МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені І. І. МЕЧНИКОВА
ФАКУЛЬТЕТ МАТЕМАТИКИ, ФІЗИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МЕТОДИ КЛАСИФІКАЦІЇ І КЛАСТЕРИЗАЦІЇ ДАНИХ

НАВЧАЛЬНО-МЕТОДИЧНИЙ ПОСІБНИК

для здобувачів факультету математики, фізики
та інформаційних технологій
спеціальності F7/123 Комп’ютерна інженерія

Одеса
ОНУ імені І. І. Мечникова
2026

УДК 004.275(076)
М545

Укладачі:
В. С. Михайленко, доктор технічних наук, професор кафедри комп'ютерних систем та технологій;
Ю. О. Гунченко, доктор технічних наук, професор кафедри комп'ютерних систем та технологій;
Л. Я. Мартинович, старший викладач кафедри комп'ютерних систем та технологій;
А. В. Камєнєва, кандидат технічних наук, доцент кафедри комп'ютерних систем та технологій.

Рецензенти:
В. Ф. Ложечніков, кандидат технічних наук, доцент кафедри комп'ютеризованих систем та програмних технологій Національного університету «Одеська політехніка»;
Ю. А. Ніцук, доктор фізико-математичних наук, професор, декан факультету математики, фізики та інформаційних технологій ОНУ імені І. І. Мечникова.

Рекомендовано до видання науково-методичною радою
ОНУ імені І. І. Мечникова.
Протокол № 2 від 16 квітня 2026 р.

Методи класифікації і кластеризації даних [Електронний ресурс] :
М545 навч.-метод. посіб. для здобувачів ф-ту математики, фізики та інформ. технологій спец. F7/123 Комп'ютерна інженерія / уклад.: В. С. Михайленко, Ю. О. Гунченко, Л. Я. Мартинович, А. В. Камєнєва. Електронні текстові дані (1 файл: 3,4 МБ). Одеса : ОНУ імені І. І. Мечникова, 2026. 98 с.
ISBN 978-966-186-392-6

Посібник містить теорію та практичні вказівки до лабораторних робіт із курсу «Методи класифікації і кластеризації даних». Розглянуто реалізацію алгоритмів та візуалізацію результатів у пакеті Orange. Описано порядок виконання робіт, вимоги до оформлення протоколів, правила захисту та критерії оцінювання. Додано перелік завдань, контрольні запитання та список рекомендованої літератури.

УДК 004.275(076)

ISBN 978-966-186-392-6

ЗМІСТ

ВСТУП

Все різноманіття даних, які вимагають аналізу чи будь-якої обробки, зорганізується як упорядкована система. Для цієї системи розробляються певні правила розподілу і кожному об'єкту (групи даних, які досліджуються) відведено певне місце згідно з ознаками. Термін «класифікація» (від латин. classic – розряд, група) означає систему впорядкованого розподілу безлічі об'єктів у логічній послідовності з підпорядкуванням на основі певних ознак. Отже, методологія класифікації є процесом розподілу безлічі об'єктів за найбільш загальними ознаками і правилами на певні підмножини [1]. Систему, що утворюється, називають класифікацією (системою класифікації). Дисципліна пов’язана з технічною складовою сучасних систем класифікації та кластерного аналізу даних. Вивчення дисципліни спрямоване на формування елементів наступних компетентностей:

- здатність застосовувати сучасні інформаційні технології, бази даних та інші електронні ресурси, спеціалізоване програмне забезпечення у науковій та навчальній діяльності;
- здатність будувати математичні, інформаційні, структурні та інші моделі для опису і подальшого вивчення процесів чи об’єктів предметних областей;
- здатність моделювати складні системи, зокрема, інформаційні та технічні системи різного призначення, інтелектуальні системи та системи підтримки прийняття рішень.

Таким чином, **метою дисципліни** є вивчення методів та алгоритмів класифікації та кластерного аналізу для створення програмного продукту та ефективного використання у практиці.

Методичні вказівки висвітлюють 10 лабораторних робіт з методів кластеризації, аналізу тексту, класифікації даних, прогнозування та містять теоретичні відомості з курсу, що дозволяє студентам глибше зрозуміти суть алгоритмів класифікації даних, інтерпретувати отримані результати, відповісти на контрольні питання та сформулювати висновок. Лабораторні роботи виконуються з метою закріплення та поглиблення теоретичних та практичних знань та вмінь, набутих у процесі засвоєння всього навчального матеріалу. Їх виконання є важливим етапом у підготовці до виконання дипломного проєкту (роботи) майбутнього фахівця з комп’ютерної інженерії.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Класифікація та кластеризація – це два «кити» машинного навчання, які допомагають упорядковувати дані. Хоча їх часто плутають, та між ними є фундаментальна різниця: **класифікація** знає, що вона шукає (навчання з учителем), а **кластеризація** намагається знайти структуру в невідомому (навчання без учителя).

Класифікація (Supervised Learning)

У цьому методі алгоритм навчається на розмічених даних. У нього є «вчитель» у вигляді правильних відповідей.

Популярні методи:

- **Логістична регресія:** Попри назву, використовується для класифікації (наприклад, «спам» чи «не спам»). Вона передбачає ймовірність належності до класу.
- **Метод k-найближчих сусідів (k-NN):** Об'єкт присвоюється до того класу, який є найпоширенішим серед його найближчих «сусідів» у просторі ознак.
- **Дерева рішень (Decision Trees):** Будується структура, схожа на дерево, де кожен вузол – це запитання про ознаку (наприклад, «Вік > 18?»), а «листя» – кінцевий клас.
- **Метод опорних векторів (SVM):** Алгоритм шукає гіперплощину, яка найкраще розділяє два класи з найбільшим відступом між ними.
- **Випадковий ліс (Random Forest):** Ансамбль з багатьох дерев рішень, що працюють разом для підвищення точності.

Кластеризація (Unsupervised Learning)

Тут немає міток або правильних відповідей. Алгоритм самостійно групує об'єкти за принципом «схожі з подібними».

Популярні методи:

- **Метод k-середніх (k-Means):** Розділяє дані на заздалегідь визначену кількість груп (). Він ітеративно обчислює центри (центроїди) і приєднує до них найближчі точки.
- **Ієрархічна кластеризація:** Будує деревоподібну структуру (дендрограму), послідовно об'єднуючи дрібні кластери у великі або навпаки.
- **DBSCAN:** Кластеризація на основі щільності. Вона чудово знаходить кластери довільної форми та ефективно відсікає «шум» (аномалії).
- **Гауссові суміші (GMM):** Припускає, що дані складаються з декількох розподілів Гаусса, що дозволяє одному об'єкту належати до кластера з певною ймовірністю.

Порівняльна таблиця

Характеристика	Класифікація	Кластеризація
Тип навчання	З учителем (Supervised)	Без учителя (Unsupervised)
Мета	Передбачити мітку класу	Знайти приховані структури
Вхідні дані	Розмічені (є відповіді)	Нерозмічені (тільки ознаки)
Приклад	Розпізнавання облич, медична діагностика	Сегментація клієнтів, стиснення зображень

Коли що обирати?

- Якщо ви хочете, щоб система навчилася відрізняти «яблука» від «груш» на основі вашого досвіду – це **класифікація**.
- Якщо у вас є кошик із невідомими фруктами, і ви хочете розкласти їх на купки за схожістю (колір, форма, розмір) – це **кластеризація.**

Детально про Кластеризацію

Метод k-середніх (k-Means)

Найпопулярніший алгоритм для швидкого групування.

1. Ви випадково ставите точки (центроїди).
2. Кожна точка даних «приписується» до найближчого центроїда.
3. Центроїд переміщується в центр своєї нової групи.
4. Процес повторюється, доки центроїди не перестануть рухатися.

- **Нюанс:** Ви повинні заздалегідь знати, на скільки кластерів () ділити дані.

DBSCAN (Density-Based Spatial Clustering)

Алгоритм, що працює як людське око: бачить «згустки» даних.

- **Як це працює:** Він шукає зони, де точки розташовані щільно. Якщо точка має достатньо сусідів поруч, вона стає частиною кластера. Якщо точка самотня в порожньому просторі – алгоритм маркує її як **шум/аномалію.**
- **Плюс:** Не треба вказувати кількість кластерів; знаходить групи будь-якої химерної форми (дуги, кільця тощо).

Ієрархічна кластеризація (Agglomerative)

Схожа на створення генеалогічного дерева.

- **Як це працює:** Спочатку кожна точка – це окремий кластер. Потім дві найближчі точки об'єднуються. Процес триває, доки всі точки не зберуться в один гігантський кластер. Результат візуалізується через **дендрограму.**
- **Плюс:** Ви можете «відрізати» дерево на будь-якому рівні, щоб отримати потрібну кількість груп.

Короткий підсумок для вибору:

- Мало даних і потрібна простота? – **k-NN.**
- Потрібна максимальна точність на складних даних? – **Random Forest.**
- Треба знайти аномалії (фрод, помилки)? – **DBSCAN.**

- Потрібно поділити клієнтів на сегменти? – **k-Means.**

Детально про Класифікацію

Метод k-найближчих сусідів (k-NN)

Це найбільш інтуїтивний алгоритм. Його логіка: «Скажи мені, хто твій друг, і я скажу, хто ти».

- **Як це працює:** Коли з'являється нова точка, алгоритм шукає найближчих до неї точок у просторі. Якщо серед 5 сусідів () троє належать до класу «А», а двоє до «Б», нова точка позначається як «А».
- **Плюс:** Простота.
- **Мінус:** Дуже повільний на великих даних, бо щоразу рахує відстані до всіх точок.

Метод опорних векторів (SVM)

Мета SVM – знайти не просто лінію розмежування, а «найширшу дорогу».

- **Як це працює:** Алгоритм шукає гіперплощину, яка розділяє класи з максимальним відступом. Точки, що лежать на межах цього відступу, називаються **опорними векторами**. Якщо дані не можна розділити прямою, SVM використовує «kernel trick» (ядерний трюк), переносячи дані у вищий вимір, де їх можна розрізати.
- **Плюс:** Дуже точний у задачах з чіткими межами (наприклад, розпізнавання тексту).

Випадковий ліс (Random Forest)

Це «демократія» в машинному навчанні.

- **Як це працює:** Ми будуємо сотні дерев рішень. Кожне дерево навчається на випадковій частині даних. Коли приходить новий об'єкт, кожне дерево «голосує» за свій клас. Перемагає той варіант, який набрав більшість голосів.
- **Плюс:** Надійна стійкість до перенавчання.

Аналітична система Orange – це програма з відкритим вихідним кодом для машинного навчання та візуалізації даних, що має великий набір дослідницьких функцій. Програмний продукт Orange (укр. Оранж), що розробляється Лабораторією біоінформатики Люблянського університету, призначена для інтелектуального аналізу даних (ІАД), статистичних досліджень та візуалізації даних. Компоненти аналітичної платформи називаються віджетами, і вони варіюються від мінімалістичної візуалізації даних, вибору підмножин та попередньої обробки до емпіричної оцінки алгоритмів навчання та прогностичного моделювання. Система стане ефективним інструментом у руках аналітика даних, дослідника та вченого.

У програмному забезпеченні Orange Data Mining використовується візуальне програмування, яке реалізується зручним графічним інтерфейсом.

В рамках візуального програмування аналітичні процедури створюються шляхом зв'язування зумовлених або розроблених користувачем блоків (віджетів), у той час як просунуті користувачі можуть використовувати Orange як програмну бібліотеку Python для маніпулювання даними та створення нових блоків (віджетів).

МЕТА, ЕТАПИ ПРОВЕДЕННЯ ТА ЗАХИСТ ЛАБОРАТОРНИХ РОБІТ

Лабораторні роботи з дисципліни виконуються з метою закріплення та поглиблення теоретичних та практичних знань та вмінь, набутих у процесі засвоєння всього навчального матеріалу дисципліни:

- закріплення, поглиблення та узагальнення теоретичних знань і розвиток навичок їх практичного застосування в галузі розробки нечітких експертних систем (НЕС);
- самостійне розв'язання задач проєктування та розробки НЕС;
- уміння користуватися відповідною довідковою літературою, програмними засобами.

Проведення лабораторних робіт містить такі етапи:

- визначення теми, завдання і повторення теоретичного матеріалу;
- безпосереднє виконання роботи;
- оформлення пояснювальної записки;
- захист.

Після виконання лабораторної роботи і вирішення всіх поставлених у ній задач студент оформлює звіт з лабораторної роботи – протокол. Виконаний протокол студент підписує і після дозволу керівника він допускається до захисту. Якщо керівник не допускає студента до захисту, то це питання обговорюється на засіданні кафедри у його присутності.

Захист лабораторної роботи – це форма перевірки якості виконання програми та знань, отриманих під час виконання лабораторних робіт та на лекціях.

Під час захисту студент робить доповідь по суті програми та відповідає на запитання.

Якість протоколу та його захист оцінюється в балах (0–5), за шкалою ECTS (A, B, C, D, E, FX, F) та за національною шкалою «відмінно», «добре», «задовільно», «незадовільно».

ЛАБОРАТОРНА РОБОТА № 1

Тема: Класифікація експериментальних даних на основі регресійного аналізу

Постановка задачі: проаналізувати експериментальні дані про поломки паливних кілець космічного корабля та застосувати регресійний аналіз для класифікації і прогнозування ймовірності поломок паливних кілець під час запуску шатла Челленджер.

Зміст протоколу:

1. Постановка задачі
2. Хід роботи (текст програми або скріншоти інтерфейсу)
3. Висновки щодо роботи
4. Відповіді на контрольні запитання

Хід роботи (приклад виконання):

Починаємо із включення файлу до завдання challenger-data.csv(рис. 1.1). Основна мета полягає в класифікації несправних елементів та прогнозуванні ймовірності виникнення несправностей у паливній системі космічного корабля Челенджер. Для цього буде використовуватися змінна Y, яку будемо визначати як target (мета). Зазначимо для класифікації, що значення 0 вказує на те, що паливне кільце шатла не має поломок, тоді як значення 1 вказує на поломку.

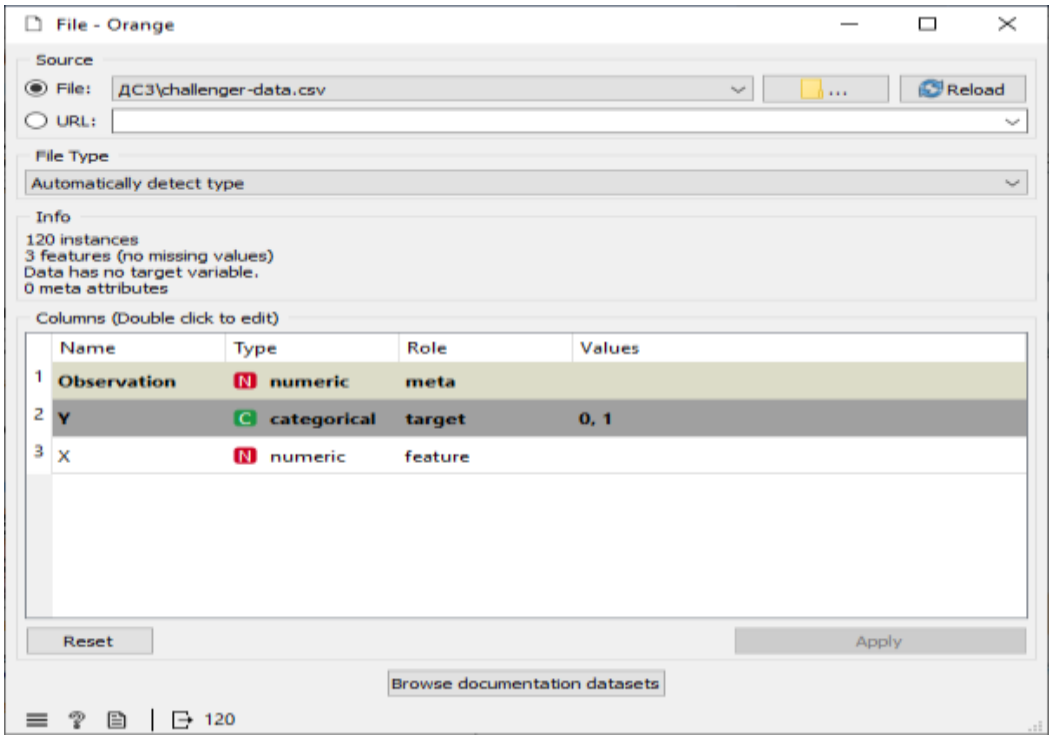


Рис. 1.1. Включення файлу з експериментальними даними

Підключимо до File в наш проєкт відмет Data Table для зручного відображення та аналізу наявних даних (рис. 1.2).

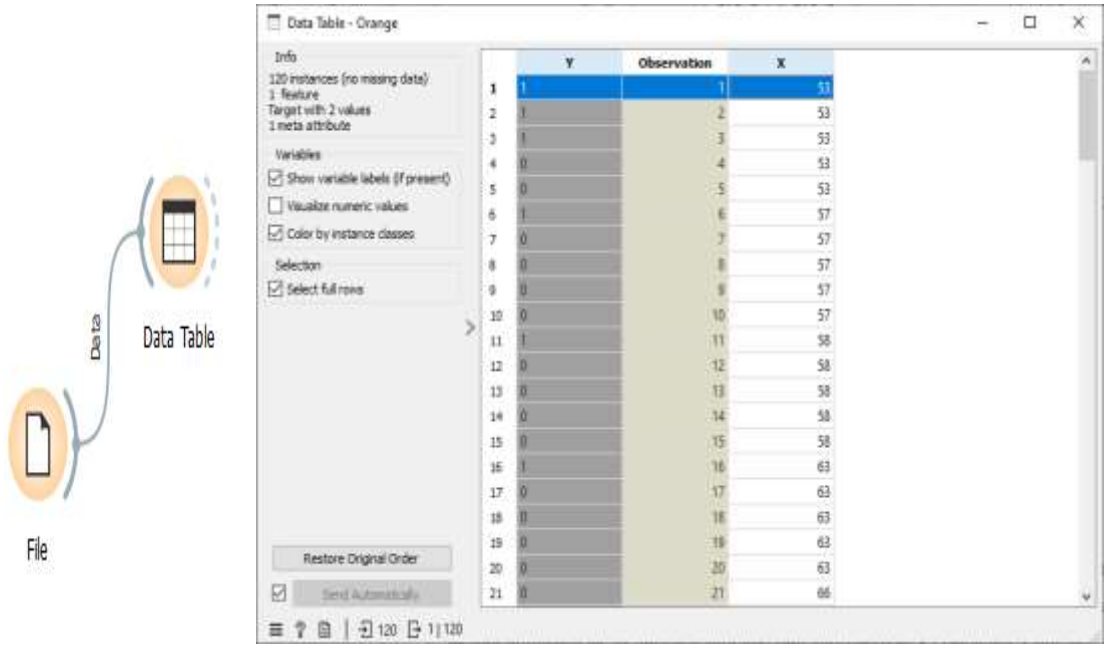


Рис. 1.2. Використання віджету Data table

Перші п’ять рядків (рис. 1.2.) відображають результати першого запуску, де вказано, які кільця виявились пошкодженими при температурі 53 ^{0}C. Наступні п’ять рядків відносяться до другого запуску при температурі 57 градусів, і так далі.

Додаємо новий модуль Distributions для візуалізації та класифікації кілець по поломкам, що дозволить нам спостерігати, що більшість кілець залишилися незіпсованими. Дивимось по Y (рис. 1.3).

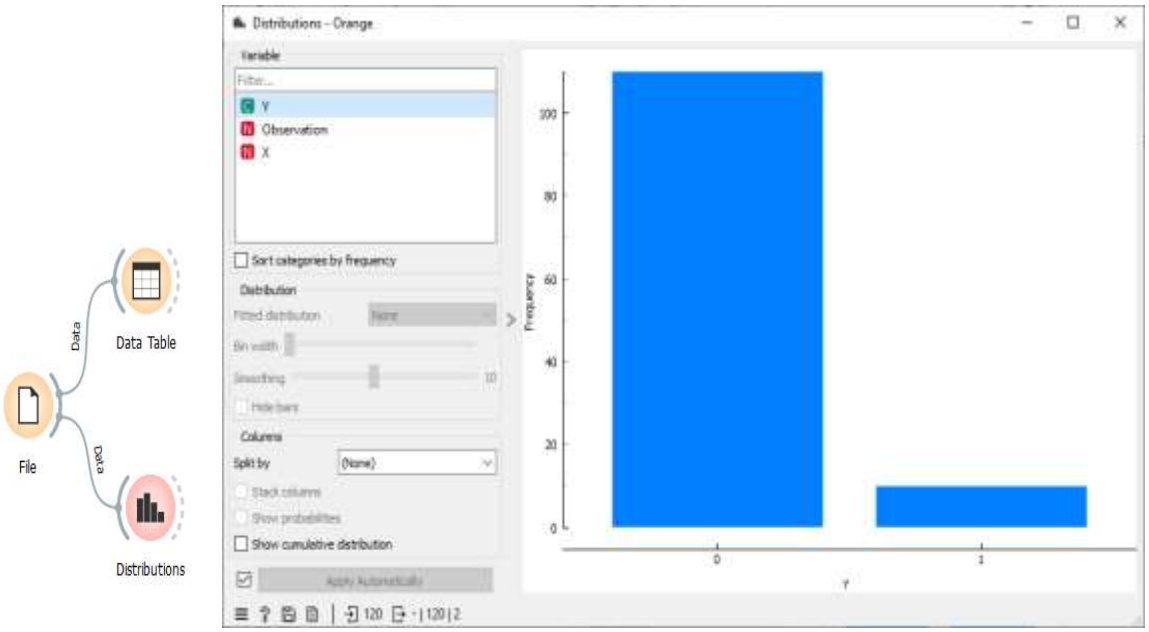


Рис. 1.3. Використання модуля Distributions

Також можна відзначити, що попередні польоти відбувались у температурному діапазоні від 50 до 90 ^{0}F (градусів за Фаренгейтом). Дивимось по X (рис. 1.4).

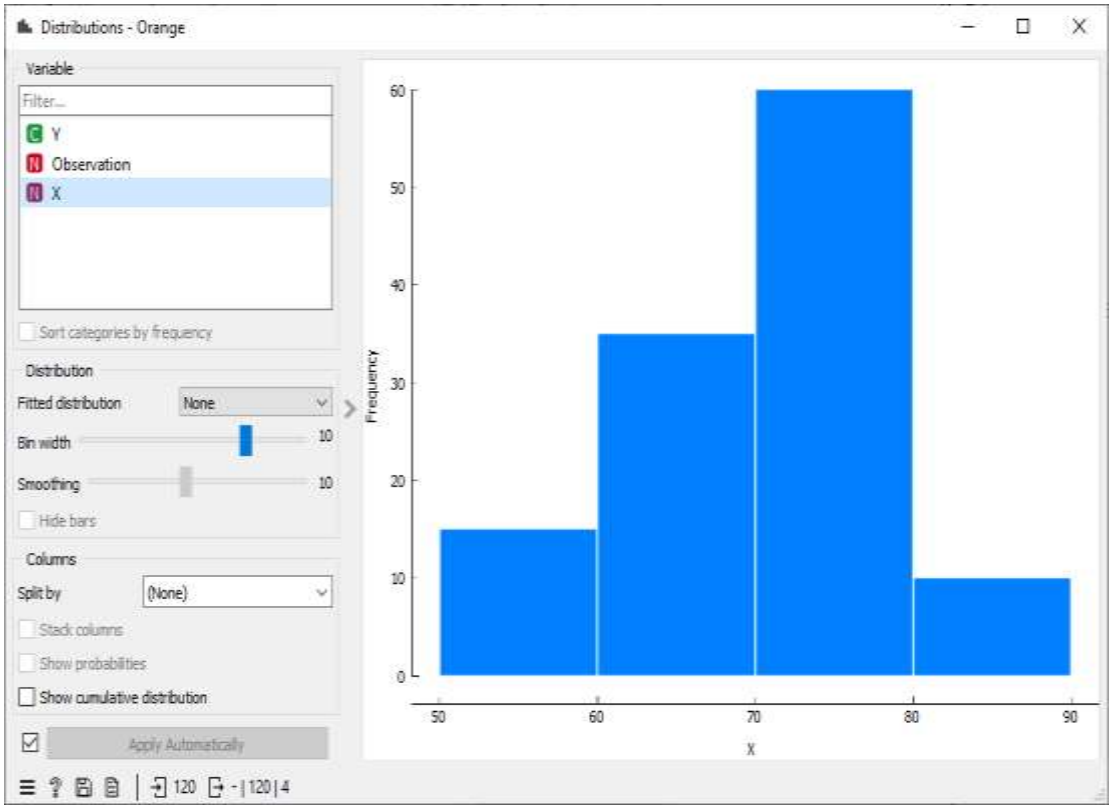


Рис. 1.4. Температурний режим попередніх запусків

Щоб вивчити, як змінюється ймовірність поломок в залежності від температури, ми можемо розподілити температуру за категоріями поломок (рис. 1.5).

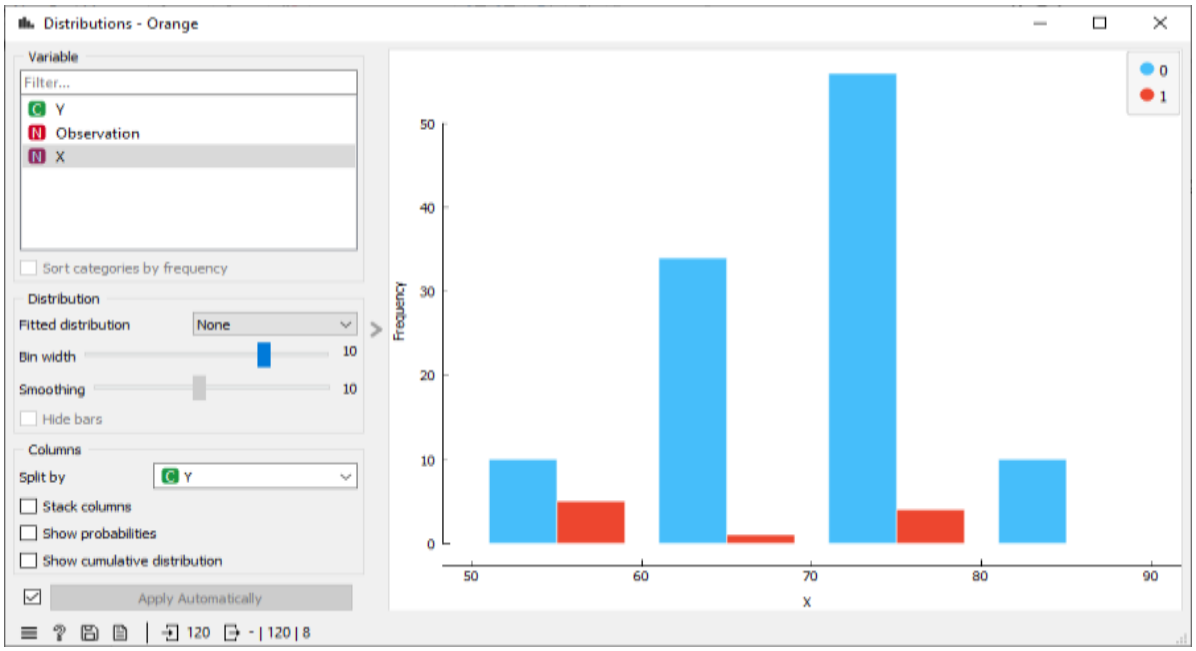


Рис.1.5. Класифікація поломок кілець при різних температурах

Під’єднаємо один до одного модулі Logistic Regression і Test and Score. Logistic Regression з’єднано з файлом (рис. 1.6). У результаті цього отримуємо модель (рис. 1.7).

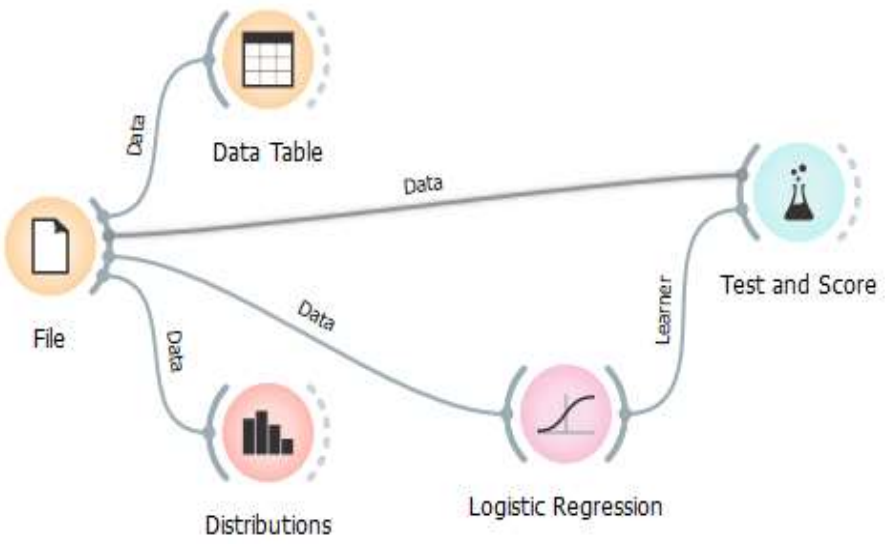


Рис. 1.6. З’єднання модулів

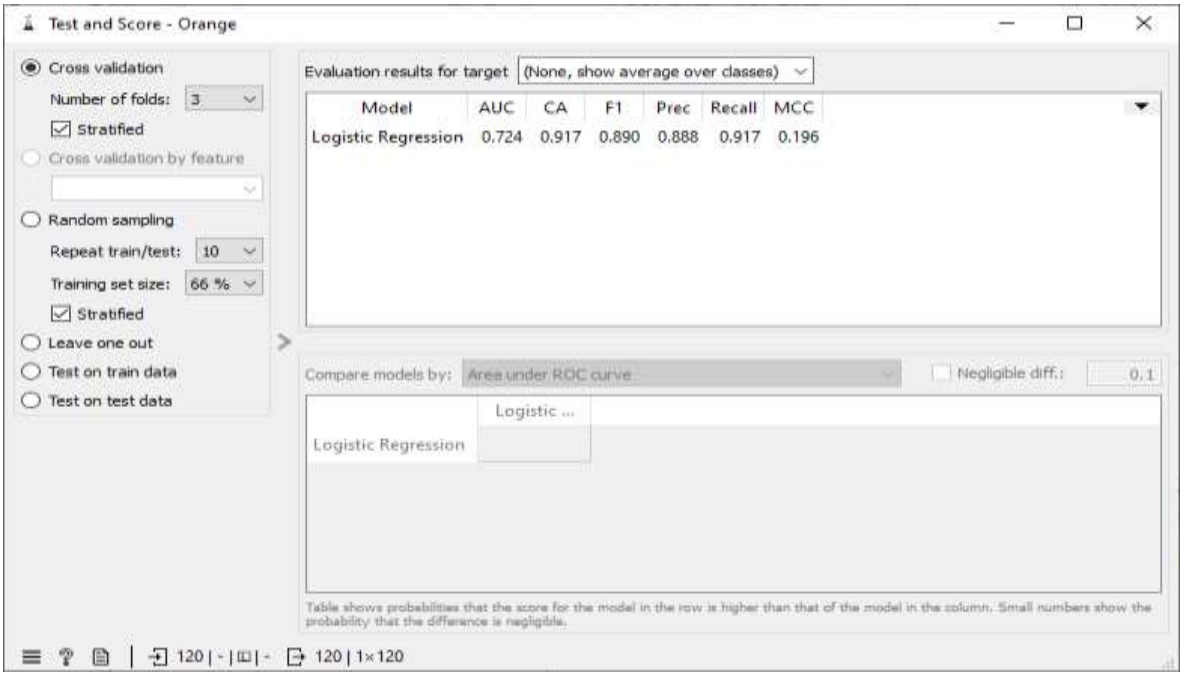


Рис. 1.7. Результат Test and Score

Наступним етапом є включення нового файлу (рис. 1.8) до програми та налаштування необхідних параметрів для того, щоб спрогнозувати ймовірність поломки при температурі 36 ^{0}F.

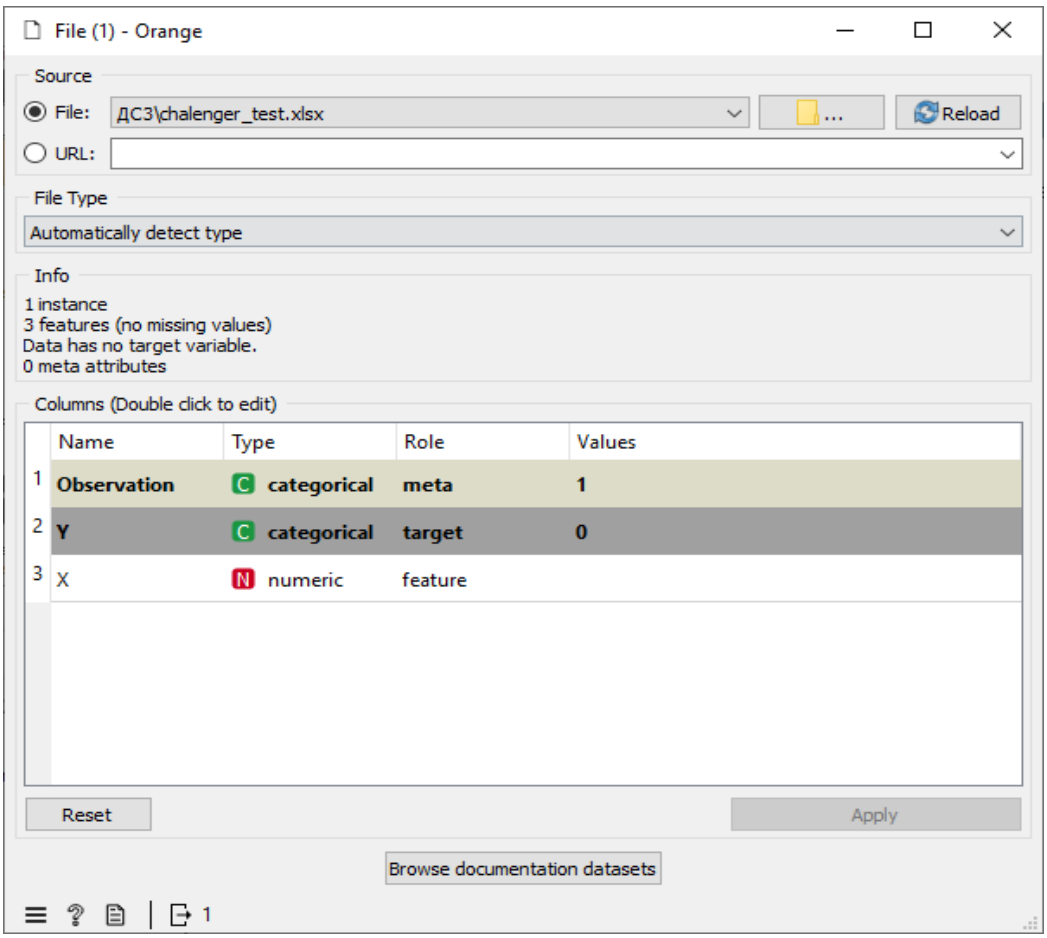


Рис. 1.8. Включення другого файлу

В даному файлі вказано параметри при температурі 36 ^{0}F (рис. 1.9), що використовуються для прогнозу ймовірності поломки кільця.

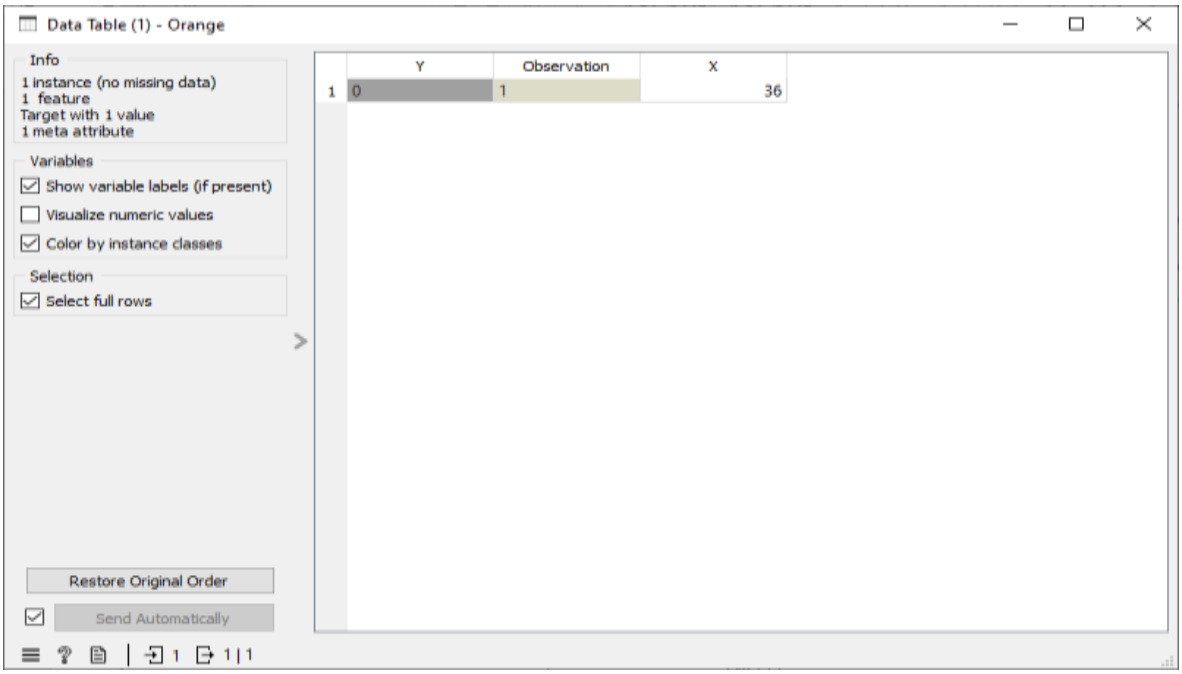


Рис. 1.9. Дані файлу з іншою температурою

Прогноз Predictions – відображення прогнозів моделей для вхідного набору даних – допоможе нам передбачити ймовірність поломки, отримавши дані з другого файлу та використовуючи результати регресійного аналізу (рис. 1.10).

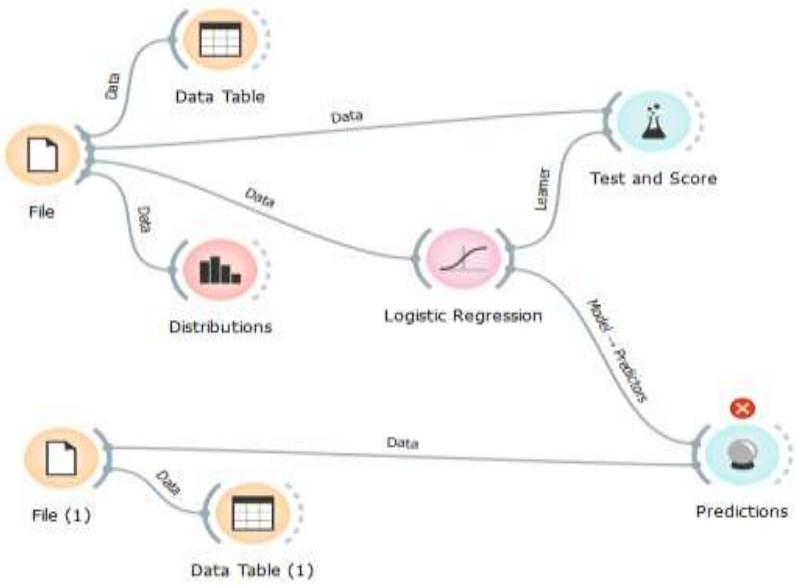


Рис. 1.10. Під'єднання модуля Predictions

У результаті аналізу отримали, що ймовірність поломки кільця за температури 36 ^{0}F складає 89 % (рис. 1.11). Крім того, загальна ймовірність поломки всіх п'яти кілець дорівнює 57 % ($0.893^5 \approx 0.57$).

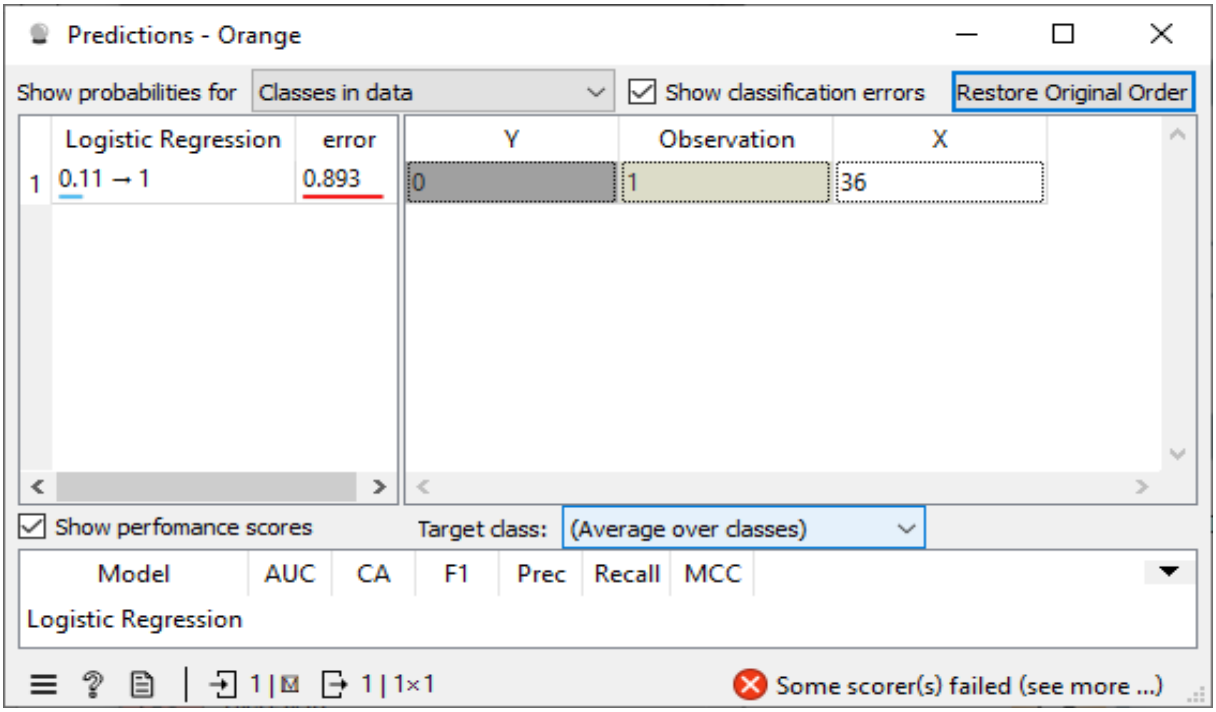


Рис. 1.11. Результат прогнозу

Отримано фінальну схему програми (рис. 1.12).

