НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

«До захисту допущено»

В.О.Завідувача кафедри

_____ О.Л. Тимощук

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.040303 Системний аналіз

на тему: «Система аналізу та категоризації текстових медичних даних з використанням SAS технологій»

Виконав:
студент (-ка) IV курсу, групи КА-53
Юрчук М.В. _____

Керівник:
доцент, к.т.н.
Терентьєв О. М. _____

Консультант з економічного розділу:
доцент, к.е.н.
Шевчук О. А. _____

Консультант з нормоконтролю:
доцент, к.т.н.
Коваленко А.Є. _____

Рецензент:

_____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студент (-ка) _____

Київ – 2019 року

РЕФЕРАТ

Дипломна робота: X с., 2 табл., рис., 2 дод., 21 джерело

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, ІНФОРМАЦІЙНИЙ ПОШУК, ТЕКСТОВА АНАЛІТИКА, МЕДИЦИНА, ОБРОБКА ПРИРОДНЬОЇ МОВИ, КЛАСТЕРИЗАЦІЯ, ХМАРНІ ОБЧИСЛЕННЯ.

Актуальність роботи: на сьогодні більша частина інформації знаходиться у неструктурованому вигляді, тому використання її звичними аналітичними моделями являється неможливим. В медичній сфері, обробляючи текстові дані, можливо значно покращити якість зворотнього зв'язку з пацієнтами, а тому і якість препаратів.

Об'єкт дослідження: медичні текстові звіти, а саме – відгуки пацієнтів.

Предмет дослідження: інформаційний пошук, інтелектуальний аналіз, метод максимальної правдоподібності, латентно-семантичний аналіз, булеві правила.

Мета роботи: дослідження існуючих методів обробки неструктурованих текстових даних та їх впровадження у системі аналізу та категоризації текстової медичної звітності.

Метод дослідження: використання знань про обробку природньої мови (NLP), математичних методів та моделей для класифікації та кластеризації текстової інформації.

Результати роботи: проведено аналіз сучасних методів інформаційного пошуку, досліджені можливості використання наявних інструментів текстової аналітики, налаштування їх під конкретну галузь, розроблена система, яка дозволяє створювати категорії з певним ступенем вірогідності виконувати задачу класифікації, отримуючи на вхід велику кількість медичних звітів.

Шляхи подальшого розвитку предмета дослідження - удосконалення обраної архітектури моделі, розширення колекції текстів, розмітка даних експертами.

ABSTRACT

Thesis: X p., X fig., X tabl., X append., X sources

The theme: “The system for analysis and categorization of textual medical data using SAS technologies”.

INTELLECTUAL ANALYSIS OF DATA, INFORMATION SEARCH, TEXT ANALYTICS, MEDICINE, NATURAL LANGUAGE PROCESSING, CLUSTERIZATION, CLOUD COMPUTING.

Relevance of work: for today most of the information is unstructured, so using it with the usual analytical models is impossible. In the medical field, processing text data can greatly improve the quality of feedback from patients, and therefore the quality of drugs.

Object of research: medical text reports, namely patient reviews.

Subject of research: information search, intellectual analysis, method of maximum likelihood, latent semantic analysis, boolean rules.

Purpose: to study existing methods of processing unstructured text data and their implementation in the system of analysis and categorization of text medical reporting.

Research method: use of knowledge about natural language processing (NLP), mathematical methods and models for classification and clustering of text information.

The results of the work: the analysis of modern methods of information search, the possibilities of using existing tools of text analytics, their adjustment for a specific branch, a system developed that allows creating categories with a certain degree of probability to perform the classification task, receiving a large number of medical reports on the entrance is investigated.

Ways of further development of the subject of research - improvement of the chosen architecture of the model, expansion of the collection of texts, markup by experts.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ТА – текстова аналітика

ТМН - технології машинного навчання

КД - класифікація документів

ПП - програмний продукт

ПК – персональний комп'ютер

ОПР - особа, яка приймає рішення

NLP (Natural Language Processing) - обробка природньої мови

SAS - компанія, що розвиває передові технології в області машинного навчання

CAS (Cloud Analytical Services) – технологія оптимізованих хмарних обчислень

SVM (Support Vector Machine) - метод опорних векторів

VTA (Visual Text Analytics) – інструмент SAS для обробки тескстуальних даних

LSA (Latent Semantic Analysis) – латентно-семантичний аналіз

SVD (Singular Value Decomposition) – метод сингулярного розкладу

PCA (Principal Component Analysis) – метод головних компонент

REGEX (Regular Expressions) – регулярні вирази

ВСТУП

Аналіз інформаційних даних використовується всюди ОПР для прийняття більш зважених рішень. У всьому світі спостережується експоненціальний зріст кількості інформації. Водночас, значна частина даних збирається у неструктурованому вигляді, що великим чином обмежує її подальше використання у аналітичних моделях. Текстові дані можуть нести важливу інформацію, наприклад відгуки та коментарі клієнтів, але через наявність у тексті людських помилок, складних зв’язків між словами, - опрацювання текстів перетворюється у серйозну задачу для машини.

Процес зворотнього зв'язку між пацієнтом та фармацевтичними компаніями довгий час відбувався шляхом звичайного анкетування, яке займає багато часу, грошових затрат та людського ресурсу. Автоматизація цього процесу дає можливість у реальному часі аналізувати потреби пацієнтів, побічні дії, що зустрічаються частіше з використанням препарату. Таким чином, доцільно впровадити систему аналізу та категоризації, як розробникам препаратів, для їх покращення, створення нових ліків, так і для незалежних дослідників, що мають на меті спостереження трендів у лікуванні певних захворювань.

Для вирішення цієї проблеми розробляються математичні методи та ПП. SAS активно розвиває свої рішення, пропонує передові технології і вже довгий час являється серйозним гравцем у сфері машинного навчання та обробки великих даних.

Таким чином, метою данної роботи є дослідження методів інформаційного пошуку та інтелектуального аналізу для подальшого

застосування їх для обробки текстових медичних відгуків від пацієнтів під час лікування.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Медична галузь, як і багато інших, постійно потребує швидкого опрацювання “зворотнього зв'язку” для того, щоб покращувати існуючі ліки або створювати нові.

Компаніям, що виготовляють лікарські засоби, дуже важливо мати точні дані про використання препаратів. Зазвичай “зворотній зв'язок” отримувався після анкетування людей та опрацьовувався працівниками компаній. Такий підхід займав дуже багато часу та дуже потребував вдосконалення.

З приходом інформаційних технологій, розвитком інтелектуального аналізу даних та розробкою інструментів текстової аналітики, цей процес вдалося значно пришвидшити та автоматизувати, що призвело до більш розробки більш якісних препаратів, зменшення вірогідності побічних дій та збільшення прибутку.

Використовуючи інструменти ТА, з’явилась можливість швидко проаналізувати відгуки хворих тим чи іншим захворюванням, зрозуміти які препарати частіше використовуються, які в них позитивні сторони, і які є недоліки.

1.1 Поняття інтелектуального та текстуального аналізу

Текстова аналітика допомагає аналітикам витягати змісти, патерни і закономірності, приховані в неструктурованих текстових даних.

ТА включає в себе інструменти і методи, які використовуються для отримання поглиблених знань про предмет аналізу з
допомогою неструктурованих даних. Ці методи можна класифікувати наступним чином:

- інформаційний пошук (informational retrieval);
- розвідувальний аналіз (exploratory analysis);
- виокремлення понять (concept extraction);
- реферування (summarization);
- категоризація (categorization);
- аналіз тональності тексту (sentiment analysis);
- управління контентом (content management);
- управління онтологіями (ontology management).

Серед цих методів розвідувальний аналіз, реферування і категоризація відносяться до інтелектуального аналізу тексту. Розвідувальний аналіз включає такі методи, як витяг тематик (topic extraction), кластерний аналіз та інші. Термін "текстова аналітика" в деякій мірі є синонімом терміна "інтелектуальний аналіз тексту" або "інтелектуальний аналіз текстових даних". Інтелектуальний аналіз тексту найкраще охарактеризувати як розділ текстової аналітики, сфокусований на застосуванні методів інтелектуального аналізу (data mining) до текстових даних з використанням обробки природної мови та машинного навчання. Інтелектуальний аналіз тексту зосереджений тільки на синтаксисі (вивченні структурних взаємозв’язків між словами). Він не стосується фонетики, прагматики і дискурсу. Аналіз тональності тексту можна розглядати як класифікаційний аналіз. Тому його вважають інтелектуальним аналізом тексту, призначеним для прогнозування (predictive text mining). З точки зору застосування описаних методів текстової аналітики

можна розділити на дві області: пошук та описову і прогнозну аналітику. Пошук включає численні методи інформаційного пошуку, тоді як описова і прогнозна аналітика включає інтелектуальний аналіз тексту і аналіз тональності тексту.

1.2 Історія розвитку сфери

Епоха інформації призвела до появи різноманітних інструментів і складної інфраструктури, призначеної для вилучення і зберігання величезних масивів текстових даних. У звіті 2009 року компанія International Data Corporation (IDC) підрахувала, що приблизно 80% даних, що зберігаються в організаціях, базуються на тексті. Окрема людина (або навіть група людей) не в змозі виділити сенс, тональності і патерни з великої кількості даних. Стаття, написана Хансом Пітером Луном і її названо "The Automatic Creation Of Literature Abstracts", є, можливо, однією з найбільш ранніх науково-дослідних робіт, присвячених текстової аналітиці. Лун (рисунок 1.1) писав про застосування машинних методів для автоматичного реферування документа. У традиційному розумінні термін "інтелектуальний аналіз тексту"
("text mining ") має на увазі автоматизоване машинне навчання і статистичні методи, які реалізують підхід bag-of-words. Цей підхід зазвичай використовується для порівняльного аналізу колекцій документів та окремих документів. З плином часу термін "інтелектуальний аналіз тексту" став позначати вільно інтегруються систему, яка запозичила методи інтелектуального аналізу даних (data mining), обробки природної мови (NLP),

інформаційного пошуку (informational retrieval) і управління знаннями (knowledge management).

Рисунок 1.1 – Ханс Пітер Лун

Текстова аналітика набирає популярність в бізнес-середовищі. Вона дозволяє отримати інноваційні та глибокі результати. ТА реалізується в багатьох галузях промисловості й щодня з'являються нові сфери застосування. В останні роки текстова аналітика більшою мірою використовувалася для пошуку трендів в текстових даних. Використовуючи дані соціальних мереж, текстова аналітика дозволяє запобігти злочинам та виявити шахрайство. У медицині текстова аналітика дозволяє підвищити результативність лікування пацієнтів і поліпшити медичну допомогу. Вченим у фармацевтичній галузі, зайнятим пошуком нових ліків, потрібен текстовий аналіз біомедичної літератури.

1.3 Огляд ринку ПП для задач текстуального аналізу

Оскільки проблема обробки неструктурованих даних залишається актуальною, багато передових компаній, що працюють в сферах інформаційних технологій та аналізу даних, вже мають свої розроблені інструменти для проведення аналізу текстуальних даних.

Аналіз ринку ПП цій сфері (рисунок 1.2) показав, що на 2019 рік компанії найбільш активно використовують Rapid Miner, Amazon Comprehend, Microsoft Text Analytics API, Google Cloud Natural Language, SAS та інші.

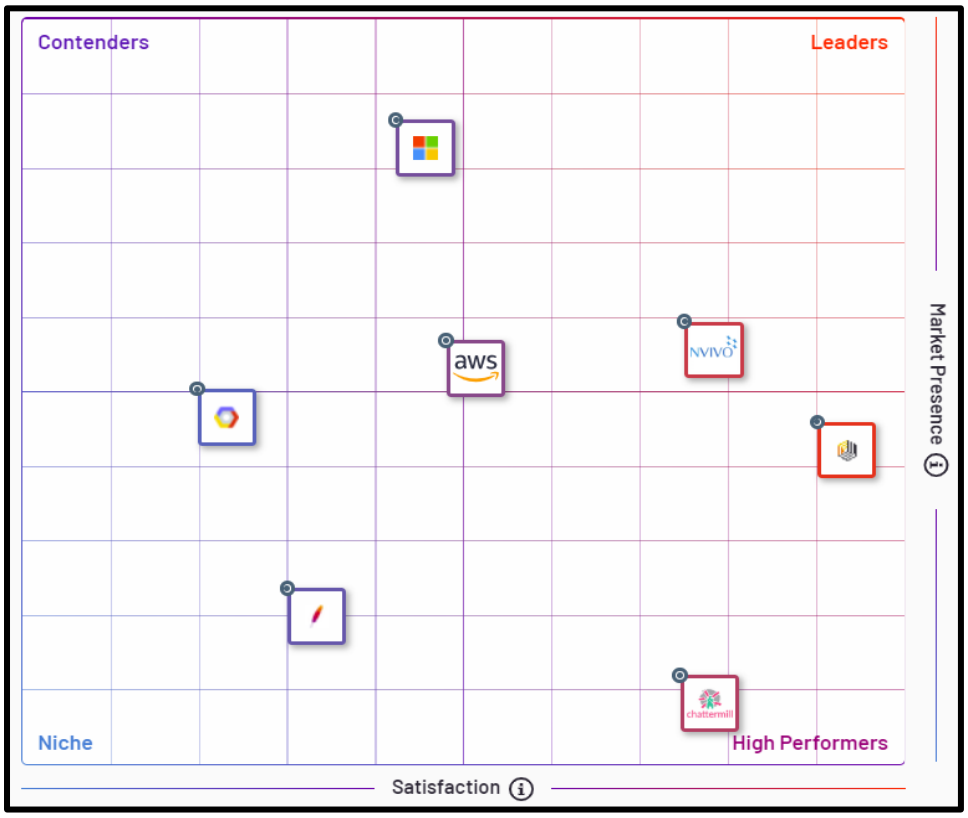


Рисунок 1.2 – аналіз ринку ПП на 2019 р.

1.3.1 SAS Viya Visual Text Analytics

Компанія SAS існує з 1976 року та займається розробкою програмних рішень у аналітичній сфері для великого, середнього бізнесу, а також для використання незалежними дослідниками.

Довгий час компанія створювала успішні ПП для настільних ПК, такі як SAS Enterprise Guide, SAS Enterprise Miner, але IT і бізнес спільнота стрімко розвиваються і стають більш вимогливим до інструментів, тому SAS випустила нову аналітичну платформу SAS Viya. Ця платформа включає в себе все краще, що було створено в компанії SAS з моменту виникнення до теперішнього часу, для того, щоб визначати сучасні тенденції класу рішень для просунутої аналітики. SAS Viya дає єдину платформу (рисунок 1.3) для такого напрямку як self-service data science с використанням можливостей in-memory, яка розроблена з використанням підходів розподілених (хмарних) обчислень і мікросервісної архітектури.

Попит народжує пропозицію і SAS Viya не виняток з правил. Якщо звернутися до офіційного визначення SAS Viya, яке було дано Джимом Гуднайта під час анонса в 2016 році на глобальному SAS форумі, то SAS Viya це: "Хмарна система, яка використовує підходи розподілених обчислень ... і дає єдину платформу для аналітики".

Ідея і цілі платформи SAS Viya - універсальна платформа для будь-якого виду аналізу на всіх стадіях проекту від підготовки даних до застосування складних алгоритмів машинного навчання.

Можна виділити 4 блоки завдань:

1. Підготовка даних;
2. Візуалізація та дослідження даних;
3. Прогнозна аналітика;
4. Просунута аналітика у вигляді алгоритмів машинного навчання.

В основі SAS Viya лежить новий унікальний метод обробки даних CAS (Cloud Analytics Service). Дві ключові особливості: перше це in - memory технологія, яка виконує всі операції з даними в оперативній пам'яті, а друге - це підхід розподілених обчислень. CAS може працювати на одному хості, але бути оптимізованим для роботи на кластері з машин - контролері і серверах обробки даних, які дозволяє зберігати і обробляти дані на різних вузлах кластера для розпаралелювання навантаження. Ідейно підхід дуже близький до концепції Hadoop систем.

Переваги концепції хмарних обчислень:

1. Доступність через великий набір API різних клієнтів. Для SAS це великий крок вперед. Немає обмежень на використання мови SAS Base для аналітики. Є можливість використовувати Python (Наприклад, з Jupiter Notebook), R, Lua і ін., виконання яких буде відбуватися в CAS на платформі SAS Viya.[1]
2. Еластичність. Можливість легко масштабувати систему, підключаючи або відключаючи вузли кластера CAS. Програма є доступною через web і організована у вигляді мікросервісів. Вони незалежні один від одного в питаннях встановлення, оновлення і роботи.
3. Висока доступність. У CAS використовується система віддзеркалення даних між вузлами кластера. Один набір даних зберігається на декількох вузлах, що зменшує ризик втрати даних.

Перемикання в разі відмови одного з вузлів відбувається автоматично зі збереженням стану виконання завдання, що часто буває критично для важких аналітичних розрахунків.

4. Підвищена безпека. Оскільки хмарне середовище може бути отримане від публічного провайдера, то реалізація повинна відповідати більш жорстким вимогам до надійності каналів передачі даних. Розгорнути платформу Viya можна де завгодно - в хмарі, на виділеній машині в своєму ЦОДі, в кластері з будь-якої кількості машин, автоматично забезпечується відмовостійкість.

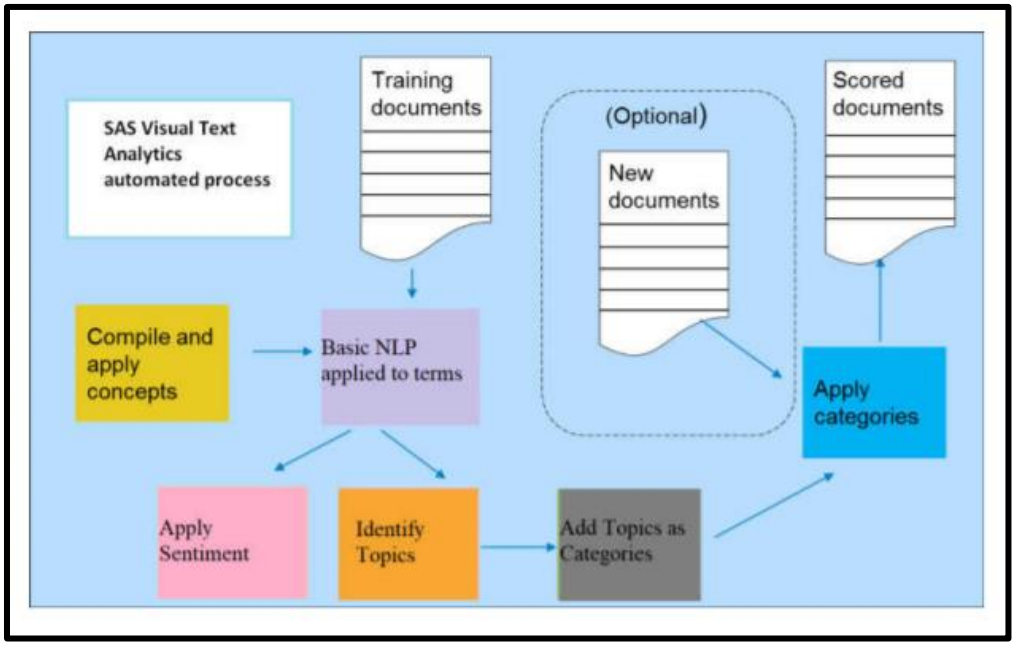


Рисунок 1.3 – Структура SAS VTA

1.3.2 Google Cloud Natural Language, Microsoft Text Analytics API, Amazon Comprehend

Дані програмні продукти створені одними з найвідоміших конкуруючих компаній та мають багато спільного. Ціль Google (рисунок 1.4), Microsoft, Amazon – якомога сильніше спростити роботу користувача з програмою. Дана ціль звужує можливості налаштування текстової аналітики, що погано для якісного дослідження, але все ж таки дає змогу вирішити велику кількість бізнес задач.

Задачі, що вирішуються даними ПП:

– Аналіз настроїв;
– Вилучення ключових фраз;
– Виявлення мови;
– Розпізнавання названих об'єктів.

В Google пропонують максимально спрощену систему, яку не потрібно налаштовувати та взаємодію з Google екосистемою.

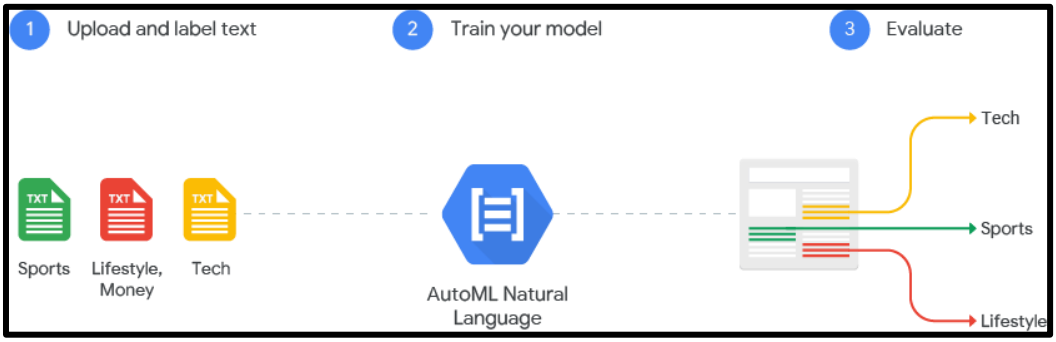


Рисунок 1.4 – Структура Google Cloud NLP

Microsoft Text Analytics API (рисунок 1.5) являється компонентою платформи Azure та має сумісність з MS Excel та програмою для просунутої візуалізованої звітності MS PowerBI. Таким чином, звичайний користувач має змогу налаштувати автоматизований процес дані-аналіз-звіт.

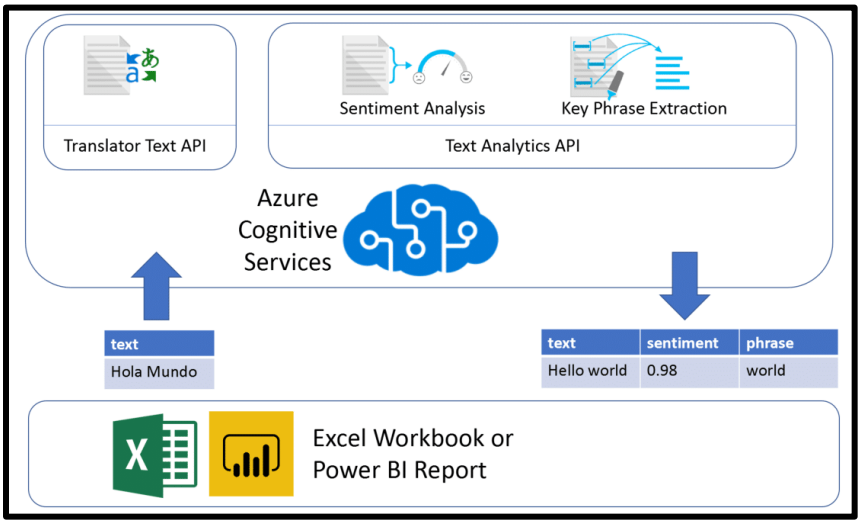


Рисунок 1.5 – Структура MS Text Analytics API

Amazon Comprehend (рисунок 1.6) теж являє собою спрощену автоматизовану систему для тектового аналізу, але має вже налаштовані рішення для конкретних задач. Так наприклад, розроблена система аналізу та категоризації текстової медичної звітності для вирішення завдань фармацевтичних компаній.

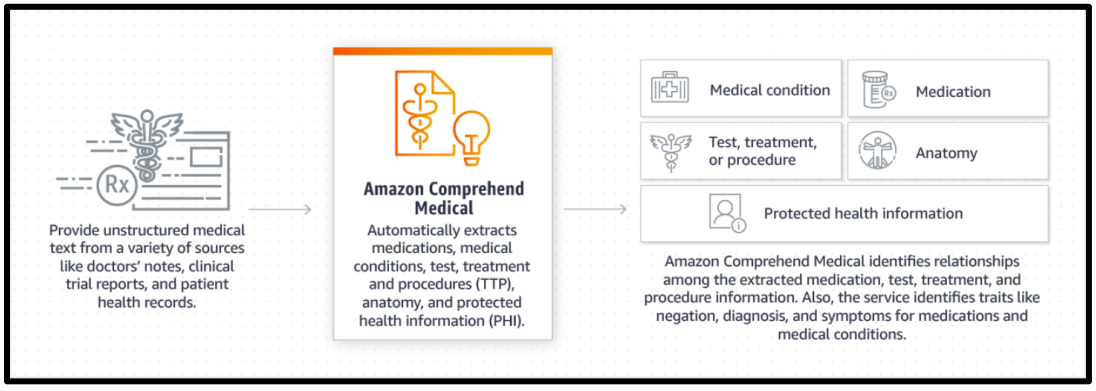


Рисунок 1.6 – Структура ПП Amazon Comprehend

1.3.3 Rapid Miner

Rapid Miner – являється розробкою невеликої компанії, що спеціалізована на інтелектуальному аналізі даних. Має широкі можливості, складніший інтерфейс, але доступніший за SAS, тому займає одну з передових позицій у використанні малим та середнім бізнесом, а також незалежними дослідниками.

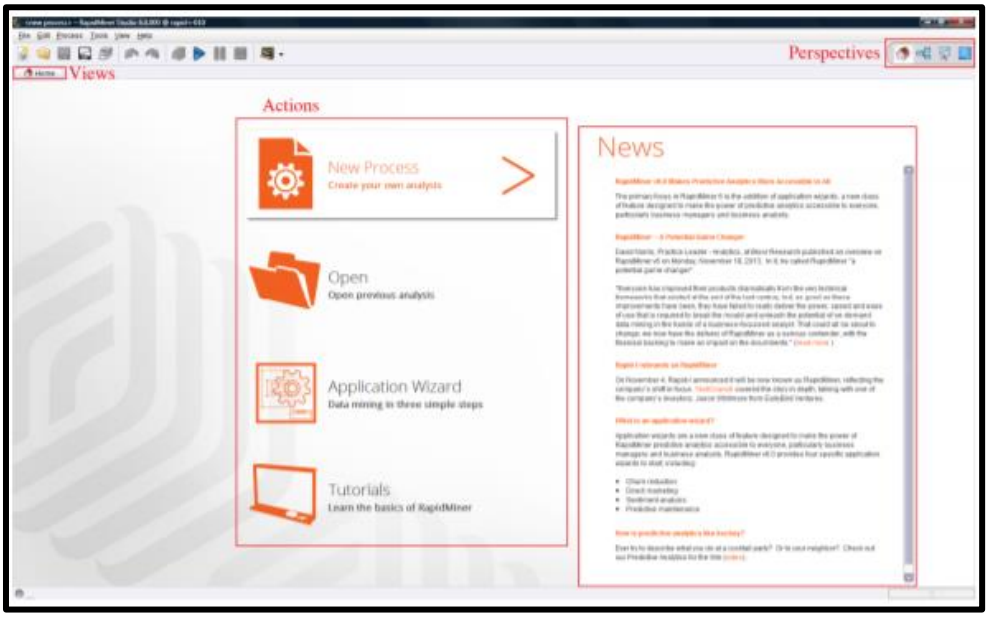


Рисунок 1.5 – вигляд ПП Rapid Miner

1.3.4 Python NLTK

NLTK (Natural Language Toolkit) – провідна відкрита платформа (рисунок 1.8) для створення NLP-програм різної складності на Python. У неї є легкі у використанні інтерфейси для багатьох мовних корпусів, а також бібліотеки для обробки текстів для класифікації, токенізації, стемінг, розмітки, фільтрації і семантичних міркувань. Також це безкоштовний опенсорсний проект, який розвивається за допомогою широкого кола розробників.

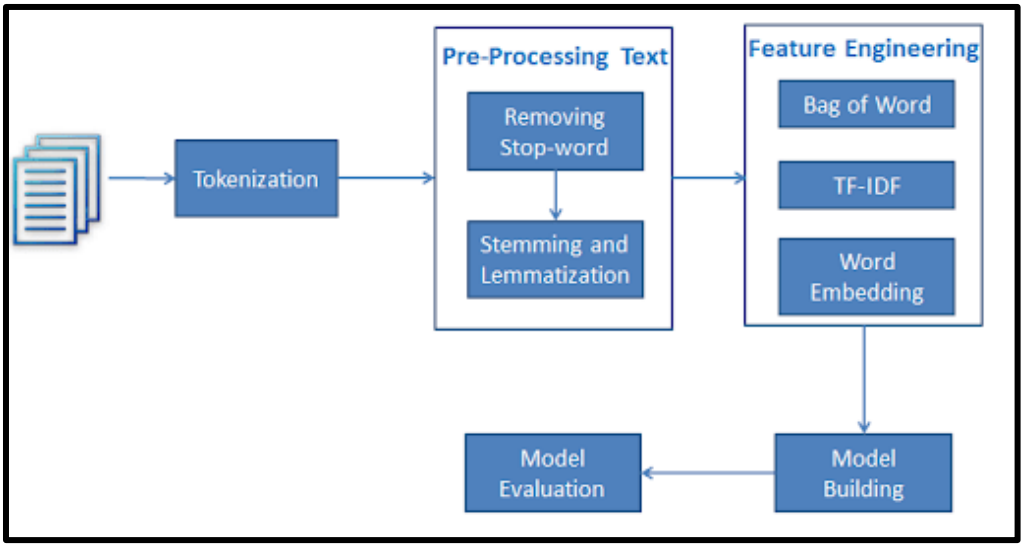


Рисунок 1.8 – Структура компонент бібліотеки NLTK

1.4 Постановка задачі

Метою дипломної роботи є дослідження існуючих методів автоматизованого аналізу і категоризації текстових медичних звітів, розробка системи для попередньої обробки даних, аналізу тональності, кластеризації тематик, класифікації категорій та перевірка якості побудованої моделі. У рамках роботи необхідно:

- розробити архітектуру системи;
- розробити ПП для аналізу та категоризації текстових медичних даних;
- протестувати комп'ютерну програму на реальних даних та провести порівняльний аналіз з іншими методами.

Для рішення цих задач необхідно дослідити вже існуючі методи інформаційного пошуку та інтелектуального аналізу.

Об'єкт дослідження – текстові відгуки пацієнтів, які потребують швидкої аналітичної обробки та мають важливу роль у прийнятті рішень фармацевтичними компаніями та у створенні нових лікарських засобів.

Предмет дослідження – математичні методи задач інформаційного пошуку та інтелектуального аналізу , а саме: булеві правила, латентно-семантичний аналіз, метод сингулярного розкладу матриці, метод максимальної правдоподібності.

1.5 Висновки до розділу 1

Медичні звіти пацієнтів несуть важливі дані для фармацевтичних компаній та для незалежних дослідників, що аналізують якість деякого препарату, або препарати, що частіше вживають при конкретному захворювані. Тому існує необхідність виконувати їх обробку якомога швидше, точніше та з невеликим застосуванням людського ресурсу.

У даному розділі розглянуто основні інструменти інформаційного пошуку та інтелектуального аналізу текстуальних даних. Було дано визначення поняттям текстової аналітики, а також досліджено основні етапи розвитку галузі.

Проведено огляд програмних продуктів компаній, які є основними лідерами на ринку програмного забезпечення призначеного для просунутого текстового аналізу, а саме SAS Viya, Google Cloud Natural Processing, RapidMiner, MS Text Analytics API, Amazon Comprehend, відкрита бібліотека Python - NLTK. Зроблено висновок, що SAS, Rapid Miner, та бібліотека Python NLTK мають великий спектр налаштувань та добре підходять як незалежним дослідникам, так і для складних бізнес-задач. Для розв’язання простих задач – краще обрати сервіс Google, Amazon або Microsoft.

Показано актуальність та перспективність дослідження, на основі чого сформульовано постановку задачі бакалаврського диплому, виділено етапи її розв’язку.

РОЗДІЛ 2 МАТЕМАТИЧНІ МЕТОДИ МОДЕЛЮВАННЯ

Процес побудови математичної моделі - формалізованого (тобто представленого у вигляді математичних співвідношень) опису комплексу чинників, що впливає на стан і функціонування досліджуваного об'єкта, і відповідного цьому опису інформаційного забезпечення - прийнято називати математичним моделюванням.

Практична користь математичного моделювання полягає в можливості отримання інформації про якісні властивості та кількісні характеристики досліджуваного об'єкта без проведення (часто складних або дорогих) експериментів, що може виправдовувати витрати на подолання труднощів, що виникають в процесі розробки або при спробах використання математичних моделей. Основна складність, з яким доводиться стикатися в математичному моделюванні, полягає в забезпеченні адекватності цієї моделі досліджуваного об'єкта. Користувачеві необхідно з'ясувати, наскільки точно дана модель відображає реальну ситуацію і наскільки надійні кількісні оцінки можуть бути отримані в процесі роботи з цією моделлю.

Досвід математичного моделювання (рисунок 2.1) у, накопичений протягом останніх декількох десятиліть, показує, що проблема адекватності в ряді випадків може бути успішно вирішена. Прикладом тому служать системи комп'ютерної імітації численних природних процесів і технічних об'єктів. [2]

З іншого боку, спроби застосування методів математичного моделювання для дослідження соціально-економічної об'єктів природи переконливо демонструють, що, незважаючи на природне бажання врахувати в моделі всі фактори, які суттєво впливають на функціонування

досліджуваного об'єкта, домогтися цього виключно важко, а іноді навіть неможливо.

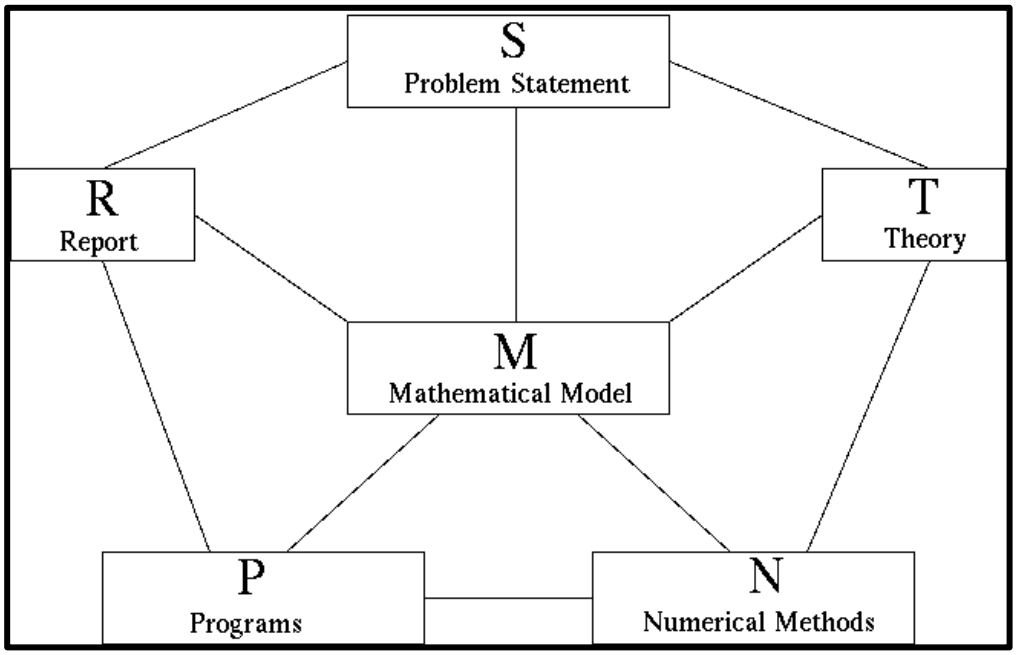


Рисунок 2.1 - Математичне моделювання

2.1 Методи категоризації текстуальних даних та аналізу тональності

Задачі категоризації текстуальних даних та задача аналізу тональності у загальному сенсі зводяться до задачі класифікації та кластеризації.[3] Існує доволі багато математичних алгоритмів, які можуть дати непоганий результат. При розв'язанні конкретної задачі класифікації найбільш доцільним є використання декількох моделей з подальшим порівнянням та вибором найкращої.

2.1.1 Булеві правила

Для задачі категоризації даних є доцільним написання булевих правил. Даний метод полягає у тому, щоб описати кожну цільову категорію булевими правилами з використанням операторів об'єднання та перетину різної складності, з необмеженою кількістю вкладених рівнів. Проте слід пам'ятати про оптимізацію даних правил для покращення ефективності роботи. Так як при описі категорій одну й ту саму функцію людина може записати у різних виглядах - для мінімізації булевих правил можна використовувати метод Квайна Мак Класкі або Карти Карно.

ПП для просунутої аналітики також дозволяє автоматизувати створення булевих функцій. Для того, щоб описати цей процес – далі розглядається компонента SAS – процедура BOOLLEAR. [4]

Робота даної компоненти складається з двох основних етапів:

1. Відбір необхідних термінів для складання правила (рисунок 2.2). Ітеративно додаються терміни до правила. Коли процес завершається, повертається правило, яке може бути використано як правило-кандидат для процесу додання правил;
2. Додання правил до складеної функції для її покращення (рисунок 2.3). Процес додання правил ітеративно створює і додає нові правила до множини правил. Коли процес завершиться, він повертає набір правил, який потім може бути використаний для категоризації тексту.

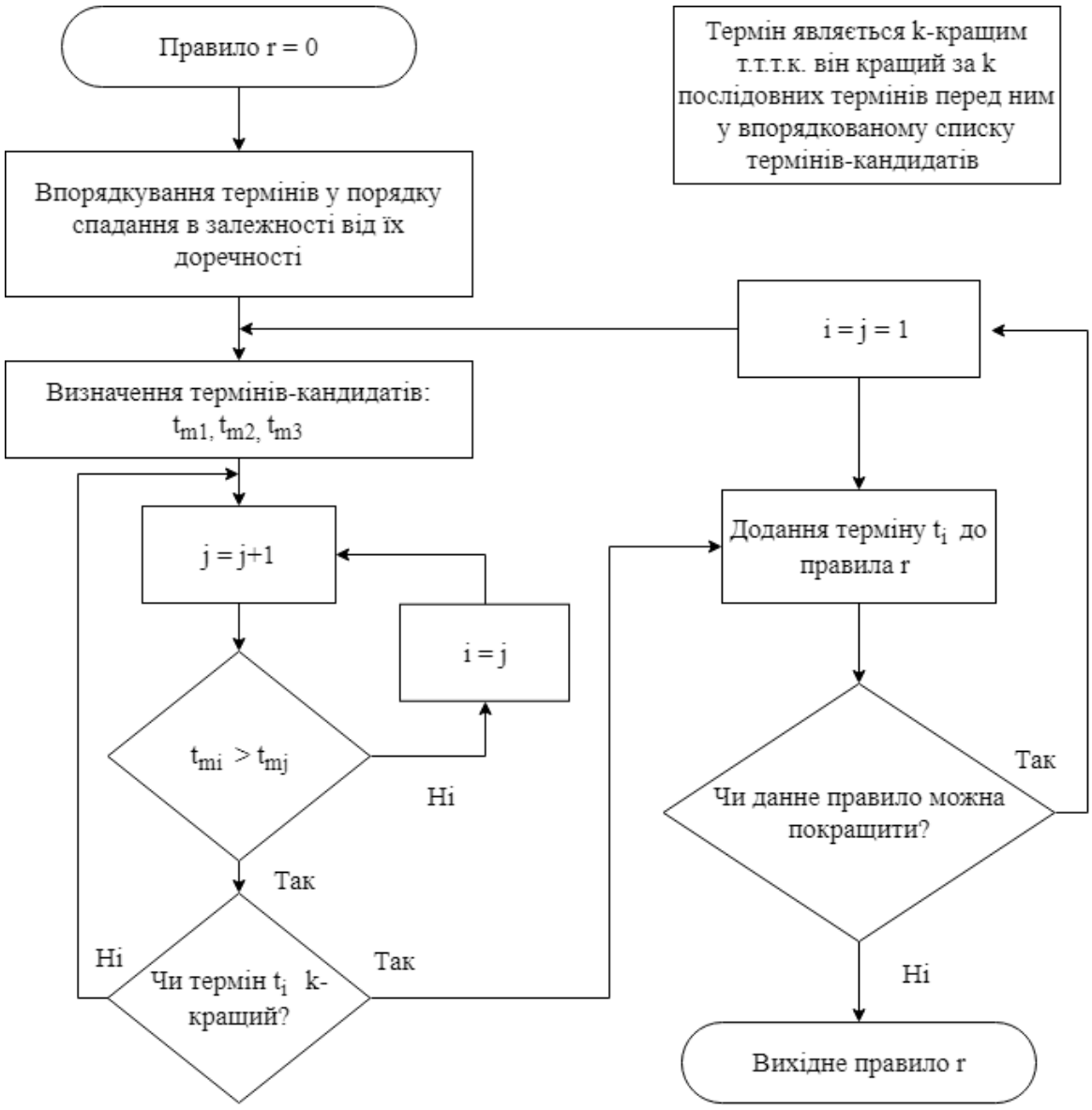


Рисунок 2.2 - Відбір необхідних термінів для складання правила

Рисунок 2.3 - Додання правил до складеної функції для її покращення

2.1.2 Метод максимальної правдоподібності

Даний метод широко використовується для вирішення задачі кластеризації. Кожна ітерація містить два кроки - обчислення математичних очікувань і максимізацію.

В основі ідеї ЕМ-алгоритму лежить припущення, що досліджувана множина даних може бути змодельована за допомогою лінійної комбінації багатовимірних нормальних розподілів, а метою є оцінка параметрів розподілу, які максимізують логарифмічну функцію правдоподібності, використовувану в якості міри якості моделі. Іншими словами, передбачається, що дані в кожному кластері підкоряються певному закону розподілу, а саме, нормальному розподілу. З урахуванням цього припущення можна визначити параметри - математичне сподівання і дисперсію, які відповідають закону розподілу елементів в кластері, найкращим чином "невластивому" до спостережуваних даними.

Таким чином, ми припускаємо, що будь-яке спостереження належить до всіх кластерів, але з різною ймовірністю. Тоді завдання полягатиме в "підгонці" розподілів суміші до даних, а потім у визначенні ймовірностей приналежності спостереження до кожного кластеру. Очевидно, що спостереження повинно бути віднесено до того кластеру, для якого ця можливість вище.

Серед переваг ЕМ-алгоритму можна виділити наступні:

• Потужна статистична основа;

• Лінійне збільшення складності при зростанні обсягу даних;

• Стійкість до шумів і перепустками в даних;

• Можливість побудови бажаного числа кластерів;

• Швидка збіжність при вдалій ініціалізації.

Однак алгоритм має і ряд недоліків. По-перше, припущення про нормальність всіх вимірювань даних не завжди виконується. По-друге, при невдалій ініціалізації збіжність алгоритму може виявитися досить повільним. Крім цього, алгоритм може зупинитися в локальному мінімумі і дати квазіоптимальне рішення.

ЕМ-алгоритм передбачає, що кластерізуемие дані підкоряються лінійної комбінації нормальних розподілів. Щільність ймовірності нормального розподілу має вигляд:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

де $\mu = E(X)$ - математичне очікування,

$\sigma 2 = E(X-\mu)^2$ - дисперсія.

Багатовимірний нормальний розподіл для q-мірного простору є узагальненням попереднього виразу. Багатовимірна нормальна щільність для q-мірного вектора $x = (x_1, x_2, x_3, \ldots, x_q)$ може бути записана у вигляді:

$$p(x) = \frac{1}{(2\pi)^{\frac{q}{2}}\sqrt{|\sum|}} exp\left\{-\frac{1}{2}(x-\mu)^T \times \sum^{-1} \times (x-\mu)\right\}$$

де Σ - ковариаційна матриця розміром q × q, яка є узагальненням дисперсії для багатовимірної випадкової величини, μ являється q-мірним вектором математичних очікувань;

| Σ | - визначник ковариационной матриці;

Т- оператор транспонування.

Введемо в розгляд функцію: $\delta^2 = (x-\mu)^T \times \sum^{-1} \times (x-\mu)$.

Алгоритм передбачає, що дані підкоряються суміші багатовимірних нормальних розподілів для qq змінних. Модель, що представляє собою суміш гауссових розподілів задається у вигляді:

$$p(x) = \sum_{t=1}^{k} w_i \times p(x|i)$$

де $p(x|i)$ - нормальний розподіл для і-го кластера,

wi - вага і-го кластера у початковій базі даних.

2.2 Методи початкової обробки даних для побудови моделі

2.2.1 Збір даних

Збір даних – це дуже важливий та відповідальний процес у інтелектуальному аналізі. Від якості та кількості даних залежать результати дослідження.

Інформацію у медичній сфері можна збирати шляхом анкетування з подальшою обробкою інформації експертами для отримання тренувальних даних. Такий метод є дуже затратним, як у часовому ресурсі, так і у грошовому, людському ресурсах, але і найбільш ефективним.

Інший спосіб – методи інформаційного пошуку, парсинг інформації, що вже є у відкритому чи обмеженому доступі на просторах мережі Internet. Наприклад, можна проаналізувати відгуки пацієнтів, що мають певне захворювання й користуються деякими препаратами. Такий метод є менш затратним, швидким, але менш ефективним, потребує написання програми для парсингу даних. Хоча наявність тренувальних даних значно полегшує

реалізацію моделей категоризації та аналізу тональності, якщо таких немає в наявності – розроблені методи інформаційного пошуку дають непогані результати для подальшого дослідження, але вимагають певного налаштування під конкретний тип даних.

Важливим етапом збору даних являється відсіювання некорректних та пустих записів, що не важливі для дослідження.

2.2.2 Токенізація та стемінг

Токенізація за реченнями - це процес поділу писемної мови на пропозиції-компоненти. Ідея виглядає досить простий. В англійській і багатьох інших мовах ми можемо виокремлювати пропозицію кожен раз, коли знаходимо певний знак пунктуації - точку. Але навіть в англійській мові ця задача нетривіальна, так як точка використовується і в скороченнях. Таблиця скорочень може сильно допомогти під час обробки тексту, щоб уникнути невірної розстановки кордонів пропозицій.

Токенізація за словами - це процес поділу пропозицій на слова-компоненти. В англійській, українській і багатьох інших мовах, що використовують ту чи іншу версію латинського алфавіту, пробіл - непоганий роздільник слів. Проте, можуть виникнути проблеми, якщо ми будемо використовувати тільки пробіл - в англійському складові іменники пишуться по-різному і іноді через пробіл. Для поліпшення точності використовуються словники з записами про відповідні ситуації.

Зазвичай тексти містять різні граматичні форми одного і того ж слова, а також можуть зустрічатися однокореневі слова. Лематизація і стемінг мають

на меті привести все зустрічаються словоформи до однієї, нормальної словникової форми. Приведення різних словоформ до однієї: cat, cats, cat’s, cat '=> cat. Лематизація і стемінг - це окремі випадки нормалізації і вони відрізняються.

Стемінг - це грубий евристичний процес, який відрізає «зайве» від кореня слів, часто це призводить до втрати словотворчих суфіксів.

Лематизація - це більш тонкий процес, який використовує словник і морфологічний аналіз, щоб в результаті привести слово до його канонічної форми - лемми.

Відмінність в тому, що стеммер (реалізація алгоритму стемінг) діє без знання контексту і, відповідно, не розрізняє слова, які мають різний зміст в залежності від частини мови. Однак у стеммері є перевага у швидкості та у простоті реалізації.

Компоненти текстової аналітики SAS використовують спеціальні словники, для задачі нормалізації, тобто реалізований алгоритм лематизації.

2.2.3 VSM

Метод інформаційного пошуку для представлення колекції документів у векторному вигляді зі спільним для усієї колекції векторним простором. Кожний документ складається з множини термінів. Усі терміни в документі можна впорядкувати й перевести у числовий вигляд, користуючись певними мірами.

У текстовому аналізі найбільш правильнішою та уживаною мірою для реалізації цієї задачі – являється метод TF-IDF, що описує частоту трапляння

конкретного терма в документі, та обернена частота трапляння терміну у документах колекції.

Кожен вектор, що представляє документ, має однакову розмірність, що співпадає з кардинальним числом термінів у колекції документів. Для спрощення обрахувань – важливо зменшити кількість термінів у колекції, шляхом видалення стоп-слів.

$$d_j = (w_{1j}, w_{2j}, \dots, w_{nj})$$

де d_j – векторний вигляд j-ого документа, w_{ij} – вага і-ого терма у j-ому документі, n – кардинальне число термінів у колекції. Вагою можуть виступати результати методу TF, або TF-IDF.

Косинусна схожість - міра подібності між двома векторами предгільбертового простору, яка використовується для вимірювання косинуса кута між ними.

Якщо дано два вектора ознак, u і v, то косинусна схожість може бути представлена, використовуючи скалярний твір і норму:

$$similarity = \cos(\theta) = \frac{AB}{|A||B|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}$$

[5]

У разі інформаційного пошуку, косинусна схожість двох документів змінюється в діапазоні від 0 до 1, оскільки частота терміна (ваги TF-IDF) не може бути негативною. Кут між двома векторами частоти терміна не може бути більше, ніж 90 °.

Одна з причин використання косинусної подібності полягає в тому, що воно ефективне для оцінювання, особливо для розріджених векторів, так як необхідно враховувати тільки ненульові вимірювання.

2.2.4 Міра TF-IDF

Центральним питанням у видобутку тексту та обробці природної мови є визначення кількісного значення термінів та документів. Одним з показників того, наскільки важливим може бути слово, є його термінова частота (tf), частота слова в документі. У документі є слова, які трапляються багато разів, але можуть бути важливими; англійською мовою, ймовірно, це такі слова, як “the”, “is”, “of” і так далі. З використанням підходу додавання таких слів до списку стоп-слів і видалення їх перед аналізом, можливо, що деякі з цих слів можуть бути більш важливими в деяких документах, ніж інші. Тому список стоп-слів не завжди є доцільним підходом до регулювання частоти слів для часто використовуваних слів.

$$\mathrm{TF(t)} = \frac{N_t}{N}$$

де $N_t -$ кількість разів термін t з′являється у документі, $N -$ загальна кількість термінів у документі.

Інший підхід полягає в тому, щоб подивитися на зворотну частоту документів терміну (idf), яка зменшує вагу для часто використовуваних слів у колекції і збільшує вагу для слів, які не використовуються у великій кількості документів. Це можна поєднати з частотою термінів, щоб обчислити tf-idf терміна, частоту терміна, що коригується, як рідко використовується. Він призначений для вимірювання того, наскільки важливим є слово в документі в колекції (або корпусі) документів. Являється евристичною величиною й застосовується у задачах інформаційного пошуку та інтелектуальному аналізі

текстових колекцій. Обернена частота документа для будь-якого даного терміну визначається як:

$$\mathrm{IDF(t)} = \ln \frac{\text{Кількість документів}}{\text{Кількість документів, до яких входить термін}}$$

В результаті, використовуючи два цих підходи, – отримали:

$$TF(t) - IDF(t) = TF(t) \times IDF(t) \qquad [6]$$

2.2.5 SVD

Сингулярний розклад матриці A є факторизацією A на добуток трьох матриць

$$\mathrm{A} = \mathrm{U} \times \mathrm{D} \times V^T$$

де стовпці матриці U і V є ортонормальними, і матриця D є діагональною з позитивними реальними записами.

SVD корисний у багатьох завданнях. По-перше, у багатьох додатках матриця даних A близька до матриці низького рангу, а важливо знайти матрицю з низьким рангом, яка є гарною апроксимацією до матриці даних. З сингулярного розкладу A ми може отримати матрицю B рангу k, яка найкраще наближається до A. Це можливо зробити для кожного k. Крім того, для всіх матриць (прямокутних або квадратних) визначено розкладення сингулярних значень на відміну від більш часто використовуваного спектрального розкладання в лінійній алгебрі.

Необхідні умови на матриці для забезпечення ортогональності власних векторів. Навпаки, стовпці V в сингулярному розкладі, називаються правими сингулярними векторами A, завжди утворюють ортогональний набір на A.

Колонки U називаються лівими сингулярними векторами, і вони також утворюють ортогональний набір. Простим наслідком ортогональності є те, що для квадратної і оберненої матриці A, інверсія A дорівнює:

$$V \times D^{-1} \times U^T$$

Щоб отримати уявлення про SVD, можна розглянути рядки n × d матриці A як n точок у d-мірному просторі і розглянемо задачу пошуку кращого k-мірного підпростору по відношенню до набору точок. Кращий – означає мінімальне значення суми квадратів перпендикулярних відстаней від точок до підпростору.

Розглядаючи одновимірний підпростір, необхідно знайти найкращу лінію відповідно до набору точок $\{x_i \mid 1 \leq \text{i} \leq \text{n}\}$ у площині, тобто знайти мінімізацію суми квадрата перпендикулярної відстані від точок до лінії. Така задача називається методом найменших квадратів.

Альтернативною задачею є пошук функції відстані, яка найкраще підходить для конкретних даних. [7]

Повертаючись до задачі підбору найкращих квадратів, розглянемо проекцію (рисунок 2.4) точки x_i на лінію.

$$x_{i1}^2 + x_{i2}^2 + \ldots + x_{id}^2 = (\text{довжина проекції})^2 + (\text{довжина від точки до лінії})^2$$

Таким чином,

$$(\text{довжина від точки до лінії})^2 = x_{i1}^2 + x_{i2}^2 + \ldots + x_{id}^2 - (\text{довжина проекції})^2$$

Мінімізацію суми квадратів довжин до лінії можна представити, як максимізацію суми квадратів довжини проекцій, оскільки $\sum_{i=1}^{n} x_{i1}^2 + x_{i2}^2 + \ldots + x_{id}^2$ являється константою.

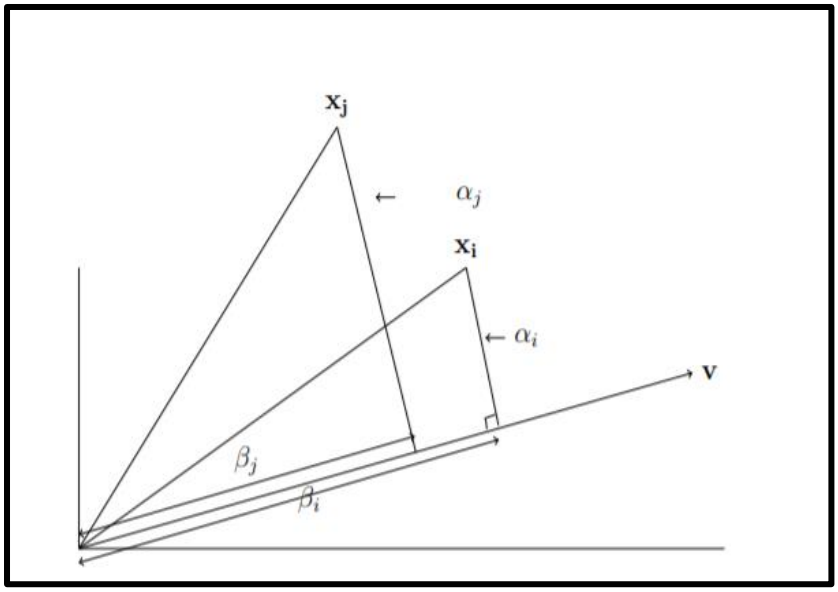


Рисунок 2.4 – проекція точки x_i на лінію в напрямку v

$$Min \sum \alpha^2 \Leftrightarrow Max \sum \beta^2$$

Аналогічно для найкращої апроксимації, ми могли б максимізувати суму квадратів довжин проекцій на підпростір замість мінімізації суми квадратів відстаней до підпростору.

Замість перпендикулярних відстаней можливо обрати іншу криву, в залежності від випадку, але в загальному випадку – така відстань дає свої переваги. [8]

Нехай A – матриця n × d з сингулярними векторами $x_1, x_2, \ldots, x_r$ та відповідними сингулярними числами $\sigma_1, \sigma_2, \ldots, \sigma_r$. Тоді $u_i = \frac{1}{\sigma_i} A v_i$, для i $= 1, 2, \ldots,$ r , - ліві сингулярні вектори і за теоремою 2.2, A може бути розкладена на суму матриць рангу 1, як

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Спочатку доведемо просту лемму, яка стверджує, що дві матриці A і B ідентичні, якщо $\forall v$: Av = Bv. Лема вказує, що в абстрактному вигляді матрицю A можна розглядати як перетворення, яке відображає вектор v на Av.

Лемма 2.1 Матриці A і B ідентичні тоді і тільки тоді, коли $\forall v$: Av = Bv.

Доведення: Очевидно, якщо A = B, то $\forall$ v: Av = Bv. Щоб переконатися, припустимо, що $\forall v$: Av = Bv. Нехай e_i – це вектор, який є усіма нулями, за винятком і-тої компоненти, яка має значення 1.

Яка має значення 1. Тепер Ae_i і-тий стовпець A і, таким чином, $A = B$, якщо для кожного і, $Ae_i = Be_i$.

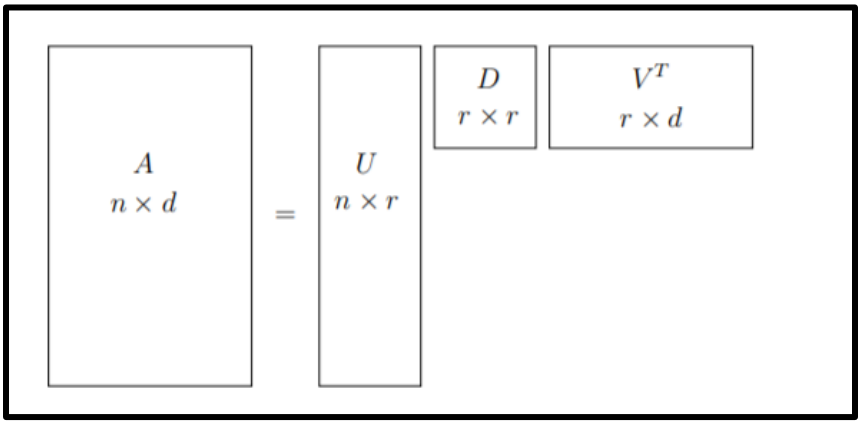


Рисунок 2.5 – SVD

Теорема 2.2 Нехай A - матриця n × d з правими сингулярними векторами $v_1, v_2, \ldots, v_r$, лівими сингулярними векторами $u_1, u_2, \ldots, u_r$ та відповідними сингулярними числами $\sigma_1, \sigma_2, \ldots, \sigma_r$. Тоді

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Доведення: для кожного сингулярного вектора v_j:

$$Av_j = \sum_{i=1}^{r} \sigma_i u_i v_i^T v_j. \qquad [9]$$

Оскільки будь-який вектор v може бути виражений як лінійна комбінація сингулярних векторів та вектора,
який перпендикулярний до v_i, $Av = \sum_{i=1}^{r} \sigma_i u_i v_i^T v_j$ та за лемою 2.1,

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Розклад (рисунок 2.5) називається розкладом сингулярних значень,
$A = UDV^T$, де стовпці U і V складаються з лівого і правого сингулярних векторів, відповідно, і D - діагональна матриця, діагональні записи якої є сингулярними значеннями A.

Для будь-якої матриці A послідовність сингулярних значень є унікальною і якщо сингулярні значення всі різні, тоді послідовність сингулярних векторів також є унікальною. Однак, коли деякі множини сингулярних значень рівні, відповідні їм сингулярні вектори охоплюють деякий підпростір. Будь-яка множина ортонормальних векторів, що охоплюють цей підпростір, може використовуватися в якості сингулярних векторів.[10]

2.2.6 PCA

SVD схожий на аналіз основних компонентів (PCA). Фактично, ці два методи дають однакові результати, якщо вони застосовуються до коваріаційної матриці термінів.[11] Іншими словами, якщо середні частоти термінів віднімалися від спостережуваних частот у кожному стовпці документа за терміновою частотною матрицею, обидва методи дадуть

однакові результати. Тим не менш, SVD зазвичай застосовується до відносно розрідженого документа за словом або терміновою частотою (лише кілька документів мають специфічні терміни), тоді як PCA зазвичай застосовується до симетричної коваріаційної матриці.[12]

З точки зору інтерпретації, PCA максимізує дисперсію послідовно витягнутих розмірів, тоді як SVD мінімізує залишкові суми квадратів відхилень розрахункових значень від спостережуваних значень у A, з урахуванням відповідних чисел. Тим не менш, інтерпретації, що використовуються в цих двох методах, дуже схожі в тому, як виокремлюють базові, так і "приховані" розміри, які фіксують більшу частину інформації, що міститься в повній матриці даних.

2.2.7 LSA

Латентно-семантичний аналіз – це метод обробки природньої мови, який дозволяє виявити взаємозв’язки між колекцією документів та термінами, що в ній трапляються, виокремити тематики з термінів, що мають сильні зв’язки.

В основі методу лежать принципи факторного аналізу, зокрема, виявлення латентних зв'язків досліджуваних об'єктів.[13] При класифікації або кластеризації документів цей метод використовується для отримання контекстно-залежних значень лексичних одиниць за допомогою статистичної обробки великих колекцій текстів.

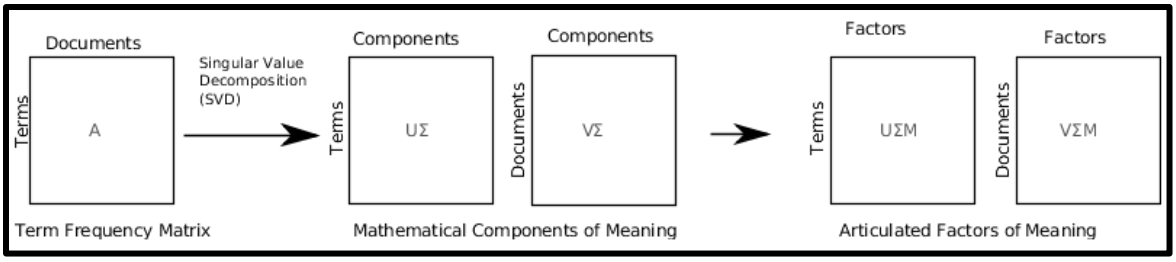


Рисунок 2.6 – алгоритм латентно-семантичного аналізу

Опис алгоритму (рисунок 2.6):

1. Нормалізація тексту, вилучення стоп-слів, стеммінг або лематизація;
2. VSA;
3. Сингулярний розклад матриці;
4. Вилучення інформації:
   - Порівняння двух термінів між собою;
   - Порівняння двох документів між собою;
   - Порівняння теміна і документа. [14]

Переваги методу:

- метод є найкращим для виявлення латентних залежностей усередині безлічі документів;
- метод може бути застосований як з навчанням, так і без навчання (наприклад, для кластеризації);
- використовуються значення матриці близькості, заснованої на частотних характеристиках документів і лексичних одиниць;
- частково знімається багатозначність і омонімія.

Недоліки:

- Істотним недоліком методу є значне зниження швидкості обчислення при збільшенні обсягу вхідних даних (наприклад, при SVD-перетворення).
- Ймовірнісна модель методу не відповідає реальності. Передбачається, що слова і документи мають нормальний розподіл, хоча ближче до реальності розподіл Пуассона. У зв'язку з цим для практичних застосувань краще підходить ймовірнісний латентно-семантичний аналіз, заснований на поліноміальному розподілі.

2.2.8 REGEX запити

Регулярні вирази, або REGEX запити, використовуються для пошуку заданної інформації у тексті. Алгоритм пошуку працює за схемою направленого дерева.

Задачі, для яких використовуються регулярні вирази:

• Виконання перевірки зазначеної кількості файлів; повідомлення про кожну строку кожного файлу, що містить повторювані слова; виділення (за допомогою стандартних Escape послідовностей ANSI) кожне повторення слова і виведення імен відповідних файлів. [15]

• Враховувати можливі розриви строк і виявляти ситуації, коли слово, що починається в кінці однієї строки, закінчується на початку наступної.

• Отримання повторів, що різні в регістрі символів (наприклад, "The ...the") і в кількості пропусків (пробіли, символи табуляції, перенесення строк і т. д.) між словами.

• Знаходити повтори, розділені тегами HTML. Теги HTML застосовуються при розбитті тексту в веб-сторінки, наприклад, для виокремлення слів жирного шрифту: «… це дуже важливо ...».

Таким чином, регулярні вирази являють собою важливий інструмент для інформаційного пошуку й використовуються у текстовій аналітиці для визначення та виокремлення понять різної складності: медичні рецепти, дозування, грошові одиниці, дати та інші.

2.3 Статистичні методи та підходи щодо оцінювання моделей

2.3.1 RMSSTD

RMSSTD є мірою однорідності в межах кластерів, статистичний показник:

$$\sqrt{\frac{SS_1 + \cdots + SS_p}{df_1 + \cdots + df_p}}$$

тобто об'єднане стандартне відхилення всіх змінних. Термін SS_j - сума квадратів j-ої змінної, обчислюється за формулою:

$$SS_j = \sum_{i=1}^{n} (x_{ij} - \overline{x_{ij}})^2$$

Великі значення RMSSTD вказують на те, що кластери не є однорідними. [16] Не існує загальних правил для оцінки того, чи є значення статистики RMSSTD малими чи великими, але відносні зміни у значеннях статистики, як число збільшення кластерів, можуть бути корисними для визначення кількості кластерів. Розрахунок статистики на кожному етапі алгоритму кластеризації, тобто для кожного числа кластерів, дозволяє побудувати значення щодо кількості кластерів. Виражене зменшення або збільшення для RMSSTD, відповідно, може вказувати на досягнення задовільного числа кластерів.

2.3.2 Відстань Кульбека-Лейблера

Міра приросту інформації використовується у процесі інтелектуального аналізу для дослідження взаємозв’язків між термінами.

Для того, щоб виміряти різницю між двома розподілами ймовірностей з однією й тією ж змінною x, існує міра, яка називається дивергенцію Кульбака-Ліблера, або просто, дивергенція KL, широко використовується в текстовій аналітиці. Концепція виникла в теорії ймовірностей і теорії інформатизації.[17]

Відстань KL, тісно пов'язана з відносною ентропією, інформаційною дивергенцією, є несиметричним показником різниці між двома розподілами ймовірностей $p(x)$ та $q(x)$. Зокрема, $q(x)$ з $p(x)$, позначена $D_{KL}(p(x),\ q(x))$, є мірою інформації, втраченої при використанні $q(x)$ для апроксимації p(x).

Нехай $p(x)$ і $q(x)$ є двома розподілами ймовірностей дискретної випадкової величини x. Тобто, $\sum \mathrm{p(x)} = 1$, $\sum q(\mathrm{x}) = 1$, і $\mathrm{p(x)} > 0$ і $q(\mathrm{x}) > 0$ $\forall$ x $\in X$. $D_{KL}(p(x),\ q(x))$ визначено в рівнянні (2.1)

$$D_{KL}(p(x)||q(x)) = \sum_{x\in X} p(x) \ln \frac{p(x)}{q(x)} \qquad (2.1)$$

Розбіжність KL вимірює очікуване кількість додаткових бітів, необхідних для кодування зразків з p(x) при використанні коду на основі $q(x)$, а не з використанням коду на основі p(x). Як правило p(x) являє собою "істинний" розподіл даних, спостережень або точно розрахований теоретичний розподіл. Міра $q(x)$ зазвичай являє собою теорію, модель, опис або апроксимацію p(x). Неперервна дивергенція KL є

$$D_{KL}(p(x)||q(x)) = \int_{-\infty}^{+\infty} p(x) \ln \frac{p(x)}{q(x)} \qquad (2.2)$$

Хоча дивергенція КЛ вимірює "відстань" між двома розподілами, це не відстань. Це пояснюється тим, що KL не є метричною мірою. [18] Вона не є симетричною: KL від p(x) до $q(x)$, як правило, не збігається з KL від $q(x)$ до p(x). Крім того, вона не повинна задовольняти трикутної нерівності. Незважаючи на це, $D_{KL}(P||Q)$ є невід'ємною мірою. $D_{KL}(P||Q) \geq 0$ і $D_{KL}(P||Q) = 0$ тоді і тільки тоді, коли $P = Q$.

При обчисленні розбіжності KL слід бути уважним. Відомо, що $\lim_{p\to 0} p \log p = 0$. Однак при $p \neq 0$, але $q = 0$, $D_{KL}\ (\ p\ |\ |q\ )$ визначається як ∞. Це означає, що якщо одна подія е є можливою (тобто $\mathrm{p}(e) > 0$), а інший розподіл передбачає абсолютно неможливим (тобто, $q(e) = 0$), то розподіли абсолютно різні. Проте на практиці два розподіли P і Q є похідними від спостережень і підрахунку вибірок, тобто від частотних розподілів. Нерозумно передбачити, що подія є абсолютно неможливою, так як необхідно брати до уваги можливість непередбачених подій.

2.4 Висновки до розділу 2

В другому розділі було проведено огляд існуючих математичних методів та інструментів, які можна використовувати для категоризації та аналізу текстуальної інформації, а саме булеві правила, метод найбільшої правдоподібності. Для задачі інформаційного пошуку метод генерації булевих правил є найкращим, але для цього необхідне використання людського ресурсу.

Досліджено процес попереднього аналізу і обробки даних для побудови моделі категоризації даних, який включає в себе: збір даних, нормалізацію тексту (токенізація, стемінг), відбір суттєвих термінів, створення матриці термін-документ, описано метод сингулярного розкладу матриці, що застосовується для зменшення розмірності матриці та для латентно-семантичного аналізу. Описаний процес пошуку важливих понять за допомогою регулярних виразів. Також були розглянуті критерії оцінки якості кластеризації (RMSSTD) та виокремлення взаємозв'язків між термінами (відстань Кульбека-Лейблера).

РОЗДІЛ 3 СИСТЕМА АНАЛІЗУ І КАТЕГОРИЗАЦІЇ ТЕКСТОВИХ МЕДИЧНИХ ДАНИХ З ВИКОРИСТАННЯМ SAS ТЕХНОЛОГІЙ

В цьому розділі наведений опис розробленої в рамках дипломної роботи комп'ютерного програмного продукту DrugReports Analytics. Система призначена для категоризації та подальшого аналізу великих об'ємів текстових даних. Дана програма реалізована за допомогою хмарних технологій SAS Viya з елементами програмування, використовуючи SAS Base.

Програмний продукт дозволяє аналітикам з різним рівнем підготовки проводити необхідну попередню обробку даних для побудови моделі категоризації та одержувати статистичні характеристики та розбиття даних на категорії на основі побудованої моделі.

Інтерфейс SAS Viya є інтуітивно зрозумілим та доступним до використання без особливої підготовки.

За технічним рівнем продукт DrugReports Analytics являється проектом, який працює в компоненті SAS Viya Text Analytics. Цей продукт спрямований на використання як однією людиною, так і командою людей, у яких відкритий до нього доступ. Усі обчислення відбуваються на серверах SAS.

3.1 Аналіз архітектури системи

У цьому розділі розглянуто систему аналізу і категоризації текстових медичних даних з використанням SAS технологій.

На рисунку 3.1 розглянуто структуру створеної системи. Розроблена система надає можливість пошуку, категоризації та аналізу інформації у різних колекціях документів. Існує можливість застосування данної системи не тільки для медичних звітів, але і для інших текстуальних даних, попередньо налаштувавши основні компоненти системи.

Створені документи для зберігання надходять до бази даних, з якої вже аналітик може підтягнути їх для виконання певного аналізу. Для аналізу і категоризації користувач може завантажити дані у налаштований додаток в програмному середовищі SAS Viya Studio, які надалі будуть зберігатися у CAS пам’яті для підвищення ефективності роботи з ними.

Після завантаження даних до головної структурної підсистеми, аналітик отримує головні характеристики текстової колекції: розподіл документів за розширенням, розподіл документів за об’ємом, кардинальне число множини.

Оскільки моделі не можуть працювати з неструктурованими даними, далі відбувається попередня обробка тексту в документах: стеммінг, об’єднання синонімів, виокремлення частин мови та трансформація тексту у числові вектори (зважження), відсіювання термів, що не впливають на зміст.

Після цього аналітиком визначаються основні поняття, які впливають на дослідження, якщо їх не має у існуючому списку понять. Створюються зв’язки між термами та виконується аналіз тональності текстів. Зі ствроненних системою зв’язків отримуються базові тематики, які аналітик, в свою чергу, може налаштувати і відкорегувати задля отримання найкращого результату. З тематик, що мають найбільший показник включення документів, створюються відповідні категорії за допомогою написання булевих правил.

Результати аналітик отримує у вигляді графіку розподілу документів по категоріям, таблиці з розподіленими документами та загальним SAS кодом, який є можливість відкорегувати.

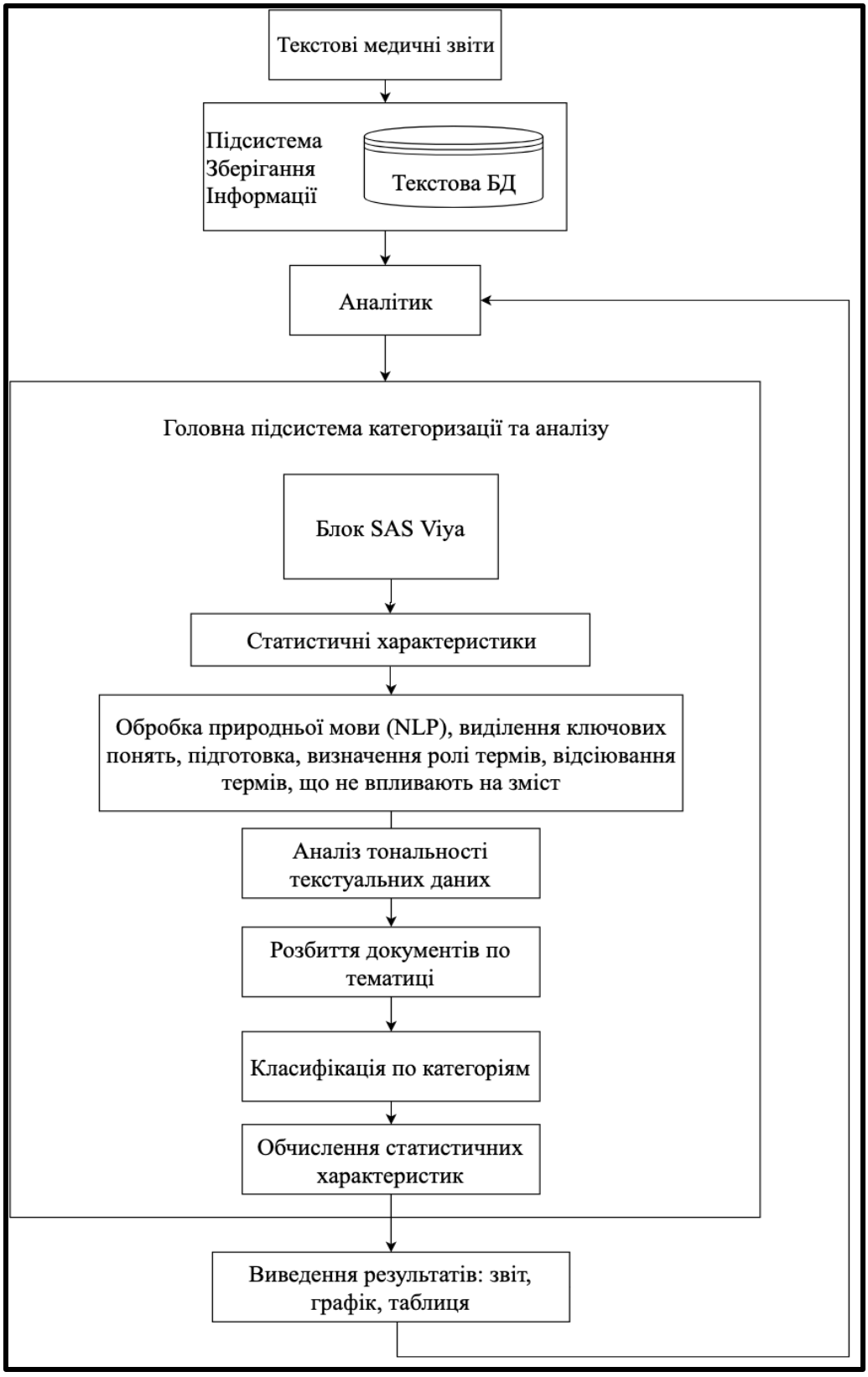


Рисунок 3.1 - Архітектура системи аналізу і категоризації

3.2 Основні технічні вимоги для коректної роботи програми

Для роботи програмного продукту необхідно мати аккаунт SAS та персональний комп’ютер з наступними мінімальними характеристиками:

а) операційна система з підтримкою одного з браузерів: Google Chrome, Safari, Mozilla Firefox;

б) доступ до мережі Інтернет, статична IP адреса;

в) оперативна пам’ять розміром: 2 гб;

г) інструменти введення та виведення: клавіатура, комп’ютерна мишка, монітор;

д) простір для вхідних даних.

3.3 Інструкція з експлуатації програмного продукту

SAS Viya Visual Text Analytics – являє собою платформу для розробки систем, спрямованих на аналіз та обробку великих текстових даних.

1. В компоненті SAS Studio користувач може завантажити дані на сервер, у CAS пам'ять та написати код програми, який буде взаємодіяти з доступними даними.
2. Develop SAS Code призначена для редагування коду проекту.
3. Manage Data дозволяє керувати усіми даними користувача.
4. Prepare Data відповідає за детальну підготовку даних перед моделюванням. Може використовуватись аналітиком для фільтрування пустих записів, балансування завантаженого датасету.

5. Explore and Visualize Data допомагає зосередитись на сенсі даних та зробити зрозумілий кожному звіт, візуалізувавши їх на дашбордах за допомогою різноманітних типів графіків.
6. Компонента Build Models є однією з найважливіших, у цій компоненті є можливість побудувати модель системи, знайти основні поняття, проаналізувати тональність інформації та категоризувати дані.
7. За допомогою додатка Manage Models являється можливим керувати розробленими моделями, дивитись налаштування та застосування створених моделей у проектах.
8. Manage Decisions дозволяє створювати моделі, які використовують інші моделі, в залежності від виконання прописаних бізнес-правил.
9. Explore Lineage допомагає швидко побачити та налаштувати зв’язки між документами і файлами у проекті.
10. Інструмент Build Graphs дає змогу розробляти свої пов’язані графіки.
11. Задля автоматизованої роботи бізнес-процесів можна за допомогою компоненти Workflow Manager.

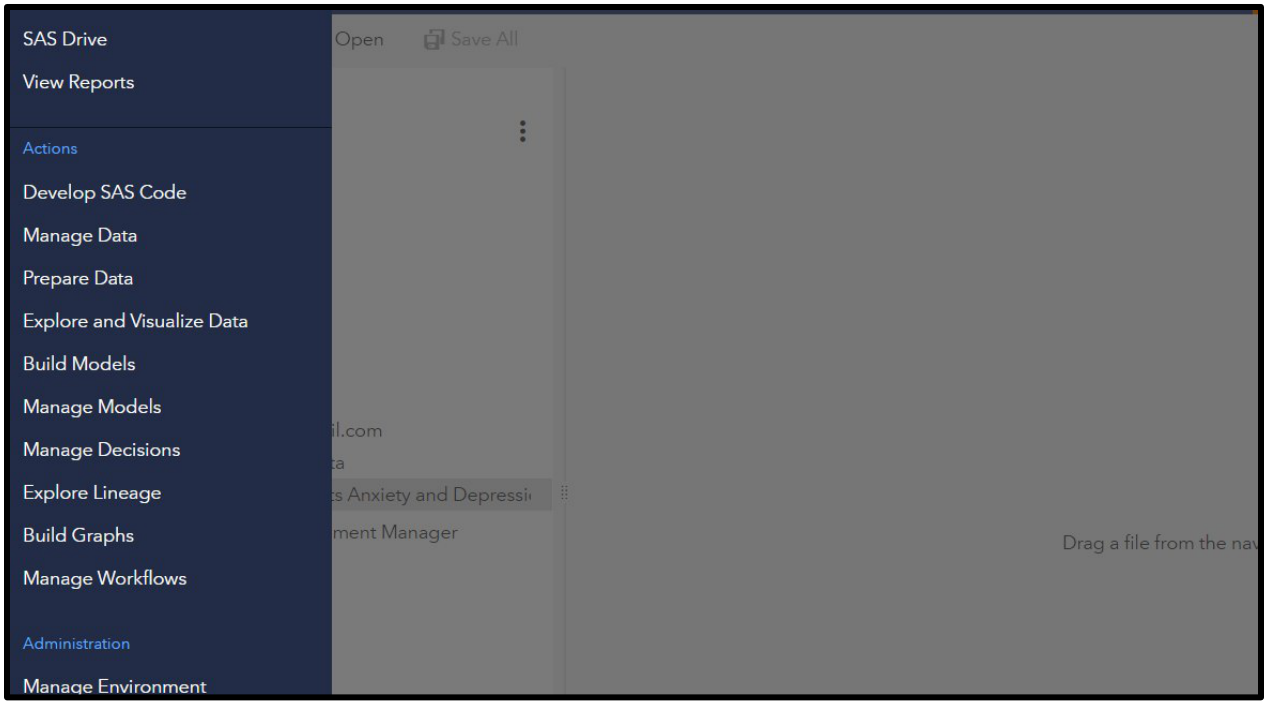


Рисунок 3.2 – SAS Viya Visual Text Analytics інтерфейс та доступні інструменти користувача

3.3.1 Завантаження даних до інструменту SAS Viya

У SAS Studio (рисунок 3.2) визначаються датасети та необхідні бібліотеки у CAS (Cloud Analytical Services) пам’яті. Конвертуються дані у формат HDFS (Hadoop Distributed File System) [19], щоб потім завантажити ці дані у CAS пам’ять. Після цього виконуються процедури, спрямовані на початковий аналіз даних.

Розглянемо на прикладі файлу з медичними текстовими звітами: drug_reports.SAS7bdat.

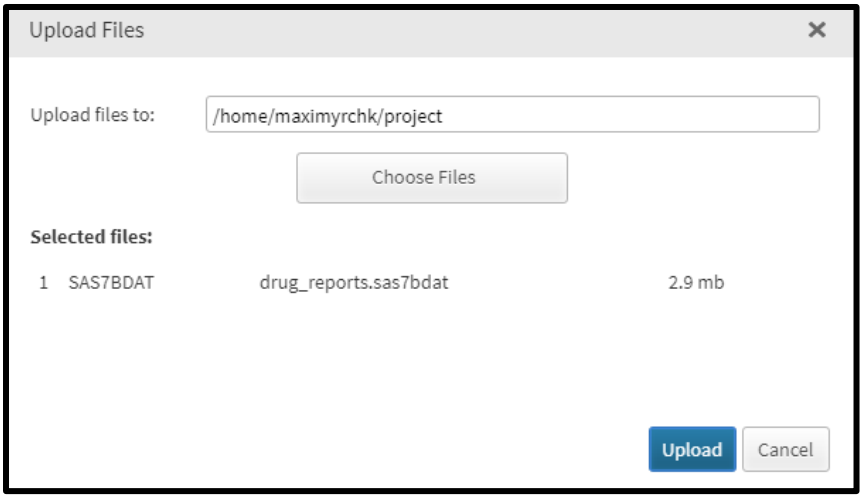


Рисунок 3.3 – загрузка датасету

Для початкового завантаження датасету (рисунок 3.3) у систему – користувач обирає файл та шлях у хмарному середовищі, куди необхідно зберегти дані. Після того, як дані завантажились, – необхідно написати SAS код, який виконає розмітку датасету у CAS пам’яті.

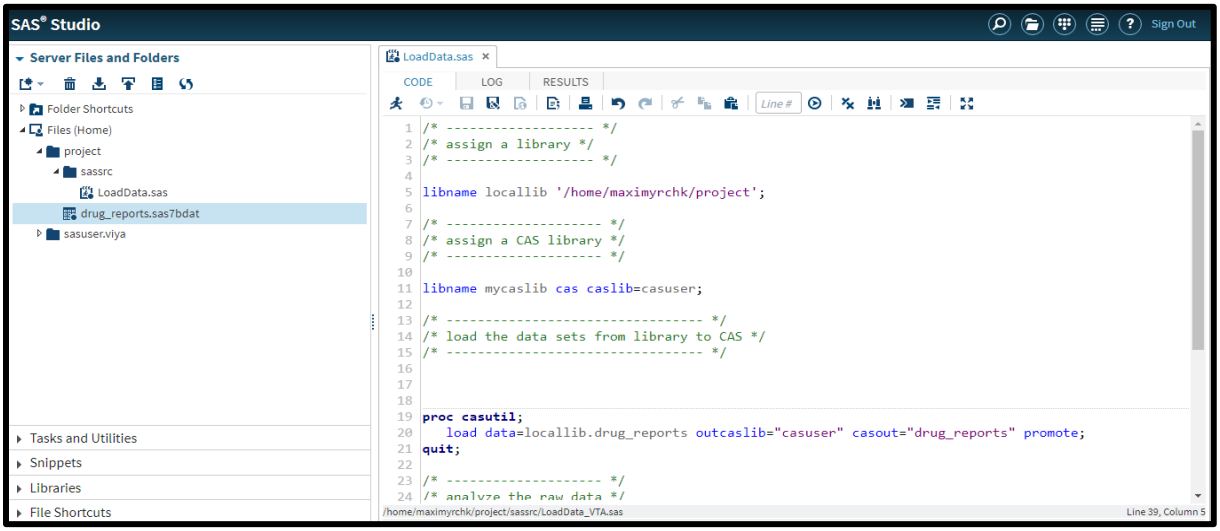
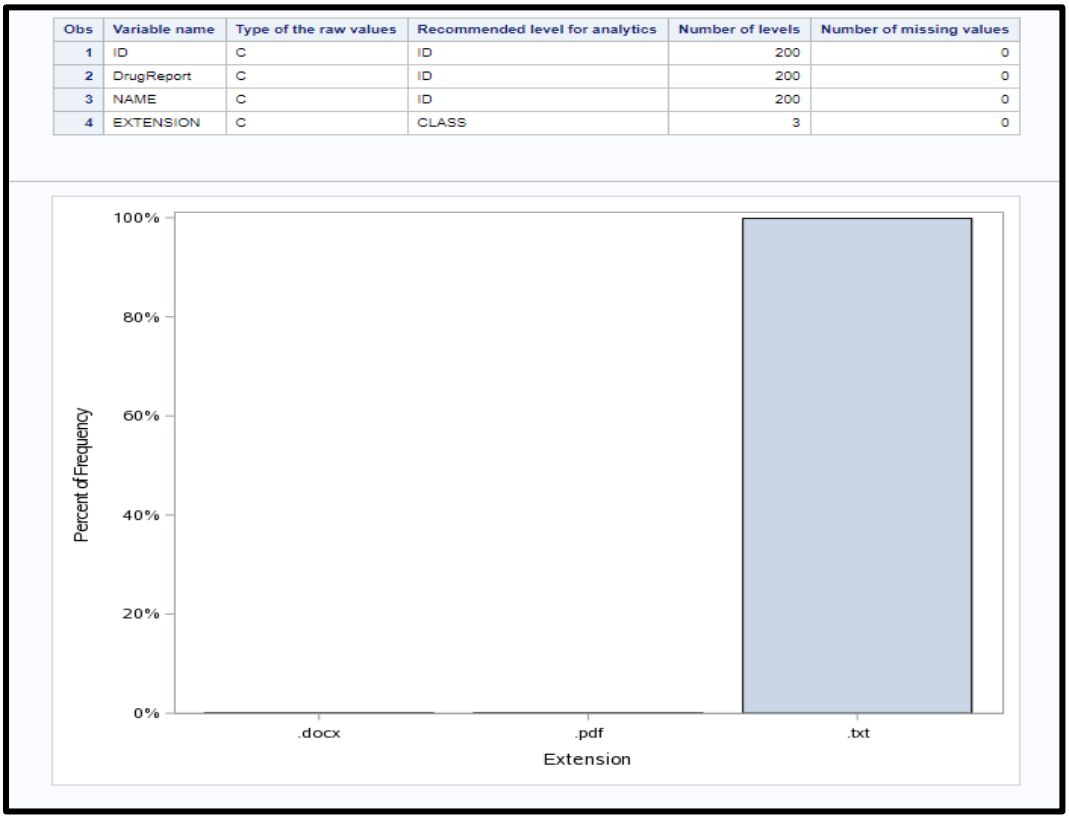


Рисунок 3.4 – процедура для розмітки та завантаження датасету у CAS пам’ять

Після запуску програми (рисунок 3.4) аналітику доступні результати у вигляді таблиці з метаданими та гістограми, яка вказує на розподіл документів за розширенням. У випадку датасету drug_reports.SAS7bdat майже всі документи були у текстовому форматі txt (рисунок 3.5). Розмітка виконалась успішно, пропущених значень немає.



Obs	Variable name	Type of the raw values	Recommended level for analytics	Number of levels	Number of missing values
1	ID	C	ID	200	0
2	DrugReport	C	ID	200	0
3	NAME	C	ID	200	0
4	EXTENSION	C	CLASS	3	0

Рисунок 3.5 – результат загрузки датасету у CAS пам’ять

Окрім цього користувач отримує таблицю з завантаженими даними та базовими характеристиками даних, такі як кардинальне число множини, розподіл документів за величиною.

Results: MYCASLIB.DRUG_REPORTS_SUMMARY_LEVELS

VARNAME	_INDEX_	_FREQ_	FREQPERCENT	NMISSPERCENT	_RAWNUM_	_RAWCHAR_	_FMTWIDTH_	_CFMT_
DrugReport	12	1	0.0707213579	0.0707213579	.	AWFUL withdrawl symptoms! I have been on this medication for about 4 years and am up to 225mg/day. If you miss one pill... your head feels like its skipping a beat, my lips get numb... it's CRAZY! I just don't know what to use though... I will be transitioning to another medication within a month! I am going to take Elevex for one week to relieve myself of the withdrawl sx... then I will be FREE! :)	1915	AWFUL withdrawl symptoms! I on this medication for about 4 y am up to 225mg/day. If you mis your head feels like its skipping lips get numb... it's CRAZY! I ju know what to use though... I wil transitioning to another medicat month! I am going to take Eleve week to relieve myself of the wi then I will be FREE! :)
DrugReport	13	1	0.0707213579	0.0707213579	.	Abidal . has just been increased today to 4 capsules	1915	Abidal . has just been increase capsules
DrugReport	14	1	0.0707213579	0.0707213579	.	Abidal 120 mg.daily works very well to relieve my 25-year depression. But it causes me terrible constipation. I have been having to take 10 Dulcolax tabs a day just to have BMs at all.	1915	Abidal 120 mg.daily works very relieve my 25-year depression. causes me terrible constipation been having to take 10 Dulcola just to have BMs at all.
DrugReport	15	1	0.0707213579	0.0707213579	.	Abidal caused me to become constipated. Once I stopped the drug, all was well again.	1915	Abidal caused me to become c Once I stopped the drug, all wa again.
DrugReport	16	1	0.0707213579	0.0707213579	.	Abidal did nothing for me. I was on 120 each day. I could do nothing during the day. I did not sleep at night. I was miserable. When the doctor started lowering my dose to take me off of it, I began feeling better. When I came off the Abidal completely, I ended up in the emergency room with high blood pressure, and I was sick for two weeks.	1915	Abidal did nothing for me. I was each day. I could do nothing du I did not sleep at night. I was mi When the doctor started lowerin to take me off of it, I began feeli When I came off the Abidal com ended up in the emergency roo blood pressure, and I was sick f weeks.
DrugReport	17	1	0.0707213579	0.0707213579	.	Abidal did nothing to help my depression or insomnia. It did however help the muscle and joint aches to some degree for a short timel	1915	Abidal did nothing to help my de insomnia. It did however help th and joint aches to some degree timel
DrugReport	18	1	0.0707213579	0.0707213579	.	Abidal did nothing to help with my depression. All it did was give me bad dreams on occasion, but didn't relieve my depression one bit.	1915	Abidal did nothing to help with m depression. All it did was give m dreams on occasion, but didn't depression one bit.

Рисунок 3.6 – сгенерована таблиця з завантаженими даними

Тепер дані проіндексовані та доступні для використання (рисунок 3.6). Наступним чином необхідно під’єднуєднати їх у Model Studio для побудови нової моделі.

У додатку Model Studio потрібно створити новий проект, описати його та обрати потрібний датасет з CAS пам’яті.

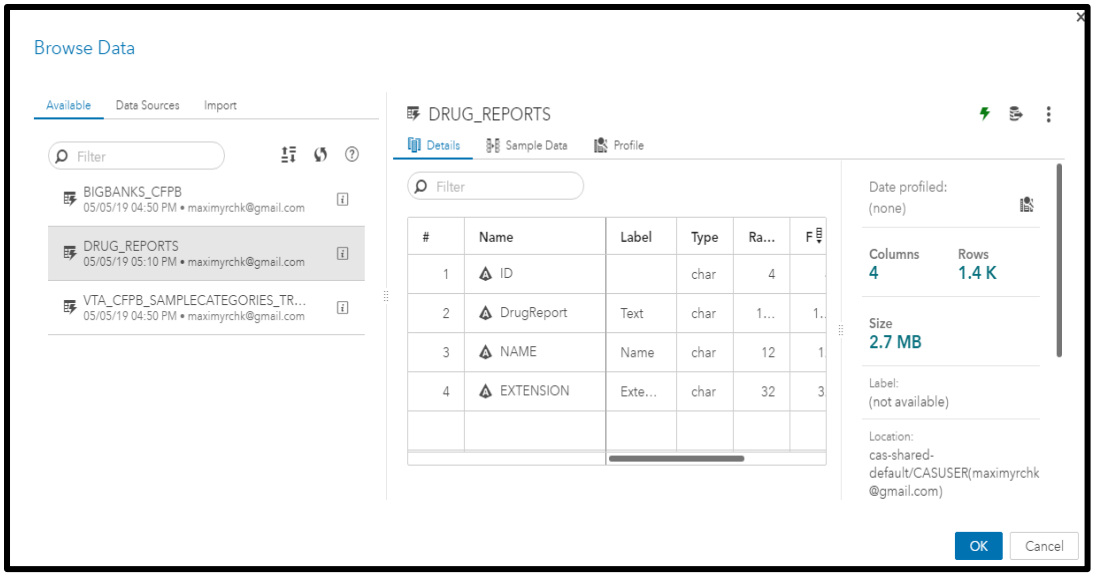


Рисунок 3.7 – перелік доступних даних до використання у проекті

Під час вибору даних користувачу відображається їх виглад та опис (рисунок 3.7). Обраний датасет має 1400 записів. Назва колонки DrugReport відповідає за текстовий зміст документів, Name, відповідно, за назву документу.

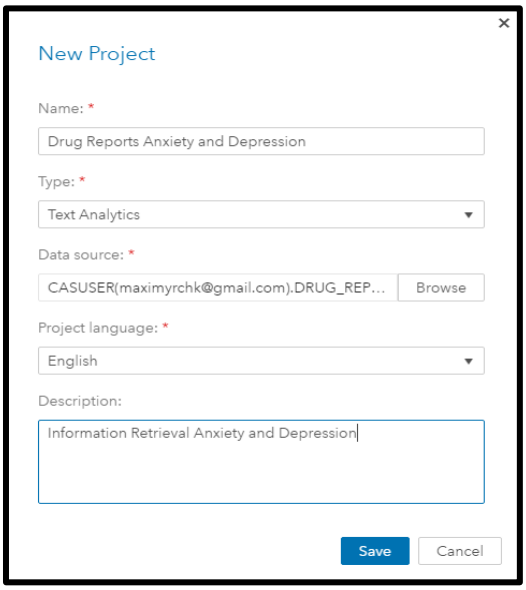


Рисунок 3.8 – створення нового проекту

Після створення (рисунок 3.8) проекту користувач має самостійно присвоїти роль тексту, який необхідно проаналізувати, відповідній змінній. В цьому випадку роль тексту відіграє змінна DrugReport (рисунок 3.9).

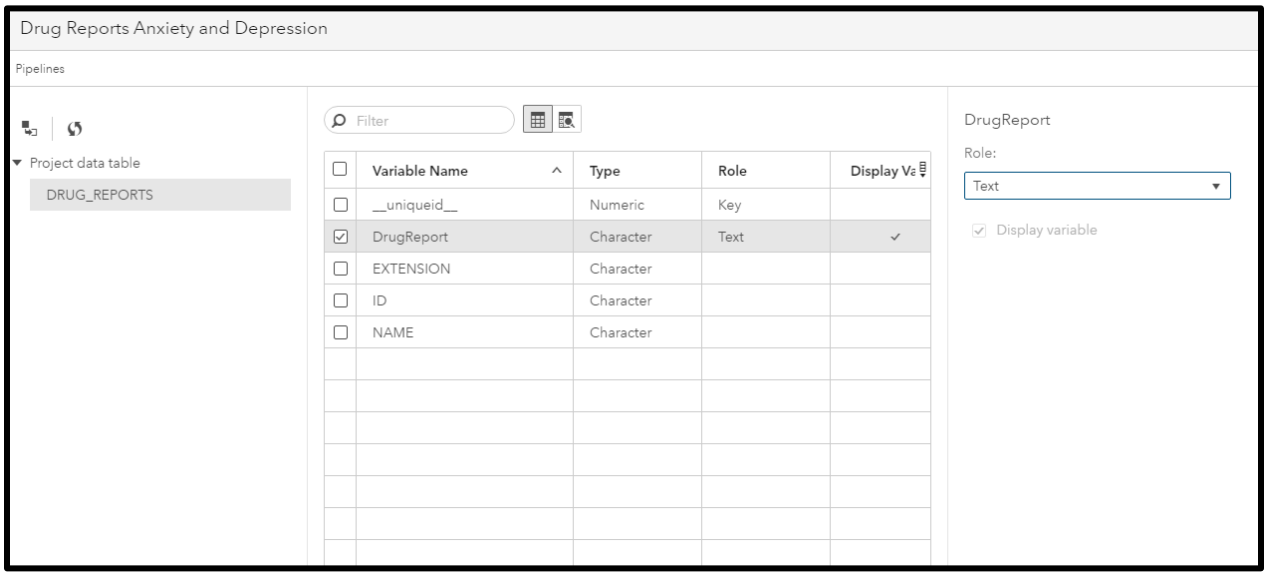


Рисунок 3.9 – призначення змінним ролей

3.3.2 Побудова моделі системи аналізу та категоризації

Основна частина створеної системи – модель аналізу і категоризації (рисунок 3.10), яка складається з 6-и послідовних частин: дані, виявлення понять, парсинг тексту, аналіз тональності, розбиття на тематики і далі на категорії.

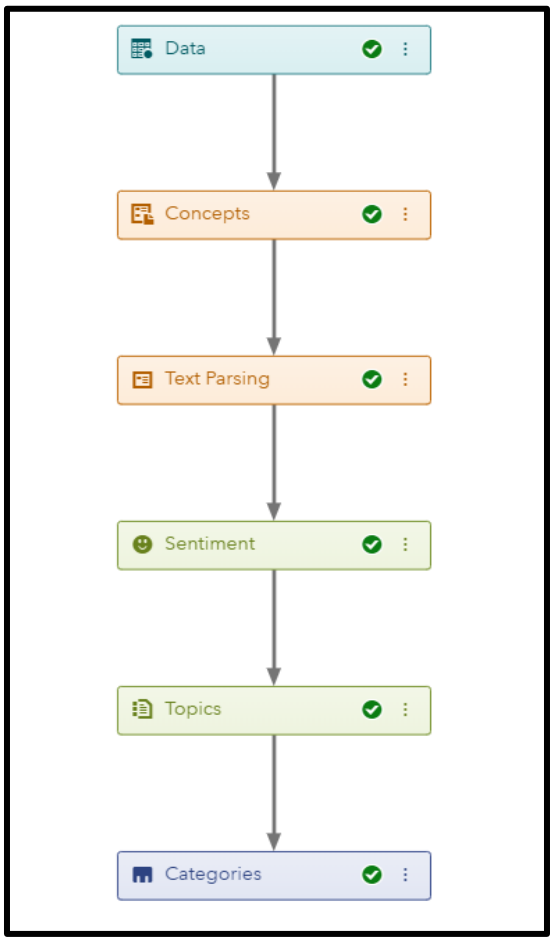


Рисунок 3.10 – базова модель

У компоненті SAS VTA Model Studio базова модель будується шляхом технології Drag And Drop (послідовного перетягування) основних вузлів

моделі. Після побудови графу – необхідно налаштувати кожен його компонент.

3.3.3 Виокремлення основних понять

В результаті натискання на вузол Concepts відкриється вікно з переліком 9 базових понять, таких як дата (nlpDate), гроші (nlpMoney), група іменників (nlpNounGroup), назва організації (nlpOrganization), проценти (nlpPercent), геолокація (nlpPlace) та час (nlpTime). Таким чином, до самостійного налаштування цієї компоненти, з тексту виділяються стандартні для будь-яких даних поняття. Щоб налаштувати даний вузол для існуючих даних аналітик повинен визначити основні поняття, які характеризують предмет його дослідження. У прикладі з медичними звітами було визначено 4 поняття: препарат (Medication), дозування (Dosage), лікарський рецепт (Prescription), та побічні ефекти (Side_Effects).

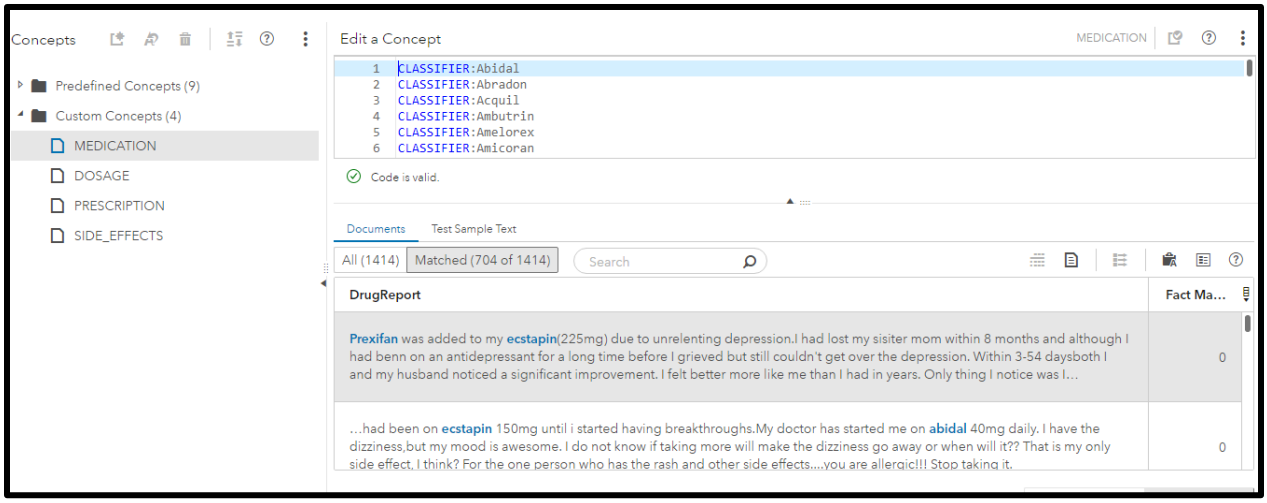


Рисунок 3.11 – поняття Medication

Концепція Medication (рисунок 3.11) включає в себе перелік різних назв медичних препаратів, які почерзі прописані за допомогою SAS коду [20]. В результаті оновленя вузла, це поняття зустрілось у 704 документах з 1414.

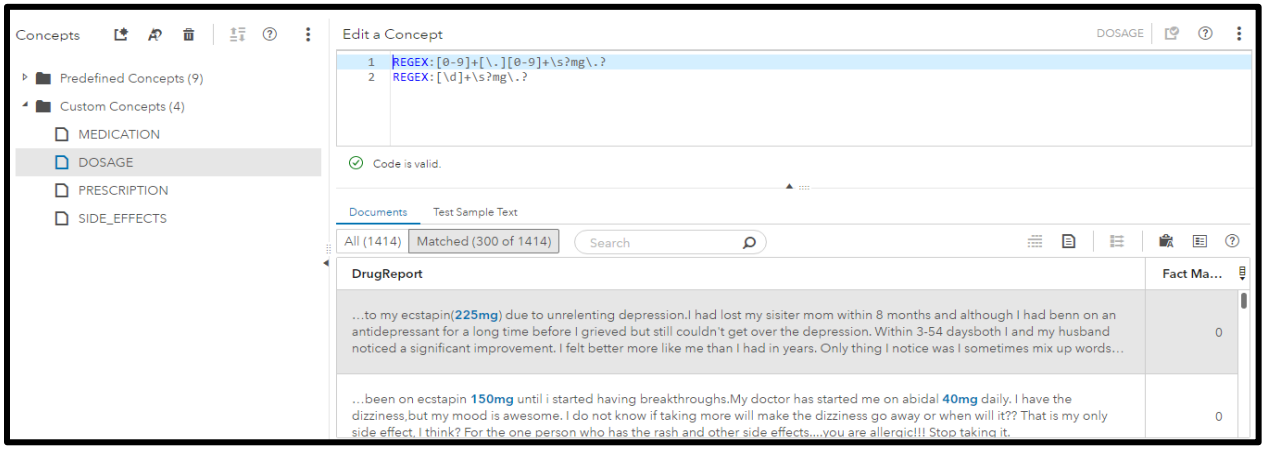


Рисунок 3.12 – поняття Dosage

Концепція Dosage прописана за допомогою REGEX запитів усіх можливих комбінацій дозування препаратів у міліграмах. На деякому прикладі даних було виявлено, що дозування представлене у тексті або у вигляді

десяткового дробу [0 − 9] + [\.][0 − 9] з надписом “mg” або ціле число [\d] з надписом “mg”. Як видно (рисунок 3.12), рекомендована доза була виявлена у 300 документах з 1414.

Рецепт лікаря, зазвичай, складається з поєднання препарату та його рекомендованої дози вживання, тому концепт Prescription прописаний як перелік поняття Medication та Dosage (рисунок 3.13). Всього таких комбінацій у колекції документів зустрілося 14.

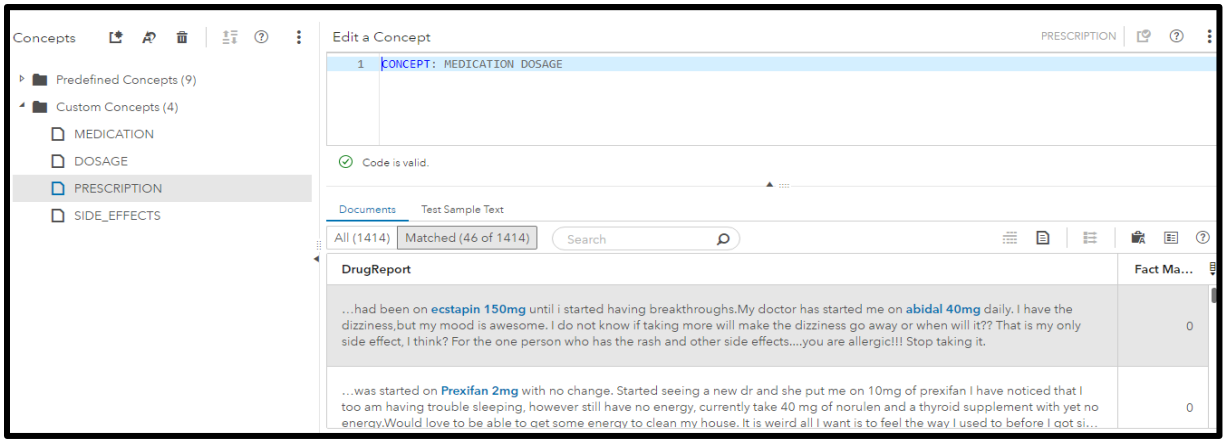


Рисунок 3.13 – поняття Prescription

Концепцію побічних дій препарату можливо задати як перелік усіх можливих випадків, таких як анемія, амнезія, агресія та інші (рисунок 3.14). Всього про побічні ефекти від препаратів йшлося у 406 документах.

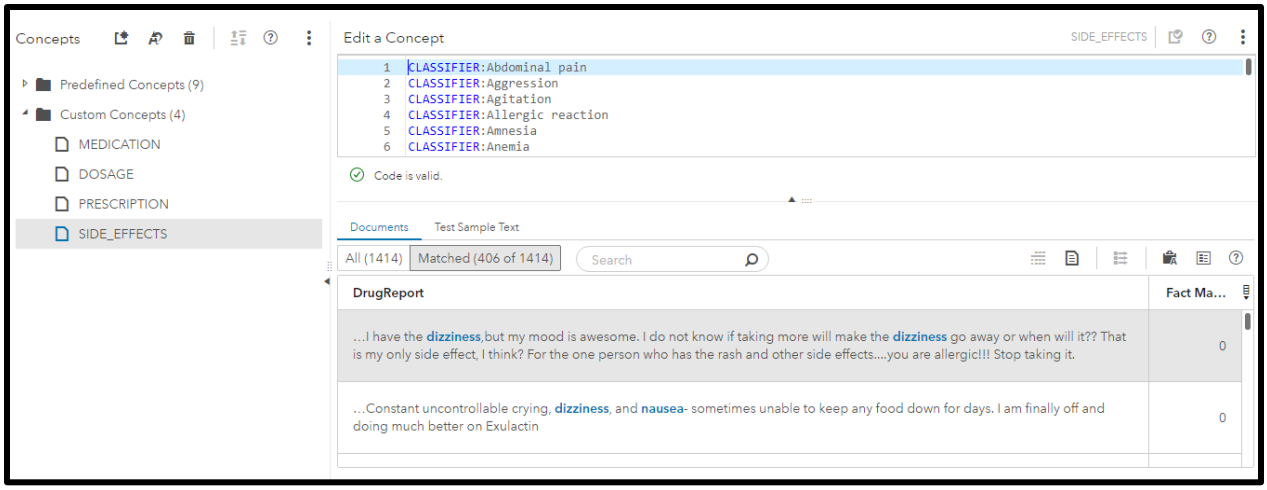


Рисунок 3.14 – поняття Side_Effects (побічні дії препарату)

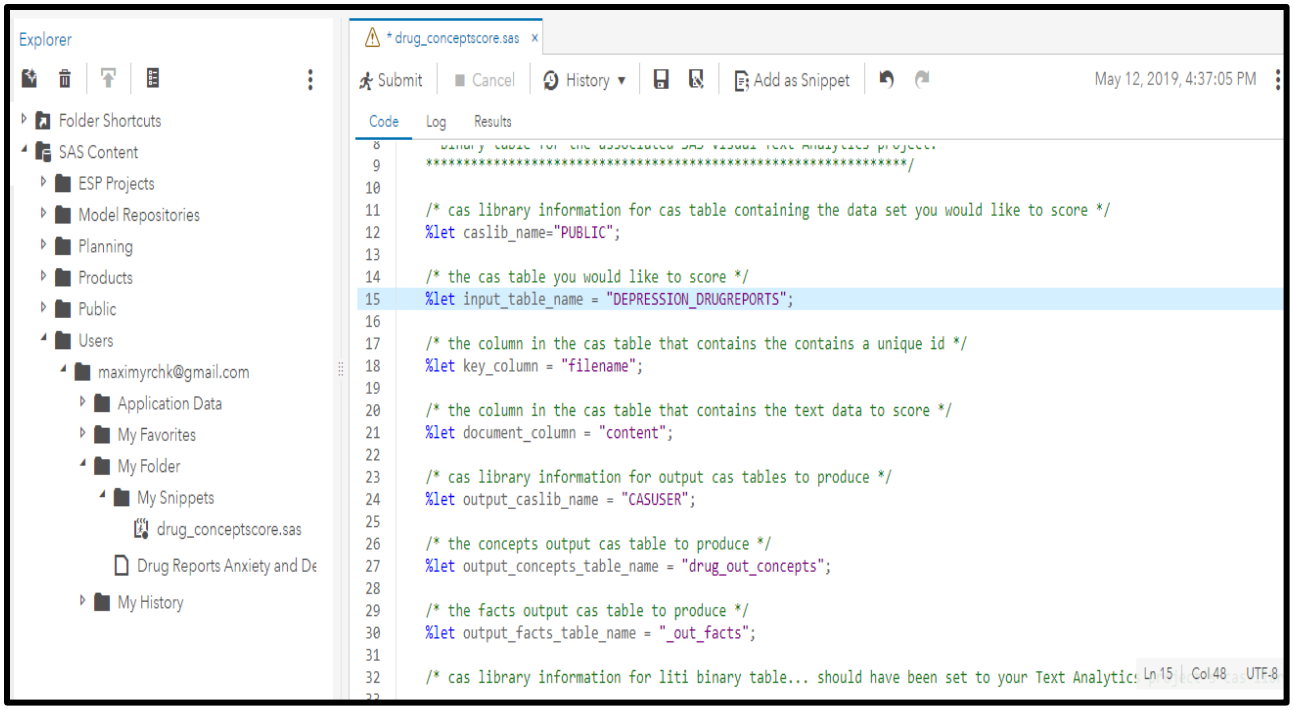


Рисунок 3.15 – SAS код для виокремлення понять

З налаштованої компоненти Concepts можна згенерувати SAS код (рисунок 3.15) для подальшого автоматизованого використання для нових колекцій документів. Для виокремлення ключових понять з медичних звітів необхідно лише додати шлях до колекції. Після запуску процедури отримуємо наступні результати (рисунок 3.16). Отримали 9820 записів з поняттями.

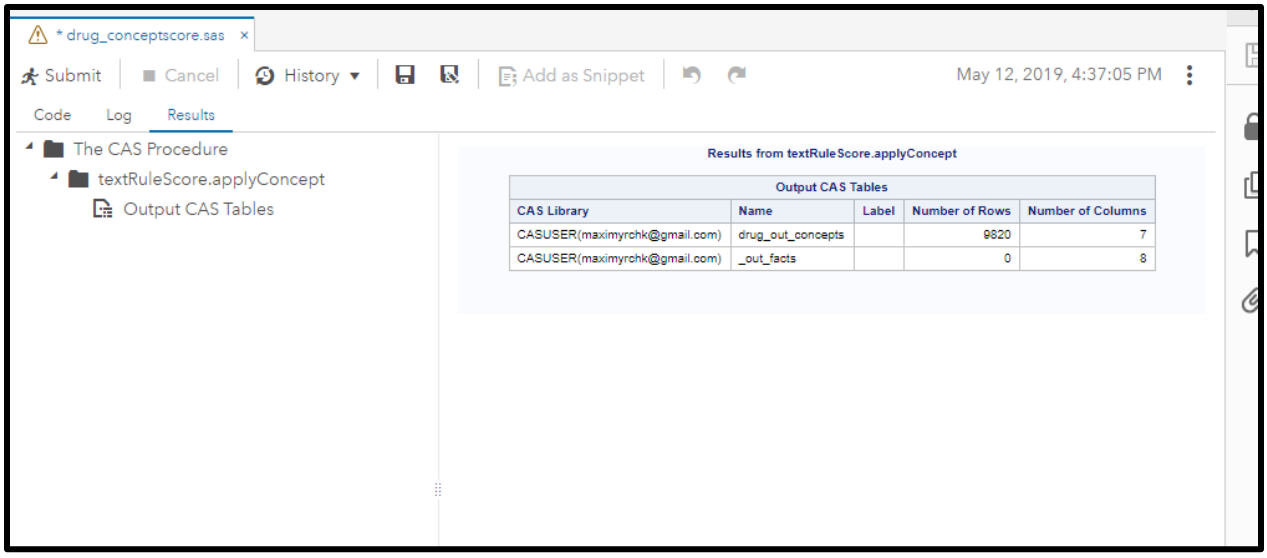


Рисунок 3.16 – результати виконання SAS коду для виокремлення понять

До результатів додається гістограма (рисунок 3.17), що зображує частотний розподіл виокремлених понять. Найбільше виявлено іменників.

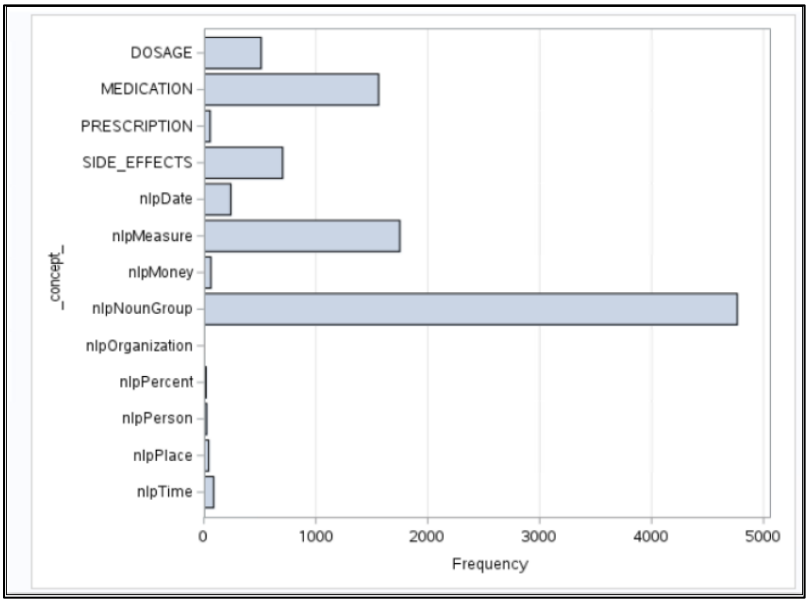


Рисунок 3.17 – результати процедури, гістограма

3.3.4 Парсинг тексту та аналіз зв’язків між термінами

Ключовим етапом у текстовій аналітиці являється витягнення, чистка та створення словника термінів з документів за допомогою NLP. Сюди входить визначення частин мови, стеммінг, парсинг витягнутих термінів для ідентифікації об’єктів, видалення стоп-слів (слів, які не впливають на суть) та перевірку правильності написання. Виділяються також змінні, що пов’язані з текстом (задані поняття на попередньому етапі).

Окрім парсингу тексту, відбувається перетворення текстуальних даних у числовий формат, у вигляді матриці термін-документ, використовуючи методи лінійної алгебри. Виокремлюються взаємозв’язки між термінами та їх сила. Після запуску вузла для кожного терміна доступна карта взаємозв’язків. Розглядаючи термін depression (рисунок 3.18) спостерігається найсильніший зв’язок з-поміж усіх препаратів – з препаратом abidal.

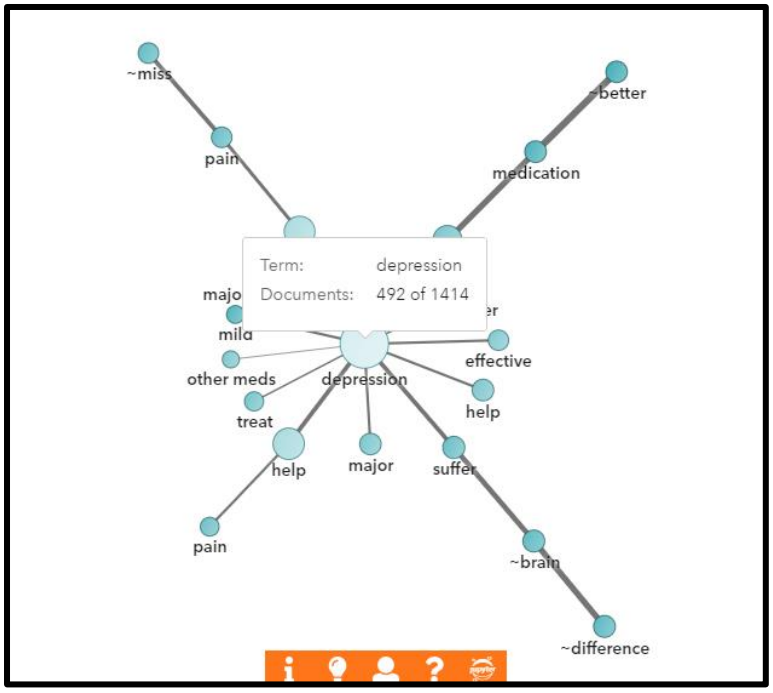


Рисунок 3.18 – зв’язки між терміном depression

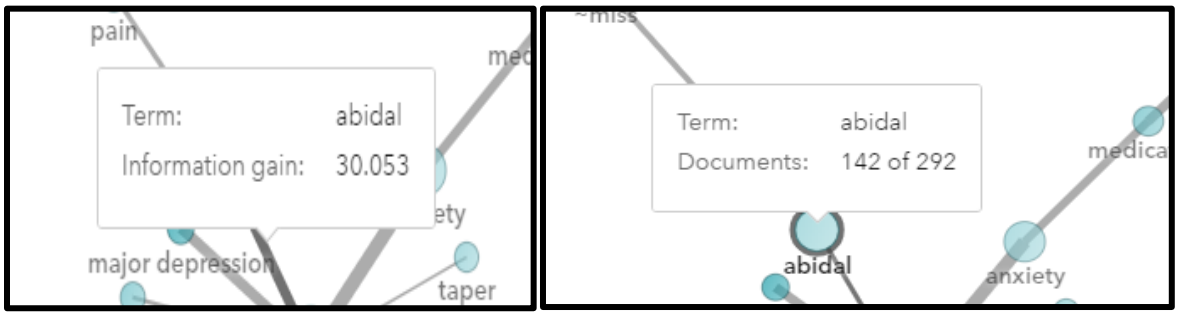


Рисунок 3.19 – препарат abidal

Згідно з картою, препарат трапляється у 142 документах (рисунок 3.19) та має розрахований показник 30.053, який відображає у кількісному вигляді скільки інформації було отримано про дану змінну, спостерігаючи змінну depression.

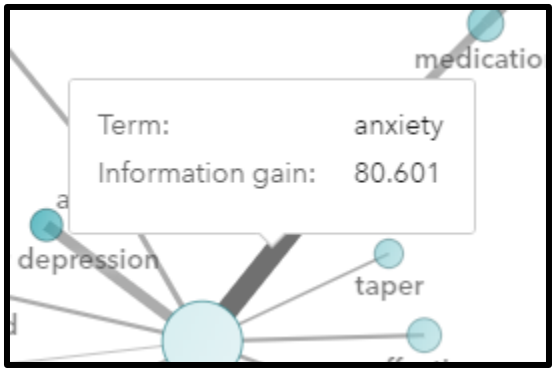


Рисунок 3.20 – термін anxiety

Найбільший інформаційний показник спостерігається у терміні anxiety, як показано на (рисунок 3.20), він складає значення 80.601.

Навпаки, розглядаючи карту терміну препарату abidal (рисунок 3.21) відстань Кульбека-Лейблера рівна 30.053 (рисунок 3.22), тобто обчислення повністю співпадають.

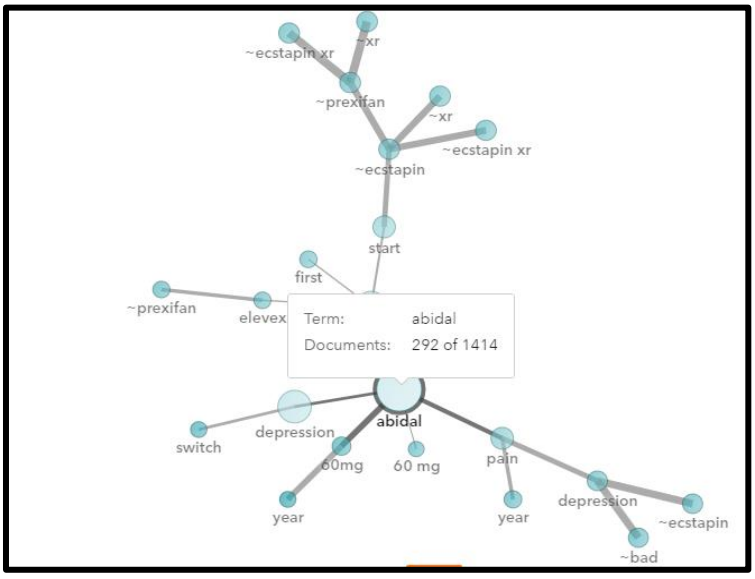


Рисунок 3.21 – карта зв’язків препарату abidal

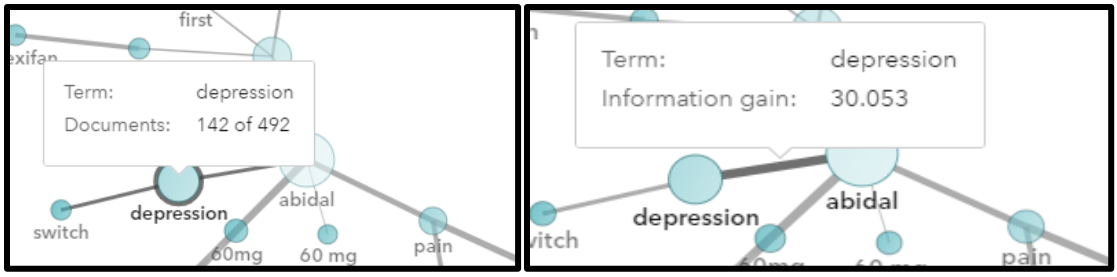


Рисунок 3.22 – показники депресії у карті зв’язків abidal

Продовжуючи аналізувати зв'язки з терміном депресія, користувачу доступний список з розрахованою мірою схожості інших слів до досліджуваного об’єкту.

