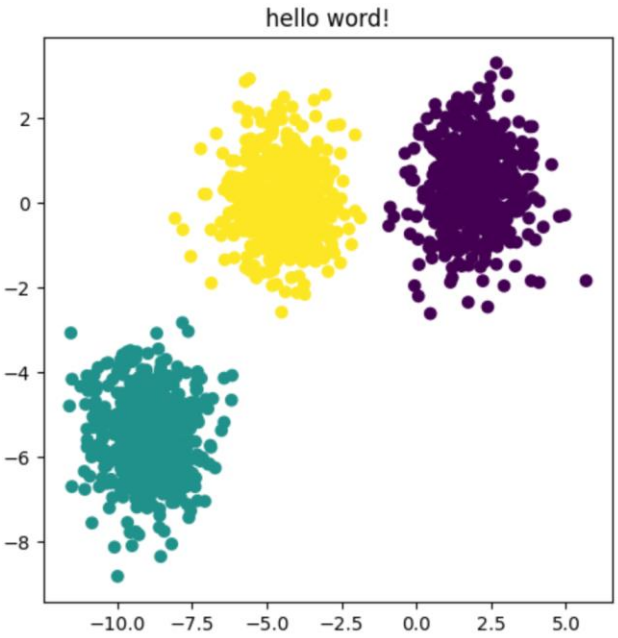
图一：

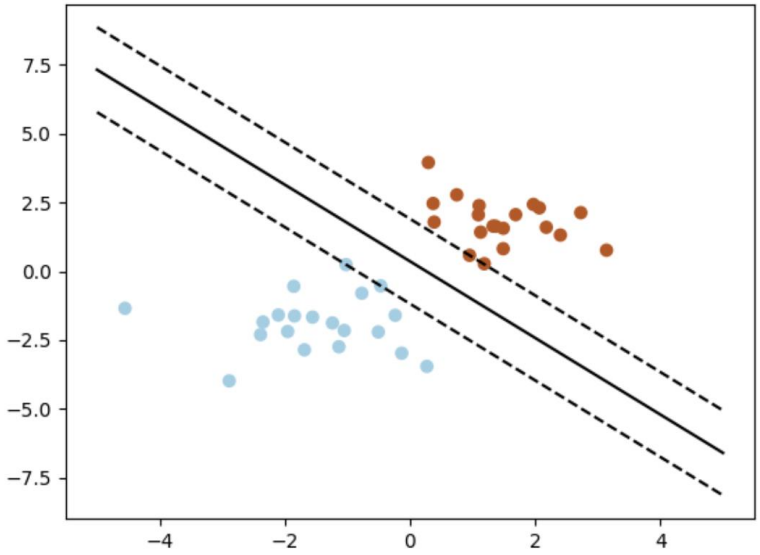


代码：

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
def show_kmeans():
    print(__doc__)
    plt.figure(figsize=(12, 12))
    n_samples = 1500
    random_state = 170
    X, y = make_blobs(n_samples=n_samples, random_state=random_state)
    y_pred = KMeans(n_clusters=3, random_state=random_state).fit_predict(X)
    plt.subplot(221)
    plt.scatter(X[:, 0], X[:, 1], c=y_pred)
    plt.title("hello word!")
    plt.show()
if __name__ == '__main__':
    show_kmeans()
```

该代码使用 Matplotlib 库将原始数据和聚类结果可视化，并在子图标题中打印“hello word!”。函数的输出是一个图形，显示了原始数据和聚类结果。

图二：



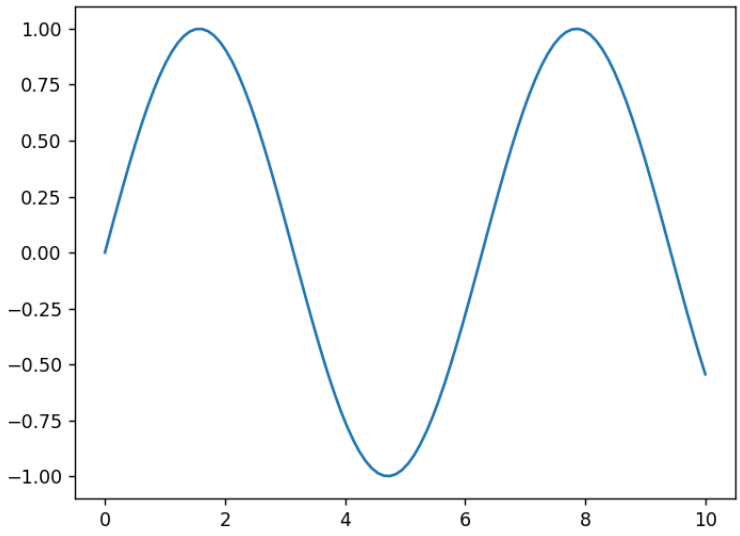
代码：

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm

np.random.seed(0)
X = np.r_[np.random.randn(20, 2) - [2, 2], np.random.randn(20, 2) + [2, 2]]
Y = [0] * 20 + [1] * 20

clf = svm.SVC(kernel='linear')
clf.fit(X, Y)

w = clf.coef_[0]
a = -w[0] / w[1]
xx = np.linspace(-5, 5)
yy = a * xx - (clf.intercept_[0]) / w[1]

b = clf.support_vectors_[0]
yy_down = a * xx + (b[1] - a * b[0])
b = clf.support_vectors_[-1]
yy_up = a * xx + (b[1] - a * b[0])

plt.plot(xx, yy, 'k-')
plt.plot(xx, yy_down, 'k--')
plt.plot(xx, yy_up, 'k--')
```

```
plt.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1],
                    s=80, facecolors='none')
plt.scatter(X[:, 0], X[:, 1], c=Y, cmap=plt.cm.Paired)

plt.axis('tight')
plt.show()
```

该代码生成包含 40 个可分离点的数据集，前 20 个标记为 10，后 20 个标记为 1，然后构造超平面，使用 svm 处理获取分离超平面并计算平面上的点，最后使用 Matplotlib 库绘制分离超平面、支持向量和原始数据点，并使用不同的颜色进行分类。

图三：

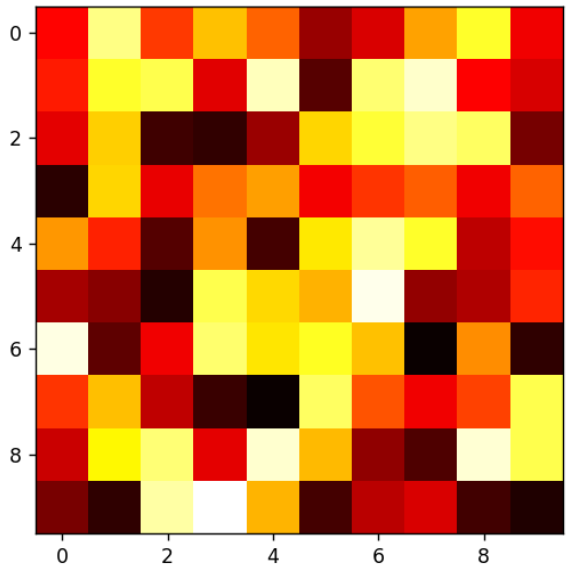


代码：

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 10, 100)
y = np.sin(x)
plt.plot(x, y)
plt.show()
```

使用 matplotlib 和 numpy 库绘制折线图。

图四：



代码：

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

data = np.random.rand(10, 10)
plt.imshow(data, cmap='hot', interpolation='nearest')
plt.show()
```

使用 matplotlib 和 numpy 库绘制热力图。