МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Кафедра вычислительных технологий**

**КУРСОВАЯ РАБОТА**

**АЛГОРИТМИЗАЦИЯ И ТЕСТИРОВАНИЕ МЕТОДОВ АНАЛИЗА БОЛЬШИХ ДАННЫХ**

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.Д. Захаров

(подпись, дата)

Факультет компьютерных технологий и прикладной математики,

3 курс, 36 группа  
Направление 02.03.02 – «Фундаментальная информатика и информационные технологии»

Научный руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Т.А. Приходько

(подпись, дата)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

Краснодар, 2018

ВВЕДЕНИЕ

В современном мире существует огромное количество задач по анализу данных. Так или иначе с этим сталкивается чуть ли не любая более или менее крупная компания вне зависимости от ее деятельности. Помимо этого, анализ данных помогает в совершенно разных отраслях: коммерции, медицине, науке, военной обороне и так далее.

Например, практически в любой крупной телекоммуникационные компании используется анализ данных, чтобы выстроить хорошие отношения со своими клиентами. Аналитики создают алгоритмы, которые предсказывают, какие клиенты потенциально ненадежные и для них применяются меры увеличения лояльности: скидки на пользование услугами, специальные тарифы и так далее.

Другим примером могут послужить множественные алгоритмы, использующиеся в медицине. Все те данные, которые собирают врачи о пациентах не могут попросту лежать без применения. Поэтому на основе таких данных, как истории болезни, планы лечения, анализы и многие другие используются аналитиками, чтобы улучшить систему здравоохранения, например, создавая алгоритм, который по набору входных данных будет ставить диагноз, а возможно и назначать лечения пациенту, может очень сильно снизить нагрузку врачей.

Еще одним очень простым примером может послужить сфера банкинга, а именно решение о выдаче кредита. Алгоритм, получая множественные данные о заемщике, например дату рождения, ежемесячный доход, кредитную историю, образование и прочее, без участия человека оценивает его шансы на своевременную выплату кредита и принимает решение о том, выдавать ли кредит или нет.

1. АНАЛИЗ ДАННЫХ
   1. Представление данных для анализа

Данные, которые можно анализировать представлены в четырех категориях:

* Численные (числовые) данные.
* Интервальные данные.
* Ранговые данные.
* Номинальные данные.

Также успешному анализу могут быть подвергнуты данные, попадающие сразу под несколько категорий из указанных выше, либо стек разнообразных данных.

При этом, существует немного больше типов этих данных:

* Структурированные.

Они представляют собой данные, которые хранятся в фиксированном поле внутри записи. Исходя из этого нетрудно догадаться, что их очень удобно хранить в таблицах или базах данных.

* Неструктурированные.

Это данные, которые могут иметь какой-то один шаблон, но содержание которых будет совершенно разное. Например, сообщения в чате.

* На естественном языке.

Это некоторая разновидность неструктурированных данных, анализ которых требует очень больших знаний и усилий.

* Машинные.

Это тип данных, который представляет собой данные, генерируемые автоматически устройством или приложением без причастия человека.

* Графовые.

Их также называют сетевыми данными. Главной их особенностью является, что в них особое внимание уделяется связям между объектами. Такой тип данных в основном используется в социальных сетях или сервисах по предоставлению таргетированнной рекламы. Также стоит отметить, что для хранения таких данных используются графовые базы данных.

* Аудио, видео и графика.

Этот тип данных является как и непростым для анализа, так и ужасно затратным для хранения. Алгоритмы, распознающие объекты на фото или видео до сих пор остаются одними из самых сложных в плане разработки.

* Потоковые.

Потоковые данные могут принимать любую из перечисленных ранее форм, но они имеют ключевую особенность – данные поступают для анализа не большими массивами за раз, а малыми объемами при возникновении некоторых событий.

Данные могут быть проанализированы либо статистическими методами, либо методами машинного обучения, но об этом подробнее будет написано позже.

Немаловажным фактором является количество этих самых данных. Для наблюдения статистического эффекта желательно, чтобы в одной группе данных было более 30 событий. Конечно, для выбранной тематики сразу понятно, что это значение будет достаточно велико, но упомянуть об этом стоит.

* 1. Шаги анализа данных

Процесс анализа данных состоит из шести основных этапов:

* Назначение цели исследования.

На этом этапе происходит описание того, что вы собираетесь исследовать, какие ресурсы для этого вам понадобятся и какова будет польза. Это очень важный этап, так как неправильно поставленная цель исследования может привести к тому, что вся работа просто-напросто будет проделана впустую.

* Сбор данных.

Также важный этап процесса анализа данных. На этом этапе происходит поиск необходимых данных, анализ качества и доступности этих данных.

* Подготовка данных.

Данные, которые можно получить для анализа далеко не совершенны. В них случаются опечатки, пробелы, нарушения логики. Также данные, полученные из разных источников, могут иметь совершенно разную структуру и форму хранения. Этот этап имеет три фазы: очистка данных – удаляются некорректные значения из источника данных, а также все данные приводятся к одному виду, интеграция данных – разные источники объединяются в один массив данных, преобразование данных – данные приводятся к подходящему формату для использования в конечных моделях.

* Исследование данных.

Исследование данных направленно на более глубокое их понимание. На этом шаге аналитики стараются понять, как переменные взаимодействуют между собой, оценить распределение данных, а также обнаружить выбросы.

* Моделирование данных.

На этом этапе знания предметной области и информации о данных, полученная на предыдущих этапах, используется для ответа на поставленный вопрос. В процессе моделирования используются методы из статистики, машинного обучения, исследования операций и так далее. Построение модели — это итеративный процесс, в ходе которого аналитик выбирает переменные модели, строит ее саму, а потом ее диагностирует. Если диагностика показала недееспособность модели в той или иной степени, то аналитик возвращается на предыдущие этапы и производит изменения, либо же вообще отказывается от данной модели.

* Отображение и автоматизация.

Этот этап не имеет как такового шаблона. Результаты могут быть представлены в различных формах, таких как презентация, отчет, график и так далее.

* 1. Краткий обзор избранных методов анализа данных

Теперь перейдем к рассмотрению непосредственно методов решения поставленной задачи. Для курсовой работы были выбраны три самых популярных метода анализа данных: метод решающих деревьев, метод k ближайших соседей и метод опорных векторов. Помимо них приводится информация о методе ФорЭл, но в практической части он не рассматривается.

1. Метод решающих деревьев (дерево принятия решений).

Он используются для задач квалификации и реже для задач регрессии.

Суть метода заключается в том, чтобы построить дерево (для машинного обучения используются бинарные деревья), где в ребрах дерева записываются атрибуты, от которых зависит целевая функция; в узлах, кроме листьев, записываются атрибуты, по которым происходит принятие решений о том, по какому ребру пойти на этой итерации; в листьях записываются непосредственно сами значения целевой функции. Чтобы принять решение о классификации нового события, необходимо пройти от корня дерева до некоторого листа, принимая на каждом шаге одно из двух решений. Цель метода – построить такую модель, которая будет предсказывать значения целевой функции на основе нескольких входных параметров. Преимуществами его являются:

* Простота в понимании и интерпретации.
* Не требует нормализации данных.
* Работает как с категориальными переменными, так и с интервальными.
* Способен работать с очень большими наборами данных без специальных процедур.

Недостатки метода:

* Проблема получения оптимального дерева является NP-полной иногда даже для простых задач. Таким образом этот алгоритм является «жадным», ведь на каждой итерации мы принимаем решения, основываясь на выборе между двумя локальными вариантами.
* Возможно создания слишком сложных конструкций, которые не полно представляют данные. Это называют «переобучением», а для решения этой проблемы существует метод регулирования глубины дерева.

1. Метод ближайших соседей.

Этот метод является одним из самых простых метрическим классификатором, который базируется на оценивании сходства объектов. Другими словами, анализируемый объект относится к тому классу, к которому относится ближайшие к нему объекты обучающей выборки.

Также этот метод часто уточняют как метод k ближайших соседей по той причине, что, если брать только одного соседа, зачастую алгоритм не устойчив к шумовым выбросам, то есть он выдает неверный ответ не только на объектных-выбросах, но и на ближних к нему объектам. Точно также, нежелательно устанавливать k по числу объектов в выборке. В этом случае алгоритм принимает константное значение на всей выборке. Чаще всего, значение k определяют по критерию скользящего контроля.

Метод k ближайших соседей можно использовать не только для задач классификации, но и для задач регрессии. В этом случае анализируемому объекту присваивается среднее значение среди его k уже известных соседей.

Иногда бывает так, что некоторые объекты в выборке являются более важными, чем другие и ставить их вровень с остальными нельзя. В этом случае алгоритм предусматривает возможность задать в соответствие каждому объекту некоторое значение – вес, который будет использоваться в дальнейшем при определении класса объекта.

Преимущества метода:

* Метод предоставляет данные, с которыми просто работать.
* Алгоритм, написанный для одной задачи, можно использовать и для других задач.
* Алгоритм очень устойчив к выбросам, так как попадание такого значения в число k соседей мала сама по себе, а помимо этого, мало будет и его влияние на ответ.
* Алгоритм легко модифицируется. Можно подобрать подходящие функции сочетаний и метрики для каждой задачи.

Недостатки метода:

* Значительная сложность выбора метрик, то есть меры близости между объектами.
* Большая зависимость результатов от выбранной метрики.
* Алгоритм требует хранение всей выборки, вследствие чего подходит лишь для относительно небольших по количеству классов и переменных задач.

1. Метод опорных векторов.

С помощью данного метода решаются задачи классификации и регрессии. Этот метод относится к набору методов линейных классификаторов, особенностью которого является уменьшение эмпирической ошибки классификации и увеличение зазора.

Суть алгоритма состоит в том, чтобы построить несколько гиперплоскостей между некоторым количеством классов, хотя зачастую этот метод остается бинарным классификатором, и выбрать такую гиперплоскость, которая будет наилучшим образом разделять объекты этих классов. Считается, что чем больше расстояние между классами (зазор), тем лучше проведена классификация.

Хоть этот метод и считается линейным классификатором, в 1992 был предложен способ создания нелинейного классификатора, путем перехода от скалярных произведений к функциям произвольных ядер, которые как раз и позволяют строить нелинейные классификаторы. Такие функции находятся в пространствах с большей размерностью, в которых уже может существовать оптимальная разделяющая гиперплоскость.

Преимущества метода:

* Метод является наиболее быстрым в поиске решающих функций.
* Метод находит разделяющую гиперплоскость самой большой ширины, что может способствовать минимизации ошибок при дальнейшей классификации.

Недостатки метода:

* Метод очень чувствителен к шумам.
* Метод требует стандартизации исходных данных.
* Отсутствие автоматического выбора подходящего ядра в случае линейной неразделимости классов.

1. Алгоритм ФорЭл.

ФорЭл (Формальный Элемент) – алгоритм кластеризации, который объединяет в один кластер объекты в областях их скопления. Другими словами, необходимо разбить выборку на некоторое неизвестное заранее количество таксонов, объекты в которых будут максимально близки друг к другу и в силу гипотезы схожести, образовывать нужные нам кластеры. Данный алгоритм применяется для задач кластеризации и задач ранжирования выборки.

Данный алгоритм требует выполнения двух условий:

* Выполнение теории компактности, которая гласит, что расположенные поблизости объекты, будут принадлежать одному кластеру с большей вероятностью.
* Наличие метрического и линейного пространств кластеризуемых объектов.

Суть алгоритма состоит в том, что на каждом шаге происходит выбор некоторого случайного объекта, вокруг которого строится сфера радиусом R. Далее внутри этой сферы следует найти объект, который является центром тяжести и делаем его центром новой сферы. Таким образом, на каждой итерации мы передвигаем нашу сферу в место наибольшего скопления объектов. После того как сфера примет свое окончательное положение, происходит удаление из выборки всех объектов, попадающих в нее, и процесс повторяется заново до тех пор, пока вся выборка не будет кластеризована.

Преимущества метода:

* При хорошем подборе радиуса R высока точность минимизации функционала качества.
* Наглядная визуализация кластеризации.
* Доказано, что алгоритм сходится за конечное время.
* Знание центров кластеризации позволяет совершать над ними операции.

Недостатки метода:

* Производительность.
* Зависимость результата от начальной разделимости выборки.
* Алгоритм неустойчив, так как зависит от начального выбранного объекта.
  1. Практическая реализация методов анализа данных

Реализация всех трех алгоритмов будет происходить в среде разработке PyCharm 2018.2 на языке программирования python 3.7. Первым будет реализован метод решающих деревьев или как его называют дерево принятия решений. Для примера будет использоваться один и тот же датасет – так называемый «Ирисы Фишера». Он включает в себя данные о 150 экземплярах ириса, по 50 экземпляров на каждый из трех видов – Ирис щетинистый (Iris setosa), ирис виргинский (Iris virginica), ирис разноцветный (Iris versicolor). Для каждого экземпляра измерялись четыре характеристики:

* Длина наружной доли околоцветника;
* Ширина наружной доли околоцветника;
* Длина внутренней доли околоцветника;
* Ширина внутренней доли околоцветника.

