```
!pip install utils
```

```
Collecting utils
  Downloading utils-1.0.2.tar.gz (13 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: utils
  Building wheel for utils (setup.py) ... done
  Created wheel for utils: filename=utils-1.0.2-py2.py3-none-any.whl size=13906 sha256=5e7ce28b1082d11ac64a84fecd2f4f44758657fa3cc4
  Stored in directory: /root/.cache/pip/wheels/b8/39/f5/9d0ca31dba85773ececf0a7f5469f18810e1c8a8ed9da28ca7
Successfully built utils
Installing collected packages: utils
Successfully installed utils-1.0.2
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from utils import *
import math
from sklearn.linear_model import LinearRegression

%matplotlib inline


# load dataset
data = pd.read_csv('/content/sample_data/california_housing_train.csv')       #store the given data in pandas data frame
data.head(5)            #checking if the data is loaded correctly
```

| | longitude | latitude | housing_median_age | total_rooms | total_bedrooms | population |
|---|---|---|---|---|---|---|
| **0** | -114.31 | 34.19 | 15.0 | 5612.0 | 1283.0 | 1015.0 |
| **1** | -114.47 | 34.40 | 19.0 | 7650.0 | 1901.0 | 1129.0 |
| **2** | -114.56 | 33.69 | 17.0 | 720.0 | 174.0 | 333.0 |
| **3** | -114.57 | 33.64 | 14.0 | 1501.0 | 337.0 | 515.0 |
| **4** | -114.57 | 33.57 | 20.0 | 1454.0 | 326.0 | 624.0 |

```
data.describe()
```

| | longitude | latitude | housing_median_age | total_rooms | total_bedrooms | |
|---|---|---|---|---|---|---|
| **count** | 17000.000000 | 17000.000000 | 17000.000000 | 17000.000000 | 17000.000000 | 1 |
| **mean** | -119.562108 | 35.625225 | 28.589353 | 2643.664412 | 539.410824 | |
| **std** | 2.005166 | 2.137340 | 12.586937 | 2179.947071 | 421.499452 | |
| **min** | -124.350000 | 32.540000 | 1.000000 | 2.000000 | 1.000000 | |
| **25%** | -121.790000 | 33.930000 | 18.000000 | 1462.000000 | 297.000000 | |
| **50%** | -118.490000 | 34.250000 | 29.000000 | 2127.000000 | 434.000000 | |
| **75%** | -118.000000 | 37.720000 | 37.000000 | 3151.250000 | 648.250000 | |
| **max** | -114.310000 | 41.950000 | 52.000000 | 37937.000000 | 6445.000000 | 3 |

```
data.isnull().sum()
```

```
longitude             0
latitude              0
housing_median_age    0
total_rooms           0
total_bedrooms        0
population            0
households            0
median_income         0
median_house_value    0
dtype: int64
```

```
x = data.drop(columns=['median_house_value'])  # Features
x = x.values
x.shape
```

```
(17000, 8)
```

```
y = data['median_house_value']  # Target
y = y.values.reshape(-1,1)
y.shape
```

```
(17000, 1)
```

```
# Add a column of ones to the feature matrix to account for the bias term
X_with_bias = np.hstack((np.ones((x.shape[0], 1)), x))
```

```
# Initialize weights (coefficients) and bias term
weights = np.random.rand(X_with_bias.shape[1], 1)  # Assuming x has 8 features
bias = np.random.rand(1)

# Compute hypothesis for each sample
hypothesis = np.dot(X_with_bias, weights)

print("Hypothesis shape:", hypothesis.shape)
```

```
Hypothesis shape: (17000, 1)
```

```
weights.shape
```

```
(9, 1)
```

```
# Compute hypothesis
hypothesis = np.dot(X_with_bias, weights)
loss = np.dot((y - hypothesis).T, (y - hypothesis))
print("cost:", loss)
```

```
cost: [[9.29737429e+14]]
```

```
temp = np.dot(X_with_bias.T, (y - (hypothesis)))
temp.shape
```

```
(9, 1)
```

```
for i in range (1000):
   hypothesis = np.dot(X_with_bias, weights)
   loss = np.dot((y-hypothesis).T, (y-hypothesis))
   weights = weights - 0.000000001*((2/17000)*(np.dot(X_with_bias.T, (y - (hypothesis)))))
   print(f"iteration = {i}, loss = {loss}, wo = {weights[0]}, w1 = {weights[1]}")
```

```
iteration = 0, loss = [[1.62427776e+68]], wo = [-5.03964689e+24], w1 = [6.02063198e+26]
iteration = 1, loss = [[1.72756937e+68]], wo = [-5.19741872e+24], w1 = [6.20911465e+26]
iteration = 2, loss = [[1.83742953e+68]], wo = [-5.36012977e+24], w1 = [6.40349798e+26]
iteration = 3, loss = [[1.95427595e+68]], wo = [-5.52793468e+24], w1 = [6.60396671e+26]
iteration = 4, loss = [[2.07855291e+68]], wo = [-5.70099291e+24], w1 = [6.81071133e+26]
iteration = 5, loss = [[2.21073293e+68]], wo = [-5.87946892e+24], w1 = [7.02392833e+26]
iteration = 6, loss = [[2.35131858e+68]], wo = [-6.06353232e+24], w1 = [7.24382032e+26]
iteration = 7, loss = [[2.5008444e+68]], wo = [-6.25335803e+24], w1 = [7.47059628e+26]
iteration = 8, loss = [[2.65987892e+68]], wo = [-6.44912646e+24], w1 = [7.70447172e+26]
iteration = 9, loss = [[2.82902682e+68]], wo = [-6.65102363e+24], w1 = [7.9456689e+26]
iteration = 10, loss = [[3.00893122e+68]], wo = [-6.85924143e+24], w1 = [8.19441702e+26]
iteration = 11, loss = [[3.20027616e+68]], wo = [-7.07397771e+24], w1 = [8.45095248e+26]
iteration = 12, loss = [[3.40378917e+68]], wo = [-7.29543656e+24], w1 = [8.71551908e+26]
iteration = 13, loss = [[3.62024405e+68]], wo = [-7.52382843e+24], w1 = [8.98836823e+26]
iteration = 14, loss = [[3.8504638e+68]], wo = [-7.75937036e+24], w1 = [9.26975923e+26]
iteration = 15, loss = [[4.09532376e+68]], wo = [-8.0022862e+24], w1 = [9.55995949e+26]
iteration = 16, loss = [[4.35575494e+68]], wo = [-8.25280679e+24], w1 = [9.8592448e+26]
iteration = 17, loss = [[4.63274753e+68]], wo = [-8.51117021e+24], w1 = [1.01678996e+27]
iteration = 18, loss = [[4.92735474e+68]], wo = [-8.77762199e+24], w1 = [1.04862171e+27]
iteration = 19, loss = [[5.2406967e+68]], wo = [-9.05241534e+24], w1 = [1.08145e+27]
iteration = 20, loss = [[5.5739648e+68]], wo = [-9.33581141e+24], w1 = [1.11530601e+27]
iteration = 21, loss = [[5.9284262e+68]], wo = [-9.62807951e+24], w1 = [1.15022192e+27]
iteration = 22, loss = [[6.30542863e+68]], wo = [-9.92949739e+24], w1 = [1.18623092e+27]
iteration = 23, loss = [[6.70640552e+68]], wo = [-1.02403515e+25], w1 = [1.22336721e+27]
iteration = 24, loss = [[7.13288146e+68]], wo = [-1.05609373e+25], w1 = [1.2616661e+27]
iteration = 25, loss = [[7.586478e+68]], wo = [-1.08915593e+25], w1 = [1.30116398e+27]
iteration = 26, loss = [[8.0689198e+68]], wo = [-1.12325318e+25], w1 = [1.34189839e+27]
iteration = 27, loss = [[8.5820412e+68]], wo = [-1.15841789e+25], w1 = [1.38390803e+27]
iteration = 28, loss = [[9.12779319e+68]], wo = [-1.19468347e+25], w1 = [1.42723283e+27]
iteration = 29, loss = [[9.70825081e+68]], wo = [-1.23208438e+25], w1 = [1.47191396e+27]
iteration = 30, loss = [[1.03256211e+69]], wo = [-1.27065617e+25], w1 = [1.51799389e+27]
iteration = 31, loss = [[1.09822514e+69]], wo = [-1.3104355e+25], w1 = [1.5655164e+27]
iteration = 32, loss = [[1.16806383e+69]], wo = [-1.35146016e+25], w1 = [1.61452666e+27]
iteration = 33, loss = [[1.24234373e+69]], wo = [-1.39376914e+25], w1 = [1.66507123e+27]
iteration = 34, loss = [[1.32134726e+69]], wo = [-1.43740266e+25], w1 = [1.71719817e+27]
iteration = 35, loss = [[1.40537481e+69]], wo = [-1.48240217e+25], w1 = [1.77095699e+27]
iteration = 36, loss = [[1.49474587e+69]], wo = [-1.52881044e+25], w1 = [1.8263988e+27]
iteration = 37, loss = [[1.58980024e+69]], wo = [-1.57667157e+25], w1 = [1.88357627e+27]
iteration = 38, loss = [[1.69089933e+69]], wo = [-1.62603105e+25], w1 = [1.94254375e+27]
iteration = 39, loss = [[1.79842756e+69]], wo = [-1.67693579e+25], w1 = [2.00335727e+27]
```

```
iteration = 40, loss = [[1.91279376e+69]], wo = [-1.72943415e+25], w1 = [2.06607463e+27]
iteration = 41, loss = [[2.03443276e+69]], wo = [-1.78357604e+25], w1 = [2.13075542e+27]
iteration = 42, loss = [[2.16380708e+69]], wo = [-1.8394129e+25], w1 = [2.19746112e+27]
iteration = 43, loss = [[2.30140861e+69]], wo = [-1.89699779e+25], w1 = [2.26625512e+27]
iteration = 44, loss = [[2.44776054e+69]], wo = [-1.95638545e+25], w1 = [2.33720279e+27]
iteration = 45, loss = [[2.60341934e+69]], wo = [-2.0176323e+25], w1 = [2.41037156e+27]
iteration = 46, loss = [[2.76897683e+69]], wo = [-2.08079655e+25], w1 = [2.48583096e+27]
iteration = 47, loss = [[2.94506252e+69]], wo = [-2.14593824e+25], w1 = [2.5636527e+27]
iteration = 48, loss = [[3.1323459e+69]], wo = [-2.21311925e+25], w1 = [2.64391074e+27]
iteration = 49, loss = [[3.33153907e+69]], wo = [-2.28240345e+25], w1 = [2.72668135e+27]
iteration = 50, loss = [[3.5433994e+69]], wo = [-2.35385666e+25], w1 = [2.81204318e+27]
iteration = 51, loss = [[3.76873242e+69]], wo = [-2.4275468e+25], w1 = [2.90007736e+27]
iteration = 52, loss = [[4.00839489e+69]], wo = [-2.50354389e+25], w1 = [2.99086755e+27]
iteration = 53, loss = [[4.26329805e+69]], wo = [-2.58192015e+25], w1 = [3.08450003e+27]
iteration = 54, loss = [[4.5344111e+69]], wo = [-2.66275007e+25], w1 = [3.18106379e+27]
iteration = 55, loss = [[4.82276486e+69]], wo = [-2.74611046e+25], w1 = [3.28065057e+27]
iteration = 56, loss = [[5.12945571e+69]], wo = [-2.83208055e+25], w1 = [3.38335504e+27]
```

```
#prediction
```

Start coding or generate with AI.