

Московский государственный технический университет имени Н. Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5

Отчёт по  
лабораторной работе № 3  
«Технологии машинного обучения»

Подготовил:

Кан Андрей Дмитриевич

Группа ИУ5-64Б

Подпись\_\_\_\_\_

Дата\_\_\_\_\_

Москва  
2021г.

**Цель лабораторной работы:** изучение способов подготовки выборки и подбора гиперпараметров на примере метода ближайших соседей.

## Задание:

1. Выберите набор данных (датасет) для решения задачи классификации или регрессии.
2. С использованием метода `train_test_split` разделите выборку на обучающую и тестовую.
3. Обучите модель ближайших соседей для произвольно заданного гиперпараметра  $K$ . Оцените качество модели с помощью подходящих для задачи метрик.
4. Произведите подбор гиперпараметра  $K$  с использованием `GridSearchCV` и/или `RandomizedSearchCV` и кросс-валидации, оцените качество оптимальной модели. Желательно использование нескольких стратегий кросс-валидации.
5. Сравните метрики качества исходной и оптимальной моделей.

## Текст программы:

```
import numpy as np
import pandas as pd
from sklearn.datasets import *
from sklearn.model_selection import train_test_split
import seaborn as sns
import matplotlib.pyplot as plt

from operator import itemgetter
import matplotlib.ticker as ticker
import math

from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score,
classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import mean_absolute_error, mean_squared_error,
mean_squared_log_error, median_absolute_error, r2_score
from sklearn.metrics import roc_curve, roc_auc_score

from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier

from sklearn.model_selection import cross_val_score, cross_validate
from sklearn.model_selection import KFold, RepeatedKFold, LeaveOneOut,
LeavePOut, ShuffleSplit, StratifiedKFold

from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer

from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.model_selection import learning_curve, validation_curve

%matplotlib inline
sns.set(style="ticks")
```

# Выборка датасета и ее разделение на тестовую и обучающую

```
wine = load_wine()
```

```
for x in wine:  
    print(x)
```

```
data  
target  
frame  
target_names  
DESCR  
feature_names
```

```
# Сформируем DataFrame  
wine_df = pd.DataFrame(data= np.c_[wine['data'], wine['target']],  
                        columns= list(wine['feature_names']) + ['target'])
```

```
wine_df
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_p_henols	flavanoids	nonflavanoid_phenols	proanthocyanins	co
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.6
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.3
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.6
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.8
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.3
...	...	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	7.7
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	7.3
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	10
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	9.3
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	9.2

178 rows × 14 columns

```
sc = MinMaxScaler()  
wine_sc = sc.fit_transform(wine.data)
```

```
X_train, X_test, Y_train, Y_test = train_test_split(  
wine_sc, wine.target, test_size=0.33, random_state=1)
```

```
X_train
```

```
array([[0.54736842, 0.05335968, 0.18181818, ..., 0.5203252 , 0.6996337 ,  
        0.15977175],  
       [0.53157895, 1.          , 0.41176471, ..., 0.20325203, 0.67032967,  
        0.07275321],  
       [0.73684211, 0.16403162, 0.67379679, ..., 0.52845528, 0.47619048,  
        0.60770328],  
       ...,  
       [0.64736842, 0.18181818, 0.47058824, ..., 0.40650407, 0.55311355,
```

```
0.13837375],
[0.5      , 0.40909091, 0.71657754, ..., 0.23577236, 0.38095238,
0.2296719 ],
[0.53157895, 0.1798419 , 0.63636364, ..., 0.5203252 , 0.45421245,
0.58987161]])
```

Y\_train

```
array([1, 1, 0, 2, 1, 2, 0, 1, 0, 1, 0, 2, 2, 2, 2, 1, 1, 0, 2, 0, 1, 2,
0, 1, 0, 2, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 2, 1, 1, 1, 1, 0, 0, 0,
2, 0, 1, 2, 2, 0, 1, 0, 1, 1, 0, 2, 1, 1, 2, 2, 1, 1, 1, 2, 2, 1,
0, 1, 0, 2, 1, 1, 0, 0, 1, 0, 0, 0, 2, 0, 2, 2, 0, 1, 1, 2, 0, 1,
1, 0, 0, 0, 1, 1, 0, 2, 2, 1, 1, 1, 0, 2, 2, 2, 2, 2, 1, 0, 0, 2,
1, 1, 2, 1, 2, 2, 1, 2, 0])
```

X\_test

