```
!pip install qiskit
```

```
Collecting qiskit
  Downloading qiskit-0.44.0.tar.gz (8.9 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Collecting qiskit-terra==0.25.0 (from qiskit)
  Downloading qiskit_terra-0.25.0-cp38-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.1/6.1 MB 12.7 MB/s eta 0:00:00
Collecting rustworkx>=0.13.0 (from qiskit-terra==0.25.0->qiskit)
  Downloading rustworkx-0.13.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 25.9 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra==0.25.0->qiskit) (1.23.5)
Collecting ply>=3.10 (from qiskit-terra==0.25.0->qiskit)
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.6/49.6 kB 4.8 MB/s eta 0:00:00
Requirement already satisfied: psutil>=5 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra==0.25.0->qiskit) (5.9.5)
Requirement already satisfied: scipy>=1.5 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra==0.25.0->qiskit) (1.10.1)
Requirement already satisfied: sympy>=1.3 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra==0.25.0->qiskit) (1.12)
Collecting dill>=0.3 (from qiskit-terra==0.25.0->qiskit)
  Downloading dill-0.3.7-py3-none-any.whl (115 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 115.3/115.3 kB 10.2 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.8.0 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra==0.25.0->qiskit) (2.
Collecting stevedore>=3.0.0 (from qiskit-terra==0.25.0->qiskit)
  Downloading stevedore-5.1.0-py3-none-any.whl (49 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.6/49.6 kB 5.0 MB/s eta 0:00:00
Collecting symengine<0.10,>=0.9 (from qiskit-terra==0.25.0->qiskit)
  Downloading symengine-0.9.2-cp310-cp310-manylinux2010_x86_64.whl (37.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 37.5/37.5 MB 29.1 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from qiskit-terra==0.25.0->qiskit) (4.7.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.0->qiskit-terra==0.25.0->q
Collecting pbr!=2.1.0,>=2.0.0 (from stevedore>=3.0.0->qiskit-terra==0.25.0->qiskit)
  Downloading pbr-5.11.1-py2.py3-none-any.whl (112 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 112.7/112.7 kB 10.1 MB/s eta 0:00:00
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy>=1.3->qiskit-terra==0.25.0->qiskit) (
Building wheels for collected packages: qiskit
  Building wheel for qiskit (pyproject.toml) ... done
  Created wheel for qiskit: filename=qiskit-0.44.0-py3-none-any.whl size=7614 sha256=c6c559ddd8b1d584b2428f74dc8bdd72819907fb8474b1db742
  Stored in directory: /root/.cache/pip/wheels/45/90/00/70879ea1304b7b44cde9f737d2a819a87346d62666b3a89de1
Successfully built qiskit
Installing collected packages: ply, symengine, rustworkx, pbr, dill, stevedore, qiskit-terra, qiskit
Successfully installed dill-0.3.7 pbr-5.11.1 ply-3.11 qiskit-0.44.0 qiskit-terra-0.25.0 rustworkx-0.13.1 stevedore-5.1.0 symengine-0.9.2
```

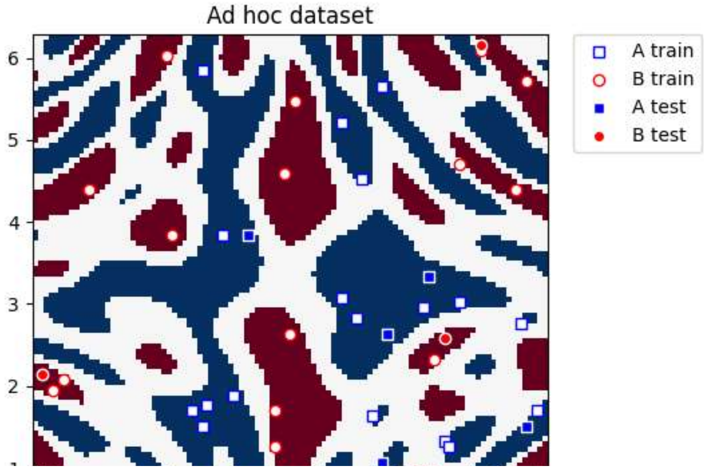
```
from qiskit.utils import algorithm_globals

algorithm_globals.random_seed = 12345
```

```
pip install qiskit_machine_learning
```

```
Collecting qiskit_machine_learning