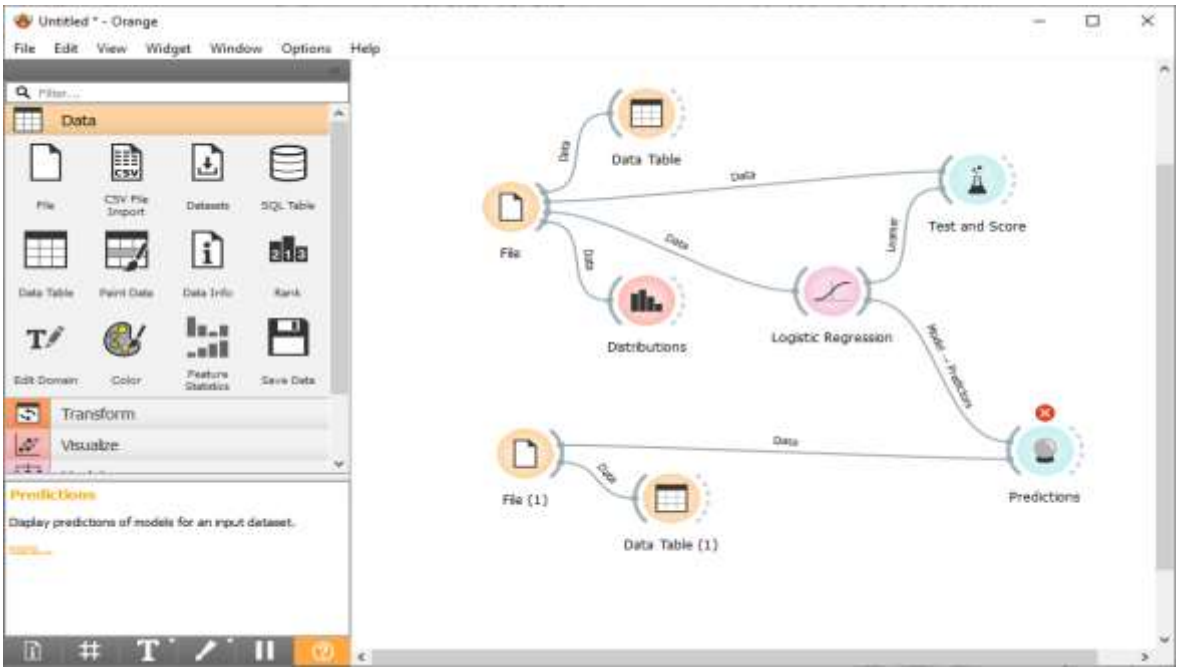


Рис. 1.12. Фінальна схема програми

Висновки щодо роботи (приклад)

Проведений аналіз згідно з класифікацією та застосуванням регресії для прогнозування ймовірності поломок кілець під час польоту шатла Челленджера свідчить про ефективність цього методу в контексті прогнозування надзвичайних ситуацій. В процесі виконання завдання було використано віджети Data Table, Distributions, Logistic Regression та Test and Score для систематичного аналізу даних.

Результати показали, що за температури 36 ^{0}F ймовірність поломки одного кільця становить 89 %, що свідчить про високий ризик виникнення несправностей. Крім того, загальна ймовірність поломки всіх п'яти кілець корабля складає 57 % ($0.893^5 \approx 0.57$), що підтверджує нестабільні умови для цієї конфігурації під час польоту. Впровадження регресійного аналізу в аерокосмічні програми може служити ефективним інструментом для попередження та управління ризиками, а також для покращення безпеки місій національного та міжнародного рівня.

Перелік питань на захист

1. Дайте визначення терміну класифікації даних.
2. Вкажіть основні можливості пакета Orange data mining.
3. Вкажіть об'єкт та ознаки класифікації у проведеній роботі.
4. Дайте визначення логістичної регресії.

ЛАБОРАТОРНА РОБОТА № 2

Тема: Функції подібності в методах класифікації

Постановка задачі: Створити програму, яка обчислюватиме функцію подібності нового об'єкта щодо наданої системи класів з бінарними характеристиками та визначатиме клас, до якого найбільш подібний наданий об'єкт.

Хід роботи:

Функції подібності виявляють суттєві ознаки об'єктів класифікації, які мають бінарне значення (0 або 1). Для виявлення міри близькості таких образів = об'єктів застосовують функції подібності.

Розглянемо два об'єкти :

$$x_i = \{x_{i1}, x_{i2}, x_{i3}, \quad \dots x_{in}\} \text{ та } x_j = \{x_{j1}, x_{j2}, x_{j3}, \quad \dots x_{jn}\},$$
$$\text{де } \{x_{i1}, x_{i2}, x_{i3}, \quad \dots x_{in}\} \text{ та } x_j$$
$$= \{x_{j1}, x_{j2}, x_{j3}, \quad \dots x_{jn}\} - \text{ ознаки цих об}'\text{єктів.}$$

Тоді:

1. Кількість однакових ознак, які є у x_i та x_j визначаються за формулою:

$$a = \sum_{k=1}^{n} x_{jk} \cdot x_{ik}$$

2. Кількість однакових ознак, яких немає у x_i та x_j визначаються за формулою:

$$b = \sum_{k=1}^{n} (1 - x_{ik}) \cdot (1 - x_{jk})$$

3. Кількість однакових ознак, яких немає у x_i, але є у x_j визначаються за формулою:

$$h = \sum_{k=1}^{n} (1 - x_{ik}) \cdot x_{jk}$$

4. Кількість однакових ознак, яких немає у x_j, але є у x_i визначаються за формулою:

$$g = \sum_{k=1}^{n} (1 - x_{jk}) \cdot x_{ik}$$

Із визначення цих параметрів можна зробити висновок: чим більш схожі об'єкти x_i та x_j, тим більше значення параметра a.

Функція подібності – це критерій, який дозволяє привести співвідношення об'єкта до одного чи іншого класу.

Наприклад, використовується функція подібності для класифікації трьох об'єктів (літак, авто, птиця), які характеризуються за ознаками, які мають бінарне значення (див. таблицю 2.1):

Таблиця 2.1

	Криля	Колеса	Двигун	Пір'я
Автомобіль	0	1	1	0
Літак	1	1	1	0
Птиця	1	0	0	1

Нехай автоматичної системі розпізнавання надано невідомий об'єкт $x = \{1, 0, 0, 1\}$. Необхідно визначити функцію подібності для кожного з класів для класифікації нового об'єкта та провести класифікацію.

1. $S_1 = \frac{a}{n}$
2. $S_2 = \frac{a}{n-b}$
3. $S_3 = \frac{a}{2a+h+g}$

Розрахуємо значення а для кожної ознаки:

$a_1 = 0 * 1 + 1 * 0 + 0 * 1 + 1 * 0 = 0$

$a_2 = 1 * 1 + 1 * 0 + 1 * 0 + 0 * 1 = 1$

$a_3 = 1 * 1 + 0 * 1 + 0 * 1 + 1 * 1 = 2$

Використаємо першу функцію належності:

$$S_1 = \frac{a_1}{n} = \frac{0}{4} = 0$$

$$S_2 = \frac{a_2}{n} = \frac{1}{4} = 0{,}25$$

$$S_3 = \frac{a_3}{n} = \frac{2}{4} = 0{,}5$$

Значення функції належності має найбільше значення для об'єкта 3 (птиця), тим самим невідомий об'єкт класифікується як птиця.

Завдання

Використати функції подібності при класифікації трьох об'єктів, у даному випадку види тварин у джунглях: 1) тигр; 2) слон; 3) мавпа. Кожен вид характеризується певними ознаками, які можна визначити за слухом та слідами. Визначення видів тварин у джунглях за допомогою слуху та слідів (див. табл. 2.2).

Таблиця 2.2. Об’єкти і критерії

Вид тварини	Рев	Тріскіт	Гіллясті сліди	Дерева
Тигр	1	0	0	1
Слон	0	1	0	0
Мавпа	0	0	1	1

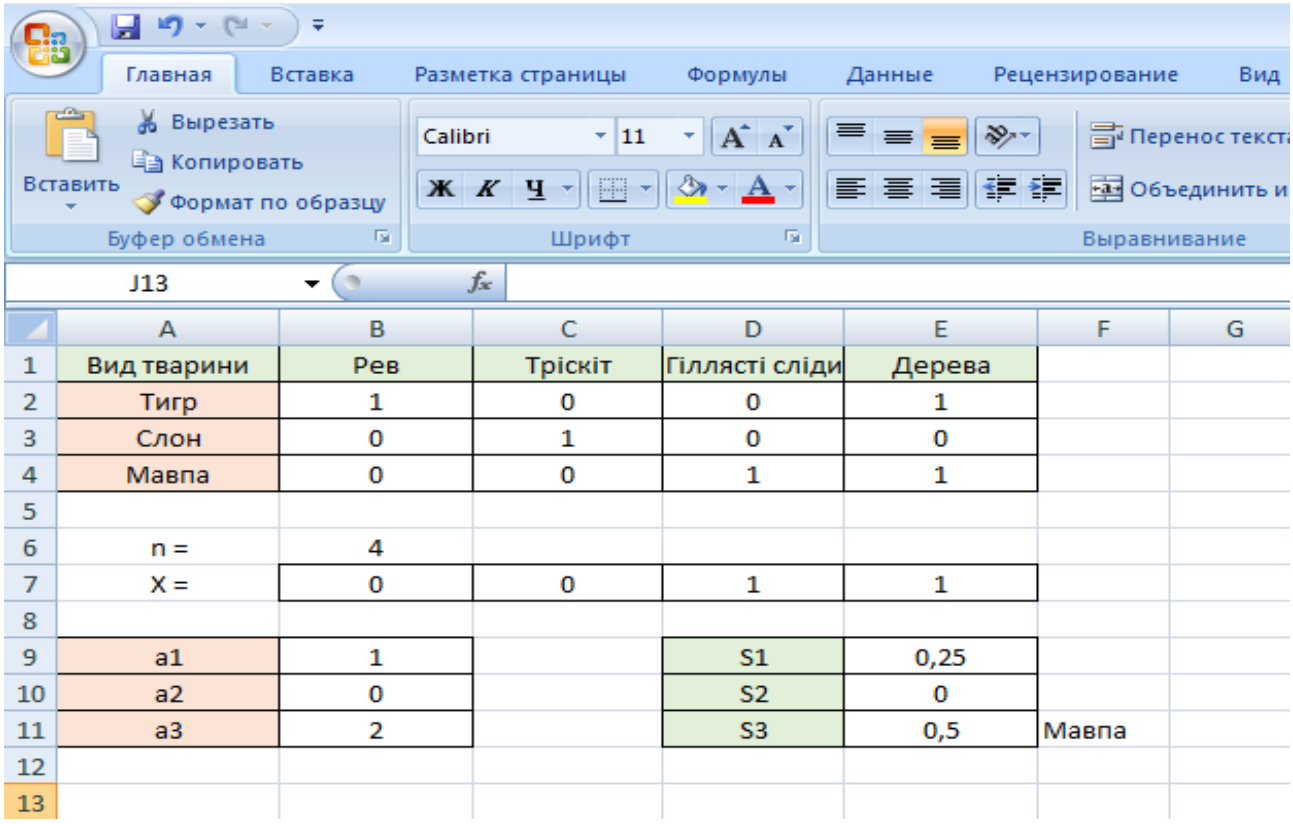
Нехай у системі розпізнавання є об’єкт X. X = {0; 0; 1; 1}. Визначаємо функцію подібності для кожного з класів для класифікації нового об’єкта у програмі (рис. 2.2).

S_1 (i, j) = $\frac{a}{n}$, де кількість ознак (4) = n.

a1	1
a2	0
a3	2

S1	0,25
S2	0
S3	0,5

Висновок: Для пред’явлення невідомого об’єкта функція подібності S_3 (мавпа) отримала максимальне значення, тим самим невідомий об’єкт класифікується як мавпа (рис. 2.1).



	A	B	C	D	E	F	G
1	Вид тварини	Рев	Тріскіт	Гіллясті сліди	Дерева		
2	Тигр	1	0	0	1		
3	Слон	0	1	0	0		
4	Мавпа	0	0	1	1		
5							
6	n =	4					
7	X =	0	0	1	1		
8							
9	a1	1		S1	0,25		
10	a2	0		S2	0		
11	a3	2		S3	0,5	Мавпа	
12							
13							

Рис. 2.1. Розрахунки в Excel

Завдання № 2

Нехай задано 3 об’єкти, серед яких вовк, заєць, курка. Даний набор може мати такі ознаки: хижак, лісна тварина, наявність 4-ох лап, пір’я

Тепер побудуємо таблицю ознак для наших образів:

Ознаки	Хижак	Лісова тварина	Наявність 4-ох лап	Пір’я
Вовк(x_1)	1	1	1	0
Заєць(x_2)	0	1	1	0
Курка(x_3)	0	0	0	1

Маємо:

X_1= {1, 1, 1, 0}

X_2= {0, 1, 1, 0}

X_3 = {0, 0, 0, 1}

Потрібно обчислити S_1, S_2, S_3 за допомогою вищевказаних формул у програмному середовищі

Код програми(Python):

```
x1 = [1,1,1,0]
x2 = [0,1,1,0]
x3 = [0,0,0,1]
val = []
n = len(x1)
a1 = 0
a2 = 0
a3 = 0
s1 = 0
s2 = 0
s3 = 0
el = 0
print("Введіть послідовність з 4 ознак-(0, 1):")
for i in range(4):
   el = int(input())
   val.append(el)
for i in range(4):
   a1 += x1[i]*val[i]
   a2 += x2[i]*val[i]
   a3 += x3[i]*val[i]
s1 = a1/4
s2 = a2/4
```

```
s3 = a3/4
print("s1 = вовк",s1)
print("s2 = заєць",s2)
print("s3 =  курка",s3)
```

Робота програми (рис. 2.2):

Користувач заносить набор ознак через клавіатуру, після чого програма обчисляє значення s у порівнянні з кожним початкове заданим образом:

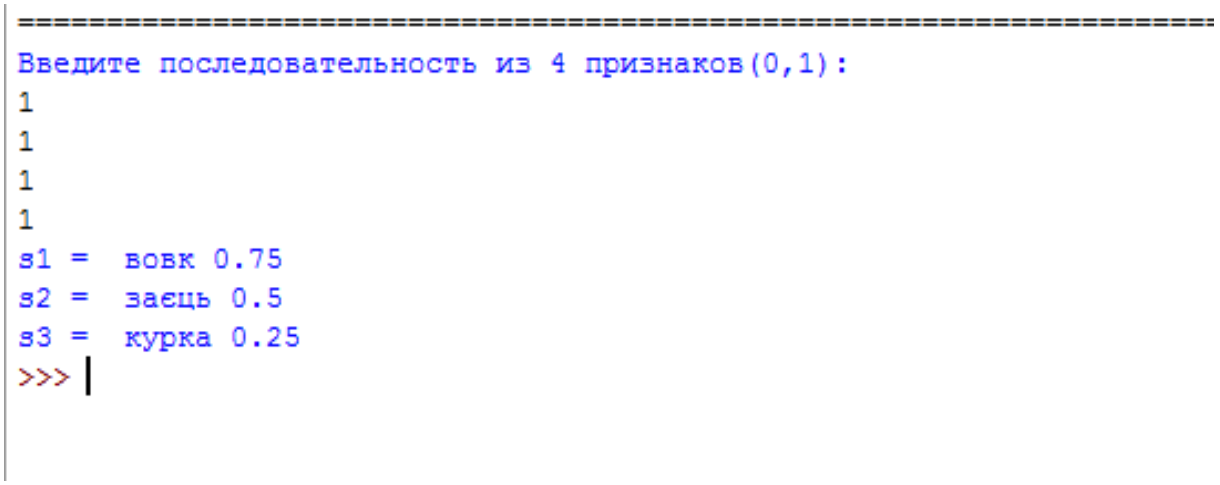


Рис. 2.2. Робота програми класифікації

Завдання № 3

Наприклад, використаємо функцію подібності для класифікації чотирьох об’єктів – музикальних інструментів(гітара, скрипка, барабанна установка, рояль), які характеризуються за ознаками, що мають бінарне значення, а саме – наявність певних елементів в інструментах, що використовуються для гри та відтворення звуків (табл. 2.2):

Таблиця 2.2. Об’єкти і ознаки

Елемент гри Вид інструмента	Струни	Клавіші	Педалі	Смички	Мембрана
Гітара	1	0	0	0	0
Скрипка	1	0	0	1	0
Барабанна установка	0	0	1	0	1
Рояль	1	1	1	0	0

Створимо програму на Python (рис. 2.3), яка зможе визначати функцію подібності $S_i = \frac{a}{n}$ для кожного з класів для класифікації нового об’єкта, який програма буде отримувати у процесі виконання:

```
instruments = {
1: [1, 0, 0, 0, 0],
2: [1, 0, 0, 1, 0],
3: [0, 0, 1, 0, 1],
4: [1, 1, 1, 0, 0],
}

n = len(instruments[1])


def get_instrument_a(instrument_id, values):
    sum = 0
for j in range(5):
        sum += instruments[instrument_id][j] * values[j]
return sum


def get_s(a):
return a / n


# Введенняпослідовністіз 5 ознак(0,1):
val = []
for i in range(5):
    el = int(input())
    val.append(el)

# Обчисленняфункційподібності
s = {}
for i in range(1, 5):
    a = get_instrument_a(i, val)
    s[i] = get_s(a)


max = 0
number = 0
# Виведеннярезультатів
for i in range(1, 5):
print(f"{instruments.get(i)} - s{i} = ", s[i])
if (s[i] > max):
        max = s[i]
        number = i
print(f'Максимальнезначенняфункціїподібностіy s{number}', '=',
max)
```

Рис. 2.3. Код програми

Користувач заносить набір ознак через клавіатуру, після чого програма обчислює значення S у порівнянні з кожним початковим заданим образом та виділяє найбільше значення, тим самим класифікуючи новий об’єкт подібним до певного класу (рис. 2.4):

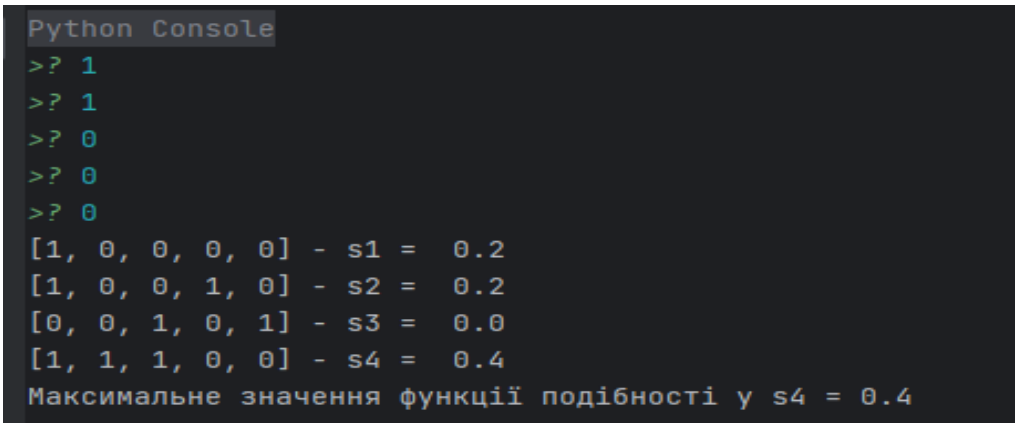


Рисунок 2.4. Приклад виконання програми

Висновок (приклад)

У результаті цієї лабораторної роботи створено програму бінарної класифікації, яка обчислює функцію подібності нового об’єкта щодо наданої системи класів з бінарними характеристиками та визначає клас, до якого найбільш подібний наданий об’єкт. Програма виконує поставлені завдання та працює без помилок у штатному режимі

Завдання № 4

Розглянемо іншу задачу, необхідно класифікувати об’єкт до одного з класів: зірка, газова планета або кам’яна планета.

	Тверда поверхня	Випромінює світло	Тверде ядро	Велика маса
Зірка	0	1	0	1
Газова планета	0	0	1	0
Кам’яна планета	1	0	1	0

Необхідно класифікувати такі об’єкти: $x_1 = \{0, 0, 1, 1\}$ та $x_2 = \{0,1, 0, 0\}$

Перелік питань на захист

1. Функції подібності в задачах класифікації
2. Дайте визначення бінарної класифікації
3. Наведіть структурну схему системи автоматичної класифікації

ЛАБОРАТОРНА РОБОТА № 3

Тема: Дерево рішень у задачах класифікації

Постановка задачі: проаналізувати дані про належність ірисів до певного виду та виконати задачу класифікації даних, побудувавши дерево рішень.

Хід роботи:

Іриси Фішера – набір даних для завдання класифікації, на прикладі якого Рональд Фішер в 1936 продемонстрував роботу розробленого ним методу дискримінантного аналізу. Іноді його називають ірисами Андерсона, оскільки дані були зібрані американським ботаніком Едгаром Андерсоном [2].

Починаємо із включення файлу iris.tab до нашого завдання(рис. 3.1). Файл є вбудованим у програму. Цільова змінна вже обрана – це вид квітки. Також у нас є 4 незалежні параметри-характеристики квітки. Усього набір даних має 150 спостережень.

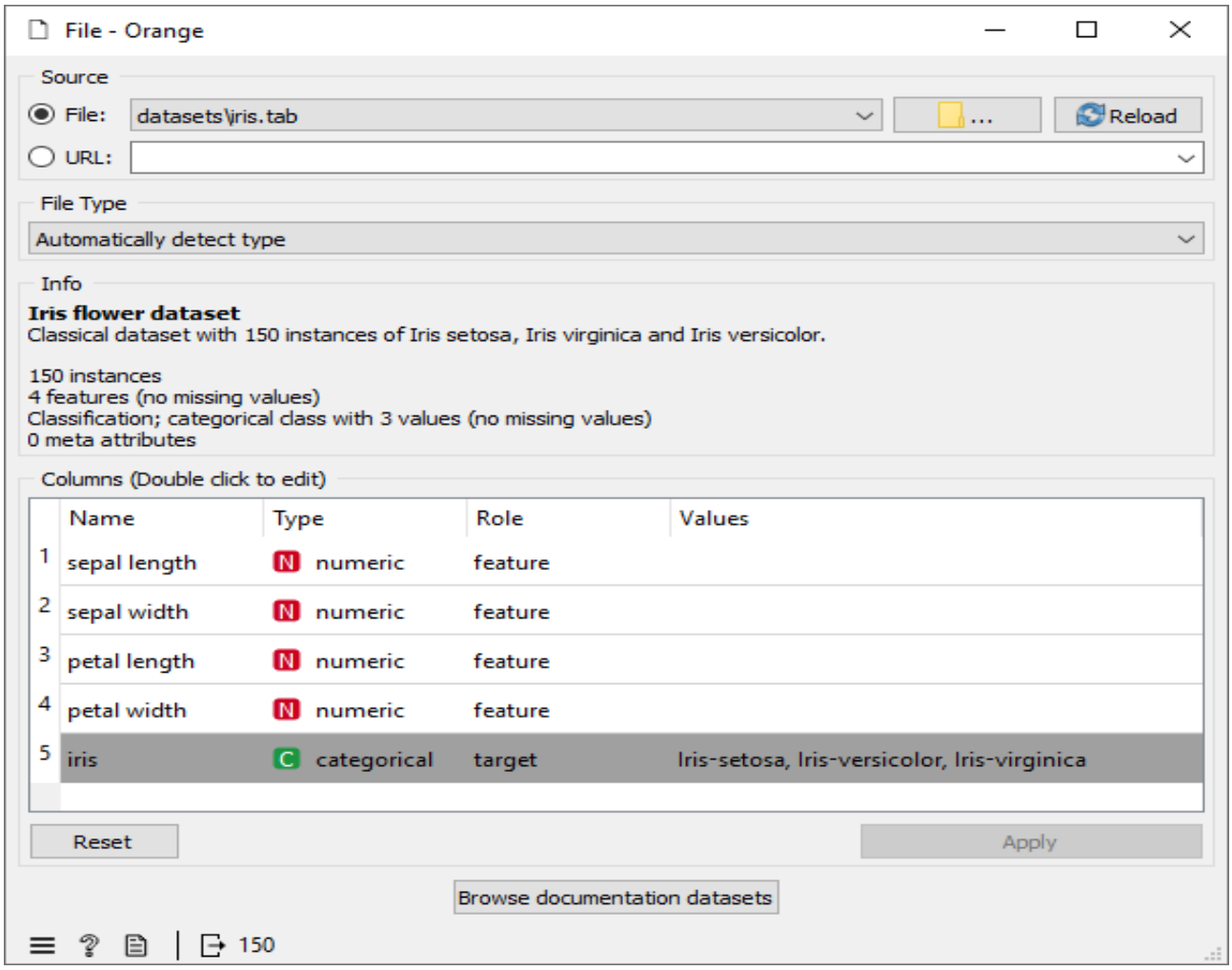
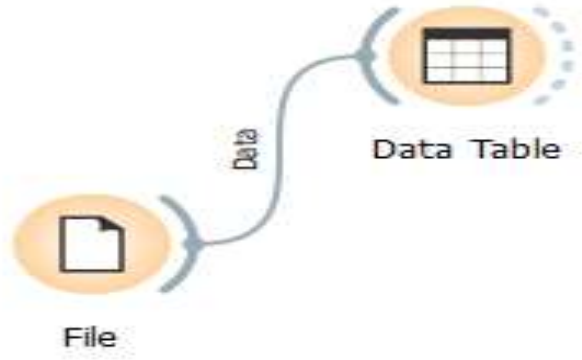


Рис. 3.1. Включення файлу з даними

Підключимо до File в наш проєкт віджет Data Table для зручного відображення та аналізу наявних даних (рис. 3.2).



Data Table - Orange

Info
150 instances (no missing data)
4 features
Target with 3 values
No meta attributes.

Variables
Show variable labels (if present)
Visualize numeric values
Color by instance classes

Selection
Select full rows

Restore Original Order
Send Automatically

150 | 150

	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.1	3.5	1.4	0.2
2	Iris-setosa	4.9	3.0	1.4	0.2
3	Iris-setosa	4.7	3.2	1.3	0.2
4	Iris-setosa	4.6	3.1	1.5	0.2
5	Iris-setosa	5.0	3.6	1.4	0.2
6	Iris-setosa	5.4	3.9	1.7	0.4
7	Iris-setosa	4.6	3.4	1.4	0.3
8	Iris-setosa	5.0	3.4	1.5	0.2
9	Iris-setosa	4.4	2.9	1.4	0.2
10	Iris-setosa	4.9	3.1	1.5	0.1
11	Iris-setosa	5.4	3.7	1.5	0.2
12	Iris-setosa	4.8	3.4	1.6	0.2
13	Iris-setosa	4.8	3.0	1.4	0.1
14	Iris-setosa	4.3	3.0	1.1	0.1
15	Iris-setosa	5.8	4.0	1.2	0.2
16	Iris-setosa	5.7	4.4	1.5	0.4
17	Iris-setosa	5.4	3.9	1.3	0.4
18	Iris-setosa	5.1	3.5	1.4	0.3
19	Iris-setosa	5.7	3.8	1.7	0.3
20	Iris-setosa	5.1	3.8	1.5	0.3
21	Iris-setosa	5.4	3.4	1.7	0.2
22	Iris-setosa	5.1	3.7	1.5	0.4

Рис. 3.2. Використання віджету Data table

Подивимося більш детально. Для цього додаємо новий модуль Distributions для візуалізації, що дозволить нам зрозуміти наскільки збалансовані дані. На рис. 3.3 бачимо, що кожен клас квітів представлений однаковою кількістю квітів – 50 екземплярів.

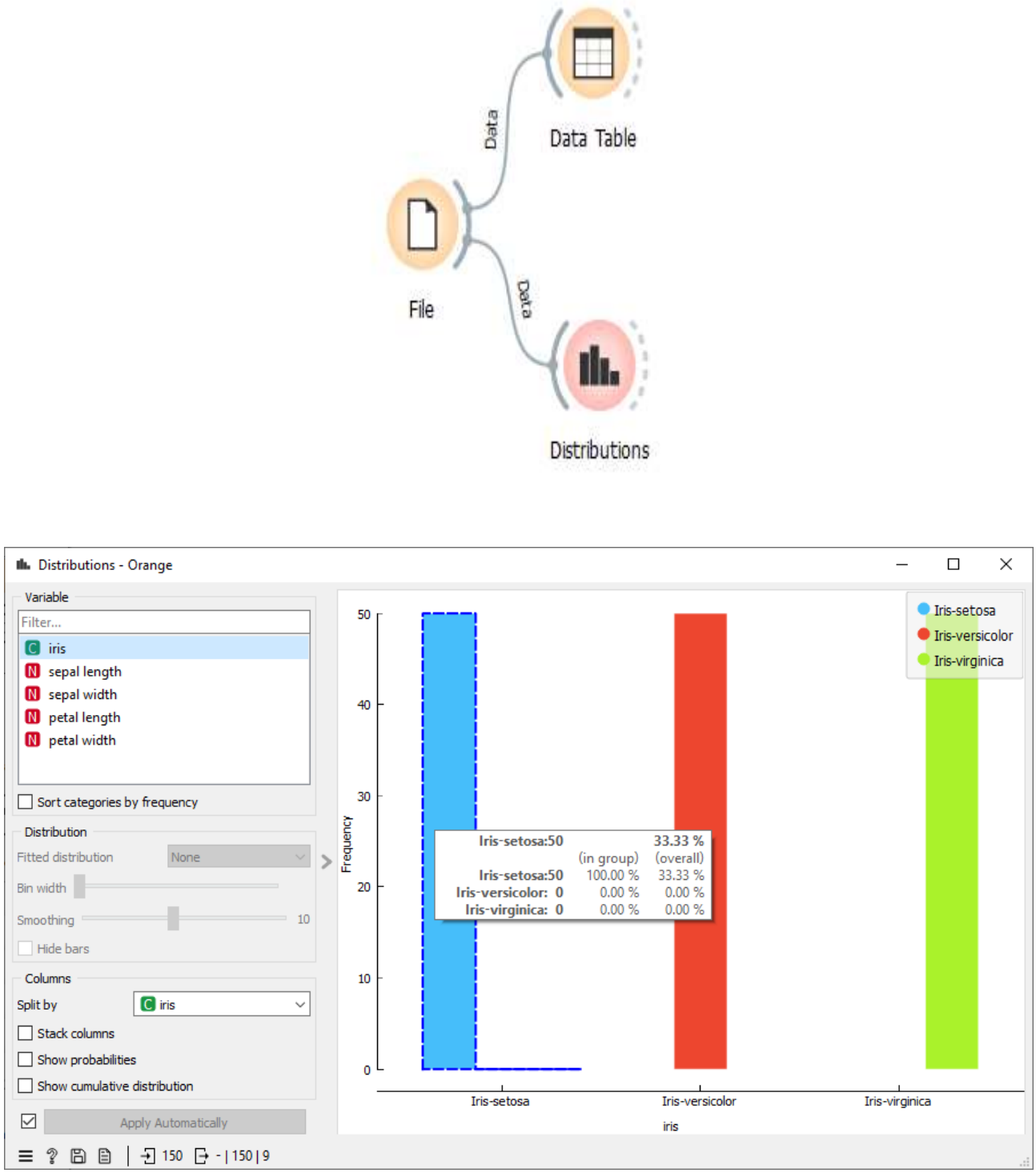


Рис. 3.3. Використання модуля Distributions

Також можна побачити розподіл довжин та ширини чашолистків і пелюсток (рис. 3.4–3.7):

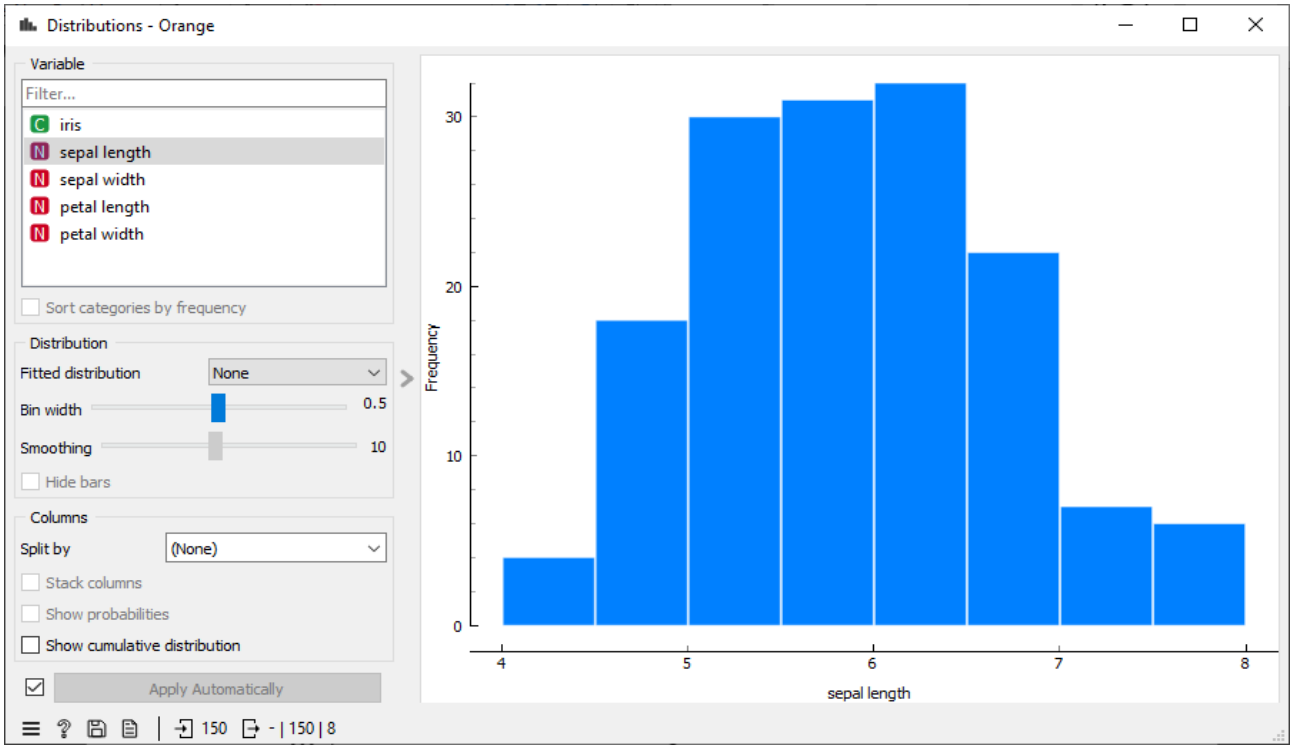


Рис. 3.4. Розподіл за довжиною чашолистків

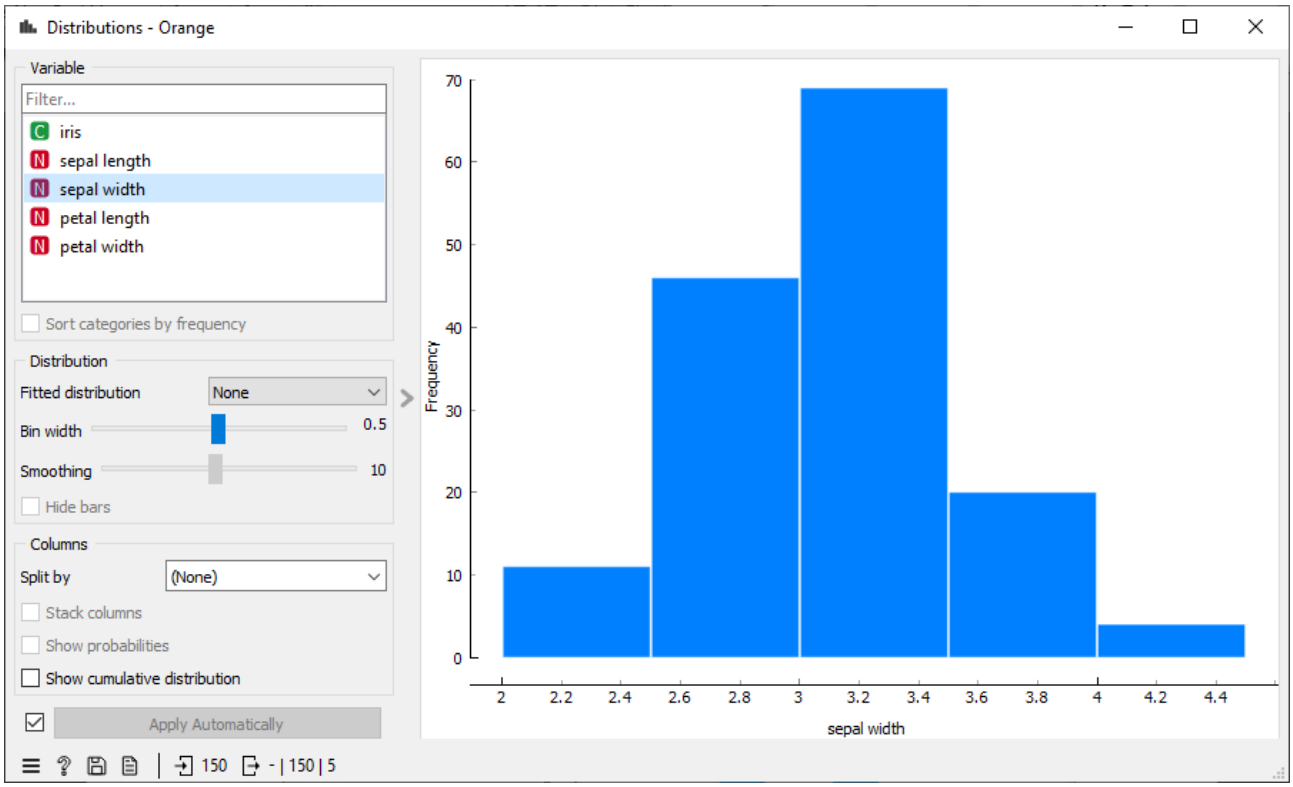


Рис. 3.5. Розподіл за шириною чашолистків

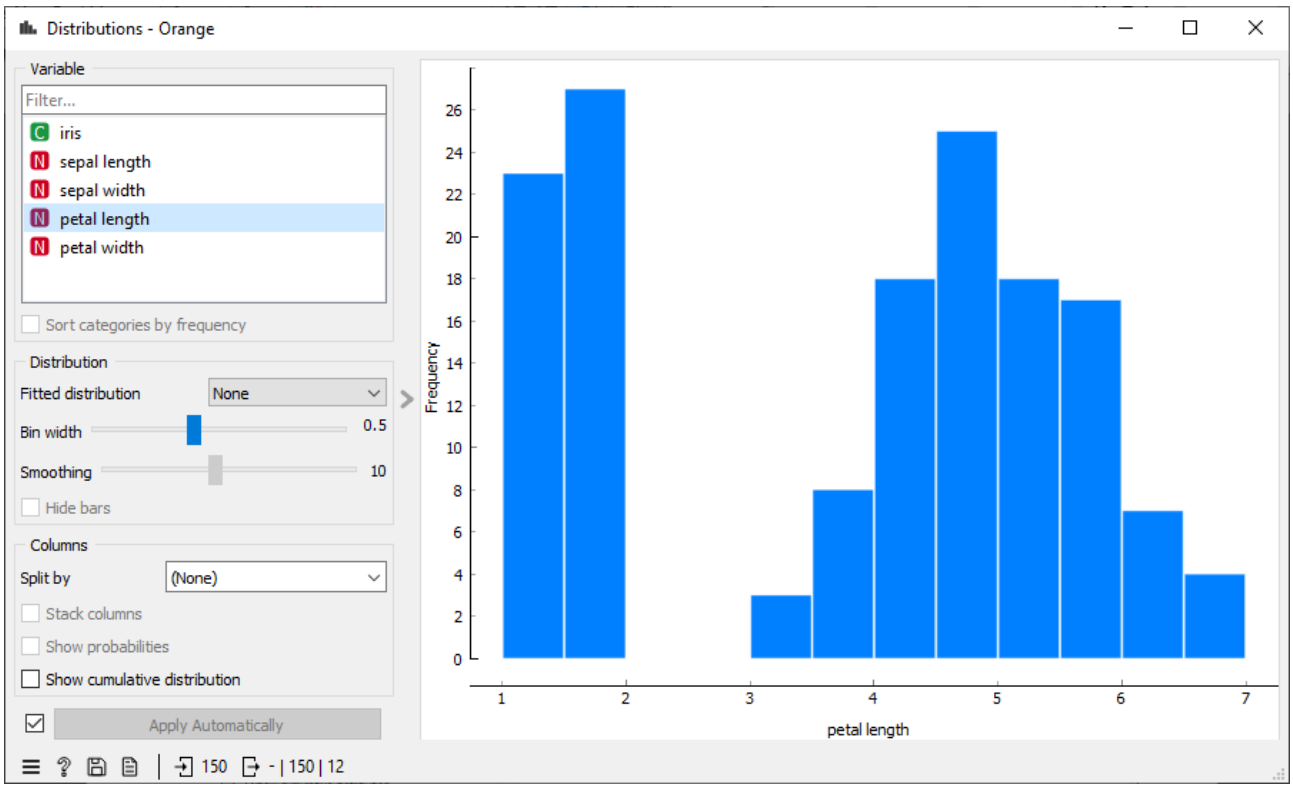


Рис. 3.6. Розподіл за довжиною пелюсток

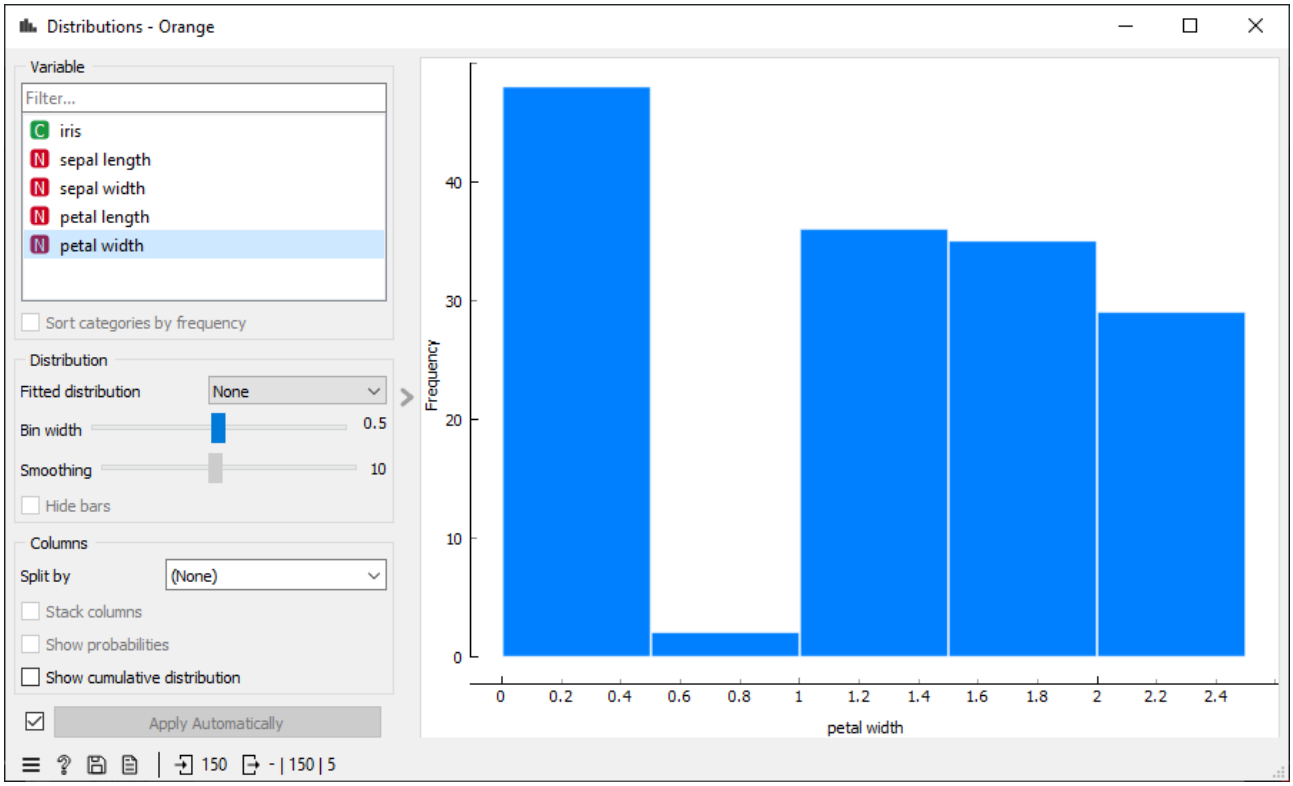


Рис. 3.7. Розподіл за шириною пелюсток

Детальніше подивимося як змінюються параметри в залежності від виду квітки на прикладі довжини чашолистків (рис. 3.8). Як бачимо, навіть на графіку наочним є те, що дані можна легко класифікувати на три класи – три види. Це можна побачити ще, наприклад, за допомогою модуля Scatter Plot (рис. 3.9).

