Task 1.1:

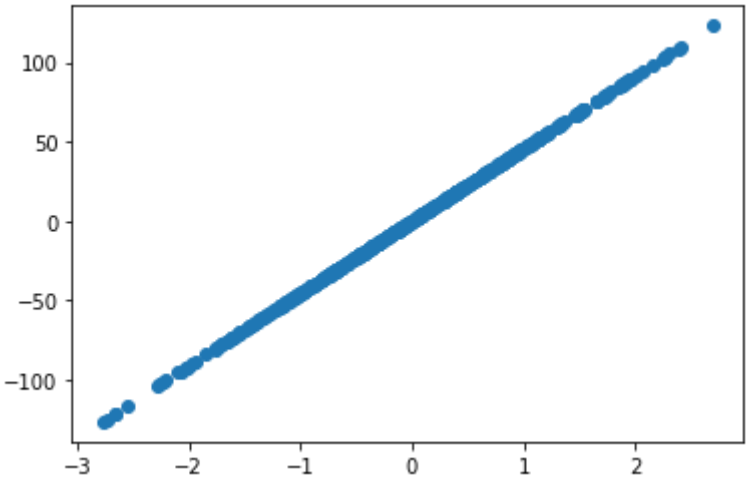
1. Generate a regression dataset with the folllwing details:
    - 500 samples
    - 1 feature
2. Visualize the dataset

```
from sklearn.datasets import make_regression

X, y = make_regression(n_samples = 500, n_features = 1, random_state = 0

type(y)
```

```
numpy.ndarray
```

```
import matplotlib.pyplot as plt

plt.scatter(X, y)
```

```
<matplotlib.collections.PathCollection at 0x7febf128df50>
```



Task 1.2:

1. Introduce noise (standard deviation of the gaussian noise= 10) in the dataset created in 2.1
2. Visualize the dataset

Task 2:

1. Generate a classification data set using make_classification with the following details:
    - 2 features
    - 2 classes
    - one cluster per class

```
from sklearn.datasets import make_classification
```

```
?make_classification
```

```
X, y = make_classification(n_samples = 500, n_features = 2, n_informativ
```

```
X.shape
```

```
(500, 2)
```

```
y.shape
```

```
(500,)
```

```
X[0:10]
```

```
array([[ 0.87749497,  1.07036032],
       [ 2.50029538,  2.73178226],
       [ 2.09022098,  2.77056377],
       [ 0.61599119,  0.52395415],
       [-2.99312332, -1.8631013 ],
       [ 0.62388252,  0.04694998],
       [ 1.43577864,  1.63717902],
       [ 1.34424638,  2.54549463],
       [-1.11984097, -1.46654328],
       [-1.1771694 , -0.85528775]])
```

```
X[0:10,1]
```

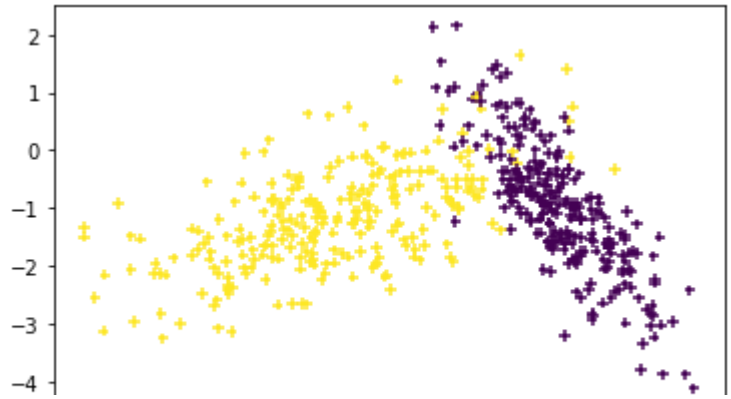
```
array([ 1.07036032,  2.73178226,  2.77056377,  0.52395415, -1.8631013 ,
        0.04694998,  1.63717902,  2.54549463, -1.46654328, -0.85528775])
```

```
y[0:10]
```

```
array([0, 0, 0, 0, 1, 1, 0, 0, 1, 1])
```

```
plt.scatter(X[:,0], X[:, 1], marker = '+', c = y, s=30)
```

```
<matplotlib.collections.PathCollection at 0x7febec38bb10>
```



