

**Московский государственный технический  
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Теория машинного обучения»

Отчет по лабораторной работе №3

Выполнила:  
студентка группы ИУ5-64  
Бредня Елизавета

Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Подпись и дата:

Москва, 2022 г.

## **Описание задания**

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 и её результаты

```
▶ from sklearn import datasets, svm
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import precision_score, recall_score

▶ data = pd.read_csv("/content/banknote.txt", index_col=False, header=None,
                     names=['DISP', 'SKEW', 'CUR', 'ENT', 'CLAS'])
data.head()



|   | DISP    | SKEW    | CUR     | ENT      | CLAS |
|---|---------|---------|---------|----------|------|
| 0 | 3.62160 | 8.6661  | -2.8073 | -0.44699 | 0    |
| 1 | 4.54590 | 8.1674  | -2.4586 | -1.46210 | 0    |
| 2 | 3.86600 | -2.6383 | 1.9242  | 0.10645  | 0    |
| 3 | 3.45660 | 9.5228  | -4.0112 | -3.59440 | 0    |
| 4 | 0.32924 | -4.4552 | 4.5718  | -0.98880 | 0    |



[ ] data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1372 entries, 0 to 1371
Data columns (total 5 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   DISP      1372 non-null    float64
 1   SKEW      1372 non-null    float64
 2   CUR       1372 non-null    float64
 3   ENT       1372 non-null    float64
 4   CLAS      1372 non-null    int64  
dtypes: float64(4), int64(1)
memory usage: 53.7 KB

[ ] X_train, X_test, y_train, y_test = train_test_split(
    data.drop('CLAS', 1), data["CLAS"], test_size=0.3, shuffle=True)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: In a fu

[ ] clf = KNeighborsClassifier(n_neighbors=55)
clf.fit(X_train, y_train)
target1_0 = clf.predict(X_train)
target1_1 = clf.predict(X_test)
accuracy_score(y_train, target1_0), accuracy_score(y_test, target1_1)

(0.9864583333333333, 0.9854368932038835)
```

```
[ ] parameters = {'n_neighbors': range (3, 100, 5)}
clf2 = KNeighborsClassifier()
grid = GridSearchCV(clf, parameters)

skf = StratifiedKFold(n_splits=7)
X = data.drop('CLAS', 1)
y = data["CLAS"]
for train_index, test_index in skf.split(X, y):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    grid.fit(X_train, y_train)
    print("Параметры: ", grid.best_params_)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: FutureWarning: In a future versio
Параметры: {'n_neighbors': 3}
```

▶ clf2 = KNeighborsClassifier(n\_neighbors=3)  
clf2.fit(X\_train, y\_train)  
target1\_0 = clf2.predict(X\_train)  
target1\_1 = clf2.predict(X\_test)  
accuracy\_score(y\_train, target1\_0), accuracy\_score(y\_test, target1\_1)

👤 (0.9991496598639455, 1.0)

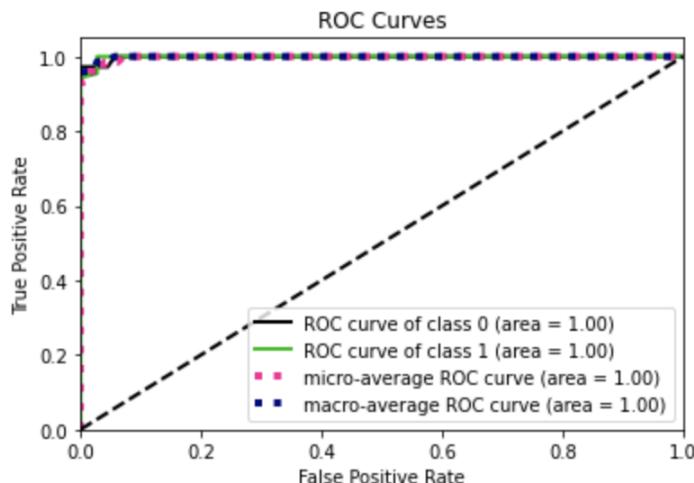
```
[ ] probs1 = clf.predict_proba(X_test)
probs2 = clf2.predict_proba(X_test)
```

```
[ ] %pip install scikit-plot
```

▶ import scikitplot as skplt
import matplotlib.pyplot as plt

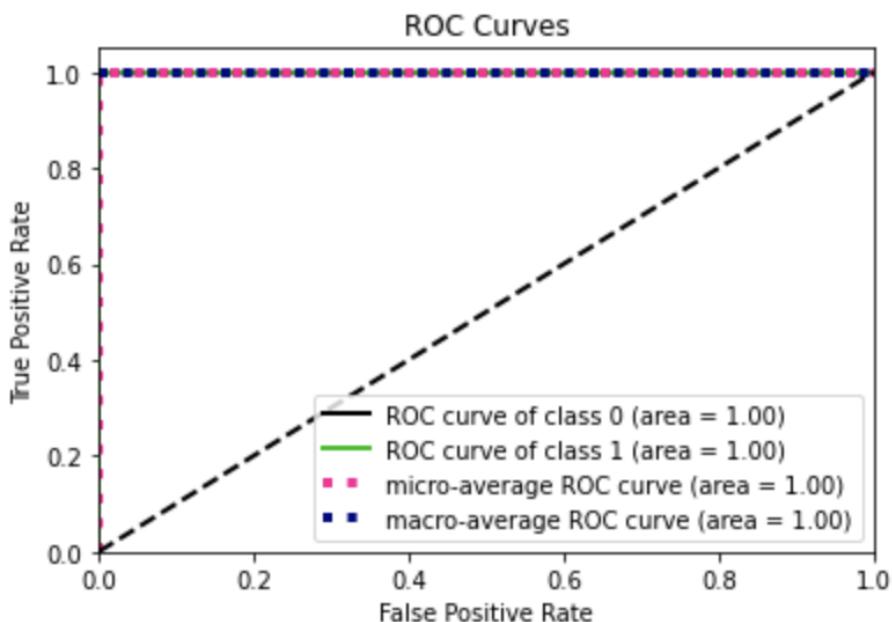
skplt.metrics.plot\_roc\_curve(y\_test, probs1)
plt.show()

👤 /usr/local/lib/python3.7/dist-packages/scikit-plot/metrics/\_roc.py:100:
warnings.warn(msg, category=FutureWarning)



```
▶ skplt.metrics.plot_roc_curve(y_test, probs2)
plt.show()

@ /usr/local/lib/python3.7/dist-packages/sklearn/utils/deprec
    warnings.warn(msg, category=FutureWarning)
```



```
▶ pred1 = clf.predict(X_test)
pred2 = clf2.predict(X_test)
print("Модель 1:")
print(precision_score(pred1, y_test))
print(recall_score(pred1, y_test))
print("Модель 2:")
print(precision_score(pred2, y_test))
print(recall_score(pred2, y_test))
```

Модель 1:  
1.0  
0.9565217391304348  
Модель 2:  
1.0  
1.0