

# Практическая работа №12: «Ансамбли».

## Оглавление

Цель работы .....	1
Задачи работы .....	1
Перечень обеспечивающих средств.....	1
Общие теоретические сведения .....	1
Описание метода.....	1
Бэггинг .....	2
Бустинг.....	3
Задание .....	5
Требования к отчету .....	5
Литература .....	5

## ***Цель работы***

Получить практические навыки решения задач регрессии и классификации с помощью различных типов ансамблей.

## ***Задачи работы***

1. Сравнить несколько моделей для решения задачи регрессии с помощью ансамблей.
2. Сравнить несколько моделей для решения задачи классификации с помощью ансамблей.

## ***Перечень обеспечивающих средств***

1. ПК.
2. Учебно-методическая литература.
3. Задания для самостоятельного выполнения.

## ***Общие теоретические сведения***

### Описание метода

Ансамбль – это модель машинного обучения, которая включает в себя набор более «слабых» моделей.

Задача, которую отдельные «слабые» модели решают плохо, т.е. с низкими значениями метрики производительности, в совокупности ансамбль решает хорошо.

#### Бэггинг

Параллельное обучение нескольких «слабых» моделей и агрегация полученных от них результатов.

Бутстрэп:

Набор данных:  $\{X_1, X_2, \dots, X_N\}$ .

Из набора данных формируется  $m$  бутстрэп-выборок, каждая длиной  $n$ .

Элементы выбираются случайным образом, с повторениями.

Основная идея: сделать выборки, а значит и модели, построенные на них как можно более различными.

Алгоритм бэггинга:

1. Выбираем алгоритм для построения «слабых» моделей.
2. Из имеющегося набора данных генерируем несколько бутстрэп-выборок.
3. На каждой из получившихся выборок строим «слабую» модель.
4. Результаты работы полученных моделей агрегируем.

Все «слабые» модели обучаются независимо, т.е. обучение можно проводить параллельно.

Случайный лес – это реализация бэггинга, когда в качестве «слабых» моделей используются деревья принятия решений.

Т.к. бэггинг предполагает, что «слабые» модели имеют большой разброс, но малое смещение, деревья для леса обычно строят без отсечения ветвей.

Чтобы избежать переобучения, к которому склонны деревья принятия решений, при построении случайного леса делается дополнительный шаг – для обучения модели используются не все параметры, представленные в наборе данных, а только некоторое их подмножество.

Обычно, для каждого дерева случайным образом отбирается некоторое заранее выбранное число параметров (одинаковое для всех деревьев).

#### Бустинг

Последовательное обучение «слабых» моделей таким образом, чтобы каждая следующая модель старалась научиться на той части данных, на которой ошибалась предыдущая.

Алгоритм:

1. Выбираем алгоритм для построения «слабых» моделей.
2. Устанавливаем одинаковую «сложность» для всех элементов набора данных.
3. Обучаем «слабую» модель на наборе данных с учётом «сложности» элементов.
4. Определяем, на каких элементах модель ошибается
5. Вычисляем новые значения «сложности» для всех элементов набора данных.
6. Если критерий остановки не достигнут, возвращаемся к шагу 3.

«Слабые» модели обучаются последовательно, поэтому полезно выбирать алгоритмы с низкой вычислительной сложностью.

#### Градиентный бустинг

На каждом шаге мы обучаем очередную «слабую» модель в сторону, противоположную градиенту текущей ошибки по отношению к текущей модели.

Если  $(X_i, y_i)$  - набор данных и  $e(y_i, \hat{y}_i)$  – функция ошибки, то

$$r_{Ni} = - \left[ \frac{\partial e(y_i, M(x_i))}{\partial M(x_i)} \right]_{M(x)=M_{N-1}(x)} \quad \text{— псевдо-остатки.}$$

«Слабая» модель  $m_N$  обучается на синтетическом наборе данных  $\{X_i, r_{Ni}\}$ .

$$M_N = M_{N-1} + a_N m_N$$

$a_N$  подбирается так, чтобы значение ошибки было минимально:

$$a_N = \arg \min_a \sum_i e(y_i, M_{N-1}(X_i) + a m_N(X_i))$$

Алгоритм:

1. Установить псевдо-остатки равными элементам набора данных.
2. Обучить наилучшую возможную «слабую» модель на псевдо-остатках.
3. Вычислить значение коэффициента обновления, который показывает, насколько должен быть учтен вклад «слабой» модели.
4. Обновить общую модель, добавив новую «слабую» модель, умноженную на её коэффициент обновления.
5. Вычислить новые псевдо-остатки, которые показывают, в каком направлении мы хотели бы обновить прогнозы модели на следующем шаге.

Пункты 2-5 повторяются столько раз, сколько «слабых» моделей мы хотим использовать.

## **Задание**

### **Пояснение**

Для сохранения результатов данной работы вам понадобится файл `іруnb`. Если требуется, для удобства можно создать также второй файл формата `doc/docx`. Названия файла или файлов должны иметь вид «*Фамилия – задание 12*».

### **Часть 1**

- Обновите свой репозиторий, созданный в практической работе №1, из оригинального репозитория:  
[https://github.com/mosalov/Notebook\\_For\\_AI\\_Main](https://github.com/mosalov/Notebook_For_AI_Main).

### **Часть 2**

- Откройте свой репозиторий в Binder (<https://mybinder.org/>).
- Откройте файл «2022 Весенний семестр\task2.іруnb».
- Изучите, при необходимости – выполните повторно, приведённый в файле код.
- По аналогии с изученным выполните два задания, приведённых в ячейках в конце ноутбука.
- Сохраните код в `іруnb`-файле. При необходимости пояснения опишите в `doc/docx`-файле.

## **Требования к отчету**

Готовые файлы загрузите в свой репозиторий, созданный в практическом задании №1 по пути: «Notebook\_For\_AI\_Main/2022 Весенний семестр/Практическое задание 12/», и сделайте пул-реквест.

## **Литература**

1. <https://neurohive.io/ru/osnovy-data-science/ansamblevye-metody-begging-busting-i-steking/>
2. <https://dyakonov.org/2016/11/14/случайный-лес-random-forest/>
3. <https://habr.com/ru/company/ods/blog/327250/>