

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

**ОТЧЕТ**  
**по лабораторной работе №7**  
**по дисциплине «Машинное обучение»**  
**Тема: Классификация (Байесовские методы, деревья)**

Студент гр. 6304

Ковынев М.В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

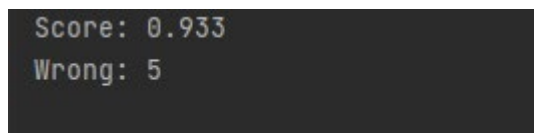
2020

## 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. Преобразованы тексты меток к числам
5. Разбили выборку на обучающую и тестовую
6. Проведена классификация наблюдений наивным байесовским методом



```
Score: 0.933
Wrong: 5
```

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

- `class_count_` — количество обучающих выборок, наблюдаемых в каждом классе
  - `class_prior_` — вероятность каждого класса
  - `classes_` — метки классов, известные классификатору
  - `epsilon_` — Абсолютная аддитивная величина
  - `sigma_` — дисперсия каждого признака по классу
  - `theta_` — среднее каждого признака по классу
7. 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.

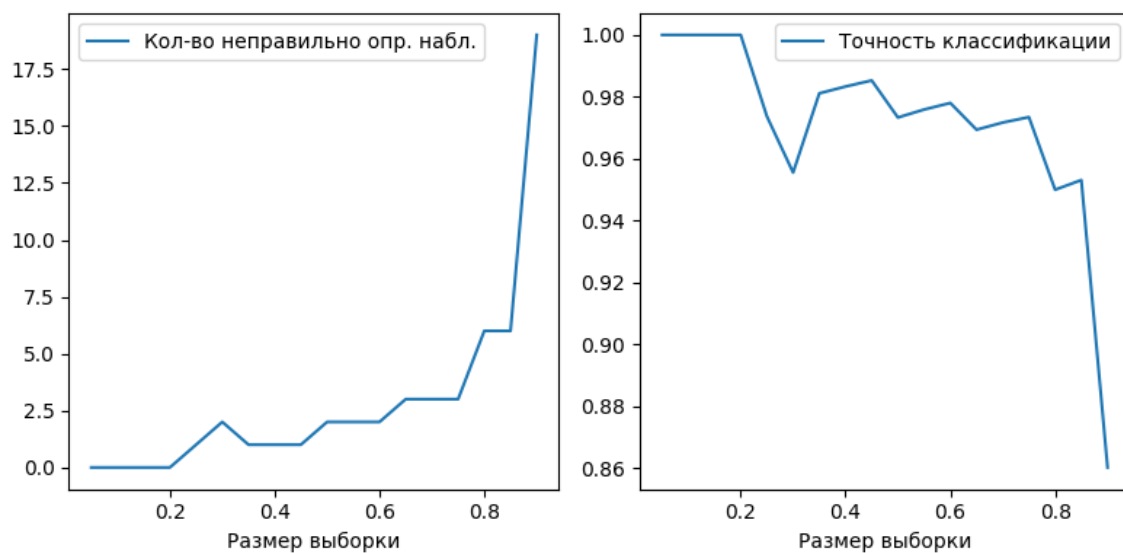


Рисунок 2 — GaussianNB

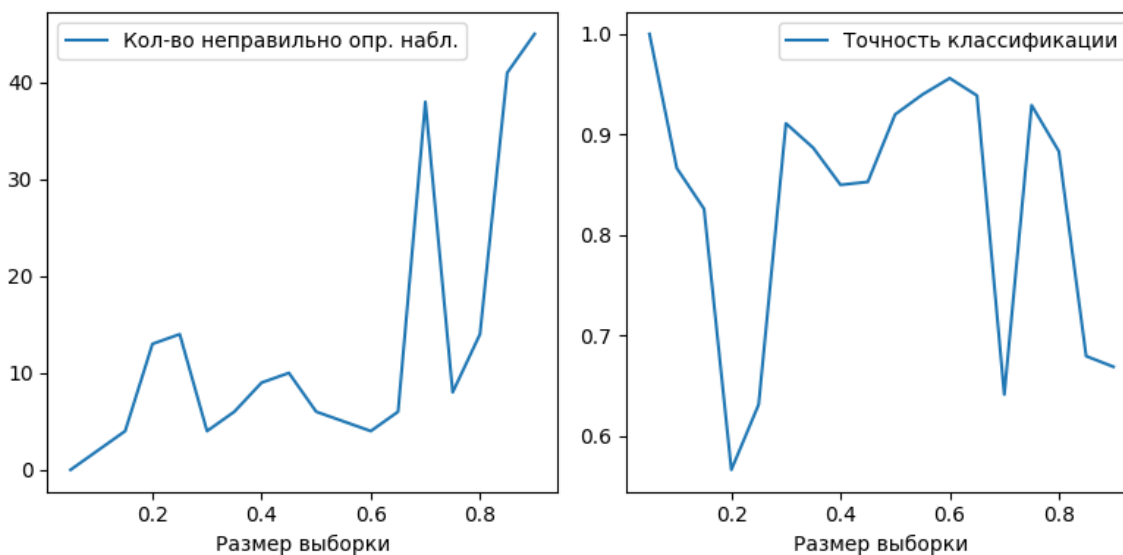


Рисунок 3 — MultinomialNB

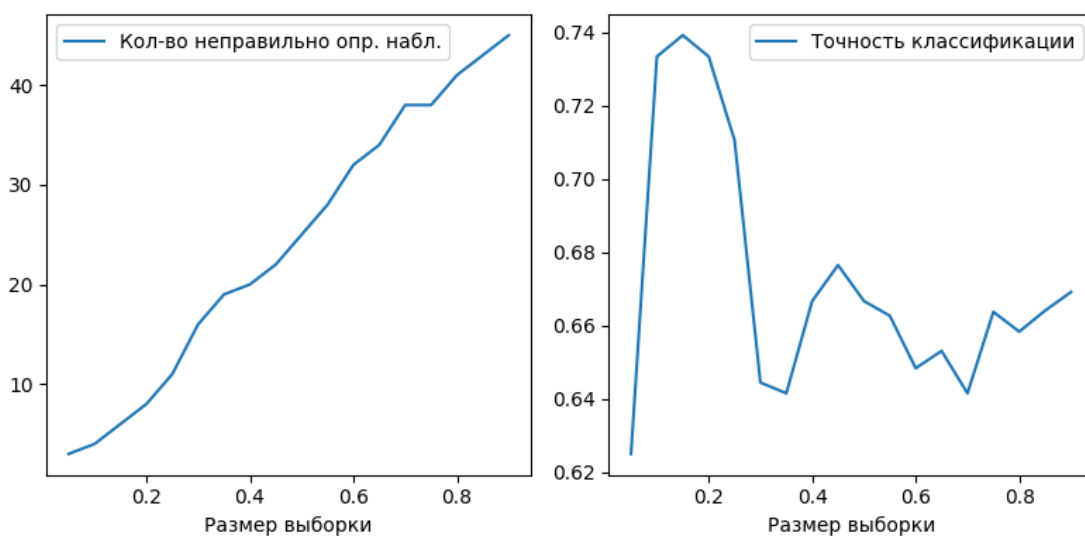


Рисунок 4 — ComplementNB

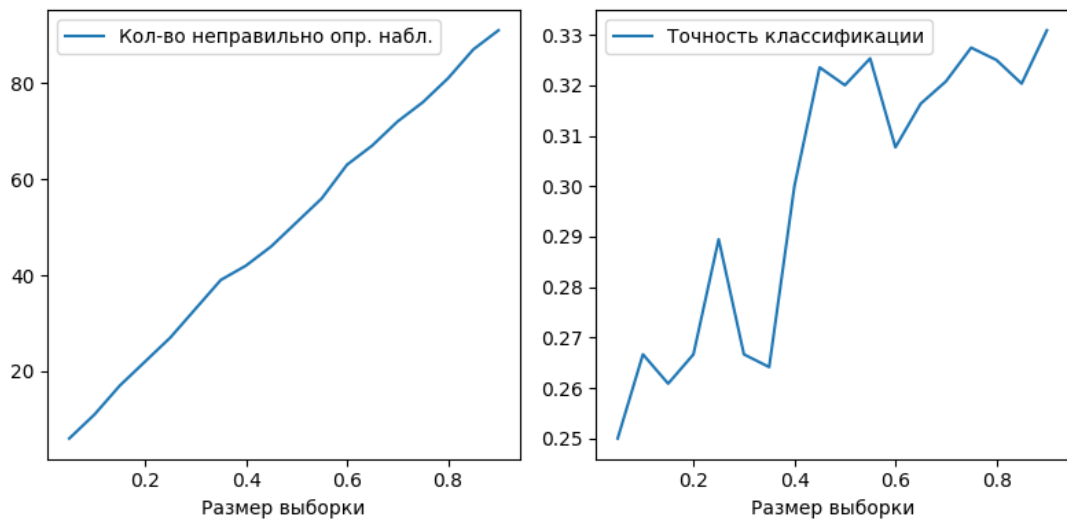


Рисунок 5 — BernoulliNB

**MultinomialNB** — полиномиальный наивный байесовский классификатор, подходит для классификации с дискретными признаками (например, подсчет слов для классификации текста). MultinomialNB реализует наивный алгоритм Байеса для полиномиально распределенных данных. Распределение для каждого класса параметризуется векторами, содержащими вероятности вхождения признаков в элемент выборки, соответствующий данному классу.

**ComplementNB** — адаптация MultinomialNB, подходит для несбалансированных наборов данных. В частности, CNB использует статистику из дополнения каждого класса для вычисления весов модели. ComplementNB часто превосходит MultinomialNB в задачах классификации текста.

**BernoulliNB** — как и MultinomialNB, этот классификатор подходит для дискретных данных. Разница в том, что в то время, как MultinomialNB работает с подсчетом вхождений, BernoulliNB предназначен для двоичных/логических признаков.

8. Классификацию при помощи деревьев на тех же данных.
9. Используя функцию `score()` выведена точность классификации
10. Выведены характеристики дерева, количество листьев и глубину, используя функции `get_n_leaves` и `get_depth`

```
Wrong classified: 6
Score: 0.92
Num of leaves: 6
Depth: 4
```

Рисунок 6 — Точность и количество наблюдений, который были неправильно определены

11. Выведено изображение полученного дерева

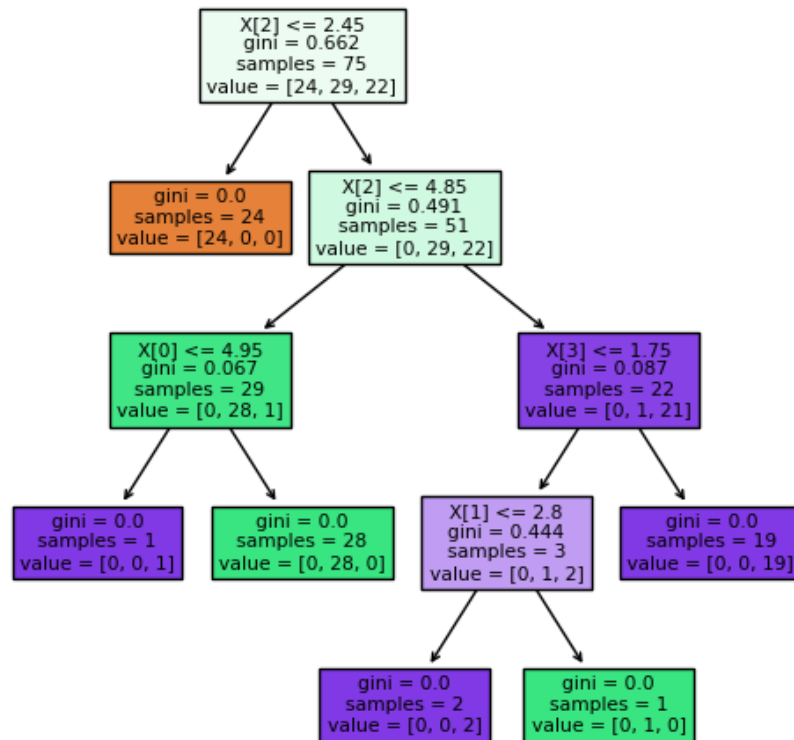


Рисунок 7 — Дерево

12. 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.

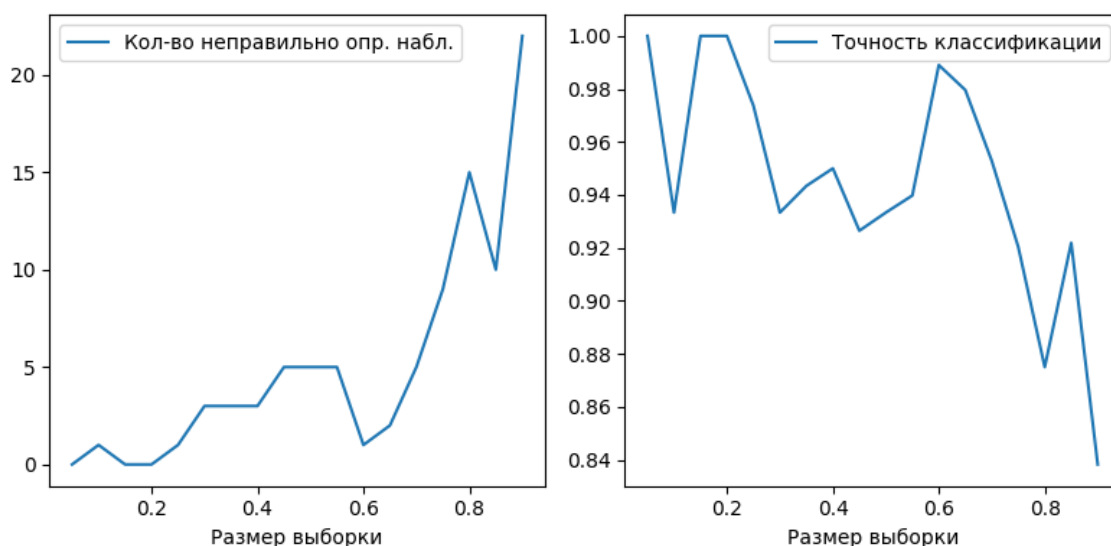


Рисунок 8 — DecisionTreeClassifier

13. Исследована работа классифицирующего дерева при различных параметрах `criterion`, `splitter`, `max_depth`, `min_samples_split`, `min_samples_leaf`
14. `criterion` — функция измерения качества разбиения. Поддерживается индекс Джини и энтропия.

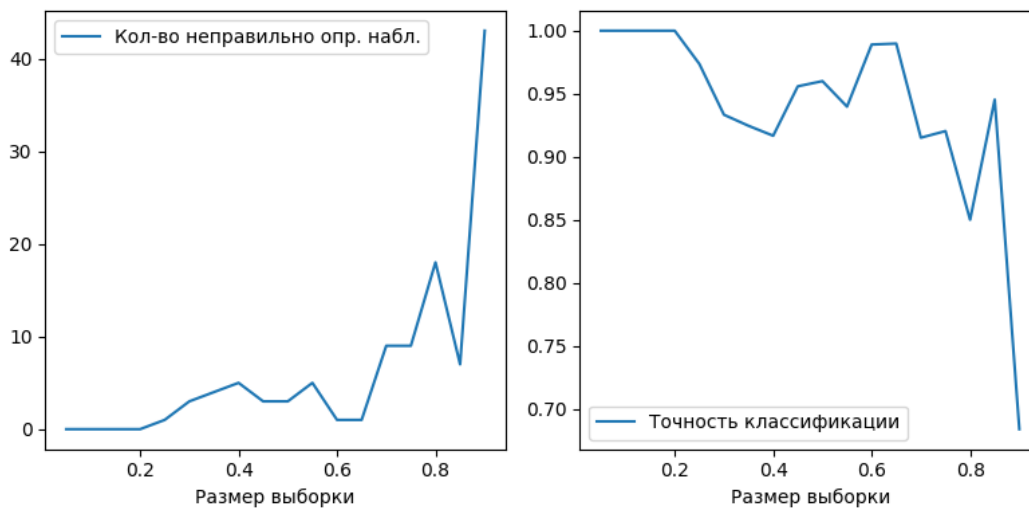


Рисунок 9 — DecisionTreeClassifier(`criterion="entropy"`)

15. `splitter` — стратегия, используемая для выбора разбиения на каждом узле. Поддерживается выбор наилучшего разбиения и случайный выбор

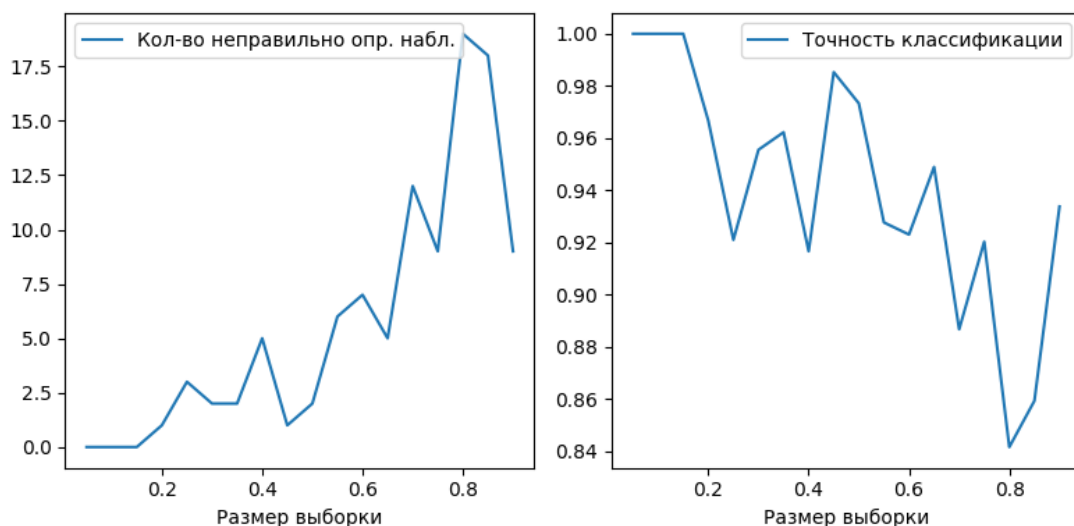


Рисунок 10 — DecisionTreeClassifier(splitter="random")

- `max_depth` — максимальная глубина дерева. Если `None`, то узлы расширяются до тех пор, пока все листья не станут чистыми или пока все листья не будут содержать менее `min_samples_split` выборок

<code>max_depth</code>	Wrong classified	Score
1	27	0.64
2	4	0.947
3	2	0.973
4	3	0.96
5	3	0.96

- `min_samples_split` — минимальное количество выборок, необходимых для разделения внутреннего узла.

<code>min_samples_split</code>	Wrong classified	Score
10	5	0.933
20	5	0.933
30	6	0.92
40	6	0.92

50	29	0.613
60	29	0.613
70	29	0.613
80	53	0.293
90	53	0.293

- `min_samples_leaf` — Минимальное количество выборок, которое требуется для конечного узла. Точка разделения на любой глубине будет учитываться только в том случае, если она оставляет не менее `min_samples_leaf` обучающих выборок в каждой из левой и правой ветвей.

<b>min_samples_leaf</b>	<b>Wrong classified</b>	<b>Score</b>
10	3	0.96
20	3	0.96
30	27	0.64
40	51	0.32
50	51	0.32
60	51	0.32
70	51	0.32
80	51	0.32
90	51	0.32

## Вывод

В ходе лабораторной работы рассмотрены такие методы классификации модуля Sklearn, как GaussianNB, MultinomialNB, ComplementNB, BernoulliNB и DecisionTreeClassifier