  Downloading qiskit_machine_learning-0.6.1-py3-none-any.whl (148 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 148.7/148.7 kB 2.6 MB/s eta 0:00:00
Requirement already satisfied: qiskit-terra>=0.22.2 in /usr/local/lib/python3.10/dist-packages (from qiskit_machine_learning) (0.25.0)
Requirement already satisfied: scipy>=1.4 in /usr/local/lib/python3.10/dist-packages (from qiskit_machine_learning) (1.10.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from qiskit_machine_learning) (1.23.5)
Requirement already satisfied: psutil>=5 in /usr/local/lib/python3.10/dist-packages (from qiskit_machine_learning) (5.9.5)
Requirement already satisfied: scikit-learn>=0.20.0 in /usr/local/lib/python3.10/dist-packages (from qiskit_machine_learning) (1.2.2)
Collecting fastdtw (from qiskit_machine_learning)
  Downloading fastdtw-0.3.4.tar.gz (133 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.4/133.4 kB 5.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: setuptools>=40.1.0 in /usr/local/lib/python3.10/dist-packages (from qiskit_machine_learning) (67.7.2)
Requirement already satisfied: dill>=0.3.4 in /usr/local/lib/python3.10/dist-packages (from qiskit_machine_learning) (0.3.7)
Requirement already satisfied: rustworkx>=0.13.0 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra>=0.22.2->qiskit_machine_l
Requirement already satisfied: ply>=3.10 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra>=0.22.2->qiskit_machine_learning)
Requirement already satisfied: sympy>=1.3 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra>=0.22.2->qiskit_machine_learning
Requirement already satisfied: python-dateutil>=2.8.0 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra>=0.22.2->qiskit_mach
Requirement already satisfied: stevedore>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra>=0.22.2->qiskit_machine_le
Requirement already satisfied: symengine<0.10,>=0.9 in /usr/local/lib/python3.10/dist-packages (from qiskit-terra>=0.22.2->qiskit_machir
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from qiskit-terra>=0.22.2->qiskit_machine_l
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.20.0->qiskit_machine_learr
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.20.0->qiskit_machir
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.0->qiskit-terra>=0.22.2->c
Requirement already satisfied: pbr!=2.1.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from stevedore>=3.0.0->qiskit-terra>=0.22.
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy>=1.3->qiskit-terra>=0.22.2->qiskit_ma
Building wheels for collected packages: fastdtw
  Building wheel for fastdtw (setup.py) ... done
```

```
    Created wheel for fastdtw: filename=fastdtw-0.3.4-cp310-cp310-linux_x86_64.whl size=471390 sha256=da0bd169c2905230e6f8ecaa38fee1ecef9f
    Stored in directory: /root/.cache/pip/wheels/73/c8/f7/c25448dab74c3acf4848bc25d513c736bb93910277e1528ef4
  Successfully built fastdtw
  Installing collected packages: fastdtw, qiskit_machine_learning
  Successfully installed fastdtw-0.3.4 qiskit_machine_learning-0.6.1
```

```python
from qiskit_machine_learning.datasets import ad_hoc_data

adhoc_dimension = 2
train_features, train_labels, test_features, test_labels, adhoc_total = ad_hoc_data(
