Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)
Факультет інформаційних технологій
(факультет)
Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня ___магістра___
(бакалавра, спеціаліста, магістра)

студента _____Куленка Сергія Олександровича_____
(ПІБ)
академічної групи _____123м-20-1_____
(шифр)
спеціальності _____123 «Комп'ютерна інженерія»_____
(код і назва спеціальності)
за освітньо-професійною програмою _____«Комп'ютерна інженерія»_____
(офіційна назва)
на тему «Комп'ютерна система формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data»_____
(назва за наказом ректора)

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	доц. Бешта Д.О.			
розділів:				
теоретичний розділ	доц. Бешта Д.О.			
синтез системи	доц. Ткаченко С.М.			
розроблення програмного забезпечення	ас. Бешта Л.В.			
Рецензент				
Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро
2022

ЗАТВЕРДЖЕНО:

завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії
(повна назва)

_____ __Гнатушенко В.В._
(підпис) (прізвище, ініціали)
«___»_____ 2022 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня _____магістр _____
(бакалавра, спеціаліста, магістра)

студенту _____Куленку С.О._____ **академічної групи** ____123м-20-1____
(прізвище та ініціали) (шифр)

спеціальності _____123 «Комп'ютерна інженерія»_____

за освітньою-професійною програмою _123 «Комп'ютерна інженерія»__
(офіційна назва)

на тему «Комп'ютерна система формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data»,

затверджену наказом ректора НТУ «Дніпровська політехніка» від 10.12.2021 р. №1036

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел сформулювати наукове завдання, конкретизувати предмет та мету досліджень	20.09.2021
Теоретичний	Обґрунтувати теоретичну базу розв'язання наукового завдання, якому присвячено роботу	25.10.2021
Синтез системи	Розробка комп'ютерної системи	15.11.2021
Розроблення програмного забезпечення	Розробка програмного забезпечення	01.12.2021
Експериментальний розділ	Проведення і обробка результатів експериментів	15.12.2021
Графічна частина	Графічні результати роботи подати у вигляді рисунків схем таблиць на 10 арк. формату А4.	10.01.2022

Завдання видано _____ доц. Беша Д.О.
(підпис керівника) (прізвище, ініціали)

Дата видачі 06 вересня 2021 р.

Дата подання до екзаменаційної комісії 10.01.2022 р.

Прийнято до виконання _____ Куленко С.О.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Кваліфікційна робота: 89 с., 19 рис., 4 табл., 22 джерела, 2 додатка.

Об’єкт дослідження: Комп’ютерна система формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data.

Мета роботи: Розробити та впровадити комп’ютерну систему формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data, для підвищення ефективності рекламних кампаній через створення теплих аудиторій за допомогою вилучення, аналізу та встановлення відповідностей у великих даних.

Одержані результати: було розроблено програмне забезпечення для збору та аналізу даних про пошуки та пропозиції турів. На основі отриманих зв'язків побудований граф, на якому виконаний алгоритм по знаходженню найбільш оптимальної безлічі співпавших пар. За допомогою даної методики було продано 39 турів на різні напрямки за 1,5 тижня за допомогою таргетованої реклами під час пандемії.

Ключові слова: BIG DATA, СОЦІАЛЬНІ ГРАФИ, ДВУДОЛЬНІ ГРАФИ, ТАРГЕТИНГ, НЕЙРОТЕХНОЛОГІЇ, APACHESPARK, DATAMINING, СЛАБО СТРУКТОРОВАНІ ДАНІ, СИЛЬНО СТРУКТУРОВАНІ ДАНІ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

ЗМІ – засоби масової інформації;

RDD – Resilient Distributed Dataset;

CRM – Customer Relationship Management;

СУБД – система управління базами даних;

БД – база даних;

ID – identifier.

ВСТУП

В останні півтора десятиріччя одним із головних факторів, що прискорює формування і розвиток інформаційного суспільства, є Інтернет. Він став не лише глобальним засобом комунікацій без територіальних і національних кордонів, але й ефективним інструментом ведення бізнесу, досліджень, впливу на аудиторію. Зі вступом світової економіки в економічну кризу роль Інтернету лише зросла, оскільки завдяки застосуванню мережевих інформаційних технологій багато бізнесів спроможні не лише знизити витрати на просування і збут своїх послуг, але й розширити існуючі і освоїти нові ринки, підвищити ефективність і адресність взаємодії зі споживачами та іншими економічними контрагентами.

Практика свідчить, що інтернет-технології урівнюють шанси на успіх малих і великих бізнесів, тих, хто міцно закріпився на ринку, і новачків. Це стало можливим тому, що витрати на просування бізнесу є невеликими, використовуються доступні й фактично стандартизовані інструменти, можливо забезпечити недосяжну за інших умов широту охоплення і при цьому адресність впливу на цільову аудиторію, забезпечується фактично миттєвий доступ на ринок будь-якої країни чи регіону, можна у реальному масштабі часу оцінювати та аналізувати ефективність бізнесу.

Особливо значного поширення набуває застосування інформаційних інтернет технологій у маркетингу як методології і практично-орієнтованого інструментарію ведення бізнесу.

На сьогоднішній час можливості інтегрованих інтернет-технологій та інструментів у маркетингу майже не досліджені, особливо їх нерозривний зв’язок з пізнавальними, комунікаційними, та соціальними можливостями в інтернеті. У глобальній комп’ютеризації, настання якої планується на найближче майбутнє, роль маркетингу в інтернеті помітно зросте.

Інтернет-маркетинг потрібно розглядати як новий вид маркетингу, який передбачає застосування традиційних та інноваційних інструментів і технологій у мережі інтернет для визначення і задоволення потреб і запитів споживачів (покупців) шляхом обміну з метою отримання продавцем прибутку.

Аудиторія у інтернеті становить значну частину населення світу, причому найбільш активну й освічену, яка, однак, нерівномірно розподілена між країнами. Статистика свідчить, що розміри інтернет-аудиторії стрімко зростають. Таким чином, потенційні можливості застосування сучасних технологій та інструментів інтернет-маркетингу є досить значними.

Маркетингові дослідження завжди пов'язані з великою кількістю інформації. Будь-яка компанія сьогодні має вкрай різнорідні за своєю структурою великі дані (Big Data). Їх використання у всьому світі, і особливо в Україні, тільки робить перші кроки. Однак за туманним визначенням «великі дані» стоїть те, що живить маркетингові дослідження та забезпечує прийняття компаніями вдалих управлінських рішень і, як наслідок, отримання найкращих позицій на ринку. Однак через розмаїтість і різнорідність потоків даних, побудувати правильні стратегії та проводити грамотні дослідження, часом буває надзвичайно складно.

Вважається, що маркетинг знаходиться на великій відстані від математичних алгоритмів та інших обчислень, однак це помилка. Переваги використання великих даних у маркетингу:

– Створення точного портрета цільового споживача;

– Передбачення реакції споживачів на маркетингові «повідомлення» та пропозиції продукту;

– Персоналізація рекламних повідомлень;

– Оптимізація виробництва та стратегій розподілу;

– Створення цифрового маркетингу та рекламно-просвітницьких компаній;

– Збереження більшої кількості клієнтів шляхом найменших витрат;

– Отримання кращого ставлення до власного продукту.

Враховуючи викладене, необхідно дослідити сучасні тенденції застосування інтернет-технологій та інструментів у бізнесі, розробити та впровадити систему формування рекламних пропозицій у суспільних мережах.

1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Стисла характеристика галузі та умов застосування системи

У зв’язку зі збільшенням популярності соціальних мереж, бізнес не міг не скористуватися можливістю використовувати їх для комерційних цілей. А розробники цих соціальних мереж швидко зрозуміли, наскільки вигідною платформою для комерції можуть стати соціальні мережі і який дохід може отримувати сама мережа. Наприклад, рекламна виручка Facebook у другому кварталі 2020 року склала $ 18,3 млрд, що на 10% більше в порівнянні з квітнем-червнем 2019 року.

Спочатку маркетинг у соціальних мережах був реалізований досить примітивними методами: лідери думок або міські групи та групи по інтересам за певні кошти та/або відсоток від продажу рекламували різні продукти. Однак доволі швидко ця ідея почала втрачати поширеність, адже довіра до такої реклами поступово втрачалася. Оскільки у соціальних мережах міститься величезний обсяг інформації про користувачів, його почали використовувати у «комерційних» цілях. Зберігання таких даних привело власників соціальних мереж до корисних можливостей. Але для роботи з таким обсягом даних потрібні певні технології та методи, які мають спільну назву Big Data.

1.2 Поняття великих даних

Big Data – це сукупність безупинно збільшуваних обсягів інформації одного контексту, але різних форматів представлення, а також методів і засобів для ефективної і швидкої обробки. Головною характеристикою Big Data є ступінь їх структурованості та варіанти представлення [13].

Яскрава ілюстрація великих даних – це безперервно надходжувана інформація з датчиків або пристроїв аудіо- і відеореєстрації, потоки

повідомлень з соцмереж, метеорологічні дані, координати геолокації абонентів стільникового зв'язку і т.д.

Таким чином, джерелами великих даних можуть бути:

– Інтернет – соціальні мережі, блоги, ЗМІ, форуми, сайти, інтернет речей (Internet of Things, IoT);

– корпоративна інформація – транзакції, архіви, бази даних і файлові сховища;

– показання приладів–датчиків, сенсорів, реєстраторів та ін.

Щоб отримати робочу гіпотезу про причини виникнення конкретних ситуацій, зокрема, як пов'язані відмови устаткування з умовами подачі напруги, або спрогнозувати майбутнє, наприклад, ймовірність своєчасного повернення кредиту приватним позичальником, аналіз великих обсягів структурованої і неструктурованої інформації виконується в кілька етапів:

– чистка даних (data cleaning) – пошук і виправлення помилок в первинному наборі інформації, наприклад, помилки ручного введення (помилки), некоректні значення з вимірювальних приладів через короткочасних збоїв і т.д. ;

– генерація предикторів (feature engineering) – змінних для побудови аналітичних моделей, наприклад, освіту, стаж роботи, стать і вік потенційного позичальника;

– побудова і навчання аналітичної моделі (model selection) для передбачення цільової (таргетной) змінної. Так перевіряються гіпотези про залежність таргетної змінної. Наприклад, скільки днів становить прострочення по кредиту для позичальника з середньою освітою і стажем роботи менше 3-х місяців.

До основних методів збору і аналізу великих даних відносять [13]:

– Data Mining – навчання асоціативним правилами, класифікація, кластерний і регресійний аналіз;

– краудсорсінг – категоризація та збагачення даних народними силами, тобто з добровільною допомогою сторонніх осіб;

– змішання і інтеграція різнорідних даних, таких як, цифрова обробка сигналів і обробка природної мови;

– машинне навчання (Machine Learning), включаючи штучні нейронні мережі, мережевий аналіз, методи оптимізації та генетичні алгоритми;

– розпізнавання образів;

– прогнозна аналітика;

– імітаційне моделювання;

– просторовий і статистичний аналіз;

– візуалізація аналітичних даних–малюнки, графіки, діаграми, таблиці.

Різнорідність великих даних обумовлює специфічні технології роботи з ними. Програмно-апаратні засоби роботи з Big Data передбачають масштабованість, паралельні обчислення і розподіленість, тому що безперервне збільшення обсягу – це одна з головних характеристик великих даних. До основних технологій відносять нереляційні бази даних (NoSQL), модель обробки інформації MapReduce, компоненти кластерної екосистеми Hadoop, мови програмування R і Python, а також спеціалізовані продукти Apache (Spark, AirFlow, Kafka, HBase і ін.).

Хоча термін «великі дані» є відносно новим, процес збору і зберігання великих обсягів інформації для подальшого аналізу має давню історію. Концепцію визначення великих даних можна сформулювати як сукупність наступних факторів:

– Об`єм. Організації збирають дані з різних джерел, включаючи бізнес-транзакції, соціальні мережі і інформацію від датчиків або машинних

даних. У минулому зберігати його було б проблемою, але нові технології (такі як HadLoop) полегшили тягар.

– Швидкість. Потоки даних відбуваються з безпрецедентною швидкістю і повинні оброблятися своєчасно. RFID-мітки, датчики і інтелектуальний облік керують необхідністю мати справу з потоками даних в близькому до реального часу.

– Різновид. Дані надходять у всіх типах форматів–від структурованих, числових даних в традиційних базах даних до неструктурованих текстових документів, електронної пошти, відео, аудіо, біржових даних і фінансових транзакцій.

– Мінливість. На додаток до зростаючих швидкостях і різновидам даних потоки даних можуть бути сильно несумісні з періодичними піками. Щоденні, сезонні і викликані подіями пікові навантаження даних складні в обробці. Тим більше при роботі з неструктурованими даними.

– Складність. Сьогоднішні дані надходять з декількох джерел, що ускладнює зв'язок, зіставлення, очищення та перетворення даних між системами. Однак необхідно з'єднувати і корелювати зв'язку, ієрархії і множинні зв'язку даних, інакше дані можуть швидко вийти з-під контролю.

Самі по собі алгоритми Big Data виникли при впровадженні перших високопродуктивних серверів (мейнфреймів), що володіють достатніми ресурсами для оперативної обробки інформації та придатних для комп'ютерних обчислень і для подальшого аналізу.

Big Data – серія підходів, інструментів і методів обробки структурованих і неструктурованих даних величезних обсягів і значного різноманіття. Дані технології застосовуються для отримання сприймаються людиною результатів, ефективних в умовах безперервного приросту, розподілу інформації по численних вузлів обчислювальної мережі.

Виходячи з визначення Big Data, можна сформулювати основні принципи роботи з такими даними:

– Горизонтальна масштабованість. Оскільки даних може бути як завгодно багато – будь-яка система, яка має на увазі обробку великих даних, повинна бути розширюваної.

– Стійкість до відмов. Принцип горизонтальної масштабованості має на увазі, що машин в кластері може бути багато. Отже, частина машин будуть виходити з нормального режиму роботи, тому методика роботи з великими даними повинні враховувати можливість таких збоїв.

– Локальність даних. Це один з найважливіших принципів проектування Big Data – рішень є принцип локальності даних – по можливості обробляємо дані на тій же машині, на якій їх зберігаємо.

Традиційним методом роботи з масивами інформації є реляційні бази даних. Однак робота з реляційною базою даних на сотні терабайт – це ще не Big Data. Нижче представлені характеристики і з відмінності в традиційних БД і BigData (табл.1.1).

Таблиця 1.1 – Характеристики традиційних БД і великих БД

Характеристика	Традиційні БД	Великі БД
Об’єм інформації	Від (1Gb 10^9 байт) до 1Tb(10^{12} байт)	Від 1Pb 10^{15} байт до 1Eb(10^{18} байт)
Спосіб зберігання	Централізований	Децентралізований
Структурованість даних	Структурована	Напівструктурована або неструктурована
Модель зберігання та обробки даних	Вертикальна модель	Горизонтальна модель
Взаємозв’язок даних	Сильний	Слабкий

У реляційних БД інформація розподілена дисперсно, тобто має місце спочатку задана чітка структура, зміна якої в уже працюючої базі пов'язано

з безліччю проблем. Таким чином, в силу своєї архітектури, реляційні БД найкраще підходять для коротких швидких запитів, що йдуть однотипним потоком. Складний же запит або зажадає перебудови структури БД, або, на догоду швидкодії, збільшення обчислювальних потужностей. Це вказує на ще одну проблему традиційних баз даних, а саме на складність їх масштабованості.

Таким чином, для роботи зі складними гнучкими запитами необхідне середовище, що дозволяє зберігати і обробляти неструктуровані дані, піддається масштабування і допускає застосування розподілених обчислень, де для обробки даних використовується не одна високопродуктивна машина, а ціла група таких машин, об'єднаних в кластер.

Термін «NoSQL» виник в червні 2009 року і був розшифрований як «Not Only SQL» – «не тільки SQL». Таким терміном позначають нереляційні БД, в яких немає внутрішніх зв'язків. БД NoSQL можуть використовувати різні моделі представлення даних в залежності від свого призначення.

Технологія NoSQL прибирає всі обмеження реляційної моделі (наприклад, трудомісткість горизонтального масштабування, недолік продуктивності в кластері), а також полегшує способи зберігання і доступу до даних. Такі БД використовують неструктурований підхід, організовуючи дані специфічних типів за малий проміжок часу і пропонуючи різні типи доступу до них.

NoSQL база даних надає механізм для зберігання та вилучення даних, який моделюється засобами, відмінними від табличних відносин, використовуваних в реляційних базах даних. Такі бази даних існували з кінця 1960-х років, але не отримали прізвисько "NoSQL" до сплеску популярності на початку двадцять першого століття, викликаного потребами компаній Web 2.0, таких як Facebook, Google і Amazon. бази

даних NoSQL все частіше використовуються в великих даних і веб-додатках реального часу. системи NoSQL також іноді називають "Notonly SQL", щоб підкреслити, що вони можуть підтримувати SQL-подібні мови запитів.

Замість цього більшість баз даних NoSQL пропонують концепцію «остаточної узгодженості», в якій зміни бази даних поширюються на всі вузли "в кінцевому рахунку" (як правило, протягом мілісекунд), тому запити даних можуть не повертати оновлені дані негайно або можуть привести до читання даних, які не є точними, проблема, відома як застарілі читання. Крім того, деякі системи NoSQL можуть демонструвати втрачені записи та інші форми втрати даних. Деякі системи NoSQL надають такі поняття, як ведення журналу з випередженням записи, щоб уникнути втрати даних. Для розподіленої обробки транзакцій в декількох базах даних, узгодженість даних є ще більш складним завданням, яка є складною як для NoSQL, так і для реляційних баз даних. Навіть поточні реляційні бази даних не дозволяють обмеженням посилальної цілісності охоплювати бази даних. Існує мало систем, які підтримують як транзакції ACID, так і стандарти x / Open XA для розподіленої обробки транзакцій.

Бази даних NoSQL – це нереляційні бази даних, які є масштабованими, оптимізованими для використання моделей даних без єдиної схеми. Бази даних NoSQL широко використовуються, тому що вони спрощують розробку, забезпечують стійкість до відмов і забезпечують низьку затримку. Такі бази даних можуть використовувати різні моделі даних, включаючи стовпчасті, документальні, графічні дані і зберігати пари "ключ-значення" в пам'яті.

Системи бази даних NoSQL використовуються для управління даними різних моделей, в тому числі для зберігання пар "ключ-значення" в пам'яті графових моделей даних і зберігання документів. Ці типи баз даних оптимізовані для додатків, яким потрібні великі обсяги даних, низька

затримка і гнучкі моделі даних. Все це досягається за рахунок пом'якшення жорстких вимог до узгодженості традиційних реляційних баз даних.

1.3 Актуальність використання великих даних у Digital Marketing

Інтерес до даних у соціальних мережах значно зріс, адже їх вилучення та глибинний аналіз та подальша робота відділу маркетингу може принести бізнесу мільйони доларів. Однак, отримати дані напряму жоден бізнес не може, адже це порушення права на конфіденційність даних, яке має кожна людина. Однак, закон ніяк не забороняє використовувати ті дані, які користувачі добровільно залишають, користуючись соціальними мережами. Важливо помітити, що люди часто групуються у певні спільноти у соціальних мережах. Це явище і було використано для того, щоб відшукати потрібну аудиторію.

Під поняттям великих даних мається на увазі робота з інформацією величезного обсягу і різноманітного складу, що достатньо часто оновлюється і знаходиться в різних джерелах з метою збільшення ефективності роботи, створення нових продуктів і підвищення конкурентоспроможності. Великі дані об'єднують техніки і технології, які витягують сенс з даних на екстремальному рівні практичності [1]. Вручну зібрати таку кількість даних, проаналізувати, відсортувати їх та ще зробити велику кількість пропозицій, базуючи свою думку на комбінаціях – було б неможливо. Але програмно це питання можна вирішити за декілька годин.

1.4 Специфіка рекламних пропозицій у Digital Marketing

Через велику кількість реклами, в багатьох з нас вже є «банерна агресія та сліпота», важливо зробити таке оголошення, яке не буде дратувати користувача. Для цього воно має бути ненав’язливим, зрозумілим, цікавим і, головне, цільовим. А щоб оголошення було

цільовим, важливо вірно підібрати свою аудиторію за допомогою глибинного аналізу великих даних.

Спираючись приклад продажу турів, що використовується у даному дослідженні, легко припустити, що тур у Херсоньску область та тур на Мальдіви варто пропонувати дуже різним людям, хоча на перший погляд – обидва сегменти мають спільний інтерес «подорожі». Але для того, щоб виявити інтерес «подорожі», комп'ютерна допомога не потрібна. Великі дані потрібні тоді, коли нам необхідно виявити більш точні характеристики, які найчастіше пов'язані безпосередньо з поведінкою користувача.

Збираючи дані, відповідні пропозиціям, можна легко влучити у ціль. Аналіз даних соціальних мереж є дуже актуальним за наступними причинами:

– Унікальне джерело даних про особисте життя і інтереси реальних людей;

– Листування, щоденники, фотоальбоми;

– Проникнення в усі сфери діяльності: до мереж IT-фахівців, трейдерів, бухгалтерів;

– Соціалізація контенту: фото, відео, музика, новини, рецепти;

– Соціалізація сервісів: магазини, форуми, рекомендації;

– Зростаючий попит на соціальні сервіси, що спрощують спілкування та обмін даними;

– Величезні можливості для таргетованого маркетингу.

Збираючи дані, які залишають люди, користуючись смартфонами, ми можемо сегментувати тих, кому пропонувати Мальдіви, окремо від тих, кому пропонувати подорожі рідною країною.

І, звісно, це стосується будь-якої ніші: пошуку професій, доставки їжі, товарного бізнесу, сфери послуг. Фільтруючи людей за цими даними, можна розробити персоналізоване оголошення для кожного сегменту:

навіть дуже вузького. Наприклад, для чоловіків, які живуть на Осокорках і мають дітей.

Тому великі дані та глибинний аналіз як ніколи актуальний для маркетингу, адже на базі отриманих результатів значно легше працювати, аніж робити припущення щодо власної аудиторії і її можливих потреб.

1.5 Аналіз методів роботи з Big Data

Соціальні мережі зберігають дані, які пов’язані не тільки з місцем проживання, маршрутом пересування, датою народження користувачів, але й з їх інтересами, сімейним станом, місцем роботи, віком дітей, улюбленим кіно та іншими доволі «персоналізованими» характеристиками.

Якщо грамотно проаналізувати зібрані дані про користувачів, можна визначити навіть такі подробиці як суми покупок у певних магазинах або кількість трафіку, який людина витрачає у соц. мережах. Або з якого пристрою вона виходить у мережу Інтернет. Це дуже корисна інформація, адже маючи її можна зрозуміти: кому запропонувати більш дорогий тур, кому – пакет інтернет-тарифу з безкоштовними соц. мережами, хто знову може повернутися на сайт за покупками або які люди найбільш схильні купляти певну продукцію на суму більш ніж N-значення або більш ніж N разів за певний проміжок часу.

Нейронні мережі навчились навіть генерувати рекламні оголошення із заданих даних (тексту та зображень), які з найбільшою ймовірністю підійдуть тій чи іншій аудиторії. Це відбувається завдяки нейротехнологіям та знанням системи основних характеристик заданої виборки людей.

Та навіть якщо «заздалегідь» самостійно визначити вибірку не вдалося, за допомогою навчання алгоритму, можна виявити «своїх» потенційних клієнтів через деякі їх дії. До того ж, тематичні спільноти у соціальних мережах збирають схожих за інтересами людей в одному місці: нам залишається тільки правильно продати їм свій продукт.

1.6 Типи та особливості соціальних даних

Соціальні дані є декількох типів – слабо-структуровані джерела у реальному часі, розмір – як мільйони об’єктів з мільярдами зв’язків, модель даних – мультограф з усіх об’єктів та зв’язків між ними – детальніше про збір (рис. 1.1) [14].

Типи соціальних даних:

– акаунти користувачів з профілями та зв’язками;

– текстовий контент;

– мультимедійний контент;

– журнали активності користувачів.

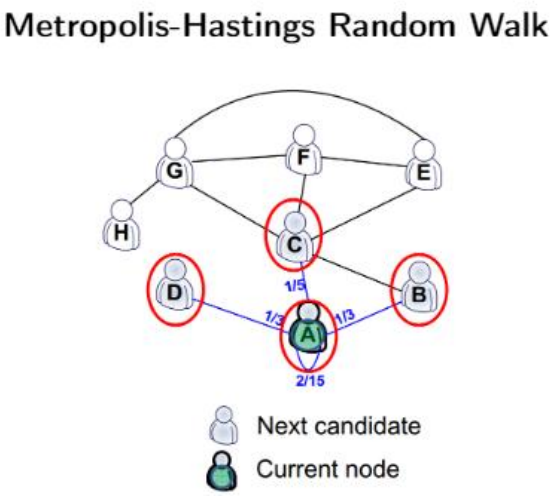


Рисунок 1.1 – Модель збору соціальних даних

Пошук глобальних суспільств користувачів (зображено на рис. 1.2). Алгоритм імітує людське спілкування між парами індивідуумів:

– пам'ять кожного вузла ініціалізується унікальною міткою співтовариства (групи);

– потім ітеративно повторюється послідовність кроків:

a. Обирається «слухає» вузол b. Кожна з вершин-сусідів обраного вузла випадковим чином обирає мітку з ймовірністю, пропорційною кількості міток даного типу в своїй пам'яті, і посилає вибрану позначку «слухає»

вузлу c. Вузол «хто слухає» вибирає найпопулярнішу з надісланих йому міток і додає її в свою пам’ять;

– для пошуку пересічних співтовариств для кожної вершини обираються найбільш популярні для них мітки;

– для пошуку непересічних спільнот для кожної вершини обирається її найбільш популярна мітка.

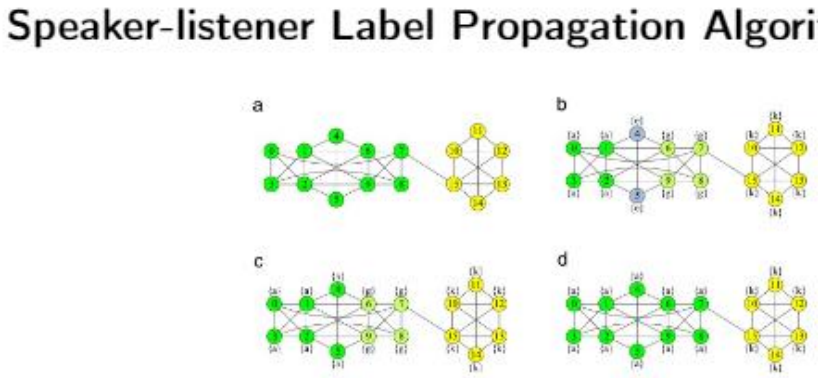


Рисунок 1.2 – Графічне зображення пошуку глобальних співтовариств користувачів

Для реалізації такого пошуку необхідно застосувати алгоритм SLPAна Spark.Bagel. Такий спосіб реалізації має наступні переваги:

– вершини графа розподіляються по вузлах кластера в RDD-структурах;

– ненаправлені ребра трансформуються у двонаправлені;

– на кожній ітерації проводиться обмін мітками між вершинами у вигляді повідомлень (BSP);

– вершини накопичують мітки в пам’яті;

– після завершення ітерацій, для отримання остаточного розбиття застосовується пост-процесінг пам'яті всіх вершин;

– добра масштабованість і висока якість результатів.

1.7 Нейротехнології та нейромаркетинг

Нейромаркетинг – прикладний розділ нейроекономіки, що представляє собою новий методологічний підхід маркетингу, що включає в себе дослідження споживчої поведінки із застосуванням інструментарію нейронаук. Нейромаркетинг вивчає споживчу поведінку (мислення, пізнання, пам'ять, емоційні реакції і т.д.), маючи важливе завдання – спрогнозувати споживчий вибір кожного користувача мережі інтернет [22].

Нейроекономічні дослідження показали принципову можливість об'єктивно вивчати процес ухвалення рішення про покупку товарів. Застосування методів нейробіології в маркетингу має ряд переваг, що підвищують ефективність маркетингових досліджень, так як дозволяють зареєструвати безпосередню (більш «об'єктивну»), а не раціональну реакцію на товар або рекламу – об'єктивно оцінити суб'єктивну реакцію споживача. Основними цілями нейромаркетинга є скорочення «вартості» маркетингових досліджень, і отримання більш достовірної маркетингової інформації.

За допомогою нейромаркетінгового апаратного тестування, можна досліджувати споживчі реакції на аудіовізуальний об'єкт, яким може стати логотип, етикетка, упаковка, обкладинка чи розворот журналу або газети, інтернет ресурс, плакат, біл-борд, рекламний ролик і товарна полку в магазині або вивіска.

Майбутні інструменти таргетингу стануть ще більш досконалими, а повідомлення, що доноситься до клієнтів і до потенційних клієнтів, більш персоналізованими. А це означає, що, отримавши максимум інформації про відвідувача, ми зможемо точніше зрозуміти мотиви ухвалення рішення про покупку і запропонувати той продукт, який він купить з більшою часткою ймовірності.

Вперше про персоналізований контент заговорили фахівці з email-маркетингу. Сегментація підписної бази і відправка кожній групі вузько

цільових повідомлень приносила збільшення показників open rate (сторінки, які відкривають користувачі) і click rate (переходи по посиланнях) як мінімум на 75-80%.

Споживачі хочуть отримувати більш релевантний контент, про що говорить ряд досліджень:

Згідно зі звітом Infosys, споживачі хочуть отримувати максимально персоналізоване пропозицію. 59% клієнтів говорять, що персоналізація впливає на їх рішення про покупку. 31% клієнтів хотіли б, щоб їх досвід покупок був більш персоналізованим, ніж насправді. 74% відвідувачів відчувають розчарування, коли контент сайту не персоналізований.

А дослідження, в ході якого було проаналізовано 650 мультиканальних маркетингових кампаній, показало, що персоналізовані кампанії в переважній більшості випадків перемагали статичні кампанії, генеруючи високий показник відгуку від цільової аудиторії (за даними з джерела MindFire).

Персоналізація активно розвивається в онлайн-торгівлі. Багатьом інтернет магазинам вдавалося досягти мільйонних результатів, використовуючи персоналізований контент.

Кілька років тому весь світ облетіла новина про незручної ситуації, що сталася з американської торгової мережею Target, яка дізналася про вагітність дівчини навіть раніше, ніж її батько. Розгніваний батько увірвався в офіс компанії, намагаючись з'ясувати, чому її улюбленої доньки, яка ще ходить в школу, приходять купони на дитячий одяг і памперси. Представникам компанії не залишалося нічого іншого, як принести свої вибачення, проте пізніше з'ясувалося, що вони мали рацію. Система спрогнозувала вагітність, спираючись на дані попередніх покупок.

Великі дані активно застосовують в банківській сфері. В основному, для оцінки кредитних ризиків фізичних або юридичних осіб. Стандартні програми аналізу кредитоспроможності враховують лише щомісячні

звітності організації і дані про її кредитної історії. Big Data ж здатна підключити до аналізу величезний пласт зовнішньої інформації.

Еволюція сервісів товарних рекомендацій підштовхує e-commerce до нового формату боротьби за покупця. Вигравatиме той магазин, який вже на сторінці входу буде генерувати товарну пропозицію з максимальною персоналізацію на основі аналізу величезного масиву даних про відвідувача і історії його переміщення. Оперуючи великими даними, можна з високою точністю передбачити потребу клієнта і показати йому відповідну рекламу в потрібному місці, в потрібний час.

1.8 Перспективи Big Data для створення контенту в рекламі

Data-driven creative (креатив, заснований на даних) – одна з найбільш важливих тенденцій галузі. Це створення повідомлень, на основі інформації про аудиторію. Такою інформацією можуть бути демографічні ознаки аудиторії, дані CRM, історія web-пошуку.

Найбільш простим і широко використовуваним в рекламній галузі прикладом подібного створення креативів є динамічні оголошення в Facebook (рис. 1.6).

Динамічні оголошення – це тип рекламної кампанії, що дозволяє автоматично створювати велику кількість однотипних текстових оголошень, адаптованих відповідно до конкретного пошуковим запитом користувача. Динамічні оголошення генеруються на основі вмісту сайту або новинної стрічки або, якщо казати безпосередньо про соціальні мережі, – на основі підібраних зображень та частин тексту.

Подібний тип кампаній найчастіше використовується інтернет-магазинами, де створення окремого оголошення під кожну товарну позицію є дуже трудомістким процесом.

Динамічні оголошення автоматично використовують в тексті реклами назву конкретного товару, яке шукає користувач. Динамічні оголошення

дозволяють створювати персоналізовані повідомлення для кожного конкретного пошукового запиту. Завдяки цьому ефективність рекламних кампаній підвищується. Однак варто звернути увагу і на більш складні механіки. Наприклад, на створення комунікаційних повідомлень за допомогою аналізу Big Data штучним інтелектом. Креативний процес являє собою аналіз існуючих або раніше створених матеріалів та їх компіляцію. Сьогодні робота фахівців з реклами та PR зводиться, по суті, до пошуку правильного алгоритму або методології, щоб кінцева комунікація приносила більший ефект для бренду, і з цим штучний інтелект зможе впоратися набагато краще людини.

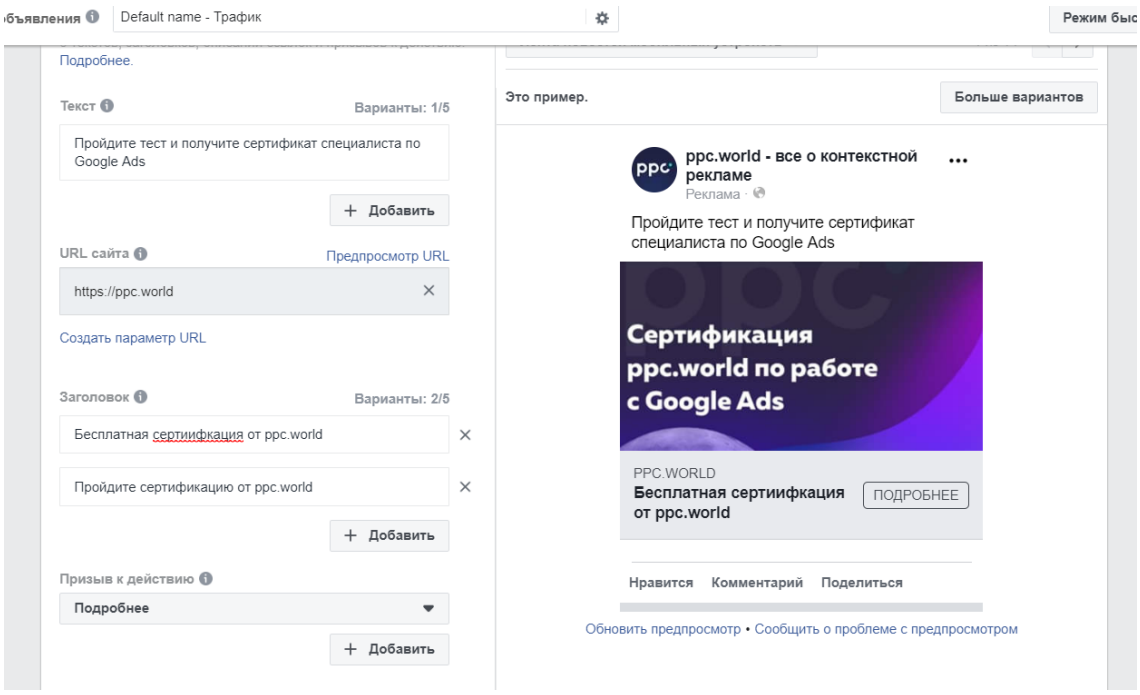


Рисунок 1.6 – Приклад створення динамічного креативу Facebook

1.9 Збір даних з соціальних мереж

Веб-інтерфейси соціальних мереж є джерелами даних реального часу і призначені для перегляду і взаємодії зі сторінками соціальної мережі в веб-браузері або для використання даних користувачів спеціалізованими

додатками. Оскільки сценарії використання інтерфейсів соціальних мереж не передбачають автоматичного збору даних безлічі користувачів з метою побудови соціального графа, то виникає ряд проблем [18]:

– Приватність даних – часто доступ до даних користувачів дозволений тільки для зареєстрованих і авторизованих учасників мережі, що вимагає підтримки емуляції користувальницької сесії за допомогою спеціальних облікових записів (акаунтів).

– Слабка структурованість даних – у багатьох випадках програмні інтерфейси (API) соціальних мереж мають обмежений функціонал, що вимагає підтримки отримання з допомогою призначеного для користувача веб-інтерфейсу статичних копій HTML-сторінок, коректної обробки їх динамічної частини(Включаючи виконання асинхронних запитів до сервера соціальної мережі), вилучення потрібних даних за допомогою алгоритму і / або шаблону і побудови їх структурованого уявлення,зручного для подальшої автоматичної обробки.

– Обмеження доступу і блокування – з метою запобігання несанкціонованого автоматичного збору даних і обмеження навантаження на інфраструктуру сервісу соціальної мережі власники сервісів часто вводять явні чи приховані обмеження на допустиму кількість запитів від одного користувача акаунта і / або IP-адреси в одиницю часу, що вимагає врахування кількості посилаються запитів, а також підтримки динамічної ротації використовуваних для збору даних для користувача акантів і IP-адрес.

– Розмірність даних обумовлює необхідність в паралельному методі збору даних, а також в методах отримання репрезентативної вибірки користувачів соціальної мережі (семплірування).

У зв'язку з постійною необхідністю отримання великих наборів даних з соціальних мереж, був розроблений фреймворк для збору даних з різних інтернет-сервісів. Розроблений інструмент підтримує скачування даних з

соціальних мереж Facebook, Twitter. Реалізовано кілька способів отримання репрезентативних вибірок користувачів соціальних мереж: семплірування методом обходу в ширину (breadth-first search, BFS) [1], по Метрополісу Гастінгсу (Metropolis-Hastings Random Walk, MHRW) і методом «лісової пожежі» (Forest Fire, FF) [18].

1.10 Постановка завдання дослідження

Соціальні мережі знають про людей значно більше, ніж здається. Поставлена задача – правильно скористатися зібраними великими даними, провести глибинний аналіз, структурувати та побудувати зв’язки типу «запит-пропозиція» для польського агентства з продажу турів.

Дослідити інструментарій за допомогою якого можна реалізувати задачу, а також проаналізувати варіанти розбору web-сторінок та експорту даних.

Зібрати та обробити велику кількість текстових даних різного типу. Частину користувачі залишили на сайті агентства, іншу – у соціальних мережах, у спільнотах для пошуку гарячих турів.

В рамках дипломної роботи розробити програмне забезпечення, яке збиратиме та аналізуватиме дані.

Розробити алгоритм, який дозволить аналізувати оброблені дані і знаходити зв'язки між ними.

На основі отриманих зв’язків побудувати граф, за яким буде працювати алгоритм по знаходженню найбільш оптимальної безлічі співпавших пар.

Завдання даної дипломної роботи: Розробити систему формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data.

1.11 Висновки по розділу

Найбільша кількість інформації завжди використовується при вивченні та розумінні власного продукту компанією. Сьогодні, у вік багатих онлайн-можливостей, проведення якісних та кількісних досліджень досить легко здійснюється в інтернеті. У тому числі й такі складні форми первинних маркетингових досліджень, як фокус-групи, об'ємні опитування у соцмережах та інше. Це лише частина великих даних для дослідження продукту. Вона дуже корисна для побудови дієвих ідей щодо товарів та послуг компанії. Також при глибокому аналізі цієї інформації компанія може створити потужну базу для просування власного продукту.

2 ТЕОРЕТИЧНИЙ РОЗДІЛ

2.1 Загальна характеристика соціальних мереж

Сайти соціальних мереж, таких як Twitter, Facebook, LinkedIn, YouTube і Wikipedia, об'єднують величезні популяції користувачів і зберігають ексабайт інформації, пов'язаної з їх повсякденною взаємодією.

Моделювання соціальних мереж виконується шляхом збору даних і аналізу вибірок, за допомогою побудови блокових і дифузійних моделей, а також методом аналізу даних, що надходять і даних тривалого спостереження. Вимірювання включають в себе централізовані оцінки поведінки груп, аналіз між мережевої взаємодії і аналіз відповідностей.

Деякі з методів аналізу мереж застосовують математики, але їх більше цікавлять кількісні характеристики структури мережі, а соціальну поведінку оцінюється виходячи з аналізу схеми з'єднань між вузлами мережі. З огляду на, що структура складних мереж неоднорідна і динамічно розвивається з часом, основний напрямок досліджень – це розробка надійних математичних методів оцінки мереж, що складаються з мільйонів вузлів. Математики і фізики нерідко використовують знання, отримані в ході вивчення біологічних систем, важливим методом вивчення поведінки яких є аналіз протяжності маршрутів і кластерний аналіз мережевої структури. У базовій формі соціальні мережі можна представити у вигляді графів, а більш складні топографії представляють у вигляді зважених, статечних, просторових мереж або випадкових графів.

Один із загальноприйнятих підходів до управління такими мережами – розбиття графа спектральним способом, коли визначається мінімальна кількість ребер між двома групами вершин. Для мереж з невідомим заздалегідь кількістю спільнот ефективний метод ієрархічної кластеризації – розбиття вузлів на кластери в залежності від ступеня зв'язності. Є також

методи кластерного аналізу, засновані на пошуку найбільшої дистанції між вузлами.

Фахівці з інформатики, спираючись на теорії соціальних і складних мереж, ведуть дослідження в області мережевих середовищ, що виконують роль інформаційних систем. Активно вивчається фундаментальне питання про подібність соціальних мереж Інтернету з колективним поведінкою людей в реальних ситуаціях. Для цього застосовуються комбіновані методи, запозичені з соціології та математики. Наприклад, метод аналізу веб-графів спирається на традиційні методи аналізу мереж, але з урахуванням особливостей Всесвітньої павутини.

Вивчення соціальних мереж перетворюється в задачу обробки Великих Даних, коли бізнес-керівникам або фахівцям з інформаційних систем потрібно прогнозувати поведінку учасників спільноти, щоб домогтися підвищення ефективності маркетингу або продажів. У багатьох соціальних сайтів є від 10 до 200 млн користувачів, тому стрижнем більшості досліджень є робота з вибірками даних. Оптимальним, хоча і сильно витратним за часом, було б витяг знання з усього зрізу даних, що для Великих Даних, які характеризуються трьома «V» (volume, velocity, variety–обсяг, швидкість, різноманіття), нереально. Уже в кінці 2011 року у Facebook було 721 млн користувачів і 68,7 млрд ребер-зв'язків між «друзями». Якщо говорити про швидкість наповнення, то Twitter і Facebook генерують 7 Тбайт і 10 Тбайт відповідно щодня. Але ці дані нерідко потрібно обробляти буквально зі швидкістю думки. Наприклад, 11 листопада 2012 року на TaoBao, найбільшої роздрібної онлайн-майданчику Китаю, пройшла розпродаж, протягом якої було зроблено 100 млн покупок, а піковий темп продажів досяг 205 тис. транзакцій в хвилину. Що ж до різноманіття – дані сьогодні надходять з різних джерел, від камер спостереження, супутників і твітів до всіляких сенсорів і електролічильників.

2.2 Обґрунтування та вибір методів дослідження

2.2.1 Соціальні графи та їх аналіз

Термін «соціальний граф» досі не має єдиного визначення. Кажучи про нього, можна розуміти контекстуальну соціограму, що описує учасників, організації, групи всередині якоїсь соціальної мережі, а також відношення між ними.

Аналіз соціальних графів – це міждисциплінна область, що фокусується на вивчення взаємозв’язаних сутностей, включаючи соціальні, біологічні, комунікаційні і комп’ютерні мережі. Головною метою цієї області є отримання пояснюючих та прогнозуючих моделей для фізичних, соціальних, технологічних, біологічних та інших феноменів. У соціальних мережах існує велика кількість типів відносин між цими сутностями. Їх дослідження допомагає краще розуміти соціум, що є дуже важливим для сучасного таргетованого маркетингу. Воно включає в себе, як приклад, розробку аналітичних мір (analytical measures), алгоритмів для виділення спільноти, прогнозування зв’язків між сутностями та аналіз їх сильно-зв’язаних вузлів (хабів).

Деякі з існуючих теоретичних досліджень знайшли своє призначення у наступному:

- моніторінг та аналіз мобільних мереж (мережі рухливості), використовуючи траєкторії як ребра, що поєднують різні географічні райони та «точки інтересу»;
- ідентифікація можливих похибок або помилок в забезпеченні приватності та безпеки;
- вірусний маркетинг у соціальних мережах;
- система, що дозволяє приймати рішення щодо пошуку та виловлення злочинців, аналізуючи злочинні тенденції;
- слідкування та прогнозування можливих епідеміологічних очагів за допомогою Twitter.

Аналіз соціальних мереж, зокрема спільнот Facebook, є одним з перших етапів обробки великих даних (тобто всіх даних, що можна уявити, а також зв’язків між сутностями) комп’ютерною системою формування рекламних пропозицій у суспільних мережах з використанням соціальних графів.

2.2.2 Характеристики соціальних графів

Говорячи про задачі на соціальному графові, часто використовують термін метрики, які в чисельній формі відображують характеристики соціальних об’єктів, сегментів або груп об’єктів та їх зв’язків. Ці метрики використовують при проведенні аналізу соціальних мереж. Найбільш спільні серед них:

– кількість вершин;

– кількість унікальних ребер;

– кількість петель;

– кількість зв’язних компонентів.

Метрики, що характеризують зв’язки:

– гомогенність – ступінь, з якою схожі учасники формують зв’язки між собою у порівнянні з несхожими;

– множинність – кількість форм, що містяться у зв’язку;

– спільність – ступінь, з якою двоє учасників відповідають один одному взаємністю у сфері будь-яких взаємодій.

Метрики, що характеризують розподіл:

– міст – індивід, слабкі зв’язки якого заповнюють структурні пробіли, забезпечуючи єдине поєднання між двома іншими індивідами або кластерами;

– центральність – визначає значення або вплив певного вузла чи групи у мережі;

– структурні пробіли – відсутність зв’язку між двома частинами мережі.

Найбільш корисними та популярними є метрики:

– degreeCentrality (ступінь вершини) – як багато людей (сутностей) пов’язані з цією людиною напряму;

– betweenness (проміжність) – яка ймовірність того, що цей вузол буде елементом найбільш короткого шляху, що поєднує дві інші сутності у мережу;

– closeness (близькість) – як швидко людина може досягнути всіх інших учасників мережі;

– eigenvector – наскільки добре ця людина пов’язана з іншими «сильно зв’язаними» людьми;

– shortestPath (найбільш короткий шлях) – мінімальна кількість зв’язків, що потребуються для встановлення наявності взаємозв’язків між іншими двома окремими користувачами;

– set of key players.

Ілюстрацію даних метрик можна побачити на рисунку (рис.2.1):

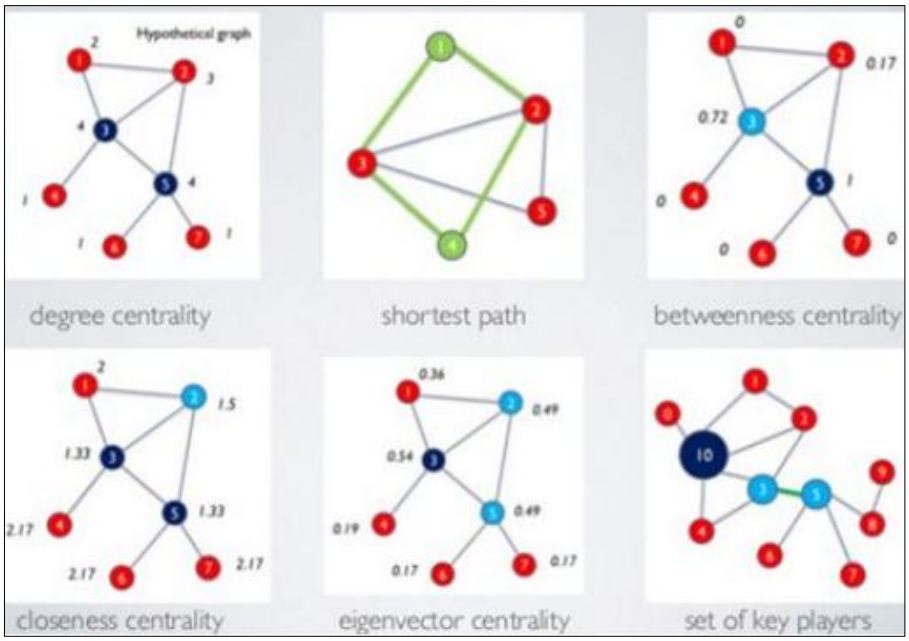


Рисунок 2.1 – Приклад метрик соціальних графів

Складовими частинами будь-якої соціальної мережі є:

– хаби – вузли, що мають високу ступінь (рис 2.2);

– мости – тип зв’язку, що поєднує 2 різні групи в одній мережі (рис. 2.3);

– острови – слабо зв’язані вузли або групи вузлів (рис 2.4);

– кластери – групи сильно пов’язаних між собою вузлів (рис 2.5).

Рисунок 2.2 – Приклад хабів в соціальному графі

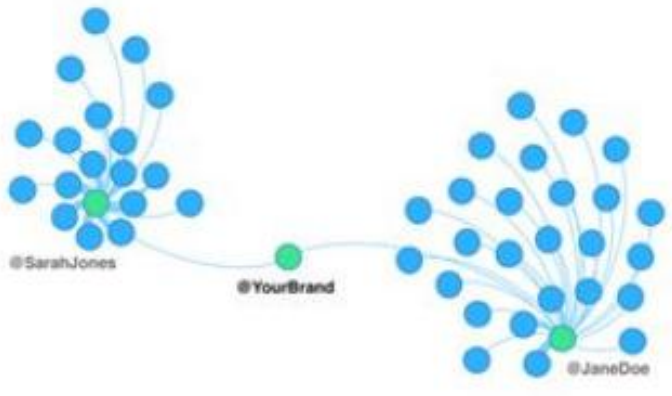


Рисунок 2.3 – Приклад мостів в соціальному графі

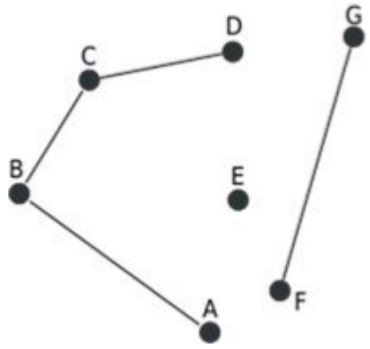


Рисунок 2.4 – Приклад островів в соціальному графі

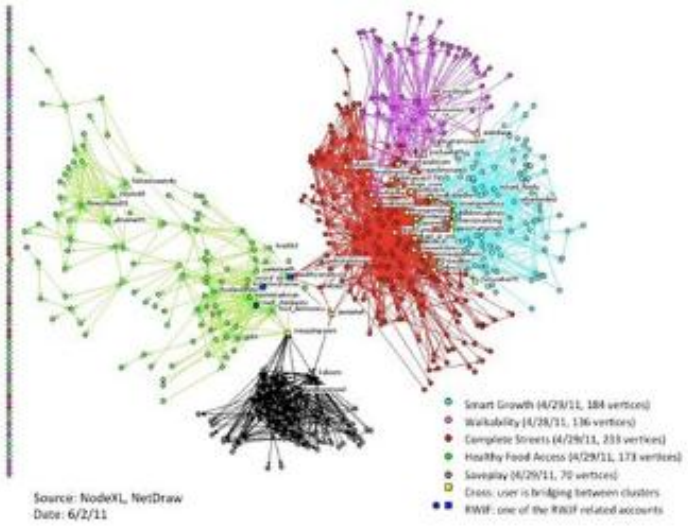


Рисунок 2.5 – Приклад кластерів в соціальному графі

У роботі об’єктом аналізу є ринок послуг, точніше – продажу турів, тобто відношення «мандрівник-туроператор». Їх взаємодію можна представити орієнтованим дводольним графом.

У комп’ютерній системі формування рекламних пропозицій у суспільних мережах соціальні графи дозволяють виділити користувачів, на яких варто розраховувати при поширенні контенту.

2.2.3 Основні алгоритми аналізу графів

Найбільш розповсюдженими задачами аналізу соціальних графів є виявлення особливостей структури. Наприклад, можна виявити спільноти, тобто, групи вузлів, які мають сильну зв’язковість між собою та слабку з вузлами не входять в дану спільноту. Також часто потребується визначити головних (найбільш впливових) членів групи. Широке втілення та застосування знаходять алгоритми, які можуть прогнозувати можливі зв’язки між сутностями, а також алгоритми рекомендацій.

Одним з таких є алгоритм PageRank. PageRank – один з алгоритмів посилання ранжирування. Алгоритм застосовується до колекції документів, пов'язаних гіперпосиланнями (таких, як веб-сторінки з всесвітньої павутини), і призначає кожному з них якийсь чисельне значення, що вимірює його «важливість» або «авторитетність» серед інших документів. Взагалі кажучи, алгоритм може застосовуватися не тільки до веб-сторінок, але і до будь-якого набору об'єктів, пов'язаних між собою взаємними посиланнями, тобто до будь-якого графу. PageRank – це числова величина, що характеризує «важливість» веб-сторінки. Чим більше посилань на сторінку, тим вона «важливіше». Крім того, «вага» сторінки A визначається вагою посилання, переданої сторінкою B. Таким чином, PageRank–це метод обчислення ваги сторінки шляхом підрахунку важливості посилань на неї. Надбудова для браузера Google Toolbar показує для кожної веб-сторінки ціле число від 0 до 10, яке вона називає PageRank, або важливістю цієї сторінки з точки зору Google. Однак механізм його розрахунку і що в точності позначає це значення, не розкривається. За деякими даними, ці значення оновлюються лише кілька разів на рік (в той час, як внутрішні значення PageRank перераховуються безперервно і показують значення PageRank сторінок на логарифмічною шкалою. Щомісяця Google оновлює алгоритми, які істотно вплинули на формування видачі. На основі цієї

інформації ви зможете проаналізувати стан свого сайту і виявити проблеми, через які виникають труднощі в піднятті сторінки наверх пошуку.

Персоналізований PageRank досить довго застосовували (і застосовують) у багатьох соціальних мережах (наприклад, Twitter) для того, щоб пропонувати користувачам акаунти, що з високою ймовірністю можуть бути цікавими для них.

Іншим класичним прикладом може бути пошук сильно зв’язаних компонентів. Пошукові системи на кшталт Google і Bing використовують в своїх інтересах той факт, що сторінки в інтернеті утворюють дуже великий спрямований граф. Щоб перетворити в нього Всесвітню Павутину, ми будемо розглядати сторінки, як вершини, а гіперпосилання між ними – як з'єднують їх ребра. Звичайно, цей граф був би величезним, так що ми обмежили його сайтами, на які є не більше десяти посилань з домашньої сторінки CS.

У системі формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data цей алгоритм може використовуватись для виділення тісно взаємодіючих груп людей, щоб пропонувати членам цієї групи, наприклад, сторінки, які вважають цікавими інші члени цієї групи (вашим друзям подобається…).

Також варто відмітити алгоритм підрахунку найкоротших шляхів у графі, що знаходить своє призначення та використання у пошуку найбільш впливових вузлів та виділенні спільнот. Більшість алгоритмів є ітеративними.

2.2.4 Генерація випадкових соціальних графів

Незважаючи на наявність коштів для збору даних із соціальних мереж і великої кількості доступних наборів даних, актуальною є задача створення моделей випадкових соціальних графів і інструментів для генерації випадкових графів із заданим набором властивостей [20].

У розроблюваній системі формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data для достовірного тестування методів аналізу соціальних даних соціальні графи повинні бути застосовані до безлічі наборів даних з різними властивостями.

Наприклад, методи пошуку спільнот користувачів в соціальному графі можуть показувати істотно різні результати в залежності від розміру початкового графа, середнього ступеня вершини, коефіцієнта кластеризації та інших структурних властивостей. Збір необхідних для достовірного тестування реальних даних ускладнений не тільки внаслідок тимчасових витрат на скачування і обробку великих масивів слабоструктурованої інформації, але і в силу складності управління процесом збору з метою отримання набору даних з конкретним набором властивостей. На малюнках нижче можна побачити, як виглядають такі масштабні графи соціальних мереж (рис. 2.6) та (рис. 2.7).

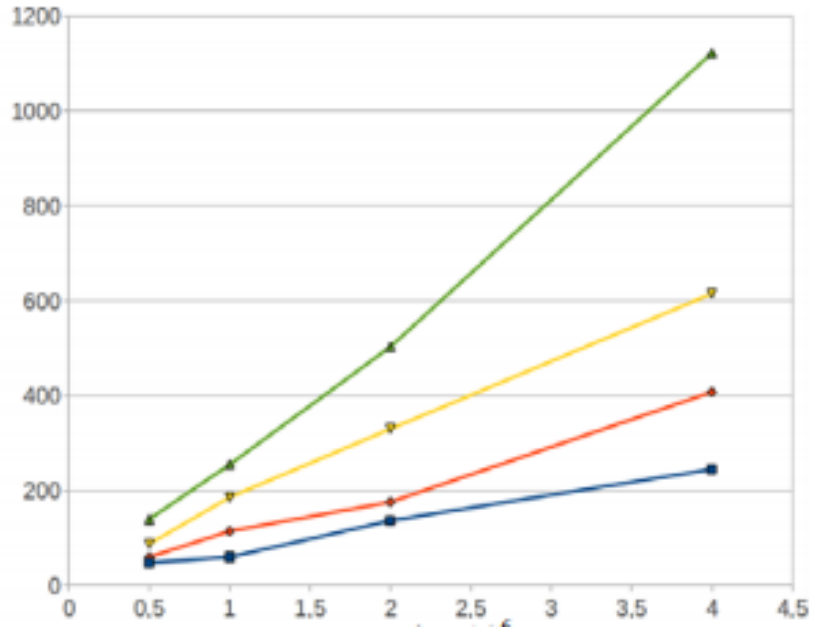


Рисунок 2.6 – Розмір графа – 10^6 вершин

Тестування часу генерації випадкових графів з заданою структурою спільнот. Вгорі: на кластерах Amazon EC2 з різною кількістю робочих вузлів типу m1.large: зелена лінія – 2 вузла, жовта лінія–4 вузли, червона лінія -8 вузлів, синя лінія–16 вузлів.

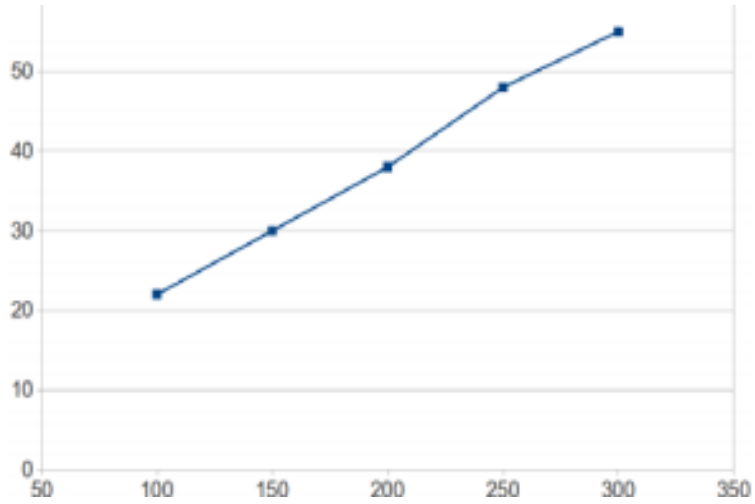


Рисунок 2.7 – Розмір графа – 10^3 вершин

2.3 Можливості в Apache Spark

Apache Spark (від англ. Spark – іскра, спалах) – фреймворк з відкритим вихідним кодом для реалізації розподіленої обробки неструктурованих і слабоструктурованих даних, що входить в екосистему проектів Hadoop. На відміну від класичного обробника з ядра Hadoop, що реалізує дворівневу концепцію MapReduce зі зберіганням проміжних даних на накопичувачах, Spark працює в парадигмі резидентних обчислень (англ. In-memory computing) – обробляє дані в оперативній пам'яті, завдяки чому дозволяє отримувати значний виграш в швидкості роботи для деяких класів задач, зокрема, можливість багаторазового доступу до завантажених в пам'ять призначених для користувача даних робить бібліотеку привабливою для алгоритмів машинного навчання [19].

Проект надає програмні інтерфейси для мов Java, Scala, Python, R. Спочатку написаний на Scala, згодом додана істотна частина коду на Java для надання можливості написання програм безпосередньо на Java. Складається з ядра і кількох розширень, таких як Spark SQL (дозволяє виконувати SQL-запити над даними), Spark Streaming (надбудова для обробки потокових даних), Spark MLlib (набір бібліотек машинного навчання), GraphX (призначене для розподіленої обробки графів). Може працювати як в середовищі кластера Hadoop під керуванням YARN, так і без компонентів ядра Hadoop, підтримує кілька розподілених систем зберігання – HDFS, OpenStack Swift, NoSQL-СУБД Cassandra, Amazon S3.

Spark – це проект Apache, який позиціонується як інструмент для «блискавичних кластерних обчислень». Проект розробляється процвітаючою вільною спільнотою, зараз є найбільш активним з проектів Apache.

Spark надає швидку та універсальну платформу для обробки даних. У порівнянні з Hadoop Spark прискорює роботу програм в пам'яті більш ніж в 100 разів, а на диску – більш ніж в 10 разів.

Крім того, код на Spark пишеться швидше, оскільки тут у вашому розпорядженні буде більше 80 високорівневих операторів. Щоб оцінити це, давайте розглянемо аналог «Hello World!» зі світу BigData: приклад з підрахунком слів (Word Count). Програма, написана на Java для містила б близько 50 рядків коду, а на Spark нам буде потрібно всього лише 4 рядки.

При вивченні Apache Spark варто відзначити ще один важливий аспект: тут надається готова інтерактивна оболонка (REPL). За допомогою REPL можна протестувати результат виконання кожного рядка коду без необхідності спочатку програмувати і виконувати всі завдання цілком. Тому написати готовий код вдається набагато швидше, крім того, забезпечується ситуативний аналіз даних [19].

Крім того, Spark має наступні ключові риси:

– в даний час надає API для Scala, Java і Python, також готується підтримка інших мов (наприклад, R) [19];

– добре інтегрується з екосистемою Hadoop і джерелами даних (HDFS, Amazon S3, Hive, HBase, Cassandra);

– може працювати на кластерах під керуванням Hadoop YARN або Apache Mesos, а також працювати в автономному режимі [19].

Ядро Spark доповнюється набором потужних високорівневих бібліотек, які без швів стикуються з ним в рамках того ж додатка. В даний час до таких бібліотек відносяться SparkSQL, Spark Streaming, MLlib (для машинного навчання) і GraphX. Зараз також розробляються інші бібліотеки та розширення Spark [19].

Ядро Spark – це базовий двигун для великомасштабної паралельної і розподіленої обробки даних. Ядро відповідає за:

– управління пам'яттю і відновлення після відмов;

– планування, розподіл і відстеження завдань кластерів;

– взаємодія з системами зберігання даних.

В Spark вводиться концепція RDD (стійкий розподілений набір даних) – незмінна і стійка до відмов розподілена колекція об'єктів, які можна обробляти паралельно. У RDD можуть міститися об'єкти будь-яких типів. RDD створюється шляхом завантаження зовнішнього набору даних або розподілу колекції з основної програми (driver program). У RDD підтримуються операції двох типів:

– Трансформації – це операції (наприклад, відображення, фільтрація, об'єднання), що здійснюються над RDD. Результатом трансформації стає новий RDD, що містить її результат;

– Дії – це операції (наприклад, редукція, підрахунок), які повертають значення, що отримується в результаті деяких обчислень в RDD.

Трансформації в Spark здійснюються в «ледачому» режимі – тобто, результат не обчислюється відразу після трансформації. Замість цього вони просто «запам'ятовують» операцію, яку слід провести, і набір даних (наприклад файл), над яким потрібно зробити операцію. Обчислення трансформацій відбувається тільки тоді, коли викликається дією, і його результат повертається основній програмі. Завдяки такому дизайну підвищується ефективність Spark. Наприклад, якщо великий файл був перетворений різними способами і переданий першої дії, то Spark обробить і поверне результат лише для першого рядка, а не стане опрацьовувати таким чином весь файл [19].

За стандартними налаштуваннями кожен трансформований RDD може переобчислюватись щоразу, коли ви виконуєте над ним нову дію. Однак RDD також можна довготривало зберігати в пам'яті, використовуючи для цього метод зберігання або кешування; в такому випадку Spark буде тримати потрібні елементи на кластері, і ви зможете запитувати їх набагато швидше.

Потенційно сфера застосування Spark, зрозуміло, далеко не обмежується сейсмологією. Ось добірка практичних ситуацій, де потрібна швидкісна, різнопланова і об'ємна обробка великих даних, для якої добре підходить Spark:

− В ігровій індустрії (обробка і виявлення закономірностей, що описують ігрові події, що надходять суцільним потоком в реальному часі. В результаті ми можемо негайно на них реагувати і робити на цьому хороші гроші, застосовуючи утримання гравців, цільову рекламу, автокорекцію рівня складності і т.д);

− В електронній комерції (інформація про транзакції, що надходить в реальному часі, може передаватися в потоковий алгоритм кластеризації, наприклад, по k-середнім або піддаватися спільній фільтрації, як у випадку ALS. Потім результати навіть можна комбінувати з інформацією з інших неструктурованих джерел даних – наприклад, з відгуками покупців або рецензіями;

− Поступово цю інформацію можна застосовувати для вдосконалення рекомендацій з урахуванням нових тенденцій), у фінансовій сфері або при забезпеченні безпеки (стек Spark може застосовуватися для виявлення шахрайства або вторгнень, або для аутентифікації з урахуванням аналізу ризиків).

Таким чином у проектованій комп’ютерній системі можна отримувати першокласні результати, збираючи величезні обсяги архівованих логів, комбінуючи їх з зовнішніми джерелами даних, наприклад, з інформацією про витоки даних, а також використовувати інформацію про з'єднання/запити, орієнтуючись, наприклад, на геолокацію по IP або на дані про час.

2.4 Можливості мови розробки Python

Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий набір корисних функцій.

Python підтримує структурний, узагальнене, об'єктно-орієнтоване, функціональне і аспектно-орієнтоване програмування. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується

розбиття програм на модулі, які, в свою чергу, можуть об'єднуватися в пакети.

Еталонної реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється під вільною ліцензією Python Software Foundation License, що дозволяє використовувати його без обмежень в будь-яких додатках. Є реалізація інтерпретатора для JVM з можливістю компіляції, CLR, LLVM, інші незалежні реалізації. Проект PyPy використовує JIT-компіляцію, яка значно збільшує швидкість виконання Python-програм.

Переваги:

– Низький поріг входження. Синтаксис Python більш зрозумілий для новачка.

– Логічний, лаконічний і зрозумілий. У порівнянні з багатьма іншими мовами Python має легкий синтаксис.

– Багатоплатформовий: підходить для різних платформ: і Linux, і Windows.

– Є реалізація інтерпретаторів для мобільних пристроїв і непопулярних систем.

– Широке застосування. Використовується для розробки веб-додатків, ігор, зручний для автоматизації, математичних обчислень, машинного навчання, в області інтернету речей. Існує реалізація під назвою Micro Python, оптимізована для запуску на мікро-контролерах (можна писати інструкції, логіку взаємодії пристроїв, організовувати зв'язок, реалізовувати розумний будинок).

– Потужна підтримка компаній-гігантів IT-індустрії. Такі компанії, як Google, Facebook, Dropbox, Spotify, Quora, Netflix, на певних етапах розробки використовували саме Python.

– Висока затребуваність на ринку праці.

У світі Python багато якісних бібліотек, так що не потрібно винаходити велосипед, якщо треба терміново вирішити якусь комерційну завдання. Для навчання є багато розумних книг, в першу чергу англійською мовою, звичайно, але і в перекладі також видана гідна література. Сьогодні багато навчальних матеріалів на Youtube: відео блоги, записи вебінарів і конференцій. Python відрізняється суворою вимогою до написання коду (вимагає відступи), що є перевагою, за моїми спостереженнями. Спочатку мова сприяє писати код організовано і красиво.

Python розвивається і не згасне ще довго. На численні оглядам і рейтингам мову займає високі позиції. Згідно DOU він знаходиться на п'ятому місці і займає третю позицію в веб-технологіях.

2.5 Визначення демографічних атрибутів користувачів

При заповненні свого профілю в соціальній мережі користувачі найчастіше помилково або навмисно не заповнюють деякі поля або дають неправдиву інформацію про факти своєї біографії, інтереси та вподобання.

Демографічні атрибути можна умовно розділити на категоріальні (Стать, національність, раса, сімейний стан, рівень освіти, професія, працевлаштованість, релігійні і політичні погляди) і чисельні (вік, рівень доходів) [21]. Умовність поділу пов'язана з тим, що значення чисельного атрибута можна відобразити в набір категорій і надалі розглядати цей атрибут як категоріальний. Зокрема, значення віку можна розділити на кілька вікових категорій, що часто застосовується на практиці. При заповненні свого профілю в соціальній мережі користувачі найчастіше помилково або навмисно не заповнюють деякі поля або дають неправдиву інформацію про факти своєї біографії, інтереси та вподобання.

В тематичних мережах (Twitter, YouTube) призначений для користувача профіль часто обмежений набором базових атрибутів,

недостатнім для вирішення багатьох задач, які передбачають персоналізацію результатів [18].

Таким чином, актуальні методи часткової ідентифікації авторів повідомлень за значеннями їх демографічних атрибутів. Зокрема, в системах інтернет-маркетингу і рекомендацій особливу важливість представляє визначення демографічних атрибутів користувача для таргетованого просування товарів і послуг в групах користувачів з однаковими значеннями атрибутів. Крім інтернет-сервісів, такі демографічні характеристики знаходять застосування в різних дисциплінах: соціологія, психологія, кримінологія, економіка, управління персоналом та ін.

Метод визначення демографічних атрибутів користувачів соціальної мережі за текстами їх записів володіє наступними особливостями:

– широкий набір підтримуваних атрибутів: стать, вік, сімейний стан, релігійні і політичні погляди;
– широкий набір підтримуваних мов;
– повністю автоматичний метод збору і розмітки корпусів повідомлень користувачів інтернету для всіх підтримуваних атрибутів мов.

Метод складається з наступних етапів:

– побудова вихідного набору даних;
– попередня обробка тексту;
– побудова простору ознак опису;
– відбір інформативних ознак;
– навчання;
– класифікація.

Всі етапи, за винятком першого, виконуються окремо для кожного атрибута.

На етапі побудови вихідного набору даних проводиться збір даних користувачів з мережі Twitter. Для кожного користувача спочатку запитується тільки його профіль у мережі Twitter. При наявності в ньому посилання на профіль того ж користувача в мережі Facebook (в якій набір призначених для користувача атрибутів істотно більше, ніж в Twitter) запитуються і зберігаються всі доступні повідомлення користувача з мережі Twitter. Після чого для поточного користувача запитується і зберігається його профіль у мережі Facebook, з якого здобуваються зазначені користувачем значення його атрибутів.

На етапі попередньої обробки тексту до текстів отриманого на попередньому етапі набору даних застосовується метод визначення мовної приналежності тексту. Після цього дані користувачів розподіляються в різні набори даних в залежності від мови користувача. Крім того, на цьому етапі здійснюється фільтрація повідомлень, авторство яких не належить користувачу (репости), оскільки цитування повідомлень інших користувачів є популярним способом поширення інформації, цей крок попередньої обробки особливо важливий для підвищення точності методу. Таким чином, елементом набору даних для кожного атрибута і мови є набір символьних рядків, отриманих з текстів повідомлень і профілю одного користувача в Twitter, а також значення атрибута у даного користувача в Facebook.

На етапі побудови простору ознак опису з повідомлень користувачів витягуються лінгвістичні ознаки. З отриманих токенів будується набір ознак у вигляді N-грам розміром від 1 до 3 з урахуванням порядку токенів.

Кожен тип ознак представлений двома підтипами: з урахуванням і без урахування регістра символів.

Підсумковий вектор ознак для користувача є бінарним, тобто містить тільки інформацію про наявність чи відсутність ознаки в його текстових даних. Кількість примірників однієї ознаки ігнорується.

На етапі відбору інформативних ознак застосовується метод, заснований на розрахунку умовної взаємної інформації. Виготовляється ітеративний відбір тих ознак, які містять найбільшу кількість інформації про значення атрибута і при цьому істотно відрізняються від ознак, обраних на попередніх ітераціях.

Таким чином, кожна ознака результуючого набору високо інформативна і слабо залежить від інших ознак.

На етапі навчання проводиться побудова моделі класифікації з використанням онлайн пасивно-агресивного алгоритму. На етапі класифікації в якості вхідних даних використовуються тексти повідомлень і поля профілю довільного користувача. Виконується алгоритм класифікація для заданого мови і атрибута. Результатом є значення атрибута обраного користувача.

2.6 Пошук описаних подій

Повідомлення користувачів соціальних мереж складають істотну частку текстового контенту сучасного Інтернету. Крім того, соціальні мережі часто виступають в ролі неформальних ЗМІ, де будь-який користувач може опублікувати новинне повідомлення про події, що відбуваються (Інформаційні приводи), що і використовує проектована комп'ютерна система рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data.

Разом з тим, щоб автоматично завантажувати набори повідомлень про невідомому заздалегідь подію є нетривіальним завданням в силу наступних чинників:

- великий обсяг вхідних даних (наприклад, користувачі Twitter публікують кілька тисяч повідомлень щосекунди);
- велика кількість нерелевантних/неінформативних повідомлень;
- користувачі можуть по-різному описувати одну і ту ж подію;

– різні події можуть збігатися за часом;

– складність поділу події і його підподій (наприклад, Олімпійські ігри і конкретний футбольний матч в рамках цього контексту).

2.7 Висновки за розділом

Великі дані можуть допомогти споживачам визначитися з найбільш прийнятними для них показниками покупки продукту, даючи при цьому маркетологам можливість сформувати чіткіший і зрозуміліший портрет їхніх клієнтів. У руках умілих маркетологів великі дані можуть бути використані для тестування та прогнозування можливої реакції споживачів на різні маркетингові повідомлення.

Інтернет є основним джерелом великих даних. Все, починаючи від веб-сайтів до аналітики соціальних медіа переходу за посиланнями рекламних оголошень, може бути легко об'єднано, проаналізовано та інтерпретовано. Зрештою, робота з таким величезним джерелом інформації, як великі дані, призвела до створення безлічі нових онлайн форм цифрового маркетингу.

3 СИНТЕЗ СИСТЕМИ

3.1 Вибір і обґрунтування принципів побудови проектованої комп’ютерної системи

Побудова комп'ютерної системи включає в себе формування технічних вимог, функціональних вимог, вимог до видів забезпечення та вимог до захисту інформації на базі яких буде зроблений вибір устаткування для створення структурної схеми комп'ютерної системи формування рекламних пропозицій.

Проектування комп'ютерної системи здійснюватиметься з урахуванням створеної структурної схеми та схеми функціональної структури.

Комп'ютера система формування рекламних пропозицій повинна виконувати наступні функції:

- Збирати та динамічно оновлювати інформацію зі сторінок Facebook (100-200 спільнот);
- Об'єднувати інформацію, так як вона часто перебуває у різних джерелах;
- Працювати з великими обсягами не структурованих даних (неоднорідні дані від 150 GB);
- Збирати дані про новоприйнятих у групах Facebook;
- Вивантажувати ID груп, на які підписано конкретного користувача;
- Конвертувати ім’я користувача в його ID та навпаки;
- Отримувати інформацію про користувачів (стать, вік, день народження, геолокація);
- Шукати користувачів, спільноти, публічні сторінки по email/номеру телефону;
- Шукати цільову аудиторію туристів;

– Шукати розширену інформацію про користувача (сімейний стан, родичі, навчання);

– Робити пошук по хештегам;

– Збирати з геолокацію.

В цілому комп’ютерна система формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data, має обслуговувати величезний потік динамічно мінливої інформації, що не в силах забезпечити одна людина або навіть злагоджена команда операторів.

3.2 Формулювання технічних вимог до комп’ютерної системи.

3.2.1 Вимоги до системи в цілому

Веб-технології, які мають бути використані у системі формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data:

Щоб створити алгоритм роботи парсеру, доведеться проаналізувати вихідний код сторінок сайту-донора, за це відповідають технології HTML, CSS і JavaScript.

Технологія DOM – дозволяє з максимальним ефектом працювати з ієрархічним деревом веб-документа (рис. 3.1).

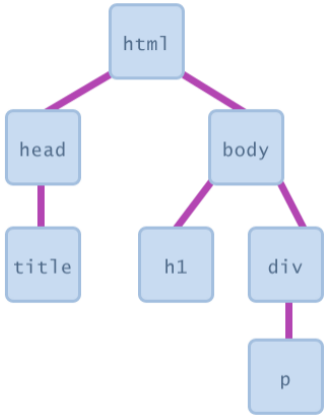


Рисунок 3.1 – Ієрархічне дерево веб-документа

На найважливішому і складному етапі – написанні аналізатора, знадобиться використовувати будь-який з інструментів текстової обробки. Один з варіантів для пошуку потрібних фрагментів тексту – скористатися регулярними виразами. Але оптимальнішим виходом буде не винахід колеса, а використання готових бібліотек для парсинга.

Для ефективної роботи з ієрархічними структурами даних потрібно скористатись парадигмою об'єктно-орієнтованого програмування. Семантичне дерево можна будувати і за допомогою багатовимірних асоціативних масивів, але, цей спосіб не логічний у випадку, коли дерево містить велику кількість вузлів, ООП відмінно підтримується всіма мовами розглянутими на сторінках Facebook.

Фінальна обробка результатів передбачає збереження даних в структурованому вигляді.

Дані буде потрібно записувати в CSV-файли та конвертувати в електронні таблиці. Іноді спарсені дані заливаються в нову базу даних за допомогою JSON. Дану javascript-технологію теж необхідно використати як додаткову можливість.

3.2.2 Вимоги до видів забезпечення

Для забезпечення функціонування системи формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data, необхідний персональний комп’ютер з рекомендованими характеристиками які вказані в таблиці 3.1.

Так як для забезпечення парсингу потрібна достатньо сильна потужність (від 4096 MB вільної оперативної пам’яті), і для збереження даних знадобиться великий об’єм жорсткого диску (від 30 GB) то можна сказати, що персональний комп’ютер з такими, або кращими характеристиками, що вказані у таблиці 3.1 рекомендується для розробки

системи формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data.

Таблиця 3.1 – Рекомендовані характеристики ПК для забезпечення функціонування комп’ютерної системи.

Тип	Найменування
Процесор	Intel Core i5, 1,8 GHz
Оперативна пам’ять	8 ГБ
Жорсткий диск	500 ГБ
Відео адаптер	Intel HD Graphics 6000
Операційна система	Windows 7/8/10

Для виводу і опрацювання інформації, що збирає система формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data, необхідний широкоформатний монітор з рекомендованими показниками які вказані у таблиці 3.2. Для аналізу даних зібраних системою потрібно відображати одразу три вікна на моніторі.

Таблиця 3.2 – Рекомендовані характеристики монітору для для виводу і опрацювання інформації, що збирає система.

Тип	Найменування
Діагональ	49"
Тип матриці	Vertical Alignment
Частота розгортки	240 ГЦ
Роздільна здатність	5120x1440
Співвідношення сторін	32:9
Роз'єми	HDMI, USB, mini-jack 3.5

Для забезпечення якісної і безпечної роботи спеціаліста, що буде обслуговувати систему формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data необхідне правильно обладнане робоче місце.

Площа робочого місця повинна становити не менше 6 м^2, для ПК з плоским дисплеєм – 4,5 м^2 (об'ємні норми на одну особу – не менше 20 м^3). У приміщеннях має проводитися щоденне вологе прибирання та систематичне провітрювання після кожної години роботи. Шумне обладнання (друкарські пристрої, сканери, сервери тощо), рівні шуму якого перевищують нормативні, має розміщуватися поза робочими місцями працівників.

Важливо, щоб спеціаліст, сидячи за комп'ютером, знаходився за добре освітленим робочим столом.

Робочий стіл слід розміщувати таким чином, щоб монітор був орієнтований бічною стороною до світлових прорізів, щоб природне світло падало переважно ліворуч.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні обладнання, що використовується. Висота робочої поверхні столу повинна становити 725 мм, робоча поверхня стола повинна мати ширину 800-1400 мм і глибину 800-1000 мм. Робочий стіл повинен мати простір для ніг заввишки не менше 600 мм, шириною – не менше 500 мм, глибиною на рівні колін – не менше 450 мм та на рівні витягнутих ніг – не менше 650 мм.

Конструкція робочого стільця або крісла повинна забезпечувати підтримання раціональної робочої пози працівника та дозволяти змінювати позу з метою зниження статичної напруги м'язів шийно-плечової ділянки та спини.

Клавіатуру слід розташовувати на поверхні столу на відстані 100-300 мм від краю, зверненого до користувача, або на спеціальній поверхні, відокремленій від основної стільниці.

Екран монітору повинен знаходитись від очей користувача на відстані 600-700 мм, але не ближче 500 мм [17].

3.2.3 Вимоги до захисту інформації

При розробці комп'ютерної системи дані будуть зберігатися на відділеному сервері бази даних, доступ до якого буде відкрито у обмеженої кількості персоналу. Захист інформації буде реалізовано за допомогою серверних технологій захисту інформації таких як SSH-ключі та фаервол.

На розробку додаткових функцій захисту інформації можна витратити багато часу і людських сил. З огляду на те, що зібрана системою інформація несе цінність тільки для турагенства (що і збирає ці дані), то в розробці додаткових систем захисту системи формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data немає необхідності.

3.3 Синтез структурної схеми за заданими показниками комп'ютерної системи

З огляду на попередні пункти розділу створюється структурна схема комп'ютерної системи формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data.

У Facebook існує ряд спільнот, де користувачі у коментарях залишають питання щодо типових турів. Головна задача маркетологів — зібрати ID користувачів та інформацію, яку вони пишуть, структурувати її у таблиці формату «користувач — запит» і отримати декілька списків аудиторій.

На першому етапі задача — розробити парсер, який видаватиме файл типу .JSON з даними «користувач — запит». Поки що дані не структуровані у таблицю. Для написання файлу використовується мова програмування Python.

Тури для пропозицій брались з сайту клієнта. API-вакансії представлені в форматі JSON, який має наступні поля:

- опис;
- країна;
- місто;
- умови (готель, переліт, умови для дітей та ін.);
- вартість;
- необхідність візи.

Виходячи з формату отриманої інформації, були обрані методи її обробки. Для таких полів, як «країна», «місто», «вартість», «громадянство» додаткової обробки не потрібно, тому що вони зберігаються в стандартному вигляді. Поля з описом туру, умовами, побажаннями були розбиті таким чином.

Спочатку вони були розбиті на пропозиції і збережені у вигляді списків, де кожен елемент – окрема пропозиція. Потім в кожному елементі (Пропозиції) була видалена пунктуація, він був розбитий на окремі слова. Після був застосований стемінг. До ключових побажань також був застосований стемінг, і вони були збережені як окремий список.

Файли з запитами були оброблені тим же способом, що і пропозиції, за винятком поля з візою. Якщо воно не порожнє, то з нього витягуються країни, куди додатково може поїхати ця людина і дані щодо візи – коли отримана і якого типу вона є.

Для вирішення цього завдання використовується модифікований ітеративний алгоритм Гейла-Шеплі. Спочатку визначено спосіб

ранжирування елементів з однієї безлічі. Потім для кожної вершини сформовано список бажаних елементів іншого безлічі. Після цього починає роботу безпосередньо алгоритм, кожна ітерація включає в себе стадію пропозиції, згоди і відмови:

1. Кожен непов'язаний ні з ким елемент з безлічі пошуковців пропонує свою кандидатуру, на основі своїх інтересів, найбільш відповідних турагентству, до якого він ще не звертався.

2. Кожен клієнт відповідає згодою на тур, який найбільше бажає (причому такий кандидат може бути обраний на попередньої ітерації) і відхиляє пропозиції інших.

Парсер віддає готовий файл, але він не придатний до роботи, тому що містить тільки ID користувача та цільний запит, який ще не дає інформації щодо його побажань системі. Система не вміє зчитувати цілісні повідомлення, але може працювати з масивами даних, що містять побажання групи користувачів, схожих за інтересами.

Для написання такої програми необхідно застосувати технологію Word2Vec або Doc2Ver, яка вміє розбивати файли або речення на прості слова, перетворюючи їх у векторну форму, а потім сортувати їх до масивів за схожими параметрами.

З двох технологій для вирішення задачі обрано Doc2Ver, тому що вона вміє працювати з цілими файлами, перетворюючи їх зміст у набор чисел (вектор). Для реалізації також застосовано мову Python.

Маючи готову модель, можна отримувати векторне уявлення як слів і пропозицій, так і цілих документів. При порівнянні тексту запиту і пропозиції це використовується наступним чином:

1. При відсутності в тексті запиту явної вказівки наявних побажань, вони додаються в асоціативний масив, якщо такі були виділені при порівнянні тексту запиту з текстом пропозиції.

2. Обробка попереднього досвіду кандидата та виділення з нього характеристик, що підходять певному туру.

3. Проходження по тексту запиту, представленому у вільній формі, здійснюючи пошук зазначених в турі умов.

Це реалізовано завдяки можливості порівняння між собою як речень, так і слів з реченнями та документом з запитами. На виході ми отримуємо 2 векторних уявлення — уявлення запитів та уявлення турів, об’єднаних у .JSON файлі. Це документ з відповідностями слів по типу класифікації.

Після того, як зібраний список ключових відповідностей для кожної категорії, можна розбити його на три класи:

– загальні для всіх категорій запити (комфорт, швидкість транспорту, інфраструктура);

– загальні для кластера запити;

– особливі запити кожного напрямку (море, гори, місцевий колорит тощо).

Робиться це наступним чином:

– власноруч створюємо список з напрямками;

– прочитуються всі файли з турами та кожен елемент «tours» перевіряється на наявність інформації в полі з ключовими вимогами і, якщо вони там є, то визначається до якої країни та міста вони відносяться;

– отримується RDD, кількість елементів якого дорівнює турам, що складається з ідентифікатора цінової політики туру напрямку і списку умов;

– З RDD, отриманого на попередньому кроці, отримати список всіх турів для кожної з напрямків;

– Використовуючи join, отримати RDD, в яких будуть всі умови для даного напрямку;

– На підставі RDD з необхідними документами та умовами для туру відфільтрувати список усіх ключових побажань, використовуючи join. Внаслідок залишаться найбільш підходящі;

– повторюються дії минулого кроку, але для RDD з умовами туру та конкретного напрямку;

– встановлюється частота появи умов/потреб в документах.

Дана класифікація дозволяє ефективніше порівнювати тури і запити, виключаючи непотрібні навички та документи.

Далі за допомогою Apache Spark створюється DataFrame (таблиці) з класифікацією за критеріями — IndexesSearch и IndexesTours. Перша містить класифікацію запитів за основними критеріями, друга — відповідно класифікацію турів. Залишається їх порівняти та створити граф відповідності турів користувачам.

Apache Spark надає зручний інструмент для роботи з даними типу DataFrame. Це розподілена колекція, де дані зберігаються в іменовані стовпці як у базі даних.

Граф побудований за допомогою GraphFrame — пакету для Apache Spark для створення графів. Для побудови використовуємо два DataFrame, один з яких буде містити в собі вершини, а інший – ребра. Вершинами вданому випадку є люди, що подали заявки і тури, а зв'язки між ними визначаються тим, наскільки тур підходить людині, і навпаки.

На підставі списків переваги, отриманих на попередньому етапі, побудовано двудольний граф, описаний в теоретичному розділі. Вузлами графа є шукачі турів і тури, у кожного з яких є властивість – список бажаних вузлів з іншої безлічі.

Після того, як граф побудований, до нього застосовано алгоритм ранжування PageRank. Він встановлює відповідність за допомогою пріорітетів важливості: чим більше посилань у стовпцю запитів користувача, тим найбільш точно тур йому підбирається. Результатом

роботи алгоритмує оптимальна паровідповідність, яка розподілила шукачів турів максимально ефективно для обох сторін.

Результати роботи можна вилучити у звичайних .CSV файлу з перевагами, звідки маркетологи можуть вручну вибрати людей, яким напряму можна запропонувати певний тур. Аудиторії відповідно до запитів можна вже відсортувати вручну або за допомогою формул у Microsoft Excel.

Схема функціональної структури комп'ютерної системи формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data представлена у додатку А.

3.4 Висновки по розділу

Таргетована реклама працює і без парсингу, але при тих же витратах конверсія буде нижчою. Це відбувається через те, що параметри налаштування рекламного кабінету можуть передбачити багато, але не все. Жоден таргет не гарантує на 100%, що профілі, які підходять під кампанію за критеріями, — потенційні клієнти.

Тому основна мета системи формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data – забезпечити безпомилкове потрапляння в цільову аудиторію.

Парсинг, що лежить в основі проектованої комп'ютерної системи – безцінний інструмент маркетолога. Збирає та вивантажує необхідну спеціалісту інформацію з цільової аудиторії, за рахунок чого підвищує ефективність кампанії.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ

4.1 Призначення та сфера застосування програми

Ідентифікація користувача є основним призначенням комп'ютерної системи, в різних соціальних мережах дає змогу отримати більш повну картину про соціальну поведінку конкретного користувача в мережі Інтернет. Оскільки пошук акаунтів користувача в різних мережах в загальному випадку вимагає наявності актуальних даних про всіх користувачів даних мереж, доцільно обмежити простір пошуку найближчими сусідами будь-якого користувача, акаунти якого в досліджуваних мережах відомі.

Так виникає завдання ідентифікації користувачів в різних соціальних мережах, в локальній перспективі мається на увазі зіставлення акаунтів користувачів в рамках списків контактів деякого центрального користувача в різних соціальних мережах. Таке завдання часто виникає при роботі з контактами користувачів в соціальних метасервісах, які, зокрема, можуть служити для об'єднання новинних потоків в підтримуваних соціальних сервісах або надання єдиної системи обміну повідомленнями. Подібна задача виникає також при використанні функції автоматичного об'єднання контактів зрізних джерел (телефонна книга, соціальні мережі, месенджери), поширеною в сучасних мобільних пристроях.

4.2 Обґрунтування технічних характеристик програми

Завдання до роботи комп'ютерної системи можна поставити так: знайти максимальне паросполучення в дводольному графі, тобто вибрати найбільшу кількість ребер, таких що жодне з них не мало б спільної вершини з іншим ребром.

Для вирішення цього завдання можна використовувати ітеративний алгоритм Гейла-Шеплі. Спочатку необхідно визначити спосіб ранжирування елементів з однієї безлічі. Потім для кожної вершини формується список бажаних елементів іншого безлічі. Після цього починає роботу безпосередньо алгоритм, кожна ітерація включає в себе стадію пропозиції, згоди і відмови:

1. Кожен непов'язаний ні з ким елемент з безлічі пошуковців пропонує свою кандидатуру, на основі своїх інтересів, найбільш відповідних турагентству, до якого він ще не звертався (незалежно від того, чи пов'язане турагентство з кимось з пошуковців чи ні).

2. Кожен клієнт відповідає згодою на тур, який найбільше бажає (причому такий кандидат може бути обраний на попередньої ітерації) і відхиляє пропозиції інших. Складність такого алгоритму $O(n^2)$, де n – число турагентств або пошуковців.

Число турагентств та пошуковців не дорівнює один одному. Крім того, внаслідок роботи алгоритму може вийти так, що на один і той самий тур претендують два кандидати з однаковою оцінкою. Через це може вийти так, що не всі вершини однієї з множин не будуть зв'язані з іншими.

Алгоритм Гейла-Шеплі гарантує, що кількість пар буде, як мінімум, дорівнює половині розміру оптимального пар. Для того щоб знайти якомога більше пар-відповідностей, можна модифікувати алгоритм:

1. Вершини, які вважає за краще пошуковець, будуть пройдені двічі.

2. Турагент буде віддавати перевагу незайнятої позиції зайнятій, якщо він має до них однаковий інтерес.

3. Якщо пошуковець прийняв пропозицію, у якого є більш бажаний тур, він буде далі приймати нові пропозиції і, в подальшому, може від нього відмовитися. В цьому випадку претендент не прибере цей тур зі свого списку.

Використовуючи ці модифікації, алгоритм гарантує, як мінімум, 2/3

від розміру оптимального паросполучення. Виходячи з написаного вище, слідує, що для роботи алгоритму потрібно створити RDD, в якому будуть зберігатися список бажаних турів, а також статус пошуковця. Це ж треба зробити для турагентств.

RDD (Resilient Distributed Dataset) – це проста, незмінна, розподілена колекція об'єктів у фреймворку Apache Spark. RDD є розподіленим набором даних, який ділиться на безліч частин, що обробляються різними вузлами в кластері. Набори RDD можуть містити об'єкти з будь-якими типами даних на мовах JAVA, Scala або Python. Останній ми і будемо використовувати.

Так як алгоритму не важливий порядок, в якому турагентства пропонують свої кандидатури або пошуковці вибирають найкращий тур, він може виконуватися в два етапи: запиту і відповіді, що спрощує його запуск в Apache Spark.

Етапи роботи алгоритму Гейла-Шеплі:

1. Всі шукачі відправляють свої пропозиції на найбільш бажані тури, а турагентства обробляють їх, вибираючи найбільш бажане і змінюючи свій статус.

2. Клієнти відправляють свої рішення, а здобувачі, якщо потрібно, вибирають з них найбільш детально визначений і змінюють свій статус.

Для того щоб додати сюди модифікації, описані вище, непотрібно міняти загальну схему роботи алгоритму. Зміни вносяться в те, як оновлюються статуси учасників, і як вони роблять пропозицію і відповідають на нього. Так, ще не обравши тур пошуковець відправляє повідомлення всіх елементів з його списку, сповіщаючи їх про те, що є такий тур. У свою чергу, пошуковець відправляє спеціальний запит на тур, який він вважає за кращий. Дані зміни не несуть великий обчислювального навантаження, тому складність алгоритму залишається такою ж.

4.3 Опис розробленої програми

Було вирішено зібрати дані з сайту та соціальних мереж клієнта. Для цього буде використовуватись парсер, який буде збирати всі доступні в даний момент запити і зберігати їх у форматі JSON.

Для кожного запиту туру існують наступні поля:

- about – інформація про тур у вільній формі;
- conditions – список ключових умов (отель, море, кількість осіб);
- cost – бажана ціна;
- position_type – формат (сімейний, екстрим, комфорт та ін.);
- position_destination – бажаний напрямок;
- position_schedule – бажаний графік відпочинку;
- position_time – бажаний час вильоту;
- personal – кількість осіб, місто;
- kids_age – вік дітей;
- cit_cit – громадянство;
- cit_visa – наявність візи;
- country – бажана країна;
- city – бажане місто;
- link – посилання на запит.

Тури для пропозицій брались з сайту клієнта. API-вакансії представлені в форматі JSON, який має наступні поля:

- опис;
- країна;
- місто;
- умови (готель, переліт, умови для дітей та ін.);
- вартість;
- необхідність візи.

Виходячи з формату отриманої інформації, були обрані методи її обробки. Для таких полів, як «країна», «місто», «вартість», «громадянство» додаткової обробки не потрібно, тому що вони зберігаються в стандартному вигляді. Поля з описом туру, умовами, побажаннями були розбиті таким чином.

Спочатку вони були розбиті на пропозиції і збережені у вигляді списків, де кожен елемент – окрема пропозиція. Потім в кожному елементі (Пропозиції) була видалена пунктуація, він був розбитий на окремі слова. Після був застосований стемінг. До ключових побажань також був застосований стемінг, і вони були збережені як окремий список.

Файли з запитами були оброблені тим же способом, що і пропозиції, за винятком поля з візою. Якщо воно не порожнє, то з нього витягуються країни, куди додатково може поїхати ця людина і дані щодо візи – коли отримана і якого типу вона є.

4.4 Опис логічної структури

Зв'язок пошуковця і турагентства головним чином визначається наявністю або відсутністю потрібних умов. Виходячи зі структури даних, представленої вище, для них є окреме поле, як у вакансіях, так і в резюме. Таким чином, можна перевірити наявність у пропозиції критеріїв, які потрібні для відповідності даному пошуковцю, після чого поставити йому оцінку і, якщо вона вище порогової, додати в список умов для подальшого розгляду. Однак досить часто клієнти залишають таке поле порожнім, а всі необхідні навички знаходяться в описі або в поле з умовами, причому як частина пропозиції, а не списком. Те ж саме відноситься до запитів. Це сильно ускладнює завдання, так як для рішення необхідно застосовувати методи NLP, зокрема Text Mining.

Для того щоб вирішити цю проблему, можна використовувати векторне представлення слів. Його ідея полягає в тому, щоб зіставити

словам і фразам елементи векторного простору розмірності n, де n значно менше кількості унікальних слів в даному корпусі. Для побудови такого уявлення існує кілька підходів.

Отримуючи на вході велику колекцію документів, в даному випадку файли з текстом запитів і турів, Word2Vec повертає подання унікальних слів цього корпусу в векторному просторі. Ці вектори розташовуються в ньому таким чином, що слова, що з'явилося в схожому контексті, знаходяться в безпосередній близькості. така особливість дозволяє порівнювати пропозиції і терміни, використовуючи зручні метрики.

4.4.1 Використання Word2Vec та Doc2Vec

Word2Vec – це набір моделей, які приймають на вхід текст і які отримують в результаті роботи уявлення слів у векторному просторі на основі контексту. Ці моделі (ContinousBagOfWordsiskipgram) представляють собою нейронну мережу, завданням якої є реконструкція контексту слів. Так, завдання CBOW – пророкування слова на підставі контексту, а завдання skipgram – передбачити контекст на основі єдиного слова. Їх архітектуру можна подивитися на малюнку (рис 4.1) [15].

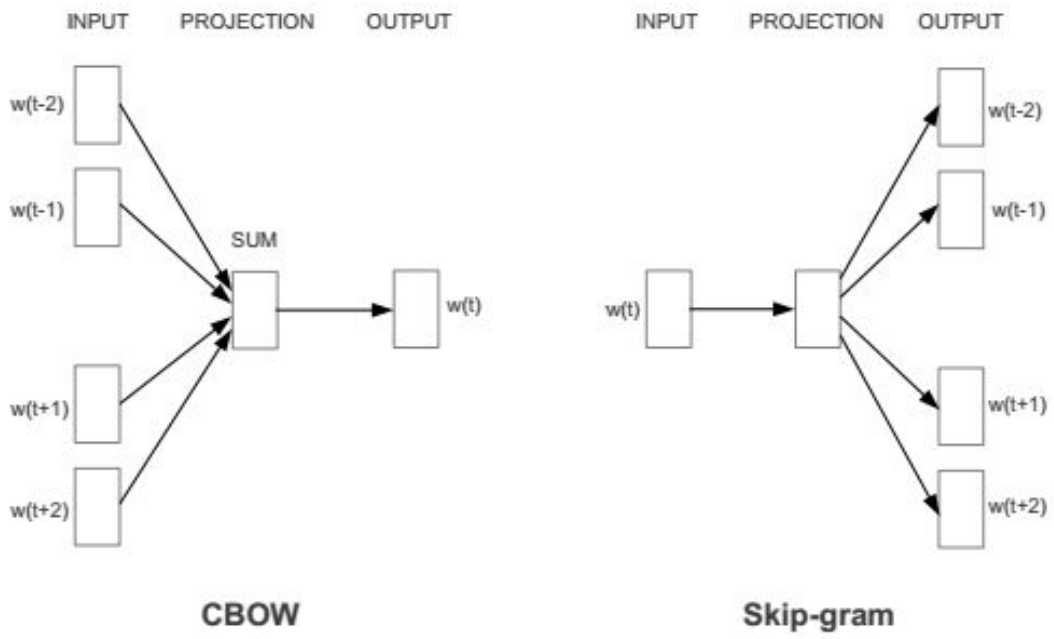


Рисунок 4.1 – Архітектура CBOWi skip-gram

Розмір векторного простору Rn задається вручну, зазвичай, n знаходиться в діапазоні від 100 до 400. Таким чином, даний метод має незаперечну перевагу у вигляді невеликої розмірності векторів. На відміну, наприклад, від методів, які працюють зі словниками, де розмірність може досягати декількох тисяч.

Принцип роботи Word2Vec можна описати таким чином: максимізація косинусної близькості для векторного уявлення слів, які з'являються в схожих контекстах, і, навпаки, її мінімізація для слів, які не зустрічаються в схожих контекстах.

Після того, як векторні уявлення отримані, з'являється можливість, наприклад, знаходити близькість між двома словами, отримувати список найбільш близьких елементів у векторному просторі і так далі. Крім того, можна отримувати вектори для цілих пропозицій, використовуючи, наприклад, усереднений вектор всіх слів в ньому. Однак даний підхід ігнорує порядок слів. Тому для роботи з пропозиціями, параграфами або цілими документами, слід використовувати Doc2Ver.

На відміну від Word2Vec, Doc2Ver використовує дві моделі: Distributed Memory і DistributedBagOfWords. Distributed Memory зіставляє кожному параграфу або пропозицією вектор, який містить в собі вектор цього параграфа або пропозиції, а також всі слова, що містяться в ньому. Таким чином, цей метод передбачає слово на підставі вектора параграфа, що дозволяє врахувати порядок слів, і по попереднім словам. DBOW, в свою чергу, передбачає появу випадкових слів в параграфі тільки на підставі вектора параграфа. архітектура обох моделей показана на малюнку (рис 4.2 і рис 4.3).

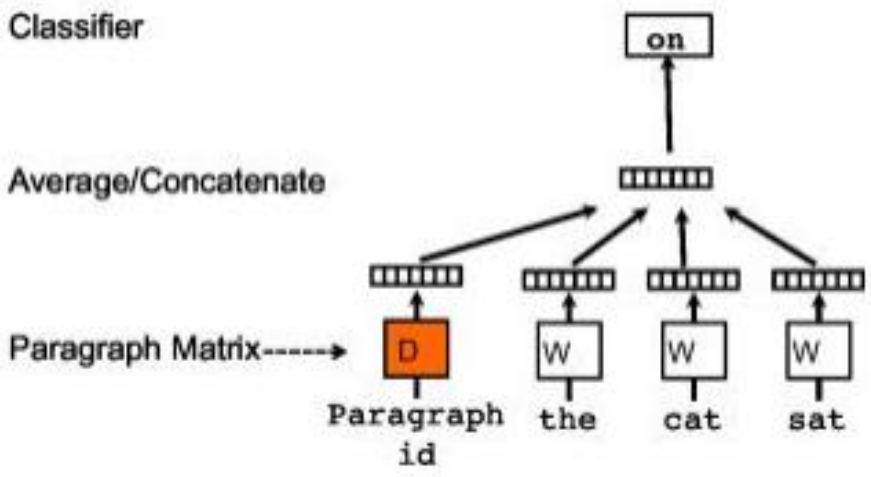


Рисунок 4.2 – Архітектура DM

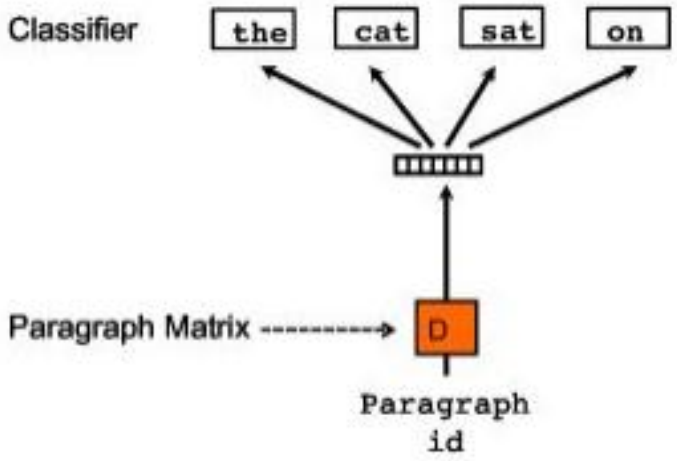


Рисунок 4.3 – Архітектура DBOW

Виходячи з вищевказаної інформації, для даної задачі Doc2Vec підходить набагато краще, тому що слова (вимоги до турів) доведеться порівнювати з пропозиціями (існуючими турами) або абзацами (Опис туру).

Для того щоб використовувати Doc2Vec, можна взяти модель, навчену, наприклад, на корпусі Вікіпедії, або ж навчити її самому. Недолік вже готової моделі в тому, що вона може бути занадто загальною і, відповідно, вважати непотрібні слова близькими один до одного. Навчання на спеціальному корпусі і навчання на ньому з додаванням сторінок Вікіпедії не дало відчутної різниці. Таким чином, вихідним корпусом можна вважати всі зібрані дані про пошук і існуючі тури. В якості вхідних даних Doc2Vec може прийняти як пропозиції (Одна строчка вхідного файлу – пропозиція), так і цілі документи (один рядок вхідного файлу – документ). Після навчання моделі можна отримати векторне подання для слова, пропозиції або документа,яких не було в тренувальному корпусі. Однак при досить великій кількості нових турів і запитів, її все ж таки варто перенавчити.

Таким чином, маючи готову модель, можна отримувати векторне уявлення як слів і пропозицій, так і цілих документів. При порівнянні тексту запиту і пропозиції це може використовуватися наступним чином:

1. При відсутності в тексті запиту явної вказівки наявних побажань, можна додавати їх в асоціативний масив, якщо такі були виділені при порівнянні тексту запиту з текстом пропозиції.

2. Обробка попереднього досвіду кандидата та виділення з нього характеристик, що підходять певному туру.

3. Проходження по тексту запиту, представленому у вільній формі, здійснюючи пошук зазначених в турі умов.

Все це може бути реалізовано завдяки можливості порівняння між собою як речень, так і слів з реченнями або документом [15].

4.4.2 Робота з даними в ApacheSpark

Apache Spark надає зручні інструменти для роботи з даними такого типу як DataFrame. Це розподілена колекція, де дані зберігаються в іменовані стовпці, тому робота з Apache Spark DataFrame схожа на роботу з таблицями в реляційних базах даних [16].

Для того, щоб створити DataFrame з JSON, необхідно:

```
From pyspark.sql import SQLContext
sqlContext = SQLContext(sc)
tours = sqlContext.read.json("tours.json")
```

Варто відзначити, що на JSON файл накладаються певні обмеження, так, наприклад, спроба вважати файл, в якому значення одного з полів займає кілька рядків, швидше за все закінчиться невдачею.

Після того, як DataFrame створений, можна звертатися до його стовпців, наприклад, наступним чином:

```
tours .select("position_title").show() — вивести значення стовпці 'position_title'
tours .filter(tours[“position_title”] = “Dominikana”).show()
```

4.4.3 Класифікація документів і ключових запитів

Деякі запити написані іншою мовою (у мого клієнта – російська). Зробити систему, яка зможе порівнювати два документа на різних мовах дуже складне завдання. Однак у даній роботі документи, що містять певний відсоток мови відмінного від російської, будуть ігноруватися.

Для початку були створені два DataFrame IndexesSearch і IndexesTour, в яких будуть два стовпці: індекс і документ. Спочатку завантажуються всі тури в RDD, та застосували функція zipWithIndex. Ті ж самі дії проводяться для файлів з запитами.

Перед тим, як почати обробку тексту, необхідно зібрати всі доступні ключові запити та класифікувати їх, щоб прискорити подальшу роботу. Як приклад категорій можна взяти дані полядirection.travarea_name, які є у запитах і пропозиціях, причому, пропозиціям варто віддавати перевагу. Таких категорій існує 23 штуки на етап пандемії. Після того, як зібраний список ключових відповідностей для кожної категорії, можна розбити його на три класи:

– загальні для всіх категорій запити (комфорт, швидкість транспорту, інфраструктура);

– загальні для кластера запити;

– особливі запити кожного напрямку (море, гори, місцевий колорит тощо).

Вирішення цього завдання можна провести в кілька етапів:

– Власноруч створити список, в якому будуть міститися всі напрямки.

– Прочитати всі файли з турами.

– Кожен елемент tours перевірити на наявність інформації в поле з ключовими вимогами і, якщо вони там є, то визначити до якої країни та міста вони відносяться.

– Отримати RDD, кількість елементів якого дорівнює турам, що складається з ідентифікатора цінової політики туру напрямку і списку умов.

– З RDD, отриманого на попередньому кроці, отримати список всіх турів для кожної з напрямків.

– Використовуючи join, отримати RDD, в яких будуть всі умови для даного напрямку.

– На підставі RDD з необхідними документами та умовами для туру відфільтрувати список усіх ключових побажань, використовуючи join. Внаслідок залишаться найбільш підходящі.

– Повторити дії минулого кроку, але для RDD з умовами туру та конкретного напрямку.

– Знайти частоту появи умов/потреб в документах.

Дана класифікація дозволяє ефективніше порівнювати тури і запити, виключаючи непотрібні навички та документи.

Наступним етапом буде класифікація всіх документів по побажаннях та умовах. Для цього необхідно створити два нових DataFrame на основі IndexesSearch і IndexesTour. Перебираючи їх елементи, відбувається звернення до полів документа, і в нові стовпці додаються дані про необхідні та достатні умови. Ці два DataFrame називаються SearchDataPos і TourDataPos.

Таким чином, при подальшому аналізі документів обмежується безліч всіляких критеріїв, виключаючи такі, які характерні для інших напрямків окрім вказаних. Це дуже зручно, адже по-перше, можна запропонувати людині альтернативний напрямок, якщо бажаний не підходить за датами (наприклад), а, по-друге, це дає змогу аналізувати ринок туризму. [10]

4.5 Висновки по розділу

Система формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data була розроблена згідно з сучасними вимогами до реалізації програмного коду, використовуючи поширену мову програмування.

При розробці враховувалися всі сучасні тенденції, вимоги до парсингу даних:

Парсинг підписників, постів, лайків, коментарів, контактів адміністрації групи у Facebook із данними;

- Збір даних про новоприйнятих у групах;
- Вивантаження ID груп, на які підписано конкретного користувача;
- Конвертація імені користувача в його ID та навпаки;
- Отримання інформації про користувачів (стать, вік, день народження, геолокація);
- Пошук користувачів, спільнот, публічних сторінок по email/номеру телефону;
- Пошук цільової аудиторії;
- Статистика вибірки;
- Розширена інформація про користувача (сімейний стан, родичі, навчання);
- Парсинг партнерів/родичів;
- Пошук по хештегам;
- Збір з геолокації;
- Фільтр груп/публічних сторінок/подій/локацій за назвою;
- Збір позначок на фото.

Спираючись на дані факти можна стверджувати, що розроблене програмне забезпечення в своїй мірі сучасне і актуальне на сьогоднішній час.

5 ЕКСПЕРИМЕНТАЛЬНИЙ РОЗДІЛ

5.1 Формулювання завдання та обґрунтування методики

Завдання дослідження — проведення аналізу над великою кількістю не структурованих даних. Для вирішення цього питання використовуються можливості Apache Spark. Фреймворк Apache Spark ефективно застосувати, якщо враховувати у кожному залишеному запиті не тільки ключову інформацію, але й інформацію про пошуковця та ключові побажання.

Отриманий з цих даних граф можна аналізувати, взявши з нього наступну інформацію:

– найбільш поширені тури;

– тури, для покупки яких потрібні певні додаткові умови (наявність візи, наявність щеплень, наявність запрошення у країну, наявність попередніх подорожей);

– найбільш релевантні для пошуковця тури;

– різні характеристику ринку туристичних послуг та його зміни;

– виділення класів турів та пошуковців.

Граф $G = (W, E)$ є дводольним, якщо множину його вершин розбити на 2 неперетинаючі підмножини: $A \cup B = W, |A| > 0, |B| > 0$, до того ж кожне ребро має початок в A, а кінець в B.

Граф $G = (W, E)$ — орієнтований та дводольний, коли його неорієнтований двійник – дводольний граф.

У системі елементами безлічі вершин A будуть пошуковці, а елементи безлічі вершин B – турагентства.

Основним завданням є оптимальний розподіл турів між пошуковцями. Так як граф є дводольним, то можна сказати, що стоїть проблема про мар'яж (Stable marriage problem):

Елементи кожної безлічі ранжують елементи іншої безлічі числом від 1 до n в залежності від своїх уподобань. Потім пару пов'язують таким

чином, що немає елементів з іншої безлічі, які хотіли б мати партнера з цієї пари замість свого. Такі зв'язки називають стійкими (stable).

Структура для зберігання вилучених даних для заявок і турів має свої подібності та відмінності. Загальним буде наявність таких полів, як дата вильоту, напрям, певні умови, вартість, наявність умов для дітей. Їхня відповідність відразу говорить про те, чи підійдуть вони одне одному. Метою є виділення з полів важливу нам інформацію про додаткові побажання, наявність візи.

Для заявки особливо важливо аналізувати документи і побажання людини, які найчастіше вони залишають у досить вільній формі. Приклади можна знайти на малюнках (рис. 5.1 та рис. 5.2).

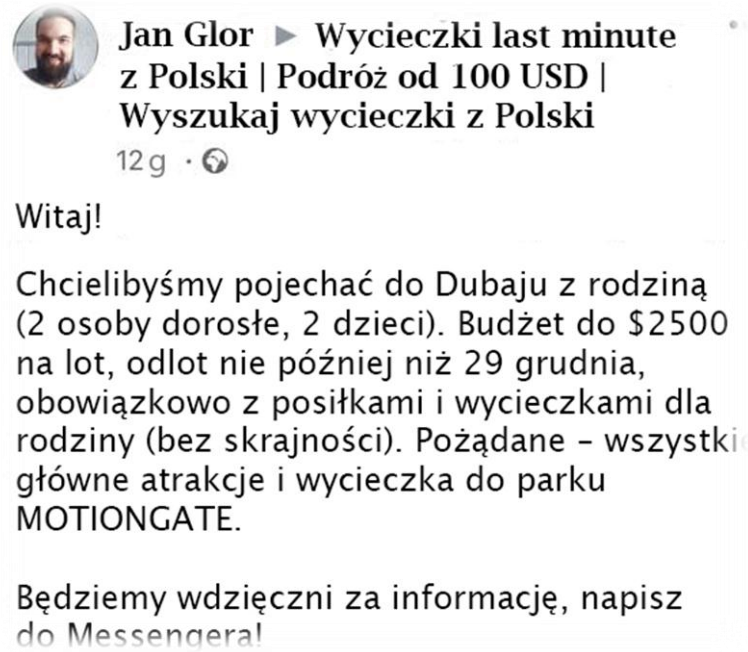


Рисунок 5.1 – Заявка на стіні Facebook у спільноті для пошуку турів з Польщі

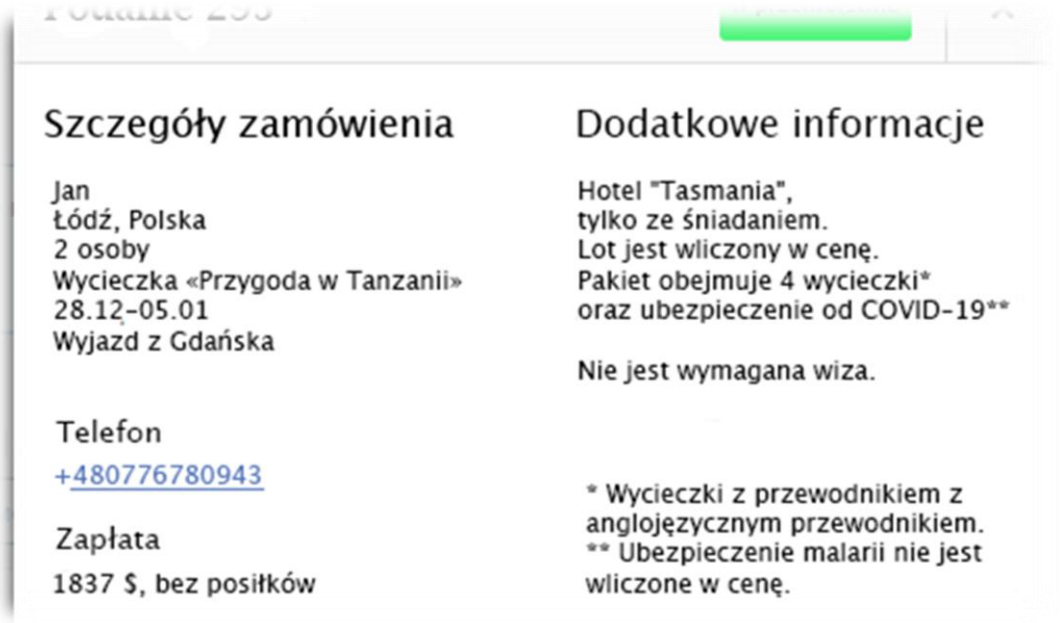


Рисунок 5.2 – Заявка на сайті клієнта

Корисною інформацією тут є назва напрямку, бажані дати, бажана ціна, кількість людей, а також опис. Варто зазначити, що для поля з описом немає яких-небудь стандартів, текст в ньому представлений у вільній формі, тому для його аналізу потрібно використовувати векторне уявлення.
Поле «додаткова інформація» також може містити інформацію про візу і побажання. Воно не має стандартів, тому для його аналізу також буде використана модель Doc2Vec.

Витяг інформації з документа проводиться в кілька кроків під час проходу по IndexesSearch: Здійснюється доступ до рядка в DataFrame і перетворення її в RDD, де один елемент – масив, що складається з індексу заявки, дати розміщення, громадянство, напрямку, інформації про місце вильоту, про ціну, про вік дітей, про «складність поїздки» та інші, а також масиву з даними щодо візи – елементами якого є масиви, що зберігають у собі інформацію про тип візи (робоча, студентська, Шенген, японська та ін.), дату початку та закінчення, особливі примітки і опис у вигляді списку пропозицій, а також масиву з полем про побажання – тобто з елементами

вільного опису та умов, які вдалося дістати з інформації про візу та умов, що вдалося визначити з побажань.

Варто відзначити, що елементи не потрібно ніяк обробляти,вони зберігаються в початковому вигляді. До даних, зазначених у спеціальному полі, був заздалегідь застосований стемінг. Елементи, що ми отримали з вільного опису отримані в результаті поєднання інформації з декількох полів JSON файлу. Елементи, що ми дістали з інформації про візу та про умови виходять в результаті застосування функції, в основі якої лежить знаходження косинусної подібності між цією пропозицією і всіма запитами, характерними для даного напрямку.

Аналіз турів здійснюється за тією ж самою схемою, що й аналіз запитів. Єдиною відмінністю будуть поля, які ми діставатимемо у результаті роботи:

- індекс туру на сайті;
- можливі дати вильоту;
- можливі часи вильоту;
- необхідність візи;
- умови для дітей;
- особливі умови;
- тип розміщення;
- вартість;
- список послуг, що входять у тур;
- список необхідних документів;
- послуги, що вдалось витягнути з опису туру;
- послуги, що вдалось витягнути з вимог до документів.

5.2 Вимоги до експерименту

Для подальшого проведення експерименту слід ввести наступні вимоги:

– використати розроблену комп’ютерну систему формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data;

– отримати спарсені дані про користовучів у файлі типу .JSON з даними «користувач — запит»;

– порівняти тури і запити, виключаючи непотрібні напрямки;

– побудувати граф, вершинами котрого є люди, що подали заявки і тури, а зв'язки між ними визначаються тим, наскільки тур підходить людині, і навпаки;

– результат роботи вилучити у .CSV файл з перевагами, звідки маркетологи зможуть вручну вибрати людей, яким по напряму можна запропонувати конкретний тур.

Визначення того, що тур підходить даному пошуковцю здійснюється в кілька етапів. Спочатку всередині кожного напрямку відбувається порівняння кожного бажаючого з кожним туром за наступними полями: країна, місце, вартість, дати вильоту, діти. Якщо хоча б по одному з цих полів виявлено розбіжність, то ця пара більше не розглядається. Якщо ж все поля задовольняють один одному, то індекс даного тура додається в RDD, що складається з індексу заявки пошуковця та відповідних турів. Такий же RDD формується для кожного туру.

Наступним етапом буде проходження по кожному елементу списку в RDD і визначення оцінки того, наскільки тур підходить пошуковцю, і навпаки. Якщо ця оцінка переходить певний поріг, то тур або пошуковець додаються в список бажаних зв'язків, який буде далі використаний для знаходження оптимального паросполучення.

Оцінку і поріг можна змінювати в залежності від того, що важливіше клієнту. У даній роботі всі знайдені поля, незалежно від того, як вони вказані в документі, враховувалися однаково. Оцінка виставлялася як відношення розміру безлічі перетину критеріїв запитів і турів до розміру безлічі об'єднання пропозицій турів і запиту. За поріг взято число 0.8.

5.3 Результати експерименту

Для отримання результатів потрібно використовуємо фрейморк з графами Apache Spark, що надає бібліотеку GraphX. Вона дозволяє створювати і працювати з орієнтованими мультиграфами, в яких кожній вершині або кожному ребру може бути поставлена у відповідність якась властивість.

Плюсами цієї бібліотеки є:

– обчислення над графом виконуються паралельно, розподілені між вузлами кластера;

– зручність роботи;

– можливість розглядати дані як граф і як колекції;

– виконання операцій над графами з тією ж ефективністю, що і над RDD;

– готові алгоритми.

Мінусами, в свою чергу, можна назвати:

– неможливість динамічно оновлювати граф, додавати або видаляти вершини і ребра;

– граф існує, поки він завантажений в пам'ять.

GraphFrames є пакетом для Spark, який, на відміну від GraphX, дозволяє будувати графи на основі DataFrame, а не RDD. Можна сказати, що він розширює можливості GraphX, роблячи доступними серіалізацію, засновану на DataFrame, а також виразні запити до графу. Однак, його

мінусом, як і в GraphX, залишається неможливість динамічно оновлювати граф. Неможливість динамічно оновлювати граф компенсується тим, що операція його створення з DataFrame вимагає досить небагато ресурсів.

В результаті порівняння цих бібліотек була обрана бібліотека GraphX, тому що її можливостей вистачає для вирішення даного завдання. До того ж вона має багатшу документацію.

Для того, щоб створити граф, потрібні два DataFrame, один з яких буде містити в собі вершини, а інший – ребра. Вершинами вданому випадку є люди, що подали заявки і тури, а зв'язки між ними визначаються тим, наскільки тур підходить людині, і навпаки.

На підставі списків переваги, отриманих на попередньому етапі, можна побудувати двочастковий граф, описаний в розділі 2. Для цього використовується бібліотека GraphX. Вузлами графа є шукачі турів і тури, у кожного з яких є властивість – список бажаних вузлів з іншої безлічі.

Після того, як граф побудований, до нього застосовується алгоритм, також описаний в розділі 2. Результатом його роботи є оптимальна паровідповідність, яка повинна розподілити шукачів турів максимально ефективно для обох сторін.

Завдяки тому, що раніше були побудовані списки переваги, маркетологи можуть вручну вибрати людей, яким напряму можна запропонувати певний тур (рис 5.3).

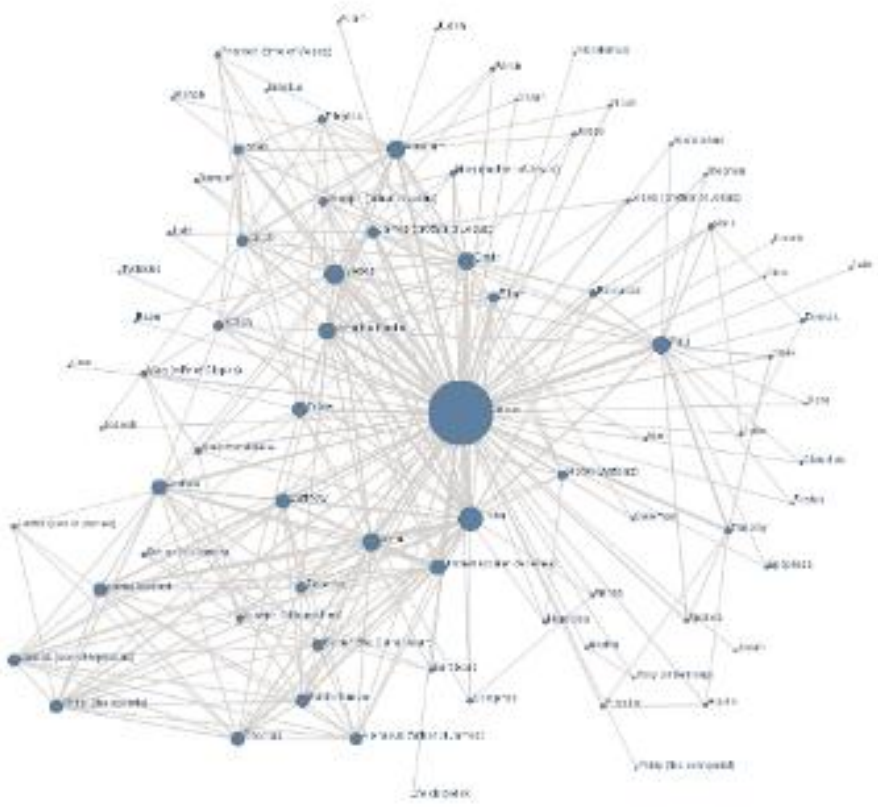


Рисунок 5.3 – Частина соціального графу відповідностей запитів та напрямків

Ось лише невеликий фрагмент відповідностей, які ми вилучили за допомогою розробленого алгоритму. Імена людей закриті зірками тому що заборонено оголошувати персональні дані користувачів.

Номер візи не вказано за причиною дотримання конфіденційності.

Таблиця 5.1 – Частковий результат підбору турів кожному типу.

Хто	Jan Koupers	Lara Ganczarek	Katarzyna Munka	Laura Mączka	Łukasz Kowalski	Justyna Kowalski	Katarzyna Czyszczoń
Дата	Після 23.01	До 29.12	Після 17.12	На 14.02	Виліт 29.12	Виліт 28.12	До 15.12

Термін	2 тижні	1 тижні	2 тижні	1 тижд.	10 днів	10 днів	3 тижні
Час	зранку	вночі	вночі	день	день	день	вночі
Вартість	Від $2000	Від $1000 на людину	До $3000	$1500	$800 на особу	До $2000 на двох	До $3000 на трьох
Харчування	+	+	Не позн	+	-	-	+
К-сть осіб	3	4	3	3	2	2	3
Діти	1 – 8 років	2 – 10 та 6 років	10 років	12 років	-	-	2 роки
Країна	Єгипет	Турція	Танзанія	Туніс	Вільний напрям	ОАЕ	Танзанія
Місто	Шарм-Ель-Шейх	Каппадокія	?	?	Вільний напрям	Дубай	?
Громадянство	PL	PL	UA, карта поляка	PL	CH	PL	UA, карта поляка
Віза	(USV) від 05.10.2018	немає	(USV) від 16.10.2019	(USV) від 02.12.2017	немає	немає	(USV) від 31.10.2019
Додаткова інф-я	Бажано у першій береговій лінії. Необхідні умови для дітей. 3-раз. харчування	Каппадокія, у пріоритеті – потрапити на польоти повітряних куль. На сім’ю 4 людини, з	Готель люкс, екскурсії, наявність ресторану зі шведським столом, переліт економ клас. Без	Пакетний тур «все включно» з перелітом на 2х. Бажано, виліт вдень туди та	Без харчування. Бажано гарна інфраструктура, але недалеко від берегової лінії (можна доїхати місцевим	З проживанням у самому Дубаї, бажано близько до центру, виліт з Варшави вдень. Харчування та екскурсі	Готель люкс, умови для немовля, харчування або ресторан на території. Екскурсії для

	вкл. Екскурії та дайвінг.	харчуванням 2 рази.	екскурсій.	зворотній напрям	транспортом)	ї не потрібні, але бажано каршерінг.	всієї сім'ї.
Тур	«Подорож до країни сфінксів». Готель Oonas Dive Club Hotel. Код туру – EG02	«Каппадокія для сім'ї» - 28.12-5.01, Проживання у Cappa Villa. Код туру – KP09.	«Африканські пригоди на острові Танзанія». Проживання у The Residence Zanzibar. Код туру – ZN03	«Все включно у Тунісі». Le Royal All inclusive. TU02	«Незвичайна Африка – подорожу Туніс». El Mouradi Hammamet All inclusive. У 10 хвилинах пішки – ТРЦ. TU05	«Арабська ніч». Готель Atlantis The Palm». Код туру DU04	«Африканські пригоди на острові Танзанія». Проживання у The Residence Zanzibar. Код туру – ZN04

5.3.1 Сутність експерименту

Було порівняно 2 рекламні кампанії: одна проводилась на «холодну» аудиторію, тобто аудиторію, яка в теорії цікавиться турами, але не факт, що шукає їх у цей самий час, та на «теплу» – визначену програмно. Перша аудиторія була зібрана класичним способом Facebook – за поведінкою та інтересами користувачів (за допомогою інформації, яку ми «добровільно» залишаємо, мандруючи інтернет-ресурсами). Друга – програмно, за допомогою ApacheSpark.

Друга аудиторія допомогла нам продати 39 турів, коли перша – усього 11. І це під час пандемії, коли Європа, по суті, стоїть на величезній паузі.

5.3.2 Сутність експерименту у фактах

Побачити порівняння продаж, а також – отриманий дохід у злотих можна нижче (рис. 5.4 та рис. 5.5).

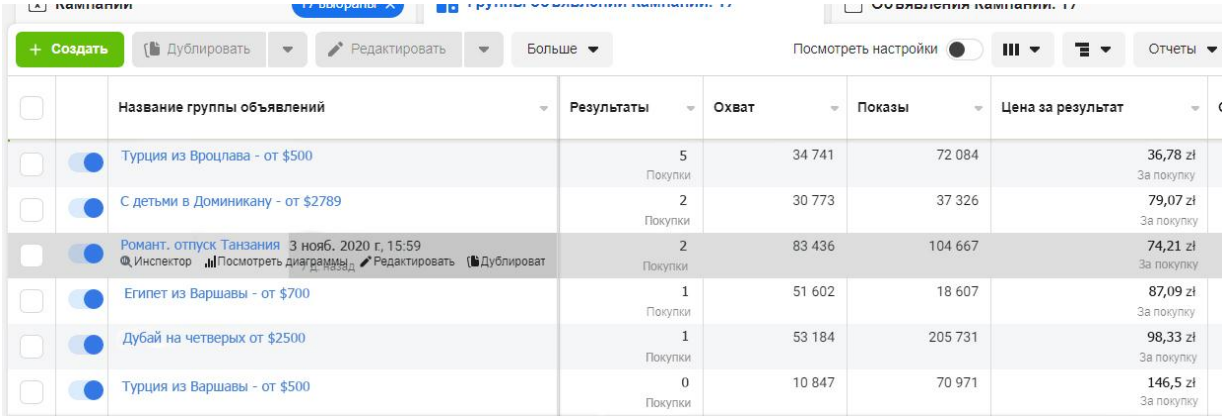


Рисунок 5.4 – Продажі на «холодну» аудиторію

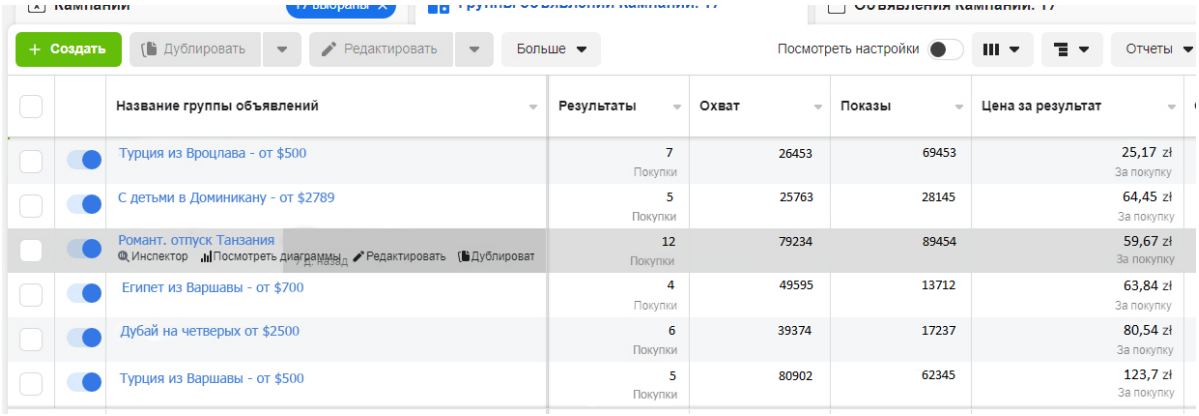


Рисунок 5.5 – Продажі на «теплу» аудиторію

5.3.3 Аналіз відповідності досліджень.

Чим заснована така відмінність у показниках? Справа в тому, що холодна аудиторія будується на основі дій користувачів – ресурсів, які вони переглядали, їх дій (часто мандрують, часто роблять онлайн замовлення, скоро мають день народження, недавно одружилися та ін.). Однак, переглядати тури може людина, яка не має у найближчий час можливості

полетіти і планує поїздку через півроку. А до «часто мандрують» можуть відноситися і ті, хто часто їздять у командировки.

Зібрана за допомогою алгоритму векторного уявлення аудиторія є максимально точною: ми знаємо, хто ці люди, куди і коли вони хочуть полетіти, на який бюджет розраховують, чи є в них діти, який формат відпочинку вони планують. Тому ми можемо зробити максимально влучне оголошення – наприклад, тому, хто шукає відпочинок з дітьми у Дубаї, показати дитячу зону парку розваг, наявність кімнати для дітей у готелі, зробити акцент на корисній їжі та двохкімнатних готелях. Тому, хто хоче екстремальний романтичний відпочинок удвох ми пропонуємо Танзанію: сафарі, квадроцикли у пустелі, закритий власний пляж, ніч у пустелі з розповіддю про зоряне небо.

Таким чином ми найбільш точно потрапляємо у бажання потенційних клієнтів і легко можемо запропонувати ідеальний варіант, базуючись на тих даних, що він залишив на сайті або будь-де у соціальних мережах.

5.3.4 Характеристика новизни результатів

Наукове значення експерименту, що провівся з використанням розробленої комп’ютерної системи формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data, полягає у обґрунтуванні доцільності моделі та її використанні для проведення експерименту.

Практичне значення результатів експерименту полягає у значущості досліджень для практики та можливі шляхи використання результатів у цифровому маркетингу.

5.4 Висновки по розділу

При початку проведення експерименту було встановлено його мету, завдання, та обрано методику проведення експерименту. Експеримент

проводився за заздалегідь сформульованими та визначеними вимогами, які відповідають суті експерименту. Проведений експеримент був вдалим, а дії які виконувалися у впродовж експерименту були докладно та чітко описаними, з кінцевим результатом. Результатом експерименту вважається виявлення усіх необхідних зв’язків між клієнтами та турами, за допомогою існуючих технологій створення динамічного креативу і BigData змогли підібрати найбільш цільові оголошення.

ВИСНОВКИ

На основі отриманих результатів можна заявити, що метод знаходження оптимальних зв'язків між шукачами і турагентствами, розглянутий у цій роботі, досить добре справляється зі своїм завданням – ми змогли продати 39 турів в Польщі під час пандемії. До використання алгоритму за такий самий проміжок часу вдалось продати лише 11.

У разі, коли є велика кількість текстових даних без будь-якої навчальної вибірки, використання векторного уявлення для слів і пропозицій має незаперечну перевагу. З його допомогою можна, не проводячи перед цим складну обробку даних, розбити документи на заздалегідь певні класи.

Точність даної моделі є допустимою для вирішення розглянутого завдання. Було показано, що алгоритм для знаходження безлічі оптимальних відповідностей пар, може застосовуватися в розглянутої області. Коригування потребує формула оцінки важливості витягнутих критеріїв залежно від галузі, у якій використовується даний алгоритм.

Реалізація описаних вище алгоритмів з використанням Apache Spark є робочою, але вимагає оптимізації. Зокрема, при витягу корисної інформації з документів, використовується занадто багато проміжних змінних. Для оптимізації використання пам'яті потрібно позначати які з них повинні залишатися в пам'яті, а які можна видалити.

Таку методику можна застосовувати у соціальних мережах, щоб запропонувати найбільш релевантну послугу чи продукт пошуковцеві. Метод векторного уявлення допоміг визначити, хто саме який тур шукає і встановити зв’язки між оферами клієнта та пошуковцями. Виявивши всі необхідні нам зв’язки, ми за допомогою вже існуючої технології створення динамічного креативу і BigData змогли підібрати найбільш цільові оголошення. Результат рекламної кампанії також був представлений у дипломній роботі.

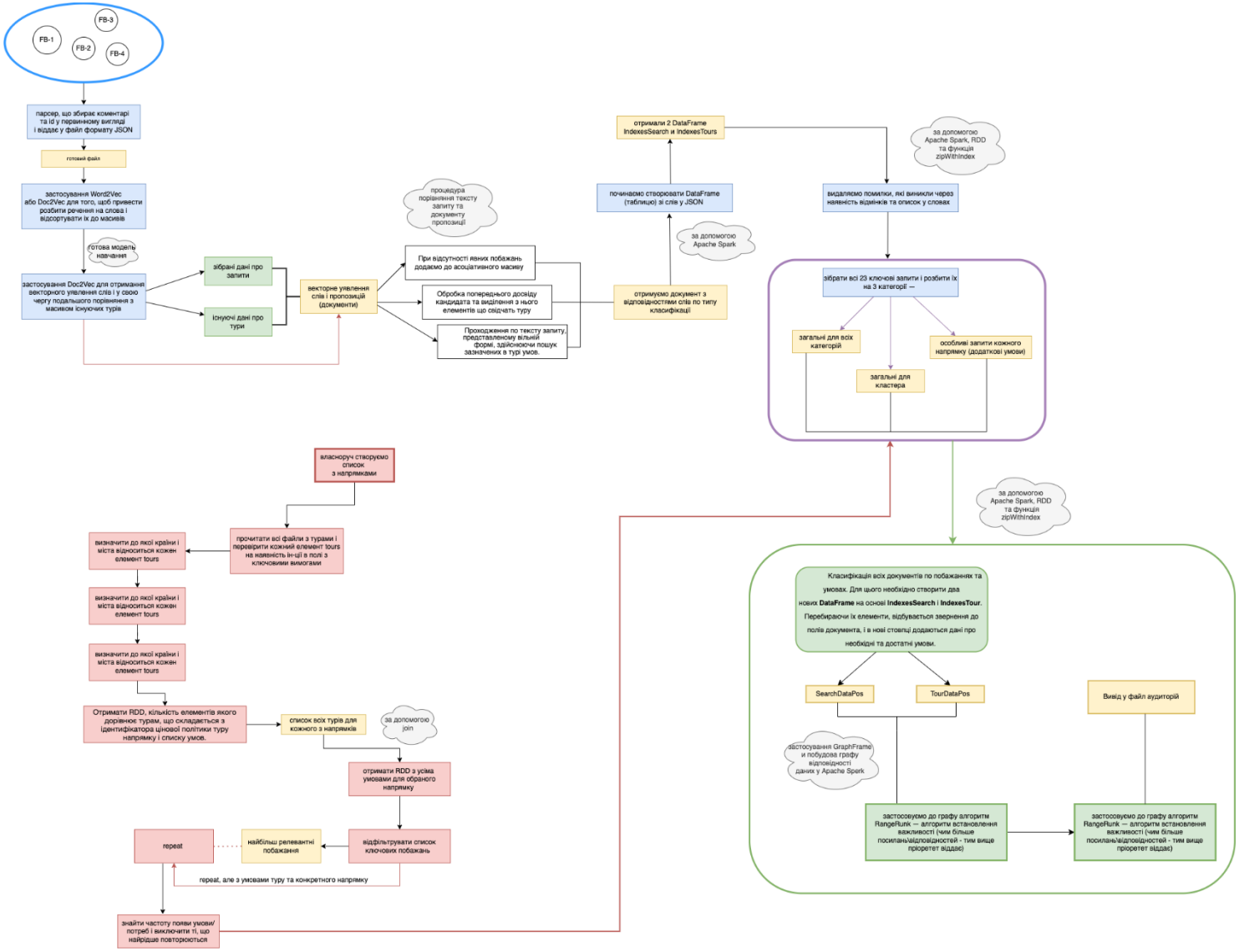
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Барсегян А. А. Технологии анализа данных: DataMining, VisualMining, TextMining, OLAP: Учебн. пос. / А. А. Барсегян. – С. Пб. : BHV, 2007. – 384 с.
2. Брянцев И. Н. Data Mining. Теория и практика / И. Н. Брянцев. – М. : БДЦ-Пресс, 2006. – 208 с.
3. Филатов В. А. Гибридные нейро-фаззи модели и мультиагентные технологии в сложных системах / В. А. Филатов, Е. В. Бодянский, В. Е. Кучеренко и др. // Системні технології, 2008. – С. 23–46.
4. Джонс М. Т. Программирование искусственного интеллекта в приложениях / М. Т. Джонс / Пер. с англ. А. И. Осипов. –М. : ДМК Пресс, 2004. – 312 с.
5. Круглов В. В. Искуственные нейронные сети. Теория и практика / В. В. Круглов, В. В. Борисов. // М. : Горячая линия – Телеком, 2001. – 382 с.
6. Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж. Ф. Люгер. – М. : Вильямс, 2005. – 739 с.
7. Рассел С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. –М. : Вильямс, 2006. – 2000 с.
8. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети / А. П. Ротштейн. – Винница: УНИВЕРСУМ-Винница, 1999. – 400 с.
9. Руденко О. Г. Штучні нейронні мережі / О. Г. Руденко, Є. В. Бодянський. –Харків : Компанія СМІТ, 2006. – 890 с.
10. Xue J. Significant remote sensing vegetation indices: A review of developments and applications / J. Xue , B. Su // Hindawi Journal of Sensors – Article ID 1353691. – 2017. – С.1–17.
11. Hansen C. How to Get the Best Word Vectors for Resume Parsing/ C. Hansen, M. Tosik, G. Goossenet al. – 2013.

12. Chris D.Improving Vector Space Word Representations Using Multilingual Correlation /D. Chris – 2014.
13. Большие данные – Сторінка https://www.bigdataschool.ru/wiki
14. Big Data: характеристики, классификация, примеры – Сторінка https://neurohive.io/ru/osnovy-data-science/big-data
15. Анализ настроений в Twitter с помощью Python – Сторінка https://www.machinelearningmastery.ru/another-twitter-sentiment-analysis-with-python-part-6-doc2vec-603f11832504
16. Apache Spark – Сторінка https://ru.wikipedia.org/wiki/Apache_Spark
17. Охрана труда за компьютером, рабочее место – Сторінка https://ru.wikipedia.org/wiki/Охрана труда за компьютером
18. Аналіз профілів учасників соціальних мереж – Сторінка https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2018/paper/download/4139/4604
19. Дослідження задачі пошуку інформації у Big Data – Сторінка https://ela.kpi.ua/bitstream/123456789/26116/1/Veryha_magistr.docx
20. Стан та удосконалення безпеки інформаційно-телекомунікаційних систем – Сторінка http://bit.nau.edu.ua/wp-content/uploads/2019/09/Zbirnyk_SITS_2015.pdf
21. Збірник студентських та магістерських тез доповідей наукових досліджень – Сторінка http://www.kntu.kr.ua/doc/zbirnyki/2017/3.pdf
22. Прикладні нейронауки – Сторінка https://ua.waykun.com/articles/prikladni-nejronauki-ce.php

ДОДАТОК А

Схема функціональної структури комп’ютерної системи формування рекламних пропозицій у суспільних мережах з використанням соціальних графів та Big Data



Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ФОРМУВАННЯ РЕКЛАМНИХ ПРОПОЗИЦІЙ У СУСПІЛЬНИХ МЕРЕЖАХ З ВИКОРИСТАННЯМ СОЦІАЛЬНИХ ГРАФІВ ТА BIG DATA

Текст програми

804.02070743.22011-01 12 01

Листів 4

```
$ python
Python 3.7.0a0 (default:98c078fca8e0, Oct 31 2016, 08:33:23)
[GCC 4.7.3] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import parser
>>> parser.suite('if 42: print("Hello world")').tolist()
[257, [269, [295, [297, [1, 'if'], [305, [309, [310, [311, [312, [315, [316, [317,
[318, [319, [320, [321, [322, [323, [324, [2, '42']]]]]]]]]]]]]]]]], [11, ':'], [304,
[270, [271, [272, [274, [305, [309, [310, [311, [312, [315, [316, [317, [318,
[319, [320, [321, [322, [323, [324, [1, 'print']]], [326, [7, '('], [334, [335, [305,
[309, [310, [311, [312, [315, [316, [317, [318, [319, [320, [321, [322, [323,
[324, [3, "'Hello world'"]]]]]]]]]]]]]]]]], [8, ')']]]]]]]]]]]]]]]]]], [4, '']]]]], [4, ''],
[0, '']]
>>>
//...

/*  simple_stmt | compound_stmt
 *
 */
static int
validate_stmt(node *tree)
{
    int res = (validate_ntype(tree, stmt)
               && validate_numnodes(tree, 1, "stmt"));

    if (res) {
        tree = CHILD(tree, 0);

        if (TYPE(tree) == simple_stmt)
            res = validate_simple_stmt(tree);
        else
            res = validate_compound_stmt(tree);
    }
    return (res);
}

static int
validate_small_stmt(node *tree)
{
    int nch = NCH(tree);
    int res = validate_numnodes(tree, 1, "small_stmt");

    if (res) {
```

```
        int ntype = TYPE(CHILD(tree, 0));

        if (  (ntype == expr_stmt)
              || (ntype == del_stmt)
              || (ntype == pass_stmt)
              || (ntype == flow_stmt)
              || (ntype == import_stmt)
              || (ntype == global_stmt)
              || (ntype == nonlocal_stmt)
              || (ntype == assert_stmt))
            res = validate_node(CHILD(tree, 0));
        else {
            res = 0;
            err_string("illegal small_stmt child type");
        }
    }
    else if (nch == 1) {
        res = 0;
        PyErr_Format(parser_error,
                     "Unrecognized child node of small_stmt: %d.",
                     TYPE(CHILD(tree, 0)));
    }
    return (res);
}

/*  compound_stmt:
 *      if_stmt | while_stmt | for_stmt | try_stmt | with_stmt | funcdef | classdef |
decorated
 */
static int
validate_compound_stmt(node *tree)
{
    int res = (validate_ntype(tree, compound_stmt)
               && validate_numnodes(tree, 1, "compound_stmt"));
    int ntype;

    if (!res)
        return (0);

    tree = CHILD(tree, 0);
    ntype = TYPE(tree);
    if (  (ntype == if_stmt)
          || (ntype == while_stmt)
```

```
          || (ntype == for_stmt)
          || (ntype == try_stmt)
          || (ntype == with_stmt)
          || (ntype == funcdef)
          || (ntype == async_stmt)
          || (ntype == classdef)
          || (ntype == decorated))
        res = validate_node(tree);
    else {
        res = 0;
        PyErr_Format(parser_error,
                     "Illegal compound statement type: %d.", TYPE(tree));
    }
    return (res);
}

//...
static expr_ty
ast_for_expr(struct compiling *c, const node *n)
{
//...
 loop:
    switch (TYPE(n)) {
        case test:
        case test_nocond:
            if (TYPE(CHILD(n, 0)) == lambdef ||
                TYPE(CHILD(n, 0)) == lambdef_nocond)
                return ast_for_lambdef(c, CHILD(n, 0));
            else if (NCH(n) > 1)
                return ast_for_ifexpr(c, n);
            /* Fallthrough */
        case or_test:
        case and_test:
            if (NCH(n) == 1) {
                n = CHILD(n, 0);
                goto loop;
            }
            // обработка булевых операций
        case not_test:
            if (NCH(n) == 1) {
                n = CHILD(n, 0);
                goto loop;
            }
```

```
            // обработка not_test
        case comparison:
            if (NCH(n) == 1) {
                n = CHILD(n, 0);
                goto loop;
            }
            // обработка comparison
        case star_expr:
            return ast_for_starred(c, n);
        /* The next five cases all handle BinOps.  The main body of code
           is the same in each case, but the switch turned inside out to
           reuse the code for each type of operator.
         */
        case expr:
        case xor_expr:
        case and_expr:
        case shift_expr:
        case arith_expr:
        case term:
            if (NCH(n) == 1) {
                n = CHILD(n, 0);
                goto loop;
            }
            return ast_for_binop(c, n);

        // case yield_expr: и его обработка

        case factor:
            if (NCH(n) == 1) {
                n = CHILD(n, 0);
                goto loop;
            }
            return ast_for_factor(c, n);
        case power:
            return ast_for_power(c, n);
        default:
            PyErr_Format(PyExc_SystemError, "unhandled expr: %d", TYPE(n));
            return NULL;
    }
    /* should never get here unless if error is set */
    return NULL;
}
```