Отчет по тестовому заданию

(исп: Ольховский Н.А.)

Введение

В качестве испытания на должность Junior Data Scientist была поставлена задача разработать ML-модель прогнозирования спроса на товар или категорию товаров из датасета Dominicks Dataset. Для разработки модели была выбрана категория обезбаливающих лекарств. В ходе работы был произведен экспресс-анализ структуры данных. Для прогнозирования было выбрано число проданных упаковок конкретного лекарства за неделю (колонка `move` таблицы `ana_sales`). Был сформирован базовый датасет и обучена базовая модель на основе CatBoost. Модель продемонстрировала неудовлетворительную скорость обучения, что препятствовало проведению экспериментов на имеющихся вычислительных мощностях. Поэтому был проведен ряд экспериментов, направленных на повышение скорости обучения. В результате было принято решение в дальнейшей работе использовать LightGBM.

После разработки ML-модели с удовлетворительной скоростью обучения, была произведена визуализация понедельного объема продаж лекарств с целью выявления трендов и оценки влияния праздничных дней на уровень продаж. В ML-модель были добавлены признаки праздников и внесены дополнительные изменения. Полученная модель продемонстрировала недостаточную точность для построения графиков эластичности спроса. Полученный результат указал на необходимость дополнительной предобработки данных и возможные пути улучшения точности модели. В это точке работа была приостановлена в связи с исчерпанием времени, выделенного на выполнение задания. Исследование может быть продолжено, если заказчик согласится следующий отчет принять после 10 января 2025.

К настоящему отчету прилагаются исходные файлы блокнотов Jupyter Notebook и другие материалы, размещенные в репозитории nikolay-olkhovsky/demo_dominiks. Структура репозитория следующая:

- `/data` -- папка для размещения рабочих датасетов
  - `/data/raw` -- исходные данные для исследования
- `/model` -- папка с обученными моделями
  - `00-baseline.cbm` -- базовая модель CatBoost
  - `01-NotNaN.cbm` -- модель CatBoost на основе данных с удаленными пропусками
  - `02-lightgbm-model.txt` -- базовая модель LightGBM
  - `03-holiday-model.txt` -- модель LightGBM на основе данных с указанием праздничных недель. Из данных удалены пропуски, а также строки с ценами меньше `0.01`
- `/notebook` - папка с исходными блокнотами, содержащими проделанную по исследованию работу
  - `01-data-info.ipynb` -- просмотр структуры данных и их основных характеристик
  - `02-baseline-dataset.ipynb` -- создание и сохранение базового датасета
  - `03-baseline-model.ipynb` -- создание и испытание базовой модели

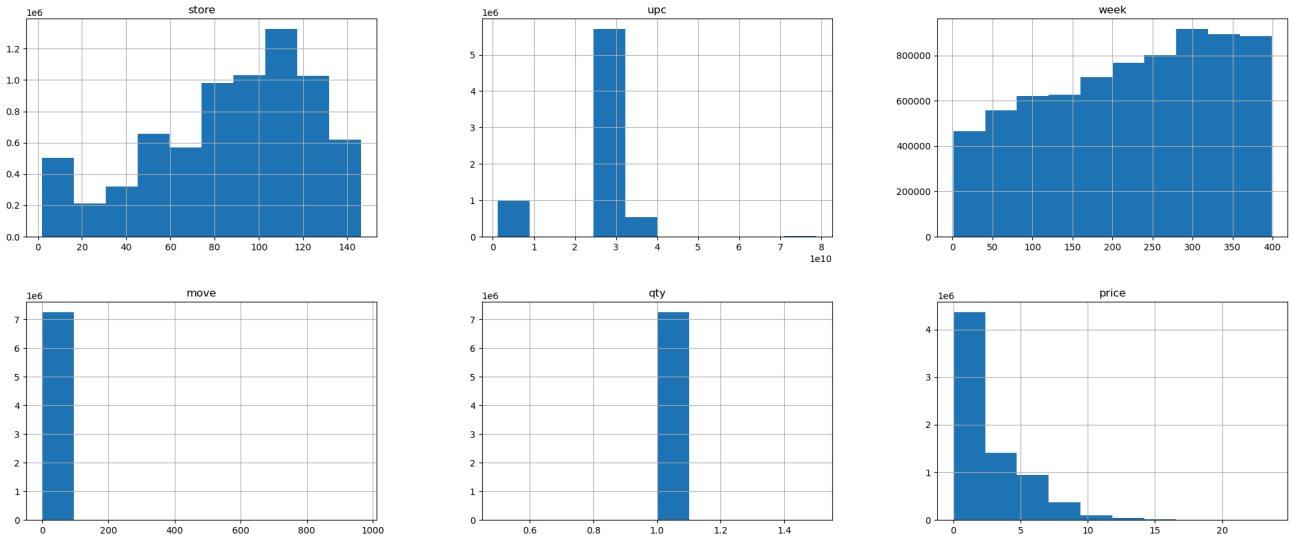
    - `04-quick-model.ipynb` -- попытка построить меньшую модель и отказаться от информации `demografic_data`
    - `05-NonNaN-model.ipynb` -- попытка выиграть в скорости за счет уменьшения датасета
    - `06-lightgbm-model.ipynb` -- построение более быстообучаемой модели на основе LightGBM
    - `07-lightgbm-NotNaN-model.ipynb` -- эксперимент по улучшению точности модели за счет более правильной обработки категориальных признаков
    - `08-data-visualization.ipynb` -- визуализация продаж по неделям, оценка влияния праздников
    - `09-holiday-model.ipynb` -- эксперимент по построению модели, учитывающей влияние праздников на продажи
    - `10-report.ipynb` -- отчет по проделанной работе
- `/reference` -- папка со справочными материалами
- `/src` -- папка с Python-скриптами

Построение базовой модели

Оценка структуры данных

Для построения прогноза спроса по анальгетикам выбраны следующие данные:

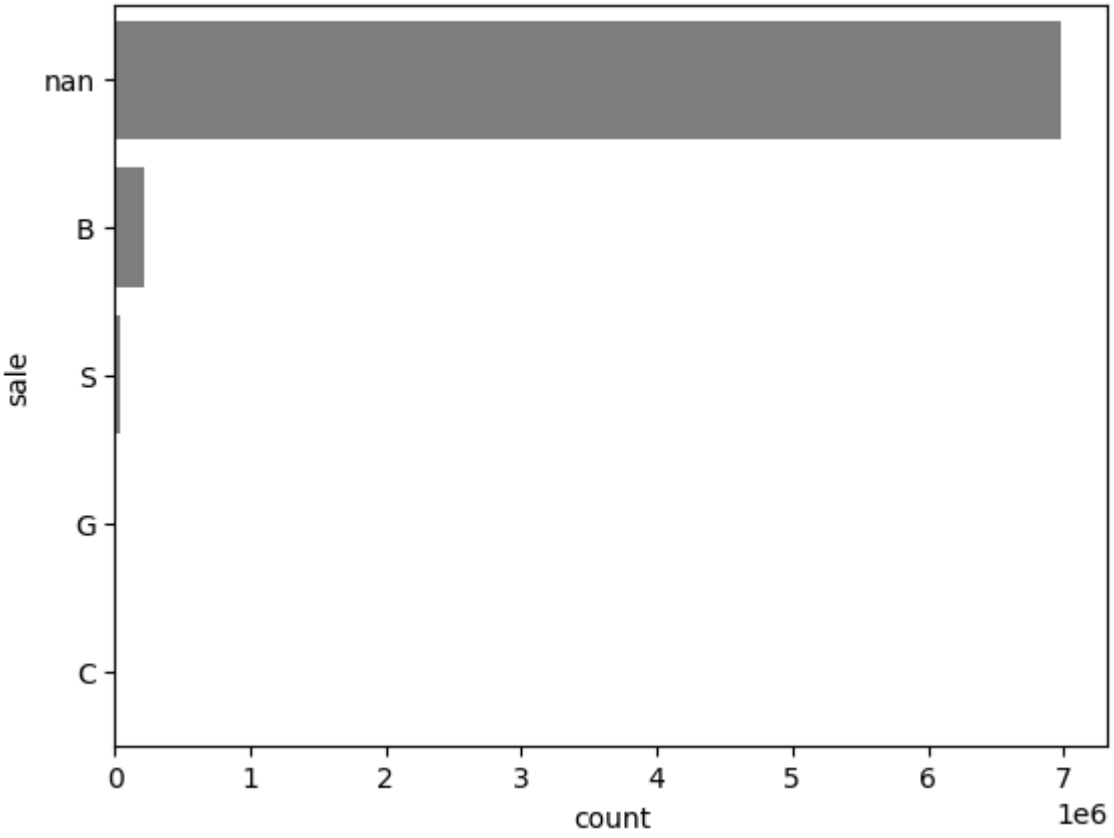
- `/data/raw`
    - `ana_sales_data.parquet` -- таблица с понедельной информацией о продажах лекарств
    - `ana_upc.parquet` -- таблица с описанием продаваемых позиций
    - `demografic_data.parquet` -- таблица с информацией о населении в районах расположения магазинов



