МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

МЕТОДЫ ПОСТРОЕНИЯ ДЕРЕВЬЕВ РЕШЕНИЙ И ИХ ПРИЛОЖЕНИЯ К АНАЛИЗУ ДАННЫХ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 38.03.05 — Бизнес-информатика

механико-математического факультета
Багникяна Михаила Петросовича

Научный руководитель
д. ф.-м. н., доцент _____ С. П. Сидоров

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2022

ВВЕДЕНИЕ

Актульность темы. В повседневной жизни мы часто сталкиваемся с различными данными, которые мы используем в разных целях. В основном это уже обработанные данные, из которых мы извлекаем необходимую нам информацию. Примерами могут быть различные социальные сети, сайты где есть статистическая информация, различные базы данных определенных компаний. Все эти данные необходимо обрабатывать, анализировать и передавать для решения задач другим лицам. В этих данных очень много полезной информации которую мы извлекаем, для дальнейшей работы с этими данными. Хранилища данных ресурсо-затратны, для работы с ними используются специальные программы. Также важно хранить эти данные в безопасном месте, чтобы предотвратить их потерю или повреждение.

Актуальность данной работы связяна с тем, что деревья решений являются очень эффективными и популярными инструментами работы с различными данными. Этот метод часто используется в анализе данных, для получения решения важных практических задач. Деревья решений будут и дальше применяться в анализе данных все больше, потому что данные растут быстрыми темпами и с ними необходимо эффективно работать, эту проблему частично решают деревья решений. Метод построения деревьев решений позволяет построить дерево где мы в графическом виде видим результаты ответов на различные вопросы на которые мы хотим ответить, анализируя наши данные. На основе построения дерева решений мы также можем предсказывать, значение целевой переменной на основе входных переменных.

Целью бакалаврской работы является изучение метода построения деревьев решений и их приложения к анализу данных с использованием языка программирования Python.

Объект исследования – методы построения деревьев решений.

Предмет исследования – применение методов построения деревьев решений к анализу реальных данных.

Для достижения поставленных целей в работе необходимо решить следующие **задачи**:

— определить основные понятия, связянные с работой с деревьями реше-

ний и анализом данных;
— изучить постановку задачи классификации и методы построения деревьев решений;
— рассмотреть основные понятия, связанные с построением деревьев решений;
— изучить необходимые библиотеки языка программирования Python для работы с данными;
— построить дерево решений с использованием языка Python;
— проанализировать реальные данные используя язык Python.

Практическая значимость проводимого исследования состоит в том, с помощью языков программирования можно анализировать множество реальных практических данных, из различных источников сбора информации, строить модели деревьев решений, находить скрытые зависимости в данных. В результате такого анализа и построения различных моделей и деревьев решений, можно извлекть из данных скрытые закономерности, которые можно применять при принятии решений, и применять результаты которые могут помочь в решении определенных проблем в различных сферах человеческой деятельности.

Структура и содержание бакалаврской работы. Работа состоит из введения, двух разделов, заключения, списка использованных источников, содержащего 12 наименований. Общий объем работы составляет 42 стр.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обосновывается значимость темы работы, формулируется цель работы и решаемые задачи, отмечается значимость метода дерева решений.

В **первом разделе** изучается задача классификации, деревья решений, и анализ данных. Основные понятия которые мы рассмотрели в первом разделе это:

1. Анализ данных
2. Постановка задачи классификации
3. Методы решения задачи классификации
4. Структура дерева решений

5. Регулирование и корректировка дерева решений
6. Области применения деревьев решений
7. Обучение дерева решений и применение
8. Интеллектуальный анализ данных и деревья решений

Кроме того, приводится постановка задачи классификации.

Пусть некто, для определенности будем говорить учитель, предъявляет ситуации и о каждой сообщает, к какому из k классов она относится. Для простоты будем считать, что $k = 2$, так как при любом другом числе классов последовательным разделением на два класса можно построить разделение и на k классов. Для этого достаточно провести k разделении по принципу: первое — отделяет элементы первого класса от всех остальных, а j-е — элементы j-го класса от всех остальных.