Это стандартный пример для задач классификации, и он отлично подойдет для сравнения трех вышеназванных алгоритмов.

Так как для всех алгоритмов используется один и тот же набор данных, то и загружать его придется только один раз. Приведем небольшой листинг кода загрузки данных:

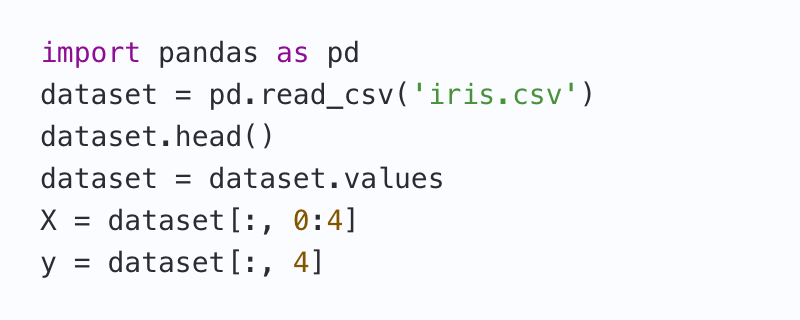


Рисунок 1 – Пример кода для загрузки датасета.

Также, большинство алгоритмов машинного обучения сильно чувствительны к шкалированию данных. Поэтому перед запуском алгоритмов чаще всего делается либо нормализация, либо так называемая стандартизация. Нормализация предполагает замену номинальных признаков так, чтобы каждый из них лежал в диапазоне от 0 до 1. Стандартизация же подразумевает такую предобработку данных, после которой каждый признак имеет среднее 0 и дисперсию 1. В Scikit-Learn уже есть готовые для этого функции:



Рисунок 2 – Пример кода для нормализации данных датасета.

Помимо стандартизации часто является целесообразно проводить анализ не по всем признакам, а по самым важным. Отбор признаков не является чем-то сложным, поскольку есть уже большое количество готовых алгоритмов, например:

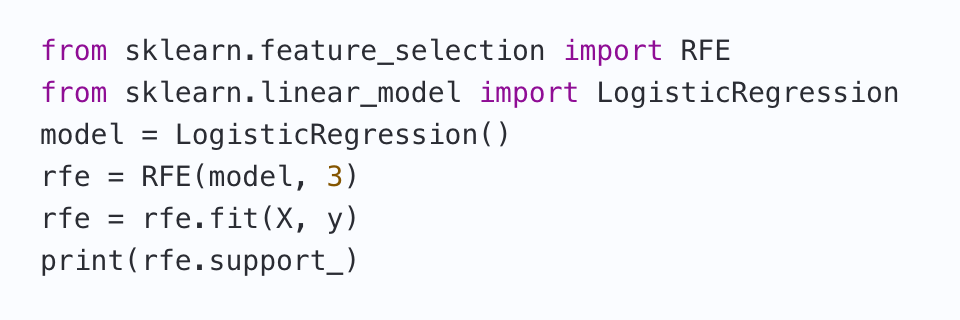


Рисунок 3 – Пример кода для отбора (в нашем случае трех) главных признаков.

Далее нам будет необходимо удалить из вектора X те переменные, которые оказались наименее важны для получения наиболее качественного результата:



Рисунок 4 – Пример кода для удаления наименее важных признаков.

Также общим моментом является разделение всего датасета на обучающую и целевую выборку. Эмпирическим путем было выяснено, что лучший результат в целом алгоритмы дают на обучающей выборке размером в 60 % от всего датасета, хотя может показаться, что это значение слишком мало.

Известно, что исходные данные представлены в виде csv-файла, в котором три вида растений занимают по 50 идущих подряд строк. Таким образом, код определения обучающей выборки может выглядеть следующим образом:

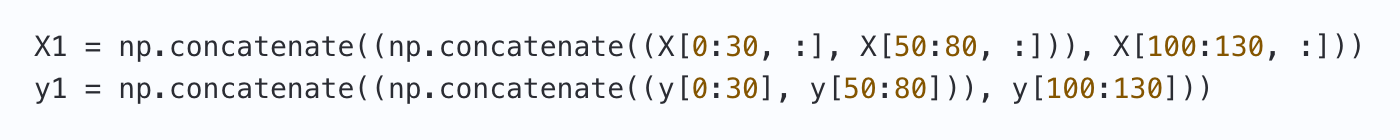


Рисунок 5 – Пример кода для определения обучающей выборки.

