Министерство образования Российской Федерации
Научно-Исследовательский Университет Высшая Школа Экономики

Построение скоринговой модели

Мирзоева Алина, БЭАД223

Москва, 2024

12 июня 2024 г.

Содержание

1 Abstract

Задача кредитного скоринга - одна из основных задач прикладной статистики и машинного обучения, решение которой позволяет банку или другим финансовым организациям оценить риски при выдаче кредита с целью минимизации убытков. В данном проекте было построено несколько моделей машинного обучения: линейной регрессии с применением WOE-преобразований, а также был применен алгоритм метода случайного леса. Рассмотривались различные значения гиперпараметров моделей и с помощью библиотек sklearn и optuna были подобраны их оптимальные значения. С помощью WOE-преобразований и подбора гиперпараметров основная модель достигла точности определения некредитоспособных заемщиков в 36.8%.

Также в рамках проекта была посчитана ожидаемая прибыль от кредитования в соответствии с предсказанной моделью вероятностью дефолта для каждого заемщика. Был получен порог равный 0.32 вероятности дефолта, при которой заемщик классифицируется как некредитоспособный.

2 Dataset

Анализ данных и построение модели проводились на датасете, содержащем данные о кредитах, выданных финтех-компанией, занимающаяся peer-to-peer кредитованием, LendingClub. В тренировочной выборке содержится 61169 записей, а в валидационной - 60334 записи. Датасет состоит из 21 фичи, описание каждой из них представлено в таблице:

№	Название признака	Описание
1	issue_d	Месяц, в который был выдан кредит
2	purpose	Цель займа, предоставленная в анкете для предоставления кредита
3	addr_state	Штат, указанный заемщиком в заявке на кредит
4	sub_grade	кредитный рейтинг, присвоенный компанией по своим внутренним правила
5	home_ownership	Статус владения жильем, указанный заемщиком при регистрации или полученный из кредитного отчета. Возможные значения: АРЕНДА, ВЛАДЕНИЕ, ИПОТЕКА, ДРУГОЕ
6	emp_title	Должность, указанная заемщиком при подаче заявки на кредит
7	installment	Ежемесячный платеж, который должен выплатить заемщик, если кредит выдан
8	dti	Коэффициент, рассчитанный с использованием общих ежемесячных платежей заемщика по всем долговым обязательствам, деленный на указанный в заявке ежемесячный доход заемщика
9	funded_amnt	Общая сумма кредита
10	annual_inc	Годовой доход, указанный заемщиком при заполнении анкеты
11	emp_length	Стаж работы в годах. Возможные значения находятся в диапазоне от 0 до 10, где 0 означает менее одного года, а 10 означает десять и более лет
12	term	Количество платежей по кредиту. Значения указаны в месяцах и могут принимать значения 36 или 60
13	inq_last_6mths	Количество запросов за последние 6 месяцев (исключая автомобильные и ипотечные запросы)
14	mths_since_recent_inq	Количество месяцев с момента последнего запроса
15	delinq_2yrs	Количество случаев просрочки более чем на 30 дней в кредитной истории заемщика за последние 2 года
16	chargeoff_within_12_mths	Количество списаний в течение 12 месяцев
17	num_accts_ever_120_pd	Количество счетов, просроченных на 120 и более дней
18	num_tl_90g_dpd_24m	Количество счетов, просроченных на 90 и более дней за последние 24 месяца
19	acc_open_past_24mths	Количество открытых счетов за последние 24 месяца
20	avg_cur_bal	Средний текущий баланс всех счетов
21	tot_hi_cred_lim	Общий кредитный лимит
22	delinq_amnt	Сумма задолженности по счетам, по которым заемщик в настоящее время просрочил платежи

Таблица 1: Описание признаков датасета

3 Exploratory data analysis

Нашей целевой переменной является переменная def: в ней содержится решение по выдаче или не выдаче кредита, а точнее будет ли у данного клиента дефолт (значение 1) или нет (значение 0).

Поэтому первым делом посмотрим на распределение результатов в нашей выборке:

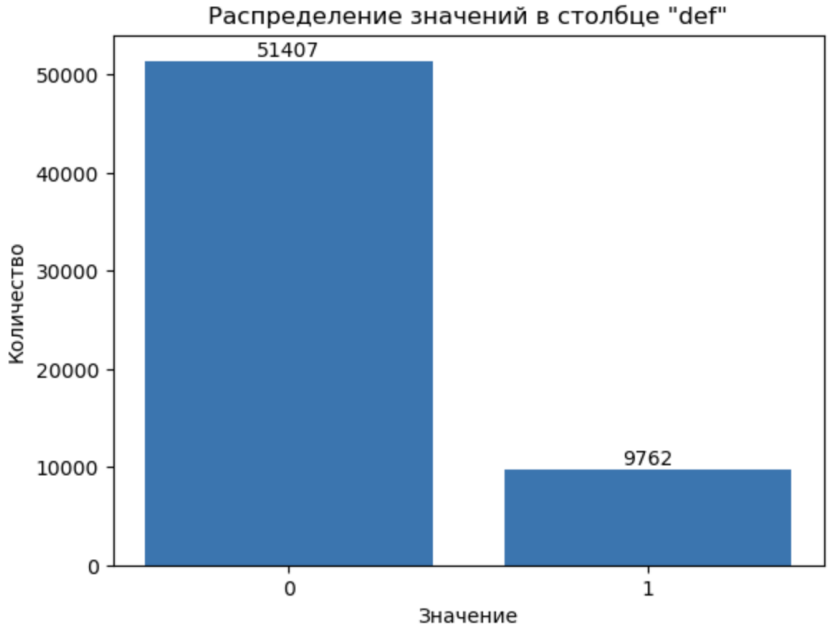


Рис. 1: Распределение целевой переменной

Как мы видим, обучающая выборка достаточно сильно смещена в сторону добросовестных заемщиков, это может ухудшить способность модели правильно классифицировать объекты из класса "1".

Также рассматривалась матрица корреляции переменных и самую высокую корреляцию продемонстрировали пара признаков installment и funded_amnt

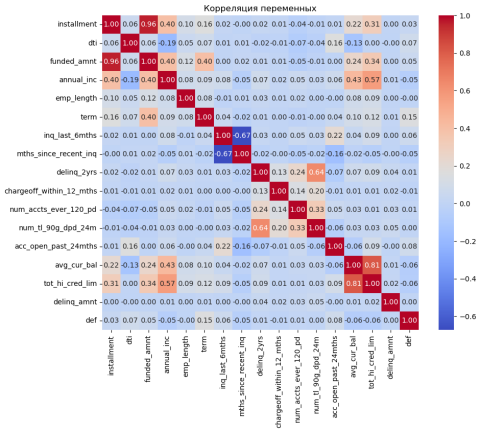
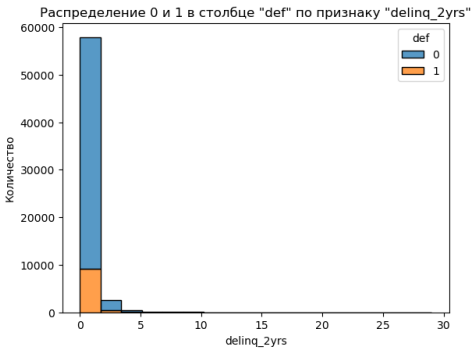


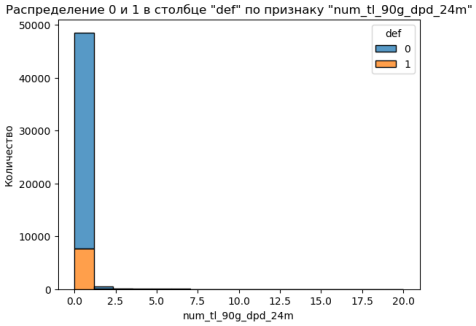
Рис. 2: Корреляционная матрица

Эти и некоторые другие переменные, показавшие высокую корреляцию, были рассмотрены детально. Мной было рассмотрено распределение таргетов на этих признаках и в случае, если распределение двух коррелирующих признаков а также распределение дефолт-рейтов признаков очень похожи, то один из признаков было решено удалить. Такая ситуация сложилась у признаков num_tl_90g_dpd_24m и delinq_2yrs. поэтому, чтобы не перегружать модель лишними признаками, признак num_tl_90g_dpd_24m был удален:

В нашем датасете содержится достаточно большое количество информации о заявке по выдаче кредита, в том числе такую базовую информацию, как дата выдачи кредита. Может быть интересным посмотреть на дефолт рейт относительно времени выдачи кредита: для этого из даты год и месяц, выделим их в отдельные признаки и предположим их предсказательную способность через дефолт-рейт:

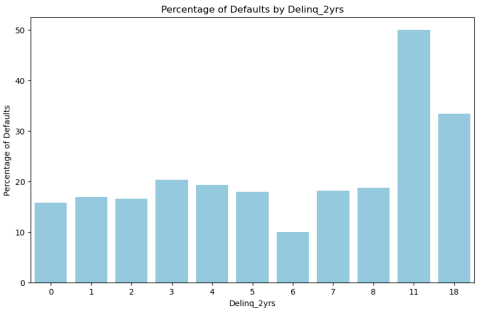


(a) Распределение признака delinq_2yrs

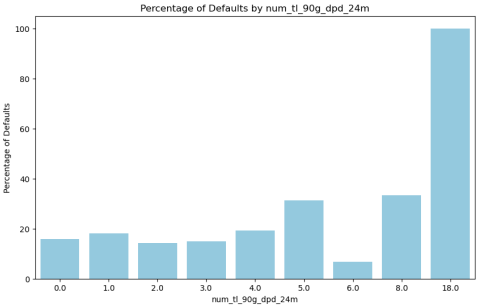


(b) Распределение признака num_tl_90g_dpd_24m

Рис. 3: Распределение таргетной переменной в двух признаках



(a) Распределение дефолт-рэйта признака delinq_2yrs



(b) Распределение дефолт-рэйта признака num_tl_90g_dpd_24m

Рис. 4: Распределение дефолт-рэйта в двух признаках

Было установлено, что дефолт рейт особо не меняется по месяцам и годам, значит эти два признака не будут слишком информативными. В целом это объяснимо: просто дата выдачи кредита не может сильно влиять на заемщика.Однако, можно было предположить, что количество дефолтов может объяснятся через макроэкономические события отдельного года, поэтому в целом для такого анализа можно добавить в датасет, например, показатель инфляции в конкретном году, не передавая конкретные даты, избегая переобучения.

Далее было произведено укрупнение по категориям в категориальных признаках:

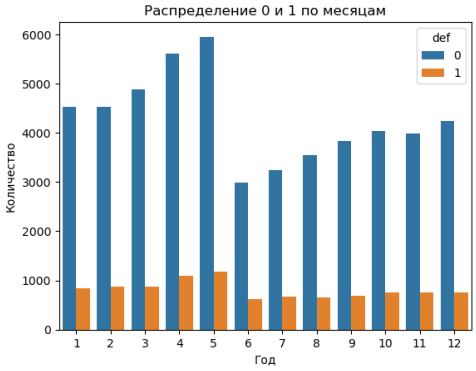
Столбец 'addr_state' был укрупнен в соответствии с разделением территории США на регионы: northwest, southwest,west, southeast, midwest. В столбце 'home_ownership"две категории 'none' и 'other' были объединены по смыслу.

Также важной частью работы с данными из анкеты является обработка переменных, которые потенциально могут привести к утечке данных о вероятности дефолта в модель, что приведет к переобучению модели.

В некоторых случаях, вероятность дефолта используется для определения процентной ставки. Более высокий PD приводит к более высокой процентной ставке, чтобы компенсировать повышенные риски.

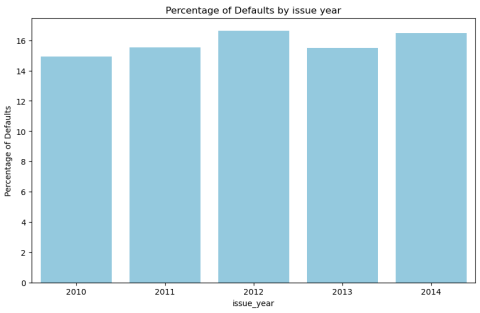
А если в модель включена переменная installment (ежемесячный платеж), то в ней может содержаться информация о процентной ставке, которая была установлена на основе вероятности дефолта. Таким образом, installment может неявно содержать информацию о вероятности дефолта клиента, которую мы и пытаемся предсказать.

(a) Распределение таргета по годам

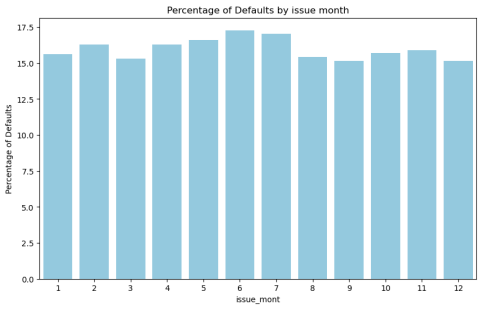


(b) Распределение таргета по месяцам

Рис. 5: Распределение таргетной переменной в двух признаках



(a) Распределение дефолт-рэйта по годам



(b) Распределение дефолт-рэйта по месяцам

Рис. 6: Распределение дефолт-рэйта в двух признаках

4 Работа с пропущенными данными

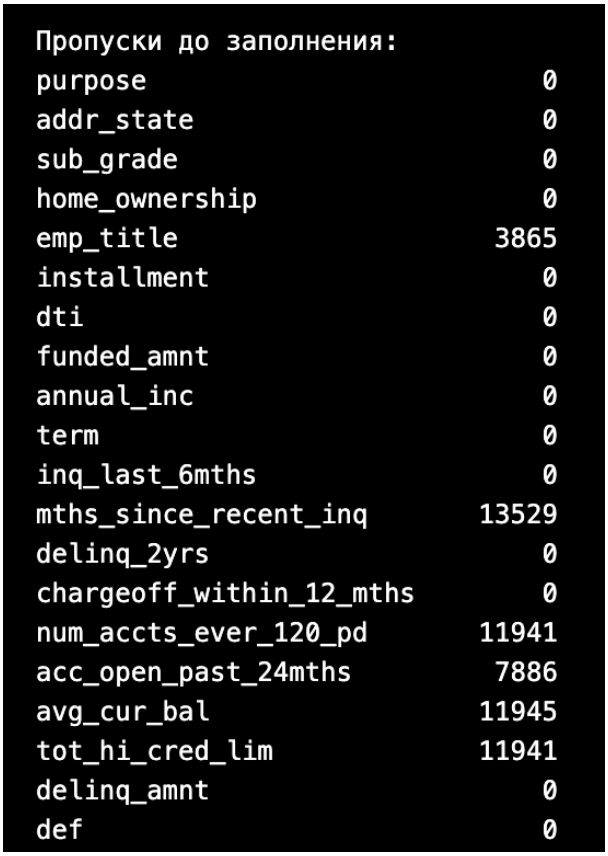


Рис. 7: Количество пропусков до заполнения

Заметим, что достаточное количество пропусков содержатся в столбцах 'emp_title' , 'emp_lenght'.

Можно предположить, что эти данные о своем трудоустройстве клиенты скрыли намеренно: поэтому нельзя однозначно исключить эти переменные, так как пропуск в данных может свидетельствовать об отсутствии работы или ее непостоянности, что может сильно повлиять на решение по выдаче кредита. Ранее было установлено, что из-за слишком большого количества уникальных значений, имеет смысл удалить перемеенную "emp_title однако создадим вспомогательный столбец в котором установим значение 1, если имел место пропуск в "emp_title и 0 в обратном случае. Это поможет иметь представление о потенциальных намерениях заемщика

Можно предположить, что пропуски в столбце 'mths_since_recent_inq' означают, что люди вообще не делали запрос на займ. Поэтому эти пропуски заполним максимальным значением в столбце, так как, как мы установили в EDA, причину, почему при нулевом inq_last_6mnths такие высокие значения mths_since_recent_inq - просто в последние 6 месяцев запроса от клиента не было, поэтому месяцев с последнего запроса намного больше 6. В целом, как было установлено, максимальное значение в столбце - около 24 месяцев, что, кажется, является достаточным сроком для повторного запроса на кредит от среднего клиента.

С остальными переменными поступим так: заполним средним значением столбцов, которые являются вторым по показателю корреляции для каждого из них, чтобы не создавать еще большую зависимость для переменной с максимальной корреляцией.

5 Генерация новых признаков

Был сгенерирован новый признак - CUR (Credit Utilization Ratio) как отношения суммы финансирования (funded_amnt) к общему кредитному лимиту (tot_hi_cred_lim) Этот показатель может быть полезен, так как отражает, насколько заемщик использует доступный ему кредитный лимит.

Высокое значение CUR может указывать на то, что заемщик близок к пределу своих кредитных возможностей, что может быть признаком финансовых трудностей.

6 WOE - преобразования

WOE-преобразования помогают преобразовать категориальные переменные в числовые значения, которые лучше подходят для модели логистической регрессии. Преобразования в woe-бины были осуществлены с помощью библиотеки scorecardpy. Для того, чтобы модель работала корректно некоторые бины были измнены вручную, чтобы восстановить их монотонность

Далее при пороге IV 0.025 было отобрано 8 самых информативных признаков:

'CUR', 'acc_open_past_24mths', 'dti', 'inq_last_6mths', 'mths_since_recent_inq', 'sub_grade', 'annual_inc', 'avg_cur'bal'

7 Подбор гиперпараметров и обучение модели

Обучение модели было произведено с помощью библиотеки sklearn. Эта же библиотека использовалась для подбора гиперпараметров модели, в ходе исследования сравнивались два подхода: Grid Search и Random Search. В качестве альтернативного метода оптимизации использовалась библиотека Optuna.

В ходе эксперимента оказалось, что подход Random Search и Grid Search дали одинаковое значение коэффициента Джини. Этот показатель на тренировочном датасете составил 37.1%. На валидационном датасете - 36.8% Параметры, подобранные с помощью библиотеки Optuna показали аналогичные результаты

Более детально с кодом Вы можете ознакомиться в ноутбуке.

8 Валидационные тесты

Более подробные значения всех оцениваемых метрик и методику проведения тестов можно посмотреть в ноутбуке

Тест М2.1: Эффективность ранжирования всей модели

Результат теста: зеленый

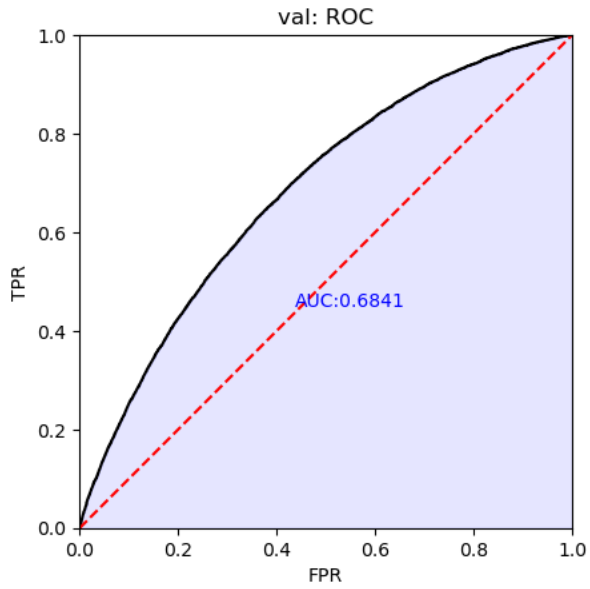


Рис. 8: Тест М2.1. Значение коэффициента Джини - 36.8%

Тест М2.2: Эффективность ранжирования отдельных факторов

Результат теста: зеленый

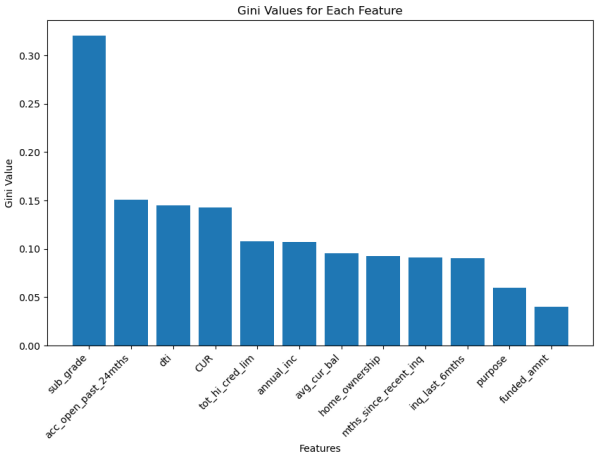


Рис. 9: Тест M2.2. Один желтый фактор, 11 зеленых факторов

Тест M2.5: Анализ вкладов факторов в формирование Джини модели

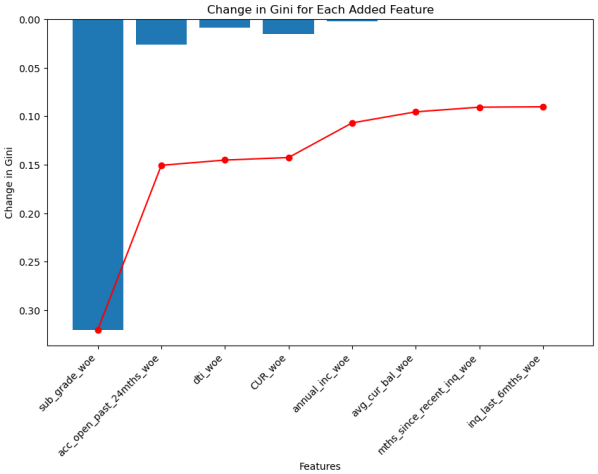


Рис. 10: Тест M2.5

Тест M2.4: Динамика коэффициента Джини

В среднем на каждой из дат значение около 35% (то есть в пределах желтого и зеленого индикаторов), что в целом неплохой результат. Однако стоит иметь ввиду, что данные по датам распределены неоднородно, что может так же влиять на качество ранжирования модели

Рис. 11: Тест M2.4

Тест M3.1: Анализ корректности дискретного преобразования факторов

Результат теста: зеленый

Все преобразования оказались монотонными по вероятности дефолта.

Графики для каждого преобразования представлены в ноутбуке.

Тест M4.1: Сравнение прогнозного и фактического TR (Target Rate) на уровне выборки

Если брать изначальные результаты модели, выдаваемые функцией predict, то модель фактически не определяет дефолтные кредиты. Был подобран порог для вероятности дефолта, при котором заемщик объявляется таковым. Таким подбором получилось сделать разницу около 8 процентов в предсказании, что соответсвует зеленому индикатору в тесте

