# Итоговая работа

Карькова Наталья, DST-3

# Постановка задачи

К вам обратился представитель крупного агентства недвижимости со следующей проблемой:

Мои риелторы тратят катастрофически много времени на сортировку объявлений и поиск выгодных предложений. Поэтому их скорость реакции, да и, сказать по правде, качество анализа не дотягивает до уровня конкурентов. А это сказывается на наших финансовых показателях. Твоя задача — разработать модель, которая бы позволила обойти конкурентов по скорости и качеству совершения сделок. Датасет прикладываю.

Задача: разработать сервис, который будет предсказывать стоимость домов, основываясь на истории предложений.

Цель: как можно точнее на основе характеристики дома предсказать его стоимость.

# Анализ данных

(код в ноутбуке «1. Features»)

Общее количество строк в наборе данных 377185.

Набор данных содержит следующие поля:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Наименование | Количество ненулевых объектов | Количество уникальных значений | Описание |
|  | status | 337267 | 159 | Статус продажи дома |
|  | private pool | 4181 | 1 | Наличие бассейна |
|  | propertyType | 342452 | 1280 | Вид собственности |
|  | street | 377183 | 337076 | Улица |
|  | baths | 270847 | 229 | Информация о ванных комнатах |
|  | homeFacts | 377185 | 321009 | Общие сведения о доме |
|  | fireplace | 103115 | 1653 | Информация о каминах и других обогревах дома |
|  | city | 377151 | 2026 | Город |
|  | schools | 377185 | 297365 | Информация о трех близлежащих школах |
|  | sqft | 336608 | 25405 | Площадь, в квадратных футах |
|  | zipcode | 377185 | 4549 | Почтовый индекс |
|  | beds | 285903 | 1184 | Информация о количестве спален |
|  | state | 377185 | 39 | Штат |
|  | stories | 226470 | 348 | Этажи |
|  | mls-id | 24942 | 24907 | Идентификатор |
|  | PrivatePool | 40311 | 2 | Еще одно поле о наличии бассейна |
|  | MlsId | 310305 | 232944 | Еще один идентификатор |
|  | target | 374704 | 43939 | Стоимость дома |

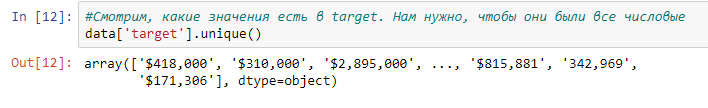
Далее будем рассматривать все поля по отдельности. Тип данных везде указан как object, что неприемлемо для обучения. Необходимо привести все данные к типу int или float.

## Обработка и очистка данных

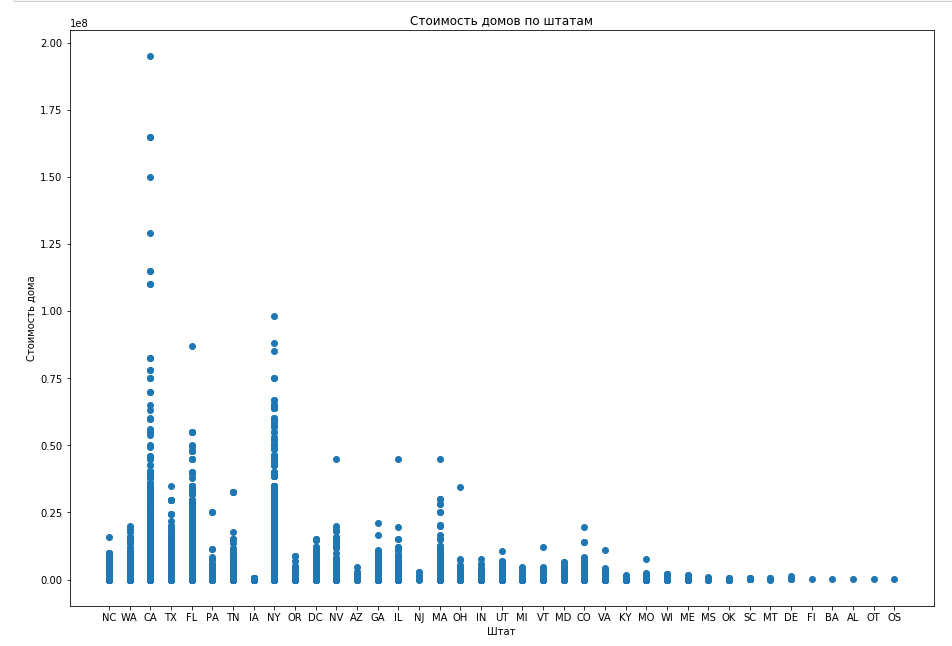
## target

Поле содержит информацию о стоимости дома. Это наша целевая переменная. Строки, в которых отсутствует значение, откидываем, так как ни для обучения, ни для тестирования они не пригодятся.

Пример значений приведен на рисунке. Для дальнейшего использования данных оставляем в них только цифры.



Можно заметить, что стоимость дома зависит от штата. Добавим эту информацию в наш датасет и сохраним ее, чтобы затем использовать для обработки входных данных.



## Status

Поле содержит информацию о статусе продажи дома.

Значение необходимо учитывать, так как есть дома для аренды, в этом случае его цена указана «за месяц».

Формируем новый столбец с признаком «продажа или аренда». Если в статусе есть слово ‘rent’, то считаем это арендой. Остальное все – продажа.

Домов на аренду значительно меньше, чем домов на продажу.

## propertyType

Поле содержит информацию о типе дома. Приводим некоторые типы к единому стилю (пример: 'single-family home','Single Family Home','Single Family'). Считаем hash, так как непонятно, сколько может быть таких типов.

## Street и city

Поле city содержит название города. Заменяем пропущенные значения на ‘no\_city’. Приводим все названия к нижнему регистру. Количество уникальных значений сократилось с 2026 до 1909. Это преобразование используем только для визуализации графиков.

Поле street содержит часть адреса дома (улица, номер дома).

Так как улиц и городов может быть много, и неизвестно, может быть в новых данных встретятся неизвестные нам сейчас значения, то считаем hash. Формируем новые столбцы:

* отдельно по улицам,
* отдельно по городам,
* вместе от целой строки «город+улица».

## baths

Поле содержит информацию о количестве ванных комнат. Иногда указывается значение без опознавательных знаков, но число с разделителем «запятая». Считаем, что это опечатка и заменяем его на «точку». Преобразуем поле в «Количество ванных комнат», отбросив все слова. Тип нового поля float.

## beds

Поле содержит информацию о количестве спальных комнат (иногда о их площади).

Так как нет возможности оставить разнородную информацию в одном поле, то решено разделить данные на два поля: Количество спален и Площадь спален. При этом площадь переведена в квадратные метры, чтобы была единая единица измерения.

## Schools

Поле содержит информацию о близлежащих школах. Количество школ – для каждого дома разное. Для каждой школы указывается ее имя, рейтинг, дистанция, какие классы.

Preschool заменяем на PK (судя по той информации, что я нашла, это одно и то же)

Выделяем в отдельные столбцы:

* количество школ по классам, включая K и PK
* общее количество школ
* минимальную, максимальную и среднюю дистанцию до школ
* максимальный рейтинг школы
* некая придуманная метрика, которая получается путем перемножения списка рейтингов на список дистанций до школ

## sqft

Поле содержит информацию о площади дома. Отсутствующие значения заменяем на 0. В существующих оставляем только цифры. Если в описании только буквы, то заменяем на 0. Приводим к формату int.

Удаляем выбросы: строки, где площадь дома более 40 млн кв. футов (так как у них указан тип как многоквартирный дом, а стоимость невелика). Для остальных «больших» площадей указан тип «land», понимаем это так, что продается дом с землей.

## Zipcode

Поле содержит информацию о почтовом индексе. Оставляем в поле только цифры (избавляемся от остальных знаков). Преобразуем в тип int. Целочисленные значения записываем в новое поле zipcode\_int.

## PrivatePool и private pool

Поля содержат информацию о наличии приватного бассейна. Судя по всему, различные названия полей пришли из разных систем, но данные содержат примерно одинаковые.

Формируем новый столбец, который содержит 1, если бассейн указан хотя бы в одном из этих полей, и 0 в противном случае.

## fireplace

Поле содержит информацию об отоплении (хотя название поля говорит вроде бы о наличии камина). Данные очень грязные. В некоторых строках написано «да», в некоторых строках идет перечисление вообще всех отопительных приборов в доме.

Преобразуем этот столбец в fireplace\_y\_n, который будет содержать информацию о том, есть (1) или нет (0) информация об отоплении. В качестве «нет информации» считаем значения 'nan', 'not applicable', 'no'.

Делаем еще одно поле с количеством характеристик обогрева (сколько вариантов перечислено в исходном поле).

## Stories

Поле содержит информацию о количестве этажей в продаваемом доме. Либо же, на каком этаже находится квартира.

Так как данные очень грязные, то просто пытаемся привести данные в тип float. Если это не получается, то убираем все буквы и приводим к типу float.

Тип float выбран из-за того, что в значениях может быть, например, «1.5 level»

## state

Поле содержит двухбуквенное обозначение штата, в котором находится дом. Кодируем его по порядку с помощью метода factorize. Сохраняем на будущее список категорий с порядковыми номерами. Это пригодится в дальнейшем, когда надо будет преобразовывать строку с входными данными.

