Name : Shubham Dutta

Year : 2nd

Stream : CST

Section : 2B

Roll : 58

Enrollment : 12019009022112

Paper Name : AI ML Advanced

Paper Code : PCC-CS495

Date : 20/05/21

Time : 11:49 AM

Signature : Shubham Dutta

Answer

1 Ans-

i) from sklearn.datasets import make_blobs
td = make_blob(n_samples = 400, centers = 3,
n_features = 3, cluster_std = 1.5,
random_state = 50)

ii) from sklearn.cluster import KMeans
km = KMeans(n_clusters = 3)
datapoint = td[0]

iii) y_pred = km.fit_predict(datapoint)
print(y_pred)

iv) import matplotlib.pyplot as plt
plt.scatter(datapoint[:, 0], datapoint[:, -1])

v) clusters = km.cluster_centers_
print(clusters)
scatter(datapoint[y_pred == 0,0], datapoint[y_pred==0,
s=80, color='blue')
scatter(datapoint[y_pred==1, 0], datapoint[y_pred==1,1],
s=80, color='red')
scatter(datapoint[y_pred = 2, 0], datapoint[y_pred=2,

82 Ans:

i) from sklearn.datasets import load_digits
digits = load_digits()

ii) from sklearn.model_selection import train_test_split

x_train1, x_test1, y_train1, y_test1 =
train_test_split(digits.data, digits.target, test_size = 0.3)

iii) import numpy as np
print np.arrange (len(x_train_1)):

iv) y_train_nonLabel = np.copy(y_train_1)
y_train_nonLabel = [np.arrange(len(x_train1)
[280:]] = -1
print (y_train_nonLabel):

Q3 Ans :-

i) from sklearn.datasets import load_digits
digits = load_digits()
print (digits.DESCR)

ii) digits.data.shape

iii) digits.target.shape

iv) print (digits.data[204])
image = digits.data[204].

v) import numpy as np
np.reshape (image, (8,8))

vi) import matplotlib.pyplot as plt
plt.imshow (np.reshape (image, (8,8)),
cmap = 'gray').

vii) from sklearn.model_selection import train_test_split
x_train1, x_test1, y_train1, y_test1 =
train_test_split (digits.data, digits.target,
test_size = 0.20)

84. i).

```
from sklearn.datasets import load_boston
boston = load_boston()
print(boston.DESCR)
```

ii) `import pandas as pd`

```
ds = pd.DataFrame(boston.data, columns=
                  boston.feature_names)
```

iii) `ds['MEDV'] = boston.target`

iv) `pd.DataFrame(ds.corr().round(3))`

v)

```
x = ds['RM']
y = ds['CRIM']
pd.DataFrame([x, y])
```

vi)

```
from sklearn.model_selection import
        train_test_split
x = pd.DataFrame(x)
y = pd.DataFrame(y)
x_train1, x_test1, y_train1, y_test1 =
        train_test_split(x, y, test_size = 0.25)
```

vii) from sklearn.linear_model import LinearRegression

```
train_model = LinearRegression()
train_model.fit(x_train1, y_train1)
x_pred = train_model.predict(x_test1)
y_pred = train_model.predict(y_test1)
```

viii) import numpy as np

```
from sklearn.metrics import mean_squared_error
np.sqrt(mean_squared_error(y_test1, y_pred))
```

ix)

```
# i)
from sklearn.datasets import make_blobs
td = make_blobs(n_samples = 400, centers = 3, n_features = 3, cluster_std =
print(td)
```

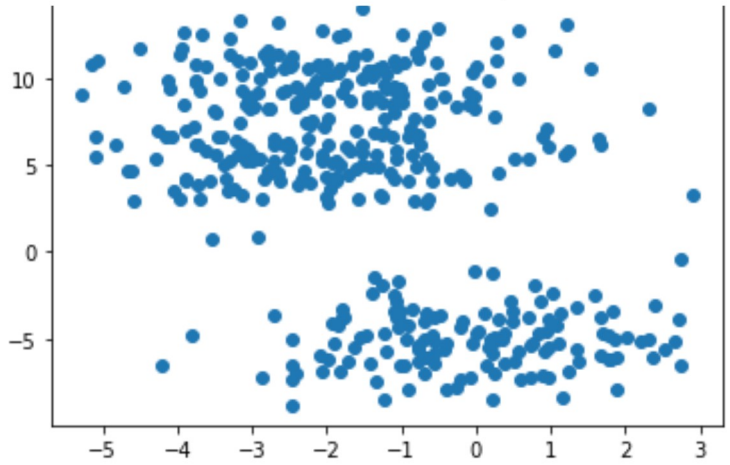
```
(array([[ 1.36954026, -4.32992761, -6.28741522],
       [-2.00051044,  2.86920882,  4.2482137 ],
       [-5.07595285, -5.71753679, 10.94815629],
       ...,
       [-1.08431735,  5.28369361,  8.01256676],
       [-0.91811619, -2.41200928,  6.65091126],
       [ 0.88024083, -6.07164711, -7.14154329]]), array([0, 2, 1, 2, 1, 2, 1,
       1, 0, 2, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 0, 2, 2, 2, 2, 1, 1, 1, 0,
       2, 0, 0, 2, 1, 2, 0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 1, 2, 2, 1, 0,
       2, 0, 2, 2, 1, 1, 1, 0, 0, 1, 1, 0, 0, 2, 2, 0, 2, 1, 0, 2, 1, 2,
       1, 0, 1, 1, 0, 1, 0, 2, 0, 1, 0, 0, 1, 2, 0, 2, 0, 1, 2, 2, 0, 0,
       2, 1, 2, 1, 1, 0, 1, 0, 2, 0, 0, 2, 0, 1, 0, 1, 2, 2, 0, 2, 0, 2,
       1, 1, 1, 0, 0, 1, 0, 2, 0, 2, 1, 0, 0, 1, 0, 2, 0, 1, 1, 1, 0, 0,
       2, 0, 1, 1, 2, 1, 1, 1, 1, 0, 2, 1, 1, 1, 1, 0, 0, 0, 2, 1, 2, 2,
       0, 1, 2, 1, 1, 1, 2, 0, 0, 1, 0, 1, 2, 1, 2, 2, 0, 2, 2, 2, 1, 2,
       2, 2, 2, 1, 1, 0, 1, 2, 2, 0, 2, 1, 2, 0, 0, 0, 1, 2, 2, 2, 0, 2,
       1, 0, 0, 1, 2, 1, 2, 0, 2, 0, 0, 1, 2, 1, 2, 1, 2, 1, 2, 2, 0, 2,
       1, 2, 0, 0, 2, 0, 1, 0, 1, 1, 1, 0, 1, 1, 2, 1, 2, 0, 1, 2, 1, 2,
       0, 1, 0, 2, 1, 1, 2, 2, 2, 0, 1, 2, 0, 1, 2, 1, 1, 2, 2, 2, 1, 2,
       1, 0, 1, 0, 0, 2, 0, 0, 1, 2, 2, 2, 1, 0, 0, 1, 2, 0, 1, 0, 2, 0,
       1, 0, 1, 1, 1, 2, 2, 2, 2, 0, 0, 0, 0, 1, 0, 1, 2, 1, 0, 1, 1, 1,
       1, 1, 0, 1, 1, 0, 0, 0, 2, 0, 2, 2, 0, 0, 0, 2, 0, 2, 2, 0, 2, 1,
       0, 1, 0, 0, 0, 1, 2, 1, 0, 0, 1, 2, 2, 0, 1, 2, 2, 0, 1, 0, 2, 0,
       2, 1, 2, 0, 0, 2, 2, 2, 1, 1, 2, 0, 0, 1, 1, 2, 0, 0, 2, 0, 2, 0,
       2, 2, 1, 0]))
```

```
# ii)
from sklearn.cluster import KMeans
km = KMeans(n_clusters=3)
datapoint = td[0]
```

```
# iii)
y_pred = km.fit_predict(datapoint)
print(y_pred)
```

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

✓ 1s completed at 11:59 AM

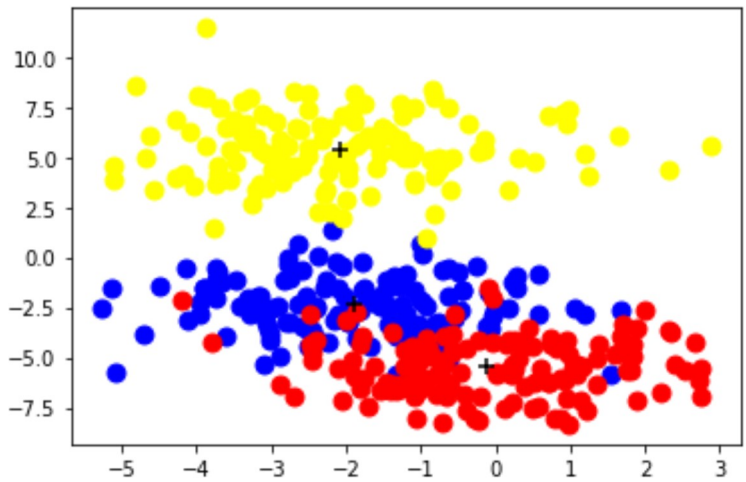


```
# v)
clusters = km.cluster_centers_
print(clusters)
```

```
[[-1.89405853 -2.38178988 10.05154914]
 [-0.10838418 -5.45628981 -5.08085495]
 [-2.07224603  5.41338941  5.22588591]]
```

```
plt.scatter(datapoint[y_pred==0,0], datapoint[y_pred==0,1], s=80, color='blu
plt.scatter(datapoint[y_pred==1,0], datapoint[y_pred==1,1], s=80, color='red
plt.scatter(datapoint[y_pred==2,0], datapoint[y_pred==2,1], s=80, color='yel
plt.scatter(clusters[0][0], clusters[0][1], marker="+", s=80, color="black")
plt.scatter(clusters[1][0], clusters[1][1], marker="+", s=80, color="black")
plt.scatter(clusters[2][0], clusters[2][1], marker="+", s=80, color="black")
```

```
<matplotlib.collections.PathCollection at 0x7f0e4047fd90>
```



```
#i
from sklearn.datasets import load_digits
digits = load_digits()


#ii
from sklearn.model_selection import train_test_split
x_train_1,x_test_1,y_train_1,y_test_1=train_test_split(digits.data,digits.ta


#iii
import numpy as np
print(np.arange(len(x_train_1)))
```

```
[   0    1    2 ... 1254 1255 1256]
```

```
#iv
y_train_nonlabel=np.copy(y_train_1)
y_train_nonlabel[np.arange(len(x_train_1))[280:]]=-1
print(y_train_nonlabel)
```

```
[ 4  1  7 ... -1 -1 -1]
```

✓ 0s completed at 11:12 AM

```
#i
from sklearn.datasets import load_digits
digits = load_digits()
print(digits.DESCR)
```

```
#ii
digits.data.shape
```

```
(1797, 64)
```

```
#ii
digits.target.shape
```

```
(1797,)
```

```
#iii
print(digits.data[204])
image = digits.data[204]
```

```
[ 0.  4. 16. 16. 16. 16.  5.  0.  0. 11. 16.  8.  5.  8.  3.  0.  0. 10.
 16.  2.  0.  0.  0.  0.  0.  3. 16.  6.  0.  0.  0.  0.  0.  0. 16.  9.
  0.  0.  0.  0.  0.  0. 12. 16.  2.  0.  0.  0.  0.  0.  6. 16. 11.  0.
  0.  0.  0.  4. 16. 12.  1.  0.  0.  0.]
```

```
#iv
import numpy as np
np.reshape(image,(8,8))
```

```
array([[ 0.,  4., 16., 16., 16., 16.,  5.,  0.],
       [ 0., 11., 16.,  8.,  5.,  8.,  3.,  0.],
       [ 0., 10., 16.,  2.,  0.,  0.,  0.,  0.],
       [ 0.,  3., 16.,  6.,  0.,  0.,  0.,  0.],
       [ 0.,  0., 16.,  9.,  0.,  0.,  0.,  0.],
       [ 0.,  0., 12., 16.,  2.,  0.,  0.,  0.],
       [ 0.,  0.,  6., 16., 11.,  0.,  0.,  0.],
       [ 0.,  4., 16., 12.,  1.,  0.,  0.,  0.]])
```

```
#v
import matplotlib.pyplot as plt
plt.imshow(np.reshape(image,(8,8)), cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7f6477404610>
```

✓ 0s completed at 10:55 AM

```
ion import train_test_split
n_1,y_test_1=train_test_split(digits.data,digits.target,test_size=0.20)
```

Not done.

vii. Using DecisionTreeClassifier from sklearn.tree with criterion as entropy and max_depth=20, train the model.
viii. Determine the R2 score
ix. Using RandomForestClassifier from sklearn.ensemble, train the model and find the R2 score

```
#i
from sklearn.datasets import load_digits
digits = load_digits()
print(digits.DESCR)
```

```
#ii
digits.data.shape
```

```
(1797, 64)
```

```
#ii
digits.target.shape
```

```
(1797,)
```

```
#iii
print(digits.data[204])
image = digits.data[204]
```

```
[ 0.  4. 16. 16. 16. 16.  5.  0.  0. 11. 16.  8.  5.  8.  3.  0.  0. 10.
 16.  2.  0.  0.  0.  0.  0.  3. 16.  6.  0.  0.  0.  0.  0.  0. 16.  9.
  0.  0.  0.  0.  0.  0. 12. 16.  2.  0.  0.  0.  0.  0.  6. 16. 11.  0.
  0.  0.  0.  4. 16. 12.  1.  0.  0.  0.]
```

```
#iv
import numpy as np
np.reshape(image,(8,8))
```

```
array([[ 0.,  4., 16., 16., 16., 16.,  5.,  0.],
       [ 0., 11., 16.,  8.,  5.,  8.,  3.,  0.],
       [ 0., 10., 16.,  2.,  0.,  0.,  0.,  0.],
       [ 0.,  3., 16.,  6.,  0.,  0.,  0.,  0.],
       [ 0.,  0., 16.,  9.,  0.,  0.,  0.,  0.],
       [ 0.,  0., 12., 16.,  2.,  0.,  0.,  0.],
       [ 0.,  0.,  6., 16., 11.,  0.,  0.,  0.],
       [ 0.,  4., 16., 12.,  1.,  0.,  0.,  0.]])
```

```
#v
import matplotlib.pyplot as plt
plt.imshow(np.reshape(image,(8,8)), cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7f6477404610>
```

✓ 0s completed at 10:55 AM

```
ion import train_test_split
n_1,y_test_1=train_test_split(digits.data,digits.target,test_size=0.20)
```

Not done.

vii. Using DecisionTreeClassifier from sklearn.tree with criterion as entropy and max_depth=20, train the model.
viii. Determine the R2 score
ix. Using RandomForestClassifier from sklearn.ensemble, train the model and find the R2 score