```
import pandas as pd
```

```
df = pd.read_csv("Mall_Customers.csv")
```

```
df.head()
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

```
df.rename(columns={'Genre': 'Gender'}, inplace=True)
df['Gender'].unique()
```

```
array(['Male', 'Female'], dtype=object)
```

```
X = df[["Gender", "Age", "Annual Income (k$)", "Spending Score (1-100)"]]
```

```
X.head()
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40

```
#label encoding
```

```
from sklearn import preprocessing
```

```
label_encoder = preprocessing.LabelEncoder()
```

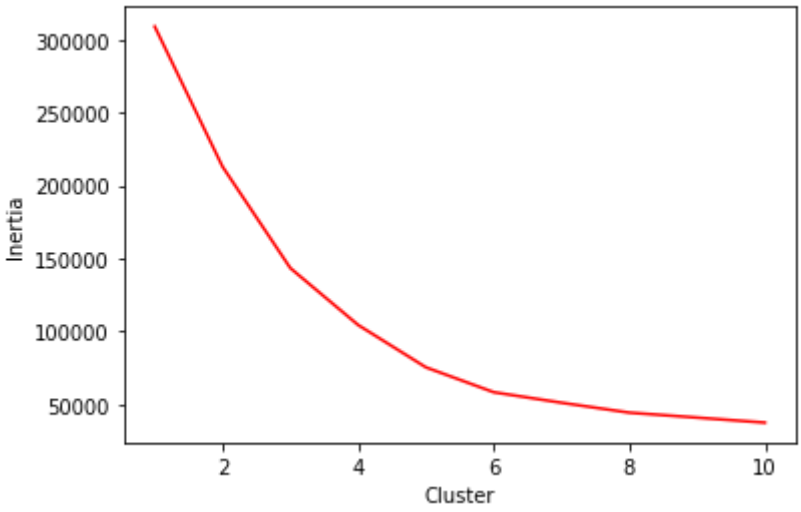
```
X["Gender"] = label_encoder.fit_transform(X["Gender"])
```

```
X["Gender"].unique()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7: SettingWithCo
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
```

```
  import sys
array([1, 0])
```

```
from sklearn.cluster import KMeans

cluster = []

for k in range(1, 11):
    kmean = KMeans(n_clusters=k).fit(X)
    cluster.append(kmean.inertia_)


import matplotlib.pyplot as plt

plt.plot(range(1, 11), cluster, 'r-')
plt.xlabel("Cluster")
plt.ylabel("Inertia")
plt.show()
```



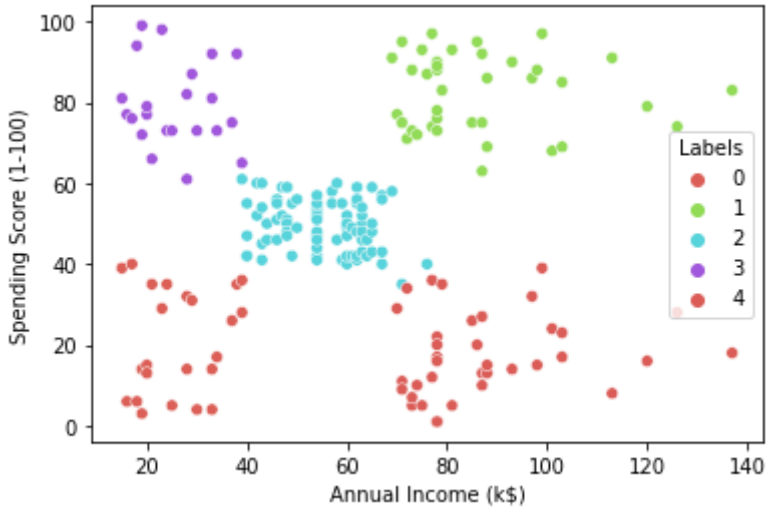

```
km = KMeans(n_clusters=5).fit(X)

X['Labels'] = km.labels_

X.head()
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Labels
0	1	19	15	39	0
1	1	21	15	81	3
2	0	20	16	6	0
3	0	23	16	77	3
4	0	31	17	40	0

```
import seaborn as sns
sns.scatterplot(X['Annual Income (k$)'], X['Spending Score (1-100)'], hue=X["Label
```

```
plt.show()
```

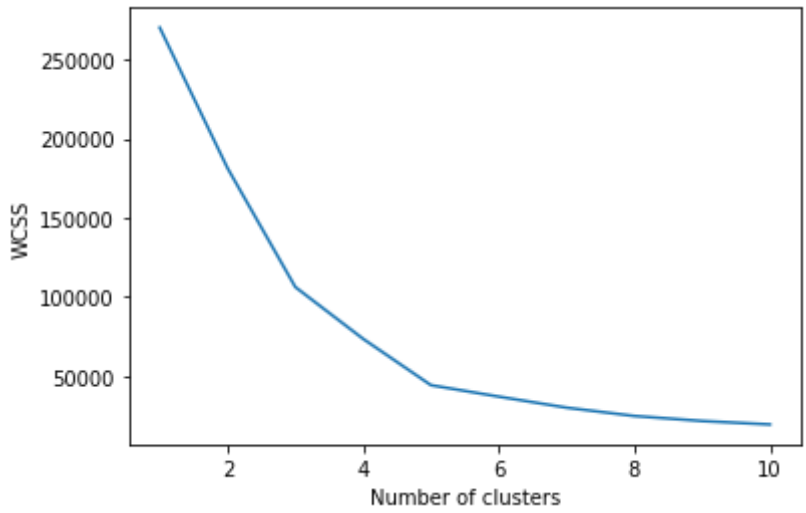
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarni
  FutureWarning
```




```
temp = pd.DataFrame()
temp['Annual Income (k$)'] = df['Annual Income (k$)']
temp['Spending Score (1-100)'] = df['Spending Score (1-100)']
```

```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(temp)
    wcss.append(kmeans.inertia_)
```

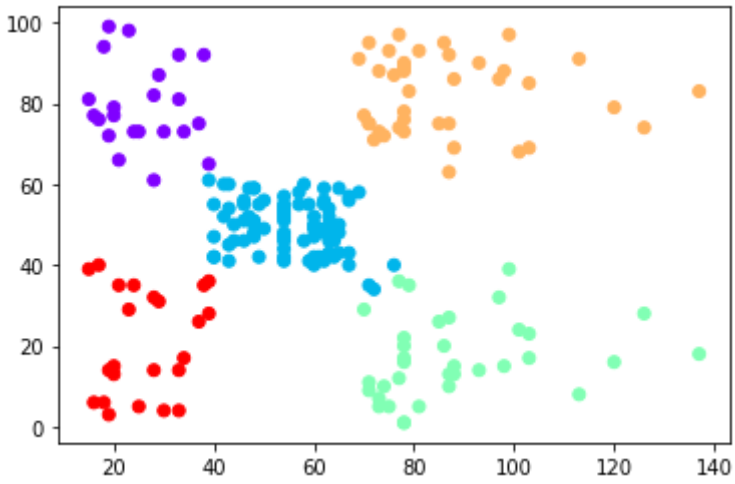
```
plt.plot(range(1, 11), wcss)
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```




```
kmean=KMeans(n_clusters=5)
kmean.fit(temp)
```

```
identified_clusters = kmean.fit_predict(temp)
temp["Predicted Value"] = identified_clusters

plt.scatter(temp["Annual Income (k$)"],temp['Spending Score (1-100)'],c=temp['Pred
```

```
<matplotlib.collections.PathCollection at 0x7fd755114150>
```



✓ 0s completed at 11:19 PM