ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет бизнеса и менеджмента

Оценка экономического эффекта и улучшений от внедрения моделей машинного обучения на данных из банковского сектора в задачах бинарной классификации

Курсовая работа студента

Сметанин Антон Александрович

2 курс, направление подготовки: 38.03.05 «Бизнес-информатика»

образовательная программа «Бизнес-информатика»

Научный руководитель

д-р наук

Масютин Алексей Александрович

Москва 2021

Содержание

Введение

В 21-ом веке нас везде окружает информация. Новости дня, акции, котировки на биржи, события, проишествия, аналитика, экспертиза, природные явления. Одним из острых желаний нашего века стало желание собрать, как можно больший объем информации, который нас окружает, а затем переработать ее, проанализировать, чтобы получить из нее полезные составляющие, а в дальнейшем, применить их для улучшения жизни человека и окружающего его мира. Для этого многие великие ученых из разных областей трудились над разработкой методов сбора, обработки, хранения и анализа данных. Это привело к формированию новой науки, известной, как Data Science.

Банковский сектор не стал исключением и так же подвергся изменениям. В его процессы были внедрены технологий из области Data Science. Банк — это юридическое лицо, нацеленное на получение прибыли, поэтому для него крайне важно, чтобы любые технологические изменения увеличивали его прибыль или открывало новые возможности для ее преумножения. Одной из функций банка является кредитование, но из-за специфики данной услуги, при уклонении клиента от выплат, банк получит убытки. Для сведения количества таких случаев к минимуму, в банках применяются разного рода системы оценки платежеспособности клиента.

Одним из таких методов является скоринг. В основе скоринга заложены математические расчёты и статистика. Данный подход может быть реализован абсолютно разными методами: может быть разработана скоринговая карта, на основе которой сотрудник сопоставит ответы клиента и посчитает набранные баллы, после чего вынесет решение, основываясь на их количестве, посмотрев в соответствующую шкалу. Так же, скоринг может быть реализован в виде компьютерного алгоритма, в который переданы все ответы клиента.

В данной работе мне предстоит рассмотреть подробнее второй способ, а также оценить экономический эффект от внедрения моделей машинного обучения в задачах скоринга. Я считаю, что подключение к работе моделей машинного обучения позволит улучшить качество анализа данных клиента и лучше оценивать его платежеспособность, также это поможет избавиться от человеческого фактора и перенаправить сотрудников, занимавшихся подсчетом баллов по скоринговым картам, на основе ответов клиента на более сложные задачи, в которых применение машинного обучения не является эффективным или экономически выгодным. Все это окажет положительное влияние на экономические показателя банка.

Целью данной курсовой работы является оценка экономического эффекта от внедрения модели машинного обучения лучшего качества, по сравнению со старой «плохой» моделью

Для реализации поставленной цели мне будет необходимо:

- Изучить область исследования
- Проанализировать алгоритмы машинного обучения
- Выбрать подходящие для данной задачи алгоритмы
- Получить данные для исследования

- Предобработать и проанализировать данные
- На основании данных обучить выбранные алгоритмы и получить модели
- Произвести сравнение моделей и выбор лучшей
- Провести тесты для лучшей модели на корректность предсказаний
- Сравнить новую модель со старой
- Проанализировать влияние конкретных параметров
- Оценить экономическое эффект от внедрения данной модели

Предметом исследования являются данные клиентов банка, на основании которых будет построена новая модель, способная лучше выполнять поставленную задачу.
Объектом исследования выступает процесс классификации клиентов по их платежеспособности.

1. Исследование существующих работ в данной области

1.1 Статья «Как мы сократили время на разработку скоринговых моделей в пять раз, переключившись на Python»

В данной статье осуществляется разработка скоринговой модели. Основными условиями авторы ставят для себя интерпретируемость модели, которая поможет более корректно работать с предсказаниями модели, а также понятно объяснить принцип работы и причину появления частного результата менеджерскому составу и регулирующим органам.
Большое внимание уделяется обработке данных перед обучением. Особенно количеству признаков и их влиянию на предсказания. Для этого прежде всего все данные проходят процедуру группировки (binning), после чего вычисляется коэффициент весомости группы (WOE). Данный коэффициент необходим для вычисления значение информативности (IV), основываясь на значения которого авторы принимали решения о исключении признаков из выборки.
По мнению авторов лучшими метриками стали: ROC-AUC, Gini, F1_score.
По результатам исследования модель с использования показала хорошие результаты.

1.2 Статья «Скоринг: прогностическая сила»

В данной статье описывается как теоретические, так и практические моменты скоринга. Но самое главное, в данной статье авторы затрагивают экономический эффект от внедрения автоматизированных скоринговых систем. А также частоту необходимого переформирования модели. Авторами выделяется 2 типа скоринга: application и поведенческий. Первый тип строится на оценке кредитоспособности клиента, чтобы сразу отсеять «плохих» заемщиков, второй же используется для прогнозирования дефолта. На затраты очень сильно влияет «серая» зона, решения в которой требую вмешательства сотрудника.
Внедрение скоринговых алгоритмов алгоритмов своей разработки, в отличии от промышленных, ведет к увеличению рисков и увеличению сроков внедрению. Так же выделяется важность имплементации скоринговых систем, в связи с тем, что покупные скоринговые карты имеют крайне низкие коэффициенты качества (по данным статьи: 0.4 в диапазоне от 0 до 1)

2.Теоретическая часть

2.1 Постановка и формализация задачи машинного обучения

Начнем с формализованного описания машинного обучения. Data Science, которая упоминалась мной в самом начале широкая область и включает в себя большое количество более мелких областей, одной из них является машинное обучение.
Машинное обучение – занимается восстановлением функции зависимости по точкам. Для более подробного объяснения потребуются ввести множество объектов и истинное множество соответствующих им ответов. Задачей машинного обучения будет найти такую функцию аппроксимации, чтобы при получении на вход множества объектов мы получали результат наиболее приближенный к истинному множеству ответов. Выборка представляет из себя матрицу с m признаков и n ответами (целевыми признаками), то есть:

$$\begin{pmatrix} x_{1.1} & \cdots & x_{m.1} & y_1 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ x_{n.1} & \cdots & x_{m.1} & y_n \end{pmatrix}$$

2.1.1 Формализация множества признаков

Каждый объект в выборке описывается множеством признаком. Формально опишем множество объектов, каждый из которых описывается множеством признаком:

$$f_j: X \to Dj, \text{где } j = 1, \dots, d$$

$$x = (x_1, \dots, x_n) - \text{признаковое описание объекта x}$$

Для удобства множество объектов и множество признаков представляют в матричном виде.

Признаки разделяют на следующие типы:

1. Бинарные – признак принадлежит множеству $\{0, 1\}$
2. Категориальные – признак принадлежит конечному множеству категорий
3. Вещественные – признак, принадлежащий множеству вещественных чисел

2.1.2 Формализация множества ответов

Целевой признак или же ответ, неотъемлемая часть любой выборки, на которой предполагается обучение при обучении с учителем. Целевой признак может иметь разный тип в зависимости от задачи. Например, в задачах

регрессии ответы представлены в вещественном типе, а в задачах классификации целевой признак представлен категориальным типом, как последовательность элементов принадлежащих ограниченному числу классов.
Разберем задачу классификации более подробно. В таких задачах мы можем встретить следующие множества ответов:

1. Бинарная классификация: $Y \in \{0,1\}$ или $Y \in \{-1,+1\}$
2. Многоклассовая классификация $Y \in \{0,\dots,n\}$

2.1.3 Введение понятий: функция потерь, модель

Как я писал выше, задача машинного обучения построить аппроксимирующую функцию. Каждый этап обучения модели сводится к задаче оптимизации. Но как мы определим, что полученная нами функция достаточно хорошо восстанавливает зависимости? Для этого используется функция потерь, которая помогает определить, насколько точные ответы выдает нам модель. На больших выборках процесс обучения происходит итерационно, поэтому нам необходимо определять, не только итоговый результат, но и промежуточные, на каждой итерации. Функция потерь позволяет нам не только определить величину ошибки алгоритма, но и оптимизировать его, основываясь на этих данных. Но чтобы это было возможно к функции потерь в первом приближении предъявляются следующие требования:

1. Функция должна быть дифференцируема
2. Определена на области значений признаков
3. Функция должна быть гладкой

Минимизируя такую функцию мы сможем оптимизировать модель добиваясь все более лучших и лучших результатов.

Модель – это семейство параметрических функций:

Для оптимизации модели применяются такие алгоритмы как: градиентный спуск, стохастический градиентный спуск, momentum, RMSProp, Adam

2.2 Алгоритмы машинного обучения

2.2.1 Решающие деревья

Алгоритм машинного обучения, являющийся в общем случае k-ичным деревом. Принцип действия заключается в подборе порогов для решающего правила в каждом узле, в котором происходит разделение выборки на части при сравнении с порогом.
Решающее правило это функция, позволяющая определить в какую вершину

необходимо отправить тот или иной объект.
Отличительной чертой решающих деревьев является универсальность, они могут быть применены как к задачам регрессии, так и к задачам классификации, без существенных доработок.
Качество разделения объектов определяется с помощью критериев.
Основными критериями являются:

1. **Misclassification criteria**

$$G(k,t) = \frac{|L|}{|Q|} * H(L) + \frac{|R|}{|Q|} * H(R), \text{где}$$

Q - Исходная выборка
L, R - Выборки после разделения
|Q|, |L|, |R| - Мощности соответствующих выборок

$$H(X) = 1 - \max\{p_0, p_1\}$$

p_0, p_1-вероятность встретить объекты соответствующего класса в выборке

Чтобы добиться наилучшего результата, критерий следует минимизировать

2. **Критерий Джинни**

$$G(k,t) = \frac{|L|}{|Q|} * H(L) + \frac{|R|}{|Q|} * H(R), \text{где}$$

Q - Исходная выборка
L, R - Выборки после разделения
|Q|, |L|, |R| - Мощности соответствующих выборок

$$H(X) = 1 - \sum_{k=0}^{K} (p_k)^2$$

p_k-вероятность соответствующего класса

Чтобы добиться наилучшего результата, критерий следует минимизировать

3. **Энтропийный критерий**

$$G(j,t) = \frac{|L|}{|Q|} * H(L) + \frac{|R|}{|Q|} * H(R), \text{где}$$

Q - Исходная выборка
L, R - Выборки после разделения
|Q|, |L|, |R| - Мощности соответствующих выборок

$$H(X) = -\sum_{k=0}^{K} p_k(\log p_k)$$

p_k-вероятность соответствующего класса

2.2.2 Random Forest

Случайный лес – это метод алгоритм машинного обучения, который реализует ансамбль из решающих деревьев. В основе данного алгоритма лежит идея объединения технологии бэггинг и метод случайных подпространств. Разберем каждый из них подробнее

Бэггинг – технология, использующая композицию алгоритмов, каждый из которых обучается параллельно. В отличии от бустинга, в данной технологии классификаторы компенсируют ошибку путем голосования, а не путем исправления ошибок друг друга.

Метод случайных подпространств – в методе случайных подпространств алгоритмы обучаются на различных подмножествах признаков, которые генерируются случайным образом. Таким образом достигается снижение коррелированности между алгоритмами в ансамбле. В случае случайного леса алгоритмами выступают решающие деревья. Ансамбль, использующий метод случайного подпространства, можно построить, используя следующий алгоритм

1. Длина вектора признаков равна D.
2. L отдельных моделей в ансамбле.
3. Для каждой отдельной модели выберем x(x<D) как число признаков
4. Для отдельной модели создадим обучающую выборку, выбрав x признаков из D,после чего обучим модель.
5. Чтобы применить модель ансамбля к новому объекту, объединим результаты отдельных L моделей мажоритарным голосованием
6. Таким образом, случайный лес — это бэггинг над решающими деревьями, при обучении которых используется метод случайных подпространств.

2.2.3 Логистическая регрессия

Логистическая регрессия – метод, который моделирует вероятность принадлежности конкретного объекта к классу из ограниченного множества. В основу данного метода заложена идея, что пространство значений может быть разделено линейной границей. В зависимости от размерности нашей матрицы признаков линейная граница будет видоизменяться, например, при двух измерениях это будет прямая линия, при 3 измерениях – плоскость и так далее.

Логистическая регрессия строится на сигмоиде, внутри которой находится

линейная регрессия. Рассмотрим более подробно, почему данный подход работает и как мы переходим от линейной функции, которая возвращает нам значения из диапазона $(-\infty, +\infty)$, к вероятности класса из диапазона $[0,1]$:

Область значения функции линейной регрессии принадлежит множеству вещественных чисел, т.е.:

$$y = Xw^T \in R$$

где X - матрица признаков, w^T - вектор весов
Вероятность объекта быть положительным классом принадлежит отрезку от 0 до 1, т.е.:

$$p_+ \in [0,1]$$

Чтобы воспользоваться линейной регрессией для классификации необходимо привести область значений линейной регрессии к области значений вероятность объекта принадлежать положительному классу:

$$\frac{p_+}{1-p_+} \in [0, +\infty)$$

Прологарифмируем это отношение и получим, что оно принадлежит множеству вещественных чисел:

$$ln\left(\frac{p_+}{1-p_+}\right) \in R$$

Теперь области значений выражений совпадают и их можно приравнять

$$ln\left(\frac{p_+}{1-p_+}\right) = Xw^T <=> \left(\frac{p_+}{1-p_+}\right) = e^{(Xw^T)}$$

Выразим p_+ из полученного уравнения

$$p_+ = \frac{e^{(Xw^T)}}{1-e^{-(Xw^T)}}$$

Упростим получившуюся дробь:

$$p_+ = \frac{e^{(Xw^T)}}{1-e^{-(Xw^T)}} = \sigma(Xw^T) \in [0,1]$$

Мы получили сигмоиду, внутри которой находится линейная регрессия.

2.3 Методы предобработки

2.3.1 Weight of evidence

В процессе банковского скоринга часто встречаются ситуации, когда не одного веса для одного признака недостаточно. Например, молодые и пожилые люди возвращают кредиты хуже, поэтому нам необходимо разделить признак возраст на несколько корзин, каждая из которых будет отвечать за свою возрастную группу. А потом рассчитать WOE для нового признака и заменить на него. Это помогает:

1. Максимизировать значимость признака в бинарной модели
2. Максимизировать равномерность заполнения интервалов, что увеличивает репрезентативность

WOE можно вычислить по следующей формуле:

$$WOE = ln\left(\frac{\%good}{\%bad}\right)$$

где
%good – относительная частота появления положительных событий
%bad - относительная частота появления негативных событий

2.3.2 Information Value

На основании WOE мы можем вычислить Information Value, которое покажет нам предсказательную силу каждого признака, а также поможет отбросить признаки, если они практически не оказывают влияние на результат работы модели.
Information Value вычисляется по следующей формуле:

$$Information\ Value\ = WOE\ *\ (\%good\ -\ \%bad)$$

где
%good – относительная частота появления положительных событий
%bad - относительная частота появления негативных событий

2.3.3 Обработка выбросов

Выбросы – точки, которые выпадают из общей тенденции последовательности, можно сказать, что они принадлежать отдельной популяции. Такое наблюдение находится очень далеко от других членов последовательности. Выбросы несут в себе опасности для итогового результата. Первое чем опасны выбросы – это искажение статистик и расчётов. При вычислении показателей их значения могут быть сильно

искажены из-за наличия выбросов, что повлечет за собой искажения реальной картины. Например, у нас есть выборка возрастов клиентов, которые заказывают товар1. В данной выборке по причине компьютерного сбоя некоторые элементы выпали из диапазона длительности жизни человека (200, 300 и тд). Теперь, когда мы захотим посчитать средний возраст клиента, который покупает товар1 мы получим искаженную картину, где средний возраст может составить более 100 лет, что не является правдоподобным. Так же выбросы могут повлечь за собой некорректный анализ признака и вместо того, чтобы исключить выбросы мы удалим корректные данные.
Одним из методов борьбы с выбросами, является межквартильный размах. Для его применения нам необходимо вычислить квантиль 25% и 75%, а затем составить неравенство:

$$q_{25} * 1.5 \leq x \leq q_{75} * 1.5$$

где

q_{25} – квантиль 25%

q_{75} – квантиль 75%

Если значения признака не попадает в данное неравенство, то объект с данным признаком удаляется

2.4 Метрики

2.4.1 TPR

TPR – отношение правильно классифицированных объектов положительного класса (TP) на сумму TP и неправильно классифицированных объектов позитивного класса (FN):

$$TPR = \frac{TP}{TP + FN}$$

Так же данную величину называют recall, она отражает насколько качественно(полно) классификатор определяет положительный класс

2.4.2 FPR

FPR – величина отражает насколько качественно(полно) классификатор определяет отрицательный класс

$$FPR = \frac{FP}{TN + FP}$$

2.4.3 ROC-кривая

Качество классификатора позволяет оценить кривая ошибок. Отражает зависимость доли верных положительных классификаций от доли ошибочно положительных при изменении порогового значения.

2.4.4 ROC-AUC

Количественная интерпретация ROC кривой. Это площадь под соответствующей кривой. Чем выше показатель к 1, тем более качественно классификатор разделяет классы. Значения данного показателя находятся в диапазоне [0,1]

2.4.5 F мера

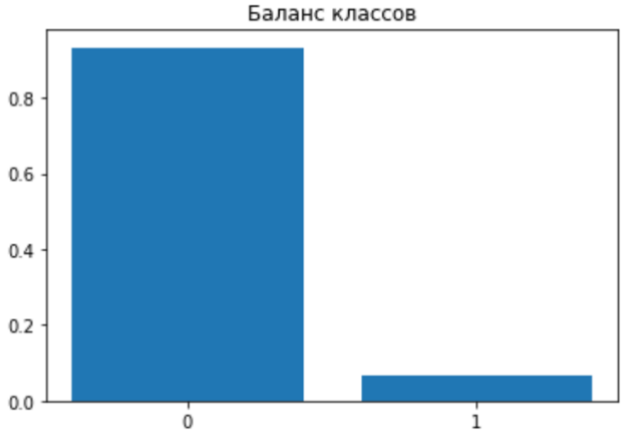
Метрика, представляющая собой гармоническое среднее между точностью и полнотой. Позволяет найти баланс между этими показателями. Так же меняя коэффициент бетта изменяет вес полноты и точности, данная особенность необходима, когда мы понимаем, что какой то из параметров для нас важнее. Общая формула F меры выглядит следующим образом:

$$F = (\beta^2 + 1)\frac{Precision * recall}{\beta^2 * precision + recall}$$

Если параметр $0 < \beta < 1$, то мы отдаем приоритет точности, в противном же случае увеличивается влияние полноты. Коэффициент не может быть равен 0, так как в этом случае независимо от наших результатов F мера будет неопределенна

3. Практическая часть

Исходные данные получены с сайта Kaggle. 9 лет назад на данном портале проводилось соревнование по кредитному скорингу. В котором было необходимо разработать модель бинарного классификатора, предсказывающего метку класса для каждого клиента банка. После получения необходимых для работы данных я начал предобработку и анализ. Но перед этим определим список библиотек, которые будут использованны в данной работе: pandas, numpy, matplotlib, seaborn, sklearn, tqdm, scipy и random. Теперь перейдем к первичной предобработке и анализу.
Первое что мне бросилось в глаза – несбалансированность классов, как видно на графике ниже, нулевой класс составляет более 80% от выборки.



Если рассмотреть данную ситуацию более подробно, то мы увидим, что нулевой класс составляет чуть больше 93% выборки:

```
0    0.93316
1    0.06684
```

Это очень большая проблема, которая ведет к некорректному процессу обучению, и если оставить все как есть, то наш классификатор практически всегда предсказывать нулевой класс. Для того чтобы это исправить мной были сделаны следующие шаги:
1. Использование SMOTE (synthetic minority oversampling technique), данный метод генерирует новые объекты минорного класса на основании уже существующих объектов, тем самым балансируя его
2. Отказ от использования такой метрики, как accuracy, так как на несбалансированных выборках она не является репрезентативной

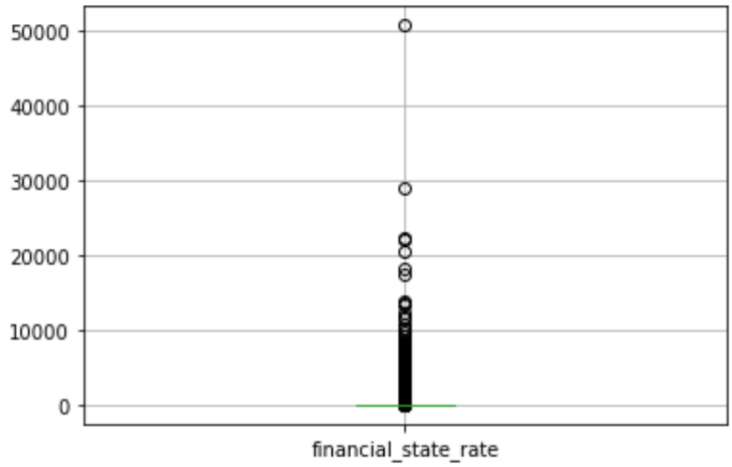
Дальше я проанализировал каждый признак по отдельности:

1. Признак age: в данном признаке присутствовало аномальное значение, у одного объекта возраст был равен 0, что не может соответствовать действительности. Эмпирическим путем я выяснил, что такой объект 1 и следующий минимальный возраст равен 21. Так как количество объектов с аномальным было крайне мало, я удалил данный объект.
2. Признак financial_state_rate: Исследуя данный признак, я нашел большое количество аномальных значений. Первым делом я посмотрел распределение

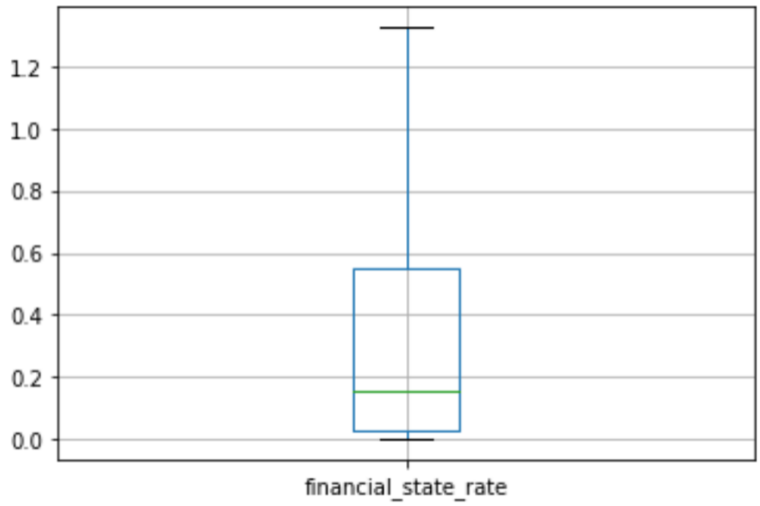
признака по квантилям и увидел, что 75% значений выборке меньше 0.55, а в остальных 25% встречаются огромные значения, которые далеко выходят за 1:

```
min          0.000000
25%          0.029867
50%          0.154176
75%          0.559044
max      50708.000000
```

Тогда я решил посмотреть данный признак на графике, и увидел, что огромное количество значений не попадают в межквартильный размах, это можно заметить на графике ниже:

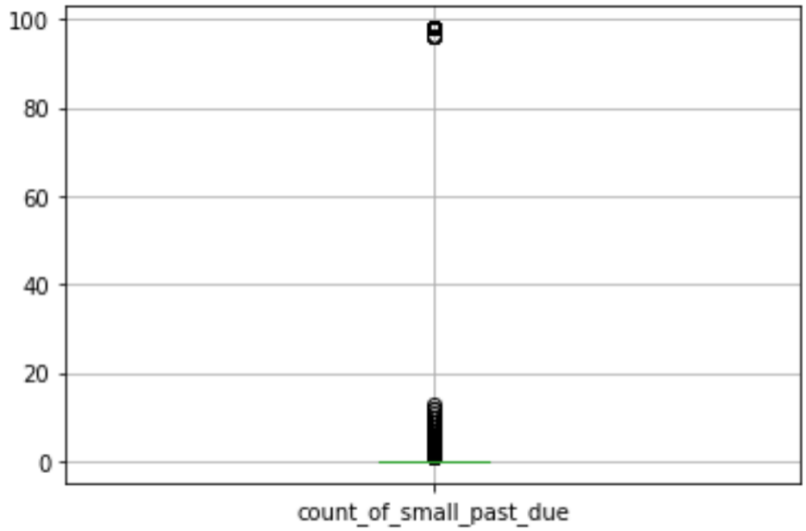


Поэтому следующим шагом необходимо было очистить данный признак от аномалий, после очистки признак стал выглядеть таким образом:



В нем отсутствуют значения, которые бы выпадали из размаха. В процессе обработки было удалено 0.5% выборки.

3. Признак count_of_small_past_due: В признаке приуставали выбросы, но из за смысловой нагрузки признака мне показалось неправильным удалять абсолютно все выбросы, поэтому мной было принято решение выкинуть только те объекты, признак у которых был больше 80, как видно на графике эти объекты очень сильно выбивались из общей картины:

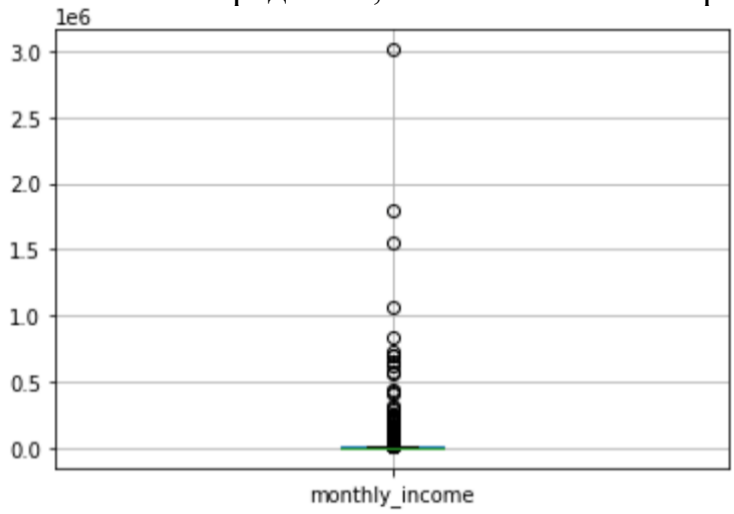


Таких объектов оказалась немного (255), поэтому целесообразнее их удалить.

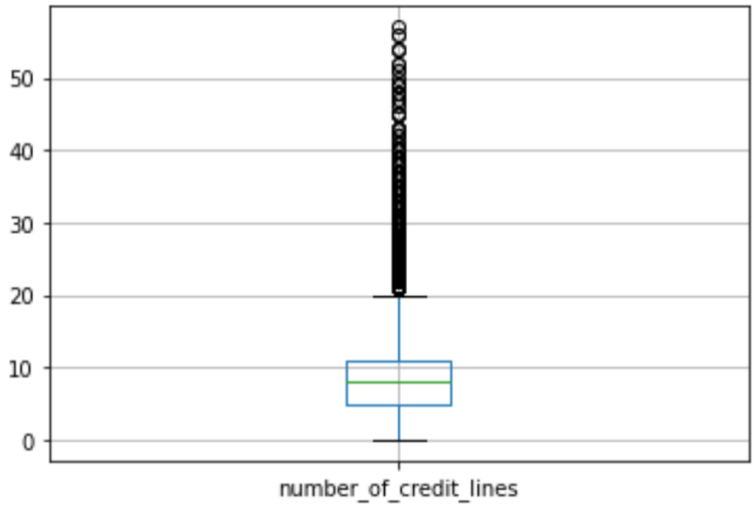
4. Признак debt_ratio: в данном признаке не может быть значений больше, но как мы видим ниже, в нем присутствуют такие объекты:

```
min            0.000000
25%            0.176107
50%            0.367165
75%            0.868529
max       329664.000000
Name: debt_ratio, dtype:
```

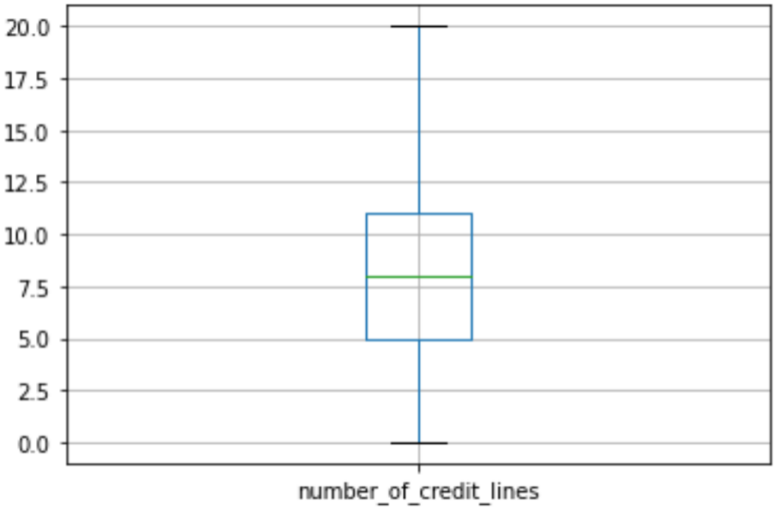
Избавимся от всех объектов, у которых данный признак был больше 1.
5. Признак monthly_income: Данный признак кроме большого количества выбросов содержит так же и пропуски. Прописки стоит заполнить медианным значением. Но перед этим, стоит очистить выбросы



6. Признак number_of_credit_lines: содержит выбросы, вот график до очистки:

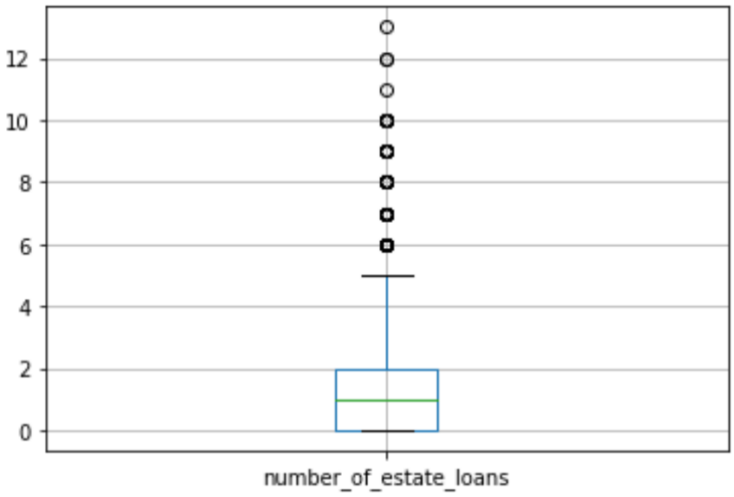


А вот после очистки:



7. Признак count_of_big_past_due: в данном признаке каких либо исправлений не потребовалось

8. Признак number_of_estate_loans: здесь мы можем видеть, что присутствует небольшое количество выбросов:

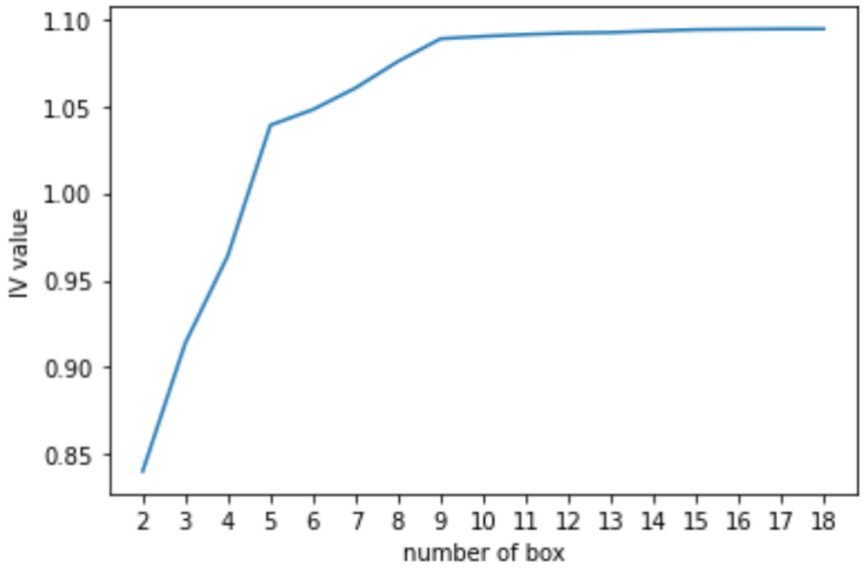


9. Признак count_of_medium_past_due: в данном признаке каких-либо исправлений не потребовалось

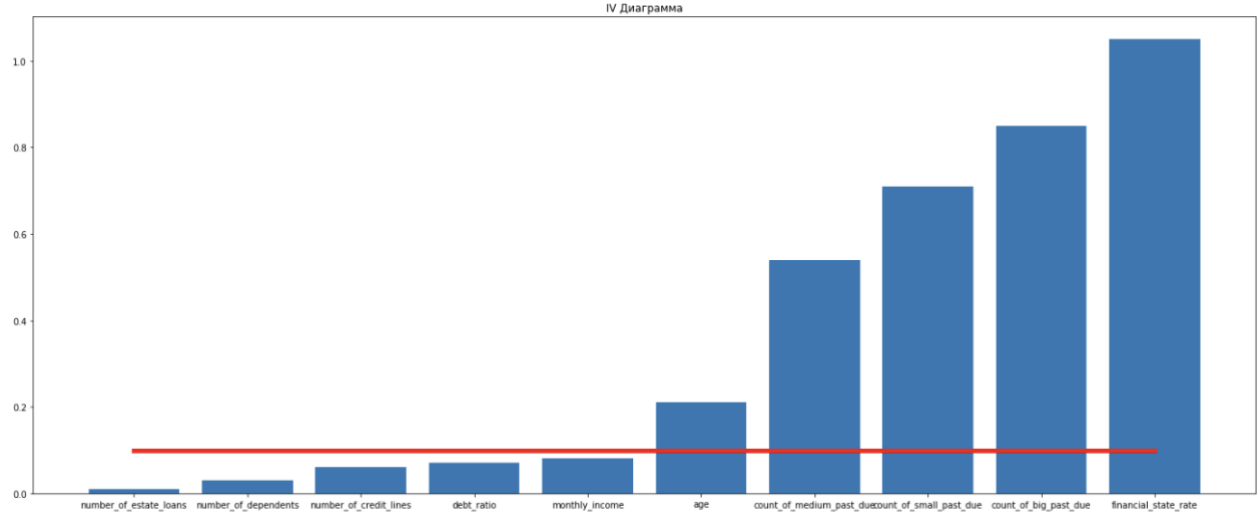
10. Признак number_of_dependents: Признак, отражающий количество подопечных, ограничим их количество до 10

После анализа и обработки признаков проверим выборку на дубликаты. Было найдено 96 дубликатов, удалим их.

Займемся категоризацией вещественных признаков, для этого применим функцию, которая рассчитывает Information value для каждого признака последовательно перебирая количество групп. Вот пример, для признака financial_state_rate:



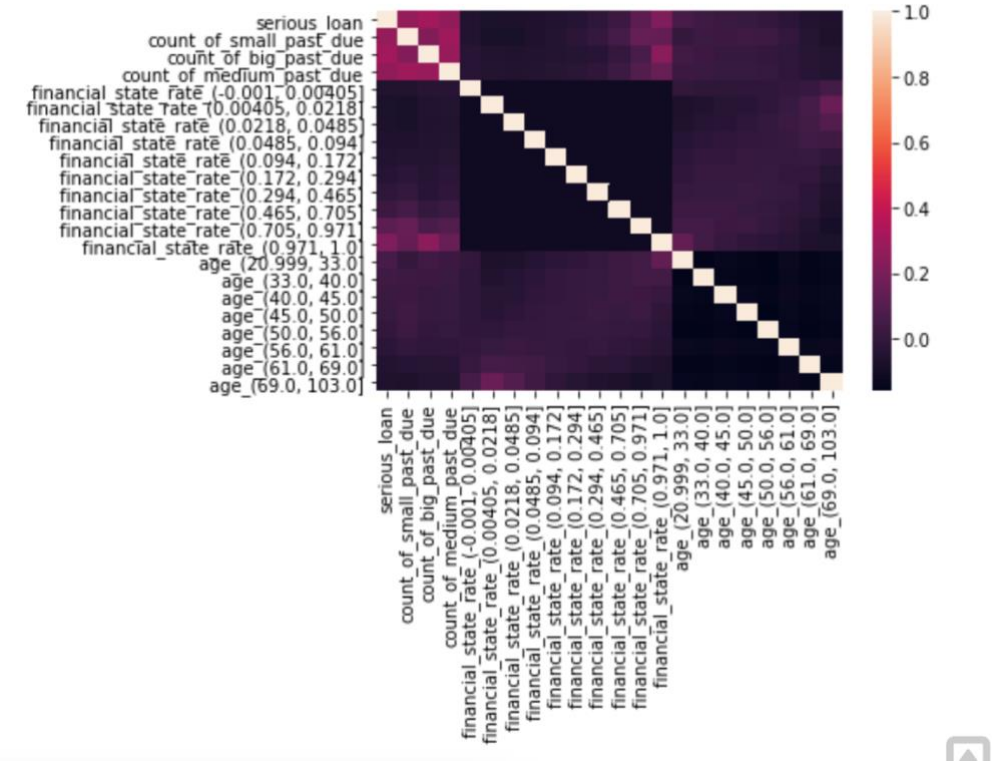
На данном графике мы можем видеть, что значение IV стабильно растет при увеличении количества групп до 10, а потом выходит на плато. На основании таких графиков выберем количество групп для каждого признака. После этого рассчитаем IV для получившихся признаков и откинем все признаки, у которых значения силы предсказания будет меньше 0.1(красная линия)



Теперь, когда выборка подготовлена, посмотрим на корреляцию признаков:

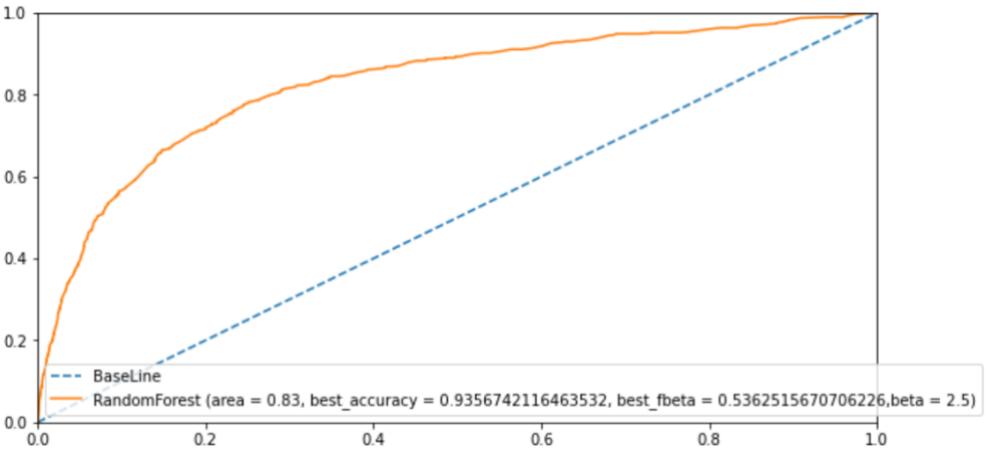
	serious_loan	count_of_small_past_due	count_of_big_past_due	count_of_medium_past_due
serious_loan	1.00	0.27	0.32	0.26
count_of_small_past_due	0.27	1.00	0.22	0.30
count_of_big_past_due	0.32	0.22	1.00	0.29
count_of_medium_past_due	0.26	0.30	0.29	1.00

После кодирования категориальных признаков:



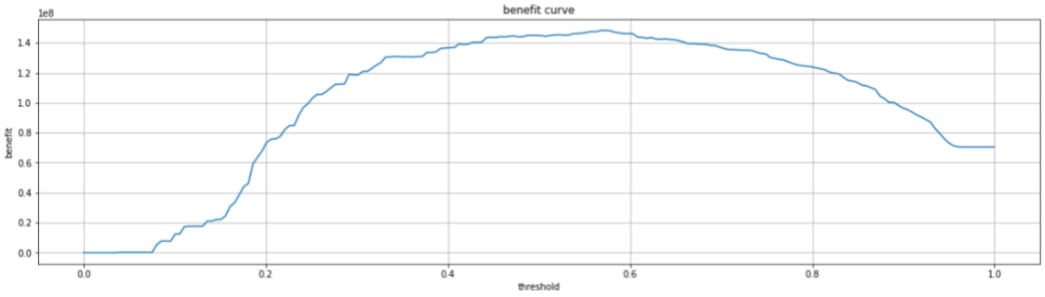
Как можно заметить, сильная корреляция отсутвует
Закодируем все категориальные признаки с помощью OneHotEncoding.
С помощью train_test_split из библиотеки sklearn разделим выборки на тренировочную, валидационную и тестовую. На тренировочную отдадим 80% выборки, а на тестовую и валидационную по 10%.

Рассмотрим модель, построенную на алгоритме случайного леса:
Прежде всего составим pipline, в который войдет SMOTE (для балансировки выборки), Стандартизация, и модель классификатора. Посмотрим на результаты модели на валидационной выборке

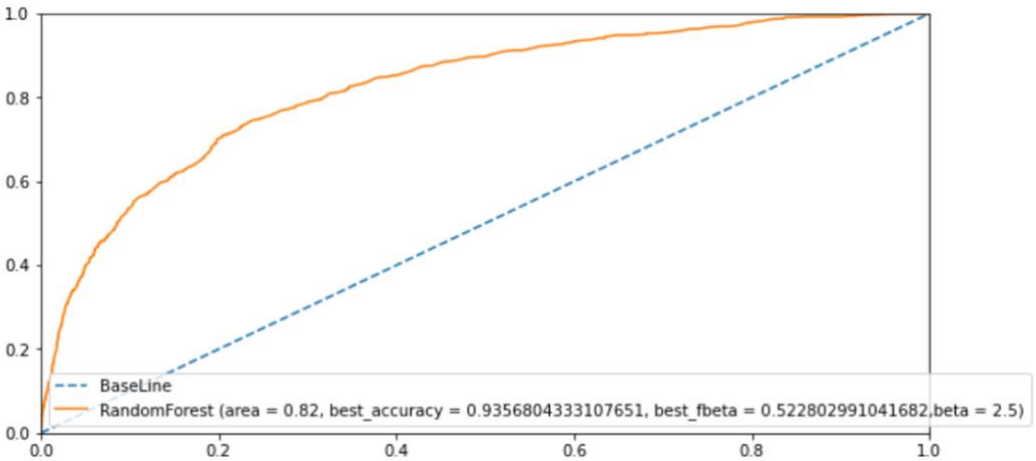


Как видно из графика площадь под ROC кривой составила 0.83 (данные значения мы достигаем при пороге равном 0.45), что является достаточно хорошим показателем. Так как для нас важна не только качество модели, но и прибыль, которая нам эта модель приносит, подберём порог таким образом, чтобы максимизировать прибыль:

```
max benefit 148306023.71381503
best_threshold 0.5678391959798995
```

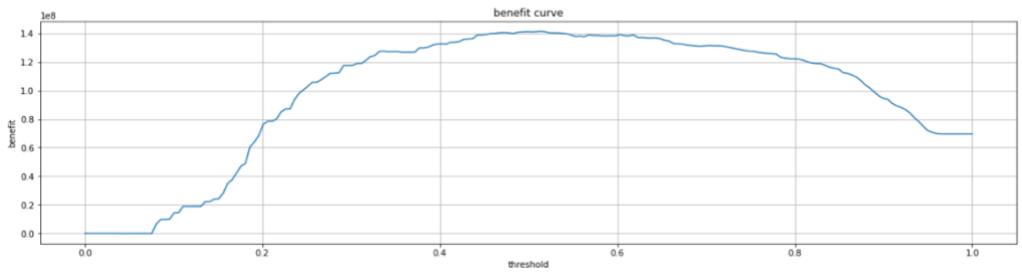


Максимальную прибыль мы получаем при более высоком пороге, который равен 0.57. Посмотрим на результаты тестовой выборки:



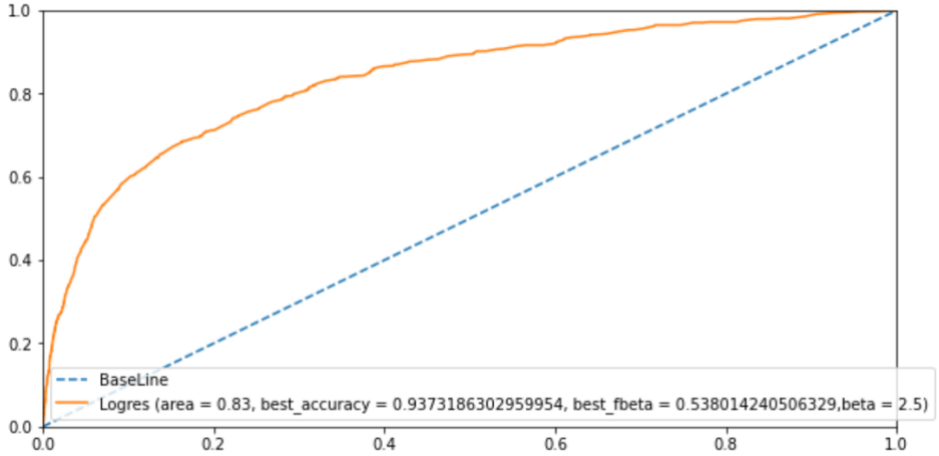
На тестовой выборке параметр площади под кривой немного меньше, что ожидаемо, так же для получения наилучшего значения площади под графиком порог возрастает до 0.5, что говорит нам о том, что модель без оптимизации порога дает отличные показатели метрики. Так же можно заметить, что и на кривой выгоды порог отличается всего лишь на 0.1

```
max benefit 141312645.42726842
best_threshold 0.5125628140703518
```



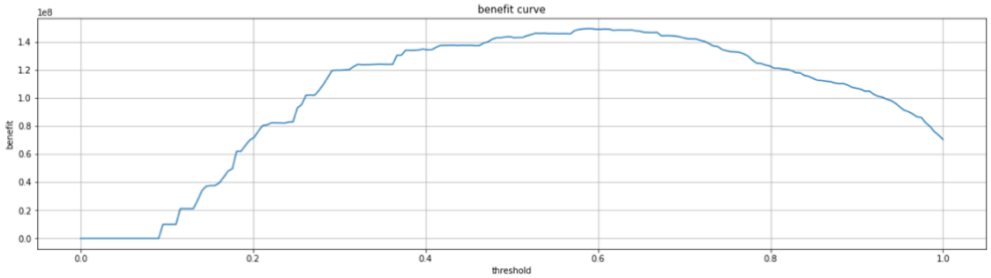
Рассмотрим модель, построенную на алгоритме логистической регрессии:

В данной модели pipeline будет отличаться только самой алгоритмом



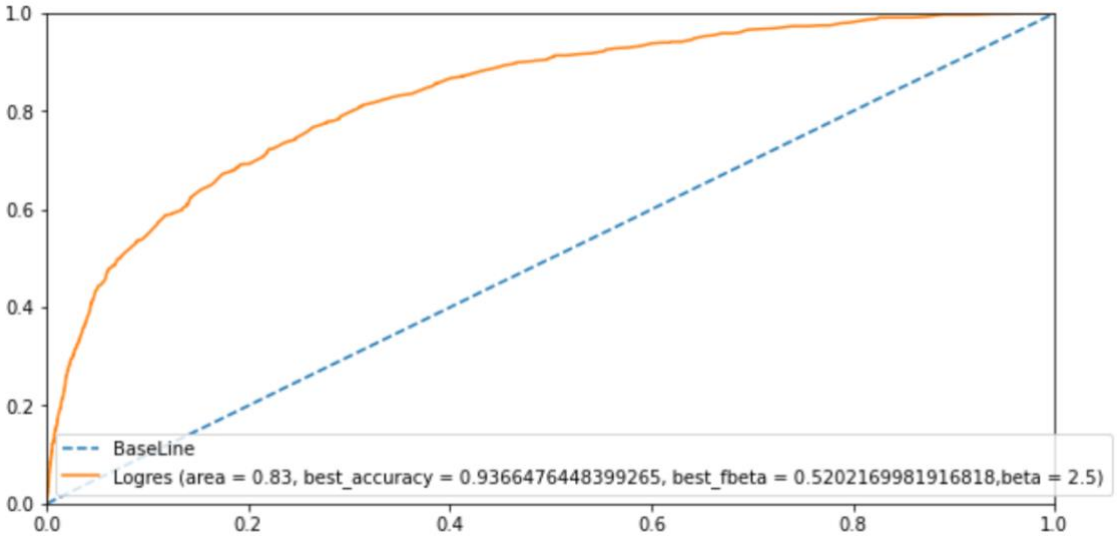
Можем наблюдать, что значения площади под графиком совпадают с решающем лесом, но вот F мера отличается на 0.002. Посмотри на кривую выгод:

```
max benefit 149369195.66297895
best_threshold 0.5879396984924623
```



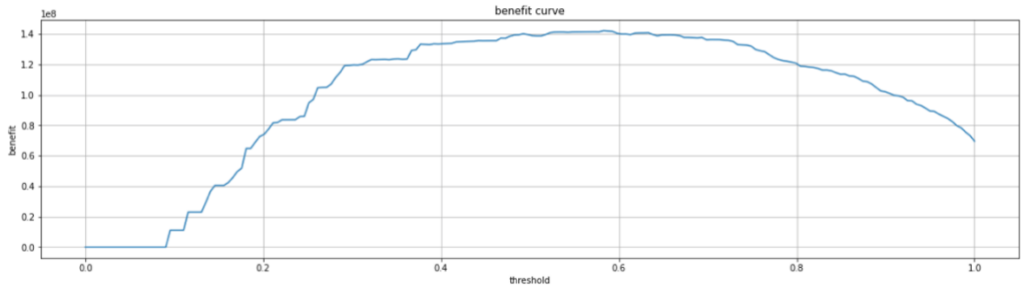
Если сравнить эту кривую выгод с кривой, полученной с помощью случайного леса, мы заметим, что прибыль отличается на 1 063 172 в пользу логистической регрессии.

Теперь посмотрим на результаты, основанные на тестовой выборке



В отличии от случайного леса, здесь на тестовой выборке значение площади под графиком не падает, а остается таким же.

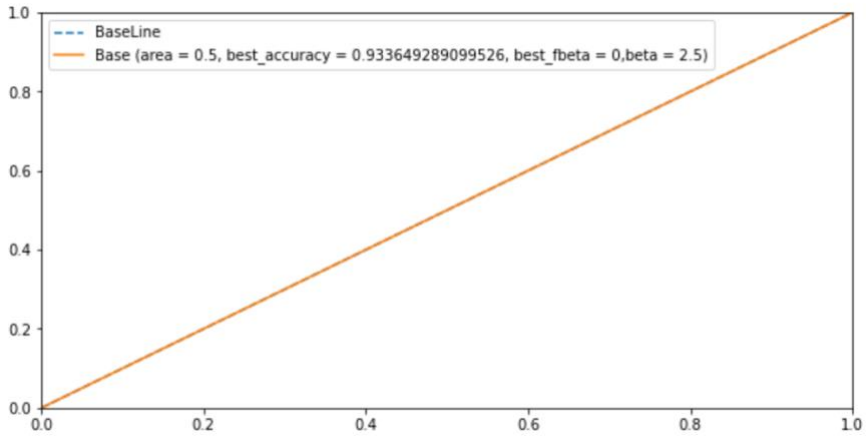
```
max benefit 142216528.91770393
best_threshold 0.5829145728643216
```



Здесь мы так же наблюдаем рост прибыли по сравнению с тестовой выборкой в модели случайного леса. Так как для банка уровень дохода является не последним приоритетом, логичнее всего будет выбрать модель, построенную на основе логистической регрессии.

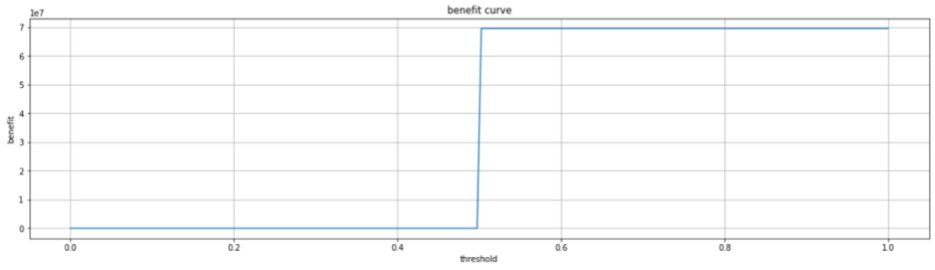
Теперь рассмотрим базовую модель:

В данном случае нам нет необходимости смотреть результаты на валидационной выборке, поэтому сразу перейдем к тестовой:



Здесь мы можем видеть, что F мера равна 0, а площадь под графиком равна 0 это означает, что наша модель ничем не отличается от случайного угадывания. Посмотрим какую прибыль можно получить, используя данную модель:

```
max benefit 69592586.60950604
best_threshold 0.5025125628140703
```



Несложно заметить, что прибыль очень сильно упала, использование данной модели не имеет смысла.

Так же были проведены тесты модели, которые показали отличные результаты:

1. Тест 1: Коэффициент Джини (Границы: больше 0.35 – зеленый, от 0.25 до 0.35 – желтый, остальные случаи – красный)

```
In [125]: get_gini(target_valid,log_prob_preds[:,1])

Out[125]: 0.6703019504080967
```

   a.

2. Тест 2: Эффективность ранжирования отдельных факторов (Границы: больше 0.35 – зеленый, от 0.25 до 0.35 – желтый, остальные случаи – красный)

   a.

name	gini
financial_state_rate_(-0.001, 0.00405]	0.590243
financial_state_rate_(0.00405, 0.0218]	0.635698
financial_state_rate_(0.0218, 0.0485]	0.504969
financial_state_rate_(0.0485, 0.094]	0.355559
financial_state_rate_(0.094, 0.172]	0.405840
financial_state_rate_(0.172, 0.294]	0.423344
financial_state_rate_(0.294, 0.465]	0.434013
financial_state_rate_(0.465, 0.705]	0.364561
financial_state_rate_(0.705, 0.971]	0.413953
financial_state_rate_(0.971, 1.0]	0.517824
age_(20.999, 33.0]	0.604199
age_(33.0, 40.0]	0.688537
age_(40.0, 45.0]	0.603575
age_(45.0, 50.0]	0.647097
age_(50.0, 56.0]	0.564277
age_(56.0, 61.0]	0.715410
age_(61.0, 69.0]	0.625961
age_(69.0, 103.0]	0.648896

4. Заключение

В ходе разработки мы выбрали «лучшую» модель, у который на тестовой выборке метрики получились следующие:

1. ROC-AUC: 0.83
2. F_2.5 = 0.52
3. Лучший пороговое значение (threshold): 0.58
4. Максимальная прибыль: 142 216 528 у.е.

«Лучшая» модель, построенная на алгоритме логистической регрессии, имеет следующие параметры:

Параметр	Вес
count_of_small_past_due	0.55526978
count_of_big_past_due	0.63494735
count_of_medium_past_due	0.3744034
financial_state_rate_(-0.001, 0.00405]	-0.25223727
financial_state_rate_(0.00405, 0.0218]	-0.39216663
financial_state_rate_(0.0218, 0.0485]	-0.36000213
financial_state_rate_(0.0485, 0.094]	-0.31934846
financial_state_rate_(0.094, 0.172]	-0.24551615
financial_state_rate_(0.172, 0.294]	0.22009412
financial_state_rate_(0.294, 0.465]	-0.11350007
financial_state_rate_(0.465, 0.705]	0.00426792
financial_state_rate_(0.705, 0.971]	0.15823045
financial_state_rate_(0.971, 1.0]	0.26564951
age_(20.999, 33.0]	0.02569094
age_(33.0, 40.0]	-0.01464682
age_(40.0, 45.0]	-0.04339154
age_(45.0, 50.0]	-0.03439827
age_(50.0, 56.0]	-0.04455185
age_(56.0, 61.0]	-0.09394314
age_(61.0, 69.0]	-0.15188677
age_(69.0, 103.0]	-0.13942765

В теме мы ставили перед собой задачу оценить экономический эффект, от внедрения новой модели. Если за старую модель брать, модель, построенную на основе случайного леса, то экономический эффект оказался положительный и составил 903 833 условных единиц (в 1.006 раза больше). Если же за старую модель брать модель предсказания, которой близки к случайным гаданиям, то положительный эффект составит 72 623 942 условных единиц (в 2.04 раза больше)

5. Источники

1. «Как мы сократили время на разработку скоринговых моделей в пять раз, переключившись на Python» [Электронный ресурс] – 2018. Режим доступа: https://habr.com/ru/company/idfinance/blog/421091/

2. «Скоринг: прогностическая сила» [Электронный ресурс] – 2008. Режим доступа: https://www.banki.ru/news/bankpress/?id=637417

3. «Гущин Александр, Методы ансамблирования обучающихся алгоритмов» [Электронный ресурс] – 2015. Режим доступа: http://www.machinelearning.ru/wiki/images/5/56/Guschin2015Stacking.pdf

4. «Виталий Радченко, Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес» [Электронный ресурс] – 2015. Режим доступа: https://habr.com/ru/company/ods/blog/324402/

5. «Е. А. Соколов, Решающие деревья» [Электронный ресурс] – 2018. Режим доступа: https://www.hse.ru/mirror/pubs/share/215285956

6. «Антон Сметанин, Логистическая регрессия» [Электронный ресурс] – 2021 Режим доступа: https://github.com/Stuksus/LogisticRegression/blob/master/LogisticRegression.ipynb

7. «Коэффициент WoE» [Электронный ресурс]. Режим доступа: https://wiki.loginom.ru/articles/coefficient-woe.html

8. «5 способов обнаружить выбросы / аномалии, которые должен знать каждый специалист по данным (код Python)» [Электронный ресурс] – 2019 Режим доступа: https://www.machinelearningmastery.ru/5-ways-to-detect-outliers-that-every-data-scientist-should-know-python-code-70a54335a623/

9. «DMAGIC, Выбросы. Часть 1: кто это такие и почему они опасны?» [Электронный ресурс] – 2012 Режим доступа: http://sixsigmaonline.ru/baza-znanij/22-1-0-291

10. «ROC-кривая» [Электронный ресурс] – 2020 Режим доступа: https://ru.wikipedia.org/wiki/ROC-кривая

11. «Кривая ошибок» [Электронный ресурс] – 2020 Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Кривая_ошибок

12. «Оценка классификатора (точность, полнота, F-мера)» [Электронный ресурс] – 2012 Режим доступа: http://bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html

6. Приложение

Приложение 1

Ссылка на код: https://github.com/Stuksus/Credit_scoring_model/blob/main/main.ipynb