

Test Case - 1: Dimensionality Reduction

```
X = np.random.rand(100, 10)
task = "dimensionality_reduction"
algorithm = "PCA"
hyperparameters = {"n_components": 5}
model = ml_function(X, None, task, algorithm, hyperparameters)

output:

New shape of X: (100, 5)
```

Test Case - 2: Anomaly Detection

```
X = np.random.rand(100, 5)
task = "anomaly_detection"
algorithm = "IsolationForest"
hyperparameters = {"contamination": 0.1}
model = ml_function(X, None, task, algorithm, hyperparameters)

output:

Number of anomalies: 10
```

Test Case - 3: Classification

```
X = np.array([[1,2], [2,1], [2,3], [3,2], [4,3], [3,4]])
y = np.array([0, 0, 0, 1, 1, 1])
task = "classification"
algorithm = "SVC"
hyperparameters = {"kernel": "linear", "C": 1}
model = ml_function(X, y, task, algorithm, hyperparameters)

output:

Accuracy: 0.5
```

Test Case - 4: Regression

```
X = np.random.rand(100, 5)
y = np.random.rand(100)
task = "regression"
algorithm = "LinearRegression"
hyperparameters = {"fit_intercept": True}
model = ml_function(X, y, task, algorithm, hyperparameters)

output:

MSE: 0.07232040625738016
```