Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Інформаційних управляючих систем_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

_____«Дослідження методів класифікації веб-сторінок на основі технології інтелектуального аналізу даних»_____
(тема)

Виконав:
студент _2_ курсу, групи _ІУСТм-20-1____
_____Сотников К. В._____
(прізвище, ініціали)

Спеціальність ____122 Комп'ютерні науки_____
(код і повна назва спеціальності)

Тип програми ___освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)

Освітня програма _Інформаційні управляючі системи та технології_____
(повна назва освітньої програми)

Керівник _____доц. Міхнова А. В._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ ___Петров К.Е.__
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерних наук_____

Кафедра_____Інформаційних управляючих систем_____

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність_____122 Комп'ютерні науки_____
(код і повна назва)

Тип програми _____освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)

Освітня програма___Інформаційні управляючі системи та технології_____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____»_____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Сотникову Кліму Володимировичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи ____«Дослідження методів класифікації веб-сторінок на основі технології інтелектуального аналізу даних»_____
_____

затверджена наказом університету від __05__ __11__ 2021__ р. № 1645 Ст_

2. Термін подання студентом роботи до екзаменаційної комісії _08_ _12_ 2021_ р.

3. Вихідні дані до роботи опис існуючих методів та моделей класифікації веб-контенту;_____ опис існуючих методів та алгоритмів інтелектуального аналізу даних; звітні матеріали передатестаційної практики_____

4. Перелік питань, що потрібно опрацювати в роботі __аналіз методів веб-класифікації та постановка задачі, дослідження методів інтелектуального аналізу даних, розробка методу класифікації веб-сторінок, розробка моделі класифікації веб-сторінок, розробка методів підвищення точності класифікації_____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Ознайомлення з предметом та об’єктом дослідження	08.11.21	
2	Огляд та аналіз науково-технічної літератури за темою кваліфікаційної роботи	07.11.21	
3	Проведення наукового дослідження щодо розробки методу класифікаціївеб-сторінок	12.11.21	
4	Розроблення програмного забезпечення	18.11.21	
5	Оформлення пояснювальної записки	30.112.21	

Дата видачі завдання _____ _____20__ р.

Студент _____
(підпис)

Керівник роботи _____ ____доц. Міхнова А. В._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи студента: 82 стор., 5 ілл., 3 табл, 16 джерел, 2 додатка.

PYTHON, HTML, CSS, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, КЛАСИФІКАЦІЯ ВЕБ-СТРОІНОК

Об’єктом дослідження є методи класифікації веб-сторінок та інтелектуального аналізу даних.

Мета роботи: модифікація існуючих методів класифікації веб-сторінок на основі інтелектуального аналізу даних для підвищення точності.

В ході проведення аналізу досліджуваної області було визначено, що існуючі методи в повній мірі не задовольняють сучасним вимогам точності і повноти класифікації веб-сторінок.

Розроблені нові методи підвищення точності класифікації веб-контенту на основі існуючих , які дозволяють виконувати класифікацію веб-сторінок з точністю 96%.

ABSTRACT

Explanatory note: 82 pages., 5 figs., 3 tables, 16 sources**.** 2 ann.

PYTHON, HTML, CSS, DATA MINING, CLASSIFICATION OF WEBPAGES

The object of the research is the methods and algorithms of data mining and classification of web pages.

Purpose of the research: the modification of the existing methods and algorithms of data mining to improve the accuracy of the classification of web pages.

During the analysis of the studied area it was determined that the existing methods do not fully meet the current requirements for the accuracy and completeness of the classification of web pages.

There were developed new methods to improve the accuracy of the web content classification model based on the existing methods and was developed model, which allows to classify web pages with an accuracy of 96%.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

Bagging - технологія класифікації, коли всі елементарні класифікатори навчаються і працюють паралельно (незалежно один від одного);

CRISP-DM - Cross-Industry Standard for Data Mining - модель, що описує основні етапи, виконання яких дозволяє організаціям отримувати максимальну вигоду від використання методів інтелектуального аналізу даних;

Cross-validation - метод оцінювання достовірності математичної моделі з метою перевірки, наскільки результати статистичного аналізу узагальнюються на незалежному наборі даних;

Data Mining - виявлення прихованих закономірностей або взаємозв'язків між змінними у великих масивах необроблених даних;

Decision tree - засіб підтримки прийняття рішень, що використовується в статистиці і аналізі даних для прогнозних моделей;

KDD - Knowledge Discovery in Databases - процес пошуку корисних знань в "сирих" даних;

kNN - k-Nearest Neighbors - найпростіший метричний класифікатор, що заснований на оцінюванні подібності об'єктів. Об'єкт, що класифікується, відноситься до того класу, якому належать найближчі до нього об'єкти навчальної вибірки;

N-gram - послідовність з n елементів, це може бути послідовність звуків, складів, слів або літер;

Random Forest - алгоритм машинного навчання, що полягає у використанні комітету (ансамблю) вирішальних дерев;

SVM - Support Vector Machine - набір схожих алгоритмів навчання з учителем, що використовуються для задач класифікації та регресійного аналізу.

TF-IDF - Term Frequency - Inverse Document Frequency - статистичний показник, що використовується для оцінки важливості слів у контексті документа, що є частиною колекції документів чи корпусу.

URL - Uniform Resource Locator - стандартизований спосіб запису адреси ресурсу в мережі.

ВСТУП

Сьогодні Інтернет займає важливу роль в житті людини. Інформаційний простір в мережі налічує вже мільйони гігабайт даних різного роду і відрізняється високим рівнем доступності для користувачів.

Легкість створення та редагування контенту в Інтернеті призводить до поширення небажаної інформації, зокрема забороненого контенту.

Інформація в Інтернеті відрізняється високою динамікою: створення нового контенту, його редагування та видалення займають кілька секунд. З огляду на кількість користувачів, які можуть створювати небажаний контент, використання традиційних методів виявлення та класифікації подібної інформації стає незручним.

Визначення тематики контенту веб-сторінок є однією з найважливіших задач багатьох інтернет-компаній. Наприклад, за умови коректної класифікації можна пропонувати користувачеві більш точну добірку рекламних блоків, що в свою чергу дозволить підвищити продаж як місць розміщення рекламних банерів, так і рекламованого товару. Крім того, захист від небажаної інформації також є однією з основних можливих сфер застосування класифікації контенту.

Для автоматизації перевірки і класифікації веб-контенту, а також для виявлення небажаних для перегляду веб-сторінок і веб-сайтів, можна використати методи інтелектуального аналізу даних. Завдання технології інтелектуального аналізу даних - виявити структури даних і знайти закономірності в слабо структурованих даних. Зважаючи на точність класифікації, що дають існуючі методи, можна зробити висновок, що такі методи потребують модифікації.

Метою роботи є дослідження існуючих методів класифікації веб-сторінок та реалізація обраного методу на основі технологій інтелектуального аналізу даних.

Основними завданнями роботи є:

- дослідження існуючих методів та моделей класифікації веб-контенту;
- дослідження існуючих методів та алгоритмів інтелектуального аналізу даних;
- вибір і вивчення інструментів інтелектуального аналізу даних;
- розробка способу класифікації веб-контенту;
- апробація запропонованого методу класифікації веб-контенту.

1 АНАЛІЗ МЕТОДІВ ВЕБ-КЛАСИФІКАЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз існуючих методів веб-класифікації

Сьогодні існує безліч наукових робіт в області класифікації веб-сторінок. Основною відмінністю класифікації веб-сторінок від звичайного тексту є гіпертекст. Зважаючи на це, можна виділити два типи класифікації:

- класифікація за вмістом цільової веб-сторінці;
- класифікація за вмістом сусідніх веб-сторінок.

Розглянемо можливі існуючі методи класифікації.

Серед атрибутів веб-сторінок розрізняють текстові та візуальні. Для класифікації текстова інформація - зручніша для використання. Для цього використовуються кілька варіантів вибору атрибутів, такі як bag-of-words, TF-IDF і n-gram. Такі методи зазвичай застосовуються в дослідженнях аналізу тексту.

Веб-сторінка використовує HTML теги, які застосовуються в якості контейнерів, в яких може знаходитись текст. Такі теги можуть бути обрані в якості атрибутів.

Веб-сторінку можна представити у вигляді ієрархії візуальних елементів, таких як навігація, контент та інші блоки. Не завжди впровадження такого методу дозволяє збільшити точність. В результаті можна об'єднати дані підходи для підвищення точності класифікації. Продуктивність моделі класифікації можна також поліпшити за рахунок зменшення розмірності даних.

Використання сусідніх веб-сторінок дозволяє значно підвищити точність. Найбільш корисними для класифікації є веб-сторінки, на які посилаються батьківська веб-сторінка цільової, а також веб-сторінки з тими ж посиланнями. При цьому сусідні веб-сторінки також можуть

додавати велику кількість шуму. Якщо розглядати зв'язки між веб-сторінками, то можна побудувати граф і на основі цього отримати вектор, використовуючи методику як в TF-IDF.

Кожна веб-сторінка має свою унікальну адресу URL, за допомогою якої можна виконати порівняно швидку класифікацію без скачування веб-сторінки. Використання n-gram при класифікації за URL веб-сторінки також може підвищити ефективність [2].

1.2 Методи фільтрації контенту

Під фільтрацією контенту мається на увазі програмне забезпечення, яке дозволяє обмежити доступ до небажаного контенту в мережі для певного кола людей.

Найчастіше фільтрація контенту відбувається на рівні веб-запитів протоколу HTTP. В такому випадку URL веб-сайту порівнюється з «чорним» списком, для такого порівняння зазвичай використовуються регулярні вирази.

«Чорні» списки потрібно часто оновлювати, адже в іншому випадку захист з їх допомогою стає малоефективним. Найбільш якісними є методи класифікації і обробки природної мови. В такому випадку класифікація веб-сайтів виконується за допомогою аналізатора кількості ключових слів за різними ознаками. Властивості, що отримуються з тексту, використовуються для визначення ступеня ймовірності відповідності небажаним категоріям. У випадку, коли ймовірність стає вище встановлених значень, відбувається блокування доступу.

Найпростіші програми дозволяють ввести слова, а система буде вести їх пошук. В той же час більш складні програми мають великий словник і мають вже готову базу посилань, що були попередньо

класифіковані. Як правило, розробники забезпечують періодичне оновлення бази посилань більш складних програм. Якщо веб-сайт не класифікований автоматично, то людина переглядає його і привласнює категорію сайту вручну.

Зрозуміло, що швидкодія класифікації – одна з найбільш важливих вимог до програм обмеження доступу.

Фільтрація контенту – це обмеження доступу користувачів до веб-сторінок. Головними методами аналізу контенту вважаються систем тематичної класифікації вмісту веб-сторінки та пошук за ключовими словами.