Будем считать, что входная ситуация описывается n-мерным вектором $x = (x_1, \ldots, x_n)$. Координаты этого вектора могут выражать те или иные характеристики объекта, например финансовые показатели предприятия, значения симптомов в задачах медицинской диагностики, значения параметров систем в технических задачах распознавания и т.д.

Последовательность ситуаций с указанием, к какому классу они относятся, называется обучающей последовательностью.

Задача заключается в том, чтобы построить такую программу, которая, используя обучающую последовательность, вырабатывала бы правило, позволяющее классифицировать вновь предъявляемые «незнакомые» ситуации (вообще говоря, отличные от входивших в обучающую последовательность).

Способность к обучению характеризуется двумя понятиями:

— качеством полученного решающего правила (вероятностью неправильных ответов — чем меньше эта вероятность, тем выше качество);

— надежностью получения решающего правила с заданным качеством (вероятностью получения заданного качества — чем выше эта вероятность, тем выше надежность успешного обучения).

Задача сводится к созданию такого обучающего устройства, которое по обучающей последовательности строило бы решающее правило, качество которого с заданной надежностью было бы не ниже требуемого.

Одним из методов решения задач классификации является деревья ре-

шений.

Дерево принятия решений (также может называться деревом классификации или регрессионным деревом) – средство поддержки принятия решений, в основном используется в машинном обучении, анализе данных и статистике. Структура дерева представляет собой «листья» и «ветки». На ребрах («ветках») дерева решения записаны атрибуты, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах – атрибуты, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение.

Подобные деревья решений широко используются в интеллектуальном анализе данных. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе. Дерево решений является одним из наиболее хороших инструментов для интеллектуального анализа данных и аналитики, эти методы могут решать различные задачи классификации и регрессии.

Данные деревья решений представляют собой иерархические древовидные структуры, которые состоят из решающих правил которые могут иметь вид «Если..., то ...» и т.д. Правила автоматически генерируются в процессе обучения на обучающем множестве, они формулируются почти на естественном языке например, если объем выпущенной продукции на предприятии состовляет 1000 т., то предприятие имеет хорошие показатели, деревья решений как модели аналитики более интерпретируемые, чем нейронные сети. Поскольку правила в деревьях решений получаются путём обобщения множества отдельных наблюдений (обучающих примеров), описывающих предметную область, то по аналогии с соответствующим методом логического вывода их называют индуктивными правилами, а сам процесс обучения — индукцией деревьев решений.

Рассмотрим алгоритм построения деревьев решений. Наша задача будет заключаться в построении иерархической классификационной модели в виде дерева из множества примеров T. Далее выделим основные требования к структуре данных и данным в деревьях решений. В основе большинства популярных алгоритмов обучения деревьев решений лежит принцип «разде-

ляй и властвуй». Алгоритмически этот принцип реализуется следующим образом. Пусть задано обучающее множество S, содержащее n примеров, для каждого из которых задана метка класса C_i, $i = 1, \ldots, k$, и m атрибутов A_i, $i = 1, \ldots, m$, которые, как предполагается, определяют принадлежность объекта к тому или иному классу. Тогда возможны три случая:

1. Все примеры множества S имеют одинаковую метку класса C_i все обучающие примеры относятся только к одному классу. Очевидно, что обучение в этом случае не имеет смысла, поскольку все примеры, предъявляемые модели, будут одного класса, который и «научится» распознавать модель. Само дерево решений в этом случае будет представлять собой лист, ассоциированный с классом C_i. Практическое использование такого дерева бессмысленно.
2. Множество S не содержит примеров оно является пустым множеством. Для него необходимо создать лист класс которого будет выбран из другого множества.
3. Множество S содержит обучающие примеры всех классов C_k. В этом случае необходимо разбить множество S на подмножества. Для этого выбираем один из атрибутов A_j множества S который содержит два или более уникальных значения $(a_1, a_2, ..., a_p)$ где p число уникальных значений. Затем множество S разбивается на p подмножеств $(S_1, S_2, ..., S_p)$, каждое из которых включает примеры, содержащие значение атрибута.

Описанная выше процедура лежит в основе многих современных алгоритмов построения деревьев решений. Очевидно, что при использовании данной методики, построение дерева решений будет происходить сверху вниз (от корневого узла к листьям). В настоящее время разработано значительное число алгоритмов обучения деревья решений: ID3, CART, C4.5, C5.0, NewId, ITrule, CHAID, CN2 и т.д. Но наибольшее распространение и популярность получили следующие:

1. ID3 (Iterative Dichotomizer 3) – алгоритм позволяет работать только с дискретной целевой переменной, поэтому деревья решений, построенные с помощью данного алгоритма, являются классифицирующими. Число потомков в узле дерева не ограничено. Не может работать с пропущенными данными.

2. Построение дерева решений алгоритмом C4.5 принципиально не отличается от его построения в ID3. На первом шаге имеется корень и ассоциированное с ним множество T, которое необходимо разбить на подмножества. Для этого необходимо выбрать один из атрибутов в качестве проверки. Выбранный атрибут A имеет n значений, что дает разбиение на n подмножеств. Далее создаются n потомков корня, каждому из которых поставлено в соответствие свое подмножество, полученное при разбиении T. Процедура выбора атрибута и разбиения по нему рекурсивно применяется ко всем n потомкам и останавливается в двух случаях После очередного ветвления в вершине оказываются примеры из одного класса (тогда она становится листом, а класс, которому принадлежат её примеры, будет решением листа). Вершина оказалась ассоциированной с пустым множеством (тогда она становится листом, а в качестве решения выбирается наиболее часто встречающийся класс у непосредственного предка этой вершины).
3. CART (Classification and Regression Tree) – алгоритм обучения деревьев решений, позволяющий использовать как дискретную, так и непрерывную целевую переменную, то есть решать как задачи классификации, так и регрессии. Алгоритм строит деревья, которые в каждом узле имеют только два потомка. Алгоритм CART предназначен для построения бинарного дерева решений. Бинарные деревья также называют двоичными, значит, что каждый узел дерева при разбиении имеет только двух потомков. Для алгоритма CART «поведение» объектов выделенной группы означает долю модального значения выходного признака. Выделенные группы – те, для которых эта доля достаточно высока. На каждом шаге построения дерева правило, формируемое в узле, делит заданное множество примеров на две части – часть, в которой выполняется правило (потомок – right) и часть, в которой правило не выполняется (потомок – left). Преимуществом алгоритма CART является определенная гарантия того, что если искомые детерминации существуют в исследуемой совокупности, то они будут выявлены. Кроме того, CART позволяет не «замыкаться» на единственном значении выходного признака, а искать все такие его значения, для которых можно найти

соответствующее объясняющее выражение.

Требования к таблице данных алгоритма C4.5:

1. Описание атрибутов. Данные для работы алгоритма, должны быть в виде плоской таблицы. Вся информация об объектах из предметной области должна описываться в виде конечного набора признаков. Каждый атрибут должен иметь дискретное или числовое значение. Сами атрибуты не должны меняться от примера к примеру.
2. Определенные классы. Каждый пример должен быть ассоциирован с конкретным классом, то есть один из атрибутов должен выбрать в качестве метки класса.
3. Дискретные классы. Классы должны быть дискретными, они должны иметь конечное число определенных значений. Каждый из которых должен относиться к конкретному классу. Но бывают случаи, когда примеры принадлежат к классу с вероятностными оценками, исключаются. Количество классов должно быть значительно меньше количества примеров.

Процесс построения дерева будет происходить сверху вниз. Сначала создается корень дерева, затем потомки корня и т.д. На первом шаге мы имеем пустое дерево (имеется только корень) и исходное множество T. Требуется разбить исходное множество на подмножества. Это можно сделать, выбрав один из атрибутов в качестве проверки. Тогда в результате разбиения получаются n (по числу значений атрибута) подмножеств и, соответственно, создаются n потомков корня, каждому из которых поставлено в соответствие свое подмножество, полученное при разбиении множества T.

Затем эта процедура рекурсивно применяется ко всем подмножествам (потомкам корня) и т.д.

Рассмотрим критерий выбора атрибута. Пусть мы имеем проверку X (в качестве проверки может быть выбран любой атрибут), которая принимает n значений $A_1, \ldots, A_n$. Тогда разбиение T по проверке X даст нам подмножества $T_1, \ldots, T_n$ при X, равном соответственно $A_1, \ldots, A_n$. Единственная доступная нам информация – то, каким образом классы распределены в множестве T и его подмножествах, получаемых при разбиении по X. Именно этим мы и воспользуемся при определении критерия.

