ХИМИКОТЕХНОЛОГИЧЕН И МЕТАЛУРГИЧЕН УНИВЕРСИТЕТ

ФАКУЛТЕТ ПО ХИМИЧНО И СИСТЕМНО ИНЖЕНЕРСТВО

КАТЕДРА „ИНФОРМАТИКА”

инж. Камал Смаил Мохаммед Ал Барзнжи

ИЗВЛИЧАНЕ НА ИНФОРМАЦИЯ ОТ ГОЛЕМИ МАСИВИ ОТ ДАННИ С ЦЕЛ УВЕЛИЧАВАНЕ НА КОНКУРЕНТНОСПОСОБНОСТТА НА ПРЕДПРИЯТИЯТА

А В Т О Р Е Ф Е Р А Т

на дисертация

за придобиване на образователната и научна степен „доктор”
по научна специалност 4.6. Информатика и компютърни науки (Информатика)

Научен ръководител: доц. д-р инж. Атанас Атанасов

Научно жури:
1. проф. д-р инж. Идилия Бачкова - председател, рецензент
2. доц. д-р инж. Веска Ганчева - рецензент
3. доц. д-р инж. Георги Ружеков
4. доц. д-р инж. Александър Ефремов
5. доц. д-р инж. Атанас Атанасов.

София, 2018

Дисертационният труд е написан на 165 страници, съдържа 58 фигури и 13 таблици. Цитирани са 107 източника.

Представеният дисертационен труд е обсъден и приет за защита на заседание на разширен научен съвет на научното звено на катедра „Информатика”, състояло се на 24. 04.2018г.

Публичната защита на дисертационния труд ще се проведе на .27.06.2018 г. от ..14.00.. часа в зала ....321....., сграда „А” на ХТМУ.

Материалите са на разположение на интересуващите се на интернет страницата на ХТМУ и в отдел „Научни дейности”, стая 406, етаж 4, сграда „А” на ХТМУ.

Глава 1. Въведение

Живеем в ерата на големите данни (Big Data), изследването на които се превръща в нова област за учените и бизнеса. Анализът на големите данни позволява на организациите да взимат по-добри решения, да откриват и предсказват промени и да осъзнават нови възможности. Големите данни разкриват ограниченията на съществуващите стратегии за извличане на информация (знания) от данни, като ни сблъскват с нови, несрещани досега ситуации, свързани с наличието на огромни количества данни. Въпреки сериозните усилия положени досега в областта на големите данни, трябва да се съгласим, че остава да бъде извършена още много работа за да могат да бъдат преодолени предизвикателствата свързани с тяхната хетерогенност, стабилност, скорост, точност, истинност, поверителност и инструктивна стойност. Извличането на информация от големи данни е способността да се екстрахира полезна информация от големи множества от данни или потоци, което не е било възможно досега поради големите обеми и разнообразието на данните, или високата скорост на потоците. Извлечената информация може да се окаже много полезна, като придобитото знание може да бъде модел за превръщане на разнообразие от видове, стилове, и форми в определена полезна информация.

Целите на техниките за извличане на информация от големи данни минават отвъд простото доставяне на някаква изисквана информация. Днес социалните мрежи, блоговете, сензорите, сайтовете за електронна търговия, търсачките, здравното обслужване, и други генерират огромни количество информация (зета байтове) във връзка с тяхната постоянна активност. Това огромно количество от информация, представляващо големи данни, не е същото като традиционни данни, в смисъл на обем (volume), разнообразие (variety), скорост (velocity), стойност (value) и истинност (veracity) на данните. Обемът характеризира количеството данни, което обикновено е огромно и нараства експоненциално. Разнообразието се отнася до вида на данните, например структурирани, полуструктурирани и неструктурирани данни и т.н. Скоростта характеризира количеството данни за единица време, което се генерира и доставя от източниците на данни и трябва да бъде обработено. Стойността се свързва с процеса на извличане на ценна информация от големи групи от социални данни и обикновено се нарича големи аналитични данни. Стойността е най-важната характеристика на всяко приложение, базирано на голям обем от данни, тъй като позволява генериране на полезна бизнес информация. Истинността касае верността, коректността, и точността на данните.

Имайки предвид гореизложеното, става ясно, че е необходимо разработването на нови по-добри алгоритми за разпознаване на често повтарящи се форми, схеми и модели,

за намиране на закономерности, за анализ на текст, за съвместно филтриране на препоръки и т.н. за да може да са получи някакво полезно знание от големите данни.

Безспорно, извличането на информация (знание) от такива данни в 21-ви век е голямо предизвикателство и в същото време дава възможност за взимането на добре информирани решения. Необходими са изследователски усилия за преодоляване на предизвикателствата и откриване на възможности за подпомагане на организациите и компаниите за взимането на точни и добре информирани решения, които да издигнат тяхната дейност и бизнес на по-високо ниво. Извличането на информация (знания) от данни е в основата на съвременната бизнес интелигентност. Съществуващите алгоритми за извличане на информация, обаче не работят директно с големите данни, а трябва да бъдат адаптирани за работа с програмни платформи за обработка на големи данни като Hadoop с MapReduce или на новата платформа с отворен код Spark. Разработката и реализацията на алгоритми за извличане на информация от големи данни е сериозно предизвикателство. Целта на настоящия дисертационен труд е разработване на скалируема софтуерна система, осигуряваща различни алгоритми за извличане на информация от големи данни с цел подпомагане на компаниите при взимане на решения, а също така, разработването и внедряването на нови алгоритми, който да могат да използват паралелната мощ на съвременните изчислителни системи. Крайната цел на предприетата изследователската работа е подпомагане работата на компаниите и бизнеса.

Глава 2. Литературен обзор

Тази глава прави обзор на големите данни, извличането на информация (знание) от тях и как големите данни могат да помогнат на компаниите и бизнеса чрез увеличаване на тяхната конкурентноспособност. Главният фокус на настоящия обзор е извличане на информация от големи данни с цел бизнес интелигентност. Също така са разгледани характеристиките и класификациите на големите данни. Хвърля се светлина върху въпроса как големите данни „добавят висока стойност“ като помагат на компаниите и бизнеса да взимат добре информирани решения с цел повишаване на растежа им. Разгледани са онлайн социални мрежи, системи за препоръки и системи за предсказване и връзката им с извличане на информация от големи данни и подпомагане на компаниите и бизнеса.

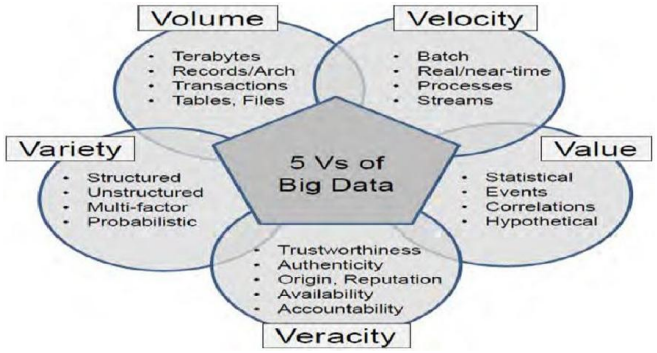
Големи данни

Големи данни (Big Data) е нов термин, използван за идентифициране на големи множества от данни с голям размер и висока степен на сложност. Големите данни се дефинират като голямо количество данни, изискващо нови технологии и архитектури, които правят

възможно извличането на полезна информация (знание) от тях чрез процеси на прихващане и анализ. Големите данни са важни, защото колкото повече данни се събират и натрупват, толкова по-точен е резултата който получаваме и толкова по-добра е способността ни да оптимизираме бизнес процеси. Големите данни са много важни за бизнес и социални цели. Данните могат да се получават отвсякъде, например от сензори, използвани за климатични измервания, от налични публикувани или споделени данни от социалните мрежи и медийни сайтове, от видео, аудио и т.н. Тези масиви от данни са познати като големи данни.

Характеристики на големите данни

Големите данни имат три основни характеристики, познати като **3V**: Volume (обем), Variety (разнообразие) и Velocity (скорост). Някои организации, изследователи и инженери, занимаващи се с големи данни разширяват този 3V модел до 5V модел с включването на нови V: Value (стойност) и Veracity (истинност). Този модел е показан на Фиг. 2.1, а характеристиките обем, разнообразие, скорост, стойност и истинност са обяснени във въведението (виж по-горе).



Фиг. 2.1: 5Vs Характеристики на Големи Данни

Класификация на големите данни

Големите данни се класифицират в различни категории с цел по-добро разбиране на техните характеристики. Класификацията е важна поради големите мащаби на данните, съхранявани в облачните структури. Класификацията се основава на пет аспекта: (източници на данните, формат на съдържанието, съхраняване на данните, междинно съхраняване на данните, обработка на данните).

Области на използване на големите данни

Важността на големите данни се състои в способността за подобряване на ефективността чрез използването на огромни количества от информация от специфичен тип. Ако тези огромни количества се обработят по подходящ начин и използват целесъобразно, организациите и бизнесите могат да получат по добър поглед върху

тяхната работа. Големите данни се използват ефективно в множество области. Някои от тях са: автомобилна промишленост, високи технологии, петролна и газова индустрия, телекомуникации, медицина, компании за продажби и търговия, пакетирани потребителски продукти, медии и бизнес, транспортен и туристически сектор, финансови услуги, социални медии и онлайн услуги, публични услуги, образование и наука, здравеопазване, правораздаване и отбранителна индустрия и т.н.

