СЛАЙД1:

здравствуйте! Меня зовут Макарова Елена

Целью моей дипломной работы был проект по умолчанию: «Предсказание стоимости недвижимости»

СЛАЙД2:

Данная работа посвящена актуальной теме, а именно, созданию модели прогнозирования стоимости недвижимости с использованием методов машинного обучения, с последующей коммерциализацией проекта в виде мобильного приложения или сервиса, предоставляющего возможность расчета стоимости недвижимости.

Задача предсказания стоимости недвижимости для определения выгодных предложений или того что цена завышена, на данный момент используется такими популярными компаниями как ЦИАН, ДомКлик, Авито, яндекс недвижимость)).

СЛАЙД3:

Для выполнения задачи были поставлены некоторые цели.

– разведывательный анализ данных

- построение различных моделей машинного обучения и выбор лучших результатов

Наши результаты естественно очень сильно зависят от имеющихся данных и их качества, поэтому хотелось немного остановится на самих данных и некоторых трудностях очистки их.

СЛАЙД4:

* Будем использовать уже готовый датасет .
* В нашем наборе данных 18 признаков описывающих недвижимость и 377185 строк
* Целевая переменная **target** - цена объекта недвижимости

СЛАЙД5:

* По первичной информации можно сказать, что
* Очень загрязненные данные, а именно:
* Много пропусков!
* Есть дубликаты
* Есть большое количество по разному записанных, НО одинаковых по значению данных.
* Сложные категориальные признаки
* Столбцы homeFacts и schools это вложенные датасеты в формате json

СЛАЙД6:

Очень много времени было потрачено на EDA и Features Engineering.

Приведу несколько примеров очистки признаков.

Целевая переменная как и многие другие признаки содержали различные символы , выбросы и требовали глубокой очистки. После очистки и логарифмирования распределение целевого признака стало нормальным.

СЛАЙД7:

Из 2х сложных json признаков (homeFact и schools), после обработки и очистки были получены новые признаки:

(Среднее, минимально и максимальное значение по рейтингу школ и расстоянию до них)

* YEAR BUILT - год постройки
* REMODELED YEAR - год реконструкции
* HEATING – отопление
* PARKING - парковка
* LOTSIZE – площадь.
* PRICE/SQFT - цена за кв.Фут
* В новых полученных признаках были свои сложности, например в цене за кв.фут значения были и в акрах и в футах. Т.е требовало приведения к одному измерению

Добавление в датасет признаков 'Heating' и 'Cooling' ухудшали метрику

А вот 'Parking’ , Price/sqft, lotsize, 'Year built' делает лучше!

СЛАЙД8:

В признаке количество ванных комнат более 100000 пропусков, а дробные значения это не опечатка

В США наличие ванной комнаты обозначается следующим образом:

* Если 1 - это значит, что это полноценная ванная комната с душем, туалетом, ванной, раковиной.
* Если 0.5 - имеет только два из четырех основных компонентов ванной комнаты, обычно унитаз и раковину
* Если 0.75 - значит чего-то одного не хватает
* Если 0.25 - комната только с одним из четырех элементов, обычно это туалет

СЛАЙД9:

* В признаке количество спален
* И так же требовалась тщательная очистка
* Пропуски: 89929 Уникальных: 1122
* Как видим по-разному записаны одни и те же значения.
* В итоге получили вот такое нормальное распределение после обработки, как видим на графике справа

СЛАЙД10:

* Так как признаки камин, бассейн и парковка были очень грязные и слишком много пропусков, целесообразнее было привести их бинарному виду., просто их наличие или отсутствие.

СЛАЙД11:

В результате всех преобразований получилось 32 признака: 26 числовых и 6 категориальных.

Во время обработки создано 19 новых признаков.

Сильная корреляция наблюдается между годом реконструкции и сколько лет назад сделана реконструкция (-0,9)

Между годом постройки и сколько лет назад построено (-0,7 ) - логично они созданы из друг друга.

(Средняя дистанция и максимальная дистанция, и рейтинги школ.)