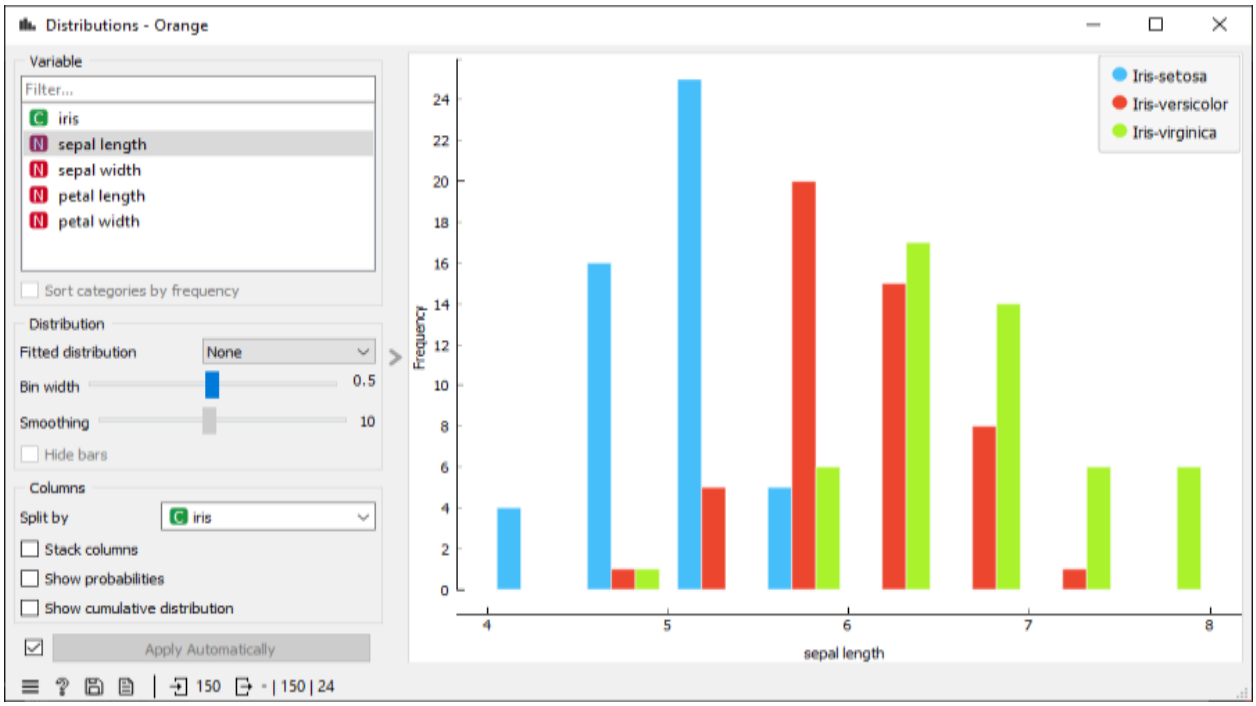


Рис.3.8. Залежність довжини чашолистків від виду квітки

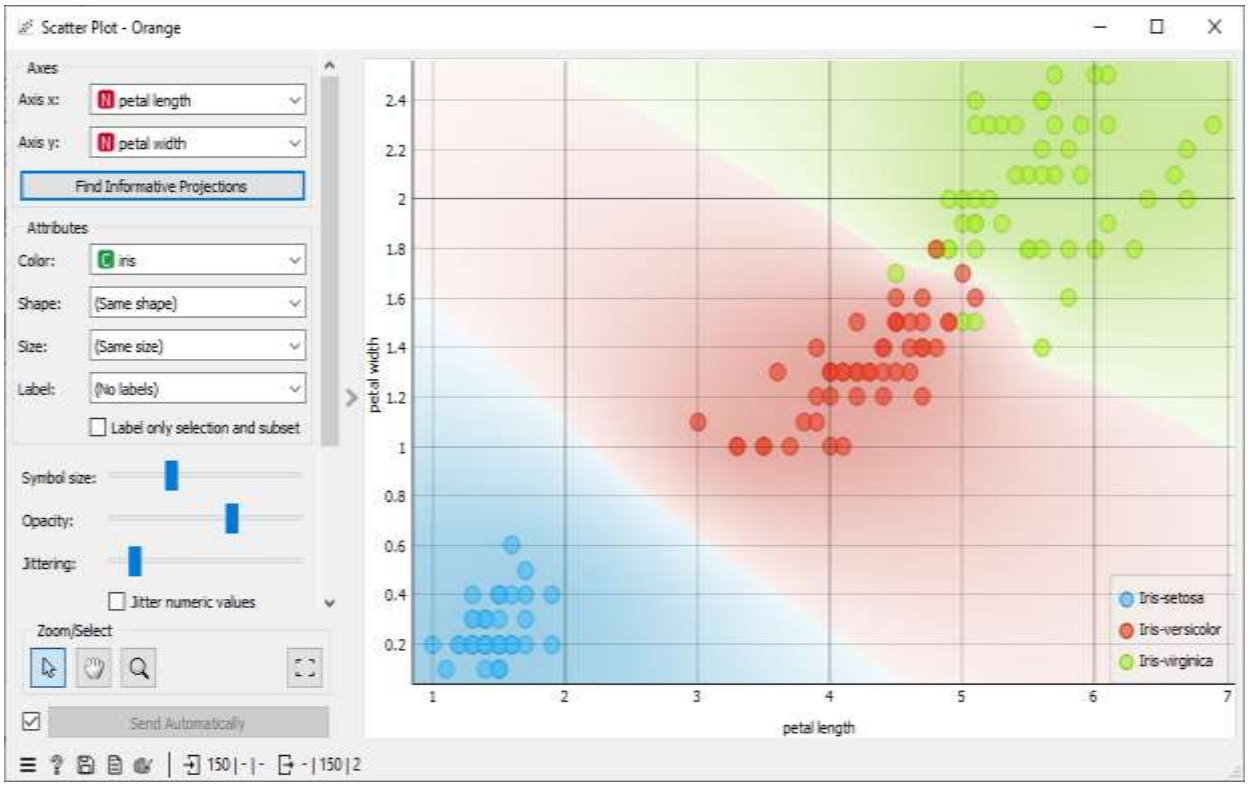


Рис. 3.9. Залежність довжини та ширини пелюсток від виду квітки

Побудуємо дерево рішень. Під’єднаємо один до одного модулі Tree і Test and Score. Tree з’єднано з файлом (рис. 3.10). У результаті цього отримуємо модель (рис. 3.11).

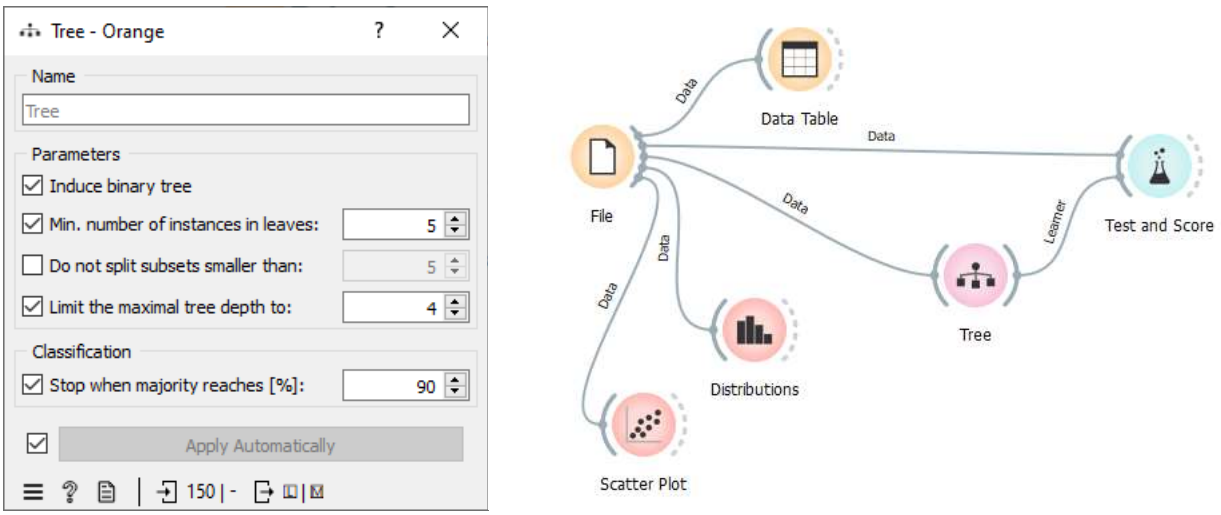


Рис.3.10. З’єднання модулів

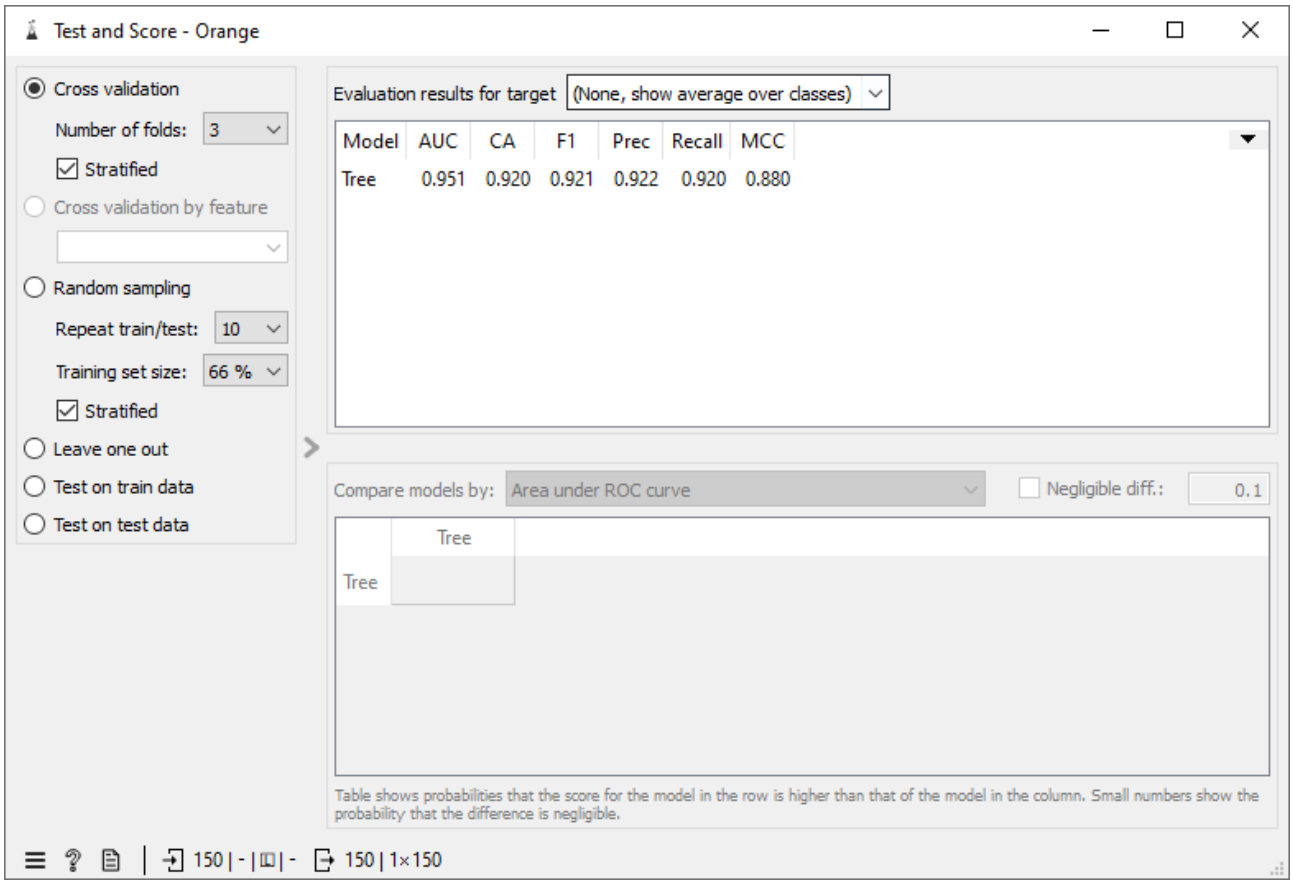


Рис. 3.11. Результат Test and Score

Наступним етапом є додавання модуля Tree Viewer (рис. 3.12) до програми. Переглянемо отримане дерево рішень (рис. 3.13).

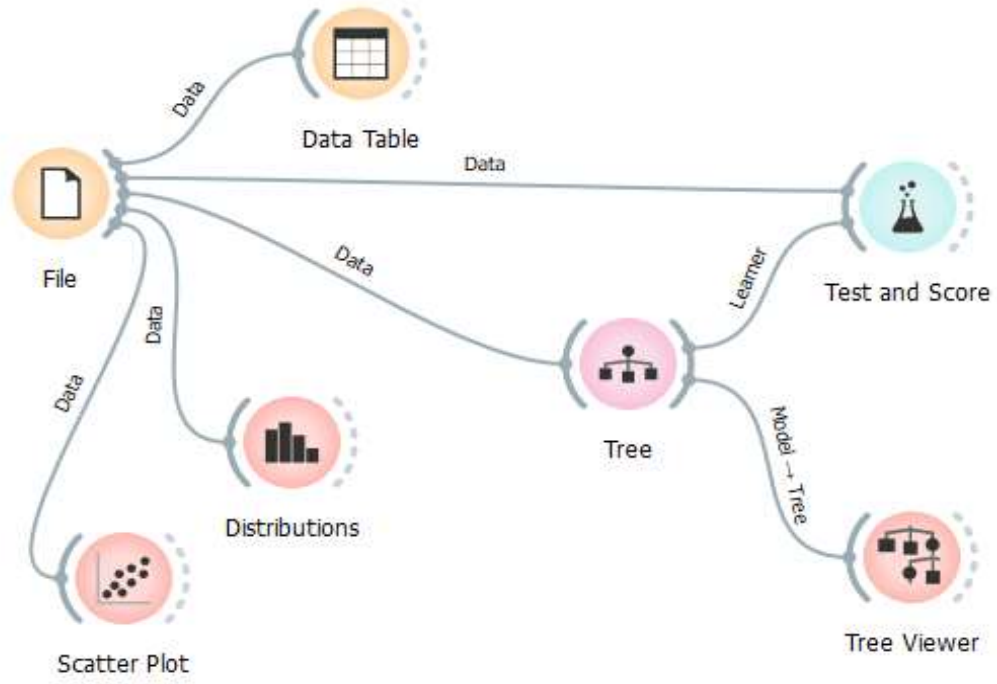


Рис. 3.12. Додавання модуля Tree Viewer

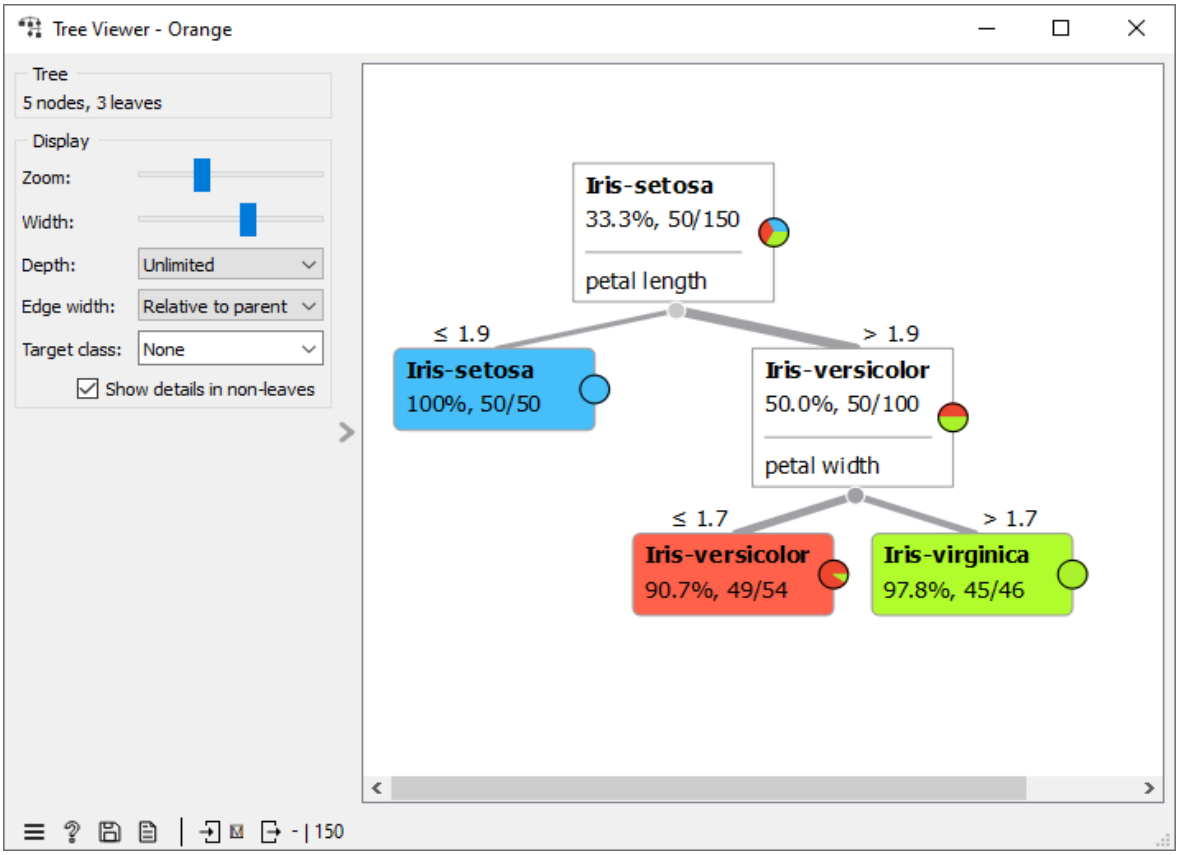


Рис. 3.13. Дерево рішень, отримане за допомогою модуля Tree

Отримане дерево рішень наочно демонструє, що якщо, наприклад, ми зупинимося у корні, то рішення «Iris-setosa» буде вірним у 33,3 % випадків, як і будь-яке інше рішення щодо визначення виду квітки (нагадаю, що ми маємо рівну кількість квіток кожного виду – по 50).

Але якщо проаналізуємо визначення рішення глибше і визначимо довжину пелюстки, то отримаємо вже можливі рішення з більшим відсотком вірогідності. Таким чином, якщо довжина пелюстки квітки менше або дорівнює 1.9, то перед нами стовідсотково саме «Iris-setosa», в іншому випадку – 50 на 50, що це «Iris-versicolor» або «Iris-virginica» (див. рис. 3.13).

Але можемо визначити далі ще один параметр і визначити вид квітки у правій вітці дерева ще більш точно. Таким чином, якщо довжина чашолистку квітки менше або дорівнює 1.7, то маємо «Iris-versicolor» з вірогідністю 90.7 %, у іншому випадку – «Iris-virginica» з вірогідністю 97.8 %.

Важливим є те, що при додаванні модуля Tree було обрано параметри для обмежень глибини дерева та мінімальну кількість елементів у листку. Якщо цього не робити або збільшити ці параметри, то дерево може «вирости» занадто великим. Внаслідок цього можливість передбачити рішення буде менш оптимальною. Важливо знайти баланс цих параметрів для правильного аналізу даних.

Висновок

Проведено аналіз даних про належність ірисів до певного виду та виконано задачу класифікації даних, а саме – побудоване дерево рішень. Дерево рішень є зручним для логічної класифікації даних з-за кількох причин:

- Простота інтерпретації: Рішення, виконані у вигляді дерева, легко читати і розуміти. Кожен вузол у дереві представляє логічне правило, що робить його зрозумілим для аналітиків і дослідників.
- Гнучкість: Дерева рішень можна легко модифікувати або розширювати шляхом додавання нових вузлів або зміни правил у вже існуючих вузлах.
- Відсутність умовності: Дерева рішень не вимагають умовних припущень про розподіл даних, оскільки вони можуть працювати як з категоріальними, так і з числовими даними без особливих перетворень.
- Відображення важливості змінних: Дерева рішень дозволяють оцінювати важливість різних змінних у процесі прийняття рішень, що може бути корисним для подальшого аналізу та оптимізації моделі.
- Ефективність: Дерева рішень можуть працювати досить швидко, особливо на великих наборах даних, тому їх використання є ефективним в різних застосунках.

Узагальнюючи, дерева рішень є потужним інструментом для логічної класифікації даних, оскільки вони поєднують в собі простоту, гнучкість і ефективність, дозволяючи аналітикам та дослідникам ефективно працювати з різноманітними наборами даних.

Перелік питань на захист

1. Дерево рішень у задачах класифікації
2. Навести етапи процесу класифікації
3. Параметри точності класифікації
4. Які проблеми виникають при класифікації?

ЛАБОРАТОРНА РОБОТА № 4

Тема: Метрика Махаланобіса у задачах класифікації

Постановка задачі: Вивчення метрики Махаллонобіса як способу вимірювання відстані між точками в багатовимірному просторі, що враховує кореляцію та варіацію даних. Написання програми, яка буде відтворювати алгоритм підрахунку відстані від нової вхідної точки до наявних двох класів згідно з метрикою Махаланобіса.

Теоретичні відомості

Метрика Махаланобіса дозволяє значно покращити процедуру навчання автоматичної системи класифікації у таких випадках:

1. Істотні ознаки об'єктів обрані неадекватно, класи погано поділяються;
2. Істотні ознаки об'єктів сильно корелюють між собою;
3. Розділяюча поверхня між класами сильно вигнута:
4. Класи можуть складатися з підкласів, що не стикаються між собою у просторі суттєвих ознак;
5. Розділяючі поверхні між класами мають дуже складну форму.

Відстань по Махалонобісу між об'єктом X $^{\rightarrow}$ та центром m $^{\rightarrow}$ деякого класу обчислюється за формулою:

$$d(\vec{X},\vec{m}) = \left\{\left(\vec{X}-\vec{m}\right)^T C^{-1}\left(\vec{X}-\vec{m}\right)\right\}^{12} \qquad (4.1)$$

де C^{-1} – матриця, зворотна до коваріаційної матриці для даного класу. $\left(\vec{X}-\vec{m}\right)$ матриця – стовпець, елементи якої – різниці однойменних координат об'єкта $\vec{X}$ і центру класу $\vec{m}$.

Для того щоб знайти відстань між об'єктом та класом у метриці Махалонобіса, використовуються такі величини як центр класу, дисперсія, середньоквадратичне відхилення та матриця підступів. Розглянемо їх визначення та методи обчислення.

Нехай є певний клас C, який містить n об'єктів, у кожного з яких є суттєві ознаки, тобто

$$\overrightarrow{X_i} = (x_{i1}, x_{i2} \ldots\ldots X_{ik}), \vec{X}_1 \in C_1, \forall i \epsilon [1,n]$$

Центром класу C_j є вектор $\vec{m}_j = (m_{j1}, m_{j2} \ldots m_{jk})$, компоненти якого – середні значення однойменних істотних ознак всіх об'єктів даного класу. Іншими словами, середнім значенням і-тої істотної ознаки об'єктів у того класу є арифметичне середнє:

$$m_{ji} = \frac{[x_{j1} + x_{j2} + \cdots x_{jk}]}{n} \quad (4.2)$$

Дисперсія – це міра «розмитості» класу або міра відхилення значень суттєвих ознак об'єктів від цього класу. У k-мірному просторі суттєвих ознак дисперсія для класу С визначається у вигляді вектора-рядка або матриці-рядка виду $S_j^2 = (S_{j1}^2, S_{j2}^2, \ldots . S_{jk}^2)$, причому елементи цієї матриці – дисперсія і-тої ознаки визначаються із співвідношення:

$$s_{ji}^2 = \left((x_{i1} - m_{ji})^2 + (x_{i2} - m_{ji})^2 + \cdots + (x_{in} - m_{ji})^2\right)/(n-1) \quad (4.3)$$

Стандартним або середньоквадратичним відхиленням називається квадратний корінь з дисперсії $\overrightarrow{S_j} = \sqrt{S_j^2} = (S_{j1}, S_{j2}, \ldots . S_{jk})$, причому середньоквадратичне відхилення і-тої ознаки об'єктів j-того класу визначається за формулою

$$S_{ji} = \sqrt{S_{ji}^2} = \sqrt{\left((x_{i1} - m_{ji})^2 + (x_{i2} - m_{ji})^2 + \cdots + (x_{in} - m_{ji})^2\right)/(n-1)} \quad (4.4)$$

Ця величина має таку ж розмірність, як і середнє значення суттєвих ознак даного класу.

Хід роботи:

Метрика Махалонобіса – це спосіб вимірювання відстані між двома точками в багатовимірному просторі, що враховує кореляцію та варіацію даних. Вона є узагальненням евклідової відстані на випадок, коли дані несферичні чи корельовані.

Суть метрики Махалонобіса полягає в обчисленні відстані між двома точками з урахуванням підступності даних. Вона використовується, наприклад, у багатовимірному статистичному аналізі та машинному навчанні для оцінки подібності чи відмінності між об'єктами при врахуванні їхньої варіабельності та взаємозв'язків. Ця метрика враховує не лише відстань між точками, але й структуру та залежності в даних, що робить її корисною у випадках, коли звичайна евклідова відстань не враховує цю інформацію.

Для знаходження відстані від точки до класу необхідно зробити такі кроки метрики:

Крок 1. Обчислити математичні очікування значень ознак класових точок.

$$m_{ij} = \frac{\left[x_{i1} + x_{i2} + \cdots + x_{in_j}\right]}{n_j}$$

Крок 2. Обчислити середньоквадратичні відхилення значень ознак точок класу.

$$S_{ij} = \sqrt{\frac{[(x_{i1} - m_i)^2 + (x_{i2} - m_i)^2 + \cdots + (x_m - m_i)^2]}{n - 1}}$$

Крок 3. Обчислити відстані.

$$d_i(\bar{x}, \bar{m}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - m_i)^2}{s_i^2}}.$$

Напишемо програму на Python, виконуючи кожен крок алгоритму (рис. 4.1):

```
importnumpyasnp

# Визначення класів та їх характеристики
class1 = np.array([[1, 2], [3, 4], [5, 6], [6, 8], [9, 10]])
class2 = np.array([[10, 12], [13, 14], [15, 16], [17, 18], [19,
20]])
print("Клас 1:", class1)
print("Клас 2:", class2)

# Обчислення математичного очікування та середньоквадратичного
відхилення для кожного класу
mean_class1, mean_class2 = np.mean(class1, axis=0),
np.mean(class2, axis=0)
std_deviation1, std_deviation2 = np.std(class1, axis=0, ddof=1),
np.std(class2, axis=0, ddof=1)

# Введення нової точки
new_point = np.array([3, 5])
print("Нова точка:", new_point)

# Розрахунок відстаней d для класу 1
d_class1 = np.sqrt(np.sum(((new_point - mean_class1) /
std_deviation1)**2))

# Розрахунок відстаней d для класу 2
d_class2 = np.sqrt(np.sum(((new_point - mean_class2) /
std_deviation2)**2))

print("Відстань від нової точки до класу 1:", d_class1)
print("Відстань від нової точки до класу 2:", d_class2)
```

Рис. 4.1. Скриншот програми

Вихідні дані об’єктів двох класів:

Клас 1: [[1, 2], [3, 4], [5, 6], [7, 8], [9, 10]]

Клас 2: [[11, 12], [13, 14], [15, 16], [17, 18], [19, 20]]

Для прикладу взято нову точку (об’єкт) з ознаками (3; 5).

Продемонструємо роботу програми. Далі наведено результат – скріншот консолі (рис. 4.2):

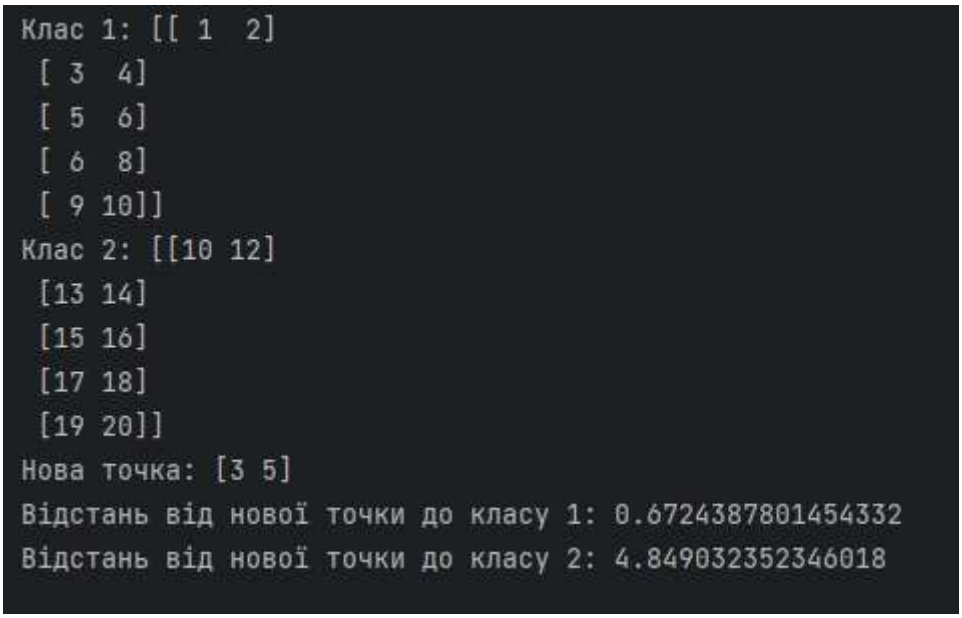


Рис. 4.2. Результат виконання програми метрики

Завдання

Нехай маємо 10 об’єктів, які мають дві значні ознаки. Вони були поділені на два класи. Також відомо новий $\bar{x} = (8, 8)$, який необхідно віднести до двох класів.

Clas №1				Clas №2		
i/j	1 - озн	2 - озн		i/j	1 - озн	2 - озн
1	1	1		1	2	1
2	3	3		2	4	2
3	5	5		3	5	3
4	8	8		4	6	5
5	9	9		5	9	8

Крок ПЕРШИЙ: Знаходимо центр кожного з класів.

$$m_1 = \frac{[x_{i1} + x_{i2} + \cdots + x_{in}]}{n}, i = 1 \ldots k$$

$$m_2 = \frac{[x_{j1} + x_{j2} + \cdots + x_{jn}]}{n}, j = 1 \ldots k$$

clas#1	
m1	m2
5,2	5,2

clas#2	
m1	m2
5,2	3,8

Крок ДРУГИЙ: Визначимо середнє квадратичне відхилення ознак класів

$$S_{11} = \sqrt{\frac{(x_{i1} - m_1)^2 + (x_{i2} - m_1)^2 + \cdots + (x_{in} - m_1)^2}{n-1}}$$

$$S_{12} = \sqrt{\frac{(x_{j1} - m_2)^2 + (x_{j2} - m_2)^2 + \cdots + (x_{jn} - m_2)^2}{n-1}}$$

Цей етап було поділено на два. Спочатку розраховано все, що під коренем. Потім розраховано з коренем.

clas#1	
S11	S12
11,2	11,2

clas#2	
S21	S22
6,7	7,7

S(new)1	
S11	S12
3,34664	3,34664

S(new)2	
S21	S22
2,588436	2,774887

Крок ТРЕТІЙ: Відстань між об’єктом і класом. $\bar{x} = (8, 8)$, для того щоб віднести об’єкт до двох класів, визначимо відстань.

$$dis = \left|\frac{\bar{x} - \bar{m}}{S}\right|$$

d1S	1,183216
d2S	1,860392

Мінімальне значення дорівнює d1S. Тому мінімальну відстань має Перший клас.

Висновок

Результати дослідження метрики Махаллонобіса показують, що ця метрика враховує структуру та залежності в даних, що робить її корисною для оцінки подібності або різниці між об'єктами в багатовимірному просторі. Відстань Махалонобіса від нової точки до класу 1 виявилася значно меншою, ніж до класу 2, що говорить про близькість нової точки до класу 1 за структурою даних та кореляцій характеристик.

Перелік питань на захист

1. Етапи метрики Махалонобіса
2. Відстань між двома класами. Наведіть відстань Евкліда
3. Наведіть відстань Чебишева
4. Наведіть відстань по Манхетону та Хемингу

ЛАБОРАТОРНА РОБОТА № 5

Тема: Методи ієрархічного кластерного аналізу. Метод ближчого сусіда

Постановка задачі: Ознайомитися з основними поняттями і методами ієрархічного кластерного аналізу. Дослідити та проаналізувати принцип роботи методу ближчого сусіда (МБС) або одиночного зв'язку в ієрархічному кластерному аналізі. Реалізувати алгоритм одиночного зв'язку для кластеризації даних. Провести тестування реалізованого алгоритму на штучних даних та проаналізувати результати.

Хід роботи:

Дана матриця 4х4, за допомогою методу найближчого сусіда потрібно обчислити кластери з найменшою відстанню та побудувати дендрограму.

Нехай задано матрицю розміром 4х4, в якій знаходяться наші дані щодо відстані між кожним кластером:

Елемент	1	2	3	4
1	0	3.06	4.13	6.42
2	3.06	0	3.50	4.22
3	4.13	3.50	0	2.32
4	6.42	4.22	2.32	0

За допомогою методу найближчого сусіда спочатку зменшимо матрицю до розмірів 3х3 шляхом об’єднання об’єктів 1 та 2 (їх рядків відповідно) та вибравши один з мінімальною відстанню до інших об’єктів (3 та 4) на місце нового кластеру 1,2:

Елемент	1,2	3	4
1,2	0	3.50	4.22
3	3.50	0	2.32
4	4.22	2.32	0

Тепер таким же чином об’єднаємо 3 та 4 і зафіксуємо результат в матриці розміром вже 2х2:

Елемент	1,2	3,4
1,2	0	3.50
3,4	3.50	0

Сходячи з наших дій, отримуємо наступне:

Перший кластер(1,2) має відстань 3.06.

Другий кластер(3,4) має відстань 2.32.

Фінальний кластер, який ми отримали після обробки матриці методом найближчого сусіда має відстань 3.50.

Реалізуємо даний алгоритм в програмному середовищі за допомогою Python :

Код програми(Python):

```
matrix = [[0, 3.06, 4.12, 6.42],
[3.06, 0, 3.50, 4.22],
[4.12, 3.50, 0, 2.32],
[6.42, 4.22, 2.32, 0]]
defclustsum(matrix,min_limit,max_limit):
min_el = 10
for i inrange(min_limit,max_limit):
for j inrange(min_limit,max_limit):
if(min_el>matrix[i][j] andmatrix[i][j] > 0):
min_el = matrix[i][j]
returnmin_el
clst1 = clustsum(matrix,0,2)
clst2 = clustsum(matrix,2,4)
clst3 = clustsum(matrix,1,3)
print(" Кластери 1 2: ", clst1,'\n',"Кластери 3 4: ", clst2,'\n',"Підсумковий кластер: ", clst3)
```

Робота програми:

```
Кластеры 1 2:  3.06
Кластеры 3 4:  2.32
Итоговый кластер:  3.5
```

Дендрограма (рис. 5.1):

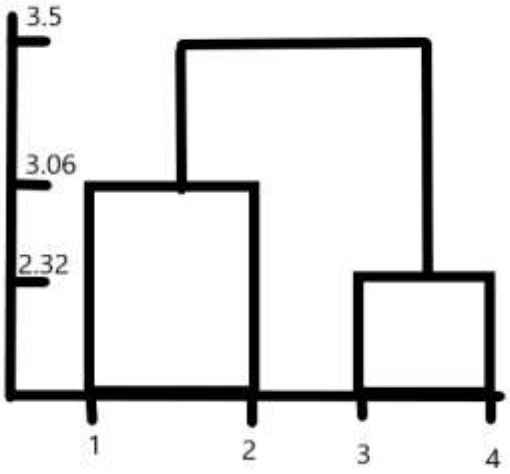


Рис. 5.1. Дендрограма МБС

Завдання

Розглянемо один із способів розподілу об'єктів за групами – ***агломеративний метод ієрархічної кластеризації.*** Він полягає у послідовному поєднанні точок у кластери. При цьому спочатку кожен об'єкт лежить в окремій групі, потім на кожному кроці найближчі кластери об'єднуються на підставі вибраних метрик відстані. Тобто дерево створюється від листків до стовбуру.

Для побудови матриці подібності (відмінності) необхідно поставити міру відстані між двома кластерами. Найчастіше використовуються такі методи визначення відстані: метод одиночного зв'язку, повного зв'язку, середнього зв'язку, центрохідний метод та метод Уорда.

У даній роботі як дистанції між кластерами будемо приймати *мінімальну* відстань між двома точками одного та іншого кластера, тобто ***метод одиночного зв'язку*** або його інша назва – ***«метод найближчого сусіда»***: $\min\{d(a,b) : a \in A, b \in B\}$, де $d(a,b)$— відстань між елементами a та b, що належать кластерам Aта B .

Як метрика відстані між точками зазвичай використовується ***евклідова міра*** (також підтримується багато інших, наприклад, кореляція, косинуса відмінність).

Виконаємо кластеризацію методом найближчого сусіда та візуалізуємо набір даних. Дана матриця відстаней:

$$A = \begin{matrix} 0 & 2.06 & 4.03 & 6.32 & 2.08 \\ 2.06 & 0 & 3.5 & 4.12 & 5.43 \\ 4.03 & 3.5 & 0 & 2.25 & 3.65 \\ 6.32 & 4.12 & 2.25 & 0 & 4.81 \\ 2.08 & 5.43 & 3.65 & 4.81 & 0 \end{matrix}$$

Для візуалізації використовується ***дендрограма***. Під дендрограмою зазвичай розуміється дерево, побудоване за матрицею мір близькості. Дендрограма дозволяє зобразити взаємні зв'язки між об'єктами із заданої множини. Для створення дендрограми потрібна матриця подібності (або відмінності), яка визначає рівень подібності пари кластерів. Найчастіше використовуються як раз таки агломеративні методи.

Напишемо програму мовою Python для реалізації вищеописаних дій:

```
importnumpyasnp
fromscipy.cluster.hierarchyimportlinkage, dendrogram
importmatplotlib.pyplotasplt

defhierarchical_clustering(data, method='single',
metric='euclidean'):
    """
```

```
Performhierarchicalclusteringonthegivendata.

Parameters:
    - data: numpyarrayorpandasdataframe
    - method: string, oneof 'ingle', 'complete', 'average',
'weighted', 'centroid', 'edian', 'ward'
    - metric: string, oneof 'euclidean', 'cosine', 'cityblock',
'correlation', etc.

Returns:
    - linkagematrix
    """
    # ScalethedatausingStandardScaler
fromsklearn.preprocessingimportStandardScaler
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)

    # Performhierarchicalclustering
linkage_matrix = linkage(scaled_data, method=method,
metric=metric)

    # Plotthedendrogram
plt.figure(figsize=(10, 6))
dendrogram(linkage_matrix, truncate_mode='level', p=3)
plt.title("HierarchicalClusteringDendrogram")
plt.xlabel("Clusters")
plt.ylabel("Distance")
plt.show()

returnlinkage_matrix

# Exampleusage
data = np.array([[0, 2.06, 4.03, 6.32, 2.08],
                 [2.06, 0, 3.5, 4.12, 5.43],
                 [4.03, 3.5, 0, 2.25, 3.65],
                 [6.32, 4.12, 2.25, 0, 4.81],
                 [2.08, 5.43, 3.65, 4.81, 0]])
linkage_matrix = hierarchical_clustering(data, method='single',
metric='euclidean')
```

Продемонструємо роботу програми. Далі наведено результат – скріншот консолі (рис. 5.1):

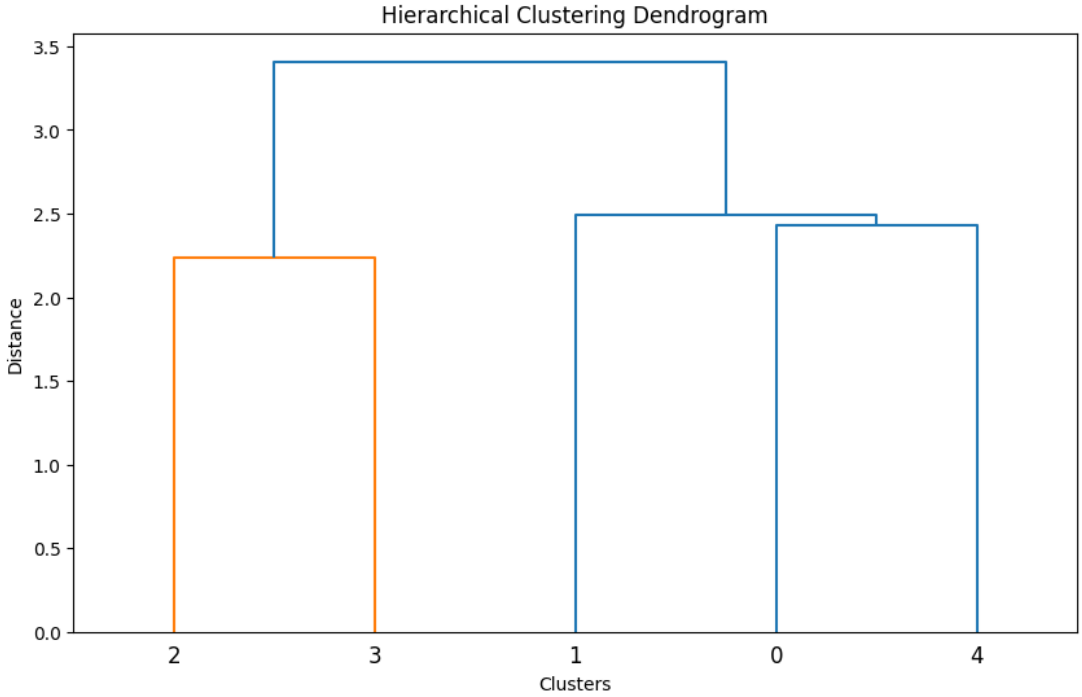


Рис. 5.2. Результат виконання програми

Висновок

У рамках даної лабораторної роботи було проведено ознайомлення з основними поняттями та методами ієрархічного кластерного аналізу, зокрема методу ближчого сусіда (одиночного зв'язку). Під час дослідження був проаналізований принцип роботи методу ближчого сусіда в ієрархічному кластерному аналізі, що дозволило краще зрозуміти його сутність та можливості в застосуванні для кластеризації даних.

Окрім теоретичного аналізу був реалізований алгоритм одиночного зв'язку для кластеризації даних за допомогою мови програмування Python. Програма успішно виконує поставлені задачі та працює у штатному режимі. Цей алгоритм був протестований на вхідних даних, а результати були візуалізовані у вигляді дендрограми.

У результаті аналізу було виявлено, що метод ближчого сусіда дійсно ефективний для кластеризації даних, особливо в тих випадках, коли у даних присутня якась структура близькості чи подібності. Алгоритм показав добрі результати на тестових даних, здатний правильно відокремлювати кластери та визначати структуру даних.

Перелік питань на захист

1. Кластерний аналіз даних. Відмінність кластеризації від класифікації
2. Етапи методу ближнього сусіда
3. Ієрархічна та фасетна класифікація даних

ЛАБОРАТОРНА РОБОТА № 6

Тема: Застосування методів прогнозування у класифікації даних (цін на вино)

Постановка задачі: Зробити візуальний аналіз даних про ціни вин. Розробити модель для прогнозування цін на вино та класифікації майбутніх цін.

Хід роботи:

1) Завантажимо wine.csv файл з даними та відкриємо його у середовищі Orange за допомогою модуля File. Обираємо налаштування параметрів та зберігаємо їх (рис. 6.1):

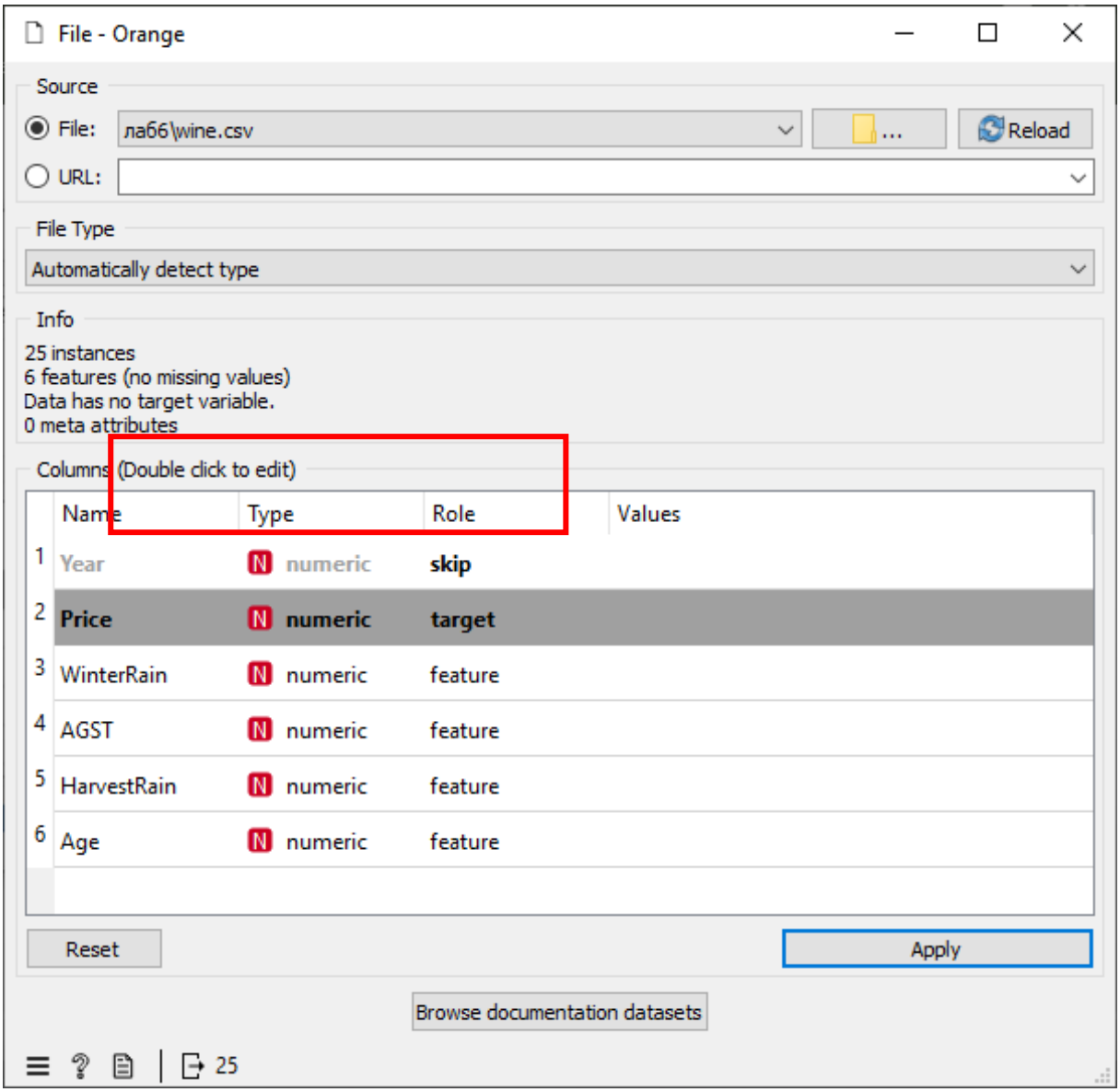
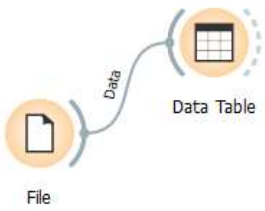


Рис. 6.1. Налаштування змінних

Ми пропускаємо параметр Year, оскільки він вже використовувався в оригінальному дослідженні. Усього дані містять 25 рядків. Щоб переглянути дані, підключимо модуль Data Table та подивимося на вміст файлу (рис. 6.2):



Data Table - Orange

Info
25 instances (no missing data)
4 features
Numeric outcome
No meta attributes.

Variables
Show variable labels (if present)
Visualize numeric values
Color by instance classes

Selection
Select full rows

Restore Original Order
Send Automatically

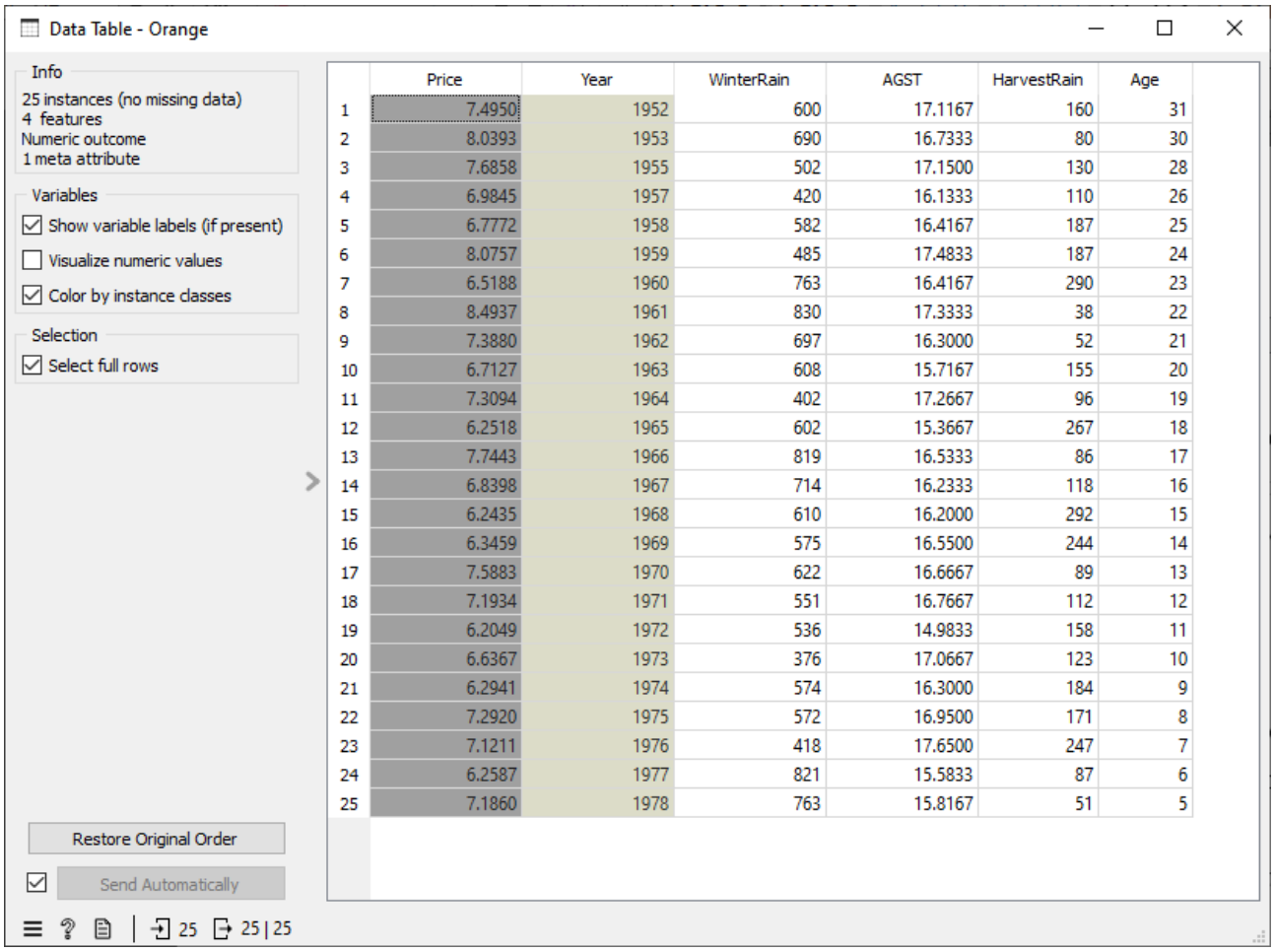
25 | 1 | 25

	Price	WinterRain	AGST	HarvestRain	Age
1	7.4950	600	17.1167	160	31
2	8.0393	690	16.7333	80	30
3	7.6858	502	17.1500	130	28
4	6.9845	420	16.1333	110	26
5	6.7772	582	16.4167	187	25
6	8.0757	485	17.4833	187	24
7	6.5188	763	16.4167	290	23
8	8.4937	830	17.3333	38	22
9	7.3880	697	16.3000	52	21
10	6.7127	608	15.7167	155	20
11	7.3094	402	17.2667	96	19
12	6.2518	602	15.3667	267	18
13	7.7443	819	16.5333	86	17
14	6.8398	714	16.2333	118	16
15	6.2435	610	16.2000	292	15
16	6.3459	575	16.5500	244	14
17	7.5883	622	16.6667	89	13
18	7.1934	551	16.7667	112	12
19	6.2049	536	14.9833	158	11
20	6.6367	376	17.0667	123	10
21	6.2941	574	16.3000	184	9
22	7.2920	572	16.9500	171	8
23	7.1211	418	17.6500	247	7
24	6.2587	821	15.5833	87	6
25	7.1860	763	15.8167	51	5

Рис. 6.2. Використання модуля Data Table для перегляду вмісту файлу

Як бачимо, наш цільовий параметр Price виділений сірим кольором. Параметр WinterRain визначає рівень опадів взимку, AGST – середня сезонна температура, HarvestRain – рівень опадів влітку та параметр Age – вік вина. Усі параметри є числовими.

Включимо параметр Year та подивимось на дані ще раз (рис. 6.3):



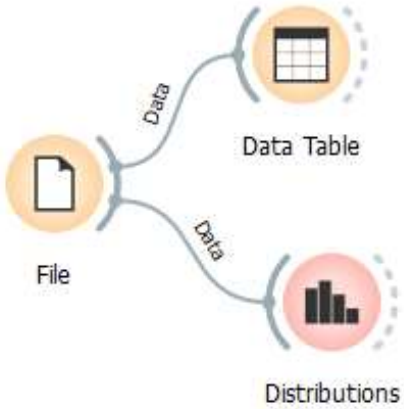
	Price	Year	WinterRain	AGST	HarvestRain	Age
1	7.4950	1952	600	17.1167	160	31
2	8.0393	1953	690	16.7333	80	30
3	7.6858	1955	502	17.1500	130	28
4	6.9845	1957	420	16.1333	110	26
5	6.7772	1958	582	16.4167	187	25
6	8.0757	1959	485	17.4833	187	24
7	6.5188	1960	763	16.4167	290	23
8	8.4937	1961	830	17.3333	38	22
9	7.3880	1962	697	16.3000	52	21
10	6.7127	1963	608	15.7167	155	20
11	7.3094	1964	402	17.2667	96	19
12	6.2518	1965	602	15.3667	267	18
13	7.7443	1966	819	16.5333	86	17
14	6.8398	1967	714	16.2333	118	16
15	6.2435	1968	610	16.2000	292	15
16	6.3459	1969	575	16.5500	244	14
17	7.5883	1970	622	16.6667	89	13
18	7.1934	1971	551	16.7667	112	12
19	6.2049	1972	536	14.9833	158	11
20	6.6367	1973	376	17.0667	123	10
21	6.2941	1974	574	16.3000	184	9
22	7.2920	1975	572	16.9500	171	8
23	7.1211	1976	418	17.6500	247	7
24	6.2587	1977	821	15.5833	87	6
25	7.1860	1978	763	15.8167	51	5

Рис. 6.3. Використання модуля Data Table для перегляду вмісту файлу

Можна побачити, що дані наявні не для усіх років. Так, наприклад, у нас відсутні дані для 1954 та 1956 років. Це може бути пов’язано з тим, що врожай тих років не залишився до того року, в який було зібрано дані для аналізу.

Також варто звернути увагу, що замість безпосередньої ціни на вино у таблиці записані логарифми ціни. Це використовується у побудові моделі з метою можливості урахування інфляції.

2) Тепер проаналізуємо одномірний розподіл атрибутів за допомогою модуля Distribution. Подивимося розподіл цін (рис. 6.4). З нього можна побачити, що дорогих вин менше, а дешевих – більше. Далі подивимося розподіл опадів взимку (рис. 6.5), середню температуру (рис. 6.6), опади влітку(рис. 6.7).



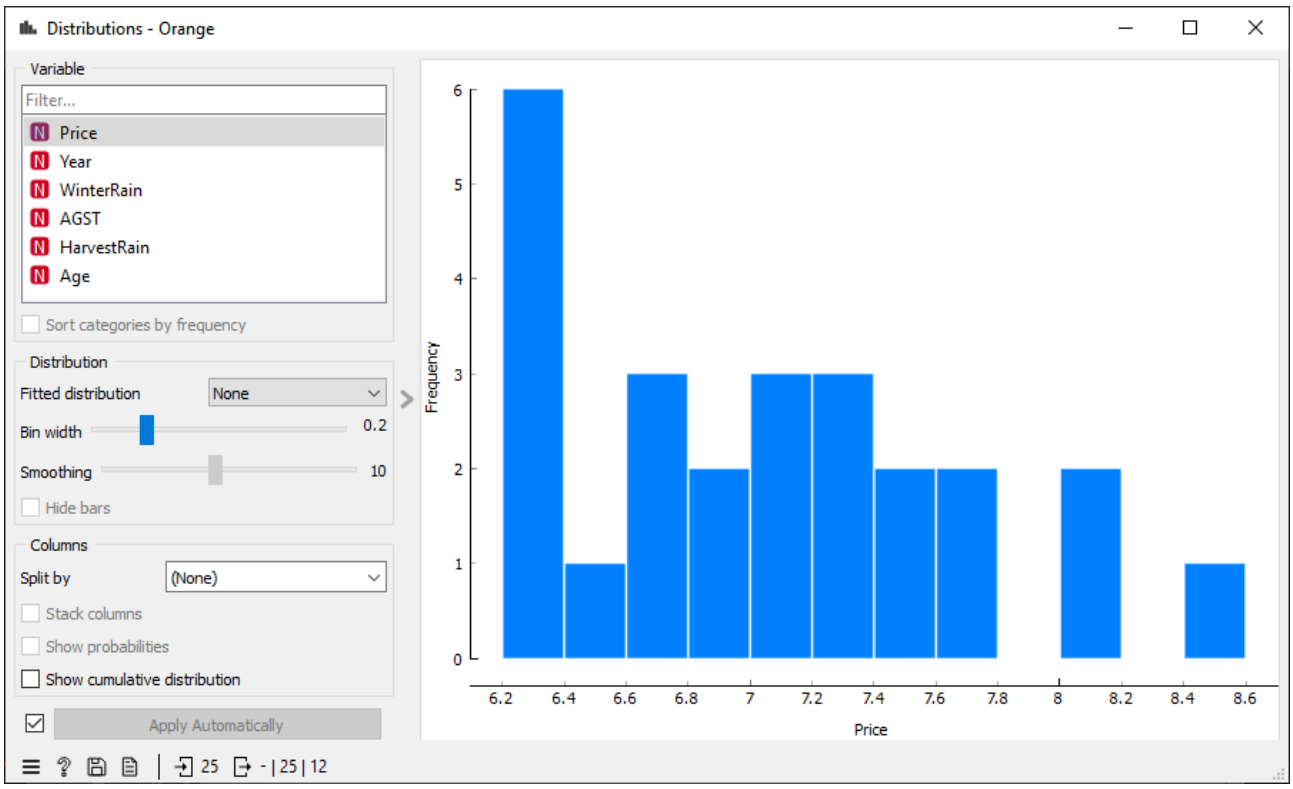


Рис. 6.4. Використання модуля Distribution для перегляду розподілу атрибутів – Price

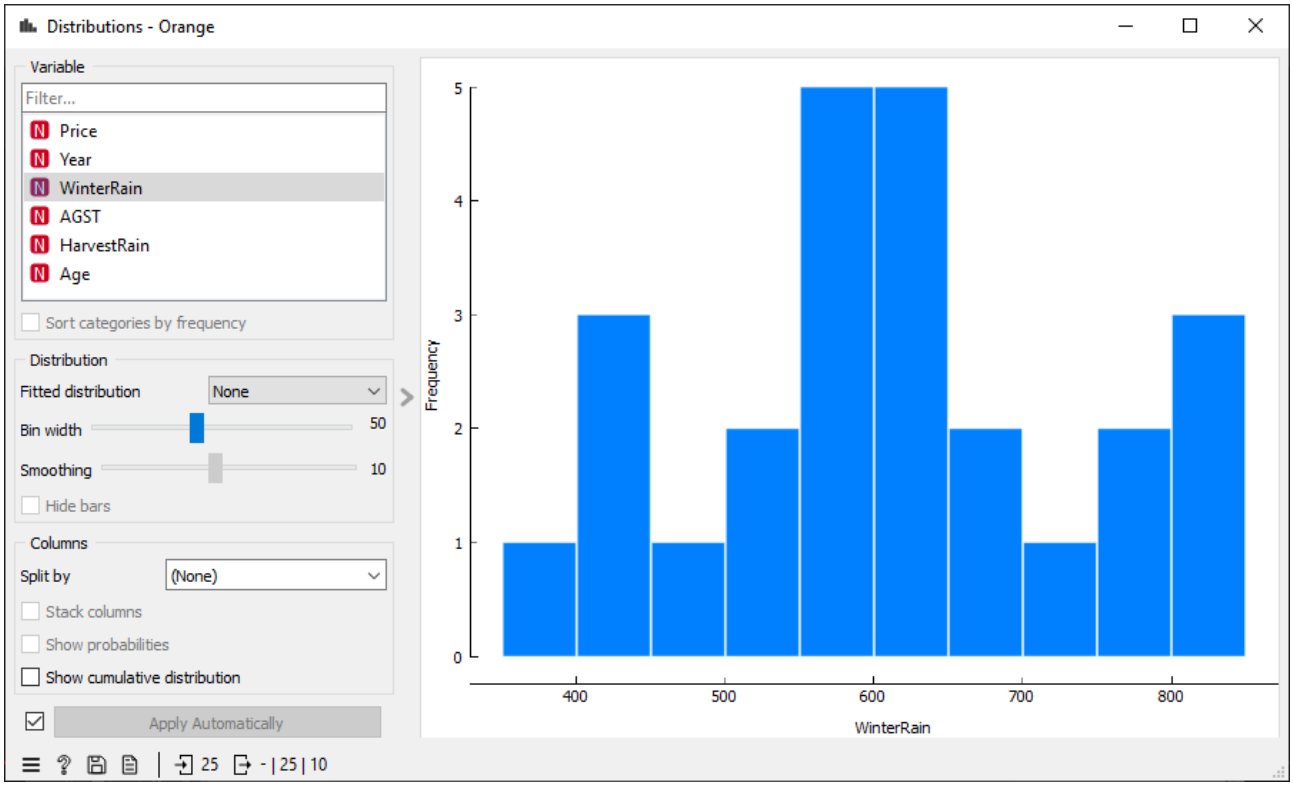


Рис. 6.5. Використання модуля Distribution для перегляду розподілу атрибутів – WinterRain

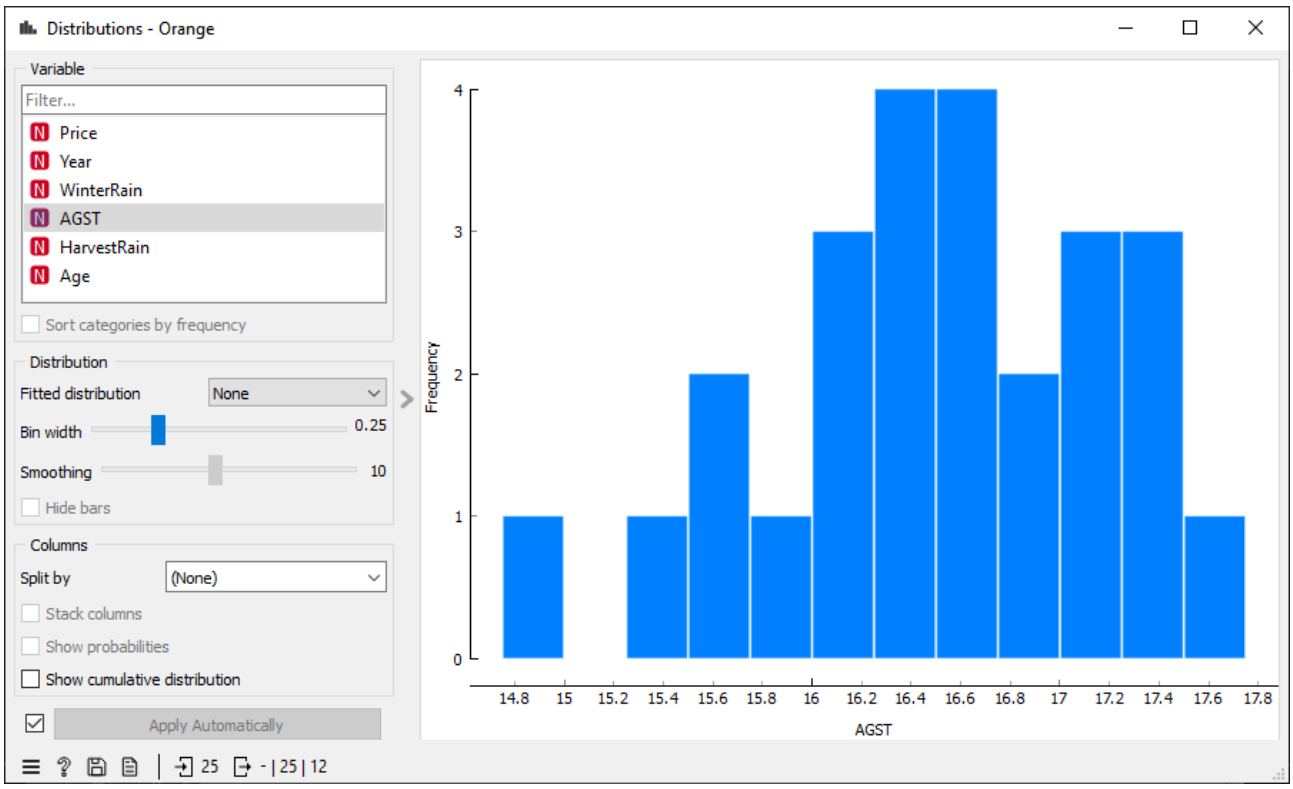


Рис. 6.6. Використання модуля Distribution для перегляду розподілу атрибутів – AGST

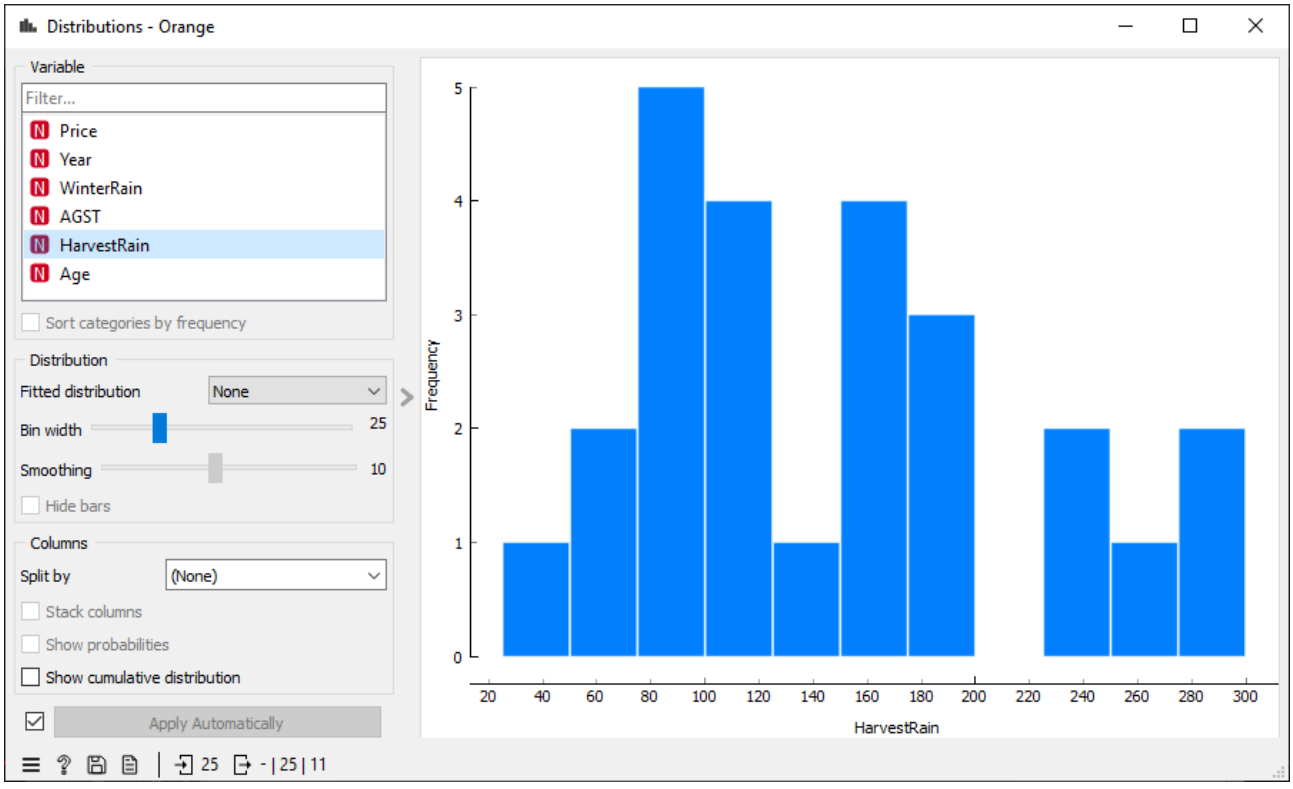
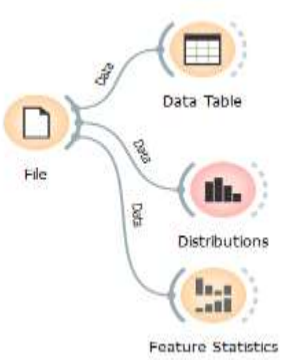


Рис. 6.7. Використання модуля Distribution для перегляду розподілу атрибутів – HarvestRain

3) Далі проаналізуємо статистику атрибутів за допомогою модуля Feature Statistics (рис. 6.8). За допомогою нього можна подивитися різноманітну статистику наших параметрів – мінімальне, максимальне та середнє значення, а також дисперсію.



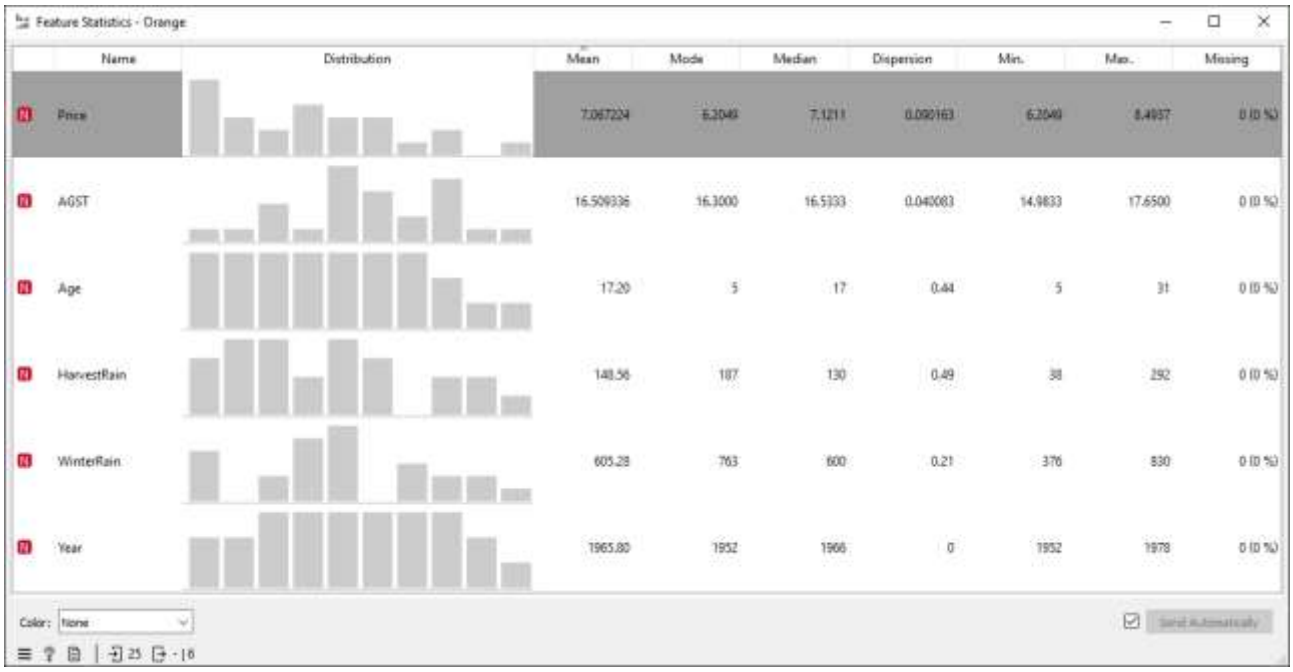
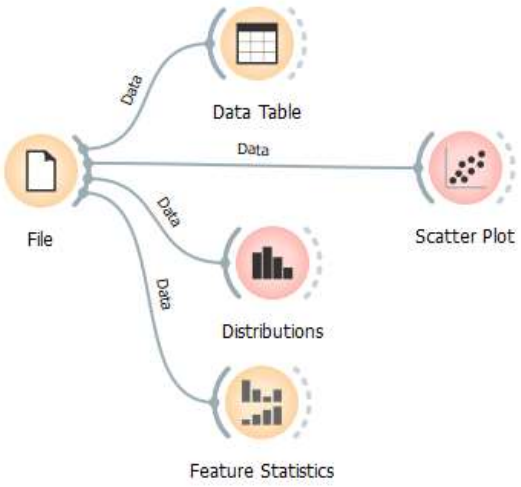


Рис. 6.8. Використання модуля Feature Statistics для аналізу статистики атрибутів

4) Провізуалізуємо дані. Для цього створимо графік за допомогою модуля Scatter Plot (рис. 6.9). На осі X обираємо параметр температури, на осі У – параметр ціни. З цього графіку бачимо прямий зв'язок, що в середньому зі зростанням температури зростає і ціна. Можна ввімкнути показ Regression Line (рис. 6.10) та побачити параметр кореляції.



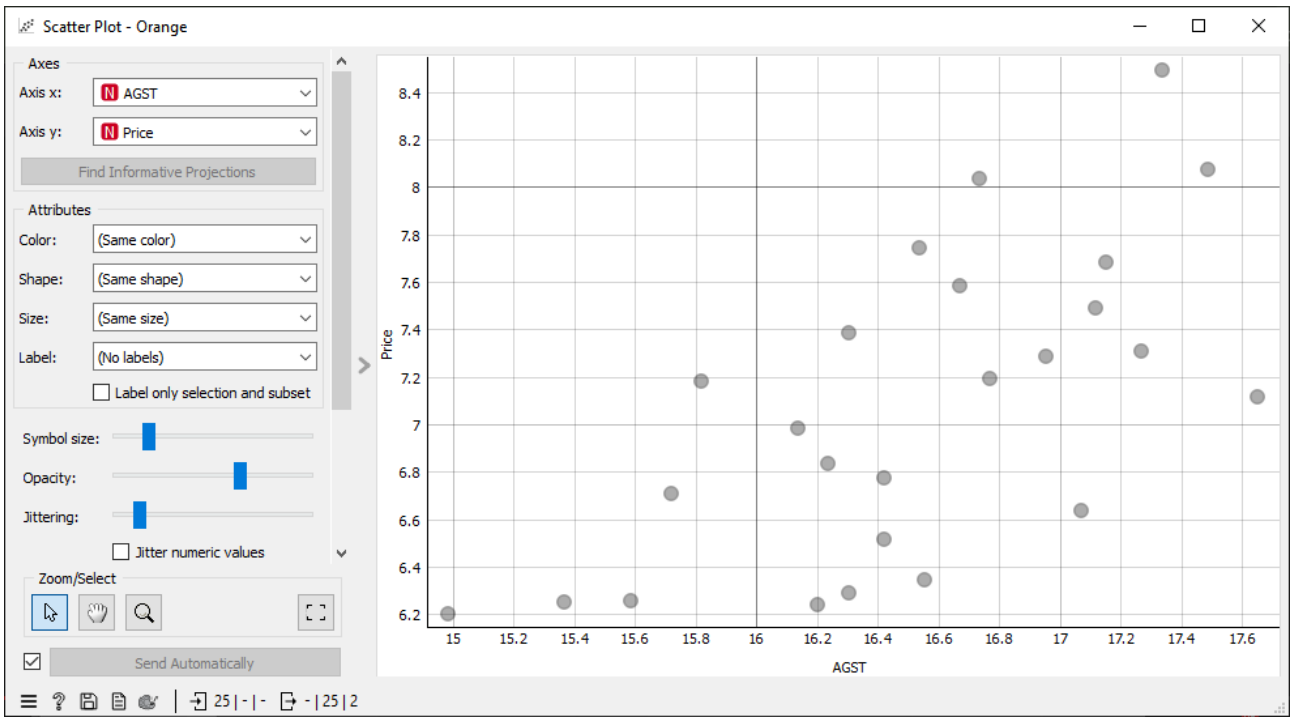


Рис. 6.9. Використання модуля Scatter Plot для візуалізації (графік залежності температури та ціни)

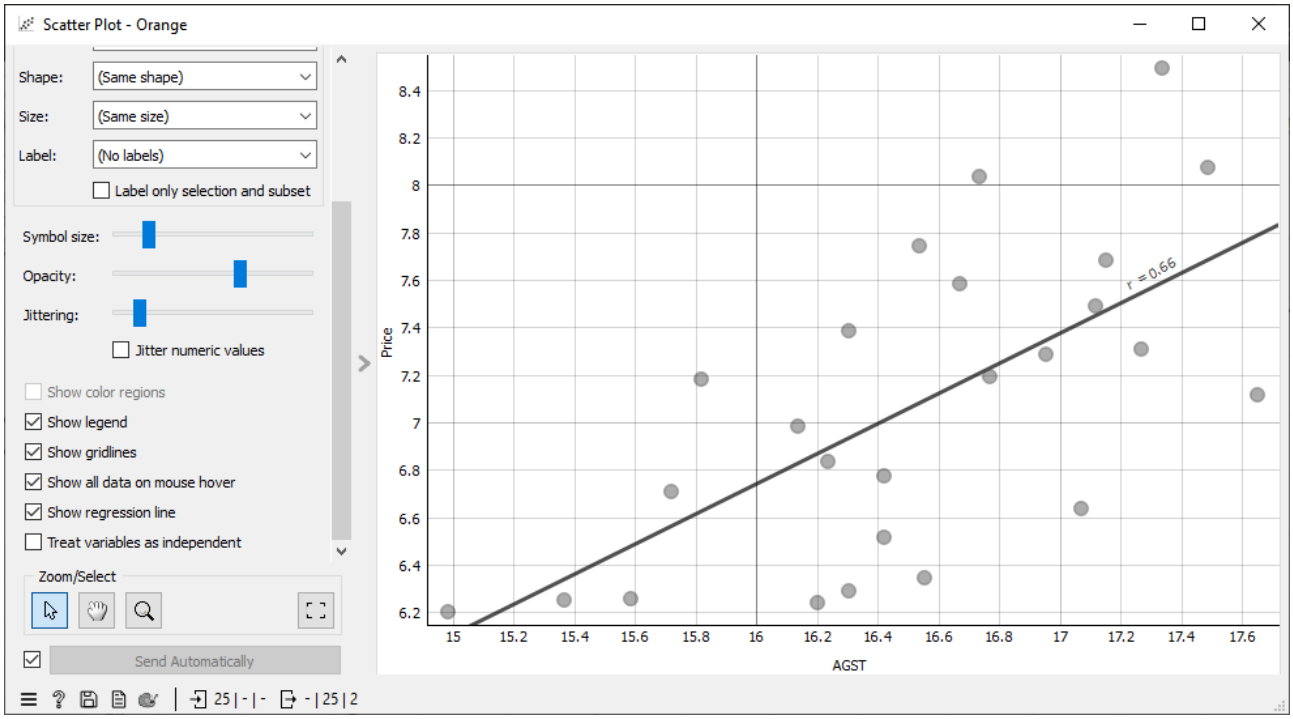


Рис. 6.10. Показ Regression Line

Подивимося ще графік залежності дощу влітку та ціни (рис. 6.11). На ньому бачимо зворотній зв’язок: коли дощів влітку було більше, то ціна була меншою.

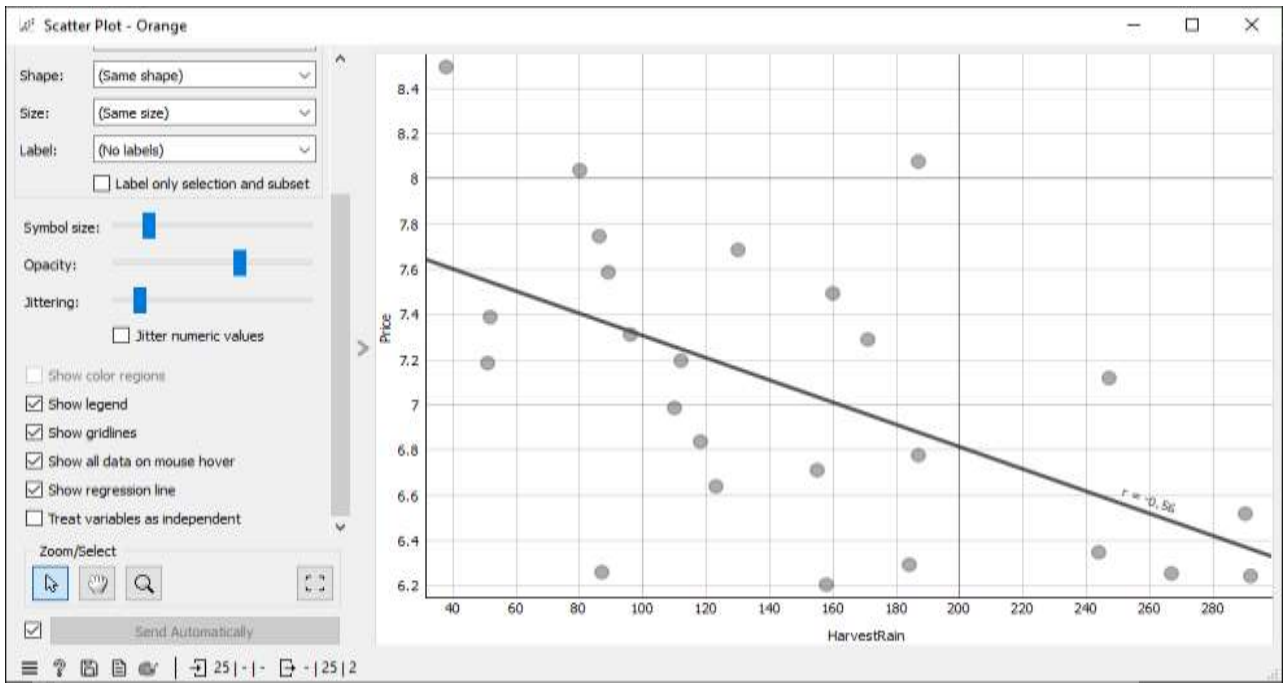


Рис. 6.11. Використання модуля Scatter Plot для візуалізації (графік залежності дощу влітку та ціни)

І ще подивимося графік із опадами взимку (рис. 6.12). В даному випадку зв’язок дуже малий – тобто цей параметр впливає на ціну зовсім мало.

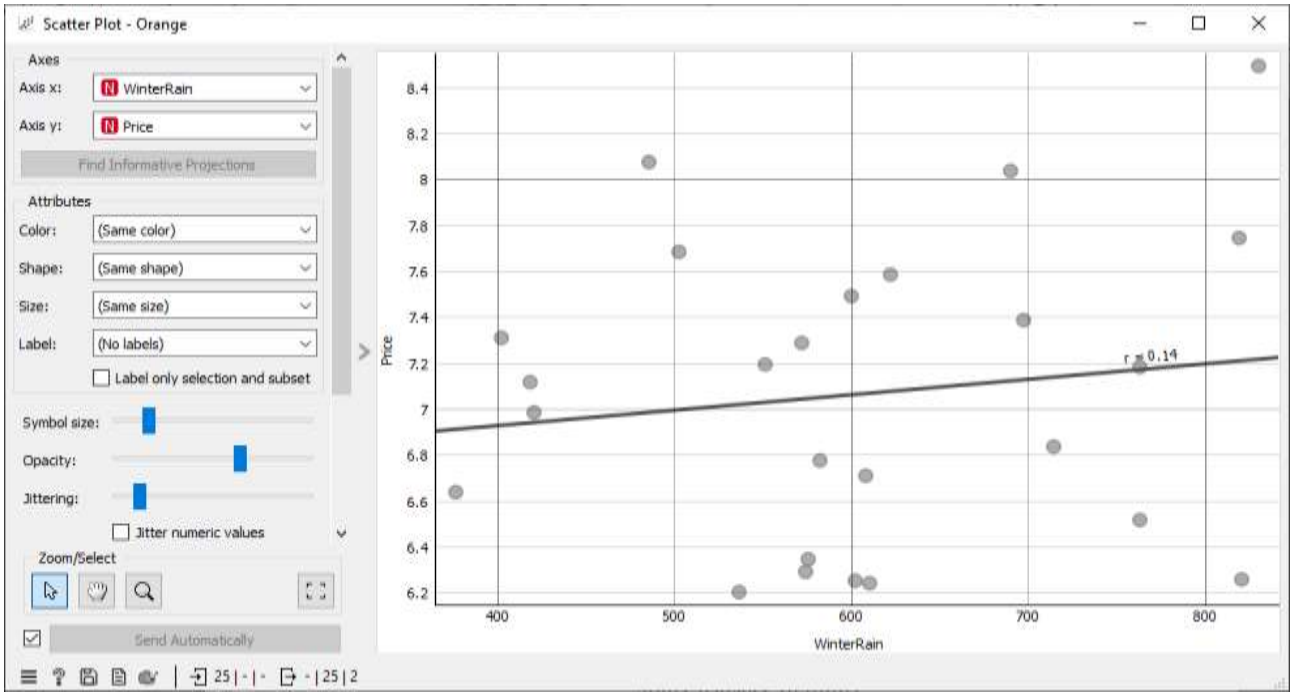


Рис. 6.12. Використання модуля Scatter Plot для візуалізації (графік залежності дощу взимку та ціни)

Переглянемо графік залежності віку та ціни (рис. 6.13). На ньому бачимо прямий зв’язок – тобто чим старше вино тим воно дорожче.

