

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №8
по дисциплине «Машинное обучение»
Тема: Классификация (линейный дискриминантный анализ, метод
опорных векторов)

Студент гр. 6304

Ковынев М.В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

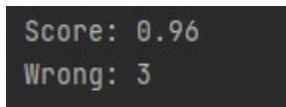
2020

Цель

Ознакомиться с методами классификации модуля Sklearn

Ход работы

1. Загрузить датасет по ссылке: <https://archive.ics.uci.edu/ml/datasets/iris> .
Данные представлены в виде data файла. Данные представляют собой информацию о трех классах цветов
2. Создан Python скрипт. Загружены данные в датафрейм
3. Выделены данные и их метки
4. Разбили выборку на обучающую и тестовую
5. Проведем классификацию наблюдений используя LDA



```
Score: 0.96
Wrong: 3
```

Рисунок 1 — Точность и количество наблюдений, который были неправильно определены

6. Параметры:

- `solver` — `svd` (разложение по сингулярным числам), `lsqr` (решение МНК), `eigen` (разложение на собственные числа)
- `shrinkage` — `auto` (автоматическое сжатие по лемме Ледуа-Вольфа), `[0, 1]`
- `priors` — класс априорных вероятностей. По умолчанию пропорции классов выводятся из данных обучения
- `n_components` — количество компонентов
- `store_covariance` — сохранение взвешанной ковариационной матрицы при `svd`
- `tol` — Абсолютный порог для того, чтобы единичное значение X считалось значимым, используется для оценки ранга X .

7. Атрибуты

- `coef_` — весовые вектора

- `intercept_` — массив прерывания
- `covariance_` — взвешенная внутриклассовая ковариационная матрица
- `explained_variance_ratio_` — процент дисперсии, объясняемой каждым из выбранных компонентов
- `means_` — средние в классах
- `priors_` — вероятности классов
- `scalings_` — масштабирование объектов в пространстве, охватываемом центроидами классов
- `xbar_` — общее среднее
- `classes_` — уникальные метки классов.

8. Построен график зависимости неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки. Размер тестовой выборки изменялся от 0.05 до 0.95 с шагом 0.05. Параметр `random_state` сделан равным номеру своей зачетной книжки - 630408.

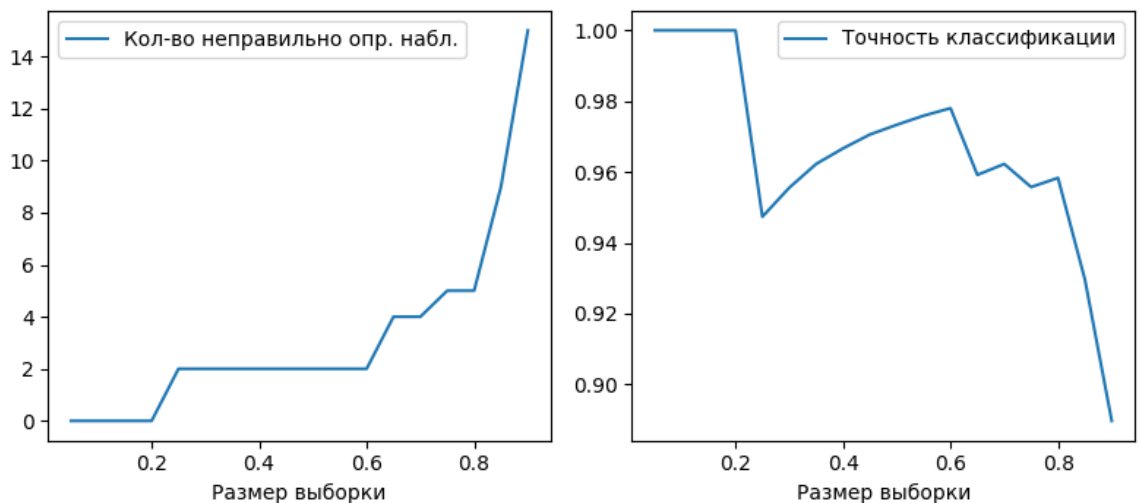


Рисунок 2 – LinearDiscriminantAnalysis

9. Функция `transform` проецирует данные для максимизации разбиения классов. LDA пытается определить атрибуты, на которые приходится наибольшая разница между классами. В частности, LDA, в отличие от PCA, является контролируемым методом, использующим известные

метки классов, то есть метод tranform позволяет уменьшить размерность данных.

