```
import os
```

```
cwd = os.getcwd()
```

```
import numpy as np
```

```
dataset = np.loadtxt('/pima-new.csv', delimiter=',')
```

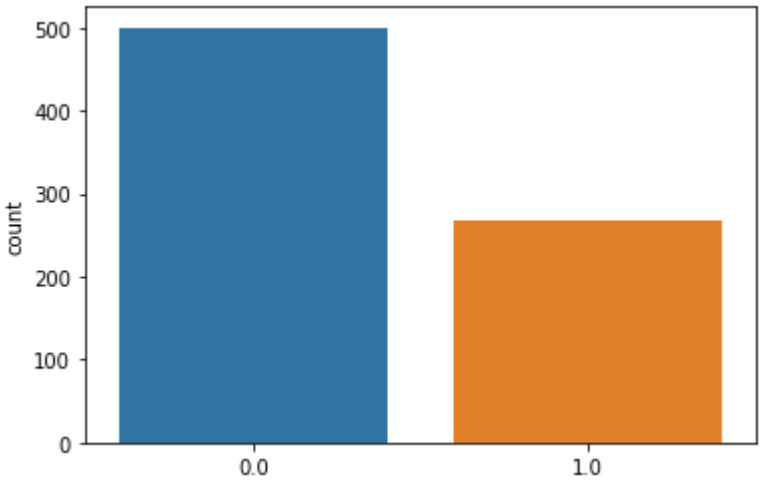
```
dataset.shape
```

```
(768, 9)
```

```
x = dataset[:,0:8]
y = dataset[:,8]
```

```
import seaborn as sns
sns.countplot(x=y)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3b806f0710>
```




```
from collections import Counter
Counter(y)
```

```
Counter({1.0: 268, 0.0: 500})
```

```
#cross validation
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(
    x, y, random_state=0, test_size=0.25)
```

```
x_test.shape
```

```
(192, 8)
```

```
x_test.shape
```

```
(192, 8)
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.datasets import mnist
```

```
(x_train, y_train),(x_test, y_test) = mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist
11493376/11490434 [==============================] - 0s 0us/step
11501568/11490434 [==============================] - 0s 0us/step
```

```
x_train.shape
```

```
(60000, 28, 28)
```

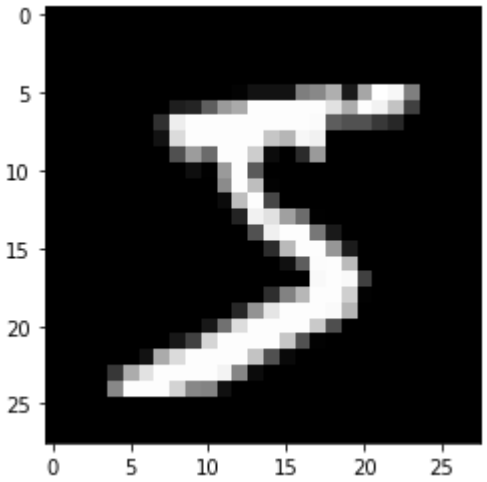
```
x_test.shape
```

```
(10000, 28, 28)
```

```
import matplotlib.pyplot as plt
```

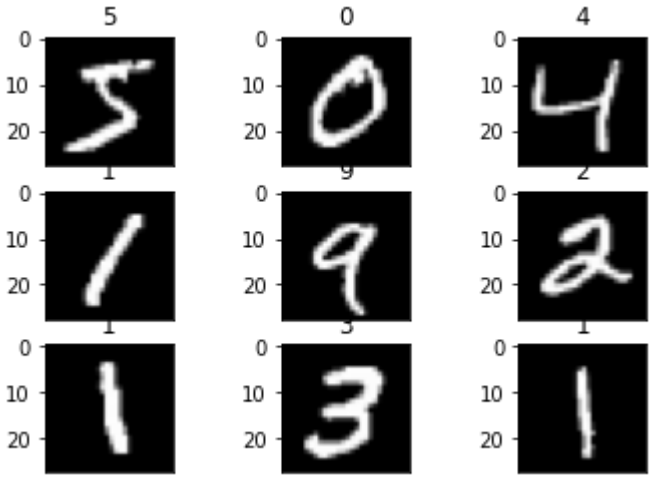
```
plt.imshow(x_train[0], cmap = 'gray')
```

```
<matplotlib.image.AxesImage at 0x7f3b7f8bc550>
```

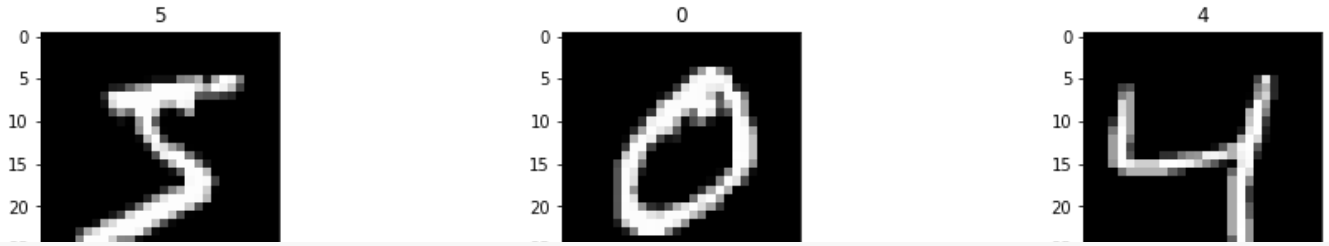


```
import matplotlib.pyplot as plt

for i in range(9):
  plt.subplot(3,3,i+1)
  plt.imshow(x_train[i], cmap = 'gray')
  plt.title(y_train[i])
  plt.xticks([])
```




```
plt.figure(figsize=(16,9))
for i in range(9):
  plt.subplot(3,3,i+1)
  plt.imshow(x_train[i], cmap = 'gray')
  plt.title(y_train[i])
  plt.xticks([])
```




```
y_train[0]
```

```
5
```

```
num_pixels = x_train.shape[1] * x_train.shape[2]
```

```
num_pixels
```

```
784
```

```
x_train = x_train.reshape(-1, num_pixels)
```

```
x_train.shape
```

```
(60000, 784)
```

```
x_test = x_test.reshape(-1, num_pixels)
```

```
x_test.shape
```

```
(10000, 784)
```

```
model = Sequential()

#input layer
model.add(Dense(num_pixels, input_dim = num_pixels, activation = 'relu'))

#hidden layer
model.add(Dense(500, activation = 'relu'))

#output layer
model.add(Dense(10, activation = 'softmax'))
```

```
from tensorflow.keras.optimizers import SGD
```

```
#compile the model
model.compile(optimizer=SGD(), metrics=['accuracy'], loss='categorical_corssentropy')
```

```
from tensorflow.keras.utils import to_categorical
```

```
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

```
y_train
```

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 1., 0.]], dtype=float32)
```