Лабораторная работа №5. Решение задач классификации с помощью байесовского классификатора и метода k-ближайших соседей.

Часть 2. Полиномиальный наивный байесовский классификатор.

Используемый набор данных: [Breast Cancer Wisconsin (Diagnostic) (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29)

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
import os
import requests

%matplotlib inline

pd.options.display.max_columns = None
```

In [2]:

```
def downloadFile(url, filePath):
    if not os.path.exists(filePath):
        req = requests.get(url)
        f = open(filePath, "wb")
        f.write(req.content)
        f.close

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsi
n/"
downloadFile(url + "/wdbc.data", "dataset/wdbc.data")
downloadFile(url + "/wdbc.names", "dataset/wdbc.names")
```

In [3]:

```
headers = ["ID", "Diagnosis", "Radius Mean", "Texture Mean", "Perimeter Mean", "Area Me
an", "Smoothness Mean",
           "Compactness Mean", "Concavity Mean", "Concave points Mean", "Symmetry Mean"
, "Fractal dimension Mean",
           "Radius SE", "Texture SE", "Perimeter SE", "Area SE", "Smoothness SE", "Comp
actness SE", "Concavity SE",
           "Concave points SE", "Symmetry SE", "Fractal dimension SE", "Radius Worst",
"Texture Worst", "Perimeter Worst",
           "Area Worst", "Smoothness Worst", "Compactness Worst", "Concavity Worst", "C
oncave points Worst",
           "Symmetry Worst", "Fractal dimension Worst"]
data = pd.read_csv("dataset/wdbc.data", names=headers)
data = data.astype({"Diagnosis": "category"})
data.sample(40)
```

Out[3]:

	ID	Diagnosis	Radius Mean	Texture Mean	Perimeter Mean	Area Mean	Smoothness Mean	Compactness Mean
138	868826	M	14.950	17.57	96.85	678.1	0.11670	0.13050
424	907145	B	9.742	19.12	61.93	289.7	0.10750	0.08333
159	871149	B	10.900	12.96	68.69	366.8	0.07515	0.03718
116	864726	B	8.950	15.76	58.74	245.2	0.09462	0.12430
131	8670	M	15.460	19.48	101.70	748.9	0.10920	0.12230
478	911685	B	11.490	14.59	73.99	404.9	0.10460	0.08228
264	889719	M	17.190	22.07	111.60	928.3	0.09726	0.08995
219	88119002	M	19.530	32.47	128.00	1223.0	0.08420	0.11300
134	867739	M	18.450	21.91	120.20	1075.0	0.09430	0.09709
27	852781	M	18.610	20.25	122.10	1094.0	0.09440	0.10660
430	907914	M	14.900	22.53	102.10	685.0	0.09947	0.22250
375	901303	B	16.170	16.07	106.30	788.5	0.09880	0.14380
520	917092	B	9.295	13.90	59.96	257.8	0.13710	0.12250
568	92751	B	7.760	24.54	47.92	181.0	0.05263	0.04362
186	874217	M	18.310	18.58	118.60	1041.0	0.08588	0.08468
65	859283	M	14.780	23.94	97.40	668.3	0.11720	0.14790
468	9113538	M	17.600	23.33	119.00	980.5	0.09289	0.20040
269	8910720	B	10.710	20.39	69.50	344.9	0.10820	0.12890
562	925622	M	15.220	30.62	103.40	716.9	0.10480	0.20870
557	925236	B	9.423	27.88	59.26	271.3	0.08123	0.04971
395	903811	B	14.060	17.18	89.75	609.1	0.08045	0.05361
217	8811779	B	10.200	17.48	65.05	321.2	0.08054	0.05907
91	861799	M	15.370	22.76	100.20	728.2	0.09200	0.10360
122	865423	M	24.250	20.20	166.20	1761.0	0.14470	0.28670
462	9113156	B	14.400	26.99	92.25	646.1	0.06995	0.05223
481	91227	B	13.900	19.24	88.73	602.9	0.07991	0.05326
434	908469	B	14.860	16.94	94.89	673.7	0.08924	0.07074
259	88725602	M	15.530	33.56	103.70	744.9	0.10630	0.16390
155	8711003	B	12.250	17.94	78.27	460.3	0.08654	0.06679
248	88466802	B	10.650	25.22	68.01	347.0	0.09657	0.07234
413	905557	B	14.990	22.11	97.53	693.7	0.08515	0.10250
69	859487	B	12.780	16.49	81.37	502.5	0.09831	0.05234
376	901315	B	10.570	20.22	70.15	338.3	0.09073	0.16600
556	924964	B	10.160	19.59	64.73	311.7	0.10030	0.07504
19	8510426	B	13.540	14.36	87.46	566.3	0.09779	0.08129
54	857438	M	15.100	22.02	97.26	712.8	0.09056	0.07081

	ID	Diagnosis	Radius Mean	Texture Mean	Perimeter Mean	Area Mean	Smoothness Mean	Compactness Mean	C
128	866458	B	15.100	16.39	99.58	674.5	0.11500	0.18070	
197	877159	M	18.080	21.84	117.40	1024.0	0.07371	0.08642	
153	87106	B	11.150	13.08	70.87	381.9	0.09754	0.05113	
531	91903901	B	11.670	20.02	75.21	416.2	0.10160	0.09453	

In [4]:

```
display(data.describe())
display(data.isna().sum())
```

	ID	Radius Mean	Texture Mean	Perimeter Mean	Area Mean	Smoothness Mean	Compac
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.0
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.1
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.0
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.0
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.0
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.0
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.1
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0.3

```
ID                         0
Diagnosis                  0
Radius Mean                0
Texture Mean               0
Perimeter Mean             0
Area Mean                  0
Smoothness Mean            0
Compactness Mean           0
Concavity Mean             0
Concave points Mean        0
Symmetry Mean              0
Fractal dimension Mean     0
Radius SE                  0
Texture SE                 0
Perimeter SE               0
Area SE                    0
Smoothness SE              0
Compactness SE             0
Concavity SE               0
Concave points SE          0
Symmetry SE                0
Fractal dimension SE       0
Radius Worst               0
Texture Worst              0
Perimeter Worst            0
Area Worst                 0
Smoothness Worst           0
Compactness Worst          0
Concavity Worst            0
Concave points Worst       0
Symmetry Worst             0
Fractal dimension Worst    0
dtype: int64
```

Пропусков в данных нет.

Подготовим данные для классификации: выберем признаки и метки и сформируем тренировочные и тестовые наборы.

In [5]:

```
X = data.drop(columns=["ID", "Diagnosis"]).copy()
y = data["Diagnosis"].copy().cat.codes

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2
4)
```

Создадим классификатор, обучим его, а затем выполним классификацию.

In [6]:

```
y_pred = MultinomialNB().fit(X_train, y_train).predict(X_test)
```

Оценим получившуюся классификацию.

In [7]:

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.97      0.93        73
           1       0.94      0.78      0.85        41

    accuracy                           0.90       114
   macro avg       0.91      0.88      0.89       114
weighted avg       0.91      0.90      0.90       114
```

In [8]:

```
fpr, tpr, _ = roc_curve(y_test, y_pred)
auc = roc_auc_score(y_test, y_pred)
```

In [9]:

```
plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b', label = "AUC = %0.2f"%auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```