

In [1]:

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

In [5]:

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

In [6]:

```
data.head()
```

Out[6]:

	total_bill	tip	sex	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

In [7]:

```
x=data[['total_bill','tip']]
```

In [8]:

```
y=data['size']
```

In [9]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split( x, y, test_size=0.33, random_state=42)
```

In [10]:

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
y_pred=lr.predict(x_test)
```

/usr/local/lib/python3.7/dist-packages/scikit-learn/\_logistic.py:940: ConvergenceWarning: lbfgs failed to converge (status=1):  
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

extra\_warning\_msg=\_LOGISTIC\_SOLVER\_CONVERGENCE\_MSG)

In [11]:

```
print(y_pred)
```

```
[2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
2 2 4 2 4 4 2]
```

In [12]:

```
from sklearn.metrics import classification_report, accuracy_score
clr=classification_report(y_test,y_pred)
acr=accuracy_score(y_test,y_pred)
print(clr, "Accuracy", acr, sep='\n')
```

	precision	recall	f1-score	support
1	0.00	0.00	0.00	3
2	0.67	0.98	0.80	48
3	0.00	0.00	0.00	12
4	0.45	0.38	0.42	13
5	0.00	0.00	0.00	3
6	0.00	0.00	0.00	2
accuracy			0.64	81
macro avg	0.19	0.23	0.20	81
weighted avg	0.47	0.64	0.54	81

Accuracy  
0.6419753086419753

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: Undefined
MetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

In [13]:

```
print(acr)
```

0.6419753086419753

In [14]:

```
from sklearn.neighbors import KNeighborsClassifier
kn=KNeighborsClassifier()
kn.fit(x_train,y_train)
y_pred1=kn.predict(x_test)
#print(y_pred1)
from sklearn.metrics import classification_report, accuracy_score
clr=classification_report(y_test,y_pred1)
acr=accuracy_score(y_test,y_pred1)
print(clr, "Accuracy", acr, sep='\n')
```

	precision	recall	f1-score	support
1	0.00	0.00	0.00	3
2	0.67	0.88	0.76	48
3	0.11	0.08	0.10	12
4	0.33	0.23	0.27	13
5	0.00	0.00	0.00	3
6	0.00	0.00	0.00	2
accuracy			0.57	81
macro avg	0.19	0.20	0.19	81
weighted avg	0.47	0.57	0.51	81

Accuracy  
0.5679012345679012

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: Undefined
MetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

In [15]:

```
from sklearn.svm import SVC
svm=SVC()
svm.fit(x_train,y_train)
y_pred2=svm.predict(x_test)
from sklearn.metrics import classification_report, accuracy_score
```

```
clr=classification_report(y_test,y_pred2)
acr=accuracy_score(y_test,y_pred2)
print(clr, "Accuracy", acr, sep='\n')
```

	precision	recall	f1-score	support
1	0.00	0.00	0.00	3
2	0.65	1.00	0.79	48
3	0.00	0.00	0.00	12
4	0.57	0.31	0.40	13
5	0.00	0.00	0.00	3
6	0.00	0.00	0.00	2
accuracy			0.64	81
macro avg	0.20	0.22	0.20	81
weighted avg	0.48	0.64	0.53	81

Accuracy

0.6419753086419753

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: Undefined
MetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

In [16]:

```
from sklearn.tree import DecisionTreeClassifier
dt=DecisionTreeClassifier()
dt.fit(x_train,y_train)
y_pred3=dt.predict(x_test)
from sklearn.metrics import classification_report , accuracy_score
clr=classification_report(y_test,y_pred3)
acr=accuracy_score(y_test,y_pred3)
print(clr, "Accuracy", acr, sep='\n')
```

	precision	recall	f1-score	support
1	1.00	0.33	0.50	3
2	0.68	0.71	0.69	48
3	0.36	0.42	0.38	12
4	0.33	0.38	0.36	13
5	0.00	0.00	0.00	3
6	0.00	0.00	0.00	2
accuracy			0.56	81
macro avg	0.40	0.31	0.32	81
weighted avg	0.55	0.56	0.54	81

Accuracy

0.5555555555555556

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: Undefined
MetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]: