TASK 1

import numpy as np

# Example dataset: House prices with square footage, number of bedrooms, and number of bathrooms

data = np.array([

[2104, 3, 2, 399900],

[1600, 3, 2, 329900],

[2400, 3, 3, 369000],

[1416, 2, 2, 232000],

[3000, 4, 3, 539900],

[1985, 4, 2, 299900],

[1534, 3, 2, 314900],

[1427, 3, 2, 198999],

[1380, 3, 2, 212000],

[1494, 3, 3, 242500]

])

# Split the data into input features X and target variable y

X = data[:, :-1]

y = data[:, -1]

# Normalize the features

def normalize\_features(X):

mu = np.mean(X, axis=0)

sigma = np.std(X, axis=0)

X\_norm = (X - mu) / sigma

return X\_norm, mu, sigma

X\_norm, mu, sigma = normalize\_features(X)

# Add a column of ones to X\_norm to account for the intercept term

m = X\_norm.shape[0]

X\_norm = np.concatenate([np.ones((m, 1)), X\_norm], axis=1)

# Compute the parameters using the normal equation

def normal\_equation(X, y):

X\_transpose = np.transpose(X)

theta = np.linalg.inv(X\_transpose @ X) @ X\_transpose @ y

return theta

theta = normal\_equation(X\_norm, y)

print("Theta parameters:", theta)

# Predict the price of a new house with given features

def predict\_price(features, theta, mu, sigma):

features\_norm = (features - mu) / sigma

features\_norm = np.concatenate(([1], features\_norm)) # Add the intercept term

price = features\_norm @ theta

return price

# Example: Predict the price of a house with 2000 sq-ft, 3 bedrooms, 2 bathrooms

features = np.array([2000, 3, 2])

predicted\_price = predict\_price(features, theta, mu, sigma)

print("Predicted price:", predicted\_price)