Найбільший показник схожості (рисунок 3.23) з депресією має препарат imitap, 0.374, наступним є amelorex, 0.303.

Term	Similarity	Role	Documents	Frequency
abidal	0.262	MEDICATION	292	433
amelorex	0.303	MEDICATION	10	10
imitap	0.374	MEDICATION	5	6
noderall	-0.114	MEDICATION	4	4

Рисунок 3.23 – препарат з найбільшим показником схожості

Спостерігаючи у таблиці схожих термінів до препарату abidal, можна побачити найбільш поширені для цього медичного засобу побічні дії (рисунок 3.24), такі як безсоння, подразливість, головна біль. В даному випадку, найбільший показник має безсоння.

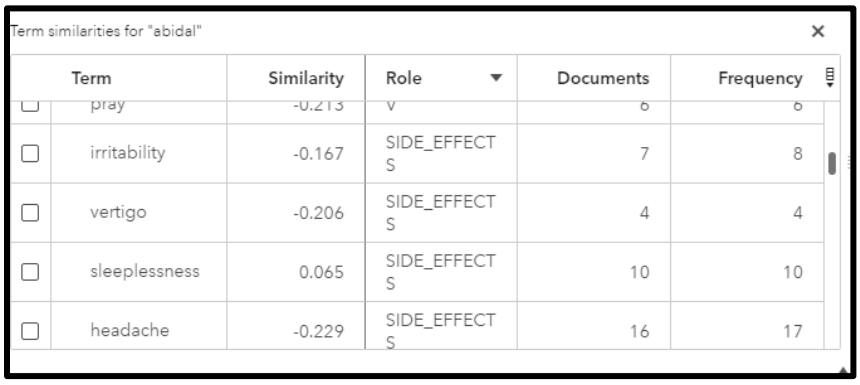


Рисунок 3.24 – побічні ефекти пов’язані з препаратом

3.3.5 Аналіз тональності та створення тематик

Вузол Sentiment відповідає за категоризацію настроїв, визначення полярності тексту: позитивної, негативної, нейтральної або змішанної, використовучи NLP.[21] На тональність (рисунок 3.25) тексту відображається у колонці Sentiment, навпроти тексту. Наступним чином виконується задача кластеризації, створення системою об’єдань з понять, які підвищуються до ролі теми. У колонці Topic аналітику доступний перелік створених тем. У колонці Documents відображається загальне число документів, що входять до конкретної теми.

Тематика з 5 термінів, що асоційовані з тематикою депресія: +допомога, +депресія, тревожність, +медицина та дійсно. Лише 161 з 1414 документів та 59 з 1522 термів співпали з тематикою, в той час, коли самостійний терм депресія був знайдений у 492 документах.

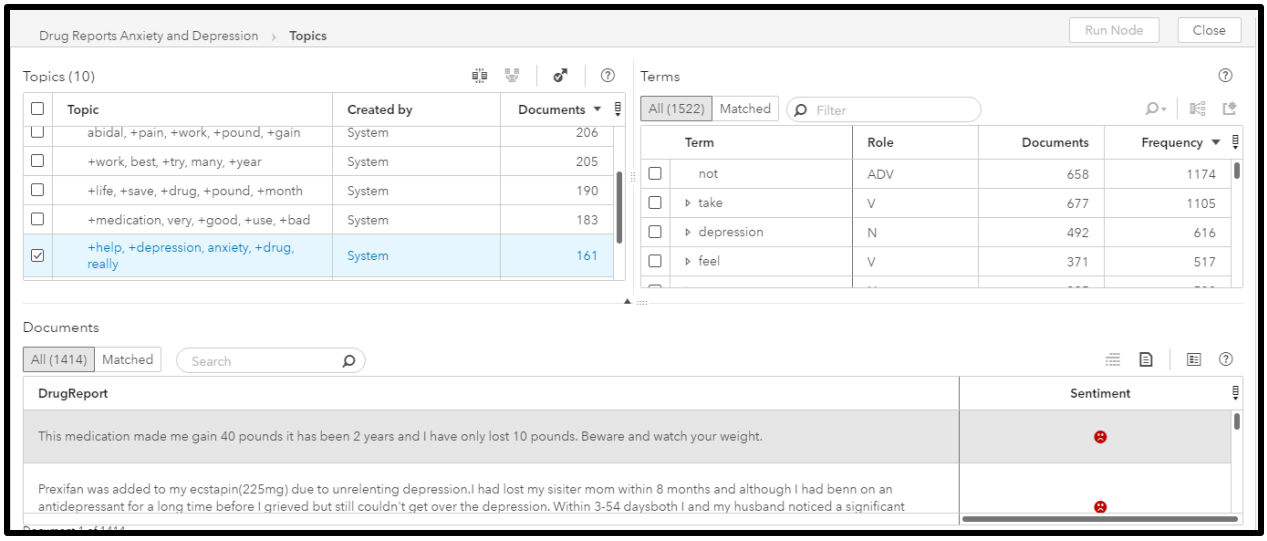


Рисунок 3.25 – зв’язки між термом depression

Щоб покращити результативність категоризації аналітику потрібно правильно налаштувати компоненту під той тип даних, який він досліджує. Один зі способів налаштування – обмежити кількість автоматичних тематик у задачі кластеризації та виставити параметри (рисунок 3.26), що відповідають за густину термінів (підвищення густини термінів означає зменшення кількості термінів, асоційованих з тематикою, але підвищення їх важливості у тематиці) та густину документів (підвищення параметру означає зменшення кількості документів, але підвищення їх релевантності до теми) .

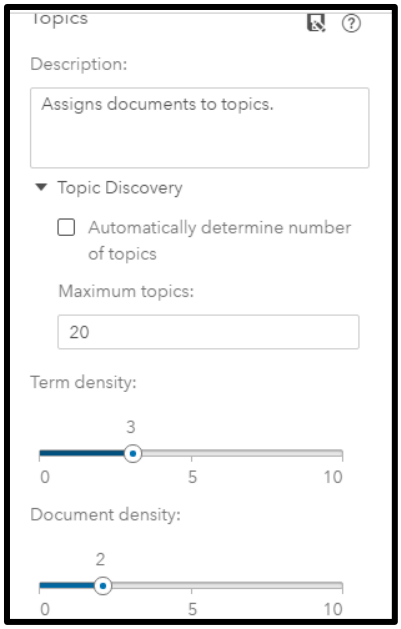


Рисунок 3.26 – параметри кластеризації

Збільшивши показник густини термінів до 3, показник густини документів до 2 та максимальну кількість тематик до 20, змінилась попередня тематика за вмістом термінів: +депресія, тривога, +допомога, естапін, паніка. Невпливові терміни відсутні, проте, кількість документів зменшилась до 58 (рисунок 3.27).

Кількість термінів, що відповідає темі: 20.

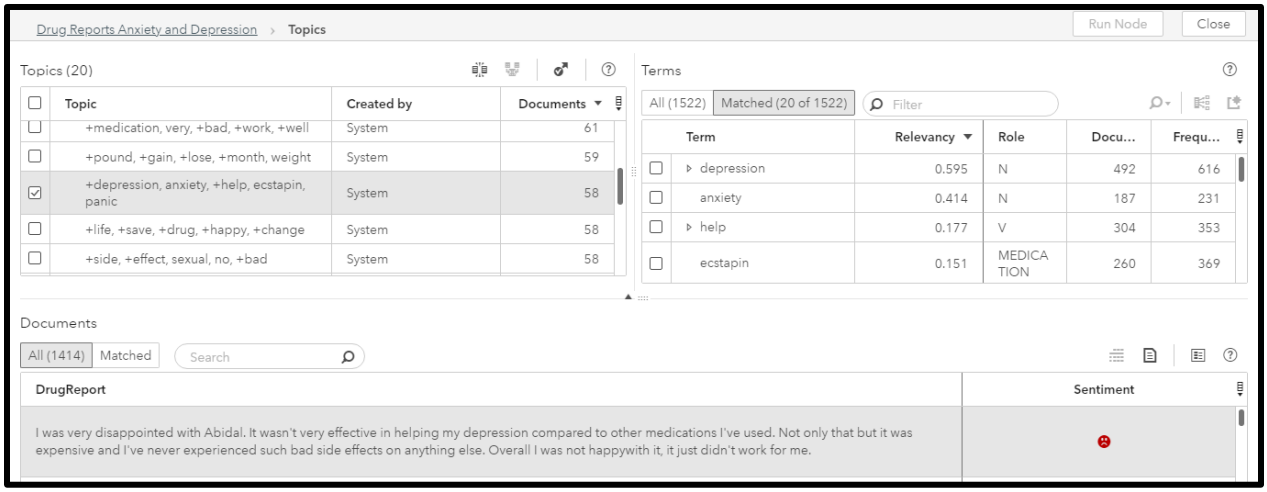


Рисунок 3.27 – зв’язки між термом depression

Також з'явилась нова тематика, пов'язана з терміном “depression”: still, +depress, +medicine, +think, +mood. До цієї тематики належать 57 (рисунок 3.28) документів та 21 термін. Підсумовуючи, збільшення максимальної кількості тематик не допомогло визначити більше пов’язаних з досліджуваним терміном документів.

Для можливого покращення результату, аналітик може створювати свої теми з вже створених системою. Оскільки 50 документів (рисунок 3.29) відповідають об'єднанню попередніх двох тематик, дана процедура не має перевагу у випадку завантажених даних та інших медичних даних, що мають схожу структуру.

В такому випадку, аналітик також має можливість створити повністю свої тематики. Було виконано запит “Anxiety” у вікні пошуку термів, в результаті отримали 2 терми: anxiety та severe anxiety. Підвищення комбінації цих термінів до тематики дало результат у 187 відповідних документів. На запит “Depress” було розпізнано 23 терміни. Об’єднання їх до тематики відповідає 632 документам.

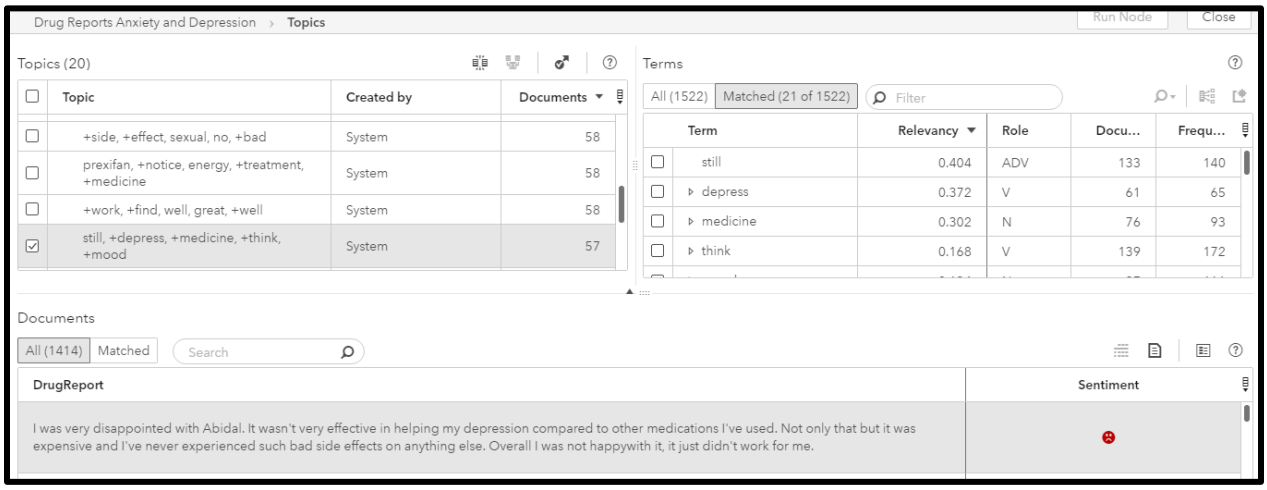


Рисунок 3.28 – нова створена системою тематика

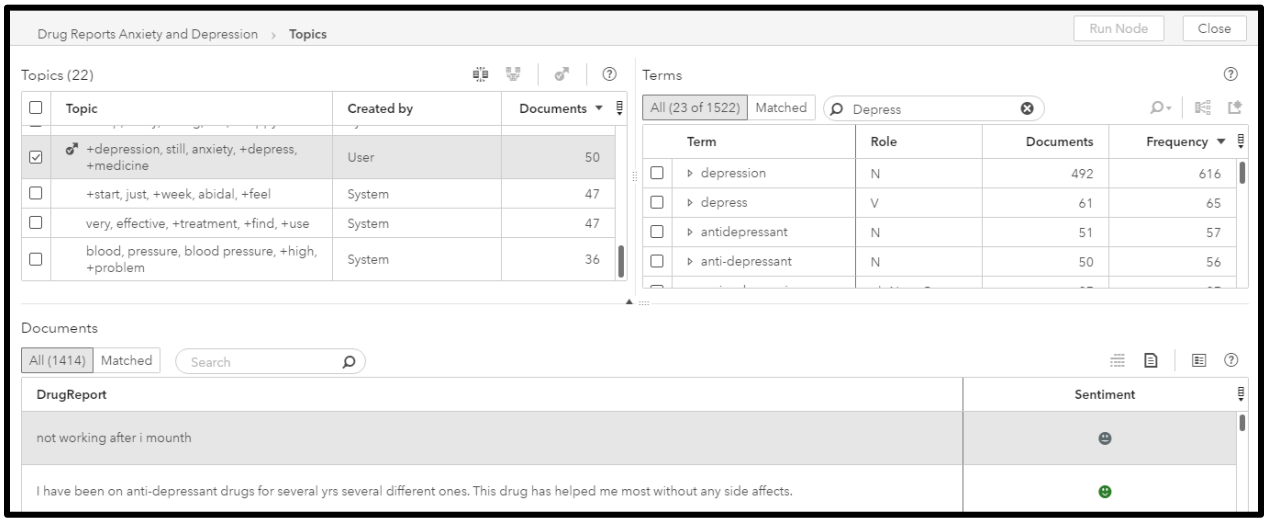


Рисунок 3.29 – об’єднання попередніх тем створених системою

Наступним кроком було об'єднання двох створених тем (рисунок 3.30). Отримана в результаті цього тематика відповідає 683 знайденим документам. Цей результат вже відповідає дійсності.

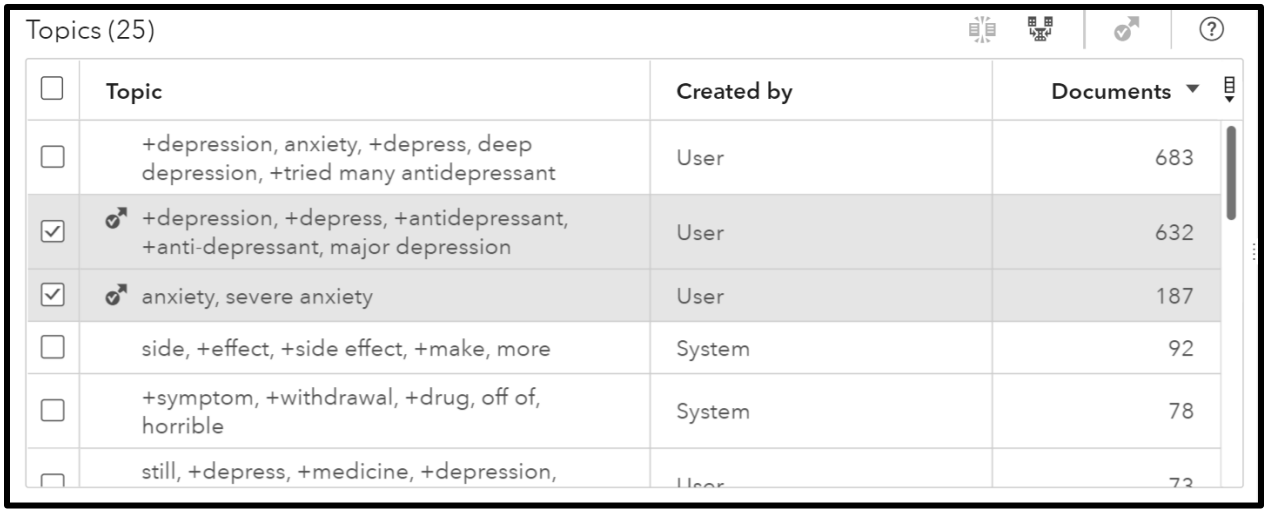


Рисунок 3.30 – зв’язки між термом depression

Метод	Швидкість реалізації	Кількість категоризованих документів	Швидкість при застосуванні
Метод максимальної правдоподібності	1	3 - 114	2
LSA	2	2 - 161	3
Самостійно створені правила	3	1 - 683	1

Таблиця 3.1 – аналіз застосованих методів для виокремлення тематик

3.3.6 Категоризація медичних звітів

Таким чином, створені тематики (таблиця 1) дали найкращі результати. Існує сенс використовувати їх для категоризації. Після натиску на кнопку: “підвищення до категорії” із тематики створюється категорія. Було підвищено три теми, створено три категорії, які можна звести до назв: “Депресія”, “Тревожність”, “Депресія та тривожність”.

Після цих дій у необхідно відкрити вузол “Categorization” й проаналізувати отримані результати. При натисканні на список категорій користувачу доступний список підвищених раніше тем. Оскільки категорії задаються булевими правилами (рисунок 3.31) – є можливість відредагувати їх у вбудованому редакторі.

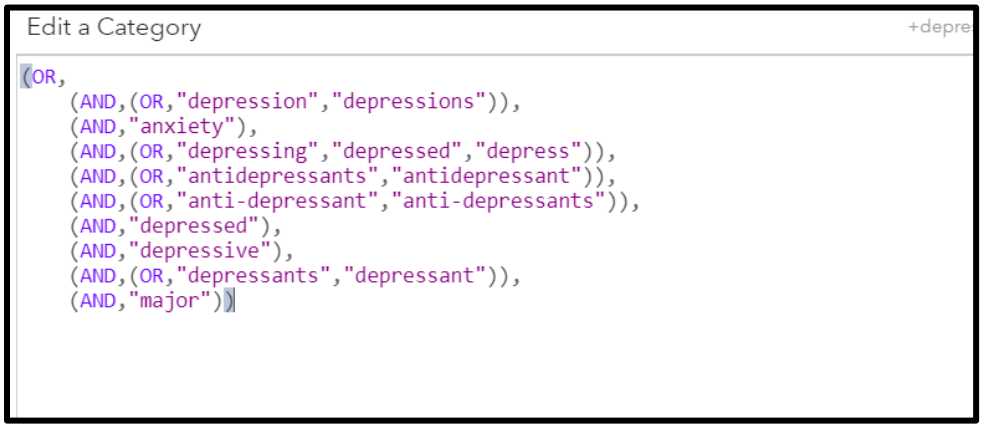


Рисунок 3.31 – булеві правила

Також аналітик може побудувати повністю самостійно нові категорії, користуючись методикою булевих правил.

3.4 Висновок до розділу 3

Спроектована та розроблена система дозволяє налаштувати автоматизований пошук в поданій на вхід колекції медичних звітів та розподіл документів, пов'язаних з депресією та тривожністю. До отриманих результатів категоризації аналітику додаються результати аналізу тональності документів та результати аналізу зв'язків між ключовими термінами. Так, наприклад, користувач може швидко побачити, які побічні дії препарату частіше трапляються у користувачів препарату, що залишили відгук, який препарат частіше використовують для конкретного захворювання: депресії, тривожності чи об'єднання цих категорій.

Вирішені ПП задачі можуть знайти застосування у компаніях, що працюють у фармацевтичній сфері, або для незалежних дослідників. Дуже важливим є те, що система може працювати автоматизовано, подаючи постійно на вхід відгуки пацієнтів. Це набагато зменшує час для опрацювання даних, та дає змогу компаніям дуже швидко реагувати на задоволеність чи незадоволеність клієнтів, швидко звертати увагу на проблемні сторони продукту для їх вирішення.

Система розроблена у хмарному середовищі SAS Viya, що надає змогу користуватись їй з будь-якої операційної системи, встановивши тільки браузер. Використання хмарного середовища пов’язано з тим, що можна мати всі можливості програмних продукті
в SAS, але використовувати їх та виконувати складні обчислення, обробляючи велику кількість даних, на майже довільному ПК.

Також у розділі був повністю описаний процес налаштування системи автоматизованої категоризації та аналізу для того, щоб була можливість використовувати систему для усіх необхідних для дослідження захворювань.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі проводиться оцінка основних характеристик програмного продукту. Даний продукт розроблений на мові програмування SAS Base в якості додатка на базі хмарних обчислень в програмному продукті SAS Viya. Програма в першу чергу призначена для аналізу та категоризації текстової медичної звітності, може використовуватися самостійно, або бути частиною експертної системи.

Нижче наведено аналіз різних варіантів реалізації модулю з метою вибору оптимального, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. Як прямі, так і побічні витрати розподіляються по продуктам та послугам у залежності від потрібних на кожному етапі виробництва обсягів ресурсів. Виконані на цих етапах дії у контексті метода ФВА називаються функціями.

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів, виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. У даному випадку – аналізу функцій програмного продукту й виявлення усіх витрат на реалізацію цих функцій.

Фактично цей метод працює за таким алгоритмом:

- визначається послідовність функцій, необхідних для виробництва продукту. Спочатку – всі можливі, потім вони розподіляються по двом групам: ті, що впливають на вартість продукту і ті, що не впливають.
- для кожної функції визначаються повні річні витрати й кількість робочих годин.
- для кожної функції на основі оцінок попереднього пункту визначається кількісна характеристика джерел витрат.
- після того, як для кожної функції будуть визначені їх джерела витрат, проводиться кінцевий розрахунок витрат на виробництво продукту.

4.1 Постановка задачі техніко-економічного аналізу

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

- програмний продукт повинен функціонувати на сучасних обчислювальній машині, з параметрами не нижчини за наступні: частота процесора 1.5Ггц, оперативна пам'ять 4Гб, місце на жосткому диску 50Мб;
- забезпечувати зручність і простоту встановлення на будь-яку апаратну систему та поєднання з іншими програмними системами(Linux, Windows та їх аналоги);

- передбачати мінімальні витрати на впровадження програмного продукту.

4.1.1 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка програмного продукту, дозволяє оцінювати ефективність навчання. Виходячи з конкретної мети, можна виділити наступні основні функції ПП:

F_1 – вибір мови програмування;

F_2 – вибір типу інтерфейсу програми;

F_3 – вибір моделі взаємодії.

Кожна з основних функцій може мати декілька варіантів реалізації.

Функція F_1:

а) мова програмування SAS Base;

б) мова програмування Python.

Функція F_2:

а) інтерфейс командного рядку;

б) віконний інтерфейс.

Функція F_3:

а) виконання обчислень на машині користувача;

б) використання клієнт-серверної архітектури.

4.1.2 Варіанти реалізації основних функцій

Варіанти реалізації основних функцій показані у морфологічній карті системи (рис. 4.1). На основі цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій (таблиця 4.1).

Рисунок 4.1 – Морфологічна карта

Морфологічна карта відображує всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів.

Таблиця 4.1 – Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
F1	*А*	Висока швидкодія, наявність багатьох реалізованих процедур, кросплатформеність	Потребує додаткове програмне забезпечення
	Б	Наявність бібліотек, з реалізаціями більшості методів для аналізу даних	Вищий поріг входу

F2	А	Менша вартість розробки, швидша розробка	Складність при тестуванні, вищий поріг входу користувачів
	Б	Нижчий поріг входу користувачів	вища вартість розробки
F3	А	Універсальність використання, вартість розробки	Необхідність оновлення клієнтської програми у випадку додавання нового офунціоналу, потребує потужні машини у користувача для обробки великих даних
	Б	Можливість додавання нових функцій без оновлення клієнтської програми користувачів, можливе користування малопотужного ПК, час обробки	Необхідність наявності інтернет-з'єднання, вартість розробки

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція *F1:*

Оскільки обидві мови можуть бути використані для розробки і пропонують різні переваги, то слід розглянути обидва варіанти.

Функція *F2:*

Оскільки для даного продукту важливим є низький поріг входу для користувача – відкидаємо а).

Функція *F3:*

Оскільки для даного продукту важливою є універсальність застосування та швидкість виконання, використаємо варіант б) як єдиний можливий.

Таким чином, будемо розглядати такий варіант реалізації ПП:

1. F1а – F2б – F3б
2. F1б – F2б – F3б

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.2 Обґрунтування системи параметрів ПП

4.2.1 Опис параметрів

На підставі даних про основні функції, що повинен реалізувати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- *X1* – швидкодія мови програмування;
- *X2* – об’єм пам’яті для коректної роботи програми;

- *X3* – час виконання обрахунків;
- *X4* – потенційний об’єм програмного коду.

X1: Відображає швидкодію операцій залежно від обраної мови програмування.

X2: Відображає необхідний для збереження та обробки даних об’єм оперативної пам’яті пристрою.

X3: Час, який витрачається на виконання обрахунків.

X4: Показує розмір програмного коду, який необхідно створити розробнику.

