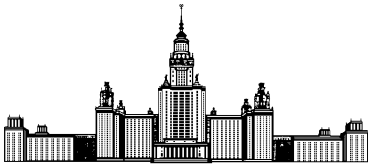
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

Отчет по преддипломной практике

«Прикладные задачи анализа данных»

Выполнила:
студентка 4 курса 417 группы
Рысьмятова Анастасия Александровна

Научный руководитель:
д.ф-м.н., профессор
Дьяконов Алксандр Геннадьевич

Руководитель практики:
специалист по анализу данных
Нехаев Антон Вадимович

Москва, 2015

Содержание

1 Введение

Преддипломная практика была пройдена в компании Алгомост. Данная компания занимается анализом и обработкой данных. В ходе преддипломной практики решались две задачи:

1. Задача кросс-продаж.
2. Задача кредитного скоринга.

Все задачи решались, используя язык python.

В данной компании автором было подписано соглашение о неразглашении, поэтому в отчете указывается не вся информация о данных.

В отчете будут описаны обе задачи, все методы их решения и проблемы, которые появлялись в ходе работы.

2 Задача 1

В задаче необходимо было отранжировать все объекты в порядке вероятности покупки страхового полиса клиентом для необходимого продукта, выбранного в качестве целевого вектора.

Целевой вектор - бинарный вектор, состоящий из нулей и единиц.

Метрикой качества, был выбран AUC.

2.1 Данные

Данные по задачи были предоставлены одной Российской страховой компанией и содержали следующие файлы:

1) *All_have* - информация о всех клиентах, имевших продаваемые сейчас продукты страхования по состоянию на 01.11.2015. Все объекты в этом файле имели целевой признак равный единице.

2) *Кампания 1* – клиенты застраховавшие/не застраховавшие целевой продукт в рамках кампании 1. Дата среза – перед стартом кампании. Кампания завершена.

3) *Кампания 2* – клиенты застраховавшие/не застраховавшие целевой продукт в рамках кампании 2. Дата среза – перед стартом кампании. Кампания активна.

4) *Кампания 3* – клиенты застраховавшие/не застраховавшие целевой продукт в рамках кампании 3. Дата среза – перед стартом кампании. Кампания активна.

Во всех данных находилась информация об 13000 клиентах (объектах) страховой компании, которые когда-либо пользовались ее услугами.

Каждый объект имел 334 признака, среди которых были категориальные, вещественные, текстовые признаки. В данных присутствовало много пропусков. Для того, чтобы

можно было использовать большинство алгоритмов машинного обучения, реализованных в библиотеки sklearn, необходимо было привести все признаки к вещественному виду. Опишем основные этапы обработки данных.

2.1.1 Пропуски в данных

Пропуски в данных удалялись по следующим принципам:

1. Если в признаке доля пропусков больше 0.95, то данный признак удалялся.
2. Если у объекта доля пропусков больше 0.95, то данный объект удалялся из обучающей выборки.
3. Если признак вещественный, то пропуски заполнялись среднем значением.
4. Если признак категориальный, то пропуск рассматривался как еще одна категория признака.

2.1.2 Константные признаки

В данных присутствовали константные признаки. Данные признаки удалялись следующим образом

1. Если в признаке присутствовало лишь одно значение, то он удалялся.
2. Если признак бинарный и доля одного из значений в признаке больше 0.95, то данный признак удалялся.

2.1.3 Текстовые признаки

Текстовых признаков было немного и все они означали название населенных пунктов. Данные признаки были переведены в категориальные, путем нумерации всех населенных пунктов натуральными числами. Также проводились эксперименты, используя другие методы кодирования категориальных признаков, но результат на кросс-валидации оставался тем же.

2.1.4 Опечатки в данных

Данные имели опечатки и несоответствия. Например, признак “Возраст ” не всегда соответствовал признаку “Дата рождения” на момент даты среза, поэтому для анализа использовалась только информация о годе рождения клиента.

После приведения данных к вещественному виду и удаления всех пропусков был запущен алгоритм Random Forest с использованием кросс-валидации по 10 фолдам. Без

настройки параметров данный алгоритм дал качество близкое к 1. Столь высокое качество получалось за счет того, что некоторые признаки как-то содержали ответ внутри себя. Приведем пример такого признака.

Рассмотрим признак "VIP клиент". Заменим все пропуски в данном признаке на '1' и построим гистограмму объектов по данному признаку. На рисунке 1 изображена гистограмма распределения признака . Красным цветом выделена гистограмма объектов по признаку "VIP клиент " для тех объектов у которых целевой признак равен 1, синим цветом для остальных.

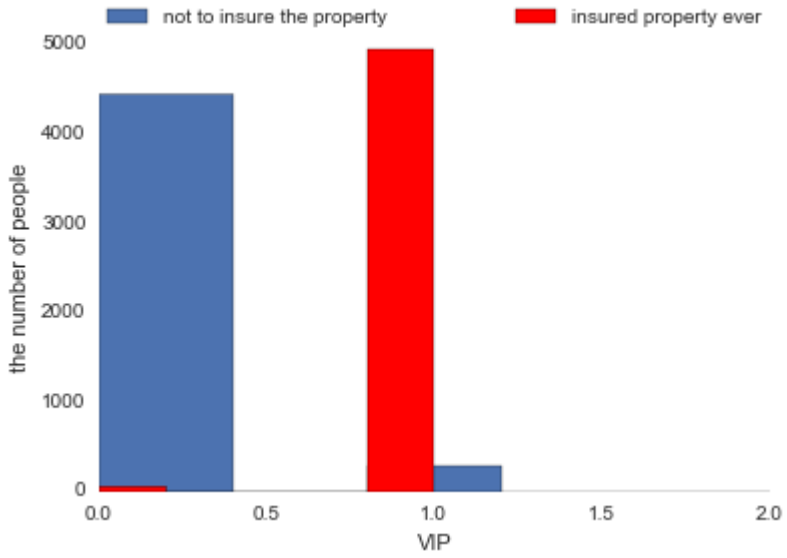


Рис. 1: Гистограмма объектов по признаку " VIP клиент "

Из данного графика видно, что объекты практически полностью разделяются по признаку "VIP клиент". Поэтому данный признак не информативен. Для корректного решения задачи необходимо было найти все подобные признаки и удалить.

Еще одной особенностью данных было то, что целевой вектор сильно зависел от признака "Возраст". Используя лишь этот признак можно было достичь неплохого результата.

Объединив все данные и удалив объекты, в которых отсутствует поле "Возраст" покажем, как распределяется возраст объектов, которые страховали когда либо застраховали необходимый продукт и которые отказались от страхования.

На Рис.2 красным цветом выделена гистограмма объектов по признаку "Возраст" для тех, кто когда-либо страховал необходимый продукт, синим цветом для тех, кто отказался от страхования. Из данного графика видно, что объекты год рождения которых меньше 1965 и больше 1985 почти всегда отказываются от страхования. Это может быть связано с тем, что данные сформированы неверно (выборка не равномерна по данному признаку).

Все признаки можно разделить на две группы:

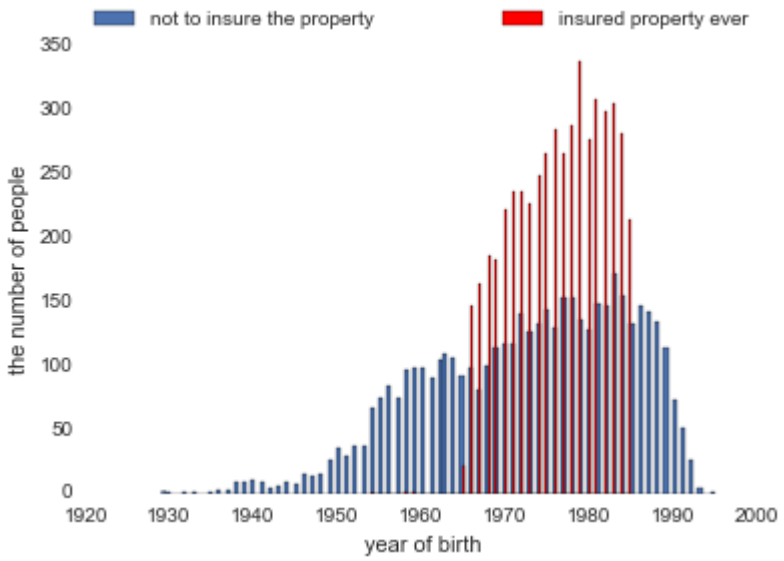


Рис. 2: Гистограмма объектов по признаку “Возраст”

1. Личные данные клиента.

2. Данные о ранее застрахованном имуществе.

В ходе решения задачи было замечено, что при удалении некоторых признаков из личных данных клиента, качество работы используемых алгоритмов машинного обучения не ухудшалось, а иногда даже становилось лучше. Было решено удалять такие признаки.

После всех преобразований в данных осталось 27 признаков.

2.2 Методы локального контроля качества

В решаемой задаче данные обладали некоторой особенностью. В файле *All_have* все объекты имели целевой признак 1, а в остальных файлах почти все объекты имели целевой признак 0. Но во всех файлах кроме *All_have* содержалась информация только о клиентах согласившихся/отказавшихся от страховки после 2014 года.

Для решения задачи использовалось несколько методов, приведем их основные идеи.

2.2.1 Метод 1

Объединить все файлы и использовать различные методы машинного обучения.

При решении задачи данным методом было проведено множество экспериментов с различными алгоритмами машинного обучения. Приведем те, которые показали наилучший результат.

1. KNN
2. Random Forest
3. Xgboost

Данные алгоритмы применялись ко всем обработанным данным. Ниже в таблице приведены результаты данных алгоритмов обучения по метрике AUC. Использовалась кросс-валидации по 10 фолдам.

Алгоритм	Качество на настроенном алгоритме
KNN	0.71
Random Forest	0.86
Xgboost	0.88

2.2.2 Метод 2

Использовать лишь те объекты, которые застраховали свое имущество после 2014 года. Данный метод более приближен к решаемой задаче. Но при использовании такого подхода возникла проблема - некоторые признаки были получены после приобретения страхового полиса. Такие признаки пришлось обрабатывать особым образом.

При решении задачи данным методом было проведено множество экспериментов с различными алгоритмами машинного обучения. Наилучший результат показали те же алгоритмы, что и для метода 1.

Алгоритмы применялись ко всем данным, в которых клиент приобрел необходимый продукт страхования начиная с 2014 года. Ниже в таблице приведены результаты данных алгоритмов обучения по метрике AUC. Использовалась кросс-валидации по 10 фолдам.

Алгоритм	Качество на настроенном алгоритме
KNN	0.73
Random Forest	0.91
Xgboost	0.92

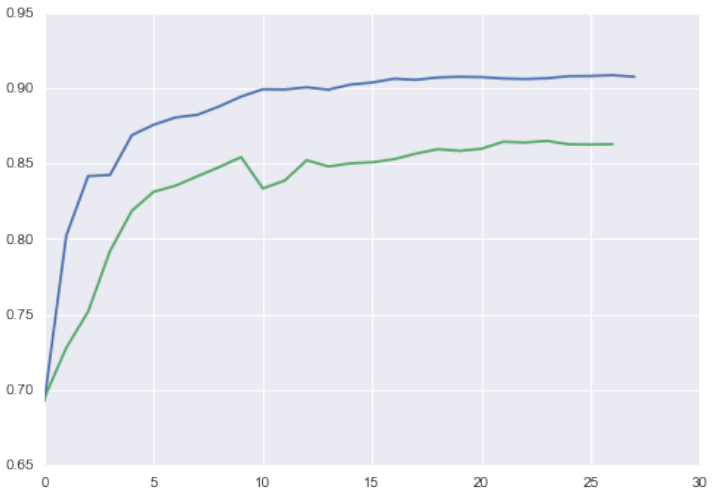


Рис. 3: График зависимости качества, от количества используемых признаков.

Покажем как менялось качество двух описанных методов при добавлении признаков с использованием алгоритма Random Forest и кросс-валидации по 10 фолдам. Признаки будем добавлять в порядке их важности в Random Forest.

В данном графике по оси абсцисс отложено количество признаков, а по оси ординат получившееся качество с помощью алгоритма Random Forest. Зеленым цветом показан метод 1, синим метод 2.

