Understanding the dataset

December 14, 2020

[2]:

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import linear_model
import seaborn as sns
from sklearn.model_selection import train_test_split
import gc

# Printing basic information about the contents of the train.csv file downloaded from kaggle.com
train_digits = pd.read_csv("train.csv")
train_digits.info()
```

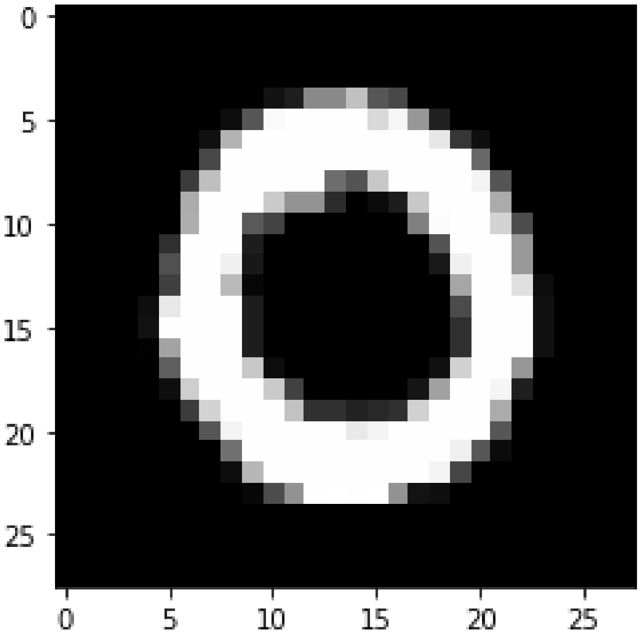
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42000 entries, 0 to 41999
Columns: 785 entries, label to pixel783
dtypes: int64(785)
memory usage: 251.5 MB
```

[9]:

```python
digit_zero = train_digits.iloc[1, 1:]
digit_zero = digit_zero.values.reshape(28, 28)
plt.imshow(digit_zero, cmap='gray')
```

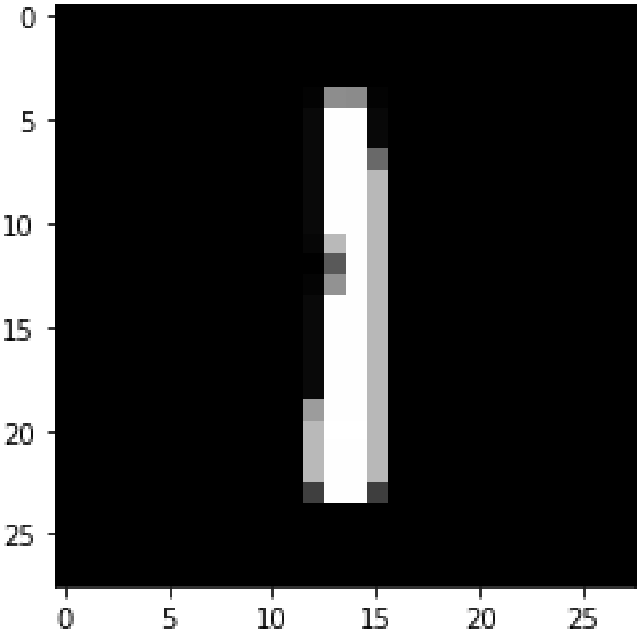
[9]:

```
<matplotlib.image.AxesImage at 0x7ffab0f38a10>
```



[8]:
```
digit_uno = train_digits.iloc[2, 1:]
digit_uno = digit_uno.values.reshape(28, 28)
plt.imshow(digit_uno, cmap='gray')
```

