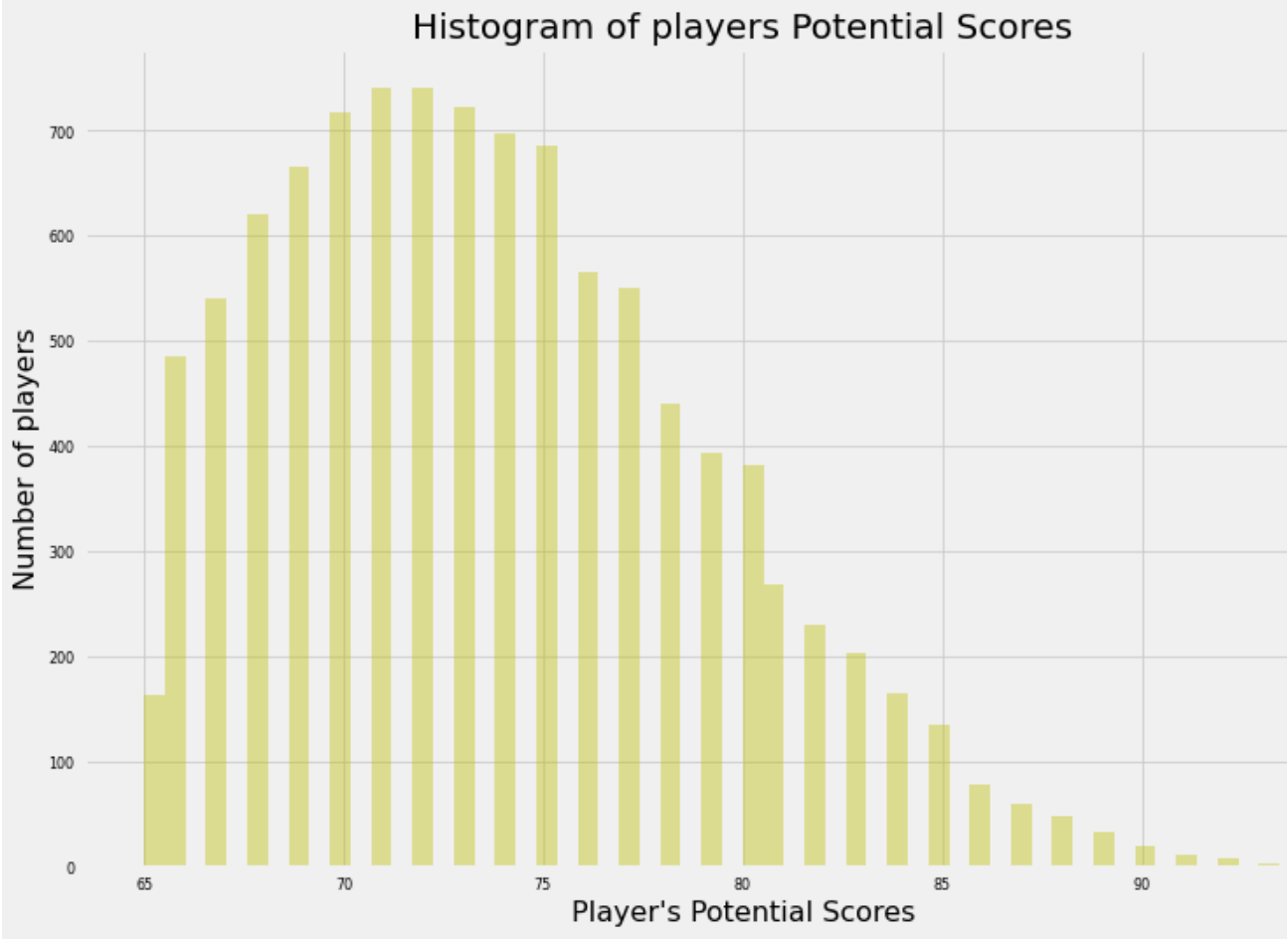



```
x = data.Special
plt.figure(figsize = (12, 8))
plt.style.use('tableau-colorblind10')

ax = sns.distplot(x, bins = 58, kde = False, color = 'm')
ax.set_xlabel(xlabel = 'Special score range', fontsize = 16)
ax.set_ylabel(ylabel = 'Count of the Players',fontsize = 16)
ax.set_title(label = 'Histogram for the Speciality Scores of the Players', fontsize = 20)
plt.show()
```

