МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет "Львівська політехніка"

Інтелектуальний аналіз даних за допомогою програмного пакета WEKA та MS Excel.

Класифікація методом дерев рішень.

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторної роботи № 12

з курсу **"Системи інтелектуального аналізу та візуалізації даних"**

для студентів за освітньою програмою Комп'ютерні науки (Проектування і програмування інтелектуальних систем та пристроїв)

Затверджено на засіданні кафедри
"Системи автоматизованого проектування"

Протокол N 1 від 28.08.2023р.

ЛЬВІВ 2023

1. МЕТА РОБОТИ

Ознайомитися та отримати навички побудови моделей класифікації за допомогою Data Mining GUI бібліотеки Weka та Excel. На практиці вивчити роботу методу побудови дерев рішень, навчитися інтерпретувати результати роботи класифікаторів.

2. КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1. Метод побудови дерев рішень

Дерева рішень (Decision Trees) є популярним методом вирішення завдань класифікації та прогнозування. Дерева рішень дозволяють візуально і аналітично оцінити результати вибору різних рішень і використовуються в галузі статистики та аналізу даних для прогнозних моделей.

Дерева рішень використовують, коли потрібно прийняти рішення в умовах невизначеності, де кожне рішення залежить від результату попередніх результатів або деяких заданих умов, що з'являються з певною ймовірністю.

Дерева рішень часто називаються деревами вирішальних правил, деревами класифікації і регресії. Якщо залежна (цільова) змінна приймає дискретні значення – це завдання класифікації, якщо залежна змінна приймає безперервні значення, то вирішується завдання прогнозування.

Дерево рішень, подібно його «прототипу» з живої природи, складається з гілок з атрибутами (від них залежить результат - цільова функція), листів зі значеннями цільової функції (вирішальні вершини - результат вибору певного значення атрибута), а також вузлів - випадкових вершин, в яких визначаються можливі варіанти розвитку подій з певного моменту (рис. 1.).

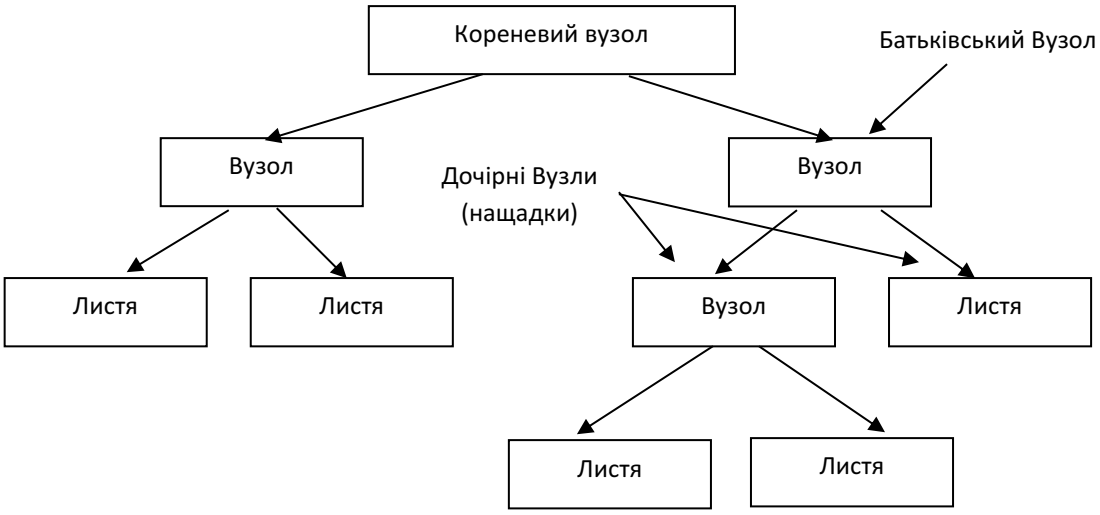

Рис. 1. Основні терміни дерева рішень

Метою процесу побудови дерева прийняття рішень є створення моделі, за якою можна було б класифікувати випадки і вирішувати, які значення може приймати цільова функція, маючи на вході кілька змінних.

У найбільш простому вигляді дерево рішень - це спосіб представлення правил «Якщо, тоді» в ієрархічній, послідовної структурі. Основою такої структури є відповіді "Так" або "Ні" на ряд питань.

Спосіб поділу вузла в дереві рішень залежить від ентропії, обчисленої з даними всередині вузла. Ентропія показує, наскільки «чистими» є дані всередині вузла дерева. Чим вища ентропія, тим менш чисті дані. Припустимо, що набір даних має m різних класів. Ентропія цього набору даних зазвичай обчислюється на основі рівняння (1):

$$H = -\sum_{k=1}^{m} P_K \log_2 (P_k) \quad (1)$$

Для детальнішої інформації зверніться до відповідної лекції.

Наочним прикладом для класифікації за алгоритмом дерев рішень є надання кредиту в банку. Як банку переконатися, чи буде повернуто кредит? У банка є тисячі профілів інших людей, які вже брали кредит. Там вказано їх вік, освіту, посаду, рівень зарплати та головне - хто з них повернув кредит, а з ким виникли проблеми.

Ці дані можна використовувати як навчальну множину, на який алгоритм навчають передбачати результат. Але, проблема в тому, що банк не може повністю довіряти відповіді машини, без пояснень: може статися збій, втручання хакерів або ненавмисна зміна певного скрипту адміністратором.

Алгоритм Дерева рішень автоматично розділяє всі дані відповідно від питання. Вирішується завдання бінарної класифікації, для вибору є лише дві відповіді на поставлене питання ("так" і "ні"). Бінарні дерева є найпростішим, окремим випадком дерев рішень. В інших випадках, відповідей і, відповідно, гілок дерева, що виходять з його внутрішнього вузла, може бути більше за двох.

Питання можуть бути не зовсім адекватними з точки зору людини, наприклад «зарплата позичальника більше, ніж 25000 гривень?», але машина придумує їх так, щоб на кожному кроці розділення було найточнішим. Так виходить дерево питань. Чим вище рівень, тим більш узагальнене питання (рис.2.).

Рис.2. Наочний приклад дерева рішень

Припустимо, існує вибірка з тисяч записів, де кожний запис – це опис характеристик клієнта та відомості щодо повернення кредиту. При навчанні дерева використано фактори:

№ паспорту	Розмір кредиту
Прізвище, ім’я, по батькові	Термін кредиту
Адреса	Мета кредиту
	Місячний дохід
Ці поля визначено як несуттєві	Місячні витрати
	Основні витрати
	Наявність власного житла

Цільовим полем буде поле Видати кредит, що приймає значення Так / Ні.

Після побудови дерева отримується модель оцінювання кредитоспроможності клієнтів певного банку у вигляді ієрархічної структури правил.

Алгоритми побудови визначають суттєві фактори. На кожному вузлу ієрархії використовується критерій, який вирішує найбільшу невизначеність. Суттєві фактори розташовуються на найближчому рівні від кореня ніж інші.

Певні фактори можуть бути замінені одним узагальненим фактором. Наприклад, Розмір кредиту, Термін кредиту, Середньомісячний дохід, Місячні витрати – можуть не використовуватися, бо існує фактор Кредит під заставу, що є їх вдалим узагальненням.

Правила, за яким визначається належність клієнта до певної групи, записуються природньою мовою:

Якщо «Кредит під заставу» - Так і
«Термін проживання в цьому місті» > 19 років і
«Наявність нерухомості» - Так і
«Наявність банківського розрахунку» - Так
Тоді «Надати кредит» - Так
Достовірність = 98%

Правильно побудоване дерево має властивості до узагальнення, тобто, якщо виникає нова ситуація (новий клієнт), то ймовірно такі ситуації вже були і клієнт буде поводитися аналогічно, як клієнти з подібними характеристиками.

Метод дерев рішень часто називають "наївним" підходом. Але завдяки цілому ряду переваг, даний метод є одним з найбільш популярних для вирішення задач класифікації.

