Національний технічний університет України «КПІ ім. Ігоря Сікорського» Факультет Інформатики та Обчислювальної Техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №4

з дисципліни «Інтелектуальний Аналіз Даних»

на тему

«МЕТОДИ ВИДОБУВАННЯ АСОЦІАТИВНИХ ПРАВИЛ

З ВЕЛИКИХ МАСИВІВ ДАНИХ»

Варіант №12

Виконав:

студент групи ІС-02

Плостак І. М.

Київ – 2021

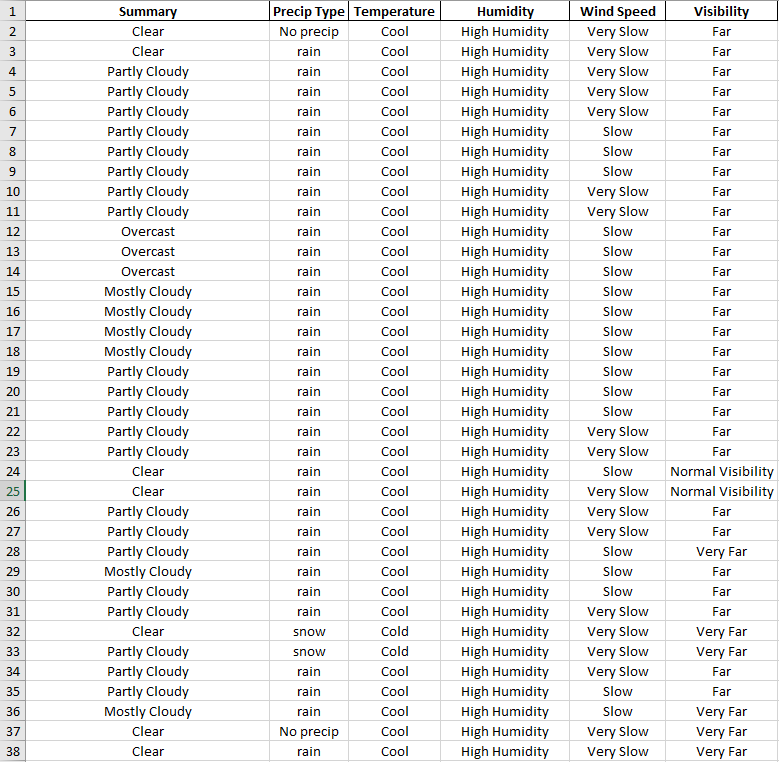
**1. Мета роботи**

Вивчити основні методи видобування асоціативних правил з великих масивів даних, навчитися використовувати спеціалізовані програмні засоби для видобування знань з масивів даних.

**2. Завдання до роботи**

1. Ознайомитися з конспектом лекцій та рекомендованою літературою за темою роботи, а також з додатком А, що містить опис програмного забезпечення для видобування асоціативних правил з великих масивів даних.
2. Сформувати набір даних для обробки та аналізу.
3. Розробити за допомогою середовища Matlab програмне забезпечення для видобування асоціативних правил з великих масивів даних або вивчити рекомендоване програмне забезпечення (пакет Armada модулю Matlab) та здійснити обробку набору даних з метою виділення асоціативних правил.
4. Оформити звіт з роботи.
5. Відповісти на контрольні питання.

**3. Набір даних для обробки (частина)**

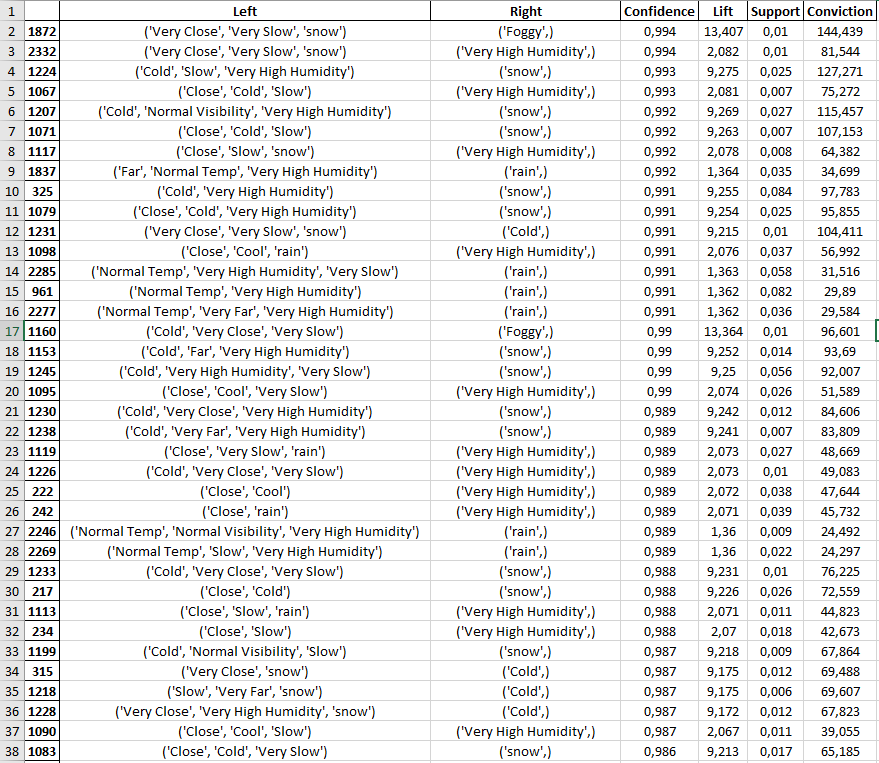


**4. Лістинг основних функцій програми**

main.py:

import pandas as pd  
from efficient\_apriori import apriori  
  
def searchWhereFrom(dataW, dataF):  
 for i in dataF:  
 if i in dataW:  
 return True  
 return False  
  
dataset = pd.read\_excel('weatherHistoryFinal.xlsx')  
dataset.head()  
  
shape1 = len(dataset)  
shape2 = len(dataset.columns)  
transactions = []  
for i in range(shape1):  
 transactions.append([str(dataset.values[i, j]) for j in range(shape2)])  
  
itemsets, rules = apriori(transactions, min\_support = 0.003, min\_confidence = 0.2, max\_length=4)  
  
listRules = []  
for rule in rules:  
 toAdd = [rule.lhs,  
 rule.rhs,  
 round(rule.confidence, 3),  
 round(rule.lift, 3),  
 round(rule.support, 3),  
 round(rule.conviction, 3)]  
 if len(rule.rhs) == 1 and searchWhereFrom(list(dataset['Summary']), list(rule.lhs)) == False:  
 listRules.append(toAdd)  
  
pdRules = pd.DataFrame(listRules)  
pdRules.columns = ["Left", "Right", "Confidence", "Lift", "Support", "Conviction"]  
pdRules = pdRules.sort\_values(['Confidence', 'Lift'], ascending=[False, False])  
pdRules.to\_excel("Rules.xlsx")

**5. Результати роботи програмного забезпечення (частина)**



**6.1. Контрольні питання**

1. Що таке асоціативне правило?

*Асоціативні правила – це відносини між елементами датасету у вигляді: якщо <умова> то <результат>.*

1. Для чого призначені асоціативні правила?

*Асоціативні правила дозволяють знаходити закономірності між пов'язаними подіями.*

1. Дати визначення понять підтримки та достовірності правила.

*Підтримка (support):*

**

*, де X - itemset, що містить у собі i-items, а T - кількість транзакцій. Тобто. Загалом це показник «частотності» даного itemset у всіх аналізованих транзакціях.*

*Коли у нас в одному itemset зустрічаються x1 та x2, то* *нам треба порахувати, у скількох транзакціях зустрічається ця парочка. Формула для більш ніж одного елементу:*

**

*, де σ - кількість транзакцій, що містять x1 і x2.*

1. Яке призначення алгоритмів пошуку асоціативних правил?

*Алгоритми пошуку асоціативних правил відбирають тих кандидатів, у яких підтримка і достовірність вище за деякі наперед задані пороги: minsupport і minconfidence.*

1. На які підзадачі розбивається задача знаходження асоціативних правил?

Задача знаходження асоціативних правил розбивається на дві підзадачі:

* *Знаходження всіх наборів елементів, які задовольняють поріг minsupport. Такі набори елементів називаються такими, що часто зустрічаються.*
* *Генерація правил наборів елементів, знайдених згідно з п.1. з достовірністю, що задовольняє поріг minconfidence.*

1. Які методи використовуються для знаходження асоціативних правил?

*Найпопулярнішими методами знаходження асоціативних правил є алгоритми, наприклад Брутфорс, Апріорі, ECLAT, FP та інші.*

1. Яким чином обираються значення параметрів minsupport та minconfidence?

