РОЗДІЛ 8
ДЕРЕВА РІШЕНЬ

Зміст

8.1 Вступ

Дерева рішень як клас методів data-mining має свої коріння в таких досить традиційних статистичних дисциплінах як лінійна регресія. Окрім цього можна знайти схожість з когнітивним напрямком науки, що відомий під назвою нейронні мережі.

Метод дерев рішень (decision trees або answer tree) – одним із самих популярних методів розв'язання завдань класифікації та прогнозування. Іноді цей метод ІАД називають також деревами вирішальних правил, деревами класифікації та регресії.

Якщо залежна, тобто цільова змінна приймає дискретні значення, то за допомогою методу дерева рішень вирішується **задача класифікації.**

Якщо ж залежна змінна приймає безперервні значення, то дерево рішень установлює залежність цієї змінної від незалежних змінних, тобто вирішується задача **чисельного прогнозування.**

Уперше дерева рішень були запропоновані Ховілендом (Hoveland) та Хантом (Hunt) наприкінці 50-х років минулого століття. Самою ранньою роботою у якій викладається суть дерев рішень являється робота Ханта "Експерименти в індукції" ("Experiments in Induction") [209, 210], яка була опублікована у 1966 році.

Дерева рішень, як метод data-mining, має багато корисних здібностей, що використовуються в різноманітних прикладних сферах для бізнесу, а не тільки для науки. Ось деякі корисні здібності дерев рішень:

- Надають візуальний максимально зрозумілий результат. Дерева рішень досить просто будувати, розуміти отриманий результат та використовувати. Здатність представити багато пояснюючих факторів процесу в простій формі step-by-step. Корисна здібність дерев рішень – будувати по ітераціям правила високої складності.
- Здібність працювати як з кількісними так і якісними (наприклад, коли цільова зміна приймає тальки два значення - good або bad) даними.

Кількісні дані включають ординарні (наприклад, high, medium, low категорії) та інтервальні (наприклад, температура пацієнта, розмір доходу родини, вага тіла) рівні змінних.

- Легко працюють з даними, що мають ефекти – незбалансованості, гніздові ефекти, взаємного перекриття, пересічення змінних та не лінійності. З цими ефектами прості одно факторні та багатофакторні статистичні підходи не працюють.
- Дерева рішень характеризуються непараметричністю, високим рівнем робастності (наприклад, легко працюють з пропусками в даних) та будують дуже близькі структури в незалежності від рівня виміру змінних (наприклад, дерево рішень дасть близький результат для випадку, коли дохід вимірюється в десятках, сотнях, тисячах, та навіть дискретних станах від 1 до 5).

Одним із перших випадків використання дерев рішень на практиці відбувся у 1956 році Белсоном (Belson) для аналізу телетрансляції [211]. На сьогодення дерева рішень досить широко використовуються як в технічних так і соціальних дисциплінах, наприклад, в маркетингу, торгівлі, контролі якості. Головною метою цього розділу є підвищення рівня розуміння дерев рішень як математичного інструменту при практичному використанні.

8.2 Рішення, рейтинги, оцінки

В загальному випадку навчальні дані використовуються для створення моделі або правила, які пов'язують вхідні змінні з цільовою. Прогнози класифікуються за трьома типами – рішення, рейтинги, оцінки.

Прогнози-рішення

В літературі замість терміна "прогноз-рішення" більш поширена назва "класифікація". Зазвичай будь яке рішення пов'язане з якоюсь дією.

Наприклад, класифікувати клієнта як хорошого або поганого. Також класифікація використовується в задачах розпізнавання почерку, виявлення шахрайства, проведення рекламних акцій за допомогою прямих розсилок.

Прогнози-рішення зазвичай відносять до категоріальної цільової змінної. Тому їх ідентифікують як первинне, вторинне та третинне рішення.

Зазвичай, коли цільова змінна має категоріальний рівень виміру (бінарний, номінальний, порядковий), то оцінка моделі припускає прогнози-рішення.

Таблиця 8.1

Типи прогнозів-рішень

Спостереження	**Вхідні характеристики клієнта**	**Рішення**
1	Стать = М Дохід = 1100	Хороший
2	Стать = Ж Дохід = 500	Поганий
3	Стать = Ж Дохід = 600	Хороший
4	Стать = М Дохід = 300	Поганий

Прогнози-рейтинги

Впорядковують спостереження на основі зв'язків вхідних змінних з цільовою. Модель намагається впорядковувати спостереження з більш високими значеннями вище спостережень з більш низькими значеннями. Спостереження з більш високими значеннями мають великі скорінгові бали. Фактично, обчислені бали несуттєві, важливий тільки відносний порядок. Приклад прогнозу-рейтингу – кредитний скорінг.

Прогнози-рейтинги можуть бути перетворені в прогнози рішення, якщо прийняти первинне рішення для спостереження вище певного значення порогу, а вторинне і третинне рішення – для спостережень нижче відповідних значень порогу.

Таблиця 8.2

Приклад прогнозів-рейтингів

Спостереження	**Вхідні характеристики клієнта**	**Рейтинги**
1	Стать = М Дохід = 1100	720
2	Стать = Ж Дохід = 500	520
3	Стать = Ж Дохід = 600	620
4	Стать = М Дохід = 300	470

В кредитного скорінгу спостереження з балом вище 700 можна охарактеризувати як невеликий ризик, між 600 і 700 – проміжні ризики, а спостереження з балом менше 600 – погані ризики.

Прогнози-оцінки

Виконують апроксимацію очікуваного значення цільової змінної, в залежності від вхідних значень. Для спостережень з чисельними цільовими змінними це число можна охарактеризувати як середнє значення цільової змінної по всім спостереженням, що мають поточні вхідні виміри. Для спостережень з категоріальними цільовими змінними це число може дорівнювати імовірності конкретного результату цільової змінної.

Таблиця 8.3

Приклад прогнозів-оцінок

Спостереження	**Вхідні характеристики клієнта**	**Оцінки**
1	Стать = М Дохід = 1100	0,65
2	Стать = Ж Дохід = 500	0,33
3	Стать = Ж Дохід = 600	0,54
4	Стать = М Дохід = 300	0,28

Прогнози-оцінки можуть бути перетворені як у прогнози-рішення, так і в прогнози рейтинги. Рішення можна оцінити по точності або проценту помилкової класифікації, рейтинги по узгодженості або не узгодженості, а оцінки по середньоквадратичній помилці.

8.3 Поняття дерев рішень

В основі дерев рішень знаходяться алгоритми, що дозволяють розділити багатьма можливими способами набір даних на сегменти, кожен з яких буде містити максимально схожі елементи. Приклад простого дерева рішень наведений на рис. 8.1, як можна побачити відбувається розбиття на дві частини – праву і ліву, відносно критерію розбиття, що за соєю суттю представляє правило вигляду IF-THEN. Набір таких правил призводить до кінцевого листа, що відображає значення ймовірності прийняття того чи іншого рішення – Target = X% або Target = Y%.

Для створення правил використовуються змінні (в наборі даних змінна – це стовбець). На кожному рівні обирається змінна відносно якої обчислюється точка для розбиття. На рис. 8.2 наведені рівні дерева рішень. Найнижчі вузли (рівень 2) називаються листами дерева або термінальними вершинами. Кожному листу дерева рішень відповідає унікальний шлях до кореня.