Тест M4.2: Тест формы калибровочной кривой

Результат теста: зеленый

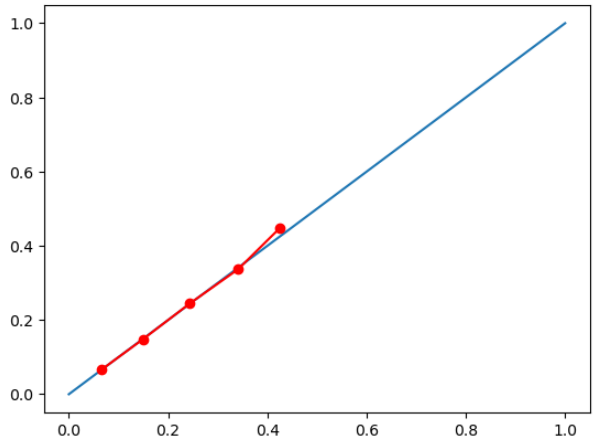


Рис. 12: Тест M4.2

	relative_difference
0	0.006408
1	0.007068
2	0.003550
3	0.012401
4	0.050777

Рис. 13: Тест M4.2

Видим, что относительная разница очень маленькая на каждом бине

Тест M5.1: Сравнение эффективности ранжирования модели на разработке и валидации

Результат теста: зеленый

Абсолютное снижение - 0.3 п.п

Относительное снижение - 0.008 процентов

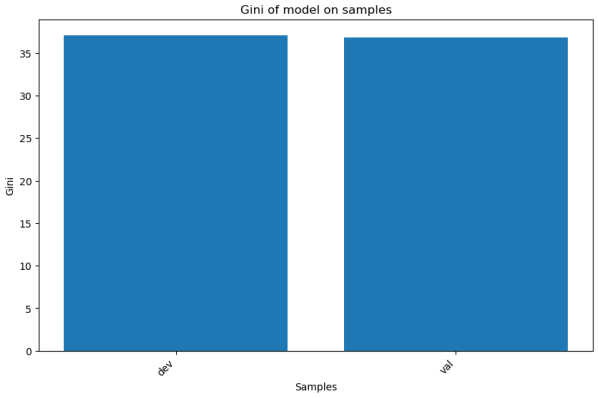


Рис. 14: Тест M5.1

Тест M5.2: Сравнение эффективности ранжирования отдельных факторов модели на разработке и валидации

Результат теста: зеленый

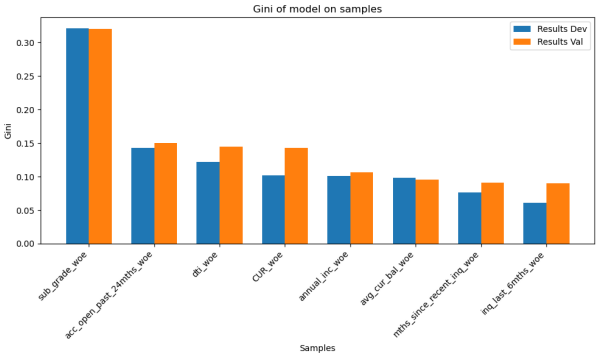


Рис. 15: Тест M5.2

	Признак	Относительное изменение	Абсолютное изменение
0	CUR	-0.140056	-0.02
1	acc_open_past_24mths	53.360489	5.24
2	dti	42.913386	4.36
3	inq_last_6mths	-10.770751	-1.09
4	mths_since_recent_inq	17.945384	1.38
5	sub_grade	-0.342253	-0.11
6	annual_inc	-12.088816	-1.47
7	avg_cur_bal	56.045752	3.43

Рис. 16: Тест M5.2

Таким образом, получаем, что все признаки входят в категорию зеленых, следовательно модель прошла валидационный тест

9 Альтернативное моделирование

Метод случайного леса (англ. random forest) — алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев. Этот алгоритм основывается на том, что случайным образом строятся несколько деревьев решений, каждое из которых определяет значение класса таргетной переменной (например, 0 или 1). После этого определяется предсказанное значение таргета : в условиях задачи классификации класс с наибольшим числом голосов от случайных деревье становится прогнозом алгоритма.

Случайность в построении деревьев достигается таким образом: случайным образом выбираются с повторениями примеры из исходных данных, чтобы создать подвыборки для каждого дерева, далее эти подвыборки делятся на узлы дерева по одному на каждый, и так строится дерево решений различной глубины. Каждое дерево строится по своей подвыборке данных, и они работают независимо друг от друга.

У функции “RandomForestClassifier“ существует несколько параметров, но остановимся подробнее на двух основных:

- n_estimators (по умолчанию: 10): Этот параметр указывает количество деревьев в случайном лесе. Большее число деревьев может улучшить качество модели, но также может увеличить время обучения.

- max_depth (по умолчанию: None): Этот параметр ограничивает максимальную глубину каждого дерева в случайном лесе. Если установлено значение None, деревья не будут ограничены по глубине.

Перед этим немного преобразуем датасет для подачи в модель Random Forest: Addr_state, purpose, sub_grade мы преобразуем так: с помощью category_encoders построим биекцию из множества значений, принимаемых переменной в множество натуральных чисел.

При подборе гиперпараметров для данной модели тем же способом, что и для основной модели, и обучении был получен коэффициент Джини 35,5%.

Ниже представлена сравнительная таблица, в которой можно увидеть, что все таки логистическая регрессия показала себя с лучшей стороны. Также логистическая регрессия с WOE-преобразованиями

более предпочтительна для задачи кредитного скоринга, так как является более интерпретируемой, что является большим плюсом в прикладных задачах бизнеса

	Модель машинного обучения	Метод подбора гиперпараметров	Гиперпараметры	Коэффициент Джини
0	Logistic Regression with WOE	Grid Search	{'penalty': 'l2', 'C': 0.004281332398719396}	0.3681
1	Logistic Regression with WOE	Optuna library	{'C': 632.2748546926065}	0.3680
2	Randomn Forest	Optuna library	{'n_estimators': 240, 'min_samples_leaf': 6, '...	0.3542

Рис. 17: Результаты моделей

10 Скоринговая карта

В конце обучения модели была построена скоринговая карта:

В обоих наборах данных наиболее распространенные скоринговые баллы находятся в диапазонах [450, 500) и [500, 550). В диапазоне [350, 450) наблюдается заметная разница в распределении: в тестовом датасете таких заемщиков больше, чем в тренировочном. В остальных диапазонах распределение баллов между наборами данных схоже.

С уменьшением скорингового балла вероятность дефолта увеличивается, что ожидаемо, так как более низкий скоринговый балл соответствует более высокому риску. Для обоих датасетов тренд вероятности дефолта по диапазонам баллов схож, что говорит о стабильности модели.

В верхней части графика указан показатель PSI (Population Stability Index), равный 0.0164.

При PSI = 0.0164, получили, что распределение скоринговых баллов практически не изменилось между выборками, и модель является стабильной.

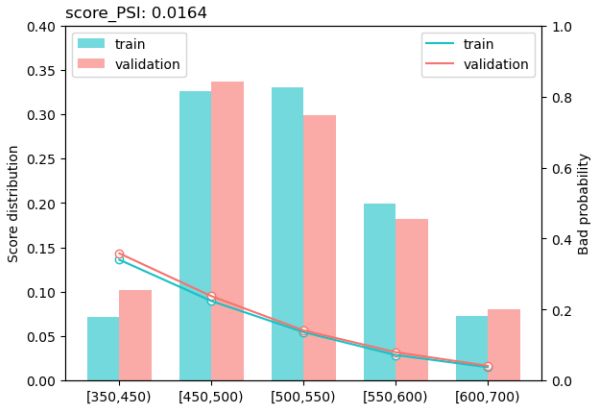


Рис. 18: Скоринговая карта

11 Подсчет ожидаемой прибыли

Основная задача кредитного скоринга - предсказать дефолт заемщика, однако глобальной целью данного прогноза является минимизация убытков банка, или, если посмотреть на задачу с другой стороны, максимизация его прибыли. Именно поэтому задача о подсчете ожидаемой прибыли является закономерным окончанием данной работы.

Итак, введем следующие экономические предпосылки для рассчета прибыли:

LGD (Loss Given Default, Потери при дефолте) – это финансовый показатель, который отражает процент убытков, которые несет кредитор в случае дефолта заемщика, т.е. невыплаты долга.

- LGD=100%, а текущая задолженность по каждому кредиту равна полю funded_amnt.

- В случае дефолта клиент полностью не выплачивает весь кредит, а процентная ставка для всех клиентов одинакова и составляет 13% годовых.

Алгоритм расчета прибыли:

Ожидаемая прибыль для каждого кредита рассчитывается по формуле:

$$\text{expected\_profit} = \text{'installment'} \times \text{'term'} - \text{'funded\_amnt'}$$

Фактическая прибыль может принимать разные значения в зависимости от предсказания модели и реального значения таргетной переменной:

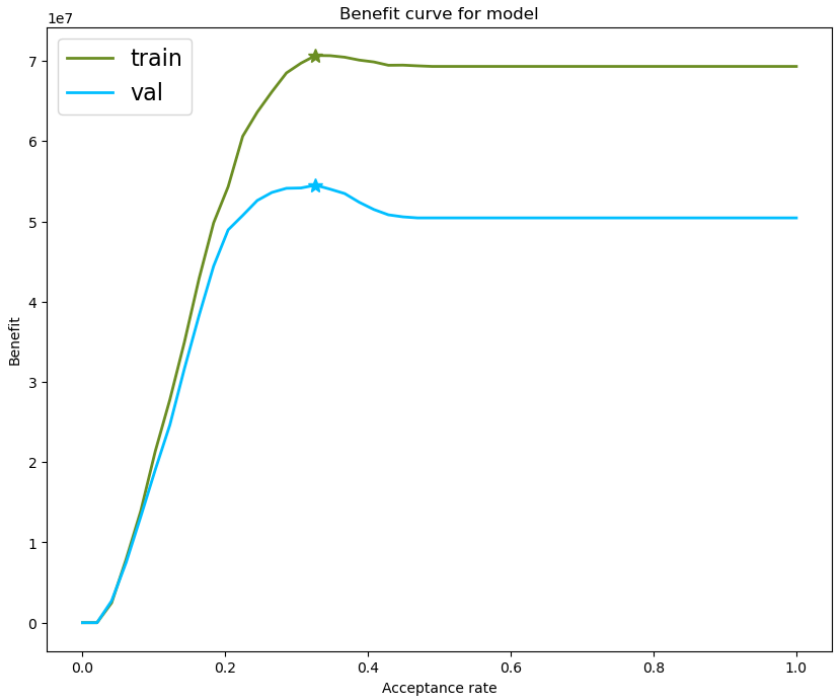


Рис. 19: График зависимости ожидаемой прибыли от пороговой вероятности

- Если предсказано и фактически нет дефолта: прибыль равна ожидаемой прибыли.
- Если предсказано отсутствие дефолта, но фактически он имеет место: прибыль равна -LGD.
- Если предсказан дефолт, независимо от фактического результата: прибыль банка равна нулю, так как банк принимает решение не выдавать клиенту кредит.

График демонстрирует, что на двух выборках максимальная прибыль была достигнута при установлении порога принятия решения о дефолте при предсказанной вероятности дефолта около 0.32 Также стоит отметить, что после значения вероятности около 0.5 прибыль выходит на плато: это можно объяснить тем, что, согласно калибровочной кривой, которая ранее была построена в валидационных тестах, мы увидели, что модель не предсказывает вероятность дефолта больше значений около 0.6, то есть при достаточно больших порогах отсечения модель будет принмать решение выдавать кредит всем клиентам, однако при этом прибыль не уходит в отрицательные значения. Это в свою очередь можно объяснить тем, что, как мы установили сначала в EDA, выборка смещена в сторону "хороших"клиентов, поэтому, с учетом наших предпосылок, убытки по дефолтным клиентам будут не такими большими, как можно было бы предполагать.

В случае, если LGD уменьшить до 80% пороговое значение вероятности увеличивается до 36% на валидационной выборке. Это в целом понятно: теперь банк несет меньше потерь, поэтому может более активно выдавать кредиты: требования к вероятности дефолта стали слабее.

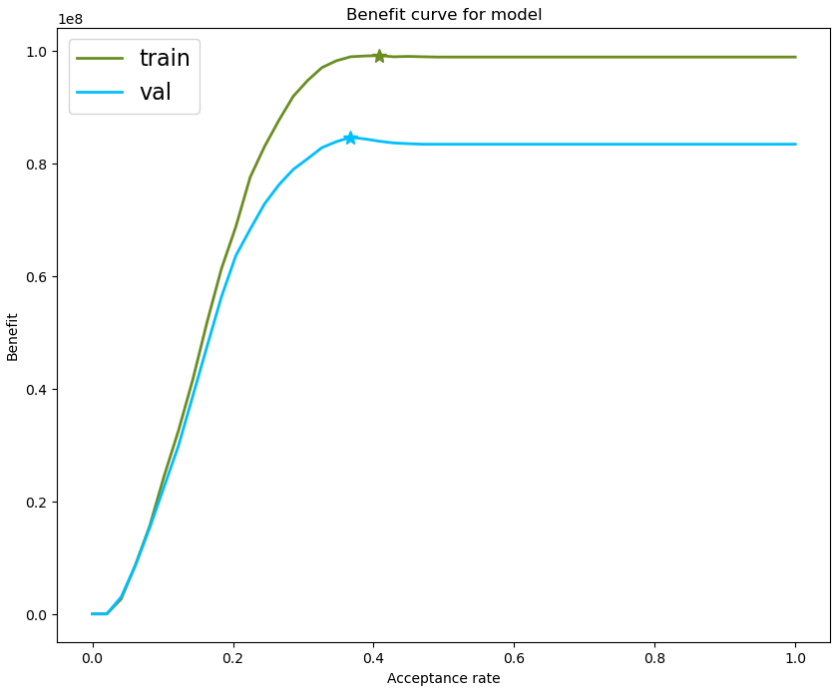


Рис. 20: График зависимости ожидаемой прибыли от пороговой вероятности при LGD = 80%