Московский Государственный Университет имени М.В.Ломоносова
Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

# Практикум осень 2022
# Ансамбли алгоритмов. Веб-сервер. Композиции алгоритмов для решения задачи регрессии.
# Отчет Ensembles

Авдеев Роман Артемович
Группа 317

Декабрь, 2022

# Оглавление

# 1 Введение

В данной работе была поставлена задача:

1. написать на языке Python собственную реализацию методов случайный лес и градиентный бустинг
2. провести серию экспериментов с выданными данными
3. написать реализацию веб-сервера с требуемой функциональностью, обернуть своё решение в docker контейнер
4. разместить весь написанный код в приватном репозитории

В моем отчете освещены первые два пункта из перечисленных выше.

Несколько слов о реализованных алгоритмах:

Алгоритм случайного леса (Random Forest) — универсальный алгоритм машинного обучения, суть которого состоит в использовании ансамбля решающих деревьев. Само по себе решающее дерево предоставляет крайне невысокое качество классификации, но из-за большого их количества результат значительно улучшается (бэггинг). Также данный алгоритм отличается широким спектром задач, где его можно использовать.

Градиентный бустинг (над решающими деревьями) - алогоритм машинного обучения, реализующий идею независимого построения алгоритмов, а далее их последовательного применения. Каждый следующий алгоритм старается уменьшить ошибку текущего ансамбля.

# 2 Эксперимент 1(предобработка данных)

Обучение и тестирвание реализованных алгоритмов проводились на данных о продажах недвижимости House Sales in King County, USA.

В этом датасете находится 21613 объектов с 21 признаком. Для начала был проведен анализ информативности каждого признака, удален столбец 'id', т.к. у подавляющего большинства объектов идентификационный номер свой, следовательно, никакой пользы при оценке стоимости недвижимости эта информация не принесет.

Целевой переменной здесь является столбец 'price'.

Далее были выбраны категориальные и вещественные признаки. Рассматривался каждый признак, а также всевозможные значения и тип данного столбца. В итоге в качестве категориальных были выбраны следующие столбцы: 'date', 'floors', 'waterfront', 'view', 'condition', 'grade', `'yr_built'`, `'yr_renovated'`, 'zipcode' - 9 признаков.

У всех вышеперечисленных признаков небольшая размерность множества значений. Опишем подробнее каждый признак:

1. столбцы 'date', `'yr_built'`, `'yr_renovated'` отвечают за принадлежность к какой-то дате(году), поэтому действительно могут трактоваться как категориальные
2. множество значений признака 'floors' равно `{1.0, 2.0, 1.5, 3.0, 2.5, 3.5}` - значения расположены в порядке убывания по частоте встречаемости. Здесь присутствуют вещественные значения, но размерность множества невелика, а также данный признак позволяет отнести дом/квартиру к определенному классу "этажности" => категориальный
3. 'waterfront' принимает только два значения: 0 и 1. Очевидно, категориальный
4. 'view', 'condition', 'grade' принимают целочисленные значения (максимальная размерность множества значений среди данных столбцов = 116). Также позволяют отнести недвижимость к определенному классу (например, некоторый тип вида для столбца 'view') => считаем перечисленные признаки категориальными

Оставшиеся 10 признаков ('id' удален, 'price' - целевая переменная => всего 19 признаков) считаем вещественными, т.к. принимают много различных вещественных значений.

Заметим, что нет столбцов с пропущенными (NaN) значениями, поэтому нет необходимости применять SimpleImputer.

Далее разделил данные на X и y, где y это столбец 'price', также перевел в numpy.ndarray.

Затем закодировал категориальные признаки с помощью счетчиков со сглаживанием(TargetEncoder). Из четвертого задания ММРО помним, что счетчики показали себя лучше, чем OHE (в моем случае), поэтому выбор пал именно на TargetEncoder. Для вещественных признаков был использован StandartScaler.

На последнем этапе обработки данных было выполнено разделение датасета на обучающую и тестовую выборки в отношении 7:3.