4.2.2 Кількісна оцінка параметрів

Головна функція F_0 – розробка програмного продукту, який кількісно оцінює відсоток покриття навчального курсу екзаменаційними білетами. На підставі даних про основні функції, що повинен реалізувати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня. На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій слід відкинути, адже, вони не відповідають задачам що були поставлені при розробці програмног о продукту. Можливі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію програмного продукту як показано у таблиці 4.2.

Таблиця 4.2 – Основні параметри ПП

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	нс/Оп	350	100	1
Об’єм пам’яті для коректної роботи	X2	Гб	2	1	0,5
Час виконання обрахунків	X3	мс	3000	2300	1500
Потенційний об’єм програмного коду	X4	кількість рядків коду	1000	500	250

За даними таблиці 4.2 будуються графічні характеристики параметрів – рисунок 4.2 – рисунок 4.5.

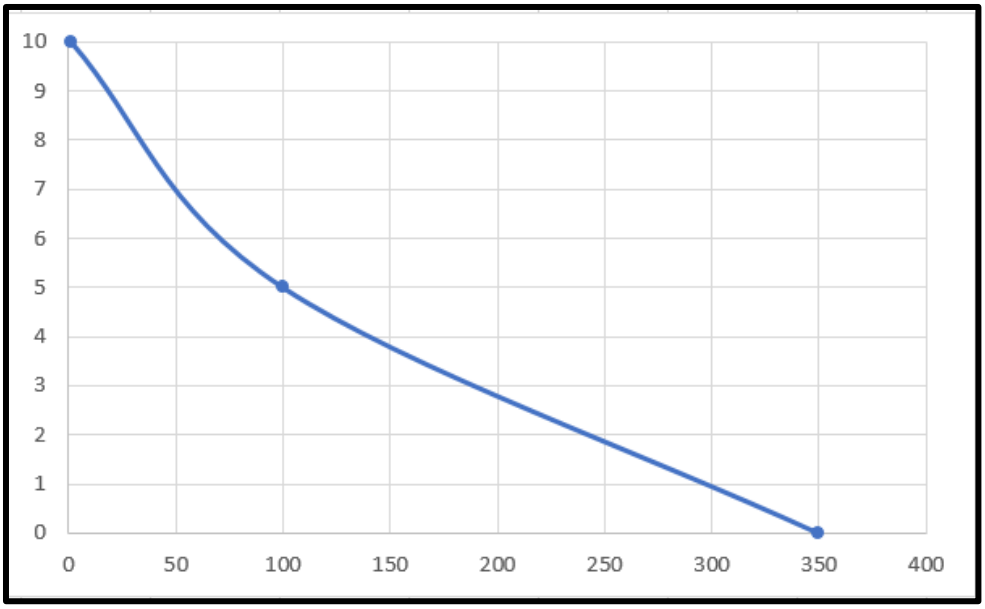


Рисунок 4.2 – X1, швидкодія мови програмування

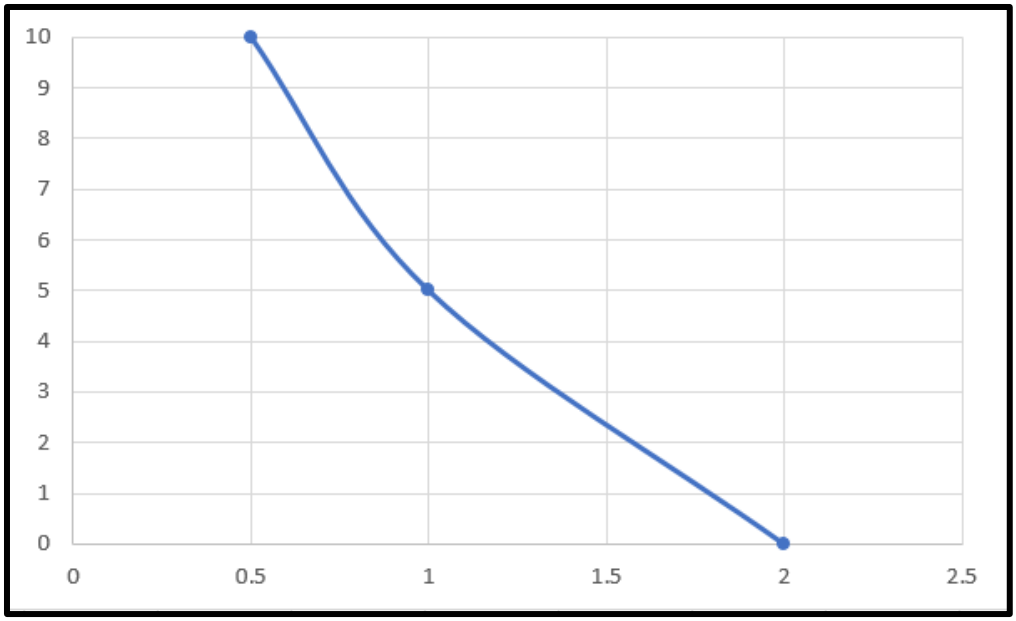


Рисунок 4.3 – X2, об’єм пам’яті для коректної роботи

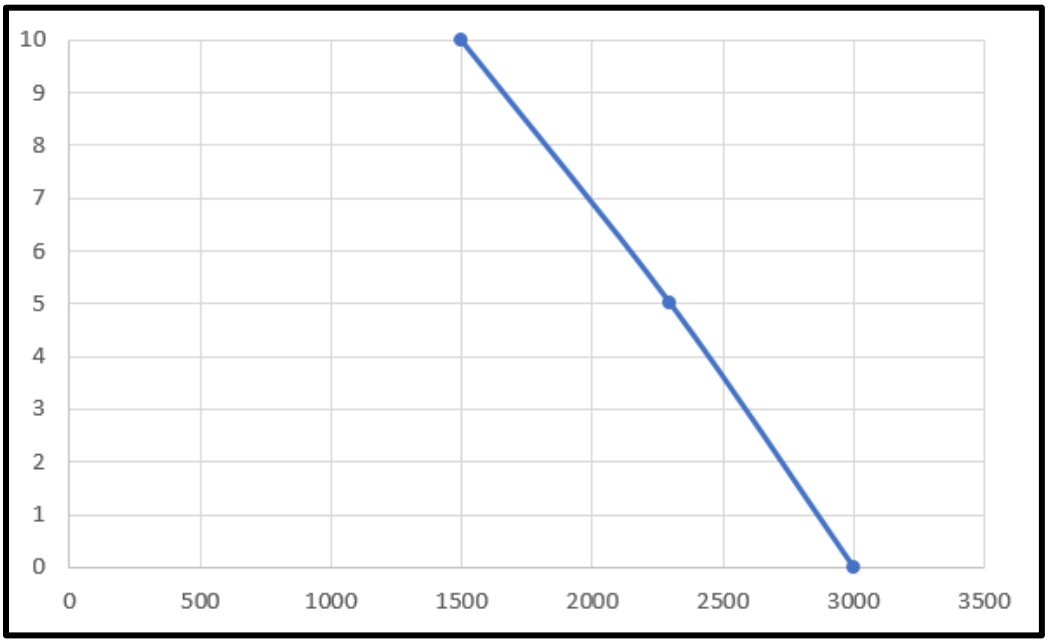


Рисунок 4.4 – X3, час виконання навчання

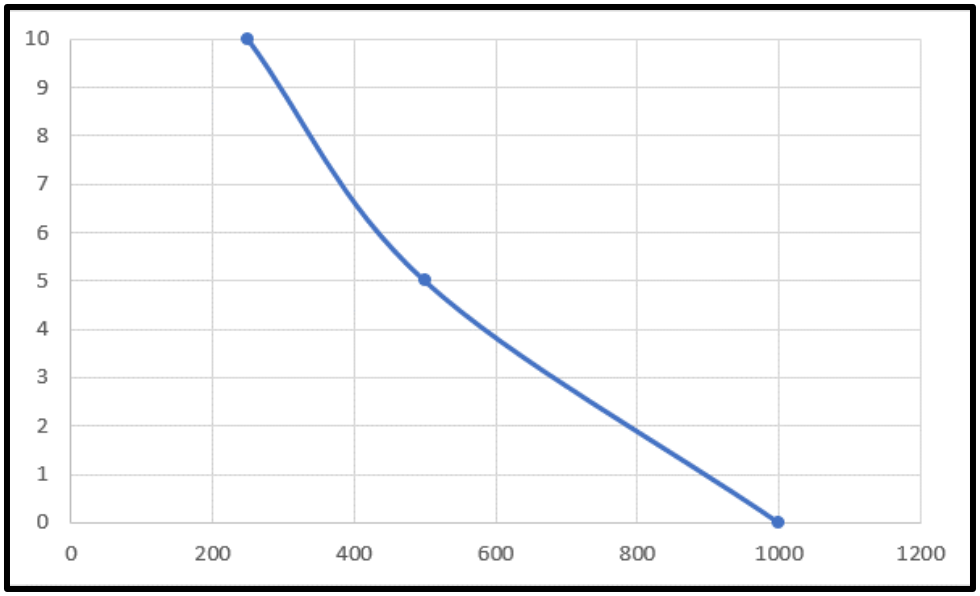


Рисунок 4.5 – X4, потенційний об’єм програмного коду

4.2.3 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту для системи аналізу і категоризації медичних текстових даних.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;

- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Таблиця 4.3 – Результати ранжування параметрів

Познач. параметра	Назва параметра	Одини ці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхи-лення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
X1	Швидкодія мови програмування	нс/Оп	2	2	1	2	1	1	2	11	-6,5	42.5
X2	Об’єм пам’яті для коректної роботи	Мб	3	3	4	3	4	3	3	23	5,5	30.25
X3	Час виконання обрахунків	Мс	1	1	2	1	2	2	1	10	-7,5	56.25
X4	Потенційний об’єм програмного коду	к-сть рядків коду	4	4	3	4	3	4	4	26	8,5	72.5
	Разом		10	10	10	10	10	10	10	70	0	201.5

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{i=1}^{N} r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70$$

де N – число експертів, n – кількість параметрів;

б) середня сума рангів:

$$T = \frac{\mathrm{R_{ij}}}{\mathrm{n}} = 17{,}5.$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T$$

Сума відхилень по всім параметрам повинна дорівнювати 0;

г)загальна сума квадратів відхилення:

$$S = \sum_{\mathrm{i=1}}^{\mathrm{N}} \Delta_{\mathrm{i}}^2 = 201{,}5.$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 201{,}5}{7^2(4^3 - 4)} = 0{,}822 > W_k = 0{,}67$$

Рангування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, який дорівнює 0,67.

Скориставшись результатами рангування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4.

Таблиця 4.4 – Попарне порівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	>	>	>	>	>	>	>	>	1,5
X1 і X3	<	<	>	<	>	>	<	<	0,5
X1 і X4	<	<	<	<	<	<	<	<	0,5
X2 і X3	<	<	<	<	<	<	<	<	0,5
X2 і X4	>	>	<	>	<	>	>	>	1,5
X3 і X4	>	>	>	>	>	>	>	>	1,5

Числове значення, що визначає ступінь переваги *i*–го параметра над *j*–тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1{,}5 \text{ при } X_i > X_j \\ 1.0 \text{ при } X_i = X_j \\ 0.5 \text{ при } X_i < X_j \end{cases}$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості $K_{вi}$ за наступними формулами:

$$K_{\text{вi}} = \frac{b_i}{\sum_{i=1}^{n} b_i}, \text{ де } b_i = \sum_{i=1}^{N} a_{ij}.$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%).На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{\text{вi}} = \frac{b_i'}{\sum_{i=1}^{n} b_i'} \text{,де } b_i' = \sum_{i=1}^{N} a_{ij} b_j.$$

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 – Розрахунок вагомості параметрів

Параметриx_i	Параметриx_j				Перша ітер.		Друга ітер.		Десята ітер	
	X1	X2	X3	X4	b_i	$K_{\text{вi}}$	b_i^1	$K_{\text{вi}}^1$	b_i^2	$K_{\text{вi}}^2$
X1	1	1,5	0,5	0,5	3,5	0,21875	13,25	0,217213	579315	0.217129
X2	0,5	1	0,5	1,5	3,5	0,21875	13,25	0,217213	579315	0.217129
X3	1,5	1,5	1	1,5	5,5	0,34375	21,25	0,348361	930119	0.348612

X4	1,5	0,5	0,5	1	3,5	0,21875	13,25	0,217213	579315	0.217129
Всього:					16	1	61	1	2.66806e+06	1

4.3 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів *X2*(об’єм необхідної оперативної пам’яті) та *X3* (Час виконання обрахунків) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра *X1 (*швидкодія мови програмування*)* обрано не найгіршим (не максимальним), тобто це значення відповідає або варіанту а) 100 нс/Оп або варіанту б) 1 нс/Оп.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 4.6):

$$K_K(j) = \sum_{i=1}^{n} K_{вi,j} B_{i,j} ,$$

де n – кількість параметрів;$K_{вi}$– коефіцієнт вагомості i–го параметра; B_i – оцінка i–го параметра в балах.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F3(X2)	Б	1300	1,4	0.217129	0.3039806
F2(X3)	Б	3	2,3	0.348612	0.8018076
F1(X1, X4)	А	600	5	0.217129	1.085645
	Б	3000	2.56	0.217129	0.55585024

За даними з таблиці 4.6 за формулою

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \ldots + K_{\text{ТУ}}[F_{zk}],$$

визначаємо рівень якості кожного з варіантів:

K_{K1} = 0.3039806 + 0.8018076+ 1.085645 = 2.1914332

K_{K2} = 0.3039806 + 0.8018076+ 0.55585024= 1.66163844

Як видно з розрахунків, кращим є перший варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.4 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;

2. Розробка програмної оболонки;

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 – до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань. Загальна трудомісткість обчислюється як

$$T_O = T_P \cdot K_П \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \qquad (5.1)$$

де T_P – трудомісткість розробки ПП; $K_П$ – поправочний коефіцієнт; $K_{СК}$ – коефіцієнт на складність вхідної інформації; K_M – коефіцієнт рівня мови програмування; $K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм; $K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: T_P = 90 людино-днів. Поправочний коефіцієнт,

який враховує вид нормативно-довідкової інформації для першого завдання: $К_П = 1.6$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: $К_{СК} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $К_{СТ} = 0.8$. Тоді, за формулою 5.1, загальна трудомісткість програмування першого завдання дорівнює:

$$Т_1 = 90 \cdot 1.6 \cdot 0.8 = 112,2 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $Т_Р = 28$ людино-днів, $К_П = 0.7, К_{СК} = 1, К_{СТ} = 0,8$:

$$Т_2 = 28 \cdot 0,7 \cdot 0,8 = 15,68 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$Т_I = (112,2 + 15,68 + 4,8 + 15,68) \cdot 8 = 1186,88 \text{ людино-годин;}$$
$$Т_{II} = (112,2 + 15,68 + 6,91 + 15,68) \cdot 8 = 1203,76 \text{ людино-годин;}$$

Найбільш високу трудомісткість має варіант II.

В розробці бере участь один програміст, з окладом 12000 грн. Визначимо зарплату за годину за формулою:

$$С_Ч = \frac{М}{T_m \cdot t} грн.,$$

де М – місячний оклад працівників;T_m – кількість робочих днів тиждень;t – кількість робочих годин в день.

$$СЧ = \frac{12000}{1 * 21 * 8} = 71,4$$

Тоді, розрахуємо заробітну плату за формулою

$$С_{ЗП} = С_ч \cdot Т_i \cdot К_Д,$$

де $С_ч$– величина погодинної оплати праці студента; $Т_i$ – трудомісткість відповідного завдання; $К_Д$ – норматив, який враховує додаткову заробітну плату.

Зарплата розробників становить:

I. $С_{ЗП} = 71,4 \cdot 1186,88 \cdot 1,2 = 101691$

II. $С_{ЗП} = 71,4 \cdot 1203,76 \cdot 1,2 = 103138$

Відрахування на соціальний внесок становить 22,0%:

I. $С_{ВІД} = С_{ЗП} \cdot 0,22 = 101691 \cdot 0,22 = 22372$

II. $С_{ВІД} = С_{ЗП} \cdot 0,22 = 103138 \cdot 0,22 = 22690$

Тепер визначимо витрати на оплату однієї машино-години. ($С_М$)

Так як одна ЕОМ обслуговує одного програміста з окладом 12000 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$С_Г = 12 \cdot М \cdot К_З = 12 \cdot 12000 \cdot 0,2 =$ 28800 грн.

З урахуванням додаткової заробітної плати:

$С_{ЗП} = С_Г \cdot (1 + К_З) = 12000 \cdot (1 + 0.2) = 14400$ грн.

Відрахування на соціальний внесок:

$С_{ВІД} = С_{ЗП} \cdot 0.22 = 14400 * 0,22 =$ 3168 грн.

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 25000 грн.

$С_А = К_{ТМ} \cdot К_А \cdot Ц_{ПР} = 1.15 \cdot 0.25 \cdot 25000 = 7187,5$ грн.,

де $К_{ТМ}$– коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача; $К_А$– річна норма амортизації; $Ц_{ПР}$– договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$С_Р = К_{ТМ} \cdot Ц_{ПР} \cdot К_Р = 1.15 \cdot 25000 \cdot 0.05 = 1437,5$ грн.,

де $К_Р$– відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$Т_{ЕФ}$ =($Д_К$ – $Д_В$ – $Д_С$ – $Д_Р$) · $t_З$· $К_В$ = (365 – 104 – 8 – 16) · 8 · 0.9 = 1706,4 годин,

де $Д_К$ – календарна кількість днів у році; $Д_В$, $Д_С$ – відповідно кількість вихідних та святкових днів; $Д_Р$ – кількість днів планових ремонтів устаткування; t –кількість робочих годин в день; $К_В$– коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$С_{ЕЛ}$ = $Т_{ЕФ}$· $N_С$· $К_З$·$Ц_{ЕН}$ = 1706,4 · 0,22 · 0,78 · 2,7515= 805,69грн.,

де $N_С$ – середньо-споживча потужність приладу; $К_З$– коефіцієнтом зайнятості приладу; $Ц_{ЕН}$ – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$С_Н$ = $Ц_{ПР}$·0.67 = 25000 · 0,67 = 16750 грн.

Тоді, річні експлуатаційні витрати будуть:

$С_{ЕКС}$ =$С_{ЗП}$+ $С_{ВІД}$+ $С_А$ + $С_Р$+ $С_{ЕЛ}$ + $С_Н$

$С_{ЕКС1}$ = 14400 + 3168+ 7187,5 + 1437,5 + 805,69+ 16750 = 43748,69 грн.

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$С_{М-Г1}$ = $С_{ЕКС}$/ $Т_{ЕФ}$ = 43748,69 / 1706,4 = 25,63 грн/год.

Оскільки в даному випадку всі роботи, які пов‘язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу складають:

$C_М = C_{М\text{-}Г} \cdot T$

I. $C_М = 25{,}63 \cdot 1186{,}88 = 30419{,}7$ грн.

II. $C_М = 25{,}63 \cdot 1203{,}76 = 30852{,}36$ грн.

Накладні витрати складають 67% від заробітної плати:

$C_Н = C_{ЗП} \cdot 0{,}67$

I. $C_Н = 101691 \cdot 0{,}67 = 68132{,}97$ грн.

II. $C_Н = 103138 \cdot 0{,}67 = 69102{,}46$ грн.

Отже, вартість розробки ПП становить:

$C_{ПП} = C_{ЗП} + C_{ВІД} + C_М + C_Н$

I. $C_{ПП} = 84743{,}23 + 18643{,}51 + 28392{,}67 + 68132{,}97 = 199912{,}38$ грн.

II. $C_{ПП} = 85948{,}46 + 18908{,}66 + 28793{,}94 + 69102{,}46 = 202753.52$ грн.

4.5 Вибір кращого варіанта ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$К_{ТЕРj} = К_{Кj} / Спп,$

$К_{ТЕР1} = 2.1914332/199912{,}38 = 0.00001096196$

$К_{ТЕР2} = 1.66163844/202753.52 = 0.00000819536;$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $К_{ТЕР1} = 0.00001096196$.

4.6 Висновки до розділу 4

В даному розділі проведено повний функціонально-вартісний аналіз програмного продукту, що було розроблено в рамках дипломного проекту. Процес аналізу можна умовно розділити на дві частини.

В першій з них проведено дослідження програмного продукту з технічної точки зору: було визначено основні функції програмного продукту та сформовано множину варіантів їх реалізації; на основі обчислених значень параметрів, а також експертних оцінок їх важливості було обчислено коефіцієнт технічного рівня, який і дав змогу визначити оптимальний, з технічної точки зору, шлях реалізації програмного продукту.

Другу частину ФВА присвячено вибору із альтернативних варіантів реалізації найбільш економічно обґрунтованого. Порівняння запропонованих варіантів реалізації в рамках даної частини виконувалось за коефіцієнтом

ефективності, для обчислення якого були обчислені такі допоміжні параметри, як трудомісткість, витрати на заробітну плату, накладні витрати.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $К_{ТЕР} = 1.096 \cdot 10^{-5}$.

Цей варіант реалізації програмного продукту має такі параметри:

- мова програмування – SAS Base;
- інтерфейс SAS Viya;
- виконання обчислень на стороні сервера.

Даний варіант програмного продукту дає користувачам можливість проводити аналіз та категоризацію текстових медичних даних на будь-якій операційній системі, що має браузер.

ВИСНОВКИ

У даній роботі були детально розглянуті задачі інформаційного пошуку та інтелектуального аналізу, а саме задача категоризації, задача аналізу взаємозв'язків та задача аналізу тональності тексту й застосовані для дослідження текстових медичних даних. Проаналізувавши теоретичні матеріали, методи обробки тексту, було розроблено автоматичну систему аналізу і категоризації для текстової медичної звітності. В якості практичного застосування системи, був розроблений додаток з використанням SAS технологій у системі SAS Viya Visual Text Analytics. У даній системі було реалізовано нормалізацію тексту, а саме лематизацію й відсічення стоп-слів, створення моделі векторного простору, виконання REGEX запитів для пошуку важливих понять, таких як, назва препарату, дозування, лікарський рецепт та побічні ефекти, проведення латентно-семантичного аналізу для дослідження зв’язків між термінами та отримання тематик, виконано кластеризацію з застосуванням методу максимальної правдоподібності.

Результати дипломної роботи:

1. Досліджені математичні моделі та методи, що використовуються для різноманітної обробки тексту;
2. Досліджені проблеми у медичній галузі, що можна розв’язати автоматизованою системою інформаційного пошуку або інтелектуальним аналізом;
3. Реалізована задача розробки автоматизованої системи інформаційного пошуку;
4. Проведений аналіз взаємозв'язків між термінами та аналіз тональності, що було важливо для виявлення препаратів, що найсильніше пов'язані з

лікуванням депресії, а також які препарати дають позитивний ефект. Також, проведено аналіз побічних ефектів, що найчастіше трапляються з прийомом того, чи іншого препарату;

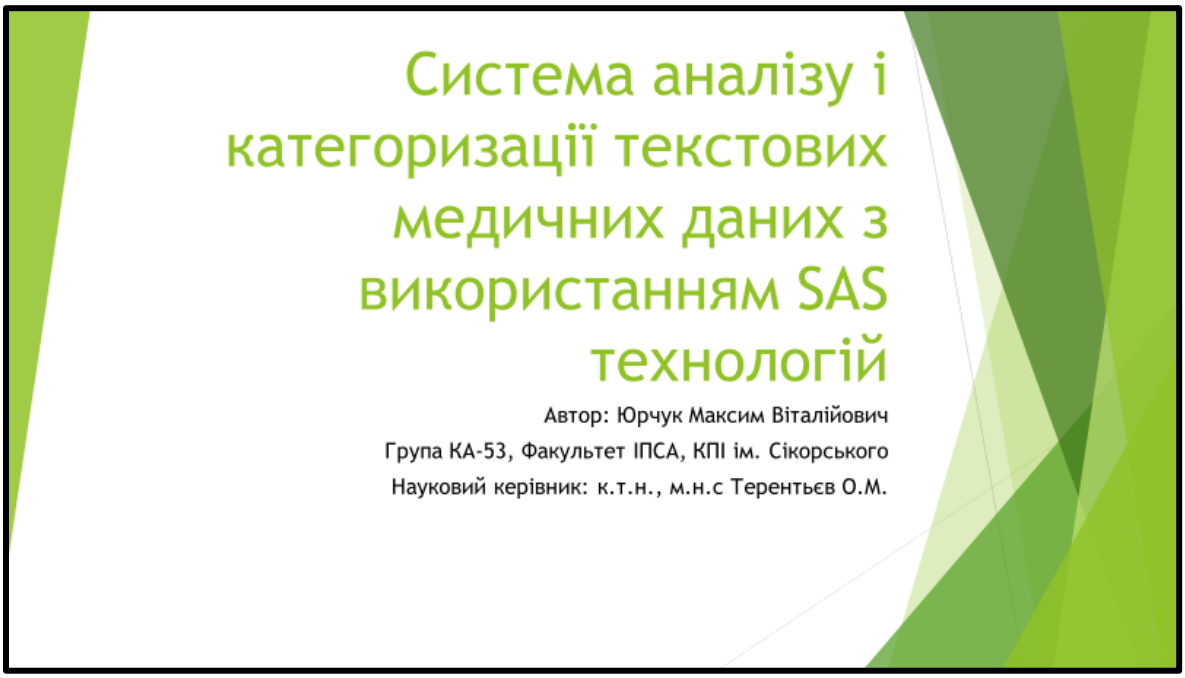
5. В результаті дослідження, покращені згенеровані системою булеві правила для категоризації документів. Після чого, кількість знайдених документів, що підлягають загальній категорії, пов’язаній з депресією та тривожністю, - зросла у 3 рази.