Данные по продажам пива и сигарет принято решение не использовать в первых версиях моделей. Они могут хорошо коррелировать с продажами обезбаливающих, но проверка этой гипотезы выходит за рамки базовой модели. Чуть более полный осмотр данных приведен в блокноте `01-data-info.ipynb`. Наиболее существенные выводы следующие:

1. Цена не распределена нормально. Много цен с нулевым значением. Вершина распределения цен без учета нулевых значений смещена влево (матожидание смещено в отрицательном направлении).

2. Числе проданных упаковок не распределено нормально. Много нулевых значений.
3. Признак `qty` не несет информации (для данной таблицы), так как всегда равен единице.
4. Все категориальные признаки распределены неравномерно. Для нас наиболее важны признаки магазина (`store`), конкретной позиции (`upc`) и маркетинговых акций (`sale`)
5. Признак `store` позволяет присоединить демографическую информацию
6. Признак `upc` позволяет присоединить информацию о продаваемых позициях. Содержит две существенные колонки: описание лекарства (`descrip`) и вид/размер упаковки (`size`). Оба этих признака сильно влияют на спрос, но чтобы разложить их на значимые части, требуется существенная работа (например, выделить из нестандартизированных описаний бренд производителя и марку лекарства). Без предобработки фактически мы имеем категориалный признак размером в `641` значение (число уникальных `upc` в `ana_sales`)
7. В таблицах `ana_sales` и `ana_upc` пропусков нет. Таблица продаж содержит более `7 млн` строк. Таблица позиций содержит `641` лекарство.
8. В таблице `demografic_data` содержит не только информация о демографии, но и эконометрическая. На `107` строк есть `10` строк, полностью из NaN. И `12` строк, в которых есть информация о населении, но пропущена статистика о ближайших складах и магазинах.



Создание базового датасета

Для создания базового датасета были выполнены следующие действия:

1. В таблице продаж заменить `None` в колонке маркетинговых акций, удалить столбец `qty` (конкретно для этого вида товаров он всегда равен `1`).
   Примечание: поле None было заменено, исходя из предположения, что для дальнейшей работы во избежание ошибок скриптов лучше, чтобы эти ячейки содержали некоторое значение, и не интерпретировались как пустые.
2. В таблице `ana_upc` убрать столбцы `case` и `nitem`

3. В таблице `demographic` ничего не менять. Гипотезу о том, что исключение NaN улучшит модель оставим на одну из будущих итераций.
4. Все три таблицы смерджить в единый датасет. Так как данных для обучения более чем достаточно, тип объединения таблиц выбран `inner`, чтобы исключить возникновение лишних `None`

Исходный код размещен в блокноте `/notebook/02-baseline-dataset.ipynb`. Датасет сохраняется в файл `/data/baseline_dataset.csv`

Создание базовой модели

Для создания базовой модели была выбрана библиотека CatBoost. Выбор был остановлен на ней, так как при обработке табличных данных себя хорошо зарекомендовал градиентный бустинг, а среди трех наиболее мощных библиотек, реализующих соответствующие алгоритмы (CatBoost, LightGBM и XGBoost), CatBoost обладает наименьшей требовательностью к предварительной обработке данных. Это (к сожалению) подтвердилось в рамках настоящего исследования: базовая модель на основе CatBoost так и осталась наиболее точной, среди испытанных.

Для создания базовой модели были выполнены следующие действия:

1. Из базового датасета были удалены столбцы `com_code`, `upc` и `store`. Как категориальные были отмечены признаки `descrip`, `size` и `sale`. В качестве целевого признака взят `move`.
2. Сформированы выборки и произведено обучение.

