Отчет о практическом задании «Ансамбли алгоритмов для решения задачи регрессии. Веб-сервер».

Практикум 317 группы, ММП ВМК МГУ.

Булкин Антон Павлович.

Декабрь 2024.

Содержание

1 Вступление

Данное задание посвящено реализации методов ансамблирования и решению задачи регрессии на основе данных о продажах недвижимости.

Задачами задания являлись:

1. Реализовать алгоритмы случайного леса и градиентного бустинга с использованием стандартных библиотек Python (*numpy*, *scipy*, *matplotlib*) и *DecisionTreeRegressor* из библиотеки *scikit-learn*.

2. Провести анализ работы алгоритмов, включая их параметры, такие как:

   - количество деревьев в ансамбле,
   - размерность подвыборки признаков для одной вершины дерева,
   - максимальная глубина деревьев
   - влияние параметра *learning_rate*.

3. Провести предобработку данных и исследовать их влияние на работу моделей.

4. Оценить алгоритмы по метрике RMSE, исследуя зависимость качества работы от изменения параметров.

2 Обработка данных

2.1 Анализ датасета

2.1.1 Характеристика датасета

Датасет `House Sales in King County, USA` содержит информацию о продажах недвижимости в округе Кинг (штат Вашингтон, США). Основные характеристики датасета:

- Датасет состоит из 21 613 записей и 21 признака.
- Столбец `price` является целевой переменной, представляющей стоимость объекта недвижимости.
- Остальные столбцы описывают различные характеристики объектов: местоположение, параметры строения, год постройки, площадь, количество спален, ванных комнат и т.д.
- В данных отсутствуют пропуски.
- Столбец `price` представляет собой дату в текстовом формате.
- Столбец `id` является ID объекта недвижимости.

2.1.2 Расположение объектов недвижимости

Рассмотрим расположение объектов недвижимости. (Рис. 1)

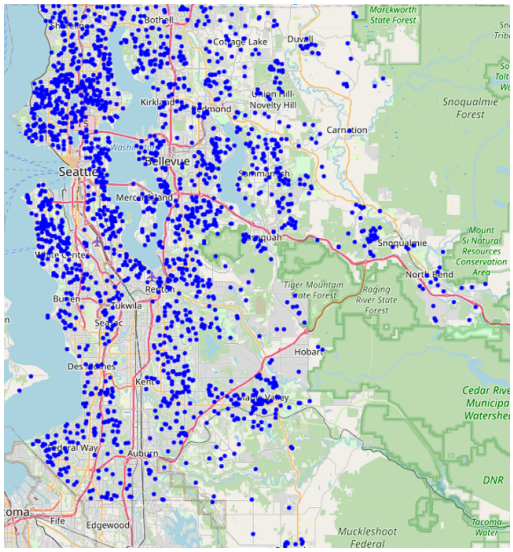

Рис. 1: Расположение случайных 2000 объектов недвижимости

Большая концентрация домов наблюдается в районе города Сиэтл и его пригородов, таких как Bellevue, Redmond, Kirkland, а также вдоль западного побережья озера Вашингтон. Менее плотная застройка наблюдается в восточной части карты, где расположены более лесные и природные зоны. Застройка вдоль главных автомагистралей и дорог указывает на зависимость доступности и плотности домов от транспортной инфраструктуры. Большинство точек сосредоточены в городской черте и прилегающих пригородах. Территории, покрытые лесами, горами, имеют низкую плотность или вовсе не включают дома. Видно, что дома расположены равномерно вдоль побережья, где вероятна высокая стоимость жилья из-за близости к воде.

2.1.3 Корреляция признаков и целевой переменной

Рассмотрим корреляционную матрицу признаков. (Рис. 2)

