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

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

Double-click (or enter) to edit

```
customers_data = customers_data.drop('CustomerID', axis=1)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
encode = LabelEncoder()
encoded_sex = encode.fit_transform(customers_data.iloc[:, 0])
print(encoded_sex)
```

```
[1 1 0 0 0 0 0 0 1 0 1 0 0 0 1 1 0 1 1 0 1 1 0 1 0 1 0 1 0 0 1 0 1 1 0 0 0
 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 1 0 0 1 1 0 0 1 0 1 0 0 0
 1 1 0 1 0 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 1 0 0 1 1 1 1
 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 1 1 1 1 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1 1 0
 0 1 1 1 0 0 0 0 1 0 1 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 0 0
 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1]
```

```
customers_data['Gender'] = encoded_sex
```

```
customers_data.head()
```

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

```
customers_data.columns
```

```
Index(['Gender', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)'], dtype='object')
```

```
from sklearn.decomposition import PCA
pca_reducer = PCA(n_components=2)
reduced_data = pca_reducer.fit_transform(customers_data)
```

```
reduced_data.shape
```

```
(200, 2)
```

```
reduced_data
```

```
array([[-3.18699448e+01, -3.30012521e+01],
       [ 7.64494048e-01, -5.68429006e+01],
       [-5.74082757e+01, -1.31249607e+01],
       [-2.16854252e+00, -5.34785900e+01],
       [-3.21740846e+01, -3.03884119e+01],
       [-2.17695183e+00, -5.22272685e+01],
       [-5.90656895e+01, -9.54376302e+00],
       [ 1.23708622e+01, -6.16180208e+01],
       [-6.63157690e+01, -3.21423160e+00],
       [-5.65556212e+00, -4.72672225e+01],
       [-5.82365979e+01, -9.13418751e+00],
       [ 1.46218043e+01, -6.21075685e+01],
       [-5.51608374e+01, -1.00799251e+01],
       [-3.02878330e-03, -5.01140754e+01],
       [-5.27646657e+01, -1.16807468e+01],
       [ 1.94661180e+00, -5.15222086e+01],
       [-3.45055496e+01, -2.37615402e+01],
       [-7.30493262e+00, -4.35151138e+01],
       [-4.12573070e+01, -1.64734737e+01],
       [ 1.61902860e+01, -5.83001207e+01],
       [-3.27400141e+01, -2.13347966e+01],
       [-9.81885245e-01, -4.44524038e+01],
       [-5.78100898e+01, -1.87057486e+00],
       [-1.52712624e+00, -4.28582471e+01],
       [-5.04819556e+01, -3.56276971e+00],
       [ 7.69078091e+00, -4.58600560e+01],
       [-3.46329913e+01, -1.50715244e+01],
       [-9.94927953e+00, -3.30225018e+01],
       [-3.38854915e+01, -1.43440485e+01],
       [ 1.33436470e+01, -4.87084876e+01],
       [-5.82990002e+01,  4.58029961e+00],
       [ 3.30609318e+00, -4.01274983e+01],
       [-5.52102909e+01,  6.08873470e+00],
       [ 2.05727386e+01, -4.89976354e+01],
       [-4.65939553e+01, -1.75635193e-01],
       [ 1.13600364e+01, -4.22934898e+01],
       [-4.23244634e+01, -2.00566039e+00],
       [ 3.95978859e+00, -3.57152924e+01],
       [-3.23506471e+01, -5.53131490e+00],
       [ 9.18744850e+00, -3.57475306e+01],
       [-3.01682815e+01, -6.09009687e+00],
       [ 2.23819886e+01, -4.41699255e+01],
       [-2.55813682e+01, -8.08036828e+00],
       [-2.71818732e+00, -2.46556548e+01],
       [-3.20580877e+01, -3.35982914e+00],
       [ 1.74869118e+00, -2.78679628e+01],
       [-1.04361893e+01, -1.79156325e+01],
       [-1.23777865e+01, -1.63365883e+01],
       [-1.66857879e+01, -1.32051927e+01],
       [-1.70637425e+01, -1.29432699e+01],
       [-1.14279946e+01, -1.47081336e+01],
       [-2.11661573e+00, -2.13930938e+01],
       [-5.86576186e+00, -1.74048397e+01],
       [-6.44140255e+00, -1.71797090e+01],
       [-1.65304147e+01, -9.75152197e+00],
       [-1.91078530e+01, -7.84724924e+00],
       [-1.22007224e+01, -1.16816450e+01],
       [-1.87466838e+01, -7.02718282e+00],
```

```
from sklearn.cluster import KMeans

km = KMeans(n_clusters=5)

cluster = km.fit(reduced_data)
```

```
plt.scatter(reduced_data[:, 0], reduced_data[:, 1], label='Datapoints')
plt.scatter(cluster.cluster_centers_[:, 0], cluster.cluster_centers_[:, 1], label='Clusters')
plt.title("Sklearn version of KMeans")
plt.legend()
plt.show()
```