*Значення цих параметрів обираються таким чином, щоб обмежити кількість знайдених правил. Якщо підтримка має велике значення, то алгоритми будуть знаходити правила, вже відомі аналітикам або настільки очевидні, що немає сенсу проводити аналіз.*

1. Що таке числові асоціативні правила?

*Числові асоціативні правила – правила, які на відміну від звичайних правил «був – не був», ґрунтуються на де-якому числовому діапазоні значень об’єкту датасету, наприклад: [Вік: 30-35] та [Сімейний стан: одружений] -> [Дохід: 1000-1500 гривень].*

1. Поясніть поняття “узагальнене асоціативне правило”.

*Узагальненим асоціативним правилом називається імплікація X⇒Y, де X⊂I, Y⊂I і X∩Y=⊘ і де жоден з елементів, що входять до набору Y, не є предком жодного елемента, що входить у X.*

1. Що називається ієрархією елементів?

*Ієрархією елементів називається ліс спрямованих ациклічних дерев, листя яких є елементи транзакцій, а внутрішніми вузлами — групи елементів.*

1. Які переваги дає введення додаткової інформації про групування елементів?

* *Будуть виявлені асоціативні правила як між окремими елементами, так і між різними рівнями ієрархії.*
* *У деяких випадках окремі елементи можуть мати дуже маленьку підтримку, проте значення підтримки всієї групи, в яку входить цей елемент, може бути більше порога мінімальної підтримки. Це призводить до того, що раніше не виявлене потенційно цікаве правило, побудоване на елементах нижнього рівня ієрархії, може бути отримане, але його елементами будуть елементи транзакції, або предки цих елементів.*
* *Введення інформації про групування елементів може використовуватися для відсікання «нецікавих» правил.*

1. Поясніть, які проблеми можуть виникнути при безпосередньому застосуванні алгоритмів знаходження асоціативних правил.

* *Елементи на верхніх рівнях ієрархії прагнуть значно більших значень підтримки в порівнянні з елементами на нижніх рівнях.*
* *З додаванням у транзакції груп збільшилася кількість атрибутів та відповідно розмірність вхідного простору. Це ускладнює завдання, а також веде до створення більшої кількості правил.*
* *Поява надлишкових правил, що суперечать визначенню узагальненого асоціативного правила. Очевидно, що практична цінність такого «відкриття» є нульовою при 100% достовірності. Отже, потрібні спеціальні оператори, які видаляють такі надлишкові правила.*

1. В чому полягає сутність виявлення узагальнених асоціативних правил?

*При пошуку асоціативних правил ми припускали, що це аналізовані елементи однорідні. Повертаючись до аналізу ринкового кошика, це товари, які мають абсолютно однакові атрибути, крім назви. Проте не важко доповнити транзакцію інформацією про те, в яку товарну групу входить товар і побудувати ієрархію товарів.*

1. Яким чином визначають “цікаві” правила? В чому полягає актуальність такого процесу?

*Правило X⇒Y називається R-цікавим щодо правила-пращура, якщо підтримка правила X⇒Y у R разів більша за очікувану підтримку правила X⇒Y щодо правила-пращура або якщо достовірність правила X⇒Y у R разів більша за очікувану достовірність правила X⇒Y щодо правила-пращура.*

1. Дати визначення понять батьківського правила (пращура) та найближчого батьківського правила.

*Називатимемо x - пращуром x і x – нащадком x.*

*Правило X ⇒ Y є найближчим предком правила X⇒Y, якщо не існує такого правила X'⇒Y', що X'⇒Y' - це пращур X⇒ і X ⇒ Y - це пращур X'⇒Y'.*

1. Порівняйте поняття цікавого та частково цікавого правила.

*Правило називається цікавим, якщо він не має пращурів або воно є R-цікавим щодо всіх своїх найближчих пращурів.*

*Правило називається частково цікавим, якщо він не має пращурів або воно є R-цікавим щодо будь-якого свого найближчого пращура.*

1. Які проблеми усуває алгоритм обчислення узагальнених асоціативних правил?

* *Розмір кожної транзакції збільшується залежно від глибини дерева від кількох десятків відсотків за кілька разів. Як наслідок, час обчислення та кількість правил збільшуються.*
* *Поява надлишкових правил, у яких за умови і у слідстві перебувають елемент та її пращур. Прикладом такого правила може бути «Якщо покупець купив Кефір, то він, швидше за все, захоче купити Молочні продукти». Цілком ясно, що практична цінність цього правила дорівнює нулю при достовірності 100%.*

1. З яких етапів складається процес обчислення узагальнених асоціативних правил?

* *Пошук часто зустрічаються безлічі елементів, підтримка яких більше, ніж заданий поріг підтримки (мінімальна підтримка).*
* *Обчислення правил на основі знайдених на попередньому етапі множин елементів, що часто зустрічаються. Основна ідея обчислення правил на основі множин, що часто зустрічаються, полягає в наступному: якщо ABCD — це часто зустрічається безліч елементів, то на основі цієї множини можна побудувати правила X⇒Y (наприклад, AB⇒CD), причому X∪Y=ABCD. Підтримка правила дорівнює підтримці множини, що часто зустрічається. Достовірність правила обчислюється за формулою conf(X⇒Y) = supp(X⇒Y)/supp(X). Правило додається до результуючого списку правил, якщо достовірність цього правила більша за поріг minconf.*
* *З результату списку правил видаляються всі «нецікаві» правила.*

1. Проаналізуйте базовий алгоритм пошуку множин, що зустрічаються часто.

* На першому етапі алгоритму підраховуються 1-елементні набори, що часто зустрічаються. При цьому елементи можуть бути на будь-якому рівні таксономії. І тому необхідно «пройтися» у всьому наборі даних, і підрахувати їм підтримку, тобто. скільки разів зустрічається у основі.
* Наступні кроки будуть складатися з двох частин: генерації потенційно часто зустрічаються наборів елементів (їх називають кандидатами) та підрахунку підтримки для кандидатів.

1. Опишіть алгоритм генерації кандидатів.

* *Об'єднання. Кожен кандидат Ck буде формуватися шляхом розширення набору (k−1), що часто зустрічається, додаванням елемента з іншого (k−1)-елементного набору.*
* *Видалення надлишкових правил. На підставі властивості анти-монотонності, слід видалити всі набори c Ck, якщо хоча б одне з його (k−1) підмножин не часто зустрічається.*

1. Яким чином використовується хеш-дерево для підрахунку підтримки кандидатів? Як відбувається процес побудови такого дерева?

*Хеш-дерево будується щоразу, коли формуються кандидати. Спочатку дерево складається тільки з кореня, яке є листом, і не містить жодних кандидатів-наборів.*

*Щоразу, коли формується новий кандидат, він заноситься в корінь дерева, і так доти, поки кількість кандидатів у корені листі не перевищить якогось порога.*

*Щойно кількість кандидатів стає більше порога, корінь перетворюється на хеш-таблицю, тобто. стає внутрішнім вузлом, і йому створюються нащадки-листя. І всі приклади розподіляються по вузлам-нащадкам відповідно до хеш-значення елементів, що входять в набір тощо. Кожен новий кандидат хешується на внутрішніх вузлах, поки він не досягне першого вузла-листа, де він і зберігатиметься, поки кількість наборів знову ж таки не перевищить порога.*

1. Виконайте порівняльний аналіз базового та покращеного алгоритмів пошуку множин, що зустрічаються часто.

*Ці два алгоритми відрізняються оптимізаціями, які введені у покращений алгоритм.*

1. За рахунок яких оптимізацій відбувається покращення базового алгоритму пошуку множин, що зустрічаються часто?

* *Доцільно один раз обчислити безліч предків для кожного елемента ієрархії: як для елемента нижнього рівня таксономії (аркуш дерева), так і для елемента внутрішнього рівня.*
* *Необхідно видаляти кандидати, які містять елемент та його предок.*
* *Немає необхідності додавати всі предки всіх елементів, що входять в транзакцію. Якщо якийсь елемент, який має предок, немає у списку кандидатів, то списку елементів з предками він позначається як віддалений. Отже, з транзакції видаляються елементи, позначені як віддалені, або проводиться заміна цих елементів з їхньої предків. До транзакції додаються лише видалені предки.*
* *Не «пропускати» транзакцію через хеш-дерево, якщо її потужність менша за потужність елементів, розташованих у хеш-дереві.*
* *Доцільно помічати транзакції як видалені та не використовувати їх при підрахунку підтримки на наступних ітераціях, якщо на поточну ітерацію до цієї транзакції не увійшов жоден кандидат.*

1. В чому полягає сутність масштабованого алгоритму пошуку асоціативних правил Apriori?

