Отчет по тестовому заданию

Для решения обеих задач я использовал логистическую регрессию. Предобработка данных заключалась в

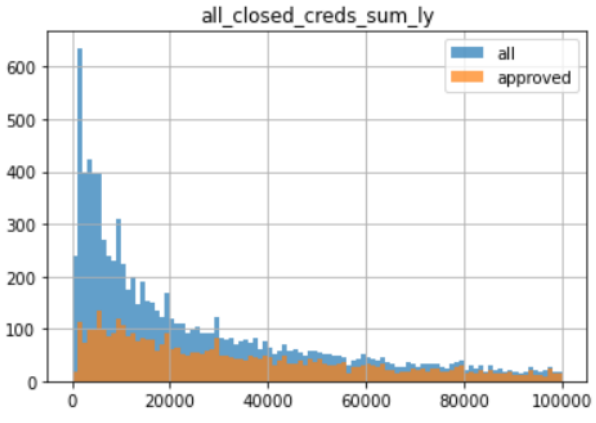
a) преобразовании категориального признака ‘work_code’ в множество бинарных признаков с помощью унитарного кодирования (One-Hot-Encoding) и отбрасывании признака ‘region’
b) отбрасывании признаков 'cred_sum_debt_all_all', 'mfo_closed_count_ly' сильно коррелирующих с другими
c) нормализации признаков с помощью MinMaxScaler или StandardScaler

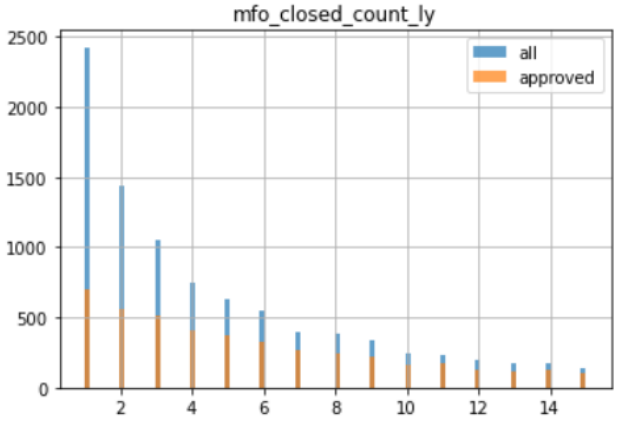
1. Сравнить признаки

a) Признак ‘all_closed_creds_sum_ly’ - сумма закрытых кредитов за последний год

Две выборки имеют одинаковый закон распределения, в начале два распределения сильно отличаются, но постепенно при увеличении аргумента оба сходятся. Это можно объяснить тем, что низкие значение (до 20 000 – 50 000) негативно сказываются на решение об одобрение.

То же самое можно сказать и про признак ‘mfo_closed_count_ly’ - количество закрытых МФО кредитов, взятых за последний год. Чем больше закрытых кредитов, тем выше вероятность одобрения.

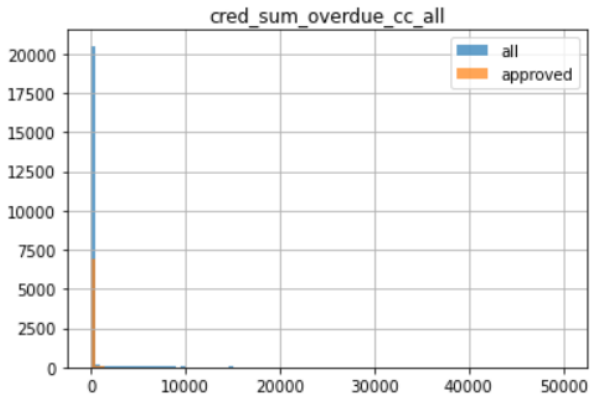


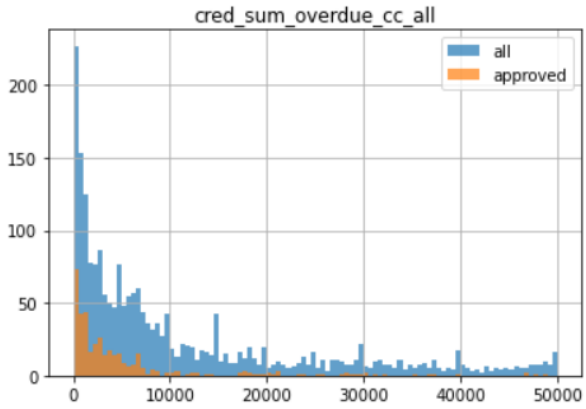


b) Признак ‘cred_sum_overdue_cc_all’ - сумма просрочек по кредитным картам

График слева содержит значение 0, а при построении графика справа нулевые значения не учитывались, чтобы посмотреть на распределение для ненулевых значений. В отличие от прошлого примера, два распределения для данного признака сильно отличаются для всех значений аргумента. Если в начале доля одобренных кредитов с конкретным значением данного относительного всех заявок с данным признаком была близка к 1/3, то с увеличением аргумента

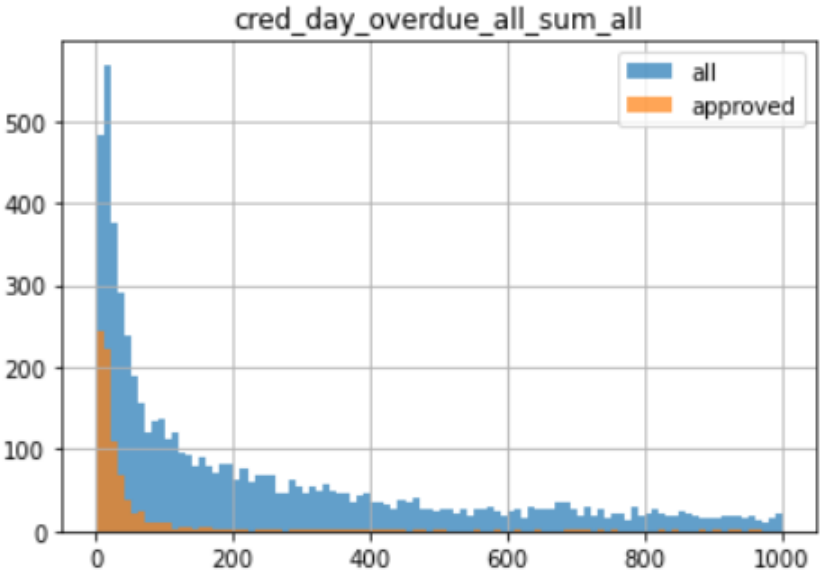
быстро падает до нуля. Таким образом, если при небольших по длительности просрочках еще есть вероятность получить новый кредит, то при больших значениях вероятность близка к нулю.





c) Подобная ситуация с признаком ‘cred_day_overdue_all_sum_all’ - суммарное количество дней просрочки текущих активных кредитов.

Только в данном случае частота одобрений для конкретного значения аргумента падает более стремительно. То есть просрочка активных кредитов более пагубно влияет на решение, чем просрочка по кредитным картам. При построение графика для этого признака нулевое значение также не учитывалось.



2. Объяснить, почему одна модель получилась лучше другой по целевой метрике

Значения метрики для двух моделей оказались равными **0.186** и **0.167** соответственно. Я могу это объяснить тем, что данных для обучения второй модели намного больше, разброс наблюдений шире, поэтому второй модели легче найти разделяющую полосу между двумя классами.

При этом первое значение зависело от разметки отклоненных заявок и от метода нормализации, и при некоторых разбиениях оказывались равными **0.178**. (Если смотреть на метрику ROC_AUC, то первая модель сильно хуже второй (0.55 и 0.83).

3. Придумать, как можно улучшить целевую метрику - процент плохих в 30% одобренных заявках

Целевая метрика - процент “плохих” среди одобренных заявок (чем ниже данная значение, тем лучше).

Метрика, учитывающая оценку двух моделей:

F-мера из вероятности Pr_1 = P{bad = 1}, рассчитанной первой моделью, и вероятности Pr_2 = P{approved = 0}, рассчитанной второй моделью:

$$F = 2 * Pr\_1 * Pr\_2 / (Pr\_1 + Pr\_2)$$

Такая комбинация дала улучшение метрики до **0.156**

Меняя веса двух вероятностей, можно добиться улучшения целевой метрики. У меня получилось, что при увеличении веса у Pr_2 значение метрики падает. И если максимально увеличить вес второй вероятности (или убрать из знаменателя первое слагаемое, что то же самое), то метрика достигнет значения **0.063**