In [1]:

```python
import numpy as np
from sklearn import svm
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
```

In [3]:

```python
data_train = pd.read_csv("data_train.csv")
data_test = pd.read_csv("data_test.csv")

X = data_train[['intensity','symmetry']]
```

In [24]:

```python
y0 = np.where(data_train["digit"] ==0, 1 ,-1)
Gamma = [-2,-1,0,1,2]
```

In [29]:

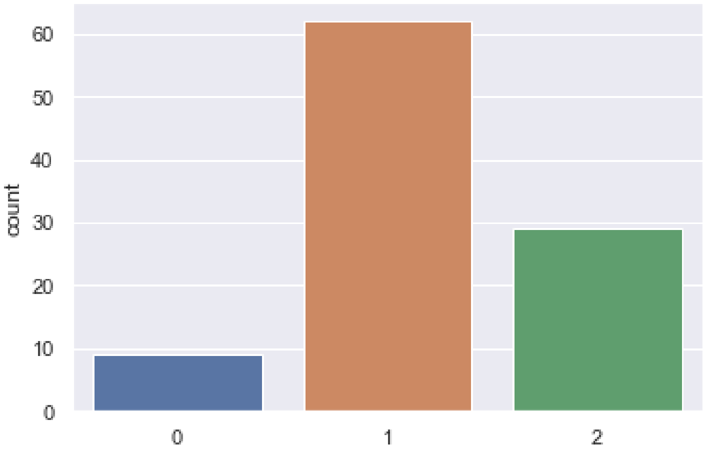
```python
BestGamma = []
for i in range(100):
    X_train, X_test, y_train, y_test = train_test_split(X, y0, test_size=1000)
    best_gamma = -10
    Prec = 0
    for gamma in Gamma:
        result = svm.SVC(C = 0.1, kernel = "rbf", gamma = 10**gamma).fit(X_train,y_t
        Eval = sum(result.predict(X_test) == y_test)
        if Eval > Prec:
            Prec = Eval
            best_gamma = gamma
    BestGamma = BestGamma + [best_gamma]
```

In [43]:

```python
import seaborn as sns; sns.set()
sns.countplot(BestGamma)
```

Out[43]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d4ceaa84a8>
```



Q16: only the gamma of 1,10,100 could be the candidate for best gamma. This may result from smaller gamma could eliminate the effect of higher polynomials.