Отчёт о проекте

Выполнил: Богданов Лев Александрович

Были использованы следующие библиотеки:
- pandas
- numpy
- matplotlib
- seaborn
- statsmodels
- sklearn

Предобработка

В данных о транзакциях были обнаружены пропуски в городах и ошибки в значениях цены и количества товара. Также обнаружены несоответствия правилам ценообразования (цены менялись чаще, чем раз в 3 дня и больше, чем на 1 золотой за раз).
- основываясь на том, что аномалии в данных разряжены, мы упорядочили данные по дате, и заполнили пропуски в городах и отрицательные значения в цене соседними значениями;
- удалены изменения цены, не соответствующие правилам (если новая цена держалась меньше 3 дней или изменилась больше чем на 1)
- основываясь на значениях аномалий, отрицательные значения в количестве товара были заменены на такие же положительные.

Данные о ценах конкурентов оказались в порядке. У некоторых групп товаров нехватало последних дней. Эти данные были заполнены предыдущими значениями, чтобы привести все даты к одному диапазону. Правила ценообразования не нарушались.

Данных о себестоимости оказалось крайне мало для прогнозирования. Логично предположить, что поставщик обновляет цены не так часто, и у нас есть данные только о фактах изменения этих цен. Диапазон дат совпадал с диапазоном основных данных, поэтому мы расширили данные, заполнив промежутки предыдущими значениями.

В данных о погоде было много наблюдений без флага погодных условий. Такие наблюдения были вынесены в отдельную категорию normal.

Анализ данных

Данные из каждого датафрейма были сгруппированы по товарам и городам и построены графики для выявления закономерностей в данных.
Определён восходящий тренд как в наших ценах, так и в себестоимости и ценах конкурентов. Присутствует инфляция.

В Анор Лондо и Фальконии присутствуют сильные просадки в ценах, однако, последние даты говорят о том, что в ближайшее время планируется рост.

Присутствуют сезонные колебания цен. Сезонность составляет 1 год. Диапазон месяцев отличается от города к городу.

Показатель количества товара колеблется в районе одних и тех же значений. Просадка была недавно, на ближайшие даты намечается рост. Присутствует недельная сезонность.

Рост и колебания цен конкурентов совпадают с нашими.

В ходе анализа, было выполнено ресемплирование наших данных по одному дню. Получен единый датафрейм.

Подготовка признаков и обучение моделей

Для решения задачи было решено использовать следующий подход:

- спрогнозировать цены конкурентов на каждый товар в каждом городе
- спрогнозировать объем и себестоимость товаров
- спрогнозировать наши цены
- сравнить полученные прогнозы с ценами конкурентов и внести правки при необходимости
- привести прогнозы в соответствие с правилами ценообразования

Для построения большого количества прогнозов была выбрана модель линейной регрессии. Она показала себя лучше бустингов и деревянных моделей, и намного быстрее.

В качестве метрики выбрана средняя абсолютная ошибка. Такие данные удобно интерпретировать.

Наш единый датафрейм нарезается на списки таблиц по каждому целевому признаку, таблицы упорядочиваются по дате. Затем генерируются признаки (номер дня, недели, месяца, год, лаги) и модель тренируется на кросс-валидации (фолды задаём через TimeSeriesSplit). В коде можно раскомментить графики отрисовки результатов (прогнозов много, решил убрать, чтобы не загромождать выводы). Лучшая модель прогоняется на всей выборке и сохраняется.

Модели хорошо обучаются, ошибки в ценах в районе нескольких серебрянных монет (0.02 - 0.08).

В каждую таблицу добавляются пустые строки до заданного горизонта прогнозирования (90 дней). Значения лагов мы получаем на день вперёд, после этого получаем предсказание, записываем его, и обновляем признаки для следующего дня.

После получения всех предсказаний, мы проверяем их на соответствие правилам ценообразования и, при необходимости, вносим правки.

Была выбрана стратегия сравнения с минимальной ценой среди конкурентов. Это позволит держаться посередине рынка и не отпугивать клиентов.

- если спрогнозированная цена оказалась меньше минимальной, ставим + 10% от минимальной
- если спрогнозированная цена больше минимальной на 20%, ставим + 15%

В итоге все данные объёдиняются в один датафрейм, из которого мы выделяем нужный нам срез, и сохраняем его в .parquet.

Варианты дальнейшей доработки:

- дооформить код из этапа анализа, обернуть в функции ресемплирование и визуализацию
- более детально проанализировать объём товара и построить для его прогнозирования отдельную модель (с учётом скользящего среднего и отдельных месяцев)
- построить модель для прогнозирования данных о погоде, чтобы использовать их в качестве признаков

- пересмотреть ценообразование с учётом себестоимости и формулы прибыли
    - сейчас цена основывается только на наших прогнозах после анализа исторических данных, и на ценах конкурентов
    - даже если учесть формулу и максимизировать прибыль, задача всё равно будет решаться выставлением цены относительно цен конкурентов, можно придумать другой способ, если получится собрать больше данных
- обернуть в функции этап предобработки данных
- написать скрипт, который будет выполнять предобработку и получать предсказания из такого же набора данных за любой период