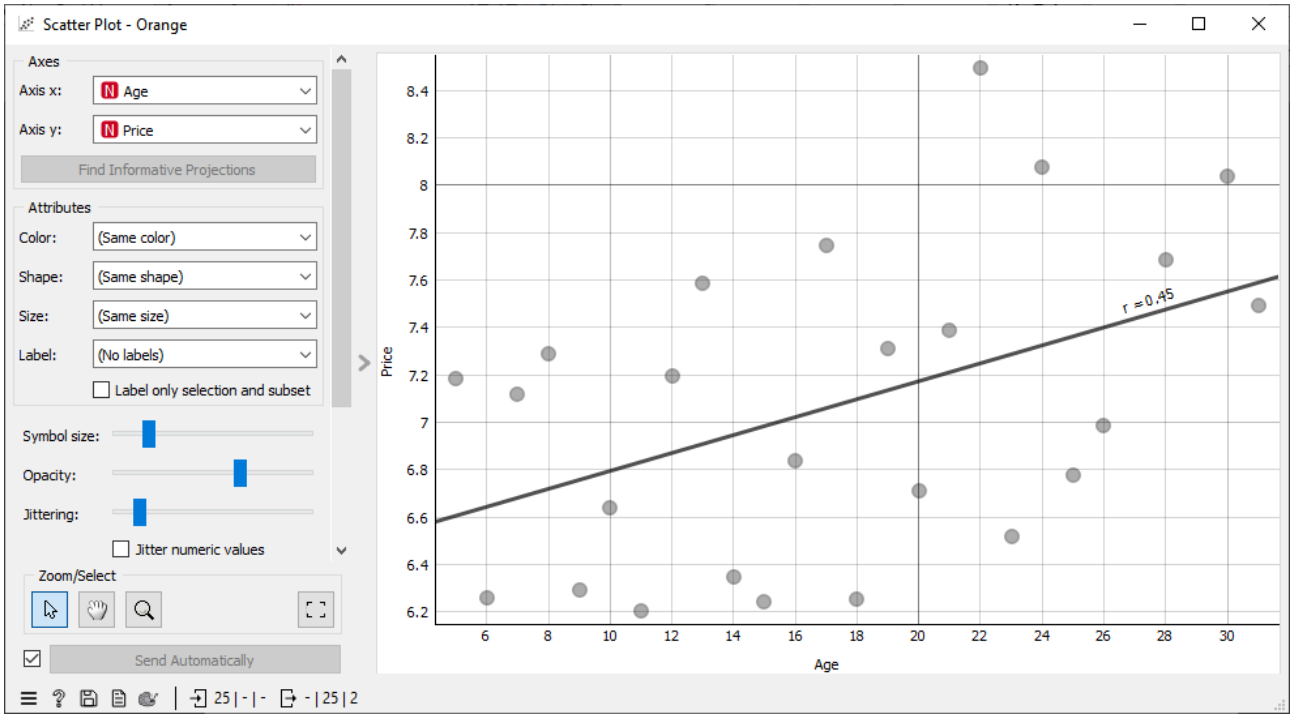


Рис. 6.13. Використання модуля Scatter Plot для візуалізації (графік залежності віку та ціни)

З аналізу візуалізованих даних можна сказати, що деякі атрибути мають позитивну кореляцію з цільовою змінною – ціна, деякі – від’ємну, а також деякі мають дуже маленьку кореляцію.

5) Можна ще провізуалізувати дані за допомогою тримірного зображення, а саме за допомогою додавання кольору (рис. 6.14). З легенди бачимо, що більш дешевші вина позначені синім кольором, а більш дорожчі – жовтим. Можна зробити висновок, що найдорожчі вина в більшості були тоді, коли температура влітку була найбільшою, а опадів було мало, а в протилежних умовах – дешевші вина. Тобто дані можна легко візуально розбити на дві групи.

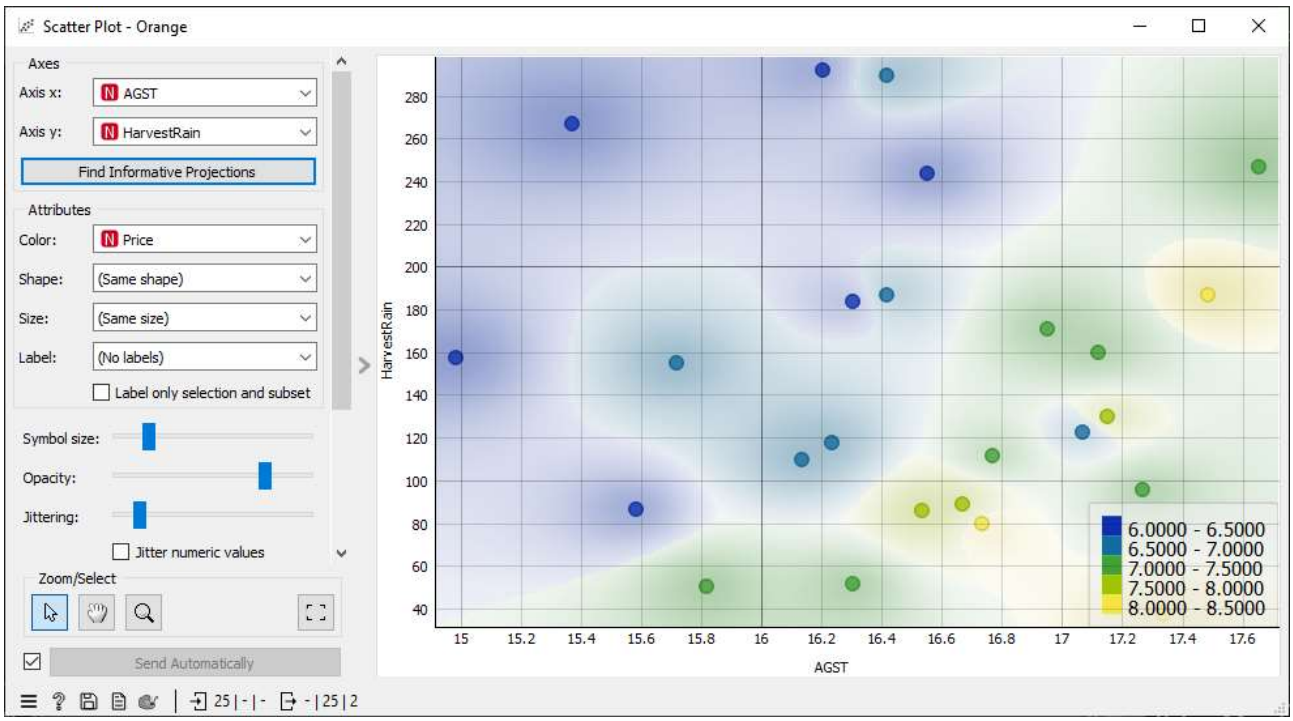


Рис. 6.14. Використання модуля Scatter Plot для візуалізації (графік залежності температури та дощу влітку і розподілу цін згідно з цією залежністю)

6) Побудуємо модель для прогнозування цін вина. Коли професор Ашенфелтер продемонстрував у своєму дослідженні можливість прогнозування ціни на вино, навіть не смакуючи його, то експерти по вінам були обурені [5].

Для побудови моделі (рис. 6.15) ми використовуємо ту саму модель, що й в дослідженні, а саме – лінійну регресію. Для цього нам знадобляться модулі Linear Regression та Test and Score, перший з яких – будує модель, а другий – перевіряє нашу модель. Також знадобиться Data Table, в якому буде результат.

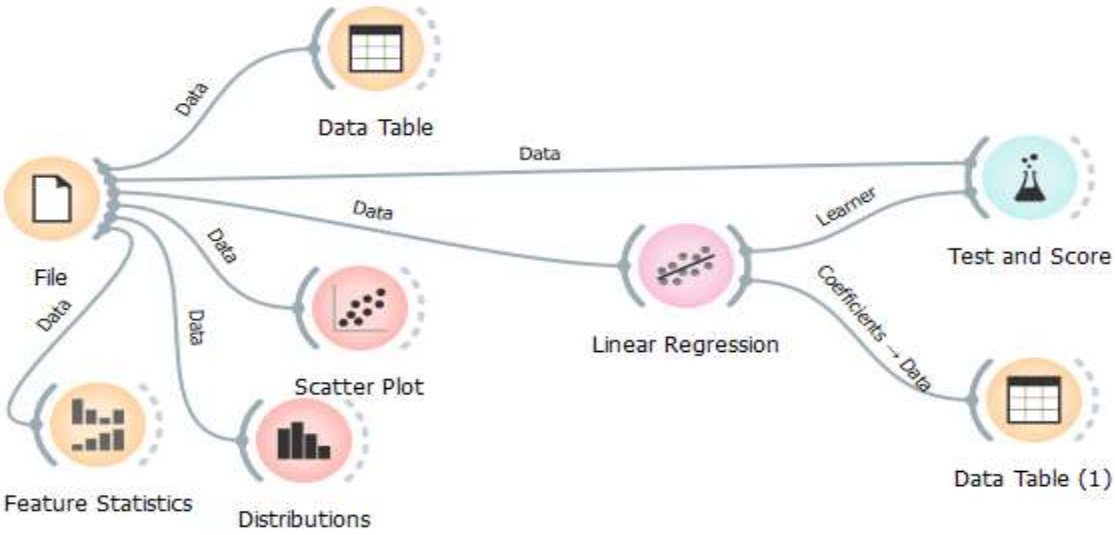


Рис. 6.15. Побудова моделі з використанням модулів Linear Regression та Test and Score

Подивимося вміст Test and Score (рис. 6.16). Отримали значення $R^2 = 0.83$. Значення цього параметру є кращим тоді, коли воно ближче до одиниці. Згідно з цим отримана модель є непоганою.

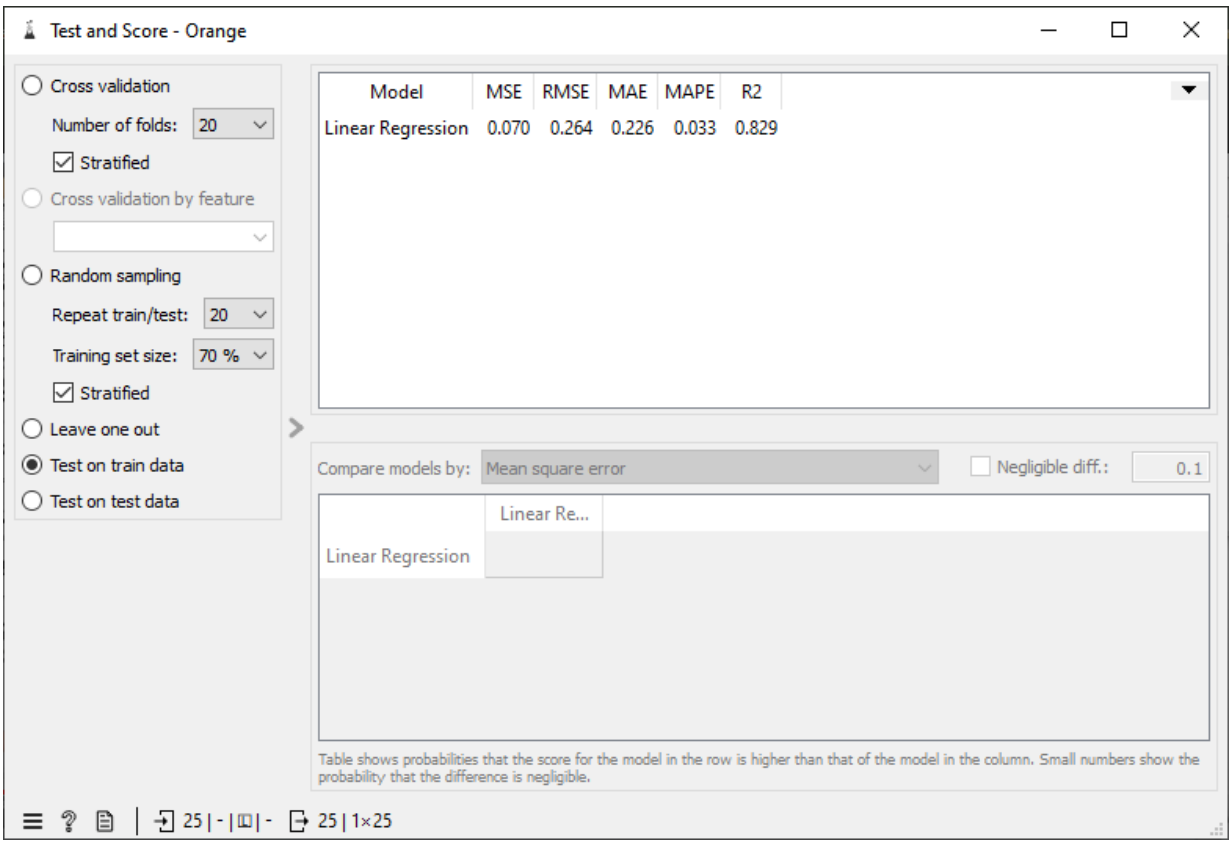


Рис. 6.16. Результати Test and Score

Переглянемо вміст Data Table(1) (рис. 6.17).

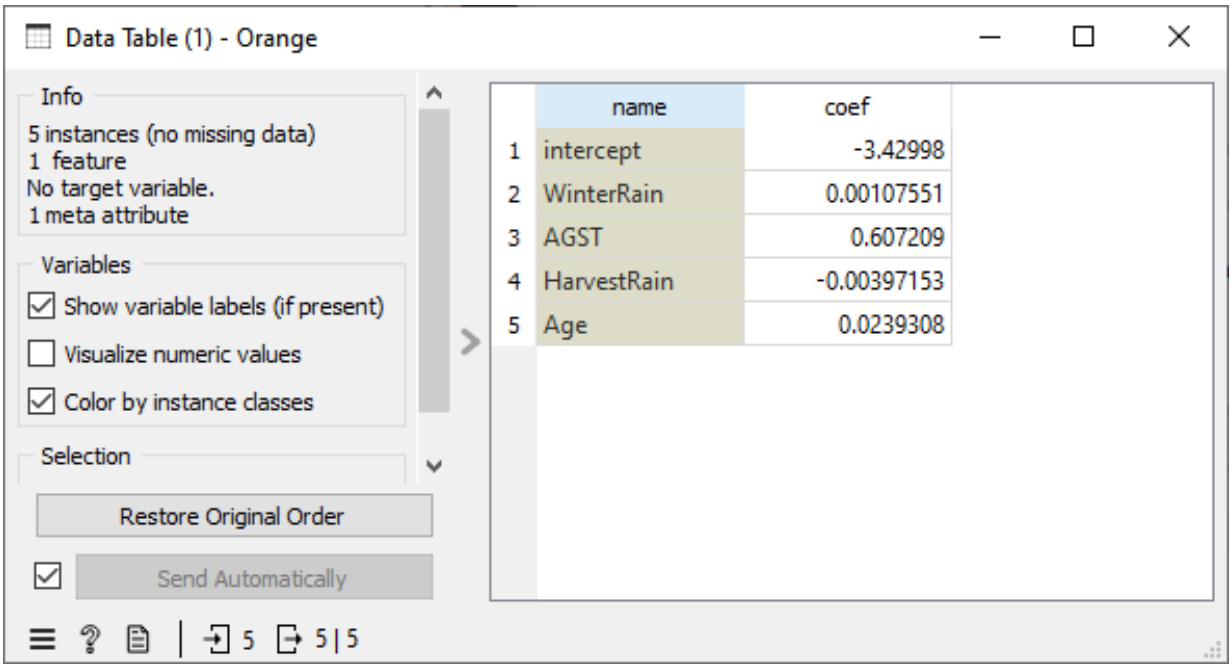


Рис. 6.17. Вміст Data Table моделі

Отримані значення параметрів являють собою коефіцієнти рівняння, необхідного для прогнозування ціни вина:

$$\log(Price) = -3.4 + 0.001 * WinterRain + 0.607 * AGST - 0.00397 * HarvestRain + 0.0239 * Age$$

Висновок (приклад)

У рамках даної лабораторної роботи зроблено візуальний аналіз даних про ціни вин та побудовано модель для прогнозування ціни вина.

В ході аналізу отримано висновок, що деякі атрибути мають позитивну кореляцію з цільовою змінною – ціна, деякі – від'ємну, а також деякі мають маленьку кореляцію. З іншого боку за допомогою візуалізації можна виявити певну категоризацію вин за ціною в залежності від інших параметрів.

Побудована модель на основі лінійної регресії має достатньо високу тестову оцінку та задовольняє умови прогнозування та класифікації цін на вино. У результаті отримано коефіцієнти лінійного рівняння, яке і надає можливість прогнозування ціни вина. Поставлене завдання виконано. Отримано коректну модель для прогнозування у середовищі Orange.

Перелік питань на захист

1. Показники точності математичної моделі прогнозу та класифікації даних
2. Типові кластери та моделі
3. Вимоги до вхідних даних у кластерному аналізу

ЛАБОРАТОРНА РОБОТА № 7

Тема: Методи ієрархічного кластерного аналізу. Повний зв'язок та незважене попарне середнє

Постановка задачі: Ознайомитися з основними поняттями і методами ієрархічного кластерного аналізу. Дослідити та проаналізувати принцип роботи методів повного зв'язку та незваженого попарного середнього в ієрархічному кластерному аналізі. Реалізувати вищезазначені алгоритми для кластеризації даних. Провести тестування реалізованих алгоритмів на штучних даних та проаналізувати результати.

Теоретичні відомості

Кластеризація (або кластерний аналіз) – це завдання розбиття безлічі об'єктів на групи, які називаються кластерами. Усередині кожної групи повинні виявитися «схожі» об'єкти, а об'єкти різних групи мають бути якомога більш відмінними. Головна відмінність кластеризації від класифікації у тому, що перелік груп чітко не заданий й у процесі роботи алгоритму.

Застосування кластерного аналізу у загальному вигляді зводиться до наступних етапів:

- Вибір вибірки об'єктів для кластеризації.
- Визначення безлічі змінних, якими оцінюватимуться об'єкти у вибірці. За потреби – нормалізація значень змінних.
- Обчислення значень міри схожості між об'єктами.
- Застосування методу кластерного аналізу до створення груп подібних об'єктів (кластерів).
- Подання результатів аналізу.

Після отримання та аналізу результатів можливе коригування обраної метрики та методу кластеризації до отримання оптимального результату.

Хід роботи:

Розглянемо один із способів розподілу об'єктів за групами – ***агломеративний метод ієрархічної кластеризації***. Він полягає у послідовному поєднанні точок у кластери. При цьому спочатку кожен об'єкт лежить в окремій групі, потім на кожному кроці найближчі кластери об'єднуються на підставі вибраних метрик відстані. Тобто дерево створюється від листків до стовбуру.

Для побудови матриці подібності (відмінності) необхідно поставити міру відстані між двома кластерами. Найчастіше використовуються такі методи

визначення відстані: метод одиночного зв'язку, повного зв'язку, центроїдний метод та метод Уорда.

У даній роботі як дистанцію між кластерами використаємо два методи:

1. **Метод повного зв'язку** (англ. complete linkage) також відомий, як «метод дальнього сусіда». Відстань між двома кластерами вважається рівною максимальної відстані між двома елементами з різних кластерів: $\max\{d(a,b) : a \in A, b \in B\}$.

2. **Незважене попарне середнє** – у цьому методі відстань між двома різними кластерами обчислюється як середня відстань між усіма парами об'єктів у них. Метод ефективний, коли об'єкти формують різні групи, проте він працює однаково добре і у випадках протяжних («ланцюжкового» типу) кластерів.

Як метрика відстані між точками зазвичай використовується ***евклідова міра*** (також підтримується багато інших, наприклад, кореляція, косинусна відмінність).

Завдання

Виконаємо кластеризацію зазначеними методами та візуалізуємо набір даних. Дана матриця відстаней:

$$A = \begin{matrix} 0 & 1.05 & 2.23 & 5.48 & 6.01 \\ 1.05 & 0 & 2.5 & 3.38 & 6.23 \\ 2.23 & 2.5 & 0 & 1.25 & 2.54 \\ 5.48 & 3.38 & 1.25 & 0 & 3.71 \\ 6.01 & 6.23 & 2.54 & 3.71 & 0 \end{matrix}$$

Для візуалізації використовується ***дендрограма***. Під дендрограмою зазвичай розуміється дерево, побудоване за матрицею мір близькості. Дендрограма дозволяє зобразити взаємні зв'язки між об'єктами із заданої множини. Для створення дендрограми потрібна матриця подібності (або відмінності), яка визначає рівень подібності пари кластерів. Найчастіше використовуються як раз таки агломеративні методи.

Напишемо програму мовою Python для реалізації вищеописаних дій:

```
import numpy as np
from scipy.cluster.hierarchy import linkage, dendrogram
import matplotlib.pyplot as plt

def hierarchical_clustering(data, method='single', metric='euclidean'):
    """
    Perform hierarchical clustering on the given data.
```

```
    Parameters:
    - data: numpy array or pandas dataframe
    - method: string, one of 'single', 'complete', 'average', 'weighted', 'centroid', 'edian',
'ward'
    - metric: string, one of 'euclidean', 'cosine', 'cityblock', 'correlation', etc.

    Returns:
    - linkage matrix
    """
    # Scale the data using StandardScaler
    from sklearn.preprocessing import StandardScaler
    scaler = StandardScaler()
    scaled_data = scaler.fit_transform(data)

    # Perform hierarchical clustering
    linkage_matrix = linkage(scaled_data, method=method, metric=metric)

    # Plot the dendrogram
    plt.figure(figsize=(10, 6))
    dendrogram(linkage_matrix, truncate_mode='level', p=3)
    plt.title(f"Hierarchical Clustering Dendrogram. {method}")
    plt.xlabel("Clusters")
    plt.ylabel("Distance")
    plt.show()

    return linkage_matrix

# Example usage
data = np.array([[0, 1.05, 2.23, 5.48, 6.01],
                 [1.05, 0, 2.5, 3.38, 6.23],
                 [2.23, 2.5, 0, 1.25, 2.54],
                 [5.48, 3.38, 1.25, 0, 3.71],
                 [6.01, 6.23, 2.54, 3.71, 0]])

linkage_matrix = hierarchical_clustering(data, method='complete',
metric='euclidean')
linkage_matrix = hierarchical_clustering(data, method='average', metric='euclidean')
```

Продемонструємо роботу програми. Далі наведено результат – скріншот консолі, а саме – дві дендрограми, перша з яких отримана методом повного зв'язку, а інша – незваженого попарного середнього (рис. 7.1):

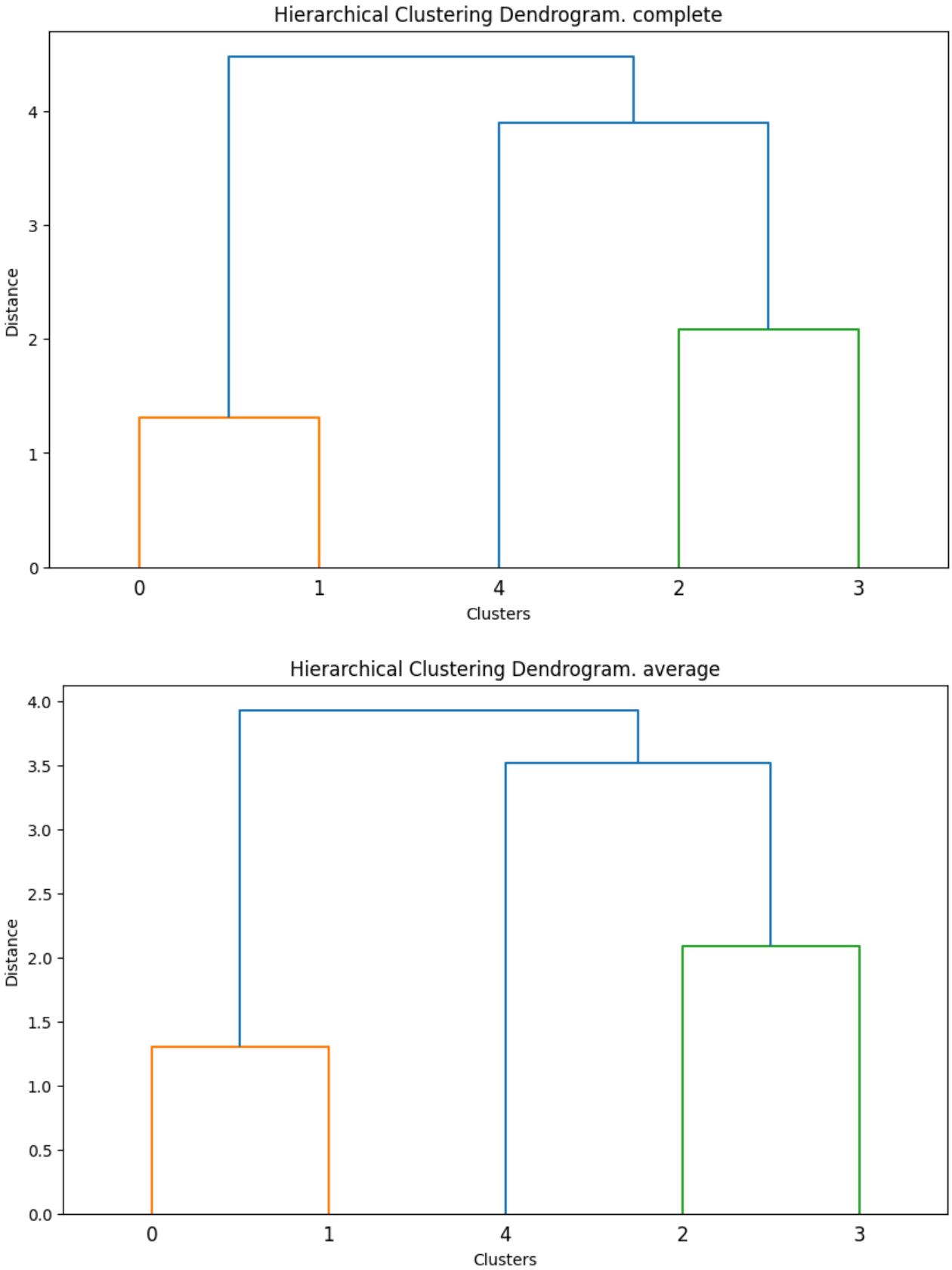


Рис. 7.1. Результат виконання програми

Висновок

У рамках даної лабораторної роботи було проведено ознайомлення з основними поняттями та методами ієрархічного кластерного аналізу, зокрема методами повного зв'язку та методом незваженого попарного середнього. Під час дослідження був проаналізований принцип роботи даних методів ієрархічному кластерному аналізу, що дозволило краще зрозуміти їх сутність та можливості в застосуванні для кластеризації даних.

Окрім теоретичного аналізу, були реалізовані алгоритми даних методів для кластеризації даних за допомогою мови програмування Python. Програма успішно виконує поставлені задачі та працює у штатному режимі. Ці алгоритми протестовані на вхідних даних, а результати візуалізовані у вигляді дендрограм.

У результаті аналізу було виявлено, що метод повного зв'язку (complete linkage) є ефективним для кластеризації даних, особливо в тих випадках, коли важливо максимально збільшити відстань між кластерами. Метод повного зв'язку забезпечує компактність кластерів, що дозволяє уникнути злиття значно віддалених точок, тим самим сприяючи чіткішому розмежуванню кластерів.

У свою чергу метод невзваженого попарного середнього (average linkage) є ефективним для кластеризації даних, особливо коли необхідно врахувати середню відстань між точками кластерів. Метод невзваженого попарного середнього добре підходить для даних з рівномірно розподіленими кластерами, оскільки він враховує середні відстані, що дозволяє отримувати збалансовані кластери [6, 7].

Перелік питань на захист

1. Етапи кластерного аналізу
2. Етапи методу повного зв'язку
3. Етапи методу незваженого попарного середнього

ЛАБОРАТОРНА РОБОТА № 8

Тема: Центроїдна модель кластеризації. Метод К-середніх

Постановка задачі: Досліджувати та відтворити кластеризацію методом К-середніх.

Теоретичні відомості

Існує безліч алгоритмів кластеризації, проте нижче буде розглянуто метод k-середніх, оскільки він є найлаконічнішим і найпростішим для розуміння.

Кластеризація методом k-середніх [8] :

- Вихідним завданням буде розподіл довільної кількості n-вимірних точок по k кластерів.
- Випадковим чином створюються k точок, надалі називатимемо їх центрами кластерів;
- Для кожної точки ставиться відповідно до найближчого до неї центр кластера;
- Обчислюються середні арифметичні точки, що належать до певного кластера. Саме ці значення стають новими центрами кластерів;
- Кроки 2 і 3 повторюються до тих пір, поки перерахунок центрів кластерів приноситиме плоди. Як тільки вирахувані центри кластерів збігатимуться з попередніми, алгоритм буде закінчено.

Хід роботи:

Алгоритм k-means (k-середніх) найбільш простий і швидкий, але в той же час досить неточний метод кластеризації у класичній реалізації. Він розбиває безліч елементів векторного простору на заздалегідь відоме число кластерів k. Дія алгоритму така, що він прагне мінімізувати середньоквадратичне відхилення на точках кожного кластера. Основна ідея полягає в тому, що на кожній ітерації перераховується центр мас для кожного кластера, отриманого на попередньому кроці, потім вектори розбиваються на кластери знову відповідно до того, який з нових центрів виявився ближчим за обраною метрикою. Алгоритм завершується, коли на якійсь ітерації немає зміни кластерів.

Головні проблеми алгоритму k-means:

- Необхідно заздалегідь знати кількість кластерів.
- Алгоритм дуже чутливий до вибору початкових центрів кластерів. Класичний варіант має на увазі випадковий вибір кластерів, що дуже часто було джерелом похибки. Як варіант рішення необхідно проводити

дослідження об'єкта для більш точного визначення центрів початкових кластерів.

- Не справляється із завданням, коли об'єкт належить до різних кластерів рівною мірою або не належить жодному. Спрямований на виділення круглих кластерів.

Завдання

Поспостерігаємо за даним методом кластеризації у середовищі Orange. Для цього створимо у модулі Paint Data тестові дані, намалювавши більшою мірою виокремлені три кластери (рис. 8.1). Нехай це будуть дані про співвідношення ціни та якості певного товару.

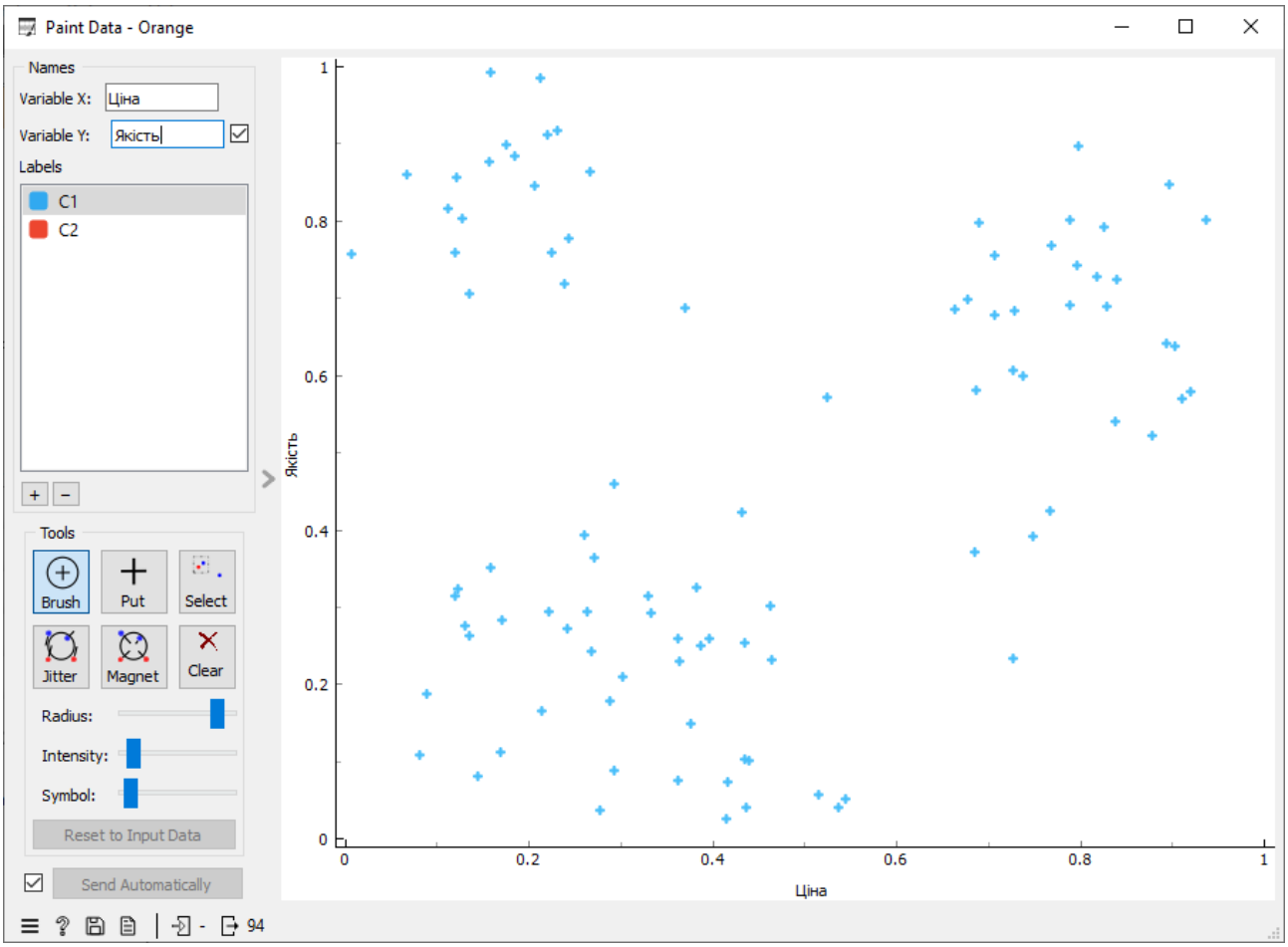


Рис. 8.1. Створення тестових даних у модулі Paint Data

Далі підключимо модуль K-Means. Обираємо у налаштуваннях кількість кластерів (рис. 8.2). Спочатку обираємо три, оскільки в цілому ми й самі бачимо, що візуально виділяється саме три кластери.

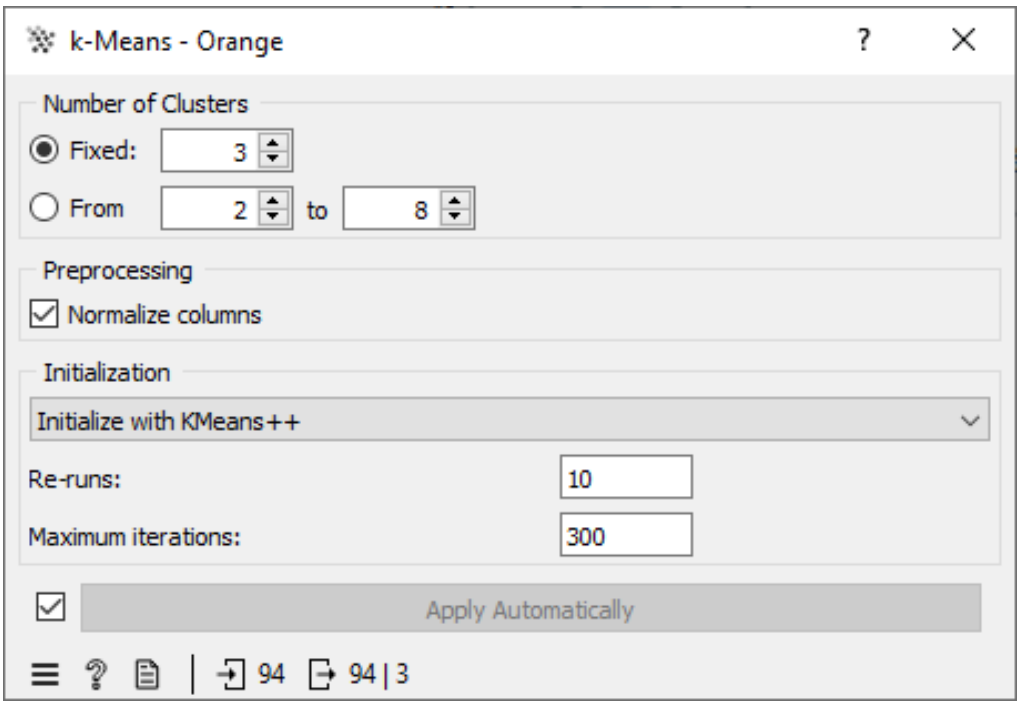


Рис. 8.2. Використання модуля k-means

Для того, щоб подивитися на результат, підключаємо модуль Scatter Plot (рис. 8.3).

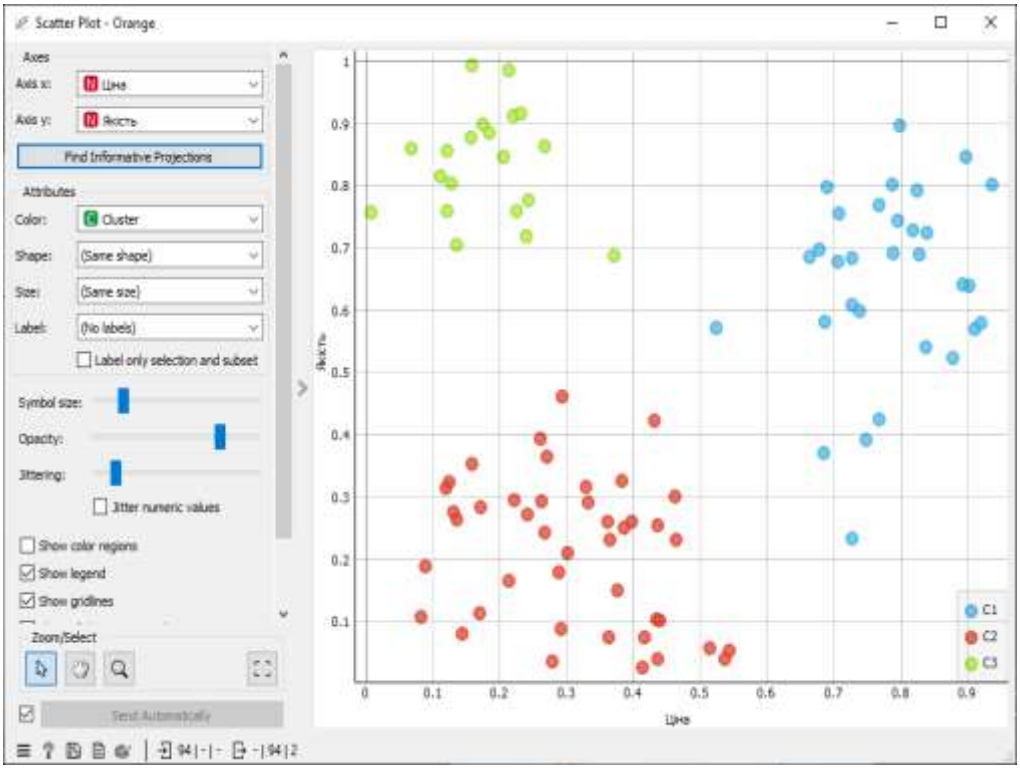


Рис. 8.3. Кластеризація методом k-means

Тепер спробуємо вибрати інші кількості кластерів (рис. 8.4–8.5):

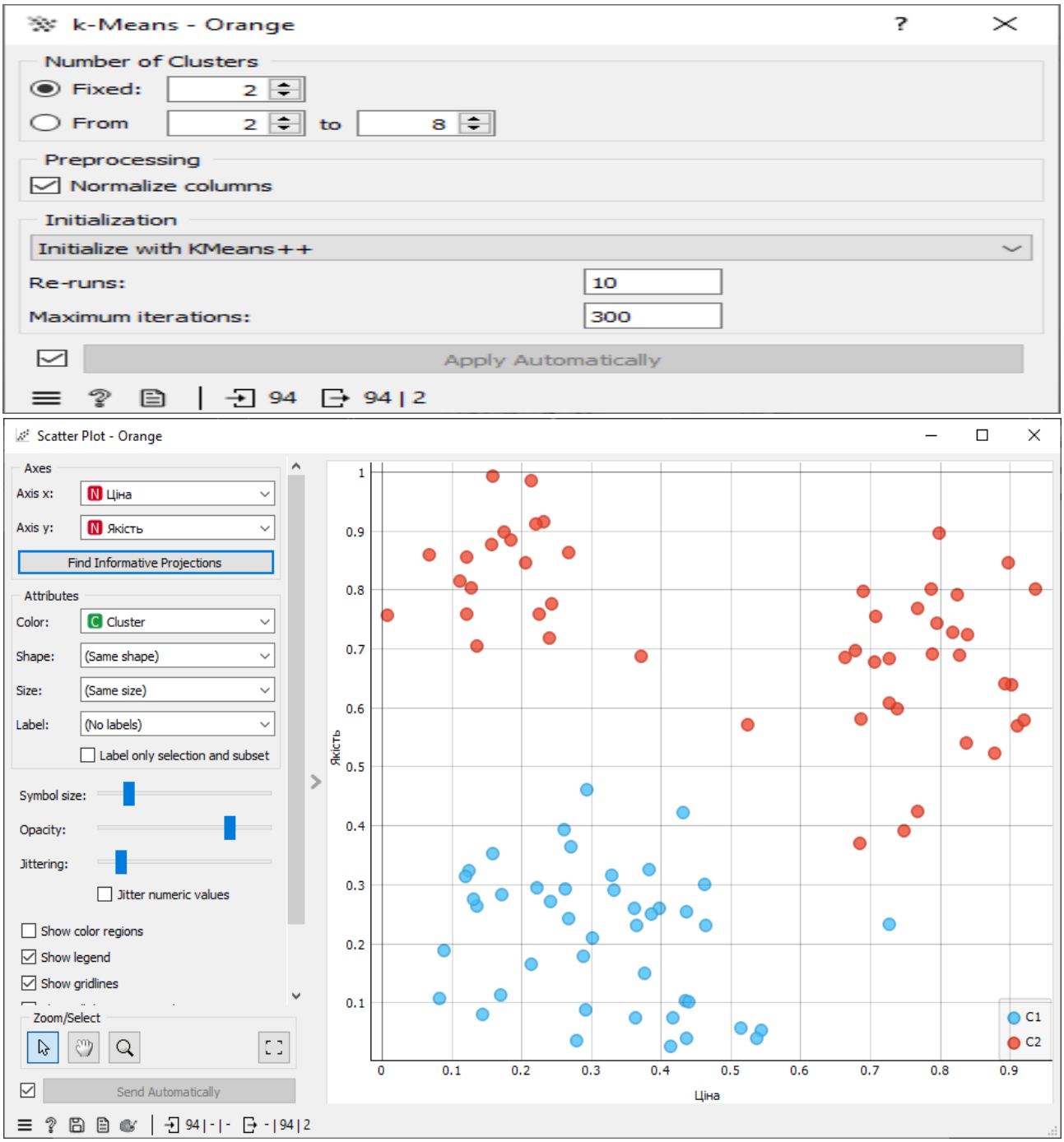


Рис. 8.4. Кластеризація методом k-means

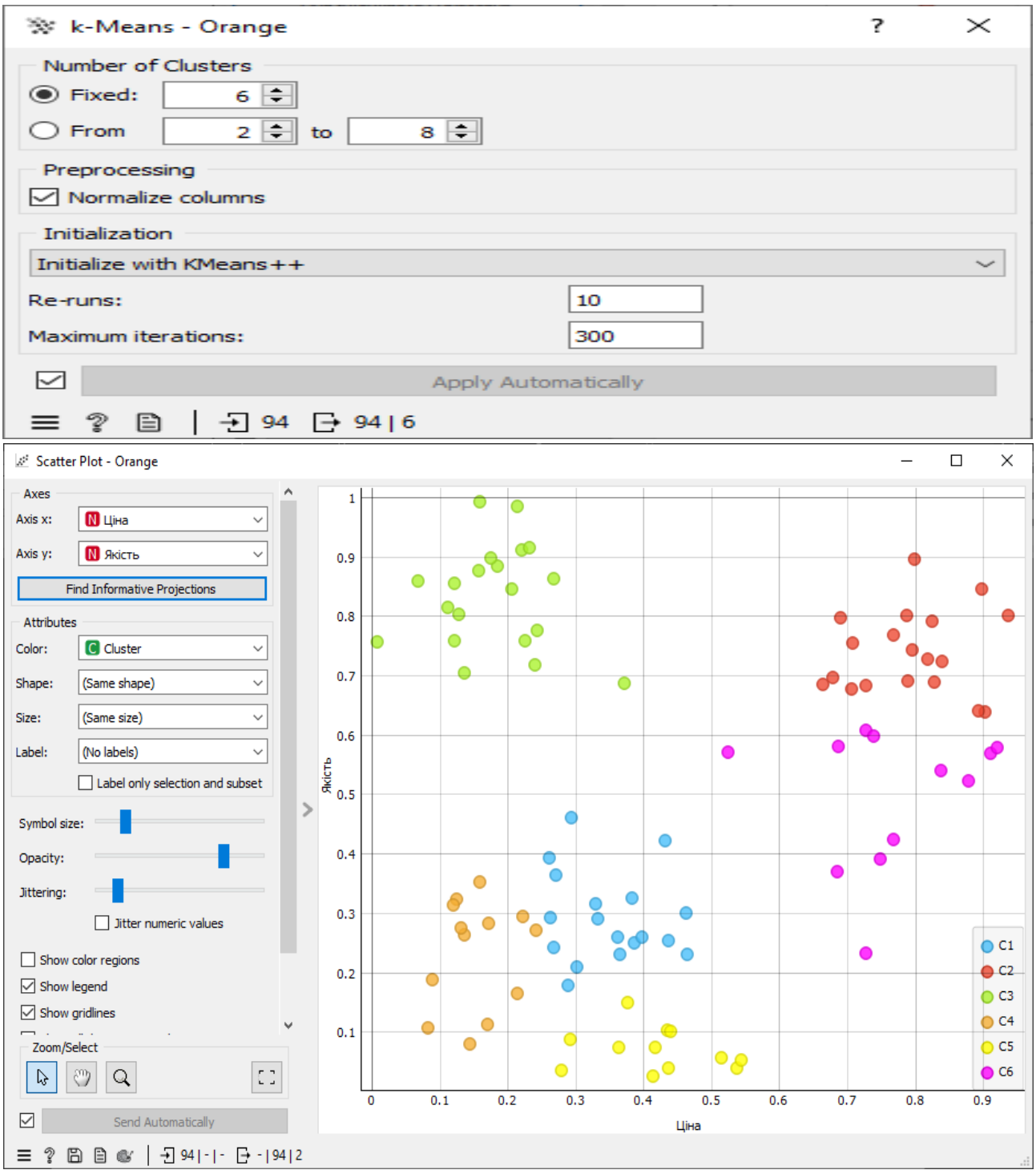


Рис. 8.5. Кластеризація методом k-means

Як бачимо, буде виділено стільки кластерів, скільки ми задаємо. А тепер спробуємо обрати налаштування для інтервалу кількості кластерів, щоб побачити, яку кількість модуль вважає найоптимальнішою (рис. 8.6).

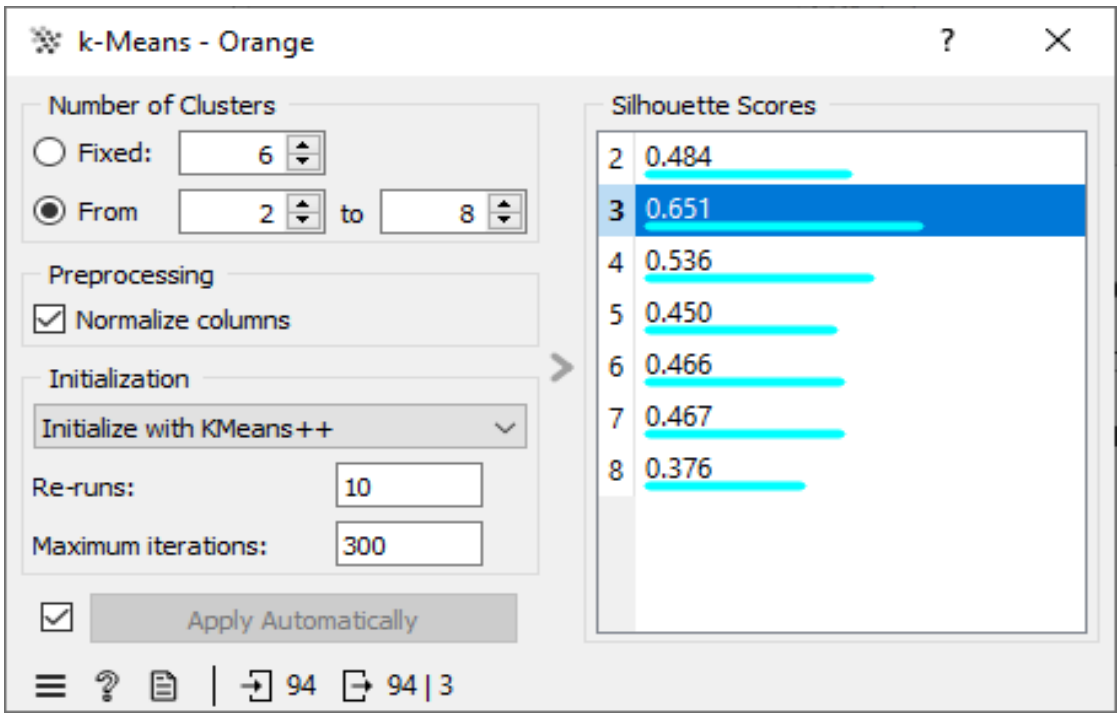


Рис. 8.6. Оцінки можливих кількостей кластерів

Тепер використаємо модуль Silhouette Plot, який допоможе побачити доречність конкретних точок у кластері, тобто чи логічним є її присвоєння до певного кластері чи вона помилкова (рис. 8.7).

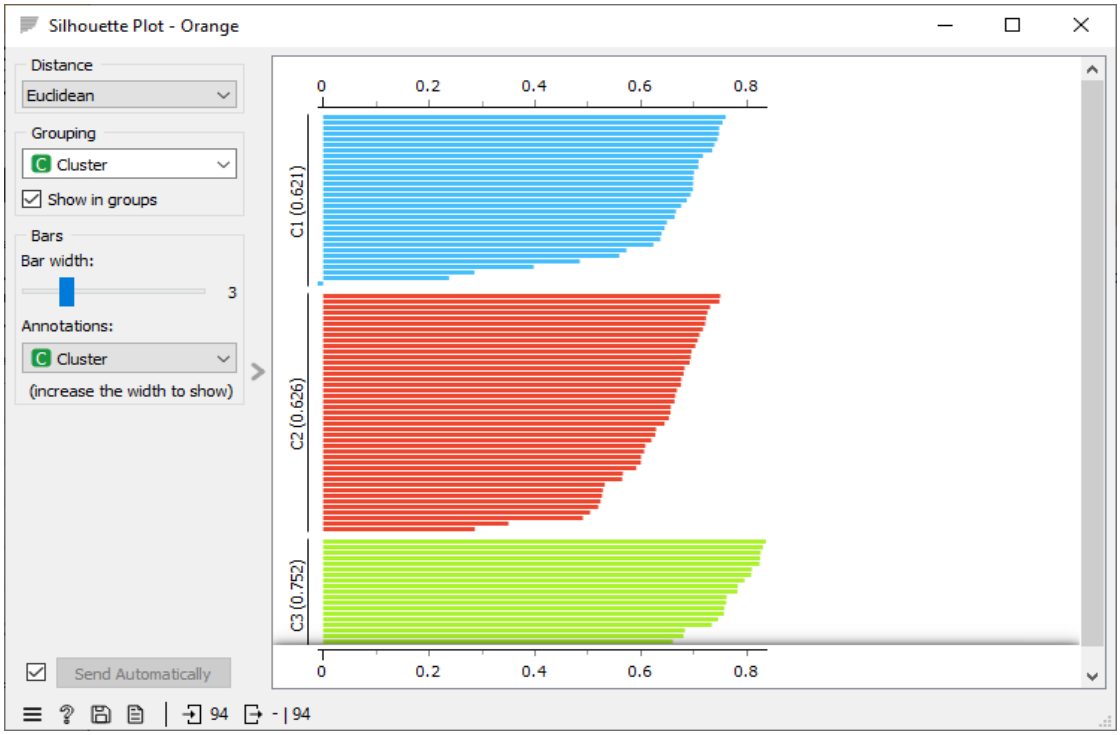


Рис. 8.7. Використання модуля Silhouette Plot

Ми можемо обирати конкретну точку на даному силуеті, щоб побачити, наприклад, які точки отримали найнижчі оцінки – тобто вони відрізняються від основної маси кластеру, до якого належать. Наприклад в даному випадку це такі точки як зображені на рис. 8.8 та 8.9, де одна з них має взагалі негативну оцінку, а інша – найнижчу у своєму кластері.

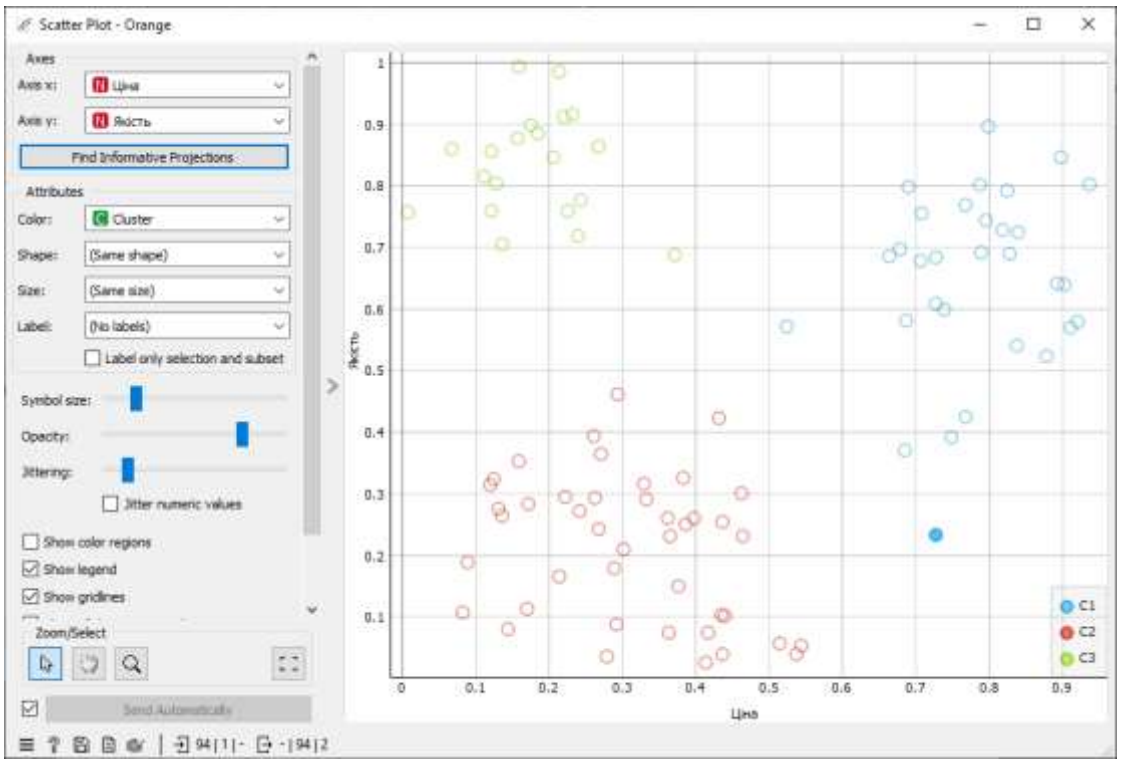


Рис. 8.8. Найбільш не підходящі точки кластеру

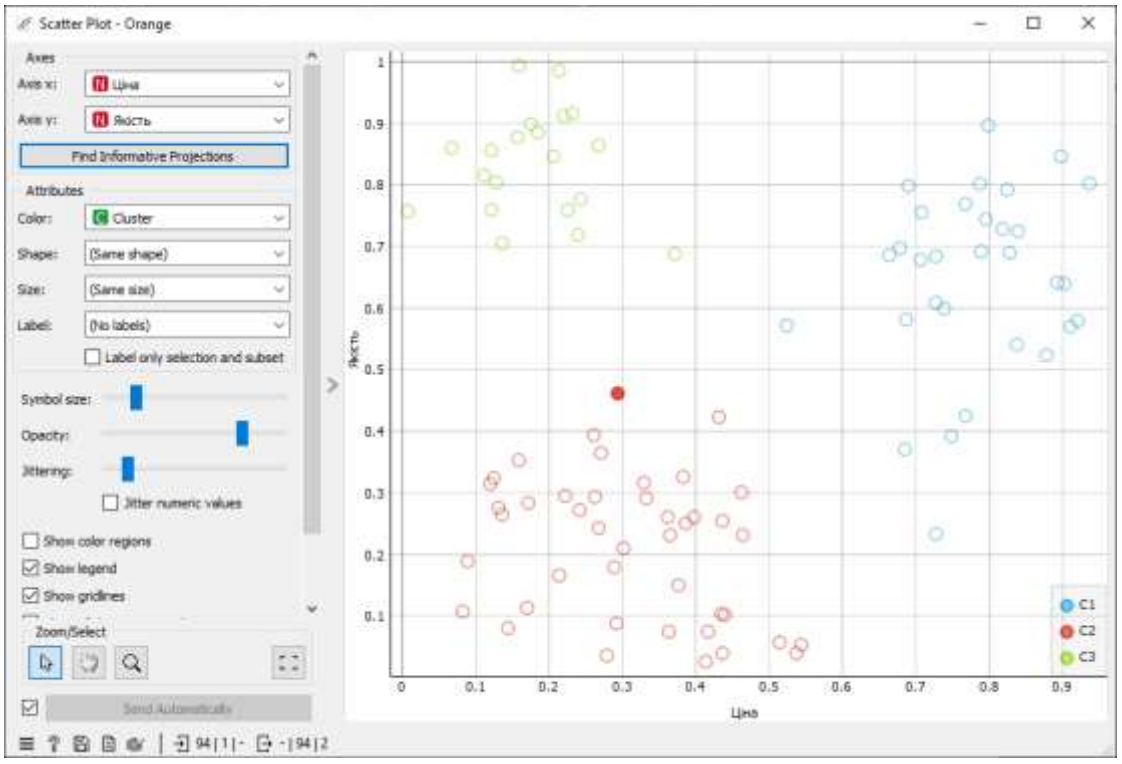


Рис. 8.9. Найбільш не підходящі точки кластеру

Аналізуючи ці дані, можна сказати, що метод k-середніх не стійкий до викидів (аномальних значень). Викиди можуть суттєво вплинути на позицію центрів кластерів, що призводить до спотворення результатів кластеризації.

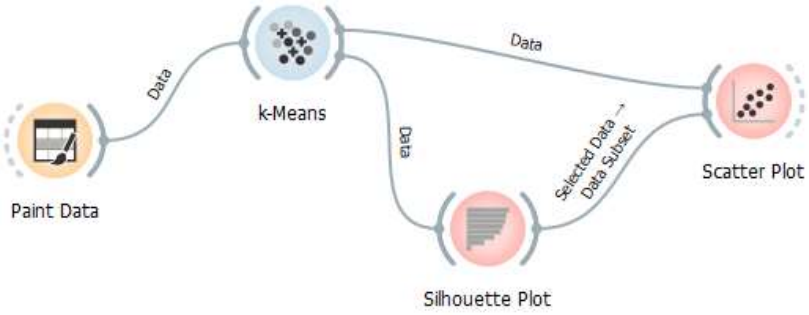


Рис. 8.10 Фінальна модель

Висновок

У рамках даної лабораторної роботи виявлено, що метод k-середніх є ефективним для розбиття даних на кластери, але потребує уважного підходу до вибору кількості кластерів і початкових центрів. У ході роботи з тестовими даними було продемонстровано, як зміна кількості кластерів впливає на результати кластеризації, підкреслюючи важливість правильного вибору цього параметра. Крім того, випадковий вибір початкових центрів часто призводить до різних результатів, що вимагає багаторазового запуску алгоритму для досягнення стабільного розподілу.

Також, було виявлено, що метод k-середніх працює найкраще для кластерів сферичної форми та однакового розміру. Під час експериментів з використанням модулю Paint Data стало очевидним, що алгоритм не справляється зі складними формами кластерів, часто розподіляючи точки неправильно, якщо форма або розмір кластерів суттєво відрізняються. Для більш точної кластеризації в таких випадках можуть знадобитися альтернативні алгоритми або попередня трансформація даних.

Окрім того, алгоритм виявився дуже чутливим до викидів, які можуть змістити центри кластерів і спотворити результати. Використання модуля Silhouette Plot дозволило виявити точки, що погано вписуються у свої кластери, показуючи таким чином вплив викидів на кластеризацію. Це підкреслює важливість попередньої обробки даних для видалення або корекції аномальних значень перед застосуванням методу k-середніх, що покращить точність і надійність кластеризації.

Перелік питань на захист

1. Етапи методу К – середніх
2. Вкажіть переваги та недоліки методу К-середніх
3. Вкажіть сучасні методи кластерного аналізу даних

ЛАБОРАТОРНА РОБОТА № 9

Тема: Визначення факторів якості кластеризації за допомогою Logistic Regression у програмному забезпеченні Orange

Постановка задачі: Провести аналіз факторів, які впливають на показники математичної моделі процесу кластеризації даних.

Хід роботи

Задано таблицю з певними характеристиками від Фрамінгемського центру дослідження, про дослідження впливу різних факторів на серце для подальшого її застосунку у програмі Orange. Після введення у програму ми розміщуємо блок «File», де ми додаємо нашу таблицю. У налаштуваннях цієї таблиці ми позначаємо змінну «TenYearCHD» як target (рис. 9.1).

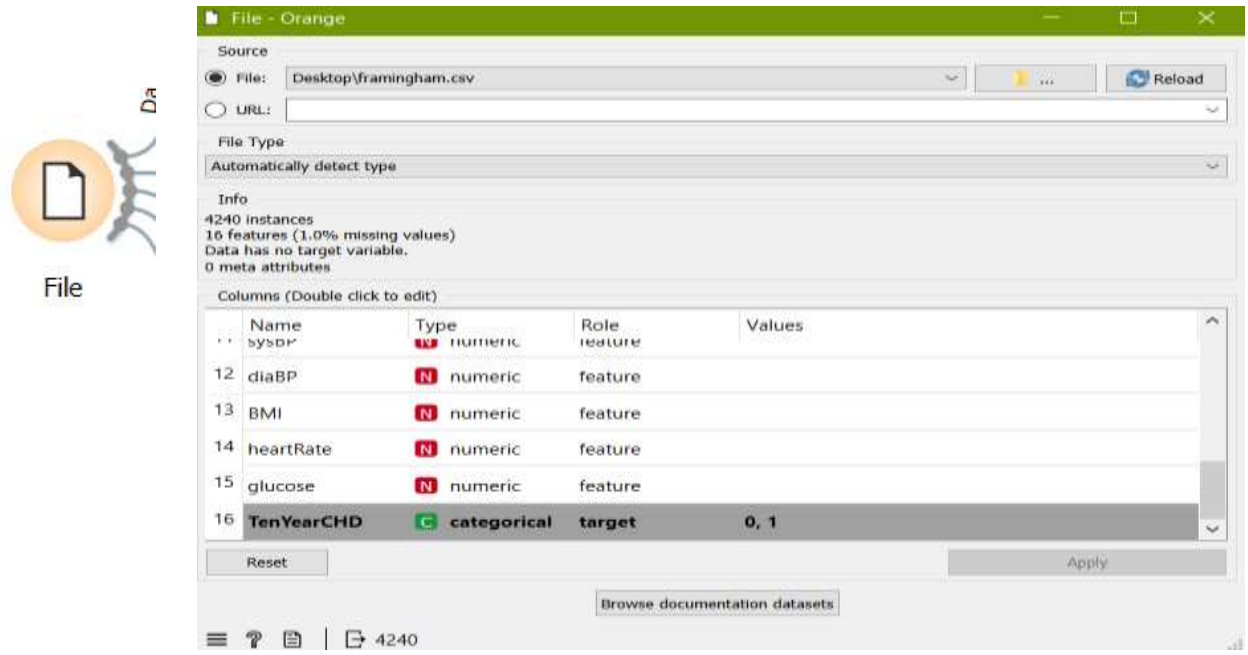


Рис. 9.1. Підключення таблиці даних та визначення головного (target) параметру

Далі додаємо у програму чотири блоки на панелі: «Data Table», що відображатиме таблицю, а також «Scatter Plot», «Distributions» та «Feature Statistics» для створення графіків (рис. 9.2).

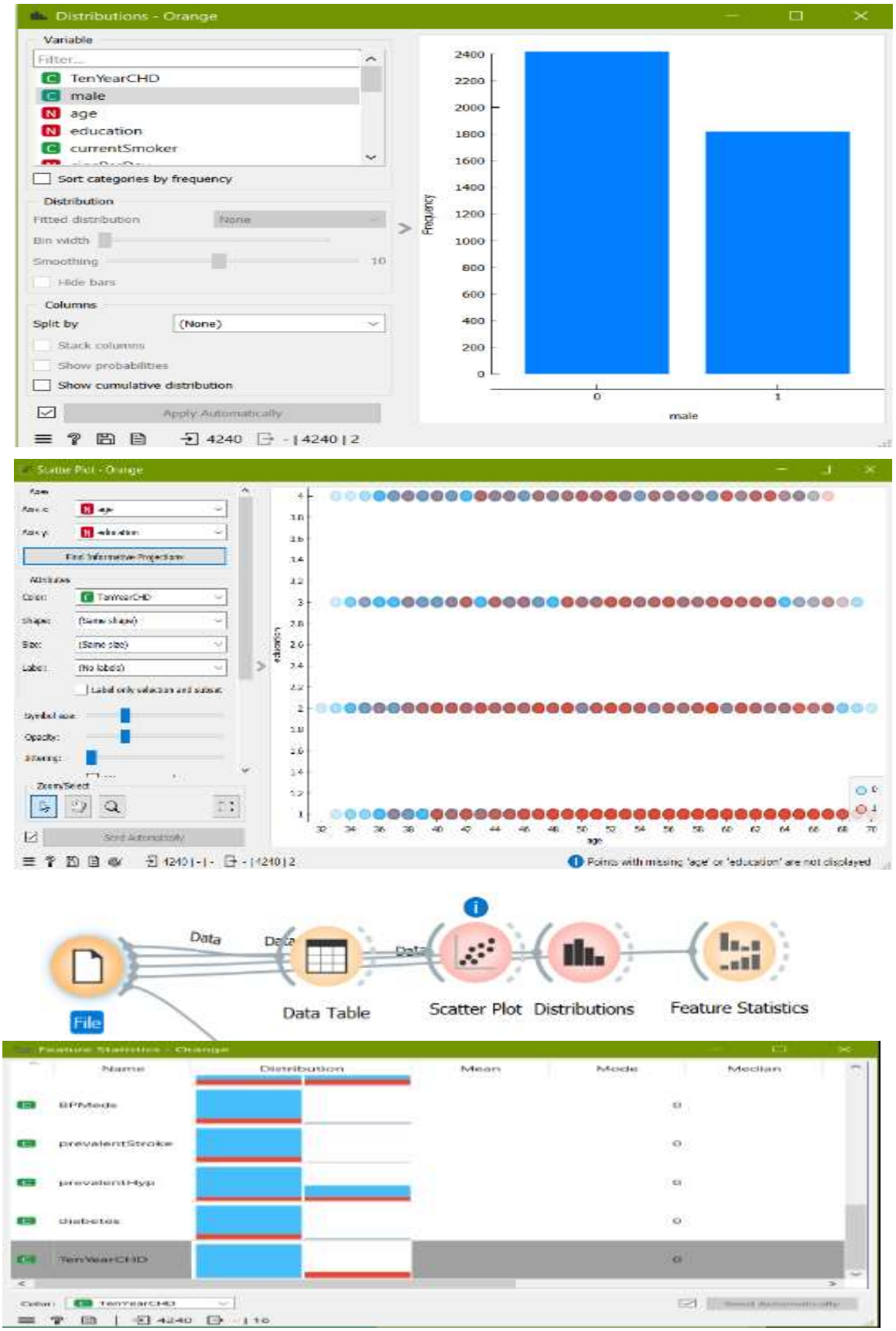


Рис. 9.2. Вигляд даних у різних блоках

Далі для розділення даних додамо блок «Data Sampler» та налаштуємо необхідні параметри:

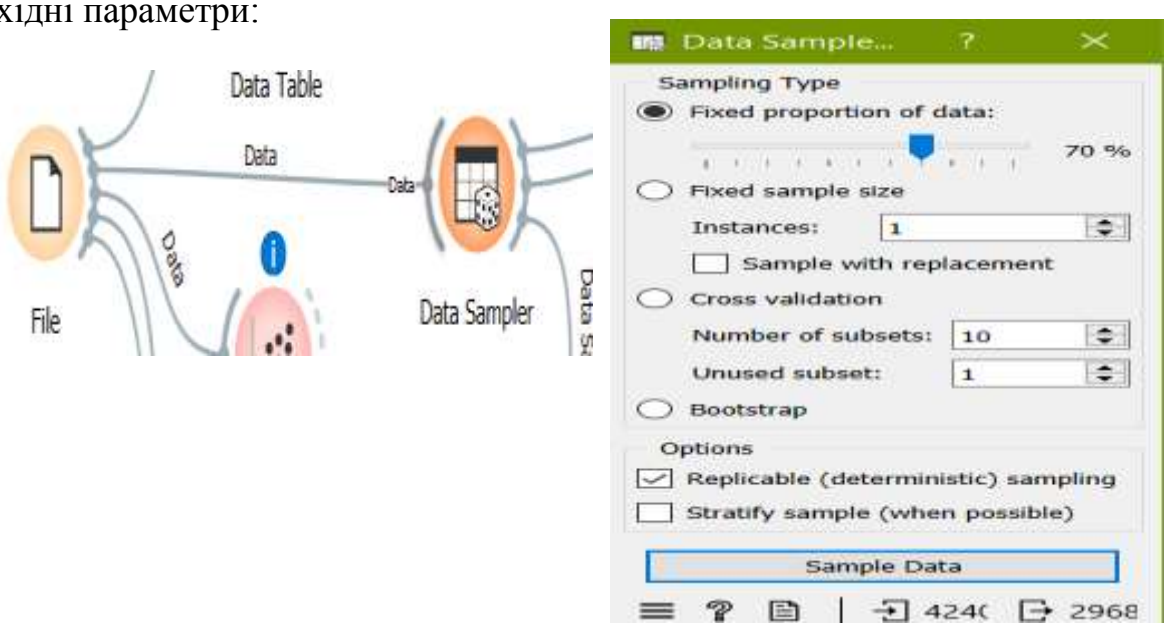


Рис. 9.3. Підключення модуля «Data Sampler»

Далі нам необхідно передати дані «Data Sempler» та «Remaining Data» блоку «Test and Score» (рис. 9.4).

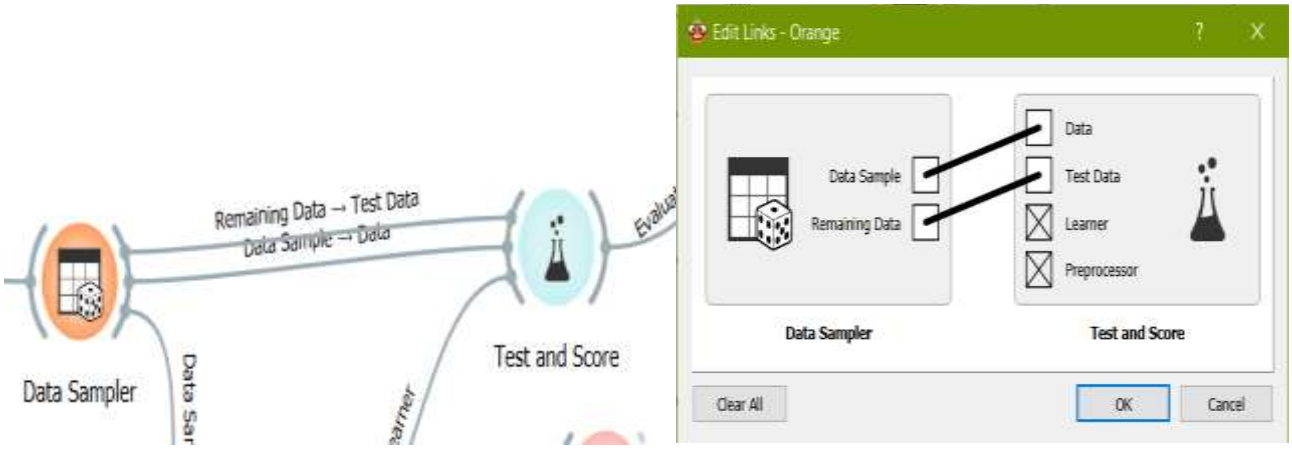


Рис. 9.4. Підключення модуля «Test and Score»

Для задач регресії додаємо модуль «Logistic Regression» – це модуль, який вирішує для нас задачу класифікації. До нього з «Data Sempler» передаємо дані та з нього передаємо дані у «Test and Score» (рис. 9.5).

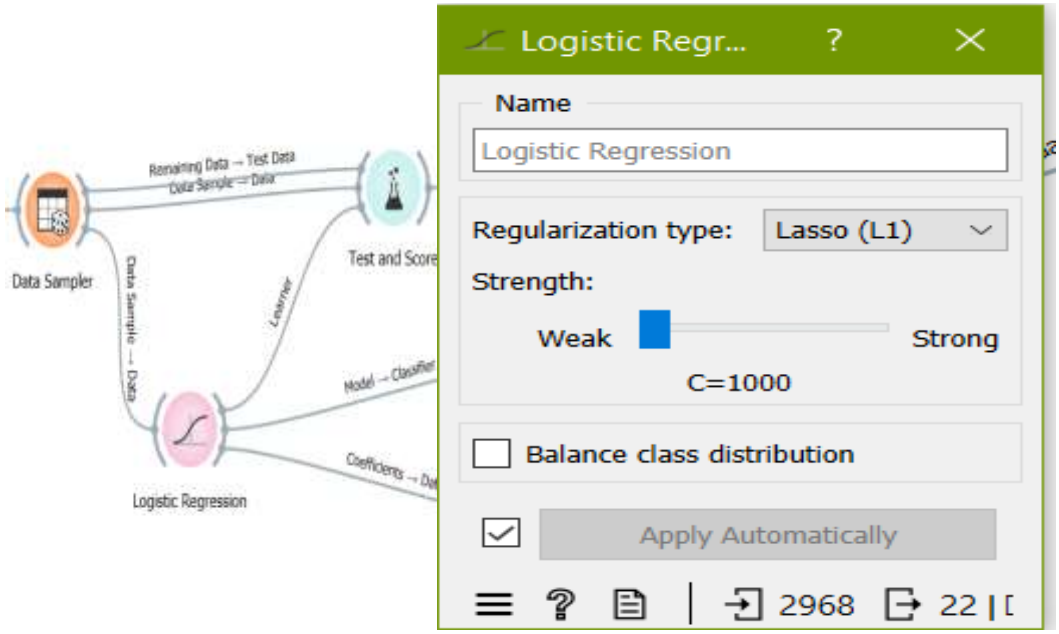


Рис. 9.5. Підключення модуля «Logistic Regression»

У результаті ми отримали параметри для оцінки моделі. Найбільш популярним параметром для оцінки у задачах класифікації є параметр AUC, його близькість до одиниці є хорошою ознакою. Нашому випадку AUC=0.726, це є хорошою ознакою, яка показує, що різниця під час проведення розслідування є (рис. 9.6).

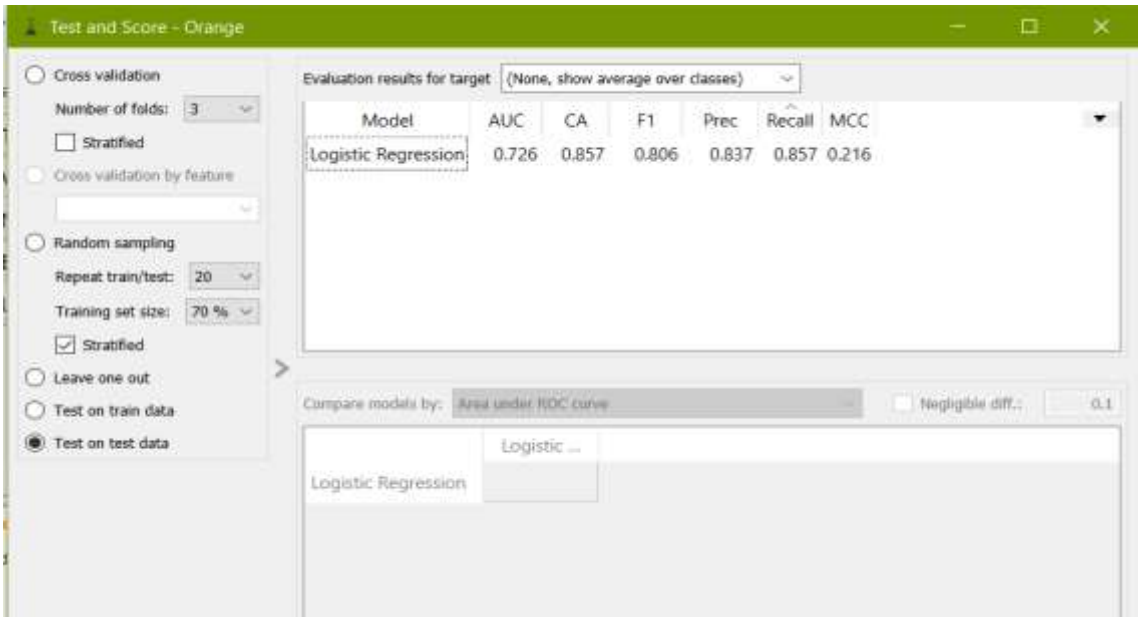


Рис. 9.6. Оцінка якості моделі

У результаті ми отримали модель, яка в змозі прогнозувати хвороби серця на 10 років вперед. Для визначення факторів ризику хвороб серця та визначення, які незалежні параметри впливають на залежні, нам необхідно додати блок «Nomogram». Завдяки цьому модулю ми можемо визначити які фактори ризику впливають на хвороби серця (рис. 9.7).

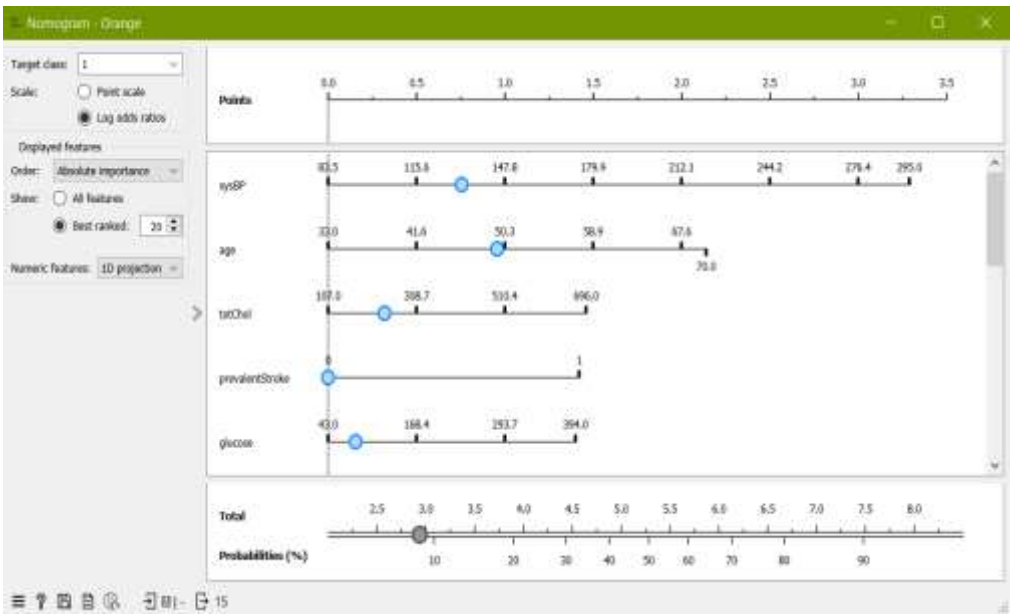


Рис. 9.7. Фактори ризику, та модель їх змінення у «Nomogram»

У результаті ми бачимо які фактори впливають на хвороби серця. Найбільш значимим фактором є Високий тиск та холестерин. За допомогою таблиці ми можемо роздивитись коефіцієнти у Logistic Regression. У результаті ми можемо побачити коефіцієнти моделі (9.8–9.9):

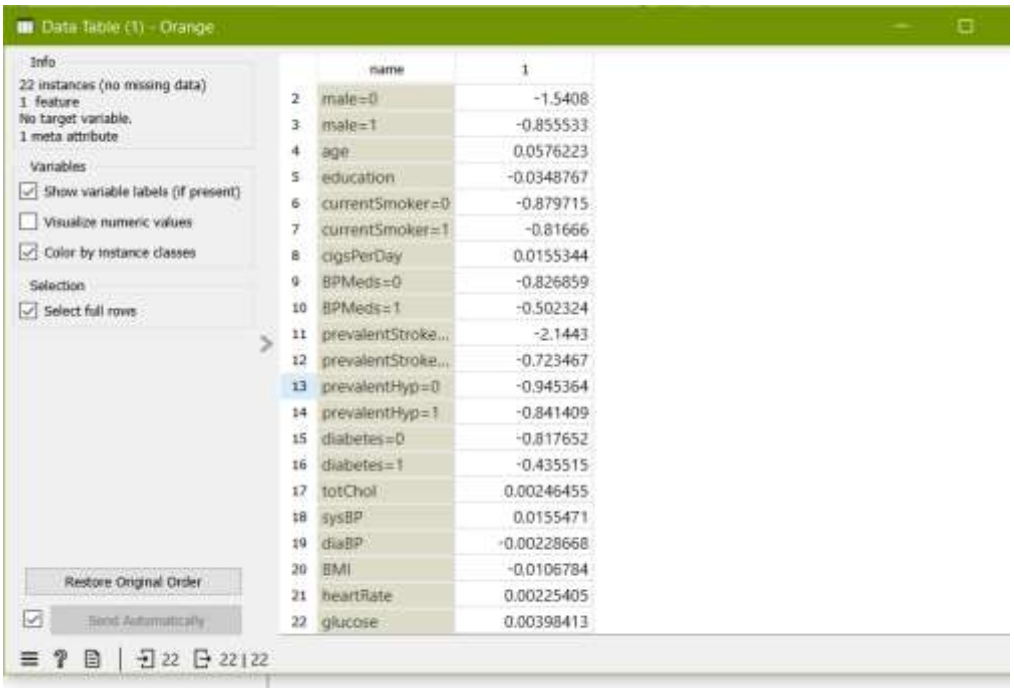


Рис. 9.8. Коефіцієнти моделі

До модуля «Test and Score» додаємо модуль «Confusion Matrix», який дозволяє деталізувати похибки під час проведення розрахунків.

Рис. 9.9. Результат отриманих даних у «Confusion Matrix»

У результаті в цьому блоці ми можемо побачити, що наша модель допустила 7 похибок першого роду, та 175 разів допустила похибку другого порядку.

Висновок

У ході виконання лабораторної роботи було проведено аналіз даних за допомогою програмного забезпечення Orange. Використали модель логістичної регресії для визначення факторів ризику хвороби серця на 10 років, використовуючи дані з Фрамінгемського центру дослідження.

Під час аналізу були використані різні графічні представлення даних, такі як діаграми розсіювання, розподілів та статистика параметрів. Крім того, за допомогою модуля «Confusion Matrix» були деталізовані похибки моделі, що дозволило нам оцінити ефективність класифікації та кластеризації. У результаті було виявлено 7 похибок першого роду та 175 похибок другого роду.

Загальний висновок полягає в тому, що використання логістичної регресії та інших методів аналізу даних у програмному забезпеченні Orange дозволяє проводити ефективний та детальний аналіз ризиків хвороби серця на основі надійних даних, отриманих в рамках дослідження.

Перелік питань на захист

1. Як окреслити межу кластерів? Скільки їх потрібно виділити?
2. Показники якості математичної моделі у кластерному аналізу
3. Стандартизація даних

ЛАБОРАТОРНА РОБОТА № 10

Тема: Аналіз методів ієрархічної кластеризації

Постановка завдання: Ознайомитися з програмою Orange та реалізувати за допомогою програмних можливостей ієрархічну кластеризацію певних даних (казок Андерсона) .

Теоретичні відомості

Orange – це набір інструментів для візуалізації даних, машинного навчання та інтелектуального аналізу даних із відкритим вихідним кодом. Він має інтерфейс візуального програмування для швидкого та якісного аналізу даних та інтерактивної візуалізації даних [10].

Ієрархічна кластеризація (англ. hierarchical clustering) – це сукупність певних алгоритмів упорядкування даних, в результаті яких на виході отримуємо ієрархію кластерів. Найкращим методом для графічного відображення такої кластеризації є дендрограма [11].