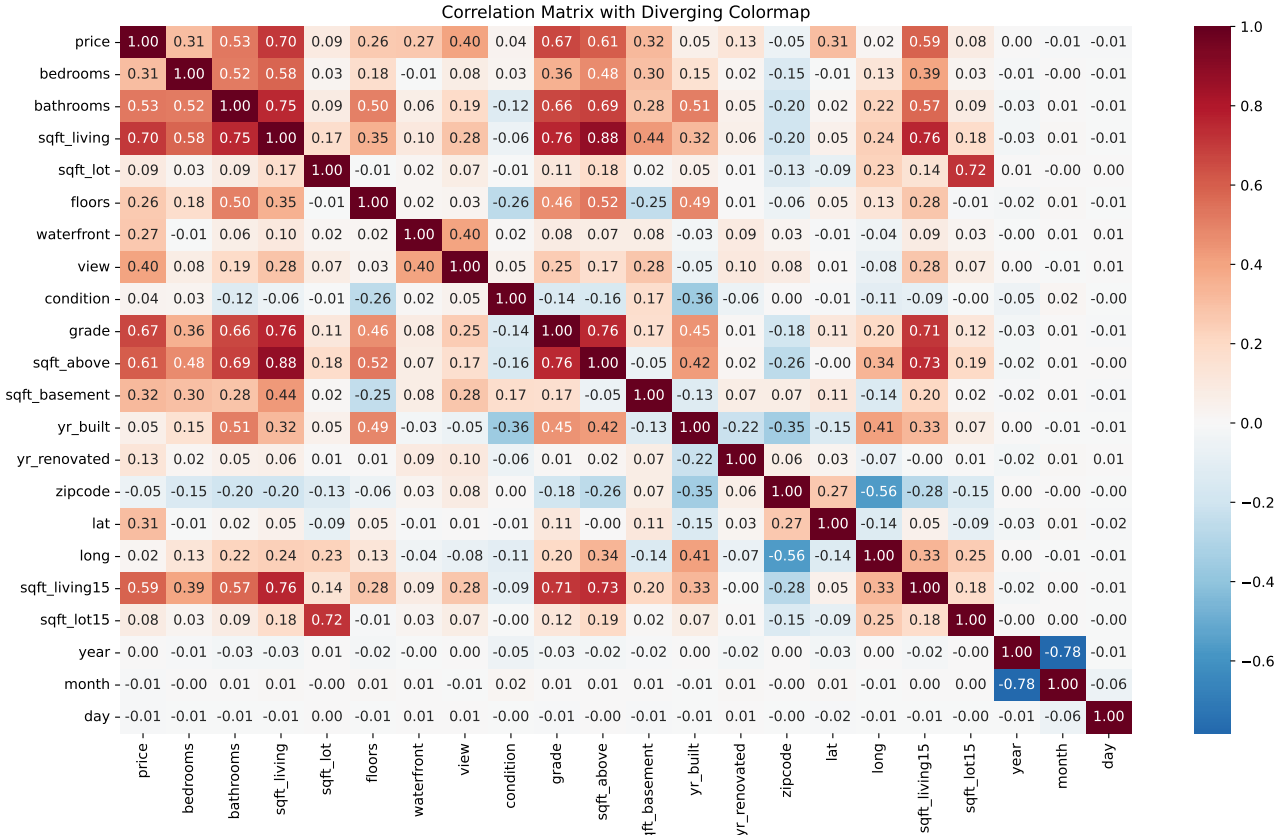

Рис. 2: Корреляционная матрица признаков и target

На цену дома *price* больше всего влияют жилая площадь *sqft_living*, качество строительства *grade* и количество ванных комнат *bathrooms*. Вид на окрестности и площадь соседних домов также имеют умеренное влияние. Связь между разме-

ром дома, его качеством и жилой площадью высокая, что может вызвать мультиколлинеарность. Признаки, такие как *condition*, *yr_renovated* и *zipcode*, имеют отрицательную цорреляцию с ценой.

Если признак имеет отрицательную корреляцию с целевой переменной, его можно оставить при обучении модели RandomForest. Важность признаков (feature importance) в RandomForest зависит от их вклада в уменьшение ошибки разделения, а не от направления корреляции. Отрицательная корреляция означает, что признак может быть полезен для предсказания, только в противоположном направлении. Деревья решений эффективно обрабатывают такие признаки. Они работают с пороговыми значениями и могут использовать признак с отрицательной корреляцией для разделения данных. Также, признак с отрицательной корреляцией может быть важным в сочетании с другими признаками, даже если его вклад по отдельности небольшой.

2.1.4 Дисперсии признаков

Рассмотрим величины дисперсии признаков. (Табл. 1)

Признак	Величина дисперсии
bedrooms	0.8650150097573724
bathrooms	0.5931512887355798
sqft_living	843533.6813681519
sqft_lot	1715658774.1754544
floors	0.29158800687709074
waterfront	0.007485225502689098
view	0.5872426169774596
condition	0.42346651239404876
grade	1.3817032893476293
sqft_above	685734.6672685045
sqft_basement	195872.66840094145
yr_built	862.7972621659763
yr_renovated	161346.2118623043
zipcode	2862.7878348129493
lat	0.01919990179600803
long	0.019832622017890593
sqft_living15	469761.23994532257
sqft_lot15	745518225.3404043
year	0.21866475249044356
month	9.705142556192985
day	74.56430497102993

Таблица 1: Дисперсии признаков датасета

Дисперсия показывает степень разброса значений признаков. Наибольшая дисперсия наблюдается у признаков *sqft_lot* и *sqft_living*, что отражает широкий диапазон размеров участков и жилых площадей. Признаки *waterfront* и *lat* и *long* имеют минимальную дисперсию, что говорит о слабо выраженных изменениях в данных. Промежуточные значения дисперсий у *grade*, *bedrooms* и *bathrooms* указывают на умеренный разброс данных, характерный для качественных категорий. Высокие дисперсии *zipcode* и *yr_renovated* могут свидетельствовать о вариативности регионов и года реконструкции.

2.2 Преобразование выборки

Проведем преобразование выборки:

- Выделим целевую переменную в отдельную переменную `Y`.
- Преобразуем значения признака `date` в формат *datetime*.
- Создадим при помощи данного признака 3 новых: `year, month` и `day`.
- Удалим лишние признаки.
- Разобьем выборку на *train* и *test* в соотношении `80:20`.

3 Исследование поведения алгоритма *RandomForest* для задачи регрессии в зависимости от параметров *n_estimators*, *max_features* и *max_depth*

3.1 Постановка задачи

Исследовать поведение алгоритма *RandomForest* для задачи регрессии в зависимости от следующих параметров:

- количество деревьев в ансамбле - *n_estimators*
- размерность подвыборки признаков для одной вершины дерева - *max_features*
- максимальная глубина дерева (в том числе случай, когда глубина не ограничена) - *max_depth*

Исследование поведения метода подразумевало анализ следующих зависимостей:

- зависимость значения *RMSE* на отложенной выборке
- зависимость *времени работы алгоритма*

3.2 Реализация

Во всех экспериментах до подбора наилучшего параметра при прочих равных увсловиях брались как тестирующие значения параметры:

- $n\_estimators = 250$
- $max\_features = 5$
- $max\_depth = 10$

После нахождения наилучшего параметра - использовался он.

3.2.1 Подбор параметра *n_ estimators*

Будем рассматривать значения $n\_estimators$ от 1 до 250.

