

Викторія В. Жебка<sup>1</sup>, Володимир І. Виноградов<sup>2</sup>,  
 Андрей П. Бондарчук<sup>3</sup>, Михайл Н. Степанов<sup>4</sup>

## ОПТИМИЗАЦИЯ РАБОТЫ АЛГОРИТМА ГРАДИЕНТНОГО БУСТИНГА С ПОМОЩЬЮ ПЕРЕКРЕСТНОЙ ПРОВЕРКИ

*В статье рассмотрены предварительные исследования перекрестной проверки. Показано как она последовательно оценивает ошибку прогнозирования путем компромисса к замещению и дисперсии и показано ее применение в качестве критериев для останковки алгоритма градиентного бустинга. Используя смоделированные данные представлено сходимость и согласованность алгоритма градиентного бустинга. Проведено сравнение результатов для разных правил останковки. Рассмотрена модель нелинейной логистической регрессии, где метод регуляризации не имеет решения. В результате проведения расчётов получено, что уровень погрешностей очень быстро уменьшается на протяжении первых нескольких итераций, потом скорость спада становится приблизительно линейной. Когда уровень погрешностей достигает наиболее низкого уровня, дальше он незначительно повышается. Это показывает, что слишком много итерации приведет к неблагоприятному результату. Учитывая высокую размерность и нелинейность этой проблемы, наш результат является обоснованным. Этот пример показывает, что применение перекрестной проверки для алгоритма градиентного бустинга есть эффективным в решении задач многомерного нелинейного моделирования проблем.*

*Ключевые слова:* градиентный бустинг, машинное обучение, критерии останковки, перекрестная проверка, нелинейная логистическая регрессия.

*Форм. 18. Лит. 10.*

DOI 10.32752/1993-6788-2019-1-222-189-196

Вікторія В. Жебка, Володимир І. Виноградов,  
 Андрій П. Бондарчук, Михайло М. Степанов

## ОПТИМИЗАЦИЯ РАБОТЫ АЛГОРИТМУ ГРАДИЕНТНОГО БУСТИНГУ ЗА ДОПОМОГОЮ ПЕРЕХРЕСНОЇ ПЕРЕВІРКИ

*У статті розглянуті попередні дослідження перехресної перевірки. Показано як вона послідовно оцінює помилку прогнозування шляхом компромісу до заміщення і дисперсії і показано її застосування в якості критеріїв для зупинки алгоритму градієнтного бустингу. Використовуючи змодельовані дані представлено збіжність і узгодженість алгоритму градієнтного бустингу. Проведено порівняння результатів для різних правил зупинки. Розглянуто модель нелінійної логістичної регресії, де метод регуляризації не має рішення. В результаті проведення розрахунків отримано, що рівень похибок дуже швидко зменшується протягом перших кількох ітерацій, потім швидкість спаду стає приблизно лінійною. Коли рівень похибок досягає найнижчого рівня, далі він незначно підвищується. Це показує, що занадто багато ітерацій призведе до несприятливого результату. З огляду на високу розмірність і нелінійність цієї проблеми, наш результат є обґрунтованим. Цей приклад показує, що застосування перехресної перевірки для алгоритму градієнтного бустинга є ефективним в розв'язанні задач багатовимірного нелінійного моделювання проблем.*

*Ключові слова:* градієнтний бустинг, машинне навчання, критерії зупинки, перехресна перевірка, нелінійна логістична регресія

<sup>1</sup> State University of Telecommunications, Ukraine.

<sup>2</sup> State University of Telecommunications, Ukraine.

<sup>3</sup> State University of Telecommunications, Ukraine.

<sup>4</sup> National Technical University of Ukraine "Kyiv Polytechnic Institute named after Igor Sikorsky", Ukraine.

