Задание 2

In [11]:
```
import numpy as np
from sklearn import datasets
from sklearn.model_selection import cross_val_score
from sklearn.naive_bayes import BernoulliNB, MultinomialNB, GaussianNB
import matplotlib.pyplot as plt

%matplotlib inline
```

Загрузим датасеты, выведем несколько строчек из обучающих выборок и названия признаков (где это возможно).

In [2]:
```
digits = datasets.load_digits()
print(digits.data[:5])
```

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

```
In [3]: breast_cancer = datasets.load_breast_cancer()
        print(breast_cancer.feature_names)
        print(breast_cancer.data[:5])
```

```
['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']
[[  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]
 [  1.14200000e+01   2.03800000e+01   7.75800000e+01   3.86100000e+02
    1.42500000e-01   2.83900000e-01   2.41400000e-01   1.05200000e-01
    2.59700000e-01   9.74400000e-02   4.95600000e-01   1.15600000e+00
    3.44500000e+00   2.72300000e+01   9.11000000e-03   7.45800000e-02
    5.66100000e-02   1.86700000e-02   5.96300000e-02   9.20800000e-03
    1.49100000e+01   2.65000000e+01   9.88700000e+01   5.67700000e+02
    2.09800000e-01   8.66300000e-01   6.86900000e-01   2.57500000e-01
    6.63800000e-01   1.73000000e-01]
 [  2.02900000e+01   1.43400000e+01   1.35100000e+02   1.29700000e+03
    1.00300000e-01   1.32800000e-01   1.98000000e-01   1.04300000e-01
    1.80900000e-01   5.88300000e-02   7.57200000e-01   7.81300000e-01
    5.43800000e+00   9.44400000e+01   1.14900000e-02   2.46100000e-02
    5.68800000e-02   1.88500000e-02   1.75600000e-02   5.11500000e-03
    2.25400000e+01   1.66700000e+01   1.52200000e+02   1.57500000e+03
    1.37400000e-01   2.05000000e-01   4.00000000e-01   1.62500000e-01
    2.36400000e-01   7.67800000e-02]]
```

Видим, что в случае digits --- **целочисленные неотрицательные признаки**, а в случае breast_cancer --- **вещественные**.

Сравним качество работы наивных байесовских оценок на датасете **digits**.

In [21]:
```
def score(X, y):
    bnb = BernoulliNB()
    mnb = MultinomialNB()
    gnb = GaussianNB()
    print('BernoulliNB: {:.2f}'.format(np.mean(cross_val_score(bnb, X, y))))
    print('MultinomialNB: {:.2f}'.format(np.mean(cross_val_score(mnb, X, y))))
    print('GaussianNB: {:.2f}'.format(np.mean(cross_val_score(gnb, X, y))))
```

In [26]:
```
X = digits.data
y = digits.target
score(X, y)
```

```
BernoulliNB: 0.83
MultinomialNB: 0.87
GaussianNB: 0.82
```

Теперь сравним качество работы наивных байесовских оценок на датасете **breast_cancer**.

In [25]:
```
X = breast_cancer.data
y = breast_cancer.target
score(X, y)
```

```
BernoulliNB: 0.63
MultinomialNB: 0.89
GaussianNB: 0.94
```

1. На датасете digits максимальное качество классификации у **MultinomialNB: 0.87**.
2. На датасете breast_cancer максимальное качество классификации у **GaussianNB: 0.94**.
3. Верны следующие утверждения: c) мультиномиальное распределение лучше показало себя на выборке с целыми неотрицательными значениями признаков, d) на вещественных признаках лучше всего сработало нормальное распределение