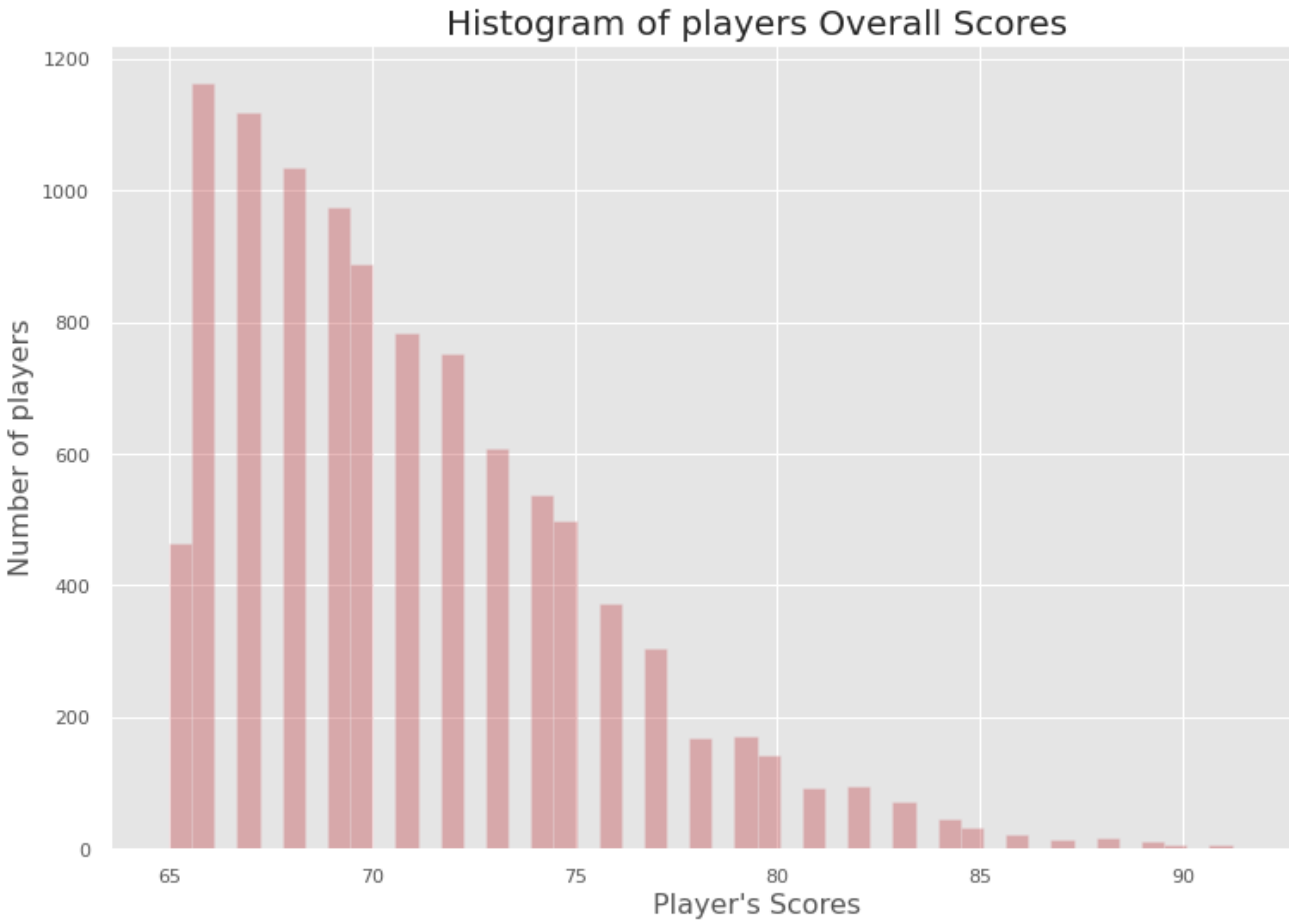



```
# To show Different potential scores of the players participating in the FIFA 2019

x = data.Potential
```

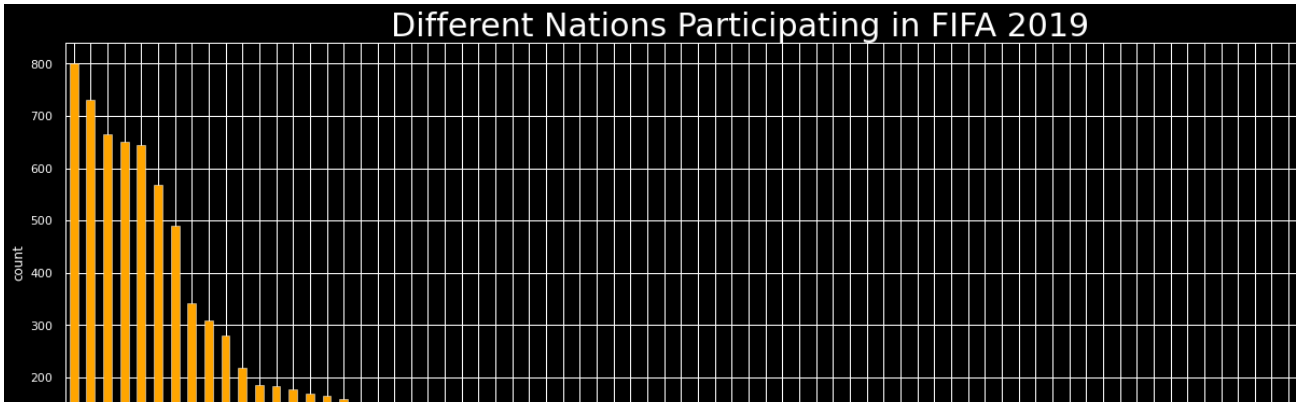
```
plt.figure(figsize=(12,8))
plt.style.use('seaborn-paper')

ax = sns.distplot(x, bins = 58, kde = False, color = 'y')
ax.set_xlabel(xlabel = "Player\'s Potential Scores", fontsize = 16)
ax.set_ylabel(ylabel = 'Number of players', fontsize = 16)
ax.set_title(label = 'Histogram of players Potential Scores', fontsize = 20)
plt.show()
```



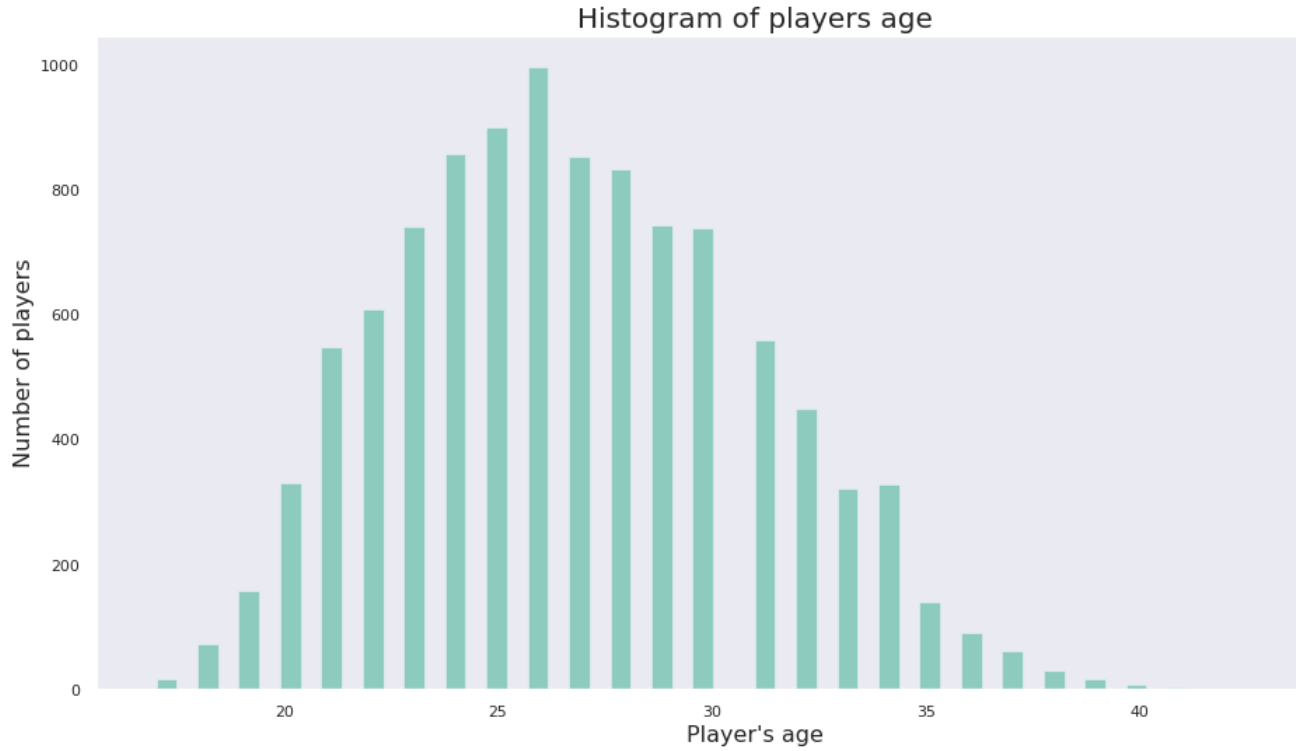

```
# To show Different overall scores of the players participating in the FIFA 2019

sns.set(style = "dark", palette = "deep", color_codes = True)
x = data.Overall
plt.figure(figsize = (12,8))
plt.style.use('ggplot')

ax = sns.distplot(x, bins = 52, kde = False, color = 'r')
ax.set_xlabel(xlabel = "Player\'s Scores", fontsize = 16)
ax.set_ylabel(ylabel = 'Number of players', fontsize = 16)
ax.set_title(label = 'Histogram of players Overall Scores', fontsize = 20)
plt.show()
```




```
# To show Different nations participating in the FIFA 2019

plt.style.use('dark_background')
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()
```

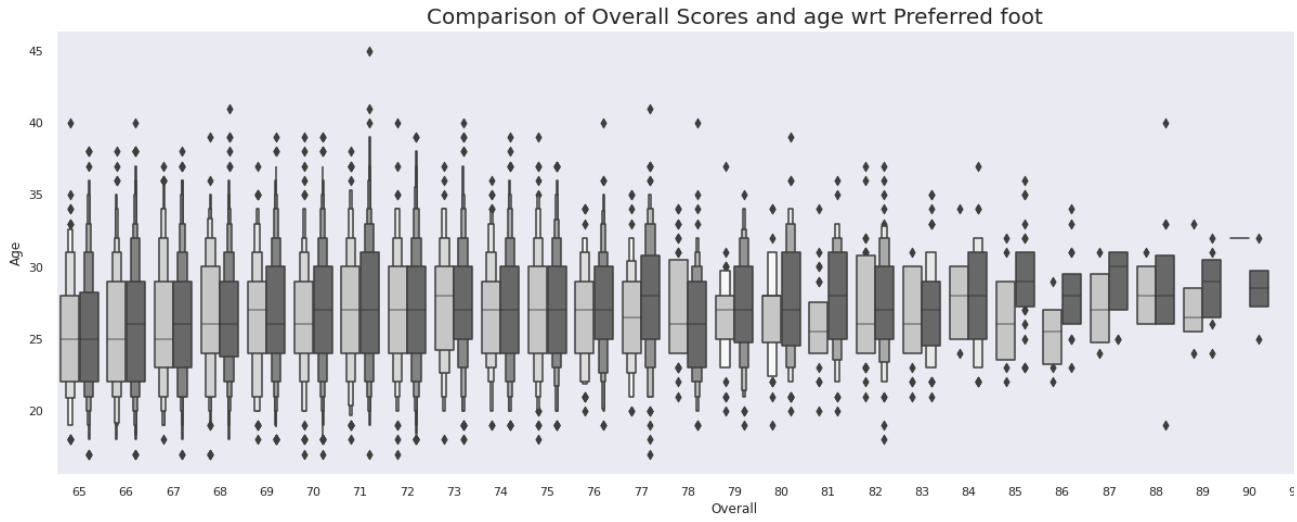



```
# To visualize age of players

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()
```




```
# violin plot
```

```
plt.rcParams['figure.figsize'] = (20, 7)
plt.style.use('seaborn-dark-palette')

sns.boxenplot(data['Overall'], data['Age'], hue = data['Preferred Foot'], palette = 'Greys
plt.title('Comparison of Overall Scores and age wrt Preferred foot', fontsize = 20)
plt.show()
```



▸ Best Players per each position with their age, club, and nationality based on their Overall Scores

[ ] ↳ *1 cell hidden*

▸ Best Players from each positions with their age, nationality, club based on their Potential Scores

[ ] ↳ *1 cell hidden*

▸ Countries with Most Players

[ ] ↳ *8 cells hidden*

▸ 15 youngest Players from the FIFA 2019

[ ] ↳ *1 cell hidden*

15 Eldest Players from FIFA 2019

[ ] ↳ *2 cells hidden*

Defining the features of players

[ ] ↳ *2 cells hidden*

Top 10 left footed footballers

[ ] ↳ *1 cell hidden*

Top 10 Right footed footballers

[ ] ↳ *2 cells hidden*

Clubs with highest number of different countries

[ ] ↳ *1 cell hidden*

Clubs with lowest number of different countries

[ ] ↳ *1 cell hidden*

Lets Create a Function to check the Player's Details

```
def playerdata(x):
    return data.loc[x,:]

x = playerdata(0)  #lionel messi, id = 0.
pd.set_option('display.max_rows', 200)
x = pd.DataFrame(x)
print(x)
```

```
                                                          0
Unnamed: 0                                                0
ID                                                   158023
Name                                               L. Messi
Age                                                      31
Photo           https://cdn.sofifa.org/players/4/19/158023.png
Nationality                                         Argentina
Flag                      https://cdn.sofifa.org/flags/52.png
Overall                                                  94
Potential                                                94
Club                                             FC Barcelona
```

Club Logo	https://cdn.sofifa.org/teams/2/light/241.png
Value	110500000.0
Wage	565000.0
Special	2202
Preferred Foot	Left
International Reputation	5
Weak Foot	4
Skill Moves	4
Work Rate	Medium/ Medium
Body Type	Messi
Real Face	Yes
Position	RF
Jersey Number	10.0
Joined	Jul 1, 2004
Loaned From	NaN
Contract Valid Until	2021
Height	5'7
Weight	159.0
LS	88+2
ST	88+2
RS	88+2
LW	92+2
LF	93+2
CF	93+2
RF	93+2
RW	92+2
LAM	93+2
CAM	93+2
RAM	93+2
LM	91+2
LCM	84+2
CM	84+2
RCM	84+2
RM	91+2
LWB	64+2
LDM	61+2
CDM	61+2
RDM	61+2
RWB	64+2
LB	59+2
LCB	47+2
CB	47+2
RCB	47+2
RB	59+2
Crossing	84
Finishing	95

Correlation heatmap

[ ] ↳ *1 cell hidden*

Modelling

```
data = pd.read_csv('/content/sample_data/data.csv')
```

```
team = data.groupby('Club',as_index=False)['Overall','Potential','Crossing','Finishing','H
team.sort_values(by='Club', ascending=True, inplace=True)
team1 = team.sort_values(by='Club', ascending=False, inplace=False)

import numpy as np
col = [0]*100
ov1 = team['Overall'].head(100).values
ov2 = team1['Overall'].head(100).values


ovt1 = team['Overall'].values
ovt2 = team1['Overall'].values
col1 = [0]*100
for i in range(100):
    col1[i]=ovt1[i]-ovt2[i]
Y_train1 = col1




for i in range(100):
    if ov1[i]>ov2[i]:
        col[i]=1
    else:
        col[i]=0

temp = pd.DataFrame({'Overall1': ov1, 'Overall2': ov2,'WinLoss': col}, columns=['Overall1'


X_train = temp[['Overall2','Overall1']].values
Y_train = temp['WinLoss']
print(X_train.shape)
print(Y_train[1])
```

```
(100, 2)
1
```