*Завдяки використанню властивості антимонотонності, він здатний обробляти великі обсяги даних за прийнятний час.*

1. Яким чином перетворюються дані для можливості використання алгоритму Apriori?

* *привести всі дані до бінарного вигляду;*
* *змінити структуру даних.*

1. Яка властивість використовується в алгоритмі Apriori? Для чого вона використовується?

*Ця властивість зветься анти-монотонності і служить зниження розмірності простору пошуку.*

*Властивості анти-монотонності можна дати і інше формулювання: зі зростанням розміру набору елементів підтримка зменшується, або залишається такою ж.*

1. Наведіть послідовність виконання алгоритму Apriori.

* *На першому етапі алгоритму підраховуються 1-елементні набори, що часто зустрічаються. І тому необхідно пройтися у всьому наборі даних, і підрахувати їм підтримку, тобто. скільки разів зустрічається у основі.*
* *Наступні кроки будуть складатися з двох частин: генерації потенційно часто зустрічаються наборів елементів (їх називають кандидатами) та підрахунку підтримки для кандидатів.*

1. Опишіть функцію генерації кандидатів в алгоритмі Apriori.

* *Об'єднання. Кожен кандидат Ck буде формуватися шляхом розширення набору (k−1), що часто зустрічається, додаванням елемента з іншого (k−1)-елементного набору.*
* *Видалення надлишкових правил. На підставі властивості анти-монотонності, слід видалити всі набори c Ck, якщо хоча б одне з його (k−1) підмножин не часто зустрічається.*

1. Як відбувається підрахунок підтримки для кожного кандидату в алгоритмі Apriori? Для чого в цій процедурі використовують хеш-дерево?

*Після створення кандидатів наступним завданням є підрахунок підтримки для кожного кандидата. Очевидно, що кількість кандидатів може бути дуже великою і потрібний ефективний спосіб підрахунку.*

*Використовується підхід, що базується на зберіганні кандидатів у хеш-дереві. Внутрішні вузли дерева містять хеш-таблиці із вказівниками на нащадків, а листя – на кандидатів. Це дерево стане нам у нагоді для швидкого підрахунку підтримки для кандидатів.*

1. Як здійснити добування правил з набору, що часто зустрічається?

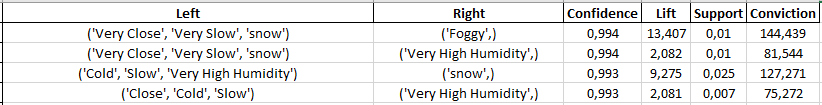
*Щоб отримати правило з набору F, що часто зустрічається, слід знайти всі його непусті підмножини. І для кожного підмножини s ми зможемо сформулювати правило s ⇒ (F − s), якщо достовірність правила conf(s ⇒(F−s))=supp(F)/supp(s) не менше порога minconf.*

**6.2. Висновки**

В результаті виконання лабораторної роботи ми ознайомилися з методами видобування асоціативних правил з великих масивів даних. Був обраний алгоритм Apriori через легкість його застосування та наявність реалізованої бібліотеки на мові Python.

Був узятий датасет даних про погоду за 2006-2016 роки: загальний опис погоди, тип опадів, температура, вологість, швидкість вітру та дальність видимості. На основі 95 тисяч записівбув проведений аналіз та знайдені асоціативні правила для цього датасету.

Наприклад:



На цьому фрагменті даних можна побачити правила, що найчастіше виконуються при максимальному розмірі ітемсету у 4 елементи.

Найкращу з них залежність має верхнє асоціативне правило. При відносній ймовірності 0,99 ці два набори подій досить сильно залежні один від одного, оскільки значення Lift досягає 13,4. Тобто ми впевнено можемо сказати, що «При дуже недалекій видимості, повільному вітрі та під час снігу буде туманна погода».

У свою чергу найнижче асоціативне правило має меншу «силу», оскільки значення Lift досягає усього лише 2. Так, це теж відносно велике значення параметру залежності подій, але якщо ми спостерігаємо «Недалеку видимість, холод та повільний вітер» то ймовірність спостерігати «дуже сильну вологість» велика, що цілком логічно, але наслідок не так жорстко прив’язаний до причини виникнення, як у попередньому судженні.