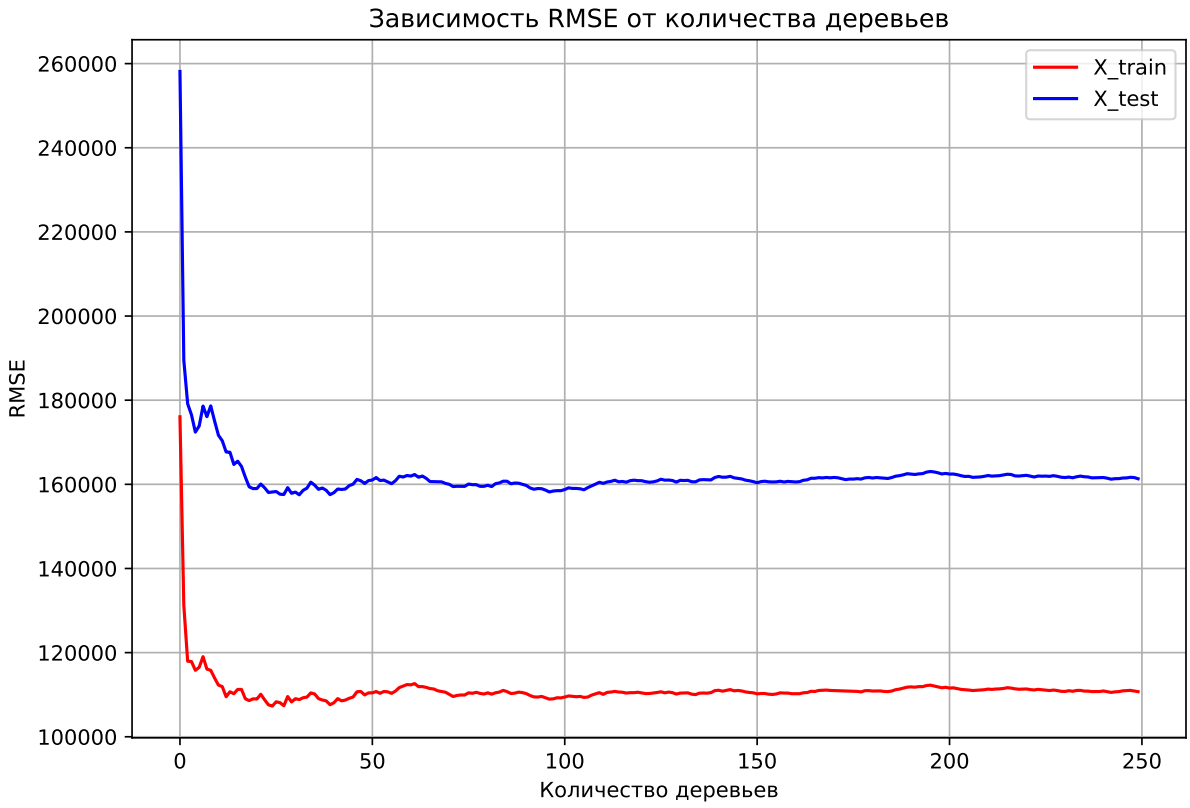

Рис. 3: Зависимость RMSE от количества деревьев в ансамбле

На рисунке 3 представлены графики зависимости значения RMSE на обучающей и отложенной выборке от параметра $n\_estimators$. Как можно заметить из графиков, наилучшее значение RMSE достигается при $n\_estimators = 171$.

3.2.2 Подбор параметра *max_features*

Будем рассматривать значения *max_features* среди следующего множества значений: {1, 2, 3, 4, 5, 7, 9, 10, 12, 15, 17, 19, 21}.

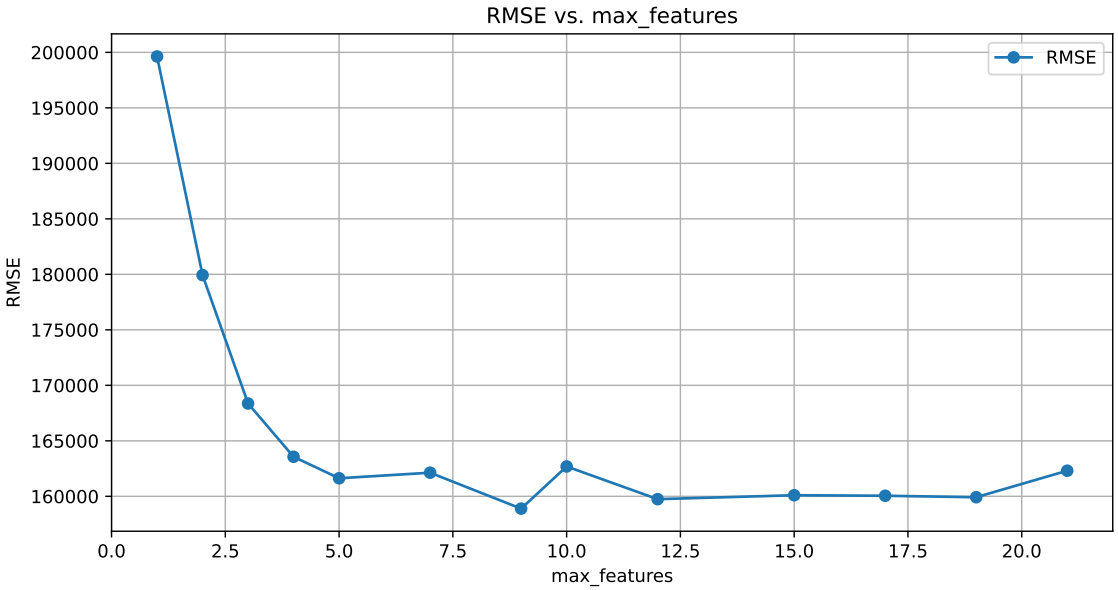

Рис. 4: Зависимость RMSE от размерности подвыборки признаков для одной вершины дерева

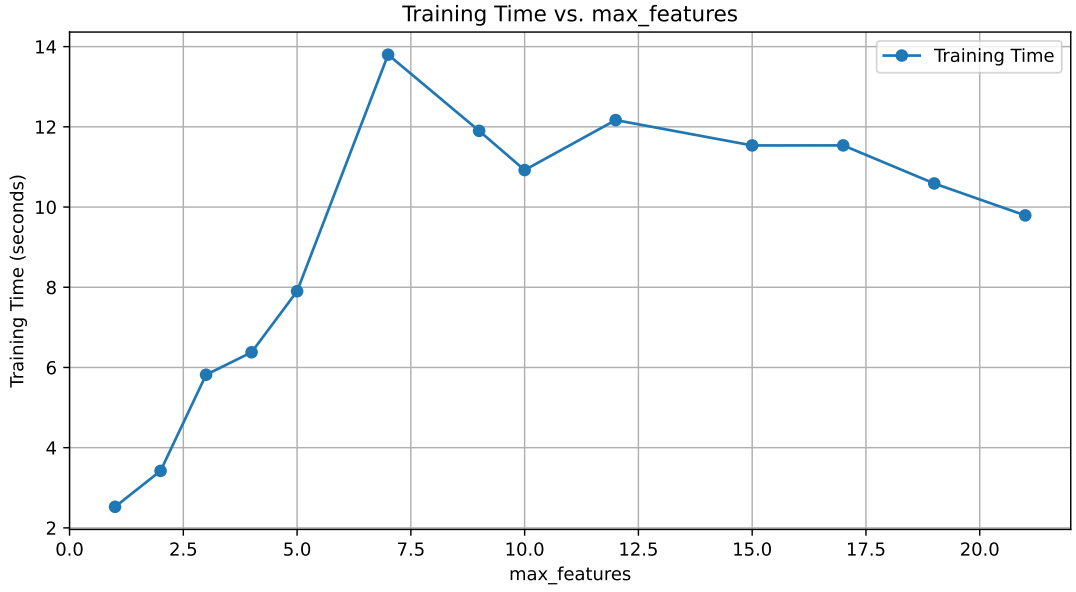

Рис. 5: Зависимость времени работы алгоритма от размерности подвыборки признаков для одной вершины дерева

На рисунках 4 и 5 представлены графики зависимости RMSE и времени работы алгоритма от параметра *max_features*. Как можно заметить из графиков, наилучшее значение RMSE достигается при *max_features* = *9*. Вне зависимости от того, что модель с данным параметром обучается дольше большинства других моделей, мы будем использовать его как наилучший, так как он обеспечивает наилучшее значение целевой функции (RMSE).

3.2.3 Подбор параметра *max_ depth*

Будем рассматривать значения *max_ depth* среди следующего множества значений: {2, 4, 6, 8, 10, 15, 20, 25, 30, 40, 50, 80, *None*}. Последний параметр соответствует неограниченной максимальной глубине дерева.

