



KNN:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

```
x, y = make_classification(n_samples=200, n_features=2,
                           n_classes=2, random_state=42, n_informative=2,
                           n_redundant=0, n_repeated=0)
```

```
x_train, x_test, y_train, y_test = train_test_split(
    x, y, test_size=0.3, random_state=42)
```

```
scales = StandardScaler()
```

```
x_train = scales.fit_transform(x_train)
```

```
x_test = scales.transform(x_test)
```

```
knn = KNeighborsClassifier(n_neighbors=3)
```

```
knn.fit(x_train, y_train)
```

```
y_pred = knn.predict(x_test)
```

```
h=0.02
```

```
x_min, y_min, x_max, y_max = x_train[:,0].min()-1, x_train[:,0].max()+1,
                               y_train[:,1].min()-1, y_train[:,1].max()+1
```

```
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                     np.arange(y_min, y_max, h))
```

```
z = knn.predict(np.c_[xx.ravel(), yy.ravel()])
```

```
z = z.reshape(xx.shape)
```

```
plt.figure(figsize=(10, 6))
```



```
plt.contourf(xx, yy, z, alpha=0.8)
plt.scatter(x_train[:, 0], x_train[:, 1],
            C=y_train, edgecolors='k', marker='o',
            s=100, label='Train Data', cmap=plt.cm.
            coolwarm)
plt.scatter(x_test[:, 0], x_test[:, 1], C=y_test,
            edgecolor='k', marker='s', s=100, label='Test Data',
            cmap=plt.cm.coolwarm)
plt.title("K-Nearest Neighbor Decision Boundary")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.legend()
plt.show()
```