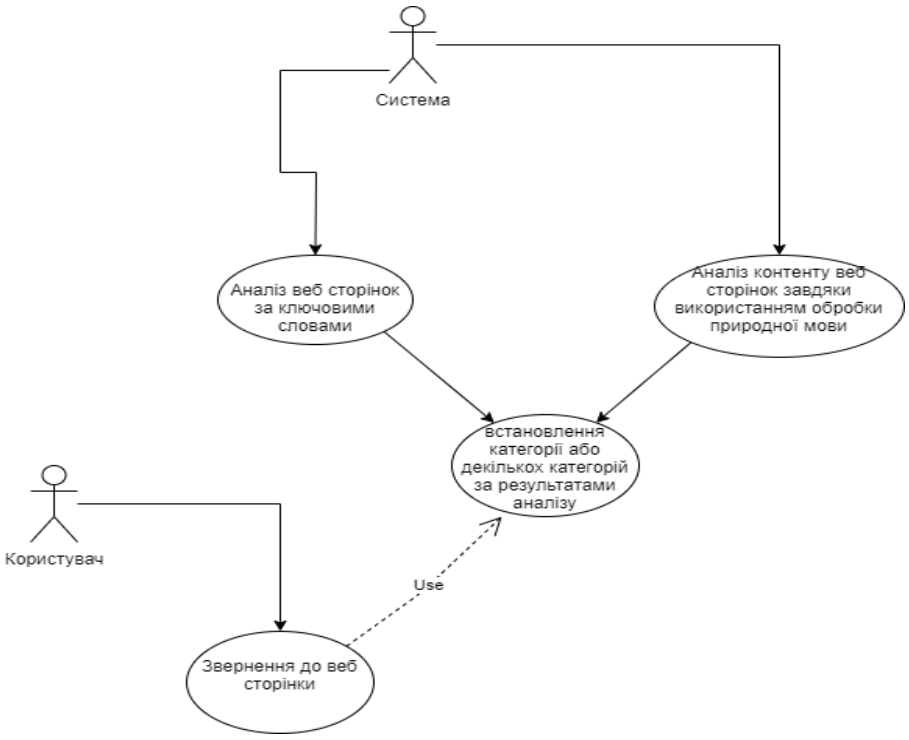

Рисунок 1.1 - Use case диаграмма роботи системи фільтрації контенту

1.2.1 Динамічне визначення тематичної категорії

Для визначення категорії вміст сторінки аналізується з використанням обробки природної мови при зверненні користувача до неї. Після чого виконується встановлення категорії або декількох категорій за результатами аналізу.

Є два види такої фільтрації:

- фільтрація вмісту запитів;
- фільтрація вмісту сайтів.

Перевагою способу динамічного визначення є те, що створюється модель, яка в онлайн режимі визначає, чи може користувач переглядати даний сайт чи ні. Складність цього методу полягає в створенні досконалої моделі розпізнавання вмісту та в подальшій класифікації конкретного сайту.

Застосування цього способу надає ряд істотних переваг:

- проведений аналіз всіх веб-сторінок знижує шанс доступу до небажаного контенту, адже існує ряд сайтів з динамічною генерацією адрес;
- зникає потреба постійного оновлення списків, тому що проводиться постійний аналіз тексту, незалежно від того, коли цей текст змінювався;
- блокування виконується на рівні однієї веб-сторінки, а не всього веб-сайту, що дозволяє вирішити проблеми, пов'язані з веб-сайтами складної класифікації, такі як новини, наприклад.

1.2.2 Списки URL

Системи фільтрації веб-контенту з використанням списків URL можуть використовувати як локальні сховища, так і віддалені бази даних.

Коли використовується віддалена база даних, система відправляє запит на доступ до веб-сайту, де виконується пошук в базі даних і приймається рішення щодо надання доступу.

Локальні системи періодично оновлюють бази даних, тому для ефективної роботи необхідно оновлювати їх якомога частіше.

Фільтрація за списками має низку істотних недоліків:

- повнота охоплення;
- списки URL не можуть містити всі можливі адреси всіх існуючих ресурсів Інтернету;
- періодичність оновлення списків;
- контент веб-сайтів постійно змінюється, відбувається поява нових веб-сайтів і міграція старих, що в свою чергу накладає відбиток на якість такого підходу до класифікації;
- соціальні мережі та блоги - основною особливістю подібних веб-сайтів є величезна кількість веб-контенту, який постійно змінюється та може бути небажаним для перегляду [1].

1.2.3 Фільтрація за ключовими словами

Метод фільтрації за ключовими словами полягає у пошуку в тексті певних ключових слів або словосполучень. Якщо веб-сторінка містить подібні слова і словосполучення, то відбувається блокування доступу до цієї веб-сторінки.

Описаний метод дозволяє відмінно фільтрувати контент за умови, коли наявність певних словосполучень або слів може однозначно визначити ступінь небезпеки для подальшого блокування веб-сторінки. В даному випадку це означає, що при вживанні таких слів контекст грати ролі не повинен. Найчастіше, коли веб-контент перевіряється на рахунок вмісту ключових слів, без аналізу контексту неможливо однозначно відповісти на питання, чи варто виконувати блокування веб-сайту чи ні.

Тому даний спосіб перевірки доцільніше використовувати як доповнення при використанні інших технологій фільтрації.

1.3 Постановка задачі подальших досліджень

Метою роботи є дослідження методів класифікації веб-сторінок за допомогою існуючих методів інтелектуального аналізу даних, модифікація цих методів, підвищення їх точності та розробка моделі, що дозволяє виконувати мультикласову класифікацію веб-сторінок.

В ході досліджень повинні бути проаналізовані існуючі методи та алгоритмів інтелектуального аналізу даних, методики класифікації веб-контенту. Базуючись на цьому, потрібно вибрати інструменти інтелектуального аналізу даних для підвищення рівня точності методів класифікації веб-контенту.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

2.1 Методи інтелектуального аналізу даних

Вперше поняття інтелектуального аналізу даних прозвучало в 1978 році. Спочатку його в основному застосовували для завдань обробки та аналізу данихв межах прикладної статистики. Також при цьому вирішувалися завдання обробки невеликих баз даних. В подальшому основнимзавданням інтелектуальногоаналізуданихставпошук прихованих знань у великих базах даних [3].

Розглянемо декілька відомих класифікацій методів інтелектуального аналізу даних за різними ознаками.

Всі методи інтелектуального аналізу даних можна розділити на дві великі групи за принципом роботи з вихідними навчальними даними. У цій класифікації верхній рівень визначається на підставі того, чи зберігаються дані після інтелектуального аналізу даних або вони дистилюються для подальшого використання.

Відповідно маємо дві групи:

Перша група методів передбачає безпосереднє використання даних, або збереження даних.

У цьому випадку вихідні дані зберігаються в явному детальному вигляді і безпосередньо використовуються на стадіях прогнозованого моделювання та/або аналізу винятків.

Недолік цієї групи методів в тому, що при їх використанні можуть виникнути складності аналізу надвеликих баз даних.

Методи цієї групи:

- кластерний аналіз;
- міркування за аналогією;
- метод найближчого сусіда;

- метод k-найближчого сусіда.

Друга група методів передбачає виявлення і використання формалізованих закономірностей, або дистиляція шаблонів.

За умови використання технології дистиляції шаблонів один зразок (шаблон) інформації витягується з вихідних даних і перетворюється в певні формальні конструкції, вид яких залежить від методу інтелектуального аналізу даних, що використовується.

Цей процес виконується на стадії вільного пошуку, у першій же групі методів дана стадія в принципі відсутня. На стадіях прогнозованого моделювання та аналізу винятків використовуються результати стадії вільного пошуку, вони значно компактніше самих баз даних.

Конструкції цих моделей можуть трактуватися аналітиком або ні ("чорні ящики").

Методи цієї групи:

- логічні методи;
- методи кростабуляції;
- методи візуалізації;
- методи, засновані на рівняннях.

Розглянемо методи другої групи більш детальніше.

Логічні методи, або методи логічної індукції, включають:

- нечіткі запити і аналізи;
- дерева рішень;
- символьні правила;
- генетичні алгоритми.

Методи цієї групи, мабуть, найбільше інтерпретуються, адже вони оформляють знайдені закономірності в досить прозорому вигляді з точки зору користувача. Отримані правила можуть включати безперервні і дискретні змінні. Слід зауважити, що дерева рішень можуть бути легко перетворені в набори символьних правил шляхом генерації одного правила по шляху від

кореня дерева до його термінальної вершини. Дерева рішень і правила фактично є різними способами вирішення однієї задачі і відрізняються лише за своїми можливостями. Крім того, реалізація правил здійснюється більш повільними алгоритмами, ніж індукція дерев рішень.

Методи крос-табуляції включають:

- агенти;
- Баєсові мережі;
- крос-табличну візуалізацію.

Методи на основі рівнянь висловлюють виявлені закономірності у вигляді математичних виразів - рівнянь. Отже, вони можуть працювати лише з чисельними змінними, і змінні інших типів повинні бути закодовані відповідним чином. Це дещо обмежує застосування методів даної групи, проте вони широко використовуються при вирішенні різних завдань, особливо завдань прогнозування.

Основні методи даної групи: статистичні методи та нейронні мережі. Статистичні методи найбільш часто застосовуються для вирішення

завдань прогнозування. Існує безліч методів статистичного аналізу даних, наприклад:

- гармонійний аналіз;
- кореляційно-регресійний аналіз;
- кореляція рядів динаміки;
- виявлення тенденцій динамічних рядів.

Інша класифікація розділяє методи інтелектуального аналізу даних

на дві групи:

- статистичні методи, що базуються на використанні усередненого досвіду, що відображається в даних, які накопичуються в БД за тривалий період;
- кібернетичні методи, що включають безліч різних математичних підходів.

Статистичні методи представляють собою чотири взаємопов’язані розділи:

- попередній аналіз природи статистичних даних - полягає в перевірці гіпотез стаціонарності, незалежності, нормальності, однорідності, а також оцінці виду функції розподілу, її параметрів тощо;
- виявлення зв'язків та закономірностей (кореляційний аналіз, лінійний та нелінійний регресійний аналіз тощо);
- багатовимірний статистичний аналіз (кластерний аналіз, компонентний аналіз, лінійний та нелінійний дискрімінантний аналіз тощо);
- динамічні моделі та прогноз на основі часових рядів.

Кібернетичні методи інтелектуального аналізу даних - це набір підходів, що об'єднані ідеєю комп'ютерної математики та використанням теорії штучного інтелекту. До цієї групи належать такі методи:

- еволюційне програмування;
- штучні нейронні мережі (розпізнавання, кластеризація);
- генетичні алгоритми (оптимізація);
- асоціативні правила (пошук аналогів, прототипів);
- дерева рішень;
- нечітка логіка;
- системи обробки експертних знань.

Методи інтелектуального аналізу даних також можна класифікувати за завданнями інтелектуального аналізу даних.

Відповідно до такої класифікації можна виділити дві групи. Перша група – це підрозділ методів, що займається вирішенням завдань сегментації (тобто завдання класифікації і кластеризації) та прогнозування.

Відповідно до другої класифікації методи інтелектуального аналізу даних можуть бути спрямовані на отримання описових і прогнозованих результатів.

Описові методи служать для знаходження шаблонів або зразків, що описують дані, які піддаються інтерпретації з точки зору аналітика.

До методів, спрямованих на отримання описових результатів, відносяться ітеративні методи кластерного аналізу, в тому числі:

- алгоритм k-медіани;
- алгоритм k-середніх;
- методи крос-табличної візуалізації;
- ієрархічні методи кластерного аналізу;
- карти Кохонена, що само організуються;
- різні методи візуалізації.

Прогнозовані методи використовують значення одних змінних для передбачення та прогнозування невідомих (пропущених) або майбутніх значень інших змінних.

До методів, що спрямовані на отримання прогнозованих результатів, відносяться такі методи:

- дерева рішень;
- лінійна регресія;
- нейронні мережі;
- метод найближчого сусіда;
- метод опорних векторів;
- тощо.

Різні методи інтелектуального аналізу даних характеризуються певними властивостями, які можуть бути визначальними при виборі методу аналізу даних. Методи можна порівнювати між собою, оцінюючи характеристики їх властивостей.

Серед основних властивостей і характеристик методів інтелектуального аналізу даних розглянемо наступні:

- масштабованість;
- точність;
- гнучкість;
- швидкість;
- інтерпретованість;

- трудомісткість;
- популярність.

