### Importing Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.axes as ax
import seaborn as sns

sns.set()
```

# About Data

Customers are asked to rate the store out of 10 i.e. "satisfaction" point is given by the customers to store.Using an algorithm store gives "Loyality points" to customers based on the number of items purchased by them in the last year + the amount of money spent by them in the store

### Loading Data

In [2]:
```
data = pd.read_csv(r'C:\Users\G.SAI KRISHNA\Desktop\ML_Projects\ML_GFG\12.K-means Clusterir
data
```

Out[2]:

| | Satisfaction | Loyalty |
|---|---|---|
| **0** | 4 | -1.33 |
| **1** | 6 | -0.28 |
| **2** | 5 | -0.99 |
| **3** | 7 | -0.29 |
| **4** | 4 | 1.06 |
| **5** | 1 | -1.66 |
| **6** | 10 | -0.97 |
| **7** | 8 | -0.32 |
| **8** | 8 | 1.02 |
| **9** | 8 | 0.68 |
| **10** | 10 | -0.34 |
| **11** | 5 | 0.39 |
| **12** | 5 | -1.69 |
| **13** | 2 | 0.67 |
| **14** | 7 | 0.27 |
| **15** | 9 | 1.36 |
| **16** | 8 | 1.38 |
| **17** | 7 | 1.36 |
| **18** | 7 | -0.34 |
| **19** | 9 | 0.67 |
| **20** | 10 | 1.18 |
| **21** | 3 | -1.69 |
| **22** | 4 | 1.04 |
| **23** | 3 | -0.96 |
| **24** | 6 | 1.03 |
| **25** | 9 | -0.99 |
| **26** | 10 | 0.37 |
| **27** | 9 | 0.03 |
| **28** | 3 | -1.36 |
| **29** | 5 | 0.73 |

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Satisfaction  30 non-null     int64  
 1   Loyalty       30 non-null     float64
dtypes: float64(1), int64(1)
memory usage: 608.0 bytes
```

In [4]:
```
data.describe()
```

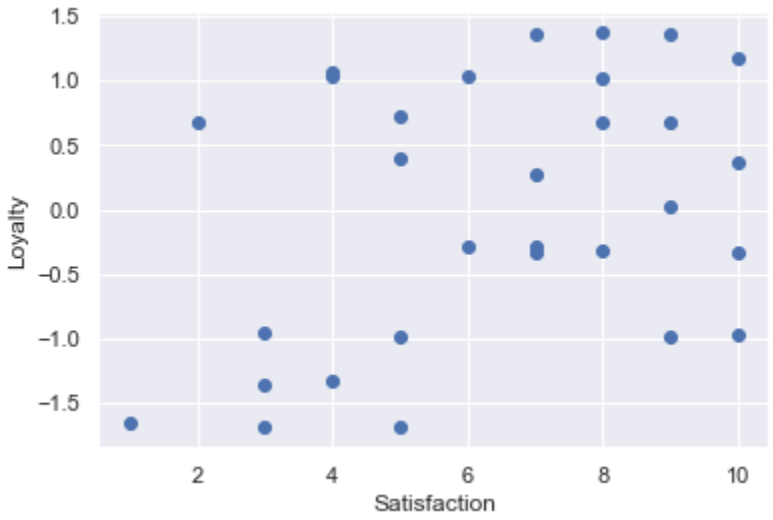
Out[4]:

| | Satisfaction | Loyalty |
|---|---|---|
| **count** | 30.000000 | 30.000000 |
| **mean** | 6.400000 | 0.001000 |
| **std** | 2.620871 | 1.016476 |
| **min** | 1.000000 | -1.690000 |
| **25%** | 4.250000 | -0.967500 |
| **50%** | 7.000000 | 0.150000 |
| **75%** | 8.750000 | 0.947500 |
| **max** | 10.000000 | 1.380000 |

## Plotting data

In [5]:
```
plt.scatter(data['Satisfaction'],data['Loyalty'])
plt.xlabel('Satisfaction')
plt.ylabel('Loyalty')
```

Out[5]:
```
Text(0, 0.5, 'Loyalty')
```




## Preparing Input data

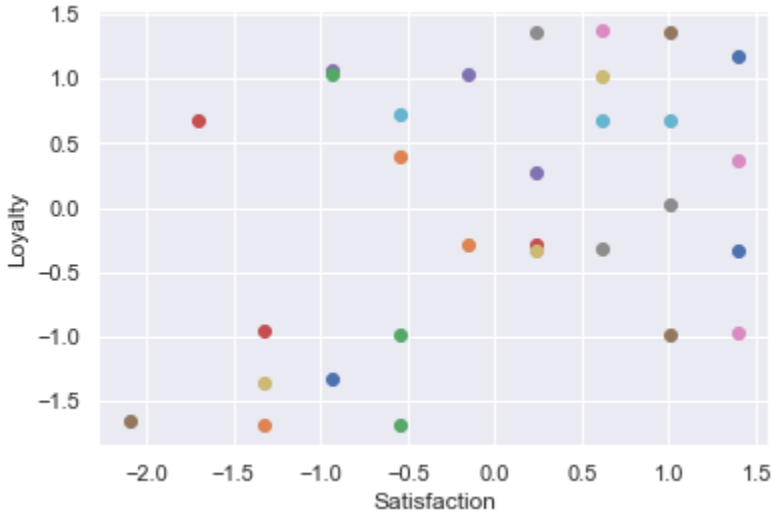
In [6]:
```
from sklearn import preprocessing

data_scaled = data.copy()
data_scaled = preprocessing.scale(data)
data_scaled
```

Out[6]:
```
array([[-0.93138063, -1.3318111 ],
       [-0.15523011, -0.28117124],
       [-0.54330537, -0.99160391],
       [ 0.23284516, -0.29117733],
       [-0.93138063,  1.05964534],
       [-2.09560642, -1.6620122 ],
       [ 1.39707095, -0.97159172],
       [ 0.62092042, -0.32119561],
       [ 0.62092042,  1.01962097],
       [ 0.62092042,  0.67941378],
       [ 1.39707095, -0.3412078 ],
       [-0.54330537,  0.38923705],
       [-0.54330537, -1.69203048],
       [-1.70753116,  0.66940768],
       [ 0.23284516,  0.26916393],
       [ 1.00899568,  1.35982816],
       [ 0.62092042,  1.37984035],
       [ 0.23284516,  1.35982816],
       [ 0.23284516, -0.3412078 ],
       [ 1.00899568,  0.66940768],
       [ 1.39707095,  1.17971847],
       [-1.31945589, -1.69203048],
       [-0.93138063,  1.03963316],
       [-1.31945589, -0.96158562],
       [-0.15523011,  1.02962706],
       [ 1.00899568, -0.99160391],
       [ 1.39707095,  0.36922486],
       [ 1.00899568,  0.02901767],
       [-1.31945589, -1.36182938],
       [-0.54330537,  0.72944425]])
```

In [7]:
```
for i in range(data_scaled.size//2):
    plt.scatter(data_scaled[i][0],data_scaled[i][1])
plt.xlabel('Satisfaction')
plt.ylabel('Loyalty')
```

Out[7]:
```
Text(0, 0.5, 'Loyalty')
```




## Elbow Method

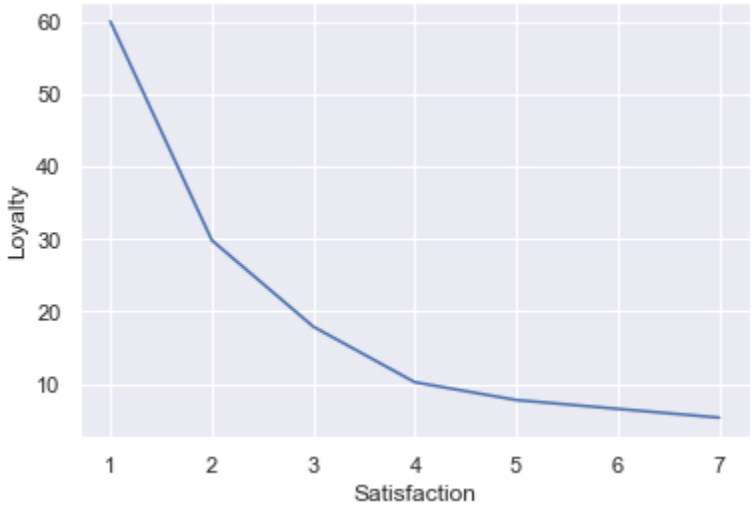
In [8]:
```
from sklearn.cluster import KMeans
```

