Модель для предсказания цены квартиры в г. Магнитогорск

Цель:

Создать модель для предсказания цены квартиры в различных районах Магнитогорска. Модель должна получать на вход параметры квартиры в формате JSON и возвращать цену в тысячах рублей. Метрика модели – MAE.

Выбор и получение исходных данных:

Источник данных – сайт недвижимости www.citystar.ru. Данные с сайта загружены парсером написанном на python, и далее отправлены в локальную реляционную базу данных под управлением Postgres. Часть записей содержат пропуски и оформлены некорректно.

Описание доступных данных (*формат*):

1) Дата подачи и дата обновления (*datetime*) – дата обновления наиболее актуальная дата по объявлению
2) Текстовое примечание (*string*) – описание квартиры в свободном стиле
3) Тип квартиры (*string*) – категория, описывающая количество комнат (пример: однокомнатная, двухкомнатная и т.д.)
4) Район (*string*) – категория, содержащая название района (пример: Орджоникидзевский)
5) Улица (*string*) – категория, в которой находится название улиц (пример: Карла Маркса)
6) Дом (*string*) – номер дома (пример: 3, 9, 57/1)
7) Этаж (*string*) – номер этажа квартиры и этажность здания через знак “/” (пример: 4/9)
8) Планировка (*string*) – описание планировки квартиры, судя по всему, в свободной форме (пример: малосемейка, хрущевка)
9) Общая площадь (*float*) – площадь всей квартиры в квадратных метрах (пример: 52.5)
10) Жилая площадь (*float*) – площадь жилой части квартиры в квадратных метрах (пример: 27.0)
11) Площадь кухни (*float*) – размер кухни в квадратных метрах (пример: 9.0)
12) Цена (*float*) – цена квартиры в тысячах рублей (пример: 3500)
13) Также доступны имя продавца, телефон, email, количество просмотров – но эти данные для модели не нужны

Выбор метода решения:

Модель. В данных есть категориальные признаки. Для моделирования в таких данных лучше использовать CatBoostRegressor, но и другие модели нужно сделать для сравнения. Итоговая модель будет обучена с использованием optuna, на 90% всей выборки, 10% останутся для теста. Анализ модели при помощи встроенного метода feature_importance и библиотеки shap.

Подготовка данных и выбор признаков. Из даты обновления извлеку месяц и год, сами даты удалю. Текстовое примечание удалю. Хотя в нем может содержаться полезная информация, оценить «привлекательность» описания сложная задача и не входит в рамки работы. Тип квартиры, район – оставлю. Улица и дом – удалю, из них можно извлечь информацию, но доступная выборка небольшая, и если в каком-либо доме будет единственная дорогая квартира, то результат предсказания квартиры в этом доме будет смещен. Этаж и этажность здания вынесу в два отдельных столбца. Планировка описана в свободной форме, и этот признак скорее всего коллинеарен другим признакам, может трактоваться по-разному и нести разный смысл. Лучше его не использовать. Было бы здорово найти название проекта дома, год постройки, год капитального ремонта, но в этой работе такую задачу не ставлю. Площади: жилая, общая, кухня – крайне важны. Цена будет целевым признаком.

Описание алгоритма решения:

1. Создание парсера, сбор данных, хранение в локальной базе данных. Использую postgres, pgAdmin, sqlalchemy, pandas, requests.
2. Подготовка данных. Итоговые признаки и преобразования перечислены выше. Создам серию обработок, оформлю в виде функций. На выходе получу датасет в виде датафрейма pandas, разобью его на трейн/тест в соотношении 90/10 %. Масштабирование в данной работе не нужно. Год, месяц, район – перекодирую one-hot encoder. Количество комнат переделаю в int тип.
3. Анализ данных. Нужно ответить на главный вопрос, репрезентативны ли данные и подходят они для моделирования процесса? Для этого для этого сделаю гистограммы распределения по годам, ценам, районам.
4. Создание прототипов моделей. Обучу на трейн сете модели: случайный лес, регрессию с регуляризацией Lasso, CatBoostRegressor. Хотя Catboost сейчас считаю лучшей моделью, для проверки адекватности необходимо сравнить его с чем-нибудь другим. Модели обучу с кросс-валидацией по небольшому выбору гиперпараметров, соберу метрики.
5. Создам сравнительную таблицу по моделям/метрикам, найду лучшую. Лучшую модель обучу при помощи поиска гиперпараметров optuna. Обучу итоговую модель.
6. Анализ модели. На тестовых данных проверю метрику MAE, для оценки признаков использую shap и feature_importance.
7. Создание локального микросервиса Flask. Сервис будет получать необработанные данные (в том виде, в котором они есть на сайте), обрабатывать данные для работы модели, и используя готовую модель делать предсказания по цене. Обращение к сервису в виде Get запроса, с данными в формате JSON. Сделать 3 пробных запроса.

Описание модели

Получил 6 пробный моделей и их метрики (целевая - MAE) на обучающем наборе.

	MAE	RMSE	R2	Predict_time
Catboost	612.928	1013.763	0.533	0.004
ElasticNet	643.382	1023.505	0.524	0.006
Ridge	645.182	1022.403	0.525	0.003
Lasso	645.341	1019.011	0.528	0.006
Случайный лес	650.865	1062.776	0.486	0.237
LGBMRegressor	662.822	1087.466	0.460	0.003

Выбрал Catboost для дальнейшего развития. Использовал optuna для поиска гиперпараметров. Получил гиперпараметры: {'learning_rate': 0.02134409209885269, 'depth': 7, 'subsample': 0.6378176007541093, 'colsample_bylevel': 0.9996306493875371, 'min_data_in_leaf': 98} и метрику MAE: 564.12.

