In [1]:

```
import numpy as np
import pandas as pd
from collections import Counter
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
df_tip=pd.read_csv("tips.csv")
```

In [3]:

```
df_tip
```

Out[3]:

	total_bill	tip	gender	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

In [17]:

```
df_tip['tip'].value_counts()
```

Out[17]:

```
2.00    33
3.00    23
4.00    12
5.00    10
2.50    10
        ..
4.34     1
1.56     1
5.20     1
2.60     1
1.75     1
Name: tip, Length: 123, dtype: int64
```

In [4]:

```
df_tip.columns
```

Out[4]:

```
Index(['total_bill', 'tip', 'gender', 'smoker', 'day', 'time', 'size'], dtype
='object')
```

In [5]:

```
df_tip.isnull().sum()
```

Out[5]:

```
total_bill    0
tip           0
gender        0
smoker        0
day           0
time          0
size          0
dtype: int64
```

In [6]:

```
df_tip['day'].value_counts()
```

Out[6]:

```
Sat     87
Sun     76
Thur    62
Fri     19
Name: day, dtype: int64
```

In [7]:

```
df_tip['gender'].value_counts()
```

Out[7]:

```
Male      157
Female     87
Name: gender, dtype: int64
```

In [8]:

```
pd.crosstab(df_tip['gender'],df_tip['smoker'].isnull())
```

Out[8]:

smoker	**False**
gender	
Female	87
Male	157

In [9]:

```
df_tip['smoker'].value_counts()
```

Out[9]:

```
No     151
Yes     93
Name: smoker, dtype: int64
```

In [10]:

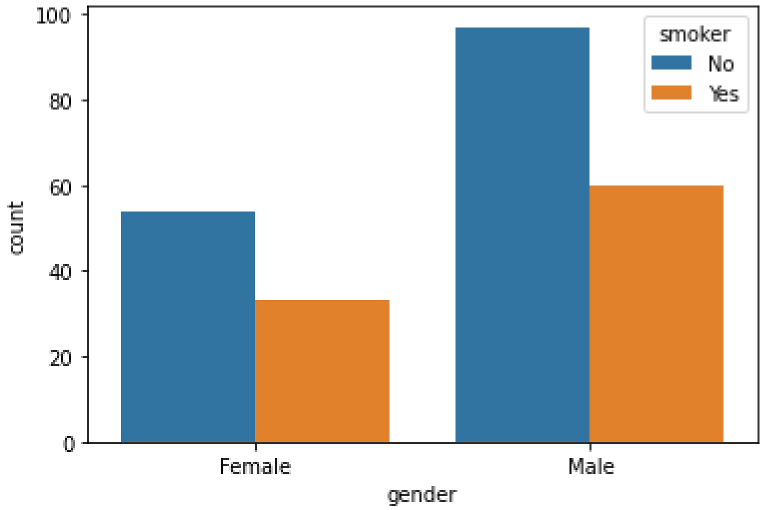
```
df_tip['time'].value_counts()
```

Out[10]:

```
Dinner    176
Lunch      68
Name: time, dtype: int64
```

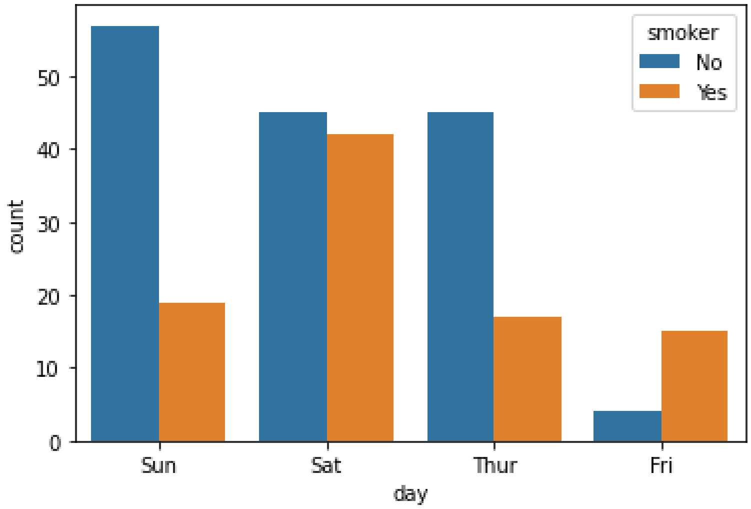
```
In [11]: sns.countplot('gender',hue='smoker',data=df_tip)
```

```
Out[11]: <AxesSubplot:xlabel='gender', ylabel='count'>
```



```
In [12]: sns.countplot('day',hue='smoker',data=df_tip)
```

```
Out[12]: <AxesSubplot:xlabel='day', ylabel='count'>
```