Logistic Regression

```
import matplotlib.pyplot as plt
import numpy as np

fig = plt.figure(figsize=(10, 10))
ax = fig.add_subplot(1, 1, 1)

ax.axis([np.min(X_train[:,0])-1., np.max(X_train[:,0])+1., np.min(X_train[:,1])-1., np.max
ax.set_xlabel('2nd half teams')
ax.set_ylabel('1st half teams')
pos = np.where(Y_train.loc[:,0] == 1)[0] #storing in array if it satisfies Y_train[:,0] ==
neg = np.where(Y_train.loc[:,0] == 0)[0] #storing in array if it satisfies y_train[:,0] ==
ax.plot(X_train[pos,0], X_train[pos,1], marker='.', color='#0F00FF', markersize=10, linest
ax.plot(X_train[neg,0], X_train[neg,1], marker='.', color='#FF00AE', markersize=10, linest
```

```
    Y_train=Y_train[1]
    ax.legend()
```

```
---------------------------------------------------------------------------
IndexingError                             Traceback (most recent call last)
<ipython-input-60-22990d043800> in <module>
```

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(penalty='none', max_iter=500, solver='lbfgs')
model.fit(X_train, Y_train.values.flatten())

Y_pred = model.predict_proba(X_train)
print("Y_pred:",Y_pred)
```

```
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 2.68958626e-047]
[0.00000000e+000 1.00000000e+000]

[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[9.99999942e-001 5.78720270e-008]
[0.00000000e+000 1.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[1.00000000e+000 0.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
[0.00000000e+000 1.00000000e+000]
```

```
 [1.00000000e+000 0.00000000e+000]
 [0.00000000e+000 1.00000000e+000]
 [0.00000000e+000 1.00000000e+000]
 [0.00000000e+000 1.00000000e+000]
 [1.00000000e+000 0.00000000e+000]
 [0.00000000e+000 1.00000000e+000]
 [1.00000000e+000 0.00000000e+000]
 [0.00000000e+000 1.00000000e+000]]
```

```
Y_pred_label = model.predict(X_train)
print("predicted label",Y_pred_label)
```

```
predicted label [1 1 1 0 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 0 0 1 0 1 0 1 0 1 1
 1 0 1 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1
 1 1 0 1 1 1 0 0 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 1 0 1]
```

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(penalty='none', max_iter=500, solver='lbfgs')
model.fit(X_train, Y_train.values.flatten())

Y_pred = model.predict_proba(X_train)
print("Y_pred:",Y_pred)


from sklearn.metrics import log_loss


log_loss(Y_train, Y_pred)
```

```
8.369562477367947e-10
```

```
from sklearn.metrics import accuracy_score


accuracy_score(Y_train, Y_pred_label)
```

```
1.0
```

```
team = data.groupby('Club',as_index=False)['Overall'].mean()
team

#label encoding
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
team['Club']= label_encoder.fit_transform(team['Club'])
team
```

	Club	Overall
0	0	65.586207
1	1	65.750000
2	2	63.384615
3	3	70.785714
4	4	65.615385
...	...	...
646	646	60.760000
647	647	66.900000
648	648	60.481481
649	649	63.545455

```
from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(team, team.Overall, test_size=0.2)
y_train
```

```
475    69.766667
460    65.285714
213    68.777778
264    70.733333
625    61.769231
         ...
232    76.678571
440    65.423077
327    60.866667
424    62.428571
527    72.296296
Name: Overall, Length: 520, dtype: float64
```

```
lab = preprocessing.LabelEncoder()
y_transformed = lab.fit_transform(y_train)

lab = preprocessing.LabelEncoder()
y_transformed_test = lab.fit_transform(y_test)
```

Logistic reg

```
team['Club']
```

```
0        0
1        1
2        2
3        3
4        4
      ...
```

```
646    646
647    647
648    648
649    649
650    650
Name: Club, Length: 651, dtype: int64
```

```
model = LogisticRegression(solver='liblinear', random_state=0)
model.fit(X_train,y_transformed)
```

```
LogisticRegression(random_state=0, solver='liblinear')
```

```
model.predict(X_test)
```

```
array([ 88, 145,  88,  88, 314, 145,  88,  88, 179, 314, 221, 314, 145,
       145,  88,  88, 145, 314, 314,  88, 179, 145, 314, 314,  88, 314,
        88,  88, 179, 145, 109, 145, 314, 145,  88,  88,  88,  88,  88,
       314, 314,  88, 145,  88,  88,  88,  88,  88, 145, 314,  88,  88,
       145, 314,  88, 139, 179,  88,  88,  88, 145, 145, 314, 145, 145,
       314, 314, 109, 179,  88,  88, 314,  88,  88, 314, 179, 314, 145,
        88, 314,  88,  88, 225, 314, 145, 109, 314,  88,  88, 109,  88,
       109, 145, 314, 314, 145,   7, 145, 145, 314, 314,  88, 109,  88,
        88,  88, 314,  88,  88,  88, 109, 145, 314, 109, 145, 314, 145,
       145,  88, 314, 314, 314,  88, 314, 109,  88, 314, 314,  88,  88,
        88])
```

KNN

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=7)

knn.fit(X_train, y_transformed)

# Predict on dataset which model has not seen before
print(knn.predict(X_test))
```

```
[ 19 206  66  83  88 191 104 205 169 218 139 202  52 112  43  88 112 202
 278 101 165  56   8  24  46 211 172 317 149 122  16  52  77 142  48   6
  66  15 234 125  77 137 142 202 163  92 173 259 212  61  21  76 186 135
  58 139 190  19  83 196  64  52 222 145 251 236 304  80  20  22  74 135
 419  38  84  33 254 271  57  24 256  46 139 134  74  26 279  15  15  26
  27  51 142  17  16 145  34  60 145 117  61 135  26  30 104  12  24 241
  43 113   7 112 202  16  11 132 251  52 110  24  59  59  22  31  26  11
 211 195  47  57 206]
```

```
print(knn.score(X_test, y_transformed_test))
```

```
0.007633587786259542
```

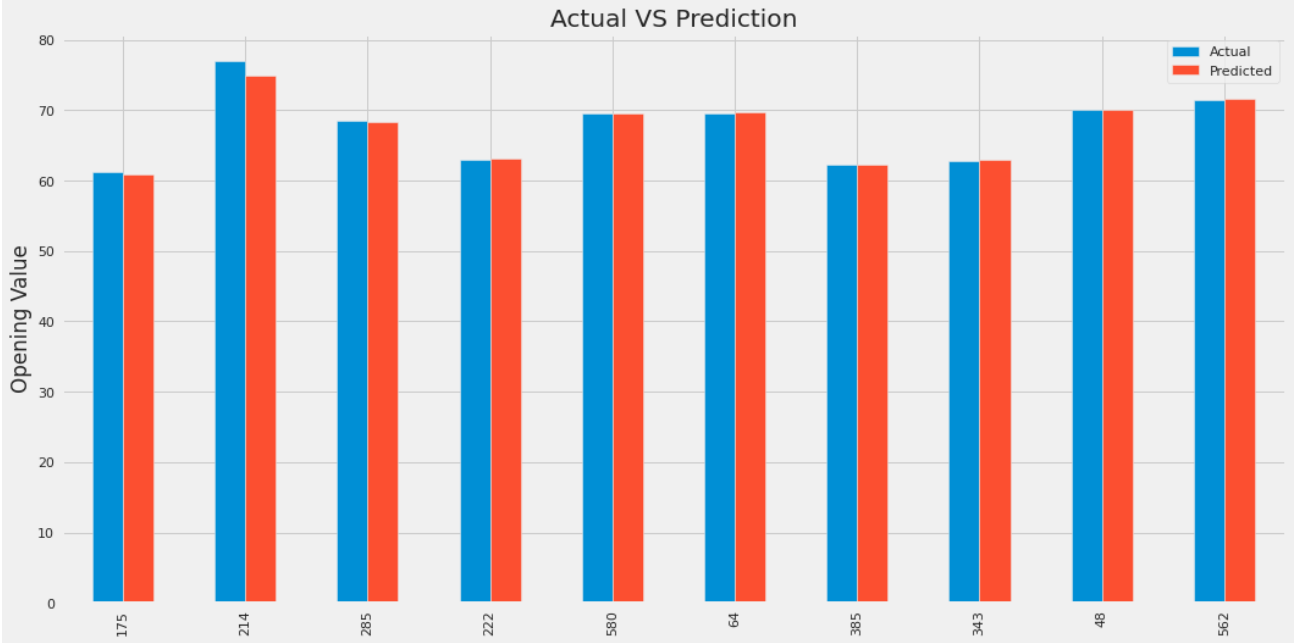
Decision Tree

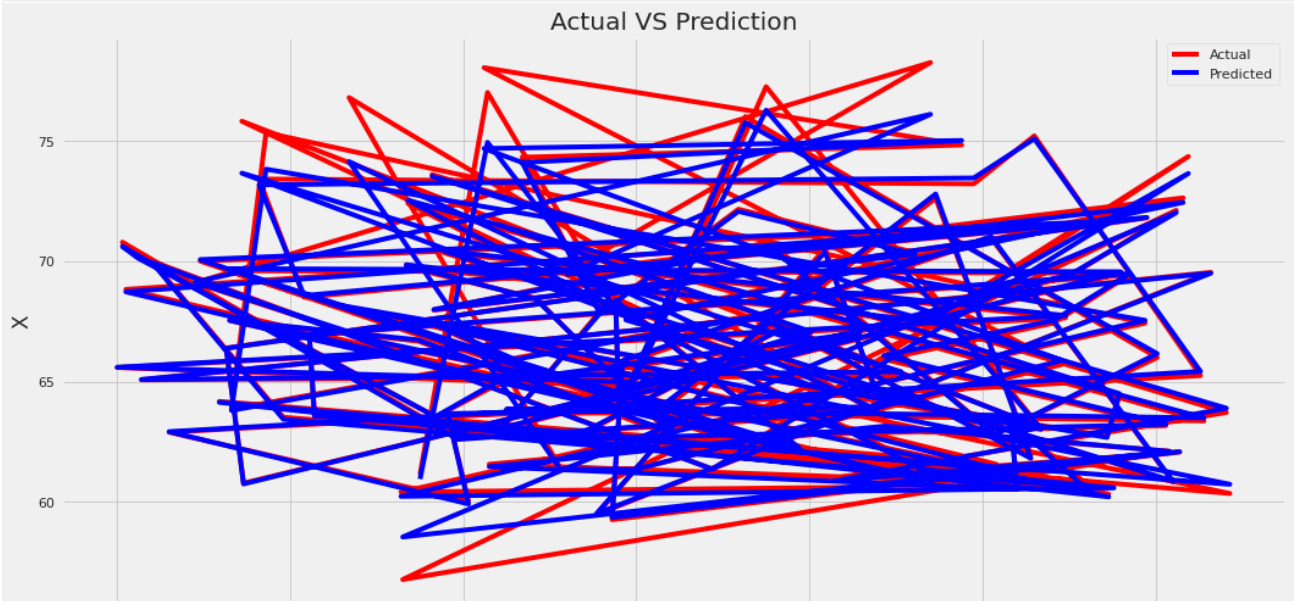
```
!pip install scikit-plot
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Requirement already satisfied: scikit-plot in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: matplotlib>=1.4.0 in /usr/local/lib/python3.8/dist-pa
Requirement already satisfied: joblib>=0.10 in /usr/local/lib/python3.8/dist-package
Requirement already satisfied: scikit-learn>=0.18 in /usr/local/lib/python3.8/dist-p
Requirement already satisfied: scipy>=0.9 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-pa
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/loca
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-package
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (f
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist
```

```
from sklearn.tree import DecisionTreeRegressor
reg = DecisionTreeRegressor(random_state=42)
reg.fit(X_train, y_transformed)
y_pred=reg.predict(np.array(X_test))
df_preds2 = pd.DataFrame({'Actual': y_test.squeeze(), 'Predicted': y_pred.squeeze()})
Test(y_pred,y_test,df_preds)
```

```
Mean Absolute Error: 178.86424171546898
Mean Squared Error: 48115.24064604512
Root Mean Squared Error: 219.35186492493088
R Square Error: -1.477743149072865
```






```
from sklearn import tree


clf = tree.DecisionTreeClassifier(random_state=42,max_depth=5)
clf = clf.fit(X_train, y_transformed)

tree.plot_tree(clf)
```

```
[Text(0.41517857142857145, 0.9166666666666666, 'X[1] <= 64.207\ngini =
0.998\nsamples = 520\nvalue = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1\n1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 2, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 1, 1,
1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2\n1, 1, 1, 1, 4, 1, 2, 1, 1, 2, 1, 1, 1, 1\n1, 1, 1,
1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n2, 1, 1,
1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 1\n1, 1, 1, 1, 2, 3, 1, 1, 1, 1, 2, 2, 1, 2\n1, 1, 1,
2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n1, 1, 1,
2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2\n1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 1, 1, 1\n2, 1, 1,
1, 1, 1, 1, 2, 1, 1, 2, 1, 2, 1\n2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 3, 1, 1, 1\n1, 1, 1,
1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1,
1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 3, 2, 1, 1, 1, 1, 1, 1\n1, 1, 1,
1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1\n1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n2, 1,
1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 2,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1\n1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]'),
 Text(0.21428571428571427, 0.75, 'X[1] <= 64.171\ngini = 0.993\nsamples =
186\nvalue = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1\n1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1\n2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 2, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 1, 1, 1, 2, 1, 1, 1, 1,
1, 1, 1, 1, 2\n1, 1, 1, 1, 4, 1, 2, 1, 1, 2, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 2, 1, 1,
1, 1, 2, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n2, 1, 1, 1, 1, 2, 2, 1, 2,
1, 1, 1, 1, 1\n1, 1, 1, 1, 2, 3, 1, 1, 1, 1, 2, 2, 1, 2\n1, 1, 1, 2, 2, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.17857142857142858, 0.5833333333333334, 'X[1] <= 64.129\ngini =
0.993\nsamples = 184\nvalue = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1\n1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 2, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 1, 1,
1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2\n1, 1, 1, 1, 4, 1, 2, 1, 1, 2, 1, 1, 1, 1\n1, 1, 1,
1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n2, 1, 1,
1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 1\n1, 1, 1, 1, 2, 3, 1, 1, 1, 1, 2, 2, 1, 2\n1, 1, 1,
2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.14285714285714285, 0.4166666666666667, 'X[1] <= 63.576\ngini =
0.993\nsamples = 182\nvalue = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1\n1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 2, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 1, 1,
1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2\n1, 1, 1, 1, 4, 1, 2, 1, 1, 2, 1, 1, 1, 1\n1, 1, 1,
1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n2, 1, 1,
```

```
1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 1\n1, 1, 1, 1, 2, 3, 1, 1, 1, 1, 2, 2, 1, 2\n1, 1, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.07142857142857142, 0.25, 'X[1] <= 63.146\ngini = 0.992\nsamples =
163\nvalue = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1\n2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 2, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 1, 1, 1, 2, 1, 1, 1, 1,
1, 1, 1, 1, 2\n1, 1, 1, 1, 4, 1, 2, 1, 1, 2, 1, 1, 1, 1\n1, 1, 1, 1, 1, 2, 1, 1,
1, 1, 2, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n2, 1, 1, 1, 1, 2, 2, 1, 2,
1, 1, 1, 1, 1\n1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.03571428571428571, 0.08333333333333333, 'gini = 0.991\nsamples = 147\nvalue
= [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1\n2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1\n1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1,
2\n1, 1, 1, 1, 4, 1, 2, 1, 1, 2, 1, 1, 1, 1\n1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1,
1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.10714285714285714, 0.08333333333333333, 'gini = 0.914\nsamples = 16\nvalue
= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 2, 2, 1, 2, 1, 1, 1, 1,
1\n1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.21428571428571427, 0.25, 'X[1] <= 63.683\ngini = 0.909\nsamples = 19\nvalue
= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 2, 3, 1, 1, 1, 1, 2, 2, 1, 2\n1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.17857142857142858, 0.08333333333333333, 'gini = 0.48\nsamples = 5\nvalue =
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.25, 0.08333333333333333, 'gini = 0.898\nsamples = 14\nvalue = [0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 1, 1, 1, 1, 2, 2, 1, 2\n1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0]'),
```

```
 Text(0.21428571428571427, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue =
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.25, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.6160714285714286, 0.75, 'X[1] <= 66.017\ngini = 0.996\nsamples = 334\nvalue
= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2\n1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 1, 1, 1\n2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 2,
1\n2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 3, 1, 1, 1\n1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1,
1\n1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1,
1\n1, 1, 1, 1, 1, 1, 3, 2, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1,
1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 2, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1,
1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]'),
 Text(0.4642857142857143, 0.5833333333333334, 'X[1] <= 65.983\ngini =
0.985\nsamples = 91\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
```

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n1, 1, 1, 2,
1, 1, 1, 1, 1, 1, 1, 1, 1, 2\n1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 1, 1, 1\n2, 1, 1, 1,
1, 1, 1, 2, 1, 1, 2, 1, 2, 1\n2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 3, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.42857142857142855, 0.4166666666666667, 'X[1] <= 65.58\ngini =
0.985\nsamples = 88\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 1, 1, 1, 1, 1, 1, 1, 1, 1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n1, 1, 1, 2,
1, 1, 1, 1, 1, 1, 1, 1, 1, 2\n1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 1, 1, 1\n2, 1, 1, 1,
1, 1, 1, 2, 1, 1, 2, 1, 2, 1\n2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.35714285714285715, 0.25, 'X[1] <= 65.276\ngini = 0.981\nsamples = 70\nvalue
= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2\n1, 1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 1, 1, 1\n2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.32142857142857145, 0.08333333333333333, 'gini = 0.976\nsamples = 47\nvalue
= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
1\n1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1\n1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2\n1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.39285714285714285, 0.08333333333333333, 'gini = 0.926\nsamples = 23\nvalue
= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 3, 1, 3, 1, 1, 1, 1\n2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.5, 0.25, 'X[1] <= 65.604\ngini = 0.92\nsamples = 18\nvalue = [0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 2, 1\n2, 1, 1, 1,
1, 2, 1, 1, 1, 1, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.4642857142857143, 0.08333333333333333, 'gini = 0.0\nsamples = 2\nvalue =
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.5357142857142857, 0.08333333333333333, 'gini = 0.914\nsamples = 16\nvalue =
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2,
1\n2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.5, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
```

SVM

```
import numpy as np
from sklearn.svm import SVR
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
rng = np.random.RandomState(42)
y = np.array(y_train)
X = np.array(X_train)
regr = make_pipeline(StandardScaler(), SVR(C=1.0, epsilon=0.2))
regr.fit(X, y)
y_pred=regr.predict(np.array(X_test))
df_preds = pd.DataFrame({'Actual': y_test.squeeze(), 'Predicted': y_pred.squeeze()})
Test(y_pred,y_test,df_preds)
```

```
Mean Absolute Error: 0.22086554342995765
Mean Squared Error: 0.30856980536598366
Root Mean Squared Error: 0.5554905988097222
R Square Error: 0.984507854631017
```