Пусть $freq(C_j, S)$ — количество примеров из некоторого множества S, относящихся к одному и тому же классу C_j. Тогда вероятность того, что случайно выбранный пример из множества S будет принадлежать к классу C_j , будет равна

$$P = \frac{f_{req}(C_j, S)}{|S|}$$

. Теория информации гласит, что количество информации в сообщении зависит от ее вероятности следующей зависимостью: $log_2\frac{1}{p}$. Данное выражение дает оценку в битах. Выражение

$$I(T) = -\sum_{j=1}^{k} \frac{f_{req}(C_j, T)}{|T|} log_2(\frac{f_{req}(C_j, T)}{|T|})$$

дает оценку среднего количества информации, необходимого для определения класса примера из множества T. В теории информации выражение называется энтропией множества T.

Ту же оценку, но только уже после разбиения множества T по X, дает следующее выражение

$$I_x(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} I(T_i)$$

. Тогда критерием для выбора атрибута будет являться следующая формула: $G(x) = I(T) - I_x(T)$. Критерий считается для всех атрибутов. Выбирается атрибут, максимизирующий данное выражение. Этот атрибут будет являться проверкой в текущем узле дерева, а затем по этому атрибуту производится дальнейшее построение дерева. То есть в узле будет проверяться значение по этому атрибуту и дальнейшее движение по дереву будет производиться в зависимости от полученного ответа. Рассмотрим процесс построения деревьев решений более подробно, и последовательно. Процесс построения деревьев решений состоит в последовательном, и рекурсивном разбиении обучающего множества на подмножества и здесь применяют решающие правила в узлах. Этот процесс разбиения продолжается до тех пор, пока все узлы в конце всех ветвей не будут объявлены листами. Объявление узла листом может получится естественным образом но при следующих условиях (когда он будет содержать один объект, или объекты только одного класса), или по достижении

условия остановки, который задает сам пользователь например, минимальное допустимое число примеров в узле или максимальная глубина дерева. Алгоритмы построения деревьев решений относят к категории так называемых жадных алгоритмов.

Для анализа данных мы использовали язык программирования Python.

В данной главе мы будем работать с языком Python, который будем использовать для анализа данных и построения деревьев решений. Стоит отметить что язык Python хорошо подходит для анализа данных. Сам язык Python появился в 1991 году, и с того момента он стал одним из самых популярных языков программирования. Python также отличный выбор для создания веб-сайтов, на этом языке можно быстро написать небольшую программу. В машинном обучении также очень часто используют язык Python, можно сказать что данный язык универсальный.

Мы подготовили и использовали реальные данные, где нам необходимо построить различные модели.

Мы будем анализировать данные и строить различные модели, данные будут реальными. В нашем случае мы будем предсказывать безопасность автомобиля. Строить мы будем 2 модели, одну с использованием индекса Джини, другая основана на вычислении энтропии. Алгоритм дерева решений, который мы будем использовать – это один из самых лучших алгоритмов для машинного обучения и анализа данных. Данный алгоритм использует древовидную структуру, где есть различные возможные комбинации, для решения конкретной задачи. Этот алгоритм относиться к классу контролируемого обучения, его можно использовать как для классификации, так и для регрессии. В настоящее время алгоритм дерева решений имеет название CART, которе расшифровывается как деревья классификации и регрессии. Для того чтобы понять концепцию получения информации, использовали энтропию. Энтропия измеряет примесь в данном наборе данных. Также мы использовали информационный выигрыш (это уменьшение энтропии). Информационный выигрыш используется для вычисления разницы между энтропией до разделения и средней энтропией после разделения набора данных на основе значений атрибутов.

Мы построили 2 модели дерева решений, одну с критериальным индек-

сом Джини и дерево решений с критериальной энтропией.

В заключении приведены результаты бакалаврской работы.

Основные результаты

1. Рассмотрены постановка задачи классификации и методы решения задач классификации.
2. Приведено определение дерева решений и методы построения деревьев решений.
3. Рассмотрены приложения, связанные с построением деревьев решений в анализе данных, с использованием языка Python.
4. Приведен пример построения дерева решений для реальных данных в задаче анализа на безопасность автомобилей.