На плакаті №2 представлена порівняльна характеристика деяких поширених методів інтелектуального аналізу даних. Оцінка кожної з характеристик проведена за наступними категоріями (впорядку зростання):

- надзвичайно низька;
- дуже низька;
- низька/нейтральна;
- нейтральна/низька;
- нейтральна;
- нейтральна/висока;
- висока;
- дуже висока.

Проаналізувавши наведену порівняльну характеристику методів інтелектуального аналізу даних можна зробити висновок, що кожен з методів має свої сильні і слабкі сторони. Але жоден метод, якою б не була його оцінка з точки зору властивих йому характеристик, не може забезпечити вирішення всього спектру завдань інтелектуального аналізу даних.

Більшість інструментів інтелектуального аналізу даних, що пропонує зараз ринок програмного забезпечення, реалізують відразу кілька методів, наприклад, дерева рішень, індукцію правил та візуалізацію, або ж нейронні мережі, карти Кохонена, що само організуються, та візуалізацію.

В універсальних прикладних статистичних пакетах (наприклад, SPSS, SAS, STATGRAPHICS, Statistica, ін.) реалізується широкий спектр найрізноманітніших методів (як статистичних, так і кібернетичних). Слід враховувати, що для можливості їх використання, а також для інтерпретації результатів роботи статистичних методів (кореляційного,

регресійного, факторного, дисперсійного аналізу та ін.) потрібні спеціальні знання в галузі статистики.

Універсальність того чи іншого інструменту часто накладає певні обмеження на його можливості. Перевагою використання таких універсальних пакетів є можливість відносно легко порівнювати результати побудованих моделей, отриманих різними методами. Така можливість реалізована, наприклад, в пакеті Statistica, де порівняння засноване на так званій "конкурентній оцінці моделей". Ця оцінка полягає в застосуванні різних моделей до одного і того ж набору даних і наступному порівнянні їх характеристик для вибору найкращої з них [4].

2.2 Завдання інтелектуального аналізу даних

В основу технології інтелектуального аналізу даних покладена концепція шаблонів, що представляють собою закономірності. В результаті виявлення цих закономірностей вирішуються завдання інтелектуального аналізу даних.

Завдання інтелектуального аналізу даних в залежності від способу їх вирішення можна розділити на два класи:

- навчання з учителем;
- навчання без вчителя.

У першому випадку потрібен навчальний набір даних, на якому створюється і навчається модель інтелектуального аналізу даних. В подальшому готова модель тестується і згодом використовується для передбачення значень в нових наборах даних.

У другому випадку мета завдань полягає у виявленні закономірностей, що наявні в існуючому наборі даних. Варто зауважити, що при цьому навчальна вибірка не потрібна.

В якості прикладу можна навести завдання аналізу споживчого кошика, коли в ході дослідження виявляються товари, що покупці найчастіше купують разом. До цього ж класу належить задача кластеризації.

Якщо говорити про класифікацію завдань інтелектуального аналізу даних за призначенням, то відповідно до неї, вони діляться на:

- описові;
- передбачливі.

Мета вирішення описових завдань - краще зрозуміти дані, що досліджуються, виявити наявні в них закономірності, навіть якщо в інших наборах даних вони не зустрічатимуться.

Для передбачливих завдань характерним є те, що в ході їх вирішення на підставі набору даних з відомими результатами будується модель для передбачення нових значень.

Основними завданнями інтелектуального аналізу даних є:

- класифікація;
- регресія;
- кластеризація;
- пошук асоціативних правил;
- пошук послідовності.

Завдання класифікації вирішується в два етапи. На першому виділяється навчальна вибірка. У неї входять об'єкти, для яких відомі значення як незалежних, так і залежних змінних. Для нашого прикладу це інформація про клієнтів, яким раніше видавалися кредити на різні суми, та інформація про їх погашення.

На підставі навчальної вибірки будується модель визначення значення залежної змінної. Її також називають функцією класифікації або регресії. Для отримання максимально точної функції для навчальної вибірки висуваються такі основні вимоги:

- кількість об'єктів, що входять до вибірки, має бути досить велика, адже чим більше об'єктів, тим точніше буде побудована на її основні функція класифікації;
- до вибірки повинні входити об'єкти, що представляють всі можливі класи;
- для кожного класу в задачі класифікації вибірка повинна містити достатню кількість об'єктів.

На другому етапі побудовану модель застосовують до об'єктів, що аналізуються, тобто до об'єктів з невизначеним значенням залежної змінної [4].

Розглянемо завдання регресії.

За допомогою регресійного аналізу можна отримати конкретні відомості про те, яку форму і характер має залежність між змінними, що досліджуються.

Метод найменших квадратів, що становить математичну основу регресійного аналізу, спочатку застосовувався в астрономії і геодезії. Надалі поєднання методу найменших квадратів та статистичних методів привело до виникнення регресійного аналізу.

Розв’язання завдання регресійного аналізу включає три етапи:

- встановлення форми залежності;
- визначення функції регресії;
- оцінку невідомих значень залежної змінної.

Для будь-яких завдань з кількісними змінними, що змінюються, представляє інтерес дослідження впливу одних змінних на інші. Таким впливом, звичайно, може бути простий функціональна зв'язок між змінними. Але для багатьох фізичних процесів це скоріше виключення, ніж правило.

Ймовірно, часто існує функціональний зв'язок, що занадто складний для розуміння або для опису простими термінами. У такому разі можна намагатися підібрати апроксимацію цього функціонального зв'язку за допомогою якої-небудь простої математичної функції (прикладом може бути

поліном), яка включає відповідні змінні, і згладжувати або апроксимувати «істинну» функцію в певній обмеженій області змін цих змінних.

При дослідженні такої спрощеної функції є можливість більше дізнатися про «справжню» залежність, що розглядається, і оцінити окремі або спільні ефекти зміни деяких важливих змінних.

Оцінка значень залежної змінної зводиться до вирішення задачі одного з наступних типів:

- оцінка значень залежної змінної всередині розглянутого інтервалу вихідних даних, при цьому вирішується завдання інтерполяції;
- оцінка майбутніх значень залежної змінної, знаходження значень поза заданого інтервалу вихідних даних, при цьому вирішується завдання екстраполяції.

Обидва завдання вирішуються шляхом підстановки в рівняння регресії знайдених оцінок параметрів значень незалежних змінних. Результат вирішення рівняння представляє собою оцінку значення цільової (залежної) змінної [5].

Кластеризація представляє собою логічне продовженням ідеї класифікації. Це завдання більш складне, адже особливість кластеризації полягає в тому, що класи об'єктів спочатку не визначені. Результатом кластеризації є розбиття об'єктів на групи.

Характеристиками кластера можна назвати дві ознаки:

- внутрішня однорідність;
- зовнішня ізольованість.

Кластеризація відрізняється від класифікації тим, що для проведення аналізу не потрібно мати окрему залежну змінну. Це завдання вирішується на початкових етапах дослідження, коли про дані мало що відомо. Її вирішення допомагає краще зрозуміти дані, і з цієї точки зору завдання кластеризації – описове завдання.

Для завдання кластеризації характерна відсутність будь-яких відмінностей як між змінними, так і між об'єктами. Навпаки, шукаються групи найбільш близьких, схожих об'єктів. Методи автоматичного розбиття на кластери рідко використовуються самі по собі, в більшості випадків тільки для отримання груп схожих об'єктів. Після визначення кластерів використовуються інші методи інтелектуального аналізу даних, щоб спробувати встановити, що означає таке розбиття, чим воно викликане.

У маркетингових дослідженнях кластерний аналіз застосовується достатньо широко. Це стосується як теоретичних досліджень, так і вирішення практичних завдань, наприклад, завдань, що стосуються проблеми згрупування різних об'єктів. При цьому вирішуються питання про групи клієнтів, продуктів тощо.

Одним з найважливіших завдань при застосуванні кластерного аналізу в маркетингових дослідженнях є аналіз поведінки споживача, а саме:групування споживачів в однорідні класи для отримання максимально повного уявлення про поведінку клієнта з кожної групи і про фактори, що впливають на його поведінку. Об'єднання споживачів в групи дозволяє спростити задачу, так як розглядати поведінку кожного споживача окремо фізично не можливо.

Важливим завданням, яке може вирішити кластерний аналіз, є позиціонування, тобто визначення ланки, в якій потрібно позиціонувати новий продукт, що пропонує ринок. В результаті застосування кластерного аналізу будується карта, за допомогою якої можна визначити рівень конкуренції в різних сегментах ринку і відповідні характеристики товару для можливості попадання в цей сегмент.

Відзначимо ряд особливостей, що притаманні завданню кластеризації.

По-перше, вирішення достатньо сильно залежить від природи об'єктів, що аналізуються. З одного боку, це можуть бути однозначно визначені кількісно об'єкти, а з іншого - об'єкти, які мають нечіткий опис.

По-друге, вирішення завдання великою мірою залежить і від ймовірних відносин об'єктів і кластерів. Також необхідно враховувати такі

властивості, як можливість або неможливість приналежності об'єктів до кількох кластерів. Необхідно точне визначення самого поняття приналежності об'єкта кластеру:

- однозначна (належить або не належить);
- ймовірнісна (ймовірність приналежності);
- нечітка (ступінь приналежності) [5].

Розглянемо завдання пошуку асоціативних правил.

В результаті вирішення завдання пошуку асоціативних правил визначаються закономірності між пов'язаними подіями в наборі даних. Знайдені залежності представляють у вигляді правил і можуть бути використані як для кращого розуміння природи даних, що аналізуються, так і для передбачення появи подій.

Спочатку завдання вирішувалося при аналізі тенденцій в поведінці покупців в супермаркетах. Аналізу піддавалися дані про покупки, що робили споживачі. При аналізі цих даних інтерес перш за все представляє інформація про те, які товари споживачі купують разом, які категорії споживачів яким товарам надають перевагу, в який період часу тощо. Така інформація дозволяє більш ефективно планувати закупівлю товарів, проведення рекламної кампанії тощо.

У медицині аналізу можуть піддаватися симптоми і хвороби, що спостерігаються у пацієнтів. У цьому випадку знання про те, які поєднання хвороб і симптомів зустрічаються найчастіше, допомагають в майбутньому правильно поставити діагноз.

Відмінність асоціації від двох попередніх задач інтелектуального аналізу даних полягає в тому, що пошук взаємозв'язків здійснюється між кількома подіями, які відбуваються одночасно [5].

Розглянемо завдання пошуку послідовності.

Різновидом пошуку асоціативних правил є пошук послідовностей, або послідовна асоціація.

Послідовна асоціація дозволяє знайти тимчасові закономірності між подіями. Завдання пошуку послідовності схоже з завданням асоціації, але її метою є встановлення закономірностей не між подіями, що настають одночасно, а між подіями, що впорядковані в часі (тобто відбуваються в деякому порядку). Іншими словами, послідовність визначається як існування високої ймовірності ланцюжка подій, що пов'язані у часі. Фактично, асоціація є окремим випадком послідовності з кроком часу, що дорівнює нулю. Це завдання інтелектуального аналізу даних також називають завданням знаходження послідовних шаблонів.

Правило послідовності може формулюватися так: після події X через певний час відбудеться подія Y.

Аналіз послідовності широко використовується, наприклад, в телекомунікаційних компаніях для аналізу даних про аварії на різних вузлах мережі. Інформація про послідовність здійснення аварій може допомогти у виявленні неполадок та попередженні нових аварій. Наприклад, якщо відома послідовність збоїв: {e5, e2, e7, e13, e6, e1, ...}, де e1 - код збою, то на підставі факту появи збою e2 можна зробити висновок про швидку появу збою e7. Знаючи це, можна провести профілактичні заходи, що допоможуть усунути причини виникнення збою. Якщо додатково володіти і знаннями про час між збоями, то можна передбачити не лише факт його появи, а й час, що також не менш важливо.