# 3 Эксперимент 2(случайный лес)

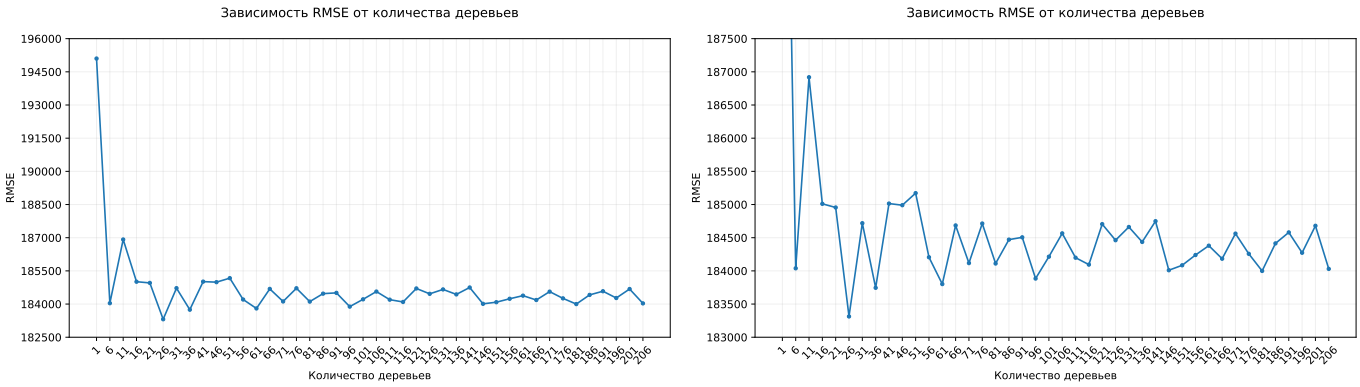
В данном эксперименте исследовались зависимости RMSE и времени работы алгоритма от следующих факторов:

1. количество деревьев в ансамбле
2. размерность подвыборки признаков для одного дерева
3. максимальная глубина дерева (+ случай, когда глубина неограничена)

## 3.1 Количество деревьев в ансамбле

Рассматривалось количество деревьев(`n_estimators`) от 1 до 210 с шагом 5.

Оценка RMSE (правый график - значения RMSE до 187000):




Минимальная ошибка (183313) достигается при `n_estimators`=26

Оценка времени работы алгоритма

Заметим, что зависимость очень близка к линейной:


## 3.2 Размерность подвыборки

Рассматривались подвыборки размером от 1 до 20 с шагом 1.
Оценка RMSE:

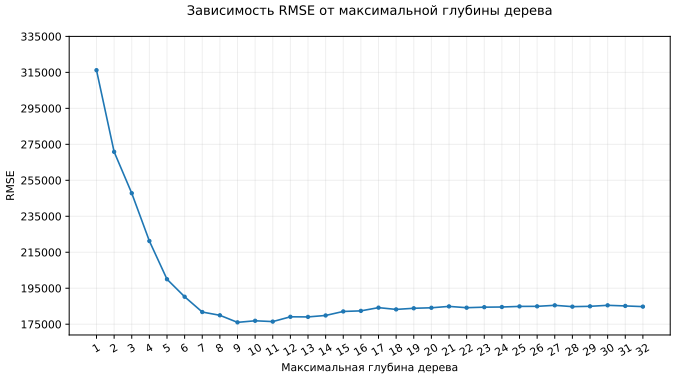

Минимальная ошибка (184003) достигается при размере подвыборки=14

Оценка времени работы алгоритма:


Быстрее всего алгоритм работает, когда размерность подвыборки=2

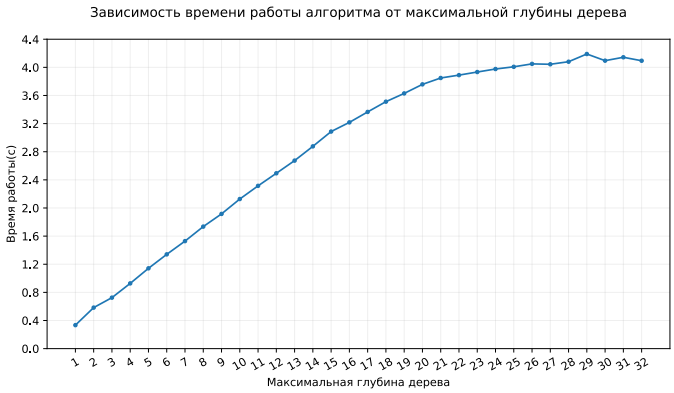
## 3.3 Максимальная глубина

Рассматривались значения `max_depth` от 1 до 32 с шагом 1.
Оценка RMSE:




Заметим, что график очень похож на график обратной экспоненты.
Минимальная ошибка (176024) достигается при `max_depth`=9

Оценка времени работы алгоритма:




При значениях от 1 до 21 график также близок к линейному, но далее стремится к $y = ln(x)$

Случай, когда глубина неограничена: RMSE=185341, время работы алгоритма=4.04(с)

# 4 Эксперимент 3(градиентный бустинг)

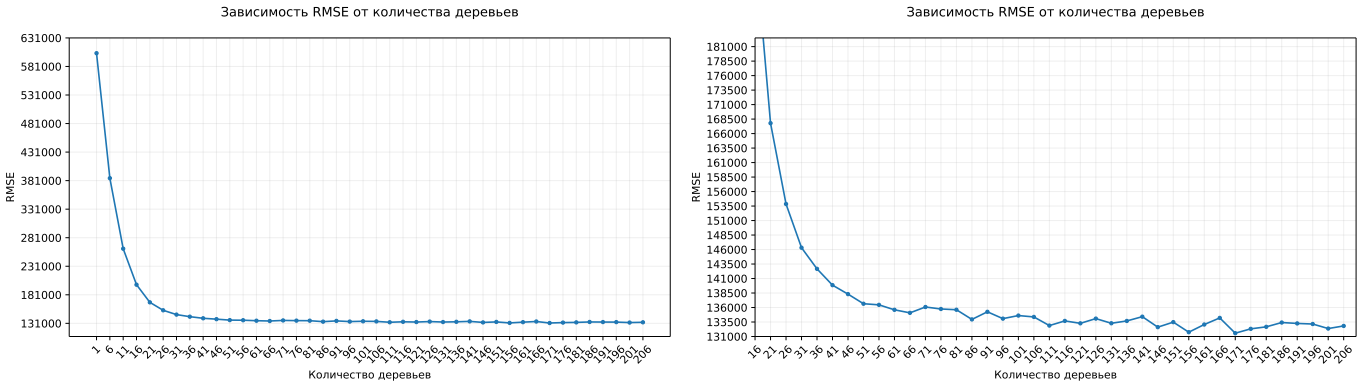
В данном эксперименте исследовались зависимости RMSE и времени работы алгоритма от следующих факторов:

1. количество деревьев в ансамбле
2. размерность подвыборки признаков для одного дерева
3. максимальная глубина дерева (+ случай, когда глубина неограничена)
4. выбранный learning rate

## 4.1 Количество деревьев в ансамбле

Рассматривалось количество деревьев(`n_estimators`) от 1 до 210 с шагом 5.

Оценка RMSE:




(Правый график - более детальное рассмотрение для RMSE $< 181000$)
Минимальная ошибка (131571) достигается при `n_estimators`=171

Общий характер зависимости RMSE от количества деревьев близок к обратно экспоненциальной.

