Module 6: Machine Learning Using Python – I

Assignment Solution

1. Consider a random 10 x 2 matrix representing Cartesian coordinates, convert them to Polar coordinates.

Solution

```
import numpy as np

Z = np.random.random((10,2))

X,Y = Z[:,0], Z[:,1]

R = np.sqrt(X**2+Y**2)

T = np.arctan2(Y,X)
```

2. Create random vector of size 50 and replace the maximum value by 0 and minimum value by 100.

Solution


```
import numpy as np

Z = np.random.random(50)

Z[Z.argmax()] = 0

Z[Z.argmin()] = 100

Expected output is : 100 should appear in Z
```

3. Create below matrix using scipy.

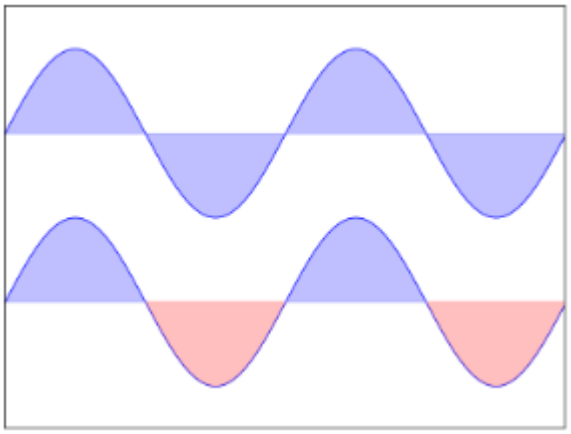
```
[[ 2.  0.  1.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  2.  0.  1.  0.  0.  0.  0.  0.  0.]
 [ 1.  0.  2.  0.  1.  0.  0.  0.  0.  0.]
 [ 0.  1.  0.  2.  0.  1.  0.  0.  0.  0.]
 [ 0.  0.  1.  0.  2.  0.  1.  0.  0.  0.]
 [ 0.  0.  0.  1.  0.  2.  0.  1.  0.  0.]
 [ 0.  0.  0.  0.  1.  0.  2.  0.  1.  0.]
 [ 0.  0.  0.  0.  0.  1.  0.  2.  0.  1.]
 [ 0.  0.  0.  0.  0.  0.  1.  0.  2.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  1.  0.  2.]]
```

Solution

```
Import numpy as np
Import scipy
N = 10
diag = np.zeros(N) + 2
udiag = np.zeros(N) + 1
ldiag = np.zeros(N) + 1
mat = scipy.sparse.dia_matrix(([diag, udiag, ldiag], [0, 2, -2]), shape=(N, N))
print mat.todense()
```

4. Reproduce given plot by correcting the below code.

```
from pylab import *
n = 256
X = np.linspace(-np.pi,np.pi,n,endpoint=True)
Y = np.sin(2*X)
plot (X, Y+1, color='blue', alpha=1.00)
plot (X, Y-1, color='blue', alpha=1.00)
show()
```



Solution

```
from pylab import *
n = 256
X = np.linspace(-np.pi,np.pi,n,endpoint=True)
Y = np.sin(2*X)
axes([0.025,0.025,0.95,0.95])
```

```
plot (X, Y+1, color='blue', alpha=1.00)
fill_between(X, 1, Y+1, color='blue', alpha=.25)
plot (X, Y-1, color='blue', alpha=1.00)
fill_between(X, -1, Y-1, (Y-1) > -1, color='blue', alpha=.25)
fill_between(X, -1, Y-1, (Y-1) < -1, color='red',  alpha=.25)
xlim(-np.pi,np.pi), xticks([])
ylim(-2.5,2.5), yticks([])
show()
```