Санкт-Петербургский Государственный Университет

Кафедра Системного Программирования

Деркунский Виктор Артурович

Применение Наивного Байесовского классификатора в задаче предсказания ухода абонентов оператора сотовой связи

Курсовая работа

Научный руководитель:

д.ф.- м.н., профессор Терехов Андрей Николаевич

Санкт Петербург, 2018

Содержание

Постановка задачи

Цель работы: по предоставленной выборке разработать классификатор, основанный на наивном байесовском классификаторе и оценить его эффективность для задачи предсказания оттока абонентов оператора мобильной связи.

Задачи:

- Провести обзор существующих решений данной задачи
- Обработать предоставленные данные
- Рассмотреть эффективность наивного байесовского классификатора в задаче предсказания оттока абонентов операторов сотовой связи
- Рассмотреть методы повышения эффективности классификатора
- Рассмотреть ансамбли, основанные на наивном байесовском классификаторе
- Рассмотреть ансамбли с другими моделями машинного обучения

Введение

В успешной отрасли бизнеса чаще всего конкуренция неизбежно возрастает. Операторы сотовой связи не стали исключениями. Компании выдвигают все более интересные предложения, и абоненты, старающиеся найти для себя выгоду, постоянно меняют операторов. Сделать это становится все проще, например, после вступления в силу закона о MNP (Mobile number probability), появилась возможность оставлять себе свой текущий номер, что являлось довольно сильным удерживающим фактором для многих людей. В итоге годовой отток абонентов операторов сотовой связи может достигать 50%.

Сложившаяся ситуация привела к тому, что стало необходимо изучать поведение абонентов. Удержание абонентов в несколько раз выгодней привлечение новых, таким образом, предотвращение оттока абонентов становится ключевым аспектом построения успешного бизнеса. Точные предсказания оттока абонентов дает возможность своевременно реагировать, используя различные подходы, пытаться сохранить пользователей в зоне риска (бонусные системы, дающие ограниченный бесплатный трафик мобильного интернета, некое число бесплатных звонков и так далее).

Задача предсказания оттока абонентов заключается в построении и обучении бинарного классификатора. Всех абонентов надо отсортировать в 2 класса: те, кто хочет остаться, и те, кто находится в зоне риска.

Терминология

- Классификатор – отображение из множества, определяемое нами, наиболее существенных признаков объекта (features) в множество классов
- Ансамбль классификаторов – модель, полученная из нескольких классификаторов, соответственно суть в том, чтобы сделать 1 наиболее эффективный классификатор
- Бустинг – последовательное построение ансамбля классификаторов, в момент построения новый классификатор при добавлении стремится исправить недочеты композиции всех предыдущих
- Бэггинг – ансамбль классификаторов, построенный на принципе их независимого обучения на каком-то подпространстве данных
- Логистическая регрессия – построение линейного классификатора, высчитывающего апостериорные вероятности принадлежности объектов к классам (мы выбираем класс с наибольшей вероятностью)
- Линейный дискриминантный анализ (далее lda) для случая 2 классов – методы статистики и машинного обучения, применяемые для нахождения линейных комбинаций признаков, которые смогут лучше всего разделить положительный и отрицательный класс.

Оценка точности классификация

Разделим все данные на 4 класса

- True Positives (TP) – верно определенные в положительный класс (уходящие абоненты)
- False Positives (FP) – ошибочно определенные в положительный класс
- True Negatives (TN) – верно определенные в отрицательный класс (удержанные абоненты)
- False Negatives (FN) – ошибочно определенные в отрицательный класс

Такие характеристики как точность и полнота являются классическими для оценки точности бинарного классификатора:

- Точность (precision) $precision = \frac{TP}{TP + FP}$
- Полнота (recall) $recall = \frac{TP}{TP + FN}$

Важность этих характеристик зависит от конкретной задачи. Соответственно есть функция, которая делает предпочтения полноте или точности, так называемая F – мера:

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

Так как не неизвестно какая из характеристик наиболее важна, то возьмем b=1 (F1 мера)

Точность и полнота регулируется при тренировки классификатора определенным параметром (predict_prob), называемым порогом. С помощью изменения порога можно увеличить precision до заданного нам значения, в следствии этого recall сильно упадет. Поэтому нам нужна независимая метрика по которой мы можем оценить эффективность классификатора. Такой метрикой является AUC (что это такое введено в следующем абзаце), которой в нашей задаче является самой главной характеристикой. Все значения

precision и recall, а также мера F1 далее представлена для более общей картины. Precision и recall специально приближены друг к другу.

ROC – кривая, которая задается параметрически: $x = \frac{FP}{TN + FP}$, $y = \frac{TP}{TP + FN}$. Следующая величина является количественной характеристикой для ROC – кривой:

- AUC (Area under ROC)

$\int_0^1 \frac{TP}{TP + FN} d \frac{FP}{TN + FP}$

($0 \le AUC \le 1$, в случае равенства 0.5, классификатор считается случайным, если меньше 0.5, то необходимо инвертировать ответы, идеальный случай является равенство 1)

Обзор решений
Наивный байесовский классификатор

Пусть у нас есть объект O с наиболее существенными характеристиками (n параметров, которые мы определяем сами), а также набор классов C, к одному из которых мы должны отнести наш классифицируемый объект. Мы выбираем такой класс, вероятность принадлежности к которому была максимальна. Математически все вышесказанное записывается вот так:

$$c = argmax_{c \in C} P(C|O)$$

Такие вероятности проблематично вычислять, но мы можем прейти к условным вероятностям, используя формулу Байеса:

$$P(C|O) = \frac{P(O|C) * P(C)}{P(O)}$$

Рассматривая формулу, мы можем упростить ее. Заметим, что так как мы ищем максимум, то на знаменатель мы можем не обращать внимание, так как это константа, зависящая от выборки данных. Вдобавок на этом шаге, мы можем выделить n параметров, наиболее точно описывающие наш объект. Тогда мы сводим нашу задачу к такой:

$$P(C|O_1 O_{2....} O_n) = \frac{P(O_1 O_2 .... O_n|C) * P(C)}{P(O_1 O_2 .. O_n)}$$

Далее заметим, что знаменатель является константой, которая определяется входными данными. Значит можно его не считать. А числитель можно переписать, используя «наивность», которая заключается в том что все n параметров независимы, то есть конечная формула выглядит так:

$$c = argmax_{c \in C} P(C|O_1 ... O_n) = argmax_{c \in C} P(c) \prod_{i \in 1..n} P(O_i|C)$$

Эти вероятности считать уже легко, в общем случае мы сильно упростили задачу и их нахождение это подсчет числа вхождение объекта в классы.

Уже написано множество работ, главная цель которых найти наиболее перспективную модель машинного обучения для предсказания оттока абонентов. Цель дальнейшего разбора выявить, как показывал себя Наивный байесовский классификатор (Naive Bayes) в аналогичных задачах, а также выбрать наиболее подходящие модели для ансамбля с ним.

Автор	Источник	Модели
V. Umayaparvathi, K. Iyakutti[1]	Сингапурский оператор сотовой связи	Деревья решений, нейронные сети
Yaya Xie, Xiu Li и др. [2]	База клиентов китайского банка	Лес случайных деревьев
А.А. Карякин и А.В. Мельников [3]	Челябинский Государственный университета	Решающее дерево, лес случайных деревьев, к-ближайших соседа, Naive Bayes, бустинг
Д.Ю. Мамонтов, Т.С.Карасев [4]	Сибирский Государственный аэрокосмический университет	Naive Bayes, к-ближайших соседа, решающее дерево, нейронная сеть, lda
М.А. Корытов [5]	Бакалаврская работа 2015 года	Бустинг над DT, лес случайных деревьев, нейронная сеть, логистическая регрессия
А.А. Сулягина [6]	Реферат 2015 года	Бустинг над DT , нейронная сеть, лес случайных деревьев

В статье [4] рассматривается эффективность 5 интересующих нас алгоритма машинного обучения, такие как к-ближайших соседа(k-NN), lda, решающее дерево(DT), Naive Bayes, нейронная сеть(ANN), на 5 задачах, схожих с нашей, то есть главное целью было найти оптимальную модель для построения бинарного классификатора(определить объект к одному из 2 классов – одобрен / не одобрен, готов / не готов и другие).
Так как модели рассматриваются на разных входных данных, можно дать достаточно независимую оценку их эффективности. В большинстве задач lda показал стабильно плохой результат, почти везде худший, однако возможно это связано с недостаточной настройкой параметров и я считаю, что данный классификатор недооценен, поэтому буду использовать его в ансамблях вместе с Naive Bayes, который наоборот показал хорошие результаты и является одним из лучших вместе с DT.
В отличие от статьи [4], в [3] Naive Bayes является худшим классификатором, это важно, потому что наши входные данные более схожи с описанными в [3] статье. Зато DT всегда показывает лучшие результаты, то есть его можно явно выделить как наиболее перспективную модель.
Наряду с ним хороший результат в [3] работе показали k-nn и Random Forest. Вдобавок к данной оценки на 5 разных задачах в [4] статье были использованы методы улучшения эффективности классификатор, такие как бэггинг и бустинг. Бустинг над решающими деревьями, оказался самым эффективным и показал отличный результат на всех задачах, кроме одной, где k-nn стал лучшим после проведенных улучшений. Бустинг показал более лучшие результаты чем бэггинг, но результаты являются сравнимыми. Также были рассмотрены различные ансамбли основанные на этих классификаторах, примечательный результат в том, что во

многих ансамблях показавших хороший результат на данных задачах, входит Naive Bayes и lda.
В работе [1] и [2] самым эффективным является алгоритм DT, однако вплотную приближаются результаты нейронных сетей.
В [5] и [6] статье лучший результат показали градиентный бустинг над решающими деревьями и лес случайных деревьев, тогда как один из самых перспективных алгоритмов – нейронные сети, наоборот показали более худший результат.
Эти работы проводились идентичными данными, поэтому именно их мы возьмем за эталон для сравнения и постараемся улучшить.
Итог:
Таким образом мы можем сделать список самых перспективных моделей машинного обучения, для рассмотрения их эффективности в ансамбле с Naive Bayes и без него:

- DT
- Random Forest
- lda
- логистическая регрессия
- k-NN
- бустинг над решающими деревьями

А также выделим результат на который мы можем опираться:

Лучший результат 2015 [6]	Precision	Recall	AUC	F1
	0.75	0.72	0.92	0.73

Данные

Были предоставлены данные 300000 абонентов и их активность за последние 3 месяца. Каждый абонент описывался через следующие характеристики:

• Количество минут и стоимость исходящих
• Количество минут и стоимость входящих
• Количество и стоимость исходящих СМС
• Количество и стоимость входящих СМС
• Количество трафика мобильного интернета и его стоимость
• Информация о количестве обращений клиента в справочные службы
• Личные данные

Была произведена работа над данными: приведение их к требуемому для обучения виду. Те колонки которые, которые описывали одни и те же вещи были агрегированы в один признак обычным сложением, например 2g, 3g и 4g трафик были сложены. Итогом такой деятельностью стало то, что наши данные стали описываться через 130 параметров.

Далее, чтобы уменьшить линейную независимость параметров, что очень существенно влияет на результат Naive Bayes методом главных компонент (PCA) размерность данных была снижена до 58 параметров.

В итоге все реализованные модели почти не изменили своих результатов, за исключением Naive Bayes, который оказался наиболее чувствителен к линейной зависимости данных.

Оценка эффективности моделей выбранных в обзоре

Первым делом был построен из самых эффективных классификаторов работ [5] и [6] – бустинг над решающими деревьями (XGBoost). Наиболее значимыми параметрами оказались – количество классификаторов, входящие в ансамбль, максимальная глубина дерева, максимальный размер подпространства данных, а так же максимальный обучаемый уровень (то есть вклад в конечный ответ)

Далее был реализован k-NN алгоритм, в процессе настройки параметров оказалось, что самые существенные изменения в классификации зависит от таких параметров, как алгоритм выбора ближайших соседей, также их количество и веса, которые придавали большее значение самым схожим объектам из обучаемой выборки.

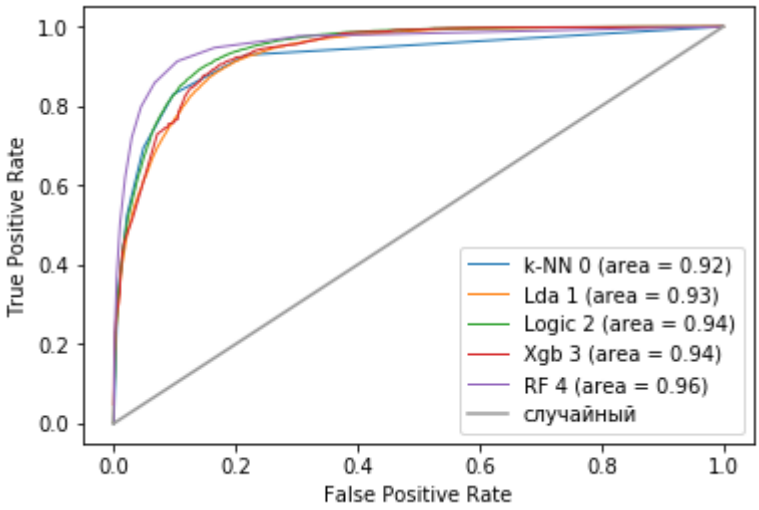
Решающее дерево, которое показывало стабильно хорошие результаты в описанных ранее работах, было тоже реализовано и настроено. Самые главные параметры оказались: максимальная глубина дерева, максимальное количество features и вид дерева (лучший результат у сбалансированного с глубиной 7)

Лес случайных деревьев должен показывать еще более лучшие результаты в задачах, так как это композиция DT, которая минимизирует ошибку допущенную одним DT. В итоге результат больше всего зависел от количества DT в ансамбле, также их глубиной и уровнем обучения.

Классификатор lda имеет специальный параметр, который

корректирует алгоритм в зависимости от числа features, так как у нас их достаточно много, то это существенно отражается на результате.

Снизу отражена результативность данных моделей машинного обучения на наши данных:



Более подробно эффективность выбранных моделей видна на табличке:

	precision	recall	AUC	F1
k-NN	0.77	0.69	0.92	0.73
lda	0.79	0.56	0.93	0.66
регрессия	0.77	0.66	0.94	0.71
Xgb	0.7	0.72	0.94	0.71
RF	0.85	0.72	0.96	0.78

Когда будем строить ансамбли следует особое внимание уделить RF, так как по раздельности он показал наилучшие результаты. Даже не улучшая его результат он показал эффективность большую чем было указанно в работе[6]

Оценка эффективности Naive Bayes

В процессе работы был реализован сам Наивный байесовский классификатор и подобраны его параметры (предварительные вероятности классов мы знаем, поэтому для повышение точности классификации можем их указать)

Далее были использованы методы улучшения эффективности классификации, такие как бэггинг с разным количеством независимо обученных классификаторов на подпространствах входных данных, где голосование проходит по принципу большинства, и где есть линейный нейрон, который пытается подобрать необходимые веса.

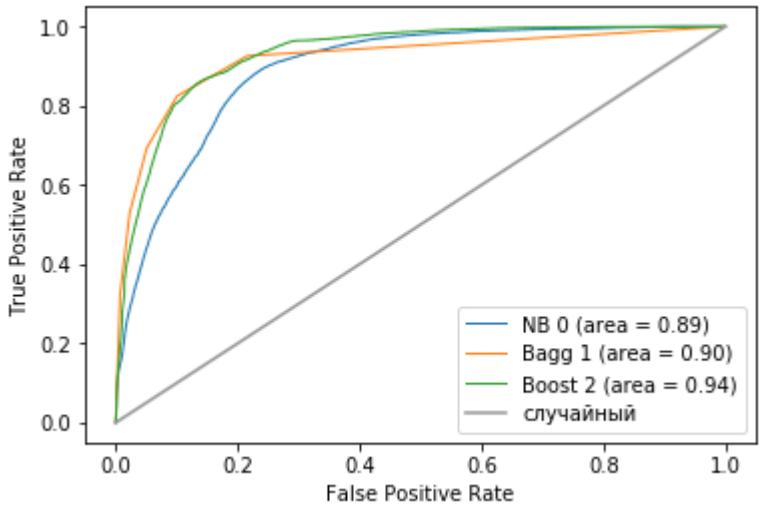
Из моделей построенных на голосовании по принципу большинства для 20, 50, 100, 150 и 200 классификаторов, обученных независимо друг от друга(кросс валидация), результат оказался идентичным, так как для обучения большего числа классификаторов данных не хватало и они обучались на схожих кусках данных много раз. Далее рассмотрели модель, когда в голосовании, значительную роль вклада классификатора в конечный ответ определяется весом, который подбирается нейроном.

Была построена модель из 4 независимо обученных Naive Bayes классификаторов и произведена настройка весов, результаты получились сравнимы с бэггингом над 20 классификаторами, где голосование определялось по принципу большинства.

Далее был рассмотрен другой метод повышение эффективности классификатора – бустинг. Его результаты сильно превзошли результаты бэггинга над Naive Bayes .
Бустинг проводился на Naive Bayes без настройки

параметров, они все были взяты с базовыми значениями, так как любая настройка была невозможна ввиду случайности обучаемой выборки. Сам же бустинг состоял из 10 таких классификаторов. Наиболее важные параметры для него оказались выбор типа самого алгоритма бустинга (был выбран дискретный , а не классический – обычно он работает медленнее и дает более плохой результат, но в нашей задаче оказался более эффективнее хотя, действительно, значительно медленнее).

В итоге давайте сравним полученные результаты (Naive Bayes, лучший результат бэггинга и бустинга):



	recall	precision	AUC	F1
NB	0.54	0.66	0.89	0.6
бэггинг	0.56	0.67	0.9	0.61
бустинг	0.72	0.76	0.94	0.74

Видно, что лучше всего себя показывает бустинг над Naive Bayes. Его результат является лучше указанных в работе [6].

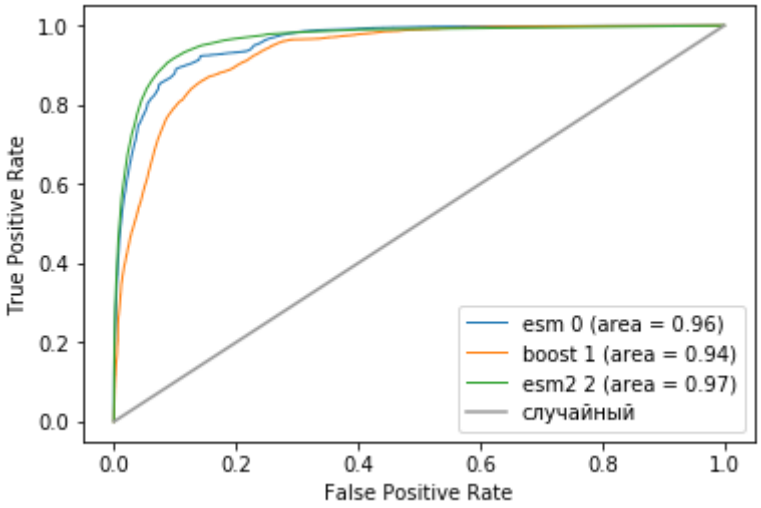
А также сравним с самыми перспективными моделями из обзора. Поэтому можно считать что Naive Bayes является эффективным методом для решения задач заданного класса (бинарной классификации)