Исходный код размещен в блокноте `/notebook/03-baseline-model.ipynb`.
Модель сохраняется в файл `/model/00-baseline.cbm`

Выводы по baseline модели

- **Достигнута точность в 0.758 правильных прогнозов.** Точность подсчитана следующим образом: у ответов модели отбрасывается дробная часть, и считается метрика accuracy с валидационными данными.
- Долго обучается (Intel Xeon E5-2670 v3, 3GHz, 32GB RAM) -> около 25 минут
- Самые важные фичи: price, descrip, week, sale, size

	Feature Id	Importances
0	price	30.487298
1	descrip	23.051027
2	week	12.909588
3	sale	9.456202
4	size	8.400924
5	sstrvol	2.415333
6	ethnic	1.967587
7	hhlarge	1.428075
8	age60	1.414026

	Feature Id	Importances
9	cpwvol5	1.272081
10	hval150	1.244064
11	income	1.124267
12	cpdist5	1.084525
13	educ	1.071258
14	sstrdist	1.053121
15	workwom	0.882992
16	age9	0.737633

Идеи на будущее:

- Добавить данные о праздниках
- Разбить size на две фичи: число и тип (таблетка, унция, миллилитр и т.п.)
- Визуализировать корреляцию/совстречаемость признаков и их влияние на таргет
- Уточнить границы понятия "спрос": сейчас пытаемся предсказать спрос на конкретное количество таблеток для каждой марки лекарства, не факт, что с точки зрения бизнеса это самое важное
- Убрать демографические признаки, либо признаки со значимостью ниже 1.5

Эксперименты на ускорение обучения модели

Главной проблемой первой версии модели стало время обучения. Для эффективной работы требовалось проводить много небольших экспериментов. Нужно было снизить время одного цикла обучения с 20-25 мин до 3-5 минут. Для этого были проверены несколько гипотез.

1. Вместо одной универсальной модели, выучившей все 641 категорию товара (и все размеры упаковок для них), можно обучить модель на одном отдельно взятом лекарстве. Это даст высокую скорость и приемлемые точность.

Достигнутая точность составила 0.455. Предположение: модель потеряла в универсальности и не смогла эффективно оценивать число проданных упаковок.
Исходный код размещен в блокноте `/notebook/04-quick-model.ipynb`.

2. Исключим из обучения демографические данные, так как базовая модель показывает их низкую значимость. Это может ускорить обучение без потери точности.

Скорость обучения не выросла. Обучение было остановлено ради экономии времени.
Исходный код размещен в блокноте `/notebook/04-quick-model.ipynb`.

3. Если исключить из датасета все строки с NaN, датасет станет меньше. Это может увеличить скорость обучения.

Датасет стал меньше на ~250 000 строк. **Скорость обучения не изменилась.** В момент проведения эксперимента была возможность по времени довести обучение до конца. Достигнутая точность составила `0.757`.

Исходный код размещен в блокноте `/notebook/05-NotNaN-model.ipynb`.
Модель сохранена в файле `/model/01-NotNaN.cbm`

4. Если обучать модель на половине датасета, это может увеличить скорость обучения.

Было проведено испытание с 50% строк датасета и с 15% строк. И в том, и в другом случае **скорость обучения существенно не уменьшилась**.
Исходный код размещен в блокноте `/notebook/05-NotNaN-model.ipynb`.

В этой точек был сделан вывод о том, что ускорить CatBoost без подбора гиперпараметров не получится. Было принято решение испытать LightGBM.

5. Модель на основе LightGBM даст существенный прирост скорости с сохранением сопоставимой точности.

При попытке загрузить в модель такие же обучающие данный, как в CatBoost, были получены предупреждения модели о том, что число уникальных значений в некоторых категориальных признаках превышают допустимое количество. Поэтому было принято решение закодировать столбцы `descrip`, `size` и `sale` соответствующими целочисленными значениями и не отмечать их как категориальные. В одном из руководств по LightGBM было рекомендовано именно так готовить high cardinality features перед подачей в модель.
Скорость обучения составила ~5 мин. Достигнутая точность: 0.763
Исходный код размещен в блокноте `/notebook/06-lightgbm-model.ipynb`.
Модель сохранена в файле `/model/02-lightgbm-model.txt`

6. Можно повысить точность LightGBM, если обучить ее на датасете без NaN.

По аналогии с проверкой Гипотезы 3 из датасета были удалены все строки, содержащие NaN. По аналогии с Гипотезой 5 признаки `descrip`, `size` и `sale` были представлены столбцами с целочисленными значениями.
Достигнутая точность: 0.761
Исходный код размещен в блокноте `/notebook/06-lightgbm-model.ipynb`.

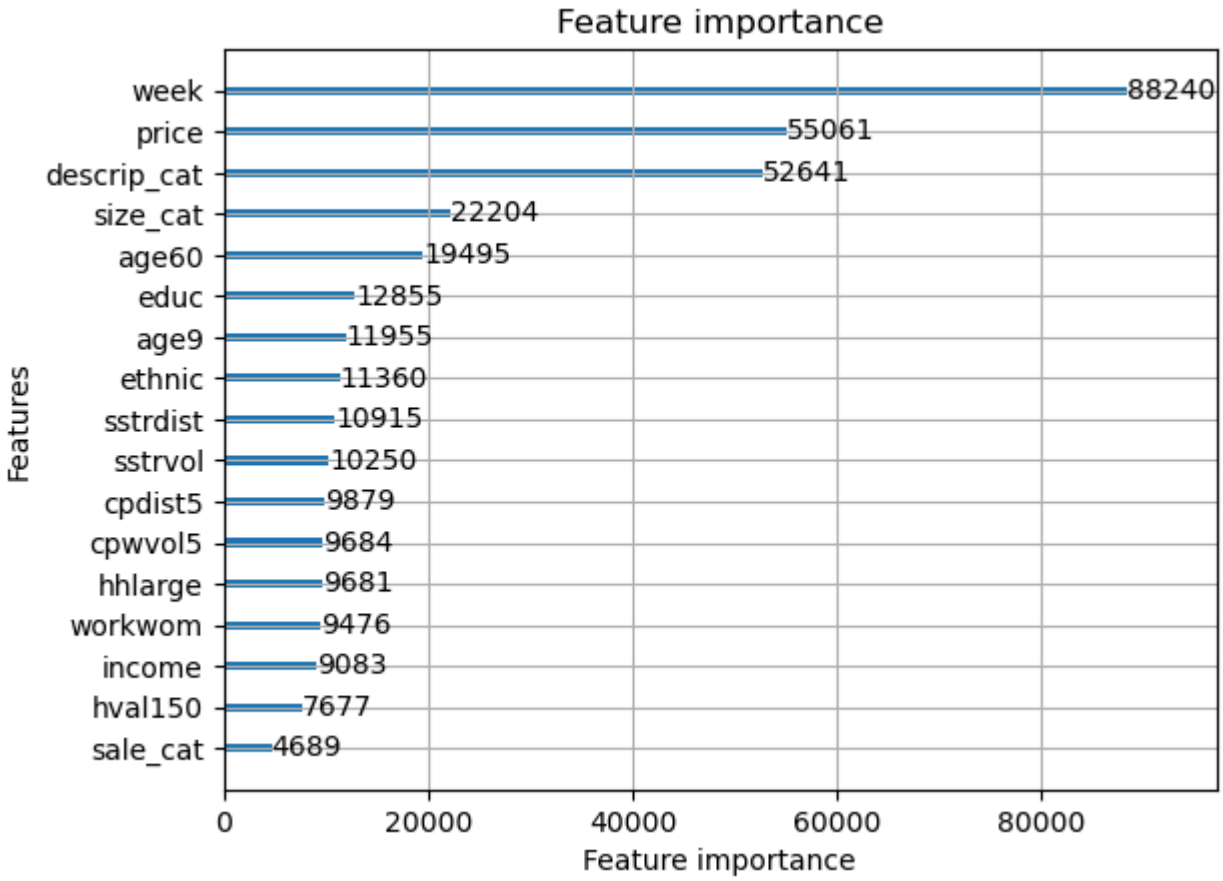
Выводы

При разработке модели `/notebook/06-lightgbm-model.ipynb` достигнута приемлемая скорость обучения (~5 мин) и несколько увеличена точность (0.763). CatBoost может по умолчанию обрабатывать категориальные признаки с большим числом уникальных значений, но без дополнительной настройки деляет это медленно.

Исключение строк, содержащих NaN, или данных о демографии покупателей, не дает существенного прироста скорости.

Интересная особенность:

- CatBoost выделил как самые значимые признаки `price`, `descrip`, `week` (самый значимый - цена)
- LightGBM сформировал несколько иную тройку `week`, `price`, `descrip` (самый значимый - номер недели, причем с большим отрывом)



Модель holiday

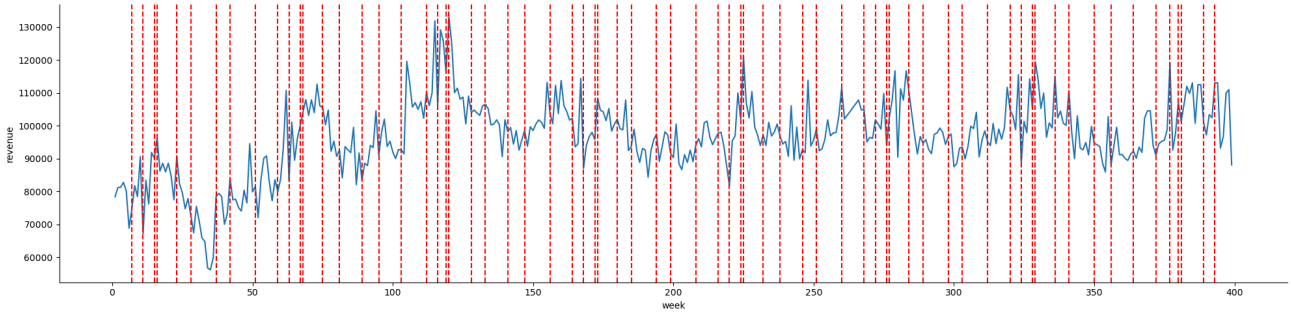
Исходные данные визуализацией сохранены в блокноте `/notebook/08-data-visualization.ipynb`.
Построение модели с учетом новых признаков сохранено в блокноте `/notebook/09-holiday-model.ipynb`.

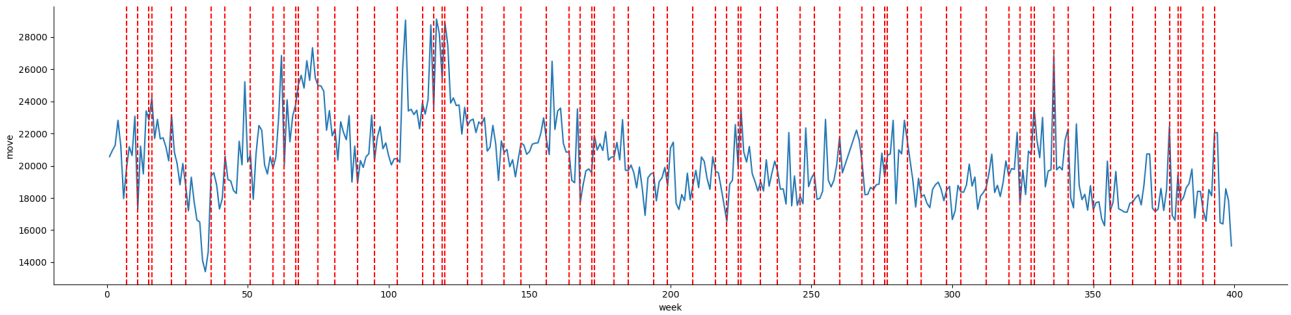
Исследование выручки и продаж упаковок по неделям

Поскольку и CatBoost, и LightGBM указали на высокую значимость признаков недели, цены, описания лекарства и его размера, данные признаки были исследованы более подробно. Для исследования зависимости цены от недель был использован `/data/baseline_dataset.csv`. С ним были произведены следующие действия:

1. Добавлен стобец `revenue` (выручка), равный произведению числа проданных упаковок `move` на цену `price`.
2. Для каждого праздника в таблицу продаж был добавлен столбец, содержащий метки 0 или 1.
3. Выручка и число проданных упаковок были сгруппированы по неделям и просуммированы.

Были построены графики выручки и количества проданных упаковок по неделям. На вертикальными красными линиями отмечены праздничные недели





Выводы по суммарной выручке

1. На графике виден тренд с 0 по 130 неделю. И боковик - со 130 недели до конца.
2. На графике хорошо видны сезонные колебания: зимой - высокие продажи с большой волатильностью, летом - более низкие продажи и меньше резких колебаний.
3. Праздники не всегда совпадают с локальными пиками. Имеет смысл рассматривать интервал +-1 неделя от праздника.

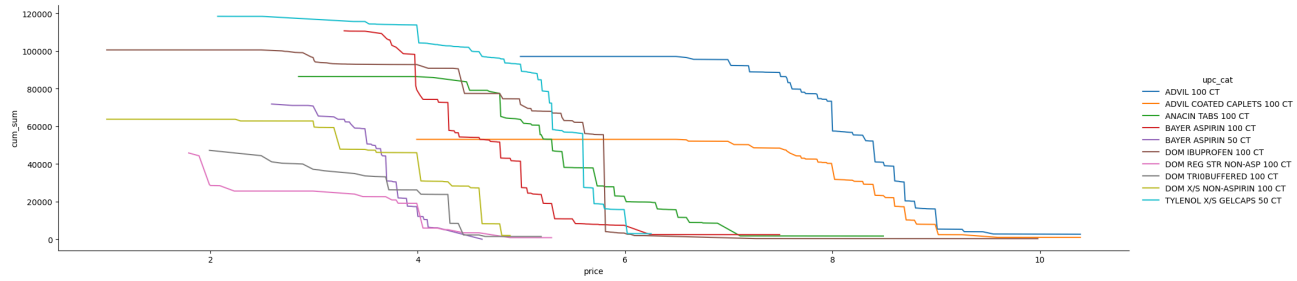
Выводы по продажам упаковок

1. На графике виден восходящий тренд с 0 по 130 неделю. Нисходящий тренд - со 130 недели до 225. Боковик - с 225 до конца. В конце - вероятен слабый нисходящий тренд. С учетом того, что после 130 недели выручка находится в боковике, получается, что сумма, которую люди готовы потратить на анальгетики остается фиксированной, а вот число упаковок потребители покупают меньше.
2. Праздники не всегда совпадают с локальными пиками. **Имеет смысл рассматривать интервал +-1 неделя от праздника.**

Исследование продаж по отдельным позициям лекарств

Из `/data/baseline_dataset.csv` были выбраны 10 наименований лекарств, о которых в таблице продаж наибольшее число записей. С данными были выполнены следующие действия:

1. Выбраны столбцы `upc`, `descrip`, `size`, `move`, `price`, `revenue`. Удалены строки, содержащие NaN.
2. Удалены строки с числом проданных упаковок менее 1.
3. Столбцы `descrip` и `size` объединены в `upc_name`.
4. Данные были сгруппированы по наименованиям лекарств (признак `upc`). Для каждого полученного наименования данные были сгруппированы по значению цены (признак `price`). Для каждой цены были просуммированы число проданных упаковок и выручка, добавлена кумулятивная сумма от самой высокой цены к самой низкой.



Выводы по распределению продаж

Для каждого вида лекарств получилась функция распределения числа проданных упаковок в зависимости от цены. С учетом того, что фактически мы имеем дело с дискретными значениями цены, видно, что для каждого лекарства число проданных упаковок соответствует нормальному распределению.

Построение модели с новыми признаками

В модель регрессии было принято решение внести три значимых изменения:

1. Прогнозировать не число проданных упаковок `move`, а выручку `revenue`. Это в дальнейшем должно упростить исследование эластичности спроса, так как спрос мы сможем оценивать не в упаковках, а в деньгах.
2. Для каждого праздника добавлены три столбца, кодирующих через 0 и 1 признаки недели перед праздником, самого праздника и недели после праздника (например, `before_christmas`, `christmas`, `after_christmas`)
3. Из датасета удалены не только строки с NaN, но и строки, в которых цена <0.01. То есть были оставлены только данные, значимые для вычисления итоговой выручки.

По результатам обучения модели **MSE составила 81.1** Причем дополнительный эксперимент по обучению модели на предсказание числа проданных упаковок (таргет `move`) подтвердил этот результат. Полученная **точность составила 0.434**.

Была проведена серия экспресс-испытаний по исключению простой ошибки в формировании входных данных и настройке модели: число листьев увеличивалось до 2048, скорость обучения уменьшалась и увеличивалась, новые категориальные признаки исключались из входных данных, признак недели `week` исключался из входных данных. К значимым изменениям это не привело.

Поскольку корень из средней квадратичной ошибки модели равен ~9.0, при средней цене на лекарства около 5-6 долларов оценивать эластичность спроса такой моделью невозможно. Для оценки эластичности требуется снизить ошибку до уровня <0.25.

Выводы

При выполнении задания на позицию Junior Data Scientist была разработана ML-модель на базе LightGBM для прогнозирования числа проданных упаковок анальгетиков из Dominicks Dataset. Время обучения модели - **менее 5 минут**. Достигаемая **точность прогнозирования = 0.763** правильных прогнозов.

При построении модели были рассмотрены библиотеки CatBoost и LightGBM. CatBoost оказался более универсальным и позволил получить точность 0.758 "из коробки", но один цикл обучения на имеющемся датасете из 7 млн строк занимал 20-25 мин. LightGBM позволил без сокращения датасета получить чуть более высокий результат при сокращении времени обучения в 5 раз.

Важно! Полученная модель может содержать серьезные ошибки и требует дополнительной проверки с точки зрения надежности. На это указывает следующий факт.

После более подробного исследования данных были сконструированы новые признаки и добавлены в обучающие данные. Одновременно с этим из данных о продажах лекарств были исключены строки, содержащие цены <0.01. Это привело к резкому падению качества прогнозов модели. Есть высокая вероятность, что высокие результаты базовой версии модели строятся не на ее хорошей обобщающей способности, а на статистических перекосах в массиве входных данных.

Возможные ошибки автора работы, приведшие к недостаточно высоким результатам работы holiday-model.

Автор переоценил возможности библиотек градиентного бустинга к обработке входных данных. Было сделано предположение, что несмотря на перекосы в значениях признаков, модель будет относительно нормально обучаться и продемонстрирует улучшение при простой очистке данных и добавлении новых значимых признаков. Автор недооценил влияние на бустинг следующих особенностей датасета:

1. Численные данные не нормализованы.
2. Категориальные признаки распределены крайне неравномерно.
3. В спросе явно присутствуют тренды, сезонные колебания (зима-лето) и сезонные пики, обусловленные праздниками.
4. Описание лекарств является одним огромным категориальным признаком, содержащим смесь важных данных (марка лекарства, производитель, вид лекарства: таблетки, капли, гель, порошок).
5. Размер упаковки также является категориальным признаком, содержащим смесь важных данных (количество лекарства, единица измерения)

Рекомендуемые шаги по предварительной обработке данных

1. Удалить строки, содержащие нулевые цены `price` и нулевое число проданных упаковок `move`. Убрать выбросы. Применить трансформации для приведения к форме нормального распределения.
2. Определить параметры трендов, сезонности и праздничных пиков на недельном графике цены. На их основе сконструировать признаки, заменяющие в датасете `week`. Убрать признак `week`
3. Из признака `descrip` выделить в отдельные столбцы производителя лекарства, марку лекарства, допинформацию о лекарстве.
4. Из признака `size` выделить столбцы с количеством лекарства в упаковке и единицей измерения (tablets, OZ, ml и т.д.)
5. Все полученные категориальные признаки сделать равномерными либо путем исключения лишних строк для слишком частых признаков, либо настройков весов, усиливающих влияние более редких признаков на обучение.

6. Оценить распределение значений в данных по демографии. При необходимости применить трансформации, чтобы все привести к нормальному распределению. Добавить признаки, явно обозначающие отсутствие блоков данных в таблице демографии (структура данных в этой таблице позволяет сделать это добавлением всего 2 столбцов).

Заключение для HR-менеджера и Data Scientist'a

Автор благодарит HR-менеджера и Data Scientist'a, рассматривавших резюме и этот отчет по тестовому заданию. Как и было написано в сопроводительном письме, это было интересное путешествие по исследованию предложенных данных, которое увлекло настолько, что автор потратил несколько больше времени на эксперименты, графики и отчет, чем изначально планировал. К сожалению, не все желаемое удалось реализовать. Автору теперь и самому крайне интересно, как будет выглядить график эластичности спроса, построенный хорошо обученной моделью. Когда ML-модель в ходе решения основной задачи по прогнозированию формирует внутри себя представление об абстрактном объекте, который иногда крайне сложно описать формулами и вычислить - в этом есть немного магии и красоты ML.

Испытательное задание, которое вы предоставили, позволило мне выявить несколько критических пробелов в своей подготовке, которые я смогу закрыть на новогодних праздниках. Даже если дело не дойдет до живого собеседования, я благодарен за предоставленную возможность протестировать себя на соответствие запросу реального работодателя.

Я буду еще больше признателен, если у Data Scientist'a найдется 10-15 минут, чтобы связаться со мной и дать обратную связь по сделанным мной выводам в конце отчета. Это сильно сэкономит мне силы и время в дальнейшей подготовке и трудоустройстве.

Наконец, если HR-менеджер и Data Scientist примут решение провести собеседование, я готов со своей стороны утверждать, что такого Juniora у вас еще не было и, вероятно, никогда не будет. По многим причинам.

С уважением,

Н.А. Ольховский