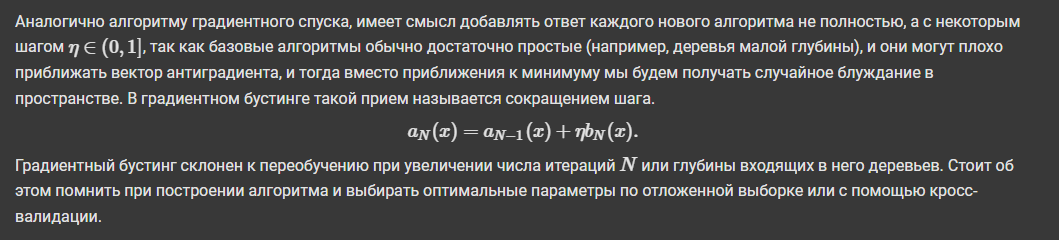
Для чего в алгоритме нужен шаг:



DecisionTreeRegressor – библиотечная реализация дерева классификации  
Особенностью градиентного бустинга является то, что мы обучаем следующее дерево с учетом ошибок предыдущего

Бэггинг – когда создаются подвыборки той же длины, что и исходная.

Метод случайных подпространств – насколько я понял, когда исходные характеристики разбиваются на несколько групп и потом над каждой группой работает определенный алгоритм

Бэггинг == бутстрап

Для уменьшения корреляции базовых алгоритмов рандомизируют сам процесс построения каждого дерева: если в стандартном методе построения деревьев мы в каждом узле выбираем j-й признак и порог t, с которым сравнивается его значение, и потом эти значения оптимизируются с помощью функции ошибки, то в методе случайного леса в каждой вершине j-й признак выбирается не из всего пространства признаков, а из его случайного подмножества размера m, **которое каждый раз выбирается заново** (в этом отличие от метода случайных подпространств, где подпространство выбирается единожды и используется для построения всего дерева).

При обучении при увеличении глубины деревьев сначала ошибка на тестовых данных уменьшалась, затем возросла из-за переобучения

В стохастическом градиентном бустинге, как я понял, отличается только то, что мы берем не целиком выборку, а берем такого же размера из изначальной, но дублируя случайные объекты, например:

[a, b, c, d, e] -> [a, a, b, b, e]  
Градиентный бустинг склонен к переобучению при увеличении числа итераций N или глубины входящих в него деревьев. Стоит об этом помнить при построении алгоритма и выбирать оптимальные параметры по отложенной выборке или с помощью кросс-валидации.

Вся инфа в lesson10 и lesson11