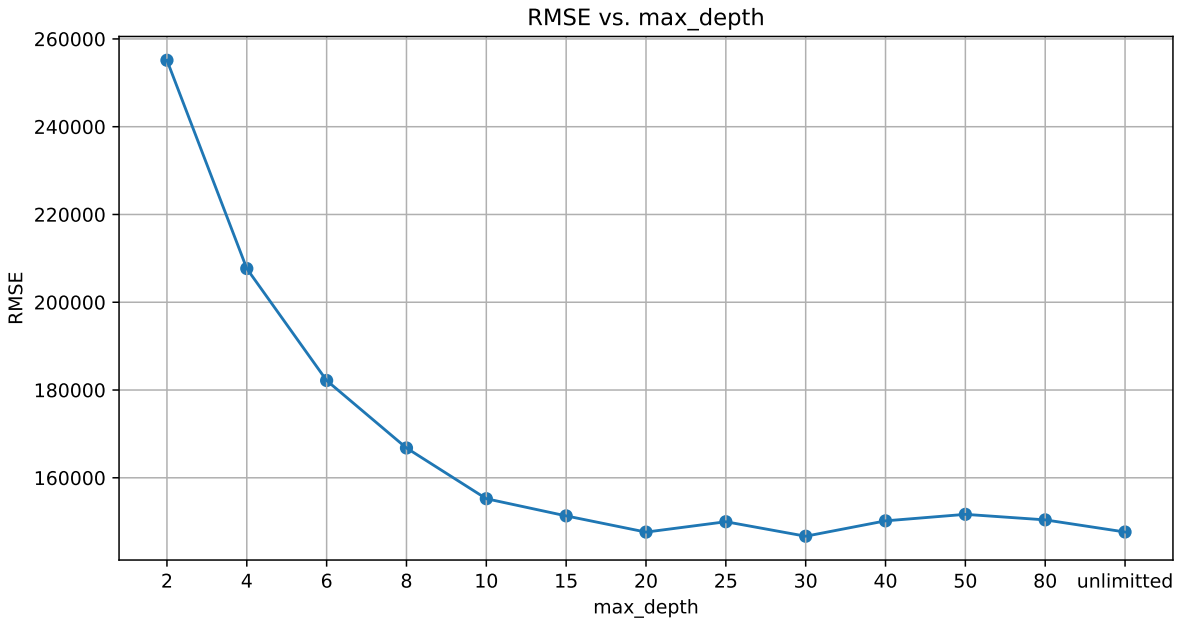

Рис. 6: Зависимость RMSE от максимальной глубины дерева

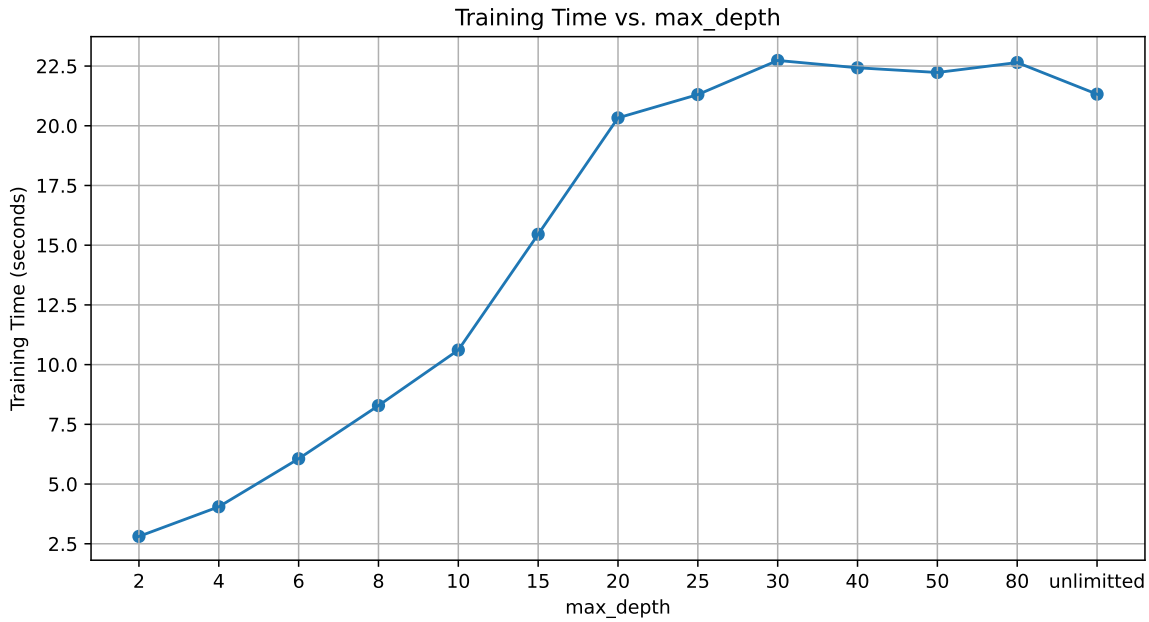

Рис. 7: Зависимость времени работы алгоритма от максимальной глубины дерева

На рисунках 6 и 7 представлены графики зависимости RMSE и времени работы алгоритма от параметра *max_ depth*. Как можно заметить из графиков, наилучшее значение RMSE достигается при *max_ depth = 30*. Вне зависимости от того, что модель с данным параметром обучается дольше большинства других моделей, мы будем использовать его как наилучший, так как он обеспечивает наилучшее значение целевой функции (RMSE).

3.3 Выводы

Проведенные эксперименты показали следующие лучшие параметры модели *RandomForest* для задачи регрессии при прочих равных условиях:

- *n_ estimators = 171*
- *max_ features = 9*
- *max_ depth = 30*

Обучим модель *RandomForest* с наилучшими подобранными параметрами и рассмотрим результаты.

Значение RMSE	Время работы алгоритма
147736.15936707004	21.98995065689087

Таблица 2: Значение RMSE и времени обучения лучшей модели *RandomForest*

В таблице 2 приведены значение RMSE на отложенной выборке и время работы алгоритма с подобранными параметрами.

4 Исследование поведения алгоритма *GradientBoosting* для задачи регрессии в зависимости от параметров *n_estimators*, *max_features* и *max_depth*

4.1 Постановка задачи

Исследовать поведение алгоритма *RandomForest* для задачи регрессии в зависимости от следующих параметров:

- количество деревьев в ансамбле - *n_estimators*
- размерность подвыборки признаков для одной вершины дерева - *max_features*
- максимальная глубина дерева (в том числе случай, когда глубина не ограничена) - *max_depth*
- параметра *learning_rate*

Исследование поведения метода подразумевало анализ следующих зависимостей:

- зависимость значения *RMSE* на отложенной выборке
- зависимость *времени работы алгоритма*

4.2 Реализация

Во всех экспериментах до подбора наилучшего параметра при прочих равных увсловиях брались как тестирующие значения параметры:

- $n\_estimators = 250$
- $max\_features = 5$
- $max\_depth = 10$

После нахождения наилучшего параметра - использовался он.

4.2.1 Подбор параметра *n_estimators*

Будем рассматривать значения *n_estimators* от 1 до 250.