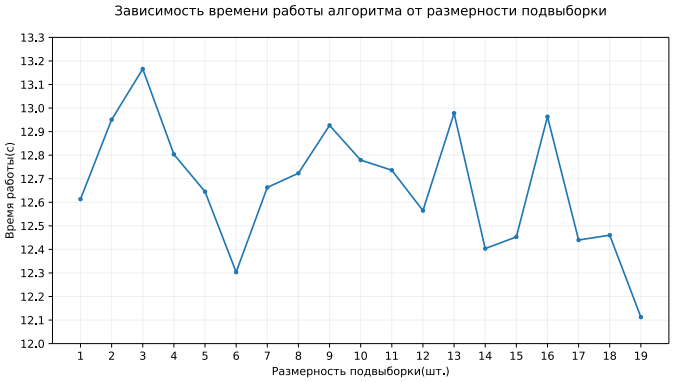
Оценка времени работы алгоритма(зависимость близка к линейной):


Заметим, что при `n_estimators` $\geq 86$ уменьшение RMSE незначительное. При этом время растет почти линейно при увеличении количества деревьев. Это значит, что в условиях реальной разработки в компании надо будет искать компромисс между пренебрежением некой долей качества и временем работы. В своем же решении я ориентируюсь на наивысшее качество (наименьшую ошибку).

## 4.2 Размерность подвыборки

Рассматривались подвыборки размером от 1 до 20 с шагом 1.

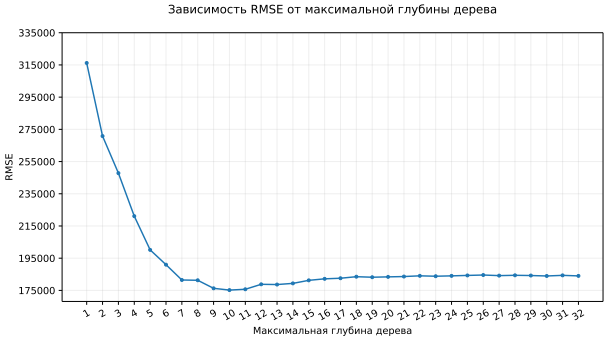
Оценка RMSE и времени работы:


Минимальная ошибка (131907) достигается при размере подвыборки=16; быстрее всего алгоритм работает, когда размерность подвыборки=19.

## 4.3 Максимальная глубина

Рассматривались значения `max_depth` от 1 до 32 с шагом 1.
Оценка RMSE и времени работы:





Заметим, что график очень похож на график обратной экспоненты.
Минимальная ошибка (175191) достигается при `max_depth`=10

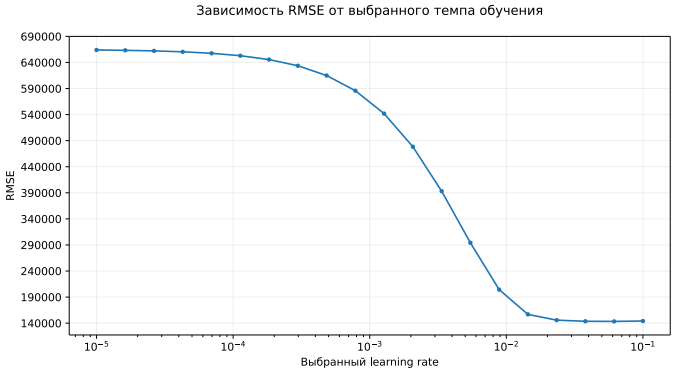
При значениях от 1 до 19 график также близок к линейному, но далее стремится к $y = ln(x)$

Случай, когда глубина неограничена: RMSE=184425, время работы алгоритма=4.18(с)

Видим, что оставлять глубину неограниченной не стоит. Минимальная ошибка достигается при `max_depth`=10. При таком параметре алгоритм работает около 21(с). Это примерно в 5 раз дольше, чем при неограниченной глубине, но зато мы получаем существенное увеличение качества.

## 4.4 Learning rate

Рассматривались значения `learning_rate` из множества $np.logspace(-5, -1, 20)$.
Оценка RMSE и времени работы:





Минимальная ошибка (143371) достигается при `learning_rate`=0.062

# 5 Общие выводы

В ходе работы над данным заданием был проанализирован предложенный датасет по продажи недвижимости, реализованы методы случайного леса(бэггинг) и градиентного бустинга. На предобработанных данных были получены лучшие комбинации гиперпараметров для каждого из алгоритмов, изучены зависимости ошибки RMSE, времени работы алгоритмов от количества деревьев, размерности подвыборки, глубины дерева и темпа обучения.