

In [27]:

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

In [28]:

```
data = pd.read_csv("shootings_2022.csv")
```

In [29]:

```
data
```

Out[29]:

	Date	State	Dead	Injured	Total	Description
0	03-03-2022	Nevada	1	6	7	A dispute between neighbors led to a shooting ...
1	02-03-2022	Maryland	1	3	4	Four men were shot in the Walbrook neighborhoo...
2	28-02-2022	California	5	0	5	A man fatally shot himself, his three children...
3	27-02-2022	California	0	4	4	An argument between two groups of people at a ...
4	27-02-2022	Louisiana	0	4	4	An argument at a business in the southeast par...
...
76	01-01-2022	Georgia	1	3	4	After officers were dispatched to respond to a...
77	01-01-2022	Wisconsin	1	3	4	A man was killed and three others wounded in a...
78	01-01-2022	Indiana	0	4	4	Four people were wounded at a New Years party ...
79	01-01-2022	Colorado	2	2	4	Two adults were killed and two wounded in an e...
80	01-01-2022	Missouri	0	4	4	Four adults were wounded in the early morning ...

81 rows × 6 columns

In [30]:

```
data.shape
```

Out[30]:

(81, 6)

In [31]:

```
data.head()
```

Out[31]:

	Date	State	Dead	Injured	Total	Description
0	03-03-2022	Nevada	1	6	7	A dispute between neighbors led to a shooting ...
1	02-03-2022	Maryland	1	3	4	Four men were shot in the Walbrook neighborhoo...
2	28-02-2022	California	5	0	5	A man fatally shot himself, his three children...
3	27-02-2022	California	0	4	4	An argument between two groups of people at a ...
4	27-02-2022	Louisiana	0	4	4	An argument at a business in the southeast par...

In [32]:

```
data.tail()
```

Out[32]:

	Date	State	Dead	Injured	Total	Description
76	01-01-2022	Georgia	1	3	4	After officers were dispatched to respond to a...
77	01-01-2022	Wisconsin	1	3	4	A man was killed and three others wounded in a...
78	01-01-2022	Indiana	0	4	4	Four people were wounded at a New Years party ...
79	01-01-2022	Colorado	2	2	4	Two adults were killed and two wounded in an e...
80	01-01-2022	Missouri	0	4	4	Four adults were wounded in the early morning ...

In [33]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 81 entries, 0 to 80
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Date            81 non-null    object
1   State           81 non-null    object
2   Dead            81 non-null    int64
3   Injured         81 non-null    int64
4   Total           81 non-null    int64
5   Description     81 non-null    object
dtypes: int64(3), object(3)
memory usage: 3.9+ KB
```

In [34]:



```
data.describe()
```

Out[34]:

	Dead	Injured	Total
count	81.000000	81.000000	81.000000
mean	1.135802	3.802469	4.938272
std	1.339338	2.288120	2.063648
min	0.000000	0.000000	4.000000
25%	0.000000	3.000000	4.000000
50%	1.000000	4.000000	4.000000
75%	1.000000	4.000000	5.000000
max	6.000000	14.000000	16.000000

In [35]:



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

Out[35]:

Date	0
State	0
Dead	0
Injured	0
Total	0
Description	0
dtype: int64	

In [36]:



```
total_victims_state = data.groupby('State').sum()
print('Total Victims = ', total_victims_state['Total'].sum())
total_victims_state.sort_values(by = 'Total', ascending = False)
```

Total Victims = 400

Out[36]:

	Dead	Injured	Total
State			
California	15	38	53
Texas	15	22	37
Georgia	9	25	34
Missouri	5	23	28
Louisiana	2	20	22
Wisconsin	8	13	21
Nevada	2	19	21
Florida	3	15	18
Illinois	3	15	18
Oregon	2	14	16
Tennessee	5	8	13
North Carolina	2	10	12
Arizona	2	9	11
Alabama	1	10	11
Mississippi	2	7	9
Pennsylvania	1	8	9
South Carolina	1	8	9
New Mexico	1	7	8
Maryland	4	4	8
Colorado	4	4	8
Virginia	1	4	5
Washington D.C.	1	4	5
Minnesota	1	3	4
New York	0	4	4
Indiana	0	4	4
Arkansas	1	3	4
Washington	1	3	4
Nebraska	0	4	4

In [37]:

```
total_victims_state.sort_values(by = 'Total', ascending = False).head()
```

Out[37]:

	Dead	Injured	Total
State			
California	15	38	53
Texas	15	22	37
Georgia	9	25	34
Missouri	5	23	28
Louisiana	2	20	22

In [38]:

```
total_victims_state.sort_values(by = 'Total', ascending = False).tail()
```

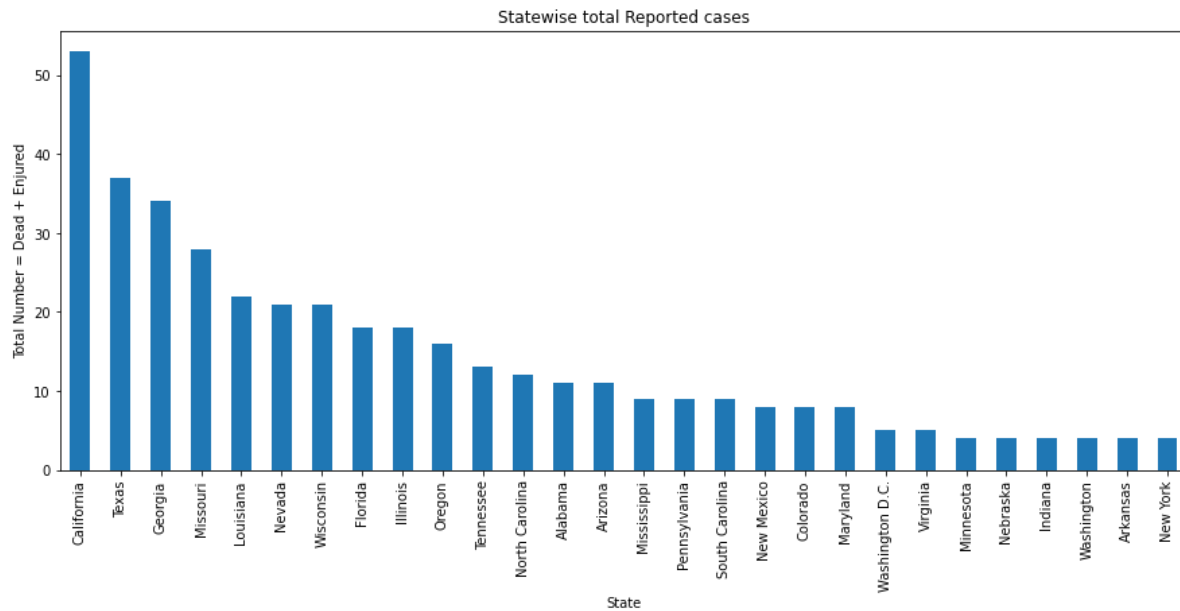
Out[38]:

	Dead	Injured	Total
State			
New York	0	4	4
Indiana	0	4	4
Arkansas	1	3	4
Washington	1	3	4
Nebraska	0	4	4

In [39]:



```
plt.subplots(figsize = (15, 6))
cr = total_victims_state['Total'].sort_values(ascending = False)
ax = cr.plot.bar()
ax.set_xlabel('State')
ax.set_ylabel('Total Number = Dead + Enjured')
ax.set_title('Statewise total Reported cases')
plt.show()
print(cr)
```



State	
California	53
Texas	37
Georgia	34
Missouri	28
Louisiana	22
Nevada	21
Wisconsin	21
Florida	18
Illinois	18
Oregon	16
Tennessee	13
North Carolina	12
Alabama	11
Arizona	11
Mississippi	9
Pennsylvania	9
South Carolina	9
New Mexico	8
Colorado	8
Maryland	8
Washington D.C.	5
Virginia	5
Minnesota	4
Nebraska	4
Indiana	4
Washington	4
Arkansas	4

New York 4
Name: Total, dtype: int64

In [40]:

```
california_victims = data[data['State'] == 'California']
```

In [41]:

