НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет прикладної математики

Кафедра прикладної математики

«До захисту допущено»

Завідувач кафедри

_____ О. Р. Чертов

«___» _____ 2016 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.040301 «Прикладна математика»

на тему: Математичне та програмне забезпечення перетворення числових даних для використання в логіко-лінгвістичних моделях

Виконав: студент IV курсу, групи КМ-23

Крівенцев Максим Олексійович _____

Керівник канд. техн. наук, _____

доцент Сирота С. В.

Консультант із нормоконтролю старший викладач _____

Мальчиков В. В.

Рецензент канд. техн. наук _____

доцент Тимощук О. Л.

Засвідчую, що в цій дипломній роботі немає запозичень із праць інших авторів без відповідних посилань.
Студент _____

Київ — 2016

Національний технічний університет україни

«Київський політехнічний інститут»

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти — перший (бакалаврський)

Напрям підготовки 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О. Р. Чертов

«___» _____ 2016 р.

ЗАВДАННЯ

на дипломну роботу студенту

Крівенцеву Максиму Олексійовичу

1. Тема роботи: «Перетворення числових даних для використання в логіко-лінгвістичних моделях»,

керівник роботи Сирота Сергій Вікторович канд. техн. наук, доцент,

затверджені наказом по університету від «06» травня 2016 р. № 1499-С.

2. Термін подання студентом роботи: «9» червня 2016 р.

3. Вихідні дані до роботи: розроблювана система повинна працювати з текстовими файлами, в яких міститься опис об'єктів та їх властивостей, що розділені комами.

4. Зміст роботи: виконати аналіз існуючих методів розв'язання поставленої задачі, обрати структуру у якій будуть зберігатись введені дані, спроектувати систему попередньої обробки числових даних для використання в логіко-лінгвістичних моделях згідно обраного алгоритму дискретизації, здійснити програмну реалізацію спроектованої системи, провести тестування розробленої системи на контрольних прикладах, оформити документацію до дипломної роботи.

5. Перелік ілюстративного матеріалу: екранні форми порівняння існуючих методів, блок-схеми розроблених алгоритмів, графіки порівняння існуючих математичних методів, схема взаємодії модулів системи, знімки екранних форм.

6. Дата видачі завдання: «22» лютого 2016 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за тематикою та збір даних	13.04.2016	
2	Проведення порівняльного аналізу математичних методів дискретизації	17.04.2016	
3	Підготовка матеріалів першого розділу дипломної роботи	20.04.2016	
4	Підготовка матеріалів другого розділу дипломної роботи	28.04.2016	
5	Розроблення програмного забезпечення, що реалізує обраний метод дискретизації неперервних атрибутів	05.05.2016	
6	Підготовка матеріалів третього розділу дипломної роботи	07.04.2016	
7	Перевірка та тестування розробленого програмного забезпечення на контрольних прикладах	15.04.2016	
8	Підготовка матеріалів четвертого розділу роботи	16.05.2016	
9	Оформлення пояснювальної записки	01.06.2016	

Студент _____ Крівенцев М. О.

Керівник роботи _____ Сирота С. В.

АНОТАЦІЯ

Дипломну роботу виконано на 57 аркушах, вона містить 2 додатки та перелік посилань на використані джерела з 21 найменування. У роботі наведено 3 рисунка та 9 таблиць.

Метою даної дипломної роботи є створення математичного та програмного забезпечення для перетворення числових даних для використання їх у логіко-лінгвістичних моделях.

У роботі проведено аналіз існуючих рішень указаної задачі – методів попередньої обробки даних для інтелектуального аналізу даних – процесу дискретизації. Виконано їх порівняння з погляду можливості використання наявних рішень у якості автономного автоматичного програмного засобу, та можливість користувачеві регулювати параметри дискретизації. Для розв'язання задачі в роботі обрано метод, що є комплексною модифікацією до існуючих методів дискретизації.

Реалізовано алгоритм, відповідно до поставлених вимог та завдання. Розроблено програмний засіб, що реалізує обраний метод. Виконано тестування розробленої системи

Ключові слова: інтелектуальний аналіз даних, машинне навчання, дискретизація, аналіз даних, попередня обробка даних.

ABSTRACT

The thesis is presented in 57 pages. It contains 2 appendixes and bibliography of 21 references. 3 figures and 9 tables are given in the thesis.

The goal of the thesis is to develop mathematical and software tools for solving the problem of Preprocessing of Numerical Data for Use in Logico-Linguistic.

In the thesis, existing solutions are analyzed, such as methods for pre-processing data for data mining, in particular – discretization of continuous variables. They are compared in terms of the possibility of using existing solutions in a stand-alone automatic and user`s possibility of change parameters of discretization system. Some modification of basics discretization algorithms is chosen. to solve the task.

Discretization algorithm is implemented according to the general requirements and task. The automated system implementing the chosen method is developed. The developed system is tested.

Keywords: data mining, machine learning, discretization, data analysis, data preprocessing.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

EFD - Equal-Frequency Discretization.

EWD - Equal-Width Discretization.

ID3 - Iterative Dichotomiser 3.

MDL - Minimum Description Length.

WEKA Waikato Environment for Knowledge Analysis.

ШІ - штучний інтелект.

ВСТУП

Однією з головних тенденцій сучасного розвитку штучного інтелекту (ШІ) є інтеграція і гібридизація різних моделей, напрямків і технологій. В першу чергу, мова йде про інтеграцію різних моделей представлення знань і способів міркувань, а також про "сходження" різних напрямків ШІ, що необхідне для розширення можливостей інтелектуальних систем [1]. Тут характерними прикладами є фреймові та продукційні моделі знань і синтетичні методи міркувань (поєднують механізми індукції, абдукції, дедукції), нейро-нечіткі мережі і нейрокомп'ютинг, заснований на знаннях, моделі м'яких обчислень і обчислювального ШІ [2].

Серед "перших ластівок" в цій галузі треба, безумовно, вказати запропоновану ще в 60-і роки XX-го століття концепцію інтегрованого логіко-лінгвістичного моделювання Поспелова Д. А., згідно з якою логічні засоби можуть використовуватися для обробки інформації, представленої в лінгвістичної формі. У руслі цієї концепції потрібне проведення свого роду "інженерного аналізу" природної мови [2].

Зв'язки між філософією та логікою, психологією і логікою, лінгвістикою і логікою завжди були предметом гострих суперечок і дискусій багатьох поколінь вчених. Але лише в останні десятиліття через поширення міждисциплінарних досліджень і розробок, спрямованих на створення інтегрованих систем практично у всіх сферах науки і техніки, ці обговорення стали здобувати важливе практичне значення.

Так у роботі [3] логіко-лінгвістичні методи опису систем засновані на тому, що поведінка системи виражається в термінах обмеженої природної мови і може бути представлена за допомогою лінгвістичних змінних. Неформально під лінгвістичною змінною розуміється така змінна, значеннями якої можуть бути не тільки числа, а й слова і словосполучення будь-якої природної або штучної мови. Вхідні і вихідні параметри системи розглядаються як лінгвістичні змінні.

Одним із основних типів відношень об’єктів в ШІ є відношення класифікації [4]. Класифікація є одним із прийомів формування діагнозів і прогнозів. Різноманіття середовищ, в яких виникають завдання, зводиться до двох нечітко розмежованих типів. Це - середовища, що утворюються неперервними об'єктами, і середовища, в яких переважає дискретність. Для середовищ першого типу характерні кількісні відносини і обчислювальні операції. У середовищах другого типу обробляється якісна, значеннєва інформація. У неперервних середовищах для вирішення завдань використовуються обчислювальні моделі, в дискретних - логіко-лінгвістичні, в яких об'єкти і ситуації представляються атрибутивними (ознаковими) описами, а процеси вирішення завдань є процесами обробки атрибутивних описів [5]. Логіко-лінгвістичні моделі більшою мірою придатні для реалізації адаптивних процесів, а також для вирішення завдань в умовах невизначеності і неповноти інформації. Вони служать основою моделювання процесів мислення і в зв'язку з цим зручні для організації людино-машинного взаємодії при автоматизації складних поведінкових актів [6].

У дослідних процесах часто виникає необхідність виділяти на шкалах ознаках інтервали, які є найбільш характерними для окремих класів об'єктів або загальними для декількох класів, тобто постає задача класифікувати певним чином об’єкти. Завдання вирішується шляхом порівняльного аналізу розподілів об'єктів різних класів на шкалах ознак. Таким чином здійснюється, наприклад, порівняння технічних та ринкових характеристик виробів, що випускаються різними фірмами; зіставлення фізико-хімічних характеристик різних матеріалів; порівняння числових показників, що характеризують підприємства або регіони в різні періоди часу. Для вдалого виконання порівняльного аналізу необхідно перед цим здійснити попередню підготовку вхідних даних та приготувати їх для обробки – здійснити дискретизації вхідних числових даних.

1 ПОСТАНОВКА ЗАДАЧІ

Метою даної дипломної роботи є розробка математичного та програмного забезпечення для підготовки числових даних, які використовуються в логіко-лінгвістичних моделях, а саме – процесу автоматичної побудови шкал по заданим вимогам, або як її ще називають - дискретизації - перетворення неперервних значень атрибутів у категоріальні.

При розробленні відповідного забезпечення потрібно розв'язати наступні завдання:

а) дослідити предметну область та цільове призначення логіко-лінгвістичних моделей;

б) дослідити існуючі методи попередньої обробки даних для використання їх в інтелектуальному аналізі;

в) дослідити математичні методи дискретизації числових даних;

г) обґрунтувати обраний математичний метод та інструментарій для реалізації обраного методу;

д) розробити програмне забезпечення, що реалізує обраний метод;

е) провести тестування програмної реалізації на контрольних прикладах.

Вхідні дані мають бути представлені у вигляді наборів даних (так званих датасетів), в яких у форматі Comma-Separated Values (CSV) міститься інформація про об'єкти, приналежність їх до певного класу, і числові значення по кожній із властивості об'єкта. Ці дані записані з роздільником – крапка з комою.

Результуючі дані представляються у вигляді двох файлів:

- у першому дані записані у аналогічному до вхідного файлу вигляді (CSV-файл), проте числові характеристики властивостей об'єктів мають бути замінені внаслідок роботи алгоритму дискретизації на номери інтервалів, до яких вони входять;

- у другому файлі мають міститись дані про розбиття неперервного атрибуту кожної властивості на скінченну кількість інтервалів, з вказанням порядкового номеру інтервалу, і його меж. Дана інформація має бути надана по кожній з обраної властивості.

Вимоги до розроблюваної системи надаються замовником.

Розроблюване програмне забезпечення має бути використане для:

- порівняльної оцінки технічних і ринкових характеристик виробів, що випускаються різними фірмами;
- дискретизації і зіставлення фізико-хімічних характеристик різних матеріалів;
- зіставлення числових показників, що характеризують підприємства або регіони в різні періоди часу тощо.

2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ДИСКРЕТИЗАЦІЇ ЧИСЛОВИХ АТРИБУТІВ

2.1 Математичні методи дискретизації числових атрибутів

У зв'язку з розвитком нових технологій, темпи зростання наукових баз даних стаю дуже великим, що створює як необхідність так і можливість видобування нових знань з баз даних. Дані в базах даних зазвичай знаходяться в змішаному форматі: номінальному, дискретному і/або неперервному. Неперервні функції також називаються такими, що містять кількісні ознаки, наприклад, людей зріст, вік, тощо.. Дискретні функції, також часто називають функціями з якісними ознаками, наприклад, стать, вчена ступінь, і так далі, тобто містять атрибути, що можуть бути обмежені тільки на декількох значеннях. Неперервні функції можна впорядковувати і на них можуть бути можуть бути визначені основні арифметичні операції. Проте, на дискретних функціях можуть бути застосовані більш складні і значущі операції, але не арифметичні.

В області машинного навчання та інтелектуального аналізу даних, існує багато алгоритмів навчання, які в першу чергу орієнтовані на обробку дискретних функцій. Але в природі, в реальному світі, дані переважно неперервні по своїй суті. Тому, багато наборів даних включають в себе неперервні змінні [7]. Одне із рішень цієї проблеми полягає в поділі числових змінних на ряд піддіапазонів і обробка кожного такого піддіапазону як категорії. Цей процес розбиття неперервних змінних і категорій, як правило, називають дискретизацією.

На жаль, кількість шляхів, якими можна дискретизувати неперервні атрибути достатньо велика. Процес шкалювання є потенційно затратним по часу, так як число можливих дискретизацій (розбиттів інтервалу значень) експоненційно зростає від числа кандидатів на межі інтервалів. Методи дискретизації часто використовують для алгоритмів класифікації, структуризації, навчання на основі екземплярів і широкого спектру алгоритмів навчання.

Використання дискретних значень у роботах в сфері інтелектуального аналізу даних має ряд переваг:

- дискретні атрибути функцій вимагають меншого обсягу пам'яті;
- дискретні значення часто наближені до вигляду, у якому представляються знання;
- в сфері машинного навчання, дискретизовані набори даних значно спрощують, пришвидшують та приводять до більш точного навчання інтелектуальних систем;
- забезпечується краща продуктивність для виокремлення правил класифікації;
- експерти зазвичай описують параметри, використовуючи лінгвістичні терміни замість точного значення. У певному сенсі дискретизація забезпечує краще сприймання атрибутів;
- забезпечується регуляризація, так як вона менш схильна до дисперсійної оцінки від малофрагментованих даних;
- обсяг даних може бути значно зменшено, так як деякі надлишкові дані можуть бути виявлені і вилучені.

Як було сказано раніше в процесі дискретизації неперервна змінна зменшує число різних значень шляхом ділення діапазону значень на скінченну кількість інтервалів, які не перетинаються, і цим інтервалам надаються певні тематичні мітки. Зазвичай це просто порядковий номер інтервалу. Згодом дані аналізуються, або задаються на більш високому рівні представлення знань, а не використовуються як індивідуальні значення, тим самим призводять до спрощеного представлення даних в області набуття знань та інтелектуальному аналізу даних. Процес дискретизації протікає в чотири етапи, які схематично показано на рисунку 2.1.

Рисунок 2.1 – основні кроки роботи алгоритму дискретизації

Метою дискретизації є знаходження набору точок для поділу діапазону значень атрибутів на невелику кількість інтервалів. Основна задача розділяється на наступні дві підзадачі. Перше завдання полягає у знаходженні кількості інтервалів, на яку необхідно розділяти область значень властивості. Лише деякі алгоритми потребують цієї дії. Частіше – користувачу надається можливість самостійно вказати кількість інтервалів початкового розбиття, або ж надається управління евристичним алгоритмам дискретизації. Друге завдання полягає в тому, щоб знайти ширину, або межі інтервалів з урахуванням діапазону значень неперервного атрибута.

Як правило, в процесі дискретизації, після сортування даних в порядку зростання або спадання змінної, що підлягає дискретизації, точки розділу діапазону значень повинні бути обрані серед всього набору даних. Загалом, алгоритм вибору граничних точок може бути, або спадним, який починається з порожнім списком граничних точок, або зростаючим, який починається з повним списком всіх значень в якості граничних точок і поступово об'єднує інтервали. В обох випадках існує критерій зупинки, який вказує, коли слід зупинити процес дискретизації.

2.2 Класифікація методів дискретизації числових атрибутів

Мотивація використовувати алгоритми дискретизації неперервних функцій ґрунтується на необхідності отримання більш високої точності обробки даних з великою кількістю атрибутів. Методи дискретизації були розроблені в залежності з від цільового призначення та цілей використання функцій з дискретизованими значеннями.

Можна виділити такі основні класи методів дискретизації:

- контрольовані (supervised), або неконтрольовані (unsupervised): В неконтрольованих методах неперервні діапазони розділені на піддіапазони, в залежності від обраного користувачем параметру. Наприклад, рівної ширини (із зазначенням діапазону значень), рівному певній частоті (кількість об'єктів в кожному інтервалі), алгоритмом кластеризації, наприклад *k*-середніх (із зазначенням кількості кластерів). Ці методи не можуть дати хороші результати в тих випадках, коли розподіл неперервних значень не є рівномірним, де випади значень мають значний вплив на діапазони. Звичайно, якщо відсутня інформація про клас, то неконтрольована дискретизація є єдиним вибором. У контрольованих методах дискретизації інформація про клас використовується, щоб знайти правильні інтервали, що обираються по опорним точкам. Контрольована дискретизація може бути різних типів: основана на помилках, або ентропії, або на основі статистичних даних в залежності від того вибираються інтервали з використанням метрики на основі помилок навчальних даних, ентропії інтервалів, або деякою статистичною мірою [8];

- ієрархічні (hierarchical), або неієрархічні (non-hierarchical): ієрархічна дискретизація обирає опорні точки для інтервалів інкрементно, утворюючи неявну ієрархію над діапазоном значень. В залежності від методу, процес може бути розділяючим і/або об'єднуючим. Що стосується неієрархічної дискретизації, то ці

методи сканують впорядковані значення тільки один раз, послідовно формуючи інтервали [9];

- спадні (top-down), або зростаючі (bottom-up), чи розділяючі (split), або об'єднуючі (merge): суть спадаючих методів полягає у тому, що вони розглядають один великий інтервал, який містить всі відомі значення ознаки, а потім розщеплює його на менші згідно з алгоритмом, поки не виконається критерій зупинки, або не буде досягнуто оптимальне число інтервалів. На відміну від цього, зростаючі методи починаються з повним переліком всіх неперервних значень функції в якості точок інтервалів і у процесі дискретизації видаляються «зливаючись»» з інтервалом. Використовуються різні критерії зупинки, наприклад – критерій хі-квадрат, або досягнення оптимального числа інтервалів [10];

- статичні (static), або динамічні (dynamic): суть статичного підходу полягає у проведені процесу дискретизації у якості етапу попередньої обробки даних, що виконується один раз, безпосередньо перед обробкою даних. Динамічний підхід полягає у тому, що інтервали розбиття початкового інтервалу значень атрибутів можуть бути розбиті, наприклад, при побудові класифікатора., як в алгоритмі C4,5 [11];

- параметричні (parametric), або непараметричні (non-parametric): параметрична дискретизація вимагає введення від користувача, таких параметрів, як максимальна кількість інтервалів дискретизації. Непараметрична дискретизація використовує тільки інформацію з даних і не вимагає від користувача ніяких додаткових даних;

- глобальні (global), або локальні (local): відмінність між глобальними і локальними методами полягає в тому, на стадіях виконання дискретизації. Глобальні методи дискретизації дискретизують дані перед ітеративним розбиттям інтервалу значень. Вони використовують весь простір дискретизації атрибутів. Локальні методи дискретизації працюють в процесі розбиття інтервалу значень [12]. Емпіричні результати показали, що глобальні методи дискретизації часто отримують кращі результати порівняно з локальними методами, так як перші використовують всю

область значень числового атрибута для дискретизації, в той час як локальні методи дають інтервали, які застосовуються до підінтервалів значення атрибутів. Таким чином, локальний метод зазвичай асоціюється з динамічним методом дискретизації;

- одномірні (univariate), або багатомірні (multivariate): методи одномірної дискретизації квантифікують одну неперервну функцію, тобто властивість, в той час, як багатомірна дискретизація розглядає одночасно кілька властивостей і виконує дискретизації атрибутів по всім наявним властивостям [13].

Для огляду математичних методів було вирішено зосередитись на деяких представниках неконтрольованих і контрольованих методів. З контрольованих методів, було обрано два методи, які відрізняються з точки зору напрямку ієрархії та налаштування критеріїв інтервалів. Перший – метод запропонований Файяд та Ірані (Fayyad, Iran) – метод рекурсивної ентропійної дискретизації – спадний метод, який заснований на оптимізації локальної міри ентропії і критерії зупинки – по опису мінімальної довжини (Minimum Description Length, MDL) [14]. Другий - хі-злиття (Chi-Merge) –зростаючий метод на основі критерію хі-квадрат. З неконтрольованих методів було обрано метод, оснований на розбитті інтервалу значень на однакові довжині інтервалу та метод, оснований на розбитті за рівномірним розподілом частоти входження атрибутів у інтервал.

2.2.1 Неконтрольовані методи дискретизації

Серед неконтрольованих методів дискретизації існують методи, що можна віднести до простих це методи EWD (equal-width discretization) та EFD (equal- frequency discretization), та більш складних, що базуються на кластеризації за k-середніми. Неперервні значення атрибутів поділяються на під-інтервали відповідно до того, який критерій визначив користувач – рівномірну довжину (EWD), чи частоту розподілу (EFD).

2.2.1.1 Метод EWD

Було обрано методи дискретизації з різних класів, щоб дослідити, які з них дозволяють більш зручніше дискретизувати дані.

Розглянемо метод дискретизації по однаковій довжині інтервалу — метод EWD, який являється одним із найпростіших методів неконтрольованої дискретизації. Суть методу полягає у тому, щоб поділити діапазон спостережуваних атрибутів на задану кількість інтервалів, рівних між собою по довжині.

Основні етапи EWD дискретизації:

а) сортування множини значень неперервного атрибуту в порядку зростання значень;

б) визначення мінімального та максимального значення з множини значень атрибутів;

в) визначення користувачем кількості інтервалів розбиття;

г) обчислення довжини інтервалів розбиття шляхом ділення довжини інтервалу всіх значень на кількість інтервалів розбиття:

$$w = \frac{(V_{max} - V_{min})}{k} \tag{2.1}$$

де w (width) — довжина інтервалу розбиття;

V_{min} — мінімальне значення неперервного атрибуту;

V_{max} — максимальне значення неперервного атрибуту;

k — кількість інтервалів розбиття;

Граничні точки інтервалів можуть буди знайдені наступним чином:

$$b = V_{min} + (i \cdot w), \tag{2.2}$$

де b (boundary)– гранична точка інтервалів;

i змінюється від 1 до $k - 1$.

Цей метод дискретизації являється чутливим до викидів (у статистиці — значення випадкової величини, що різко виділяється з експериментальної вибірки), які можуть значно спотворити спектр розподілу даних. Обмеження цього методу визначаються через нерівномірний розподіл точок даних: деякі інтервали можуть містити набагато більше точок даних, ніж інші.

Необхідні для роботи даного методу параметри задаються наступним чином:

- вказуються користувачем;
- обчислене за формулою Стерджеса [15]:

$$k = [\log_2 n + 1]; \tag{2.3}$$

- обчислене за формулою Скотта:

$$k = \frac{(V_{max} - V_{min})}{h}, \tag{2.4}$$

$$h = \frac{3{,}5 \cdot \sigma}{\sqrt[3]{n}}; \tag{2.5}$$

- обчислене за правилом Фрідмана-Діаконіса [16]:

$$k = \frac{(V_{max} - V_{min})}{h}, \tag{2.6}$$

$$h = \frac{2 \cdot IQR}{\sqrt[3]{n}}; \tag{2.7}$$

де k – кількість інтервалів розбиття;

n – кількість об'єктів, властивості яких дискретизуються;

V_{min} — мінімальне значення неперервного атрибуту;

V_{max} — максимальне значення неперервного атрибуту;

σ — стандартне відхилення;

IQR (inter-quartile range) — міжквантильний проміжок (різниця між нижнім і верхнім квартилем множини значень).

Формули Скотта (2.4, 2.5) та Фрідмана-Діаконіса (2.6, 2.7) враховують розподіл кожного атрибута і дають різну кількість інтервалів, в той час, як формула Стерджеса (2.3), аналогічно до вибору користувачем – постійно дають однакову кількість інтервалів.

В таблиці 2.1 наведено приклад роботи алгоритму EWD дискретизації.

Таблиця 2.1 – Приклад роботи EWD дискретизації.

Значення даних	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Оригінальні неперервні дані	10	50	15	20	12	25	40	30
Опис методу	Відсортовані дані: $\{10; 12; 15; 20; 25; 30; 40; 50\}$ Визначимо $V_{min} = 10$, $V_{max} = 50$, За формулами (2.4) та (2,6) $k = 2$, Тоді $w = 20$ і $b = 10 + 20 = 30$							
Дискретизовані дані*	1	2	1	1	1	1	2	1
Опис методу	За формулою (2.3) $k = 4$ Тоді $w = 10$ і $b_i = 20, 30, 40, i = 1..k - 1$							
Дискретизовані дані**	1	4	1	2	1	2	4	3

У таблиці 2.1 наведено приклад застосування алгоритму дискретизації до набору даних із 8 чисел. Далі підрахувавши за формулами (2.3), (2.4) та (2.6) кількість інтервалів розбиття отримали дискретизовані дані, де числові значення замінені на

номера інтервалів до яких вони потрапили. Знаком «*» відмічено дані, що були дискретизовані за формулами (2.4) та (2,6), а «**» - за формулою (2.3).

2.2.1.2 Метод EFD

Розглянемо метод дискретизації на основі рівної частоти розподілення — метод EFD, який являється також одним з найлегших методів неконтрольованої дискретизації. Суть методу полягає у тому, щоб поділити діапазон відсортованих спостережуваних атрибутів на інтервали таким чином, щоб кожен інтервал містив приблизно однакову кількість об’єктів.

Основні етапи EFD дискретизації:

а) сортування множини значень неперервного атрибуту в порядку зростання значень;

б) визначення мінімального та максимального значення з множини значень атрибутів;

в) визначення користувачем кількості інтервалів розбиття;

г) поділ інтервалу неперервних даних на підінтервали таким чином, щоб кожен підінтервал містив однакову кількість атрибутів:

$$f = \frac{n}{k} \tag{2.8}$$

де f (frequency) – кількість об'єктів, що входять в інтервал;

k – кількість інтервалів розбиття;

n – кількість об'єктів, властивості яких дискретизуються.

У випадку використання EFD дискретизації, велика кількість схожих за значенням атрибутів, можуть потрапити до різних підінтервалів розбиття, не

дивлячись на те, що вони можуть бути навіть однакові. Цей алгоритм намагається подолати обмеження методу EWD шляхом ділення інтервалу значень властивості на підінтервали з однаковим розподілом точок даних в них.

Деякі екземпляри даних з однаковим значенням повинні бути розміщені в одному й тому ж самому підінтервалі, але це не завжди можливо зробити у випадку використання EFD методу.

У таблиці 2.2 наведено приклад розв'язання задачі дискретизації EFD методом.

Таблиця 2.2 – Приклад роботи EFD дискретизації.

Значення даних	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Оригінальні неперервні дані	10	50	15	20	12	25	40	30
Опис методу	Відсортовані дані: $\{10; 12; 15; 20; 25; 30; 40; 50\}$ Визначимо $V_{min} = 10$, $V_{max} = 50$, Припустимо, $k = 2$, Тоді за формулою (2.8) $f = \frac{8}{2} = 4$, В кожному інтервалі розбиття має бути по 4 значення.							
Дискретизовані дані	1	2	1	1	1	2	2	2

2.2.1.3 Метод оснований на кластеризації k-середніми

Кластеризація методом k–середніх являється одним із найбільш популярних методів кластеризації. Він також використовується для дискретизації неперервних атрибутів завдяки тому, що він обчислює міру схожості, що залежить від відстаней між кожним об'єктом та його кластером. Насправді, так як неконтрольована

дискретизація проводиться одночасно лише з одною змінною, то цей процес є еквівалентним до процесу одномірного кластерного аналізу *k*-середніми.

Кластеризація методом *k*-середніх неієрархічний алгоритм кластеризації, що працює на множині об’єктів і передбачає, що кількість кластерів визначено. Маємо масив спостережень (об'єктів), кожен з яких має певні значення по ряду ознак. Відповідно до цих значень об'єкт розташовується у багатовимірному просторі. Тоді алгоритм буде складатись із таких кроків:

- визначається кількість кластерів, що необхідно утворити;
- випадковим чином обирається *k* спостережуваних об’єктів, які на цьому кроці вважаються центрами кластерів;
- кожне спостереження «приписується» до одного з n кластерів — того, відстань до якого найкоротша;
- розраховується новий центр кожного кластера як елемент, ознаки якого розраховуються як середнє арифметичне ознак об'єктів, що входять у цей кластер;
- відбувається така кількість ітерацій (повторюються кроки 3-4), поки кластерні центри стануть стійкими (тобто при кожній ітерації в кожному кластері опинятимуться одні й ті самі об'єкти), дисперсія всередині кластера буде мінімізована, а між кластерами — максимізована.

Теоретично, кластери, утворені таким чином, повинні мінімізувати суму квадратів відстані між точками даних усередині кожного кластера над сумою квадратів відстані між точками даних з різних кластерів.

Основним обмеженням дискретизації на основі кластеризації за *k*-середніми є те, що результат дискретизації в основному залежить від заданого значення кількості кластерів *k*, початково обраних центроїдів кластерів, а також він чутливий до статистичних викидів.

У роботі [17] описано контрольований метод кластеризації, що являється варіацією методу, що оснований на методі *k*-середніх і використовує байесівський інформаційний критерій для прийняття рішення про поділ кластеру на підкластери. Цей алгоритм автоматично обирає кількість дискретних інтервалів, без будь-якого

втручання користувача. Інші типи методів кластеризації також можуть бути використані в якості основоположних методів для проведення дискретизації неперервних числових атрибутів. На відміну від методу k-середніх, який являється ітеративним методом, ієрархічні методи можуть бути або агломеративними (від слова агломерат - накоплення), або дивізимними (від англ. division – поділення, розділення). Дивізимні методи кластеризації починають роботу з одного кластера, який містить всі досліджувані об'єкти, а агломеративні методи створюють кластер для кожного об'єкту. Далі ці кластери попарно об'єднуються до тих пір, поки не буде досягнена бажана кількість кластерів. Обидва ці класи методів кластеризації містять конструктивні недоліки, а саме, який кластер розділити, або які кластери об'єднати для того, аби досягти бажаної кількості кластерів.

Після того як кластеризацію було виконано, граничні точки інтервалів дискретизації визначаються як мінімум та максимум інтервалу значень кластеру і середнє значення між цими точками.

У таблиці 2.3 наведено приклад розв'язання задачі дискретизації методом k-середніх, припускаючи, що бажана кількість отриманих кластерів рівна двом.

Таблиця 2.3 – Приклад роботи дискретизації заснованої на кластеризації

Значення даних	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Оригінальні неперервні дані	10	50	15	20	12	25	40	30
Опис методу	Відсортовані дані: $\{10; 12; 15; 20; 25; 30; 40; 50\}$ Припустимо, $k = 2$, 15 та 30 – два випадково обраних центроїди кластерів							
Дискретизовані дані	1	2	1	1	1	1	2	2

2.2.2 Контрольовані методи дискретизації

Контрольовані методи дискретизації використовують інформацію про клас, щоб знайти правильні інтервали, що обираються по опорним точкам. Серед контрольованих методів найбільш простими є метод на основі ентропії, Контрольована дискретизація може бути різних типів: основана на помилках, або ентропії, злиття інтервалів, або злиття за критерієм χ^2 (хі-квадрат).

2.2.2.1 Метод рекурсивної ентропійної дискретизації

Одним з контрольованих ієрархічних методів дискретизації, що був запропонований і визначений Файяда і Ірані, називається дискретизацію на основі ентропії, або метод рекурсивної ентропійної дискретизації. Цей метод використовує правило мінімальної ентропії, що описані у роботі [18].

Ентропія є мірою невизначеності і вимірює кількість інформації, що необхідна, аби вказати, до якого класу належить екземпляр. Спочатку оцінюється один великий інтервал, що містить всі відомі значення ознаки, а потім рекурсивно розбиває цей інтервал на кілька підінтервалів, поки не наступить критерій зупинки, наприклад принцип мінімальної довжини опису MDL (minimum description length), або не буде досягнуто оптимальної кількості інтервалів.

Для початку приведемо визначення ентропії (міри невизначеності) для множини об’єктів вибірки S:

$$Ent(S) = -\sum_{x \in X} p(x)\, log_2\, p(x) \tag{2.9}$$

де X – множина класів в S;

$p(x)$ – відношення потужності класа x до потужності множини S.

У формулі (2.9) використовується функція двійкового логарифму log_2, так, як інформація закодована в бітах. Коли $Ent(S) = 0$, множина S являється ідеально класифікованою, тобто всі її об'єкти належать одному класу.

На основі цієї ентропії Д. Росс Куінлан розробив алгоритм під назвою ID3 (Iterative Dichotomiser 3), щоб знаходити найкращу точку поділу на дерева рішень. ID3 використовує жадібний пошук, щоб знайти потенційні точки розділу межах неперервного діапазону за формулою (2.10):

$$Ent(S,T) = -p_{left}\sum_{j=1}^{m} p_{j,left}\log p_{j,left} - p_{right}\sum_{j=1}^{m} p_{j,right}\log p_{j,right} \quad (2.10)$$

де $p_{j,left}$ – ймовірнсть, того, що екземпляри, що належать до класу j, знаходиться по ліву сторону від потенційної точки T;

$p_{j,right}$ – ймовірнсть, того, що екземпляри, що належать до класу j, знаходиться по праву сторону від потенційної точки T.

Точка розділу з найнижчою ентропією вибирається так, щоб розділити діапазон на два інтервали, далі поділ на два інтервали продовжується до тих пір, поки не виконається критерій зупинки.

Файяд і Ірані запропонувати узагальнений критерій зупинки, що формується на основі принципу MDL, який зупиняє дискретизацію, якщо виконується рівняння (2.11):

$$EntGain(S,T) = Ent(S) - Ent(S,T) < \delta \quad (2.11)$$

де T – потенційна точка розділу інтервалу, що ділить множину S на дві підмножини S_1(ліву) та S_2(праву) і виконується рівняння (2.12):

$$\delta = \frac{\left[\log_2(n-1) + \log_2(3k-2) - [m\,Ent(S) - m_1\,Ent\,(S_1) - m_2\,Ent\,(S_2)]\right]}{n} \quad (2.12)$$

де m – кількість класів в кожній підмножині S_i;

n – кількість даних в множині S.

У таблиці 2.4 представлено результат роботи даного алгоритму на простому прикладі із набором даних, що містить 8 екземплярів.

Таблиця 2.4 – Приклад роботи дискретизації заснованої на кластеризації

Значення даних	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Оригінальні неперервні дані	10	50	15	20	12	25	40	30
Значення класів	1	1	1	1	2	1	2	2
Значення ентропії	0.8622	0.9371	0.9502	0.9056	0.7946	0.9371	0.8622	0.9437
Опис методу	Кількість значень класів = 2 $\min[Ent_A(S)] = 0.7946$ вказує на атрибут зі значенням Тож, критерієм зупинки буде кількість інтервалів, що дорівнює 2.							
Дискретизовані дані	1	2	1	1	1	1	2	2

2.2.2.2 Метод оснований на критерії χ^2

Хі-квадрат (χ^2) – статистична міра, яка проводить перевірку значущості на відносини між значеннями ознаки та класу.

Стверджується, що в точній дискретизації, частоти відносно класу повинен бути досить послідовним у межах інтервалу, але два сусідні інтервали не повинні мати аналогічну відносну частоту класу. У даному випадку критерій χ^2 перевіряє гіпотезу про те, що ознаки атрибутів, що містяться у двох суміжних інтервалах не залежать від класу. Якщо вони між собою незалежні, то вони повинні бути об'єднані в один інтервал. У протилежному випадку – залишитись окремими [19]. Спадний метод, що оснований на порівнянні за критерієм хі-квадрат називається методом хі - розщеплення (Chi Split). Правило за яким завершується розбиття інтервалу заснований на введеному користувачем значені критерію χ^2, за яким здійснюється перевірка на те, чи здійснювати розділення інтервалу, якщо підінтервали схожі, чи ні.

Даний метод шукає найкращий поділ на інтервали за рахунок максимізації критерію χ^2, який застосовується для двох підінтервалів, що суміжні до точки розділу: інтервал розбивається, якщо обидва інтервали істотно розрізняються за статистичними характеристиками. Критерій зупинки даного методу зазвичай базується на основі даних, що вводяться користувачем, а саме – поріг для значення χ^2 для злиття двох інтервалів в один, якщо вони достатньо схожі.

Зростаючий алгоритм, що використовує критерій χ^2 називається хі - злиття (Chi Merge).

Він шукає кращого злиття суміжних інтервалів шляхом мінімізації критерію хі – квадрат і застосовується локально до двох суміжних інтервалів. Інтервали об'єднуються, якщо вони статистично подібні. Критерій зупинки заснований на введеному користувачем значення хі-квадрат, необхідне для того, аби відхилити злиття, якщо два суміжних інтервали недостатньо схожі.

Алгоритм Chi Merge ініціалізується сортуванням значень неперервного атрибуту за зростанням, а потім здійснюється побудова початкової дискретизації, в якій кожен екземпляр поміщається в свій власний інтервал. Якщо два суміжних інтервалів мають схожий розподіл класів, то інтервали можуть бути об'єднані. В методі хі-злиття кожна окрема величина числового атрибута спочатку поміщається у свій окремий інтервал. Далі χ^2 тести виконуються для кожної пари суміжних інтервалів і суміжні інтервали з найменшими значеннями χ^2 об'єднуються разом. Цей процес об'єднання продовжується рекурсивно, поки не буде задовільнено критерій зупинки, тобто до таких значень χ^2, що не перевищують заданий поріг, або не досягнеться визначена кількість інтервалів. Поріг χ^2 визначається рівнем значущості і ступенями свободи, що на одиницю менше ніж число класів.

В наборі даних, у яких об'єкти визначені p класами, формула (2.13) визначає параметр χ^2 за яким точка, що розділяє два суміжних інтервали в залежності від класу p.

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{p} \frac{\left(A_{ij} - E_{ij}\right)^2}{E_{ij}}, \tag{2.13}$$

$$R_i = \sum_{j=1}^{p} A_{ij}, \tag{2.14}$$

$$C_i = \sum_{i=1}^{m} A_{ij}, \tag{2.15}$$

$$R_i = \sum_{j=1}^{p} C_j, \tag{2.16}$$

$$E_{ij} = \frac{R_i \cdot C_j}{N}, \tag{2.17}$$

де p – кількість класів;

A_{ij} – кількість різних значень в i-тому інтервалі, j-того класу;

R_i – кількість атрибутів в i-тому інтервалі;

C_j – кількість атрибутів в j-тому класі;

N – загальна кількість атрибутів у виборці для дискретизації;

E_{ij} – очікувана частота входження A_{ij}.

Приклад таблиці спряженості у випадку роботи алгоритму χ^2 представлена у таблиці 2.5.

Таблиця 2.5 – Таблиця спряженості

	Клас 1	Клас 2	Загалом
Інтервал 1	A_{11}	A_{12}	R_1
Інтервал 2	A_{21}	A_{22}	R_2
Загалом	C_1	C_2	N

У таблиці 2.6 представлено результат роботи алгоритму, що базується на основі злиття за критерієм χ^2.

Таблиця 2.6 – Таблиця спряженості

Значення даних	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Оригінальні неперервні дані	10	50	15	20	12	25	40	30
Значення класів	1	1	1	1	2	1	2	2
Опис роботи методу	Відсортовані дані: $\{10; 12; 15; 20; 25; 30; 40; 50\}$ Метод ініціалізується із інтервалами розбиття, кількість яких рівна кількості об'єктів на всьому інтервалі значень. Далі об'єднуючи суміжні інтервали, для яких значення класів однакове отримаємо розбиття на 5 інтервалів: I1=10, I2= 12, I3= 15,20,25, I4=30,40, I5= 50							

Продовження таблиці 2.6

<table>
<tr><td>Значення даних</td><td>V_1</td><td>V_2</td><td>V_3</td><td>V_4</td><td>V_5</td><td>V_6</td><td>V_7</td><td>V_8</td></tr>
<tr><td rowspan="9">Опис роботи методу</td><td colspan="8">Далі обраховуємо значення χ^2 для кожного інтервалу і визначаємо суміжні інтервали з найменшими значеннями χ^2 які будуть об’єднані разом. Продовжується до тих пір, поки кількість інтервалів не буде рівно 2.</td></tr>
<tr><td colspan="2">10</td><td colspan="2">12</td><td colspan="2">15, 20, 25</td><td colspan="2">30, 40</td><td colspan="2">50</td></tr>
<tr><td colspan="2">$\chi^2 = 2$</td><td colspan="2">$\chi^2 = 4$</td><td colspan="2">$\chi^2 = 5$</td><td colspan="2">$\chi^2 = 2{,}94$</td></tr>
<tr><td colspan="2">10, 12</td><td colspan="2">15, 20, 25</td><td colspan="2">30, 40</td><td colspan="2">50</td></tr>
<tr><td colspan="2">$\chi^2 = 1{,}875$</td><td colspan="2">$\chi^2 = 5$</td><td colspan="2">$\chi^2 = 2{,}94$</td></tr>
<tr><td colspan="3">10, 12, 15, 20, 25</td><td colspan="3">30 40</td><td colspan="2">50</td></tr>
<tr><td colspan="3">$\chi^2 = 2{,}94$</td><td colspan="3">$\chi^2 = 2{,}94$</td></tr>
<tr><td colspan="4">10, 12, 15, 20, 25</td><td colspan="4">30, 40, 50</td></tr>
<tr><td>Дискретизовані дані</td><td>1</td><td>2</td><td>1</td><td>1</td><td>1</td><td>1</td><td>2</td><td>2</td></tr>
</table>

Обмеженням методу хі-злиття є те, що він не може бути використаний для дискретизації даних для завдань методів неконтрольованого навчання (кластеризації). Крім того, метод хі-злиття тільки намагається виявити кореляції першого порядку, таким чином, не може виконатись вірно, коли існує кореляція другого поряду без відповідної кореляції першого порядку, що може статись коли атрибут корелює лише в присутності якого-небудь іншого стану. Другим недоліком методу є відсутність глобальної оцінки. При визначенні того які інтервали необхідно об’єднати, метод розглядає лише суміжні інтервали, при цьому ігноруючи всі інші інтервали. Із-за цього обмеженого локального аналізу, якщо це можливо, створення великих, відносно рівномірних інтервалів може бути відвернено малоймовірним проходом по значенням, що містяться в ньому. Одним із можливих варіантів рішення цієї проблеми може бути виконання тесту на критерій χ^2 для трьох чи більше

інтервалів одночасно. Формула (2.13) легко може бути перетворена і використана для загального випадку шляхом заміни значення параметра m в розрахунку χ^2.

2.3 Порівняння математичних методів

Існує багато методів дискретизації, кожен з яких має свої особливі характеристики і є ефективними у різних ситуаціях. Кожен з цих методів має свої сильні сторони. Порівняльний аналіз методів дискретизації, що були описані у попередніх пунктах розділу 2 базується на приналежності їх до різних класів методів, що описані у розділі 2.2 та на обмеженнях, що на них накладені. Ця інформація викладену у таблиці 2.7, що наведена нижче.

Таблиця 2.7 – Порівняння методів дискретизації

Методи / Критерії оцінки	EWD	EFD	k-середніми	Ентропійна дискретизація	За критерієм χ^2
Контрольований/ неконтрольований	Неконтро-льований	Неконтро-льований	Неконтро-льований	Контро-льований	Контро-льований
Динамічний/ статичний	Статичний	Статичний	Статичний	Статичний	Статичний
Глобальний/ локальний	Глобальний	Глобальний	Локальний	Локальний	Глобальний
Розділяючий/ об'єднуючий	Розділяючий	Розділяючий	Розділяючий	Розділяючий	Об'єднуючий
Критерій зупинки	Стерджеса; Скотта; Фрідмана; Фіксована кількість інтервалів	Фіксована кількість інтервалів	Відсутність змін центру мас кластерів	Поріг значеня ентропії; Фіксована кількість інтервалів	Поріг значеня χ^2; Фіксована кількість інтервалів

Продовження таблиці 2.7

Методи / Критерії оцінки	EWD	EFD	k-середніми	Ентропійна дискретизація	За критерієм χ^2
Чутливість до викидів	Так	Ні	Так	Ні	Ні
Дублювання значень в різних інтервалах	Ні	Так	Ні	Ні	Ні
Обчислювальна складність дискретизації одного атрибуту для n об'єктів	$O(n)$	$O(n)$	$O(ikn)$, де i –кількість інтерацій; k –кількість інтервалів;	$O(n \log n)$	$O(n \log n)$

Додатково було проведено тестування вище описаних методів дискретизації на декількох тестових наборах даних, що були завантажені з репозиторію «UCI Machine Learning Repository» [20]. Було обрано наступні набори даних: «Ecoli», «Glass», «Indian Diabetes», «Ionosphere», «Iris», та «Wine». Всі вони містять тільки числові, неперервні атрибути, які необхідно дискретизувати.

Тестування даних алгоритмів було проведено у системі WEKA (Waikato Environment for Knowledge Analysis), версії 3.8.0 - вільному програмному забезпеченні для аналізу даних та машинного навчання [21].

Тестування методів дискретизації використовувалось на задачі класифікації. Тому якість дискретизованих даних можна оцінити застосувавши до вище зазначених наборів даних методи класифікації. Методи k-середніх і метод ентропійної дискретизації використовуються у якості вбудованих методів попереднього етапу обробки даних у класифікаторах k-середніх та класифікаторі на основі ентропії. Дискретизацію, що базується на критерії χ^2 буде перевірено шляхом обробки даних на класифікаторі на основі зростаючих пірамідальних мереж. Так, як методи EWD та EFD використовуються не у якості вбудованого методу одного із класифікатора, то застосуємо їх

до алгоритму для побудови дерев рішень C4.5. Алгоритми класифікації буде розглянуто у двох варіантах – з навчальною вибіркою 66,6% екземплярів бази та 75%.

Для реалізації даних алгоритмів у системі WEKA існують наступні класифікатори:

- J48 – імплементація алгоритму C4.5 у WEKA;
- IBk – кластеризація методом *k*–середніх;
- KStar – класифікатор на основі екземплярів, що використовує функцію ентропії.

Дані, що були отримані в результаті роботи алгоритмів на різних наборах даних представлено у таблиці 2.8. У цій таблиці наведено значення, які описують відсоток правильної класифікації екземплярів, тобто алгоритм класифікації спочатку навчається на 66,6%, або 75% об'єктах вибірки, і намагається класифікувати інші об'єкти, основуючись на даних, що були ним проаналізовані.

Таблиця 2.8 – Порівняння методів дискретизації на тестових наборах даних

Датасет	Навчальна вибірка	EWD	EFD	*k*-середніми	Ентропійна дискретизація	За критерієм χ^2
Ecoli	66,6%	65,7895%	77,193%	79,4531%	79,4619%	82,1431%
Ecoli	75%	73,8095%	79,7619%	80,9452%	80,9512%	77,3822%
Glass	66,6%	61,6438%	58,9041%	70,4192%	73,2378%	77,4521%
Glass	75%	64,1509%	62,2642%	71,8819%	73,4375%	64,1488%
Indian Diabetes	66,6%	77,7778%	78,5441%	73,8293%	71,0912%	73,4421%
Indian Diabetes	75%	77,6042%	80,7292%	71,3541%	70,8322%	68,7537%
Ionosphere	66,6%	84,0336%	82,3529%	87,0012%	86,4821%	87,2301%
Ionosphere	75%	81,8182%	87,5%	84,3021%	86,1921%	83,9221%
Iris	66,6%	96,0784%	92,1569%	98,0012%	98 %	96 %

Продовження таблиці 2.8

Датасет	Навчальна вибірка	EWD	EFD	*k*-середніми	Ентропійна дискретизація	За критерієм χ^2
Iris	75%	94,5946%	89,1892%	97,3021%	97.3003%	94.5921%
Wine	66,6%	86,8852%	86,8852%	96,6073%	96.6134%	96.6098%
Wine	75%	93,1818%	84,0909%	88,6378%	93.1812%	99,8921%

На рисунку 2.1 можна побачити графічну репрезентацію даних, що були отримані у ході експерименту. По вертикальній шкалі відкладаються відсотки точності отриманих результатів. Можна побачити, що дані класифіковані методами *k*-середніх, ентропійною дискретизацією та дискретизації на основі критерію χ^2 показали кращі результати, аніж найпростіші методи дискретизації - EWD та EFD.

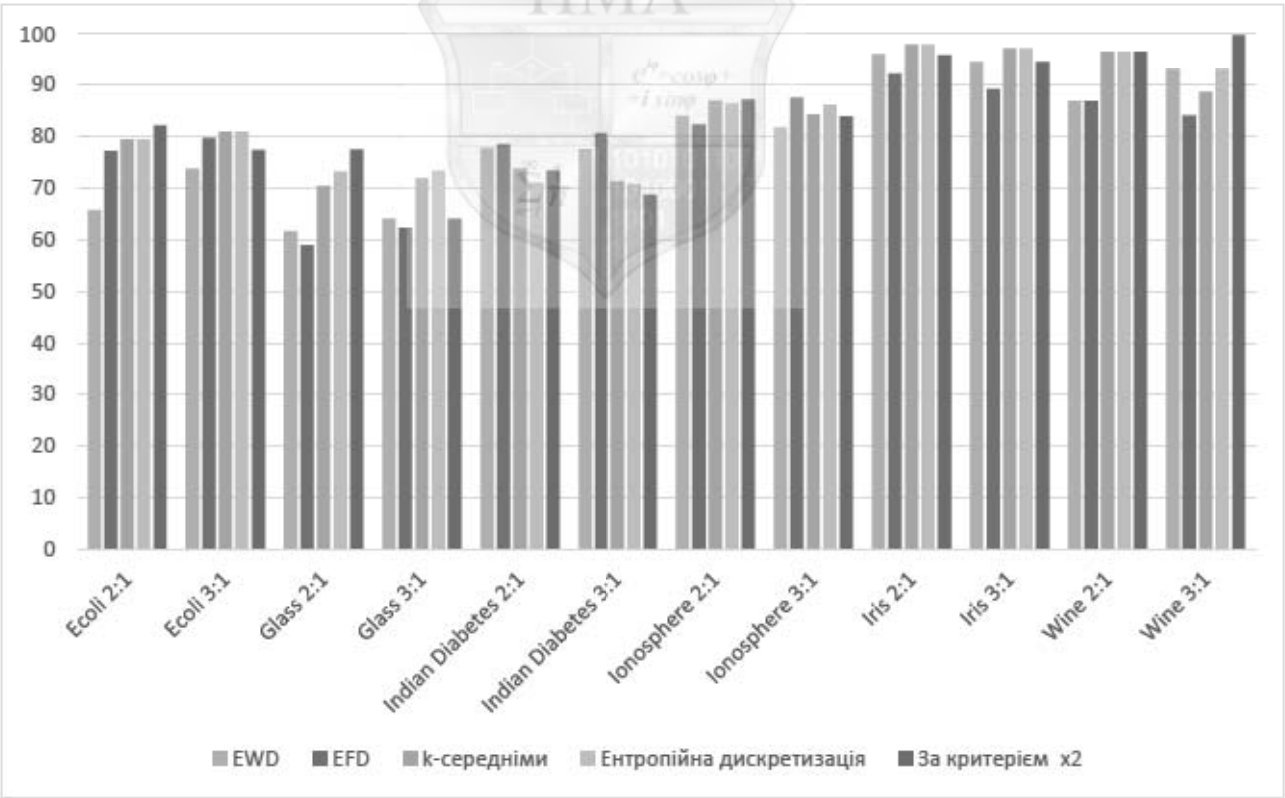

Рисунок 2.2 – Графічне представлення отриманих результатів роботи порівнюваних методів дискретизації

Дискретизація неперервних функцій грає важливу роль в попередньої обробці даних для численних алгоритмів машинного навчання та інтелектуального аналізу даних на наборах даних з числовими атрибутами. Так як зростання кількісті можливих значень числових атрибутів робить процес навчання інтелектуальних систем менш неефективним, то однією з найважливіших цілей алгоритму дискретизації є значне скорочення кількості дискретних інтервалів неперервних значень атрибутів. В даній роботі описано необхідність дискретизації для підвищення ефективності алгоритмів навчання, таксономію методів дискретизації, основні ідеї та недоліки деяких методів, наведено опис алгоритмів, що відносяться до контрольованих та неконтрольованих методів. Також аналізуючи розібрані приклади реалізації алгоритмів можна побачити, що такі неконтрольовані методи як метод кластеризації *k*-середніми, може функціонувати аналогічно, як і контрольовані методи, що використовують дані про ентропію об'єкта, або критерій хі-квадрат. Було проведено порівняльний аналіз існуючих математичних методів на основі приналежності їх до певних класів, за критерієм зупинки, чутливості до зашумлених даних на різних датасетах, що показує недоліки та переваги кожного методу, в залежності від того, на яких типах даних було проведено дискретизацію. Питання вибору єдиного правильного методу стоїть відкритим, адже неможливо обрати єдиний універсальний метод. Відповідь на це питання змінюється в залежності від типів наборів даних, цілей алгоритмів навчання та від кожної конкретної ситуації

2.4 Огляд існуючих програмних рішень

2.4.1 WEKA

Серед існуючих програмних рішень для розв'язання задачі дискретизації неперервних числових атрибутів шкал широко використовується система для аналізу даних та машинного навчання WEKA.

WEKA — вільне програмне забезпечення для аналізу даних та машинного навчання, написане на мові програмування Java в університеті Уайкато (Нова Зеландія), розповсюджується за ліцензією GNU GPL. У WEKA зосереджено набір засобів візуалізації та алгоритмів для аналізу даних і вирішення задач прогнозування, разом з графічною оболонкою для доступу до них.

WEKA дозволяє виконувати такі завдання аналізу даних, як попередня обробка даних (preprocessing) для методів машинного навчання та інтелектуального аналізу даних, відбір ознак (feature selection), кластеризацію, класифікацію, регресійний аналіз та візуалізацію отриманих результатів.

Перші версії WEKA були реалізовані за допомогою C, C++ та LISP. У 2005 році керівництвом проекту було прийнято рішення переписати програмний комплекс на мові програмування Java.

На відміну від інших проектів в WEKA акцент зроблено на наданні середовища для роботи спеціалісту предметної області, а не експерту з машинного навчання. Це проявляється в широкому наборі інтерактивних засобів для керування даними, візуалізації результатів, засобів інтеграції з базами, кросвалідації, та багатьма іншими що доповнюють базові засоби машинного навчання [21].

WEKA надає прямий доступ до бібліотеки реалізованих в ній алгоритмів. Це дозволяє легко використовувати вже реалізовані алгоритми з інших систем, реалізованих на Java. Наприклад, ці алгоритми можна викликати з MATLAB. Зокрема, інтерфейс доступу до алгоритмам WEKA з MATLAB реалізований в деяких алгоритмічних пакетах машинного навчання таких, як Spider і MATLABArsenal.

Система WEKA має ряд додатково встановлюваних розширень, що дозволяють інтегрувати роботу з такими системами машинного навчання та інтелектуального аналізу даних:

- Environment for DeveLoping KDD-Applications Supported by Index-Structures (ELKI) – інтегроване середовище розробки застосунків для видобутку знань із баз даних. Проект схожий на WEKA, більш зосереджений на проблемах кластерного аналізу і інших неконтрольованих методах;

- Konstanz Information Miner (KNIME) – вільно розповсюджувана система, розроблена на мові програмування Java, для перетворення даних, статистики, машинного навчання і візуалізації. Користувачу надається можливість використовувати оператори WEKA та R;

- Massive Online Analysis (MOA) – вільно розповсюджуване програмне забезпечення для обробки великих масивів даних, споріднений до WEKA проект, що було розроблено у тому ж університеті Уайкато (Нова Зеландія);

- Neural Designer – система аналізу даних для глибинного навчання, рішення проблем розпізнавання образів та моделювання високорівневих абстракцій, розроблено на мові C++;

- Orange – система машинного навчання для сфери біоінформатики, що розроблена на мові Python і надає широкі можливості для візуалізації ;

- RapidMiner (YALE) – середовище для проведення експериментів і рішення задач машинного навчання та інтелектуального аналізу даних, тексту, предикативного аналізу, бізнес-аналітики, прототипіювння, тощо.

Для використання WEKA з систем, реалізованих на інших платформах, є можливість виконувати керування засобами WEKA за допомогою інтерфейсу командного рядка.

2.4.2 RapidMiner

RapidMiner (колишня назва YALE) - середовище для проведення експериментів і рішення задач машинного навчання та інтелектуального аналізу даних. Експерименти описуються у вигляді суперпозиції довільного числа довільним чином вкладених операторів, і легко будуються засобами візуального графічного інтерфейсу RapidMiner-а.

Система була розроблена у 2001 році групою із підрозділу штучного інтелекту Дортмундського університету (Artificial Intelligence Unit of Dortmund University of Technology). Реалізовано на мові програмування Java. З вільним клієнтським доступом до Java API [22].

RapidMiner - відкритий програмний продукт, вільно розповсюджуваний під ліцензією GNU AGPLv3. RapidMiner може працювати і як окремий додаток, і як «інтелектуальне ядро», що вбудований в інші додатки, включаючи комерційні. Додатками RapidMiner-а можуть бути як дослідні (модельні), так і прикладні (реальні) завдання інтелектуального аналізу даних, включаючи аналіз тексту (text mining), аналіз мультимедіа (multimedia mining), аналіз потоків даних (data stream mining).

Основні функціональні можливості:

- RapidMiner надає більше 400 операторів для всіх найбільш відомих методів машинного навчання, включаючи введення і виведення, попередню обробку даних і візуалізацію.
- RapidMiner інтегрує в себе оператори WEKA;
- має вбудовану мову сценаріїв, що дозволяє виконувати масивні серії експериментів;
- концепція багаторівневого представлення даних (multi-layered data view) забезпечує ефективну і прозору роботу з даними;
- графічна підсистема забезпечує багатовимірну візуалізацію даних і моделей.

Одним із основних недоліків даної системи, порівняно з рішеннями, що пропонують схожий функціонал, є те, що система комерційно орієнтована і безкоштовна версія програмного забезпечення має занадто обмежений функціонал, що надається на тестовий період – 14 діб.

У вбудованих компонентах RapidMiner аналітика класичних баз даних, порівняно з іншими класичними рішеннями, істотно гірша. Багато даних аналізуються не на зовнішніх серверах, в цих випадках платформа намагається

агрегувати дані на локальній машині, що може мати критичні наслідки, у разі обробки великих масивів даних, внаслідок обмеженості ресурсів локальної машини.

2.4.3 Microsoft R Open (Revolution R)

Revolution R – комерційний продукт для статистичної обробки даних на мові програмування R (мова програмування для статистичної обробки даних і роботи з графікою) і створення програмних рішень з його використанням. Дане середовище, оптимізоване для багатопоточних обчислень, а також, містить набір бібліотек, для масово-паралельної обробки в рамках концепції «великих даних».

З січня 2016 року, компанія Microsoft, що купила компанію Revolution Analytics заявила про перейменування пакету Revolution R Open на Microsoft R Open і відкрила доступ до безкоштовного завантаження по підписці MSDN.

Завдяки участі Microsoft в участі розробки даного програмного продукту стала можлива його інтеграція з наступними технологіями, використовуючи мову програмування R:

- Microsoft R Server / R Server для Azure HDInsight - середовище виконання R-скриптів з поліпшеними показниками швидкості роботи з матрицями, математичними функціями, має поліпшену підтримку багатопоточності і дозволяє виконувати R-скрипти безпосередньо на Spark-кластері в хмарному серсіві Azure. Таким чином розв'язана проблема тошо, що дані, які необхідно оброблювати не переносяться до RAM-пам'яті локальної машини на якій виконується R-скрипт;
- Data Science VM – віртуальна машина за технологією Azure, що дозволяє використовувати інструменти для проведення складних обчислень на віртуальній машині, обробки отриманих даних, інтеграції і зв'язку між собою різних баз даних, отримання звітів і аналізу отриманих результатів. Будування моделей, розроблення прототипів інтелектуальних баз даних облегшується завдяки можливості управління

системою мовою програмування R. Використання Data Science VM дозволить користувачам R швидко почати роботу по дослідженню даних і моделювання в хмарних сервісах без необхідності налаштовувати середовище;

- Azure Machine Learning - хмарний сервіс для виконання задач, що пов'язані із машинним навчанням. Azure ML – центральний сервіс, що використовується для навчання моделей в хмарному середовищі. Управління і статистичний аналіз даних можливий за допомогою інтеграції мови програмування R і виконання R-скриптів в Azure ML Studio.

2.5 Порівняння існуючих програмних рішень

Кожне із розглянутих в розділі 2.4 програмних рішень має ряд власних переваг і недоліків щодо основного функціоналу систем. Але, досліджувані в даній роботі алгоритми дискретизації рідко виступають у якості автономної системи. Так, як процес дискретизації являється невідмінним процесом машинного навчання, то алгоритми дискретизації представлені у вище описаних системах у якості підсистем попередньої обробки даних, що реалізують основний функціонал програмних засобів.

Тому, поставлена перед нами задача не є повністю реалізованою в жодному із зазначених програмних рішеннях.

2.6 Висновки до розділу

На основі порівняльного аналізу існуючих математичних методів розв’язання задачі (розділ 2.3, таблиці 2.7, 2.8 та розділу 2.5), можна зробити висновок, що на сьогоднішній день існують повноцінно працюючі методи, що розв’язують поставлену

задачу, але автономного програмного забезпечення, що реалізує алгоритм дискретизації числових атрибутів в категоріальні знайдено не було. Алгоритми дискретизації лише являються процесом попередньої обробки даних і присутні у якості вбудованих підсистем різноманітних класифікаторах, засобів кластеризації систем машинного навчання. Остаточно визначити найкращий метод неможливо, так як універсального методу не існує, і для кожної окремої задачі необхідно враховувати її особливості. Порівняно з існуючими реалізації математичного методу власна реалізація дещо відрізняється від них, та буде розглянута у розділі 3. Тому для реалізації власного методу дискретизації є необхідність в розробці прикладного математичного забезпечення. Є необхідність розробити реалізацію методу дискретизації, що зможе використовуватись як автономний програмний засіб, а не бути у якості підсистеми певного математичного пакету.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Опис методу дискретизації

В ході роботи було запропоновано модифікацію до існуючих методів дискретизації. Алгоритм дискретного аналізу класів об'єктів за ознакою D і складається в послідовному виконанні наступних операцій:

а) формування шкали значень ознаки D_i:

Нехай задана система ознак $A = \{A_1, A_2, \dots, A_n\}$, для кожної з ознак заданий домен D_i – множина значень ознаки A_i. Тоді в атрибутивних моделях кожен об'єкт q буде представляється наступною множиною:

$$D_q = \left\{ d_i^q \middle| d_i^q \in D_i,\ i \in 1..n \right\} \tag{3.1}$$

Алгоритми дискретизації розбивають множину значень атрибута A_i на деяку сукупність підмножин, що не перетинаються. Тобто утворюється розбиття $\wp(D_i)$ множини значень ознаки D_i. Далі домен D_i замінюється на $\widetilde{D_i}$, де D_i має неперервну область значень, а $\widetilde{D_i}$ – скінченна множина. До того ж, існує функціональне відображення від множини неперервних значень на скінченну множину $g: D_i \to \widetilde{D_i}$.

На першому етапі операція визначення меж шкали полягає в сортуванні неперервних значень ознаки D_i за зростанням і знаходженні найбільшого і найменшого значення на множині всіх аналізованих об'єктів:

$$\alpha = \min_{q \in Q} d_i^q, \tag{3.2}$$

$$\beta = \max_{q \in Q} d_i^q \tag{3.3}$$

Визначення розміру початкового інтервалу розбиття здійснюється за формулою 3.4:

$$w = \frac{\beta - \alpha}{k} \tag{3.4}$$

де w (width) — довжина інтервалу розбиття;

α — мінімальне значення неперервного атрибуту;

β — максимальне значення неперервного атрибуту;

k — початкова кількість інтервалів розбиття.

Від вибору k залежить точність визначення меж інтервалів, час роботи алгоритму, зручність візуалізації шкали. З урахуванням цих чинників використовуємо значення $k = 100$.

Наступним етапом є визначення меж початкових інтервалів розбиття. Ліва і права межа початкових інтервалів обчислюється наступним чином:

$$b_1^l = d_i^{\min}; b_1^r = d_i^{min} + w;$$
$$b_m^l = b_{m-1}^r;\ b_m^r = b_m^l + w; \quad m = 2 \dots k \tag{3.5}$$

де b_m^l — ліва границя m -того інтервалу;

b_m^r — права границя m -того інтервалу;

В результаті даного розбиття множини D_i на k інтервалів отримаємо скінченну множину:

$$\widetilde{D_i} = \left\{I_1^{(1)}, I_2^{(1)}, \dots, I_n^{(1)}, \dots, I_k^{(1)}\right\} =$$
$$= \left\{\left[\alpha, x_1^{(1)}\right), \left[x_1^{(1)}, x_2^{(1)}\right), \dots, \left[x_{n-1}^{(1)}, x_n^{(1)}\right), \dots, \left[x_{k-1}^{(1)}, \beta\right]\right\}, \tag{3.5}$$

де k —кількість інтервалів розбиття;

$\widetilde{D_i}$ — скінченна множина значень ознаки D_i;

$x_i^{(1)}$ — граничні точки інтервалів, $i = 1..k - 1$;

Множина $\widetilde{D}_i$ розбиття ознаки D_i має потужність $Card(\widetilde{D}_i) = k$;

б) Формування розподілів об'єктів досліджуваних класів на шкалі ознаки:

Для кожного інтервалу $I_n^{(1)}$ по кожній властивості V визначається множина вагів $\{w_1^n, w_2^n, \dots, w_j^n, \dots, w_l^n\}$, що означають число об'єктів кожного з класів, значення яких потрапляють в цей інтервал. Визначаються за формулою:

$$w_j^n = Card\left\{q | d_i^q \in I_n^{(1)}\right\}, \ \ j = 1..l \tag{3.6}$$

де $l = Card(V)$ – кількість значень властивості, що досліджуються;

w_j^n — кількість складених об'єктів q, які мають однакове значення властивості v_j, і значення релевантної первинної властивості p кожного q потрапляє в виділений інтервал. $I_n^{(1)}$;

V — цільова властивість об'єкта. Іншими словами, це значення властивості об'єкта, значення якої невідомо для об'єкта прогнозу, але відомо для об'єктів навчальної вибірки.

в) Узагальнення кожного з інтервалів $I_n^{(1)}$:

Непусті інтервали, які межують з порожніми інтервалами, переглядаються, починаючи від лівої межі, і розширюються в обидва боки шляхом приєднання до них кількох сусідніх порожніх початкових інтервалів. Максимальне число порожніх початкових інтервалів, яке може бути приєднано до непорожньої інтервалу з кожного боку, є параметром алгоритму і задається користувачем;

г) Визначення типу інтервалів $I_n^{(1)}$:

Тип інтервалів визначається за наступними критеріями:

- $I_n^{(1)}$— пустий, якщо для всіх j виконується $w_j^n = 0$;
- $I_n^{(1)}$— чистий, якщо існує єдине w_j^n, таке, що $w_j^n > 0$;
- $I_n^{(1)}$— змішаний, якщо існує декілька вагів w_j^n таких, що кожен $w_j^n > 0$.

Змішані інтервали поділяються на:

- $I_n^{(1)}$— рівномірно-змішані, якщо:

$$\max_{j=1..l} w_j^n - \min_{j=1..l} w_j^n < \Delta; \tag{3.7}$$

- $I_n^{(1)}$— змішані, з переважанням по значенню властивості v_τ, якщо:

$$\max_{j=1..l} w_j^n - \min_{j=1..l} w_j^n > \Delta; \tag{3.8}$$

Значення Δ — вводиться користувачем у якості параметра дискретизації. Значення властивості v_τ визначаємо відповідно з $w_\tau^n = \max_{j=1..l} w_j^n$;

г) Об’єднання чистих інтервалів;

Сусідні чисті інтервали, що містять значення ознак об'єктів одного класу, об'єднуються;

д) Об’єднання змішаних інтервалів з превієлюванням по певній властивості:

Інтервали змішаного типу з превієлюванням, які розташовані поруч, об'єднуються. Додатковою умовою об'єднання суміжних змішаних інтервалів з превалюванням за значенням властивості v_τ є однакове значення властивості для кожного з інтервалів, які об'єднуються;

е) Об’єднання рівномірно змішаних інтервалів:

Сусідні інтервали, що на минулому кроці були визначені як рівномірно змішані об’єднуються.

В результаті на деякому кроці s отримуємо наступну скінченну множину $\widetilde{D}_\iota$ інтервалів:

$$\widetilde{D}_\iota = \left\{I_1^{(s)}, I_2^{(s)}, \dots, I_o^{(s)}\right\} =$$
$$= \left\{\left[\alpha, x_1^{(s)}\right), \left[x_1^{(s)}, x_2^{(s)}\right), \dots, \left[x_{o-1}^{(s)}, \beta\right]\right\} \tag{3.9}$$

де, $I_o^{(s)}$ — деякий o-тий інтервал розбиття на кроці s;

o — кількість інтервалів s-того розбиття, $o = Card(\widetilde{D}_\iota),\ o \leq k$.

Отримані інтервали використовуються для дискретизації значень первинних неперервних властивостей об'єктів, які входять в множину прототипів і множину моделей.

Наведений алгоритм дискретного аналізу націлений на забезпечення роздільності і компактності розподілів класів об'єктів в просторі ознак. Це досягається шляхом виділення найбільш характерних інтервалів, а саме, інтервалів, що містять значення ознак тільки або переважно одного класу (чисті, або змішані з превієлюванням інтервали).

3.2 Висновки до розділу

У цьому розділі розроблено математичне забезпечення системи дискретизації неперервних числових атрибутів. У реалізації алгоритму було зосереджено увагу на виконанні усіх поставлених у меті роботи завдань.

В результаті дискретного аналізу виділяються інтервали найбільшого розміру, в результаті чого зменшується загальна кількість сформованих номінальних ознак. Користувач має можливість узагальнювати сформовані інтервали на основі його розуміння проблеми, для вирішення якої проводиться дискретний аналіз.

Так, у розробленому математичному забезпеченні, на відміну від існуючих методів реалізована можливість для користувача розширювати виділені інтервали за рахунок сусідніх порожніх інтервалів і для змішаних інтервалів, користувач має можливість вказати кількість об’єктів класу, необхідну для переважання над іншим класом, аби вказаному інтервалу було надано характеристику «з превієлюванням» за одним із класів.

4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

До розроблюваної системи було висунуто ряд вимог, що дозволятимуть їй в повному обсязі реалізувати поставлену задачу та допомогти під час процесу її проектування і розробки. Замовником було визначено наступні вимоги щодо програмної реалізації:

а) номінальні ознаки, сформовані в результаті дискретизації шкал, повинні забезпечувати роздільність класів в просторі ознак;

б) у просторі ознак, сформованому в результаті дискретизації, області розподілів об'єктів різних класів повинні бути максимально компактними;

в) роздільне і компактне розподілення класів досягається, якщо метод дискретизації виділяє найбільш характерні інтервали в розподілах об'єктів порівнюваних класів на шкалах числових ознак. Найбільш характерними є інтервали, в яких переважають значення ознак деякого класу. Метод дискретизації повинен забезпечувати виділення саме таких інтервалів;

г) при дотриманні вимог роздільності і компактності, а також описаної вище вимоги «в» метод дискретизації повинен виділяти якомога більші інтервали;

д) метод дискретизації повинен забезпечувати задану точність визначення меж виділених інтервалів;

е) користувач повинен мати можливість розширювати виділені інтервали за рахунок сусідніх порожніх інтервалів. Величина зсуву межі виділеного інтервалу задається користувачем на основі його розуміння проблеми, для вирішення якої виконується дискретизація;

ж) для змішаних інтервалів, користувач повинен мати можливість вказати кількість об’єктів класу, необхідну для переважання над іншим класом, аби вказаному інтервалу було надано характеристику «з превієлюванням» за одним із класів.

4.1 Структура програми

Програмне забезпечення для перетворення дискретизації числових даних було спроектовано і розроблено з дотриманням парадигм об'єктно-орієнтованого програмування. Розроблене програмне забезпечення складається із трьох основних модулів:

а) модуль імпортер даних;

б) модуль дискретизації;

в) модуль експорту результатів.

Доцільність розбиття програмного комплексу на три модулі обґрунтовується наявністю спеціальних форматів, у яких збережені вхідні файли, і у які необхідно вивести результуючі дані. Тому, модуль імпортеру даних і модуль експорту результатів являються основними компонентами взаємодії користувача з програмою.

Модуль імпортер даних виконує наступні функції:

- зчитування файлу з даними;
- розпізнання типу файлу;
- розпізнання структури файлу;
- передача структури файлу та даних до модуля дискретизації.

Основним функціональним модулем системи є модуль дискретизації, що здійснює обробку інформації за розглянутим у розділі 3 алгоритмом. Основний функціонал модуля дискретизації зосереджено на наступному:

- прийняття даних про файл і структуру даних від модуля імпортеру;
- обробка даних за допомогою алгоритму дискретизації;
- виведення результуючих даних на модуль експорту.

Після того, як числові атрибути було переведено у категоріальні дані, результат роботи необхідно вивести у результуючі файли. Це функціонал модуля експорту результатів. Він виконує такі функції:

- прийняття результуючих даних від модуля обробки даних;
- збереження файлу з категоріальними атрибутами;
- збереження файлу з проміжками дискретизації

4.2 Формат вихідних даних

4.2.1 Вихідні дані для ініціалізації системи

Вихідними даними для ініціалізації системи є перші два рядки файлу з даними, що збережено у форматі CSV. У першому рядку записано кількість атрибутів, що описують об'єкт-екземпляр, вказано число порожніх початкових інтервалів, яке може бути приєднано до непорожнього інтервалу з кожного боку. Третє число, що записане в першому рядку – кількість об'єктів класу, необхідна для переважання над іншим класом, аби вказаному інтервалу було надано характеристику «з превієлюванням» за одним із класів.

У другому рядку міститься заголовковий запис. Структура заголовкового запису описує екземпляр об'єкта, назви його властивостей, кількість властивостей, загальну кількість атрибутів.

Приклад даних для ініціалізації системи:

4; 14; 30;

<Назва об'єкта>;<Клас об'єкта>;<Властивість1>;<Властивість2>.

4.2.2 Вихідні дані алгоритму дискретизації

Вихідними даними для роботи алгоритму дискретизації являється текстовий файл у форматі CSV, перші два рядки якого ініціалізують роботу системи. В інших

рядках міститься інформація про досліджувані об’єкти. З огляду на формат файлу та структуру запису, всі дані у файлі відформатовано наступним чином:

а) у першому рядку міститься кількість атрибутів;

б) дані у файлі записані через роздільник – крапку з комою;

в) кожен рядок представляє собою новий досліджуваний об’єкт;

г) структура рядка і типи даних наступні:

1) <назва об’єкта> - до 20 будь-яких символів;

2) <клас до якого належить об’єкт> - до 20 будь-яких символів;

3) <властивість n> - дійсне число, що характеризує дану властивість.

Зазвичай, числові датасети вже представлені у заданому форматі. Користувачу немає необхідності самому форматувати файл з даними.

4.3 Формат результуючих даних

Результуючі дані представляються у вигляді двох файлів:

- у першому файлі результуючі дані записані у аналогічному до вхідного файлу вигляді. У форматі CSV-файлу. Проте числові характеристики властивостей об’єктів мають бути замінені внаслідок роботи алгоритму дискретизації на цілі числа - номери інтервалів, до яких вони входять;

- у другому файлі мають міститись дані про розбиття неперервного атрибуту кожної властивості на скінченну кількість інтервалів, з вказанням порядкового номеру інтервалу, його меж і назви інтервалу розбиття. Дана інформація має бути надана по кожній з обраної властивості.

4.4 Результати випробування

Випробування розробленого в рамках даної роботи програмного забезпечення відбувалось аналогічним методом, що описано у розділі 2.3, під час порівняльного аналізу існуючих математичних методів дискретизації. З репозиторію «UCI Machine Learning Repository» [20] для проведення випробування системи було обрано наступні набори даних: «Ecoli», «Glass», «Ionosphere» та «Iris». Всі вони містять в якості атрибутів тільки дійсні числа.

Випробування проводилось для задачі класифікації, як однієї із тих, де дискретизація числових атрибутів відіграє велике значення. Було використано реалізацію алгоритмів класифікації, що представлені в системі для машинного навчання WEKA.

Розглянемо набір даних «Iris». Датасет складаються з даних про 150 вимірювань ірисів з трьох видів — Iris setosa, Iris virginica і Iris versicolor, по 50 вимірювань на вид. Для кожного екземпляра вимірювалися чотири характеристики (в сантиметрах):

Датасет «Ionosphere» містить інформацію, про випромінювання в іоносфері Землі. Даний набір даних складається з 351-го об’єкта і містить 34 числових характеристики і одну бінарну.

Датасет «Ecoli» містить інформацію про 8 характеристик 336 досліджуваних об'єктів бактерії кишкової палички. Всі характеристики – дійні числа.

Датасет «Glass» складається з 214 об'єктів з 10-ма характеристиками скла.

Класифікацію було проведено алгоритмом C4,5 на долі навчальної вибірки 66,6% і 75%. Порівняння буде проводитись з двома методами: *k*-середніми і EFD. Результати випробування записані у таблицю 4.1, у якій записані відсоткові значення правильної класифікації об’єктів. Графічне відображення таблиці 4.1 зображено на рисунку 4.1. На вертикальній шкалі відкладаються значення успішно класифікованих об’єктів у відсотках.

Таблиця 4.1 – випробовування системи

Датасет	Навчальна вибірка	Реалізований метод	EFD	*k*-середніми
Ecoli	66,6%	75,7226%	77,193%	79,4531%
Ecoli	75%	83,1468%	79,7619%	80,9452%
Glass	66,6%	79,3721%	70,4192%	77,4521%
Glass	75%	74,6223%	71,8819%	64,1488%
Ionosphere	66,6%	93,3291%	87,0012%	87,2301%
Ionosphere	75%	92,3333%	84,3021%	83,9221%
Iris	66,6%	98,0411%	98,0012%	96 %
Iris	75%	100%	97,3021%	94.5921%

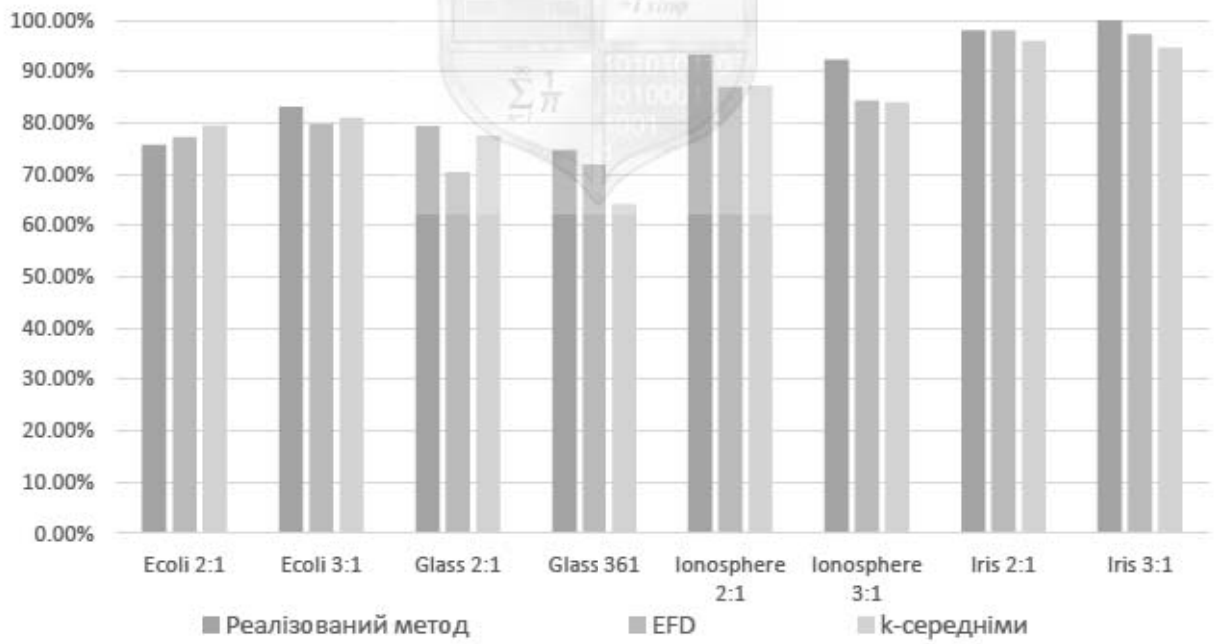

Рисунок 4.1 – Випробування системи та порівняння з існуючими методами

Отримані результати свідчать про те, що розроблене програмне забезпечення працює коректно, і дають більший відсоток правильної класифікації ніж існуючі методи.

4.5 Висновки до розділу

У цьому розділі було описано розроблене програмне забезпечення перетворення числових даних для використання в логіко-лінгвістичних моделях.

Дискретизовані дані, що були отримані в результаті роботи були перевірені на класифікаторі системи WEKA. Отримані дані перевищують в середньому на 5% значення точності класифікації об'єктів за вихідними даними, щ обули дискретизовані іншими методами.

ВИСНОВКИ

У роботі було розглянуто основні підходи до підготовки числових даних для використання в логіко-лінгвістичних моделях. Серед можливих засобів попередньої обробки даних було розглянуто алгоритм шкалювання числових даних (дискретизації).

Алгоритм дискретизації дозволяє здійснювати порівняльний аналіз числових характеристик об'єктів різних класів, а також створений для перетворення числових ознак в номінальні на етапі попередньої обробки даних при вирішенні дискретних аналітичних задач (набуття знань з даних, класифікація, діагностика, прогнозування).

Було проведено аналіз існуючих математичних методів та програмних рішень, що реалізують дані методи, проведено їх порівняльний аналіз. Відповідно до вимог замовника було реалізовано метод, що найліпше підходить для перетворення числових даних, що у подальшому будуть використовуватись для класифікації об’єктів. Данна реалізація може бути використана у якості автономного програмного забезпечення для досягнення наступних цілей:

- порівняльної оцінки технічних і ринкових характеристик виробів, що випускаються різними фірмами;
- дискретизації і зіставлення фізико-хімічних характеристик різних матеріалів;
- зіставлення числових показників, що характеризують підприємства або регіони в різні періоди часу тощо.

Система здійснює поділ шкал числових параметрів на інтервали шляхом зіставлення розподілів значень цих параметрів, що характеризують різні класи об'єктів. В результаті виділяються інтервали, що в найбільшій мірі характеризують досліджувані класи об'єктів. Застосування системи з метою попередньої обробки даних в комплексі із засобами вирішення аналітичних задач значно підвищує точність класифікації, діагностики або прогнозування.

Відмінною рисою є те, що описана у роботі реалізація алгоритму дискретизації може працювати автономним комплексом та автоматично, або із вказанням необхідного розбиття інтервалу значень атрибутів з урахуванням розподілів об'єктів різних класів, що дозволяє формувати більш ефективні класифікаційні правила при вирішенні дискретних аналітичних задач для цих класів.

Розроблене математичне програмне забезпечення було реалізовано програмними засобами. Було проведено випробовування даного програмного засобу на наборах даних, порівняно їх з вже існуючими реалізаціями методів. В результаті порівняльного аналізу, метод дискретизації, що було реалізовано в рамках даної дипломної роботи показав результати, в середньому на 5-7% кращими, для методів класифікації, ніж існуючі реалізації

ПЕРЕЛІК ПОСИЛАНЬ

1. Тарасов В.Б. Логико-лингвистические модели: прошлое, настоящее и будущее // Политехнические чтения: Сб. тр. Вып. 7. Искусственный интеллект - проблемы и перспективы / Политехн. музей; науч. ред. Г.Г. Григорян, В.Л. Стефанюк. М.: РАИИ, 2006. - С. 48-68.

2. Маккарти Дж., Хайес Р. Некоторые философские проблемы в задаче построения искусственного интеллекта // Кибернетические проблемы бионики. - М.: Мир, 1973. - С. 40-87.

3. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. [пер. с англ.] – М.: Мир, 1976. − 166 с.

4. Поспелов Д.А. Логико-лингвистические модели в системах управления. - М.: Энергоатомиздат, 1981. -231с.

5. Валькман, Ю.Р. Интеллектуальные технологии исследовательского проектирования: формальные системы и семиотические модели. - К. : Port-Royal, 1998. - 249 с.

6. Гладун В. П . Партнерство с компьютером . – К .: «Port-Royal», 2000. – 128 с.

7. Morgan Kaufmann Publishers, 2011. Jiawei Han, Micheline Kamber Jian Pei.

8. J. Dougherty, R. Kohavi, M. Sahami. Supervised and unsupervised discretization of continuous features. In Proceedings of the 12th International Conference on Machine Learning (1995), pp. 194-202

9. R. Kerber. Discretization of Numeric Attributes. Proceedings of the 10th National Conference on Artificial Intelligence, MIT Press, 1992, pp.123-128.

10. F. Hussain, H. Liu, Ch. L. Tan, M. Dash. Discretization: An Enabling Technique. Technical Report – School of Computing, Singapore, June 1999.

11. J.Quinlan.C4.5: Programs for Machine Learning. M. Kaufmann, San Mateo, CA, 1993.

12. J. Dougherty, R. Kohavi, M. Sahami. Supervised and unsupervised discretization of continuous features. In Proceedings of the 12th International Conference on Machine Learning (1995), pp. 194-202

13. S. Bay. Multivariate discretization of continuous variables for set mining. In Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2000), pp. 315–319

14. H. Sturges. The Choice of a Class Interval. J. American Statistical Association: 1926, pp. 65–66.

15. D. Scott. On Optimal and Data-based Histograms. Biometrika 66 (3), 1979, pp. 605–610.

16. D. Freedman, P. Diaconis. On the Histogram as a Density estimator: L2 Theory. Probability Theory and Related Fields (Heidelberg: Springer Berlin) 57 (4): (December 1981), pp. 453–476

17. Haiyang Hua, Huaici Zhao, "A Discretization algorithm of Continuous attribute based on Supervised Clustering" ,Chinese Conference on Pattern Recognition, vol. 8, no. 3, 2009, pp. 1-5.

18. U.Fayyad, K.Irani. Multi-interval Discretization of Continuous-Valued Attributes for Classification Learning. Proceedings of the 13th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Mateo, CA, 1993, pp.1022-1027.

19. R. Kerber. Discretization of Numeric Attributes. Proceedings of the Tenth National Conference on Artificial Intelligence, MIT Press, Cambridge, MA, 1992, pp.123-128.

20. A. Asuncion, D.J. Newman. UCI Machine Learning Repository. University of California, Irvine, CA, School of Information and Computer Science. Режим доступу: http://archive.ics.uci.edu/

21. I. Witten, E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. 2nd Edition, Morgan Kaufmann, San Francisco, 2005. Режим доступу: http://www.cs.waikato.ac.nz/ml/weka

22. Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, Timm Euler. YALE: Rapid Prototyping for Complex Data Mining Tasks // International Conference on Knowledge Discovery and Data Mining (KDD-06), 2006. Режим доступу: http://rapidminer.com/