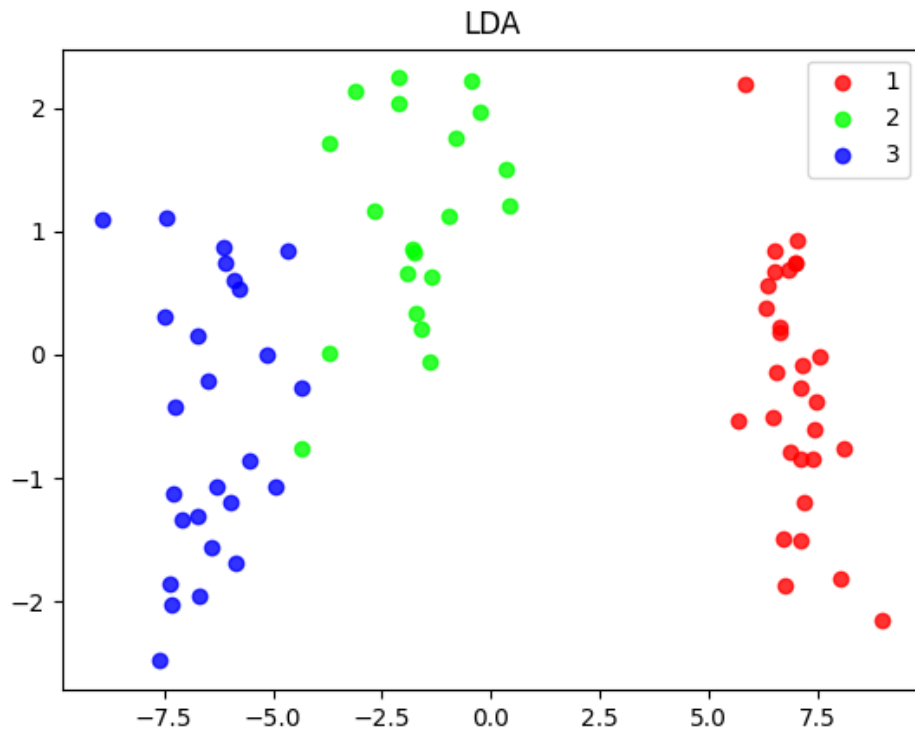


Рисунок 3 — Результат работы tranform

10. Исследована работа классификатор при различных параметрах solver, shrinkage

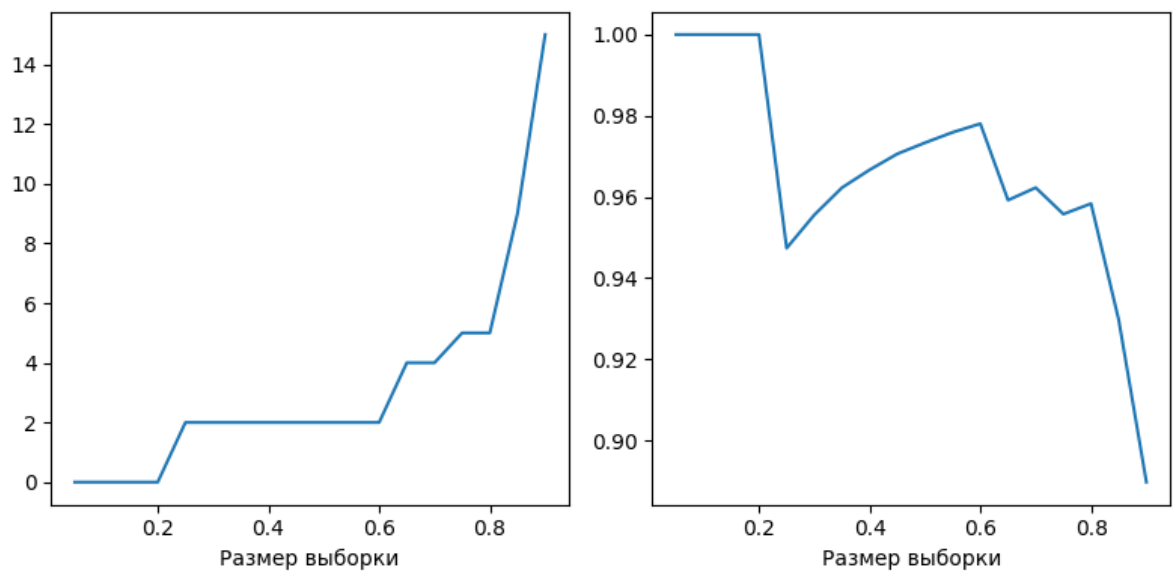


Рисунок 4 — solver=svd, shrinkage=None

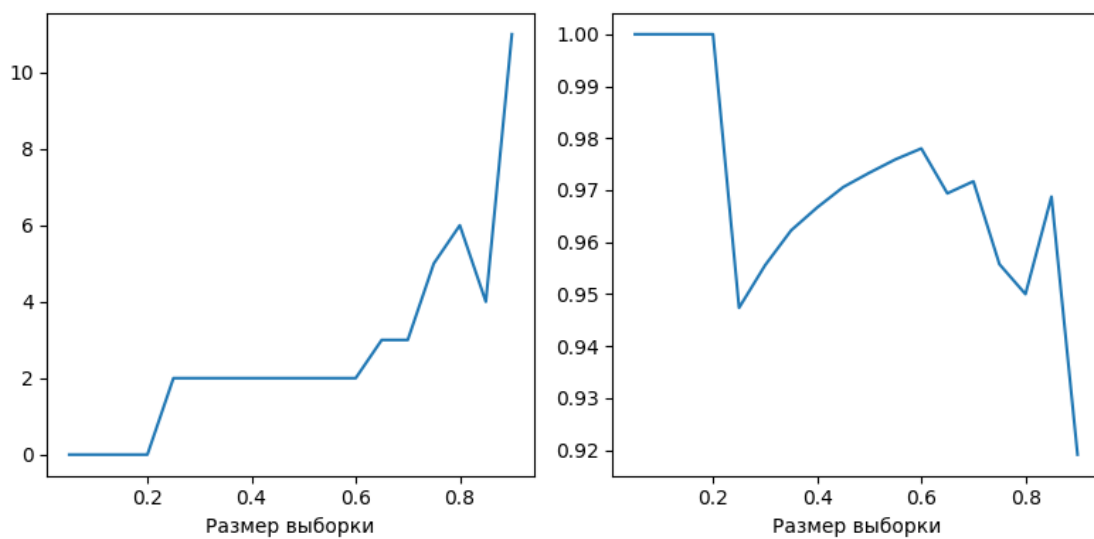


Рисунок 5 — solver=lsqr, shrinkage=auto

