РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту на тему:

«Применение алгоритмов классификации в задаче прогнозирования оттока клиентов»

Мамаев А.А.

Москва

2017

Оглавление

Введение

Задача прогнозирования оттока клиентов остро стоит перед компаниями работающими на потребительском секторе рынка (B2C) близком к насыщению по предоставляемой ими услуге. Конкурентная среда рынков с подобной структурой формирует особые условия, в которых привлечение новых клиентов оказывается значительно дороже удержания имеющихся. Особенности сектора B2C (большой объем клиентской базы, недостаточно тесные взаимоотношениями между продавцом и потребителем) приводят к большим затратам на обнаружение клиентов склонных к оттоку. Применение математических методов классификации для решения данной задачи позволяет значительно повысить эффективность удержания клиентов. Класс пользователей с высоким риском ухода выделяется классификатором на основе информации, собираемой компанией: анкет, данных о составе услуг, о частоте их использования, о регулярности платежей, и т.д.

В представленной работе описан процесс создания классификатора для конкретного эмипирческого материала – 40 тысяч клиентов French Telecom company Orange – одного из мировых лидеров в области телекоммуникационных услуг (более 170 млн. пользователей). Рассмотрены различные методы предобработки данных и отбора значимых признаков. Оценено влияние предобработки на качество линейных методов классификации, «случайного леса», градиентного бустинга над решающими деревьями. Опробован «stacking»-подход к решению задачи. Проведен расчет экономического эффекта от применения разработанной модели.

Программная реализация алгоритмов обработки и классификации выполнена на языке Python 2.7 [1] в интерактивной оболочке Jupyter Notebook с использованием библиотек pandas [2], skipy [3], sklearn [4], seaborn [5].

1 Основные понятия и определения

Задано множество *объектов* X, множество *допустимых ответов* $Y = \{-1, +1\}$, и существует *целевая функция* (target function) $y^*: Y \to X$, значения которой $y_i = y^*(x_i)$ известны только на конечном подмножестве объектов $\{x_1, \dots, x_l\} \in X$. Совокупность пар $X^l = (x_i, y_i)_{i=1}^l$ называется *обучающей выборкой* (training sample). [6]

Задача классификации заключается в том, чтобы по выборке X^l *восстановить зависимость* y^*, то есть простроить *решающую функцию* (decision function) $a: X \to Y$, которая приближала бы целевую функцию $y^*(x)$, причем не только на объектах обучающей выборки, но и на всём множестве X.

Решающая функция должна допускать эффективную компьютерную реализацию; по этой причине будем называет её *алгоритмом классификации* или просто *классификатором*.

1.1 Объекты и признаки

Признак f объекта x - это результат измерения некоторой характеристики объекта. Формально признаком называется отображение $f: X \to D_f$, где $D_f -$ множество допустимых значений признака.

В зависимости от природы множества D_f признаки делятся на несколько типов.

Если $D_f = \{-1, +1\}$, то $f -$ *бинарный* признак;

Если $D_f -$ конечное множество, то $f -$ категориальный признак.

Если $D_f -$ конечное упорядоченное множество, то $f -$ порядковый признак.

Если $D_f = \mathbb{R}$, то $f -$ вещественный признак.

Пусть имеется набор признаков $f_1, \dots, f_n$. Вектор $(f_1(x), \dots, f_n(x))$ называют признаковым описанием объекта $x \epsilon X$. В дальнейшем мы не будем различать объекты из X и их признаковые описания, полагая $X = D_{f_1} \times \dots \times D_{f_n}$. Совокупность признаковых описаний всех объектов выборки X^l, записанную в виде таблицы размера $\ell \times$ n, названются *матрицей объектов-признаков:*

$$F = \begin{pmatrix} f_1(x_1) & \cdots & f_n(x_1) \\ \vdots & \ddots & \vdots \\ f_1(x_l) & \cdots & f_n(x_l) \end{pmatrix}$$

1.2 Модель алгоритмов классификации и метод обучения

Моделью алгоритмов классификации называется параметрическое семейство отображений $A = \{ a(x, \theta) | \theta \epsilon \Theta \}$, где $g: X \times \Theta \to Y -$ некоторая фиксированная функция, $\Theta -$ множество допустимых значений параметра θ, называемое пространством параметров или пространством поиска (*search space*).

Широко используются *линейные модели алгоритмов классификации* с вектором параметров $\theta = (\theta_1, \dots, \theta_n) \epsilon \Theta = \mathbb{R}^n$:

$$a(x,\theta) = sign \sum_{j=1}^{n} \theta_j f_j(x) = sign \langle x,\theta \rangle$$

Иными словами если $\langle x,\theta \rangle > 0$, то алгоритм a относит объект x к классу +1, иначе к классу −1.

Процесс подбора оптимальных параметров модели θ по обучающий выборке X^l называются *настройкой* (fitting) или *обучением* (training, learning) алгоритма $a \in A$.

Метод обучения (learning algorithm) – это отображение $\mu: (X \times Y)^l \to A$, которое произвольной конечной выборке $X^l = (x_i, y_i)_{i=1}^l$ ставит в соответствие некоторый алгоритм $a \in A$. Говорят также, что метод μ строит алгоритм а по выборке X^l.

Итак, в задачах обучения по прецедентам чётко различаются два этапа.

На этапе *обучения* метод μ по выборке X^l строит алгоритм $a = \mu(X^l)$.

На этапе *применения (*prediction*)* для новых объектов x выдает ответы $y = a(x)$.

Этап обучения, как правило, сводится к поиску параметров модели, доставляющих оптимальное значение заданному функционалу качества.

1.3 Функционал качества

Функция потерь (loss function) – это неотрицательная функция $\mathcal{L}(a,x)$, характеризующая величину ошибки алгоритма a на объекте x. Если $\mathcal{L}(a,x) = 0$, то ответ а (x) называется корректным.

Функционал качества алгоритма a на выборке X^l:

$$Q(a,X^l) = \frac{1}{l}\sum_{i=1}^{l} \mathcal{L}(a,x_i)$$

Функционал Q называют также функционалом *средних потерь* или *эмпирическим риском*, так как он вычисляется по эмпирическим данным $(x_i, y_i)_{i=1}^l$.

Пример функционала с «логистической» функцией потерь для задачи бинарной классификации:

$$Q(a,X^l) = \frac{1}{l}\sum_{i=1}^{l} (1+y_i)\ln\left(1+e^{-y_i \cdot \langle \theta, x_i \rangle}\right) + (1-y_i)\ln\left(1+e^{-y_i \cdot \langle \theta, x_i \rangle}\right)$$

Классический *метод обучения*, называемый *минимизацией эмпирического риска*, заключается в том, чтобы найти в заданной модели A алгоритм a, доставляющий минимальное значение функционалу качества Q на заданной обучающей выборке X^l:

$$\mu(X^l) = \arg \underbrace{min}_{\theta}\ Q(a,X^l)$$

1.4 Вероятностная постановка задачи обучения

Элементы множества X – это не реальные объекты, а лишь доступные данные о них. Данные могут быть неточными, поскольку измерения значений признаков $f_j(x)$ и целевой зависимости $y^*(x)$ обычно выполняются с погрешностями. Данные могут быть неполными, поскольку измеряются не все мыслимые признаки, а лишь физически доступные для измерения. В результате одному и тому же описанию x могут соответствовать различные объекты и различные ответы. В таком случае $y^*(x)$, строго говоря, не является функцией. Устранить эту некорректность позволяет вероятностная постановка задачи.

Вместо существования неизвестной целевой зависимости $y^*(x)$ предположим существование неизвестного вероятностного распределения на множестве $X \times Y$ с плотностью $p(x,y)$, из которого случайно и независимо выбираются ℓ наблюдений $X^l = (x_i, y_i)_{i=1}^l$. Такие выборки называются *простыми или случайными одинаково распределёнными* (independent identically distributed, i.i.d.).

Вероятностная постановка задачи считается более общей, так как функциональную зависимость $y^*(x)$ можно представить в виде вероятностного распределения $p(x,y) = p(x)p(y|x)$, положив $\mathrm{p}(y|x) = \delta(y - y^*(x))$, где $\delta(z)$ – дельта-функция.

1.5 Принцип максимума правдоподобия

При вероятностной постановке задачи вместо модели алгоритмов $g(x,\theta)$, аппроксимирующей неизвестную зависимость $y^*(x)$, задаётся модель совместной плотности распределения объектов и ответов $\phi(x,y,\theta)$, аппроксимирующая неизвестную плотность $p(x,y)$. Затем определяется значение параметра θ, при котором выборка данных $X\ell$ максимально правдоподобна, то есть наилучшим образом согласуется с моделью плотности.

Если наблюдения в выборке X^l независимы, то совместная плотность распределения всех наблюдений равна произведению плотностей $p(x,y)$ в каждом наблюдении: $p(X^l) = p(x_1,y_1) \cdot \ldots \cdot p(x_l,y_l)$.

Подставляя вместо $p(x,y)$ модель плотности $\phi(x,y,\theta)$, получаем *функцию правдоподобия* (likelihood):

$$L(\theta, X^l) = \prod_{i=1}^{l} \phi(x_i, y_i, \theta)$$

Чем выше значение правдоподобия, тем лучше выборка согласуется с моделью. Значит, нужно искать значение параметра θ, при котором значение $L(\theta, X^l)$ максимально. В математической статистике это называется *принципом максимума правдоподобия*. [7]

1.6 Связь максимизации правдоподобия с минимизацией эмпирического риска

Вместо максимизации L удобнее минимизировать функционал $-\ln L$ поскольку он аддитивен (имеет вид суммы) по объектам выборки:

$$-L(\theta, X^l) = \sum_{i=1}^{l} \phi(x_i, y_i, \theta) \to \underbrace{min}_{\theta}$$

Этот функционал совпадает с функционалом эмпирического риска $Q(a, X^l)$, если определить вероятностную функцию потерь $\mathcal{L}(a_\theta, x) = -\ell \ln \phi(x, y, \theta)$. Такое определение потери вполне естественно, чем хуже пара (x_i, y_i) согласуется с моделью ϕ, тем меньше значение плотности $\phi(x_i, y_i, \theta)$ и выше величина потери $\mathcal{L}(a_\theta, x)$.

Верно и обратное – для многих функций потерь возможно подобрать модель плотности $\phi(x, y, \theta)$ таким образом, чтобы минимизация эмпирического риска была эквивалентна максимизации правдоподобия.

Таким образом, существуют два родственных подхода к формализации задачи обучения: первый основан на введении функции потерь, второй – на введении вероятностной модели порождения данных. Оба в итоге приводят к схожим (иногда даже в точности одинаковым) оптимизационным задачам. Обучение – это оптимизация.

1.7 Аппроксимация и регуляризация эмпирического риска

В случае линейной классификации естественный способ определить качество того или иного алгоритма – вычислить для объектов обучающей выборки долю неправильных ответов:

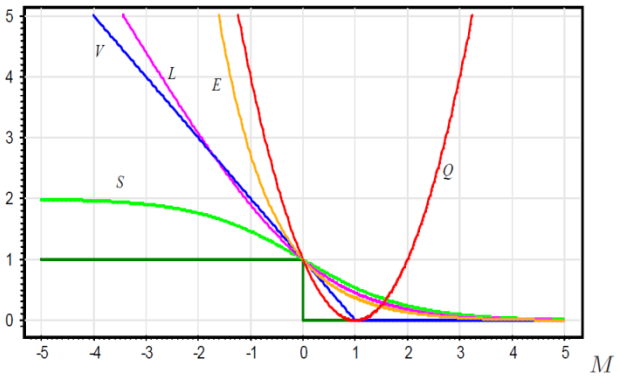
$$Q(a, X^l) = \frac{1}{l}\sum_{i=1}^{l} [a(x_i) \neq y_i]$$

Введем применительно к линейным методам классификации понятие отступа (margin) $M_i(\theta) = y_i\langle x_i, \theta\rangle$ объекта x_i. Если $M_i(\theta) < 0$, то алгоритм $a(x, \theta)$ допускает ошибку на объекте x_i. Чем больше отступ, тем правильнее и надежнее классификация объекта x_i.

С помощью введенного понятия отступа можно переписать выражение для $Q(a, X^l)$ в следующем виде:

$$Q(a, X^l) = \frac{1}{l}\sum_{i=1}^{l} [y_i\langle x_i, \theta\rangle < 0] = \frac{1}{l}\sum_{i=1}^{l} [M_i(\theta) < 0]$$

$M_i(\theta)$ – пороговая функция, с разрывом в точке 0, что делает невозможным применение к ней легко реализуемых методов оптимизации гладких функций (рисунок 1).



$Q(M) = (1-M)^2$ — квадратичная;
$V(M) = (1-M)_+$ — кусочно-линейная;
$S(M) = 2(1+e^M)^{-1}$ — сигмоидная;
$L(M) = \log_2(1+e^{-M})$ — логистическая;
$E(M) = e^{-M}$ — экспоненциальная.

Рисунок 1 – Непрерывные аппроксимации пороговой функции потерь $M_i(\theta)$.

Для применения методов оптимизации используют различные виды непрерывных функций мажорирующих (оценивающих сверху) пороговую функцию: квадратичную, кусочно-линейную, сигмоидную, лоигстическую, экспоненциальную.

1.8 Оптимизация функционала качества

Для гладких оценок пороговой функции потерь минимизация функционала качества $Q(a, X^l)$ может выполняться методом градиентного спуска.

$$Q(a, X^l) = \frac{1}{l}\sum_{i=1}^{l} \mathcal{L}(a, x_i) \rightarrow min,$$

В этом методе выбирается некоторое начальное приближение для вектора весов θ, затем запускается итерационный процесс, на каждом шаге которого вектор θ изменяется в направлении наиболее быстрого убывания функционала $Q(a, X^l)$. Это направление противоположно вектору градиента $Q'(\theta) = \left(\frac{\partial Q(\theta)}{\partial \theta_j}\right)_{j=1}^{n}$:

$$\theta := \theta - \eta Q'(\theta),$$

где $\eta > 0$ величина шага в направлении антиградиента, называемая также *темпом обучения* (learning rate).

Распишем градиент функции потерь $\mathcal{L}$:

$$\theta := \theta - \eta \sum_{i=1}^{l} \mathcal{L}'(y_i \langle x_i, \theta \rangle)\, x_i y_i$$

Каждый объект обучающей выборки вносит аддитивный вклад в изменение вектора θ, но вектор θ изменяется только после перебора всех l объектов.

Инициализация весов на нулевом шаге может производиться различными способами. Стандартная рекомендация взять небольшие случайные значения.

Критерием останова является стабилизация значения $Q(a, X^l)$ и/или вектора весов θ. Практичная реализация должна предусматривать стандартизацию данных, отсев выбросов, регуляризацию (сокращение весов), отбор признаков, и другие эвристики для улучшения сходимости.

1.9 Проблема переобучения и обобщающей способности

Минимизацию эмпирического риска следует применять с известной долей осторожности. Если минимум функционала $Q(a, X^l)$ достигается на алгоритме a, то это ещё не гарантирует, что a будет хорошо приближать целевую зависимость на произвольной контрольной выборке $X^k = (x'_i, y'_i)_{i=1}^{k}$.

Когда качество работы алгоритма на новых объектах, не вошедших в состав обучения, оказывается существенно хуже, чем на обучающей выборке, говорят об эффекте *переобучения* (overtraining) или *переподгонки* (overfitting). При решении практических задач с этим явлением приходится сталкиваться очень часто.

Легко представить себе метод, который минимизирует эмпирический риск до нуля, но при этом абсолютно не способен обучаться. Получив обучающую выборку X^l, он запоминает её и строит алгоритм, который сравнивает предъявляемый объект x с обучающими

объектами x_i из X^l. В случае совпадения $x = x_i$ алгоритм выдаёт правильный ответ y_i. Иначе выдаётся произвольный ответ. Эмпирический риск принимает наименьшее возможное значение, равное нулю. Однако этот алгоритм не способен восстановить зависимость вне материала обучения. Отсюда вывод: для успешного обучения необходимо не только запоминать, но и обобщать.

Обобщающая способность (generalization ability) метода μ характеризуется величиной $Q(\mu(X^l), X^k)$ при условии, что выборки X^l и X^k являются представительными. Для формализации понятия «представительная выборка» обычно принимается стандартное предположение, что выборки X^l и X^k простые, полученные из одного и того же неизвестного вероятностного распределения на множестве X.

Для эмпирической оценки обобщающей способности применяют метод скользящего контроля (cross-validation, CV).

1.10 Метрики качества классификации

Метрики качества используются:

- для задания функционала ошибки при обучении;
- для подбора гиперпараметров при измерении качества на скользящем контроле. В том числе можно использовать другую метрику, которая отличается от метрики, с помощью которой построен функционал ошибки;
- для оценивания итоговой модели: пригодна ли модель для решения задачи.

Для оценки работы бинарного классификатора и подбора гиперпараметров моделей в данной работе используются следующие показатели:

- $TP\ (True\ positive)-$ объекты верно отнесенные к классу $+1$;
- $FP\ (False\ positive)-$ объекты ошибочно отнесенные к класса $+1$;
- $TN\ (True\ negative)-$ объекты верно отнесенные к классу -1;
- $FN\ (False\ negative)-$ объекты ошибочно отнесенные к классу -1.

На основе этих показателей определяются метрики качества классификатора precision (точность) и recall (полнота), FPR (false positive rate):

$$precision = \frac{TP}{TP + FP}, \qquad recall = \frac{TP}{TP + FN}, \qquad FRP = \frac{FP}{FP + FN}$$

Рассмотрим модель алгоритмов классификации $a(x, \theta) = sign(\langle x, \theta \rangle - \theta_0)$, где $\theta_0\ \epsilon\ R -$ аддитивный гиперпараметр, в теории нейронных сетей его называют *порогом активации*. На практике после построения алгоритма значение данного порога может неоднократно пересматриваться изменяя соотношение $precision/recall$. Поэтому применяется специальная метрика качества - $AUC - ROC$ (площадь под $ROC -$ кривой), которая показывает, что происходит с числом ошибок обоих типов (FP, FN), если изменяется порог активации.

Каждая точка на ROC-кривой соответствует некоторому алгоритму, некоторому значению θ_0. В общем случае это даже не обязательно кривая, дискретное множество алгоритмов также может быть отображено в тех же координатах в виде точечного графика.

Для построения ROC-кривой по оси X откладывается доля ошибочных положительных классификаций FPR, а по оси Y – recall. ROC-кривая монотонно не убывает и проходит из точки (0, 0) в точку (1, 1). Чем выше проходит ROC-кривая, тем выше качество классификации. Идеальная ROC-кривая проходит через левый верхний угол - точку (0, 1). Наихудший алгоритм соответствует диагональной прямой, соединяющей точки (0, 0) и (1, 1); её часто изображают на графике как ориентир.

2 Исходные данные

Исходный набор данных (см. таблицу 1) включает в себя 40 тысяч объектов и 230 анонимизированных признаков с пропусками и «зашумленных» :

- 190 вещественных признаков;
- 40 номинальных, с хешированными значениями;
- 18 признаков не определены ни для одного из объектов;
- 114 заполнены менее, чем для 3% объектов;
- 57 вещественных признаков имеют менее 20 уникальных значений;
- 16 номинальных имеют более 1800 уникальных значений.

Классы не сбалансированы, доля класса оттока $+1$ $-$ 7.44 %

Таблица 1 – Dataset

№	Var1	Var2	Var3	…	Var189	Var190	Var191	…	Var229	Var230	Y
0	NaN	NaN	NaN	…	NaN	NaN	NaN	…	NaN	NaN	-1
1	NaN	NaN	NaN	…	276.0	NaN	NaN	…	mj86	NaN	-1
2	NaN	NaN	NaN	…	NaN	NaN	NaN	…	mj86	NaN	-1
3	NaN	NaN	NaN	…	NaN	NaN	NaN	…	NaN	NaN	1
4	NaN	NaN	NaN	…	NaN	NaN	NaN	…	NaN	NaN	-1
5	0.0	NaN	NaN	…	174.0	NaN	NaN	…	NaN	NaN	-1

3 Постановка задачи

Цель работы: разработать алгоритм предсказания ухода абонентов по представленной выборке и оценить экономический эффект от его применения.

Задачи:

- Предобработка исходных данных;
- Корреляционный и статистический анализ данных;
- Построение простых baseline-моделей, оценка потенциального качества решения;
- Отбор значимых признаков;
- Проведение экспериментов по сравнению моделей и настройке их параметров;
- Выбор модели с наилучшим результатом;
- Экономический анализ эффекта от внедрения модели.

4 Предобработка данных

Перед проведением корреляционного и статистического анализа из выборки были удалены выбросы:

- значения вещественных признаков лежащие за пределами 99-го процентиля были заменены на их математическое ожидание;
- пропущенные значения вещественных признаков были заменены их математическое ожидание.
- значения категориальных признаков частота встречаемости которых была менее 0.5% от числа объектов с указанными значениями этого признака были заменены на «Rare»;
- пропущенные значения категориальных признаков были заменены на «Missing».

При построении baseline-моделей варьировались пороги замены как для вещественных (от 95 до 99 процентилей), так и для категориальных признаков (от 0.5% до 2.5%).

Рассматривались следующие методы заполнения пропущенных значений вещественных признаков:

- значениями математического ожидания соответствующего признака;
- нулями;
- значениями, предсказанными линейной регрессией (для значимых вещественных признаков количество пропущенных значений в которых было менее 20% , а количество уникальных значений более 300).

Для построения финальных моделей из данных были удалены категориальные переменные имеющие более 100 уникальных значений, все вещественные переменные имеющие менее 80 уникальных значений обрабатывались, так же как и категориальные – с помощью Dummy-кодирования.

Dummy-кодированием или one-hot-кодированием [8] называют метод позволяющий некоторый категориальный признак f принимающий q значений $\{a_1 \dots a_q\}$ заменить на q бинарных признаков следущим образом:

$$f^k = I\,[\,f = a_k],\ k \in \{1 \dots q\},$$

где $I\,[E]$ – индикатор события E, т.е.

$$I[E] = \begin{cases} 1, & \text{если } E \text{ исттинно,} \\ 0, & \text{если } E \text{ ложно.} \end{cases}$$

К полученным таким образом описаниям объектов можно применять многие классические методы работы с вещественными признаками.

Одним из недостатков dummy-кодирования является сильно увеличивающаяся размерность пространства объектов. Многие алгоритмы не способны обрабатывать полученные матрицы данных во многих реальных задачах. В связи с этим описание объектов приходится хранить в разреженном формате и использовать приспособленные методы. Например, логистическую регрессию, многие реализации которой позволяют работать с разреженным представлением данных.

5 Корреляционный и статистический анализ.

5.1 Вещественные признаки

Для каждой вещественной переменной количество уникальных значений которой больше 80 с помощью статичтическго критерия Манна-Уитни проверялась нулевая гипотеза H_0: среднее значение переменной в двух группах отток и не отток не отличаются, против двухсторонней альтернативы H_1: среднее значение переменной в двух группах отличается.

Таблица 2 – Критерий Манна-Уитни

выборки:	$X_1^{n_1} = (X_{11}, \ldots, X_{1n_1})$, $X_2^{n_2} = (X_{21}, \ldots, X_{2n_2})$,
нулевая гипотеза:	$H_0 \colon F_{X_1}(x) = F_{X_2}(x)$;
альтернатива:	$H_1 \colon F_{X_1}(x) = F_{X_2}(x + \Delta), \Delta <\neq> 0$;
статистика:	$X_{(1)} \leq \ldots \leq X_{(n_1+n_2)}$ — вариационный ряд объединённой выборки $X = X_1^{n_1} \bigcup X_2^{n_2}$, $R_1(X_1^{n_1}, X_2^{n_2}) = \sum_{i=1}^{n_1} \operatorname{rank}(X_{1i})$;
нулевое распределение:	табличное.

Данные содержат 81 подобную вещественную переменную, следовательно имеется 81 выборка, каждая своего размера и из своего распределения. Каждой выборке соответствует своя нулевая гипотеза H_{0_i} и альтернатива H_{1_i}. Каждая из гипотез проверяется своей статистикой. Для каждой из статистик свое нулевое распределение. Иными словами перед нами стоит задача множественной проверки гипотез.

Таблица 3 – Множественная проверка гипотез

данные:	$\mathbf{X} = \{X_1^{n_1}, \ldots, X_m^{n_m}\}, \; X_i \sim \mathbf{P}_i$;
нулевые гипотезы:	$H_i \colon \mathbf{P}_i \in \omega_i$;
альтернативы:	$H_i' \colon \mathbf{P}_i \notin \omega_i$;
статистики:	$T_i = T(X_i^{n_i})$;

Для решения поставленной задачи используется метод Холма. Метод Холма — это нисходящая процедура множественной проверки гипотез со следующими уровнями значимости:

$$\alpha_1 = \frac{\alpha}{m}, \; \alpha_1 = \frac{\alpha}{m-1}, \ldots, \alpha_1 = \frac{\alpha}{m-i+1}, \alpha_m = \alpha \, ,$$

где m количество проверяемых гипотез, α - порог для достигаемого уровня значимости.

Если достигаемый уровень значимости меньше, чем α, то нулевая гипотеза отвергается в пользу альтернативы.

С достигаемым уровнем значимости $\alpha = 0.05$ нулевая гипотеза о равенстве средних значений переменной в группах отток и не отток отвергается для следующих 24 вещественных переменных: Var113, Var112, Var119, Var28, Var25, Var21, Var22, Var6, Var3, Var160, Var183, Var106, Var125, Var74, Var95, Var140, Var84, Var13, Var162, Var73, Var85, Var81, Var177, Var109.

5.2 Категориальные признаки

3.2.1 Для категориальных переменных с единственным значением была посчитана доля оттока среди всех объектов для которых они были определены:

Признак	Доля оттока
Var191	0.049369
Var213	0.046067
Var215	0.062167
Var224	0.055891

3.2.2 Для бинарных категориальных переменных и вещественных переменных с двумя уникальными значениями был посчитан коэффициент корреляции Мэтьюса с целевой переменной y.

Коэффициент корреляции Мэтьюса – мера силы взаимосвязи между двумя бинарными переменными. Для того чтобы его вычислить, необходимо использовать таблицу сопряженности:

X_1	X_2	
	0	1
0	a	b
1	c	d

В стоках таблицы сопряженности находятся значения одного признака, по столбцам – второго, в каждой ячейке – количество объектов, на которых реализовалась эта пара. Коэффициент корреляции Мэтьюса вычисляется по данных из таблицы сопряженности следующим образом:

$$MCC_{X_1X_2} = \frac{ad - bc}{\sqrt{(a+b)(a+c)(b+d)(c+d)}},$$

причем для корректного использования критерия должны выполняться условия:

$$n = a + b + c + d > 40$$

$$\frac{(a+c)(a+b)}{n}, \frac{(a+c)(c+b)}{n}, \frac{(b+d)(a+b)}{n}, \frac{(b+d)(c+b)}{n} > 5$$

Корреляция Мэтьюса лежит в диапазоне от -1 до 1. $MCC_{X_1X_2} = 0$ соответсвует случаю полного отсутсвия взаимосвязи между переменными. $MCC_{X_1X_2} = 1$ соответсвуют ситуации $b = c = 0$ т.е. в выборке отсутствуют объекты на которых значения X_1 и X_2 отличаются. $MCC_{X_1X_2} = -1$ – противоположенная ситуация: в выборке нет ни одного объекта, на которых значения двух бинарных признаков совпадают.

Значения корреляции Мэтьюса признаков для которых выполняются условия корректного использования:

Признак	MCC_{XY}
Var130	0.0437
Var201	0.0097
Var208	0.0095
Var211	-0.032
Var218	0.0448

3.2.3 Категориальные переменные с количеством уникальных значений больше 2

Для прочих категориальных переменных и вещественных переменных с менее чем 80 уникальными значениями был посчитан коэффициент корреляции V Крамера с целевой переменной y. Корреляция V Крамера это обобщение корреляции Мэтьюса.

Пусть X_1 принимает K_1 различных значений, а X_2 - K_2 различных значений. Можно составить таблицу сопряженности, у которой в i строке и стобце k будет стоять n_{ij} - количество объектов выборки, на которых $X_1 = i$, а $X_2 = j$.

X_1	X_2				
	1	…	j	…	K_2
1					
…					
i			n_{ij}		
…					
K_1					

На основании этой таблицы сопряженности вычисляется мера взаимосвязи между X_1 и X_2. Эта мера называется коэффициентом V Крамера :

$$\phi(X_1^n, X_2^n) = \sqrt{\frac{\chi^2(X_1^n, X_2^n)}{n(\min(K_1 K_2) - 1)}},$$

где $\chi^2(X_1^n, X_2^n)$ - значение статистики хи-квадрат.

Критерий хи-квадрат для таблиц сопряженности может применяться при выполнении следующих условий:

- объем выборки n > 40;
- ожидаемое количество элементов в каждой ячейке таблицы меньше 5, не более, чем в 20% ячеек.

Значения коэффициента корреляции V Крамера признаков для которых выполняются условия корректного использования критерия хи-квадрат:

№	Признак	Корреляция	Уникальных значений
0	Var126	0.175901	51
1	Var7	0.084238	6
2	Var206	0.083779	21
3	Var212	0.078319	78
4	Var205	0.072558	3
5	Var228	0.070700	30
6	Var144	0.062611	10
7	Var229	0.062432	4
8	Var193	0.062371	50
9	Var65	0.057621	13
10	Var207	0.057221	14
11	Var225	0.056515	3
12	Var227	0.055619	7
13	Var226	0.048516	23
14	Var221	0.047711	7
15	Var210	0.040914	6
16	Var78	0.037950	13
17	Var35	0.037610	12
18	Var132	0.037497	18
19	Var44	0.037079	8
20	Var173	0.036529	4
21	Var143	0.036347	4
22	Var181	0.035721	7
23	Var72	0.034150	8

3.2.4 Визуализация данных.

Визуализцая вещественных переменных (рисунок 2) позволила определить линейную зависимость переменных Var21 и Var22: Var22 = 1.25·Var21, и исключить одну из них при построении модели. Зависимость переменных Var21 и Var160 – параболическая.

5.3 Выводы

Проведенный анализ позволил выделить 24 вещественных признака для которых средние значения признака для классов отток и не отток статистически значимо отличается.

Расчет доли класса оттока в переменных с единственным значением позволил выделить две переменные для которых рассчитанные доли отличаются от средней по выборке более чем на 60%.

Расчет коэффициентов корреляции Мэтьюса и V-Крамера позволил отранжировать категориальные признаки по «значимости».

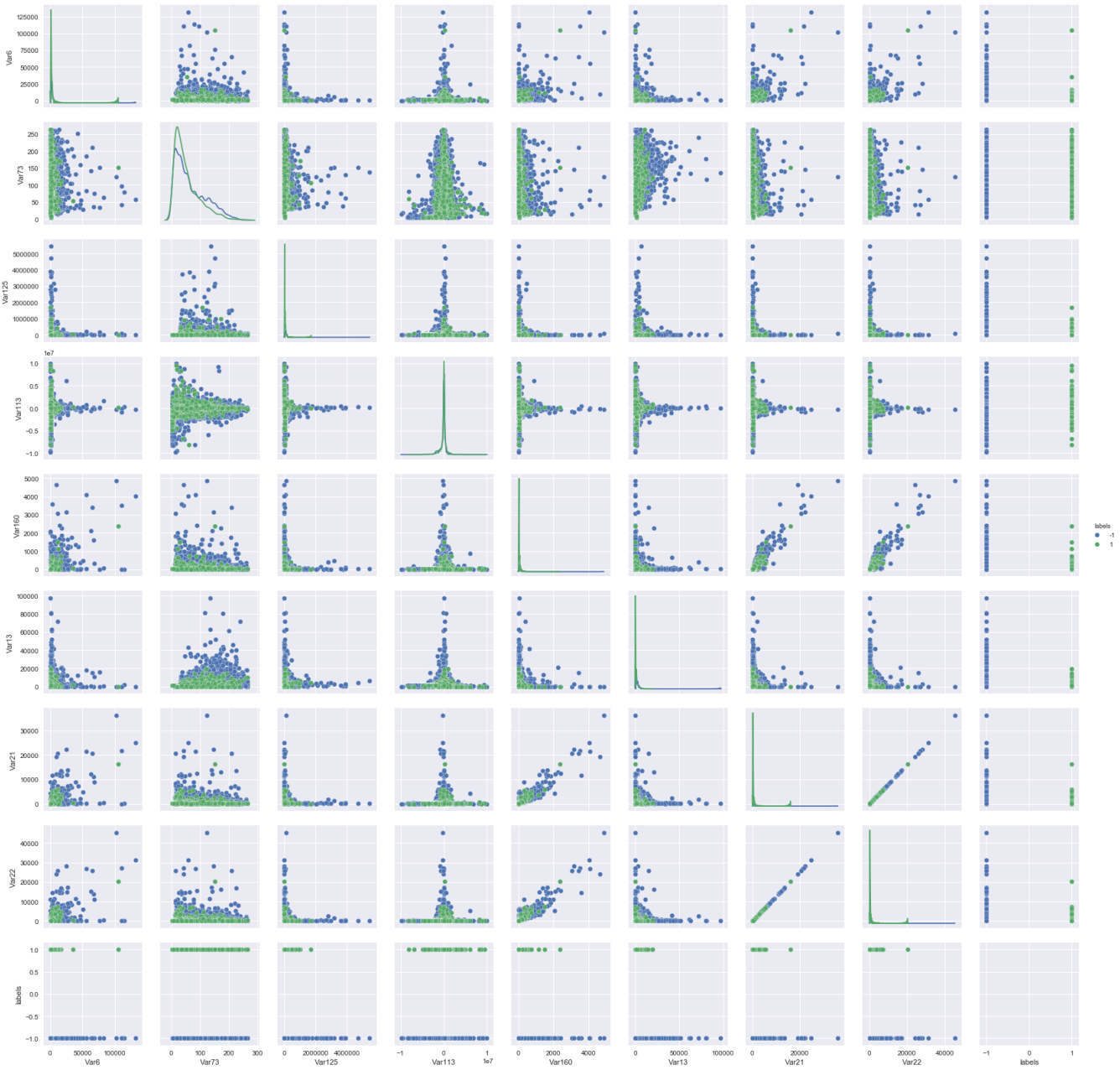


Рисунок 2 – Визуализация вещественных переменных

6 Построение простых baseline-моделей

Для построения baseline-моделей были выбраны вещественные по типу признаки [Var1: Var190]. Проведена процедура нормализации $f_i = \frac{f_i - \mathbb{E}(f)}{\sigma_f}$.

В качестве алгоритмов были выбраны логистическая регрессия, случайный лес.

Оценка AUC, Precision и Recall по классу оттока выполнялась с помощью кросс-валидации на 10 фолдов.

Алгоритм	**AUC**	**Precision**	**Recall**
Заполнение пропусков средними, без замены экстремальных значений			
Логистическая регрессия с балансировкой классов	0.6541±0.015	0.11	0.67
Случайный лес	0.6918±0.019	0.14	0.49
Заполнение пропусков средними, замена 1% экстремальных значений			
Логистическая регрессия с балансировкой весов	0.6517±0.017	0.11	0.67
Случайный лес	0.6936±0.02	0.14	0.49
Заполнение пропусков средними, замена 2.5% экстремальных значений			
Логистическая регрессия с балансировкой весов	0.6430±0.011	0.11	0.66
Случайный лес	0.6896±0.013	0.14	0.48
Заполнение пропусков нулями, без замены экстремальных значений			
Логистическая регрессия с балансировкой весов	0.6325±0.017	0.10	0.66
Случайный лес	0.6937±0.019	0.14	0.51
Заполнение пропусков нулями, замена 1% экстремальных значений			
Логистическая регрессия с балансировкой весов	0.6324±0.013	0.10	0.67
Случайный лес	0.6923±0.016	0.14	0.53
Заполнение пропусков нулями, замена 2.5% экстремальных значений			
Логистическая регрессия с балансировкой весов	0.6231±0.016	0.10	0.66
Случайный лес	0.6881±0.015	0.14	0.52

С вероятностью ~68%, можно утверждать, что замена более 1% экстремальных значений приводит к снижению качества работы алгоритма, а заполнение пропусков нулями снижает качество логистической регрессии.

Для двух моделей построены кривые обучения – зависимости качества обучения от размера обучающей выборки и кривые Precision-Recall. На рисунке 3 представлены кривые для случайного леса в случае заполнения пропусков средними и замены 1%

экстремальных значений. На рисунке 4 кривые для логистической регрессии с балансировкой классов.

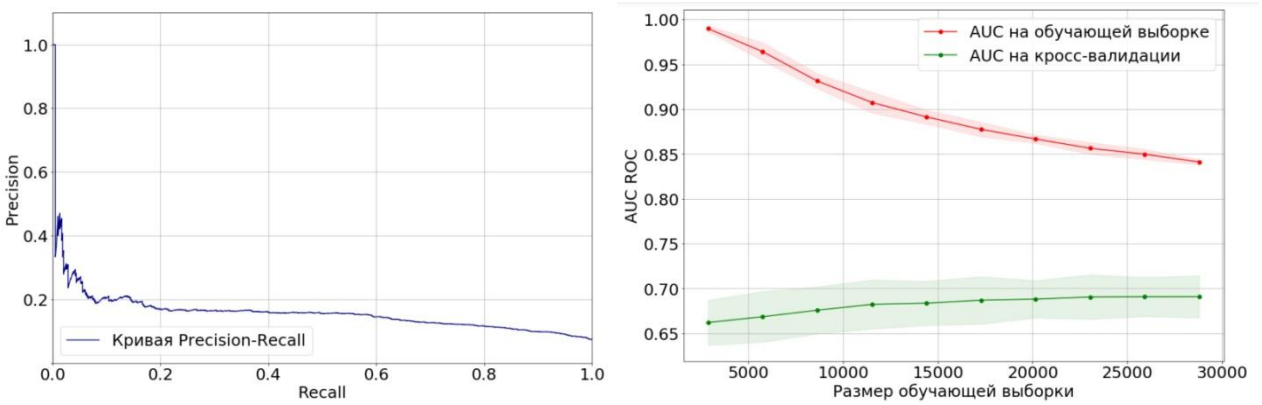


Рисунок 3 - Кривая Precision-Recall и кривая обучения для случайного леса

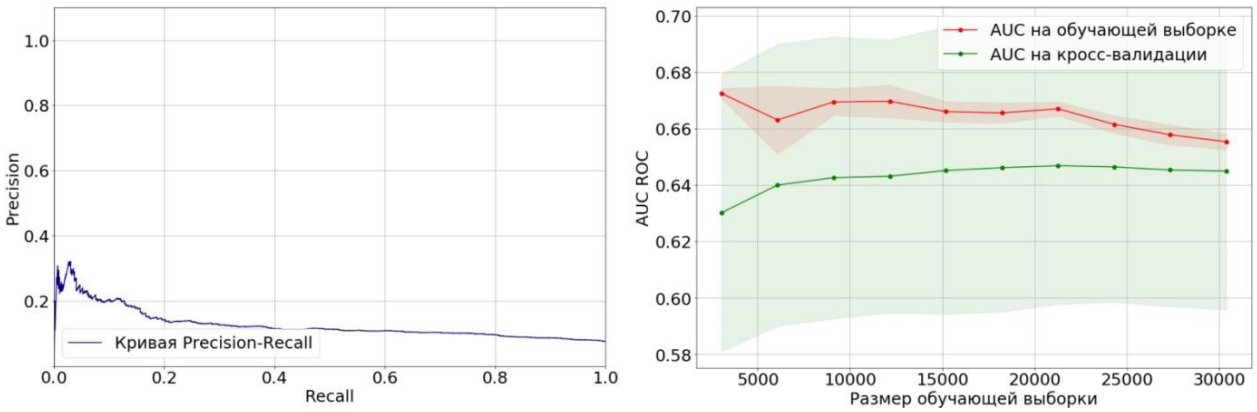


Рисунок 4 - Кривая Precison-Recall и кривая обучения для логистической регрессии

7 Сравнение моделей и настройка параметров

7.1 Балансировка классов для логичстической регрессии

Минимизируемый функционал ошибки с логистической функцией потерь для задачи бинарной классификации позволяет изменять «стоимость» ошибок на объектах разных классов с помощью введения дополнительного гиперпараметра w:

$$Q(a, X^l) = \frac{1}{l}\sum_{i=1}^{l} w \cdot (1 + y_i)\ln\left(1 + e^{-y_i \cdot \langle \theta, x_i \rangle}\right) + (1 - y_i)\ln\left(1 + e^{-y_i \cdot \langle \theta, x_i \rangle}\right)$$

При $w > 1$ ошибка классификации на объекте класса $+1$ будет вность больший вклад в аддитивный функционал ошибки, чем ошибка классификации на объекте -1.

В таблице и на рисунке 5 поведены значения метрик качества классификации логистической регрессии для различных значений w.

Вес оттока	AUC	Precision	Recall
1	0.6834±0.011	0.33	0.01
5	0.7198±0.010	0.28	0.19
10	0.7232±0.010	0.16	0.54
15	0.7245±0.010	0.12	0.76
20	0.7254±0.010	0.10	0.85
25	0.7259±0.010	0.10	0.90

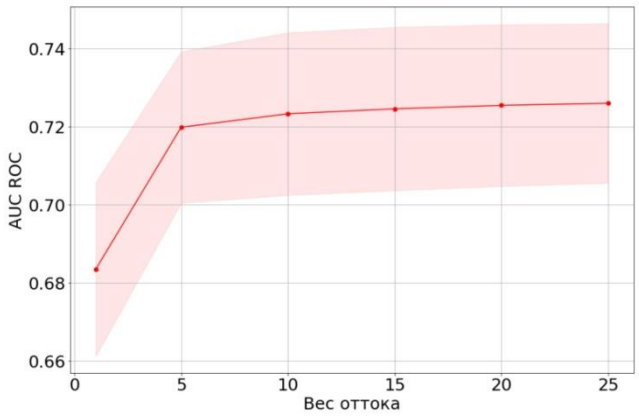


Рисунок 5 – $AUC\ ROC\ (w)$

7.2 Undersampling для грдиентного бустинга над решающими деревьями

Для балансировки классов при построении алгоритмов с решающими деревьями используются методы oversampling и undersampling. Oversampling – увеличение доли минорного класса за счет создания дубликатов объектов минорного класса. Undersampling – увеличение доли минорного класса за счет удаления объектов доминирующего класса.

Undersampling сокращает размер обучающей выборки, но согласно кривой обучения (рисунок 6) это сокращение объема незначительно сказывается на качестве композиции решающих деревьев, чего нельзя сказать о качестве логичитческой регрессии. Уменьшение объема выборки ведет к сокращению времени на построение композиции, поэтому для дальнейшего использования была применена именно техника undersampling'а

Измерение метрик качества проводилось не на кросс-валидации, что было бы не корректно вследствии изменения баланса классов в обучающей выборке, а на отложенной (hold-out) выборке размером 20% от X^l отделенной от обучающей до проведения процедуры undersampling'а.

Как видно из расчетов изменение баланса классов не сказываясь на значении метрики AUC ведет к перераспределению полноты и точности.

Доля класса оттока	AUC	Precision	Recall
0.1	0.7435	0.46	0.05
0.2	0.7369	0.35	0.14
0.3	0.7428	0.26	0.28
0.4	0.7388	0.20	0.47
0.5	0.7409	0.14	0.71

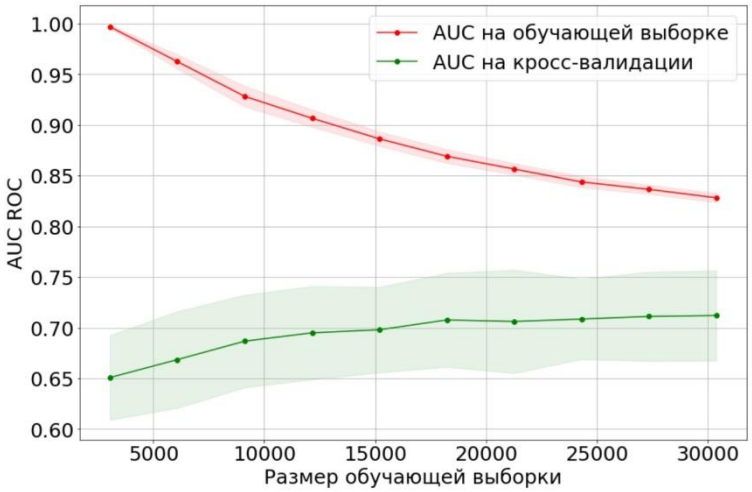


Рисунок 6 - Кривая обучения для градиентного бустинга над решающими деревьями

7.3 Отбор переменных

Для отбора переменных для логистической регрессии использовался алгоритм линейной регрессии с регуляризатором LASSO. При данном типе регуляризации дополнительно к оптимизируемому функционалу качества вводится ограничительное неравенство предотвращающее слишком большие абсолютные значения коэффициентов:

$$\frac{1}{2l}\sum_{i=1}^{l}|y - \langle\theta, x_i\rangle|^2 + \alpha \cdot ||\theta||$$

где $\alpha -$ параметр регуляризации. Чем больше α тем больше параметров θ_j обнуляется. Происходит селекция признаков. Образно говоря параметр $alpha$ зажимает вектор коэффициентов, лишая его избыточных степеней свободы. Важность признака для алгоритма можно оценить по модулю значения θ_i перед ним.

После one-hot кодирования обучающая выборка содержала 650 признаков, изменение весов перед ними с ростом $alpha$ продемонстрировано на рисунке 7. Для получения важности исходного категориального признака модули весов перед всеми признаками производными от данного категориального складывались. Top 10 признаков для линейной регрессии с регуляризатором LASSO приведен в таблице (см. ниже) .

№	Признак	Вес
1	Var126	1.321241
2	Var189	0.827889
3	Var226	0.531673
4	Var210	0.426528
5	Var218	0.391977
6	Var206	0.119127
7	Var205	0.084792
8	Var73	0.083380
9	Var24	0.082144
10	Var212	0.067319

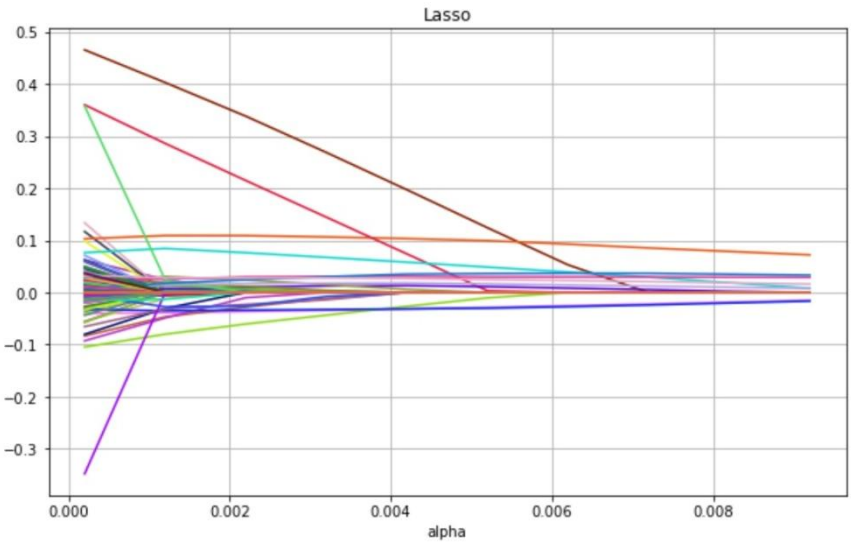


Рисунок 7 – Динамика изменения весов с ростом alpha

Признаки важные для градиентного бустинга над решающими деревьями приведены на рисунке 8.

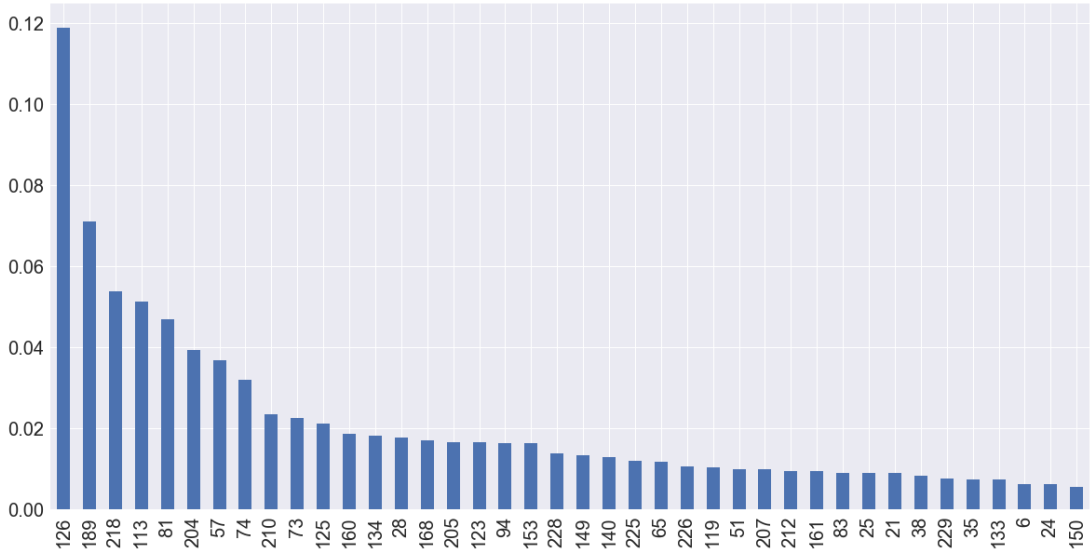


Рисунок 8 - Переменные важные для градиентного бустинга над деревьями

Для обоих алгоритмов наиболее важным оказался категориальный признак имеющий наибольшее значение корреляции V-Крамера - Var126.

В Топ 10 наиболее важных признаков 5 признаков общие для обоих алгоритмов.

Топ 10 наиболее важных признаков для логистической регрессии имеет 4 пересечения с Топ 10 по значению корреляции V-Крамера.

7.4 Голосование моделей

Существует несколько наиболее известных корректирующих операций:

- простое голосование (Simple Voiting)

$$a(x) = F\big(a_1(x), \dots, a_T(x)\big) = \frac{1}{T}\sum_{t=1}^{T} a_t(x)$$

- взвешенное голосование (Weighted Voiting)

$$a(x) = F\big(a_1(x), \dots, a_T(x)\big) = \frac{1}{T}\sum_{t=1}^{T} w_t a_t(x)$$

$$\sum_{t=1}^{T} w_t = 1, w_t \geq 0$$

- “мягкое” голосование (Soft Voiting)

$$a(x) = F\big(a_1(x), \dots, a_T(x)\big) = \max\left(a_1(x), \dots, a_T(x)\right)$$

- смесь экспертов (Mixture of Experts):

$$(x) = F\big(a_1(x), \dots, a_T(x)\big) = \frac{1}{T}\sum_{t=1}^{T} w_t\,(x)\; a_t(x)$$

$$\sum_{t=1}^{T} w_t(x) = 1, w_t \geq 0$$

Для реализации в проекте была принята модель «мягкого» голосования, принимающая в качестве итогового значения предсказание того из голосующих алгоритмов, который наиболее уверен в классификации объекта. В качестве голосующих алгоритмов были выбраны три алгоритма логистической регрессии с разными весовыми коэффициентами и алгоритм градиентного бустинга, обученный на undersampling выборке с долей класса оттока 0.3. Полученное на кросс-валидации значение AUC составило 0.7322±0.008.

7.5 Стекинг

Идея стекинга состоит в том, чтобы обучить метаклассификатор M не только на исходной выборке X^l, но и на предсказаниях (метапризнаках) полученных с помощью базовых классификаторов a. Метапризнаки полученные с помощью базовых классификаторов a для выборки X^l, будем обозначать $MF(X^l, a)$

Введем следующие обозначения:

a – базовый алгоритм;

$a.fitting(X^l, Y)$ – функция обучения классификатора;

$a.predict(X^l)$ – функция предсказывающая целевую переменную для X^l классификатором a;

M – метаклассификатор;

$MF(X^l, a)$ – метапризнак, полученный классификатором a для выборки X^l;

P – финальное предстказание стекинга;

$concatV(X_i, X_j)$ - операция конкатенирования по стобцам;

$concatH(X_i, X_j)$ - операция конкатенирования по строкам;

Алгоритм

Зафиксировать разбиения $\left\{ \left\{ \left(X_{nk}^{\frac{l}{N}}, Y_{nk}^{\frac{l}{N}} \right), k = 1 \dots K \right\}, \; n = 1 \dots N \right\}$

для $n = 1 \dots N$

  для $k \; = \; 1 \dots \; K$

    $a_{nk} := a$

    $a_{nk}.fitting(\; X^l / X_{nk}^{\frac{l}{N}}, \; Y^l / Y_{nk}^{\frac{l}{N}} \;)$

    $MF\left(X_{nk}^{\frac{l}{N}}, a_{nk} \right) := a_{nk}.predict(X_{nk}^{\frac{l}{N}})$

    $MF(\; Z, a_{nk}) := a_{nk}.predict(Z)$

  $MF\left(X_n^{\frac{l}{N}}, a_n \right) := concatH\left(\left\{ MF\left(X_{nk}^{\frac{l}{N}}, a_{nk} \right), k = 1 \dots K \right\} \right)$

  $MF(\; Z, a_n) := \frac{1}{K} \sum_{k=1}^{K} MF(\; Z, a_{nk})$

$MF(\; X^l, a) := \frac{1}{N} \sum_{k=1}^{N} MF\left(X_n^{\frac{l}{N}}, a_n \right)$

$MF(\; Z, a) := \frac{1}{N} \sum_{k=1}^{N} MF(\; Z, a_n)$

$M.fitting(\; concatV(X, MF(X, a), \; Y \;)$

$P := \; M.predict(concatV(Z, MF(Z, a), \; Y \;)$

В качестве базового алгоритма была выбрана логичтическая регрессия:

$$LogisticRegression(C = 0.05, class\_weight = \{1: 5, -1: 1\})$$

В качестве метаалгоритма:

$$GradientBoostingClassifier(\max\_depth = 4, n\_estimators = 800,$$

$$learning\_rate = 0.01, subsample = 0.66)$$

Точечные оценки AUC метаалгоритма приведены в таблице:

K	N	AUC
2	100	0.7311
3	50	0.734
5	25	0.7297
7	16	0.721
10	10	0.717

7.6 Построение финальной модели

Для построения финальной модели из выборки были удалены 10 категориальных переменных имеющих худшую корреляцию V Крамера и 25 вещественных переменных наименее важных для градиентного бустинга. Для обучения использовалась undersumpling подвыборка с долей минорнго класса 0.3. В качестве алгоритма использован $GradientBoostingClassifier$ с параметрами $\max\_depth = 6$, $n\_estimators = 800$, $learning\_rate = 0.01, subsample = 0.66$. AUC модели на кросс-валидации составил 0.7426.

8 Экономический анализ

Исходными данными для экономического анализа являлись кривая Precision-Recall финальной модели, плотность распределения величины среднемесячного чека абонента, график зависимости вероятности принятия предложения оператора от размера бонуса предоставляемого абоненту (в долях от среднемесячного чека).

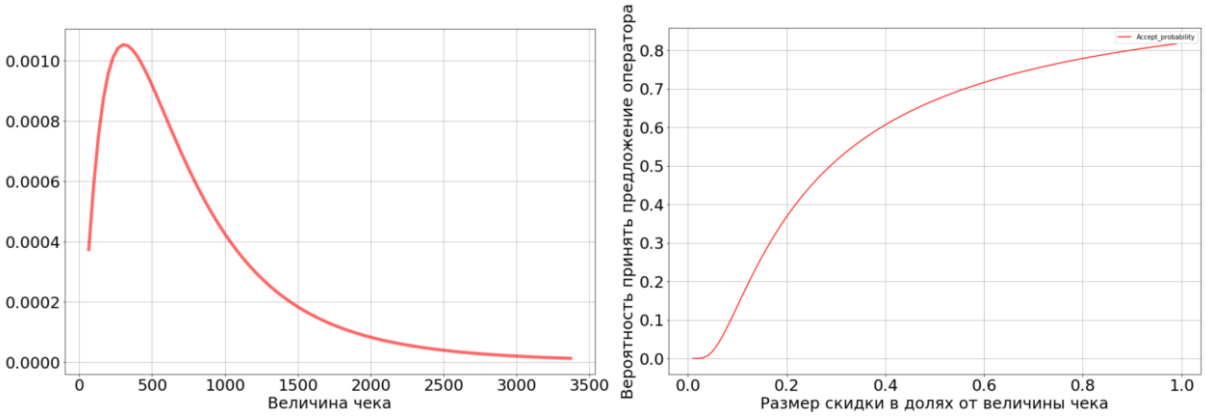


Рисунок 9 - Плотность распределения величины среднемесячного чека абонента и график зависимости вероятности принятия предложения оператора

Согласно выбранному распределению средний размер среднемесячного чека равен 798, средний чек Top 5% пользователей равен 2108.

Предлагаемая зависимость вероянотности принятия предложения от размера чека отражает поведенческие особенности абонента. Бонус в 5% от среднемесячного чека не останавливает от ухода ни одного из собравшихся сменить оператора. 10% бонус останавливает около 10%.

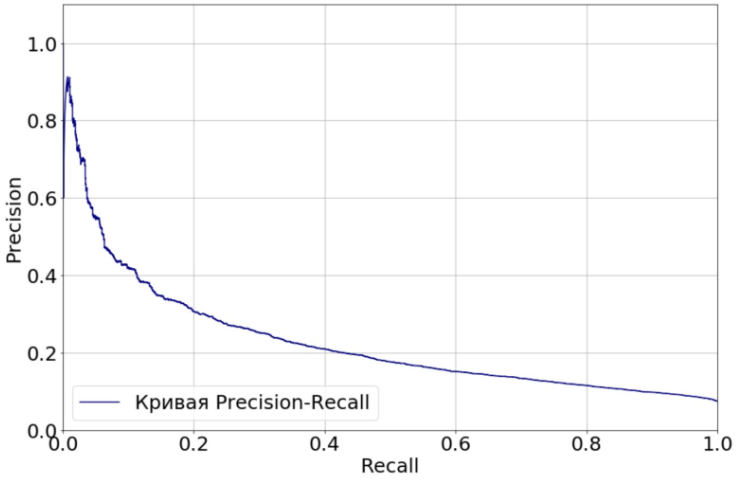


Рисунок 10 - Кривая Precision-Recall финальной модели

Доход от проводимой компании оценивался по формуле:

$$Revenue = live\_time \cdot TP \cdot accept\_probability(cost\_of\_retention) \cdot top \cdot avarage\_check(top) - cost\_of\_retention \cdot (TP \cdot accept\_probability(cost\_of\_retention) + FP) \cdot top - \alpha \cdot ((TP + FP) \cdot Top)^{1.4}$$

TP – количество пользователей верно отнесенных алгоритмом к классу отток;

FP – количество пользователей ошибочно отнесенных алгоритмом к классу отток;

top – доля наиболее платёжеспособных пользователей принимающих участие в компании по удержанию, иными словами доля пользователей из всех отнесенных алгоритмом к классу отток которым будут предложены скидки;

$avarage\_check\ (Top)$ – среднемесячный чек пользователя принимающего участе в компании по удержанию;

$cost\_of\_retention$ – размер предлагаемого бонуса;

$accept\_probability\ (cost\_of\_retention)$ – вероятность принятия предложения от оператора и продолжения договорных отношений с ним;

$live\_time$ – среднее время в месяцах на которое удерживается пользователь за счет предлагаемого бонуса;

$\alpha \cdot ((TP + FP) \cdot Top)^{1.4}$ – условное слагамое накладных расходов на проведение компании по удрежанию.

Произведена оценка дохода для Precision = 0.42, Recall = 0.1, $live\_time$ = 2.

На рисунке 11 приведена зависимость дохода от величины топа пользователей при зафиксированном размере предлагаемого бонуса - 300.

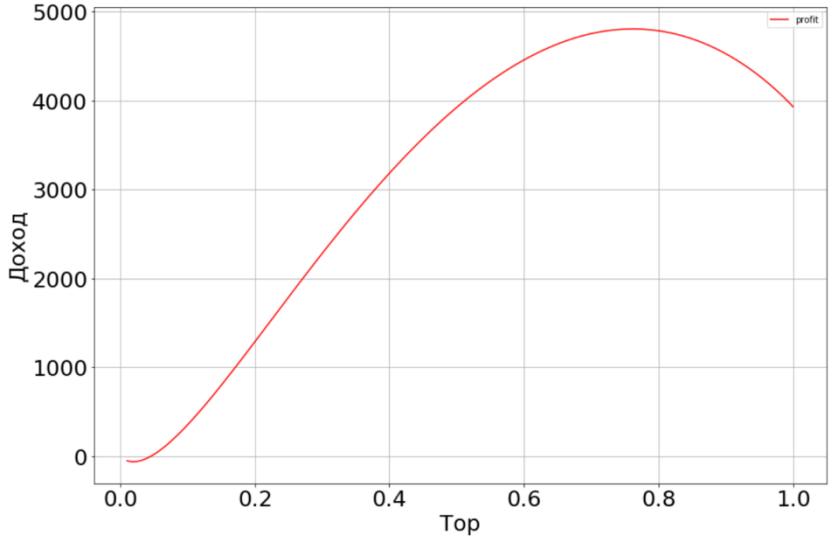


Рисунок 11 - Кривая зависимости дохода от топа пользователей

На рисунке 12 приведена зависимость дохода от предлагаемого бонуса при зафиксированном размере топа - 0.6.

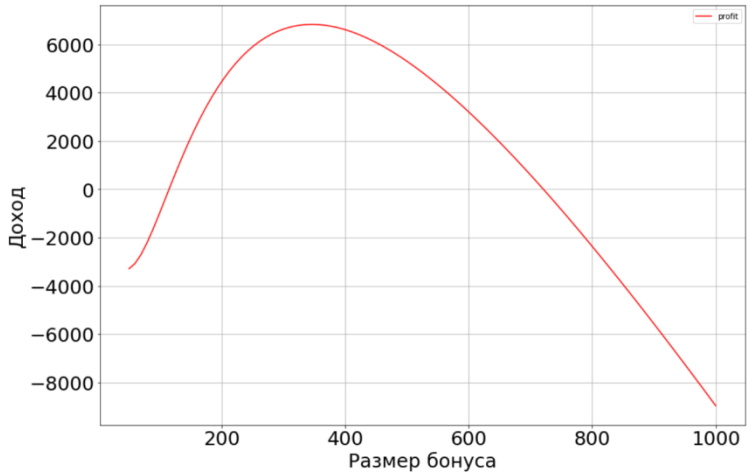


Рисунок 12 - Кривая зависимости дохода от размера бонуса

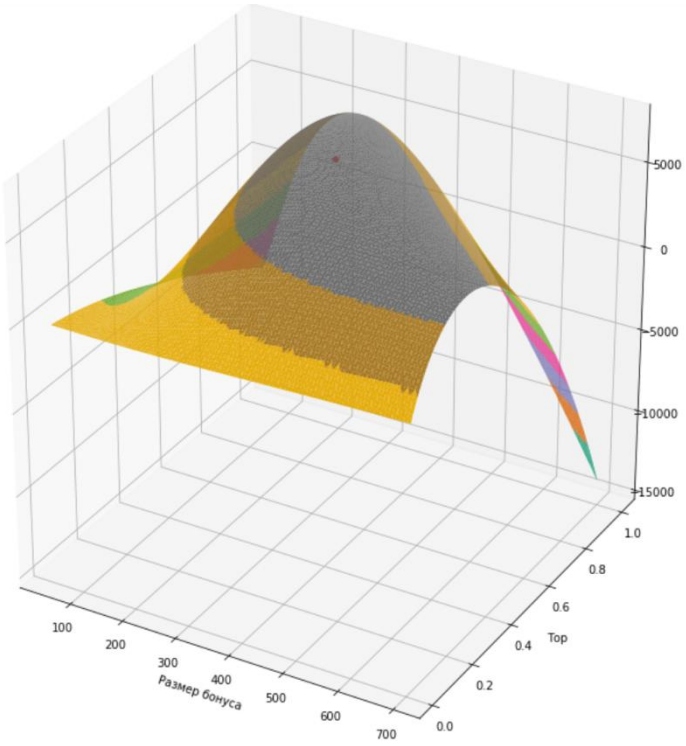


Рисунок 13 - Зависимость дохода от размера бонуса и Top'а

Доходность модели составила 6854 ед. или около 8 среднемесячных чеков. 8 чеков на 156 человек участвовавших в компании по удержанию. Оптимальный размер предлагаемого бонуса ($cost\_of\_retention$) составил 368 ед., оптимальный размер Top'а 56%.

Рост точности (precision) модели на 3% увеличивает её доходность на 19%, рост полноты (Recall) на 3% увеличивает доходность модели на 17%.

Из графиков и трехмерной поверхности видно узкую область параметров применимости модели.

Заключение

Результатом работы является экономически эффективный алгоритм бинарной классификации решающий задачу предсказания оттока клиентов, построенный на базе конкретного эмипирческого материала – 40 тысяч абонентов French Telecom company Orange.

В ходе работы выборка, объекты которой описывались 230 анонимизированными признаками (190 вещественных и 40 хешированных категориальных) , была очищена от «выбросов». Проведен её корреляционный и статистический анализ. Посчитаны корреляции Мэтьюса и V-Крамера для категориальных признаков. Для каждого из вещественных признаков проверена гипотеза о равенстве средних значений в группах «отток» и «не отток».

На базе предобработанных вещественных признаков были построены baseline модели позволившие оценить потенциально возможное качество классификации (логистическая регрессия, градиентный бустинг над решающими деревьями, «случайный лес»).

Над категориальными признаками проведена процедура dummy-кодирования.

Проведены эксперименты с семействами алгоритмов логистической регрессии и градиентного бустинга:

- исследовано влияние различных методов заполнения пропусков в данных;
- опробованы различные способы устранения влияния дисбаланса классов на качество классификации;
- произведен отбор значимых переменных, устранены шумовые переменные;
- произведен поиск оптимальных свободных параметров моделей.

Рассмотрены способы ассемблирования алгоритмов:

- голосование моделей;
- стекиг алгоритмов.

Построена финальная модель и оценен её экономический эффект.

Список литературы

1. Язык программирования python. http://python.org.
2. Библиотека pandas. https://pandas.pydata.org
3. Библиотека SciPy для инженерных и научных расчётов. https://www.scipy.org/
4. Библиотека sklearn для машинного обучения на python. http://scikit-learn.org.
5. Библиотека Seaborn для статистической графики. http://seaborn.pydata.org/
6. К.В. Воронцов. Математические методы обучения по прецедентам (Теория обучения машин). http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf
7. К.В. Воронцов. Вероятностное тематическое моделирование. http://www.machinelearning.ru/wiki/images/2/22/Voron-2013-ptm.pdf.
8. А.Г. Дьяконов. Методы решения задач классификации с категориальными признаками. http://alexanderdyakonov.narod.ru/sw-factors-dyakonov.pdf.