

Supervised and Unsupervised Learning: Code Examples

1. Supervised Learning: Logistic Regression

In this example, we classify whether a flower is of the species Iris Setosa based on its petal dimensions.

Code:

```
from sklearn.datasets import load_iris

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score


# Load the Iris dataset

iris = load_iris()

X = iris.data[:, :2] # Using only the first two features for simplicity

y = (iris.target == 0).astype(int) # Binary classification: Is it Iris Setosa?


# Split the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)


# Train the model

model = LogisticRegression()

model.fit(X_train, y_train)


# Make predictions

y_pred = model.predict(X_test)
```

```
# Evaluate the model
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))
```

2. Unsupervised Learning: K-Means Clustering

In this example, we cluster data into groups using the Iris dataset.

Code:

```
from sklearn.cluster import KMeans
```

```
import matplotlib.pyplot as plt
```

```
from sklearn.datasets import load_iris
```

```
# Load the Iris dataset
```

```
iris = load_iris()
```

```
X = iris.data[:, :2] # Using only the first two features for visualization
```

```
# Apply K-Means Clustering
```

```
kmeans = KMeans(n_clusters=3, random_state=42)
```

```
y_kmeans = kmeans.fit_predict(X)
```

```
# Visualize the clusters
```

```
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, cmap='viridis')
```

```
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red', marker='X',  
label='Centroids')
```

```
plt.xlabel('Feature 1')
```

```
plt.ylabel('Feature 2')
```

```
plt.title('K-Means Clustering')
```

```
plt.legend()
```

```
plt.show()
```