Завдання 1. Ієрархічна кластеризація

При створенні нового проєкту в Orange перед користувачем буде поки ще не заповнене елементами робоче поле, на яке будемо перетягувати елементи з відповідною панелі зліва. Для початку виберемо елемент «Corpus» та додамо його до проєкту (рис. 10.1):

Рис. 10.1. Додавання елемента «Corpus»

Тепер подвійним натисканням по нашому елементу відкриємо його вікно та виставимо певні настройки згідно зі скріншотом, вказаним нижче (рис. 10.2):

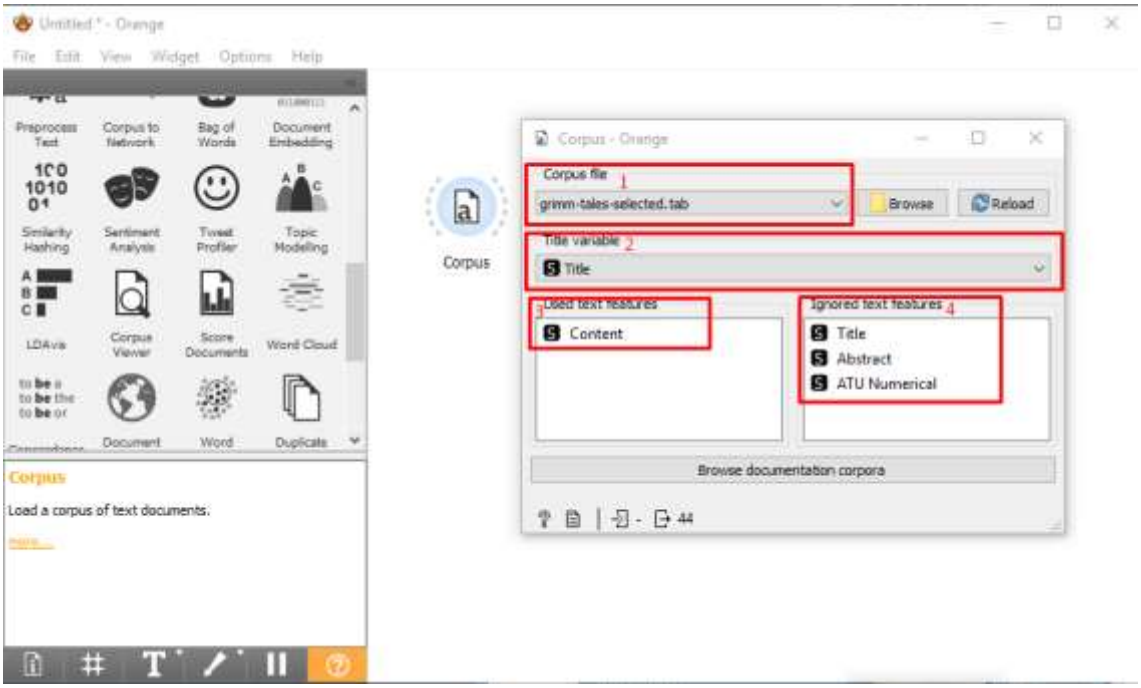


Рис.10.2. Вікно елемента «Corpus», в якому виставляємо такі налаштування

Тепер, після зберігання параметрів, вказаних вище, наш елемент містить в собі набір об’єктів текстових даних, а якщо буде точніше: казки братів Грім.

Переглянути зміст «Corpus» можна за допомогою елемента «Corpus viewer» наступним чином (рис. 10.3):

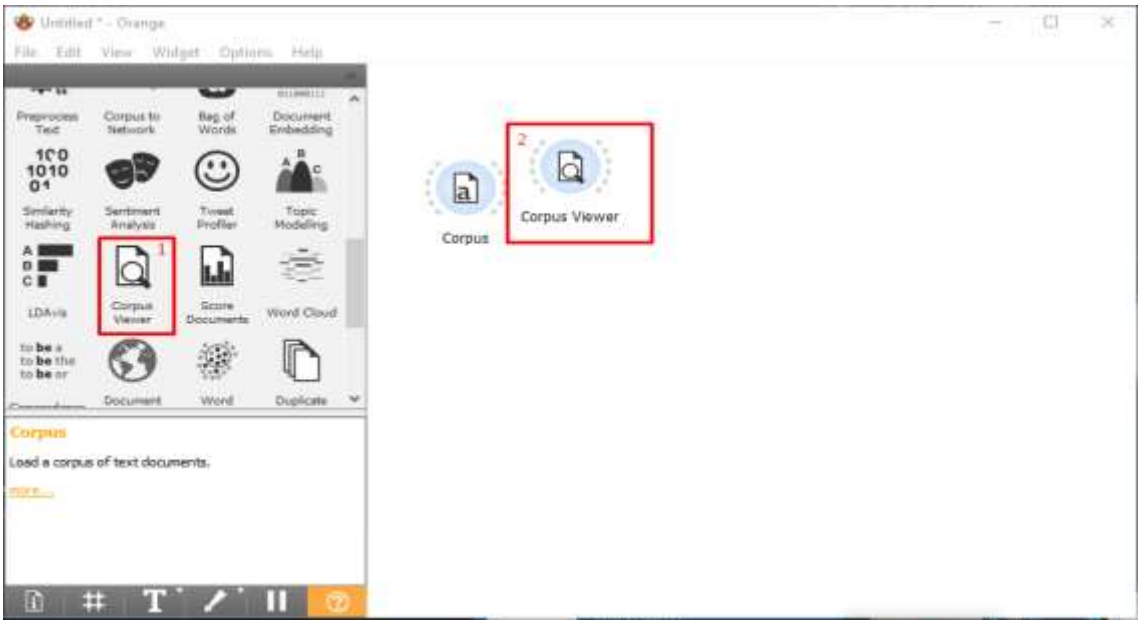


Рис.10.3. Додавання «Corpus viewer»

Тепер зв’яжемо наші елементи на схемі та переглянемо зміст елементу «Corpus viewer» (рис. 10.4):

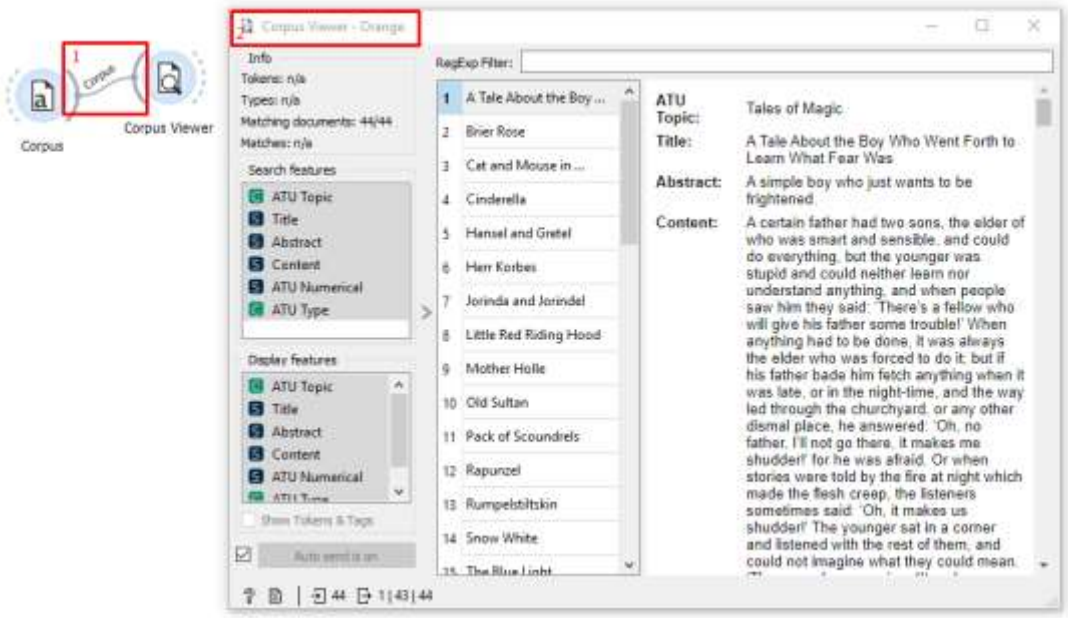


Рис. 10.4. Зв’язуємо елементи та переглядаємо зміст казок за допомогою «Corpus viewer»

Щоб продемонструвати вміння Orange працювати та аналізувати дані, чудово підійдуть елементи «Preprocess text», за допомогою яких ми обробимо та виділимо з текстових даних тільки текст та «Word cloud», підключений наступним для перегляду найчастіших слів, використаних у казках (рис. 10.5):

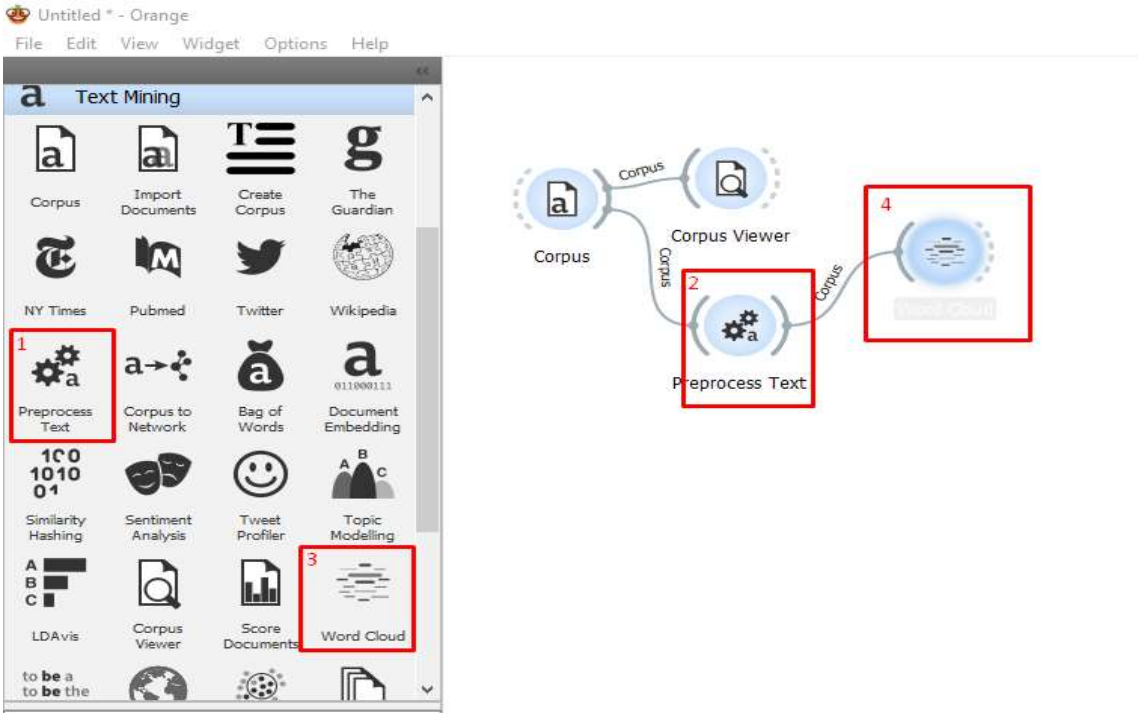


Рис.10.5. Додаємо нові елементи до схеми

Тепер ознайомимось детальніше з вікнами наших нових елементів, налаштуємо «Preprocess text» та переглянемо результат у вікні «Word cloud»:

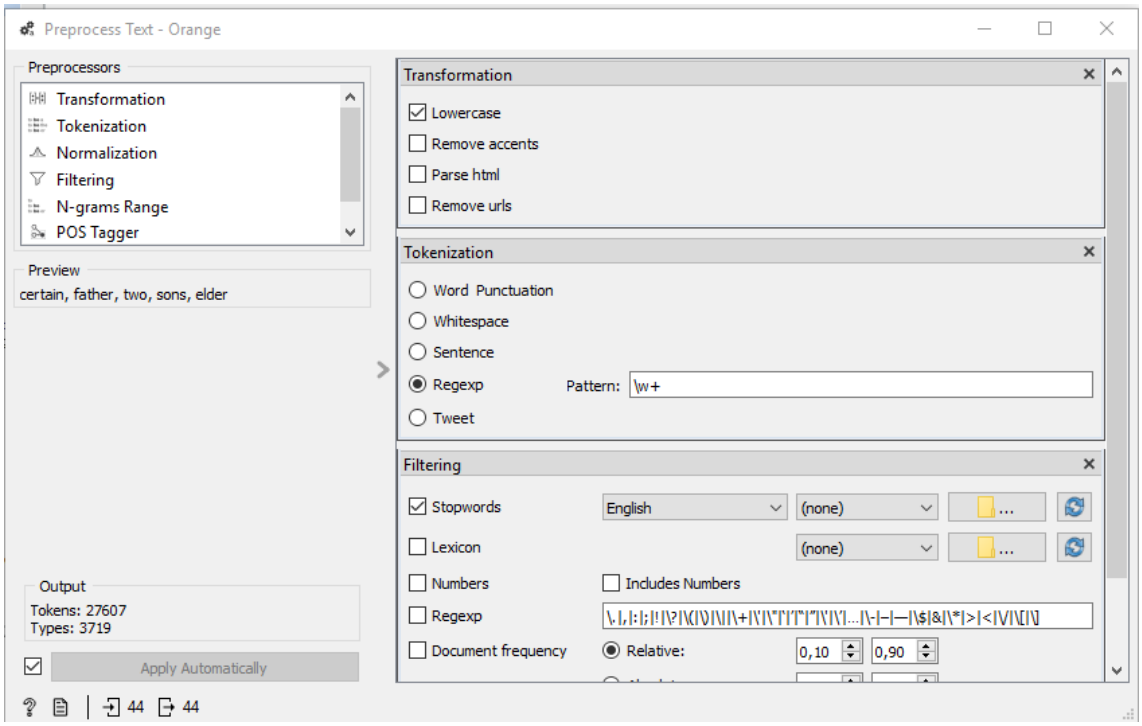


Рис. 10.6. Вікно елемента «Preprocess text»

Після того, як наші дані проходять через даний елемент та на виході маємо тільки слова, переглядаємо результат «Word cloud» (рис. 10.7):

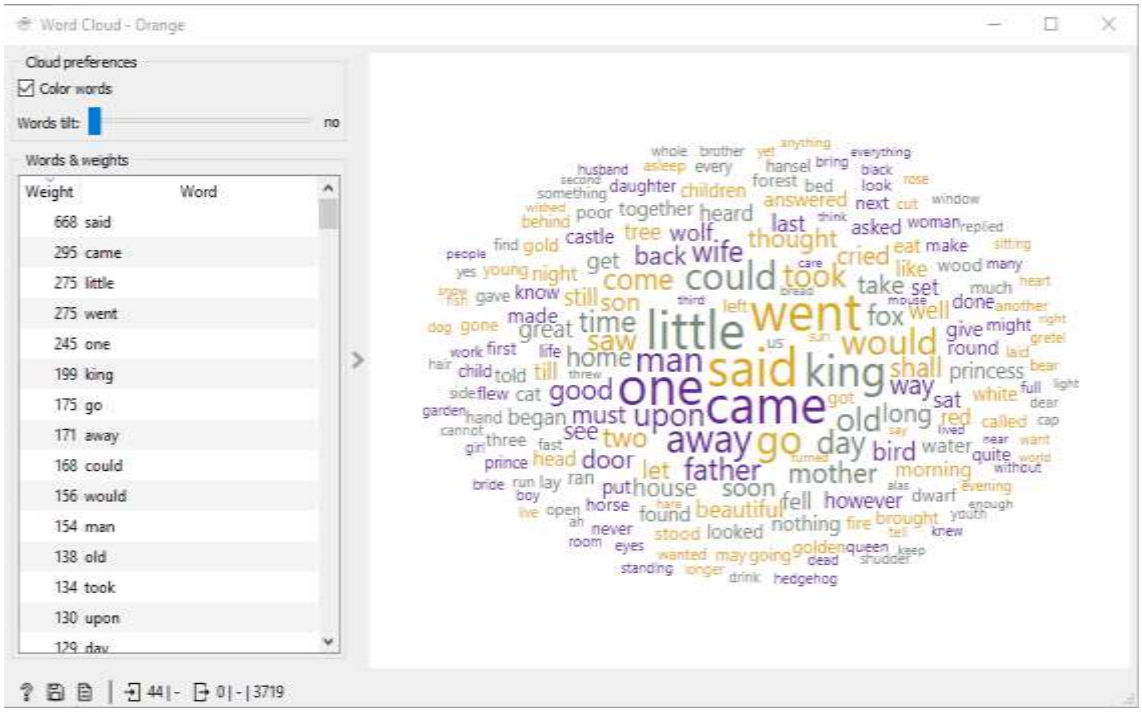


Рис. 10.7. Вікно елемента «Word cloud»

Тепер, коли ми ознайомились з деякими можливостями програмного середовища Orange та навчились будувати схеми, за допомогою яких можна аналізувати дані, переходимо до обробки нашого прикладу методом ієрархічної кластеризації.

Для цього до схеми додаємо 2 нових елемента “Bag a Words”, “Distances”. Перший використовуємо для зміни типів даних з текстового до табличних, а другий відповідно, сходячи з назви, для знаходження відстані між об’єктами (рис. 10.8):

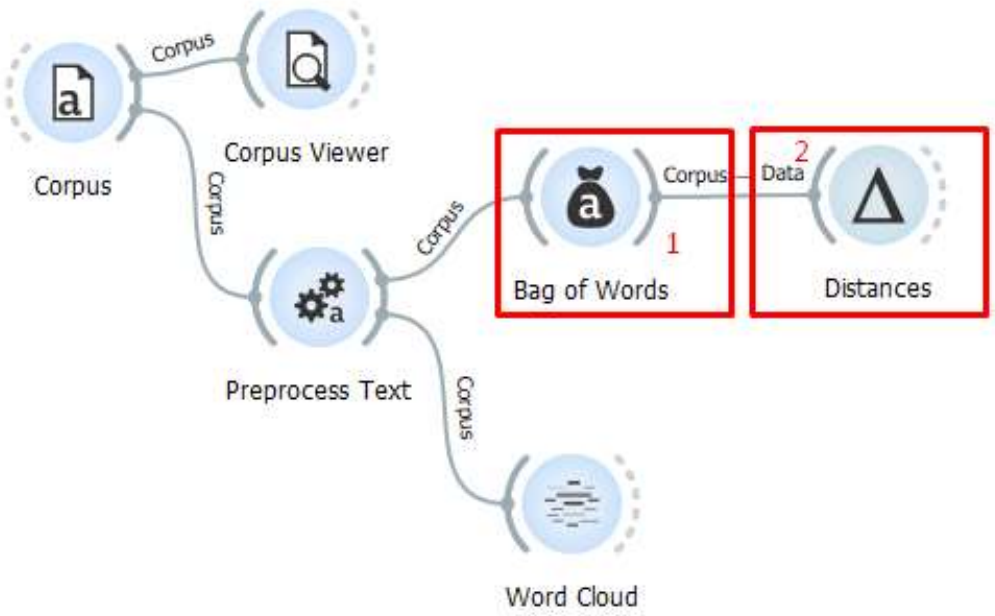


Рис. 10.8. Додаємо нові елементи

Тепер, якщо відкрити вікно елементу «Distances», можемо бачити, що користувач лишає за собою право вибирати різноманітні типи відстані, деякі з них ми вже використовували раніше (Евклідова відстань) (рис. 10.9):

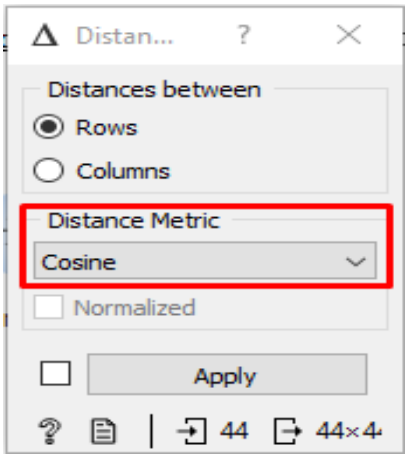


Рис. 10.9. Вибираємо відстань Cosine у вікні «Distances»

Після вибору відстані все що нам залишилося – додати до схеми елемент «Hierarchical Clustering» та переглянути результат кластеризації (рис. 10.10–10.11):

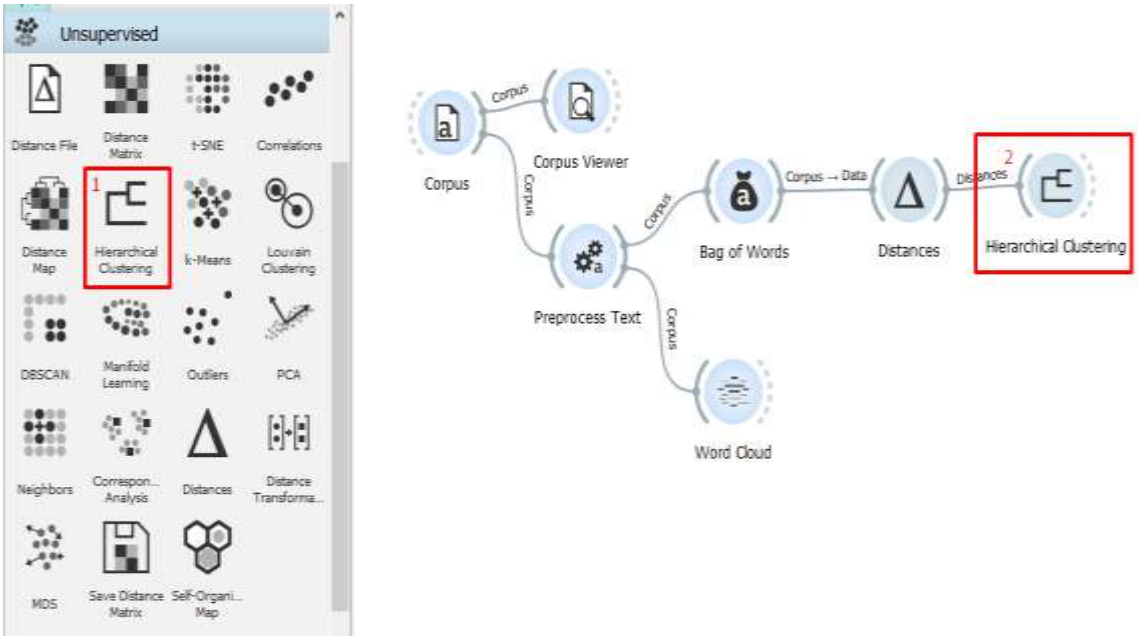


Рис.10.10. Додаємо елемент для ієрархічної кластеризації

Переглянемо, як програма Orange впоралась з таким завданням:

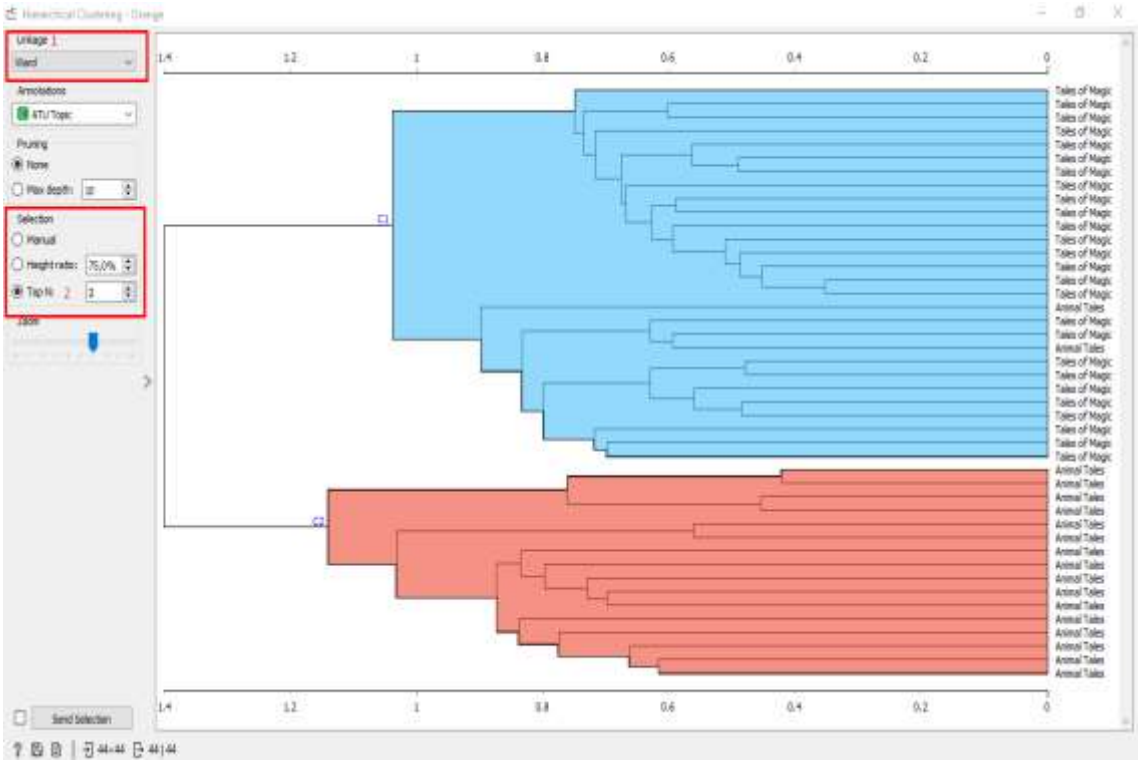


Рисунок 10.11. Ієрархічна кластеризація казок братів Грім

Як бачимо, програма поділила всі об’єкти на 2 кластери:

- Магічні казки (синій колір);
- Казки про тварин(червоний колір).

Для досягнення такого результату нам довелось лише побудувати просту схемку наступного вигляду (рис. 10.12):

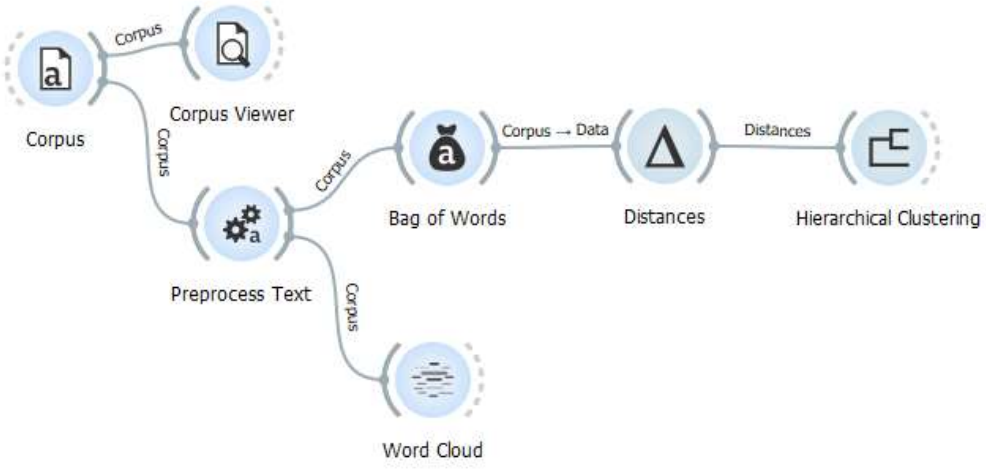


Рис. 10.12. Схема для ієрархічної кластеризації

Завдання 2. Кластеризація казок Андерсона

Тепер, коли ми ознайомились з методом ієрархічної кластеризації на прикладі, що наведений вище, можемо за допомогою програмних можливостей середовища Orange кластеризувати нові дані за тематикою з попередніх даних, а також на основі нових даних побудувати монограму.

Для цього побудуємо схему з наступною структурою, додаючи до неї нові елементи, що були відсутні в попередньому проєкті (рис. 10.13).

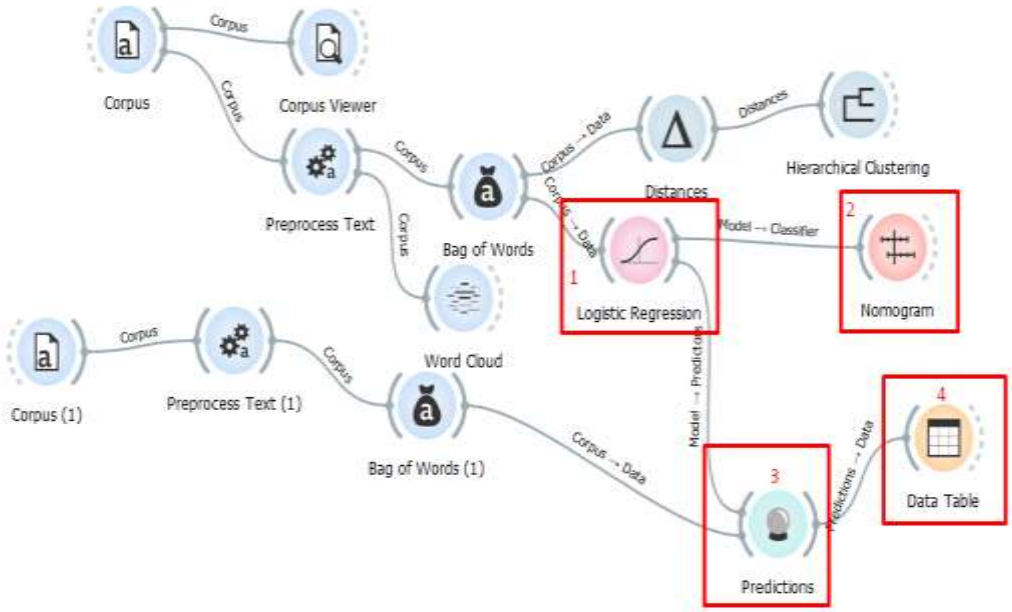


Рис. 10.13. Будуємо схему з новими елементами для класифікації казок Андерсона

Тепер можемо ознайомитися з монограмою, відміченою на рис. 10.13 цифрою 2:

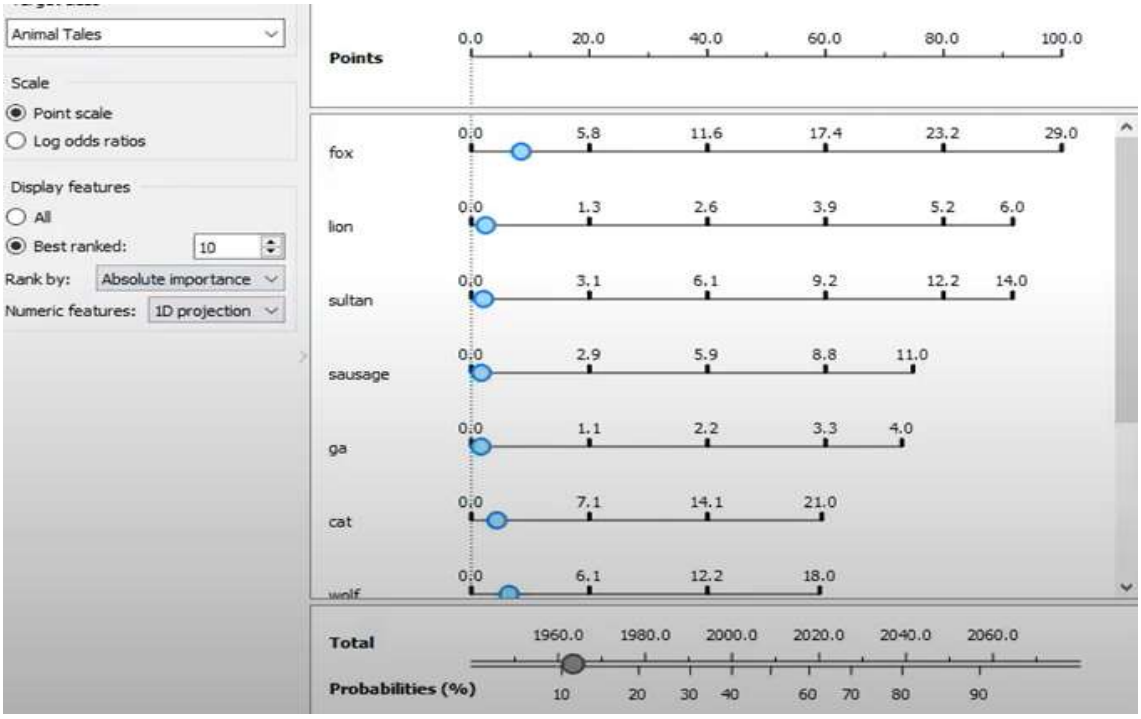


Рис.10.14. Номограма на основі наших даних

Елемент «Data Table», відмічений на рисунку 13 цифрою 4, містить в собі нові дані про класифікацію казок Андерсона на основі тематик казок братів Грім (рис. 10.15):

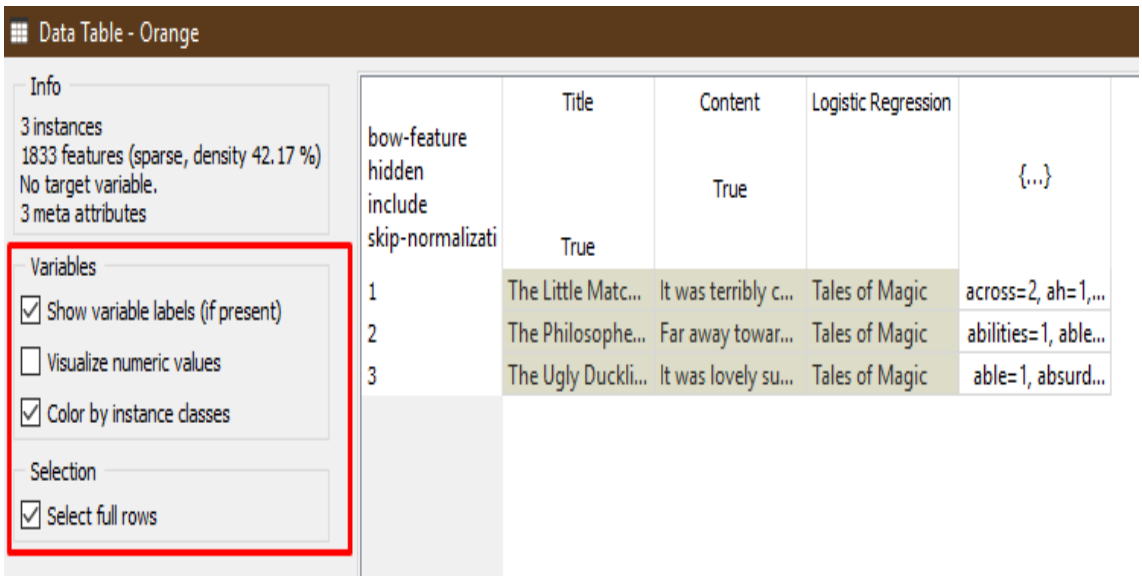


Рис. 10.15. Таблиця з даними класифікації казок Андерсона

Висновок

Виконавши це завдання, ми ознайомились з новими елементами “Logistic Regression”(для вивчення моделі логічної регресії), “Nomogram” для графічного відтворення класифікаторів, “Predictions” для виводу даних “Data Table”, за допомогою якого переглянули результат кластеризації казок.

Перелік питань на захист

1. Наведіть визначення терміна дендрограма
2. Етапи методу кластеризації Густафсона-Кесселя
3. Метод нечітких К-середніх

ЛАБОРАТОРНА РОБОТА № 11

Тема: Розробка нейромережевої системи розпізнавання зображень.

Постановка завдання: Ознайомитися з програмою Orange та реалізувати за допомогою програмних можливостей нейромережеву систему.

Теоретичні відомості

Нейромережева технологія розпізнавання зображень – це галузь штучного інтелекту (ШІ), яка використовує нейронні мережі для аналізу, інтерпретації та класифікації візуальної інформації. Ця технологія знаходить застосування в багатьох сферах, таких як медицина, транспорт, роздрібна торгівля, безпека, розваги та інше.

Етапи:

1. **Збір даних:** Нейромережу навчають на великому наборі зображень, які позначені відповідними категоріями.
2. **Архітектура нейромереж:** Для розпізнавання зображень зазвичай використовуються згорткові нейронні мережі (Convolutional Neural Networks, CNN). Вони складаються з шарів, які виділяють ключові ознаки зображення (краї, текстури, форми тощо).
3. **Навчання:** Під час навчання модель вивчає закономірності в даних, щоб згодом розпізнавати подібні ознаки на нових зображеннях.
4. **Інференція:** Після навчання модель може класифікувати або аналізувати нові зображення, наприклад, визначати об’єкти на фото, ідентифікувати осіб або розпізнавати текст.

Хід роботи:

Першим чином підключимо модуль імпортування зображень в Orange. Зображення взяті з сайту Kaggle, одного з відкритих датасетів. (рис. 11.1)

https://www.kaggle.com/datasets/hereisburak/pins-face-recognition

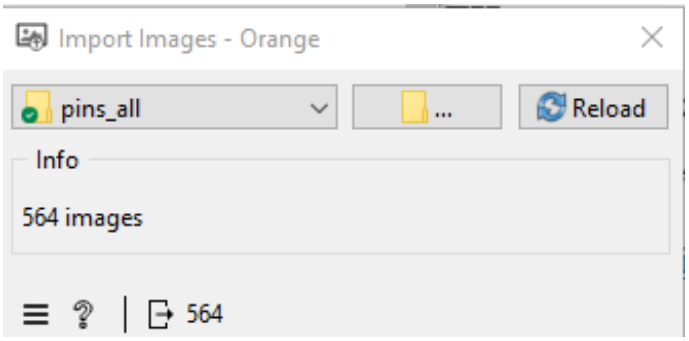


Рис. 11.1. Імпортування зображень у Orange

Далі підключимо Image Viewer та передивитися зображення (11.2).

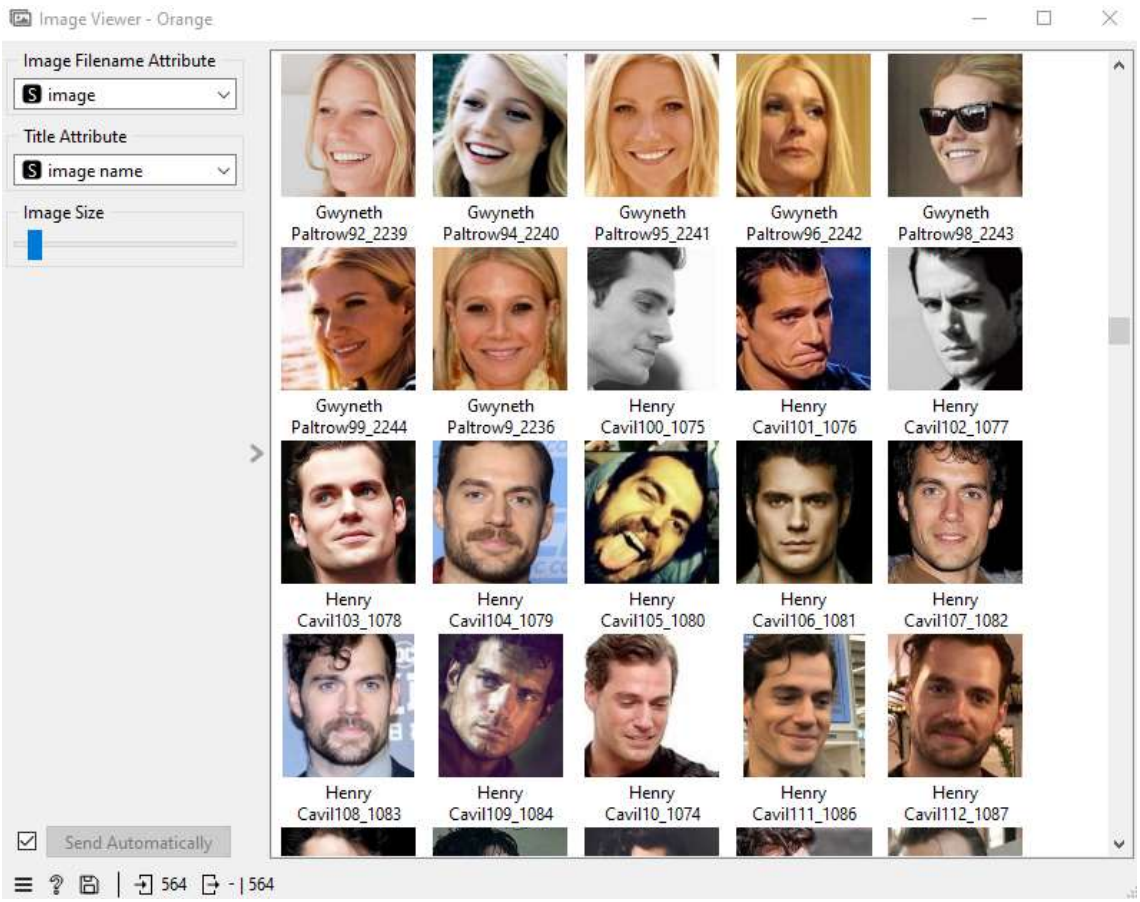


Рис. 11.2. Зображення

З датасету обирається зображення 3 людей, 2 чоловіки і 1 жінка. Підключимо модуль Image Embedding. (рис. 11.3)

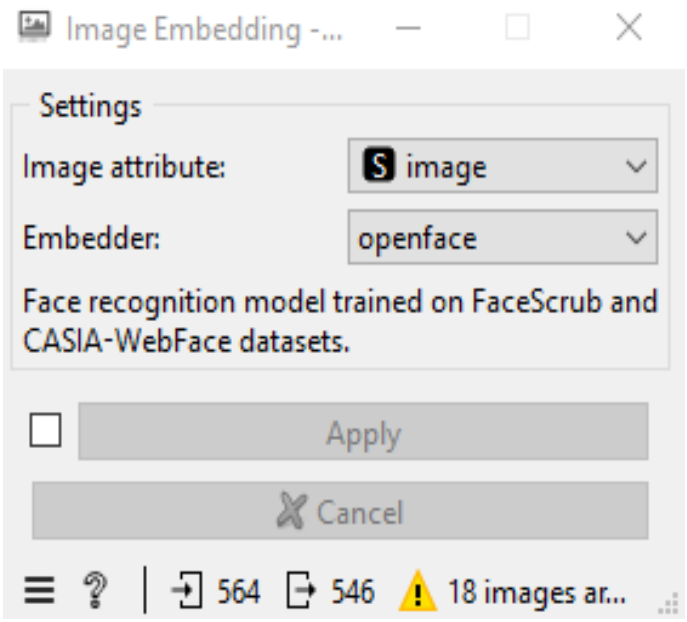
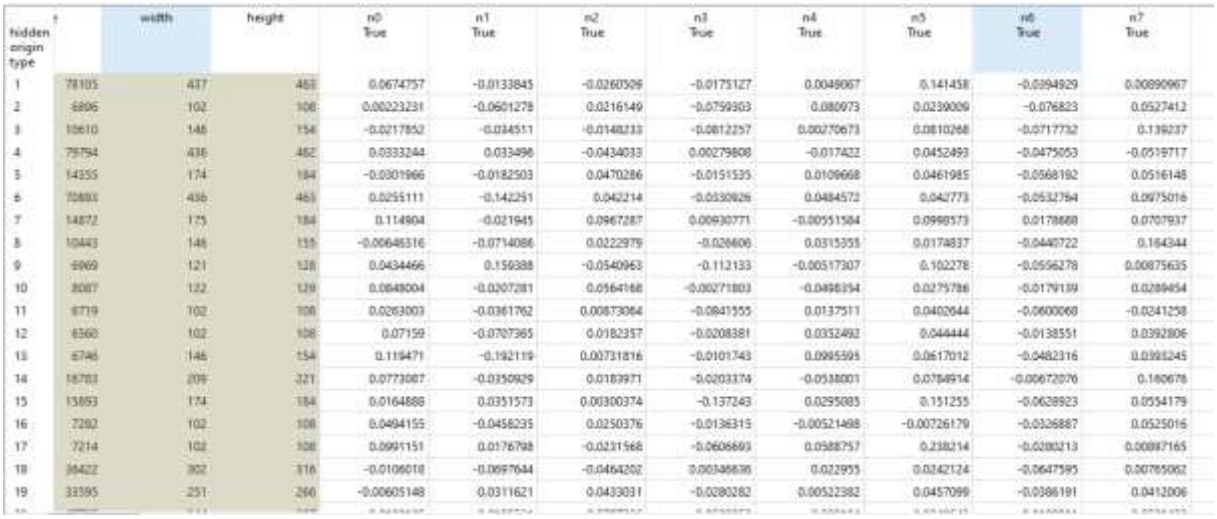


Рис. 11.3. Image Embedding

Цей модуль використовує велику модель глибинного навчання, в моєму випадку openface, для аналізу наданих зображень. Він перетворює зображення у набір параметрів, зрозумілих машині, які потім можна передивитись у таблиці. Обираємо модель openface тому, що вона, як виходить з назви, натренована саме на людських обличчях.



hidden origin type		width	height	n0 True	n1 True	n2 True	n3 True	n4 True	n5 True	n6 True	n7 True
1	78105	437	463	0.0674757	-0.0133845	-0.0260509	-0.0175127	0.0049067	0.141458	-0.0394929	0.00890967
2	6896	102	108	0.00223231	-0.0601278	0.0216149	-0.0759303	0.080973	0.0239009	-0.076823	0.0527412
3	10610	146	154	-0.0217852	-0.034511	-0.0148233	-0.0812257	0.00270673	0.0810268	-0.0717732	0.139237
4	79794	438	462	0.0333244	0.033496	-0.0434033	0.00279808	-0.017422	0.0452493	-0.0475053	-0.0519717
5	14355	174	184	-0.0301966	-0.0182503	0.0470286	-0.0151535	0.0109668	0.0461985	-0.0568192	0.0516148
6	70493	436	463	0.0255111	-0.142251	0.042214	-0.0330926	0.0484572	0.042773	-0.0532764	0.0975016
7	14872	175	184	0.114804	-0.021945	0.0967287	0.00930771	-0.00551584	0.0968573	0.0178688	0.0707937
8	10443	148	155	-0.00648316	-0.0714086	0.0222979	-0.026606	0.0315355	0.0174837	-0.0440722	0.164344
9	6969	121	128	0.0434466	0.159388	-0.0540963	-0.112133	-0.00517307	0.102278	-0.0596278	0.00875635
10	8087	122	129	0.0848004	-0.0207281	0.0564168	-0.00271803	-0.0498354	0.0275786	-0.0179139	0.0289454
11	6719	102	108	0.0263003	-0.0361762	0.00873064	-0.0841555	0.0137511	0.0402644	-0.0600068	-0.0241258
12	6560	102	108	0.07159	-0.0707365	0.0182357	-0.0208381	0.0352492	0.044444	-0.0138551	0.0392806
13	6746	146	154	0.119471	-0.192119	0.00731816	-0.0101743	0.0995595	0.0617012	-0.0482316	0.0393245
14	16703	209	221	0.0773087	-0.0350929	0.0183971	-0.0203374	-0.0538001	0.0784914	-0.00672076	0.160678
15	15893	174	184	0.0164888	0.0351573	0.00300374	-0.137243	0.0295085	0.151255	-0.0628923	0.0554179
16	7202	102	108	0.0494155	-0.0458235	0.0250376	-0.0136315	-0.00521498	-0.00726179	-0.0326887	0.0525016
17	7214	102	108	0.0991151	0.0176798	-0.0231568	-0.0606693	0.0588757	0.238214	-0.0280213	0.00887165
18	36422	302	316	-0.0106018	-0.0697644	-0.0464202	0.00346836	0.022955	0.0242124	-0.0647595	0.00765062
19	33595	251	266	-0.00605148	0.0311621	0.0433031	-0.0280282	0.00522382	0.0457099	-0.0386191	0.0412006

Рис. 11.4. Таблиця параметрів

Всього у кожного зображення 127 параметрів.

