

Task 1: Create a Linear Dataset

Your goal is to create a simple dataset consisting of a single feature and a label as follows:

- Assign a sequence of integers from 6 to 20 (inclusive) to a NumPy array named feature.
- Assign 15 values to a NumPy array named label such that:

Snapshots:

Code:

```
import numpy as np
feature = np.arange(6, 21)
print(feature)
label = (feature * 3) + 4
print(label)
```

Figure 1 Task 1 (Code)

Output:

```
[ 6  7  8  9 10 11 12 13 14 15 16 17 18 19 20]
[22 25 28 31 34 37 40 43 46 49 52 55 58 61 64]
```

Figure 2 Task 1 (Output)

Task 2: Add Some Noise to the Dataset

- To make your dataset a little more realistic, insert a little random noise into each element of the label array you already created. To be more precise, modify each value assigned to label by adding a *different* random floating-point value between -2 and +2.
- Don't rely on broadcasting. Instead, create a noise array having the same dimension as label.

Snapshots:

Code:

```
noise = (np.random.random([15]) * 4) - 2
print(noise)
label = label + noise
print(label)
```

Figure 3 Task 2 (Code)

Output:

```
[-1.23464776  1.93997833  1.58754278 -1.30213315  1.57071026  1.14559956
 -0.45835645 -1.72122176  0.74343461  0.11895447 -1.37305733  1.46107659
 1.00290929  0.77250068  0.23289265]

[18.82815506 25.35124871 29.92249875 29.37294403 37.53638112 38.0077609
 39.86915968 40.01690992 46.39634177 48.36522965 50.95373596 58.01272679
 59.28442874 63.24233248 65.32433049]
```

Figure 4 Task 2 (Output)

Task 3:

- Create a dataset with 2 features with 30 values/exaples each generating randomly between -10 to +10 inclusive
- Create a label for these features such that each label is twice the sum of features

Snapshots:

Code:

```
import numpy as np
import matplotlib.pyplot as plt
features=(np.random.random([30,2])*20)-10
feature1=(features[:,0])
feature2=(features[:,1])
print(feature1)
print(feature2)
label=(feature1+feature2)*2
print(label)
plt.scatter(feature1,label)
plt.scatter(feature2,label)
plt.show()
```

Figure 5 Task 3 (Code)

Output:

```
[ 7.80283658 -6.40453857  8.25901268  4.41556267  7.0726599  -0.22547998
 4.68178219  3.11685339  6.74494659 -9.8329049  -0.03635661  4.70642853
-0.67204747  8.75535176  9.41012751  5.31120577 -8.45852772  2.29402984
 5.8554843  -9.82642311 -4.10099033  0.76525024 -9.04301315 -1.14012617
-9.00283775 -3.83781615  4.33270507 -3.35849953 -0.9349376  0.81548826]
[ 2.65479724 -4.62055919 -6.00484074 -7.45044529 -1.73158076 -5.2042009
 1.5328792  3.93497115  0.40795793 -4.58079812 -8.17323088  7.73905136
-7.0176676  6.54425398  3.47086914  7.70339931 -7.45230157  1.13652698
-7.77626075  9.01186896 -6.75420157 -4.96753175  7.48671793 -1.63983159
 0.71499775 -8.08202848 -8.8001341  -8.41104259 -7.16076924 -3.69088884]
[ 20.91526764 -22.05019551  4.50834388 -6.06976525  10.68215828
-10.85936176  12.42932277  14.10364908  14.30580905 -28.82740604
-16.41917499  24.89095978 -15.37943015  30.59921149  25.76199331
 26.02921016 -31.82165858  6.86111363 -3.8415529  -1.6291083
-21.7103838  -8.40456302 -3.11259044 -5.55991551 -16.57567999
-23.83968926 -8.93485805 -23.53908425 -16.1914137  -5.75080115]
```

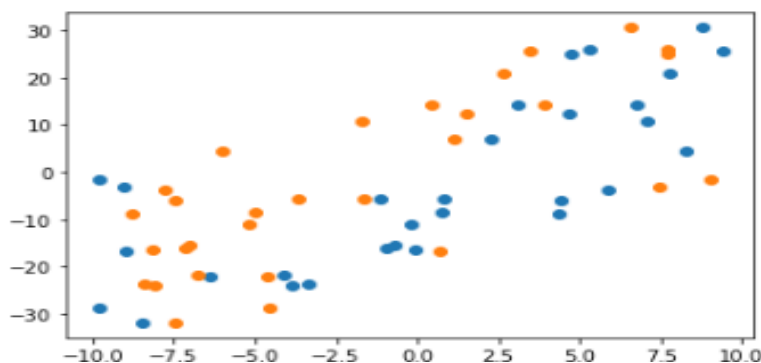


Figure 6 Task 3 (Output)

Name: Qasim Khan
CMSID: 49375