Использую эти гиперпараметры для обучения модели и получаю итоговую модель.

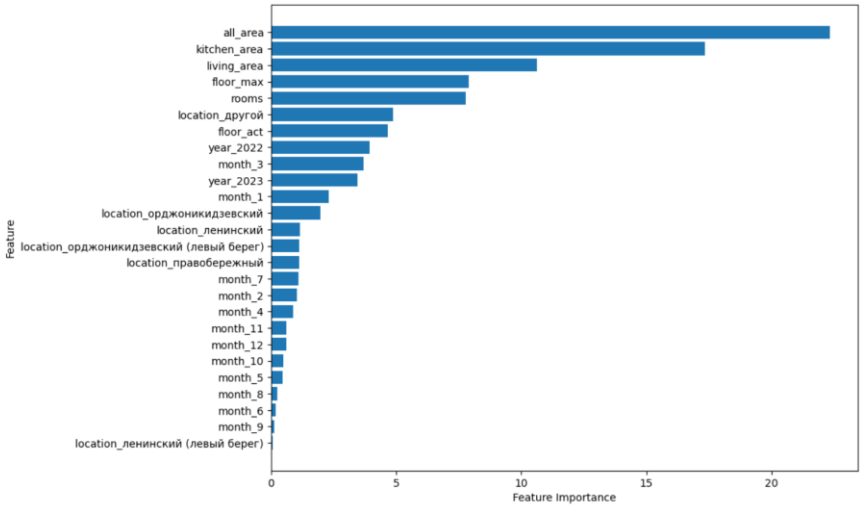
Описание качества модели/полученных результатов

MAE на тестовой выборке 272.96 тысяч рублей.

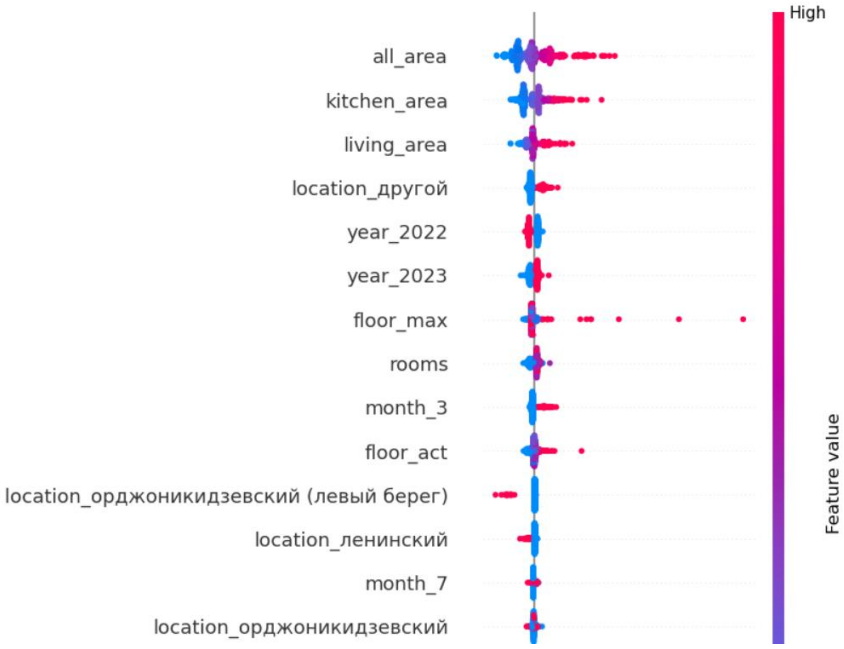
Микросервис на Flask работает локально, обращается к итоговой модели. Метод запроса – Get в формате JSON. По запросу возвращает предсказанную цену. На вход идут сырые данные в том виде в котором они есть на сайте, однако поля должны быть заполнены полностью.

Описание результатов тестирования модели

Самое интересное! Анализируем признаки:



Из важности признаков самые важные для предсказания – площади квартиры и комнаты. Но стоит отметить важные для модели признаки этажности здания и этаж квартиры, и месяц/год продажи. Только потом идут определенные районы. Рассмотрим эти признаки в shap.



Чем больше площадь, тем больше цена, логично.

Район (location), год, месяц – были бинарно кодированы, поэтому красная точка обозначает присутствие этого признака.

Видно, что в 2023 году цены поднялись на все квартиры.

Самое лучшее время для продажи – март, в течении этого месяца цены выше.

Более высокий этаж квартиры повышает цену, общая этажность дома тоже влияет, но не всегда.

На левом берегу квартиры дешевле, как и в ленинском районе. Районы, указанные как «Другие» (отличные от пять наиболее популярных), дороже.

Выводы

1) Модель построена, работает как сервис, возможно запрашивать оценочную цену квартиры

2) Квартиры подорожали во времени, локация, этаж, месяц продажи – важные признаки

Если нужно продолжать работу над моделью, то я бы сделал следующие шаги:

- Собрать бы больше данных. Модель можно обучить на большей выборке и это точно улучшить качество.

- Сервис для предсказания работает с запросами, в которых заполнены все данные. Возможно сделать его более устойчивым к незаполненным полям.

- Собрать информацию о доме, например год постройки, капитального ремонта

- Поискать сторонние решения и проанализировать их, думаю много можно найти полезного для доработки модели

- Совместить несколько моделей

- Создать больше признаков, например признак – «первый этаж»