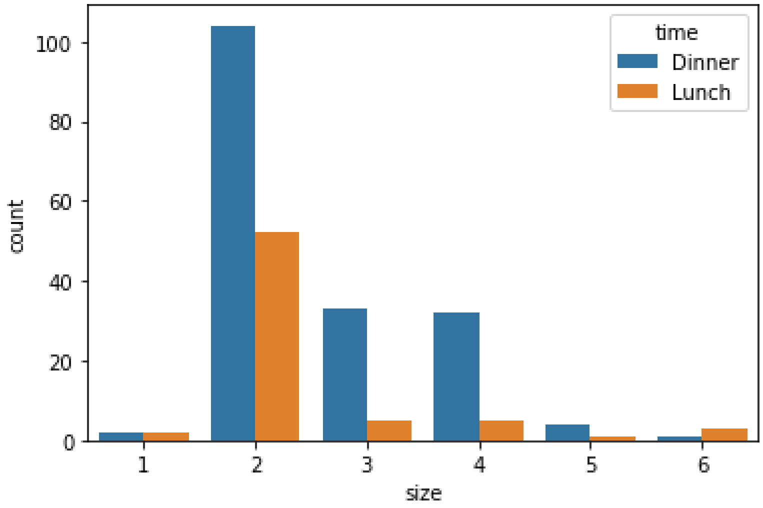


```
In [13]: df_tip['size'].value_counts()
```

```
Out[13]: 2    156
         3     38
         4     37
         5      5
         1      4
         6      4
         Name: size, dtype: int64
```

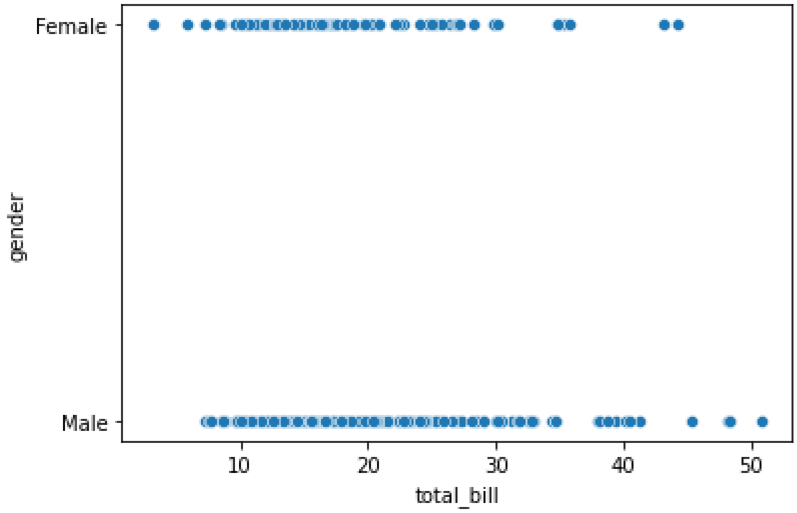
```
In [14]: sns.countplot('size',hue='time',data=df_tip)
```

```
Out[14]: <AxesSubplot:xlabel='size', ylabel='count'>
```



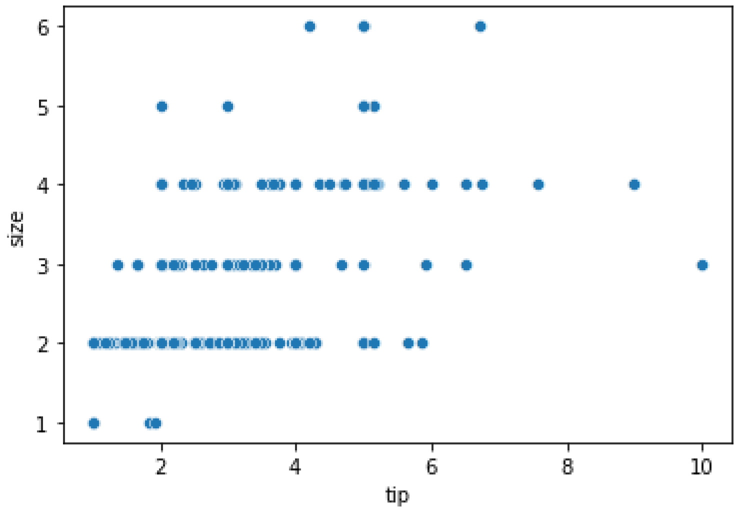
```
In [18]: sns.scatterplot(y='gender',x='total_bill',data=df_tip)
```

```
Out[18]: <AxesSubplot:xlabel='total_bill', ylabel='gender'>
```



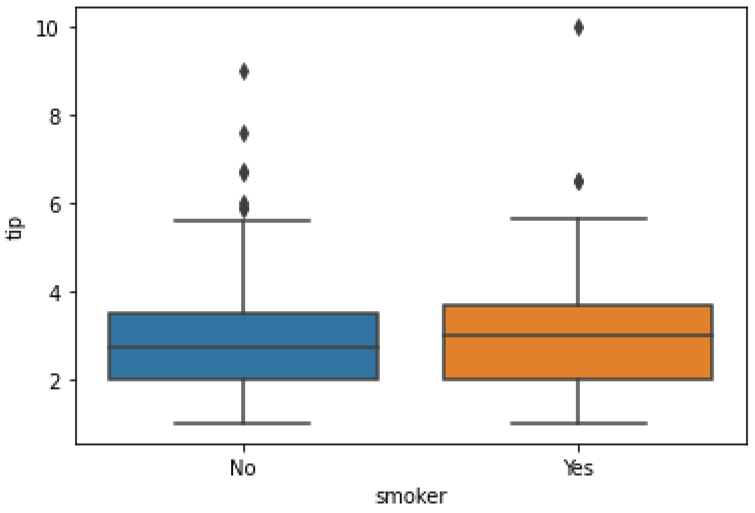
In [20]:
```
sns.scatterplot(y='size',x='tip',data=df_tip)
```

Out[20]:
```
<AxesSubplot:xlabel='tip', ylabel='size'>
```



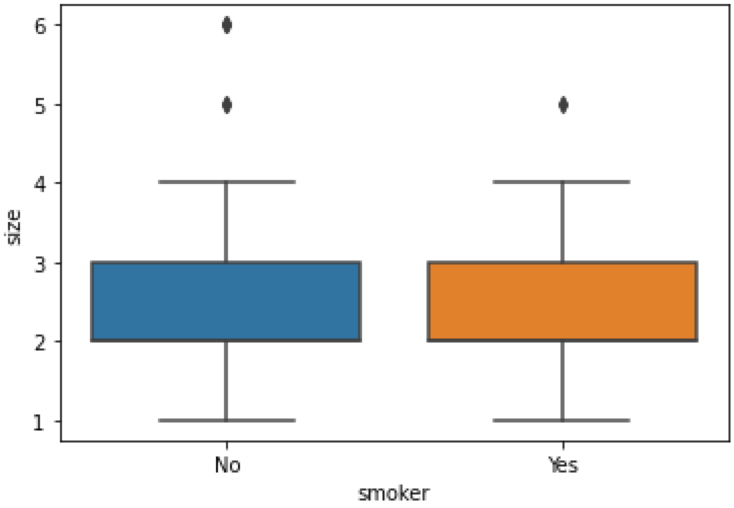
In [22]:
```
sns.boxplot(x = "smoker",y = "tip", data =df_tip)
```

Out[22]:
```
<AxesSubplot:xlabel='smoker', ylabel='tip'>
```



In [24]:
```
sns.boxplot(x = "smoker",y = "size", data = df_tip)
```

Out[24]:
```
<AxesSubplot:xlabel='smoker', ylabel='size'>
```



In [28]:
```
df_tip['smoker'].replace(['No','Yes'],[0,1],inplace=True)
```

In [29]:
```
df_tip
```

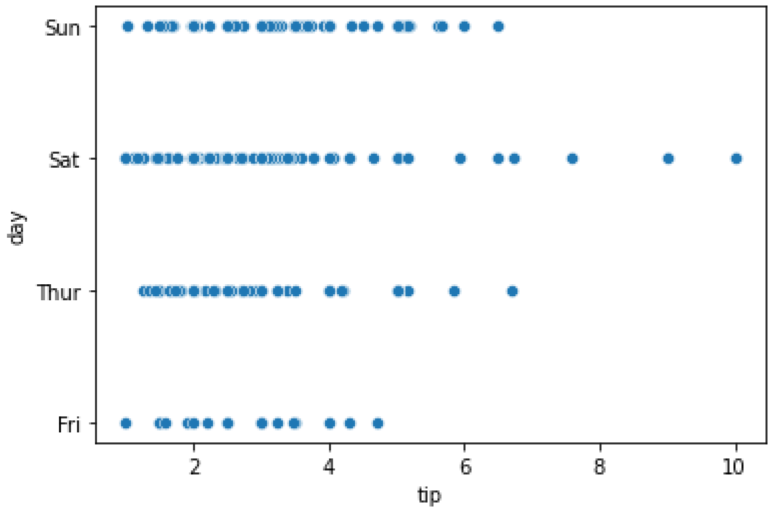
Out[29]:

	total_bill	tip	gender	smoker	day	time	size
0	16.99	1.01	Female	0	Sun	Dinner	2
1	10.34	1.66	Male	0	Sun	Dinner	3
2	21.01	3.50	Male	0	Sun	Dinner	3
3	23.68	3.31	Male	0	Sun	Dinner	2
4	24.59	3.61	Female	0	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	0	Sat	Dinner	3
240	27.18	2.00	Female	1	Sat	Dinner	2
241	22.67	2.00	Male	1	Sat	Dinner	2
242	17.82	1.75	Male	0	Sat	Dinner	2
243	18.78	3.00	Female	0	Thur	Dinner	2

244 rows × 7 columns

```
In [38]: sns.scatterplot(y='day',x='tip',data=df_tip)
```

```
Out[38]: <AxesSubplot:xlabel='tip', ylabel='day'>
```



Make data cloumns is numerical

```
In [39]: df_tip
```

Out[39]:

	total_bill	tip	gender	smoker	day	time	size
0	16.99	1.01	Female	0	Sun	Dinner	2
1	10.34	1.66	Male	0	Sun	Dinner	3
2	21.01	3.50	Male	0	Sun	Dinner	3
3	23.68	3.31	Male	0	Sun	Dinner	2
4	24.59	3.61	Female	0	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	0	Sat	Dinner	3
240	27.18	2.00	Female	1	Sat	Dinner	2
241	22.67	2.00	Male	1	Sat	Dinner	2
242	17.82	1.75	Male	0	Sat	Dinner	2
243	18.78	3.00	Female	0	Thur	Dinner	2

244 rows × 7 columns

In [40]:
```
df_tip['gender'].replace(['Female','Male'],[0,1],inplace=True)
```

In [41]:
```
df_tip
```

Out[41]:

	total_bill	tip	gender	smoker	day	time	size
0	16.99	1.01	0	0	Sun	Dinner	2
1	10.34	1.66	1	0	Sun	Dinner	3
2	21.01	3.50	1	0	Sun	Dinner	3
3	23.68	3.31	1	0	Sun	Dinner	2
4	24.59	3.61	0	0	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	1	0	Sat	Dinner	3
240	27.18	2.00	0	1	Sat	Dinner	2
241	22.67	2.00	1	1	Sat	Dinner	2
242	17.82	1.75	1	0	Sat	Dinner	2
243	18.78	3.00	0	0	Thur	Dinner	2

244 rows × 7 columns

In [43]:
```
df_tip['day'].value_counts()
```

Out[43]:
```
Sat     87
Sun     76
Thur    62
Fri     19
Name: day, dtype: int64
```

In [44]:
```
df_tip['day'].replace(['Sun','Sat','Thur','Fri'],[0,1,2,3],inplace=True)
```

In [45]:
```
df_tip
```

Out[45]:

	total_bill	tip	gender	smoker	day	time	size
0	16.99	1.01	0	0	0	Dinner	2
1	10.34	1.66	1	0	0	Dinner	3
2	21.01	3.50	1	0	0	Dinner	3
3	23.68	3.31	1	0	0	Dinner	2
4	24.59	3.61	0	0	0	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	1	0	1	Dinner	3
240	27.18	2.00	0	1	1	Dinner	2
241	22.67	2.00	1	1	1	Dinner	2
242	17.82	1.75	1	0	1	Dinner	2
243	18.78	3.00	0	0	2	Dinner	2

244 rows × 7 columns

In [46]:
```
df_tip['time'].value_counts()
```

Out[46]:
```
Dinner    176
Lunch      68
Name: time, dtype: int64
```

In [47]:
```
df_tip['time'].replace(['Dinner','Lunch'],[0,1],inplace=True)
```

```
In [48]: df_tip
```

Out[48]:

	total_bill	tip	gender	smoker	day	time	size
0	16.99	1.01	0	0	0	0	2
1	10.34	1.66	1	0	0	0	3
2	21.01	3.50	1	0	0	0	3
3	23.68	3.31	1	0	0	0	2
4	24.59	3.61	0	0	0	0	4
...	...	...	...	...	...	...	...
239	29.03	5.92	1	0	1	0	3
240	27.18	2.00	0	1	1	0	2
241	22.67	2.00	1	1	1	0	2
242	17.82	1.75	1	0	1	0	2
243	18.78	3.00	0	0	2	0	2

244 rows × 7 columns

```
In [49]: #df_tip.describe()
```

Out[49]:

	total_bill	tip	gender	smoker	day	time	size
count	244.000000	244.000000	244.000000	244.000000	244.000000	244.000000	244.000000
mean	19.785943	2.998279	0.643443	0.381148	1.098361	0.278689	2.569672
std	8.902412	1.383638	0.479967	0.486667	0.933244	0.449276	0.951100
min	3.070000	1.000000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	13.347500	2.000000	0.000000	0.000000	0.000000	0.000000	2.000000
50%	17.795000	2.900000	1.000000	0.000000	1.000000	0.000000	2.000000
75%	24.127500	3.562500	1.000000	1.000000	2.000000	1.000000	3.000000
max	50.810000	10.000000	1.000000	1.000000	3.000000	1.000000	6.000000

Scaling OF Data

```
In [50]: #from sklearn.preprocessing import StandardScaler
         #scaler=StandardScaler()
```

```
In [51]: #scaler.fit(df_tip.drop('smoker',axis=1))
```

```
Out[51]: StandardScaler()
```

```
In [52]: #scaled_features=scaler.transform(df_tip.drop('smoker',axis=1))
```

In [53]:
```
#scaled_features
```

Out[53]:
```
array([[-3.14711305e-01, -1.43994695e+00, -1.34335316e+00,
         -1.17934719e+00, -6.21581561e-01, -6.00192629e-01],
       [-1.06323531e+00, -9.69205340e-01,  7.44405889e-01,
         -1.17934719e+00, -6.21581561e-01,  4.53382921e-01],
       [ 1.37779900e-01,  3.63355539e-01,  7.44405889e-01,
         -1.17934719e+00, -6.21581561e-01,  4.53382921e-01],
       ...,
       [ 3.24629502e-01, -7.22971264e-01,  7.44405889e-01,
         -1.05613181e-01, -6.21581561e-01, -6.00192629e-01],
       [-2.21286504e-01, -9.04025732e-01,  7.44405889e-01,
         -1.05613181e-01, -6.21581561e-01, -6.00192629e-01],
       [-1.13228903e-01,  1.24660453e-03, -1.34335316e+00,
          9.68120829e-01, -6.21581561e-01, -6.00192629e-01]])
```

In [57]:
```
#df_stand=pd.DataFrame(scaled_features,columns=df_tip.columns[:-1])
#df_stand.head(10)
```

Out[57]:

	total_bill	tip	gender	smoker	day	time
0	-0.314711	-1.439947	-1.343353	-1.179347	-0.621582	-0.600193
1	-1.063235	-0.969205	0.744406	-1.179347	-0.621582	0.453383
2	0.137780	0.363356	0.744406	-1.179347	-0.621582	0.453383
3	0.438315	0.225754	0.744406	-1.179347	-0.621582	-0.600193
4	0.540745	0.443020	-1.343353	-1.179347	-0.621582	1.506958
5	0.619537	1.239659	0.744406	-1.179347	-0.621582	1.506958
6	-1.239955	-0.722971	0.744406	-1.179347	-0.621582	-0.600193
7	0.798507	0.088153	0.744406	-1.179347	-0.621582	1.506958
8	-0.534203	-0.751940	0.744406	-1.179347	-0.621582	-0.600193
9	-0.563469	0.167817	0.744406	-1.179347	-0.621582	-0.600193

In [62]:
```
#df_stand['smoker']=df_tip['smoker']
```

Dependent And Independent Set

In [76]:
```
x=df_tip.drop('smoker',axis=1)
```

In [77]:

```
x
```

Out[77]:

	total_bill	tip	gender	day	time	size
0	16.99	1.01	0	0	0	2
1	10.34	1.66	1	0	0	3
2	21.01	3.50	1	0	0	3
3	23.68	3.31	1	0	0	2
4	24.59	3.61	0	0	0	4
...	...	...	...	...	...	...
239	29.03	5.92	1	1	0	3
240	27.18	2.00	0	1	0	2
241	22.67	2.00	1	1	0	2
242	17.82	1.75	1	1	0	2
243	18.78	3.00	0	2	0	2

244 rows × 6 columns

In [78]:

```
y=df_tip['smoker']
```

In [79]:

```
y
```

Out[79]:

```
0      0
1      0
2      0
3      0
4      0
      ..
239    0
240    1
241    1
242    0
243    0
Name: smoker, Length: 244, dtype: int64
```

Train and Test split

In [80]:

```
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [81]:
```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 170 entries, 27 to 68
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   total_bill  170 non-null    float64
 1   tip         170 non-null    float64
 2   gender      170 non-null    int64  
 3   day         170 non-null    int64  
 4   time        170 non-null    int64  
 5   size        170 non-null    int64  
dtypes: float64(2), int64(4)
memory usage: 9.3 KB
```

In [82]:
```
X_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 74 entries, 159 to 140
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   total_bill  74 non-null     float64
 1   tip         74 non-null     float64
 2   gender      74 non-null     int64  
 3   day         74 non-null     int64  
 4   time        74 non-null     int64  
 5   size        74 non-null     int64  
dtypes: float64(2), int64(4)
memory usage: 4.0 KB
```

Logistic Regression Algorithms

In [83]:
```
from sklearn.linear_model import LogisticRegression
```

In [84]:
```
logmodel=LogisticRegression()
```

In [85]:
```
logmodel.fit(X_train,y_train)
```

Out[85]:
```
LogisticRegression()
```

In [86]:
```
predictions=logmodel.predict(X_test)
```

In [87]:
```
from sklearn.metrics import classification_report
```

```
In [88]: print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.67      0.93      0.78        45
           1       0.73      0.28      0.40        29

    accuracy                           0.68        74
   macro avg       0.70      0.60      0.59        74
weighted avg       0.69      0.68      0.63        74
```

SVM Algorithm

```
In [89]: from sklearn import svm
```

```
In [90]: clf=svm.SVC(kernel='linear',C=1.0)
```

```
In [91]: clf.fit(X_train,y_train)
```

```
Out[91]: SVC(kernel='linear')
```

```
In [92]: prediction=clf.predict(X_test)
```

```
In [93]: from sklearn.metrics import classification_report
```

```
In [94]: print(classification_report(y_test,prediction))
```

```
              precision    recall  f1-score   support

           0       0.67      0.96      0.79        45
           1       0.80      0.28      0.41        29

    accuracy                           0.69        74
   macro avg       0.74      0.62      0.60        74
weighted avg       0.72      0.69      0.64        74
```

```
In [ ]:
```