Отчет

2. до 15% баллов за отчет в свободной форме о решении контеста, включающий:
   (a) Краткую формулировку задачи
   (b) Описание итогового решения: как готовились данные, что использовалось в качестве таргета, какой алгоритм обучался и все комментарии, которые могут быть полезны для воспроизведения вашего решения
   (c) Рассказ о подходах, которые вы пробовали, и том, как реализация тех или иных идей сказывалась на качестве вашего решения
   (d) Код вашего итогового решения
   (e) Описание того, как вы оценивали качество при решении контеста: как делали кросс-валидацию, насколько она коррелировала с результатом по leaderboard

(a) Требовалось предсказать продажи на выборке test, модифицируя имеющийся baseline — алгоритм, обучающийся на train выборке. При написании кросс-валидации или придумывании другого критерия для алгоритма, нужно было учесть, что данные зависят во времени и не могут обучаться на будущем.

(b,c)
Были проверены разные регрессоры (модели оставлялись обучаться и предсказывать на ночь, утром наблюдались результаты) такие, как

- RandomForestRegressor,
- GradientBoostingRegressor,
- DesicionTreeRegressor,
- BaggingRegressor,

с различными параметрами, например, num_estimators(порядком меньше 100), max_depth.

В итоге, был выбран RandomForestRegressor, который показывал лучший результат. Для него был сделан перебор параметров (пример финального кода приложу)
Перебор некоторых параметров,

- например, max_features, не дал результата (результат по умолчанию лучше),
- некоторых, например, min_samples_split дал улучшение (сделала равным 3).

Результаты, полученные на RandomForestRegressor, улучшила преобразованием feature item_id. Использовала OneHotEncoder для преобразования item_id в совокупность бинарных признаков (1 — это нужный номер товара, 0 — нет).

(d) Код: Baseline_RF_5_ohe.ipynb, результаты: baseline_submission_ohe_best.tsv

(e) Изначально была написана метрика, которая делила выборку на какое-то количество частей, по тем, что находятся раньше во времени обучалась, по тем, что находятся позже предсказывала, считала SMAPE для нескольких случаев, и выдавала среднее арифметическое.

Впоследствии выборка test была разделена на 12 частей, тесты проводились на последних трёх 1/12ых, а обучение на первых девяти 1/12ых. Использовала np.random.permutation(), чтоб мешать элементы test выборки, и брала какую-то часть из всей перемешанной выборки.

Моя оценка качества показывала примерно на 1-2 SMAPE хуже, чем в leaderboard. Например, 25.08 при 23.13 в leaderboard. Подробней можно посмотреть в файле analys_min_smape.ipynb.