    training_size=20,
    test_size=5,
    n=adhoc_dimension,
    gap=0.3,
    plot_data=False,
    one_hot=False,
    include_sample_total=True,
)


import matplotlib.pyplot as plt
import numpy as np


def plot_features(ax, features, labels, class_label, marker, face, edge, label):
    # A train plot
    ax.scatter(
        # x coordinate of labels where class is class_label
        features[np.where(labels[:] == class_label), 0],
        # y coordinate of labels where class is class_label
        features[np.where(labels[:] == class_label), 1],
        marker=marker,
        facecolors=face,
        edgecolors=edge,
        label=label,
    )


def plot_dataset(train_features, train_labels, test_features, test_labels, adhoc_total):

    plt.figure(figsize=(5, 5))
    plt.ylim(0, 2 * np.pi)
    plt.xlim(0, 2 * np.pi)
    plt.imshow(
        np.asmatrix(adhoc_total).T,
        interpolation="nearest",
        origin="lower",
        cmap="RdBu",
        extent=[0, 2 * np.pi, 0, 2 * np.pi],
    )

    # A train plot
    plot_features(plt, train_features, train_labels, 0, "s", "w", "b", "A train")

    # B train plot
    plot_features(plt, train_features, train_labels, 1, "o", "w", "r", "B train")

    # A test plot
    plot_features(plt, test_features, test_labels, 0, "s", "b", "w", "A test")

    # B test plot
    plot_features(plt, test_features, test_labels, 1, "o", "r", "w", "B test")

    plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left", borderaxespad=0.0)
    plt.title("Ad hoc dataset")

    plt.show()


plot_dataset(train_features, train_labels, test_features, test_labels, adhoc_total)
```



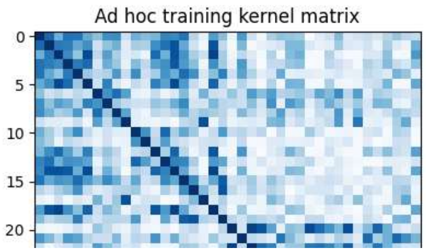
```
from qiskit.circuit.library import ZZFeatureMap
from qiskit.primitives import Sampler
from qiskit.algorithms.state_fidelities import ComputeUncompute
from qiskit_machine_learning.kernels import FidelityQuantumKernel

adhoc_feature_map = ZZFeatureMap(feature_dimension=adhoc_dimension, reps=2, entanglement="linear")

sampler = Sampler()

fidelity = ComputeUncompute(sampler=sampler)

adhoc_kernel = FidelityQuantumKernel(fidelity=fidelity, feature_map=adhoc_feature_map)
```

```
<ipython-input-8-6c5e6aa359ce>:3: DeprecationWarning: ``qiskit.algorithms`` has been migrated to an independent package: https://github.
  from qiskit.algorithms.state_fidelities import ComputeUncompute
```

```
from sklearn.svm import SVC

adhoc_svc = SVC(kernel=adhoc_kernel.evaluate)

adhoc_svc.fit(train_features, train_labels)

adhoc_score_callable_function = adhoc_svc.score(test_features, test_labels)

print(f"Callable kernel classification test score: {adhoc_score_callable_function}")
```

```
Callable kernel classification test score: 1.0
```

```
adhoc_matrix_train = adhoc_kernel.evaluate(x_vec=train_features)
adhoc_matrix_test = adhoc_kernel.evaluate(x_vec=test_features, y_vec=train_features)

fig, axs = plt.subplots(1, 2, figsize=(10, 5))

axs[0].imshow(
    np.asmatrix(adhoc_matrix_train), interpolation="nearest", origin="upper", cmap="Blues"
)
axs[0].set_title("Ad hoc training kernel matrix")

axs[1].imshow(np.asmatrix(adhoc_matrix_test), interpolation="nearest", origin="upper", cmap="Reds")