Теперь осталось лишь определить нашу контрольную выборку, которая будет включать в себя все остальные значения:

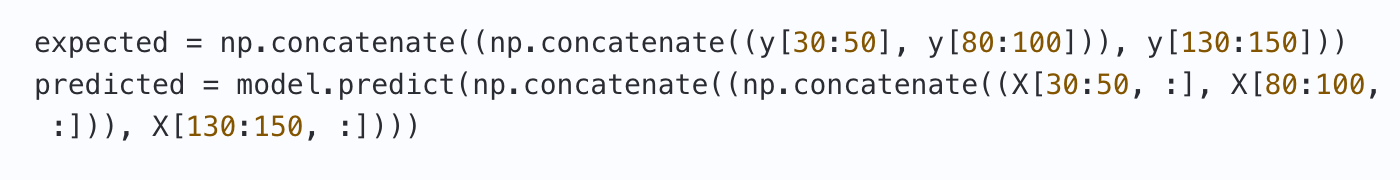


Рисунок 6 – Пример кода для определения контрольной выборки.

Наконец общими моментами покончено, так что можно переходить к самим методам анализа данных.

1. Метод решающих деревьев (дерево принятия решений).

Каждый лист представляет собой значение целевой переменной, изменённой в ходе движения от корня по листу. Каждый внутренний узел соответствует одной из входных переменных. Дерево может быть также «изучено» разделением исходных наборов переменных на подмножества, основанные на тестировании значений атрибутов. Это процесс, который повторяется на каждом из полученных подмножеств. Рекурсия завершается тогда, когда подмножество в узле имеет те же значения целевой переменной, таким образом, оно не добавляет ценности для предсказаний. Процесс, идущий «сверху вниз», индукция деревьев решений (TDIDT), является примером поглощающего «жадного» алгоритма, и на сегодняшний день является наиболее распространённой стратегией деревьев решений для данных, но это не единственная возможная стратегия. В интеллектуальном анализе данных, деревья решений могут быть использованы в качестве математических и вычислительных методов, чтобы помочь описать, классифицировать и обобщить набор данных, которые могут быть записаны следующим образом:

Зависимая переменная Y является целевой переменной, которую необходимо проанализировать, классифицировать и обобщить.

Вектор x состоит из входных переменных и т. д., которые используются для выполнения этой задачи.

Стоит сразу отметить, что данный пример является задачей многоклассовой классификации, что идеально подходит для метода решающих деревьев.

Дальше не будет ничего сложно, поскольку построение модели является уже заключительной частью анализа данных и средства языка Python оставляют программистам лишь небольшую часть работы.

В нашем случае необходимо лишь проделать 3 шага. Первый – указать, какой метод требуется использовать:



Рисунок 7 – Пример кода для определения модели DTC.

Второй шаг – обучить (натренировать) модель на отобранной ранее обучающей выборке:



Рисунок 8 – Пример кода для обучения DTC.

И последний шаг – сделать предсказания для некоторого набора данных, то есть для отобранной ранее целевой выборки и вывести статистику относительно прогноза:

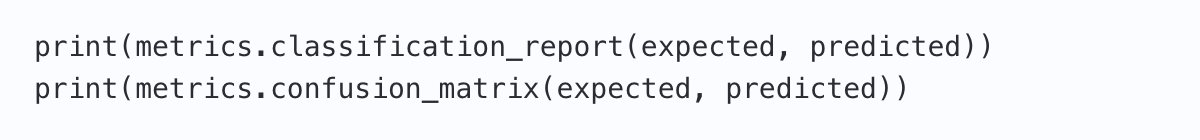


Рисунок 9 – Пример кода для осуществления прогноза метода DTC.

В результате работы данного метода, был получен некоторый прогноз, согласно которому были угаданы все 20 ирисов первого вида, один ирис второго вида был принят за ирис третьего вида и один ирис третьего вида был принят за ирис второго вида.

Исходя из этих результатов, можно сделать вывод, что алгоритм отлично справляется с поставленной задачей и выдает в среднем 97 % правильных предсказаний.