Пошук послідовності застосовується і в маркетингу. Наприклад, може бути встановлено, що після покупки квартири мешканці в 60% випадків протягом двох тижнів купують холодильник, а протягом двох місяців в 50% випадків купується телевізор. Вирішення задач, що подібні до даної, широко застосовується в менеджменті, наприклад, при управлінні циклом роботи з клієнтом.

Завдання прогнозування вирішуються в різноманітних областях людської діяльності, таких як наука, медицина, економіка, виробництво тощо.

Прогнозування є важливим елементом організації управління як окремими господарюючими суб'єктами, так і економіки в цілому. Прикладами можуть бути наступні завдання:

- прогнозування руху грошових коштів;
- прогнозування урожайності агрокультури;
- прогнозування фінансової стійкості підприємства.

Типовим в сфері маркетингу є завдання прогнозування ринків. В результаті вирішення даного завдання оцінюються перспективи розвитку кон'юнктури певного ринку, зміни ринкових умов в майбутньому, визначаються тенденції ринку (структурні зміни, потреби покупців, зміни цін). Крім економічної і фінансової сфери, завдання прогнозування ставляться в найрізноманітніших областях: медицині, фармакології, технічних науках.

Розвиток методів прогнозування безпосередньо пов'язаний з розвитком інформаційних технологій, зокрема, із зростанням обсягів даних,що зберігаються,та ускладненням методів і алгоритмів.

В результаті вирішення завдання прогнозування на основі особливостей історичних даних оцінюються пропущені або ж майбутні значення цільових чисельних показників.

Прогнозування направлено на визначення тенденцій динаміки конкретного об'єкта або події на основі ретроспективних даних, тобто аналізу його стану в минулому та сьогодні. Таким чином, рішення завдання прогнозування вимагає деякої навчальної вибірки даних.

У найзагальніших рисах рішення задачі прогнозування зводиться до вирішення таких підзадач:

- вибір моделі прогнозування;
- аналіз адекватності та точності побудованого прогнозу.

Для вирішення таких завдань широко застосовуються методи математичної статистики, нейронні мережі і т.д [5].

2.3 Підходи ведення проектів інтелектуального аналізу даних

Для вирішення описаних вище завдань існують дві популярні методології ведення проектів інтелектуального аналізу даних:

- Knowledge Discovery in Databases (KDD);
- Cross Industry Standard Process for Data Mining (CRISP-DM).

Технологія Knowledge Discovery in Databases включає в себе питання:

- підготовки даних;
- вибору інформативних ознак;
- очищення даних;
- застосування методів інтелектуального аналізу даних;
- післяобробки даних;
- інтерпретації отриманих результатів.

Весь процес KDD представлений у вигляді схеми на рисунку 2.1.

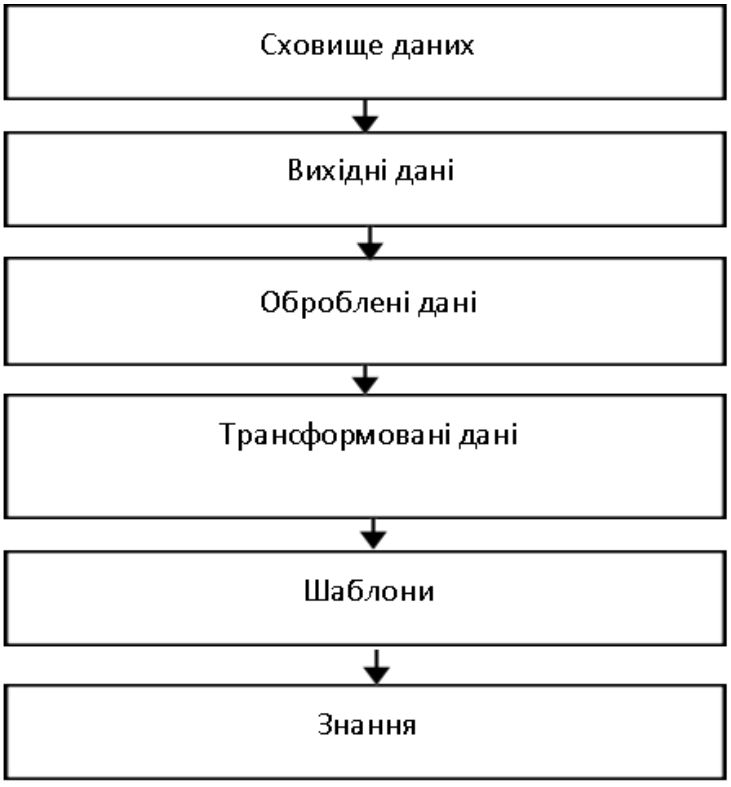

Рисунок 2.1 - Процесс Knowledge Discovery in Databases

Розглянемо процес Knowledge Discovery in Databases детальніше.

Підготовка вихідного набору даних. Цей етап полягає в створенні набору даних, в тому числі з різних джерел, вибору навчальної вибірки тощо. Для цього повинні існувати розвинені інструменти доступу до різних джерел даних. Бажано мати підтримку роботи зі сховищами даних і наявність семантичного шару, що дозволяє використовувати для підготовки вихідних даних не технічні терміни, а бізнес-поняття.

Передобробка даних. Для того, щоб ефективно застосовувати методи інтелектуального аналізу даних, слід звернути увагу на питання передобработки даних. Дані можуть містити пропуски, шуми, аномальні значення тощо. Крім того, дані можуть бути надлишкові, недостатні тощо. У деяких задачах потрібно доповнити дані деякою апріорною інформацією.

Якщо на вхід системи подати дані в існуючому вигляді, то, зрозуміло, що на виході ми одразу не отримаємо корисні знання. Дані повинні бути якісні і коректні з точки зору використовуваного методу інтелектуального аналізу даних. Тому перший етап KDD полягає в передобробці даних. Більш того, іноді розмірність початкового простору може бути дуже велика, і тоді бажано застосовувати спеціальні алгоритми зниження розмірності.

Трансформація, нормалізація даних. Цей крок необхідний для приведення інформації до придатного для подальшого аналізу виду. Для цього потрібно виконати, наприклад, приведення типів, квантування тощо. Крім того, деякі методи аналізу вимагають, щоб вихідні дані були в якомусь певному вигляді. Наприклад, нейронні мережі працюють тільки з числовими даними, причому вони повинні бути нормалізованими.

Інтелектуальний аналіз даних. На цьому етапі застосовуються різні алгоритми для знаходження знань. Це алгоритми кластеризації, нейронні мережі, дерева рішень, встановлення асоціацій тощо.

Постобработка даних. Даний етап полягає в інтерпретації результатів та застосуванні отриманих знань в бізнес додатках.

Knowledge Discovery in Databases передбачає послідовність дій, яку необхідно виконати, щоб з вихідних даних отримати знання. Але він не визначає набір методів обробки або алгоритми, що придатні для аналізу. Перевагою технології є те, що даний підхід універсальний і не залежить від предметної області [5].

Cross Industry Standard Process for Data Mining включає шість основних етапів:

- розуміння бізнесу — перша фаза процесу спрямована на визначення цілей проекту і вимог з боку бізнесу, потім ці знання конвертуються в постановку задачі інтелектуального аналізу даних і попередній план досягнення цілей проекту. Кроки:
    1) визначити бізнес мету;
    2) оцінити ситуацію;
    3) визначити цілі аналізу даних;
    4) скласти план проекту;
- розуміння даних. — друга фаза починається зі збору даних і ставить за мету познайомитися з даними якомога ближче, для котрої необхідно виявити проблеми, що пов’язані з якістю даних, якщо такі є, зрозуміти, які дані є в наявності, спробувати відшукати цікаві набори даних або сформувати гіпотези про наявність прихованих закономірностей в даних. Кроки:
    1) зібрати вихідні дані;
    2) описати дані;
    3) дослідити дані;
    4) перевірити якість даних.
- підготовка даних — фаза підготовки даних ставить за мету отримання підсумкового набору даних з вихідних різнорідних та різноформатних даних, що в подальшому будуть використовуватися при моделюванні. Кроки:
    1) визначити бізнес мету;
    2) оцінити ситуацію;
    3) визначити цілі аналізу даних;

4) скласти план проекту.

- моделювання — до даних застосовуються різноманітні методики моделювання, будуються моделі, а їх параметри налаштовуються на оптимальні значення. Зазвичай для вирішення будь-якої задачі аналізу даних існує кілька різних підходів. Деякі підходи ставлять особливі вимоги для подання даних. Таким чином досить часто потрібно повернутися на крок назад до фази підготовки даних. Кроки:

1) вибрати методику моделювання;
2) зробити тести для моделі;
3) побудувати модель;
4) оцінити модель.

- оцінка — модель вже побудована і отримані кількісні оцінки її якості, перевірка, що всі поставлені бізнес-цілі були досягнуті.

Основна мета етапу - пошук важливих бізнес-задач, яким не було приділено належної уваги. Кроки:

1) оцінити результати;
2) зробити перевірку процесу;
3) визначити наступні кроки.

- розгортання — фаза розгортання залежить від вимог та, відповідно до них, може бути простою (наприклад, складання фінального звіту) або складною (наприклад, автоматизація процесу аналізу даних для вирішення бізнес задач). Зазвичай фазу розгортання виконує клієнт. Навіть якщо аналітик не бере участь в розгортанні, важливо, щоб клієнт чітко розумів, що йому потрібно зробити для того, щоб почати використовувати отриману модель. Кроки:

1) запланувати розгортання;
2) запланувати підтримку і моніторинг розгорнутого рішення;
3) зробити підсумковий звіт;
4) зробити огляд проекту.

Відповідно до загальних принципів і методологій отримаємо наступний алгоритм виконання інтелектуального аналізу даних:

- постановка завдання аналізу;
- збір даних;
- підготовка даних (фільтрація, доповнення, кодування);
- підбір параметрів, вибір моделі і алгоритму навчання;
- навчання моделі;
- аналіз якості навчання, якщо незадовільний перехід на п. 3 або п. 4;
- аналіз виявлених закономірностей, якщо незадовільний перехід на п.1, 3 або 4.

Всі етапи життєвого циклу представлені у вигляді схеми на рисунку 2.2.

Можливе переміщення вперед та назад між фазами. Залежно від результату фази або її підзадачі приймається рішення, до якої фази переходити далі. Стрілки показують найбільш важливі і часті переходи між фазами.

Зовнішнє коло символізує циклічну природу аналізу даних. Процес аналізу даних триває і після фази розгортання. Знання, отримані під час процесу, можуть породити нові більш тонкі питання бізнесу.

Подальший процес аналізу даних вигідно проводити, використовуючи знання, що були отримані раніше [6].

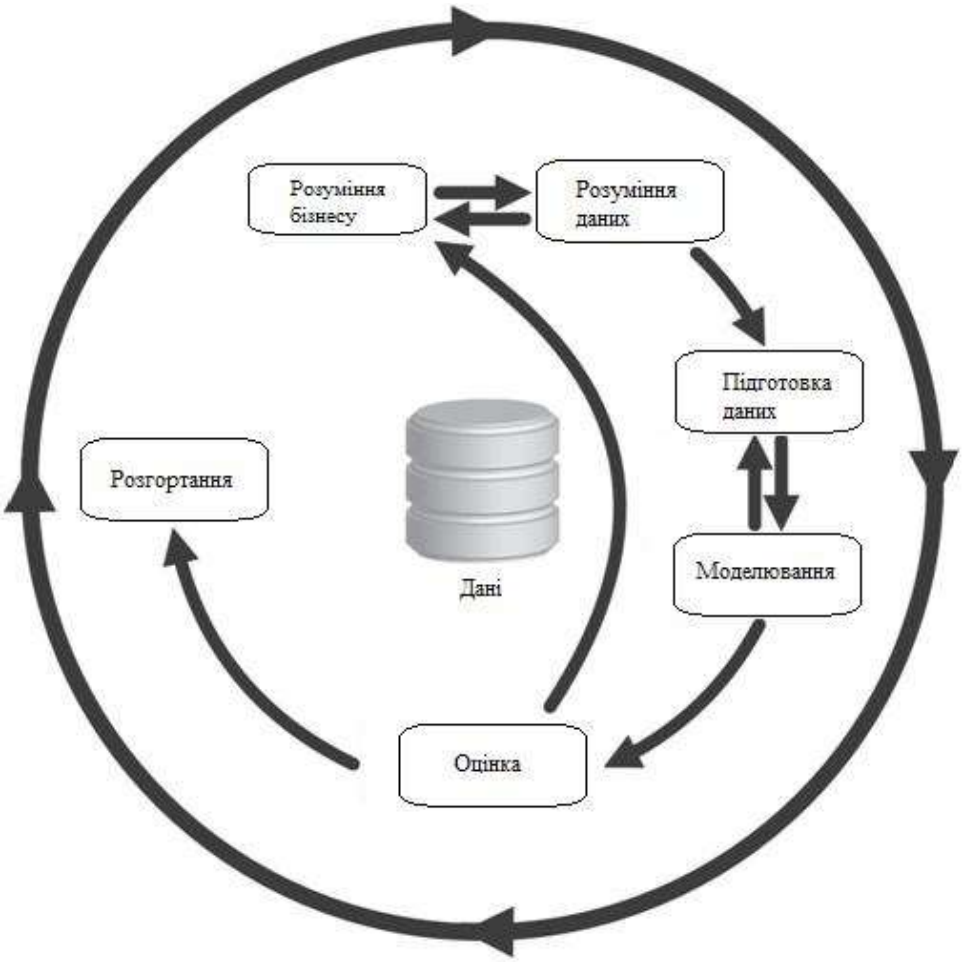

Рисунок 2.2 - Життєвий цикл дослідження даних CRISP-DM

2.4 Вибір інструмента для розробки моделі

Для роботи з даними було вирішено використовувати мову програмування Python і бібліотеку машинного навчання scikit-learn (sklearn).

В Python основний акцент робиться на продуктивності та можливості легко читати код. Досить велике коло програмістів використовує для аналізу даних і статистичних прийомів саме цю мову.

В Python є велика база бібліотек. Python має велику спільноту розробників, але вона дещо неоднорідна, оскільки мова універсальна.

Проте саме наука про дані стрімко займає все більш значні позиції в середовищі розробників, що використовують мову Python.

Однією з бібліотек для роботи з даними є sklearn. Бібліотека sklearn надає реалізацію цілого ряду алгоритмів для навчання з учителем і навчання без учителя. Незважаючи на те, що весь інтерфейс бібліотеки представлений на Python, але використання бібліотек, написаних на C, у внутрішній реалізації деяких частин scikit-learn, дозволяє значно підвищити швидкість роботи. Прикладом може бути використання NumPy для роботи з масивами і для операцій з матрицями або використання LAPACK і LibSVM. Для прискорення роботи використовується Cython. Бібліотека Scikit-learn поширюється під ліцензією "Simplified BSD License" і має релізи для безлічі різних операційних систем, включаючи MacOS і Windows. Розробники бібліотеки заохочують цим комерційне та академічне використання sklearn.

Дослідження показують, що час роботи інструментів Rapid Miner, Weka, мабуть через те що вони написані мовою Java, більше доби, тоді як час роботи інструментів Python і R - не більше години. Так як швидкість інструментів Python і R виявилася приблизно однаковою, то підставами для вибору стали більш звичними, R - функціональна мова програмування, де принципово інший підхід до програмування. В результаті залишилися два інструменти і основною особливістю бібліотеки sklearn є її гнучкість та можливість розширювати функціонал. Саме через свою швидкість роботи і гнучкість були вибрані технології Python і sklearn [8].

.

3 РОЗРОБКА МОДЕЛІ КЛАСИФІКАЦІЇ ВЕБ-СТОРІНОК

3.1 Визначення точності класифікації веб-сторінок

Проект, що розробляється повинен відповідати таким вимогам: отримана моделі повинна відносити веб-сторінку до однієї з 15 категорій з точністю передбачення більше 70%.

Для оцінки точності класифікації скористаємося крос-валідацією, де для розрахунку використовується матриця неточностей (таблиця 3.1).

Таблиця 3.1 - Матриця неточностей

		Вірні результати	
		1	0
Результат моделі	1	TP	FP
	0	FN	TN

У таблиці 3.1 міститься інформація про кількість вірно і невірно встановлених значень категорій:

- TP - істинно-позитивний;
- TN - істинно-негативний;
- FP - хибно-позитивний;
- FN - хибно-негативний.

Вимірювання точності класифікації виконується з використанням наступних метрик:

- accuracy;
- precision;
- recall;
- F1 score.

Метрика accuracy показує частку документів, за якими модель класифікації приймає правильне рішення.

$$A(\text{accuracy}) = \frac{tp + fp}{N}, \quad (3.1)$$

де tp - істинно-позитивний, fp - хибно-позитивний, N - розмір навчальної вибірки.

Метрика precision характеризує, скільки отриманих від моделі класифікації позитивних відповідей є правильними. Однак це не дає уявлення про те, чи всі правильні відповіді повернула модель класифікації.

$$P(\text{precision}) = \frac{tp}{tp + tn}, \quad (3.2)$$

де tp - істинно-позитивний, tn - істинно-негативний.

Метрика повноти recall характеризує здатність моделі класифікації «вгадувати» якомога більше число позитивних відповідей з очікуваних. Можна відзначити, що хибно-позитивні відповіді ніяк не впливають на цю метрику.

$$R(\text{recall}) = \frac{tp}{tp + fn}, \quad (3.3)$$

де tp - істинно-позитивний, fp - хибно-позитивний.

Precision і recall надають повну оцінку моделі класифікації. Зазвичай при побудові подібного роду систем доводиться весь час балансувати між цими двома метриками. Якщо підвищити recall, роблячи модель класифікації більш «оптимістичною», це призводить до падіння precision через збільшення числа хибно-позитивних відповідей.

Якщо ж змінювати модель класифікації, роблячи її більш «песимістичною», наприклад, суворіше фільтруючи результати, то зростання precision одночасно призведе до падіння recall через бракування якогось числа правильних відповідей.

$$F_1 = 2 * \frac{P * R}{P + R} \quad (3.4)$$

Метрика F1 score досягає свого максимуму 1 (100%), якщо P = R = 100%. Величина F1 score є однією з найпоширеніших метрик для

подібного роду систем.

В обчисленні F1 score для завдання класифікації є два основні підходи:

- сумарний F1 score: результати по всіх класах зводяться в одну єдину таблицю, по якій потім обчислюється метрика F1 score;
- середній F1 score: для кожного класу формується своя таблиця розподілу і своє значення F1 score, потім береться просте арифметичне середнє для всіх класів.

При класифікації веб-сторінок існує по кілька представників категорій з боку як забороненого, так і дозволеного контенту. В такому випадку метрика precision дозволить більш якісно оцінити модель, так як оцінка виконується за кількістю помилкових влучень, тобто необхідно виконати точну класифікацію для кожної з категорії, а всі інші відмітити як «невідомі» і в подальшому передати на перевірку експертам. При мультикласовій класифікації використовуються метрики усереднення macro-averaging і micro-averaging:

$$P_{micro} = \frac{\sum_{i=1}^{k} tp_i}{\sum_{i=1}^{k} (tp_i + fp_i)}, \quad (3.5)$$

де tp - істинно-позитивний, fp - хибно-позитивний, k – кількість класів.

$$P_{macro} = \frac{\sum_{i=1}^{k} \frac{tp_i}{tp_i + fp_i}}{k}, \quad (3.6)$$

де tp - істинно-позитивний, fp - хибно-позитивний, k – кількість класів.

Варіант macro-averaging дає кожному класу однакову вагу в результуючій метриці, а micro-averaging - кожному документу. За умови, коли вага класів однакова, з точки зору вартості помилки, має сенс використовувати macro-averaging. Інакше має сенс використовувати micro-averaging і додати більше документів цього класу в тестову вибірку. У

нашому випадку класи по вартості помилки не рівні, тому для оцінки скористаємося метрикою precision micro-averaging.

При навчанні моделі можлива ситуація, коли виходить досить складна модель, яка точно підходить під дані на навчання, але при цьому дає високу помилку на дані, що не входять в навчання. Уникнути подібної ситуації можна, якщо використовувати крос-валідацію.

Всю вибірку поділяють на дві підвибірки, які можна розбивати за різними принципами: навчальну і контрольну (тестову). Для отриманих підвибірок виконується навчання на алгоритмі машинного навчання, потім оцінюється і обчислюється середня помилка на примірниках контрольної підвибірки. Оцінкою крос-валідації є середня оцінка по всім підвибірках величина помилки на тестових підвибірках.

Якщо вибірка незалежна, то середня помилка крос-валідації дає незміщену оцінку ймовірності помилки. Це помітно відрізняє її від середньої помилки на навчальній підвибірки, яка може виявитися зміщеною (оптимістично заниженою) оцінкою ймовірності помилки, що пов'язано з явищем перенавчання.

До явища перенавчання можна віднести випадок, коли будується дуже складна модель, що дозволяє отримати відмінний результат на навчальній вибірці, але низьку точність при перевірці на тестовій вибірці. Таким чином при використанні крос-валідації можна оцінити рівень перенавчання, а точніше отримати більш правильну оцінку побудованої моделі.

Крос-валідація є стандартною методикою тестування і порівняння алгоритмів класифікації, регресії і прогнозування.

- повна крос-валідація (complete cross validation);
- випадкові розбиття (random cross validation);
- контроль на відкладених даних (hold-out cross validation);

- контроль за окремими об'єктами (leave-one-out cross validation або LOO CV);
- контроль за k блоками (k-fold cross validation);
- контроль за n × k блоками (n × k- fold cross validation).

Керуючись даними «Perception, Sensing & Instrumentation Lab» зазвичай використовується контроль за k блоками або контроль за окремими об'єктами в залежності від кількості вибірки. Відомо, що контроль за k блоками залежить від правильно обраного параметра «k», а при досить великій кількості вибірки краще використовувати контроль за окремими об'єктами [9].

3.2 Збір даних

Одним з найбільш важливих етапів для вирішення завдань за допомогою методів інтелектуального аналізу даних є збір навчальної та тестової вибірки. Так як модель буде модифікуватися, необхідно зберегти дані, щоб вони були доступні для дослідження впливу тих чи інших параметрів. У нашому випадку зберігання в файлі буде не таким зручним, так як необхідно буде використовувати відношення, а в базах даних це дуже просто реалізовано. При зборі бажано зберігати «сирі» дані, в нашому випадку це вихідний код веб-сторінок. Спочатку було задумано збирати не тільки цільові веб-сторінки, але і їх дочірні сторінки, які можна отримати за посиланнями на веб-сторінки. Виявилося, що це дуже трудомістка за часом задача. Так, якщо одна сторінка завантажується в середньому секунду, то якщо взяти тисячу веб-сторінок з десятьма дочірніми сторінками (посиланнями), отримується десять тисяч сторінок.

Перед тим, як почати збір даних, необхідно отримати доступ до ресурсу, звідки можна отримати дані. Найчастіше знайти «небажаний» або

заборонений контент можна тільки в тому випадку, якщо безпосередньо займатися подібною діяльністю.

Сайт urlblacklist.com використовується для SquidGuard. SquidGuard - програмний модуль (плагін) для проксі-сервера Squid, призначений для побудови системи фільтрації небажаного веб-контенту. SquidGuard працює за принципом "чорного списку", в якому перераховані небажані сайти, домени та їх ip-адреси. Якщо браузер намагається перейти на таку адресу, то автоматично спрацьовує перенаправлення на сторінку з попередженням або на інший заданий вами сайт. Недолік методу - потрібна підтримка "чорного списку" небажаних сайтів в актуальному стані, тому що постійно з'являються нові і нові сайти. Категорії, які розглядаються в якості заборонених представлені в таблиці 3.2.

Таблиця 3.2 - Заборонені категорії

Категорія	Зміст
alcohol	Інформація про алкогольні продуктах
drugs	Інформація про наркотики
gambling	Інформація про азартні ігри
adults	Порнографічний контент
smoking	Інформація про тютюнову продукцію
violence	Інформація про насилля
weapons	Інформація про зброю
terrorism	Інформація про тероризм
suicide	Інформація про суїцид

Також було отримано список дозволених категорій (таблиця 3.3).

Таблиця 3.3 - Дозволені категорії

Категорія	Зміст
news	Веб-сайти новин
sport	Спортивні веб-сайти
education	Веб-сайти навчальних установ
finance	Економіка і фінанси
shopping	Інтернет-магазини
whitePages	Контент для дітей

Для збору інформації необхідно розібратися в досліджуваній області. Типовий веб-сайт зазвичай складається з набору веб-сторінок, а веб-сторінки зазвичай являють собою текстові файли у форматі *.html. Веб-сторінки можуть містити посилання на інші файли в інших форматах, а також гіперпосилання. Гіперпосиланням називають частину гіпертекстового документа, що посилається на інший елемент в документі, на інший об'єкт або на елементи цього об'єкта.

Для створення HTML-сторінки використовуються HTML-теги. Кожен HTML-тег має своє призначення і дає інформацію браузеру про те, якйого слі двідображати. Крім HTML-тегів, використовуються ідентифікатори і класи, що дозволяють змінити відображення при використанні каскадних таблиць стилів (CSS). Зазвичай при верстці (написанні HTML коду) структура веб-сторінки розділяється на 3 блоки:

- шапка – header;
- контент – content;
- підвал - footer.

У кожному блоці знаходиться специфічна інформація щодо веб-сайту, така як назва сайту, основний текст і реквізити. Наприклад, зазвичай в якості нижньої частини (підвалу) веб-сторінки встановлюють клас або ідентифікатор «footer», де може знаходиться назва компанії, авторські

права, дублікат основного меню веб-сайту, контактна або коротка інформація про компанію.

Також вводяться теги, що являють собою збагачення семантичного змісту веб-сторінки, такі зміни призводять до уточнення типу мультимедіа об'єктів (відео, зображення, звук тощо), розширення універсальних блокових і текстових елементів для визначення інформаційного посилу інформації. Дизайнери веб-сайтів зазвичай усвідомлено використовують стандарт 5-ої версії HTML, тому що такий код простіше читати, а також спрощується його аналіз пошуковими роботами.

Для класифікації веб-сайтів необхідно враховувати безліч факторів, крім тестового контенту. Наприклад, порталам і сайтам новин притаманний матеріал різного змісту, тому на класифікацію впливають дані про батьківські та дочірніх сторінки. Також категорія сайту в цілому може бути визначена тільки за поєднанням єдиного домена. Це необхідно враховувати. Правильна і гнучка вибірка контенту необхідна для роботи, що пов'язана з класифікацією.

Для отримання даних з сайтів необхідно написати модуль. Інформація повинна бути розділена на безліч різних категорій, що дозволить спростити роботу з нею. Орієнтовна класифікація даних:

- посилання (текст в тегу «a» і посилання «href»);
- текст (весь текст сторінки);
- заголовки;
- спеціалізовані теги (meta, head, title);
- теги для конкретизації вибірки (strong, div, теги навігації);
- зображення (шлях «src», текст опису «alt»);
- тощо.

Завдяки переходу майже всього Інтернету на HTML5 з'явилася можливість розкладання сторінки на семантичні одиниці, що дозволить спростити класифікацію. Тому в якості додаткової вибірки буде ще кілька категорій:

- контент (вміст «div.content, div # content, article, section»);
- шапка (вміст «#header, .header, header»);
- підвал (вміст «#footer, .footer, footer»);
- навігація (вміст «.nav a, #nav a, #menu a, .menu a, ul a»).

Вибірка виконана з урахуванням використання стандартів як HTML4, так і HTML5 - при наявності необхідного класу або ідентифікатора тег ставиться за стандартом 4-й версії, інакше за стандартом 5-ої версії.

Тепер, коли нам відомі основні можливі групи контенту, можна розглянути варіант збереження отриманих даних. Так як модель буде модифікуватися, необхідно зберегти дані, щоб вони були доступні для дослідження впливу тих чи інших параметрів [1].

У нашому випадку зберігання в файлі буде не таким зручним, так як необхідно буде використовувати відношення, а в базах даних це дуже просто реалізовано. При зборі бажано зберігати «сирі» дані, в нашому випадку це вихідний код веб-сторінок. Так само спрощується пошук, сортування та організація даних. Простим запитом можна отримати необхідну вибірку.

Для зберігання отриманої інформації необхідно розробити структуру бази даних. Таблиці повинні зберігати інформацію про вміст сайту, а також про додаткові дані сторінки, такі як домен, батько і категорії.

Основна інформація буде зберігається в таблиці «pages»:

- url - адреса сторінки;
- html_code - код сторінки;
- html_text - текст сторінки;
- title - заголовок сторінки;
- meta_keyword - ключові слова;
- meta_description – опис;
- p_text - вміст параграфів;
- content - вміст блоку «контент»;

- header - вміст блоку «шапка»;
- footer - вміст блоку «підвал»;
- error - http код відповіді;
- domain_id - ключ на таблицю домену.

Для зберігання в базі даних можна скористатися структурою складається з таблиць:

- domains - таблиця з доменами;
- category_page - таблиця з категоріями веб-сторінок;
- categories - таблиця з категоріями;
- relatives - таблиця зі зв'язками між сторінками;
- pages - таблиця з веб-сторінками;
- tag_ * - таблиці за структурою і змісту ідентичні, де «*»: «a», «h1», «H2», «h3», «decorations», «div», «italic», «img», «nav», «bold», «u» та інші, що містять інформацію з відповідних їм тегами.

Схема представлена на рисунку 3.1.

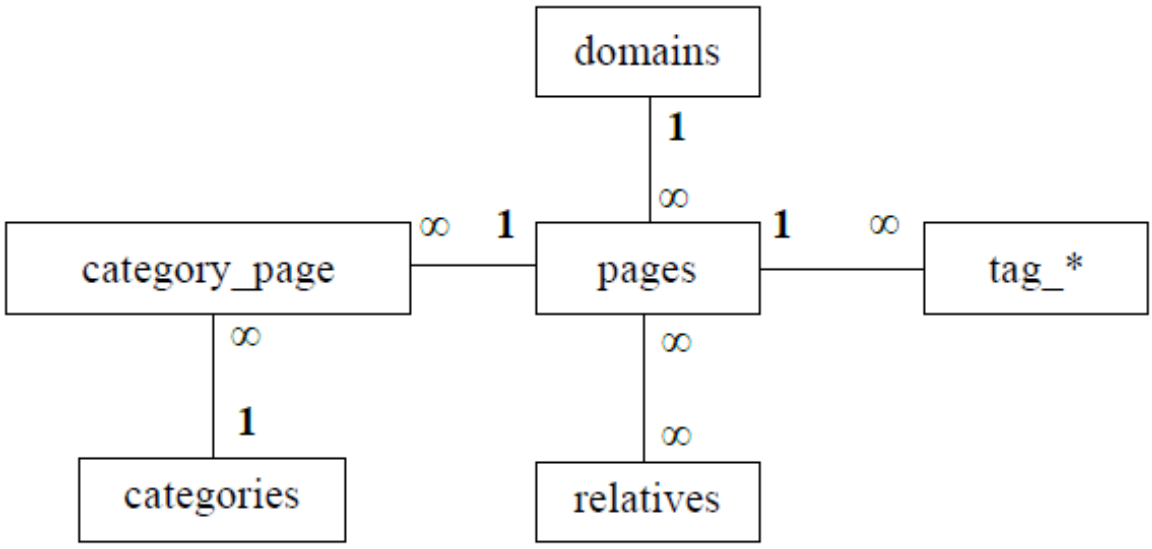

Рисунок 3.1 - Схема структури бази даних

У базі даних зберігаються дані всіх веб-сторінок по групах в окремих таблицях, домени, категорії, що використовуються для класифікації, відношення збережених веб-сторінок, код помилки веб-сторінки.

3.3 Навчання моделі

В даному розділі піде мова про методи та способи навчання моделей інтелектуального аналізу даних для класифікації веб-сторінок на основі бібліотеки scikit-learn.

Для роботи з sklearn необхідно встановити безпосередньо інтерпретатор мови програмування python і бібліотеки numpy, scipy. Для роботи з вибіркою бази даних - pandas. Роботу з методами обробки тексту можна здійснити з використанням бібліотеки NLTK. За установку бібліотек відповідає спеціалізований python менеджер - pip. Він дозволяє не тільки встановлювати необхідні бібліотеки, але також відслідковувати оновлення і залежні бібліотеки під час встановлення.

Працювати з python буде набагато зручніше в середовищі розробки. У нашому випадку буде використана інтерактивна консоль ipython notebook. Інтерфейс середовища представляє собою вікно браузера, в середовищі існує можливість створення в якості проектів Notebook. У проекті є форма для введення коду, але крім цього є можливість додавання декількох форм, кожна з форм є лише одним з етапів скрипта. Дані в даному випадку записуються в оперативну пам'ять, і при запуску наступної форми вона буде використовувати попередні результати обчислень.

Для підключення до бази даних скористаємося встановленим раніше пакетом pandas. Зазвичай робота на мовах програмування з базами даних організована з використанням спеціальних драйверів. В даному випадку

використовувалася PostgreSQL база даних, тому необхідно встановити пакет psycopg2.

Після установки драйверів можна приступити до установки з'єднання, де config - це dict з настройками для підключення:

```
conn = psycopg2.connect(host=config["hostname"],
                        user=config["username"],
                        password=config["password"],
                        database=config["database"])
```

Отримане підключення дозволяє працювати з базою, використовуючи курсор. Технічно це дуже незручний спосіб обробки даних в умовах інтелектуального аналізу даних. Тому скористаємося встановленим пакетом pandas, він сформує спеціальний об'єкт типу DataFrame з отриманих даних, над яким можна виконувати різні види операцій, такі як вибірка за умовою, обробка даних, обчислення агрегованих функцій і ін.

3.4 Підготовка даних

Так як в нашому випадку в якості даних виступає текст, то для підготовки даних можна розглядати фільтрацію і векторизацію. Фільтрація тексту виконується з метою зменшення шуму, її можна реалізувати двома методами:

Приведення до загального вигляду - серед різнорідних слів, схожих за коренем або іншими властивостями, слова вирівнюються і спрощуються, тим самим зменшується їх унікальність. Існує ряд позитивних і негативних сторін, які залежать від реалізації методу. Можливе формування урізаних слів, які не будуть до обробки мати нічого спільного, але після стануть ідентичні за написанням. З іншого боку, якщо

недостатньо сильно фільтрувати, то можна отримати ряд різних за написанням слів, але ідентичних за змістом.

Видалення слів з низьким смисловим навантаженням - даний метод розглядає використання списку заготовлених слів для видалення, звичайно на видалення відправляються прийменники і артиклі, а також слова зміст яких не впливає на роботу моделі класифікації. Дані методи можуть використовуватися як окремо, так і спільно.

Якість роботи залежить багато в чому від їх реалізації в залежності від завдання і даних.

Для роботи з текстом в бібліотеці sklearn прийнято його оцифрувати і представити у вигляді вектора (векторизувати), для цього є три класи HashingVectorizer, CountVectorizer, TfidfVectorizer.

Навчання з вчителем в бібліотеці scikit-learn виконується по єдиному інтерфейсу. Необхідно розділити вибірку, що навчається, на атрибути (ознаки, властивості або features) і мітки (категорії, класи або label), зазвичай в математиці позначаються X, y відповідно:

```
X, y = data.text, data.categories
```

Для виконання навчання моделі з використанням алгоритму Random Forest, скористаємося крос-валідацією k-fold з параметром k = 5. Число вибрано з міркувань оптимальності за часом обробки і одержуваної помилки точності. У sklearn представлено кілька реалізацій методу: KFold, StratifiedKFold, LabelKFold. Найбільше підходить StratifiedKFold, тому що алгоритм являє собою розподіл вибіркина підвибірки, що містять кожну з категорій.

Почнемо з фільтрації тексту за наступним алгоритмом:

1) привести всі символи до одного регістру і видалити «не слова»;

2) виключити загальні слова (стоп-слова);

3) провести стемінг і лематизацію;

4) вказати шаблон токенізаціі (розбиття тексту на слова - токени) і модель n-грами слів (кількість можливих слів в токені).

Крок за кроком виконуємо фільтрацію, а також навчання з крос-валідацією за метрикою precision_micro.

Необхідно привести всі символи до одного регістру методом «lower()». Для видалення всіх «не слів» скористаємося регулярними виразами, в такому випадку шаблон виглядає так: «[^ a-zA-Z]». Точність моделі з використанням такого фільтра: 0.660.

Лематизація не дозволяє підвищити точність як спільно зі стемінгом, так і окремо. Якщо використовуємо тільки стемінг, то отримаємо модель з точністю: 0.665.

Реалізуємо можливість фільтрації по стоп-словами. Для цього скористаємося вже існуючими списками від Google, MySQL і Word Analytics. Виконаємо об'єднання списків, при цьому прибравши всі перетини і збіги. Для фільтрації по стоп-словам існує функція в бібліотеці sklearn, скористаємося нею і подивимося на отриманий результат. Точність в такому випадку збільшилася до 0.690.

Якщо поглянути на роботу пошуку і видалення стоп-слів з тексту, то можна звернути увагу на те, що багато слів зі списків залишаються в тексті через їх формозміни, наприклад, «sites», де додавання закінчення не змінює кардинально сенсу слова, а значить, його можна було б видалити.

Ще якщо розглянути матрицю векторів, можна звернути увагу на існування спільнокореневих слів, які також об'єднані за змістом. В такому випадку введемо фільтр для очищення закінчень і перевіримо, що з цього вийде. Для фільтрації простіше використовувати також регулярні вирази, але фільтрація буде виконуватися вже окремо від pipline sklearn, шаблон для очищення буде виглядати так: «ing | ly | ed | ious | ies | ive | es | s | ment». В такому випадку збільшуємо точність до 0.698.

Отримана точність моделі недостатня відповідно до технічного завдання. На точність класифікації дуже сильно може вплинути

застосований алгоритм машинного навчання, тому для поліпшення можемо замінити алгоритм класифікації і ще раз протестувати модель.

3.5 Застосування алгоритмів машинного навчання

Скористаємося раніше описаними алгоритмами машинного навчання на отриманих даних. Всі алгоритми представлені в бібліотеці scikit-learn для використання.

В результаті:

- kNN - 0.432
- SVM - 0.725
- Logistic Regression - 0.758
- Decision Tree - 0.645

Підбір параметрів в цілому можна назвати емпіричним. Точність моделі класифікації залежить як від векторизації, так і від алгоритму класифікації.

У більшості випадків рішення-переможці на Kaggle представляють собою лінійну комбінацію декількох алгоритмів. Ґрунтуючись на цьому, можна застосувати композицію алгоритмів для проектованої системи.

У машинному навчанні існує кілька алгоритмічних композицій: bagging, boosting, blending, stacked generalization.

Для композиційних алгоритмів використовуються комбінаційні правила: algebraic combiners (mean rule, sum rule тощо.), voting based methods (majority plurality / voting, weighted majority voting), Borda count, Dempster-Schafer rule і ін.

Використання комбінації алгоритмів на основі blending дозволило досягти кращого результату серед багатьох інших алгоритмів. Основна

ідея blending в об'єднанні кількох алгоритмів в один: якщо існують два алгоритму a1 (x) і a2 (x), то їх комбінація представляє собою:

$$a\ x\ =\ \alpha a1\ x\ +\ \ 1-\alpha\ \ a2(x);\alpha\ \in [0,1], \qquad (3.7)$$

де параметр α вибирається за результатами крос-валідації.

Спробуємо поліпшити модель класифікації, використавши мета-алгоритми (ансамблі). Кращий результат показав алгоритм Bagging з Random Forest: 0.728.

Завдання мультикласової класифікації, передбачають використування стратегією «one versus rest». Принцип такої стратегії полягає в навчанні окремих моделей класифікації для кожного з класів і отриманні максимального з них по ймовірності відношення до категорії. Така стратегія також реалізована в бібліотеці sklearn. В результаті точність моделі: 0.763.

На даному етапі можно зробити висновок, що застосування Random Forest алгоритму спільно з алгоритмом Bagging і стратегією «one versus rest» дозволяє отримати найкращий результат.Була отримана модель класифікації з точністю, яка підходить встановленим вимогам [10].

4 РОЗРОБКА МЕТОДІВ ПІДВИЩЕННЯ ТОЧНОСТІ КЛАСФІКАЦІЇ

4.1 Методи збільшення точності класифікації веб-сторінок

Збільшимо точність класифікації веб-сторінок, використовуючи існуючі методики і розробивши власні. Для цього проведемо дослідницьку роботу, присвячену класифікації веб-сторінок на основі методів і алгоритмів інтелектуального аналізу даних.

Одним з найважливіших етапів в підготовці даних на навчання є векторизація. Основними параметрами в цьому випадку виступають алгоритм отримання атрибутів, кількість слів (n-gram) і максимальна кількість атрибутів. У бібліотеці sklearn присутні кілька реалізацій векторизації даних:

- HashingVectorizer (HV) - перетворює вхідний текст в вектор зі значеннями кількості входження слова в текст.
- TfidfVectorizer (TF) - перетворює в вектор зі значеннями відношення числа входження слова до загальної кількості слів документа.

В якості додаткового параметра в TF-IDF можна використовувати сублінійну функцію (SB, sublinear), яка замінює стандартний підрахунок TF і дозволяє прибрати «звичайні» слова з розрахунків. Результати представлені на плакаті №6.

Проаналізувавши отримані результати, можна зробити висновок, щонайвищу точність дає використання векторизації з TF-IDF+ sublinear з 5000 атрибутів і n-gram (1, 2).

При скачуванні даних з мережі Інтернет існує проблема в тому, що є досить багато даних і визначити, чи вірно були промарковані веб-сторінки, чи не скінчилася оренда домену тощо, досить складно. Тому було вирішено провести фільтрацію веб-сторінок за «правильно класифікованими». Для цього векторизуємо всю вибірку, а потім

навчаємося на отриманому векторі і робимо прогноз по ньому ж. У цього методу є шанс того, що буде перенавчання, але наявність невірно передбачених менше 5%. Тому приймемо їх за недоступні веб-сайти або категорії, які були спочатку з помилкою і відкинемо їх з вибірки на навчання.

Також стверджується можливість використання PCA (метод головних компонент). Це дозволяє виконати зменшення аналізованої безлічі даних до розміру, який буде оптимальним з точки зору розв'язуваної задачі. Даний метод використовується в якості підготовки даних перед класифікацією.

Найважливішим фактором в інтелектуальному аналізі даних є правильне використання атрибутів. На веб-сторінках можна як атрибути вибрати теги, наприклад, «title», «a», «meta» і блоки «header», «content» і «footer», які були отримані селекторами по тегу, id і class. Так як на цей текст був акцент з боку розробників веб-сайту, то можливо роль їх набагато вища.

Використання окремих моделей класифікації для кожного з атрибутів не дає підвищення точності. Спробуємо поєднати їх з існуючим, тим самим збільшивши кількість входжень ключових слів. В результаті досвідченим шляхом обчислено, що теги «title» і «meta» (description, keywords) збільшують точність.

Ще одна складність класифікації полягає у визначенні приналежності тексту, який може належати одночасно до безлічі категорій, але при цьому ні до однієї з наявних, наприклад, «новини».

Спочатку додамо категорію новини. Виконаємо пошук слів високою вагою і використовуємо в якості ключових слів: «news, finance, sport, political, politics, health, tech, technology, culture, art, weather, ecomony, business, lifestyle, world, national, travel, celebrity, movies , music, fashion».

Використання ключових слів не нове і майже нічим не відрізняється від того, як визначаються інші категорії. Тому введемо критерій оцінки і

підрахунку - робити обчислення при наявності хоча б двох входжень основного ключового слова, в даному випадку «news». Таким чином у нас є два рівня ключових слів, де на першому рівні знаходиться основне, а на наступному - інші, за якими виконується вже перевірка. Для збільшення відсотка відповідності використовуємо багатовимірний вектор стоп-слів з використанням синонімів - це дозволяє отримати більш коректний відсоток входження слів без підрахунку слів з однаковим змістом [1].

4.2 Метод ієрархічної класифікації

Стратегія «one versus rest» дозволила поліпшити точність моделі. Тому використовуємо схожий метод з різницею в тому, що ми можемо використовувати моделі з окремо підібраними параметрами і алгоритмами. Для цього використаємо поодинокі бінарні моделі класифікації, що навчені тільки під свої категорії. Для отримання підсумкової категорії використовуємо метод голосування. Підвищити точність також вдалося за рахунок округлення вихідних даних моделей класифікації.

Метод голосування є не найкращим, тому спробуємо його поліпшити. В якості заміни скористаємося ще однією моделлю класифікації, яка буде навчена за результатами попередніх. В такому випадку необхідно розділити навчання на кілька рівнів. Для початку проводиться розподіл даних на перший (L_1) і другий (L_2) рівень. Вибірка ділиться при повному змішуванні порядку (дозволяє більш коректно визначати точність), при цьому на кожному рівні присутнє рівне співвідношення даних з кожної категорії. Першому рівню встановлюються категорії в бінарному вигляді (0, 1) відповідно до приналежності. При навчанні L_1 (атомарних моделей класифікації) навчаються по кожній категорії з L_1. При навчанні L_2 (рефері) використовуються результати

атомарних моделей класифікації (відсоток збігу з категорією). У підсумку на першому рівні ми отримуємо прогноз по кожній категорії, а потім на другому рівні рефері видає остаточне рішення по ним.

Дана методика дозволяє зв'язати кілька моделей класифікації, навчених на різних атрибутах з різними алгоритмами. Використовуючи раніше отримані атрибути, можна побудувати модель для класифікації. Таким чином є можливість вибрати найбільш підходящі алгоритми і параметри для кожної з категорії і для рефері [11].

4.3 Метод класифікації за допомогою «сусідніх» веб-сторінок

Гіпертекстові особливості, такі як посилання, також можуть бути використані в якості атрибутів. Найпростішим способом використання посилання є набір дочірних сторінок в якості атрибутів для навчання, тому спочатку розглянемо можливість їх використання.

Для того щоб істотно скоротити часові витрати, можна спробувати обмежити кількість скачуваних дочірніх сторінок. Виконувати збір шести випадкових посилань, де дві з них знаходяться на початку сторінки, в середині та інші в кінці. Для початку проведемо класифікацію по новій вибірці з використанням тільки цільової веб-сторінки. Далі виконаємо класифікацію по дочірнім сторінкам, отримані результати повинні бути нижче через наявність викидів, які обумовлені не тільки змістом різнорідних категорій, а й помилкою при класифікації.

Проводячи класифікацію тільки за контентом цільової веб-сторінки, крім результатів прогнозів, також можна підраховувати відсоток упевненості. Це дозволить розділити веб-сторінки за результатами на групи з високим (> ~ 70%) і низьким відсотком впевненості. Розділивши вибірку на ці групи, розглянемо взаємозв'язок з класифікацією по дочірніх сторінках.

Для веб-сторінок з високою впевненістю дочірні сторінки найімовірніше будуть збігатися з цільовою, як для вірно, так невірно класифікованих.

При класифікації веб-сторінок з низькою упевненістю дочірніх сторінок можливість збігу не так очевидна. Проведемо класифікацію по дочірніх сторінках тільки для веб-сторінок з низькою упевненістю, а по цільовим - з високою. Даний метод може дозволити підвищити точність і використовувати новий вид атрибутів, але точність і швидкість отримання результату залежить від кількості дочірніх сторінок.

Зазвичай веб-сторінки з більш складним контентом для класифікації, наприклад, «news» не дають позитивного результату через вміст змішаного набору категорій. Також як і цільові веб-сторінки, що містять зображення, не дозволяють отримати дочірні сторінки. Тому розглянемо можливість використання сусідніх веб-сторінок.

Розглянемо схему того, як можуть бути побудовані зв'язки між веб-сторінками, використовуючи посилання (рис. 4.1).

На схемі розташовані веб-сторінки зі зв'язками на двох рівнях (radius 1-2) в залежності від їх дистанції щодо цільової веб-сторінки (Target Page). На першому рівні цільова веб-сторінка може мати батьківську (Parent) і дочірні (Child) веб-сторінки.

На другому рівні Parent веб-сторінки можуть мати батьківські (Grandparent) і дочірні (Sibling), аналогічно Child веб-сторінки мають дочірні (Grandchild) і батьківські (Spouse).

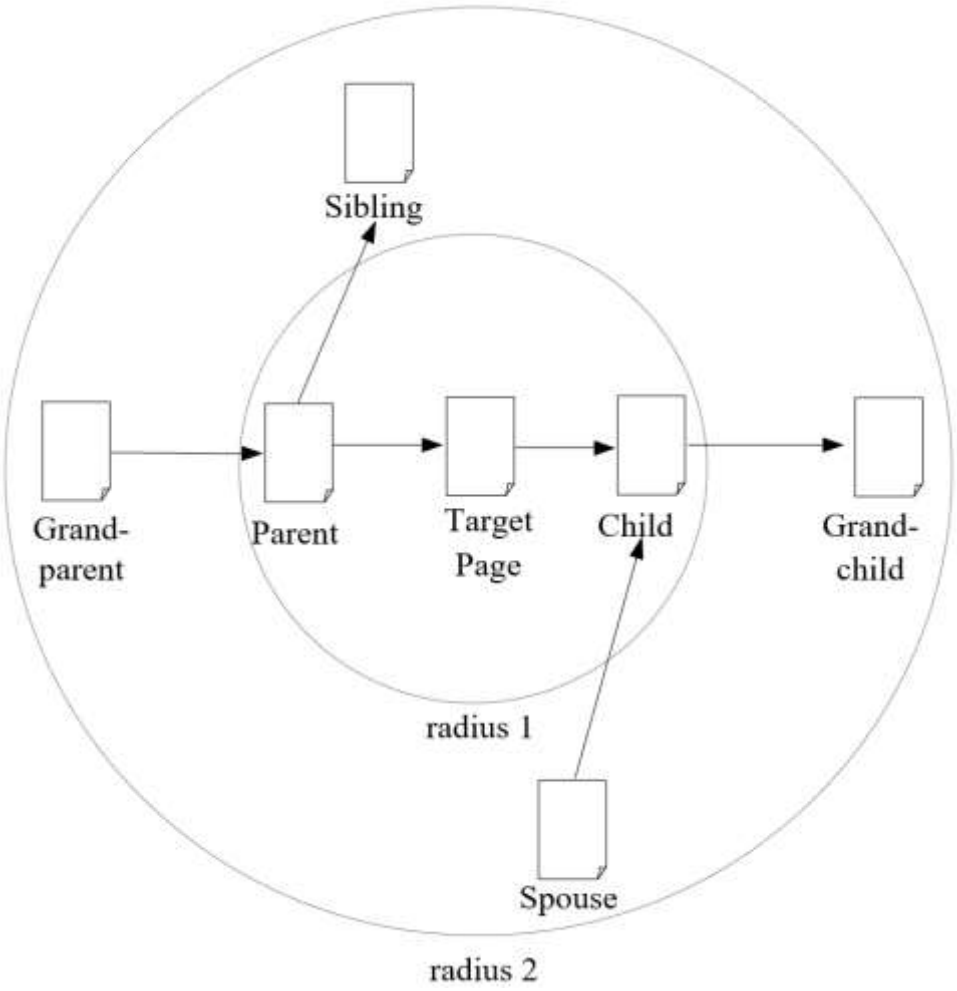

Рисунок 4.1 - Представлення зв'язків між веб-сторінками

На практиці отримати вибірку за подібною схемою можна лише з використанням пошукового робота для збору великої кількості веб-сайтів різних тематик. Такий підхід є практично неможливим за короткий проміжок часу, в той час як інформація, а точніше веб-сторінки з'являються і змінюються щомиті. Практичне застосування даного методу залишається під питанням.

Для отримання батьківських сторінок скористаємося доступними «Backlink» сервісами. Розглядаючи методи скачування, з'явилася ідея підрахунку посилань. Списки відомих доменів веб-сайтів з категоріями можна використовувати в якості атрибутів. Таким чином ми отримуємо вектор, в якому атрибути - колонки з кількістю посилань на категорію [11].

4.4 Оцінка ефективності запропонованих методів

Розглянуті методи були перевірені на всіх доступних алгоритмах класифікації в бібліотеці sklearn.

Методи, засновані на класифікації «сусідніх» веб-сторінок, не дозволяють отримати очікуване поліпшення точності. Можливо, це пов'язано з тим обмеженим набором «сусідніх» веб-сторінок, який використовувався в даному дослідженні. Також процес класифікації, при використанні «сусідніх» веб-сторінок займає чимало часу через необхідність їх скачування. Результати проведеної роботи представлені на плакаті №7.

Якщо розглядати процес класифікації по текстовим даними цільової веб-сторінки, то, як показала практика, одним з найважливіших факторів є використання якісної вибірки, яка не буде містити порожніх або невірно промаркованих веб-сторінок. Це було показано після навчання моделей класифікації на відфільтрованої вибірці по «правильно передбаченим».

Проведені дослідження показують, що найбільш простими ефективним методом класифікації, що були досліджені в даній роботі, є класифікація на основі ієрархічної моделі з використанням бінарних моделей класифікації з рефері.

ВИСНОВКИ

В ході роботи було проведено аналіз існуючих методів класифікації веб-сторінок, який дозволив зробити висновок, що такі методи класифікації у повній мірі не задовольняють сучасним вимогам точності та повноти класифікації.

Проведені дослідження показують,що найбільш простимі ефективним способом класифікації, що досліджувалися у даній роботі, є класифікація на основі ієрархічної моделі з використанням бінарних моделей класифікації з рефері. Варто зауважити, що як показала практика, в процесі класифікації одним з найважливіших факторів є використання якісної вибірки, яка не буде містити порожніх або невірно промаркованих веб-сторінок. Такий висновок зроблено післянавчання моделей класифікації на від фільтрованої вибірці по «правильно передбаченим».

На основі виконаних досліджень були розроблені нові методи підвищення точності класифікації веб-контенту на основі існуючих , які дозволяють виконувати класифікацію веб-сторінок з точністю 96%.

Також виявлено, що найбільш складно класифікувати веб-сторінки, які не містять тексту. Оскільки зазвичай людина оцінює вміст веб-сторінки на основі зображень, то таких сторінок досить багато. В майбутньому можливе додавання атрибутів такого типу, що допоможе поліпшити якість класифікації. Тому одним з можливих напрямків подальших досліджень може бути використання нових атрибутів, що базуються на зображеннях.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Abdallah T. A., Iglesia B. URL-based web page classification-a new method for URL-based web page classification using n-gram language models // SCITEPRESS Digital Library-KDIR 2014-Intern. conf. on Knowledge Discovery and Information Retrieval. Rome, Italy, 2014.
2. Belmouhcine A., Benkhalifa M. Implicit Links based Web Page Representation for Web Page Classification // Proc. of the 5th Intern. conf. on Web Intelligence, Mining and Semantics. Larnaca, Cyprus, 2015.
3. Чубукова И. А. Data mining - Електрон. дані (1 файл) – Режим доступу: http://lnfm1.sai.msu.ru/~rastor/Books/Chubukova-Data_Mining.pdf (дата звернення 20.09.2021).
4. Солонин Е.Б. Интеллектуальные технологи поиска и анализа данных - Електрон. дані (1 файл) – Режим доступу: http://www.study.urfu.ru/Aid/Publication/13334/1/Solonin.pdf (дата звернення 15.10.2021).
5. Knowledge Discovery in Databases – обнаружение знаний в базах данных - Електрон. дані (1 файл) – Режим доступу: https://basegroup.ru/community/articles/kdd (дата звернення 15.10.2021).
6. CRISP-DM // MachineLearning.ru. Профессиональный информационно-аналитический ресурс - Електрон. дані (1 файл) Режим доступу: http://www.machinelearning.ru/wiki/index.php?title=Crisp-dm (дата звернення 10.09.2021)
7. Завдання Data Mining - Електрон. дані (1 файл) – Режим доступу: http://studopedia.com.ua/1_11366_zavdannya-data-mining.html (дата звернення 20.09.2021).

8. Python - Електрон. дані (1 файл) – Режим доступу: https://ru.wikipedia.org/wiki/Python (дата звернення 16.11.2021).
9. Оцінка класифікатора (точність, повнота, F-міра) - Електрон. дані (1 файл) – Режим доступу: http://bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html (дата звернення 30.10.2021).
10. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks // Information Processing & Management. 2009.
11. Qi X., Davison B.D. Web page classification: Features and algorithms // Journal ACM Computing Surveys. 2009.
12. Data mining - Електрон. дані (1 файл) – Режим доступу: https://ru.wikipedia.org/wiki/Data_mining (дата звернення 10.10.2021).
13. TF-IDF - Електрон. дані (1 файл) – Режим доступу: https://ru.wikipedia.org/wiki/TF-IDF (дата звернення 13.10.2021).
14. Web mining - Електрон. дані (1 файл) – Режим доступу: https://ru.wikipedia.org/wiki/Web_mining (дата звернення 13.01.2021).
15. Інтелектуальний аналіз даних. Класифікація і регресія - Електрон. дані (1 файл) – Режим доступу: http://ukrbukva.net/print:page,1,4473 intellektual-nyiy-analiz-dannyh-klassifikaciya-i-regressiya.html(дата звернення 30.09.2021).
16. Построение модели и алгоритма кластеризации в интеллектуальном анализе данных - Електрон. дані (1 файл) – Режим доступу: https://cyberleninka.ru/article/v/postroenie-modeli-i-algoritma-klasterizatsii-v-intellektualnom-analize-dannyh (дата звернення 12.11.2021).