```
array([[0.7      , 0.49802372, 0.63101604, 0.48453608, 0.40217391,
0.29310345, 0.0464135 , 0.69811321, 0.12302839, 0.39249147,
0.3902439 , 0.2014652 , 0.28673324],
[0.36578947, 0.17193676, 0.44385027, 0.61340206, 0.41304348,
0.35172414, 0.36919831, 0.39622642, 0.3785489 , 0.0665529 ,
0.47154472, 0.61904762, 0.04778887],
[0.68684211, 0.46640316, 0.64171123, 0.2371134 , 0.5      ,
0.59310345, 0.56751055, 0.0754717 , 0.39432177, 0.32593857,
0.3902439 , 0.76556777, 0.40442225],
[0.31052632, 0.08893281, 0.20855615, 0.31958763, 0.88043478,
0.3      , 0.19831224, 0.01886792, 0.65930599, 0.13395904,
0.6504065 , 0.65934066, 0.31383738],
[0.72105263, 0.22924901, 0.70588235, 0.33505155, 0.48913043,
0.69655172, 0.51687764, 0.49056604, 0.40063091, 0.42832765,
0.52845528, 0.60805861, 0.78245364],
[0.64736842, 0.56324111, 0.44385027, 0.45876289, 0.19565217,
0.22068966, 0.02953586, 0.8490566 , 0.14826498, 0.37713311,
0.26829268, 0.2014652 , 0.21540656],
[0.19210526, 0.38339921, 0.8342246 , 0.48453608, 0.35869565,
0.26551724, 0.35654008, 0.88679245, 0.20189274, 0.21501706,
0.6097561 , 0.45054945, 0.23466476],
[0.88157895, 0.22332016, 0.54545455, 0.07216495, 0.34782609,
0.8      , 0.69620253, 0.30188679, 0.8044164 , 0.53071672,
0.58536585, 0.63369963, 0.90513552],
[0.35      , 0.61067194, 0.54545455, 0.53608247, 0.19565217,
0.45517241, 0.12236287, 0.69811321, 0.19873817, 0.54351536,
0.06504065, 0.11355311, 0.17261056],
[0.44473684, 0.19960474, 0.49197861, 0.61340206, 0.15217391,
0.13793103, 0.29957806, 0.66037736, 0.38485804, 0.17235495,
0.32520325, 0.42124542, 0.14978602],
[0.81315789, 0.14624506, 0.51336898, 0.31958763, 0.27173913,
0.42068966, 0.44092827, 0.24528302, 0.3659306 , 0.31740614,
0.56097561, 0.56776557, 0.7146933 ],
[0.58157895, 0.36561265, 0.80748663, 0.53608247, 0.52173913,
0.62758621, 0.49578059, 0.49056604, 0.44479495, 0.25938567,
0.45528455, 0.60805861, 0.32596291],
[0.31315789, 0.10869565, 0.31016043, 0.43298969, 0.23913043,
0.47586207, 0.35864979, 0.49056604, 0.52681388, 0.12116041,
0.30894309, 0.64102564, 0.02425107],
[0.75      , 0.22727273, 0.65775401, 0.22680412, 0.33695652,
0.78275862, 0.67932489, 0.0754717 , 0.40694006, 0.35409556,
0.32520325, 0.83882784, 0.58273894],
[0.21315789, 0.02964427, 0.65240642, 0.3814433 , 0.26086957,
0.42068966, 0.39451477, 0.16981132, 0.61198738, 0.15102389,
0.25203252, 0.66300366, 0.17261056],
[0.1      , 0.      , 0.60962567, 0.53608247, 0.19565217,
```

0.51724138, 0.35232068, 0.54716981, 0.32492114, 0.15358362,  
0.50406504, 0.38095238, 0.11126961],  
[0.54736842, 0.22924901, 0.74331551, 0.76804124, 0.5  
0.42068966, 0.19831224, 0.24528302, 0.36277603, 0.49658703,  
0.10569106, 0.02197802, 0.10485021],  
[0.64473684, 0.21146245, 0.56149733, 0.51030928, 0.32608696,  
0.59310345, 0.55696203, 0.24528302, 0.45741325, 0.32593857,  
0.45528455, 0.80586081, 0.45791726],  
[0.36578947, 0.35770751, 0.48663102, 0.58762887, 0.2173913 ,  
0.24137931, 0.3164557 , 1. , 0.31861199, 0.12116041,  
0.30894309, 0.74358974, 0.02639087],  
[0.67105263, 0.18181818, 0.53475936, 0.43814433, 0.39130435,  
0.64827586, 0.60126582, 0.16981132, 0.48580442, 0.47952218,  
0.49593496, 0.58974359, 0.88231098],  
[0.73684211, 0.1798419 , 0.6631016 , 0.34020619, 0.26086957,  
0.50689655, 0.55907173, 0.16981132, 0.59305994, 0.36860068,  
0.61788618, 0.76923077, 0.70399429],  
[0.36842105, 0.15612648, 0.4973262 , 0.56185567, 0.17391304,  
0.60689655, 0.592827 , 0.49056604, 0.42902208, 0.22696246,  
0.17073171, 0.57509158, 0.05278174],  
[0.48157895, 0.12055336, 0.51336898, 0.3814433 , 0.56521739,  
0.18275862, 0.19198312, 0.1509434 , 0.16719243, 0.24061433,  
0.22764228, 0.00732601, 0.2510699 ],  
[0.51578947, 0.18379447, 0.6631016 , 1. , 0.75 ,  
0.8 , 0.53797468, 0.1509434 , 0.48895899, 0.17662116,  
0.67479675, 0.81684982, 0.5042796 ],  
[0.66578947, 0.1916996 , 0.50802139, 0.28865979, 0.51086957,  
0.74827586, 0.62236287, 0.39622642, 0.60883281, 0.41382253,  
0.38211382, 0.77289377, 0.36875892],  
[0.47894737, 0.5 , 0.65240642, 0.58762887, 0.39130435,  
0.23103448, 0.05485232, 0.88679245, 0.17350158, 0.3668942 ,  
0.31707317, 0.30769231, 0.20827389],  
[0.75526316, 0.18577075, 0.40641711, 0.27835052, 0.33695652,  
0.73103448, 0.64345992, 0.1509434 , 0.54574132, 0.4112628 ,  
0.3495935 , 0.75457875, 0.5042796 ],  
[0.78684211, 0.18577075, 0.45454545, 0.27835052, 0.2826087 ,  
0.57586207, 0.41983122, 0.24528302, 0.49526814, 0.29180887,  
0.45528455, 0.84981685, 0.53994294],  
[0.86052632, 0.23320158, 0.72727273, 0.48453608, 0.54347826,  
0.62758621, 0.5907173 , 0.37735849, 0.49211356, 0.41979522,  
0.4796748 , 0.50549451, 0.7146933 ],  
[0.48421053, 0.76482213, 0.59893048, 0.56185567, 0.17391304,  
0.24827586, 0.06540084, 0.64150943, 0.14195584, 0.54351536,  
0.04878049, 0.21611722, 0.24750357],  
[0.33157895, 0.13241107, 0.3315508 , 0.27835052, 0.16304348,  
0.54137931, 0.4556962 , 0.30188679, 0.42902208, 0.13822526,  
0.6097561 , 0.53846154, 0.10699001],  
[0.64473684, 0.18379447, 0.68449198, 0.61340206, 0.20652174,  
0.55862069, 0.16033755, 0.73584906, 0.59305994, 0.89334471,  
0.07317073, 0.18681319, 0.24393723],  
[0.46315789, 0.38142292, 0.59893048, 0.58762887, 0.45652174,  
0.17241379, 0.21518987, 0.20754717, 0.2681388 , 0.81228669,  
0. , 0.07326007, 0.14407989],  
[0.83421053, 0.20158103, 0.5828877 , 0.2371134 , 0.45652174,  
0.78965517, 0.64345992, 0.39622642, 0.49211356, 0.46672355,  
0.46341463, 0.57875458, 0.83594864],  
[0.11052632, 0.32806324, 0.56684492, 0.48453608, 0.2826087 ,  
0.66206897, 0.51687764, 0.35849057, 0.44794953, 0.16808874,  
0.2601626 , 0.77655678, 0.24750357],  
[0.15526316, 0.24703557, 0.49197861, 0.3814433 , 0.30434783,  
0.70344828, 0.40506329, 0.0754717 , 0.29652997, 0.16808874,  
0.55284553, 0.61904762, 0.04778887],  
[0.25526316, 0.15217391, 0.56684492, 0.58762887, 0.17391304,  
0.16206897, 0.19198312, 0.69811321, 0.38485804, 0.19795222,

0.46341463, 0.50549451, 0.12268188],  
[0.44473684, 0.21146245, 0.44919786, 0.42268041, 0.17391304,  
0.42068966, 0.46202532, 0.24528302, 0.42902208, 0.22354949,  
0.55284553, 0.68498168, 0.31098431],  
[0.27631579, 0.1284585, 0.60962567, 0.61340206, 0.15217391,  
0.54482759, 0.41139241, 0.56603774, 0.19873817, 0.13822526,  
0.36585366, 0.7032967, 0.07631954],  
[0.80789474, 0.25296443, 0.55614973, 0.42268041, 0.35869565,  
0.61034483, 0.5443038, 0.35849057, 0.6214511, 0.41979522,  
0.4796748, 0.54212454, 0.55777461],  
[0.71315789, 0.18379447, 0.47593583, 0.29896907, 0.52173913,  
0.55862069, 0.54008439, 0.1509434, 0.38170347, 0.38993174,  
0.35772358, 0.70695971, 0.55777461],  
[0.35263158, 0.03952569, 0., 0., 0.19565217,  
0.34482759, 0.04852321, 0.28301887, 0.00315457, 0.05716724,  
0.46341463, 0.2014652, 0.17261056],  
[0.71052632, 0.71541502, 0.48128342, 0.61340206, 0.19565217,  
0.10344828, 0.02742616, 0.73584906, 0.23343849, 0.4556314,  
0.24390244, 0.17582418, 0.17261056],  
[0.83947368, 0.64229249, 0.61497326, 0.13402062, 0.63043478,  
0.69655172, 0.56962025, 0.13207547, 0.52681388, 0.32593857,  
0.33333333, 0.82783883, 0.34379458],  
[0.83947368, 0.18972332, 0.5026738, 0.29381443, 0.52173913,  
0.76551724, 0.56118143, 0.24528302, 0.51104101, 0.43515358,  
0.37398374, 0.74725275, 0.4935806 ],  
[0.53157895, 0.20355731, 0.39572193, 0.32989691, 0.40217391,  
0.69655172, 0.56118143, 0.28301887, 0.51104101, 0.32081911,  
0.32520325, 0.76190476, 0.43295292],  
[0.34210526, 0.04940711, 0.31550802, 0.21649485, 0.7173913,  
0.31724138, 0.3185654, 0.41509434, 0.74132492, 0.18088737,  
0.47154472, 0.38095238, 0.33666191],  
[0.71842105, 0.15612648, 0.71657754, 0.45876289, 0.67391304,  
0.67931034, 0.50632911, 0.69811321, 0.29652997, 0.35153584,  
0.62601626, 0.63369963, 0.68259629],  
[0.83157895, 0.16798419, 0.59893048, 0.30412371, 0.41304348,  
0.8, 0.75738397, 0.35849057, 0.45741325, 0.6331058,  
0.6097561, 0.56776557, 1. ],  
[0.70789474, 0.13636364, 0.60962567, 0.31443299, 0.41304348,  
0.83448276, 0.70253165, 0.11320755, 0.51419558, 0.47098976,  
0.33333333, 0.58608059, 0.71825963],  
[0.2, 0.27470356, 0.75935829, 0.92268041, 0.23913043,  
0.39655172, 0.40084388, 0.8490566, 0.42586751, 0.14675768,  
0.39837398, 0.42857143, 0.13409415],  
[0.82368421, 0.34980237, 0.59893048, 0.48453608, 0.22826087,  
0.24137931, 0.07594937, 0.58490566, 0.26182965, 0.71843003,  
0.11382114, 0.16117216, 0.2724679 ],  
[0.60263158, 0.49407115, 0.54545455, 0.56185567, 0.23913043,  
0.32758621, 0.08860759, 0.60377358, 0.26498423, 0.60921502,  
0.05691057, 0.12820513, 0.26533524],  
[0.71578947, 0.19565217, 0.56149733, 0.27835052, 0.20652174,  
0.55862069, 0.51054852, 0.30188679, 0.44164038, 0.36860068,  
0.54471545, 0.5970696, 0.74322397],  
[0.27631579, 0.21541502, 0.51336898, 0.40721649, 0.11956522,  
0.2137931, 0.24472574, 0.73584906, 0.38801262, 0.09556314,  
0.48780488, 0.36630037, 0.14407989],  
[0.37894737, 0.1541502, 0.44919786, 0.43298969, 1.,  
0.52413793, 0.407173, 0.35849057, 0.90536278, 0.11262799,  
0.55284553, 0.4981685, 0.4700428 ],  
[0.74736842, 0.22924901, 0.77005348, 0.45360825, 0.40217391,  
0.67931034, 0.55485232, 0.45283019, 0.42586751, 0.27474403,  
0.62601626, 0.78021978, 0.45435093],  
[0.22105263, 0.7055336, 0.55080214, 0.53608247, 0.13043478,  
0.64827586, 0.56751055, 0.1509434, 0.78864353, 0.12969283,  
0.2195122, 0.86813187, 0.07275321],

