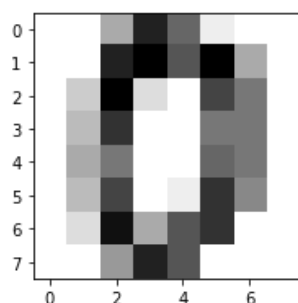


Задание 2

```
In [1]: from sklearn import datasets
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import BernoulliNB
from sklearn.model_selection import cross_val_score
import matplotlib.pyplot as plt
import pandas
%matplotlib inline
```

Посмотрим на датасеты

```
In [2]: # Первый датасет содержит в себе 1797 изображения размера 8*8,
# где каждое изображение задано вектором размера 64
# Каждый элемент вектора задает оттенок клеточки, как на рисунке ниже
# Это и есть признаки
digits = datasets.load_digits()
plt.figure(1, figsize=(3, 3))
plt.imshow(digits.images[0], cmap=plt.cm.gray_r, interpolation='nearest')
plt.show()
```



```
In [3]: print(digits.data[:3])
```

```
[[ 0.  0.  5. 13.  9.  1.  0.  0.  0.  0. 13. 15. 10. 15.
   5.  0.  0.  3. 15.  2.  0. 11.  8.  0.  0.  4. 12.  0.
   0.  8.  8.  0.  0.  5.  8.  0.  0.  9.  8.  0.  0.  4.
  11.  0.  1. 12.  7.  0.  0.  2. 14.  5. 10. 12.  0.  0.
   0.  0.  6. 13. 10.  0.  0.  0.]
 [ 0.  0.  0. 12. 13.  5.  0.  0.  0.  0.  0. 11. 16.  9.
   0.  0.  0.  0.  3. 15. 16.  6.  0.  0.  0.  7. 15. 16.
  16.  2.  0.  0.  0.  0.  1. 16. 16.  3.  0.  0.  0.  0.
   1. 16. 16.  6.  0.  0.  0.  0.  1. 16. 16.  6.  0.  0.
   0.  0.  0. 11. 16. 10.  0.  0.]
 [ 0.  0.  0.  4. 15. 12.  0.  0.  0.  0.  3. 16. 15. 14.
   0.  0.  0.  0.  8. 13.  8. 16.  0.  0.  0.  0.  1.  6.
  15. 11.  0.  0.  0.  1.  8. 13. 15.  1.  0.  0.  0.  9.
  16. 16.  5.  0.  0.  0.  0.  3. 13. 16. 16. 11.  5.  0.
   0.  0.  0.  3. 11. 16.  9.  0.]]
```

```
In [4]: # Классификацией является разделение опухоли на доброкачественную и зло
качественную
breast_cancer = datasets.load_breast_cancer()
print(breast_cancer.feature_names)
print(breast_cancer.target_names)

['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
['malignant' 'benign']
```

```
In [5]: print(breast_cancer.data[:3])

[[ 1.79900000e+01  1.03800000e+01  1.22800000e+02  1.00100000e+03
   1.18400000e-01  2.77600000e-01  3.00100000e-01  1.47100000e-01
   2.41900000e-01  7.87100000e-02  1.09500000e+00  9.05300000e-01
   8.58900000e+00  1.53400000e+02  6.39900000e-03  4.90400000e-02
   5.37300000e-02  1.58700000e-02  3.00300000e-02  6.19300000e-03
   2.53800000e+01  1.73300000e+01  1.84600000e+02  2.01900000e+03
   1.62200000e-01  6.65600000e-01  7.11900000e-01  2.65400000e-01
   4.60100000e-01  1.18900000e-01]
 [ 2.05700000e+01  1.77700000e+01  1.32900000e+02  1.32600000e+03
   8.47400000e-02  7.86400000e-02  8.69000000e-02  7.01700000e-02
   1.81200000e-01  5.66700000e-02  5.43500000e-01  7.33900000e-01
   3.39800000e+00  7.40800000e+01  5.22500000e-03  1.30800000e-02
   1.86000000e-02  1.34000000e-02  1.38900000e-02  3.53200000e-03
   2.49900000e+01  2.34100000e+01  1.58800000e+02  1.95600000e+03
   1.23800000e-01  1.86600000e-01  2.41600000e-01  1.86000000e-01
   2.75000000e-01  8.90200000e-02]
 [ 1.96900000e+01  2.12500000e+01  1.30000000e+02  1.20300000e+03
   1.09600000e-01  1.59900000e-01  1.97400000e-01  1.27900000e-01
   2.06900000e-01  5.99900000e-02  7.45600000e-01  7.86900000e-01
   4.58500000e+00  9.40300000e+01  6.15000000e-03  4.00600000e-02
   3.83200000e-02  2.05800000e-02  2.25000000e-02  4.57100000e-03
   2.35700000e+01  2.55300000e+01  1.52500000e+02  1.70900000e+03
   1.44400000e-01  4.24500000e-01  4.50400000e-01  2.43000000e-01
   3.61300000e-01  8.75800000e-02]]
```

Выборка представлена действительными числами, которые могут быть меньше единицы

Сравним cross_val_score

```
In [6]: def compare_score(naive_bayes_estimator, dataset):
         estimator = naive_bayes_estimator
         score = cross_val_score(naive_bayes_estimator, dataset.data, dataset
         .target)

         return score.mean()
```

```
In [7]: naive_bayes_estimators = [GaussianNB(), MultinomialNB(), BernoulliNB()]

for i in range(3):
    print(str(naive_bayes_estimators[i]))
    print('digits: ' +
          "{:.4f}".format(compare_score(naive_bayes_estimators[i], digits)))
    print('breast_cancer: ' +
          "{:.4f}".format(compare_score(naive_bayes_estimators[i], breast_cancer)))
    print('\n')
```

GaussianNB(priors=None)
digits: 0.8186
breast_cancer: 0.9367

MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
digits: 0.8709
breast_cancer: 0.8946

BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)
digits: 0.8258
breast_cancer: 0.6274

Ответы на вопросы:

1. На **breast_cancer** максимальное значение получилось: 0.9367 (GaussianNB)
2. На **digits** максимальное значение получилось: 0.8709 (MultinomialNB)
3. (d) - верное утверждение