МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«Южный федеральный университет»

Институт высоких технологий и пьезотехники

|  |  |
| --- | --- |
| Описание: https://pp.vk.me/c623122/v623122412/1acb7/yTJRs3eyFiE.jpg | Кафедра прикладной информатики и инноватики  Направление подготовки: 09.03.03 Прикладная информатика |

**Отчет на тему «Лабораторная работа №1: Метод Наивный Байеса»**

**По дисциплине «Основы машинного обучения»**

**Выполнил:**

студент 3 курса бакалавриата 6 группы

Кузин М.А.

**Проверил:**

доцент, к. ф-м. н.

Ефимов А.И.

Ростов-на-Дону

2022

**Цели и задачи работы**

Изучить принцип работы метода машинной инженерии Наивного Байеса для задач классификации. Реализовать на практике данный метод с помощью стандартных методов, библиотеки scikit-learn для Toy’s datasets классификации, создать с помощью pandas и numpy свою реализацию Наивного Байеса для заданных датасетов. Также для Toy’s datasets, написать программу, которая реализует ROC-кривую по данным обучения.

**Использованные инструменты**

Python 3.9

Библиотеки: Sklearn(datasets, KNeighborsClassifier и т.д.), numpy, matplotlib(pyplot)

**Краткая теория**

**Наи́вный ба́йесовский классифика́тор** — простой вероятностный классификатор, основанный на применении теоремы Байеса со строгими (наивными) предположениями о независимости.

Все параметры модели могут быть аппроксимированы относительными частотами из набора данных обучения. Это оценки максимального правдоподобия вероятностей. Непрерывные свойства, как правило, оцениваются через нормальное распределение. В качестве математического ожидания и дисперсии вычисляются статистики — среднее арифметическое и среднеквадратическое отклонение соответственно.

Если данный класс и значение свойства никогда не встречаются вместе в наборе обучения, тогда оценка, основанная на вероятностях, будет равна нулю. Это проблема, так как при перемножении нулевая оценка приведет к потере информации о других вероятностях. Поэтому предпочтительно проводить небольшие поправки во все оценки вероятностей так, чтобы никакая вероятность не была строго равна нулю.

*ROC-кривая*

Кривые ROC обычно показывают истинную положительную скорость по оси Y и ложноположительную скорость по оси X. Это означает, что верхний левый угол графика является “идеальной” точкой - ложноположительная скорость равна нулю, а истинная положительная скорость равна единице. Это не очень реалистично, но это означает, что большая площадь под кривой (AUC) обычно лучше.

“Крутизна” ROC-кривых также важна, поскольку идеально максимизировать истинную положительную скорость при минимизации ложноположительной скорости.

ROC-кривые обычно используются в двоичной классификации для изучения выходных данных классификатора. Чтобы расширить кривую ROC и область ROC до классификации с несколькими метками, необходимо выполнить бинаризацию выходных данных. На каждую метку можно нарисовать одну ROC-кривую, но можно также нарисовать ROC-кривую, рассматривая каждый элемент матрицы индикаторов меток как двоичное предсказание (микроосреднение).

Другой мерой оценки классификации с несколькими метками является макросреднение, которое придает одинаковый вес классификации каждой метки.

**Ход работы**

*Часть 1. Реализация метода Наивного Байеса с помощью Numpy и Pandas*

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - 1. Импорт нужных библиотек.

Для реализации данного метода классификации удобно использовать ООП. (Рис. 2)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - 2. Реализация метода Наивного Байеса (Метод обучения).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - 3. Реализация метода Наивного Байеса (Методы подсчета процентов правильности).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - 4. Реализация метода Наивного Байеса (Кроссвалидация и дальнейшее обучение).

*Часть 2. Реализация метода Наивного Байеса с помощью библиотеки scikit-learn*

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - 5. Реализация метода Наивного Байеса с помощью библиотеки scikit-learn.

В библиотеке scikit-learn уже реализованы все необходимые методы

*Часть 3. ROC-кривая*

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - 6. Импорт библиотек и реализация функции для вывода ROC-кривой.

Для вывода графика ROC-кривой используем библиотеку matplotlib(pyplot). (рис.5)

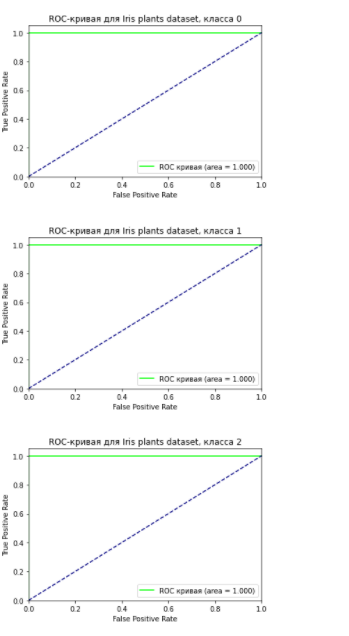


Рисунок - 7. Вывод ROC-кривых.

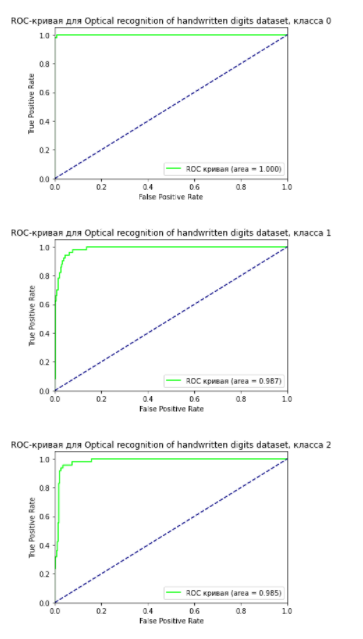


Рисунок - 8. Вывод ROC-кривых.

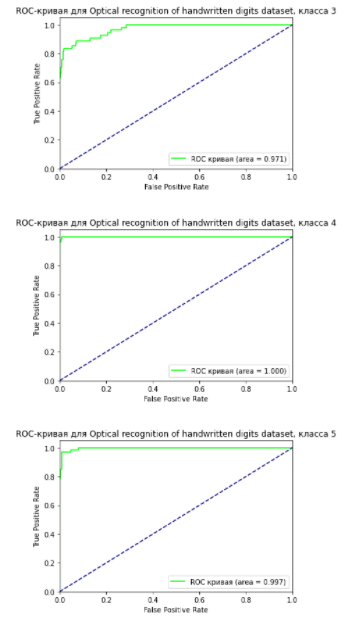


Рисунок - 9. Вывод ROC-кривых.

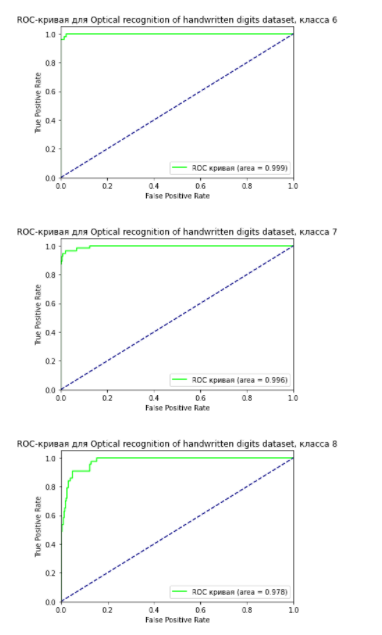


Рисунок - 10. Вывод ROC-кривых.

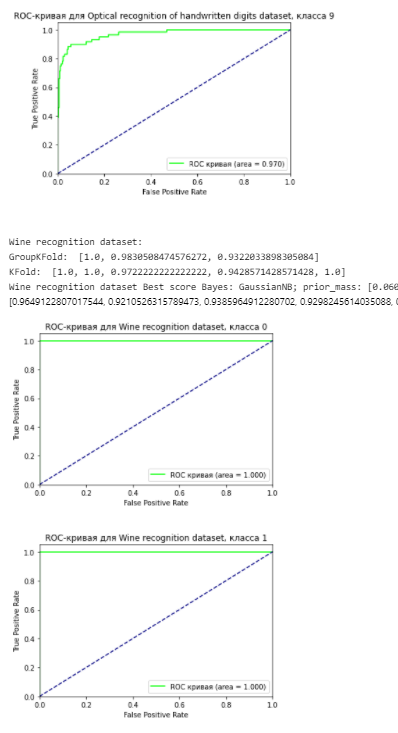


Рисунок - 11. Вывод ROC-кривых.

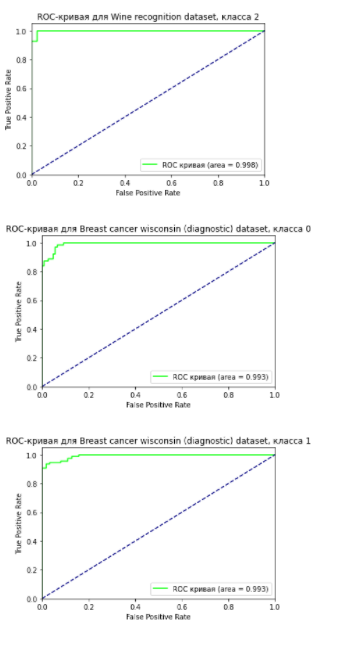


Рисунок - 12. Вывод ROC-кривых.

**Вывод**

Изучили принцип работы метода машинной инженерии Наивного Байеса для задач классификации. Реализовали на практике данный метод с помощью стандартных функций, библиотеки scikit-learn для Toy’s datasets классификации. Также для этих датасетов, написали программу, которая реализует ROC-кривую по данным обучения.