```
[0.41315789, 0.33992095, 0.44919786, 0.40721649, 0.26086957,  
0.22068966, 0.06751055, 0.94339623, 0.16719243, 0.49658703,  
0.20325203, 0.11355311, 0.29743224]])
```

## Обучение модели и оценка ее качества

```
# 2 ближайших соседа
```

```
reg1_1 = KNeighborsClassifier(n_neighbors=2)  
reg1_1.fit(X_train, Y_train)  
target1_1 = reg1_1.predict(X_test)  
len(target1_1), target1_1
```

```
(59,  
array([2, 1, 0, 1, 0, 1, 1, 0, 2, 1, 0, 0, 1, 0, 1, 1, 2, 0, 1, 0, 0, 1,  
       2, 0, 0, 2, 0, 0, 0, 2, 1, 2, 2, 0, 1, 1, 1, 1, 1, 0, 0, 1, 2, 0,  
       0, 0, 0, 0, 0, 0, 1, 2, 2, 0, 1, 0, 0, 1, 2]))
```

```
accuracy_score(Y_test, target1_1)
```

```
0.9322033898305084
```

## Кросс-валидация

```
scoring = {'precision': 'precision_weighted',  
          'recall': 'recall_weighted',  
          'f1': 'f1_weighted'}
```

```
kf = KFold(n_splits=5)  
scores = cross_validate(KNeighborsClassifier(n_neighbors=2),  
                        wine_sc, wine.target, scoring=scoring,  
                        cv=kf, return_train_score=True)
```

```
scores
```

```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/  
_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being  
set to 0.0 in labels with no true samples. Use `zero_division` parameter to  
control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/  
_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being  
set to 0.0 in labels with no true samples. Use `zero_division` parameter to  
control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/  
_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being  
set to 0.0 in labels with no true samples. Use `zero_division` parameter to  
control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/  
_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being  
set to 0.0 in labels with no true samples. Use `zero_division` parameter to  
control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
{'fit_time': array([0.00362945, 0.00061607, 0.00065041, 0.00089145,  
0.00047374]),  
'score_time': array([0.0074923 , 0.0051558 , 0.00435805, 0.00483489,  
0.00375938])}
```

```

'test_precision': array([1.          , 0.97337963, 1.          , 0.87643785, 1.
]),
'train_precision': array([0.98131094, 0.98665678, 0.96753961, 0.98003609,
0.96776661]),
'test_recall': array([1.          , 0.91666667, 0.80555556, 0.8          ,
0.88571429]),
'train_recall': array([0.97887324, 0.98591549, 0.96478873, 0.97902098,
0.96503497]),
'test_f1': array([1.          , 0.93930517, 0.89230769, 0.83576966, 0.93939394]),
'train_f1': array([0.97929199, 0.98598228, 0.96415013, 0.9789498 ,
0.96510759])}]

```

```

kf1 = LeaveOneOut()
scores1 = cross_validate(KNeighborsClassifier(n_neighbors=2),
                        wine_sc, wine.target, scoring=scoring,
                        cv=kf1, return_train_score=True)

```

```

scores1

```

```

/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/
_classification.py:1221: UndefinedMetricWarning: Precision is ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/
_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being
set to 0.0 in labels with no true samples. Use `zero_division` parameter to
control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/
_classification.py:1221: UndefinedMetricWarning: Precision is ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/
_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being
set to 0.0 in labels with no true samples. Use `zero_division` parameter to
control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/
_classification.py:1221: UndefinedMetricWarning: Precision is ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/
_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being
set to 0.0 in labels with no true samples. Use `zero_division` parameter to
control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/
_classification.py:1221: UndefinedMetricWarning: Precision is ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/
_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being
set to 0.0 in labels with no true samples. Use `zero_division` parameter to
control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/
_classification.py:1221: UndefinedMetricWarning: Precision is ill-defined and
being set to 0.0 in labels with no predicted samples. Use `zero_division`
parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))

```



```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
/home/ripperonik/anaconda3/lib/python3.8/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
{'fit_time': array([0.00096059, 0.00065112, 0.00068831, 0.00055265, 0.00053096,
                    0.00048018, 0.00052309, 0.00046158, 0.00045609, 0.00045848,
                    0.00045466, 0.000453 , 0.00046015, 0.00045443, 0.00046253,
                    0.0004518 , 0.0004549 , 0.00045395, 0.00045657, 0.00045347,
                    0.00045061, 0.00044918, 0.00045991, 0.00045347, 0.00044918,
                    0.00045061, 0.00045443, 0.00046182, 0.00045419, 0.00045466,
                    0.00045419, 0.0004518 , 0.00046134, 0.000453 , 0.00045085,
                    0.00045705, 0.00045514, 0.00044847, 0.00045204, 0.00044823,
                    0.00049806, 0.00050664, 0.00081825, 0.00049138, 0.00050497,
                    0.00050473, 0.00047874, 0.00049591, 0.00052547, 0.00048995,
                    0.00048709, 0.00051641, 0.00048923, 0.00048661, 0.00049472,
                    0.0004971 , 0.00049138, 0.00050449, 0.00049758, 0.00048637,
                    0.00048852, 0.00048423, 0.00049734, 0.00049019, 0.0005002 ,
                    0.00048876, 0.00049949, 0.00049281, 0.00049806, 0.00048923,
                    0.00051761, 0.00048518, 0.00053215, 0.00048304, 0.00049281,
                    0.00049543, 0.00048089, 0.00049353, 0.00047731, 0.00049305,
                    0.00048542, 0.00049543, 0.00050974, 0.00056601, 0.00052357,
                    0.00055599, 0.00048828, 0.00049305, 0.00049019, 0.00047851,
                    0.00046778, 0.00048351, 0.00049591, 0.00046659, 0.00049019,
                    0.00047898, 0.00045061, 0.00045395, 0.00044966, 0.000453 ,
                    0.00045443, 0.00045395, 0.00045538, 0.00048947, 0.00054622,
                    0.00048804, 0.00067258, 0.00057721, 0.00046682, 0.00046539,
                    0.0004549 , 0.00044799, 0.0004518 , 0.00045419, 0.00055575,
                    0.00048065, 0.00048256, 0.00051236, 0.00045538, 0.00051236,
                    0.00046253, 0.00045586, 0.00045848, 0.00046897, 0.00045466,
                    0.00045276, 0.0004549 , 0.0004549 , 0.00045347, 0.00044775,
                    0.00045109, 0.00045133, 0.00045013, 0.00044799, 0.00045609,
```