Дерева знайшли свою нішу в областях з високою відповідальністю: діагностиці, медицині, фінансах. У чистому вигляді дерева сьогодні використовують рідко, але їх комбінації лежать в основі складних систем і часто обробляють навіть результати від нейромережі. Наприклад, коли задається питання до пошукової системи, ранжуванням результатів займаються саме дерева рішень.

2.2. Виконання класифікації у Weka

Наведений нижче приклад є перекладом та інтерпретацією англомовних вказівок до виконання класифікація на прикладі даних, отриманих вд дилерськиих центрів BMW.

https://www.programmersought.com/article/9237166886/

У нашому випадку було взято датасет bmw-training.arff з 3000 записів та проведено його рощеплення на власне bmw-training.arff (1999 записів) та bmw-test.arff з рештою записів.

Одним із ваших завдань є порівняти отримані результати у першоджерелі із 4500 записами та результатами, отриманими на 3000 записів.

Набір даних, який буде застосований для прикладу класифікаційного аналізу, містить інформацію (bmw-training.arff, bmw-test.arff), зібрану центром продажу компанії BMW. Центр починає рекламну компанію, пропонуючи розширену дворічну гарантію своїм постійним клієнтам. Подібні компанії вже проводилися, так що центр продажу має у розпорядженні 3000 екземплярів даних щодо попередніх продажів з розширеною гарантією. Цей набір даних охоплює наступні атрибути:

- Розподіл за доходами [0=$0-$30k, 1=$31k-$40k, 2=$41k-$60k, 3=$61k-$75k, 4=$76k-$100k, 5=$101k-$150k, 6=$151k-$500k, 7=$501k+];
- Рік / місяць покупки першого автомобіля BMW;
- Рік / місяць покупки останнього автомобіля BMW;
- Чи скористався клієнт розширеною гарантією?

Файл даних у форматі Attribute-Relation File Format (ARFF) буде виглядати наступним чином, див. Лістинг 1.

Лістинг 1. Файл даних для класифікаційного аналізу у Weka

```
@attribute IncomeBracket {0,1,2,3,4,5,6,7} % Групи за доходом
@attribute FirstPurchase numeric % перша покупка
@attribute LastPurchase numeric % остання покупка
@attribute responded {1,0} % відгук
```

```
@data
4,200210,200601,0
5,200301,200601,1
...
```

При побудові моделі класифікації набір даних зазвичай ділять так, щоб частина даних використовувалася для побудови моделі (навчання), частина – для перевірки її коректності (тестування), щоб переконатися, що модель не є навченою тільки під конкретний набір даних.

Розділіть вибраний набір даних на два файли *.arff в співвідношенні «2/3» (навчальні дані bmw-training.arff) та «1/3» (тестові дані bmw-test.arff) від загальної кількості даних. Завантажте файл «2/3» в програмний пакет Weka.

2.2.1. Завантаження даних у Weka Explorer

Коли файл з навчальними даними готовий, його потрібно завантажити у Weka. Запустіть Weka і виберіть опцію Explorer. У результаті відкриється закладка Preprocess у вікні Explorer. Натисніть кнопку Open File і виберіть створений вами ARFF-файл. Вікно Weka Explorer з завантаженими даними показано на Рис. 3. Зауваження: в пропонованому навчальному файлі містяться 1999 записів.

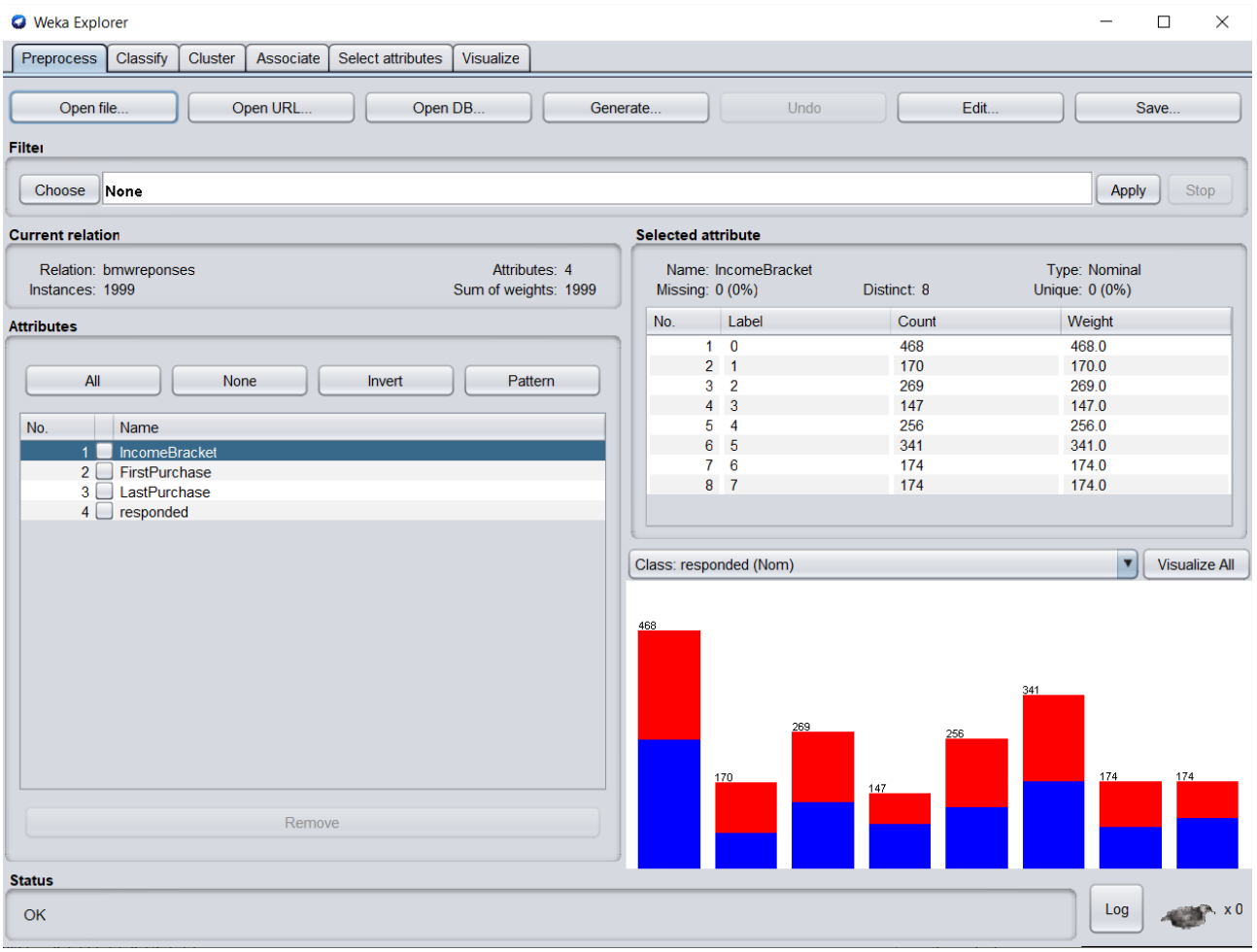

Рис. 3. Дані дилерського центру BMW

У цьому вікні ви можете перевірити дані, на підставі яких ви збираєтесь будувати модель. У лівій частині вікна Explorer показані атрибути даних (Attributes), які відповідають заголовкам стовпців таблиці, також вказано кількість екземплярів даних (Instances), тобто рядків таблиці. Якщо виділити мишкою однин з заголовків стовпців, тоді в правій частині вікна з'являться значення відповідного атрибуту для різних екземплярів даних. Також існує можливість візуального аналізу даних за допомогою кнопки Visualize All.

2.2.2. Побудова моделі класифікації у Weka Explorer

Виберіть метод класифікації (див. Рис.4.): відкрийте закладку Classify, виберіть опцію trees, потім опцію, наприклад опцію **J48**. (див. Таблицю 1).

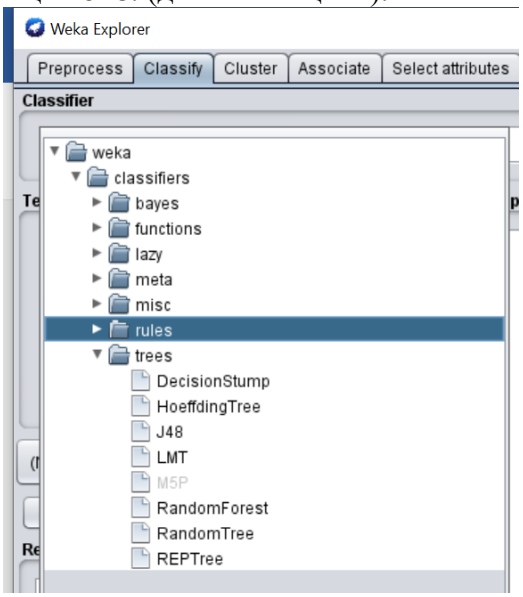

Рис. 4. Вибір методу класифікації даних

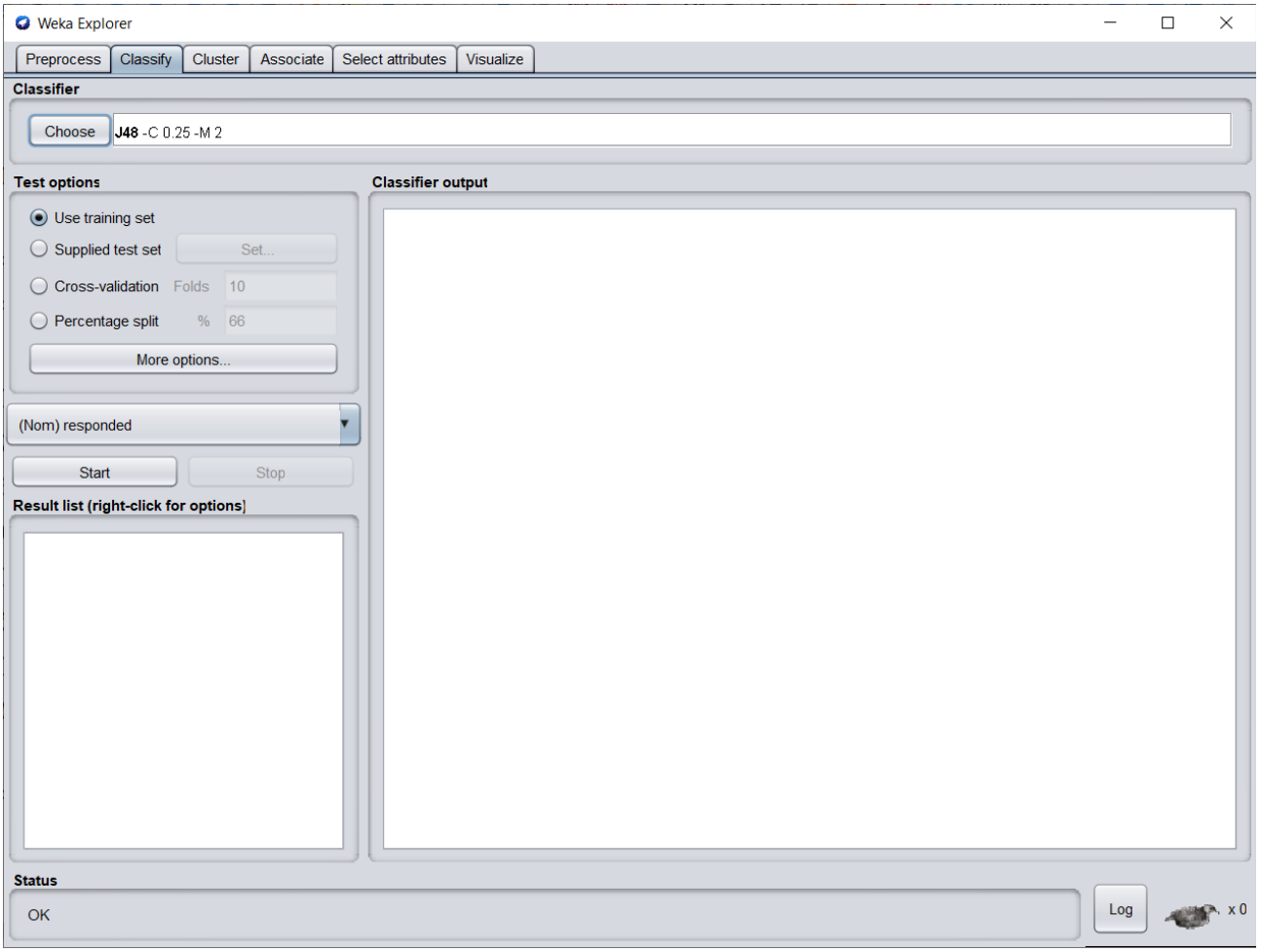

Рис. 5. Вибір алгоритму класифікації даних BMW у навчальному режимі

Тепер можна розпочати побудову моделі класифікації засобами пакету Weka. Переконайтеся, що обрана опція Use training set, для того щоб пакет Weka при побудові моделі використовував саме ті дані, які ви тільки що завантажили з файлу. Натисніть Start. Результуюча модель повинна виглядати так, як показано на Лістинг 2.

Лістинг 2. Результат роботи класифікаційної моделі Weka

```
Number of Leaves  :     13

Size of the tree :      19


Time taken to build model: 0.01 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.02 seconds

=== Summary ===

Correctly Classified Instances        1151               57.5788 %
Incorrectly Classified Instances       848               42.4212 %
Kappa statistic                          0.1554
Mean absolute error                      0.4806
Root mean squared error                  0.4902
Relative absolute error                 96.1559 %
Root relative squared error             98.0592 %
Total Number of Instances             1999

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.470    0.314    0.610      0.470    0.531      0.160    0.597     0.583     1
                 0.686    0.530    0.554      0.686    0.613      0.160    0.597     0.561     0
Weighted Avg.    0.576    0.420    0.582      0.576    0.571      0.160    0.597     0.572

=== Confusion Matrix ===

   a   b   <-- classified as
 480 541 |   a = 1
 307 671 |   b = 0
```

Що означають всі ці числа? Як нам зрозуміти, наскільки хороша отримана модель? І де, власне, це так зване «дерево», яке повинні були отримати в результаті? Цілком закономірні питання. Давайте відповімо на кожне з них по черзі:

- Що означають всі ці числа? Найбільш суттєві дані - це показники класифікації "Correctly Classified Instances" (57.6%) і "Incorrectly Classified Instances" (42.4%). Крім того, слід звернути увагу на число в першому рядку стовпця ROC Area (0.597). Нарешті, таблиця Confusion Matrix показує кількість хибнопозитивних (541) і помилково негативних (671 розпізнавань).
- Як зрозуміти, наскільки хорошою є отримана модель? Оскільки показник точності нашої моделі - 57,6%, то в початковому розгляді її не можна назвати досить хорошою.
- Де дерево? Ви зможете побачити дерево, якщо клацнете правою кнопкою мишки в панелі результуючої моделі. У контекстному меню виберіть опцію Visualize tree. На екрані відобразиться візуальне уявлення класифікаційного дерева нашої моделі (рис.6.), проте в даному випадку картинка мало чим нам допоможе. Ще один спосіб побачити дерево моделі - прокрутити вгору висновок у вікні Classifier Output, там ви знайдете текстовий опис дерева з вузлами і листям.

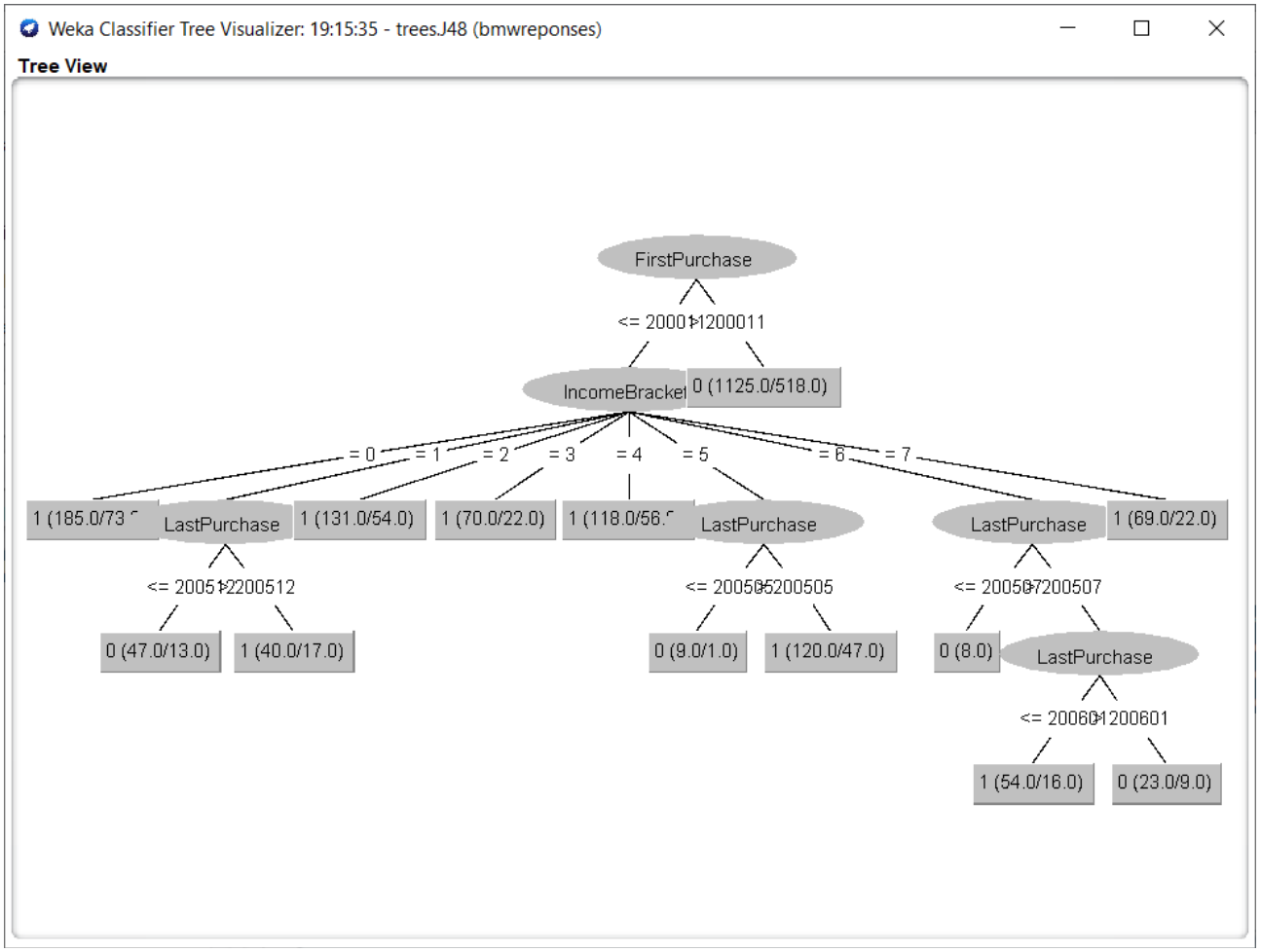

Рис.6. Графічне відображення моделі класифікації

2.2.3. Тестування моделі класифікації

Тестова перевірка моделі класифікації дозволяє уникнути зайвого перенавчання моделі. Оскільки модель класифікації будується для класифікації некласифікованих екземплярів, при перевірці її оптимальності використовується тестовий набір даних. Таким чином, гарантується, що побудована модель класифікації зможе з досить високою ймовірністю визначити клас ще некласифікованого екземпляру.

Залишився останній етап перевірки класифікаційного дерева: нам треба пропустити тестовий набір даних через отриману модель і перевірити, наскільки результати класифікації будуть відрізнятися від реальних даних. Перевірку моделі треба провести на тестовому наборі даних «1/3» і оцінити, наскільки результати класифікації відрізняються від тестових класів. Для цього в секції Test options виберіть опцію Supplied test set і натисніть Set. Вкажіть файл з даними, що містить тестові «1/3» дані, які не були включені в навчальний набір (див. Рис.7).

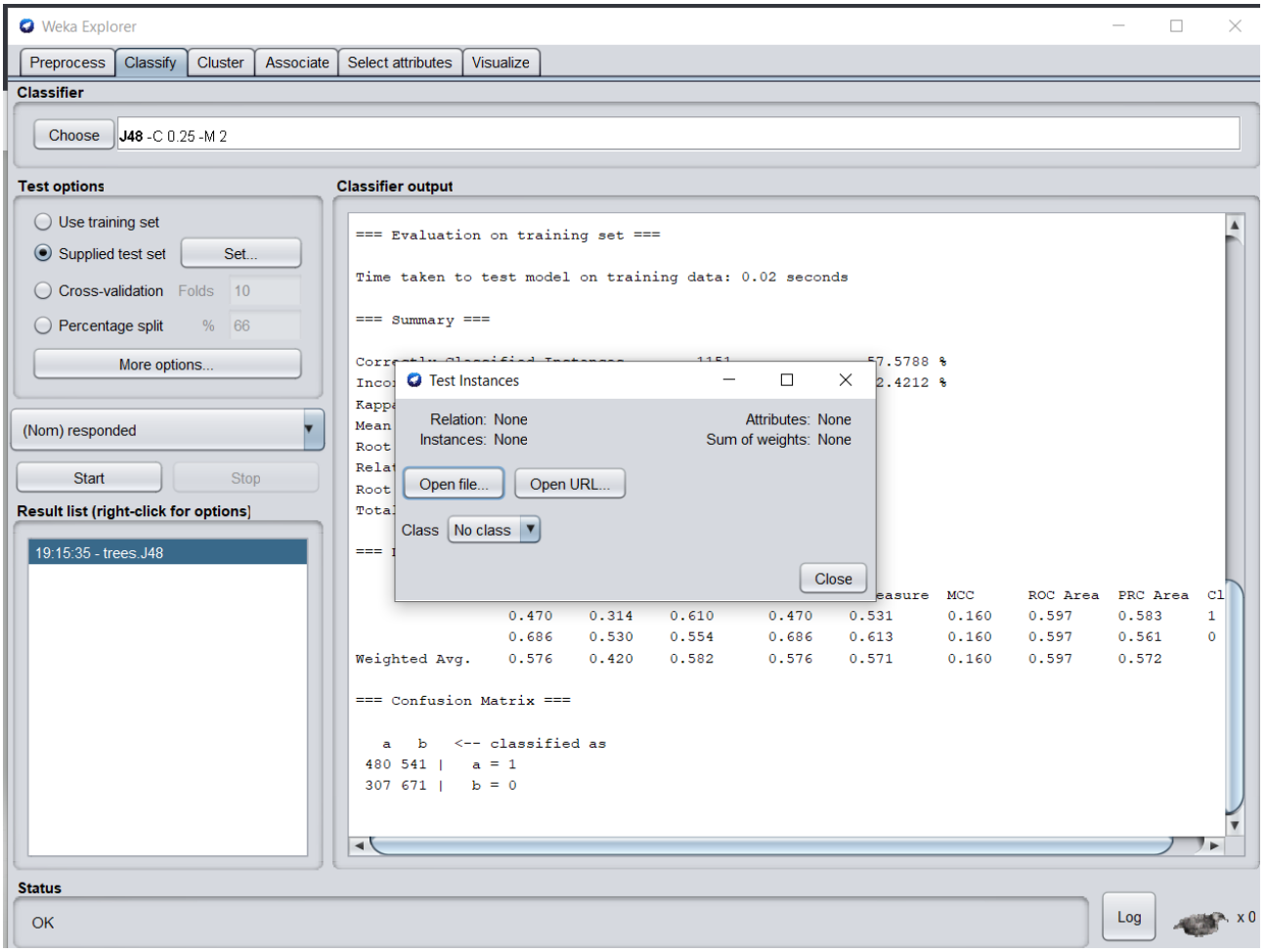

Рис. 7 Задання тестової вибірки

При натисканні на кнопку Start, Weka проведе класифікацію тестових даних і виведе інформацію про оптимальність побудованої моделі (див. Рис. 8).

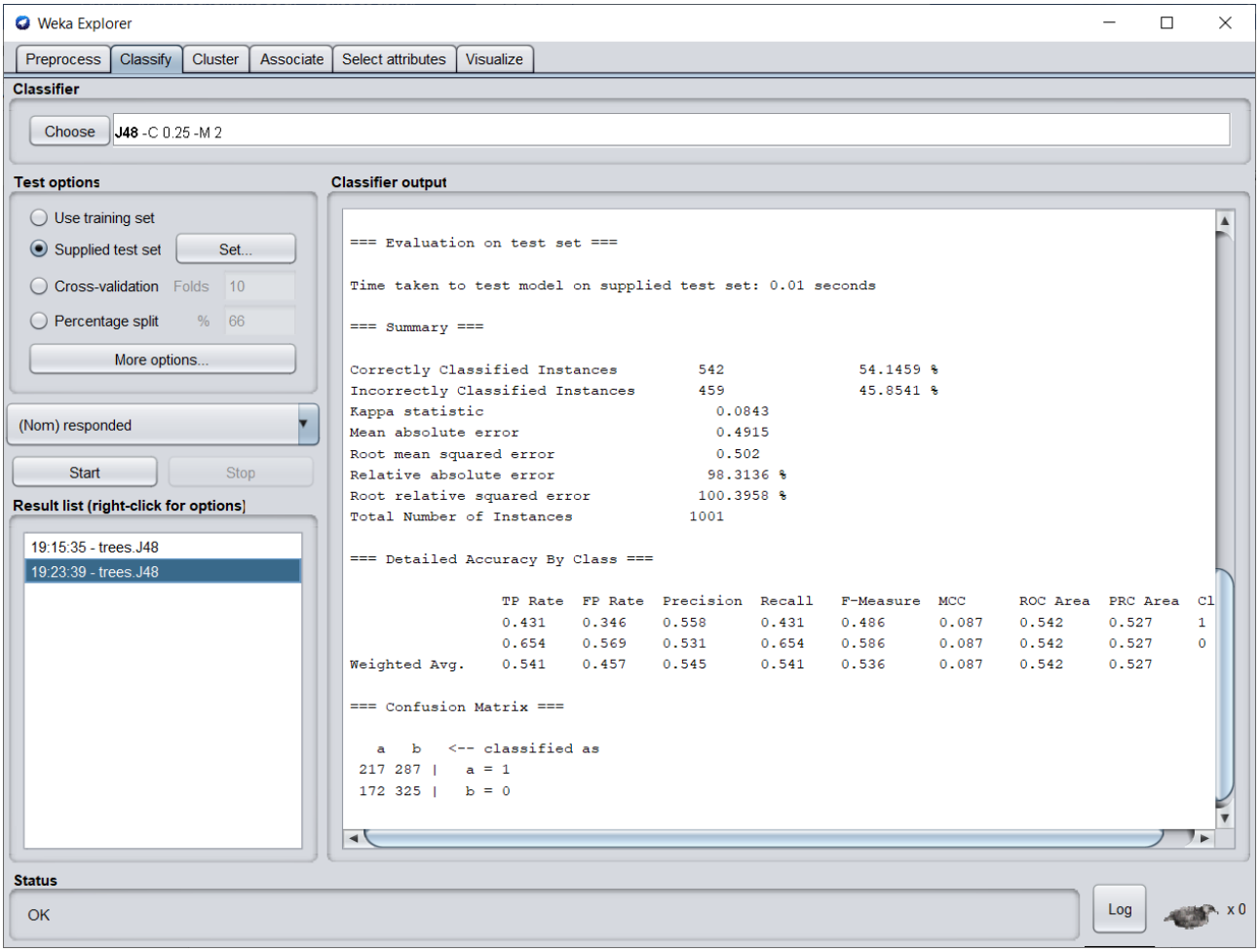

Рис. 8. Перевірка моделі класифікації

Порівнюючи показник Correctly Classified Instances для тестового набору (54,1%) з цим же показником для навчального набору (57,6%), видно, що точність моделі для двох різних наборів даних приблизно однакова. Це означає, що нові дані, які будуть класифікуватися за допомогою цієї моделі в майбутньому, не зменшать точність її роботи. Для підвищення точності класифікації рекомендується збільшити кількість навчальних і тестових даних та число атрибутів класів.

2.2.4. Параметри налаштування алгоритмів класифікації

Розглянемо параметри налаштування алгоритму, що використовується у лабораторній роботі (табл.1).

Таблиця 1. Параметри налаштування класифікаторів

Метод	**Параметр**
J48	*binarySplits* – використовувати бінарний поділ на категоріальних атрибутах для побудови дерев. *confidenceFactor* – довірчий рівень, використовується для відсікання гілок (малі значення - сильніше відсікання). *minNumObj* – мінімальна кількість примірників у листі. *reducedErrorPruning* – який алгоритм відсікання гілок використовувати

	saveInstanceData – чи зберігати навчальну інформацію для візуалізації. *subtreeRaising* – чи використовувати операцію підняття піддерев при відсіканні гілок. *unpruned* – чи залишити дерево повним. *useLaplace* – використовувати оціночну функцію Лапласа для підрахунку ймовірностей в листках

2.2.5. Інтерпретація результатів класифікації в WEKA

Розглянемо результати роботи класифікаторів в WEKA (Classifier output).
Секція «*Run information*» містить наступну інформацію:

- метод класифікації (scheme);
- назва набору даних, на якому проводилося навчання (relation);
- кількість примірників у вихідній вибірці (instances);
- атрибути, що характеризують об'єкти вибірки (attributes);
- відомості про тестову вибірку (test mode).

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     bmwreponses
Instances:    1999
Attributes:   4
              IncomeBracket
              FirstPurchase
              LastPurchase
              responded
Test mode:    user supplied test set:  size unknown (reading incrementally)
```

Секція «*Classifier model*» містить параметри налаштованого класифікатора і час, затрачений для побудови моделі. Залежно від типу класифікатора дана область буде містити різну інформацію:

- для алгоритмів, що будують правила, будуть відображені отримані правила;
- для байєсівських класифікаторів будуть перераховані розраховані ймовірності для всіх можливих комбінацій атрибут-значення-клас;
- для класифікаторів, заснованих на побудові дерев, відображається текстове представлення отриманого дерева; в дужках навпроти кожного листа вказана кількість примірників, які до нього віднесені; якщо в лист потрапляють екземпляри декількох класів, через слеш буде вказана кількість примірників, які відносяться до домішок;
- для функціональних методів виводяться значення коефіцієнтів побудованої функціональної моделі;
- для методу k найближчих сусідів відображаються налаштування класифікатора.

```
=== Classifier model (full training set) ===

J48 pruned tree
------------------

FirstPurchase <= 200011
|   IncomeBracket = 0: 1 (185.0/73.0)
```

```
|   IncomeBracket = 1
|   |   LastPurchase <= 200512: 0 (47.0/13.0)
|   |   LastPurchase > 200512: 1 (40.0/17.0)
|   IncomeBracket = 2: 1 (131.0/54.0)
|   IncomeBracket = 3: 1 (70.0/22.0)
|   IncomeBracket = 4: 1 (118.0/56.0)
|   IncomeBracket = 5
|   |   LastPurchase <= 200505: 0 (9.0/1.0)
|   |   LastPurchase > 200505: 1 (120.0/47.0)
|   IncomeBracket = 6
|   |   LastPurchase <= 200507: 0 (8.0)
|   |   LastPurchase > 200507
|   |   |   LastPurchase <= 200601: 1 (54.0/16.0)
|   |   |   LastPurchase > 200601: 0 (23.0/9.0)
|   IncomeBracket = 7: 1 (69.0/22.0)
FirstPurchase > 200011: 0 (1125.0/518.0)
```

Секція «*Predictions*» буде відображена, якщо в налаштуваннях тестування класифікатора обрана опція «Output predictions». У ній для всіх примірників тестової вибірки будуть відображені результати класифікації, отримані за допомогою навченого класифікатора.

```
=== Predictions on test set ===

    inst#     actual  predicted error prediction
        1       2:0       2:0       0.54
        2       1:1       1:1       0.575
        3       2:0       2:0       0.54
        4       2:0       2:0       0.54
        5       2:0       1:1   +   0.686
        6       2:0       2:0       0.54
        7       2:0       1:1   +   0.588
        8       1:1       1:1       0.605
        9       2:0       2:0       0.54
       10       2:0       1:1   +   0.704
...
```

Секція оцінки побудованої моделі «*Evaluation*» містить кілька підпунктів.

«*Summary*» містить загальну статистику роботи класифікатора:

- кількість та відсоток правильно і неправильно класифікованих примірників (Correctly and Incorrectly Classified Instances), загальна кількість примірників (Total Number of Instances);
- параметр Каппа (Kappa statistic);
- статистичні параметри помилки класифікації (Mean absolute error, Root mean squared error, Relative absolute error, Root relative squared error).

«*Detailed Accuracy By Class*» містить наступні параметри точності класифікації по кожному з класів: TP Rate, FP Rate, Precision, Recall, F-Measure, ROC Area.

«*Confusion Matrix*» містить матрицю помилок.

2.3. Класифікація методом дерев рішень за допомогою Excel

Створіть у файлі робочий аркуш під назвою level-1 (level означає, що це перший рівень вузлів дерева). Дані у робочому аркуші мають бути такі, як показано на рис.9.

	A	B	C	D	E
1	**Temperature**	**Humidity**	**Windy**	**Outlook**	**Play**
2	hot	high	FALSE	overcast	yes
3	cool	normal	TRUE	overcast	yes
4	mild	high	TRUE	overcast	yes
5	hot	normal	FALSE	overcast	yes
6	mild	high	FALSE	rainy	yes
7	cool	normal	FALSE	rainy	yes
8	cool	normal	TRUE	rainy	no
9	mild	normal	FALSE	rainy	yes
10	mild	high	TRUE	rainy	no
11	hot	high	FALSE	sunny	no
12	hot	high	TRUE	sunny	no
13	mild	high	FALSE	sunny	no
14	cool	normal	FALSE	sunny	yes
15	mild	normal	TRUE	sunny	yes

level-1

Рис.9. Дані про гольф в Excel

Нам потрібно налаштувати наші дані у відповідній таблиці, щоб ми могли автоматично заповнювати формули. Ми бачили таку організацію таблиць у попередніх лабораторних.

1. Введіть «Розмір вибірки» в клітинку A16 і 14 у клітинку B16. 14 - це розмір даних.
2. Введіть «p∗log(p)» у клітинку E17. p∗log(p) представляє PKlog2(Pk) у рівнянні (1).
3. Введіть «ентропія» в клітинку F17.
4. Об’єднайте клітинки B18 і B19 і введіть «Відтворити» всередині об’єднаної клітинки.
5. Введіть «так» у клітинку C18 і «ні» у клітинку C19.

Зараз частина нашого робочого аркуша виглядає як Рис.10, за винятком того, що на нашому аркуші немає чисел у комірках D18 або D19. Незабаром ми їх обчислимо.

	A	B	C	D	E	F
14	cool	normal	FALSE	sunny	yes	
15	mild	normal	TRUE	sunny	yes	
16	Sample-size	14				
17					p*log(p)	entropy
18		Play	yes	9		
19			no	5		
20						

Рис.10. Набір даних про гольф і налаштування таблиці для Play

Дотримуйтеся цих інструкцій, щоб завершити обчислення ентропії для цільової змінної Play:

6. У комірці D18 введіть формулу =COUNTIFS($E$2:$E$15,C18). Це підраховує кількість «так» для Play.
7. Автозаповніть клітинки D18 до клітинки D19. D19 підраховує, скільки «ні» для Play.
8. У клітинку E18 введіть формулу =D18/$B$16*LOG(D18/$B$16,2). Ця формула обчислює Pyes × Log2(Pyes).
9. Автозаповніть кілтинки від E18 до E19. E19 обчислює Pno × Log2(Pno).
10. Об’єднайте клітинки F18 і F19 і введіть формулу =−SUM(E18:E19) усередині об’єднаної клітинки. Результатом є ентропія для цільової змінної Play.

Ваш аркуш має виглядати так, як показано на Рис.11.

	A	B	C	D	E	F
16	Sample-size	14				
17					p*log(p)	entropy
18		Play	yes	9	-0.40977638	0.94028596
19			no	5	-0.53050958	
20						

Рис.11. Обчислена ентропія гри

Настав час налаштувати відповідні таблиці для чотирьох атрибутів. Хитрість полягає в тому, щойно перша формула визначена правильно, її можна використати для вертикального автозаповнення щоб працювати з різними атрибутами, розташованими в окремих стовпцях (див. рис.9). Це може здатися складним, але функція INDEX може допомогти.

Функція INDEX потребує масив (насправді це таблиця) як перший вхідний параметр. Якщо другий параметр (рядок) дорівнює 0, функція повертає стовпець, визначений номером стовпця у масиві. У наборі даних «Temperature», «Humidity», «Windy» і «Outlook» є стовпцями 1, 2, 3 і 4. Вираз INDEX($A$2:$E$15,0,1) отримує стовпець «Temperature», тоді як INDEX($A$2 :$E$15,0,4) отримує стовпець Outlook.

Дотримуйтесь цих інструкцій, щоб налаштувати допоміжні таблиці для чотирьох атрибутів:

11. Введіть 1, 1, 1, 2, 2, 3, 3, 4, 4 і 4 у комірки A22:A31 відповідно.

12. Введіть "так", "ні", "p∗log(p)-так", "p∗log(p)-ні", "weighted", "entropy" та "info gain" в клітинках D21: J21 відповідно. У стовпці «weighted» зберігається зважена ентропія для кожного значення атрибута.

13. Об’єднайте комірки B22:B24 і введіть «Temperature» в об’єднану комірку.

14. Введіть «hot», «mild» і «cool» у клітинки C22, C23 і C24 відповідно.

Частина вашого робочого аркуша виглядає так, як на рис.12. Зверніть увагу, що під час вирівнювання клітинок A22:A24 до C22:C24 «hot», «mild» і «cool» відповідають 1, оскільки всі вони є значеннями атрибута

Температура, яка є першим стовпцем у таблиці A1:E15.

	A	B	C	D	E	F	G	H	I	J
18		Play	yes	9	-0.40977638	0.94028596				
19			no	5	-0.53050958					
20										
21				yes	no	p*log(p)-yes	p*log(p)-r	weighted	entropy	info gain
22	1		hot							
23	1	Temperature	mild							
24	1		cool							
25	2									
26	2									
27	3									
28	3									
29	4									
30	4									
31	4									

Рис.12. Процес налаштування таблиці

15. Об’єднайте комірки B25 і B26 і введіть «Humidity» у об’єднану комірку.

16. Введіть «high», «normal» у клітинках C25 і C26 відповідно.

17. Об’єднайте клітинки B27 і B28 і введіть «Windy» в об’єднану клітинку.

18. Введіть «TRUE» в клітинку C27 і «FALSE» в клітинку C28.

19. Об’єднайте комірки B29:B31 і введіть «Outlook» у об’єднану комірку.

20. Введіть «overcast», «rainy» і «sunny» в клітинках C29, C30 і C31 відповідно.

Частину вашого аркуша можна порівняти з рис.13.

	A	B	C	D	E	F	G	H	I	J
16	Sample-size	14								
17					p*log(p)	entropy				
18		Play	yes	9	-0.40977638	0.94028596				
19			no	5	-0.53050958					
20										
21				yes	no	p*log(p)-yes	p*log(p)-no	weighted	entropy	info gain
22	1		hot							
23	1	Temperature	mild							
24	1		cool							
25	2	Humidity	high							
26	2		normal							
27	3	Windy	TRUE							
28	3		FALSE							
29	4		overcast							
30	4	Outlook	rainy							
31	4		sunny							

Рис.13. Налаштування всіх таблиць

Підготувавши допоміжні таблиці, дотримуйтеся цих інструкцій, щоб обчислити ентропію та приріст інформації:

21. Введіть наступну формулу в клітинку D22:

=COUNTIFS(ІНДЕКС($A$2:$E$15,0,$A22), $C22,$E$2:$E$15,D$21)

Комірка A22 посилається на 1; тому ІНДЕКС($A$2:$E$15,0,$A22)

отримує стовпець температури. Ця формула підраховує кількість точок даних, у яких Temperature яких є «hot», а «Play» — «yes».

22. Автоматично заповніть клітинки D22 до E22, а потім автоматично заповніть клітинки D31:E31. Це показано на рис.14.

	A	B	C	D	E	F
20						
21				yes	no	p*log(p)-yes
22	1		hot	2	2	
23	1	Temperature	mild	4	2	
24	1		cool	3	1	
25	2	Humidity	high	3	4	
26	2		normal	6	1	
27	3	Windy	TRUE	3	3	
28	3		FALSE	6	2	
29	4		overcast	4	0	
30	4	Outlook	rainy	3	2	
31	4		sunny	2	3	
32						

Рис.14. Автозаповнення атрибутів

23. Введіть наступну формулу в клітинку F22:

=IF(ISERROR(D22/SUM($D22:$E22)*LOG(D22/SUM($D22:$E22),2)),0,D22/SUM($D22:$E22)*LOG(D22/SUM($D22:$E22),2))

Можливо, що SUM($D22:$E22) поверне 0, і тому буде помилка ділення на нуль у виразі D22/SUM($D22:$E22). Крім того, log функція не може приймати 0 як вхідні дані. Таким чином, функція ISERROR використовується тут для виявлення таких помилок. Якщо виникає помилка, повертається 0. Примітка: використання функції IF разом із ISERROR є критичною тут.

Ця формула обчислює Pyes × Log2(Pyes) для значення атрибута «hot».

24. Автозаповніть з комірки F22 до комірки G22, а потім разом автозаповніть комірки F31:G31. Примітка: комірка G22 обчислює Pno × Log2(Pno) для значення атрибута «hot».

25. У клітинку H22 введіть формулу =−SUM($D22:$E22)/$B$16*(F22+G22). Розраховане значення є зваженою субентропією для значення «hot» атрибута Temperature.

26. Автозаповніть клітинки H22 до клітинки H31.
Частина вашого аркуша має виглядає так, як рис.15.

	A	B	C	D	E	F	G	H	I	J
21				yes	no	p*log(p)-yes	p*log(p)-r	weighted	entropy	info gain
22	1		hot	2	2	-0.5	-0.5	0.285714		
23	1	Temperature	mild	4	2	-0.389975	-0.5283	0.393555		
24	1		cool	3	1	-0.311278	-0.5	0.231794		
25	2	Humidity	high	3	4	-0.523882	-0.4613	0.492614		
26	2		normal	6	1	-0.190622	-0.4011	0.295836		
27	3	Windy	TRUE	3	3	-0.5	-0.5	0.428571		
28	3		FALSE	6	2	-0.311278	-0.5	0.463587		
29	4		overcast	4	0	0	0	0		
30	4	Outlook	rainy	3	2	-0.442179	-0.5288	0.346768		
31	4		sunny	2	3	-0.528771	-0.4422	0.346768		

Рис.15. Розраховані індивідуальні значення ентропії

27. У клітинку I22 введіть формулу =SUMIFS(H$22:H$31,$A$22:
A$31,A22) і автозаповніть від комірки I22 до I31. Формула в клітинці I22 обчислює ентропію для атрибута Temperature.

28. У комірку J22 введіть формулу =F$18-I22, щоб отримати інформаційний приріст для атрибута Temperature. Автозаповніть з комірки J22 по J31. Порівняйте ваш результат із рис.16.

	D	E	F	G	H	I	J
21	yes	no	p*log(p)-yes	p*log(p)-r	weighted	entropy	info gain
22	2	2	-0.5	-0.5	0.285714	0.911063	0.029223
23	4	2	-0.389975	-0.52832	0.393555	0.911063	0.029223
24	3	1	-0.31127812	-0.5	0.231794	0.911063	0.029223
25	3	4	-0.52388247	-0.46135	0.492614	0.78845	0.151836
26	6	1	-0.19062208	-0.40105	0.295836	0.78845	0.151836
27	3	3	-0.5	-0.5	0.428571	0.892159	0.048127
28	6	2	-0.31127812	-0.5	0.463587	0.892159	0.048127
29	4	0	0	0	0	0.693536	0.24675
30	3	2	-0.44217936	-0.52877	0.346768	0.693536	0.24675
31	2	3	-0.52877124	-0.44218	0.346768	0.693536	0.24675

Рис.16. Розраховано ентропію та приріст інформації

29. Об’єднайте клітинки I22:I24, I25:I26, I27:I28, I29:I31, J22:J24, J25:J26, J27:J28 і J29:J31 відповідно. Обчислення завершено, як показано на Рис.17. Оскільки атрибут Outlook має найбільший приріст інформації, його вибрано для поділу вузла дерева level-1.

	C	D	E	F	G	H	I	J
18	yes	9	-0.40977638	0.94028596				
19	no	5	-0.53050958					
20								
21		yes	no	p*log(p)-yes	p*log(p)-r	weighted	entropy	info gain
22	hot	2	2	-0.5	-0.5	0.285714		
23	mild	4	2	-0.389975	-0.52832	0.393555	0.911063	0.029223
24	cool	3	1	-0.31127812	-0.5	0.231794		
25	high	3	4	-0.52388247	-0.46135	0.492614	0.78845	0.151836
26	normal	6	1	-0.19062208	-0.40105	0.295836		
27	TRUE	3	3	-0.5	-0.5	0.428571	0.892159	0.048127
28	FALSE	6	2	-0.31127812	-0.5	0.463587		
29	overcast	4	0	0	0	0		
30	rainy	3	2	-0.44217936	-0.52877	0.346768	0.693536	0.24675
31	sunny	2	3	-0.52877124	-0.44218	0.346768		

Рис.17. Level-1 розділення буде базуватися на Outlook

30. Ми можемо нарисувати просту «деревоподібну діаграму», як показано на рис.18. Оскільки зважена ентропія H-outlook-overcast дорівнює нулю, дочірній вузол overcast(4,0) є листовим вузлом. Наступне завдання — розділити вузли дощовий (3,2) і сонячний (2,3).

	C	D	E	F	G	H	I	
25	high	3	4	-0.523882466	-0.46135	0.492614068	0.78845	0
26	normal	6	1	-0.190622075	-0.40105	0.295836389		
27	TRUE	3	3	-0.5	-0.5	0.428571429	0.892159	0
28	FALSE	6	2	-0.311278124	-0.5	0.4635875		
29	overcast	4	0	0	0	0		
30	rainy	3	2	-0.442179356	-0.52877	0.346768069	0.693536	(
31	sunny	2	3	-0.528771238	-0.44218	0.346768069		
32								
33								
34				Outlook				
35								
36		rainy(3,2)		overcast(4,0)		sunny(2,3)		
37								
38								

Рис.18. Проста деревоподібна діаграма

Дотримуйтесь цих інструкцій, щоб розділити вузол rainy(3,2):

31. Скопіюйте робочий аркуш level-1 і перейменуйте новий робочий аркуш level-2-rainy.

32. На робочому аркуші level-2-rainy введіть «rainy» у клітинку F1.

33. У комірці B16 введіть формулу = COUNTIFS($D$2:$D$15,$F$1). Будуть враховані лише ті комірки, для яких значення Outlook є «rainy».

Переконайтеся, що ваш робочий аркуш виглядає точно так, як на рис.19.

	A	B	C	D	E	F
1	**Temperature**	**Humidity**	**Windy**	**Outlook**	**Play**	**rainy**
2	hot	high	FALSE	overcast	yes	
3	cool	normal	TRUE	overcast	yes	
4	mild	high	TRUE	overcast	yes	
5	hot	normal	FALSE	overcast	yes	
6	mild	high	FALSE	rainy	yes	
7	cool	normal	FALSE	rainy	yes	
8	cool	normal	TRUE	rainy	no	
9	mild	normal	FALSE	rainy	yes	
10	mild	high	TRUE	rainy	no	
11	hot	high	FALSE	sunny	no	
12	hot	high	TRUE	sunny	no	
13	mild	high	FALSE	sunny	no	
14	cool	normal	FALSE	sunny	yes	
15	mild	normal	TRUE	sunny	yes	
16	Sample-size	5				

Рис.19. Робота над розділенням вузла дерева rainy(3,2)

Продовжуйте, як описано далі, щоб виконати завдання:

34. Комірка D18 містить формулу = COUNTIFS($E$2:$E$15,C18). Вставте «,$D$2:$D$15,$F$1» відразу після «C18» у формулу, щоб формула мала вигляд

=COUNTIFS($E$2:$E$15;C18;$D$2:$D$15;$F$1)

Знову ж таки, це включатиме лише ті клітинки, для яких значення Outlook є «rainy».

35. Автозаповніть клітинки D18 до клітинки D19.

36. Вставте ",$D$2:$D$15,$F$1" у формулу клітинки D22 і переконайтеся, що її формула стає

=COUNTIFS(INDEX($A$2:$E$15,0,$A22),$C22, $E$2:$E$15,D$21,$D$2:$D$15,$F$1)

37. Автозаповніть від D22 до E22, а потім разом автозаповніть клітинки D31:E31.

Частина нашого аркуша має виглядати так, як на рис. 20.

	A	B	C	D	E	F	G	H	I	J
16	Sample-size	5								
17					p*log(p)	entropy				
18		Play	yes	3	-0.44217936	0.970950594				
19			no	2	-0.52877124					
20										
21	1			yes	no	p*log(p)-yes	p*log(p)-r	weighted	entropy	info gain
22	1		hot	0	0	0	0	0		
23	1	Temperature	mild	2	1	-0.389975	-0.52832	0.5509775	0.950978	0.019973
24	1		cool	1	1	-0.5	-0.5	0.4		
25	2	Humidity	high	1	1	-0.5	-0.5	0.4	0.950978	0.019973
26	2		normal	2	1	-0.389975	-0.52832	0.5509775		
27	3	Windy	TRUE	0	2	0	0	0	0	0.970951
28	3		FALSE	3	0	0	0	0		
29	4		overcast	0	0	0	0	0		
30	4	Outlook	rainy	3	2	-0.442179356	-0.52877	0.970950594	0.970951	0
31	4		sunny	0	0	0	0	0		

Рис.20. Усі значення ентропії та інформаційного приросту обчислено для rainy(3,2)

38. Видається, що вузол rainy(3,2) має бути розгалужений на основі Windy. Оскільки обидва дочірні вузли Windy-t(0,2) і Windy-f(3,0) мають нульову ентропію, обидва вони є листовими вузлами. Ми можемо змінити існуючу «діаграму», як на рис.21.

Рис.21. Розділений вузол rainy(3,2) на Windy

Розбити вузол sunny(2,3) досить легко. Дотримуйтеся наступних інструкцій:
39. Зробіть копію аркуша level-2-rainy, перейменуйте новий аркуш level-2-sunny.
40. На робочому аркуші level-2-sunny змініть текст у клітинці F1 на «sunny».
Ось і все, усі обчислення автоматично виконує Excel. Результат виглядає так, як на рис. 22.

	A	B	C	D	E	F	G	H	I	J
16	Sample-size	5								
17					p*log(p)	entropy				
18		Play	yes	2	-0.5287712	0.97095059				
19			no	3	-0.4421794					
20										
21	1			yes	no	p*log(p)-yes	p*log(p)-n	weighted	entropy	info gain
22	1		hot	0	2	0	0	0		
23	1	Temperature	mild	1	1	-0.5	-0.5	0.4	0.4	0.57095
24	1		cool	1	0	0	0	0		
25	2	Humidity	high	0	3	0	0	0	0	0.97095
26	2		normal	2	0	0	0	0		
27	3	Windy	TRUE	1	1	-0.5	-0.5	0.4	0.95098	0.01997
28	3		FALSE	1	2	-0.5283208	-0.39	0.5509775		
29	4		overcast	0	0	0	0	0		
30	4	Outlook	rainy	0	0	0	0	0	0.97095	0
31	4		sunny	2	3	-0.5287712	-0.4422	0.970950594		

Рис.22. Усі значення ентропії та інформаційного приросту, обчислені для sunny(2,3)

З результатів, показаних на робочому аркуші level-2-sunny, можна помітити, що вузол дерева sunny(2,3) має розгалужуватися на основі Humidity. Оскільки обидва згенеровані дочірні вузли мають нульову ентропію, вони обидва є листовими вузлами. Таким чином, більше немає необхідності розщепляти дерево. Результат показано на рис. 23.

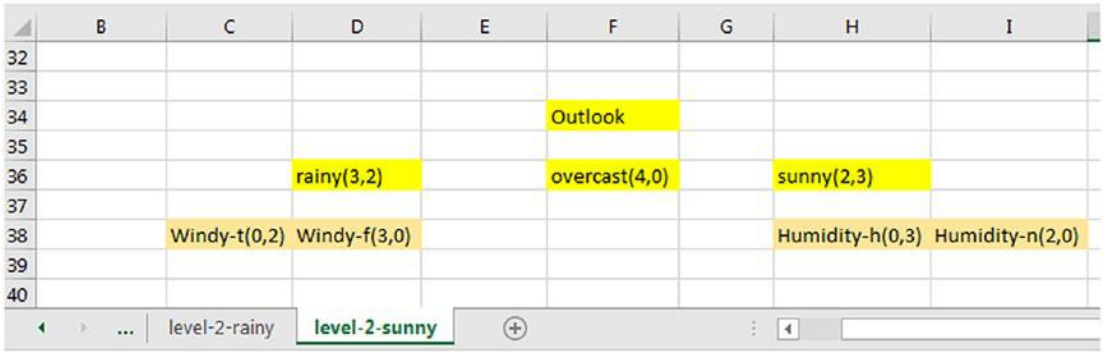

Рис.23. Побудова дерева рішень завершена

Як видно атрибут Temperature не використовувався для розбиття дерева.

3. ЛАБОРАТОРНЕ ЗАВДАННЯ

1. Для індивідуального завдання вирішіть задачу класифікації за допомогою наступного алгоритму:

- метод побудови дерев рішень C4.5 (trees.J48).

2. Змінюючи параметри налаштування алгоритму, спробуйте досягти найвищої якості навчання класифікатора.

3. Для цього ж датасету побудуйте дерево рішень у Excel.

4. Порівняйте отримані результати отримані у різних системах.

5. У звіті надайте результати роботи роботи алгоритму, його налаштування.

4. КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке дерева рішень в машинному навчанні?
2. Які основні етапи побудови дерева рішень?
3. Які алгоритми використовуються для побудови дерев рішень?
4. Як визначити критерій поділу при побудові дерева рішень?
5. Що таке глибина дерева рішень і чому вона важлива?
6. Що таке обрізка дерева і навіщо він використовується?
7. Як використовується дерево рішень для класифікації?
8. Що є перевагами і недоліками використання дерев рішень у порівнянні з іншими алгоритмами машинного навчання?
9. Як оцінюється якість класифікації методом дерев рішень?
10. Як впливає кількість точок даних на якість побудови дерева рішень?
11. Чому важливо враховувати перенавчання при використанні дерев рішень?
12. Яка роль відіграє вибір різних атрибутів при побудові дерева рішень?
13. Що таке ентропія?
14. Що таке індекс Джині? Як він використовується в деревах рішень?

5. ЗМІСТ ЗВІТУ

1. Тема і мета роботи.
2. Завдання до роботи.
3. Результати виконання завдань.
4. Висновки, що відображують результати виконання роботи та їх критичний аналіз.