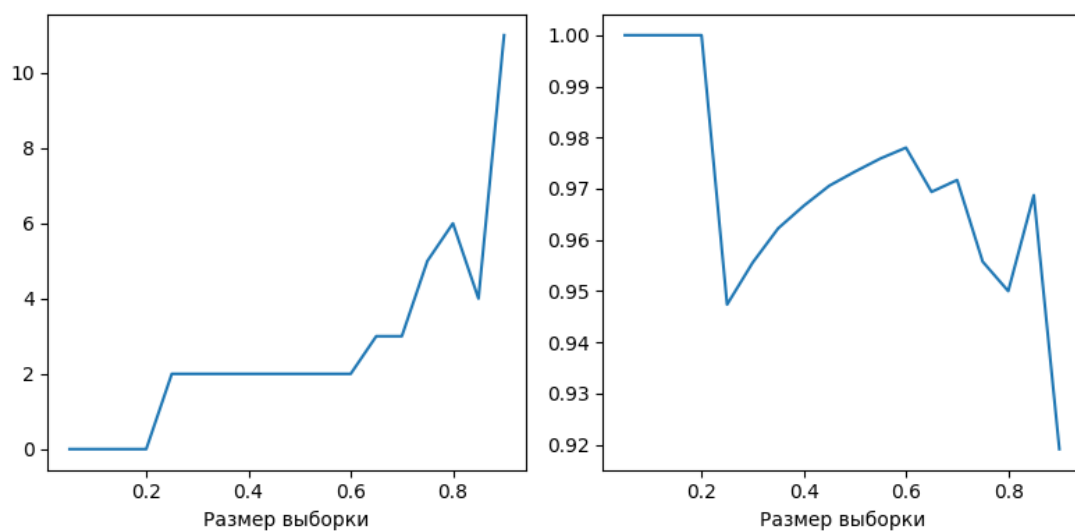


Рисунок 6 — solver=eigen, shrinkage=auto

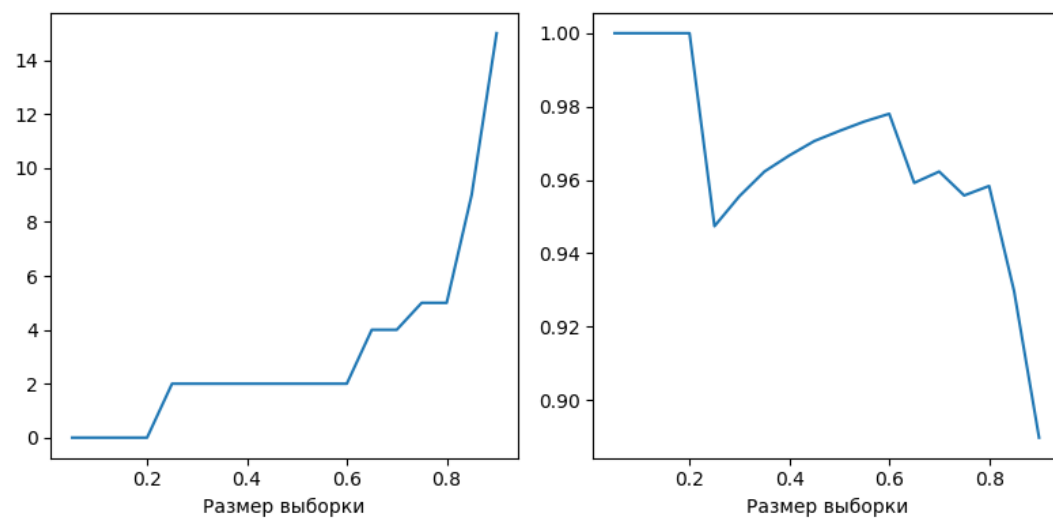


Рисунок 7 — solver=lsqr, shrinkage=None

11. Задана априорная вероятность классу с номером 1 равную 0.7, остальным классам задана равные априорные вероятности