```
0.00045133, 0.00044966, 0.00045013, 0.00045133, 0.00058913,
0.00050044, 0.00057602, 0.00051451, 0.00051022, 0.00048065,
0.00046206, 0.00051165, 0.00045133, 0.00045419, 0.00044894,
0.00045013, 0.00045156, 0.00045347, 0.000453 , 0.00045991,
0.00045538, 0.00044823, 0.00044894, 0.00045228, 0.000458 ,
0.00045133, 0.00045371, 0.0004549 , 0.00045013, 0.00045347,
0.00045252, 0.00045276, 0.00045753, 0.00045061, 0.00045371,
0.00045419, 0.00045347, 0.00045133, 0.00045347, 0.00045252,
0.00046015, 0.00046301, 0.00045776]),
'score_time': array([0.00368166, 0.00296474, 0.00332165, 0.00237823,
0.00226593,
0.00225019, 0.00225377, 0.0021441 , 0.00228524, 0.00211668,
0.0021143 , 0.00211239, 0.0021143 , 0.00213385, 0.00212789,
0.00212216, 0.00213432, 0.0021069 , 0.00212693, 0.0021143 ,
0.00210261, 0.00211096, 0.0022192 , 0.00210595, 0.00209594,
0.00209785, 0.00209975, 0.0021081 , 0.00210667, 0.00215816,
0.00211358, 0.00211072, 0.00211787, 0.00210357, 0.00211334,
0.00210452, 0.00211382, 0.00212812, 0.00207663, 0.00211525,
0.00220871, 0.00217104, 0.00250244, 0.00234556, 0.00221324,
0.00216484, 0.00220656, 0.002141 , 0.00224447, 0.00214982,
0.00215077, 0.0022974 , 0.00214815, 0.00219917, 0.00214481,
0.00220203, 0.00214696, 0.00217748, 0.00222778, 0.00214243,
0.00219059, 0.00214982, 0.00278425, 0.00219154, 0.00212669,
0.00260568, 0.00214171, 0.00220585, 0.00216317, 0.0021801 ,
0.00220418, 0.00268173, 0.00219846, 0.00260925, 0.00218081,
0.0021739 , 0.00219417, 0.0021534 , 0.00213957, 0.00220513,
0.00229883, 0.0021615 , 0.00220871, 0.00302744, 0.00270104,
0.00235653, 0.00224662, 0.00213504, 0.00216246, 0.00214052,
0.0024097 , 0.00216866, 0.00215364, 0.00212765, 0.00211883,
0.00212502, 0.00207806, 0.00209713, 0.00209141, 0.00211382,
0.00210261, 0.00208569, 0.00208879, 0.00211978, 0.00230098,
0.00215745, 0.00253057, 0.0022707 , 0.00218177, 0.00224853,
0.00208712, 0.00209737, 0.00209665, 0.00210166, 0.00233698,
0.00220346, 0.00210905, 0.00215936, 0.00270224, 0.00216556,
0.00212097, 0.00249124, 0.00211906, 0.00249386, 0.00208759,
0.0020771 , 0.00209284, 0.00219607, 0.00208092, 0.00209332,
0.00209808, 0.00208187, 0.00208807, 0.00208044, 0.00208068,
0.00209451, 0.00208831, 0.00210118, 0.00219417, 0.00221896,
0.00214839, 0.00266218, 0.00216889, 0.00217152, 0.00209951,
0.0021081 , 0.00213695, 0.00207877, 0.00210214, 0.00207925,
0.00208735, 0.00210238, 0.00208187, 0.00221157, 0.00209594,
0.00209737, 0.00212288, 0.00212812, 0.00210929, 0.00209188,
0.00209713, 0.00210524, 0.00207496, 0.00211668, 0.00209498,
0.00209713, 0.00209308, 0.00207019, 0.00218081, 0.00211716,
0.00207472, 0.00211883, 0.00208902, 0.00208831, 0.00209332,
0.00206804, 0.00210595, 0.00209045]),
'test_precision': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 0., 1., 1., 0., 1., 1.,
1., 1., 1., 0., 1., 0., 1., 1., 1., 1., 1., 1., 1., 1., 1., 0., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 0.,
1., 1., 0., 1., 0., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1.] ),
'train_precision': array([0.97399336, 0.97399336, 0.97399336, 0.97399336,
0.97399336,
0.97399336, 0.97399336, 0.97399336, 0.97399336, 0.97399336,
0.97399336, 0.97399336, 0.97399336, 0.97399336, 0.97399336,
0.97399336, 0.97399336, 0.97399336, 0.97399336, 0.97399336,
```

[illegible]

[illegible]

# Оптимизация гиперпараметров

## Leave one out

```
n_range = np.array(range(1,70,2))
tuned_parameters = [{'n_neighbors': n_range}]
tuned_parameters

[{'n_neighbors': array([ 1,  3,  5,  7,  9, 11, 13, 15, 17, 19, 21, 23, 25, 27,
29, 31, 33,
35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67,
69])}]

%%time
clf_gs = GridSearchCV(KNeighborsClassifier(), tuned_parameters, cv=kf1,
scoring='accuracy')
clf_gs.fit(X_train, Y_train)

CPU times: user 5.4 s, sys: 16.4 ms, total: 5.42 s
Wall time: 5.43 s

GridSearchCV(cv=LeaveOneOut(), estimator=KNeighborsClassifier(),
param_grid=[{'n_neighbors': array([ 1,  3,  5,  7,  9, 11, 13, 15,
17, 19, 21, 23, 25, 27, 29, 31, 33,
35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67,
69])}],
scoring='accuracy')

clf_gs.cv_results_

{'mean_fit_time': array([0.00046182, 0.00041849, 0.00044304, 0.00040617,
0.00040489,
0.0004043 , 0.00040969, 0.00040566, 0.00040025, 0.00040062,
0.00040291, 0.00040128, 0.00040129, 0.00040148, 0.00040085,
0.00040104, 0.00040351, 0.00040162, 0.00040181, 0.00040111,
0.00040157, 0.00040119, 0.00040137, 0.00040149, 0.00040152,
0.00040177, 0.00042401, 0.00040202, 0.00040162, 0.00031492,
0.00031354, 0.0003141 , 0.00031317, 0.0003133 , 0.00031423]),
'std_fit_time': array([1.08563884e-04, 4.20835294e-05, 8.32594692e-05,
9.71204800e-06,
5.37929331e-06, 4.86559900e-06, 4.29522330e-05, 1.47133767e-05,
2.82893262e-06, 3.40932864e-06, 1.13157730e-05, 2.59953439e-06,
3.31976860e-06, 3.33274638e-06, 2.68403004e-06, 3.41901651e-06,
6.28205451e-06, 3.97894030e-06, 3.28075114e-06, 2.61671808e-06,
2.84521067e-06, 2.58114953e-06, 3.25142307e-06, 2.86902303e-06,
3.58173677e-06, 2.64324782e-06, 5.09376554e-05, 2.86178749e-06,
2.88841771e-06, 9.09102185e-06, 4.54328942e-06, 9.76318196e-06,
2.65614435e-06, 2.23180336e-06, 2.86143401e-06]),
'mean_score_time': array([0.00074346, 0.00066797, 0.00071483, 0.00064828,
0.00064612,
0.00064065, 0.00064578, 0.00065143, 0.00063571, 0.00063576,
0.00063876, 0.00063688, 0.0006369 , 0.00063882, 0.00063796,
0.00063877, 0.00064377, 0.00063993, 0.00064017, 0.0006404 ,
0.00063958, 0.00064044, 0.00064464, 0.00064106, 0.00064082,
0.00064103, 0.00102594, 0.00064513, 0.00064664, 0.00066149,
0.00057627, 0.00057929, 0.00057564, 0.00057556, 0.00057625]),
'std_score_time': array([1.92571582e-04, 7.07607239e-05, 1.32409334e-04,
3.19042087e-05,
2.88472320e-05, 2.00930727e-05, 5.36240240e-05, 6.80384546e-05,
1.96216083e-05, 2.14175976e-05, 3.23475759e-05, 2.07088689e-05,
1.97625846e-05, 2.74464954e-05, 2.04236900e-05, 2.08189564e-05,
```

```

3.00098346e-05, 2.26242817e-05, 2.04318177e-05, 2.80177724e-05,
2.08402135e-05, 2.11666841e-05, 3.05206080e-05, 2.03421522e-05,
2.07915199e-05, 1.74800579e-05, 3.78255426e-03, 2.03480754e-05,
3.02343766e-05, 9.02445926e-04, 5.94499577e-06, 1.68938753e-05,
4.62023640e-06, 5.19071514e-06, 7.10245842e-06]],
'param_n_neighbors': masked_array(data=[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21,
23, 25, 27, 29,
31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57,
59, 61, 63, 65, 67, 69],
mask=[False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False, False,
False, False, False],
fill_value='?',
dtype=object),
'params': [{ 'n_neighbors': 1},
{ 'n_neighbors': 3},
{ 'n_neighbors': 5},
{ 'n_neighbors': 7},
{ 'n_neighbors': 9},
{ 'n_neighbors': 11},
{ 'n_neighbors': 13},
{ 'n_neighbors': 15},
{ 'n_neighbors': 17},
{ 'n_neighbors': 19},
{ 'n_neighbors': 21},
{ 'n_neighbors': 23},
{ 'n_neighbors': 25},
{ 'n_neighbors': 27},
{ 'n_neighbors': 29},
{ 'n_neighbors': 31},
{ 'n_neighbors': 33},
{ 'n_neighbors': 35},
{ 'n_neighbors': 37},
{ 'n_neighbors': 39},
{ 'n_neighbors': 41},
{ 'n_neighbors': 43},
{ 'n_neighbors': 45},
{ 'n_neighbors': 47},
{ 'n_neighbors': 49},
{ 'n_neighbors': 51},
{ 'n_neighbors': 53},
{ 'n_neighbors': 55},
{ 'n_neighbors': 57},
{ 'n_neighbors': 59},
{ 'n_neighbors': 61},
{ 'n_neighbors': 63},
{ 'n_neighbors': 65},
{ 'n_neighbors': 67},
{ 'n_neighbors': 69}],
'split0_test_score': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1.,
1.]),
'split1_test_score': array([0., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1.]),
'split2_test_score': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1.]),

```

[illegible]

[illegible]



[illegible]

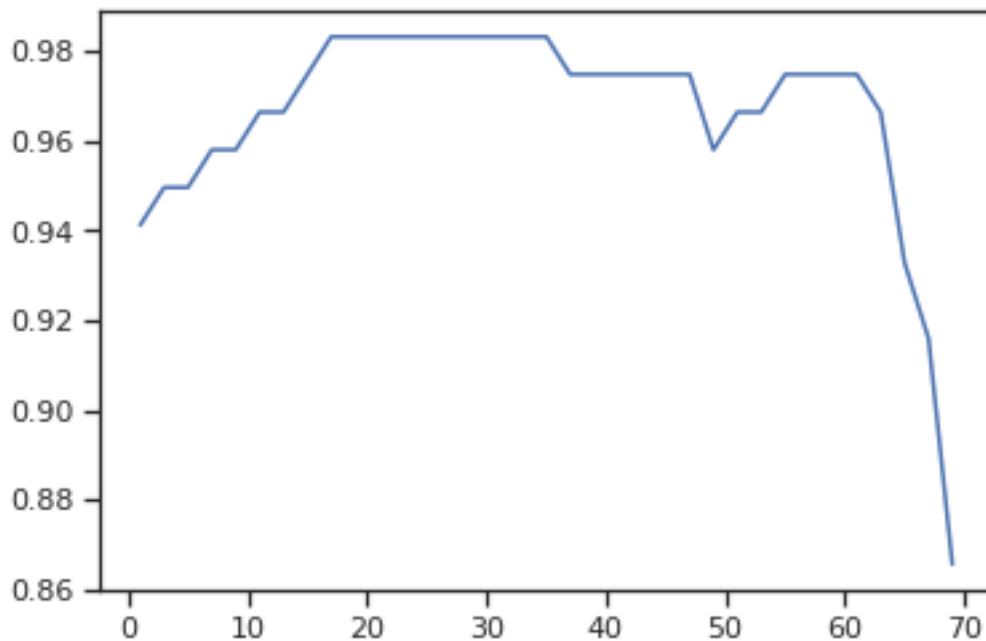
[illegible]

[illegible]

[illegible]

[illegible]