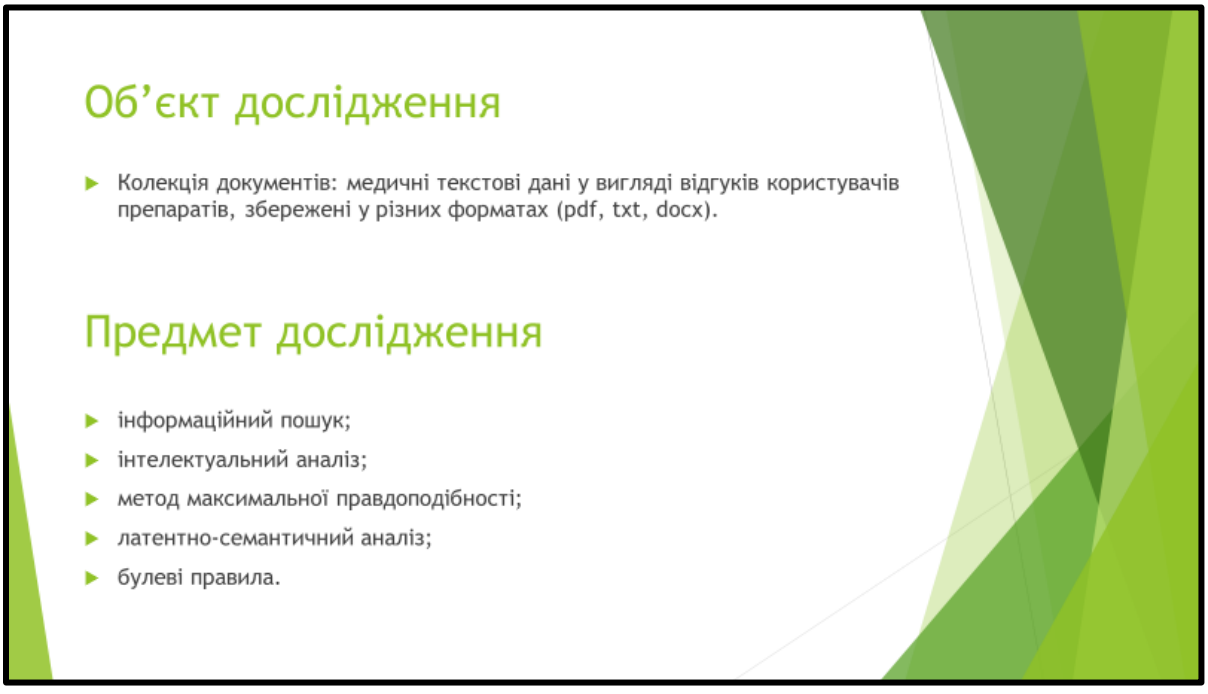
СПИСОК ЛІТЕРАТУРИ

1. SAS Visual Text Analytics, URL:https://www.SAS.com/ru_ru/software/visual-text-analytics.html
2. Бідюк П.І., Романенко В.Д., Тимощук О.Л. Аналіз часових рядів (навчальний посібник) — Київ: Політехніка, 2010. — 317 с.
3. А.В. Антонов. Методы классификации и технология Галактика - Зум. Научно-техническая информация. Сер. 1. Вып. 6. 2004.
4. Документація SAS, URL: https://support.sas.com/en/documentation.html
5. Introduction to Information Retrieval, URL: https://nlp.stanford.edu/IR-book/
6. International Journal of Computer Applications (0975 – 8887) Volume 181 – No.1, липень 2018, Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents
7. Функції відстані, URL: http://stefansavev.com/blog/better-euclidean-distance-with-the-svd-penalized-mahalanobis-distance/
8. Singular Value Decomposition, MIT, URL:http://web.mit.edu/be.400/www/SVD/Singular_Value_Decomposition.htm
9. Dan Kalman, A Singularly Valuable Decomposition: The SVD of a Matrix, The American University Washington, DC 20016 February 13, 2002
10. В.В. Стрижов. “Информационное моделирование”. Конспект лекций.
11. Jolliffe, L.T., Principal component analysis – 2nd ed., 2002
12. Rasmus Elsborg Madsen, Lars Kai Hansen and Ole Winther, Singular Value Decomposition and Principal Component Analysis, лютий 2004
13. Landauer, T. K., Foltz, P. W., & Laham, D. Introduction to Latent Semantic Analysis, 1998

14. Cambridge University Pres, April 1, 2009, URL:https://nlp.stanford.edu/IR-book/pdf/18lsi.pdf
15. Mastering Regular Expressions. Third Edition. Jeffrey E. F. Friedl
16. Comparison of Clustering Techniques for Cluster Analysis, Piyatida Rujasiri, and Boonorm Chomtee
17. Advances in Information Retrieval: 25th European Conference on IR Research, ECIR 2003, Pisa, Italy, April 14–16, 2003. Proceedings (pp.305-319)
18. Kullback-Leibler Divergence, Anna-Lena Popkes, лютий 2, 2019
19. Tom White, Hadoop: The Definitive Guide, p.736
20. Терентьев А.Н., Домрачев В.М., Костецкий Р.И, SAS Base: Основы программирования, Эдельвейс, 2014, 304 с.
21. Daniel Jurafsky, James H. Martin Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Second Edition. Pearson Education International. — 2009. — 1024 pp

ДОДАТОК А Ілюстративний матеріал доповіді





Актуальність

На сьогодні більша частина інформації (90%) знаходиться у неструктурованому вигляді, тому використання її звичними аналітичними моделями являється неможливим.

В медичній сфері, обробляючи текстові дані, можливо значно покращити якість зворотнього зв'язку з пацієнтами, а тому і якість препаратів.

Мета роботи

Дослідження існуючих методів обробки неструктурованих текстових даних та їх впровадження у системі аналізу та категоризації текстової медичної звітності.

Перелік основних методів

- Класифікація тексту на основі генерації булевих правил;
- Кластеризація тексту з застосуванням методу максимальної правдоподібності;
- Латентно-семантичний аналіз

Застосовані математичні методи

- Метод максимальної правдоподібності
- Сингулярний розклад матриці
- Метод головних компонент

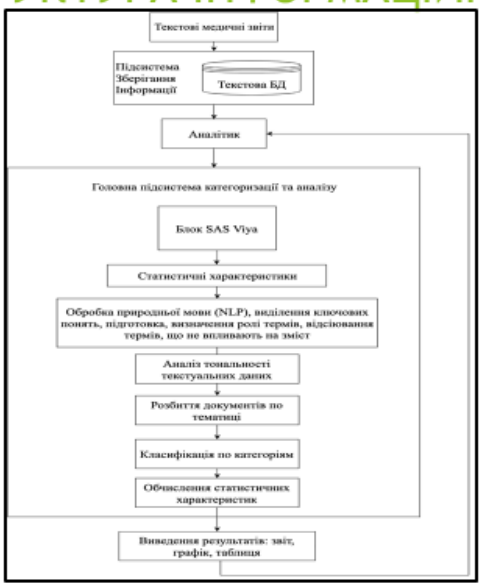
Статистичні міри, що використовувались для оцінки якості моделей

- RMSSTD
- F1-score
- Загальна точність
- Відстань Кульбека-Лейблера

F1 – score та загальна точність

- $\text{precision} = \frac{TP}{TP+FP}$
- $\text{recall} = \frac{TP}{TP+FP}$
- $F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$
- $\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$

СТРУКТУРА ІНФОРМАЦІЙНОЇ СИСТЕМИ

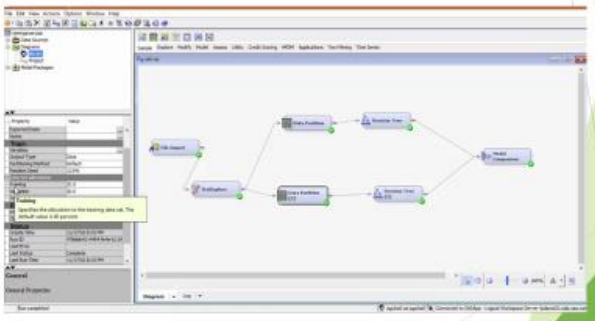


Програмне забезпечення, яке було використано в дипломній роботі

SAS Viya
(хмарні обчислення)

- Develop SAS Code
- Manage Data
- Prepare Data
- Explore and Visualize Data
- Build Models
- Manage Models
- Manage Decisions
- Explore Lineage
- Build Graphs
- Manage Workflows

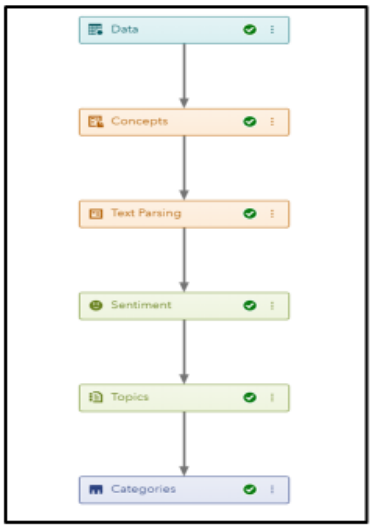
SAS Enterprise Miner
(локальний інструмент)

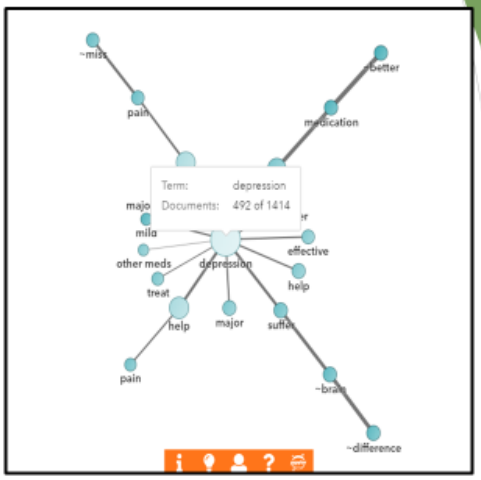


SAS технології

- Нормалізація, стемінг
- Виокремлення понять
- Створення тематик з поєднанням важливих понять
- Розподіл документів по створеним категоріям
- Аналіз тональності текстів

Результати



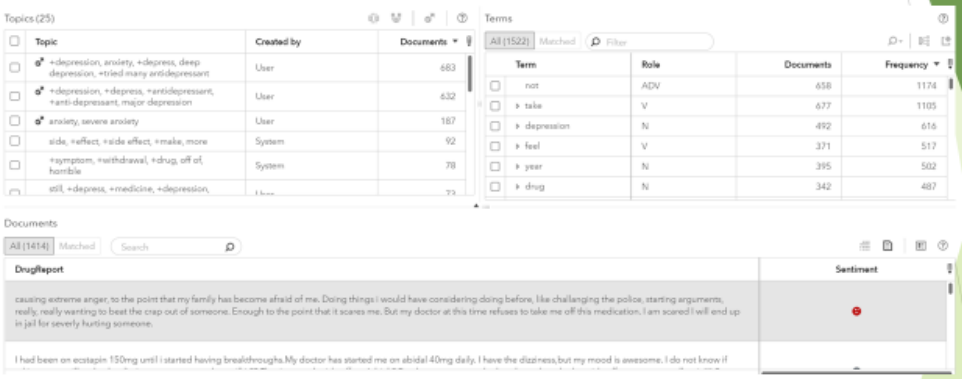


Взаємозв'язок понять

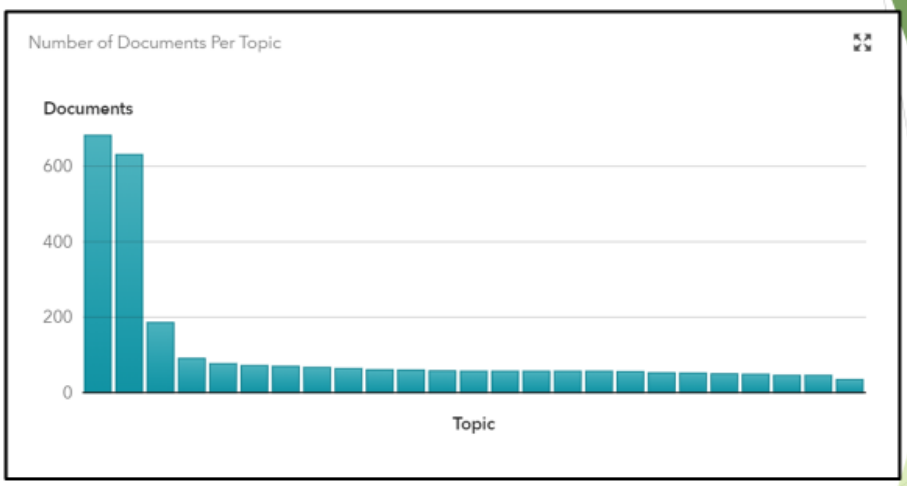
Результати

- Проведено аналіз сучасних методів інформаційного пошуку;
- Досліджені можливості використання наявних інструментів текстової аналітики;
- Налаштування їх під конкретну галузь;
- Розроблена система, яка дозволяє виконувати задачу класифікації, аналізу тональності, розподілу текстової інформації по категоріям отримуючи на вхід велику кількість медичних звітів.

Тематики



Об’єднання понять та підвищення їх до категорій



Порівняльна таблиця характеристик моделей класифікації з використанням WOE та без використання WOE

	Навчальна вибірка (Train)		Тестова вибірка (Test)	
	Загальна точність	F1	Загальна точність	F1
Модель без використання WOE	0.774	0.673	0.728	0.631
Модель з використанням WOE	0.803	0.701	0.785	0.682

ДОДАТОК Б

1. SAS Base код для розмічення датасету у CAS пам’яті:

```
/* ------------------- */
/* assign a library */
/* ------------------- */

libname locallib '/home/maximyrchk/project';

/* -------------------- */
/* assign a CAS library */
/* -------------------- */

libname mycaslib cas caslib=casuser;

/* --------------------------------- */
/* load the data sets from V9 to CAS */
/* --------------------------------- */

proc casutil;
  load data=locallib.asrs outcaslib="casuser" casout="asrs" promote;
quit;

proc casutil;
  load data=locallib.asrs_id outcaslib="casuser" casout="asrs_id" promote;
quit;

proc casutil;
  load data=locallib.asrs_newreports outcaslib="casuser" casout="asrs_newreports" promote;
quit;

proc casutil;
  load data=locallib.asrs_rdu_sna outcaslib="casuser" casout="asrs_rdu_sna" promote;
quit;
```

```
proc casutil;
  load data=locallib.CFPBcomplaints outcaslib="casuser" casout="CFPBcomplaints" promote;
quit;

proc casutil;
  load data=locallib.CFPBcomplaintsm outcaslib="casuser" casout="CFPBcomplaintsm" promote;
quit;

proc casutil;
  load data=locallib.drug_reports outcaslib="casuser" casout="drug_reports" promote;
quit;

proc casutil;
  load data=locallib.movies_plus outcaslib="casuser" casout="movies_plus" promote;
quit;

proc casutil;
  load data=locallib.sasgf_2013_papers_cl outcaslib="casuser" casout="sasgf_2013_papers_cl" promote;
quit;

/*----  The rules demo data is not referenced in the course notes. It is a small  ----*\
 *----  sampling of several text data sets. You can use it for experimentation,   ----*
 *----  since the small size will cause rapid execution. By default, it is not    ----*
 *----  loaded into memory.                                                       ----*
proc casutil;
  load data=locallib.vta_rules_demo outcaslib="casuser" casout="vta_rules_demo" promote;
quit;
\*----  ----*/

/* -------------------- */
/* analyze the raw data */
/* -------------------- */
```

```
proc cardinality data=mycaslib.asrs maxlevels=200
     outcard=mycaslib.asrs_summary out=mycaslib.asrs_summary_levels;
run;

proc print data=mycaslib.asrs_summary label;
  var _varname_ _type_ _rlevel_ _cardinality_ _nmiss_ _min_ _max_ _mean_;
run;

proc sgplot data=mycaslib.asrs;
  vbar Target02 / stat=percent;
run;

proc sgplot data=mycaslib.asrs;
  vbar Target05 / stat=percent;
run;

proc sgplot data=mycaslib.asrs;
  histogram Size / nbins=100 SHOWBINS;
  density Size;
run;


proc cardinality data=mycaslib.cfpbcomplaintsm maxlevels=200
     outcard=mycaslib.complaints_summary out=mycaslib.complaints_summary_levels;
run;

proc print data=mycaslib.complaints_summary label;
  var _varname_ _type_ _rlevel_ _cardinality_ _nmiss_ _min_ _max_ _mean_ _stddev_ _skewness_
_kurtosis_;
run;

proc freq data=mycaslib.cfpbcomplaintsm;
  tables Issue / list missing;
run;
```

```
proc sgplot data=mycaslib.cfpbcomplaintsm;
  vbar Issue / stat=percent;
run;

proc cardinality data=mycaslib.drug_reports maxlevels=200
    outcard=mycaslib.drug_reports_summary out=mycaslib.drug_reports_summary_levels;
run;

proc print data=mycaslib.drug_reports_summary label;
  var _varname_ _type_ _rlevel_ _cardinality_ _nmiss_;
run;

proc sgplot data=mycaslib.drug_reports;
  vbar Extension / stat=percent;
run;


proc cardinality data=mycaslib.movies_plus maxlevels=200
    outcard=mycaslib.movies_plus_summary out=mycaslib.movies_plus_summary_levels;
run;

proc print data=mycaslib.movies_plus_summary label;
  var _varname_ _type_ _rlevel_ _cardinality_ _nmiss_ _min_ _max_ _mean_ _stddev_ _skewness_
_kurtosis_;
run;

proc sgplot data=mycaslib.movies_plus;
  vbar Made_Money / stat=percent;
run;

proc sgplot data=mycaslib.movies_plus;
  histogram revenue / nbins=100 SHOWBINS;
  density revenue;
```

```
run;

proc sgplot data=mycaslib.sasgf_2013_papers_cl;
  vbar Business_Analytics / stat=percent;
run;

proc sgplot data=mycaslib.sasgf_2013_papers_cl;
  vbar section / stat=percent;
run;
```

Визначення поняття Medication:

```
CLASSIFIER:Abidal
CLASSIFIER:Abradon
CLASSIFIER:Acquil
CLASSIFIER:Ambutrin
CLASSIFIER:Amelorex
CLASSIFIER:Amicoran
CLASSIFIER:Amlican
CLASSIFIER:Aquiven
CLASSIFIER:Attentor
CLASSIFIER:Bifental
CLASSIFIER:Captalan
CLASSIFIER:Celifen
CLASSIFIER:Cenerol
CLASSIFIER:Concordan
CLASSIFIER:Detall
CLASSIFIER:Donital
CLASSIFIER:Ecstapin
CLASSIFIER:Elevex
CLASSIFIER:Escalan
CLASSIFIER:Espican
CLASSIFIER:Essequal
CLASSIFIER:Exulactin
CLASSIFIER:Formilan
```

CLASSIFIER:Fortifex
CLASSIFIER:Gemulex
CLASSIFIER:Habillan
CLASSIFIER:Halinol
CLASSIFIER:Hydrazan
CLASSIFIER:Ibuprofen
CLASSIFIER:Imitap
CLASSIFIER:Indrazine
CLASSIFIER:Intanol
CLASSIFIER:Isolex
CLASSIFIER:Maril
CLASSIFIER:Meriflex
CLASSIFIER:Nextam
CLASSIFIER:Noderall
CLASSIFIER:Noricam
CLASSIFIER:Norulen
CLASSIFIER:Parzonal
CLASSIFIER:Perinol
CLASSIFIER:Placilam
CLASSIFIER:Prexifan
CLASSIFIER:Promican
CLASSIFIER:Pronizen
CLASSIFIER:Quiescal
CLASSIFIER:Raniculex
CLASSIFIER:Reculan
CLASSIFIER:Reqlyx
CLASSIFIER:Reqlyz
CLASSIFIER:Respirex
CLASSIFIER:Respitan
CLASSIFIER:Revinor
CLASSIFIER:Septamyl
CLASSIFIER:Suprizol
CLASSIFIER:Sustify
CLASSIFIER:Tacifen

CLASSIFIER:Tanilor

CLASSIFIER:Taplex

CLASSIFIER:Tenactol

CLASSIFIER:Vanquixal

CLASSIFIER:Zindol

Визначення поняття Dosage:

REGEX:[\d]+\s?mg\.?

REGEX:[0-9]+[\.][0-9]+\s?mg\.?

Визначення поняття Prescription:

CLASSIFIER: MEDICATION DOSAGE

Визначення поняття Side_effects:

CLASSIFIER:Abdominal pain

CLASSIFIER:Aggression

CLASSIFIER:Agitation

CLASSIFIER:Allergic reaction

CLASSIFIER:Amnesia

CLASSIFIER:Anemia

CLASSIFIER:Back pain

CLASSIFIER:Blindness

CLASSIFIER:Blurred vision

CLASSIFIER:Bone pain

CLASSIFIER:Breast pain

CLASSIFIER:breathing

CLASSIFIER:Chest pain

CLASSIFIER:Chills

CLASSIFIER:Cold symptoms

CLASSIFIER:Colitis

CLASSIFIER:Colored vision

CLASSIFIER:Confusion

CLASSIFIER:Constipation

CLASSIFIER:Cough

CLASSIFIER:Diabetes mellitus

CLASSIFIER:Diarrhea

CLASSIFIER:Dizziness

CLASSIFIER:Dry mouth
CLASSIFIER:Dyspepsia
CLASSIFIER:Exhaustion
CLASSIFIER:Fainting
CLASSIFIER:Fast heartbeat
CLASSIFIER:Fever
CLASSIFIER:Fluid in the lungs
CLASSIFIER:Frequent urination
CLASSIFIER:Hair loss
CLASSIFIER:Hallucinations
CLASSIFIER:Headache
CLASSIFIER:Heart attack
CLASSIFIER:Heart palpitations
CLASSIFIER:Heartburn
CLASSIFIER:Hepatitus
CLASSIFIER:High blood pressure
CLASSIFIER:Hostile
CLASSIFIER:Hostility
CLASSIFIER:Hyperactive
CLASSIFIER:Hyperglycemia
CLASSIFIER:Impulsive
CLASSIFIER:Indigestion
CLASSIFIER:Insomnia
CLASSIFIER:Intestinal bleeding
CLASSIFIER:Irregular heartbeat
CLASSIFIER:Irritability
CLASSIFIER:Irritable
CLASSIFIER:Itching
CLASSIFIER:Jaundice
CLASSIFIER:Joint pain
CLASSIFIER:Ketoacidosis
CLASSIFIER:Kidney failure
CLASSIFIER:Leg cramps
CLASSIFIER:Liver damage

CLASSIFIER:Loss of appetite
CLASSIFIER:Low blood cell counts
CLASSIFIER:Low blood pressure
CLASSIFIER:Lower respiratory infection
CLASSIFIER:Moodiness
CLASSIFIER:Muscle pain
CLASSIFIER:Nausea
CLASSIFIER:Nightmares
CLASSIFIER:Overly excited
CLASSIFIER:Palpitations
CLASSIFIER:Pancreatitis
CLASSIFIER:Panicky
CLASSIFIER:Personality disorder
CLASSIFIER:Postural hypotension
CLASSIFIER:Pulmonary thrombosis
CLASSIFIER:Rectal bleeding
CLASSIFIER:Seizures
CLASSIFIER:Severe skin reactions
CLASSIFIER:Severely restless
CLASSIFIER:shortness of breath
CLASSIFIER:Shortness of breath
CLASSIFIER:Skin rash
CLASSIFIER:Sleeplessness
CLASSIFIER:Slow heartbeat
CLASSIFIER:Sore throat
CLASSIFIER:Spasm
CLASSIFIER:Speech disorder
CLASSIFIER:Stomach bleeding
CLASSIFIER:Stomach pain
CLASSIFIER:Stroke
CLASSIFIER:Sweating
CLASSIFIER:Sweats
CLASSIFIER:Swelling
CLASSIFIER:Thrombosis

CLASSIFIER:tinnitus

CLASSIFIER:Tiredness

CLASSIFIER:Upper respiratory infection

CLASSIFIER:Upset Stomach

CLASSIFIER:urinary retention

CLASSIFIER:Urinary tract infection

CLASSIFIER:vertigo

CLASSIFIER:Vomiting

CLASSIFIER:Weakness

CLASSIFIER:Weight gain

CLASSIFIER:Weight loss

Правило для відношення до категорії Depression&Anxiety:

```
(OR,
   (AND,(OR,"depression", "depressions"))
   (AND,"anxiety"),
   (AND,(OR,"depressing", "depressed", "depress"),
   (AND,(OR,"antidepressants", "antidepressant"),
   (AND,(OR,"anti-depressant", "anti-depressants"),
   (AND,"depressed"),
   (AND,"depressive"),
   (AND,(OR,"depression", "depressions"),
   (AND,"major"),
   )
```