

ФИНАЛЬНЫЙ ПРОЕКТ

Python для аналитики 2.0

Постановка задачи



Запрос клиента

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

SMART-цель

До 01.06.2020 (включительно) разработать работающий сервис, который будет предлагать оптимальную цену на объект недвижимости исходя из его вводных параметров. Процент отклонения от цены (ошибка) не будет превышать 15%.





MVP

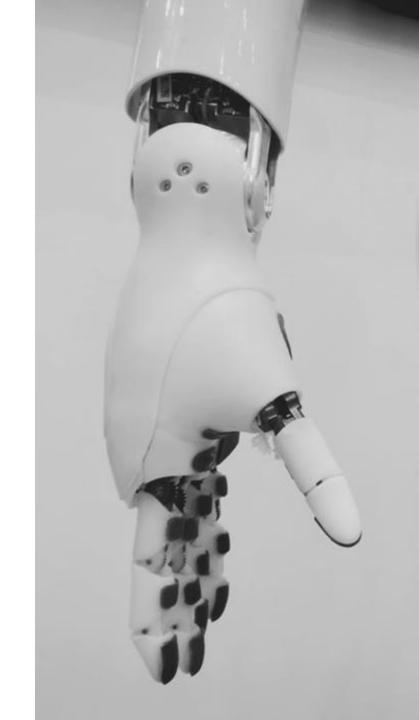
Разработанный сервис который предлагает оптимальную цену на объект недвижимости исходя из его вводных параметров.

Определение задачи для ML

Динамическое ценообразование: изменение стоимости товара или услуги в зависимости от множества параметров

Выбор алгоритма

Так как у нас имеется target-переменная (целевая), а множество значений для target-переменной бесконечно, то лучше всего для решения данной задачи подойдет класс обучение с учителем, подкласс регрессия.



Исследование рынка

Выбор основных критериев



Территориальное расположение

Цены на недвижимость могут значительно различаться в разных штатах.



Инфраструктура

Близость дома к важным для жизни объектам (метрополитен, школа, больница и т.д.)



Тип недвижимости

Разные типы недвижимости обладают различной статустностью и уровнем комфорта.



Особенности объекта

Площадь и планировка, год постройки, а также наличие лифта, мусоропровода, отопления, паркинга и т.д.

Обработка данных



Очистка данных

Первичный анализ структуры и содержания предоставленного датасета выявил следующие проблемы:

- наличие пропусков информации
- разные форматы данных
- дублирование столбцов
- присутствие «мусора» в большом количестве
- отсутствие данных в числовом формате

Оценка данных



status

Показывает на каком этапе находится продажа. Необходимо унифицировать данные в зависимости от наличия покупателя



street

Адрес объекта. Данные довольно сложно перевести в числовой формат. Чтобы избежать перегрузки информацией удалим эти данные



fireplace

Показывает наличие камина. Желательна унификация в числовом формате



private pool

Показывает наличие собственного бассейна. Желательна унификация в числовом формате



baths

Количество ванных комнат. Данные имеют разный формат. Необходимо выделить числовой показатель



city

Город объекта. Данные довольно сложно перевести в числовой формат. Чтобы избежать перегрузки информацией удалим эти данные



propertyType

Тип недвижимости. Необходимо унифицировать данные в зависимости от комфортности и статустности жилья



homeFacts

Содержит словарь с информацией о доме. Необходимо вытащить нужные нам данные в отдельные графы



schools

Содержит словарь с информацией о ближайших школах. Необходимо вытащить нужные нам данные в отдельные графы

Оценка данных



sqft

Жилая площадь. Данные имеют разный формат. Необходимо выделить числовой показатель



state

Штат объекта. Необходимо выделить числовой показатель. Например, по уровню преступности



PrivatePool

Дублирует столбец private pool. Требуется объединение данных



zipcode

Данные довольно сложно перевести в числовой формат. Чтобы избежать перегрузки информацией удалим эти данные



stories

Количество этажей. Данные имеют большое количество пропусков. Чтобы избежать перегрузки информацией удалим эти данные



MIsId

Дублирует столбец mls-id. Требуется объединение данных



beds

Количество жилых комнат. Данные имеют разный формат. Необходимо выделить числовой показатель



mls-id

Показывает присутствие объекта в базе мультилистинга. Желательна унификация в числовом формате



target

Цена. Данные имеют разный формат. Необходимо выделить числовой показатель

Используемые приёмы

Для очистки данных я использовала:

- Объединение столбцов с общими значениями
- Указание нового значения при отсутствии данных
- Выделение общего признака и стандартизация данных
- Очистка данных от лишних символов
- Замена пропусков на среднее значение (медиану)
- Удаление строк/столбцов с пропусками
- Обогащение данных новыми признаками на основе имеющихся данных