axs[1].set_title("Ad hoc testing kernel matrix")

plt.show()
```


Ad hoc training kernel matrix

Ad hoc testing kernel matrix

```
adhoc_svc = SVC(kernel="precomputed")

adhoc_svc.fit(adhoc_matrix_train, train_labels)

adhoc_score_precomputed_kernel = adhoc_svc.score(adhoc_matrix_test, test_labels)

print(f"Precomputed kernel classification test score: {adhoc_score_precomputed_kernel}")
```

```
Precomputed kernel classification test score: 1.0
```

```
from qiskit_machine_learning.algorithms import QSVC

qsvc = QSVC(quantum_kernel=adhoc_kernel)

qsvc.fit(train_features, train_labels)

qsvc_score = qsvc.score(test_features, test_labels)

print(f"QSVC classification test score: {qsvc_score}")
```

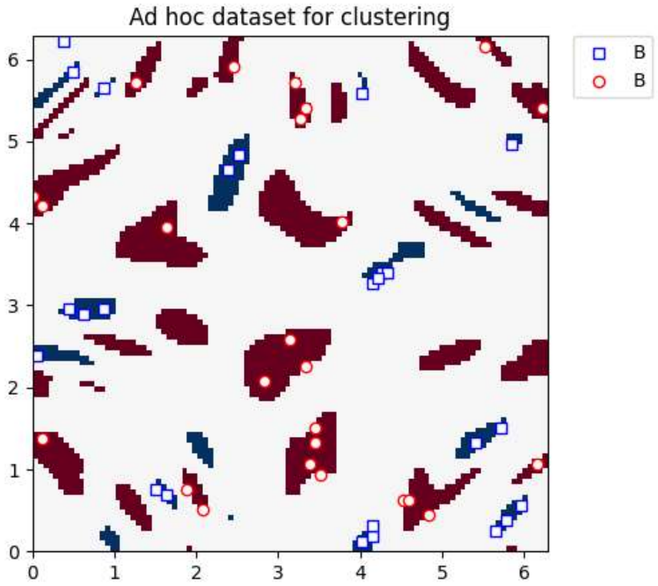
```
QSVC classification test score: 1.0
```

```
print(f"Classification Model                    | Accuracy Score")
print(f"---------------------------------------------------------")
print(f"SVC using kernel as a callable function | {adhoc_score_callable_function:10.2f}")
print(f"SVC using precomputed kernel matrix     | {adhoc_score_precomputed_kernel:10.2f}")
print(f"QSVC                                    | {qsvc_score:10.2f}")
```

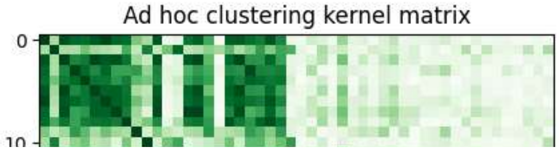
```
Classification Model                    | Accuracy Score
---------------------------------------------------------
SVC using kernel as a callable function |       1.00
SVC using precomputed kernel matrix     |       1.00
QSVC                                    |       1.00
```

```
adhoc_dimension = 2
train_features, train_labels, test_features, test_labels, adhoc_total = ad_hoc_data(
    training_size=25,
    test_size=0,
    n=adhoc_dimension,
    gap=0.6,
    plot_data=False,
    one_hot=False,
    include_sample_total=True,
)
```

```
plt.figure(figsize=(5, 5))
plt.ylim(0, 2 * np.pi)
plt.xlim(0, 2 * np.pi)
plt.imshow(
    np.asmatrix(adhoc_total).T,
    interpolation="nearest",
    origin="lower",
    cmap="RdBu",
    extent=[0, 2 * np.pi, 0, 2 * np.pi],
)

# A label plot
plot_features(plt, train_features, train_labels, 0, "s", "w", "b", "B")

# B label plot
plot_features(plt, train_features, train_labels, 1, "o", "w", "r", "B")

plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left", borderaxespad=0.0)
plt.title("Ad hoc dataset for clustering")

plt.show()
```



```
adhoc_feature_map = ZZFeatureMap(feature_dimension=adhoc_dimension, reps=2, entanglement="linear")

adhoc_kernel = FidelityQuantumKernel(feature_map=adhoc_feature_map)


adhoc_matrix = adhoc_kernel.evaluate(x_vec=train_features)

plt.figure(figsize=(5, 5))
plt.imshow(np.asmatrix(adhoc_matrix), interpolation="nearest", origin="upper", cmap="Greens")
plt.title("Ad hoc clustering kernel matrix")
plt.show()
```



```
from sklearn.cluster import SpectralClustering
from sklearn.metrics import normalized_mutual_info_score

adhoc_spectral = SpectralClustering(2, affinity="precomputed")

cluster_labels = adhoc_spectral.fit_predict(adhoc_matrix)

cluster_score = normalized_mutual_info_score(cluster_labels, train_labels)

print(f"Clustering score: {cluster_score}")
```

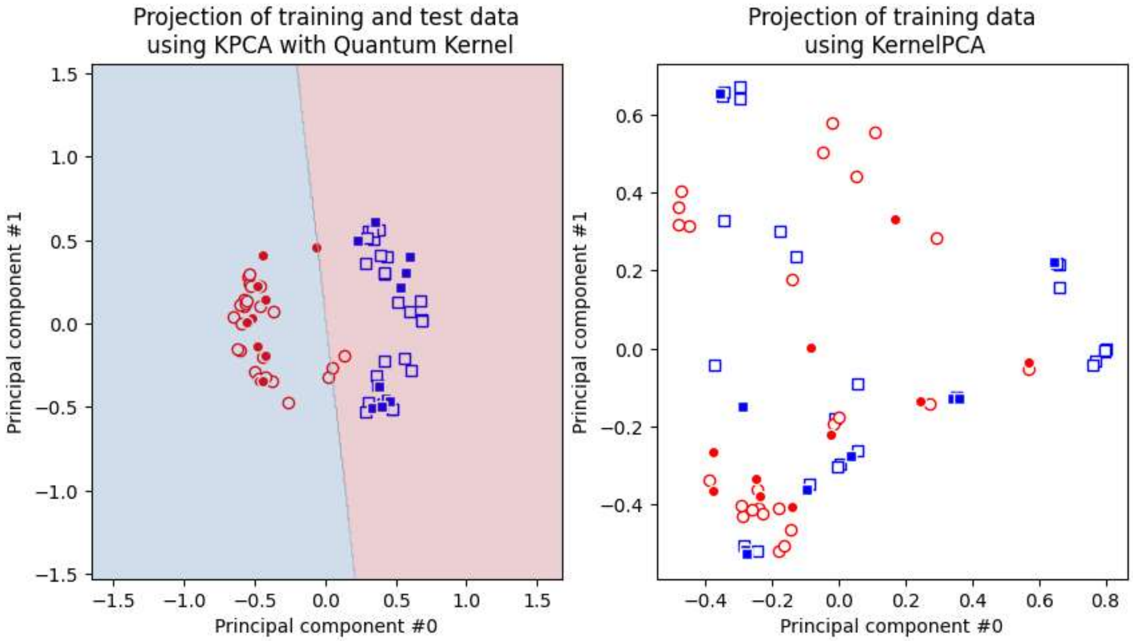
```