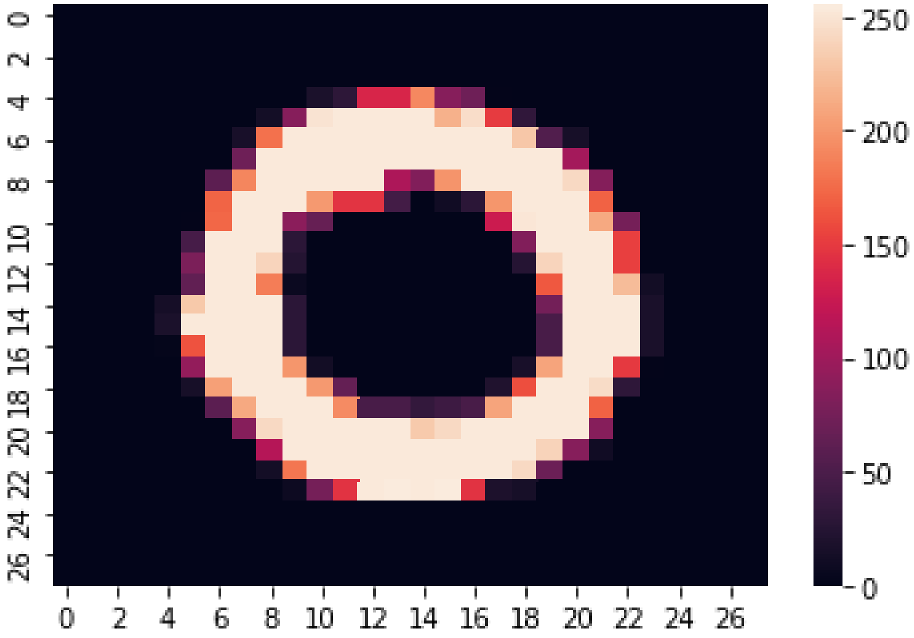
[8]:
```
<matplotlib.image.AxesImage at 0x7ffab0f56510>
```



[5]:
```
##The training data set comprises of pixel intensity information for the␣
 ↪handwritten number mentioned as the label.
# visualising the array as pixel intensity information
print(digit_zero[5:-5, 5:-5])
sns.heatmap(digit_zero)
```

```
[[  0   0   0  13  86 250 254 254 254 254 217 246 151  32   0   0   0   0]
 [  0   0  16 179 254 254 254 254 254 254 254 254 254 231  54  15   0   0]
 [  0   0  72 254 254 254 254 254 254 254 254 254 254 254 254 104   0   0]
 [  0  61 191 254 254 254 254 254 109  83 199 254 254 254 254 243  85   0]
 [  0 172 254 254 254 202 147 147  45   0  11  29 200 254 254 254 171   0]
 [  1 174 254 254  89  67   0   0   0   0   0   0 128 252 254 254 212  76]
 [ 47 254 254 254  29   0   0   0   0   0   0   0   0  83 254 254 254 153]
 [ 80 254 254 240  24   0   0   0   0   0   0   0   0  25 240 254 254 153]
 [ 64 254 254 186   7   0   0   0   0   0   0   0   0   0 166 254 254 224]
 [232 254 254 254  29   0   0   0   0   0   0   0   0   0  75 254 254 254]
 [254 254 254 254  29   0   0   0   0   0   0   0   0   0  48 254 254 254]
 [163 254 254 254  29   0   0   0   0   0   0   0   0   0  48 254 254 254]
 [ 94 254 254 254 200  12   0   0   0   0   0   0   0  16 209 254 254 150]
 [ 15 206 254 254 254 202  66   0   0   0   0   0  21 161 254 254 245  31]
 [  0  60 212 254 254 254 194  48  48  34  41  48 209 254 254 254 171   0]
 [  0   0  86 243 254 254 254 254 254 233 243 254 254 254 254 254  86   0]
 [  0   0   0 114 254 254 254 254 254 254 254 254 254 254 239  86  11   0]
 [  0   0   0  13 182 254 254 254 254 254 254 254 254 243  70   0   0   0]]
```

[5]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ffac405bdd0>
```



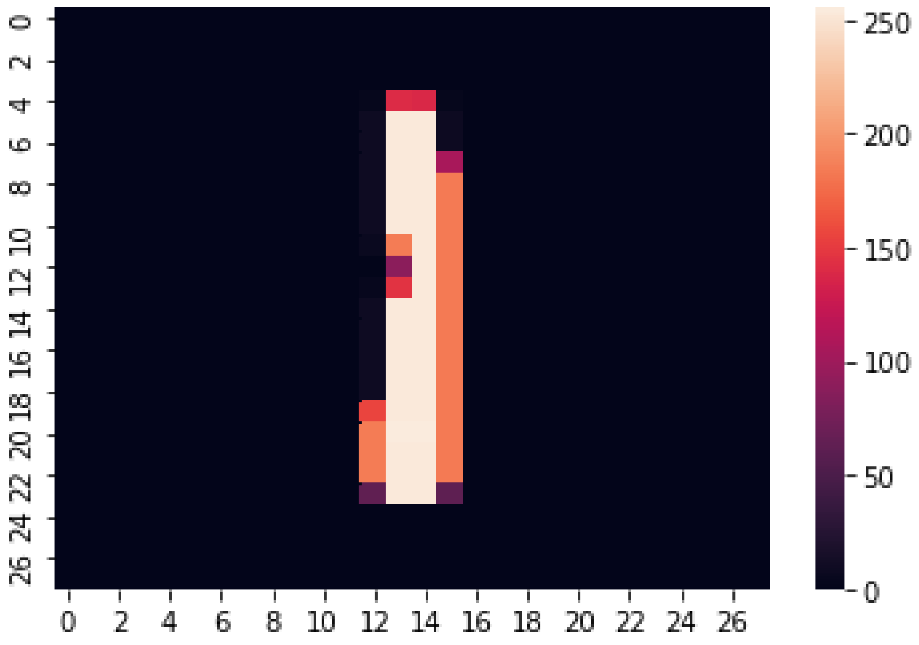
[6]:

```
# visualising the array as pixel intensity information
print(digit_uno[5:-5, 5:-5])
sns.heatmap(digit_uno)
```

```
[[  0   0   0   0   0   0   0   9 254 254   8   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   9 254 254   8   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   9 254 254 106   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   9 254 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   9 254 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   9 254 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   6 185 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0  89 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   4 146 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   9 254 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   9 254 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   9 254 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   9 254 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   9 254 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0 156 254 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0 185 255 255 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0 185 254 254 184   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0 185 254 254 184   0   0   0   0   0   0   0]]
```

[6]:

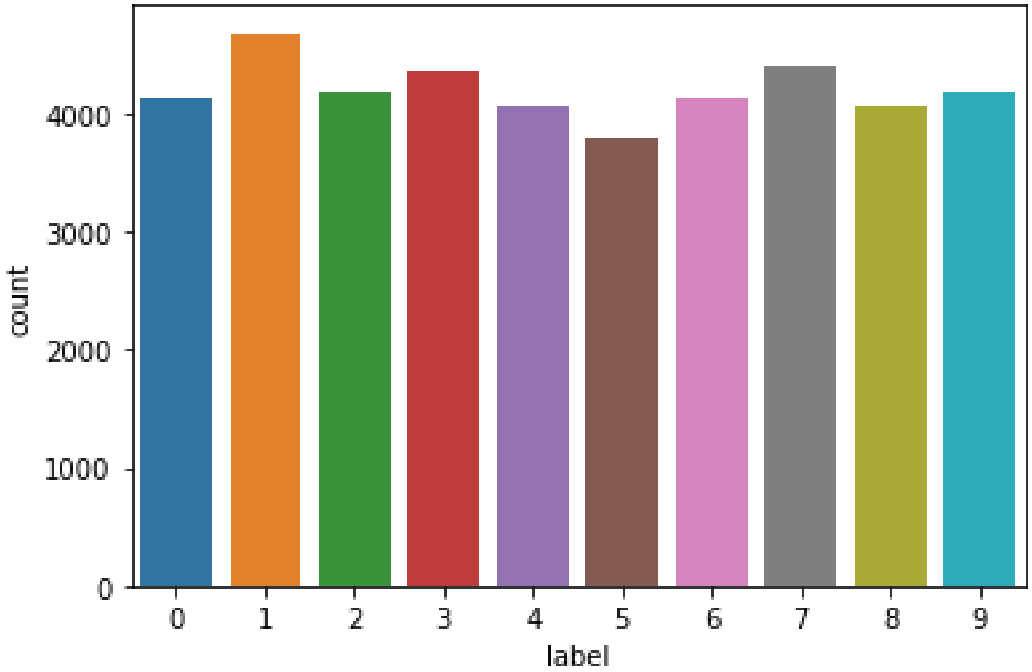
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ffac1493590>
```



[37]:

```
print("The label distribution for the 10 classes in the training data set is as␣
 ↪follows")
sns.countplot(train_digits['label'])
print(train_digits.label.value_counts())
```

```
The label distribution for the 10 classes in the training data set is as follows
1    4684
7    4401
3    4351
9    4188
2    4177
6    4137
0    4132
4    4072
8    4063
5    3795
Name: label, dtype: int64
```



We see that selection is little biased towards digit 1 and the sample count for label 1 is around 30% higher than sample 5, and this problem persists. On average, the dataset is balanced. This is an important factor in considering the choices of models to be used, especially SVM, since SVMs rarely perform well on imbalanced data.