Из данного графика можно сделать вывод, что метод 2, всегда показывает результат лучше метода 1 и причем при добавлении некоторого признака если качество улучшалось в методе один, то оно почти всегда улучшалось и в методе 2. Так же видно, что после добавления 20 признаков качество почти не улучшалось.

2.3 Результаты

В качестве окончательного результата была выбрана модель использующая алгоритм Xgboost и метод 2. На локальном контроле данная модель показала 0.92 по метрике AUC.

3 Задача 2

Задача кредитного скоринга возникает в банках и других кредитных организациях при принятии решений о выдаче кредита. Задача заключается в том, чтобы на основе некоторой информации о заявителе обоснованно принять решение - стоит ли ему выдавать кредит или нет.

В решаемой задаче необходимо было отранжировать все объекты по вероятности возврата кредита.

Целевой вектор - бинарный вектор, состоящий из нулей и единиц.

Метрикой качества, был выбран AUC.

3.1 Данные

Данные по задачи были предоставлены одним из Российских банков. В данных имелась информация о 180000 клиентах, (объектах) которым был выдан кредит в этом банке. Значение целевого вектора для каждого объекта было равно 0 - если клиент не вернул кредит вовремя, 1 - если вернул вовремя.

Каждый объект имел 202 признака, среди которых были категориальные, вещественные, текстовые признаки. В данных присутствовало много пропусков. Аналогично прошлой решаемой задачи, было решено привести все признаки к вещественному виду.
Основные этапы обработки данных совпали с этапами в предыдущей задачи.

3.2 Признаки

Все признаки в данной задаче можно было разделить на две группы:

1. Личные данные клиента, заполненные им в анкете.

2. Данные о кредитах получаемых ранее.

Как и в прошлой задаче пришлось столкнуться с проблемой пропусков в данных, эта проблема была решена аналогично задачи 1.

При удалении многих признаков качество алгоритмов машинного обучения не падало, а иногда даже улучшалось, поэтому такие признаки было решено удалить.

Удалось придумать признаки, которые улучшали результат:

- Доход полученный банком от клиента

  В данных имелась информация о процентной ставке, сроке выплаты кредита и суммы кредита. На основе этой информации можно вычислить сумму которую выплатит клиент за все время.

- Остаток денежных средств у клиента в месяц.

  В данных имелась информация о всех доходах и кредитах клиента. Можно вычислить остаток денежных средств на каждый месяц после выплаты по всем кредитам.

3.3 Алгоритмы решения

При решении данной задачи было проведено множество экспериментов с различными алгоритмами машинного обучения. Приведем те, которые показали наилучший результат.

1. Vowpal Wabbit

   В данной задаче было много текстовых признаков и в основном это была некоторая информация о клиентах, которую они заполнили о себе в анкете. Поэтому было решено на таких признаках использовать Vowpal Wabbit, так как с помощью него удобно работать с текстовыми данными и это очень качественная реализация стохастического градиентного спуска для линейных моделей.

2. Random Forest

   Данный алгоритм применялся ко всем обработанным данным.

3. Xgboost

   Данный алгоритм применялся ко всем обработанным данным.

Ниже в таблице приведены результаты данных алгоритмов обучения по метрике AUC. Использовалась кросс-валидации по 5 фолдам.

Алгоритм	Качество на настроенном алгоритме
Random Forest	0.82
Xgboost	0.84
Vowpal Wabbit	0.75

3.4 Результаты

В качестве окончательного результата была выбрана модель использующая линейную комбинацию алгоритмов Xgboost, Random Forest, Vowpal Wabbit. На локальном контроле данная модель показала 0.85 по метрике AUC.

4 Заключение

В ходе выполнения преддипломной практики был получен опыт работы с реальными задачами анализа данных.

Приобретены навыки в программировании на языке python.

Список литературы

[1] Scikit-learn http://scikit-learn.org/stable/

[2] NumPy Tutorial http://wiki.scipy.org/Tentative_NumPy_Tutorial

[3] Python Tutorial https://docs.python.org/3.4/tutorial/

[4] Воронцов К. В., лекции LaTeX, http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf