

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідуючий кафедрою _____ О.О. Ємець
(підпис, ініціали, прізвище)

« ____ » _____ 2020 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ**

на тему

РОЗРОБКА ТРЕНАЖЕРУ З ТЕМИ «КЛАСИФІКАТОР NAIVE BAYES»
ДИСЦИПЛІНИ «КОМП'ЮТЕРНИЙ АНАЛІЗ СТАТИСТИЧНИХ ДАНИХ»

зі спеціальності 122 «Комп'ютерні науки та інформаційні технології»

Виконавець роботи

Шаркий Ярослав Іванович _____ «__» _____ 2020 р.
(підпис)

Науковий керівник

к.ф.-м.н, Олексійчук Юрій Федорович _____ «__» _____ 2020 р.
(підпис)

Полтава – 2020 р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПОСТАНОВКА ЗАДАЧІ.....	6
2 ІНФОРМАЦІЙНИЙ ОГЛЯД	8
3 ТЕОРЕТИЧНА ЧАСТИНА.....	11
3.1 Класифікатор Naïve Bayes	11
3.2 Розробка алгоритму роботи тренажеру	17
4 ПРАКТИЧНА РЕАЛІЗАЦІЯ	20
4.1 Розробка тренажеру	20
4.2 Тестування тренажеру	33
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	48
ДОДАТОК А. КОД РОЗРОБЛЕНОГО ТРЕНАЖЕРА	50

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
GUI	Graphical user interface – Графічний інтерфейс користувача
JavaFX	Платформа та набір інструментів для створення графічного інтерфейсу користувача
Java	Об'єктно-орієнтована мова програмування
IntelliJ IDEA	Інтегроване середовище розробки для створення програмного забезпечення на мові програмування Java
Naive Bayes	Метод класифікації

ВСТУП

В Полтавському університеті економіки і торгівлі і, зокрема, на кафедрі математичного моделювання та соціальної інформатики в останні роки впроваджується дистанційна форма освіти.

Дистанційна освіта дає багато додаткових можливостей для студентів. До її переваг можна віднести наступне.

1. Можливість навчатися, перебуваючи в іншому місті, або, навіть, закордоном.

2. Можливість навчатися у зручний час. Це дозволяє поєднувати навчання з роботою або навчанням в іншому навчальному закладі.

3. Можливість навчатися у зручному для себе темпі.

Дистанційне навчання ґрунтується насамперед на самонавчанні студентів. Для цього необхідне гарне інформаційне забезпечення. Частково воно реалізується через дистанційні курси. Але для виконання практичних завдань часто цього недостатньо.

Тому на кафедрі математичного моделювання та соціальної інформатики створено навчальні тренажери, як складові різних дистанційних курсів [1-7]. Як правило, кожен тренажер призначений, щоб допомогти користувачу зрозуміти певну тему. З дисципліни «Комп'ютерний аналіз статистичних даних» також створюються навчальні тренажери. Наприклад, створений тренажер з теми «1-R алгоритм» [6-7]. З теми «Класифікатор Naive Bayes» тренажер до цього не був розроблений, хоча ця тема не проста і використання тренажеру може суттєво допомогти студентам. Отже, розробка тренажеру з теми «Класифікатор Naive Bayes» дистанційного навчального курсу «Комп'ютерний аналіз статистичних даних» є **актуальною**.

Метою роботи є програмна реалізація тренажеру з теми «Класифікатор Naive Bayes» дисципліни «Комп'ютерний аналіз статистичних даних». Для досягнення цієї мети потрібно розв'язати такі основні **задачі**:

- 1) розробка алгоритму роботи тренажеру;
- 2) побудова блок-схеми алгоритму;

3) вибір технології та мови програмування;

4) програмна реалізація тренажеру.

Об'єктом розробки є програмне забезпечення для навчання студентів.

Предмет розробки – програмне забезпечення навчального тренажеру з теми «Класифікатор Naive Bayes» дисципліни «Комп'ютерний аналіз статистичних даних».

Методи: методи класифікації, методи Naive Bayes, методи розробки на мові програмування Java.

Робота складається із вступу, чотирьох розділів, висновків та списку літератури. Також робота має додаток, в якому розміщений програмний код тренажеру.

Для розробки тренажеру використовується мова програмування Java, для створення графічного інтерфейсу користувача (GUI) – технологія JavaFX. Тренажер створений за допомогою інтегрованого середовища розробки (IDE – Integrated development environment) IntelliJ IDEA Community.

Новизною роботи є розроблений тренажер з теми «Класифікатор Naive Bayes» дисципліни «Комп'ютерний аналіз статистичних даних». Його основними перевагами є:

- ✓ незалежність від апаратної та операційної платформи;
- ✓ можливість інтеграції в дистанційний курс;
- ✓ можливість запускати із комп'ютера користувача.

1 ПОСТАНОВКА ЗАДАЧІ

Головною задачею бакалаврської роботи є створення навчального тренажеру з теми «Класифікатор Naive Bayes» дисципліни «Комп'ютерний аналіз статистичних даних».

Тренажер призначений для того, щоб допомогти користувачу розібратися з тим, як працює класифікатор Naive Bayes. Для цього буде запропоновано певні дані і поставлена задача. Використовуючи дані, користувач повинен розв'язати поставлену задачу. Задача буде розв'язуватися покроково. На кожному кроці користувач повинен дати відповідь на конкретне питання або обрахувати певне значення. Якщо користувач успішно із цим справляється, то відбувається перехід на наступний крок. У випадку помилкової відповіді користувач отримує підказку і повторює спробу. Такий підхід дозволяє відносно просто розв'язати складну задачу. Також у користувача буде мотивація продовжувати роботу з тренажером, а не кинути її після невдачі на деякому етапі.

Тренажер призначений для студентів різних форм навчання, але, найперше, для студентів, які самостійно опановують дисципліну. В першу чергу, це студенти дистанційної та заочної форм навчання.

Тема «Класифікатор Naive Bayes» вивчається в рамках дисципліни «Комп'ютерний аналіз статистичних даних» після теми «1-R алгоритм», з якої вже створений навчальний тренажер [6-7]. Тому однією з рекомендацій до створюваного тренажера є схожість графічного інтерфейсу з тренажером з теми «1-R алгоритм». 1-R алгоритм – це також класифікатор. Тому для цих задач будуть використовуватися схожі приклади. Це має суттєво спростити адаптацію користувача до роботи з цим тренажером.

Основні вимоги до тренажеру:

1. Незалежність від апаратної та операційної платформи. Тренажер буде використовуватися студентами при самостійному навчанні, тому повинен працювати на різних операційних системах.

2. Простота у використанні. Тренажер буде використовуватися при вивченні лише однієї теми. Ця тема вивчається недовго. Тому в користувача не буде мотивації довго розбиратися з роботою тренажеру. Тренажер повин мати простий та інтуїтивно зрозумілий інтерфейс і принцип роботи.

3. Можливість інтеграції в дистанційний курс з дисципліни «Комп'ютерний аналіз статистичних даних». Дистанційні курси в Полтавському університеті економіки і торгівлі базуються на системі Moodle. Тренажер має бути створений таким чином, щоб його можна було підключити до цієї системи.

Для розв'язання поставленої задачі потрібно виконати такі етапи.

1. Аналіз навчальних тренажерів з різних тем та дисциплін.
2. Вивчення та аналіз теоретичного матеріалу з теми «Класифікатор Naive Bayes».
3. Вибір мови програмування та технології, на якій буде створений тренажер.
4. Розробка алгоритму роботи тренажеру.
5. Побудова блок-схеми алгоритму роботи тренажеру.
6. Безпосередньо програмна реалізація тренажеру.
7. Тестування тренажеру і усунення виявлених помилок.

Після виконання цих етапів буде створений тренажер з теми «Класифікатор Naive Bayes», який може використовуватися при вивченні дисципліни «Комп'ютерний аналіз статистичних даних».

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

Розвиток мережі Інтернет, зростання пропускних спроможностей мереж сприяють розвитку дистанційної освіти в Україні та світі. Дистанційна освіта потребує відповідного програмного забезпечення.

Для пришвидшення розробки web-сайтів існує велика кількість CMS (Content Management System) — систем керування контентом. Існують CMS призначені для певних типів сайтів: блоги, форумів, інтернет-магазини та ін. Також існують CMS для освітніх сайтів — системи керування навчанням (Learning Management System, LMS). Прикладом такої системи є Moodle.

Moodle — це безкоштовна, відкрита система управління навчанням. Орієнтована насамперед на організацію взаємодії між викладачем та учнями, хоча підходить і для організації дистанційних курсів, а також підтримки очного навчання. Moodle має також українську локалізацію.

Типова функціональність Moodle включає:

- здачу завдань;
- дискусійні форуми;
- завантаження файлів;
- оцінювання;
- обмін повідомленнями;
- календар подій;
- новини та анонси подій (для різних рівнів: сайт, курс, навчальна група);
- тестування;
- вікі.

Розробники можуть створювати додаткові модулі з новою додатковою функціональністю.

У Вищому навчальному закладі Укоопспілки "Полтавський університет економіки і торгівлі" для дистанційного навчання також використовується система Moodle.

Навчальні тренажери створюються для розширення можливостей системи Moodle під час практичної підготовки студентів. На кафедрі математичного моделювання та соціальної інформатики створена значна кількість тренажерів [1-7]. Як правило, тренажер призначений для того, щоб допомогти студенту розібратися з конкретною темою або як розв'язувати деякий тип прикладів з певної дисципліни.

З дисципліни «Комп'ютерний аналіз статистичних даних» створений навчальний тренажер з теми «1-R алгоритм» [6-7]. Тренажер розроблений на мові програмування Java. Головне вікно тренажеру складається з двох частин (рис. 2.1).

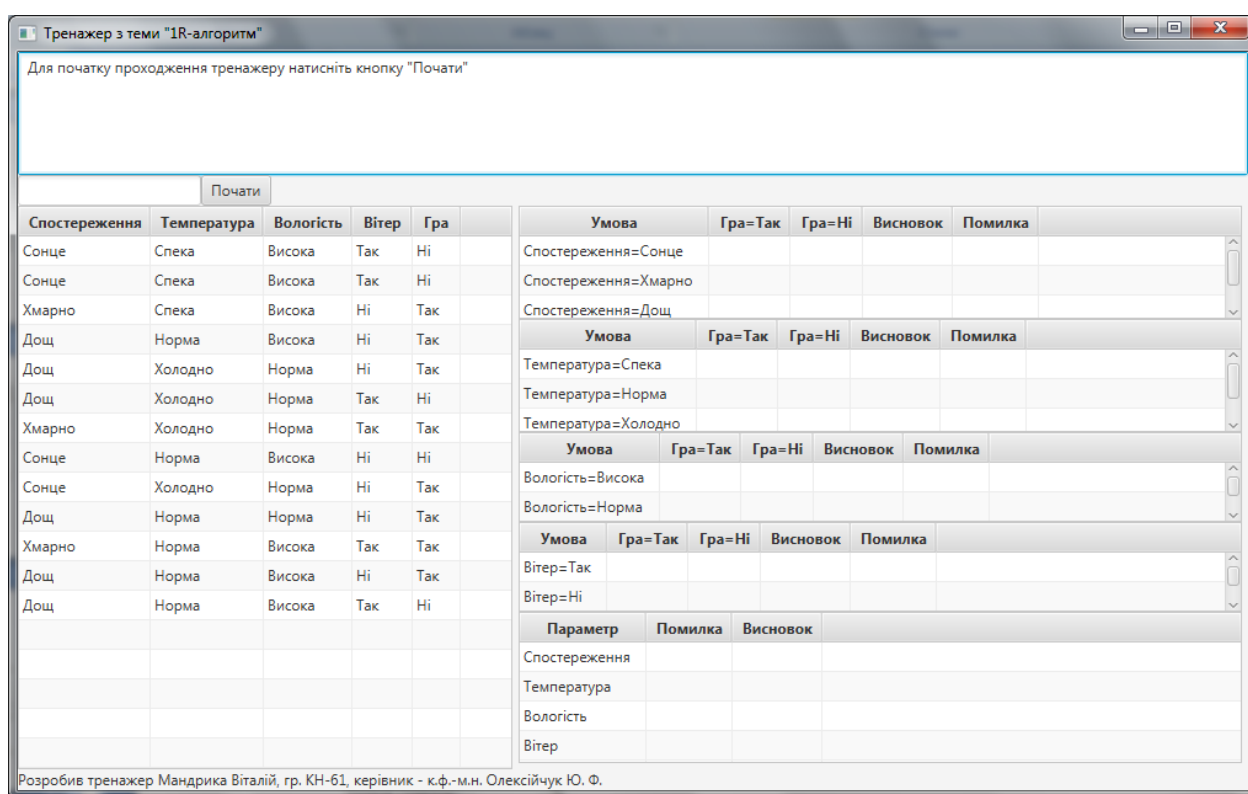


Рис. 2.1. Головне вікно тренажеру з теми «1-R алгоритм»

В лівій частині вікна показана вибірка, за допомогою якої будуються правила класифікації. В правій частині розміщені додаткові таблиці, які необхідні для того, щоб розв'язати задачу.

У верхній частині тренажеру розміщений блок, в якому на кожному етапі задається певне питання, а користувач повинен дати відповідь на це питання (ввести

числове значення). У випадку неправильної помилки користувачу дається підказка і він повторює спробу.

Роботу тренажеру успішно завершено!

Відповідь

Спостереження	Температура	Вологість	Вітер ^Δ	Гра
Сонце	Норма	Висока	Ні	Ні
Дощ	Норма	Висока	Ні	Так
Дощ	Норма	Висока	Ні	Так
Хмарно	Спека	Висока	Ні	Так
Дощ	Норма	Норма	Ні	Так
Дощ	Холодно	Норма	Ні	Так
Сонце	Холодно	Норма	Ні	Так
Дощ	Норма	Висока	Так	Ні
Сонце	Спека	Висока	Так	Ні
Сонце	Спека	Висока	Так	Ні
Дощ	Холодно	Норма	Так	Ні
Хмарно	Норма	Висока	Так	Так
Хмарно	Холодно	Норма	Так	Так

Умова	Гра=Так	Гра=Ні	Висновок	Помилка
Спостереження=Сонце	1	3	Ні	1
Спостереження=Хмарно	3	0	Так	0
Спостереження=Дощ	4	2	Так	2

Умова	Гра=Так	Гра=Ні	Висновок	Помилка
Температура=Спека	1	2	Ні	1
Температура=Норма	4	2	Так	2
Температура=Холодно	3	1	Так	1

Умова	Гра=Так	Гра=Ні	Висновок	Помилка
Вологість=Висока	4	4	Ні	4
Вологість=Норма	4	1	Так	1

Умова	Гра=Так	Гра=Ні	Висновок	Помилка
Вітер=Так	2	4	Ні	2
Вітер=Ні	6	1	Так	1

Параметр	Помилка	Висновок
Спостереження	3	
Температура	4	
Вологість	5	
Вітер	3	Найкраща класифікація

Розробив тренажер Мандрика Віталій, гр. КН-61, керівник - к.ф.-м.н. Олексійчук Ю. Ф.

Рис. 2.2. Вікно тренажеру з теми «1-R алгоритм» після розв'язання прикладу

Після проходження всіх етапів студент отримує розв'язання початкової задачі (рис. 2.2).

В результаті аналізу різних тренажерів, було зроблено висновок, що оптимальним вибором для реалізації тренажеру буде мова програмування Java. Графічний інтерфейс буде подібний до тренажеру з теми «1-R алгоритм» (рис. 2.1). В лівій частині будуть розміщуватися початкові дані, а правій – розв'язання. Графічний інтерфейс користувача (GUI) тренажера буде створений за допомогою технології JavaFX.

3 ТЕОРЕТИЧНА ЧАСТИНА

3.1 Класифікатор Naive Bayes

Розглянемо основний теоретичний матеріал, який необхідний для побудови алгоритму роботи тренажеру та його реалізації [8].

Задача класифікації — це формалізована задача, задана множиною об'єктів або ситуацій, що розбита деяким чином на кілька класів. Для деяких об'єктів відомо, до яких класів вони відносяться. Множина таких об'єктів називається навчальною вибіркою. Для інших об'єктів заздалегідь невідомо, до якого із класів вони належать. Необхідно побудувати алгоритм, який зможе віднести кожен об'єкт до певного класу, тобто класифікувати об'єкт.

Приклад задачі. Дано інформацію про проведення спортивних ігор при різних погодних умовах (табл. 3.1).

Задача полягає у побудові правил, які дозволять для будь-якого набору значень незалежних змінних (Спостереження, Температура, Вологість та Вітер) визначити: гра відбудеться чи ні.

Таблиця 3.1. Дані про спортивні змагання

Спостереження	Температура	Вологість	Вітер	Гра
Сонце	Спека	Висока	Ні	Ні
Сонце	Спека	Висока	Так	Ні
Хмарно	Спека	Висока	Ні	Так
Дощ	Норма	Висока	Ні	Так
Дощ	Холодно	Норма	Ні	Так
Дощ	Холодно	Норма	Так	Ні
Хмарно	Холодно	Норма	Так	Так
Сонце	Норма	Висока	Ні	Ні
Сонце	Холодно	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так

Сонце	Норма	Норма	Так	Так
Хмарно	Норма	Висока	Так	Так
Хмарно	Спека	Норма	Ні	Так
Дощ	Норма	Висока	Так	Ні

Для розв'язання поставленої задачі можуть бути використані різні методи. Розглянемо один із них.

Метод Naïve Bayes для класифікації використовує кілька незалежних змінних. Метод використовує формулу Байеса для розрахунку ймовірності. Вважається, що всі змінні незалежні між собою, тому назва naïve (наївний). Це не завжди так, тобто дуже часто спостерігається залежність між змінними, але алгоритм часто застосовний на практиці.

$P(y = c_r)$ — ймовірність того, що деякий об'єкт належить класу c_r .

E — подія, що полягає у рівності незалежних змінних певним значенням, $P(E)$ — її ймовірність.

Ідея методу полягає у розрахунку умовної ймовірності належності об'єкта до класу c_r при рівності його незалежних змінних певним значенням.

$$P(y = c_r | E) = \frac{P(E | y = c_r) P(y = c_r)}{P(E)} \quad (3.1)$$

Вважаючи, що незалежні змінні набувають своїх значень незалежно одна від одної, виразимо $P(E | y = c_r)$ через добуток ймовірностей для кожної незалежної змінної:

$$P(E | y = c_r) = P(x_1 = c_p^1 | y = c_r) \times P(x_2 = c_d^2 | y = c_r) \times \dots \times P(x_m = c_b^m | y = c_r) \times P(y = c_r) / P(E) \quad (3.2)$$

Ймовірність рівності незалежної змінної конкретному значенню, при умові належності залежної змінної класу c_r :

$$P(x_h = c_d^h | y = c_r) = \frac{P(x_h = c_d^h \text{ and } y = c_r)}{P(y = c_r)} \quad (3.3)$$

Приклад: Використовуючи формулу (3.3), знайдемо наступні ймовірності

$$P(\text{Спостереження} = \text{Сонце} | \text{Гра} = \text{Так})$$

$$P(\text{Спостереження} = \text{Хмарно} | \text{Гра} = \text{Так})$$

$$P(\text{Спостереження} = \text{Дощ} | \text{Гра} = \text{Так})$$

$$P(\text{Спостереження} = \text{Сонце} | \text{Гра} = \text{Ні})$$

$$P(\text{Спостереження} = \text{Хмарно} | \text{Гра} = \text{Ні})$$

$$P(\text{Спостереження} = \text{Дощ} | \text{Гра} = \text{Ні})$$

Розглянемо окремо ситуації, коли гра відбулася (табл. 3.2) та коли гра не відбулася (табл. 3.3).

Таблиця 3.2. Дані про спортивні змагання, коли гра відбулася

Спостереження	Температура	Вологість	Вітер	Гра
Хмарно	Спека	Висока	Ні	Так
Дощ	Норма	Висока	Ні	Так
Дощ	Холодно	Норма	Ні	Так
Хмарно	Холодно	Норма	Так	Так
Сонце	Холодно	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Сонце	Норма	Норма	Так	Так
Хмарно	Норма	Висока	Так	Так
Хмарно	Спека	Норма	Ні	Так

Як видно з таблиці 1.2, гра відбулася у 9 випадках. При цьому у 2-х випадках було сонце, у 4-х випадках – хмарно, і в 3-х випадках – дощ. Тому

$$P(\text{Спостереження} = \text{Сонце} | \text{Гра} = \text{Так}) = 2/9$$

$$P(\text{Спостереження} = \text{Хмарно} | \text{Гра} = \text{Так}) = 4/9$$

$$P(\text{Спостереження} = \text{Дощ} | \text{Гра} = \text{Так}) = 3/9$$

Таблиця 3.3. Дані про спортивні змагання, коли гра не відбулася

Спостереження	Температура	Вологість	Вітер	Гра
Сонце	Спека	Висока	Ні	Ні
Сонце	Спека	Висока	Так	Ні
Дощ	Холодно	Норма	Так	Ні
Сонце	Норма	Висока	Ні	Ні
Дощ	Норма	Висока	Так	Ні

Гра не відбулася в 5 випадках. В трьох із них було сонце, в двох – дощ і в жодному випадку не було хмарно. Тому

$$P(\text{Спостереження} = \text{Сонце} | \text{Гра} = \text{Ні}) = 3/5$$

$$P(\text{Спостереження} = \text{Хмарно} | \text{Гра} = \text{Ні}) = 0/5$$

$$P(\text{Спостереження} = \text{Дощ} | \text{Гра} = \text{Ні}) = 2/5$$

Аналогічно можна знайти наступні ймовірності для температури:

$$P(\text{Температура} = \text{Спека} | \text{Гра} = \text{Так}) = 2/9$$

$$P(\text{Температура} = \text{Норма} | \text{Гра} = \text{Так}) = 4/9$$

$$P(\text{Температура} = \text{Холодно} | \text{Гра} = \text{Так}) = 3/9$$

$$P(\text{Температура} = \text{Спека} | \text{Гра} = \text{Ні}) = 2/5$$

$$P(\text{Температура} = \text{Норма} | \text{Гра} = \text{Ні}) = 2/5$$

$$P(\text{Температура} = \text{Холодно} | \text{Гра} = \text{Ні}) = 1/5$$

Також знайдемо ймовірності для вологості. Варто відмітити, що вологість має лише два можливих значення: висока та норма.

$$P(\text{Вологість} = \text{Висока} | \text{Гра} = \text{Так}) = 3/9$$

$$P(\text{Вологість} = \text{Норма} | \text{Гра} = \text{Так}) = 6/9$$

$$P(\text{Вологість} = \text{Висока} | \text{Гра} = \text{Ні}) = 4/5$$

$$P(\text{Вологість} = \text{Норма} | \text{Гра} = \text{Ні}) = 1/5$$

Аналогічно знайдемо ймовірності для вітру:

$$P(\text{Вітер} = \text{Так} | \text{Гра} = \text{Так}) = 3/9$$

$$P(\text{Вітер} = \text{Ні} | \text{Гра} = \text{Так}) = 6/9$$

$$P(\text{Вітер} = \text{Так} | \text{Гра} = \text{Ні}) = 3/5$$

$$P(\text{Вітер} = \text{Ні} | \text{Гра} = \text{Ні}) = 2/5$$

Ймовірність $P(y = c_r)$ рівна відношенню об'єктів у навчальній вибірці, що належать класу c_r , до загальної кількості об'єктів у вибірці.

$$P(\text{Гра} = \text{Так}) = 9/14$$

$$P(\text{Гра} = \text{Ні}) = 5/14$$

Нехай необхідно визначити, чи відбудеться гра при наступних умовах (події E):

Спостереження = Сонце

Температура = Холодно

Вологість = Висока

Вітер = Так

Потрібно обрахувати наступні ймовірності:

$$\begin{aligned}
P(\text{Гра} = \text{Так} | E) &= P(\text{Спостереження} = \text{Сонце} | \text{Гра} = \text{Так}) \times \\
&\times P(\text{Температура} = \text{Холодно} | \text{Гра} = \text{Так}) \times \\
&\times P(\text{Вологість} = \text{Висока} | \text{Гра} = \text{Так}) \times \\
&\times P(\text{Вітер} = \text{Так} | \text{Гра} = \text{Так}) \times P(\text{Гра} = \text{Так}) / P(E)
\end{aligned}$$

$$\begin{aligned}
P(\text{Гра} = \text{Ні} | E) &= P(\text{Спостереження} = \text{Сонце} | \text{Гра} = \text{Ні}) \times \\
&\times P(\text{Температура} = \text{Холодно} | \text{Гра} = \text{Ні}) \times \\
&\times P(\text{Вологість} = \text{Висока} | \text{Гра} = \text{Ні}) \times \\
&\times P(\text{Вітер} = \text{Так} | \text{Гра} = \text{Ні}) \times P(\text{Гра} = \text{Ні}) / P(E)
\end{aligned}$$

Підставляючи конкретні значення які були обраховані раніше, отримуємо:

$$P(\text{Гра} = \text{Так} | E) = \left(\frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14} \right) / P(E) = 0,0053 / P(E)$$

$$P(\text{Гра} = \text{Ні} | E) = \left(\frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14} \right) / P(E) = 0,0206 / P(E)$$

Нормалізуємо ймовірності по формулі

$$P'(y = c_r | E) = \frac{P(y = c_r | E)}{\sum P(y = c_i | E)} \quad (3.4)$$

$$P'(\text{Гра} = \text{Так} | E) = \frac{0,0053}{0,0053 + 0,0206} = 0,205$$

$$P'(\text{Гра} = \text{Ні} | E) = \frac{0,0206}{0,0053 + 0,0206} = 0,795$$

Тобто, можна зробити висновок, що гра за таких погодних умов не відбудеться.

При використанні формули Байєса може виникнути ситуація, коли в навчальній вибірці не буде жодного об'єкта, де змінна x_h приймає значення c_d^h і

відноситься до класу c_r . В такому випадку відповідна ймовірність буде рівною 0, а значить і ймовірність правила рівна 0. В такому випадку до кожної ймовірності додається деяка додатна величина. Така методика називається оціночною функцією Лапласа.

Перевагою Naive Bayes є нечутливість до пропущених значень. При підрахунку ймовірностей вони просто пропускаються для всіх правил.

3.2 Розробка алгоритму роботи тренажеру

Основною задачею навчального тренажеру є навчити користувача застосовувати алгоритм Naive Bayes для прийняття рішення в задачі класифікації. Користувачу буде показаний певний набір даних та поставлено питання. Тобто, потрібно буде вказати з якою ймовірністю відбудеться (чи не відбудеться) певна подія за даних умов. Студент повинен по-кроково розв'язати цю задачу. На кожному кроці буде поставлене конкретне питання, на яке потрібно дати відповідь.

Розглянемо на прикладі.

Користувачу виводиться один із наборів даних, наприклад, такий як в табл. 1.1.

Далі користувачу виводиться завдання: визначити, чи відбудеться гра при наступних умовах:

Спостереження = Хмарно

Температура = Норма

Вологість = Висока

Вітер = Ні

Користувач повинен обрахувати почергово наступні ймовірності:

$$P(\text{Спостереження} = \text{Сонце} | \text{Гра} = \text{Так}) = 2/9$$

$$P(\text{Спостереження} = \text{Сонце} | \text{Гра} = \text{Ні}) = 3/5$$

$$P(\text{Температура} = \text{Холодно} | \text{Гра} = \text{Так}) = 3/9$$

$$P(\text{Температура} = \text{Холодно} | \text{Гра} = \text{Ні}) = 1/5$$

$$P(\text{Вологість} = \text{Висока} | \text{Гра} = \text{Так}) = 3/9$$

$$P(\text{Вологість} = \text{Висока} | \text{Гра} = \text{Ні}) = 4/5$$

$$P(\text{Вітер} = \text{Так} | \text{Гра} = \text{Так}) = 3/9$$

$$P(\text{Вітер} = \text{Так} | \text{Гра} = \text{Ні}) = 3/5$$

Після цього виводиться новий блок, в якому потрібно обрахувати ще 2 ймовірності:

$$P(\text{Гра} = \text{Так}) = 9/14$$

$$P(\text{Гра} = \text{Ні}) = 5/14$$

Потім користувач повинен буде ввести значення ймовірностей $P(\text{Гра} = \text{Так} | E)$ та $P(\text{Гра} = \text{Ні} | E)$ – спочатку лише знаменник, а потім нормоване значення.

Алгоритм роботи тренажеру:

1. Користувачу показується таблиця з даними та умова завдання.
 2. Користувач по чергово вводить умовні ймовірності: ймовірність появи певного погодного явища в залежності від того, відбулася гра чи ні. Якщо відповідь правильна, то перехід на крок 4. Інакше – на крок 3.
 3. Виводиться інформації про помилку та підказка. Перехід на крок 2.
 4. Користувач вводить ймовірність того, що гра відбулася, та ймовірність того, що гра не відбулася. Якщо відповідь правильна, то перехід на крок 6. Інакше – на крок 5.
 5. Виводиться інформації про помилку та підказка. Перехід на крок 4.
 6. Користувач обраховує чисельники ймовірностей даної в умові завдання події та протилежної до неї. Якщо відповідь правильна, то перехід на крок 8. Інакше – на крок 7.
 7. Виводиться інформації про помилку та підказка. Перехід на крок 6.
 8. Користувач обраховує ймовірність даної в умові завдання події. Якщо відповідь правильна, то кінець роботи алгоритму. Інакше – на крок 9.
 9. Виводиться інформації про помилку та підказка. Перехід на крок 8.
- Для цього алгоритму побудована блок-схема (рис. 3.1).

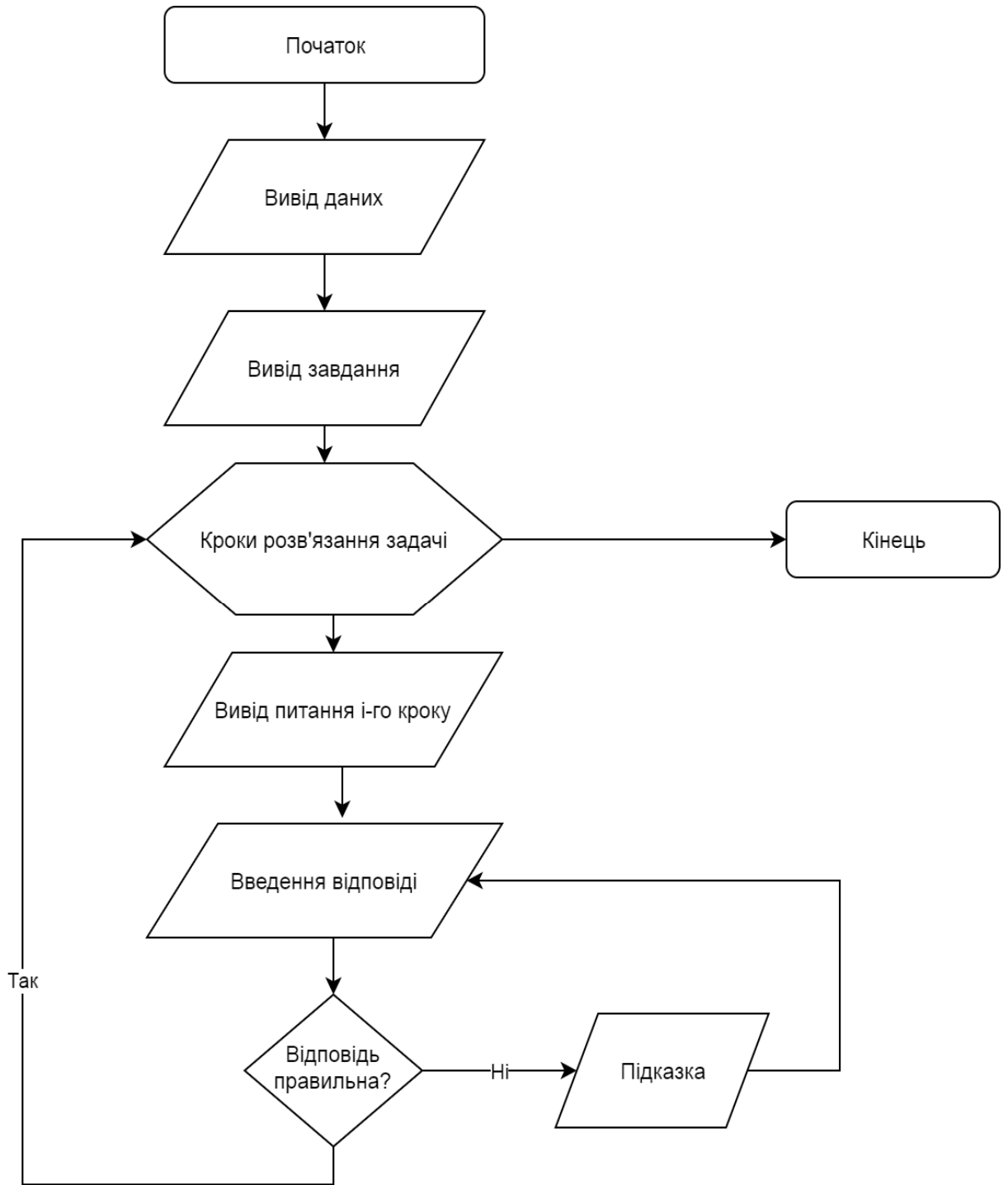


Рис 3.1 Блок-схема алгоритму роботи тренажера

Структурно файл form.fxml також складається із трьох частин. В корені файлу знаходиться елемент BorderPane. В ньому містяться елементи: top, left, center.

В елементі top реалізована верхня частина вікна:

```
<top>
  <VBox spacing="20">
    <padding><Insets top="15" right="15" bottom="15"
left="15"/></padding>
    <Label fx:id="stepInfo" text="Ознайомтеся із даними в лівій
частині вікна. Ви можете відсортувати їх по будь-якому стовпцю.&#13;
В правій частині вікна завдання. Вам потрібно по чергово порахувати
ймовірності. &#13;
Обрахуйте першу ймовірність. Відповідь введіть звичайним дробом,
наприклад, 2/5 та натисніть Enter."/>
  </VBox>
</top>
```

В елементі left виводяться дані. Вся логіка даних реалізована динамічно в java-кодї, тому ця частина має невеликий розмір:

```
<left>
  <TableView fx:id="tableData" editable="false">
    <columns>
      <TableColumn text="Спостереження"></TableColumn>
      <TableColumn text="Температура"></TableColumn>
      <TableColumn text="Вологість"></TableColumn>
      <TableColumn text="Вітер"></TableColumn>
      <TableColumn text="Гра"></TableColumn>
    </columns>
  </TableView>
</left>
```

Елемент center складається із двох частин. В верхній частині виводиться завдання:

```
<center>
  <VBox spacing="10">
    <padding><Insets top="15" right="15" bottom="15"
left="15"/></padding>
    <Label fx:id="questionLabel" text="Визначити, чи відбудеться
гра при наступних умовах (подія E):"/>
    <Label fx:id="var1" text="Спостереження = Сонце"/>
    <Label fx:id="var2" text="Температура = Холодно"/>
  </VBox>
</center>
```

```
<Label fx:id="var3" text="Вологість = Висока"/>
<Label fx:id="var4" text="Вітер = Так"/>
```

В нижній частині виводиться інформація, яка необхідна для розв'язання поставленого завдання:

```
<Label text="Знайдіть наступні ймовірності:"/>
<GridPane>
  <Label fx:id="p11Label"
text="P(Спостереження=Сонце|Гра=Так)=" GridPane.columnIndex="0"
GridPane.rowIndex="0"/>
  <Label fx:id="p12Label"
text="P(Спостереження=Сонце|Гра=Hi)=" GridPane.columnIndex="0"
GridPane.rowIndex="1"/>
  <Label fx:id="p21Label" text="P(Температура =
Холодно|Гра=Так)=" GridPane.columnIndex="0" GridPane.rowIndex="2"/>
  <Label fx:id="p22Label" text="P(Температура =
Холодно|Гра=Hi)=" GridPane.columnIndex="0" GridPane.rowIndex="3"/>
  <Label fx:id="p31Label" text="P(Вологість =
Висока|Гра=Так)=" GridPane.columnIndex="0" GridPane.rowIndex="4"/>
  <Label fx:id="p32Label" text="P(Вологість =
Висока|Гра=Hi)=" GridPane.columnIndex="0" GridPane.rowIndex="5"/>
  <Label fx:id="p41Label" text="P(Вітер = Так|Гра=Так)="
GridPane.columnIndex="0" GridPane.rowIndex="6"/>
  <Label fx:id="p42Label" text="P(Вітер = Так|Гра=Hi)="
GridPane.columnIndex="0" GridPane.rowIndex="7"/>
  <Label text="P(Гра=Так)=" GridPane.columnIndex="0"
GridPane.rowIndex="8"/>
  <Label text="P(Гра=Hi)=" GridPane.columnIndex="0"
GridPane.rowIndex="9"/>
```

Також в цій частині розміщені текстові поля, куди користувач буде вводити інформацію:

```
<TextField fx:id="p11" GridPane.columnIndex="1" GridPane.rowIndex="0"
onAction="#step1"/>
<TextField fx:id="p12" GridPane.columnIndex="1" GridPane.rowIndex="1" disable="true"
onAction="#step2"/>
<TextField fx:id="p21" GridPane.columnIndex="1" GridPane.rowIndex="2" disable="true"
onAction="#step3"/>
<TextField fx:id="p22" GridPane.columnIndex="1" GridPane.rowIndex="3" disable="true"
onAction="#step4"/>
<TextField fx:id="p31" GridPane.columnIndex="1" GridPane.rowIndex="4" disable="true"
onAction="#step5"/>
<TextField fx:id="p32" GridPane.columnIndex="1" GridPane.rowIndex="5" disable="true"
onAction="#step6"/>
<TextField fx:id="p41" GridPane.columnIndex="1" GridPane.rowIndex="6" disable="true"
onAction="#step7"/>
```

```

<TextField fx:id="p42" GridPane.columnIndex="1" GridPane.rowIndex="7" disable="true"
onAction="#step8"/>
<TextField fx:id="p01" GridPane.columnIndex="1" GridPane.rowIndex="8" disable="true"
onAction="#step9"/>
<TextField fx:id="p02" GridPane.columnIndex="1" GridPane.rowIndex="9" disable="true"
onAction="#step10"/>
</GridPane>

```

Потім створені елементи керування для фінальної відповіді:

```

<Label text="Знайдіть чисельники наступних ймовірностей:"/>
<GridPane>
  <Label text="P(Гра=Так|E)=" GridPane.columnIndex="0" GridPane.rowIndex="0"/>
  <Label text="P(Гра=Hi|E)=" GridPane.columnIndex="0" GridPane.rowIndex="1"/>
  <TextField fx:id="p1e" GridPane.columnIndex="1" GridPane.rowIndex="0"
onAction="#step11"/>
  <TextField fx:id="p2e" GridPane.columnIndex="1" GridPane.rowIndex="1"
onAction="#step12"/>
  <Label text="/P(E)" GridPane.columnIndex="2" GridPane.rowIndex="0"/>
  <Label text="/P(E)" GridPane.columnIndex="2" GridPane.rowIndex="1"/>
</GridPane>
<Label text="Знайдіть нормовані значення ймовірностей:"/>
<GridPane>
  <Label text="P(Гра=Так|E)=" GridPane.columnIndex="0" GridPane.rowIndex="0"/>
  <Label text="P(Гра=Hi|E)=" GridPane.columnIndex="0" GridPane.rowIndex="1"/>
  <TextField fx:id="p1" GridPane.columnIndex="1" GridPane.rowIndex="0"
onAction="#step13"/>
  <TextField fx:id="p2" GridPane.columnIndex="1" GridPane.rowIndex="1"
onAction="#step14"/>
</GridPane>
</VBox>
</center>

```

Більшість елементів змінюються динамічно під час виконання програми.

Розглянемо структуру проекту. Окрім файлу form.fxml, проект містить:

- 1) пакет variables;
- 2) клас Data у файлі Data.java;
- 3) клас Fraction у файлі Fraction.java;
- 4) клас Question у файлі Question.java;
- 5) клас RowTable у файлі RowTable.java;
- 6) клас Controller у файлі Controller.java;
- 7) клас Main у файлі Main.java.

Розглянемо кожен із елементів детальніше. Пакет `variables` містить такі файли:

- 1) інтерфейс `Variable`;
- 2) перелік (enum) `Game`;
- 3) перелік `Humidity`;
- 4) перелік `Observation`;
- 5) перелік `Temperature`;
- 6) перелік `Wind`.

Інтерфейс `Variable` реалізують всі наведені вище переліки. Це гарантує, що всі переліки реалізують потрібні методи. Код інтерфейсу:

```
public interface Variable {
    String getMeaning();
    String getObservation();
}
```

Кожен перелік описує одну із змінних з набору даних. В переліку вказується набір можливих значень відповідної змінної. Наприклад, код переліку `Temperature` має такий вигляд:

```
public enum Temperature implements Variable {
    HEAT("Спека"), NORM("Норма"), COLD("Холодно");
    private String meaning;
    private String observation = "Температура";
    Temperature(String meaning) {
        this.meaning = meaning;
    }
    @Override
    public String getMeaning() {
        return meaning;
    }
    @Override
    public String getObservation() {
        return observation;
    }
}
```

Код інших переліків аналогічний і наведений в додатку.

Клас `Data` описує дані, для яких застосовується метод `Naive Bayes`. Клас містить 2 поля:

- private Variable[][] table – двовимірний масив, у якому зберігаються дані;
- private int countVar – кількість змінних у даних.

В конструкторі дані заносяться в масив. Всього є 3 різних набори даних, один із них вибирається відповідно до номеру варіанту. Номер варіанту передається конструктору як параметр. Нижче наведений частковий код конструктора:

```
public Data(int var) {
    if(var==0) {
        table = new Variable[countVar][];
        Observation[] observation = {
            Observation.SUN,
            Observation.SUN,
            Observation.CLOUDY,
//...
        };
        Temperature[] temperature = {
            Temperature.HEAT,
            Temperature.HEAT,
            Temperature.HEAT,
//...
        };
        Humidity[] humidity = {
            Humidity.HIGH,
            Humidity.HIGH,
            Humidity.HIGH,
//...
        };
        Wind[] wind = {
            Wind.NO,
            Wind.YES,
            Wind.NO,
//...
        };
        Game[] game = {
            Game.NO,
            Game.NO,
            Game.YES,
//...
        };
        table[0] = observation;
        table[1] = temperature;
        table[2] = humidity;
        table[3] = wind;
        table[4] = game;
//...
    }
}
```

Також клас Data має 2 методи, які повертають відповідні поля:

```

public Variable[][] getTable() {
    return table;
}

public int getCountVar() {
    return this.countVar;
}

```

Ще 2 методи обраховують ймовірності. Перший метод обраховує умовну ймовірність певного погодного явища в залежності від того, відбулася гра чи ні:

```

public Fraction probability(Variable variable, Game game){
    int num=0;
    int denom=0;
    int line=-1;
    if(variable instanceof Observation){
        line=0;
    } else if(variable instanceof Temperature){
        line=1;
    } else if(variable instanceof Humidity){
        line=2;
    } else if(variable instanceof Wind){
        line=3;
    }
    for(int i=0;i<table[line].length;i++){
        if(table[4][i]==game){
            denom++;
            if(table[line][i]==variable){
                num++;
            }
        }
    }
    return new Fraction(num,denom);
}

```

Другий метод обраховує ймовірність того, що гра відбулася/не відбулася:

```

public Fraction probabGame(Game game){
    int num=0;
    int denom=table[4].length;
    for(int i=0;i<table[4].length;i++){
        if(table[4][i]==game){
            num++;
        }
    }
    return new Fraction(num,denom);
}

```

Як бачимо, обидва ці методи повертають тип `Fraction`. Цей тип описується за допомогою однойменного класу. Цей клас дозволяє працювати з ймовірностями, як із звичайними дробами. Клас `Fraction` має 2 поля:

- `private int numerator` – чисельник дробу;
- `private int denominator` – знаменник дробу.

Клас також містить конструктор та 3 методи. Перший метод `equals()` дозволяє порівняти дробу – рівні вони чи ні:

```
@Override
public boolean equals(Object obj) {
    Fraction fraction=(Fraction)obj;
    return fraction.numerator*denominator ==
fraction.denominator*numerator;
}
```

Другий метод повертає дріб у текстовому вигляді:

```
@Override
public String toString() {
    return numerator + "/" + denominator;
}
```

Третій метод повертає дріб у вигляді десяткового дробу, тобто тип `double`:

```
public double getValue() {
    return (double)numerator/denominator;
}
```

Клас `Question` описує конкретну задачу, яку має розв'язати користувач. Також цей клас обраховує усі ймовірності, які вводить користувач при розв'язанні задачі. Ці значення використовуються для перевірки того, чи правильно введена відповідь на певному кроці.

Задача формується випадковим чином. Змінній `Спостереження` привласнюється одне із трьох значень: Сонце, Хмарно, Дощ. Причому значення вибирається випадковим чином з однаковою ймовірністю. Змінній `Температура`

привласнюється одне із трьох значень: Спека, Норма, Холодно. Змінній Вологість – одне із двох значень: Висока, Норма. Змінній Вітер також одне із двох значень: Так, Ні.

Тобто, може бути сформована одна із 36 задач ($3*3*2*2=36$). Враховуючи, що ця задача розв'язується для одного із трьох наборів даних, маємо 108 ($36*3=108$) різних постановок задач. Одну із таких постановок можна побачити на рис. 4.1.

Генерація задачі реалізована в конструкторі класу Question:

```
public Question() {
    double rnd=Math.random();
    if(rnd<0.334) {
        observation=Observation.SUN;
    } else if (rnd<0.667) {
        observation=Observation.CLOUDY;
    } else {
        observation=Observation.RAIN;
    }
    rnd=Math.random();
    if(rnd<0.334) {
        temperature=Temperature.HEAT;
    } else if (rnd<0.667) {
        temperature=Temperature.NORM;
    } else {
        temperature=Temperature.COLD;
    }
    rnd=Math.random();
    if(rnd<0.5) {
        humidity=Humidity.HIGH;
    } else {
        humidity=Humidity.NORM;
    }
    rnd=Math.random();
    if(rnd<0.5) {
        wind=Wind.YES;
    } else {
        wind=Wind.NO;
    }
}
```

Задача зберігається в полях:

private Observation observation – спостереження;

private Temperature temperature – температура;

private Humidity humidity – вологість;

private Wind wind – вітер.

Обраховані ймовірності зберігаються у наступних полях:

private Fraction [] fractions=new Fraction[10] – зберігаються умовні ймовірності, які обраховуються на перших десяти кроках тренажеру;

private double pYesE – чисельник ймовірності, що гра відбудеться;

private double pNoE – чисельник ймовірності, що гра не відбудеться;

private double pYes – нормована ймовірність, що гра відбудеться;

private double pNo – нормована ймовірність, що гра не відбудеться.

В методі public void calcFractions(Data data) відбувається обрахунок цих ймовірностей. Повний код класу – в додатку.

Клас RowTable призначений для виведення даних. Його код також наведений в додатку.

В класі Controller визначається логіка роботи програми та її взаємодія з користувачем.

Метод initialize() запускається на початку роботи тренажеру. В цьому методі спочатку генерується номер варіанту і дані:

```
int var=(int) (Math.random()*2.999);
this.myData =new Data(var);
```

Після цього дані виводяться користувачу:

```
data= FXCollections.observableArrayList();
TableColumn[] tableCol=new TableColumn[myData.getCountVar()];
for(int i = 0; i< myData.getCountVar(); i++) {
    tableCol[i] = tableData.getColumns().get(i);
    tableCol[i].setText(myData.getTable()[i][0].getObservation());
}
tableCol[0].setCellValueFactory(new
PropertyValueFactory<String,RowTable>("col1"));
tableCol[1].setCellValueFactory(new
PropertyValueFactory<String,RowTable>("col2"));
tableCol[2].setCellValueFactory(new
PropertyValueFactory<String,RowTable>("col3"));
tableCol[3].setCellValueFactory(new
PropertyValueFactory<String,RowTable>("col4"));
tableCol[4].setCellValueFactory(new
PropertyValueFactory<String,RowTable>("col5"));
for(int j = 0; j< myData.getTable()[0].length; j++) {
```

```

    data.add(new RowTable(myData.getTable()[0][j].getMeaning(),
myData.getTable()[1][j].getMeaning(),
        myData.getTable()[2][j].getMeaning(),
myData.getTable()[3][j].getMeaning(),
myData.getTable()[4][j].getMeaning()));
}
tableData.setItems(data);

```

Потім генерується задача і обраховуються ймовірності, які необхідні для її розв'язку:

```

this.question=new Question();
question.calcFractions(this.myData);

```

Після цього заповнюється права частина вікна – виводиться задача та кроки, які необхідно пройти, щоб її розв'язати:

```

var1.setText(question.getObservation().getObservation()+"="+question.g
etObservation().getMeaning());
var2.setText(question.getTemperature().getObservation()+"="+question.g
etTemperature().getMeaning());
var3.setText(question.getHumidity().getObservation()+"="+question.getH
umidity().getMeaning());
var4.setText(question.getWind().getObservation()+"="+question.getWind(
).getMeaning());
p11Label.setText("P("+question.getObservation().getObservation()+"="+q
uestion.getObservation().getMeaning()+" |Гра=Так)");
p12Label.setText("P("+question.getObservation().getObservation()+"="+q
uestion.getObservation().getMeaning()+" |Гра=Hi)");
p21Label.setText("P("+question.getTemperature().getObservation()+"="+q
uestion.getTemperature().getMeaning()+" |Гра=Так)");
p22Label.setText("P("+question.getTemperature().getObservation()+"="+q
uestion.getTemperature().getMeaning()+" |Гра=Hi)");
p31Label.setText("P("+question.getHumidity().getObservation()+"="+ques
tion.getHumidity().getMeaning()+" |Гра=Так)");
p32Label.setText("P("+question.getHumidity().getObservation()+"="+ques
tion.getHumidity().getMeaning()+" |Гра=Hi)");
p41Label.setText("P("+question.getWind().getObservation()+"="+question
.getWind().getMeaning()+" |Гра=Так)");
p42Label.setText("P("+question.getWind().getObservation()+"="+question
.getWind().getMeaning()+" |Гра=Hi)");

```

Метод `parseToFraction()` перетворює введений текст в дріб (об'єкт класу `Fraction`). Дріб необхідно вводити у форматі "чисельник/знаменник". Наприклад, "2/5". Виключенням є числа 0 і 1. Їх можна вводити без риски дроби, хоча варіанти "0/5" або "6/6" також будуть коректні.

```

private Fraction parseToFraction(String text){
    String [] s=text.split("/");
    Fraction fraction=null;
    if(s.length==1){
        if(s[0].equals("0")){
            return new Fraction(0,1);
        } else if(s[0].equals("1")){
            return new Fraction(1,1);
        } else{
            msgError();
            return fraction;
        }
    }
    if(s.length!=2){
        msgError();
        return fraction;
    }
    try{
        int num=Integer.parseInt(s[0]);
        int denom=Integer.parseInt(s[1]);
        fraction=new Fraction(num,denom);
    }catch (Exception e){
        msgError();
    }
    return fraction;
}

```

Також контролер містить методи, які викликаються при введенні користувачем відповіді на певному кроці. Наприклад:

```

public void step3(ActionEvent actionEvent) {
    if(p21.isEditable()){
        Fraction fraction=parseToFraction(p21.getText());
        if(fraction!=null){
            if(fraction.equals(question.getFractions(stepNumber-1))){
                p22.setDisable(false);
                p22.requestFocus();
                p21.setEditable(false);
                errors=0;
                stepNumber++;
            }else{
                msgWrongAns();
            }
        }
    }
}

```

Також в контролері є методи, які виводять повідомлення про неправильну відповідь. Наприклад:

```
private void msgWrongAns () {
    errors++;
    if(errors==1) {
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Неправильна відповідь");
        alert.setHeaderText("Неправильна відповідь");
        alert.setContentText("Спробуйте ще раз");
        alert.showAndWait();
    } else{
        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Неправильна відповідь");
        alert.setHeaderText("Неправильна відповідь");
        alert.setContentText("Правильна відповідь:
"+question.getFractions(stepNumber-1));
        alert.showAndWait();
    }
}
```

Клас Main є головним класом проекту. Він визначає розміри екрану та заголовок. Також містить стандартний код для JavaFx-проектів. Код класу:

```
public class Main extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception{
        Parent root =
FXMLLoader.load(getClass().getResource("form.fxml"));
        primaryStage.setTitle("Тренажер Naive Bayes");
        primaryStage.setScene(new Scene(root, 800, 750));
        primaryStage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

Отже, ми розглянули реалізацію тренажеру з теми «Класифікатор Naive Bayes» дисципліни «Комп'ютерний аналіз статистичних даних».

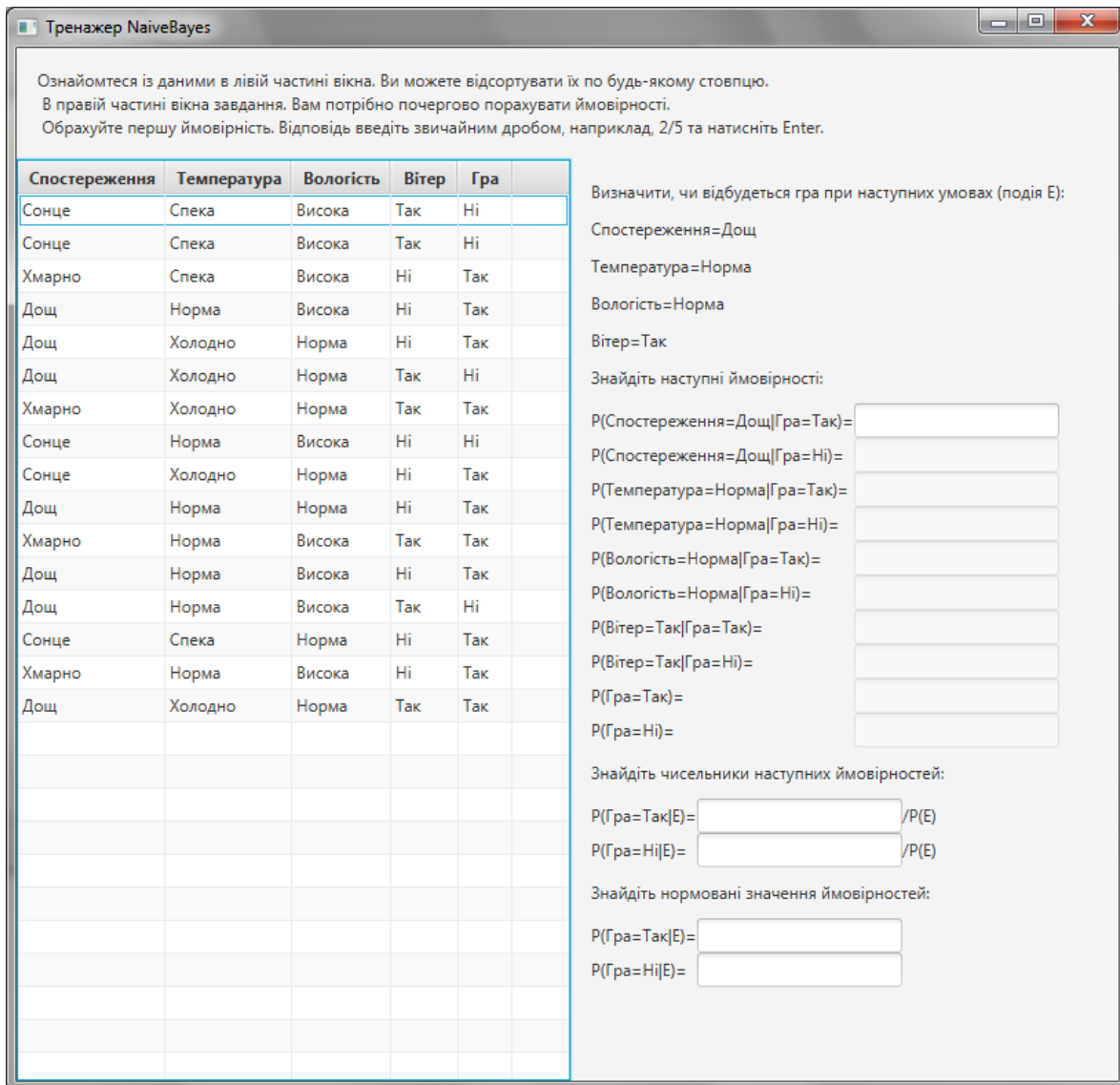


Рис. 4.3. Вікно тренажеру після запуску з іншим завданням

Для перевірки правильності роботи тренажеру, розв'яжемо покрокову одну із задач (рис. 4.4). Для зручності підрахунку дані можна відсортувати по останньому стовпцю.

На першому кроці потрібно порахувати умовну ймовірність

$$P(\text{Спостереження}=\text{Дощ} | \text{Гра}=\text{Так})$$

Всього є 11 записів, коли гра відбулася. В 4-х із них був дощ. Тому правильна відповідь: 4/11.

Після введення неправильної відповіді (4/12) виводиться повідомлення про помилку (рис. 4.5).

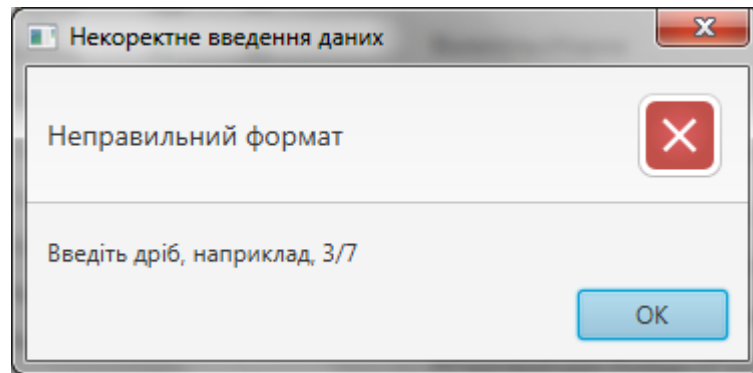


Рис. 4.6. Повідомлення про неправильний формат відповіді

Якщо повторно ввести неправильну відповідь, то з'являється повідомлення із правильною відповіддю:

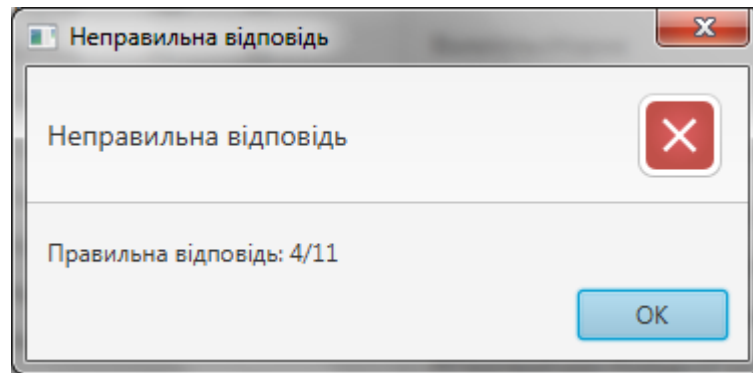


Рис. 4.7. Діалогове вікно з правильною відповіддю

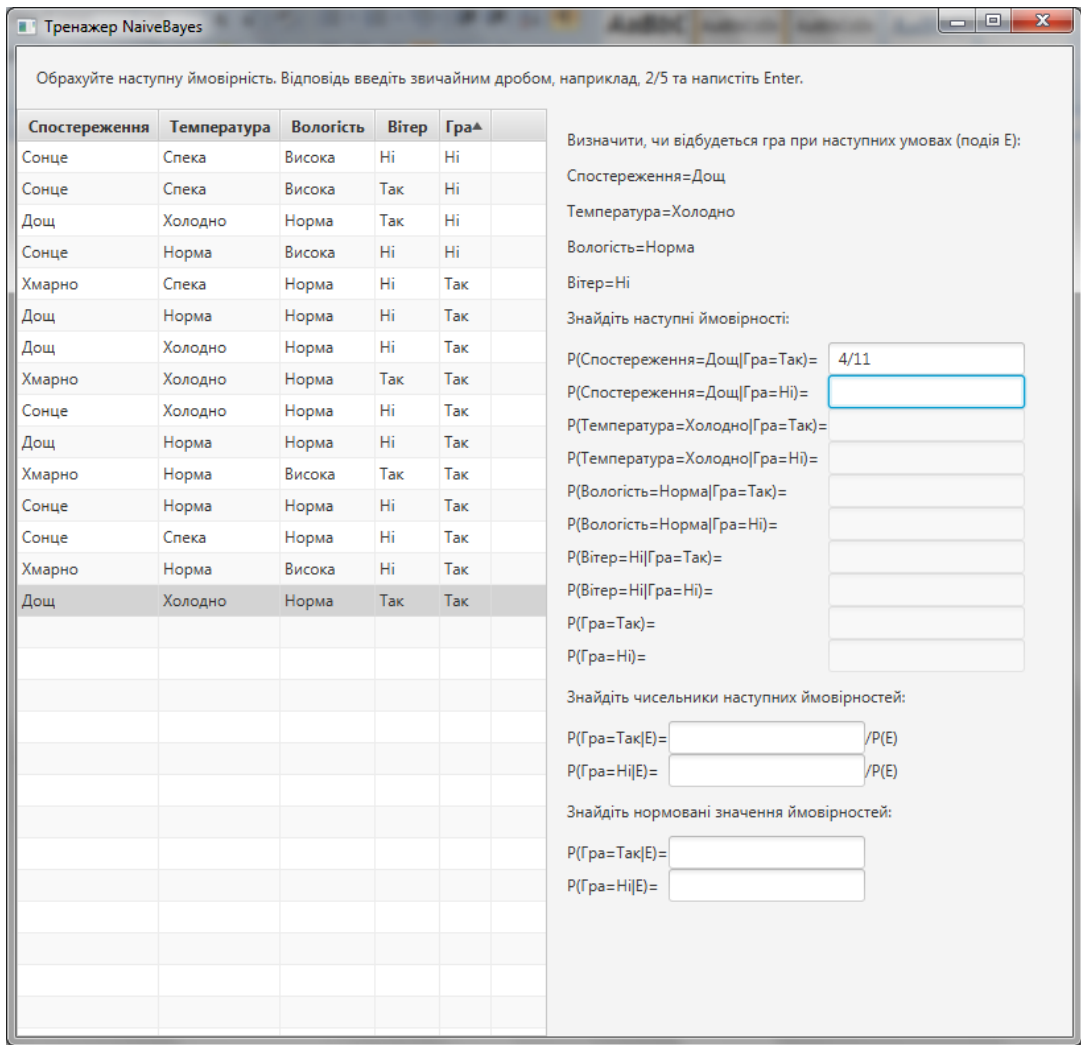
Після введення правильної відповіді (4/11), відбувається перехід до наступного кроку. При цьому правильне значення вноситься у відповідну комірку, комірка стає недоступною для редагування, а наступна комірка стає активною.

На цьому кроці потрібно порахувати умовну ймовірність

$$P(\text{Спостереження}=\text{Дощ} \mid \text{Гра}=\text{Ні})$$

Всього є 4 записи, коли не було гри. Лише в одному із цих випадків був дощ. тому правильна відповідь: 1/4.

Після введення правильної відповіді відбувається перехід на наступний крок (рис. 4.9).



4.8. Вигляд тренажера після правильної відповіді на перше запитання

На третьому кроці (рис. 4.9) потрібно порахувати умовну ймовірність $P(\text{Температура}=\text{Холодно} \mid \text{Гра}=\text{Так})$

Як було визначено раніше, всього є 11 записів, коли гра відбулася. В 4-х із цих випадків було холодно (другий стовпець із таблиці з даними).

Тому правильна відповідь 4/11.

На наступному кроці потрібно порахувати умовну ймовірність $P(\text{Температура}=\text{Холодно} \mid \text{Гра}=\text{Ні})$

Всього є 4 випадки, коли гра не відбулася. В одному із них було холодно. Тому правильна відповідь 1/4.

Вигляд тренажера після цих двох кроків зображений на рис. 4.9.

Тренажер NaiveBayes

Обрахуйте наступну ймовірність. Відповідь введіть звичайним дробом, наприклад, 2/5 та натисність Enter.

Спостереження	Температура	Вологість	Вітер	Гра
Сонце	Спека	Висока	Ні	Ні
Сонце	Спека	Висока	Так	Ні
Дощ	Холодно	Норма	Так	Ні
Сонце	Норма	Висока	Ні	Ні
Хмарно	Спека	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Дощ	Холодно	Норма	Ні	Так
Хмарно	Холодно	Норма	Так	Так
Сонце	Холодно	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Хмарно	Норма	Висока	Так	Так
Сонце	Норма	Норма	Ні	Так
Сонце	Спека	Норма	Ні	Так
Хмарно	Норма	Висока	Ні	Так
Дощ	Холодно	Норма	Так	Так

Визначити, чи відбудеться гра при наступних умовах (подія E):
Спостереження=Дощ
Температура=Холодно
Вологість=Норма
Вітер=Ні

Знайдіть наступні ймовірності:

$P(\text{Спостереження}=\text{Дощ}|\text{Гра}=\text{Так})=$

$P(\text{Спостереження}=\text{Дощ}|\text{Гра}=\text{Ні})=$

$P(\text{Температура}=\text{Холодно}|\text{Гра}=\text{Так})=$

$P(\text{Температура}=\text{Холодно}|\text{Гра}=\text{Ні})=$

$P(\text{Вологість}=\text{Норма}|\text{Гра}=\text{Так})=$

$P(\text{Вологість}=\text{Норма}|\text{Гра}=\text{Ні})=$

$P(\text{Вітер}=\text{Ні}|\text{Гра}=\text{Так})=$

$P(\text{Вітер}=\text{Ні}|\text{Гра}=\text{Ні})=$

$P(\text{Гра}=\text{Так})=$

$P(\text{Гра}=\text{Ні})=$

Знайдіть чисельники наступних ймовірностей:

$P(\text{Гра}=\text{Так}|E)=$ / $P(E)$

$P(\text{Гра}=\text{Ні}|E)=$ / $P(E)$

Знайдіть нормовані значення ймовірностей:

$P(\text{Гра}=\text{Так}|E)=$

$P(\text{Гра}=\text{Ні}|E)=$

4.9. П'ятий крок тренажеру

На цьому кроці (рис. 4.10) потрібно порахувати умовну ймовірність $P(\text{Вітер}=\text{Ні} | \text{Гра}=\text{Так})$

Легко порахувати, що було 8 випадків, коли був вітер і гра відбулася. Тому правильна відповідь 8/11.

Тренажер NaiveBayes

Обрахуйте наступну ймовірність. Відповідь введіть звичайним дробом, наприклад, 2/5 та натисніть Enter.

Спостереження	Температура	Вологість	Вітер	Гра▲
Сонце	Спека	Висока	Ні	Ні
Сонце	Спека	Висока	Так	Ні
Дощ	Холодно	Норма	Так	Ні
Сонце	Норма	Висока	Ні	Ні
Хмарно	Спека	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Дощ	Холодно	Норма	Ні	Так
Хмарно	Холодно	Норма	Так	Так
Сонце	Холодно	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Хмарно	Норма	Висока	Так	Так
Сонце	Норма	Норма	Ні	Так
Сонце	Спека	Норма	Ні	Так
Хмарно	Норма	Висока	Ні	Так
Дощ	Холодно	Норма	Так	Так

Визначити, чи відбудеться гра при наступних умовах (подія E):
Спостереження=Дощ
Температура=Холодно
Вологість=Норма
Вітер=Ні

Знайдіть наступні ймовірності:

$P(\text{Спостереження}=\text{Дощ}|\text{Гра}=\text{Так})=$

$P(\text{Спостереження}=\text{Дощ}|\text{Гра}=\text{Ні})=$

$P(\text{Температура}=\text{Холодно}|\text{Гра}=\text{Так})=$

$P(\text{Температура}=\text{Холодно}|\text{Гра}=\text{Ні})=$

$P(\text{Вологість}=\text{Норма}|\text{Гра}=\text{Так})=$

$P(\text{Вологість}=\text{Норма}|\text{Гра}=\text{Ні})=$

$P(\text{Вітер}=\text{Ні}|\text{Гра}=\text{Так})=$

$P(\text{Вітер}=\text{Ні}|\text{Гра}=\text{Ні})=$

$P(\text{Гра}=\text{Так})=$

$P(\text{Гра}=\text{Ні})=$

Знайдіть чисельники наступних ймовірностей:

$P(\text{Гра}=\text{Так}|E)=$ / $P(E)$

$P(\text{Гра}=\text{Ні}|E)=$ / $P(E)$

Знайдіть нормовані значення ймовірностей:

$P(\text{Гра}=\text{Так}|E)=$

$P(\text{Гра}=\text{Ні}|E)=$

4.10. Сьомий крок тренажеру

На восьмому кроці потрібно обрахувати умовну ймовірність

$$P(\text{Вітер}=\text{Ні} \mid \text{Гра}=\text{Ні})$$

Із чотирьох випадків коли не було гри у двох не було вітру. Формально можна записати 2/4, але математично правильніше можна записати 1/2. Тренажер приймає за правильну відповідь будь який із варіантів.

Варто зауважити, що інші еквівалентні дроби (3/6, 4/8 і т.п.) теж будуть прийматися як правильні відповіді, оскільки фактично, вони і є такими.

У випадку введення неправильної відповіді 2 рази, виводиться повідомлення із підказкою (рис. 4.12). В цій в підказці правильна відповідь записана у форматі 2/4, оскільки студенту більш зрозуміло, як отримати таке число.

Тренажер NaiveBayes

Обрахуйте наступну ймовірність. Відповідь введіть звичайним дробом, наприклад, 2/5 та напістіть Enter.

Спостереження	Температура	Вологість	Вітер	Гра
Сонце	Спека	Висока	Ні	Ні
Сонце	Спека	Висока	Так	Ні
Дощ	Холодно	Норма	Так	Ні
Сонце	Норма	Висока	Ні	Ні
Хмарно	Спека	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Дощ	Холодно	Норма	Ні	Так
Хмарно	Холодно	Норма	Так	Так
Сонце	Холодно	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Хмарно	Норма	Висока	Так	Так
Сонце	Норма	Норма	Ні	Так
Сонце	Спека	Норма	Ні	Так
Хмарно	Норма	Висока	Ні	Так
Дощ	Холодно	Норма	Так	Так

Визначити, чи відбудеться гра при наступних умовах (подія E):
Спостереження=Дощ
Температура=Холодно
Вологість=Норма
Вітер=Ні

Знайдіть наступні ймовірності:

$P(\text{Спостереження}=\text{Дощ}|\text{Гра}=\text{Так})=$

$P(\text{Спостереження}=\text{Дощ}|\text{Гра}=\text{Ні})=$

$P(\text{Температура}=\text{Холодно}|\text{Гра}=\text{Так})=$

$P(\text{Температура}=\text{Холодно}|\text{Гра}=\text{Ні})=$

$P(\text{Вологість}=\text{Норма}|\text{Гра}=\text{Так})=$

$P(\text{Вологість}=\text{Норма}|\text{Гра}=\text{Ні})=$

$P(\text{Вітер}=\text{Ні}|\text{Гра}=\text{Так})=$

$P(\text{Вітер}=\text{Ні}|\text{Гра}=\text{Ні})=$

$P(\text{Гра}=\text{Так})=$

$P(\text{Гра}=\text{Ні})=$

Знайдіть чисельники наступних ймовірностей:

$P(\text{Гра}=\text{Так}|E)=$ / $P(E)$

$P(\text{Гра}=\text{Ні}|E)=$ / $P(E)$

Знайдіть нормовані значення ймовірностей:

$P(\text{Гра}=\text{Так}|E)=$

$P(\text{Гра}=\text{Ні}|E)=$

4.11. Восьмий крок тренажеру

Неправильна відповідь

Неправильна відповідь

Правильна відповідь: 2/4

OK

4.12. Підказка із правильною відповіддю на восьмому кроці

Після обрахунку усіх умовних ймовірностей потрібно визначити ймовірність того, що гра відбулася.

Всього у нас 15 записів. В 11 із них гра відбулася, тому правильна відповідь: 11/15.

Тренажер NaiveBayes

Обрахуйте наступну ймовірність. Відповідь введіть звичайним дробом, наприклад, 2/5 та натисність Enter.

Спостереження	Температура	Вологість	Вітер	Гра▲
Сонце	Спека	Висока	Ні	Ні
Сонце	Спека	Висока	Так	Ні
Дощ	Холодно	Норма	Так	Ні
Сонце	Норма	Висока	Ні	Ні
Хмарно	Спека	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Дощ	Холодно	Норма	Ні	Так
Хмарно	Холодно	Норма	Так	Так
Сонце	Холодно	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Хмарно	Норма	Висока	Так	Так
Сонце	Норма	Норма	Ні	Так
Сонце	Спека	Норма	Ні	Так
Хмарно	Норма	Висока	Ні	Так
Дощ	Холодно	Норма	Так	Так

Визначити, чи відбудеться гра при наступних умовах (подія E):

Спостереження=Дощ
Температура=Холодно
Вологість=Норма
Вітер=Ні

Знайдіть наступні ймовірності:

$P(\text{Спостереження}=\text{Дощ}|\text{Гра}=\text{Так})=$
 $P(\text{Спостереження}=\text{Дощ}|\text{Гра}=\text{Ні})=$
 $P(\text{Температура}=\text{Холодно}|\text{Гра}=\text{Так})=$
 $P(\text{Температура}=\text{Холодно}|\text{Гра}=\text{Ні})=$
 $P(\text{Вологість}=\text{Норма}|\text{Гра}=\text{Так})=$
 $P(\text{Вологість}=\text{Норма}|\text{Гра}=\text{Ні})=$
 $P(\text{Вітер}=\text{Ні}|\text{Гра}=\text{Так})=$
 $P(\text{Вітер}=\text{Ні}|\text{Гра}=\text{Ні})=$
 $P(\text{Гра}=\text{Так})=$
 $P(\text{Гра}=\text{Ні})=$

Знайдіть чисельники наступних ймовірностей:

$P(\text{Гра}=\text{Так}|E)=$ / $P(E)$
 $P(\text{Гра}=\text{Ні}|E)=$ / $P(E)$

Знайдіть нормовані значення ймовірностей:

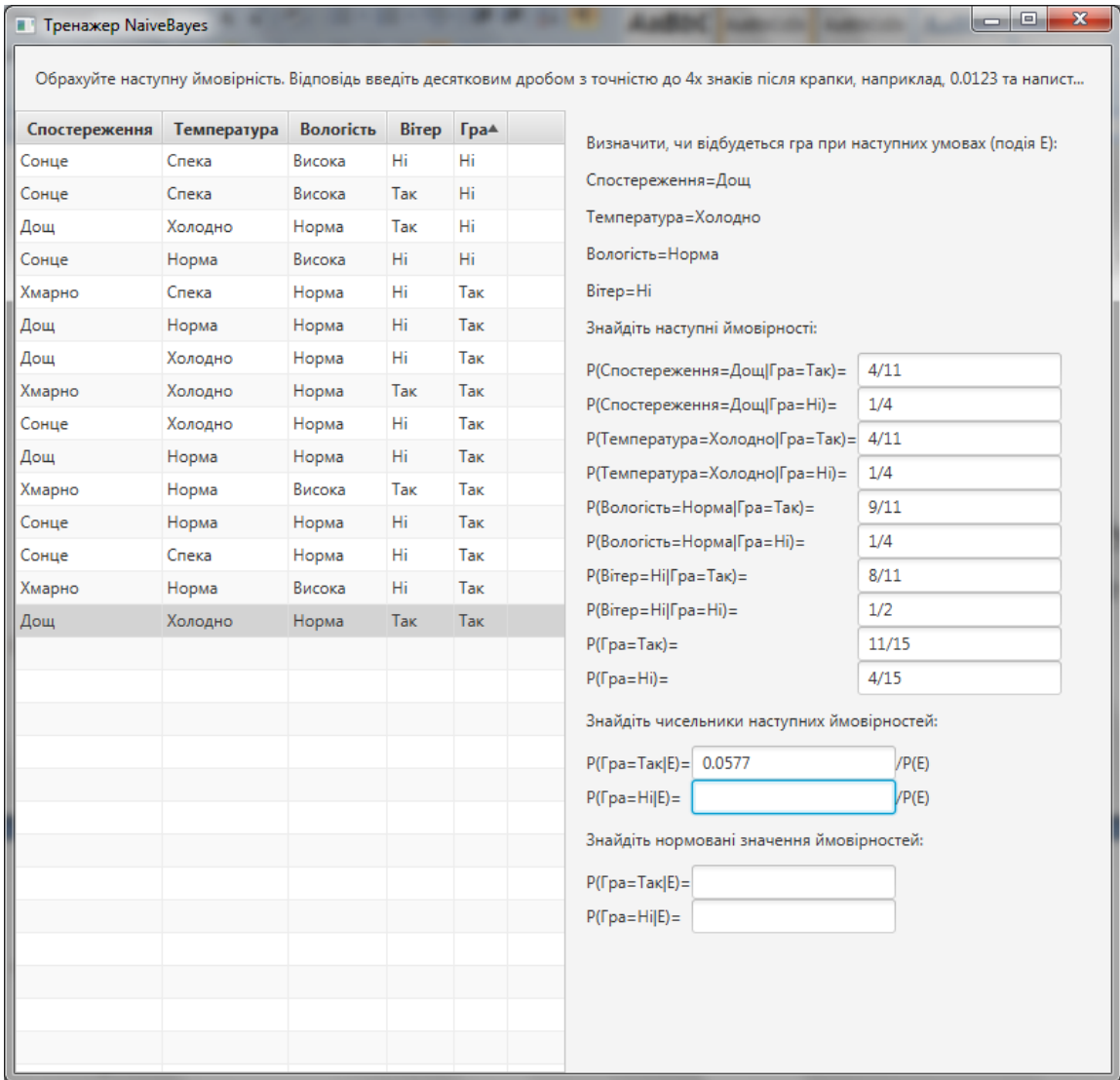
$P(\text{Гра}=\text{Так}|E)=$
 $P(\text{Гра}=\text{Ні}|E)=$

4.13. Десятий крок тренажеру

Після введення правильної відповіді відбувається перехід на наступний крок (рис. 4.13). На цьому кроці необхідно обрахувати ймовірність того, що гра не відбулася. Із 15 записів гра не відбулася у 4-х. Тому правильна відповідь 4/15.

Ввівши правильну відповідь, переходимо до наступного кроку роботи тренажеру (рис. 4.14). На цьому кроці потрібно порахувати чисельник умовної ймовірності

$$P(\text{Гра}=\text{Так}|E).$$



4.15. Дванадцятий крок тренажера

На цьому кроці потрібно порахувати чисельник умовної ймовірності $P(\text{Гра}=\text{Ні}|E)$.

Для обрахунку цієї ймовірності потрібно перемножити наступні ймовірності:
 $1/4 * 1/4 * 1/4 * 1/2 * 4/15 = 0.002$

Якщо ввести відповідь неправильно двічі, то отримаємо підказку (рис.4.16).

Як бачимо, ми отримали результати:

$$P(\text{Гра}=\text{Так}|E) = 0.0577/P(E)$$

$$P(\text{Гра}=\text{Ні}|E) = 0.002/P(E)$$

Використовуючи ці два значення, можна порахувати шукані ймовірності без знаходження значення $P(E)$:

$$P(\text{Гра}=\text{Так}|E) = 0.0577 / (0.0577 + 0.002) = 0.9652$$

$$P(\text{Гра}=\text{Ні}|E) = 0.002 / (0.0577 + 0.002) = 0.0348$$

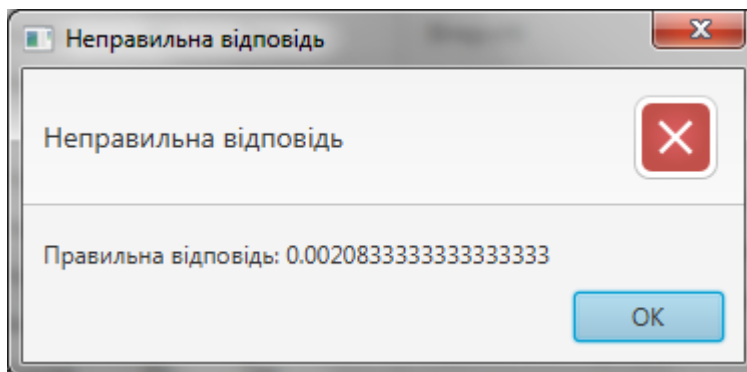


Рис. 4.16. Підказка на дванадцятому кроці

Ви успішно пройшли тренажер!

Спостереження	Температура	Вологість	Вітер	Гра▲
Сонце	Спека	Висока	Ні	Ні
Сонце	Спека	Висока	Так	Ні
Дощ	Холодно	Норма	Так	Ні
Сонце	Норма	Висока	Ні	Ні
Хмарно	Спека	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Дощ	Холодно	Норма	Ні	Так
Хмарно	Холодно	Норма	Так	Так
Сонце	Холодно	Норма	Ні	Так
Дощ	Норма	Норма	Ні	Так
Хмарно	Норма	Висока	Так	Так
Сонце	Норма	Норма	Ні	Так
Сонце	Спека	Норма	Ні	Так
Хмарно	Норма	Висока	Ні	Так
Дощ	Холодно	Норма	Так	Так

Визначити, чи відбудеться гра при наступних умовах (подія E):
Спостереження=Дощ
Температура=Холодно
Вологість=Норма
Вітер=Ні

Знайдіть наступні ймовірності:

$P(\text{Спостереження}=\text{Дощ}|\text{Гра}=\text{Так}) =$

$P(\text{Спостереження}=\text{Дощ}|\text{Гра}=\text{Ні}) =$

$P(\text{Температура}=\text{Холодно}|\text{Гра}=\text{Так}) =$

$P(\text{Температура}=\text{Холодно}|\text{Гра}=\text{Ні}) =$

$P(\text{Вологість}=\text{Норма}|\text{Гра}=\text{Так}) =$

$P(\text{Вологість}=\text{Норма}|\text{Гра}=\text{Ні}) =$

$P(\text{Вітер}=\text{Ні}|\text{Гра}=\text{Так}) =$

$P(\text{Вітер}=\text{Ні}|\text{Гра}=\text{Ні}) =$

$P(\text{Гра}=\text{Так}) =$

$P(\text{Гра}=\text{Ні}) =$

Знайдіть чисельники наступних ймовірностей:

$P(\text{Гра}=\text{Так}|E) =$ $/P(E)$

$P(\text{Гра}=\text{Ні}|E) =$ $/P(E)$

Знайдіть нормовані значення ймовірностей:

$P(\text{Гра}=\text{Так}|E) =$

$P(\text{Гра}=\text{Ні}|E) =$

Рис. 4.17. Вигляд головного вікна після проходження тренажеру

Ці два значення обраховуються на двох останніх кроках алгоритму.

Після введення цих значень користувач отримує розв'язаний приклад (рис. 4.17). Для проходження тренажеру ще раз потрібно закрити та запустити його знову.

Отже, було проведене тестування роботи тренажеру. Помилки або проблем у роботі не було виявлено.

Робота користувача та тестування роботи спрощене за рахунок того, що в тренажері активним і доступним для введення завжди є лише одне текстове поле.

ВИСНОВКИ

В бакалаврській роботі розглядається розробка тренажеру з теми «Класифікатор Naive Bayes» дисципліни «Комп'ютерний аналіз статистичних даних».

Не дивлячись на те, що тренажери значно допомагають студентам опанувати навчальний матеріал, з теми «Класифікатор Naive Bayes» тренажер до цього ще не був розроблений.

Основними результатами роботи є розробка алгоритму роботи тренажера, його блок-схеми, реалізація та тестування тренажеру.

Робота складається зі вступу, чотирьох частин, висновків, списку літератури та додатку, в якому розміщений початковий код тренажеру.

В першій частині наведена постановка задачі.

В другій частині зроблено інформаційний огляд.

Третій розділ присвячений розробці алгоритму роботи тренажеру, побудові його блок-схеми.

В четвертому розділі розроблено тренажер та проведено його тестування.

Для створення тренажеру вибрана мова програмування Java. Реалізація тренажеру виконувалася за допомогою інтегрованого середовища розробки IntelliJ IDEA.

Основними перевагами створеного тренажеру є:

- ✓ незалежність від апаратної та операційної платформи;
- ✓ можливість інтеграції в дистанційний курс;
- ✓ можливість запускати із комп'ютера користувача.

Результати роботи можуть використовуватися при вивченні студентами дисципліни «Комп'ютерний аналіз статистичних даних».

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ємець О.О. Про розробку тренажерів для дистанційних курсів кафедрою ММСІ ПУЕТ / О.О. Ємець // Інформатика та системні науки (ІСН-2015): матеріали VI Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19–21 берез. 2015 р.). – Полтава: ПУЕТ, 2015. – С. 152-161.
2. Ольховська О. В. Технології підтримки системи дистанційного навчання в Полтавському університеті економіки і торгівлі / О. В. Ольховська, Д. М. Ольховський // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.). – Полтава: ПУЕТ, 2016. – С. 219-221.
3. Чілікіна Т. В. Огляд тренажерів з дисципліни "Математичний аналіз" на прикладі розробок студентів напряму "Інформатика" / Т. В. Чілікіна // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.). – Полтава: ПУЕТ, 2016. – С. 329-330.
4. Парфьонова Т. О. Про розробку тренажерів для дистанційного навчального курсу "Алгебра і геометрія" / Т. О. Парфьонова // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.) – Полтава: ПУЕТ, 2016.
5. Марченко Д. А. Алгоритмізація та програмна реалізація тренажера з теми «Дослідження на збіжність числових рядів» дистанційного навчального курсу «Математичний аналіз» / Д. А. Марченко // Інформатика та системні науки (ІСН-2015): матеріали VI Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 19–21 берез. 2015 р.) – Полтава: ПУЕТ, 2015. (<http://dspace.puet.edu.ua/handle/123456789/2500>)
6. Мандрика В. М. Тренажер з теми «1-R алгоритм» дисципліни «Комп'ютерний аналіз статистичних даних» / В. М. Мандрика, Ю. Ф. Олексійчук // Комп'ютерні науки і прикладна математика (КНіПМ-2018): матеріали науково-

- практичного семінару. Випуск 1 – Полтава: Кафедра ММСІ ПУЕТ, 2018. – С. 27-30.
7. Мандрика В. М. Розробка тренажеру з теми «1-R алгоритм» дисципліни «Комп'ютерний аналіз статистичних даних» / В. М. Мандрика, Ю. Ф. Олексійчук // Комп'ютерні науки і прикладна математика (КНіПМ-2018): матеріали науково-практичного семінару. Випуск 2 – Полтава: Кафедра ММСІ ПУЕТ, 2018. – С. 10-13.
 8. Барсегян А. А. Методы и модели анализа данных: OLAP и DataMining. / А. А. Барсегян, М. С. Куприянов, В. В. Степаненко, И. И. Холод – СПб.: БХВ-Петербург, 2004. – 336 с.
 9. Тюрин Ю.Н., Макаров А.А. Анализ данных на компьютере / Под ред. В. Э. Фигурнова. – М.: ИНФРА-М, Финансы и статистика, 1995. – 384 с.
 10. Айвазян С. А., Енюков И. С., Мешалкин Л.Д. Прикладная статистика: исследование зависимостей: Справ.изд. – М.: Финансы и статистика, 1985. – 487 с.
 11. Берк К., Кэйри П. Анализ данных с помощью Microsoft Excel.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 560 с.
 12. Эккель Б. Философия Java. Библиотека программиста. 4-е изд./Б. Эккель — СПб: Питер, 2009. — 640 с.
 13. Шилдт Г. Полный справочник по Java, 7-е издание./ Г. Шилдт — К.: Издательский дом "Вильямс", 2007. — 1034 с.
 14. Машнин Т. С. Java FX 2.0: разработка RIA-приложений / Т. С. Машнин — СПб.: БХВ-Петербург, 2012. — 320 с.
 15. Монахов В. В. Язык программирования Java и среда NetBeans /В. В. Монахов — 3-е издание. — СПб.: «БХВ-Петербург», 2011. — 704 с.

ДОДАТОК А. КОД РОЗРОБЛЕНОГО ТРЕНАЖЕРА