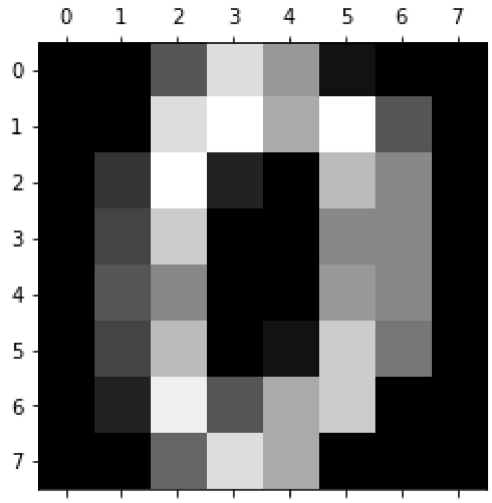
In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_digits
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
```

In [2]:
```
# 1.We shall use the same dataset used in previous assignment -digits.
# Make a 80-20 train/test split.
digits = load_digits()
plt.gray()
plt.matshow(digits.images[0])
```

Out[2]:
```
<matplotlib.image.AxesImage at 0x1d4d4217048>

<Figure size 432x288 with 0 Axes>
```



In [5]:
```
images = digits.images.reshape(digits.images.shape[0], -1)
labels = digits.target

x_train, x_test, y_train, y_test = train_test_split(images, labels, test_size=
0.20, random_state=10)
```

In [10]:
```
# 2.Using scikit learn perform a LDA on the dataset. Find out the number of co
mponents
# in the projected subspace.
lda_model = LDA(n_components = 5)           # T0 Get 95% of Accuracy n_compone
nts should be 5
lda_model.fit(x_train, y_train)

print(lda_model.explained_variance_ratio_)
```

```
[0.34582698 0.2204826  0.19641704 0.13912627 0.09814711]
```

In [12]:
```
# 3.Transform the dataset and fit a logistic regression and observe the accura
cy.
# Compare it with the previous model based on PCA in terms of accuracy and mod
el complexity.
x_train = lda_model.transform(x_train)
x_test = lda_model.transform(x_test)

log_model = LogisticRegression()
log_model.fit(x_train, y_train)

predicted_values = log_model.predict(x_test)

print("Accuracy Score\n")
print(metrics.accuracy_score(predicted_values, y_test)*100)
```

```
Accuracy Score

90.55555555555556

C:\Users\hp\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:94
0: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

In [ ]: