Первоначальная очистка данных:

1. Удаляем данные за 15 число, а также те, что выходят за рамки месяца.
2. Столбец fulldate содержит даты, которые не совпадают с ts, причем они соответствуют дубликатам записей, судя по одинаковым значениям lac, cid, ts – удаляем.
3. ID из разных подгрупп по условию не должны пересекаться, так как относятся к разным периодам времени и способу идентификации, поэтому найдем пересечение групп ID и посмотрим в какой группе они в основном проявляли активность – как оказалось во второй, поэтому удаляем строки с ними из первой половины месяца.
4. По итогу количество ID оказалось не равным, что говорит о том, что не все пользователи проявляли активность в течении всего месяца, а значит не имеют пары.
5. Также удалим ошибочные записи с эталонными ID попавшие не в свою группу.

Подготовка обучающего и предсказательного датасета:

1. Считаем для каждого юзера количество действий по каждой категории фичей lac и cid, а также общее количество действий, и долю действий по каждой категории фичей lac и cid.
2. Далее у каждого пользователя определяем топ-20 категорий фичей lac и cid, проанализировав самых активные эталонные пары можно отметить, что большая часть действий и пересечений находится в этом диапазоне рангов.
3. После анализа взаимосвязи распределения времени активности внутри дня и по дням недели, можно сделать следующие выводы:
    - Дисперсия распределения времени активности внутри дня юзера слабо меняется для эталонных пользователей - но сильно отличается между разными юзерами;
    - Мода активности по дням недели особенно сильно разнится у разных юзеров - можно усилить эффект экспоненцированием и в дальнейшем вычислить дельту.
4. Генерируем все возможные пары ID, которые гарантированно относятся к False классу (берем случайную вырезку в 0.05% для того чтобы не было слишком большого дисбаланса классов - в дальнейшем будем балансировать, но все равно это скажется на полноте при предсказании класса True), путем кросс-джойна эталонный ID одной группы с неизвестными другой и наоборот, конкатенируем, True и False пары, джойним к этим парам метрики из пункта 1,2,3.
5. Для всех этих пар рассчитываем следующие метрики, говорящие о различиях в паттерне поведения юзеров:
    - долю совпадений в топ-20 по использованию категориальных фичей cid и lac (в случае если ID соответствуют разным пользователям, метрика будет уменьшаться до нуля, в обратном случае стремиться к единице)
    - дисперсию распределений разности долей использования топ-20 категориальных фичей cid и lac (в случае если ID соответствуют разным пользователям, метрика будет расти, в обратном случае стремиться к нулю)
    - модуль доли количества общей активности относительно максимума в паре (в случае если ID соответствуют разным пользователям, метрика будет расти до 1, в обратном случае стремиться к нулю)
    - модуль доли экспоненты моды распределения дней недели активности относительно максимума в паре (в случае если ID соответствуют разным пользователям, метрика будет расти до 1, в обратном случае стремиться к нулю)
    - модуль доли дисперсии распределения часов активности относительно максимума в паре (в случае если ID соответствуют разным пользователям, метрика будет расти до 1, в обратном случае стремиться к нулю).
6. Все старые метрики относящиеся к 1 пользователю удаляем.
7. Создаем аналогично датасет в котором будем предсказывать неизвестные пары, кросс-джойним все одинокие ID и далее по пунктам 1,2,3,5,6.

8. Методом SMOTE балансируем классы в обучающем датасете, путем генерации строк со схожими значениями метрик.

Обучение, валидация и предсказание:

1. Разбиваем на тренировочный и тестовый датасеты наш обучающий датасет.
2. Используя кросс-валидацию с 4 фолдами рассчитываем метрики F1, точность и полноту.
3. Далее обучаем модель и проверяем на тестовом датасете.
4. Проводим обучение на предсказательном датасете – появляются коллизии (один ID состоит сразу в нескольких группах), избавляемся от которых с помощью удаления пар с меньшей вероятностью.

При обучении увидел определенную закономерность:

При увеличении обучающей выборки путем большего сэмпла (до 50%, выше не тянет ноутбук) False класса и соответственно балансировки True растут метрики точности на тренировочном и тестовом датасетах вплоть до 99,9999%, НО в тоже время усиливается скрытое влияние реального дисбаланса классов (искусственное наращивание имеет свои пределы) и от того на предсказании находит гораздо больше False пар.

Итого при сэмпле False класса в 0,05% имеем 1613 пар и значение метрик на уровне 90,23%.

При сэмпле False класса в 50 % имеем около 1400 пар и значение метрик на уровне 99,93%.