НО в дальнейшем было проверенно! Что вариант при сохранении всех признаков , дал лучше результат!

СЛАЙД12:

* Многие алгоритмы машинного обучения не могут работать с категориальными данными напрямую. Они требуют, чтобы данные были числовыми. Поэтому перед обучением модели требуется кодировка признаков
* В данном проекте я проверила несколько вариантов кодировки
* 1)Frequency Encoding –( для признаков с высокой спектром значений)

LabelEncoder – (Преобразует все значения категориальных признаков в числа)

* 2) *One-Hot Encoding – создание dummy переменных.*
* Прежде чем приступать к моделированию, нужно выполнить «центровку» и «стандартизацию» данных путём их масштабирования.: выбрала для этого MinMaxScaler, он не искажает расстояния между значениями в каждой функции

СЛАЙД13:

* В качестве наивной модели я использовала линейную регрессию
* И модель с логарифмированием целевой переменной показала лучше результат. Поэтому далее в моделях будем использовать логарифмирование целевого признака,

Так же , если не удалять коррелирующие признаки, то результат немного становится лучше!

СЛАЙД 14:

* Далее посмотрела какие результаты получатся у модели случайного леса (RandomForestRegressor)
* И модель показала очень хороший результат!! (один из лучших, если с подбором параметров)

СЛАЙД 15:

* Далее еще посмотрела какие результаты получатся у таких трех моделей (ExtraTreesRegressor, CatBoostRegressor, XGBRegressor )
* И модели показали неплохой результат!!
* К алгоритму XGBoost был произведен подбор гиперпарапетров и крос-валидация

СЛАЙД 16:

* Далее посмотрела какие результаты получатся у (**асамбля) Bagging со случайным лесом и с градиентным бустингом**
* (Bagging (от Bootstrap aggregation) — это один из первых и самых простых видов ансамблей.)

СЛАЙД 17:

* **Стекинг**— один из самых популярных способов [ансамблирования](https://en.wikipedia.org/wiki/Ensemble_learning) алгоритмов, т.е. использования нескольких алгоритмов для решения одной задачи машинного обучения.
* В качестве базовых моделей выбрала две с лучшими результатами - XGBRegressor & RandomForestRegressor
* Для обработки результатов взяла линейную регрессию
* Хороший результат! ( на экране)

СЛАЙД 18:

* Так же по вашим рекомендациям я еще добавила несколько моделей: первая это – LGBMRegressor.
* Модель выдает хорошие результаты при меньшем времени!

СЛАЙД 19:

* Вторая добавленная модель – из deep learning модели - TabNet, к сожалению она обучается очень долго и при попытки улучшения с помощью параметров занимало больше 30 часов и все слетало( поэтому я оставила первоначальные результаты которые смогла получить на этой модели.

СЛАЙД 20 :

**Результат**

Во время исследования все модели запускались много раз. Проверяли следующие моменты:

* Оставлять или нет корелирующие признаки по дате постройки и реконструкции, мин, макс и средн. дистанцией и рейтингом школ.
* учитывать или нет недвижимость построенная раньше 1750 г,
* логорифмировать или нет числовые признаки и сам таргет!,
* Выбор обработки категориальных данных: Dumming, LabelEncoding
* Проводить или нет стандартизацию данных перед обучением.

По логике все эти процедуры должны улучшать обучение, однако, например

* учет или не учет очеь старой недвижимости практически не изменяет результат. Просто таких очень мало в процентах.
* Не удаление кореллирующих признаков немного улучшило результаты (1-2%).
* Обработки категориальных данных - Dumming показало лучше результат, чем LabelEncoding
* Логорифмирование таргета тоже показывает лучше результат
* Предобработка признаков проводилась вручную, каждый отдельно, так как были очень грязные данный!

В результате лучшие результаты показали модели RandomForestRegressor (11%), LGBMRegressor (9%) и стекинг моделей, в качестве базовых моделей ( XGBRegressor & RandomForestRegressor) (10%)