import matplotlib.pyplot as plt

%matplotlib inline

from sklearn.datasets import load\_digits

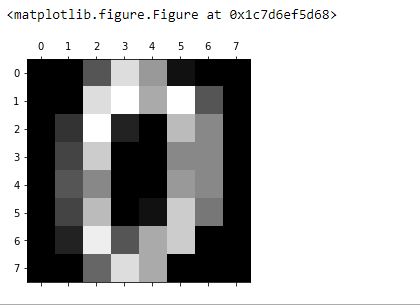
digits = load\_digits()

import pylab as pl

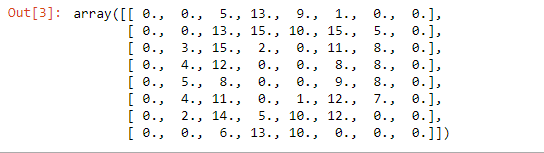
pl.gray()

pl.matshow(digits.images[0])

pl.show()



digits.images[0]



images\_and\_labels = list(zip(digits.images,digits.target))

plt.figure(figsize=(5,5))

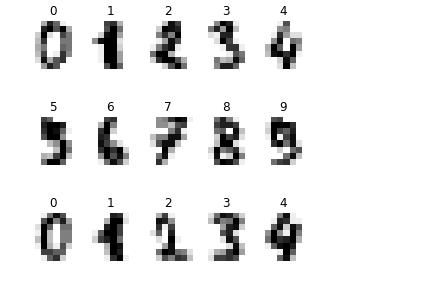
for index, (image, label) in enumerate(images\_and\_labels[:15]):

plt.subplot(3, 5, index + 1)

plt.axis('off')

plt.imshow(image, cmap=plt.cm.gray\_r, interpolation='nearest')

plt.title('%i' % label)



import random

from sklearn import ensemble

#Define variables

n\_samples = len(digits.images)

x = digits.images.reshape((n\_samples, -1))

y = digits.target

#Create random indices

sample\_index=random.sample(range(len(x)),len(x)/5) #20-80

valid\_index=[i for i in range(len(x)) if i not in sample\_index]

#Sample and validation images

sample\_images=[x[i] for i in sample\_index]

valid\_images=[x[i] for i in valid\_index]

#Sample and validation targets

sample\_target=[y[i] for i in sample\_index]

valid\_target=[y[i] for i in valid\_index]

#Using the Random Forest Classifier

classifier = ensemble.RandomForestClassifier()

#Fit model with sample data

classifier.fit(sample\_images, sample\_target)

#Attempt to predict validation data

score=classifier.score(valid\_images, valid\_target)

print ('Random Tree Classifier:\n')

print 'Score\t'+str(score)

i=220

pl.gray()

pl.matshow(digits.images[i])

pl.show()

classifier.predict(x[i])