Используемые библиотеки

- Pandas
- Numpy
- Re





Используемые функции/методы

- isnull()
- fillna()
- apply()
- describe()
- groupby()
- replace()

- extract()
- split()
- compile()
- drop()
- read_excel()
- merge()

eval()

def

- lambda
- to_numeric

Визуализация и нормализация

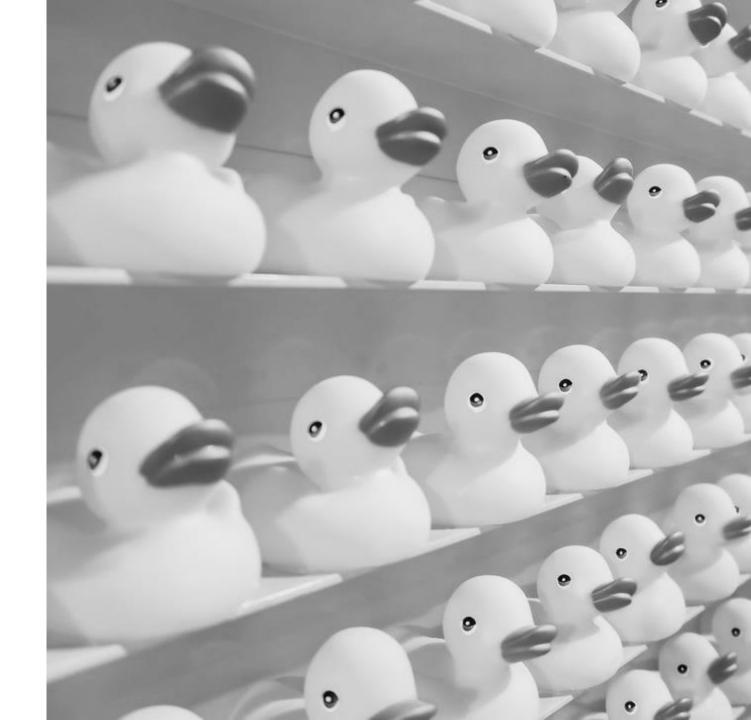


Построение гистограмм позволило оценить распределение данных по выборке внутри столбца и значений в признаке, а также выявить очевидные выбросы.

С помощью корреляционного графика была проведена проверка данных на наличие линейной зависимости и сильной корреляции.

Нормализация данных

Для снижения чувствительности алгоритма к масштабу признаков проведем нормализацию методом MinMax..



Используемые библиотеки

- Pandas
- Seaborn
- Sklearn





Используемые функции/методы

- hist()
- bins
- corr()
- get_dummies()

- MinMaxScaler()
- fit_transform()

Разработка модели

Выбор алгоритма

Мы уже ранее определили, что для решения задачи динамического ценообразования нам подойдет алгоритм обучение с учителем (регрессия).

Попробуем найти зависимостей между определяющими переменными и определяемой переменной с помощью линейной регрессии.





Построение модели

Для начала разделим данные на признак и целевую переменную, а затем всю выборку – на тренировочную и тестовую.

Попробуем самостоятельно построить классификатор с помощью методов оптимизации. Для этого реализуем функцию вычисления градиента функции MSE, шаг градиентного спуска и процедуру оптимизации.

Оптимизируем параметр линейной регрессии theta на всех данных, сделаем предсказание, посчитаем значение ошибок.

Т.к. значение ошибок получилось очень высоким, мы заново разобьём выборку на train/test, оптимизируем theta, сделаем предсказания и посчитаем ошибки MSE и RMSE.



Использование библиотек

Поскольку построенная модель имеет довольно высокие значения MSE и RMSE, применим имеющиеся модели в пакете scikit-learn.

Начнём с простой линейной регрессии и импортируем пакет NumPy и класс LinearRegression из sklearn.linear_model. Валидация модели показала среднюю погрешность в размере \$321 736.

Попробуем использовать градиентный бустинг. Для этого импортируем GradientBoostingRegressor из sklearn.ensemble. Валидация модели – \$220 087.

Проверим коэффициент детерминации R². Для нашей модели он получился равным 0.63.

Используемые библиотеки

- Pandas
- Numpy
- Sklearn
- Pyplot





Используемые функции/методы

- LinearRegression()
- GradientBoostingRegressor() •
- mean_squared_error()
- train_test_split()
- gradient_step()
- r2_score()

- transpose()
-) dot()
- ones()
- hstack()
- fit()
- predict()

Результат



В ходе выполнения финального проекта мне не удалось выполнить изначально заявленную цель (отклонение не более 15%). При среднем значении стоимости объекта недвижимости \$ 544 848 размер среднего отклонения составил примерно \$220 087, а коэффициент детерминации – 0.63.

Разработчик



Дарья ХудалееваМенеджер проектов департамента внутреннего аудита