```
california_victims.head()
```

Out[41]:

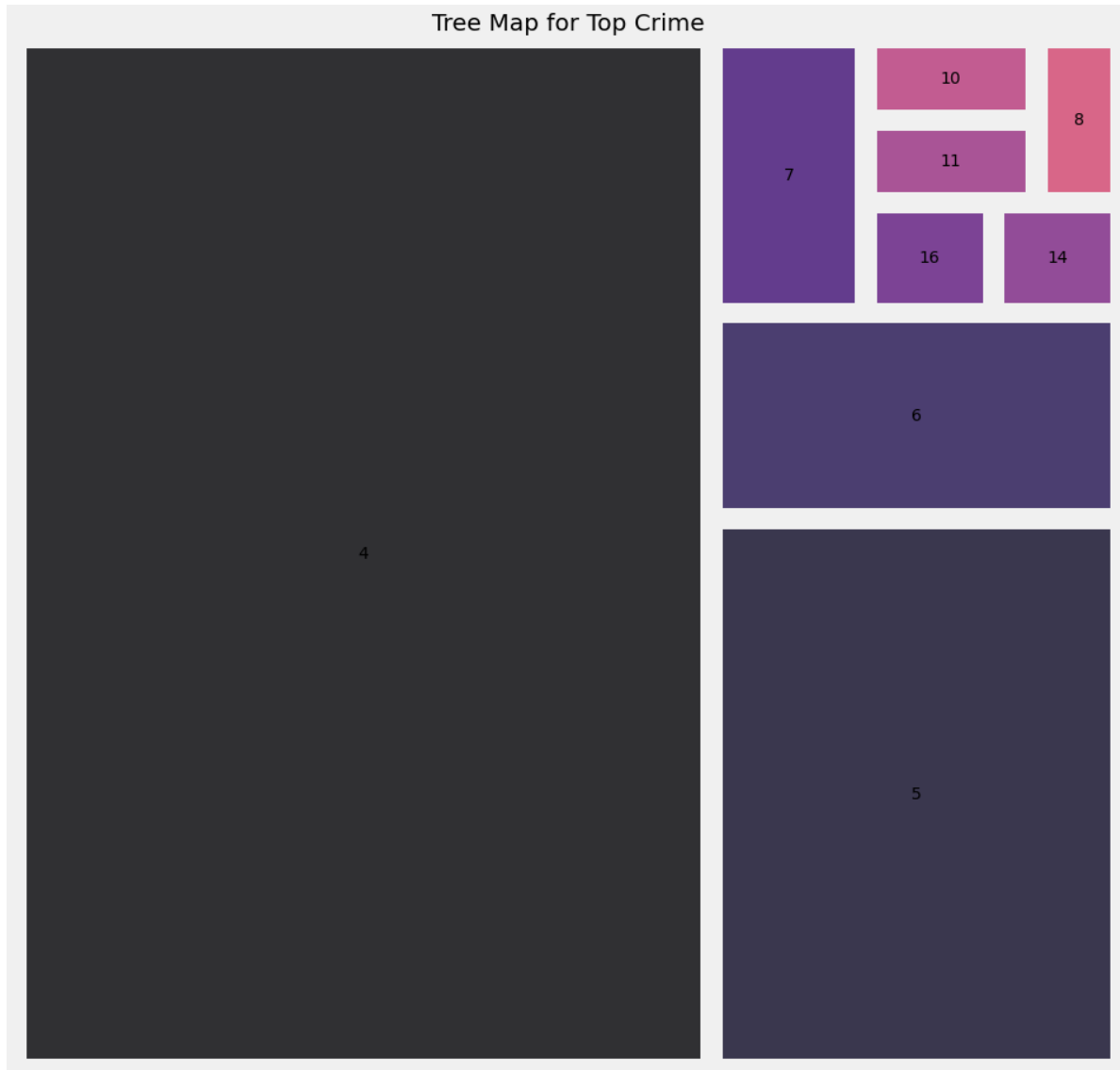
	Date	State	Dead	Injured	Total	Description
2	28-02-2022	California	5	0	5	A man fatally shot himself, his three children...
3	27-02-2022	California	0	4	4	An argument between two groups of people at a ...
21	19-02-2022	California	0	7	7	A man shot seven people outside a bar in the e...
28	12-02-2022	California	0	4	4	Outside an Italian restaurant hosting a party ...
33	06-02-2022	California	2	2	4	A chance encounter between two rival gangs at ...

In [45]:



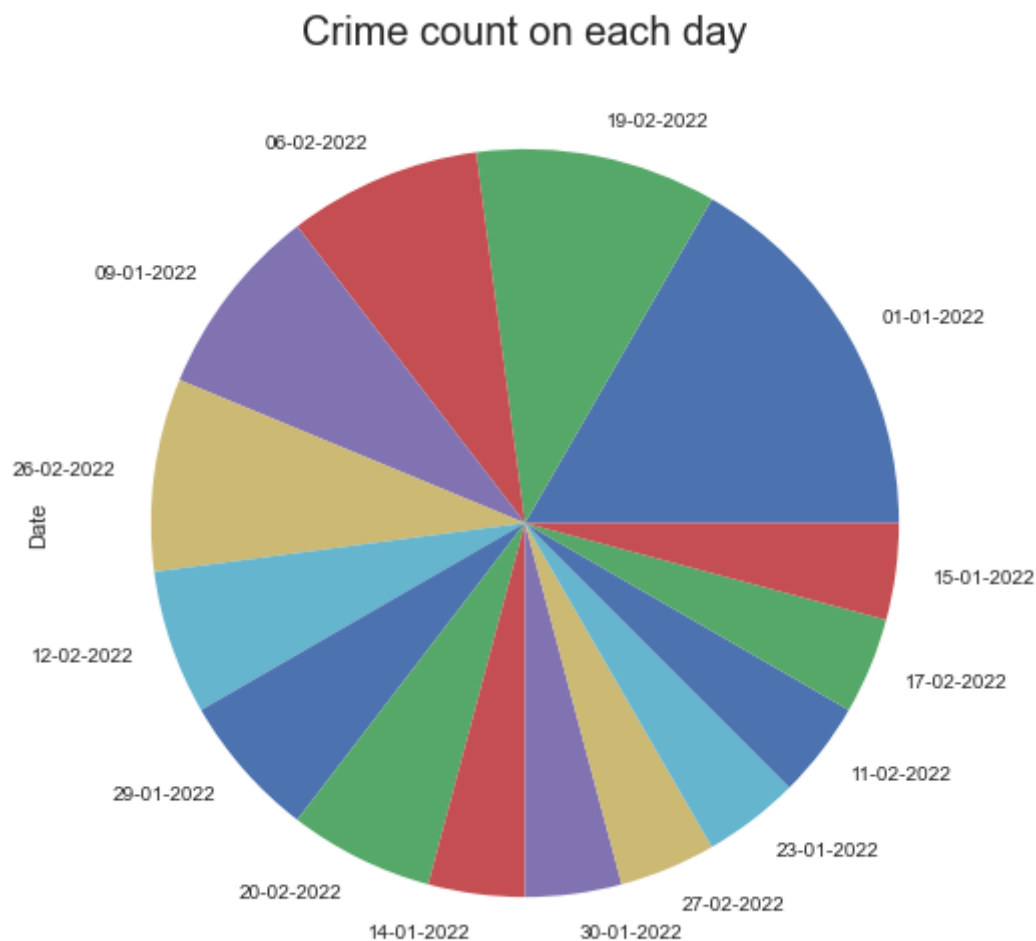
```
import squarify

y = data['Total'].value_counts().head(25)
plt.rcParams['figure.figsize'] = (15, 15)
plt.style.use('fivethirtyeight')
color = plt.cm.magma(np.linspace(0, 1, 15))
squarify.plot(sizes = y.values, label = y.index, alpha=.8, color = color, pad = True)
plt.title('Tree Map for Top Crime', fontsize = 20)
plt.axis('off')
plt.show()
```



In [47]:

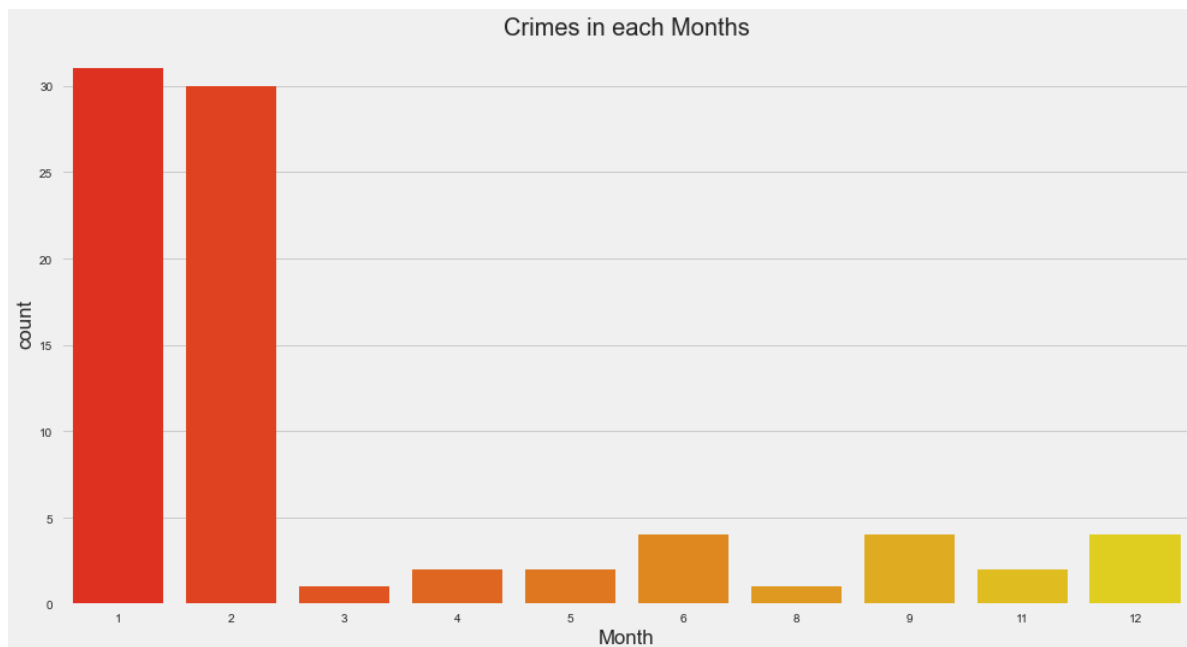
```
plt.style.use('seaborn')
data['Date'].value_counts().head(15).plot.pie(figsize = (15, 8))
plt.title('Crime count on each day',fontsize = 20)
plt.xticks(rotation = 90)
plt.show()
```



In [48]:



```
data['Date'] = pd.to_datetime(data['Date'])
data['Month'] = data['Date'].dt.month
plt.style.use('fivethirtyeight')
plt.rcParams['figure.figsize'] = (15, 8)
sns.countplot(data['Month'], palette = 'autumn',)
plt.title('Crimes in each Months', fontsize = 20)
plt.show()
```

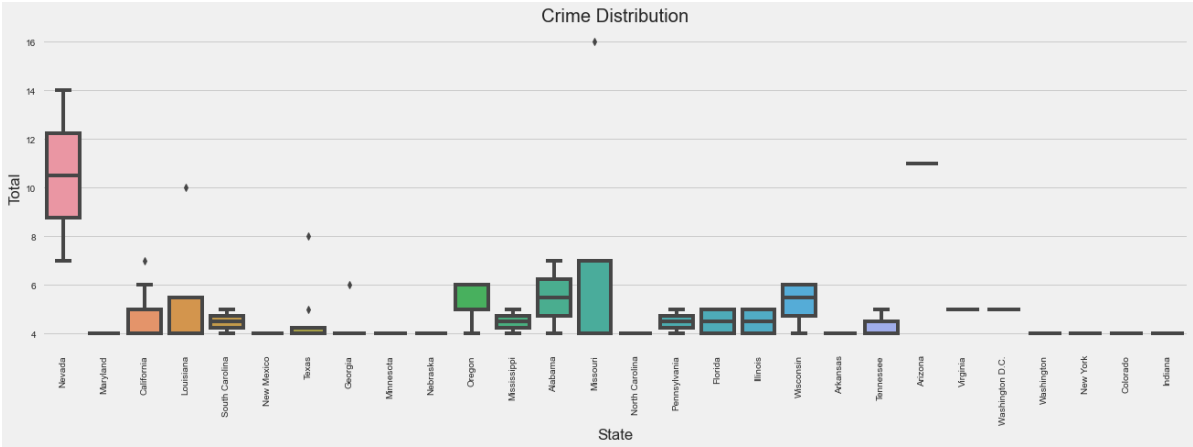


In [58]:

```
plt.figure(figsize=(20,6))
ax = sns.boxplot(x='State',y='Total',data=data)
plt.xticks(rotation=90)
ax.set_title("Crime Distribution")
```

Out[58]:

Text(0.5, 1.0, 'Crime Distribution')

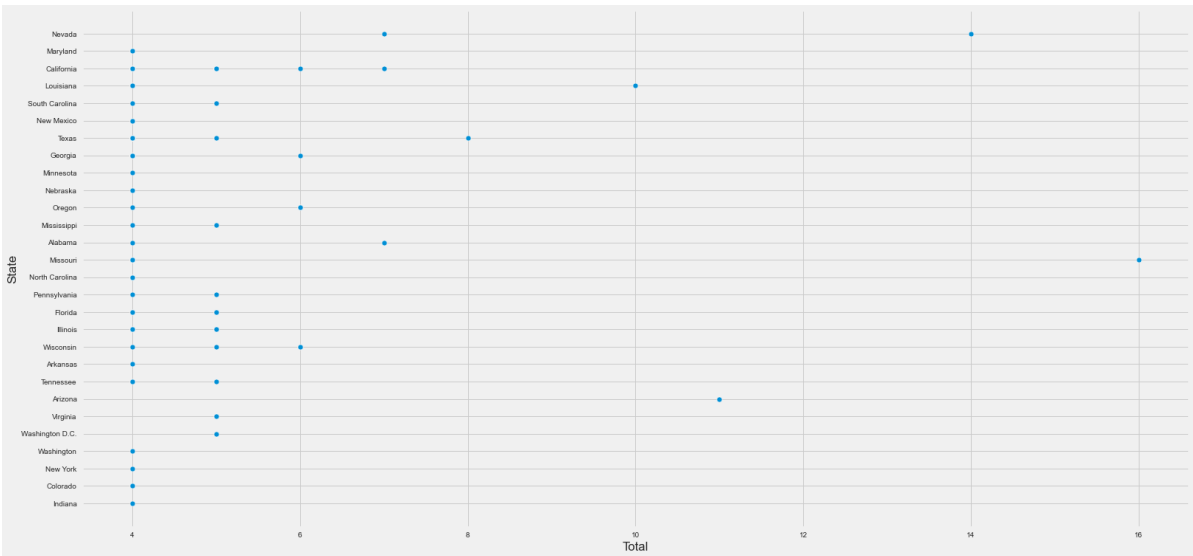


In [60]:

```
plt.figure(figsize=(24,12))
sns.scatterplot(x="Total",y="State",data=data)
```

Out[60]:

<matplotlib.axes._subplots.AxesSubplot at 0x819a575e0>



In [50]:

```
data.head()
```

Out[50]:

	Date	State	Dead	Injured	Total	Description	Month
0	2022-03-03	Nevada	1	6	7	A dispute between neighbors led to a shooting ...	3
1	2022-02-03	Maryland	1	3	4	Four men were shot in the Walbrook neighborhoo...	2
2	2022-02-28	California	5	0	5	A man fatally shot himself, his three children...	2
3	2022-02-27	California	0	4	4	An argument between two groups of people at a ...	2
4	2022-02-27	Louisiana	0	4	4	An argument at a business in the southeast par...	2

In [51]:

```
x = data.drop(['Date', 'State', 'Description'], axis = 1)
y = data['Total']
```

In [52]:

```
x.shape
```

Out[52]:

(81, 4)

In [53]:

```
y.shape
```

Out[53]:

(81,)

In [54]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [55]:

```
model = LogisticRegression()  
model.fit(X_train, y_train)
```

Out[55]:

```
LogisticRegression()
```

In [56]:

```
y_pred = model.predict(X_test)
```

In [57]:

```
print("Training Accuracy :", model.score(X_train, y_train))  
print("Testing Accuracy :", model.score(X_test, y_test))
```

```
Training Accuracy : 0.9375  
Testing Accuracy : 0.8235294117647058
```

In [61]:

```
from sklearn.linear_model import LinearRegression  
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [62]:

```
model1 = LinearRegression()  
model1.fit(X_train, y_train)
```

Out[62]:

```
LinearRegression()
```

In [63]:

```
y_pred = model1.predict(X_test)
```

In [64]:

```
print("Training Accuracy :", model1.score(X_train, y_train))  
print("Testing Accuracy :", model1.score(X_test, y_test))
```

```
Training Accuracy : 1.0  
Testing Accuracy : 1.0
```