1. Random Forest Classifier

from sklearn.datasets import load\_iris

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# Load Iris dataset

iris = load\_iris()

X, y = iris.data, iris.target

# Split the dataset 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 Random Forest model

rf = RandomForestClassifier(n\_estimators=100)

rf.fit(X\_train, y\_train)

# Make predictions

y\_pred = rf.predict(X\_test)

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

2. Principal Component Analysis (PCA)

from sklearn.datasets import load\_wine

from sklearn.decomposition import PCA

import matplotlib.pyplot as plt

# Load Wine dataset

data = load\_wine()

X, y = data.data, data.target

# Apply PCA to reduce to 2 components

pca = PCA(n\_components=2)

X\_pca = pca.fit\_transform(X)

# Plot the reduced data

plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c=y, cmap='viridis', edgecolor='k')

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.title('PCA of Wine Dataset')

plt.colorbar()

plt.show()

3. Linear Regression on a Simple Dataset

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

# Example dataset: house sizes (sq ft) and prices

X = np.array([[800], [1000], [1200], [1500], [1800]]).reshape(-1, 1)

y = np.array([150000, 200000, 250000, 300000, 350000])

# Train a linear regression model

model = LinearRegression()

model.fit(X, y)

# Predict house prices

predictions = model.predict(X)

# Plot the results

plt.scatter(X, y, color='blue')

plt.plot(X, predictions, color='red')

plt.xlabel('House Size (sq ft)')

plt.ylabel('Price ($)')

plt.title('Linear Regression')

plt.show()