Наступним кроком підключимо модуль Distances для аналізу схожості зображень і модуль Hierarchial Clustering для їх поділу на кластери

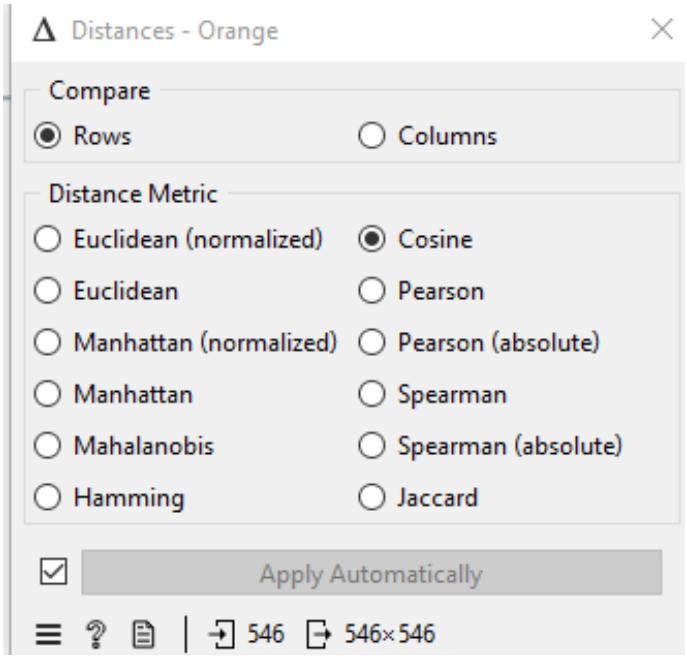


Рис. 11.5. Модуль Distances

Тут обираємо косинус як метрику дистанцій через те, що вона найбільше підходить для аналізу зображень (рис. 11.6).

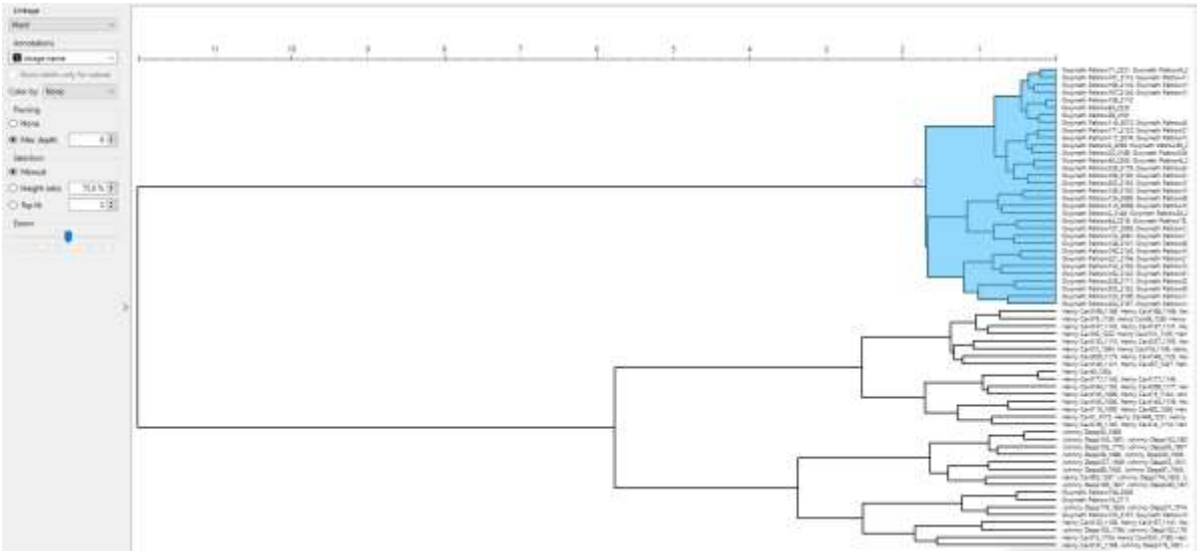


Рис. 11.6. Модуль Hierarchial Clustering

Можна побачити, як зображення поділились на кластери, а саме: 2 великих кластери, що відповідають за стать зображеної людини, 2 кластери менше у випадку чоловічої статі, що відповідають двом персонам.

Підключивши модуль Image Viewer, ми можемо побачити зображення саме з обраного кластеру.

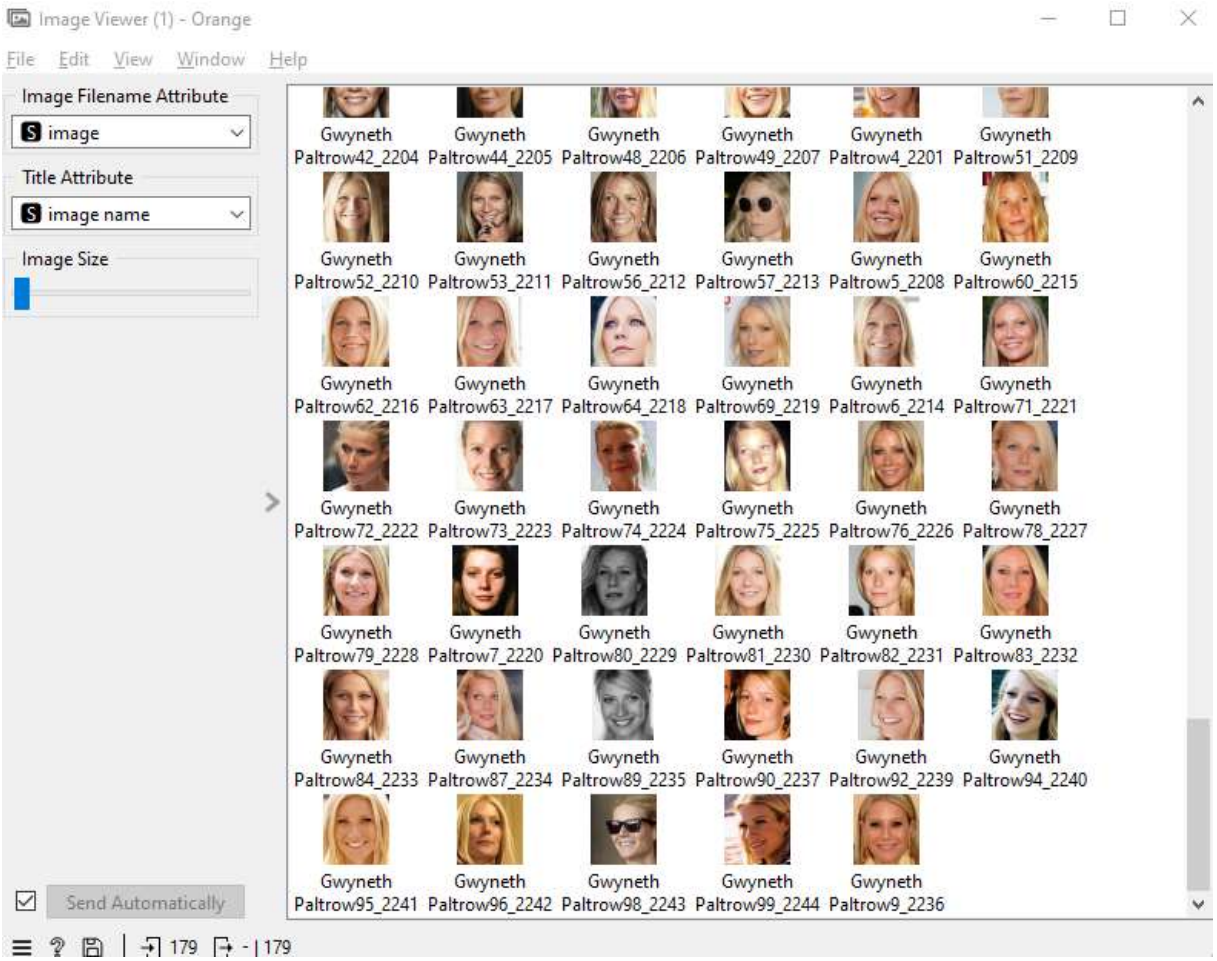


Рис. 11.7. Модуль Image Viewer

Як видно, відображаються лише фото Гвінет Пелтроу.

Наступним кроком підключимо до модуля Image Embedding 3 модулі: Image Viewer, Neighbors та Image Viewer таким чином:

Обравши зображення у Image Viewer (2) ми побачимо 10 (такі налаштування в модулі Neighbors) східних з ним зображень у Image Viewer (3).

Рис. 11.8. Підключення модулів до Image Embedding

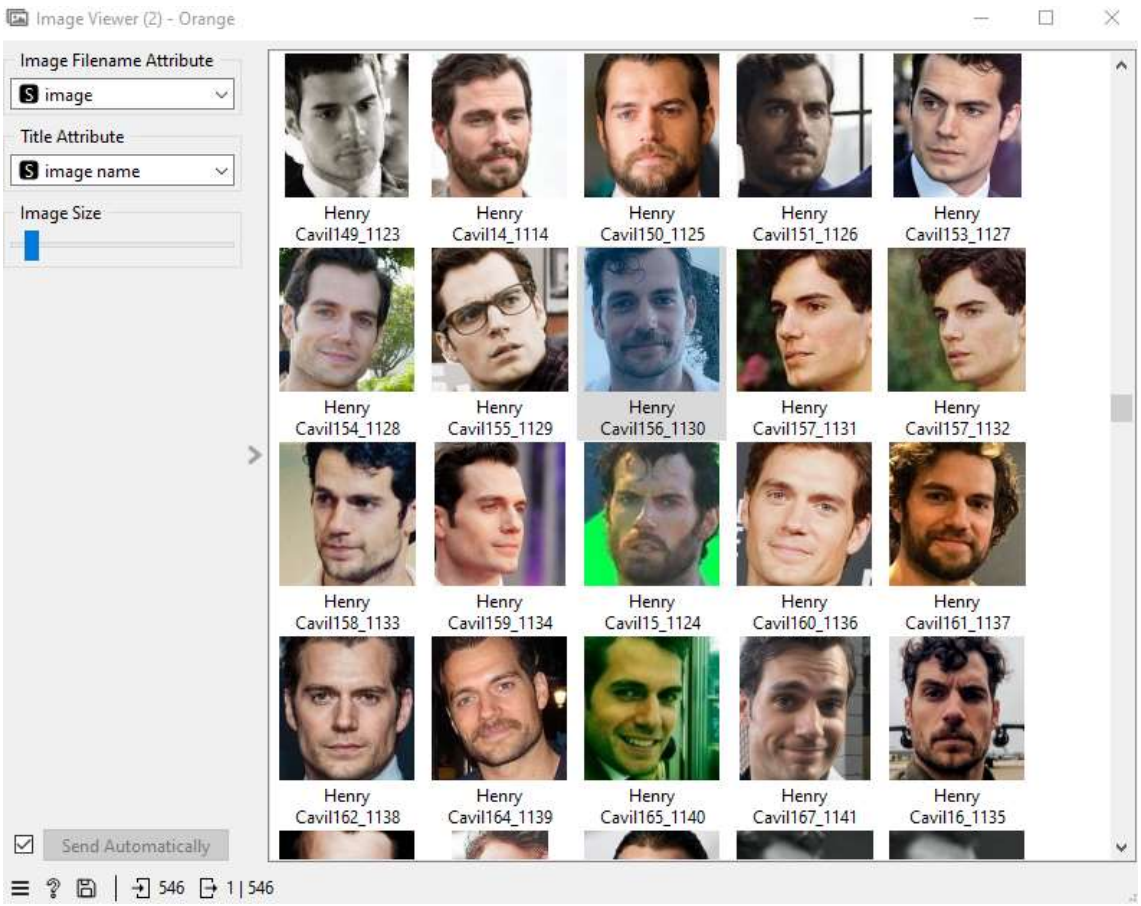


Рис. 11.9. Image Viewer (2)

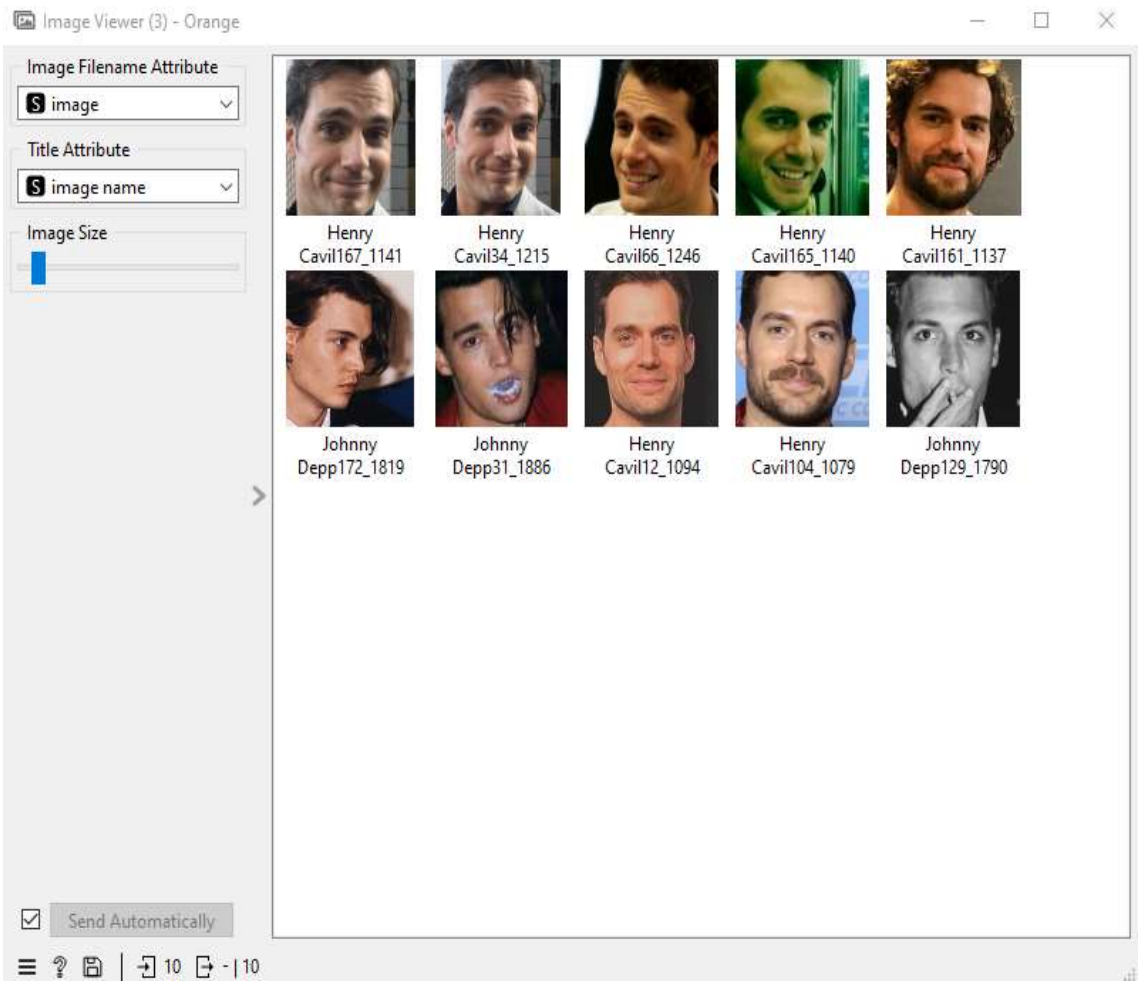


Рис. 11.10. Image Viewer (3)

Як бачимо, присутня похибка адже застосунок обрав також деякі схожі фото Джонні Деппа, але в нашому випадку вона прийнятна.

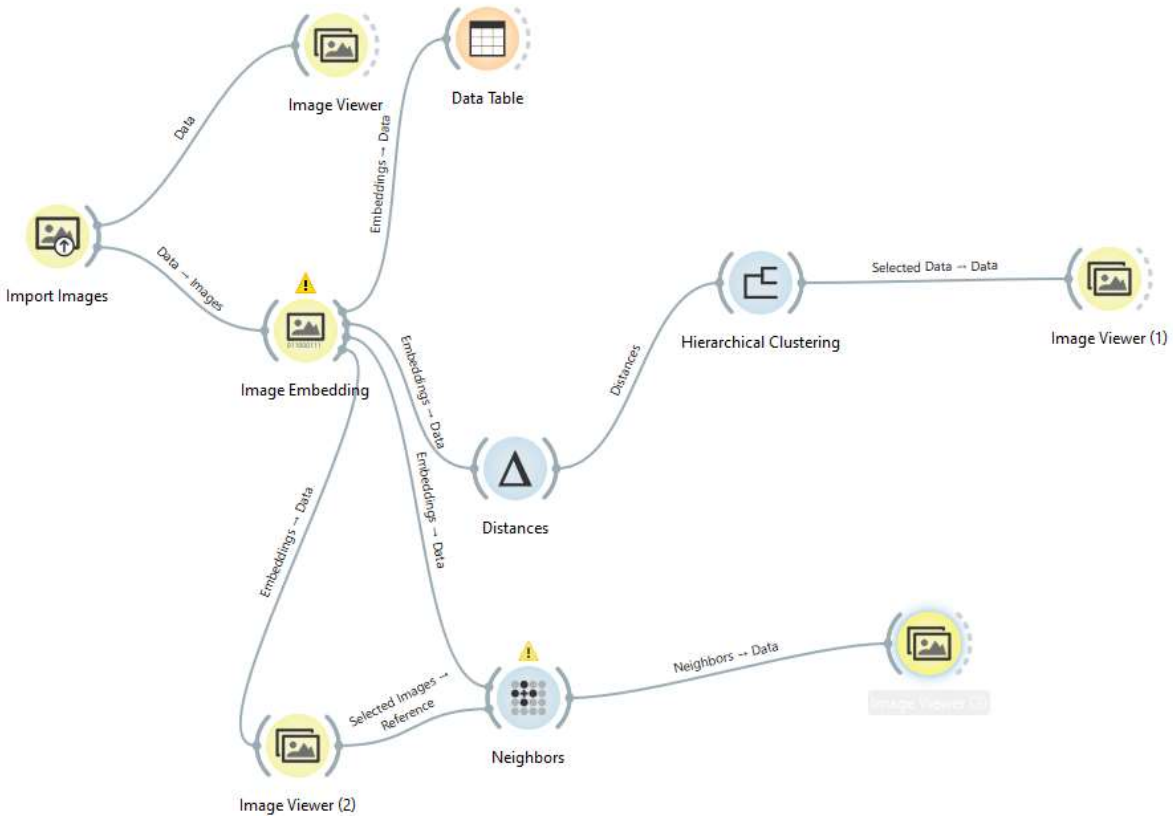


Рис. 11.11. Фінальна модель

Класифікація

Задачу класифікації почнемо знову з підключення модуля Import Images.

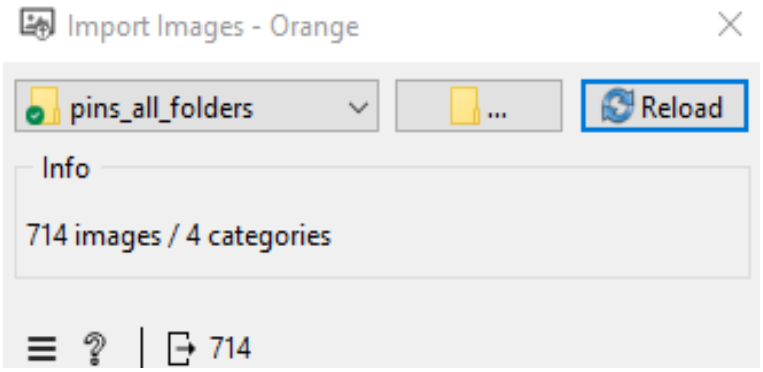


Рис. 11.12. Модуль Import Images

Але цього разу зображення розбиті по папкам і мають тег з відповідним ім’ям папки. Цей тег буде використовуватись для класифікації.

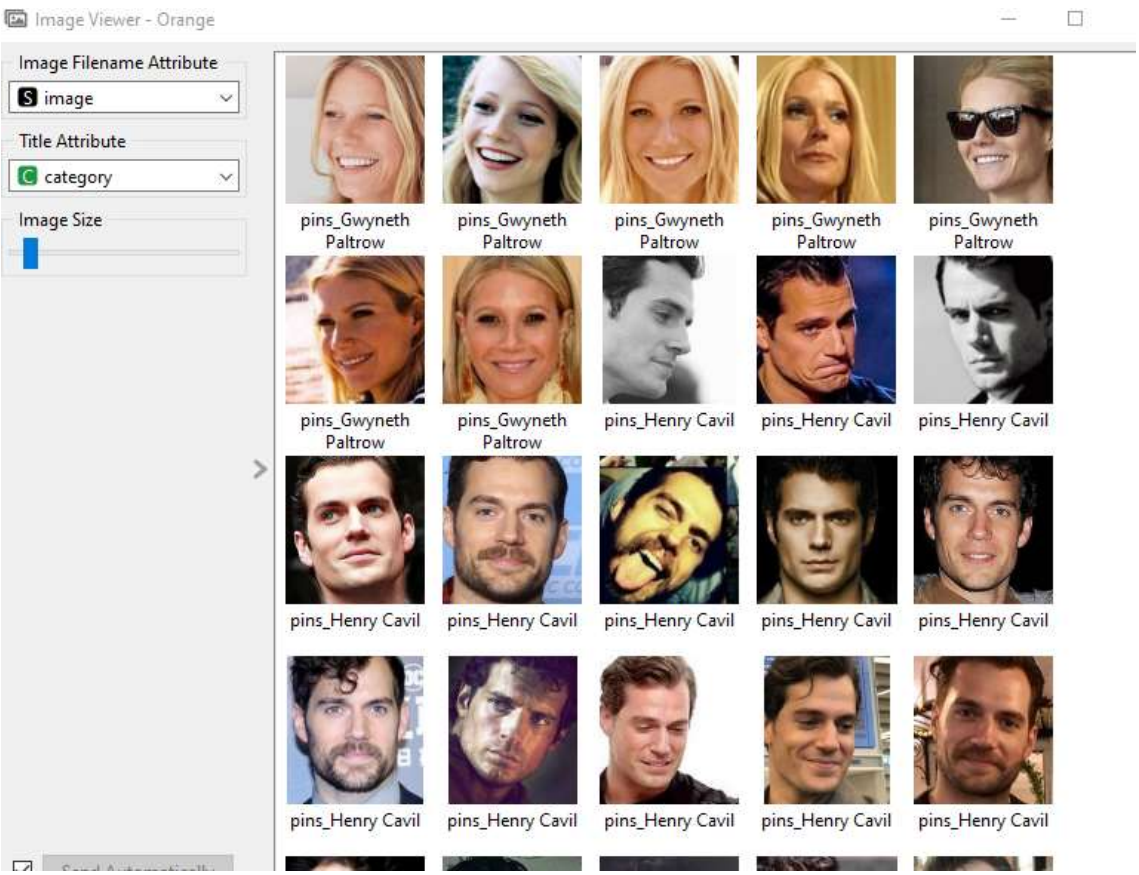


Рис. 11.13. Image Viewer

Тепер підключимо модуль Images Embedding.

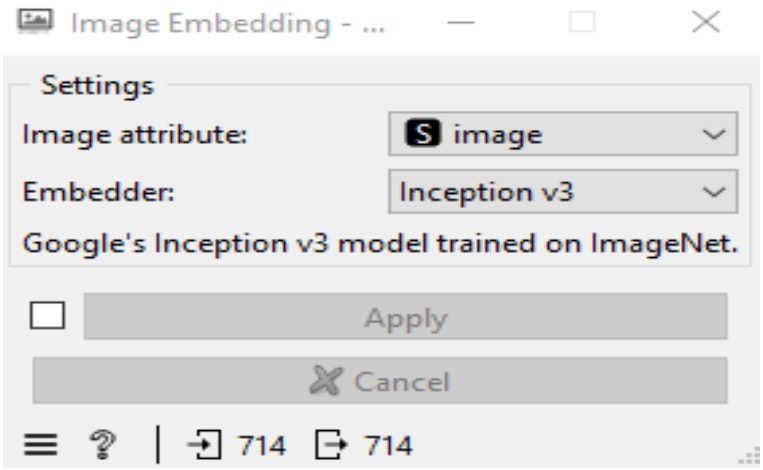


Рис. 11.14. Images Embedding

Цього разу в тестових цілях я обрав більш універсальну модель Inception v3, хоча ми все ще працюємо з обличчями.

Підключимо до Images Embedding модулі Logical Regression та Test and Score аби побачити характеристики результату роботи.

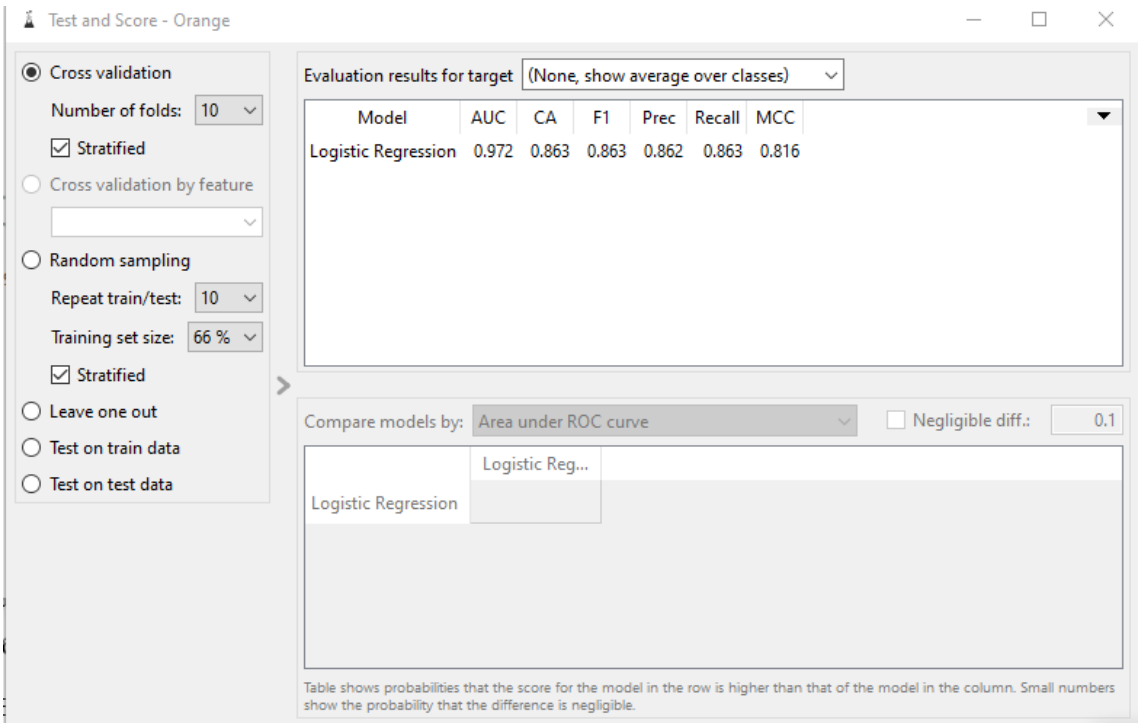


Рис. 11.15. Test and Score

Результат AUC 0.972 є дуже добрим результатом, але ми можемо підключити модуль Confusion Matrix, щоб побачити де модель спрацювала хибно.

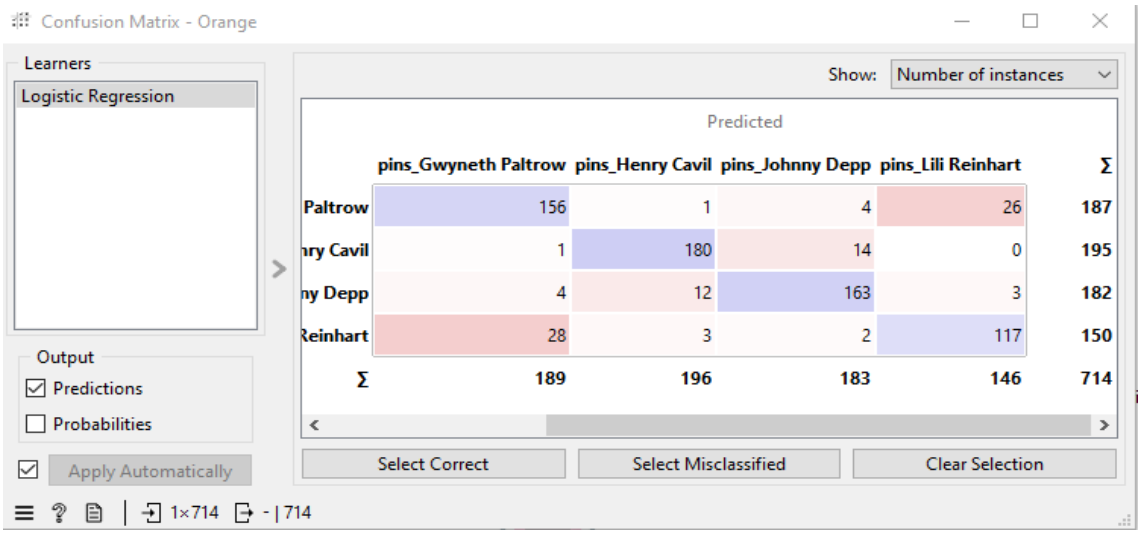


Рис. 11.16. Confusion Matrix

Видно, що в більшості випадків результат роботи правильний, але є декілька хибних варіантів. Цікаво, що схибила модель найбільше з особами одної статі. Гвінет Пелтроу більш за все путалася з Лілі Райнхарт і навпаки, а Джонні Депп – з Генрі Кавілом.

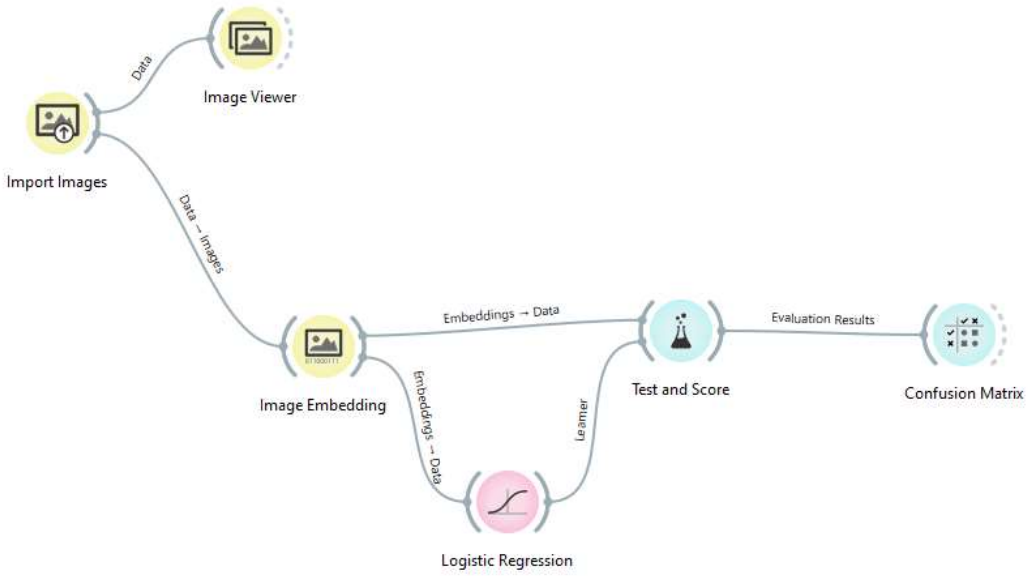


Рис. 11.17. Фінальна модель

Висновок

Таким чином, у ході виконання роботи з кластеризації та класифікації зображень було використано середовище Orange і відкритий датасет із зображеннями осіб. Для кластеризації застосовувалася модель глибинного навчання OpenFace через модуль Image Embedding, яка конвертувала зображення в набір числових параметрів. Далі модуль Distances із косинусною метрикою дозволив аналізувати схожість між зображеннями, а модуль Hierarchical Clustering успішно розподілив їх на кластери за статтю та індивідуальними ознаками. У класифікації використовувалася модель Inception v3, що забезпечила високу якість при прогнозуванні класів зображень, тегованих за іменами папок. Аналіз Confusion Matrix показав, що помилки класифікації переважно виникали між особами однієї статі. Робота продемонструвала високу ефективність глибинних моделей для задач кластеризації та класифікації зображень, але підкреслила важливість вибору відповідної моделі та метрик для конкретної задачі.

Перелік питань

1. Що таке машинне навчання, і які його основні типи?
2. Чим відрізняється навчання з учителем (supervised learning) від навчання без учителя (unsupervised learning)?
3. Що таке функція втрат (loss function) і як вона використовується?
4. Що таке модель, параметри і гіперпараметри в контексті машинного навчання?
5. Як працює крос-валідація і навіщо вона потрібна?

ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

1. Оформлення тексту

Текстовий та графічний матеріали записки друкують комп'ютерним способом на одному боці односортних білих аркушів формату А4 (розмір 210 х 297 мм) через 1,5 міжрядковий інтервал, текст вирівнюють по ширині аркуша. Текстовий редактор – Word з пакета Microsoft Office, Open Office Writer, Star Office Writer та ін. Шрифт – Times New Roman Cyr, 14.

2. Оформлення ілюстрацій

Усі ілюстрації в пояснювальній записці (креслення, схеми, фотографії, діаграми, графіки) називають рисунками. Кількість ілюстрацій має бути достатньою для пояснення тексту, який викладається. Ілюстрації потрібно розміщувати як по тексту записки (якомога ближче до відповідних частин тексту), так і в кінці його або наводити в додатках. Ілюстрації належить виконувати у відповідності до вимог стандартів ЄСКД і ЕСПД за допомогою різних графічних редакторів та систем автоматизованого проєктування.

Усі ілюстрації послідовно нумерують у межах розділу арабськими цифрами. Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, наприклад, «Рис 2.5 Граф алгоритму». Посилання на ілюстрації подають так: «... на рис. 2.5 ...». Повторне посилання на ілюстрацію наводять із скороченням слова «дивись», наприклад, «... див. рис. 2.5 ...». Допускається нумерація ілюстрацій у межах лабораторної роботи.

Ілюстрації повинні мати назву, яку розміщують під ілюстрацією в одному рядку з її номером, наприклад, «Рис. 3.2. Схема». За потреби під назвою ілюстрації записують пояснювальні дані.

Розмір шрифту всіх без винятку надписів у рисунках має бути таким самим, як і в тексті пояснювальної записки.

Ілюстрації розміщують так, щоб їх можна було розглядати, не повертаючи аркуш або повертаючи його за ходом стрілки годинника.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

Основна література

1. Berry L. Data Mining Techniques For Marketing, Sales, and Customer Relationship Management / L. Berry, G. Linoff; Second Edition. – Indianapolis, Indiana: Wiley Publishing, Inc., 2004. – 672 p.
2. Berry M. Lecture Notes in Data Mining / M.Berry, M.Browne .– Singapore: World Scientific Publishing Co., 2006. – 237 p.
3. Charu C. Aggarwal Data Mining. The Textbook. / C.A.Charu; – Springer, 2015. – 746 p.
4. Christopher M. Bishop Pattern Recognition and Machine Learning / M.B.Christopher. – Springer, 2006. – 758 p.
5. Data Analysis, Machine Learning and Applications / C.Preisach, H.Burkhardt, L.Schmidt-Thieme, R.Decker and etc.; Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation, Albert-Ludwigs-Universität Freiburg, March 7-9, 2007 .– Berlin Heidelberg: Springer-Verlag, 2008. – 703 p.
6. de Oliveira J. Valente Advances in Fuzzy Clustering and its Applications / J. Valente de Oliveira, W. Pedrycz .– West Sussex England: John Wiley & Sons Ltd, 2007. – 457 p.
7. Han J. Data Mining Concepts and Techniques / J.Han, M.Kamber, J.Pei; Third Edition. – 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann Publishers is an imprint of Elsevier, 2012. – 740 p.
8. Hastie T. The Elements of Statistical Learning Data Mining, Inference, and Prediction / T.Hastie, R.Tibshirani, J.Friedman; Second Edition .– Springer, 2017. – 764 p.
9. Nisbet R. Handbook of statistical analysis and data mining applications / R.Nisbet, J.Elder, G.Miner. – Elsevier Inc., 2009. – 860 p.

Додаткова література

1. Babuska R. Improved Covariance Estimation for Gustafson-Kessel Clustering / R.Babuska, d.V.van, U.Kaymak // 2002 IEEE International Conference on Fuzzy Systems FUZZ-IEEE '02 .– Honolulu, HI, USA, 2002. – P. 1081–1085.
2. Biemann C. Chinese Whispers – an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems / C.Biemann // Workshop on TextGraphs at HTL-NAACL. – New York City, June 2006. – P. 73–80 .— Режим доступу: https://www.researchgate.net/publication/228670574_Chinese_whispers_An_efficient_graph_clustering_algorithm_and_its_application_to_natural_language_processing_problems

3. Chin-Tang Chang A Fuzzy K-means Clustering Algorithm Using Cluster Center Displacement / Chin-Tang Chang, Jim Z.C. Lai and Mu-Der Jeng // Journal of Information Science and Engineering. – 2011. — № 27. – P. 995–1009.
4. Clustering with Minimum Spanning Trees / Y. Zhou, O. Grygorash, T .F. Hain та ін. – Elsevier, 2010. – 22 p.
5. Jain A. K. Data Clustering: A Review / A. K. Jain, M. Murty, P. J. Flynn // ACM Computing Surveys. – 1999. – Vol. 31, No. 3. – P. 264–323.
6. Mercer D. P. Clustering large datasets / D.P.Mercer. – Linacre College, 2003. – 50 p.
7. Semi-supervised graph clustering: a kernel approach / B.Kulis, S.Basu, I.Dhillon, R.Mooney // Mach Learn. – 2009. – № 74. – P. 1–22.

Додаток А

Одеський національний університет імені І. І. Мечникова

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту/факультету)

Кафедра комп’ютерних систем та технологій

(повна назва кафедри)

Звіт з лабораторної роботи № __

з дисципліни ДС 2 «Методи класифікації образів»

Тема: «Метод к-середніх»

Виконав(ла): студент (ка) денної форми навчання
спеціальності 123 «Комп’ютерна інженерія»
курс___гр. ___ № З.К._____

Прізвище Ім’я По-батькові_____

Керівник д. т. н., (к. т. н.) проф.(доц.) каф___
Прізвище І. Б. _____

Одеса – 2026

Навчальне видання

МЕТОДИ КЛАСИФІКАЦІЇ І КЛАСТЕРИЗАЦІЇ ДАНИХ

НАВЧАЛЬНО-МЕТОДИЧНИЙ ПОСІБНИК

для здобувачів факультету математики, фізики
та інформаційних технологій
спеціальності F7/123 Комп'ютерна інженерія

Електронне видання мережевого використовування

Укладачі:
Михайленко Владислав Сергійович
Гунченко Юрій Олександрович
Мартинович Лариса Ярославівна
Камєнєва Алла Вікторівна

В авторській редакції

Затв. авт. 20.05.2026. Шрифт Times New Roman.
Системні вимоги: операційна система сумісна з програмним забезпеченням
для читання файлів формату PDF.
Обсяг 3,4 МБ. Зам. № 3152.

Видавець:
Одеський національний університет імені І. І. Мечникова
вул. Змієнка Всеволода, буд. 2, м. Одеса, 65001, Україна
Свідоцтво суб'єкта видавничої справи ДК № 8592 від 23.03.2026 р.
Тел.: (048) 723 28 39, e-mail: druk@onu.edu.ua