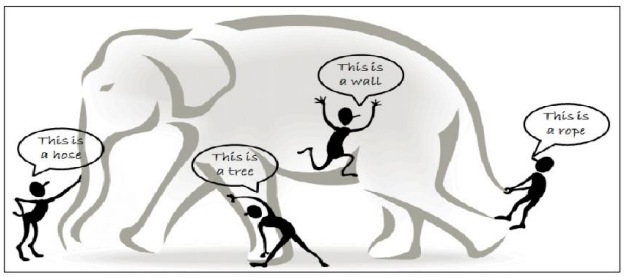
Извличане на данни за бизнес интелигентност

Извличането на информация от големи данни може да предостави обширни знания за бизнес анализи и интелигентност, които могат да бъдат използвани за взимането на добре информирани решения. Предприемачите в реалния свят имат постоянна нужда от такава дейност, за да могат да взимат бизнес решения. Fania и Miller изказват мнение, че извличането на знания от големи данни може да предостави по-богат и дълбок поглед върху конкретните начинания и повиши операционната ефективност и конкурентоспособността на компаниите. Интел ИТ има крило за анализиране на големи данни, които са под формата на неструктурирани данни. През 2012г. Интел стартира множество проекти, свързани с големите данни. Те включват системи за препоръки (прогнози), пазарно разузнаване, дизайн и валидация на чипове, разпознаване за откриване на вреден софтуер. Едновременно с това съществуват и ситуации, известни като жаргон „Удавени в Данни и Гладуващи за Знание“. Такива ситуации са решени с реализацията на разпределени програмни платформи като Hadoop и появата на изчисленията и услугите в облак като "инфраструктура като услуга" (Infrastructure as a Service (IaaS)), "платформа като услуга" (Platform as a Service (PaaS)) и софтуер като услуга (Software as a Service (SaaS)).

Нуждата от извличане на знания от големи данни

Wu *et al*., изказват мнение, че извличането на знания от големите данни е от съществено значение за обширна бизнес интелигентност (BI). Без взимането под внимание на характеристиките на големите данни, такива като обем (данни от порядъка на peta bytes), скорост (потоци от данни) и разнообразие (структурирани, полустуктурирани, и неструктурирани), може да се придобие погрешна представа за ситуацията. Фиг. 2.4 илюстрира нуждата от разглеждане на големите данни за извличането на информация. Хората, разглеждащи части от слона стигат до заключението че виждат маркуч, стена, дърво и въже. Примерът илюстрира, че може да се направят грешни заключения когато се базираме само на частични данни и липсва цялостен подход. Извличането на информация от големите данни ни дава възможност да открием скрити (латентни) знания в данните.

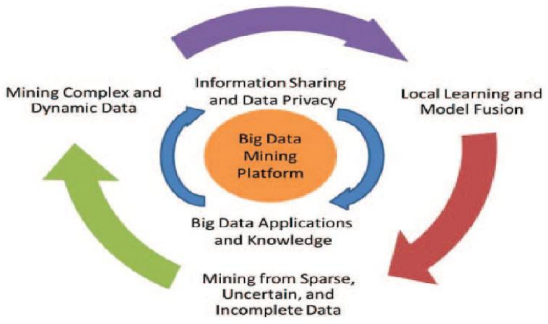
Такива бизнес знания могат да бъдат използвани за взимането на точни решения, които водят до бизнес растеж.



Фиг. 2.4: Ограниченото количество данни може да доведе до погрешни изводи.

Извличане на информация от големи данни

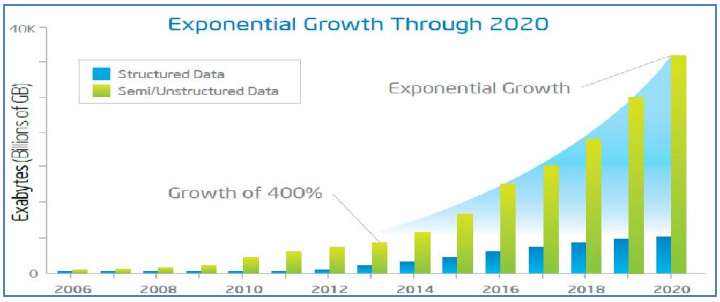
Bifet, изследва извличане на информация от големи данни в реално време. Анализ на данни в реално време е необходим, когато имаме потоци от данни от различни източници. Тъй като много организации произвеждат данни в реално време, налице са всички предпоставки данните никога да не са в покой. Такива данни растат динамично. За обработката им е нужна платформа (софтуерна рамка), която да може да реализира обработката на такива данни. Специално издание за извличане на информация от големи данни е предназначено за изследване на процеса в две свързани области, а именно здравни грижи и биомедицина. Wu *et al*., предлагат теория позната като HACE за характеризиране на чертите на големи данни и обработката им. Те комбинират два подхода: data-driven (основан на данни) и demand-driven (основан на нужда/търсене), за да създадат интелигентен модел за извличане на информация от данни. Платформата е показана на Фиг. 2.8. и служи за обработка на данни като осигурява нужните фази за извличане на информация от тях. Платформата за извличане на информация може да помогне откриването на бизнес знания в големите данни. Siddaraju *et al*, изследват MapReduce платформата за анализ на големи данни.



Фиг. 2.8: Платформа (софтуерна рамка) за обработка на големи данни.

Превръщане на големите данни в „голяма стойност“

Превръщането на големите данни в „голяма стойност“, т.е. извличането на знания, които реално могат да подпомогнат бизнес начинанията е от основна важност за компании и бизнеси. В доклад на Intel е проведено изследване, свързано със систематичен изчислителен анализ на големи данни и статистика – Big Data аналитикс. В него се предлага практична стратегия за „добавяне на голяма стойност“ към бизнес начинанията, чрез извличане на информация от големи данни. Според него много бизнеси са получили променящо играта предимство посредством обработка на големи данни. Потребителите на мобилни телефони и социални мрежи допринасят за експоненциалното нарастване на данните. В допълнение Интернет на нещата (Internet of Things (IoT)) също допринася за нарастването на данните. Текущото нарастване на големите данни до 2020 година е показано на Фиг. 2.10



Фиг. 2.10: Експоненциално нарастване на големите данни.

Както се вижда от фигурата, има структурирани, полуструктурирани, и неструктурирани данни, които са част от големите данни. Количеството данни расте експоненциално в интервала 2006 г. – 2020 г. Количеството данни се измерва в Exabyte (EB). В този интервал има подобно нарастване не само на големите данни но и на операциите по извличане на информация от данните. Това показва, че извличането на информация „добавя стойност към начинанията“ т.е. реално подпомага бизнесите и компаниите, когато разбира се получаването на информация е приложено за бизнес интелигентност. Има три модела за използване като "Извлечи, Преобразувай, Зареди" (Extract Transform Load (ETL)), интерактивни запитвания (Interactive Queries), и предсказващ анализ (Predictive Analysis).

Големите данни и уязвимостта на социалните мрежи

Mansour прави преглед на големите данни и свързани въпроси по отношение на социалните мрежи. Изследователят изказва мнение, че големите данни могат да доведат до уязвимост на социалните мрежи. Това се дължи на потенциална злоупотреба с лична

информация и разпространение на зловредно садържание из акоунтите на потребителите на социалната мрежа. Тъй като големите данни могат да бъдат използвани за широкообхватна бизнес интелигентност, винаги съществува възможност за злоупотреба с чувствителни данни в приложенията в социалните мрежи.

Как големите данни „добавят голяма стойност“

Според Ningyuxin и Liyueling, "добавянето на голяма стойност“, т.е. действително подпомагане работата на компаниите и бизнесите е възможно при разбиране на възможностите за бизнес интелигентност/разузнаване, което извличането на информация от големи данни предоставят в области като бизнес администрация, индустриални структури, публични ресурси, управление, иновации, и т.н. Дълбоки промени в администрацията е възможна при получаване на стойността за правене на промоция в маркетинга. Преценката на клиента и разбиранете на клиента може да помогне на организациите да се развиват по бързо. Освен това, точността на решенията при взимане на такива се повишава когато се използва анализ на големи данни. Анализът на големи данни може да подобри бъдещи индустриални структури. Експоненциалното нарастване на данните може да помогне в бизнес интелигентността и за осъществяване на необходими оптимизации за подобряване на индустриите. Публичните услуги могат да бъдат подобрени чрез анализ на големи данни, тъй като това може да помогне за извършване на анализ в реално време и взимане на бързи решения. Извличането на информация от големи данни може да помогне на правителства да разберат реалностите и да помогне за взимане на експертни решения. Големите данни могат да представят иновации, които да доведат до преобразуване на съществуващите системи в оптимизирани такива.

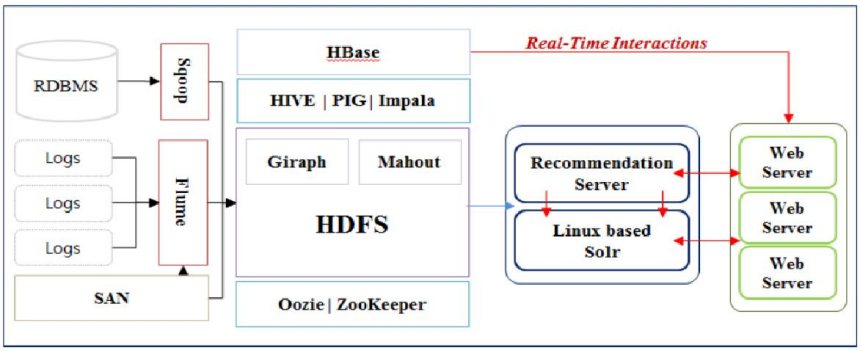
Извличане на мнение от големи данни

Weichselbraun, Gingl, и Scharl, предлагат метод за поставяне на семантични бази на знанието, които могат да бъдат използвани за семантичен и лексикален анализ. Такива бази могат да бъдат използвани за извличане и анализ на мнение (Sentiment Analysis). Целта им е да прихванат данни от социалните медии и да ги приложат за обогатяване на основите на познанието. За тази цел се използва тренировъчен корпус, като се извършват процеси по откриване на емоционални и контекстни термини и идентифициране на двусмислени термини. Те се класифицират в позитивни и негативни контекстни термини. Чрез използване на двете концепции и семантиката на изречението се генерират по-полезни лексикални бази данни, които имат концептуално ядро, асоциирано с позитивни или негативни концепции. Обогатяването и интегрирането на концепциите се прави с цел придобиване на по-богати бази знание. Те се използват в реалността за анализ на мнения и

извличане на мнения. Концепцията за граф и подграф, идентифициране на кандидат концепциите и финализиране на кандидатите се прави преди обогатяването на базите знание.

Система за препоръки с платформа за големи данни в облак

Hadoop е облачна изчислителна система с отворен код, притежаваща всични необходими характеристики като мащабируемост, надеждност, и устойчивост на грешки (толеранс към грешки). J. Kim и S. Hwang предложиха проект, една от целите на който е изследване на Hadoop и как неговата резпределена файлова система (Hadoop Distributed File System (HDFS)) поддържат облачни изчислителни способности за предоставяне на облачни услуги. Главната цел на проекта е изграждането на система за търсене и препоръки, използвайки облачно бзираната технология Hadoop. В конкурентен пазар, главната цел на всяка организация за създаване на софтуер е максимизиране на създадената стойност за дадена инвестиция. Следователно, управлението на ресурсите е решаващ фактор при максимизиране на стойност на предпиемаческо ниво. На Фиг. 2.14 е показана разработка на такава умна, машабируема система за препоръки и търсене. Конфигурирн е облачен клъстер от четири сървъра. За съхранение на данните, на върха на Hadoop, е избрана HBase разпределена, колонно-ориентирана база данни. Препоръките са базирани на KNN алгоритъм за намиране на заместител на лице напускащо проекта.



Фиг. 2.14: Система за препоръки с Hadoop в Облак

Изводи към Глава 2

В тази глава е направен литературен обзор на големите данни, извличане на информация от тях с цел „добавяне на голяма стойност“, т.е. подобряване работата и повишаване конкурентността на компании и бизнеси. Някои от направените изводи са:

- Големите данни, след обработка, могат да послужат за широкообхватна и прецизна бизнес интелигентност. Ако не се използват големи данни, бизнес информацията не е достатъчна за взимане на правилни решения. Казано накратко, резултатите от бизнес интелигентността могат да бъдат подвеждащи (изкривени).

- От литературния обзор става ясно, че обработката на големи данни става в разпределена среда. Необходима е разпределена програмна рамка от типа на Hadoop за да могат да бъдат обработвани големи данни. В общия случай, за съхранение на големи данни се използват центрове за данни асоциирани с облак.
- Откритите полета за изследователска работа могат да бъдат обобщени така: Големите данни са относително ново явление. При обработка им алгоритмите трябва да се съобразяват с техните характеристики като обем, разнообразие, и скорост.
    - ✓ Обемът касае големи количества данни с експоненциален растеж.
    - ✓ Разнообразието касае структурирани, неструктурирани и полуструктурирани данни.
    - ✓ Скоростта касае непрекъснати потоци от нови данни, които достигат източниците на данни.
- Имайки предвид гореизложеното, става ясно, че са необходими нови алгоритми в областта на намиране на често-повтарящи се модели, съвместно филтриране за препоръки, и т.н. за да може да бъде извлечена информация (знание) от големите данни. Целта на настоящата дисертационна работа е изграждането на обширна, разширима програмна рамка (платформа), която да осигурява различни алгоритми за извличане на знания от големи данни с цел повишаване на конурентноспособността на компании и бизнеси.

ЦЕЛ НА ДИСЕРТАЦИОННАТА РАБОТА:

Цел на дисертационния труд е изграждането на разпределена, софтуерна платформа, която осигурява алгоритми за генериране на препоръки за повишаване на конкурентноспособността на фирмите. Платформата трябва да има възможност да комбинира два различни подхода за формиране на препоръки с цел по-добри резултати. Комбинирането на резултатите на два алгоритъма предоставя по-добри възможности за обширна бизнес интелигентност добър подход за „добавяне на стойност към начинанията“, т.е. повишаване конкурентостта на компаниите. За постигане на основната цел са поставени следните под-цели:

1. Анализ на извличането на информация от големи данни и различни предизвикателства и проблеми свързани с големи данни. Изясняване на техниките за обработка и съхранение на големи данни и анализ на предизвикателствата, свързани с бази данни с големи мащаби, намиращи се в различни области. Разясняване работата на алгоритмите за машинно обучение и библиотеките за машинно обучение с големи данни.

2. Анализ на съществуващите техники в системите за препоръки и подобряване на някои от тях, особено техниките за съвместно филтриране, за изпълнение на част от поставения план за изграждане платформата (цел на дисертацията) за генериране на препоръки при големи количества данни.
3. Анализ на различните видове мерки за подобие (similarity measures) и метрики за оценка (evaluation metrics) и използването на част от тях за оценка, предсказващ анализ, формиране на препоръки при големи множества от данни.
4. Предлагане на архитектура за генериране (формиране) на препоръки при големи данни посредством разработка и развитие на съвместни техники за филтриране базирани на памет (Memory-Based Collaborative filtering techniques), базирани на потребител (User-Base) и базирани на предмет/стока (Item-Based).
5. Разработване и комбиниране на алгоритми за съвместно филтриране, предметно-базирани алгоритми и алгоритми на алтернативна регресия на най-малките квадрати (Alternating Least Square (ALS)) за формиране на нова хибридна система за препоръки (Hybrid Recommender System) за разрешаване на някои от предизвикателствата и проблемите, стоящи пред методите за генериране на добри препоръки от големи бази данни.
6. Разработване на алгоритми за анализ на мненията (Sentiment analysis) на основата на известните класификационни алгоритми за машинно обучение, анализ на текст и генериране на препоръки, като Naïve Bayes и Support Vector Machine.
7. Разрааботване на препоръки и предсказване на тенденции, като (Key Performance Indicators), необходими на компаниите и бизнесите за по-добро разбиране на потребителите и пазара и за взимане на добри решения.
8. Оценяване на разработените алгоритми на основата на известни тестови комплекти от данни (Data Sets), съвместими със системи за препоръки и машинно обучение.

Глава 3 . Извличане на информация от големи данни и машинно обучение

В тази глава е разяснено извличането на информация от големи данни и въпроси и предизвикателства, акцентиращи на отличителните характеристики на големите данни. Също така са дискутирани методи и техники за работа с големи данни. В допълнение са анализирани алгоритмите за машинно обучение и библиотеките за машинно обучение за работа с големи данни.

Извличане на информация от големи данни

Днес социални мрежи, търсачки, сензори, уеб-блогове, електронна търговия и някои други приложения генерират големи обеми от данни (зета байта) по време на

тяхната рутинна работа. Това огромно количество от данни, разглеждано като големи данни, е различно от традиционните данни от гледна точка на обем, разнообразие, и скорост. Извлечените и придобитите от данните знания са изключително полезни, защото представляват различни видове структури или модели. Целите на техниките за извличане на информация от големи данни минават отвъд обикновено доставяне на изисквана информация или дори разкриване на някакви скрити връзки и модели между числените параметри. Процесът на откриване на полезни знания от големи множества данни и потоци, които могат да помогнат при взимане на конкретни решения, е сложен и е свързан с редица предизвикателства.

Предизвикателста, свързани с извличането на информация от големи данни

Анализът на големи данни се разпределя на множество стъпки (нива), всяка от които с висока степен на сложност. Някои от стъпките са: придобиване на данни и записване, извличане на информация и прочистване, интерпретация на данни, обобщаване и илюстрация, моделиране и анализ, обработка на заявки, и интерпретация. Всяка от тези фази е свързана с някакво предизвикателство. Хетерогенност, разнообразие, скорост, мащаб/обем, сложност, навременност, и поверителност са някои от предизвикателствата на извличането на информация и знания от големите данни.

Хетерогенност и непълнота. Проблемите на анализа на големи данни произтичат не само от големите мащаби, но и от наличието на смесени типове данни, базирани изцяло на различни модели или политики в събираните и съхранени данни. Данните могат да са както структурирани, така и неструктурирани. На практика около 80% от данните генерирани от компаниите и фирмите са неструктурирани. Те не могат да бъдат записани във формат ред/колона както структурираните данни. Трансформирането на тези данни в структуриран формат за по късен анализ е главното предизвикателство пред извличането на информация от големите данни. Наличието на непълнота в данните създава несигурност в определен етап на анализа и трябва да бъде контролирано посредством статистически анализ на данните. Ефективното справяне с този проблем също е предизвикателство. „Непълни данни“ означава липсата на стойности в някои области за определени образци.

Мащаб и сложност: Управлението на големи и бързо растящи обеми данни е сериозно предизвикателство. Традиционните софтуерни програми средства не са достатъчни за справянето с нарастващите обеми данни. Анализът на данни, подредбата, извличането, и моделирането са също предизвикателство поради мащаба и сложността на данните, които трябва да бъдат анализирани.

Скорост: При големите данни скоростта е много важно обстоятелство. Способността за бърз достъп и бързо извличане на големи данни не е само субективно предпочитание, а е задължително, особено при потоци от данни, където трябва да приключим процеса на обработка/извличане в определен период от време, в противен случай резултатът от обработката/извличането ще е много по-малко ценен или дори безполезен.

Навременност: Колкото по-голямо е множеството от данни, толкова повече време отнема анализът му. Конструирането, на система която коректно се справя с дължината е възможно да доведе до система, която обработва определена дължина от данни по-бързо. Но когато говорим за скорост в контекста на големи данни не се има предвид само бързината на обработка на данни. Има множество ситуации, в които крайният резултат на анализа е нужен без забавяне.

Поверителност и сигурност: Тъй като достъпът до големите данни изискват се извършва от разнообразни домейни, то сигурността и поверителността на информацията играят основна роля в изследването и технологиите на големи данни. Разработването на алгоритми за генериране на случайни множества от лични данни така, че да може да има определена поверителност е ключов изследователски проблем.

Техники за извличане на информация от големи данни

Анализът на големи данни се превръща в основен инструмент за автоматизирано откриване на знания свързани с регулярно повтарящи се форми, модели и схеми, и скрити политики. Тези големи множества от данни са твърде и сложни за да могат хората ефективно да извличат полезна информация от тях без помощта и ресурсите на изчислителните инструменти на новите технологии. Платформата Hadoop с MapReduce или Apache Spark предоставят нови и интересни методи за обработка и трансформиране на сложни, неструктурирани или огромни количества от данни в смислено знание.

Платформа Hadoop

Apache Hadoop е платформа (софтуерна рамка) с отворен код на Java за заявки и обработка на огромни количества данни на големи хардуерни клъстери. Apache Hadoop е готова за ползване облачна изчислителна технология, която се използва от много компании и бизнеси за обработка на големи данни. Тя произлиза от Google's MapReduce и Google File System (GFS). Hadoop има два главни компонента, а именно: Hadoop Разпределена Файлова Система (Hadoop Distributed File System (HDFS)) и MapReduce програмна рамка. Системата за съхранение не е физически отделена от системата за обработка. Фигура 3.1 показва главните компоненти на платформата Hadoop, както и компонентите на MapReduce (Job Tracker and Task Trackers) и HDFS (Name Node and Data Nodes).

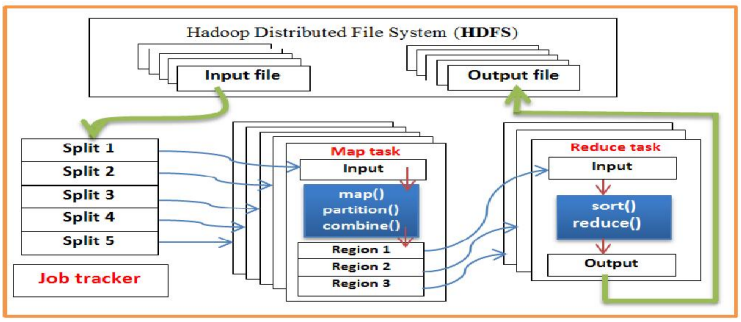


Figure 3.1. Платформа Hadoop

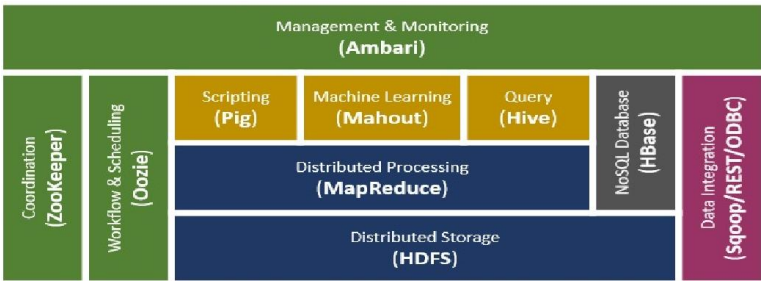
Екосистема Hadoop

Apache Hadoop е екосистема, разработена от Apache Software Foundation за разрешаване на предизвикателствата, свързани с големи данни. Сегашната Hadoop екосистема, както е показано на Фиг. 3.3, включва определен брой асоциирани компоненти включително в Apache Projects, които са изградени около платформата Hadoop, и са част от Екосистемата Hadoop, като най-важните компоненти са обяснени тук:

HDFS: Разпределена файлова система със значителна устойчивост на (толеранс към) грешки, отговорна за съхранението на данните в клъстери, и използвана в системи за съхранение.

MapReduce: Ефективен подход за паралелно програмиране за разпределена обработка на големи количества данни на клъстери. Използва се за системи са обработка.

Apache Mahout: Библиотека за извличане на данни и машинно обучение (мащабируема).



Фиг. 3.3. Екосистема Apache Hadoop

Apache Mahout

Apache Mahout е съвкупност от библиотеки за машинно обучение разработена, така че да е мащабируема и надеждна. Apache Mahout е проект с отворен код, който предоставя безплатна реализация на мащабируеми и разпределени алгоритми за машинно обучение в областта на Препоръки, Клъстери и Класификации. Той предоставя както неразпределени, така и разпределени (Map-Reduce) алгоритми за препоръки. Целта на безплатната библиотека за машинно обучение Mahout е изграждането на мащабируеми инструменти за машинно обучение и софтуерна рамка за извличане и анализ на данни в разпределена

среда. Както е показано на фигура 3.9, библиотеката Mahout разполага с много алгоритми за подобие и позволява на разработчиците да ги интегрират в съвместна система за филтриране и препоръки (Collaborative Filtering Recommender Systems). Понастоящем, библиотеката Mahout е подходяща за приложения, които изискват мащабиране към големи множества от данни, защото е отворена за реализации, които работят върху Apache Hadoop. Компании като Adobe, Facebook, LinkedIn, Twitter, и Yahoo използват Mahout при разработка на техни проекти.

Classification	Similarity Measures
Clustering	Pearson Correlation
Recommender/Collaborative Filtering	Spearman Correlation
Evolutionary Algorithms	Euclidean Distance
Pattern Mining	Tanimoto Coefficient
Regression	Log Likelihood Similarity
Dimension reduction	Neighborhood Measures
Similarity Vectors	Nearest N Users Algorithm

Фиг. 3.9: Списък с Алгоритми, Подобия и Мерки на Mahout

Apache Spark

Apache Spark е бърза библиотека с общо предназначение за обработка на данни от голям мащаб. Spark поддържа кеширане на основната памет и диспечеризация от типа loop-aware . В допълнение Spark реализира MapReduce и е Java-базирана (както е и Hadoop). Тези характеристики позволяват на потребителите да използват съществуващата Hadoop приложна логика към Spark чрез неговия Scala потребителски интерфейс. Spark предоставя програмна рамка за разработване на приложения в реално време с по-ефективно използване на паметта за обмяна и споделяне на информация между различните модули на дадено приложение. Това позволява на Spark да изпълнява входно-изходни операции и изчисленията по ефективно, без да чака HDFS да му предостави дял или множество дялове. Освен това, Spark предоставя разпределена абстракция на главната памет (Resilient Distributed Datasets (RDDs), която позволява потребителите да изпълняват изчисления в паметта на големи системи. Когато потребителите изпълняват задачи, системата създава множества данни/записи от входа (RDDs) и разпределя тези записи в главната памет. Когато множеството от данни е по-голямо от главната памет, Spark прилага механизъм за разпределяне в серии и съхраняване на частите на множеството във вторични хранилища.

Apache Spark има следните характеристики:

Бързодействие: Изпълнява програми до **100** пъти по-бързо от Hadoop MapReduce в паметта, и **10** пъти по-бързо на диск.

Лесен за употреба: Лесно писане на приложения на Java, Scala, Python, and R. Spark предоставя 80 оператора от високо-ниво, които правят лесно изграждането на паралелни апликации.

Работи навсякъде: Spark работи на Hadoop YARAN, Apache Mesos, самостоятелен клъстерен режим, EC2, или в облак.

Общост: Комбинира SQL, потоци (*streaming*), и комплексни анализи.

Spark поддържа набор от библиотеки за работа с бази данни (SQL и DataFrames), за машинно обучение MLlib и др. Те са показани на фигура 3.10, в която Spark SQL е модул за работа със структурирани данни. За разлика от базовия Spark RDD интерфейс, интерфейсът, осигурен от Spark SQL дава повече информация, както за структурата на данните, така и за изчисленията, които се извършват. Spark Streaming прави по-лесно изграждането на мащабируеми, устойчиви на грешки приложения за поточна обработка. Приложенията се разработват чрез оператори от високо ниво. GraphX е интерфейс за диаграми и диаграмно-паралелно изчисления. Той е гъвкав, работи безпроблемно както за диаграми така и за колекции. MLlib е мащабируема библиотека за машинно обучение. Притежава висококачествени алгоритми, 100 пъти по-бърз е от MapReduce и работи на съществуващите Hadoop клъстери.

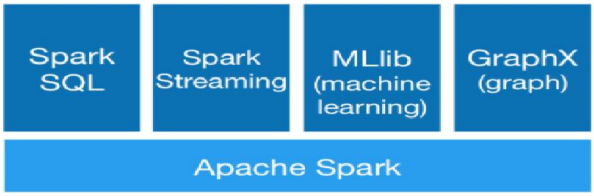


Figure 3.10: Apache Spark вградени библиотеки

NoSQL Бази данни

Базите данни могат да бъдат разделени на три типа:

RDBMS - Система за Управление на Релационни Бази-Данни -;

OLAP - Аналитична Онлайн Обработка (Online Analytical Processing);

NoSQL- non-SQL или не-релационни бази-данни,

NoSQL се използват за облачни бази данни, за големи бази данни и са разработени за съхранение, разпределяне и достъп до данни чрез използване на методи различни от тези на релационните бази данни (RDBMS's). NoSQL технологията първоначално е създадена и използвана от Интернет лидери като Facebook, LinkedIn, Google, Amazon и други, чиято работа изисква система за управление на бази данни, която да чете и пише данни навсякъде по света, като мащабира, доставя и обработва масивни множества от

данни за милиони от потребители. Има четири общи типа (най-често срещани категории) от NoSQL бази данни. Таблица 3.1 показва оценка на NoSQL бази данни.

1. Съхранение на ключова-стойност (Key-value stores), катоDyanmoDB и BerkeleyDB.
2. Документно-ориентирани (Document-oriented), като MongoDB и CouchDB.
3. Колонно-ориентирани (Column-oriented), като HBase и Cassandra.
4. Диаграмни Бази Данни (Graph Database), катоOrientDB и Neo4J.

Таблица 3.1 Оценка на NoSQL бази данни по ключови атрибути

Модел на Данни	**Представяне**	**Мащабируемост**	**Гъвкавост**	**Сложност**	**Функционалност**
Key-value store	Високо	Висока	Висока	Никаква	Пром. (Никаква)
Column Store	Високо	Висока	Средна	Ниско	Минимална
Document Store	Високо	Пром. (Висока)	Висока	Ниско	Пром. (Ниско)
Graph Database	Променливо	Променлива	Висока	Висока	Диаграмна Теория

Изчисления в облак

В днешната софтуерна ера, непрекъснатото нарастване на количествата от данни изисква наличието на еластично мащабируеми центрове за данни, които да предоставят удобен достъп с високо качество, надеждност, и сигурност. Тези нужди доведоха до развитието на изчисленията в облак – една от появяващите се технологии на днешното време. Изчисленията в облак предоставят компютърни услуги, компютърен софтуер, и съхранение на данни далеч от локалната инфраструктура чрез облачно-базирани решения, в Интернет, където услугите или софтуера или данните се намират на далечни машини. Изчисленията в облак обикновено имат преден и заден край. Те са лесно достъпни. Клиентите могат да влязат в системата и да ползват данните отвсякъде. Изчисленията в облак спестяват на организациите притеснения относно място и надзор на записващите устройства и сървърите. Намаляват разходите за хардуер и софтуер. Някои големи компании като Amazon, Google, Microsoft, Yahoo, и IBM използват изчисленията в облак.

Машинно обучение

Машинното обучение е клон на изкуствения интелект. То помага на компютрите да се обучават и действат като човешки същества посредством алгоритми и данни. Целта на изчисленията в машинното обучение е да изведат предсказващи модели от текущите и историческите данни. Предполага се, че самообучаващият се алгоритъм ще се подобрява с повече тренировки и опит и в частност, че алгоритмите за машинно обучение могат да постигнат изключително високи резултати в тесни области, използвайки моделни тренировки от големи множества от данни. Машинното обучение се използва в разнообразни задачи и различни области. Голям брой приложения използват машинно обучение и техния брой расте всеки ден. Задачите пред машинното обучение могат да

бъдат групирани в Класификация, Регресия, Клъстеризация, Разпознаване на аномалии, Препоръки, и Пространствена редукция.

Алгоритми за машинно обучение

Алгоритмите за машинно обучение използват данни, за да тренират модел. Процесът на трениране на модел се нарича още доставяне на данни за модел. Както се вижда от фигура 3.15, в зависимост от типа на тренировъчните данни, алгоритмите за машинно обучение се групират в две основни категории: контролирано (надзиравано) машинно обучение и неконтролирано (ненадзиравано) машинно обучение.

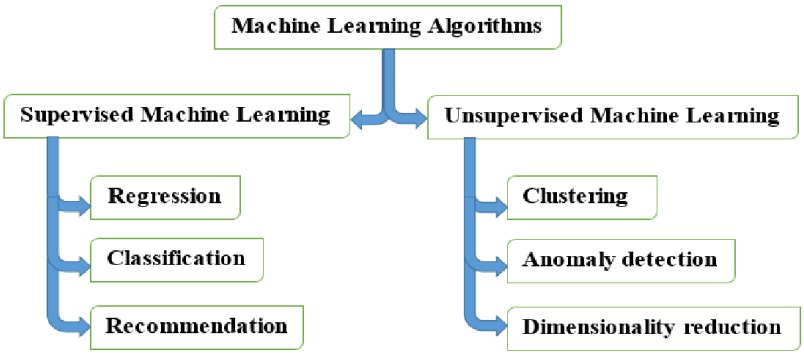


Figure 3.15: Machine learning algorithms

Алгоритми за контролирано (надзиравано) машинно обучение

Контролираното машинно обучение се извършва с множества от данни, на които са сложени означения (етикети). Всяко наблюдение в тренировъчното множество от данни има набор от характеристики и етикет. Алгоритъмът за машинно обучение се учи от данните да оценява и приближава (апроксимира) връзката между променливата на отговора (response variable), т.е. зависимата променлива, наричана още етикет (label) и една или повече прогнозни променливи (predictor variables), т.е. независими променливи, наричани още характеристики (features). Етикетите в едно тренировъчно множество могат да бъдат генерирани ръчно или извлечени от друга система. Алгоритмите за контролирано машинно обучение се групират в машини за препоръки, регресионни, и класификационни алгоритми. Често използвани алгоритми за машинно обучение за регресионни задачи включват: Линейна регресия (Linear regression), Дървовидно вземане на решения (Decision trees), и Ансамбли от дървета (Ensembles of trees), за класификационни задачи: Логистична Регресия (Logistic Regression), Поддържаща векторна машина (Support vector machines (SVM)), Бейс Модел (Naïve Bayes) и Изкуствени Невронни Мрежи (Artificial Neural Network), а за Препоръки: Съвместно филтриране (Collaborative filtering) с ALS.

Алгоритми за неконтролирано (ненадзиравано) машинно обучение

Неконтролираното машинно обучение имаме когато на множеството от данни не са приложени етикети и желаният изход не е известен. При това обучение моделите се опитват да открият скрити структури в неетикирани данни или да сведат (редуцират) данните до техните най-съществени черти и характеристики. Тъй като не са зададени етикети за това кое е правилно, няма и мярка за грешка, с която да оценим обучения модел. С неконтролирания модел няма верен или грешен отговор, просто изпълняваме машинния алгоритъм и виждаме какви форми, схеми и модели са налични. Неконтролирано обучение се използва за групиране (Clustering), засичане на аномалии и пространствена редукция (намаляване на размерността). Широко използваните алгоритми за неконтролирано машинно обучение включват: Метод на k-средни (k-means), Метод на Главните Компоненти (Principal Component Analysis), и Декомпозиция по Сингулярни Стойности (Singular Value Decomposition).

Библиотеки с големи данни за машинно обучение.

Apache Mahout

Проектът Apache Mahout има за цел изграждането на мащабируема библиотека за машинно обучение. Mahout включва мащабируеми алгоритми за машинно обучение. Някои от алгоритмите могат да се използват за задачи като Препоръки, Класификация и Групиране. Главната характеристика на Mahout са неговата мащабируемост. Той работи както на единичен възел, така и на клъстер (група) от машини. Неговите алгоритми се изпълняват на Hadoop така, че работят добре в разпределена среда. Mahout предлага различни системи за препоръки, въпреки че може да си изградите собствени. По подразбиране системите за препоръки са: Система за препоръки базирана на потребител (User-based recommender), Система за препоръки базирана на продукт (Item-based recommender), и Система за препоръки „Slope One“.

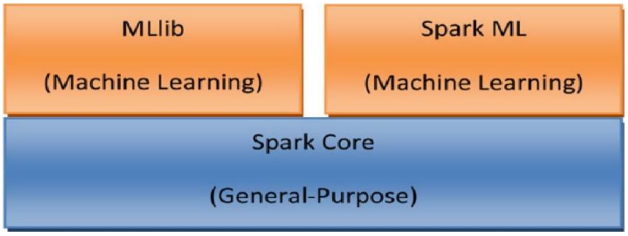
Apache Spark

Apache Spark е платформа за обработка на данни с големи мащаби и е по-нова програмна рамка изградена на същите принципи като Hadoop. За разлика от Mahout, тя не е обвързана с MapReduce. Вместо това, използва кеширане в паметта за да извлече работно множество от данни, да го обработи, и да повтори заявката. Има данни, че приложенията на Spark работещи директно с данни съхранени върху диск са до десет пъти по-бързи от тези на Mahout.

Spark осигурява две библиотеки за машинно обучение, MLlib и Spark ML (позната също като Pipelines API). Spark MLlib предоставя алгоритмите за машинно обучение базирани на RDD формата – Гъвкави Разпределени Множества от Данни. Новият Spark ML се характеризира с голяма мащабируемост и изключително лесна употреба. Spark ML

е базиран на множества от данни и позволява използването на SQL. Задачите за извличане на характеристики и манипулацията с тях са много лесни тъй като ползват и Spark SQL заявки.

Spark ML библиотеката допълнителна характеристика наречена "Pipeline", което гарантира конвейерна обработка на алгоритмите за машинно обучение. Както се вижда на фигура 3.16 и двете библиотеки осигуряват абстракции от по-високо-ниво за машинно обучение отколкото ядрото на Spark. Spark библиотеките съдържат реализации на много алгоритми и средства които могат да бъдат използвани при решаването на общи задачи като Регресия, Класификация, Групиране, Намаляване на Размерността, Извличане на Характеристики, Извличане на Често повтарящи се модели, и Препоръки.



Фиг 3.16 MLlib и Spark ML работят на Spark.

Изводи към Глава 3.

Количествата данни на световно ниво растат експоненциално поради експлозията от данни в социалните мрежи, машините за търсене, новите източници на данни, медийните сайтове, търговията на борсата, и т.н. Големите данни стават нова област за научни изследвания и бизнес приложения. Анализът на големите данни помага на компаниите да взимат по добри решения, да предсказват и разпознават промени и да осъзнават нови възможности. В тази глава беше направено следното:

- Дискутирани са въпросите и предизвикателствата, свързани с извличането на информация от големите данни, такива като хетерогенност, мащабируемост, скорост, точност, достоверност, поверителност, и инструктивна стойност.
- Обяснена е работата на платформите за анализ на големи данни като Mahout, Apache Spark, NoSQL бази данни, и изчисленията в облак, за ефикасен анализ на големи данни и решаване на предизвикателствата при обработка на множества от данни с големи мащаби в различни домейни.
- В допълнение, разяснена е работата на алгоритмите за машинно обучение с популярните и известни библиотеки за машинно обучение с големи данни. Те помагат на организациите да разберат по-добре изискванията и потребностите на клиентите си и пазара с цел вземан на по-добри решения. Също така, помагат на изследователи и учени да извличат знания от големите данни.

Глава 4. Системи за препоръки

Тази глава обяснява работата на системите за препоръки, класификацията на системите за препоръки (традиционни/базови техники и модерни техники на системи за препоръки) за работа с големи данни, и въпроси свързани с тях. Също така са дискутирани някои метрики за подобие и оценка използвани за изчисляване на препоръки на основата на големи множества от данни. Накрая е описана предложената архитектура за формиране на препоръки чрез анализ на големи данни, чрез разработване на филтриращи техники, базирани на памет, базирани на потребител и базирани на продукт, използвайки Apache Mahout на Hadoop.

Системи за препоръки

Препоръката е предложение, което може да помогне за взимане на добро решение по-бързо. Системите за препоръки (RS) са технологии на изкуствения интелект, които са се превърнали в неразделна част от работата на много компании, предприятия и бизнеси. Системите за препоръки са важна част на информационните системи и екосистемата за електронна търговия. Те представляват мощен метод, който позволява на потребителите да филтрират големи количества информация и продуктови пространства. Повечето компании използват системи за препоръки, които представляват софтуер за избор на продукт, който да бъде препоръчан на индивидуалния клиент. Системите за препоръки са глобални в днешния пазар и имат голяма търговска важност, което се потвърждава от големия брой системи за препоръки, които се продават от компаниите. Успешните системи за препоръки използват данни за минали покупки и удовлетвореност на клиента, за да правят висококачествени персонализирани препоръки. Системите за препоръки принадлежат към класа на персонализираните, информационни, филтриращи технологии, които смислено подсказват от кой наличен продукт или стока клиентът би могъл да се заинтересува. Има много различни техники, които могат да се приложат за персонализация в системите за препоръки. Всички тези техники имат, както своите предимства, така и недостатъци.

Класификация на системите за препоръки

Системите за препоръки са станали основни приложения в електронната търговия и информационния достъп. Съществуват няколко основни типа техники за препоръки. Най-известната класификация включва съвместни, базирани на съдържание и базирани на знание техники. Различни методи се комбинират при хибридните препоръчители с цел оптимизиране на предсказванията и разрешаване на проблема с тесните места (bottlenecks) при индивидуалните техники. Както е показано на фигура 4.1 тези техники могат да бъдат

класифицирани в две групи: Традиционни, базови техники за препоръки и Модерни техники за препоръки.

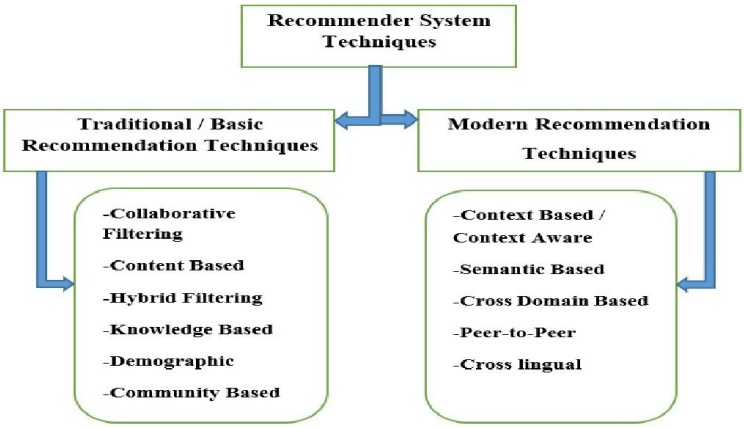


Figure 4.1. Класификация на Техниките на Системите за Препоръки

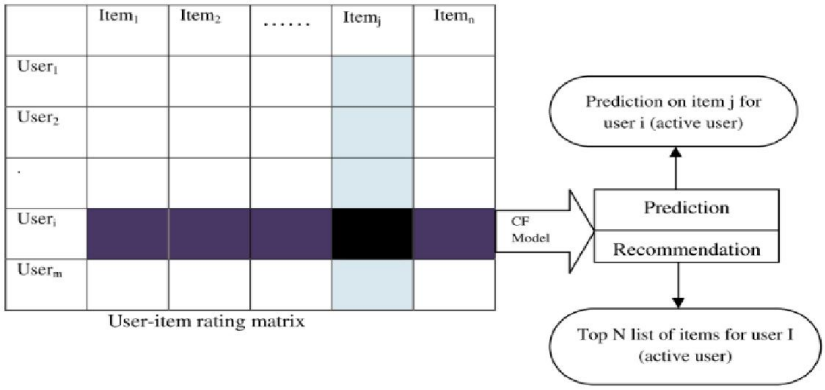
Традиционни, базови техники на системите за препоръки

Базовите подходи на системите за препоръки работят с два вида данни, които са (1) взаимодействие потребител-продукт, например рейтинги или поведение на потребителя при покупка на стоки и (2) атрибутна информация за потребителя и продукта, например тестов профил, ключови думи, и т.н. Тук ще бъдат дискутирани тези базови техники. Три от тях са главни техники, познати като филтриращи техники за препоръки. Това са: съвместното филтриране (Collaborative Filtering (CF)), филтрирането, базирано на съдържание (Contend Based Filtering (CBF)) и хибридното филтриране (Hybrid Filtering (HF)). Но три други техники за създаване на системи за препоръки заслужават да бъдат споменати: системи базирани на знание, демографки системи и системи базирани на общност (community-based). Те не са толкова широко разпространени както CF и CBF.

Съвместно филтриране (CF)

Терминът „съвместно филтриране“ означава съвместно използване на рейтинги от множество потребители за предсказване на липсващи рейтинги. Техниката за съвместно филтриране работи чрез изграждане на база данни от тип матрица потребител-продукт (user-item matrix), съдържаща предпочитанията на даден потребител към даден продукт. Тази техника съпоставя потребители с подобни интереси и предпочитания като изчислява подобието между техните профили и прави препоръки. Препоръките, формирани чрез CF могат да бъдат или предсказание или препоръка. Предсказанието има числена стойност, **Ti**, и представлява предсказания рейтинг на продукт **j** за потребител **i**, докато препоръката е списък с **N** продукта, които потребителят ще хареса най-много (фигура 4.2.).

Главното предимство на CF е, че няма нужда да знае нищо за продуктите, за да прави препоръки. Главното предизвикателство при разработка на съвместни филтриращи методи е, че оригиналната матрица съдържаща рейтингите е разредена. Зададените рейтинги в тази матрица се наричат още наблюдавани (observed). В насоящата дисертационна работа термините „зададен“ и „наблюдаван“ ще бъдат използвани взаимозаменяемо. Незададените рейтинги ще бъдат наричани „ненаблюдавани“ или „липсващи“. Основната идея на техниките за съвместно филтриране е, че незададените (липсващите) рейтинги могат да бъдат предсказани, понеже наблюдаваните рейтинги често са силно корелирани по различни потребители и продукти. Както е показано на Фигура 4.3, техниките за съвместно филтриране на системите за препоръки могат да бъдат групирани в два класа: базирани на памет (Memory Based) и базирани на модел (Model Based).



Фиг. 4.2: Процес на Съвместно Филтриране.

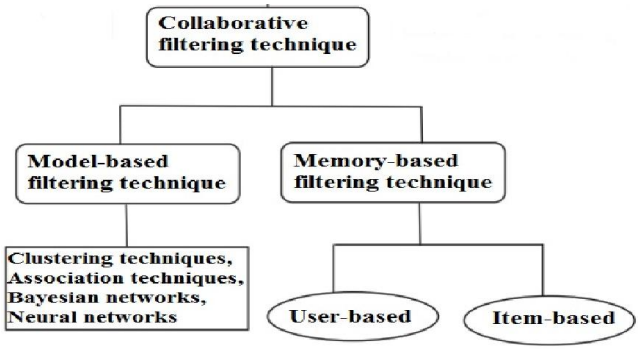


Figure. 4.3: Техники за Съвместно Филтриране на Системите за Препоръки

Филтриращи техники, базирани на памет:

Техниките, базирани на памет се наричат още Техники базирани на съседство (neighborhood-based). Поради ефективността на тези техники, те са получили широко разпространение в реални приложения. Съвместно филтриране на базата на памет може да бъде постигнато по два начина: посредством техники, базирани на потребителя и

посредством техники, базирани на продукта. Съвместното филтриране на базата на потребител изчислява подобието между потребители като сравнява техните рейтинги за един и същ продукт. След това се изчислява (предсказва) липсващият рейтинг за даден продукт за активния потребител като се осредняват с тегло рейтингите на дадения продукт по потребители, подобни на активния потребител. Теглата са подобията на тези потребители с целевия продукт (target-item). Филтриращи техники базирани на продукт изчисляват предсказанията, използвайки подобието между продукти а не между потребители.

Филтриращи техники, базирани на модел:

Тези техники използват рейтингите за да обучат модел с цел подобряване на представянето на Съвместните Техники за Филтриране. Процесът на изграждане на модел може да включва техники за машинното обучение или техники на извличане на информация от данни. Примери за такива техники са: Дървовидно вземане на решения (decision trees), модели базирани на правило (rule-based models), Бейс методи (Bayesian methods), модели на латентния фактор (latent factor models) и Техника за пространствена редукция или Намаляване на размерността (Dimensionality Reduction technique), както и такива като Разлагане по Сингулярни Стойности (Singular Value Decomposition), Техника за Допълване на Матрица (Matrix Completion Technique), Регресия (Regression) и Групиране (Clustering). Техниките, базирани на модел анализират матрицата потребител-продукт, за да установят връзки между продуктите. Те използват тези връзки за да сравнят списъка с топ-N препоръки. Техниките, базирани на модел разрешават проблемите с разреденост асоциирани със системите за препоръки и имат високо ниво на покритие дори за разредени рейтинг матрици.

Техники за филтриране, базирани на съдържание

Техниките за филтриране, базирани на съдържание, (Content-Based Filtering (CBF)) са други техники, използвани от системите за препоръки. При системите за препоръки CBF всеки продукт е представен с вектор на характеристиките (feature vector) или атрибутен профил (attribute profile). Характеристиките съдържат числени стойности и номинални стойности представящи определени аспекти от продукта такива като цвят, цена, и т.н. Системите за препоръки CBF работят с профили на потребители създадени в началото. Те използват главно тагове и ключови думи за по-добро и ефективно филтриране. При методите базирани на съдържание, рейтингите и поведението по време на закупуване се комбинират с информация за съдържание, налична в продуктите. Описанията на продукта, които са снабдени с рейтинги, се използват за тренировъчни

данни за създаване на класификация спрямо потребителя (user-specific classification) или регресионен модел.

Хибридна филтрираща система за препоръки

Хибридната комбинация включва най-малко две различни техники с цел преодоляване на непълнотата на единичните методи използвани при разделянето. Хибридните системите за препоръки комбинират две или повече техники за препоръки, за да постигнат по-добро представяне и по-добри резултати с по малко от недостатъците на всяка една индивидуална техника. Използвайки хибридни подходи можем да избегнем някои от ограниченията и проблемите на чистите системи за препоръки, като проблема със студения старт (cold-start problem) и др. Хибридите могат да бъдат особено полезни когато включените алгоритми покриват различни случаи или аспекти на множеството от данни. Има различни стратегии за получаване на хибридизация и те са класифицирани в седем категории основани на класификацията на Burke's на хибридните методи чрез анализ на хибридните системи за препоръки. Хибридните методи са:

- **Претеглен (Weighted):** Означава реализацията на различни методи отделни и след това комбиниране на резултатите за формиране на списък с препоръки.
- **Каскаден (Cascade) :** Едната система за препоръки подобрява резултатите на друга.
- **Смесен (Mixed):** Препоръките на няколко различни системи за препоръки се представят едновременно. Подобно е на претеглената хибридизация, но резултатите не се комбинират в една крайна препоръка.
- **Превключващ (Switching):** Системата превключва между различните техники за препоръки в зависимост от текущата ситуация. Това означава, че се превключва на различни алгоритми, като се използва алгоритъмът, който се очаква да даде по-добри резултати в дадения контекст.
- **Комбинация на характеристики (Feature combination):** Характеристики от различни източници на препоръки се събират в един единствен алгоритъм за препоръки.
- **Нарастване на характеристиките (Feature augmentation):** Изходът от една техника се използва за вход на друга техника.
- **Мета-ниво (Meta-level):** Моделът, обучен от една система за препоръки се използва за вход на друга. Разликата от предходния вид е, че целият модел се използва за вход.

Предизвикателства и проблеми свързани с техниките на системите за препоръки

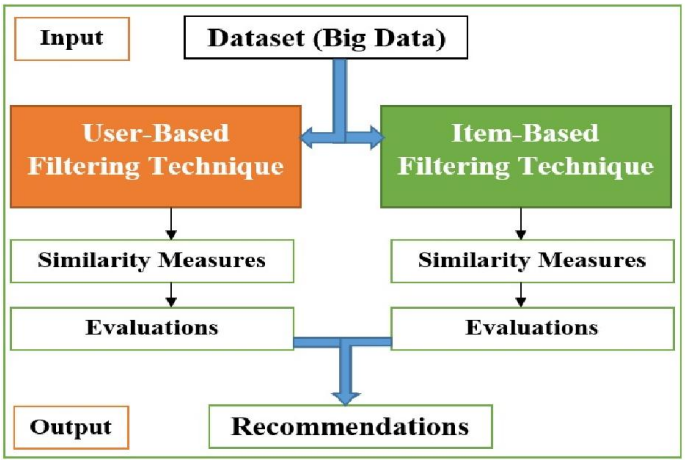
Системите за препоръки са били и все още са много успешни, но широка им употреба е извадила наяве някои действителни проблеми. По-долу са изброени някои проблеми и предизвикателства:

- **Студен Старт (Cold Start):** Означава ситуация в която системата за препоръки няма достатъчна информация за потребителя или продукта, за да направи предсказание.
- **Разреденост (Sparsity)**: е проблемът с липса на информация, т.е. когато само малък брой от общия брой налични продукти са оценени (даден им е рейтинг) от потребителите.
- **Мащабируемост (Scalability):** Техника за препоръки, която е ефективна за множество от данни с малък размер може да не успее да формира нужния брой препоръки когато обемът на данните нарасне.
- **Щастлива случайност (Serendipity):** Това е свързано с липсата на изненада в препоръките. Продуктите които не са свързани с потребителя, може никога да не бъдат препоръчани.
- **Свръх (Прекалена) Специализация (Super (Over) Specialization)**: Само продукти подобни на тези които преди това са оценени от потребителя ще бъдат препоръчвани. Изследването на нови категории е невъзможно.
- **Сива овца и Черна Овца (Gray Sheep and Black Sheep)**: Ако даден потребител има рядък вкус и предпочитание, препоръката може да не е точна, тъй като няма „близки съседи“. Този проблем е наречен сива овца. Черни овци са тези потребители които нямат подобни или имат много малък брой хора които корелират с тях.
- **Синонимия (Synonymy):** е свойството много подобни продукти да имат различни имена.
- **Доверие (Trust):** Гласовете на хора с кратка история може да не са толкова достоверни, колкото гласовете на хора които имат богата история в техните профили.
- **Първоначален-Рейтинг** (**Early-Rater**): Когато нов продукт се появи , не може да бъде препоръчан преди някой потребител да го посети.
- **Измама (Fraud):** Системите за препоръки са непрекъснато приемани от комерсиални уебсайтове поради техните икономически ползи за продавачи и предлагащи услуги.
- **Поверителност (Privacy):** е най-големия проблем. За да получи най-точната и вярна информация.

Предложена архитектура за генериране на препоръки на основата на големите данни

В настоящата дисертационна работа се предлага система за генериране (формиране) на препоръки при наличие на голямо количество данни чрез филтриращи техники, базирани на памет (базирани на потребител и базирани на продукт). Тези техники не изискват предварително познаване на свойствата и характеристиките на продукта, а само информация за матрицата с рейтингите. Тези алгоритми за препоръки са

реализирани на платформата Hadoop, използвайки Apache Mahout, инструмент за машинно обучение, за осигуряване на мащабируема система за ефективна обработка на големи масиви от данни. Това са техники от анализ на големи данни (Big Data Analytics). Както е показано на фигура 4.4 системата взима големи данни (множество от данни) като вход. След това се използват два алгоритъма за формиране на препоръки. В първата фаза се извършва филтриране, базирано на потребителя, а след това се извършва филтриране базирано на продукта. Накрая се извършва сравняване и се дискутират резултати от техниките за да се определи тяхното качество при формиране на препоръки и генериране на резултати включващи всички предимства на двете техники и изключващи недостатъците в същото време.