```
%%time
clf_rs = RandomizedSearchCV(KNeighborsClassifier(), tuned_parameters, cv=kf1,
scoring='accuracy')
clf_rs.fit(X_train, Y_train)
```

CPU times: user 1.66 s, sys: 7.79 ms, total: 1.67 s  
Wall time: 1.67 s

```
RandomizedSearchCV(cv=LeaveOneOut(), estimator=KNeighborsClassifier(),
param_distributions=[{'n_neighbors': array([ 1,  3,  5,  7,
9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33,
35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67,
69])}],
scoring='accuracy')
```

```
clf_rs.best_score_, clf_rs.best_params_

(0.9831932773109243, {'n_neighbors': 23})
```

## K-Fold

```
%%time
clf_gs = GridSearchCV(KNeighborsClassifier(), tuned_parameters, cv=kf,
scoring='accuracy')
clf_gs.fit(X_train, Y_train)
```

CPU times: user 366 ms, sys: 62 µs, total: 366 ms  
Wall time: 366 ms

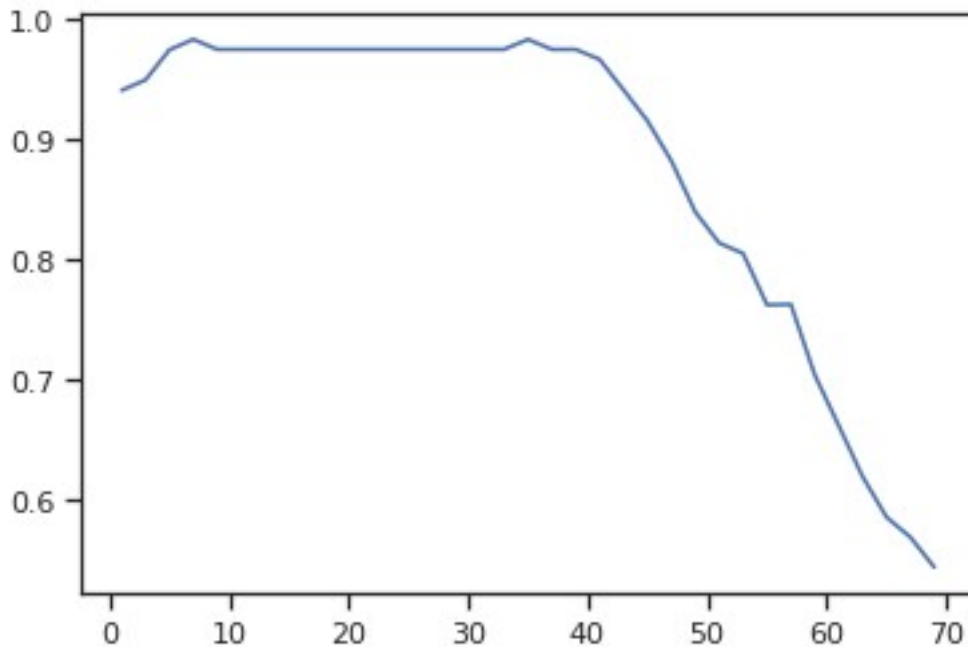
```
GridSearchCV(cv=KFold(n_splits=5, random_state=None, shuffle=False),
estimator=KNeighborsClassifier(),
param_grid=[{'n_neighbors': array([ 1,  3,  5,  7,  9, 11, 13, 15,
17, 19, 21, 23, 25, 27, 29, 31, 33,
35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67,
69])}],
scoring='accuracy')
```

```
clf_gs.best_score_, clf_gs.best_params_
```

```
(0.9833333333333334, {'n_neighbors': 7})
```

```
# Изменение качества на тестовой выборке в зависимости от K-соседей  
plt.plot(n_range, clf_gs.cv_results_['mean_test_score'])
```

```
[<matplotlib.lines.Line2D at 0x7f28b6fc3f40>]
```



```
%%time  
clf_rs = RandomizedSearchCV(KNeighborsClassifier(), tuned_parameters, cv=kf,  
scoring='accuracy')  
clf_rs.fit(X_train, Y_train)
```

```
CPU times: user 121 ms, sys: 0 ns, total: 121 ms  
Wall time: 118 ms
```

```
RandomizedSearchCV(cv=KFold(n_splits=5, random_state=None, shuffle=False),  
estimator=KNeighborsClassifier(),  
param_distributions=[{'n_neighbors': array([ 1,  3,  5,  7,  
9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33,  
35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67,  
69])}],  
scoring='accuracy')
```

```
clf_rs.best_score_, clf_rs.best_params_
```

```
(0.9833333333333334, {'n_neighbors': 35})
```