Viktoriia V. Zhebka, Volodymyr I. Vynohradov,  
Andrii P. Bondarchuk, Mikhailo M. Stepanov

## OPTIMIZING THE OPERATION OF THE GRADIENT BOOSTING ALGORITHM WITH THE HELP OF CROSS-CHECK

*This article reviews preliminary cross-validation studies. It shows how it consistently estimates the prediction error by tradeoffs of substitution and variance, and shows its application as criteria for stopping the gradient boosting algorithm. Using the simulated data, the convergence and consistency of the gradient boosting algorithm is presented. The results are compared for different stopping rules. A nonlinear logistic regression model is considered, where the regularization method has no solution. As a result of the calculations, it was found that the level of errors decreases very quickly during the first few iterations, then the decay rate becomes approximately linear. When the level of errors reaches the lowest level, then it rises slightly. This shows that too much iteration will lead to an unfavorable outcome. Given the high dimensionality and nonlinearity of this problem, our result is reasonable. This example shows that the cross validation of the gradient boosting algorithm is effective in solving multidimensional nonlinear modeling problems.*

**Keywords:** gradient boosting, machine learning, stopping criteria, cross validation, nonlinear logistic regression.

**Постановка проблеми.** На сегодня важные данные получаются из больших наборов неструктурированных данных. Много зарубежных компаний, где собирается большое количество данных применяют методы машинного обучения и интеллектуальному анализу данных для их обработки. Именно у больших неструктурированных данных компании ищут подсказки для решения проблем. Но очень часто инструменты, которые используются компаниями не позволяют получить достоверную информацию. Не менее важной проблемой является скорость получения информации. С целью улучшения процесса обработки данных используются алгоритмы машинного обучения.

**Анализ последних исследований и публикаций.** К методам и моделям машинного обучения относятся следующие: линейная регрессия, логистическая регрессия, бинарная регрессия, линейный дискриминантный анализ, деревья принятия решения, наивный Байесов классификатор, K – Ближайших соседей (KNN), сети векторного квантования (LVQ), метод опорных векторов (SVM), случайный лес, бустинг (Adaboost), кластеризация, поиск ассоциативных правил, нейронные сети и др.

В [1] сравнивались два алгоритма: дерево решений и нейронные сети. Авторы отдали преимущество дереву решений, хотя в ходе исследования точность каждого алгоритма составляла более 90%.

В [2] авторы рассматривали алгоритмы логистической и линейной регрессия, линейный дискриминантный анализ, дерево решений, K – Ближайших соседей, нейронная сеть.

В [3] рассмотренные такие алгоритмы как K – Ближайших соседей, градиентный бустинг, наивный байес, случайный лес. Однако именно случайный лес дал наилучшую точность (91%) и полноту (87%).

В [4] исследователь Нил А. Акильдири исследовал статистические данные телекоммуникационной компании с помощью алгоритмов случайный лес и

методу опорних векторов. Наилучший результат был получен при использовании метода случайного леса. В результате проведенного моделирования точность для модели случайный лес, которую было создано для прогнозирования состояния клиентов телекоммуникационной компании, составляла 0,89. Для метода опорных векторов этот показатель составлял лишь 0,647.

В [5] освещены результаты исследования методов логистической регрессии (точность 0,87) и случайного леса (точность 0,96).

Усовершенствование в данной области возможно путем применения комбинации методов. Например, в [5] освещено исследования, которые определяют, что эффективные есть комбинация регрессии Кокса (модель пропорциональных рисков), которая позволяет прогнозировать риск и оценить влияние на него независимых сменных, с бинарной регрессией, которая определяет вероятность отказы от услуги и определения факторов, которые влияют на нее. В указанном исследовании установлено, что применение комбинации этих методов позволит на основе исторических данных о пользователях определить причины и вероятность оттока клиентов в установленном временном промежутке.

В процессе рассмотрения разных комбинаций алгоритмов остановимся на понятии бустингу – ансамблевом методе, главным преимуществом которого является последовательное использование слабых алгоритмов, причем каждый алгоритм обращает внимание на ошибки предыдущего.

**Цель исследования:** разработать критерий для остановки алгоритма градиентного бустинга, что позволит обеспечить сходимость и согласованность алгоритма градиентного бустинга.

**Основные результаты исследования.** Практическая проблема для алгоритма градиентного бустинга заключается в том, чтобы своевременно остановить итерацию. Не считаясь с тем, что по мере того, как размер выборки  $n$  и количество итераций  $m$  стремятся к бесконечности, модель градиентного бустингу совпадает с реальной моделью, но на практике мы не хотим выполнять много итераций, поскольку это приведет к проблеме переоборудования. Итак, нам нужен критерий, чтобы остановить итерацию, как только подобранная модель станет достаточно точной. Наиболее часто применимым есть метод определения оптимального количества итераций – перекрестная проверка.

Если нашей целью является построение статистической модели с хорошими характеристиками прогнозирования, то в идеале мы хотим найти такой параметр налаживания, чтобы соответствующая модель имела наименьшую погрешность прогнозирования.

$$T_n = \{(x_1, y_1), \dots, (x_n, y_n)\} \quad (1)$$

Пусть  $n$  – набор выборок,  $f_n^{(M)}$  модель подобрана к  $T_n$  с помощью некоторого алгоритма  $A$ . Оптимальный алгоритм  $A$  имеет ожидаемую погрешность.

$$\Delta = EL(Y, \hat{f}_n^{(M)}(X)) = \min_f EL(Y, f(X)), \quad (2)$$

где математическое ожидание берется на общее распределение  $D$  пары  $(X, Y)$ .

В целом, нет возможности определить точное значение ожидаемой погрешности прогнозирования, поскольку  $D$  не известно. Вместо этого нам необходимо найти альтернативные образа оценки ожидаемой погрешности прогнозирования. Единым вариантом является оценка обучения,

$$\hat{\delta} = \frac{1}{n} \sum_{t=1}^n L(y_t, \hat{f}_n^{(M)}(x_t)), \quad (3)$$

которая представляет собой среднее арифметическое значение функции потерь на все учебных выборках. К сожалению, ошибка обучения не является хорошей оценкой реальной ошибки прогнозирования. Причина простая, если мы учим и тестируем модель с использованием одного и того же набора данных, то модель «знает» данные перед тестированием и, соответственно, имеет преимущество. Фактически, ошибка обучения по обыкновению есть слишком оптимистичной. Модели, которые «слишком хорошо» отвечают учебным данным, могут очень плохо предусматривать будущие данные. Для фиксированного учебного набора  $T_n$  и функции потерь  $L$  определим различие между ожидаемой ошибкой предсказания  $\Delta$  и ошибкой обучения  $\hat{\delta}$ . То есть

$$\varphi = \varphi(T_n) = \Delta - \hat{\delta} \quad (4)$$

Другими словами,  $\varphi(T_n)$  это величина, на которую ошибка обучения занижает реальную ошибку прогнозирования. Пусть  $M$  математическое ожидание  $\varphi(T_n)$

$$M = E\varphi(T_n) \quad (5)$$

где математическое ожидание берется на учебной избирателе  $T_n$ . В большинстве случаев  $M$  неизвестно из-за того, что распределение  $D$  неизвестное. Поэтому его нужно оценивать по учебным выборкам. Пусть  $M_i$  – оценка  $M$ . Соответствующая оценка  $\Delta$  будет определяться как

$$\bar{\Delta} = M + \hat{\delta} \quad (6)$$

Чтобы последовательно оценить ошибку предсказания  $\Delta$ , нам нужен метод, который исправит смещение ошибки обучения  $\hat{\delta}$ . Одним из наиболее распространенных методов есть перекрестная проверка (CV). Построение и оценка модели, которая основывается на том же наборе данных, приведет к завышению оптимистичной оценки эффективности прогнозирования. Перекрестная проверка устраняет эту проблему, распределяя учебный и тестовый наборы данных, чтобы модель не «видела» тестовые данные до того, как она будет протестирована.

Предположим у нас есть  $n$  образцов. Метод перекрестной проверки случайным образом делит выборку на два набора:

Учебный набор, содержит  $n_t$  образцов, и набор проверок, который содержит  $n_v = n - n_t$  образцов. Модель строится с использованием учебного набора, а погрешность прогнозирования оценивается с помощью перекрестной проверки. Конечная оценка погрешности прогнозирования получается путем усреднения погрешностей из всех возможных разделов набора данных:

$$\bar{\Delta} = \left[ n_v \binom{n}{n_v} \right]^{-1} \sum \| y_s - \hat{f}^{-s}(X_s) \|^2_2, \tag{7}$$

где  $X_s$  и входные значения и значения отклика выборок в проверяющем наборе  $S$ .  $f^{-s}$  – модель, подобрана из учебного набора  $S^c$ .

Среди всех возможных вариантов для  $n_v$ , налаживание  $n_v=1$ , приводит к простейшей версии перекрестной проверки, поскольку для этого необходимое наименьшее количество операций. В случае, когда  $n$  направленное к бесконечности, в то же время, когда признаков  $k$  фиксированное количество, оценка CV с исключением одного единственного почти не смещена для реальной ошибки прогнозирования. Однако CV с исключением одного значения не является хорошей оценкой  $\Delta$ , поскольку среднеквадратичная ошибка

$$MSE^{(CV)} = E(\Delta^{(CV)} - \Delta)^2 \tag{8}$$

по обыкновению высокая. Это не сложно объяснить в терминах концепции оптимизма. Для CV с единичным исключением ожидаемый оптимизм оценивается как

$$M^{(CV)} \frac{1}{n} \sum_{i=1}^n L(y_i, \hat{f}^{-i}(X_i)) - \partial = \Delta^{(CV)} - \partial, \tag{9}$$

где  $f^{-i}$  – модель, подобранная из всех учебных выборок, кроме  $i$ -той.

Из этого формулирования мы видим что учебные наборы, которые используются CV с исключением одного единственного для соответствия модели  $f^{-i}$  слишком близкие к исходному набору данных, который не может быть хорошим отображением его реального распределения. Таким образом  $\Delta^{(CV)}$  будет иметь большую дисперсию. Соответственно дисперсия  $M^{(CV)}$  будет высокой. В результате среднеквадратичная погрешность  $MSE^{(CV)}$  также высокая, поскольку формула Эфрона показывает что  $Var(M^{(CV)})$  является важным компонентом  $MSE^{(CV)}$ .

Соответственно, чтобы улучшить производительность  $\Delta^{(MCV)}$  при оценке  $\Delta$  нам необходимо уменьшить его дисперсию с учебным набором  $T_n$ . В этом случае уместной будет идея увеличить размер набора для проверки. Это приводит к CV – delete –  $d$ , где  $d = n_v$  и  $d < I$ . Интуитивно понятно, когда  $d$  становится очень большим, размер  $n_i$  учебного набора будет очень малым, и учебный набор в одном подходе обучения будет очень отличаться от другого. Беря среднее значение моделей, которые учились на этих наборах данных, можно контролировать дисперсию оценки.

$$\bar{\Delta}^{(kCV)} = \frac{1}{k} \sum_{j=1}^k \frac{1}{|S_j|} \| y_{S_j} - \hat{f}^{-j}(X_{S_j}) \|^2_2, \tag{10}$$

Другой альтернативой delete –  $d$  – CV есть тест на повторное обучение (RTL). В некоторых других источниках это также называется перекрестной

проверкой Монте-Карло (*MCCV*). Вместо суммирования по всем возможной разбиениям с  $n_v = d$ , *MCCV* просто берем  $B$  случайных подмножеств  $S^*_1, \dots, S^*_B$  размера  $d$  и оцениваем как:

$$\bar{\Delta}^{(MCCV)} = \frac{1}{Bd} \sum_{i=1}^B \|y_{S_i^*} - \hat{f}^{-S_i^*}(X_{S_i^*})\|_2^2. \tag{11}$$

По данным исследований,  $\Delta^{KCV}$  асимптотично эквивалентно  $\Delta^{MCCV}$  при  $n \rightarrow +\infty$  при определенных условиях.

Для  $j$ -кратной перекрестной проверки, когда количество сум  $k$  небольшое, систематическая погрешность может быть проблемой, в зависимости от того, как эффективность метода обучения зависит от размера учебной выборки. Как правило, производительность прогнозирования должна увеличиваться с увеличением размера учебных выборок  $n$ . Теперь предположим, что  $n = 300$ . Если  $k = 2$ , то используем только половину учебных выборок для построения модели, а другую половину используем как набор для проверки. В этом случае оцененная погрешность предсказания будет приближаться к погрешности предсказания модели, подобранной с использованием  $300/2 = 150$  учебных выборок.

С другой стороны, если  $j$  слишком большое, то производительность оценщика перекрестной проверки  $j$ -кратного размера будет приближена к перекрестной проверке с исключением по одному, которая имеет большую дисперсию. Соответственно,  $j$  необходимо избирать для достижения компромисса между смещением и дисперсией. Хотя задачи сильно отличаются одна от одной, в целом выбор  $j = 10$  широко используется на практике. Эффективность 10-кратной перекрестной проверки доказанная на многих примерах.

10-кратную перекрестную проверку удобно использовать, чтобы избрать правильное количество итераций для алгоритма градиентного бустинга. Будем использовать следующий алгоритм: разделим учебный набор  $T$  случайным образом на 10 подмножеств  $T_1, \dots, T_{10}$ , которые имеют приблизительно одинаковый размер. Для  $i = 1, \dots, 10$ , реализовать машину повышения градиента на множестве  $T/T_i$  с  $p$  итерациями, где  $p$  – заранее заданное целое число, которое достаточно большое. Для  $j = 1, \dots, p$ , пусть будет моделью, аппроксимированного алгоритма градиентного бустинга на  $j$ -той итерации на множестве  $T/T_i$ . Мы используем среднюю погрешность теста на  $f^j_{-i}$ , чтобы оценить погрешность прогнозирования  $\Delta^{(j)}$  на  $j$ -той итерации, и т.п.

$$\Delta^{(j)} = \frac{1}{10} \sum_{i=1}^{10} \frac{1}{|T_i|} \sum_{(x_i, y_i) \in T_i} L(y_i, \hat{f}^j_{-i}(x_i)). \tag{12}$$

Оптимальное количество операций  $j$  это такое количество, при котором оценена выше погрешность прогнозирования минимальная. То есть,

$$k^* = \arg \min_{k \in \{1, \dots, m\}} \Delta^{(j)}. \tag{13}$$

После определения  $j$ , мы снова реализуем алгоритм градиентного бустинга на всех учебных выборках с  $j$  итерациями и получим  $f^{j^*}$  в качестве конечного результата.

Сейчас рассмотрим модель нелинейной логистической регрессии. Пусть  $Y$  – случайная переменная величина, распределение которой зависит от функции регрессии  $f(X)$ :

$$Y_1 = \text{Bernoulli} \left( \frac{e^{f(X)}}{1 + e^{f(X)}} \right). \quad (14)$$

Регрессия  $f$  є лінійною комбінацією функцій-індикаторів.

$$f(X) = \sum_{j=1}^k \beta_j 1_{\{x_j \geq r_j\}} + \varepsilon. \quad (15)$$

Кроме того, мы предполагаем, что модель есть многомерной, однако разреженной. Пусть

$$k = 600, \quad \beta_{1-6} = (1, 0.8, 0.9, -0.9, -0.9, -0.9)^T, \quad \beta_i = 0 \text{ для } i > 6.$$

Также предположим, что каждая с независимых сменных  $X_i \in [0, 1]$  и нелинейных параметров

$$r_{1-6} = (0.61, 0.55, 0.37, 0.75, 0.26, 0.48).$$

Предполагается, что случайный шум  $\varepsilon$  есть  $N(0, 0.01)$ .

Данная модель может быть применена для испытаний, в которых, как известно, необходимо предусмотреть определенный бинарный результат.

Предположим, что мы имеем  $n = 400$  образцов. В данном случае с небольшим размером выборки и многомерность традиционного алгоритм подбора модели логистической регрессии не обеспечивает сходимость. Также невозможно применить методы регуляризации через нелинейный параметр  $r_i$  в базисной функции. К этой проблеме мы применяем машину для повышения градиента.

Что касается модели регулярной регрессии, применяем экспоненциальные потери как функцию потерь. На каждой итерации мы используем функцию оптимизации  $R$ , чтобы максимизировать квадратную корреляцию выборки  $1_{\{x_i \geq r\}}$  с негативным градиентом как односменную функцию  $r$  для каждой независимой сменной  $X_i$ , и мы избираем ковариант  $X_i$  и соответствующий параметр  $r_i$  таким, что основная функция  $1_{\{x_i \geq r\}}$  имеет самую большую квадратную корреляцию выборки с отрицательным градиентом.

После того, как мы избрали оптимальную базисную функцию, мы вычисляем размер шага  $k$  с данной базисной функцией. Однако, чтобы сделать алгоритм более стабильным, вместо того, чтобы использовать полный размер шага  $k$ , мы множим его на коэффициент  $\nu = 0.1$ . Мы принимаем сокращенный размер шага, чтобы не зайти далеко в неправильном направлении.

Сначала мы устанавливаем количество итераций  $m$  равной 150. Для каждого  $j \leq p$  мы используем 10-кратную перекрестную проверку для оценки погрешности прогнозирования модели  $f^j$ , полученную на  $j$ -той итерации.

Вместо оценки экспоненциальных потерь вычисляем коэффициент погрешности

$$E_i^j = \frac{1}{|T_i|} \sum_{(x_i, y_i) \in T_i} 1_{\{y_i \neq 1_{\{\hat{f}_i^j(x_i) \geq 0\}}\}} \quad (16)$$

Здесь под  $T_i$  понимаются случайные подмножества пробного подбора и  $\hat{f}_i^j$  — модель, установленная на  $j$ -той итерации без использования  $T_i$ ,  $i = 1, \dots, 10$ .

Используем среднюю величину

$$E^j = \frac{1}{10} \sum_{i=1}^{10} E_i^j \quad (17)$$

для оценки уровня погрешностей, который отвечает итерационному числу  $j$ . Избираем число  $j^*$  с наименьшим прогнозируемым уровнем погрешностей как оптимальное количество итераций. Повторяем наш алгоритм для  $j^*$  раз на все пробных образцах и выводим соответствующую модель  $\hat{f}^{j^*}$  как наш конечный результат.

Для проверки результата прогнозирования полученной модели генерируем тестовый набор размером  $N = 1000$ . Для  $j = 1, \dots, p$  оцениваем среднюю квадратичную погрешность, используя среднее значение выборки на тестовом наборе. То есть

$$E(\hat{f}^j(X) - f(X))^2 \approx \frac{1}{N} \sum_{i=1}^N (\hat{f}^j(X_i) - f(X_i))^2.$$

**Выводы данного исследования.** В результате проведения расчётов получено, что уровень погрешностей очень быстро уменьшается на протяжении первых нескольких итераций, потом скорость спада становится приблизительно линейной. Когда уровень погрешностей достигает наиболее низкого уровня, дальше он незначительно повышается. Это показывает, что слишком много итерации приведет к неблагоприятному результату. Учитывая высокую размерность и нелинейность этой проблемы, наш результат является обоснованным. Этот пример показывает, что применение перекрестной проверки для алгоритма градиентного бустинга есть эффективным в решении задач многомерного нелинейного моделирования проблем.

1. Condamoor Ravi Building Predictive Models for Customer Churn in Telecom. Access mode: <https://www.experfy.com/blog/building-predictive-models-for-customer-churn-in-telecom>.

2. Canale A., Lunardon N. Churn prediction in telecommunications industry. A study based on bagging classifiers telecom // Carlo Alberto Notebooks, 2014. Vol. 350. P. 1–11. Access mode: <https://www.carloalberto.org/assets/working-papers/no.350.pdf>.

3. Khan A.A., Sanjay J., Sepehri M.M. Applying data mining to customer churn prediction in an Internet service provider // Int. J. Comput. Appl., 2010. Vol. 9. No. 7. P. 8–14. Access mode: <http://www.ijcaonline.org/volume9/number7/pxc3871889.pdf>.

4. Telecom Churn Analysis. Access mode: <https://towardsdatascience.com/customer-churn-analysis-4f77cc70b3bd>

5. Neal A. Akyildirim Brief Overview of Customer Churn Analysis and Prediction with Decision Tree Classifier. Access mode: [http://rstudio-pubs-static.s3.amazonaws.com/277278\\_427ca6a7ce7c4eb688506efc7a6c2435.html](http://rstudio-pubs-static.s3.amazonaws.com/277278_427ca6a7ce7c4eb688506efc7a6c2435.html)

6. Григорчук Т.И., Максименко З.В., Розанова Л.Ф., Бикбулатова Г.Р. Скоринговое моделирование финансовых потоков от взыскания // Электронный научный журнал «Нефтегазовое дело», 2015. № 5. С. 630-655.
7. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Springer, 2008.
8. Tropp, J., Anna, C. and Gilbert, C. (2007). Signal recovery from random measurements via orthogonal match pursuit, IEEE Transactions on Information Theory 53, 4655-4666.
9. Wang, S., Nan, B., Zhu, J. and Beer, D. (2008). Doubly penalized Buckley James method for survival data with high-dimensional covariates, Biometrics 64, 132-140.
10. Wasserman, L. and Roeder, K. (2009). High-dimension variable selection, The Annals of Statistics 37, 2179-2201.

- 
1. Condamoor Ravi Building Predictive Models for Customer Churn in Telecom. Access mode: <https://www.experfy.com/blog/building-predictive-models-for-customer-churn-in-telecom>.
  2. Canale A., Lunardon N. Churn prediction in telecommunications industry. A study based on bagging classifiers telecom // Carlo Alberto Notebooks, 2014. Vol. 350. P. 1–11. Access mode: <https://www.carloalberto.org/assets/working-papers/no.350.pdf>.
  3. Khan A.A., Sanjay J., Sepehri M.M. Applying data mining to customer churn prediction in an Internet service provider // Int. J. Comput. Appl., 2010. Vol. 9. No. 7. P. 8–14. Access mode: <http://www.ijcaonline.org/volume9/number7/pxc3871889.pdf>.
  4. Telecom Churn Analysis. Access mode: <https://towardsdatascience.com/customer-churn-analysis-4f77cc70b3bd>
  5. Neal A. Akyildirim Brief Overview of Customer Churn Analysis and Prediction with Decision Tree Classifier. Access mode: [http://rstudio-pubs-static.s3.amazonaws.com/277278\\_427ca6a7ce7c4eb688506efc7a6c2435.html](http://rstudio-pubs-static.s3.amazonaws.com/277278_427ca6a7ce7c4eb688506efc7a6c2435.html)
  6. Нрыгорчук Т.И., Максименко З.В., Розанова Л.Ф., Бикбулатова Г.Р. Skorynhovoe modelyrovanye fyansovyykh potokov ot vzyiskaniya // Электронный научный журнал «Нефтегазовое дело», 2015. № 5. С. 630-655.
  7. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Springer, 2008.
  8. Tropp, J., Anna, C. and Gilbert, C. (2007). Signal recovery from random measurements via orthogonal match pursuit, IEEE Transactions on Information Theory 53, 4655-4666.
  9. Wang, S., Nan, B., Zhu, J. and Beer, D. (2008). Doubly penalized Buckley James method for survival data with high-dimensional covariates, Biometrics 64, 132-140.
  10. Wasserman, L. and Roeder, K. (2009). High-dimension variable selection, The Annals of Statistics 37, 2179-2201.