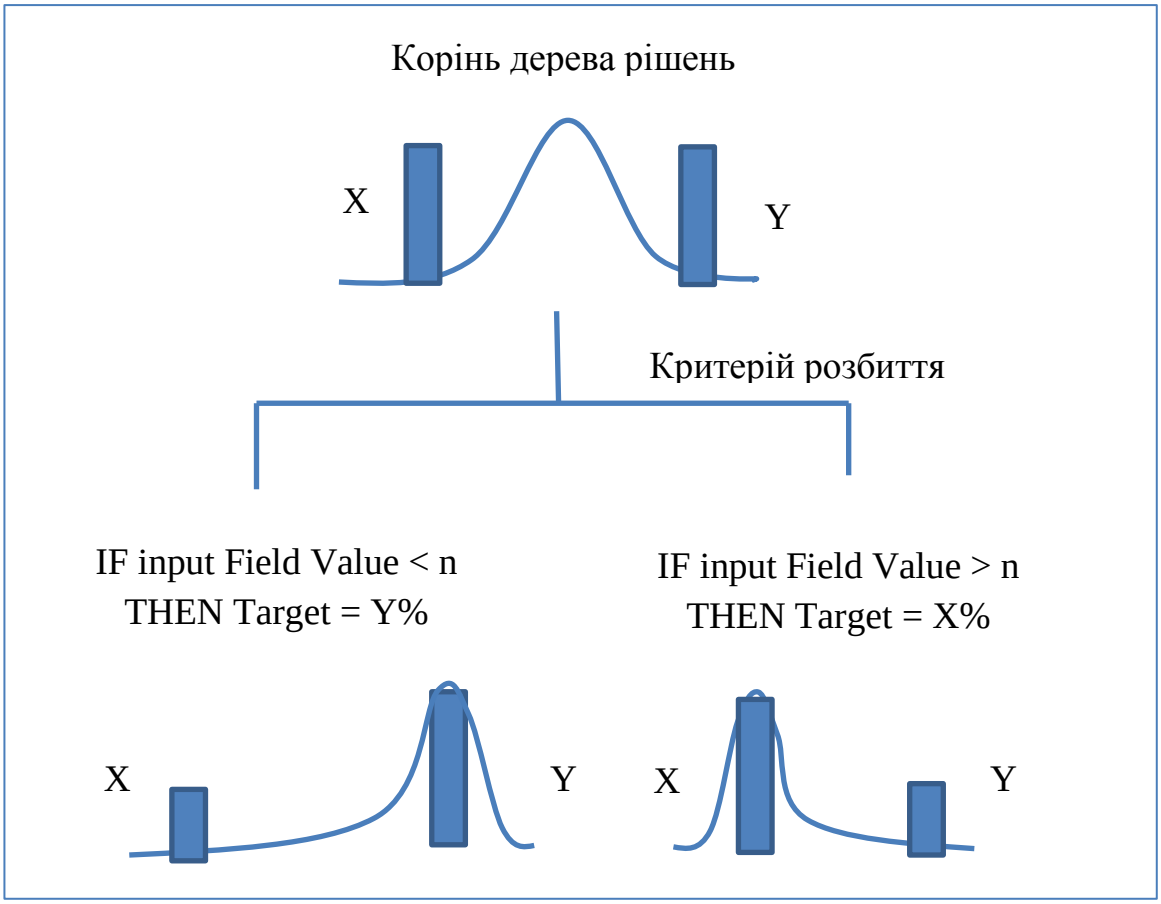

Рис. 8.1. Приклад дерева рішень

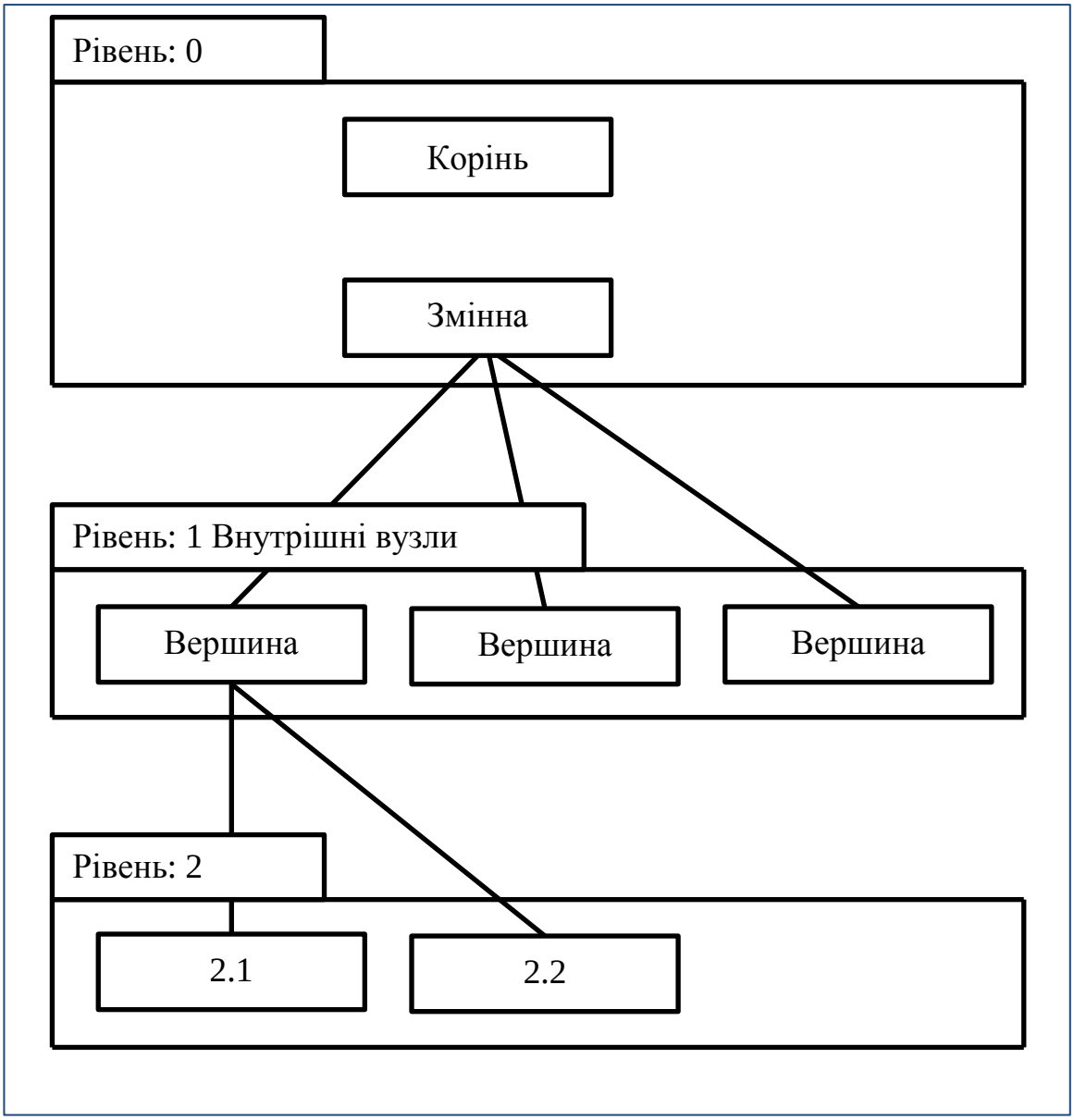

Рис. 8.2. Рівні дерева рішень

У найбільш простому виді дерево рішень – це спосіб подання правил в ієрархічній, послідовній структурі. Основа такої структури – відповіді “Так” або “Ні” на ряд запитань.

На рис. 8.3 наведені приклади дерев рішень, задачею яких є дати відповідь на питання: “Чи видавати кредит?” Для того щоб вирішити задачу, необхідно визначити ймовірність дефолту клієнта банку. Для цього необхідно відповісти на ряд запитань, що знаходяться у вузлах (вершинах) цього дерева, починаючи з його кореня.

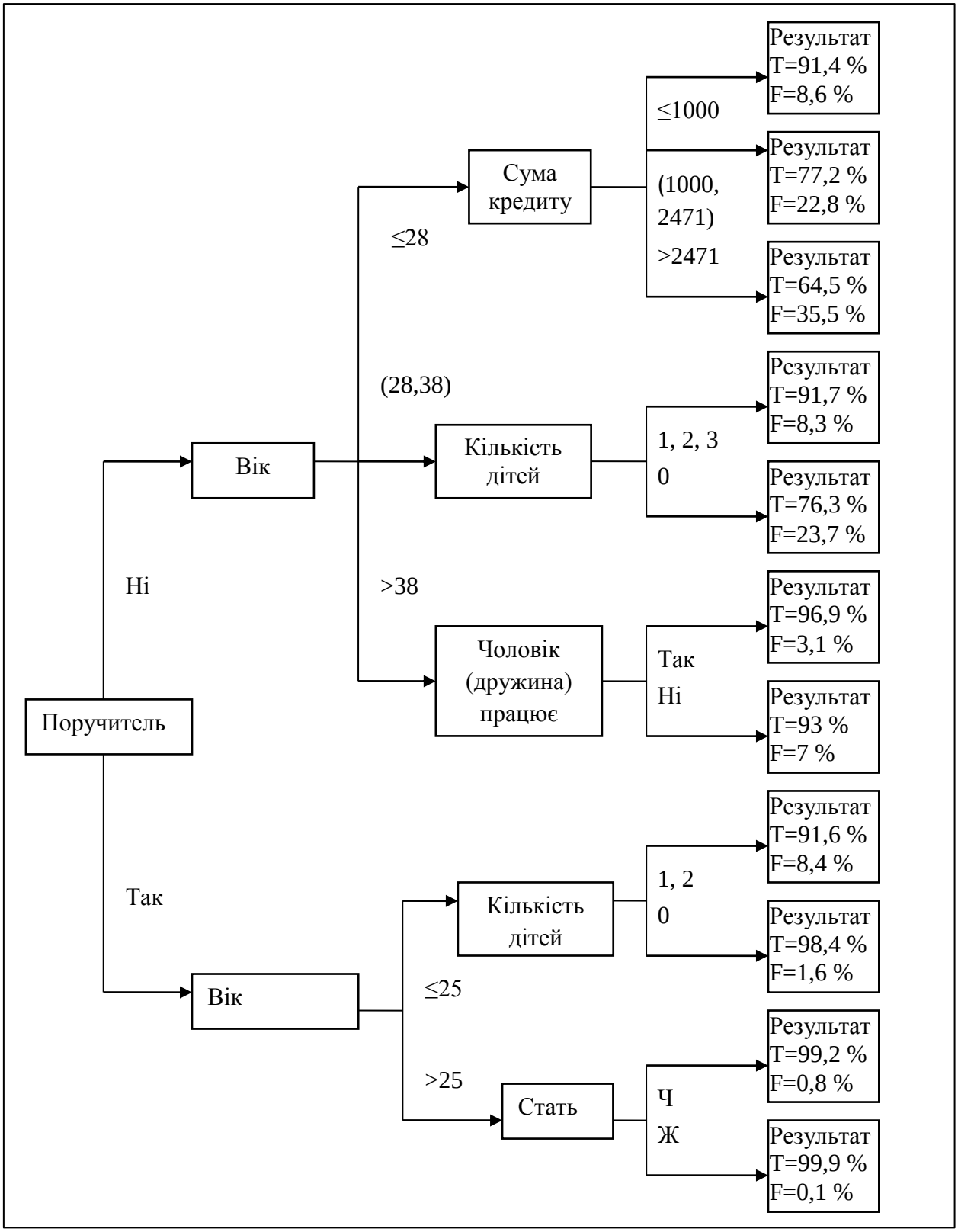

Рис. 8.3 Система кредитного скорингу у вигляді дерева рішень

Як приклад, розглянемо дерево рішень, наведене на рис 8.3. Змінна “Поручитель” – вершина перевірки, тобто умова. Якщо відповідь позитивна –

“Так”, то здійснюється перехід до верхньої частини дерева до вершини “Вік”, при негативній – до нижньої частини дерева. Таким чином, внутрішній вузол (вершина) дерева – це вузол перевірки певної умови. Далі йде наступне питання, і так далі до тих пір, поки не буде досягнуто кінцевого вузла (вершини) дерева, який представляє вузол рішення. Для нашого дерева кінцевим вузлом є вершина “Результат”, що приймає два значення: “T” та “F” (T = true та F = false), де “F” – ймовірність неповернення клієнтом кредиту виданого банком (дефолту).

В результаті проходження від кореня дерева “Поручитель” до його кінцевої вершини “Результат” розв’язується задача класифікації. На основі отриманих значень ймовірностей “T” та “F” визначається ступінь кредитоспроможності клієнта і приймається рішення щодо видачі або відмови у наданні кредиту.

8.4 Використання дерев рішень разом з іншими методами моделювання

Дерева рішень можуть використовуватися разом з іншими методами моделювання, наприклад, регресіними для вибору вхідних змінних аналізу та створення фіктивних змінних.

Розглянемо приклад побудови стратифікаційної регресії, як показано на рис. 8.4 різні набори даних (страти) можуть відповідати різним регресійним рівнянням. Саме дерева рішень як найкраще підходять для визначення страт.

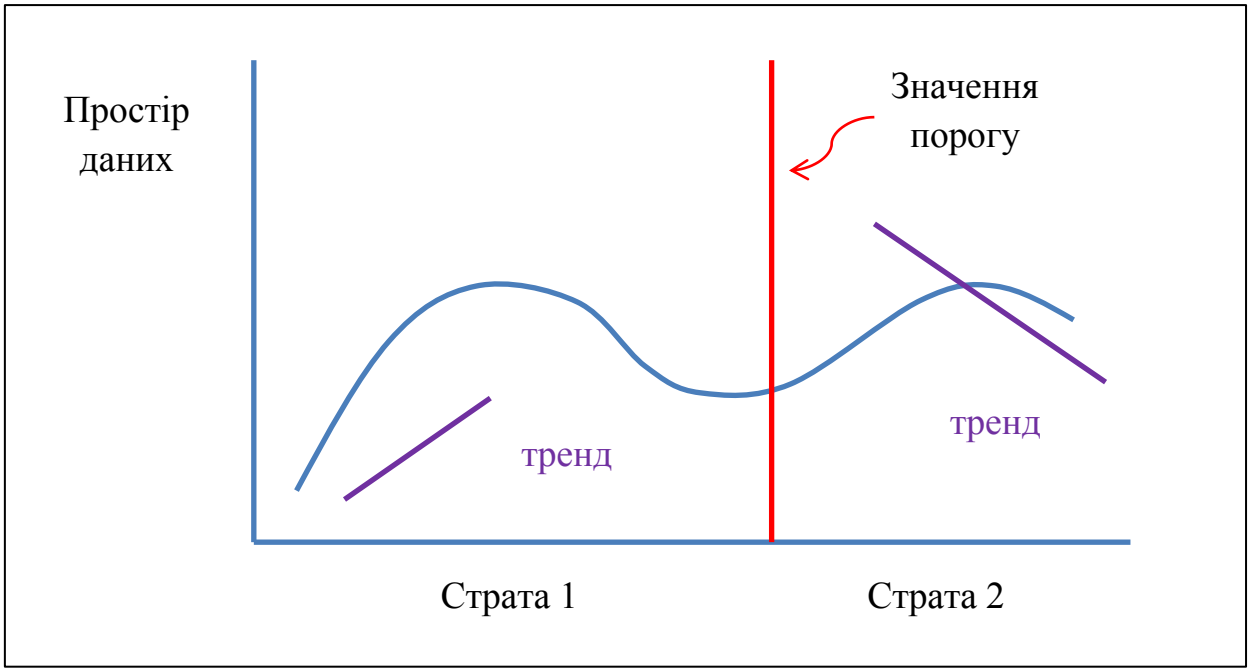

Рис. 8.4 Стратифікаційна регресія

Окрім наведеного прикладу побудови стратифікаційної регресії, дерева рішень також можуть використовуватися для зменшення рівнів категоріальних змінних. Зазвичай цю процедуру називають – оптимальне стиснення значень (optimal collapsing of values). Типовим підходом являється об'єднання суміжних категорій разом. На рис. 8.5 наведений приклад, коли в наявності є змінна, що складається з 10 категоріальних значень. Рис. 8.6 демонструє варіант стиснене до 5 категорій:

- 1 і 2 – асоційовані з екстремально низькими і екстремально великими значеннями цільової змінної
- 3, 4, 5 і 6
- 7 і 8
- 9
- 10

Рівень										
		X								
			X							
				X				X		
					X	X	X			
	X								X	X
Значення										
	1	2	3	4	5	6	7	8	9	10

Рис. 8.5 Початкові 10 категорій

Рівень										
		X								
			X							
				X				X		
					X	X	X			
	X								X	X
Значення										
	1	2	3	4	5	6	7	8	9	10

Рис. 8.6 Стиснення до 5 категорій

На рис. 8.7 наведений приклад стиснення до 4 категорій:

- 1
- 2, 3 і 4 – асоційовані з екстремально великими значеннями цільової змінної
- 5, 6 і 7 – асоційовані з екстремально низькими значеннями цільової змінної
- 8, 9 і 10

