МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**по курсу**

«Data Science Pro»

**Тема:** «**Прогнозирование вероятности дефолта клиентов»**

Слушатель **Щекина Татьяна Сергеевна**

Москва, 2025

**Содержание**

[Введение 3](#__RefHeading___Toc5172_1330607712)

[1 Аналитическая часть 4](#__RefHeading___Toc5200_1330607712)

[1.1 Постановка задачи 4](#__RefHeading___Toc5198_1330607712)

[1.2 Описание используемых методов 6](#__RefHeading___Toc5196_1330607712)

[1.2.1 Логистическая регрессия 6](#__RefHeading___Toc5210_1330607712)

[1.2.2 Дерево решений 6](#__RefHeading___Toc9083_1330607712)

[1.2.3 Градиентный бустинг LightGBM 6](#__RefHeading___Toc9079_1330607712)

[1.2.4 Адаптивный бустинг AdaBoost 6](#__RefHeading___Toc7347_1832855893)

[1.2.5 Экстремальный градиентный бустинг XGBoost 6](#__RefHeading___Toc7349_1832855893)

[1.2.6 Многослойный перцептрон 6](#__RefHeading___Toc5533_368929236)

[1.2.7 Метрики качества моделей классификации 6](#__RefHeading___Toc5194_1330607712)

[1.3 Разведочный анализ данных 8](#__RefHeading___Toc5192_1330607712)

[2 Практическая часть 9](#__RefHeading___Toc5190_1330607712)

[2.1 Предобработка данных 9](#__RefHeading___Toc5188_1330607712)

[2.2 Разработка и обучение моделей 10](#__RefHeading___Toc5186_1330607712)

[2.3 Тестирование модели 11](#__RefHeading___Toc5184_1330607712)

[2.4 Разработка нейронной сети 12](#__RefHeading___Toc5182_1330607712)

[2.5 Создание удаленного репозитория GitHub и загрузка результатов 12](#__RefHeading___Toc5178_1330607712)

[Заключение 13](#__RefHeading___Toc1505_1560672135)

[Библиографический список 14](#__RefHeading___Toc1507_1560672135)

**Введение**

Тема данной работы — прогнозирование дефолта клиентов по платежам, на основании данных, предоставленных банком Home Credit Bank в 2018 году для соревнования на платформе Kaggle.

Многим людям сложно получить кредит из-за недостаточной или отсутствующей кредитной истории. Задача состоит в построении модели, которая будет оценивать вероятность беспроблемного обслуживания долга заявителем на основании имеющейся статики, которая включает в себя не только кредитную историю клиентов, но также и многочисленные альтернативные данные, включая информацию о телекоммуникационных компаниях и транзакциях, здании, в котором проживает клиент, рейтинга региона, для прогнозирования платежеспособности своих клиентов.

1. **Аналитическая часть**
   1. **Постановка задачи**

Сформулированная организаторами соревнования Kaggle задача является классической задачей классификации: нулевой класс соответствует беспроблемным кредитам, первый класс — кредитам, по которым заявитель допускал просрочки более некоторого количества дней.

В рамках этой работы поставлена задача:

* провести разведочный анализ данных
* при необходимости провести предобработку данных
* добавить в основной датасет агрегированные показатели, используя данные из бюро кредитных историй, данные кредитных карт и предыдущих кредитных заявок
* провести отбор наиболее информативных переменных-предикторов
* применить методы классического машинного обучения к задаче классификации, сравнить их эффективность
* разработать нейронную сеть для решения задачи классификации
* при построении моделей провести поиск гиперпараметров модели с помощью поиска по сетке с перекрестной проверкой, количество блоков равно 10.
* создать публичный репозиторий в GitHub и разместить там код и документацию исследования

Предоставленный датасет состоит из 7 основных файлов в табличном виде. Рассмотрим их по-порядку:

1. application\_{train|test} – это основная таблица, разбитая на 2 файла — для обучения и тестирования, содержит статические данные для всех кредитных заявок. Одна строка представляет один кредит в выборке данных. В обучающей выборке содержится 307 тысяч заявок, в тестовой — 48 тысяч заявок. Количество признаков (не считая целевой переменной и технического идентификатора заявки SK\_ID\_CURR ) — 120. Для данной исследовательской работы будет использоваться только файл с обучающей выборкой: 70% выборки используется для обучения моделей, 30% - для тестирования моделей.
2. bureau - все предыдущие кредиты клиента, предоставленные другими финансовыми учреждениями, о которых было сообщено в кредитное бюро (для клиентов, имеющих кредит в нашей выборке). Для каждого кредита в нашей выборке имеется столько строк, сколько кредитов было у клиента в кредитном бюро до даты подачи заявки. Содержит 1.7 млн записей и 15 признаков (не считая ключей SK\_ID\_CURR и SK\_BUREAU\_ID, которые используются для джойна между таблицами).
3. bureau\_balance - ежемесячные остатки по предыдущим кредитам в кредитном бюро. В этой таблице есть одна строка для каждого месяца истории каждого предыдущего кредита, сообщенного в кредитное бюро, т.е. в таблице есть строки (количество кредитов в выборке \* количество относительных предыдущих кредитов \* количество месяцев, в которых у нас есть некоторая история, наблюдаемая для предыдущих кредитов). Таблица содержит 27.3 млн записей и 2 признака.
4. POS\_CASH\_balance - ежемесячные справки о балансе предыдущих кредитов наличными, которые клиент имел в банке Home Credit. По одной записи для каждого кредита, каждый месяц. Таблица содержит 10.0 млн записей и 6 признаков.
5. credit\_card\_balance - ежемесячные балансы кредитных карт, которые заявитель имел в Home Credit. Таблица содержит 3.8 млн записей и 21 признак.
6. previous\_application - все предыдущие заявки на кредиты Home Credit клиентов, имеющих кредиты в нашей выборке. Таблица содержит 1.6 млн записей и 35 признаков.
7. installments\_payments - история погашения ранее выданных кредитов в Home Credit, относящихся к кредитам в нашей выборке. Имеется a) одна строка для каждого произведенного платежа плюс b) одна строка для каждого пропущенного платежа. Таблица содержит 13.6 млн записей и 6 признаков.
   1. **Описание используемых методов**
      1. **Логистическая регрессия**
      2. **Дерево решений**
      3. **Градиентный бустинг LightGBM**
      4. **Адаптивный бустинг AdaBoost**
      5. **Экстремальный градиентный бустинг XGBoost**
      6. **Многослойный перцептрон**
      7. **Метрики качества моделей классификации**

Существует множество различных метрик для оценки качества классификаторов. К наиболее широко используемым относятся:

1. матрица ошибок (confusion matrix)
2. правильность (accuracy), в некоторых источниках также называется меткостью
3. правильность с учетом цены (cost-sensitive accuracy)
4. точность, полнота и F-мера (precision и recall, F-score)
5. площадь под кривой рабочей характеристики приемника (Receiver Operating Characteristic, ROC) - AUC

Матрица ошибок для задачи бинарной классификации представляет собой матрицу размером 2 x 2, по одной оси располагается метка, предсказанная моделью, по другой оси — фактическая метка. При этом в различных источниках горизонтальная и вертикальные оси будут быть подписаны по-разному. В этой работе буду использовать матрицу ошибок в том виде, как представлено на рисунке 1.

|  |  |  |
| --- | --- | --- |
|  | Положительная предсказанная метка | Отрицательная предсказанная метка |
| Положительная фактическая метка | TP | FN |
| Отрицательная фактическая метка | FP | TN |

Рисунок 1 — матрица ошибок

Правильность определяется как отношение правильно классифицированных экземпляров к общему количеству:

Правильность = TP + TN / (TP + TN + FP + FN) (1)

В нашем случае датасет несбалансирован — отрицательных меток в 11 раз больше, чем положительных. В этом случае имеет смысл использовать правильность с учетом цены, то есть присваивать ложно-положительным и ложно-отрицательным результатам разные веса.

Точность (precision) это отношение истинно положительных прогнозов к общему количеству положительных прогнозов:

Точность = TP / (TP + FP) (2)

Полнота (recall) определяется как отношение истинно положительных прогнозов к обществу количеству положительных экземпляров:

Полнота = TP / (TP + FN) (3)

Полнота и точность не зависят, в отличие от правильности, от соотношения классов и подходят лучше для нашей задачи.

F-мера представляет собой агрегированный критерий качества, объединяющий собой полноту и точность с некоторыми весами, которые зависят от задачи и предпочтений заказчика. F-мера рассчитывается как среднее гармоническое с весами по формуле.

F-меру удобно использовать в процессе оптимизации параметров алгоритма, ее значение достигает максимума при полноте и точности, равными единице, и близка к нулю, если один из аргументов близок к нулю.

ROC кривая строится на основании комбинации доли истинно-положительных результатов (TPR) и доли ложно-положительных результатов (FPR). Диапазон оценок достоверности дискретизируется, затем каждое дискретное значение используется в качестве порога при прогнозировании. Чем больше площадь под ROC кривой, тем лучше классификатор.

Организаторы соревнования Kaggle определили в качестве основной метрики качества моделей площадь под кривой ROC. В работе будут использоваться несколько метрик качества классификаторов, поскольку не все рассматриваемые алгоритмы возвращают вероятность принадлежности к классу, а не сам класс.

* 1. **Разведочный анализ данных**

Разведочный анализ данных включает в себя:

* нормировку данных — нормировка данных может быть полезна для дальнейшей интерпретации весов модели (например, для логистической регрессии)
* анализ на аномалии (пропущенные данные, выбросы)
* корректировка аномалий там, где это возможно

Разведочный анализ данных, перечисленный выше, был также дополнен агрегированным показателями, отсутствующими в исходных таблицах:

1. количество ранее выданных кредитов клиента
2. количество типов ранее выданных кредитов — этот параметр показывает, насколько диверсифицирован кредитный портфель клиента
3. среднее количество кредитов одного типа
4. доля активных кредитов от общего числа
5. доля пролонгированных кредитов — событие пролонгации может служить негативным сигналом, увеличивающим вероятность дефолта в будущем
6. отношение просроченных платежей клиента к общему размеру долга клиента
7. доля кредитов, у которых дата погашения в прошлом
8. **Практическая часть**
   1. **Предобработка данных**

Визуализация первичных данных позволяет наглядно увидеть, какие из признаков имеют значительные выбросы. Некоторые признаки из датасета уже нормализованы (такие как внешние рейтинги EXT\_RATINGS 1-3).

Гистограммы распределения признаков до устранения выбросов и ящики с усами приведены в приложении А.

На основании визуализации определены признаки, которые не требуют удаления выбросов или для которых будет использоваться собственный критерий

|  |  |
| --- | --- |
| **Признак** | **Критерий выброса** |
| REGION\_POPULATION\_RELATIVE | > 0.04 |
| AMT\_REQ\_CREDIT\_BUREAU\* | Заменить на индикатор I[var > 1.0] |
| CREDIT\_ENDDATE\_PAST\_PERCENTAGE | Выбросы отсутствуют |

Предобработка данных состояла из последовательных шагов:

1. Анализ бинарных и категориальных признаков. Для категориальных признаков с большим количеством категорий (более 10) была сделана смысловая группировка с целью сокращения количества категорий.
2. Добавление дополнительных числовых признаков, которые характеризуют кредитную нагрузку клиента, а также агрегированные показатели из бюро кредитных историй:

* Отношение кредита к залогу (LTV)
* Отношение долга к доходу (DTI)
* Количество ранее выданных кредитов клиента
* Количество типов ранее выданных кредитов (POS/потребительный и т. д.)
* Среднее количество кредитов каждого типа
* Доля активных кредитов от общего числа
* Доля пролонгированных кредитов
* Отношение просроченных платежей к общему долгу на момент заявки
* Доля кредитов, у которых дата погашения в прошлом
* Отношение одобренной суммы кредита к выбранной сумме кредита

Данные признаки получены соединением таблиц по ключу SK\_ID\_CURR. Поскольку не по всем кредитам есть информация в кредитном бюро, то по отсутствующим значениям признаки заполнены нулями.