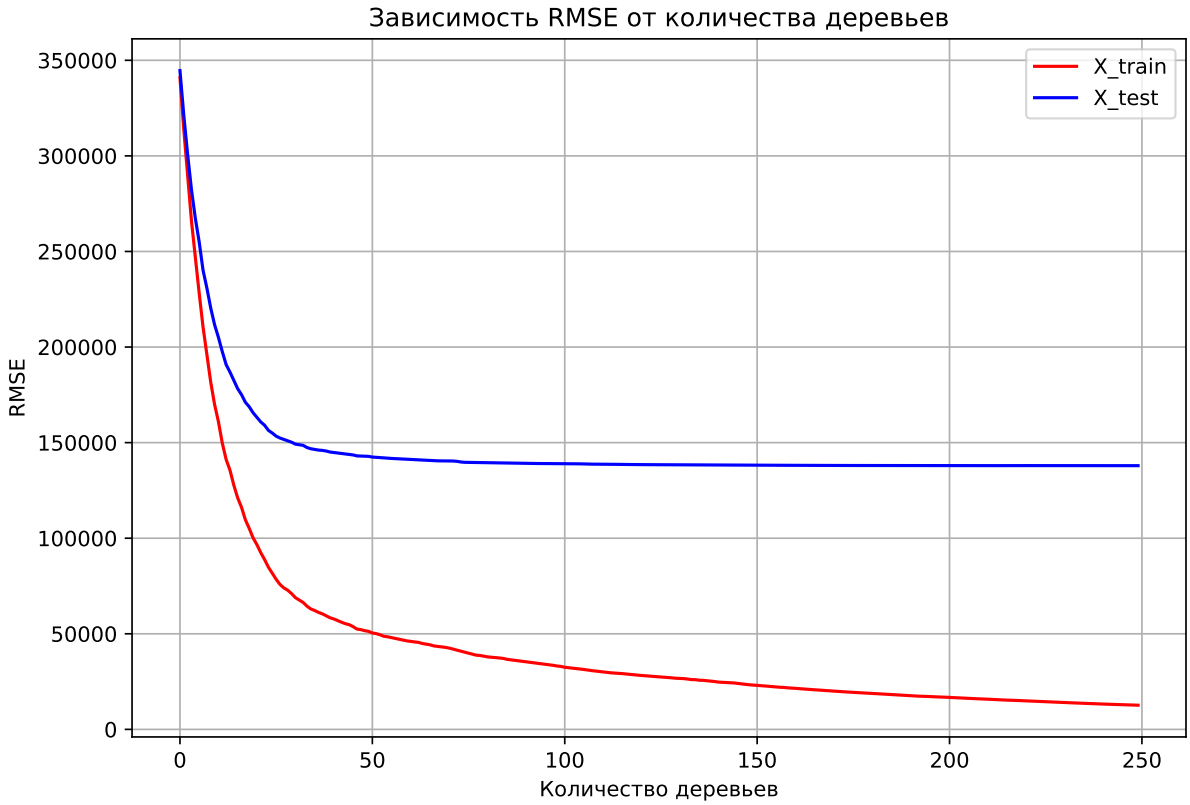

Рис. 8: Зависимость RMSE от количества деревьев в ансамбле

На рисунке 8 представлены графики зависимости значения RMSE на обучающей и отложенной выборке от параметра *n_estimators*. Как можно заметить из графиков, наилучшее значение RMSE достигается при *n_estimators* = *211*.

4.2.2 Подбор параметра *max_features*

Будем рассматривать значения *max_features* среди следующего множества значений: {1, 2, 3, 4, 5, 7, 9, 10, 12, 15, 17, 19, 21}.

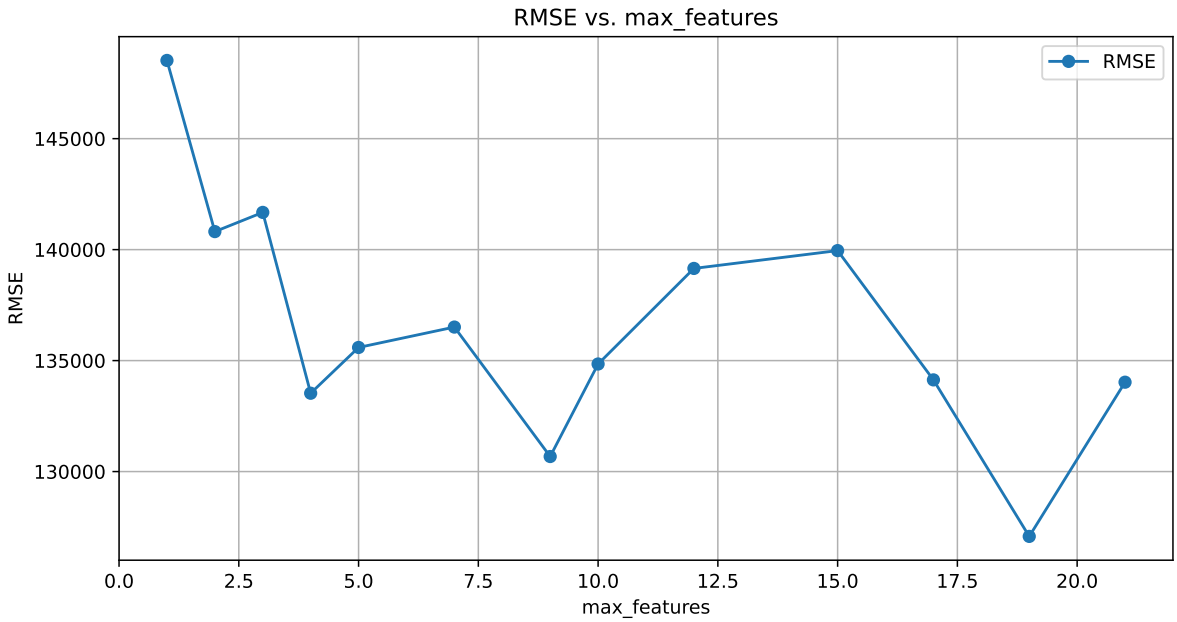

Рис. 9: Зависимость RMSE от размерности подвыборки признаков для одной вершины дерева

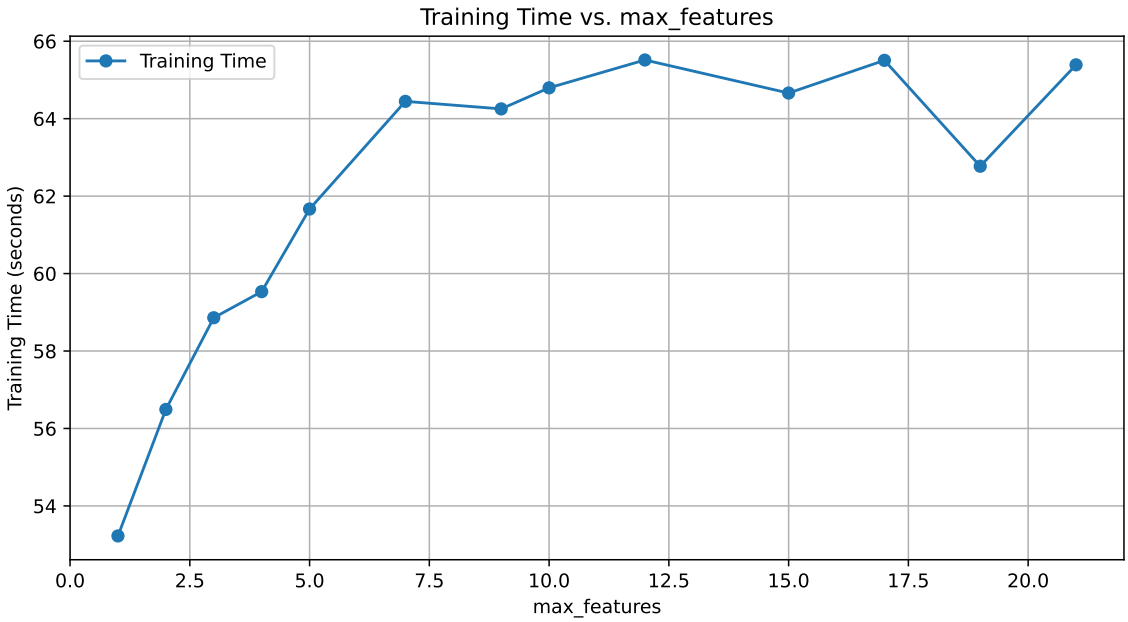

Рис. 10: Зависимость времени работы алгоритма от размерности подвыборки признаков для одной вершины дерева

На рисунках 9 и 10 представлены графики зависимости RMSE и времени работы алгоритма от параметра *max_features*. Как можно заметить из графиков, наилучшее значение RMSE достигается при *max_features* = *19*. Вне зависимости

от того, что модель с данным параметром обучается дольше большинства других моделей, мы будем использовать его как наилучший, так как он обеспечивает наилучшее значение целевой функции (RMSE).

4.2.3 Подбор параметра *max_ depth*

Будем рассматривать значения *max_ depth* среди следующего множества значений: {2, 4, 6, 8, 10, 15, 20, 25, 30, 40, 50, 80, *None*}. Последний параметр соответствует неограниченной максимальной глубине дерева.

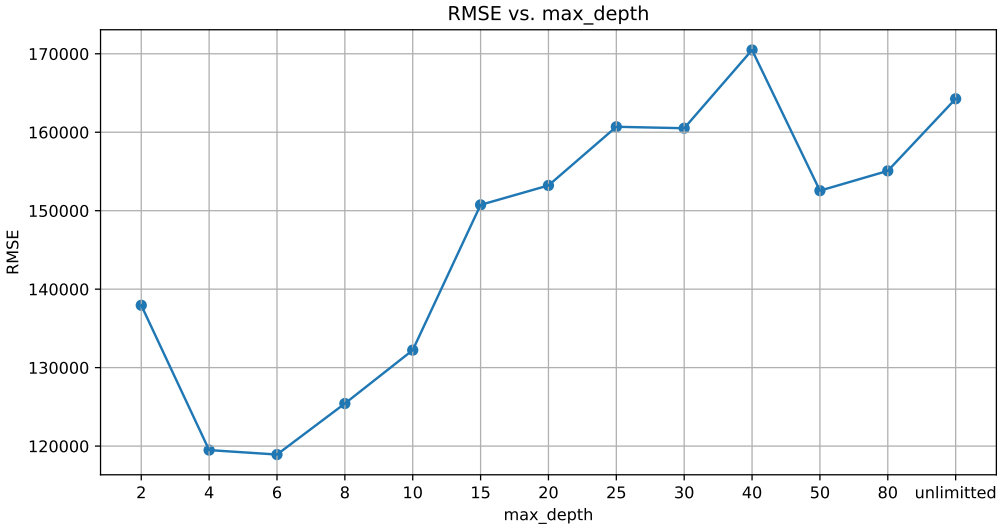

Рис. 11: Зависимость RMSE от максимальной глубины дерева

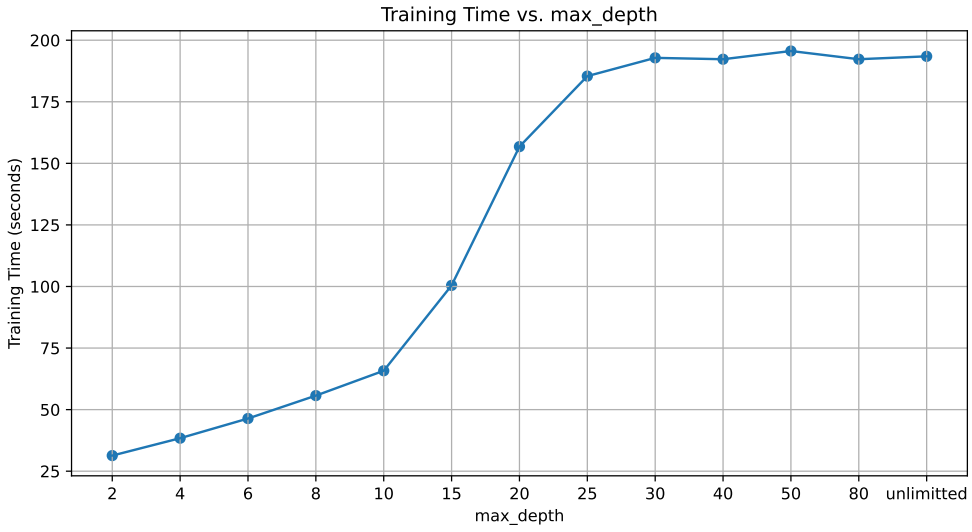

Рис. 12: Зависимость времени работы алгоритма от максимальной глубины дерева

На рисунках 11 и 12 представлены графики зависимости RMSE и времени работы алгоритма от параметра *max_ depth*. Как можно заметить из графиков, наилучшее значение RMSE достигается при *max_ depth* = *6*. Вне зависимости от того, что модель с данным параметром обучается дольше большинства других моделей, мы будем использовать его как наилучший, так как он обеспечивает наилучшее значение целевой функции (RMSE).

4.2.4 Подбор параметра *learning_ rate*

Будем рассматривать значения *learning_ rate* среди множества чисел, расположенных от `0.01` до `1`, взятых в количестве двадцати штук.

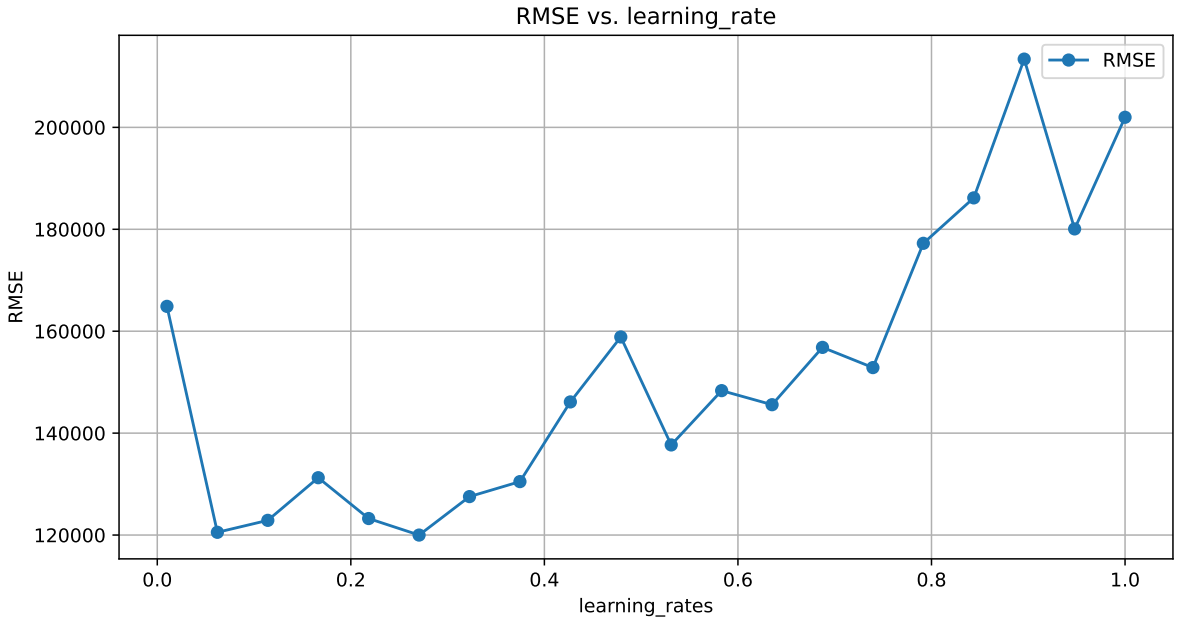

Рис. 13: Зависимость RMSE от параметра *learning_ rate*

Рис. 14: Зависимость времени работы алгоритма от параметра *learning_rate*

На рисунках 13 и 14 представлены графики зависимости RMSE и времени работы алгоритма от параметра *learning_rate*. Как можно заметить из графиков, наилучшее значение RMSE достигается при $learning\_rate = 0.27052631578947 37$.

4.3 Выводы

Проведенные эксперименты показали следующие лучшие параметры модели *RandomForest* для задачи регрессии при прочих равных условиях:

- $n\_estimators = 211$
- $max\_features = 19$
- $max\_depth = 6$
- $learning\_rate = 0.27052631578947 37$

Обучим модель *GradientBoosting* с наилучшими подобранными параметрами и рассмотрим результаты.

Значение RMSE	Время работы алгоритма
123530.98922074263	46.27242064476013

Таблица 3: Значение RMSE и времени обучения лучшей модели *GradientBoosting*

В таблице 3 приведены значение RMSE на отложенной выборке и время работы алгоритма с подобранными параметрами.

5 Итог

В ходе работы проведён анализ методов ансамблирования для задачи регрессии на данных о продажах недвижимости. Реализованы алгоритмы *RandomForest* и *GradientBoosting*, исследованы их параметры и выявлены оптимальные. Random Forest показал лучшие результаты при параметрах $n\_estimators = 171$, $max\_features = 9$ и $max\_depth = 30$, обеспечив RMSE `147736.16` за `21.99` секунды, тогда как для *GradientBoosting* оптимальными стали $n\_estimators = 211$, $max\_features = 19$, $max\_depth = 6$ и $learning\_rate = 0.27$, что привело к RMSE `123530.99` за `46.27` секунд. Отличия в параметрах объясняются разными подходами моделей: *RandomForest* снижает шум через усреднение множества глубоких деревьев, а *GradientBoosting* акцентируется на последовательном исправлении ошибок, используя меньшую глубину деревьев для предотвращения переобучения. Таким образом, *GradientBoosting* обеспечивает лучшую точность, а *RandomForest* — более быстрое выполнение.

Список литературы

[1] Материалы семинаров ММРО и Практикума 3 курса ВМК МГУ, https://github.com/mmp-practicum-team/mmp_practicum_fall_2024/blob/main/Seminars/08-text-processing-and-logreg/seminar.pdf