In [8]:
```
import pandas as pd
import numpy as np
from math import sqrt
import time

from sklearn.model_selection import train_test_split
from sklearn.datasets import load_digits
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import accuracy_score, recall_score, precision_score
from sklearn.utils import shuffle
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression



import matplotlib.pyplot as plt
#https://stackoverflow.com/questions/21154643/python-line-profiler-installat
%load_ext line_profiler
```

```
The line_profiler extension is already loaded. To reload it, use:
  %reload_ext line_profiler
```

In [2]:
```
digits = load_digits()
print(digits.data.shape)


#image representatio of the data
# plt.gray()
# plt.matshow(digits.images[0])
# plt.show()


X = digits.data
y = digits.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, ra
```

```
(1797, 64)
```

KNN

In [3]:
```
len(X_train),len(X_test)
```

Out[3]:
```
(1347, 450)
```

In [5]:
```
k = 3
start = time.time()
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)
pred = model.predict(X_test)
acc = accuracy_score(y_test, pred)
end = time.time()
print(f"Time Taken: {end-start} sec")
print("K = "+str(k)+"; Accuracy: "+str(acc))
```

```
Time Taken: 0.22458481788635254 sec
K = 3; Accuracy: 0.9777777777777777
```

Logistic regression

In [11]:
```
start = time.time()
clf = LogisticRegression(random_state=0,max_iter=5000).fit(X_train, y_train)
pred = model.predict(X_test)
acc = accuracy_score(y_test, pred)
end = time.time()
print(f"Time Taken: {end-start} sec")
print("Accuracy: "+str(acc))
```

```
Time Taken: 2.324965000152588 sec
Accuracy: 0.9777777777777777
```

In [ ]:
```
1
```