1. Удаление признаков с высоким процентом отсутствующих значений (в качестве критерий выбран 25%).
2. Значения типа «бесконечность» заменяются отсутствующими значениями NaN
3. Применяется анализ на выбросы методом межквартильного размаха. Метод 3-сигм работает плохо в нашем случае, поскольку данные в основном имеют асиметричное распределение
4. Для признаков с процентом заполнения более 75% применяется стандартный Imputer из библиотеки sklearn: для категориальных и бинарных переменных используется наиболее частотное значение, для числовых — среднее значение.
5. Для категориальных переменных применяется метод One-Hot Encoding, цель которого преобразовать каждый признак с N категориями в N-1 бинарный признак.
6. Для непрерывных переменных применяется стандартизация MinMaxScaler() либо StandardScaler().
   1. **Разработка и обучение моделей**

В данной части приводится список моделей, которые будут использоваться для прогноза модуля упругости при растяжении и прочности при растяжении.

Для решения задачи прогнозирования дефолта обученные модели из следующих классов:

* Логистическая регрессия
* Дерево решений
* Градиентный бустинг LightGBM
* Адаптивный бустинг AdaBoost
* Экстремальный градиентный бустинг XGBoost
* Многослойная полносвязная нейронная сеть (перцептрон)
  1. **Тестирование модели**

Каждый из рассмотренных классификаторов продемонстрировал точность выше, чем базовая модель, а модель LightGBM показала наилучший результат.

|  |  |  |
| --- | --- | --- |
| **Классификатор** | **Тренировочная выборка** | **Тестовая выборка** |
| Адаптивный бустинг AdaBoost | 0.7603 | 0.7577 |
| Градиентный бустинг LightGBM | 0.8286 | 0.7676 |
| Дерево решений | 0.7294 | 0.7278 |
| Логистическая регрессия | 0.7453 | 0.7488 |
| Многослойная нейронная сеть | 0.7610 | 0.5000 |
| Экстремальный градиентный бустинг XGBoost | 0.8980 | 0.7565 |

Рисунок — Сравнение метрики AUC

на тренировочном и тестовом датасетах

|  |  |  |
| --- | --- | --- |
| **Классификатор** | **Тренировочная выборка** | **Тестовая выборка** |
| Адаптивный бустинг AdaBoost | 0.2672 | 0.2674 |
| Градиентный бустинг LightGBM | 0.3107 | 0.2739 |
| Дерево решений | 0.2508 | 0.3362 |
| Логистическая регрессия | 0.2581 | 0.2595 |
| Многослойная нейронная сеть | 0.3336 | 0.2946 |
| Экстремальный градиентный бустинг XGBoost | 0.4040 | 0.2852 |

Рисунок — Сравнение метрики F1-score

на тренировочном и тестовом датасетах

* 1. **Разработка нейронной сети**

Для построения и обучения нейросети используется библиотека tensorflow.keras.

На вход нейронной сети подается стандартизованный и сбалансированный датасет. Датасет сбалансирован методом undersampling.

* 1. **Создание удаленного репозитория GitHub и загрузка результатов**

Для целей данного исследования был создан публичный репозиторий на GitHub по адресу <https://github.com/Spring-31/HC_default_risk>, в котором размещена данная пояснительная записка, метаданные по исходному датасету, результирующие характеристики моделей, рабочие ноутбуки Jupyter.

**Заключение**

Для достижения поставленной цели был проведен полноценный pipeline машинного обучения, включающий в себя следующие этапы:

* изучение теоретических основ анализа данных и машинного обучения: типы классификаторов (деревья решений, ансамблевые методы, логистическая регрессия), методы обучения (понятие тренировочной, тестовой и валидационной метрики), метрик качества классификаторов
* изучение библиотек python для применения теоретических знаний на практике: scikit-learn, tensorflow, keras классификаторы библиотек scikit-learn
* знакомство с особенностями предметной области, в которой решается задача (оценка риска дефолта клиентов-физических лиц)
* проведение первичного анализа данных с использованием статистических методов
* обработка данных: извлечение данных, работа с типами данных, идентификация и удаление аномалий, заполнение отсутствующих значений, стандартизация данных
* инженерия признаков: генерация новых признаков (извлечение статистик, агрегация данных), преобразование данных, объединение данных, отбор значимых признаков)
* определение метрик измерения эффективности моделей машинного обучения, выбор наиболее подходящей для решаемой задачи метрики
* подбор оптимальных гиперпараметров для повышения эффекттивности работы моделей машинного обучения

В рамках проведенного исследования удалось познакомиться с каждым из перечисленных выше этапов. Наиболее эффективная модель классического машинного обучения характеризуется значением AUC 0.7676, в то время как показатель лидеров в соревновании Kaggle превышает 0.802.

Что касается разработки многослойного перцептрона, обучить хоть сколько-то эффективную нейронную сеть не удалось. При беглом анализе результатов соревнования Kaggle также не удалось найти решение, в котором использовались бы полносвязные нейронные сети. Среди лучших решений применялись модели LightGBM, XGBoost, Random Forest и ExtraTrees. Например, команда, занявшая 4-е место, использовала стек из моделей LightGBM, Random Forest, ExtraTree и Linear Regression.

С моей точки зрения, одно из слабых мест разработанных классификаторов — слишком большое количество признаков, не оптимальный выбор нормализаторов — для асиметричных распределений использование StandardScaler не оптимально.

Дальнейшие шаги по улучшению качества моделей могли бы включать:

* Более тонкую настройку стандартизации непрерывных переменных
* Снижение размерности, сокращение признаков в выборке, например методом главных компонент (PCA)
* Дальнейший подбор гиперпараметров (невозможен в текущем исследовании из-за большого количества признаков)

**Библиографический список**

1. Anna Montoya, inversion, KirillOdintsov, and Martin Kotek. Home Credit Default Risk. <https://kaggle.com/competitions/home-credit-default-risk> , 2018. Kaggle.
2. VanderPlas, Python Data Science Handbook [Текст]: Python Data Science Handbook: Essential Tools for Working with Data [2 ed.]. 2022
3. Бурков Андрей. Машинное обучение без лишних слов [Текст] — Спб.: Питер, 2020. - 192 с.
4. Рашка, Себастьян, Мирджалили, Вахид. Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2, 3-е изд.: Пер. с англ. - СПб. : ООО "Диалектика", 2020. - 848 с.
5. Уэс Маккинни. Python и анализ данных: Первичная обработка данных с применением pandas, NumPy и Jupiter / пер. с англ. А. А. Слинкина. 3-е изд. – М.: МК Пресс, 2023. – 536 с.: ил.
6. Логистическая регрессия кредитного риска – Режим доступа <https://fastercapital.com/content/Credit-Risk-Logistic-Regression--How-to-Use-Logistic-Regression-to-Estimate-the-Probability-of-Default.html>
7. Документация по библиотеке numpy: – Режим доступа: https://numpy.org/
8. Документация по библиотеке pandas: – Режим доступа: <https://pandas.pydata.org/>
9. Документация по библиотеке matplotlib: – Режим доступа: https://matplotlib.org/
10. Документация по библиотеке numpy: – Режим доступа: https://scikit-learn.org/stable/api/index.html
11. Документация по библиотеке sklearn: – Режим доступа: https://scikit-learn.org/stable/api/index.html
12. Документация по библиотеке Tensorflow Keras – Режим доступа: <https://www.tensorflow.org/guide/keras>
13. Сравнение 11 классификаторов https://www.kaggle.com/code/davidfumo/comparing-11-classification-models
14. Документация по библиотеке LightGBM – Режим доступа: <https://lightgbm.readthedocs.io/en/stable/>
15. Документация по библиотеке seaborn: – Режим доступа: <https://seaborn.pydata.org/tutorial.html>
16. Loginom Вики. Алгоритмы: – Режим доступа: <https://wiki.loginom.ru/algorithms.html>