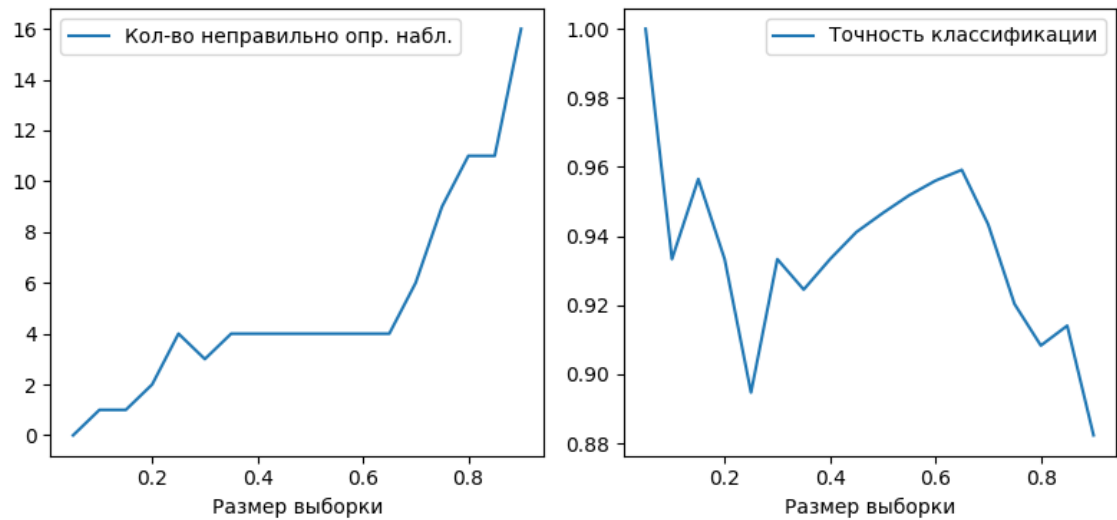


Рисунок 7 — LinearDiscriminantAnalysis(priors=[0.15, 0.7, 0.15])

12. Классифицируем при SVM на тех же данных

13. Используя функцию score() выведена точность классификации

```
Score: 0.96
Wrong: 3
```

Рисунок 8 — Точность и количество наблюдений, который были неправильно определены

14. Выведена следующая информация

```
clf.support_vectors_ [[4.5 2.3 1.3 0.3]
 [5.4 3.9 1.7 0.4]
 [5.1 3.3 1.7 0.5]
 [5. 3. 1.6 0.2]
 [5.1 2.5 3. 1.1]
 [6.2 2.2 4.5 1.5]
 [5.7 2.9 4.2 1.3]
 [5.7 2.8 4.5 1.3]
 [6.6 3. 4.4 1.4]
 [6.4 2.9 4.3 1.3]
 [4.9 2.4 3.3 1. ]
 [6.7 3.1 4.4 1.4]
 [5.7 2.6 3.5 1. ]
 [6.3 2.5 4.9 1.5]
 [6.7 3. 5. 1.7]
 [5.5 2.4 3.7 1. ]
 [6.6 2.9 4.6 1.3]
 [5.6 3. 4.1 1.3]
 [5.9 3.2 4.8 1.8]
 [6.3 2.3 4.4 1.3]
 [5.9 3. 5.1 1.8]]
```

```

[6.4 2.8 5.6 2.1]
[6.5 3.2 5.1 2. ]
[6.2 3.4 5.4 2.3]
[5.7 2.5 5.  2. ]
[6.9 3.1 5.4 2.1]
[7.2 3.  5.8 1.6]
[7.9 3.8 6.4 2. ]
[6.  3.  4.8 1.8]
[6.4 3.2 5.3 2.3]
[6.7 3.  5.2 2.3]
[5.8 2.7 5.1 1.9]
[6.3 2.9 5.6 1.8]]
clf.support_ [16 26 36 59  2  4  6 33 34 37 40 42 54 57 58 60 64 65 66 67
1 11 14 17
19 20 23 41 44 55 56 62 71]
clf.n_support_ [ 4 16 13]

```

Атрибут

- `support_` - индексы опорных векторов
- `support_vectors_` – сами опорные вектора,
- `n_support_` – количество опорных векторов для каждого класса.

15. Построен график зависимости неправильно классифицированных наблюдений и точности классификации от размера тестовой выборки. Размер тестовой выборки изменялся от 0.05 до 0.95 с шагом 0.05. Параметр `random_state` сделан равным номеру своей зачетной книжки - 630408.

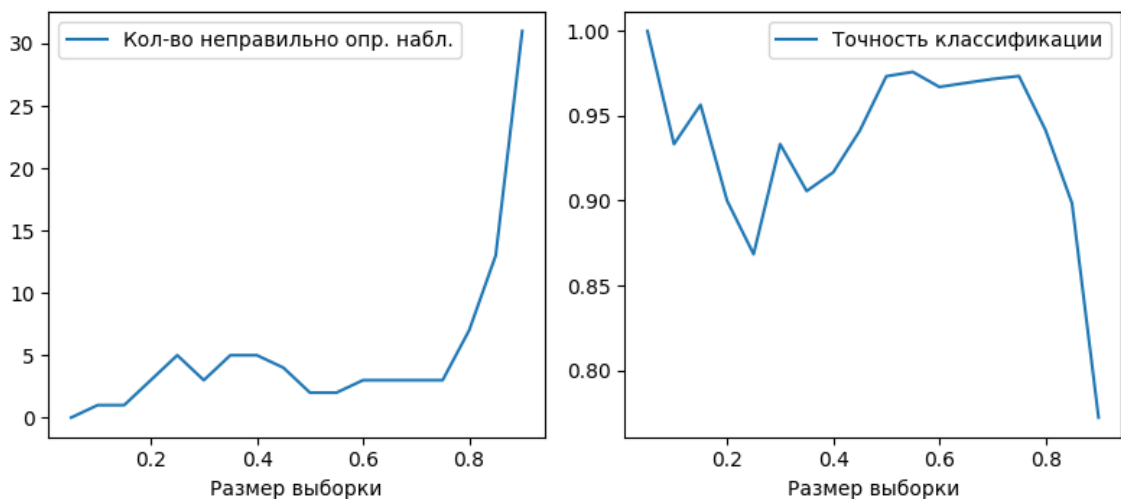


Рисунок 8 — `svm.SVC()`

16. Исследована работа метода опорных векторов при различных значениях kernel, degree, max_iter

kernel	Wrong classified	Score
linear	2	0.973
poly	6	0.953
rbf	4	0.953
sigmoid	54	0.333

degree	Wrong classified	Score
1	5	0.946
2	6	0.96
3	6	0.953
4	5	0.96
5	3	0.97

max_iter	Wrong classified	Score
1	9	0.92
2	8	0.94
3	5	0.95
4	3	0.96
5	1	0.98
6	3	0.96
7	3	0.96
8	4	0.96

17.. Проведено исследование для методов NuSVC и LinearSVC

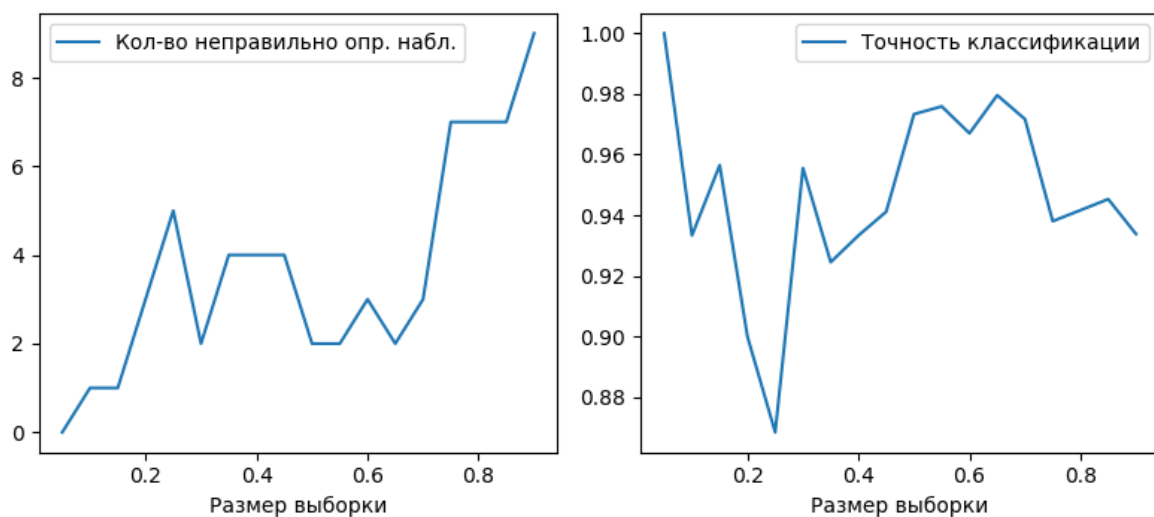


Рисунок 9 — svm.NuSVC()

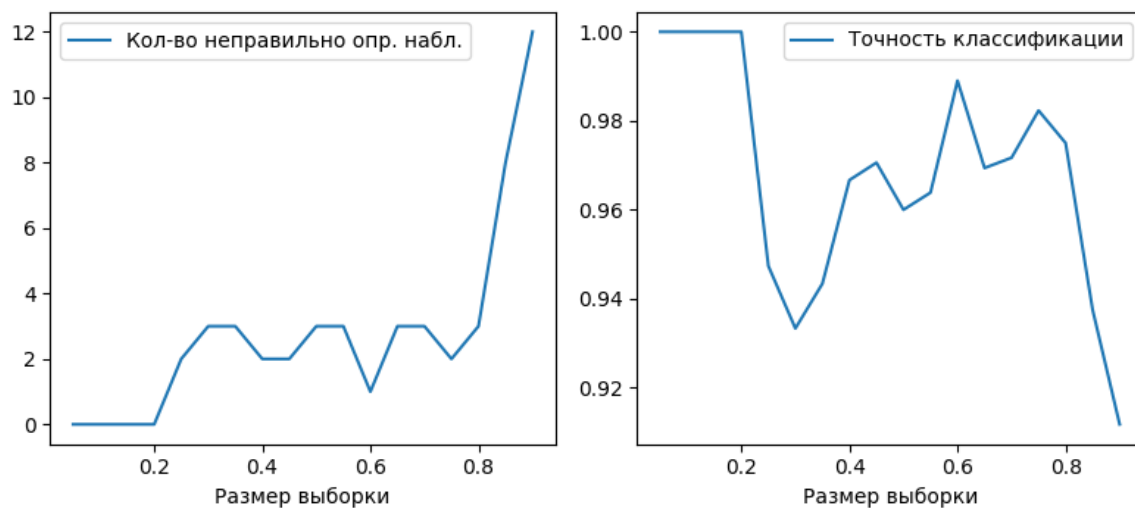


Рисунок 10 — svm.LinearSVC()

- NuSVC подобен SVC, но использует параметр для управления количеством опорных векторов.
- LinearSVC аналогично SVC с линейным ядром, но лучше масштабируется для большого числа выборок.

Вывод

В ходе лабораторной работы рассмотрены такие методы классификации модуля Sklearn, как LinearDiscriminantAnalysis, SVC, NuSVC и LinearSVC.