## homeFacts

Поле содержит комплексную информацию по дому, разложенную в словарь со следующими метками: {'Cooling', 'Heating', 'Parking', 'Price/sqft', 'Remodeled year', 'Year built', 'lotsize'}

Разбираем эту информацию по соответствующим полям. Затем 'Cooling', 'Heating', 'Parking' заменяем на 1, если там есть какая-то информация, на 0, если ее нет.

'Price/sqft' – избавляемся от лишних символов, переводим в тип int

'Remodeled year', 'Year built' – выделяем год. Удаляем выброс со странным очень большим годом в Годе постройки.

'lotsize' – выделяем числа. Переводим все в единый формат (акры в квадратные футы)

## mls-id и MlsId

Выявлено, что поля mls-id и MlsId идентичны, содержат некий идентификатор дома. Дома, имеющие один и тот же идентификатор, но в различных полях, имеют практически идентичное описание в других полях. Однако, стоимость иногда отличается.

Объединяем оба идентификатора в один столбец.

Удаляем все уже обработанные столбцы и удаляем дубликаты строк (их оказалось порядка 80).

Кроме того, есть строки, в которых одинаковые значения в полях, но разные значения целевой переменной. Такие строки удаляем совсем, так как нам неизвестно, какие значения более правильные.

## Дополнительно

Так как у нас есть поля «стоимость 1 квадратного фута» (price) и «площадь дома» (sqft), и еще некий lot, то добавим в датасет поля

price\_sqft = price \* sqft

price\_lot = price \* lot

## Результат обработки и очистки данных

Все исходные данные обработаны. В результате получилось 46 колонок с данными с типом int или float. Сохраняем датасет в файл.

После всего этого проведем анализ получившегося датасета с помощью feature\_selector (код в ноутбуке «2. Feature selector»). Он предлагает удалить 10 колонок, наименее важных. Удалим их и сохраним датасет в отдельный файл.

# Выбор алгоритма для обучения

(код в ноутбуке «3. Learning»)

Для предсказания стоимости дома сравним несколько алгоритмов:

* линейная регрессия
* случайный лес
* градиентный бустинг

Перед началом выбора алгоритма разделим наш датасет на две части: для обучения и для валидации в соотношении 80:20. Также зафиксируем RANDOM\_SEED=34 для возможности дальнейшего воспроизведения всех экспериментов.

Сравнивать алгоритмы будем по нескольким метрикам:

* Mean Absolute Error (MAE), которая показывает среднее отклонение предсказанных значений от фактических
* RMSE
* R2-score – коэффициент детерминации (чем ближе к 1, тем лучше)

Сравнивать будем практически на дефолтных параметрах.

Кроме того, посмотрим модели на двух датасетах: обработанный и датасет с удаленными колонками после feature-selector.

Подсчитанные метрики в ноутбуке.

Как видим, наилучший результат дает алгоритм Градиентного бустинга на сокращенном датасете. Его и возьмем для прототипа.

Так как для прототипа понадобится наша обученная модель, то завернем ее в файл с помощью pickle.

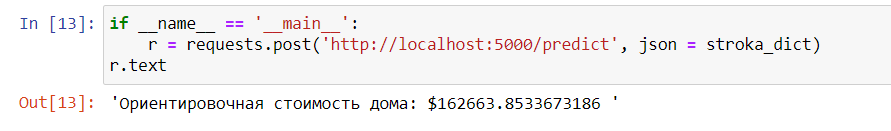
# Прототип

Для прототипа реализуем сервер на Flask. Он будет развернут по адресу <http://localhost:5000>. Файлы для запуска находятся в папке Prototype.

Для получения предсказания необходимо будет обратиться по ссылке <http://localhost:5000/predict> и передать серверу исходные значения.

Исходные значения передаются в виде словаря. Поля точно такие же, как в исходном датасете. Для имитации работы запустим сервер и обратимся к нему из ноутбука (код в ноутбуке «5. Запуск прототипа». В качестве данных можно передать строку из исходного датасета, либо же сформировать ее самостоятельно. Образец строки приведен в ноутбуке.

Результатом работы сервиса будет являться ориентировочная стоимость дома по его указанным характеристикам.



# Выводы

В ходе работы удалось создать модель, предсказывающую стоимость дома, и продемонстрировать ее работу. Опробованы несколько алгоритмов для обучения. Все они дали примерно одинаковую точность на имеющихся данных.

Для улучшения точности предсказаний необходимо более глубокая работа с данными: создание новых признаков, подключение данных извне.