In [9]:
```
#Within cluster sum of squares
wcss = []
for i in range(1,8):
    kmeans = KMeans(n_clusters=i,random_state=0)
    kmeans.fit(data_scaled)
    wcss.append(kmeans.inertia_)
wcss
```

Out[9]:
```
[59.999999999999986,
 29.818973034723143,
 17.913349527387968,
 10.247181805928422,
 7.792695153937187,
 6.569489487091783,
 5.363006425346783]
```

In [10]:
```
plt.plot(range(1,len(wcss)+1),wcss)
plt.xlabel('Satisfaction')
plt.ylabel('Loyalty')
```

Out[10]:
```
Text(0, 0.5, 'Loyalty')
```




## K-Means Clustering

In [11]:
```
kmeans_cluster = KMeans(4)
kmeans_cluster.fit(data_scaled)
```

Out[11]:
```
KMeans(n_clusters=4)
```

In [12]:
```
cluster_data = data.copy()
cluster_data['cluster_pred'] = kmeans_cluster.fit_predict(data_scaled)
cluster_data
```

Out[12]:

| | Satisfaction | Loyalty | cluster_pred |
|---|---|---|---|
| **0** | 4 | -1.33 | 1 |
| **1** | 6 | -0.28 | 2 |
| **2** | 5 | -0.99 | 1 |
| **3** | 7 | -0.29 | 2 |
| **4** | 4 | 1.06 | 3 |
| **5** | 1 | -1.66 | 1 |
| **6** | 10 | -0.97 | 2 |
| **7** | 8 | -0.32 | 2 |
| **8** | 8 | 1.02 | 0 |
| **9** | 8 | 0.68 | 0 |
| **10** | 10 | -0.34 | 2 |
| **11** | 5 | 0.39 | 3 |
| **12** | 5 | -1.69 | 1 |
| **13** | 2 | 0.67 | 3 |
| **14** | 7 | 0.27 | 2 |
| **15** | 9 | 1.36 | 0 |
| **16** | 8 | 1.38 | 0 |
| **17** | 7 | 1.36 | 0 |
| **18** | 7 | -0.34 | 2 |
| **19** | 9 | 0.67 | 0 |
| **20** | 10 | 1.18 | 0 |
| **21** | 3 | -1.69 | 1 |
| **22** | 4 | 1.04 | 3 |
| **23** | 3 | -0.96 | 1 |
| **24** | 6 | 1.03 | 3 |
| **25** | 9 | -0.99 | 2 |
| **26** | 10 | 0.37 | 0 |
| **27** | 9 | 0.03 | 2 |
| **28** | 3 | -1.36 | 1 |
| **29** | 5 | 0.73 | 3 |

## Clustering Labels

In [13]:
```
kmeans_cluster.labels_
```

Out[13]:
```
array([1, 2, 1, 2, 3, 1, 2, 2, 0, 0, 2, 3, 1, 3, 2, 0, 0, 0, 2, 0, 0, 1,
       3, 1, 3, 2, 0, 2, 1, 3])
```

## Visualization

In [14]:
```
plt.scatter(data['Satisfaction'],data['Loyalty'],c=cluster_data['cluster_pred'],cmap='rainb
plt.xlabel('Satisfaction')
plt.ylabel('Loyalty')
```

Out[14]:
```
Text(0, 0.5, 'Loyalty')
```