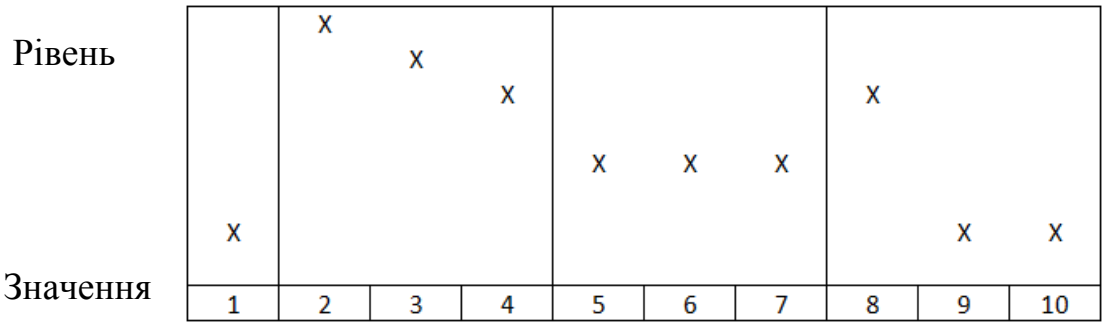

Рис. 8.7 Стиснення до 4 категорій

На рис. 8.5-8.7 розглядалися випадки ординарних значень, але якщо змінна номінального типу, то найбільш оптимальним буде розбиття, що наведено на рис. 8.8, з трьох категорій:

- 1, 5, 6, 7, 9 і 10
- 3, 4, 8
- 2

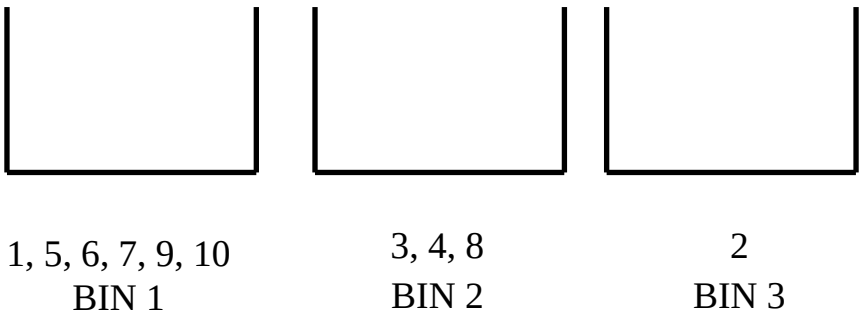

Рис. 8.8 Стиснення до 3 категорій

Як показано на рис. 8.5-8.8 можливі різноманітні варіанти стиснення категоріальних значень, інтуїтивно зрозуміло, що кожна ново сформована стисненням категорія може бути використана як частина дерева рішень. Наприклад, BIN 3 з рис. 8.8 може виступати в ролі окремого листа дерева рішень, що буде прогнозувати екстремально великі значення цільової змінної.

Дерева рішень намагаються знайти сильний зв'язок між вхідними параметрами та цільовою змінною. Коли множина вхідних змінних ідентифікується як та що має сильний зв'язок з цільовою, всі значення групуються в групи, що утворюють частини дерева.

Дерева рішень здатні працювати як з кількісними так і якісними змінними. Кількісні змінні, наприклад, зріст та вага, можуть використовуватися в таких арифметичних операціях як додавання, віднімання, ділення та множення. По відношенню до кількісних змінних, наприклад, стать та колір, арифметичні операції не можуть бути застосовані. Деякі змінні можуть мати характеристики як кількісної, так і якісної змінної, наприклад, розмір взуття, 44 розмір більший за 43, але він не вдвічі більший за 22.

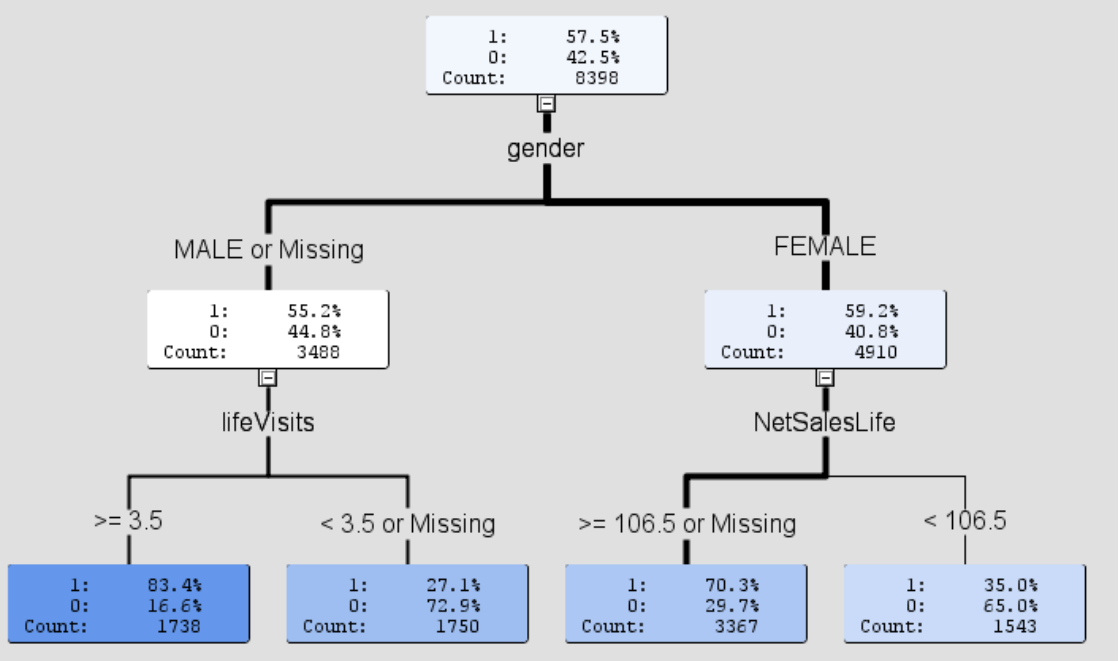

Рис. 8.9 Приклад дерева рішень з категоріальною цільовою змінною

На рис. 8.9 наведений приклад дерева рішень в системі SAS Enterprise miner, коли цільова змінна бінарного типу, тобто приймає два значення: 1 – клієнт купив товар, 0 – не купив. По інформації з кореневої вершини видно,

що навчальна вибірка складається з 8398 клієнтів, причому 57,5% з них купили товар, а 42,5% ні.

Під кореневою вершиною знаходиться вершина Gender (стать), по якій відбулося розбиття, як видно жінки (59,2%) на 4% частіше у порівнянні з чоловіками (55,2%) купують товар.

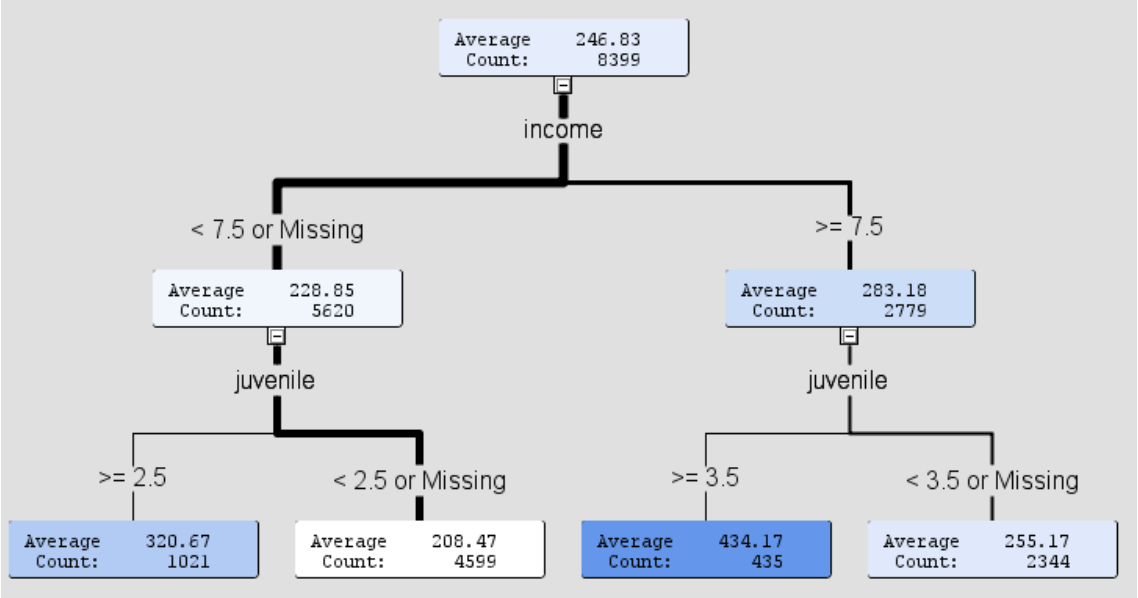

Рис. 8.10 Приклад дерева рішень з інтервальною цільовою змінною

На рис. 8.10 наведений приклад дерева рішень в системі SAS Enterprise miner, коли прогнозуються інтервальна цільова змінна – вартість покупки. Як можна побачити з інформації, що наведена у кореневій вершині, для побудови дерева рішень використовується 8399 записів в базі даних про клієнтів, при чому середня вартість покупки 246,83 $.

Кожен лист дерева може бути описаний правилами IF-THEN, наприклад:

if juvenile >= 3.5 AND income >= 7.5

then

Number of Observations = 435

Predicted: NetSales = 434.16

8.5 Тести значимості при побудові дерев рішень

CHi-squared Automatic Interaction Detector). Це найбільш відомий метод побудови дерев рішень, згідно з яким для одержання оптимальної розбивки використається критерій зв'язку між категоріальними змінними χ^2 (у випадку, якщо цільова змінна є кількісною, використається F – критерій). Дані для аналізу цільової змінної та змінні-фактори можуть бути як кількісними, так і категоріальними, однак кількісні змінні-фактори при побудові дерева перетворюються в категоріальні.

При побудові дерев рішень методом CHAID використовуються статистичні тести з наступних причин:

- Статистичний тест надає доказ того, що виявлений зв'язок значимо відрізняється від будь якого іншого обраного випадковим чином.
- В залежності від рівня значимості проведеного тесту для відповідної змінної визначається сама значима змінна яка використовується для розбиття.
- Побудова дерева рішень припиняється як тільки процедура перевірки статистичної значимості видає незадовільний результат.

На практиці в статистиці використовуються наступні рівні значимості наведені в табл. 8.4.

Таблиця 8.4

Рівні значимості p-value для chi-square при побудові дерева рішень

Рівень значимості	**Опис**
0,001	Екстремально сильний
0,01	Сильний
0,05	Достатньо сильний
0.1	Недостатньо сильний
0,15	Екстремально слабкий

На рис. 8.11 наведені рівні значимості, як видно чим краще точка розбиття відповідної змінно робить розбиття на різні сегменти тим далі один від одного знаходяться графіки розподілів.

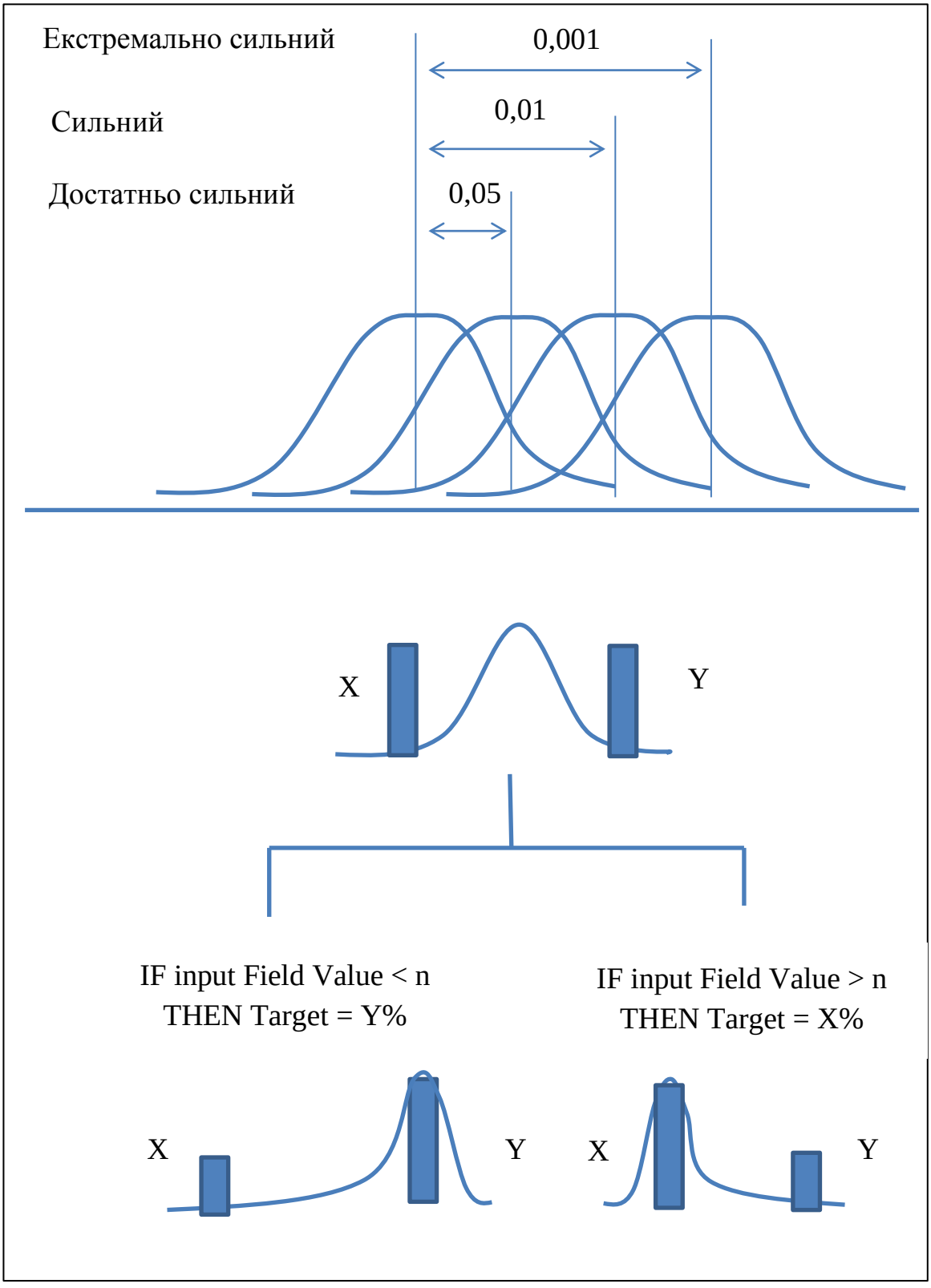

Рис. 8.11 Рівні значимості

8.6 Пошук точки розділення при побудови дерев рішень

Пошук розділення починається з вибору вхідної змінної для розбиття наявних навчальних даних. Якщо шкала виміру обраної вхідної змінної **категоріальна**, то кожне унікальне значення використовується в якості потенційної точки розділення даних. Якщо вхідна змінна інтервальна, то розглядається середнє значення цільової змінної в кожному категоріальному рівні вхідної змінної. Середні значення мають таку ж саме роль що і унікальні значення категоріальних змінних. Для обраної вхідної змінної та фіксованої точки розділу генеруються дві групи.

Спостереження з вхідними значеннями, меншими точки розділення, мають назву – **лівої гілки**. Спостереження з вхідними даними, більшими точки розділення, мають назву – **правої гілки.**

Велика різниці в пропорціях значень цільової змінної вказує на якісне розділення.

В загальному випадку статистика Пірсона може біти використана для випадку багатомірних розділень і цільових змінних з багатьма значеннями (не тільки бінарні), то статистика перетворюється в значення ймовірності або p-значення.

Насамперед p-значення вказує на ймовірність отримання значення статистики, що спостерігається, при допущенні ідентичних пропорцій цільових змінних в кожному з напрямів гілок. Для великих наборів даних p-значення може бути дуже близьким до нуля, саме тому якість розділення описується значенням

$$logworth = -\log_{10}(chi - square\ for\ p - value)$$

Як наслідок хоча б одне значення logworth повинно бути більшим за значення порогу, для того щоб по даній вхідній змінній було виконане розділення. В SAS Enterprise Miner в стандартних налаштуваннях системи

значення порогу відповідає chi-square p-value 0,20 або приблизно значенню logworth=0,7.

$$logworth = -\log_{10}(0{,}2) = 0{,}7$$

8.7 Поправки Бонферроні, Касса та за глибиною при побудови дерев рішень

Найкраще розділення для вхідної змінної – це розділення, що має найбільше значення logworth. Існує декілька другорозрядних факторів, що роблять пошук розділень складними.

По-перше, налаштування алгоритму побудови дерев дозволяють робити відповідне розбиття даних. Налаштування, такі як мінімальна кількість спостережень, що необхідне для пошуку розділень, і мінімальна кількість спостережень в листі примусово задають кількість спостережень в частині, що отримана, примусово задають кількість спостережень в частині, отримані після розділення. Це мінімальна кількість спостережень знижує кількість потенційних точок розбиття для кожної вхідної змінної при пошуку розділень.

По-друге, коли перевіряють на незалежність стовбці в таблиці спряженості, можна отримати значно більші значення хі-квадрат статистики, навіть при відсутності різниці в пропорціях результатів цільової змінної між гілками розділення. По мірі збільшення кількості можливих точок розбиття, ймовірність отримання великих значень також збільшується. Так вхідна змінна зі множиною унікальних значень має більшу ймовірність випадково привести до великого значення logworth, чим вхідна змінна всього з декількома різноманітними значеннями.

Статистики зустрічають схожі проблеми, коли об’єднують результати багатьох статистичних тестів. По мірі збільшення кількості, ймовірність невірного позитивного результату також зростає. Щоб підтримати загальну ймовірність в статистичних результатах, різко збільшують p-значення

кожного тесту в стільки разів, скільки тестів було зроблено. Якщо збільшене p-значення показує значимий результат, то значимість загальних результатів гарантована. Цей тип корегування p-значення відомий як поправка Бонферроні.

Так кожна точка розділення відповідає статистичному тесту, поправки Бонферроні автоматично використовуються до обчислень значень logworth для вхідної змінної. Ці поправки мають назву поправки Касса, на честь засновника алгоритму побудови дерев рішень для системи SAS Enterprise Miner, "штрафують" вхідні змінні з багатьма точками розділення зменшуючи значення logworth розділення на число, що дорівнює логарифму кількості різноманітних вхідних значень. Це еквівалентно поправці Банферроні, тому що віднімання цієї константи із значень logworth еквівалентно множенню відповідного хі-квадрат p-значенню на кількість точок розділення.

Ця поправка дає більш чітке порівняння вхідних змінних з більшою та малою кількістю рівнів в алгоритмі пошуку розділень.

$$logworth - \log(N) = -\log(p - value) - \log(N) = -\log(p - value \times N)$$

По-третіх. Для вхідних даних з пропущеними значеннями фактично генеруються два набори значень logworth, що скоректовані по Кассу. Для першого набору спостереження, що мають пропущені значення вхідних змінних, включають в ліву гілку таблиці спряженості, і після цього обчислюється значення logworth. Для другого набору значень logworth, спостереження з пропусками переносяться до правої гілки. Після цього обирається найкраще розділення і набору можливих розділень в лівій та правій гілках відповідно.

Процес розбиття повторюється для кожної вхідної змінної в навчальних даних. Вхідні змінні чиї підправлені значення logworth не більше значення порогу виключаються з аналізу.

Після того як знайдено найкраще розділення для кожної вхідної змінної, алгоритм порівнює усі відповідні значення logworth кращих

розділень. Розділення з найбільшим підправленим logworth вважається кращим.

Так як значимість другого та подальших розділень залежить від значимості попередніх розділень, алгоритм знову зустрічається з проблемою множинного порівняння. Для виправлення наслідків цієї проблеми алгоритм збільшує значення порогу на число, що залежить від кількості розділів, що були здійсненні раніше, вище поточного рівня розділення. Для двійкових розділень значення порогу збільшується на

$$\log_{10}(2) = 0{,}3 \times d$$

де d – це глибина розділення в дереві рішень.

Дані діляться згідно кращому розділенню, що створює друге правило розбиття. Процес повторюється в кожному листі до тих пір, доки не залишиться допустимих розділень, чиї підправлені значення logworth перебільшують значення порогу, що скореговані по глибині. Даний процес закінчує пошук розділень алгоритму побудови дерев.

Розбиття простору вхідних даних змінних має назву – максимального дерева. Вирощування максимального дерева заснований виключно на статистичних вимірах розділення навчальних даних. Максимальне дерево, як правило не спроможне надати якісний ступінь узагальнення на незалежному наборі перевірочних даних.

8.8 Приклад використання функції logworth при побудови дерев рішень

Розглянемо випадок, коли є тільки дві вхідні змінні процесу – координати точки x_1 та x_2 в квадраті розміром 1 на 1, дивись рис. 8.12, та бінарна цільова змінна, значення якої треба прогнозувати – колір точки, синій або жовтий.

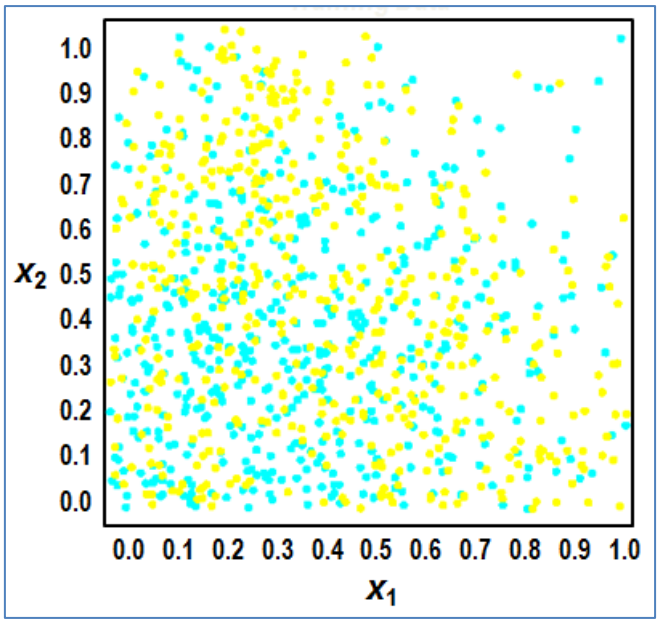

Рис. 8.12 Приклад прогнозування кольору точки

8.8.1 Алгоритм пошуку розділення, на основі знаходження максимального значення logworth

Крок 1. Для змінної x_1 здійснюється перебір всіх значень з кроком 0,01

Крок 2. Для кожного x_1 будується матриця контингенції.

На рис. 8.13 наведений приклад побудованої матриці контингенції для значення $x_1 = 0{,}52$. Ця матриця містить інформацію про кількість точок відповідного кольору в лівій та правій частинах, для прикладу з рис. 8.12.

Рис. 8.13 Матриці контингенції для $x_1 = 0{,}52$

Крок 3. Обчислюється значення функції logworth за формулою:

$$logworth = -\log_{10}(chi - square\ for\ p - value)$$

Чисельний приклад. В якості більш наочного прикладу розглянемо випадок, коли на площині розміщені 10 точок – п’ять кругів (сині точки) та п’ять хрестиків (червоні точки). В табл. 8.5 наведені координати.

Таблиця 8.5

Координати точок на площині

x1	x2	Target	Color
0,1	0,1	o	Blue
0,1	0,9	x	Red
0,2	0,3	o	Blue
0,2	0,6	o	Blue
0,3	0,8	o	Blue
0,6	0,1	x	Red
0,6	0,9	x	Red
0,8	0,1	x	Red
0,8	0,5	o	Blue
0,8	0,8	x	Red

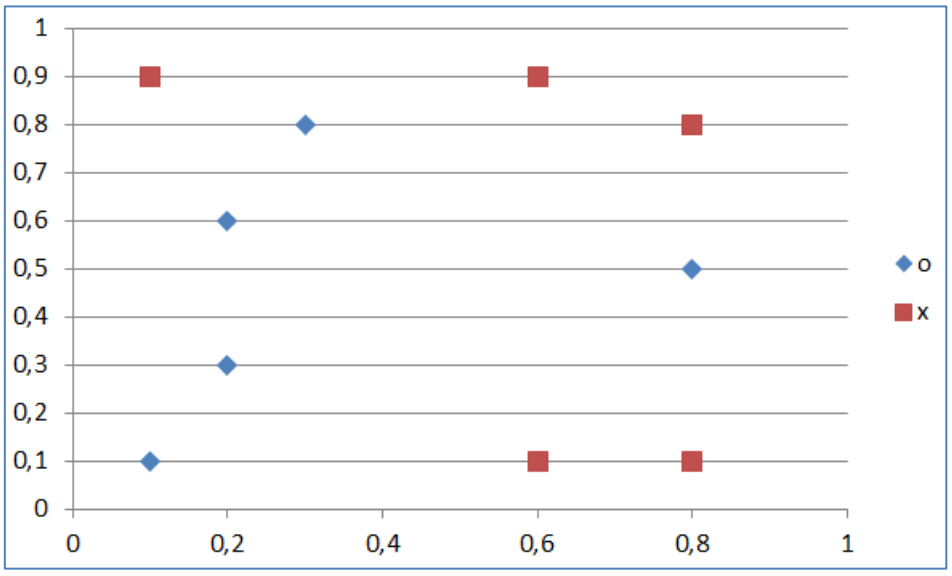

Рис. 8.14 Діаграма розсіювання по координатам з табл.. 8.5.

Розіб’ємо площину на дві частини по значенню $x_1 = 0{,}5$.

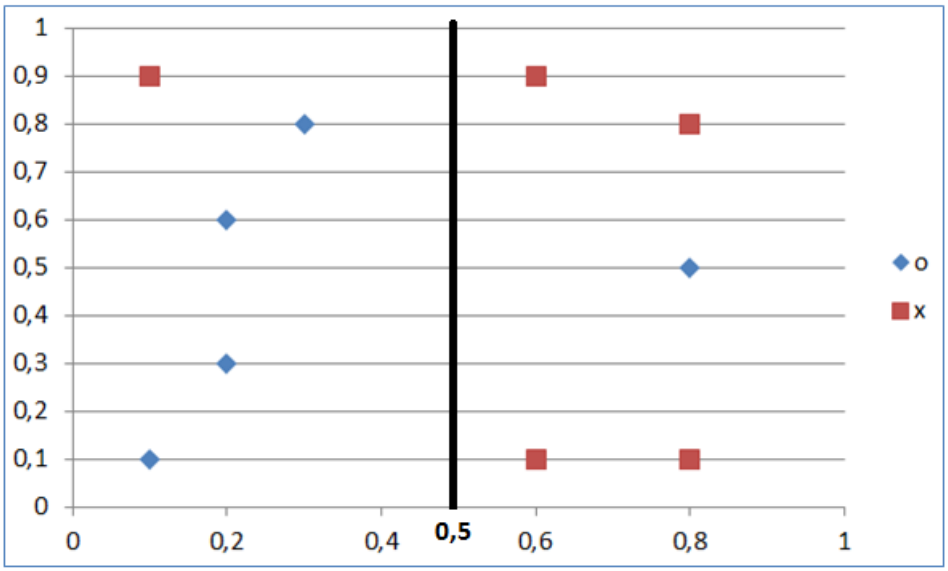

Рис. 8.15 Розбиття вибірки на дві частини по $x_1 = 0{,}5$

В табл. 8.6 наведена кількість елементів кожного типу в правій та лівій частинах, що містяться в дійсності, а в табл. 8.7 кількість, що очікується.

Таблиця 8.6

Observed		
	Left	Right
x	1	4
o	4	1

Таблиця 8.7

Expected		
	Left	Right
x	5	5
o	5	5

Для обчислення значення chi-square використовується формула

$$\chi^2 = chi - square = \sum_{k=1}^{n} \frac{(observed - expected)^2}{expected}$$

Для прикладу, що розглядається, значення chi-square обчислюється як:

$$\chi^2 = \frac{(1-5)^2}{5} + \frac{(4-5)^2}{1} = 3{,}4$$

Для того щоб отримати значення ймовірності для $\chi^2 = 3{,}4$ використовується спеціальна табл. 8.8 значень розподілу.

Таблиця 8.8

Таблиця значень ймовірностей хі-квадрат

	Значення ймовірності (chi-square p-value)										
Количество степеней свободы	0,95	0,9	0,8	0,7	0,5	0,3	0,2	**0,1**	**0,05**	0,01	0
1	0	0	0,1	0,2	0,5	1,07	1,64	**2,71**	**3,84**	6,64	10,8
2	0,1	0,2	0,5	0,7	1,4	2,41	3,22	4,6	5,99	9,21	13,8
3	0,35	0,6	1	1,4	2,4	3,66	4,64	6,25	7,82	11,3	16,3
4	0,71	1,1	1,7	2,2	3,4	4,88	5,99	7,78	9,49	13,3	18,5
5	1,14	1,6	2,3	3	4,4	6,06	7,29	9,24	11,1	15,1	20,5
6	1,63	2,2	3,1	3,8	5,4	7,23	8,56	10,6	12,6	16,8	22,5
7	2,17	2,8	3,8	4,7	6,4	8,38	9,8	12	14,1	18,5	24,3
8	2,73	3,5	4,6	5,5	7,3	9,52	11	13,4	15,5	20,1	26,1
9	3,32	4,2	5,4	6,4	8,3	10,7	12,2	14,7	16,9	21,7	27,9
10	3,94	4,9	6,2	7,3	9,3	11,8	13,4	16	18,3	23,2	29,6
	Не значимі								Значимі		

Окрім цього для визначення chi-square p-value можна скористатися готовими комп'ютерними програмами з Інтернету, наприклад Statistics Calculators 3:

http://www.danielsoper.com/statcalc3/calc.aspx?id=11

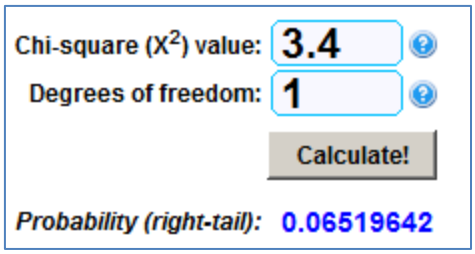

Рис. 8.16 Результати обчислення chi-square p-value програмою Statistics Calculators 3

Маючи p-value = 0,06519 для chi-square = 3,4 можна обчислити значення функції logworth як:

$$logworth = -\log_{10}(chi - square\ for\ p - value) =$$
$$= -\log_{10}(c0{,}06519) = 1{,}1858$$

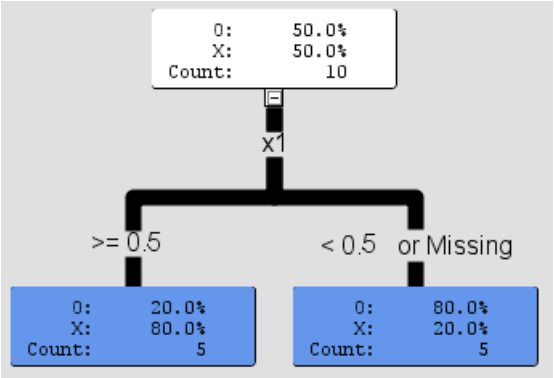

Рис. 8.17 Дерево рішення отримане в результаті розбиття вибірки на дві частини по $x_1 = 0{,}5$

Рис. 8.18 Статистичні характеристики розбиття по $x_1 = 0{,}5$ в SAS Enterprise Miner

Кінець чисельного прикладу.

Крок 4. Таким чином для кожного значення x_1 обчислюється значення logworth, на рис. 8.19 представлена у вигляді графіку красного кольору.

Будемо вважати, що найбільше значення logworth по x_1 досягається при $x_1 = 0{,}52$ для якого $logworth = 0{,}95$.

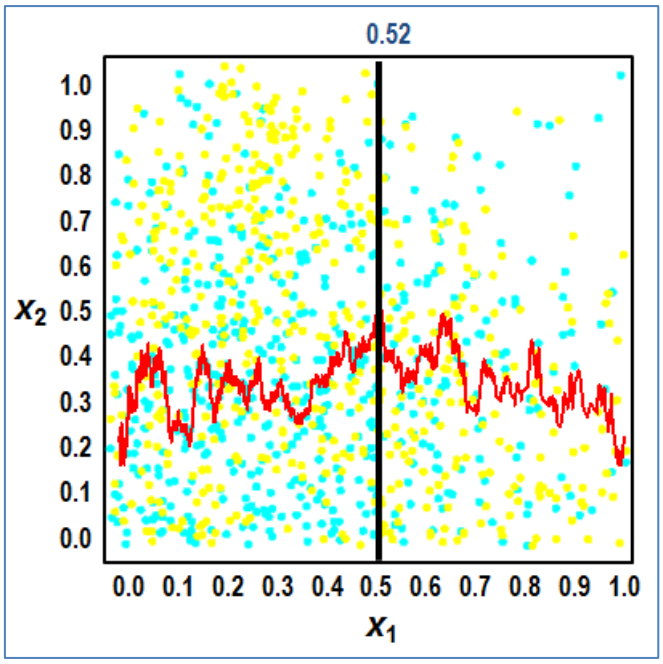

Рис. 8.19 Знаходження максимального значення logworth по x_1

Крок 5. Після того як на попередньому кроці 4 знайдена оптимальна точка розбиття, переходять до аналізу наступної змінної x_2.

Для змінної x_2 з кроком 0,01 виконується перебір всіх значень, на відрізку від 0 до 1.

Крок 6. Для кожного x_2 будується своя власна матриця контингенції. На рис. 8.19 наведений приклад побудованої матриці контингенції для $x_2 = 0{,}63$. Ця матриця містить інформацію про кількість точок відповідного кольору в ***верхній*** та ***нижній*** частинах, тому що переріз простору відбувається вертикально.

Рис. 8.20 Матриці контингенції для $x_2 = 0{,}63$

Крок 7. Для кожного значення x_2 обчислюється значення logworth, на рис. 8.21 представлена у вигляді графіку красного кольору.

Будемо вважати, що найбільше значення logworth по x_2 досягається при $x_2 = 0{,}63$ для якого $logworth = 4{,}92$.

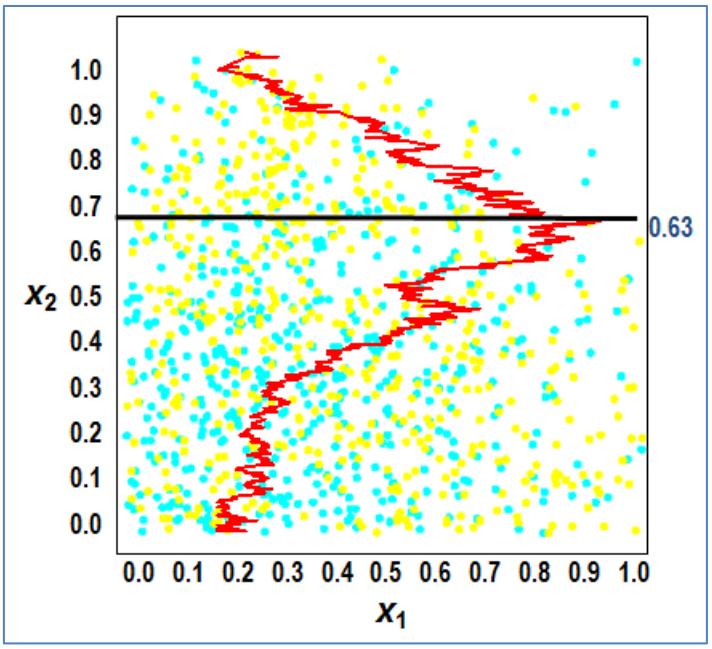

Рис. 8.21 Знаходження максимального значення logworth по $x_2 = 0{,}63$

Після того як знайдені максимальні значення logworth для x_1 та x_2 можна визначити за якою змінною буде відбуватися перше розбиття. В якості критерію знову виступає максимальне значення logworth.

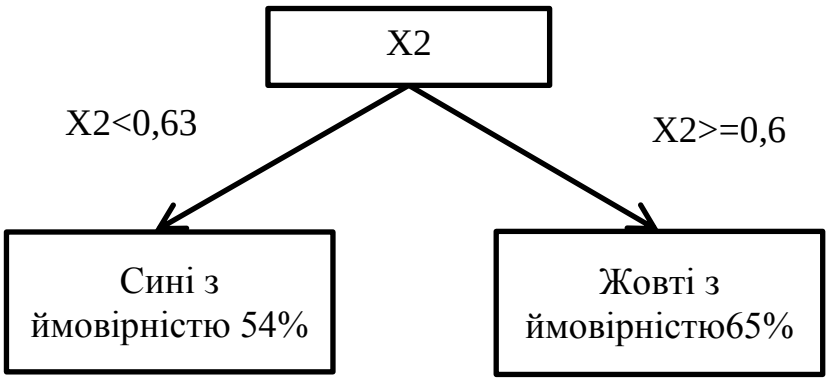

Рис. 8.22 Розбиття по змінній x_2

8.8.2 Вибір кореневої вершини

Наведений у попередньому підрозділі алгоритм на кроках 1-8 шукає оптимальні точки розділення для кожної змінної, при цьому важливу роль грає значення logworth. Як можна побачити на графіку з рис. 8.23 чим менше значення p-value, тим більше logworth.

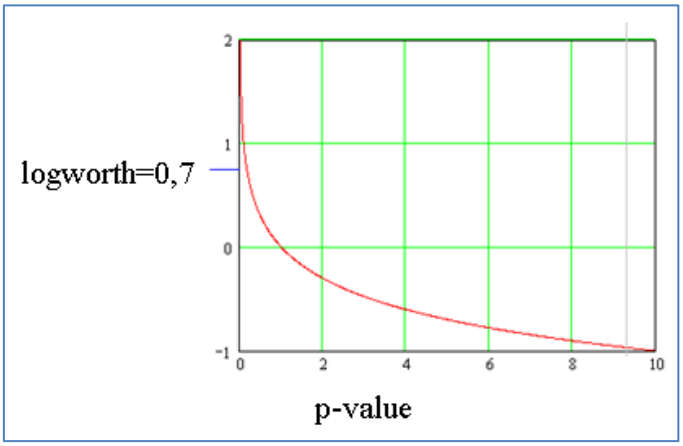

Рис. 8.23 Графік залежності функції logworth від значення p-value

В загальному випадку вхідних змінних може бути набагато більше за дві, тому для вибору кореневої змінної роблять наступне:

1. для кожної вхідної змінної обчислюють максимальне значення logworth, тобто знаходять відповідну точку розділення.

2. серед обчислених максимальних значень logworth для кожної змінної знаходять саме максимальне і надалі змінну, якій відповідає знайдене значення logworth, використовують в якості кореневої вершини – першої точки розділення.

8.8.3 Сороченя максимального дерева

В загальному випадку набір даних для аналізу поділяють на два набори – тренінгів (навчальний) та перевірочний (валідаційний).

Після того як побудоване максимальне дерево рішень на основі тренінгового набору даних починається етап скорочення дерева рішення (prunning).

В системі SAS Enterprise Miner будується окрім максимального дерева рішень також набір математичних моделей наростаючої складності. Тому вирішення задачі скорочення дерева рішень полягає в використанні валідаційного набору даних для оптимізації статистики, що збільшить ефективність фінальної моделі. Існують різні типи статистики, що відповідають кожному з трьох існуючих в загальному випадку типів прогнозів – рішень, рейтингів та оцінок.

Вибір статистики для визначення кращої математичної моделі залежить від типу прогнозу, що цікавить аналітика.

Таблиця 8.9

Таблиця вибору статистики в залежності від типу прогнозу

Тип прогнозу	**Статистика**	**Напрям**
Рішення	Помилкова класифікація	менше
Рішення	Середній дохід	більше
Рішення	Середній збиток	менше
Рішення	Статистика Калмогорова-Смірнова (KS)	більше
Рейтинги	Індекс ROC (узгодженість)	більше
Рейтинги	Коефіцієнт Джині	більше
Оцінки	Середня квадратична похибка	менше
Оцінки	Критерій Шварца-Байєса (SBC)	менше
Оцінки	Лагорифмічна правдоподібність	менше

8.9 Висновки до восьмого розділу

Даний розділ присвячено широковідомому підходу щодо прогнозного моделювання – деревам рішень. Наведено теоретичне обґрунтування використання методу CHAID із наведенням чисельного прикладу обчислення логарифмічної функції корисності (logworth).

В загальному випадку процедура побудови дерева рішень складається з двох етапів:

- вирощування максимального дерева рішень (growing maximum decision tree);

- скорочення максимального дерева рішень (prunning).

На етапі побудови максимального дерева рішень використовуються поправки Бонферроні, Касса та за глибиною для уникнення ситуацій, коли статистичний тест значимості може видавати хибний результат.

Якість побудованого дерева рішень залежить від репрезентативності вибірки даних та кваліфікації аналітика, який повинен коректно зробити розділення даних на навчальний та валідаційний набори даних. В залежності від цілей аналізу аналітик на етапі формалізації задачі обирає тип прогнозу цільової змінної, що в свою чергу приводить до необхідності вибору відповідної статистики для визначення кращої математичної моделі у видляді дерева рішень.

Дерева рішень знайшли дуже широке коло практичного використання в задачах, що потребують класифікації та прогнозування. Наприклад, оператори мобільного зв’язку на основі виявлених груп споживачів розробляють спеціальні маркетингові компанії для кожного виявленого сегменту. При чому для мільйонної бази абонентів дерево рішень спроможне виділити специфічні сегменти розміром до тисячі абонентів, що в умовах конкурентної боротьби є досить ваговим фактором, тому що в глобальному масштабі відтік одного проценту клієнтів обертається мільйонними збитками.