In [12]:

```
from sklearn.datasets import load_iris
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
```

In [2]:

```
# Load the Iris dataset
iris = load_iris()
```

In [5]:

```
iris.data.shape
```

Out[5]:

```
(150, 4)
```

In [6]:

```
iris.target.shape
```

Out[6]:

```
(150,)
```

In [3]:

```
# Split the data into features and target
X = iris.data
y = iris.target
```

In [7]:

```
# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

In [8]:

```
# Perform PCA with two components
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
```

In [11]:

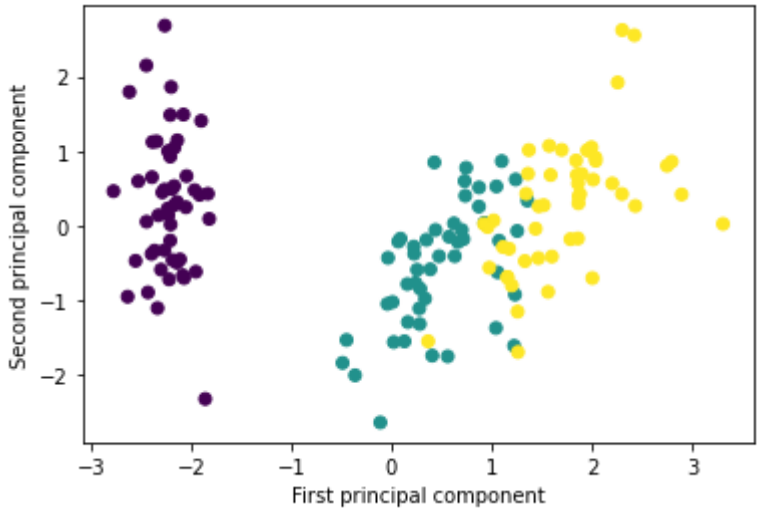
```
X_pca.shape
```

Out[11]:

```
(150, 2)
```

In [9]:

```
# Create a scatter plot of the first two principal components
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y)
plt.xlabel('First principal component')
plt.ylabel('Second principal component')
plt.show()
```



Before PCA

In [16]:

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.33, random_
```

In [20]:

```
y_train.shape
```

Out[20]:

```
(100,)
```

In [14]:

```
from sklearn.linear_model import LinearRegression
```

In [15]:

```
lr = LinearRegression()
```

In [21]:

```
lr.fit(X_train, y_train)
```

Out[21]:

```
LinearRegression()
```

In [22]:

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

In [26]:

```
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score
```

In [28]:

```
# Calculate the mean squared error and R-squared value
mse = mean_squared_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)
```

In [25]:

```
# Print the results
print("Mean squared error: {:.2f}".format(mse))
print("Root mean squared error: {:.2f}".format(rmse))
print("R-squared value: {:.2f}".format(r2))
```

```
Mean squared error: 0.04
Root mean squared error: 0.21
R-squared value: 0.94
```

After PCA

In [29]:

```
X_train_2, X_test_2, y_train_2, y_test_2 = train_test_split(X_pca, y, test_size=0.33, ra
```

In [31]:

```
X_train_2.shape
```

Out[31]:

```
(100, 2)
```

In [32]:

```
lr.fit(X_train_2, y_train_2)
```

Out[32]:

```
LinearRegression()
```

In [33]:

```
y_pred_2 = lr.predict(X_test_2)
```

In [34]:

```
mse = mean_squared_error(y_test_2, y_pred_2)
rmse = mean_squared_error(y_test_2, y_pred_2, squared=False)
r2 = r2_score(y_test_2, y_pred_2)
```

In [35]:

```
print("Mean squared error: {:.2f}".format(mse))
print("Root mean squared error: {:.2f}".format(rmse))
print("R-squared value: {:.2f}".format(r2))
```

```
Mean squared error: 0.07
Root mean squared error: 0.26
R-squared value: 0.90
```

In [ ]:

```

```