Task 3

Create clustering dataset with the following details:

- 2 features
- 3 clusters

```
from sklearn.datasets import make_blobs
```

```
X, y = make_blobs(n_samples = 500, n_features = 2, centers = 3)
```

```
X.shape
```

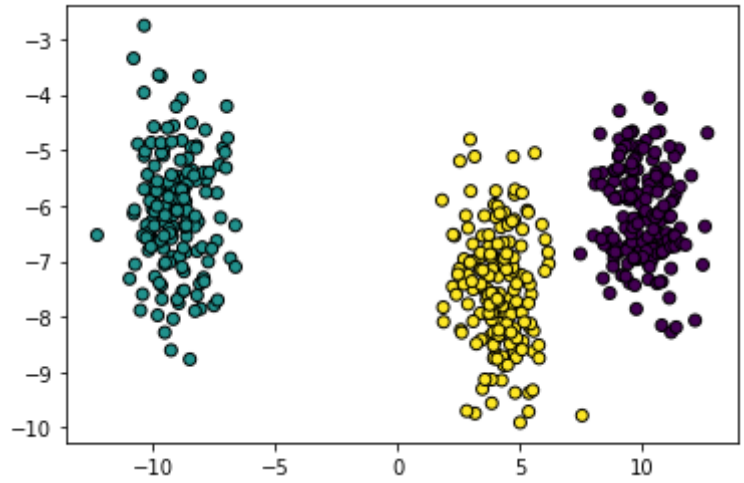
```
(500, 2)
```

```
y[0:10]
```

```
array([1, 0, 2, 0, 2, 0, 2, 0, 2, 2])
```

```
plt.scatter(X[:, 0], X[:, 1], edgecolor = 'k', c = y)
```

```
<matplotlib.collections.PathCollection at 0x7febec3f6b50>
```



Task 4

1. Load diabetes dataset from sklearn library.
2. Examine the number of features and samples in it.
3. Is it a clssification or regression data set?

```
from sklearn.datasets import load_diabetes

X, y = load_diabetes(return_X_y = True)

# type(diabetes_data)
type(X)
```

```
    numpy.ndarray
```

```
diabetes_data[1]
```

```
    array([151.,  75., 141., 206., 135.,  97., 138.,  63., 110., 310., 101.,
            69., 179., 185., 118., 171., 166., 144.,  97., 168.,  68.,  49.,
            68., 245., 184., 202., 137.,  85., 131., 283., 129.,  59., 341.,
            87.,  65., 102., 265., 276., 252.,  90., 100.,  55.,  61.,  92.,
           259.,  53., 190., 142.,  75., 142., 155., 225.,  59., 104., 182.,
           128.,  52.,  37., 170., 170.,  61., 144.,  52., 128.,  71., 163.,
           150.,  97., 160., 178.,  48., 270., 202., 111.,  85.,  42., 170.,
           200., 252., 113., 143.,  51.,  52., 210.,  65., 141.,  55., 134.,
            42., 111.,  98., 164.,  48.,  96.,  90., 162., 150., 279.,  92.,
            83., 128., 102., 302., 198.,  95.,  53., 134., 144., 232.,  81.,
           104.,  59., 246., 297., 258., 229., 275., 281., 179., 200., 200.,
           173., 180.,  84., 121., 161.,  99., 109., 115., 268., 274., 158.,
           107.,  83., 103., 272.,  85., 280., 336., 281., 118., 317., 235.,
            60., 174., 259., 178., 128.,  96., 126., 288.,  88., 292.,  71.,
           197., 186.,  25.,  84.,  96., 195.,  53., 217., 172., 131., 214.,
            59.,  70., 220., 268., 152.,  47.,  74., 295., 101., 151., 127.,
           237., 225.,  81., 151., 107.,  64., 138., 185., 265., 101., 137.,
           143., 141.,  79., 292., 178.,  91., 116.,  86., 122.,  72., 129.,
           142.,  90., 158.,  39., 196., 222., 277.,  99., 196., 202., 155.,
            77., 191.,  70.,  73.,  49.,  65., 263., 248., 296., 214., 185.,
            78.,  93., 252., 150.,  77., 208.,  77., 108., 160.,  53., 220.,
           154., 259.,  90., 246., 124.,  67.,  72., 257., 262., 275., 177.,
            71.,  47., 187., 125.,  78.,  51., 258., 215., 303., 243.,  91.,
           150., 310., 153., 346.,  63.,  89.,  50.,  39., 103., 308., 116.,
           145.,  74.,  45., 115., 264.,  87., 202., 127., 182., 241.,  66.,
            94., 283.,  64., 102., 200., 265.,  94., 230., 181., 156., 233.,
            60., 219.,  80.,  68., 332., 248.,  84., 200.,  55.,  85.,  89.,
            31., 129.,  83., 275.,  65., 198., 236., 253., 124.,  44., 172.,
           114., 142., 109., 180., 144., 163., 147.,  97., 220., 190., 109.,
           191., 122., 230., 242., 248., 249., 192., 131., 237.,  78., 135.,
           244., 199., 270., 164.,  72.,  96., 306.,  91., 214.,  95., 216.,
           263., 178., 113., 200., 139., 139.,  88., 148.,  88., 243.,  71.,
            77., 109., 272.,  60.,  54., 221.,  90., 311., 281., 182., 321.,
            58., 262., 206., 233., 242., 123., 167.,  63., 197.,  71., 168.,
```

```
       140., 217., 121., 235., 245.,  40.,  52., 104., 132.,  88.,  69.,
       219.,  72., 201., 110.,  51., 277.,  63., 118.,  69., 273., 258.,
        43., 198., 242., 232., 175.,  93., 168., 275., 293., 281.,  72.,
       140., 189., 181., 209., 136., 261., 113., 131., 174., 257.,  55.,
        84.,  42., 146., 212., 233.,  91., 111., 152., 120.,  67., 310.,
        94., 183.,  66., 173.,  72.,  49.,  64.,  48., 178., 104., 132.,
       220.,  57.])
```

```
?load_diabetes
```

```
type(diabetes_data)
```

```
    sklearn.utils.Bunch
```

```
diabetes_data.data
```

```
    array([[ 0.03807591,  0.05068012,  0.06169621, ..., -0.00259226,
             0.01990842, -0.01764613],
           [-0.00188202, -0.04464164, -0.05147406, ..., -0.03949338,
            -0.06832974, -0.09220405],
           [ 0.08529891,  0.05068012,  0.04445121, ..., -0.00259226,
             0.00286377, -0.02593034],
           ...,
           [ 0.04170844,  0.05068012, -0.01590626, ..., -0.01107952,
            -0.04687948,  0.01549073],
           [-0.04547248, -0.04464164,  0.03906215, ...,  0.02655962,
             0.04452837, -0.02593034],
           [-0.04547248, -0.04464164, -0.0730303 , ..., -0.03949338,
            -0.00421986,  0.00306441]])
```

```
X = diabetes_data.data
```

```
X.shape
```

```
    (442, 10)
```

```
y = diabetes_data.target
```

```
y.shape
```

```
    (442,)
```

```
diabetes_data.feature_names
```

```
    ['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6']
```

```
y
```

```
    array([151.,  75., 141., 206., 135.,  97., 138.,  63., 110., 310., 101.,
            69., 179., 185., 118., 171., 166., 144.,  97., 168.,  68.,  49.,
```

```
 68., 245., 184., 202., 137.,  85., 131., 283., 129.,  59., 341.,
 87.,  65., 102., 265., 276., 252.,  90., 100.,  55.,  61.,  92.,
259.,  53., 190., 142.,  75., 142., 155., 225.,  59., 104., 182.,
128.,  52.,  37., 170., 170.,  61., 144.,  52., 128.,  71., 163.,
150.,  97., 160., 178.,  48., 270., 202., 111.,  85.,  42., 170.,
200., 252., 113., 143.,  51.,  52., 210.,  65., 141.,  55., 134.,
 42., 111.,  98., 164.,  48.,  96.,  90., 162., 150., 279.,  92.,
 83., 128., 102., 302., 198.,  95.,  53., 134., 144., 232.,  81.,
104.,  59., 246., 297., 258., 229., 275., 281., 179., 200., 200.,
173., 180.,  84., 121., 161.,  99., 109., 115., 268., 274., 158.,
107.,  83., 103., 272.,  85., 280., 336., 281., 118., 317., 235.,
 60., 174., 259., 178., 128.,  96., 126., 288.,  88., 292.,  71.,
197., 186.,  25.,  84.,  96., 195.,  53., 217., 172., 131., 214.,
 59.,  70., 220., 268., 152.,  47.,  74., 295., 101., 151., 127.,
237., 225.,  81., 151., 107.,  64., 138., 185., 265., 101., 137.,
143., 141.,  79., 292., 178.,  91., 116.,  86., 122.,  72., 129.,
142.,  90., 158.,  39., 196., 222., 277.,  99., 196., 202., 155.,
 77., 191.,  70.,  73.,  49.,  65., 263., 248., 296., 214., 185.,
 78.,  93., 252., 150.,  77., 208.,  77., 108., 160.,  53., 220.,
154., 259.,  90., 246., 124.,  67.,  72., 257., 262., 275., 177.,
 71.,  47., 187., 125.,  78.,  51., 258., 215., 303., 243.,  91.,
150., 310., 153., 346.,  63.,  89.,  50.,  39., 103., 308., 116.,
145.,  74.,  45., 115., 264.,  87., 202., 127., 182., 241.,  66.,
 94., 283.,  64., 102., 200., 265.,  94., 230., 181., 156., 233.,
 60., 219.,  80.,  68., 332., 248.,  84., 200.,  55.,  85.,  89.,
 31., 129.,  83., 275.,  65., 198., 236., 253., 124.,  44., 172.,
114., 142., 109., 180., 144., 163., 147.,  97., 220., 190., 109.,
191., 122., 230., 242., 248., 249., 192., 131., 237.,  78., 135.,
244., 199., 270., 164.,  72.,  96., 306.,  91., 214.,  95., 216.,
263., 178., 113., 200., 139., 139.,  88., 148.,  88., 243.,  71.,
 77., 109., 272.,  60.,  54., 221.,  90., 311., 281., 182., 321.,
 58., 262., 206., 233., 242., 123., 167.,  63., 197.,  71., 168.,
140., 217., 121., 235., 245.,  40.,  52., 104., 132.,  88.,  69.,
219.,  72., 201., 110.,  51., 277.,  63., 118.,  69., 273., 258.,
 43., 198., 242., 232., 175.,  93., 168., 275., 293., 281.,  72.,
140., 189., 181., 209., 136., 261., 113., 131., 174., 257.,  55.,
 84.,  42., 146., 212., 233.,  91., 111., 152., 120.,  67., 310.,
 94., 183.,  66., 173.,  72.,  49.,  64.,  48., 178., 104., 132.,
220.,  57.])
```

Task 5

1. Fetch `kddcup99` dataset from sklearn library.
2. Examine the number of features and samples in it.
3. Is it a clssification or regression data set?

```
from sklearn.datasets import fetch_kddcup99

kdd_data = fetch_kddcup99()

X = kdd_data.data
```

```
y = kdd_data.target
```

```
kdd_data.feature_names
```

```
['duration',
 'protocol_type',
 'service',
 'flag',
 'src_bytes',
 'dst_bytes',
 'land',
 'wrong_fragment',
 'urgent',
 'hot',
 'num_failed_logins',
 'logged_in',
 'num_compromised',
 'root_shell',
 'su_attempted',
 'num_root',
 'num_file_creations',
 'num_shells',
 'num_access_files',
 'num_outbound_cmds',
 'is_host_login',
 'is_guest_login',
 'count',
 'srv_count',
 'serror_rate',
 'srv_serror_rate',
 'rerror_rate',
 'srv_rerror_rate',
 'same_srv_rate',
 'diff_srv_rate',
 'srv_diff_host_rate',
 'dst_host_count',
 'dst_host_srv_count',
 'dst_host_same_srv_rate',
 'dst_host_diff_srv_rate',
 'dst_host_same_src_port_rate',
 'dst_host_srv_diff_host_rate',
 'dst_host_serror_rate',
 'dst_host_srv_serror_rate',
 'dst_host_rerror_rate',
 'dst_host_srv_rerror_rate']
```

```
X.shape
```

```
(494021, 41)
```

Double-click (or enter) to edit

```
y[490000:494021]
```

```
array([b'smurf.', b'smurf.', b'smurf.', ..., b'normal.', b'normal.',
```

```
       b'normal.'], dtype=object)
```

```
set(y)
```

```
{b'back.',
 b'buffer_overflow.',
 b'ftp_write.',
 b'guess_passwd.',
 b'imap.',
 b'ipsweep.',
 b'land.',
 b'loadmodule.',
 b'multihop.',
 b'neptune.',
 b'nmap.',
 b'normal.',
 b'perl.',
 b'phf.',
 b'pod.',
 b'portsweep.',
 b'rootkit.',
 b'satan.',
 b'smurf.',
 b'spy.',
 b'teardrop.',
 b'warezclient.',
 b'warezmaster.'}
```

```
import numpy as np
```

```
unique, counts = np.unique(y, return_counts = True)
unique
```

```
array([b'back.', b'buffer_overflow.', b'ftp_write.', b'guess_passwd.',
       b'imap.', b'ipsweep.', b'land.', b'loadmodule.', b'multihop.',
       b'neptune.', b'nmap.', b'normal.', b'perl.', b'phf.', b'pod.',
       b'portsweep.', b'rootkit.', b'satan.', b'smurf.', b'spy.',
       b'teardrop.', b'warezclient.', b'warezmaster.'], dtype=object)
```

```
counts
```

```
array([  2203,     30,      8,     53,     12,   1247,     21,      9,
            7, 107201,    231,  97278,      3,      4,    264,   1040,
           10,   1589, 280790,      2,    979,   1020,     20])
```

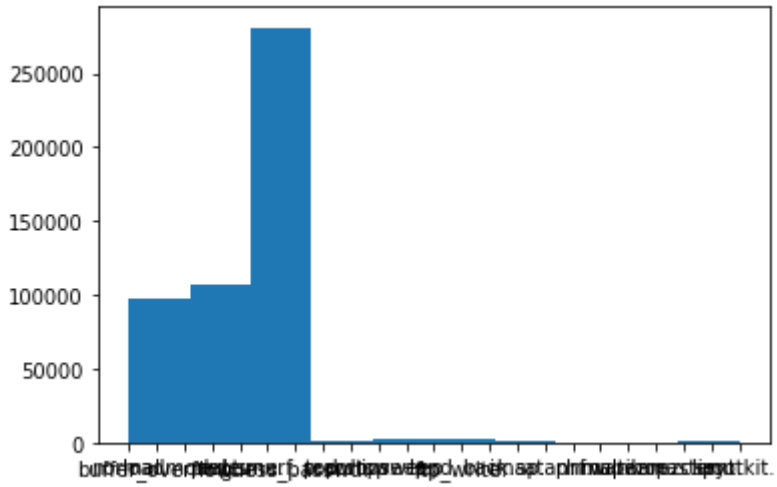
```
np.asarray((unique, counts)).T
```

```
array([[b'back.', 2203],
       [b'buffer_overflow.', 30],
       [b'ftp_write.', 8],
       [b'guess_passwd.', 53],
       [b'imap.', 12],
       [b'ipsweep.', 1247],
       [b'land.', 21],
       [b'loadmodule.', 9],
```

```
       [b'multihop.', 7],
       [b'neptune.', 107201],
       [b'nmap.', 231],
       [b'normal.', 97278],
       [b'perl.', 3],
       [b'phf.', 4],
       [b'pod.', 264],
       [b'portsweep.', 1040],
       [b'rootkit.', 10],
       [b'satan.', 1589],
       [b'smurf.', 280790],
       [b'spy.', 2],
       [b'teardrop.', 979],
       [b'warezclient.', 1020],
       [b'warezmaster.', 20]], dtype=object)
```

```
plt.hist(y)
```

```
(array([9.73170e+04, 1.07204e+05, 2.80843e+05, 1.24300e+03, 2.28700e+03,
        2.23200e+03, 1.60100e+03, 2.35000e+02, 2.70000e+01, 1.03200e+03]),
 array([ 0. ,  2.2,  4.4,  6.6,  8.8, 11. , 13.2, 15.4, 17.6, 19.8, 22. ]),
 <a list of 10 Patch objects>)
```



Task 6:

1. Open https://www.openml.org/d/37
2. Load diabetes data using its ID.
3. print the number of features in it.
4. Is it a regression dataset or classification?

Another: miceprotein, machine_cpu

```
from sklearn.datasets import fetch_openml
```

```
?fetch_openml
```

```
X, y = fetch_openml('diabetes', return_X_y = True)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/datasets/_openml.py:418: UserW
  " {version}.".format(name=name, version=res[0]["version"])
```

type(X) type(y)

X.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 8 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   preg    768 non-null    float64
 1   plas    768 non-null    float64
 2   pres    768 non-null    float64
 3   skin    768 non-null    float64
 4   insu    768 non-null    float64
 5   mass    768 non-null    float64
 6   pedi    768 non-null    float64
 7   age     768 non-null    float64
dtypes: float64(8)
memory usage: 48.1 KB
```

X.head(10)

	preg	plas	pres	skin	insu	mass	pedi	age
0	6.0	148.0	72.0	35.0	0.0	33.6	0.627	50.0
1	1.0	85.0	66.0	29.0	0.0	26.6	0.351	31.0
2	8.0	183.0	64.0	0.0	0.0	23.3	0.672	32.0
3	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21.0
4	0.0	137.0	40.0	35.0	168.0	43.1	2.288	33.0
5	5.0	116.0	74.0	0.0	0.0	25.6	0.201	30.0
6	3.0	78.0	50.0	32.0	88.0	31.0	0.248	26.0
7	10.0	115.0	0.0	0.0	0.0	35.3	0.134	29.0
8	2.0	197.0	70.0	45.0	543.0	30.5	0.158	53.0
9	8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0

y.unique()

```
['tested_positive', 'tested_negative']
Categories (2, object): ['tested_positive', 'tested_negative']
```

Task 7

```
import numpy as np
```

np.random.normal(size = 10)

```
temp = np.random.normal(size = 50)
temp
```

```
array([ 0.52291843, -0.29938688,  0.85037684,  2.15357886, -0.61455386,
       -2.11850519,  0.61234041,  2.39049604,  0.04559833, -0.28799228,
       -0.24725298, -0.49389015,  0.41676837, -0.8898663 , -0.05551963,
        1.32924722,  0.15193694,  0.32803023,  2.08426563,  1.71901009,
       -0.59266108, -0.36393329, -0.93564768, -0.8748603 ,  1.53638015,
        0.55116969,  0.52837961, -1.26878289,  0.33069881,  1.45391123,
        0.02529613, -0.39623791,  0.42238372, -1.12927833, -0.17574915,
       -1.96194783, -1.03809132,  0.2131697 ,  1.70974338, -0.41964833,
        0.81750428, -0.36852918, -0.51252091, -1.39963692,  1.14888416,
        2.97107096, -0.28516822,  0.76367312,  0.65959502,  0.24949665])
```

```
class_0_elements = temp.reshape(-1, 2)
```

```
class_1_elements = np.random.normal(size = 50).reshape(-1, 2)
```

```
X = np.concatenate([class_0_elements, class_1_elements])
```

```
X
```

```
array([[ -8.61961136,  -8.97335403],
       [ -7.16760454,  -7.6597931 ],
       [ -7.84546468,  -7.98178813],
       [ -8.08914561,  -9.82998074],
       [ -7.34647144,  -7.65124555],
       [ -7.10464764,  -8.32388811],
       [ -9.8713271 ,  -6.97383097],
       [ -7.49217827,  -6.17851757],
       [ -7.306525  ,  -7.11003984],
       [ -7.67866691,  -6.81523517],
       [ -7.2654799 , -10.052367  ],
       [ -5.99801163,  -7.59363156],
       [ -7.35977044,  -7.26238949],
       [ -8.1668052 ,  -7.62344952],
       [ -6.95470455,  -6.42908929],
       [ -7.76806924,  -9.4919028 ],
       [ -9.81740767,  -7.80433911],
       [ -8.48453481,  -9.05359352],
       [ -7.71813695,  -9.55816486],
       [ -7.28675518,  -8.13345911],
       [ -8.70068901,  -7.93830352],
       [ -8.70349689,  -5.77948043],
       [ -7.69165553,  -7.42779004],
```

```
       [ -7.48628085,  -8.93168105],
       [ -7.44651209,  -8.11567253],
       [  9.19605966,   7.96056987],
       [  8.369878  ,   7.39113422],
       [  8.08724116,   6.23618131],
       [  7.31218888,   8.69172511],
       [  8.45475126,   7.52897044],
       [  7.47063789,   6.82392204],
       [  7.6495622 ,   6.55626544],
       [  8.72203235,   8.99870473],
       [  7.89335121,   9.69059351],
       [  7.49882082,   8.66942748],
       [  8.0580962 ,   8.544451  ],
       [  8.01609692,   8.16659656],
       [ 10.15039441,   7.10243242],
       [  6.70615829,   9.53546456],
       [  6.69397729,   8.23830874],
       [  6.48161616,   9.06151735],
       [  9.19543054,   7.54360539],
       [  7.51469702,   7.00905139],
       [  9.13396735,   7.10192206],
       [  6.7375084 ,   6.0938758 ],
       [  9.16873424,   6.01063204],
       [  7.34218531,   7.76178278],
       [  7.72614281,   7.98889501],
       [  8.3910241 ,   9.06029407],
       [  8.86110608,   7.14818031]])
```

```
class_0_labels = np.zeros(25)
```

```
class_1_labels = np.ones(25)
```

```
class_1_labels
```

```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1.])
```

```
y = np.concatenate([class_0_labels, class_1_labels])
```

```
X
```

```
array([[ 0.52291843, -0.29938688],
       [ 0.85037684,  2.15357886],
       [-0.61455386, -2.11850519],
       [ 0.61234041,  2.39049604],
       [ 0.04559833, -0.28799228],
       [-0.24725298, -0.49389015],
       [ 0.41676837, -0.8898663 ],
       [-0.05551963,  1.32924722],
       [ 0.15193694,  0.32803023],
       [ 2.08426563,  1.71901009],
       [-0.59266108, -0.36393329],
       [-0.93564768, -0.8748603 ],
       [ 1.53638015,  0.55116969],
```

```
[ 0.52837961, -1.26878289],
[ 0.33069881,  1.45391123],
[ 0.02529613, -0.39623791],
[ 0.42238372, -1.12927833],
[-0.17574915, -1.96194783],
[-1.03809132,  0.2131697 ],
[ 1.70974338, -0.41964833],
[ 0.81750428, -0.36852918],
[-0.51252091, -1.39963692],
[ 1.14888416,  2.97107096],
[-0.28516822,  0.76367312],
[ 0.65959502,  0.24949665],
[-0.81055949,  0.45714978],
[ 0.63879224, -2.05247085],
[ 0.66867208, -0.31818555],
[ 1.22598487, -0.92219917],
[-0.91611836, -0.17097335],
[-1.32420174,  1.25836234],
[-2.18481679,  0.09881505],
[ 1.30033673, -0.8073465 ],
[ 0.42083267, -0.92842308],
[ 1.64155896,  0.31365576],
[-1.25912588, -0.44317041],
[ 0.31191316, -0.84266489],
[-1.05697013, -0.15187655],
[ 1.24718285, -0.14606193],
[-1.83667986,  1.10416816],
[-0.82498059, -0.1196364 ],
[-0.82213646,  0.32515594],
[ 0.19335724, -1.25064791],
[ 0.71664809, -1.09190562],
[-1.63503312, -1.29715087],
[-1.17022425,  0.51683808],
[ 0.78682161,  0.01398688],
[-0.52432637, -0.41991571],
[-0.76412541, -1.54192833],
[ 1.84080367, -0.88281496]])
```

```
y
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

```
plt.plot(X[:, 0], X[:, 1], 'ro' )
```

```
[<matplotlib.lines.Line2D at 0x7febd42d94d0>]
```


```
import seaborn as sns
```

```
sns.scatterplot(X[:, 0], X[:, 1], hue = y)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarni
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7febe276ae50>
```