Ансамблирование с другими методами машинного обучения

Разделим 2 вида ансамблей, с вхождением Naive Bayes, как составной части ансамбля, и без его участия.
Далее основываясь на выбранные нами самые перспективные методы, описанные в обзоре, были построены все возможные ансамбли построенные по 2 схемам:

- количество классификаторов равно 3
- входит / не входит Naive Bayes, соответственно либо схема №1 / №2
- веса распределены следующем образом: у Naive Bayes вес 1, а у двух оставшихся сумма весов тоже равна 1 / либо все веса выбираются произвольным образом
- классификаторы выбираются из тех, что были рассмотрены

Достигнутые результаты:



Более подробно:

	precision	recall	AUC	F1
Ансамбль 1)	0.75	0.83	0.96	0.79
Бустинг 2)	0.72	0.76	0.94	0.74
Ансамбль 3)	0.76	0.84	0.97	0.8

1) – это 1 метод построения ансамбля, когда Naive Bayes имеет существенное влияние на конечный результат. Вместе с ним в композицию входит лес случайных деревьев и k-NN
2) – это уже описанный бустинг над Naive Bayes. Представлен в данной таблице для того, чтобы можно было наглядно увидеть последовательное улучшение эффективности классификатора
3) – это 2 метод построения ансамбля, Naive Bayes в него не вошел, а такие модели как лес случайных деревьев, логистическая регрессия вместе с lda показали самый лучший результат.

Все модели включались как с полностью настроенными параметрами, так и с полностью стандартными значениями, веса перебирались с шагом 0.2 . Лучше всего себя показали ансамбли с композициями из разных моделей обучения, однако и классификаторы на алгоритме бустинга над Naive Bayes показали очень достойный результат.

Общие результаты:

Naive Bayes(NB)

	Accuracy	Precision	Recall	AUC	F1
Лучший результат прошлых лет[6]	0.88	0.72	0.75	0.92	0.73
NB	0.89	0.54	0.66	0.89	0.6
Бэггинг NB	0.9	0.56	0.67	0.9	0.61
Бустинг NB	0.91	0.72	0.76	0.94	0.74
Ансамбль с NB(*)	0.92	0.75	0.83	0.96	0.79
Ансамбль без NB(**)	0.93	0.76	0.84	0.97	0.8

*ансамбль из леса случайных деревьев+ NB+ k-NN

**ансамбль из леса случайных деревьев+lda+логистической регрессии

Итог:

- Улучшены результаты прошлых лет[6]
- Реализованы классификаторы и настроены их параметры
- Проведен сравнительный анализ реализованных классификаторов

Заключение

По предоставленной выборке были построены классификаторы на основе наивного байесовского классификатора, а также на основе ансамблей построенных методом бэггинга и бустинга. Еще были рассмотрены ансамбли на других методах машинного обучения, с помощью различных моделей ансамблирования удалось достичь результата лучше чем в сравниваемых работах, поэтому можно считать что результат работы успешен.

Список литературы:

[1] V Umayaparvathi and K Iyakutti. Applications of data mining techniques in telecom churn prediction. International Journal of Computer Applications, 42(20):5–9, 2012.
[2] Shin-Yuan Hung, David C Yen, and Hsiu-Yu Wang. Applying data mining to telecom churn management. Expert Systems with Applications, 31(3):515– 524, 2006.
[3] А.А. Карякин и А.В. Мельников. Сравнение моделей прогнозирования оттока клиентов интернет — провайдеров. https://jmlda.org/papers/doc/2017/no4/Karyakina2017Churn.pdf (дата обращения: 20.05.2018)
[4] Д.Ю. Мамонтов, Т.С.Карасев. Журнал «Современные технологии поддержки принятия решений в экономике ». Статья «Эффективность методов интелектуального анализа данных при решении задач банковской сферы».(стр. 250-256),2017
[5] Максим Корыстов. Дипломная работа. Применение методов машинного обучения для предсказания поведения абонентов сотовой связи. 2015
[6] Сулягина Анастасия. Курсовая работа. Применение методов машинного обучения для предсказания поведения абонентов сотовой связи. 2015