     Clustering score: 0.7287008798015754
```

```
adhoc_dimension = 2
train_features, train_labels, test_features, test_labels, adhoc_total = ad_hoc_data(
    training_size=25,
    test_size=10,
    n=adhoc_dimension,
    gap=0.6,
    plot_data=False,
    one_hot=False,
    include_sample_total=True,
)


feature_map = ZZFeatureMap(feature_dimension=2, reps=2, entanglement="linear")
qpca_kernel = FidelityQuantumKernel(fidelity=fidelity, feature_map=feature_map)


matrix_train = qpca_kernel.evaluate(x_vec=train_features)
matrix_test = qpca_kernel.evaluate(x_vec=test_features, y_vec=test_features)


from sklearn.decomposition import KernelPCA

kernel_pca_rbf = KernelPCA(n_components=2, kernel="rbf")
kernel_pca_rbf.fit(train_features)
train_features_rbf = kernel_pca_rbf.transform(train_features)
test_features_rbf = kernel_pca_rbf.transform(test_features)

kernel_pca_q = KernelPCA(n_components=2, kernel="precomputed")
train_features_q = kernel_pca_q.fit_transform(matrix_train)
test_features_q = kernel_pca_q.fit_transform(matrix_test)


from sklearn.linear_model import LogisticRegression

logistic_regression = LogisticRegression()
logistic_regression.fit(train_features_q, train_labels)

logistic_score = logistic_regression.score(test_features_q, test_labels)
print(f"Logistic regression score: {logistic_score}")
```

```
     Logistic regression score: 1.0
```

```
fig, (q_ax, rbf_ax) = plt.subplots(1, 2, figsize=(10, 5))


plot_features(q_ax, train_features_q, train_labels, 0, "s", "w", "b", "A train")
plot_features(q_ax, train_features_q, train_labels, 1, "o", "w", "r", "B train")

plot_features(q_ax, test_features_q, test_labels, 0, "s", "b", "w", "A test")
plot_features(q_ax, test_features_q, test_labels, 1, "o", "r", "w", "A test")

q_ax.set_ylabel("Principal component #1")
q_ax.set_xlabel("Principal component #0")
q_ax.set_title("Projection of training and test data\n using KPCA with Quantum Kernel")

# Plotting the linear separation
h = 0.01  # step size in the mesh
```

```
# create a mesh to plot in
x_min, x_max = train_features_q[:, 0].min() - 1, train_features_q[:, 0].max() + 1
y_min, y_max = train_features_q[:, 1].min() - 1, train_features_q[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))

predictions = logistic_regression.predict(np.c_[xx.ravel(), yy.ravel()])

# Put the result into a color plot
predictions = predictions.reshape(xx.shape)
q_ax.contourf(xx, yy, predictions, cmap=plt.cm.RdBu, alpha=0.2)

plot_features(rbf_ax, train_features_rbf, train_labels, 0, "s", "w", "b", "A train")
plot_features(rbf_ax, train_features_rbf, train_labels, 1, "o", "w", "r", "B train")
plot_features(rbf_ax, test_features_rbf, test_labels, 0, "s", "b", "w", "A test")
plot_features(rbf_ax, test_features_rbf, test_labels, 1, "o", "r", "w", "A test")

rbf_ax.set_ylabel("Principal component #1")
rbf_ax.set_xlabel("Principal component #0")
rbf_ax.set_title("Projection of training data\n using KernelPCA")
plt.show()
```


Projection of training and test data
using KPCA with Quantum Kernel
Principal component #1
Principal component #0
Projection of training data
using KernelPCA
Principal component #1
Principal component #0