Фиг. 4.4: Предложената Архитектура за формиране на препоръки за големи данни

Филтриращи техники базирани на памет за формиране на препоръки:

Поради ефективността на тези техники те са станали широко използвани от много приложения в реалния свят. Техниките, базирани на памет могат да бъдат приложени по два начина, посредством техники базирани на потребител и техники базирани на продукт. Препоръките които се формират на базата на тези техники могат да бъдат или предсказания или препоръки. Предсказанието е числена стойност, докато препоръката е списък с топ-N продукта, които потребителя ще хареса най-много. Техниките за филтриране, базирани на **потребител** изчисляват подобието между потребителите като сравняват рейтингите им за един и същ продукт и след това предсказание за рейтинга на даден продукт за активния потребител като тегловно средно от рейтингите на продукта по потребители подобни на активния потребител. Теглата са равни на подобието между даден потребител и с целевия продукт (target item). Техниките за филтриране, базирани на **продукт** изчисляват предсказанията използвайки подобието между продукти, а не подобието между потребители. Те изграждат модел на подобие между продуктите чрез

извличане на всички продукти, оценявани от активния потребител от матрицата потребител-продукт. Псевдокодовете на техниките, базирани на потребител и на продукт са дадени по-долу.

User-based Technique pseudo code:

1. **procedure** USER-BASED COLLABORATIVE FILTERING
2. **for all** $item_i$ which $user_u$ has no preference **do**
3. **for all** $item_j$ which $user_u$ has preference **do**
4. Compute a similarity **s** between $item_i$ and $item_j$
5. **end for**
6. Add $user_u$**'s** preference for $item_j$ **,** weighted by **s**, to a running average
7. **end for**
8. **return** top items, ranked by weighted average
9. **end procedure**

Item-based Technique pseudo code:

1. **procedure** ITEM-BASED COLLABORATIVE FILTERING
2. **for all** $item_i$ which $user_u$ has no preference **do**
3. **for all** $user_v$ which has a preference for $item_i$ **do**
4. Compute a similarity **s** between $user_u$ and $user_v$
5. **end for**
6. Add $user_v$**'s** preference for $item_i$ **,** weighted by **s**, to a running average
7. **end for**
8. **return** top items, ranked by weighted average
9. **end procedure**

Експериментална Оценка

Множество от данни:

Използвани са MovieLens масиви (множества) от данни. Та са събрани от изследователския проект GroupLens на Университета в Минесота. Тези масиви от данни са често използвани за съвместно филтриране и системи за препоръки. Използвано е MovieLens 100k множество от данни като главния фокус е отделен на таблицата с рейтингите. Тя се състои от 100000 рейтинга със стойности от 1 до 5, направени от 943 потребителя на 1682 филма, като всеки потребител е оценявал(класирал) поне по 20 филма.

Мерки за Подобие:

Системите за препоръки съдържат много метрики за подобие, които идват от областта на машинното обучение. Те са важни за системите за препоръки. Всяка метрика за подобие е свързана с методи за работа с векторни пространства. Има различни начини за дефиниране на подобие. Тъй като на всяка дефиниция за подобие съответства различна формула за изчисляване на подобието, различните дефиниции дават различни стойности на подобие. Мерки за подобие се използват от системите за препоръки за определяне на подобие между продукти и/или потребители в системата. Това се използва най-често за съвместно филтриране и хибридни системи които инкорпорират аспекти на съвместно филтриране. Мерки за подобие се използват често за определени оценъчни метрики от системите за препоръки. Има често използвани мерки за подобие. За предложената в дисертацията система се използвани Коефициентът на корелация на Пиърсън (Pearson Correlation Coefficient (PCC)) и подобие с лог-правдоподобност (Log-Likelihood Similarity). **Коефициентът на корелация на Пиърсън (PCC):** За това подобни, предпочитания на отделните потребители са основата, от която се изчислява подобието както между потребители, също така и между продукти. **Подобие с лог-правдоподобност** е подобно на *Tanimoto* подобието. Тази метрика за подобие се опитва да определи колко силно неправдоподобно е двама потребителя да нямат подобни вкусове (предпочитания), колкото по-неправдоподобно е, толкова по-подобни трябва да са двамата потребителя.

По-подробно описание на мерките и техните формули за подобие са дадени в дисертационния труд.

Метрики за Оценка на Системите за Препоръки:

Качеството на алгоритмите за препоръки може да бъде оценено използвайки различни метрики. Видът на използваната метрика зависи от типа на техниката за филтриране. В настоящата работа за оценка на техниките базирани на потребител и на техниките базирани на продукт се използва Средно Квадратична Грешка (*Root Mean Square Error* (RMSE)), Прецизност (Precision), Recall, и F1 Score. Тези критерии (метрики) са използвани широко за сравнение и оценка на представянето на системите за препоръки. В контекста на нашето множество от филми, RMSE ще оценява колко добре Системата за Препоръки може да предскаже рейтинга, който даден потребител би поставил на даден филм в скала от 1 до 5 звезди. RMSE „наказва“ в по-голяма степен по-големите абсолютни грешки. Колкото по- малка е стойността на RMSE толкова по-висока е точността на предсказанието. **Precision** (*P*) е частта от правилно препоръчаните продукти, които са релевантни за дадения потребител и е мярка за това колко от предсказаните продукта са били успешни.

$$\textbf{Precision} = \frac{Correctly\ recommended\ items}{Total\ recommended\ items} \quad \text{or} \quad \textbf{P = TP / (TP +FP)} \quad \textbf{(4.13)}$$

Recall (*R*) може да бъде дефиниран като частта на релевантните продукти, които са също част от множеството на препоръчаните продукти и е мярка за това колко добра е системата за препоръчване на продукти, от които потребителят се интересува. F-Мярката, **F-measure** (или F1-score),дефинирана по-долу помага да сведем *Precision* и *Recall* до една единствена мярка. Получаващата се стойност прави сравнението между алгоритмите и множествата от данни много просто и директно.

$$\textbf{Recall} = \frac{Correctly\ recommended\ items}{Total\ useful\ recommended\ items} \quad \text{or} \quad \textbf{R = TP/ (TP +FN)} \quad \textbf{(4.14)}$$

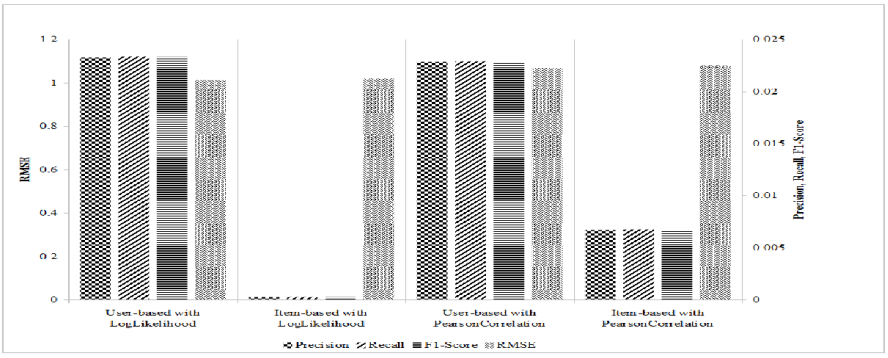
$$\textbf{F} - \textbf{measure} = F1 = \frac{2 * P * R}{P + R} \quad \textbf{(4.15)}$$

Резултати и Дискусия

За всеки неизвестен рейтинг (на даден продукт за даден потребител) са намерени най-подобните продукти, които са били оценени от дадения потребител (или най-подобните потребители, които са оценили дадения продукт) и е направено предсказание за неизвестния рейтинг като тегловно средно от „СЪСЕДНИ“ рейтинги. Подобието е изчислено като е използван коефициентът на Корелация на Пиърсън и подобие с Лог-правдоподобност (Log-Likelihood Similarity). За оценка на представянето са използвани следните метрики (критерии): RMSE, Precision, Recall, и F1-Score. Също така определен брой продукти (5 продукта) са препоръчани на дадения потребител (с потребителски номер user_id=25 със съседство (neighbourhood) size=100). Всяка система за препоръки е помолена да оцени (изчисли) стойностите на предпочитанията за тестовите данни и след това резултатите са сравнени с действителните стойности на предпочитанията, за да може да се измери качеството на направените препоръки (т.е. точността на предсказанията). Така, за работата на всяка система за препоръки е съпоставена стойност. Колкото по-ниска е тази стойност, толкова по-добре работи системата, т.е. предсказанията, които системата прави са по-близки до действителните предпочитания на потребителя. Таблица 4.5 и фигура 4.5 показват оценка за работата на четири Техники за Препоръки оценени чрез различни критерии (метрики).

Техника за препоръки с *Метрика за подобие*	RMSE	Precision	Recall	F1-Score
Потребител базирана с *LogLikelihood*	1.012003	0.023284	0.023358	0.023321
Продукт базирана с *LogLikelihood*	1.019956	0.000299	0.000299	0.000299
Потребител базирана с *PearsonCorrelation*	1.068607	0.02287	0.0229	0.022885
Продукт базирана с *PearsonCorrelation*	1.080572	0.006716	0.006754	0.006735

Таблица 4.5: Оценка на работата на четири техники за препоръки (колона 1) чрез четири различни критерия за оценка на предсказанията.



Фиг. 4.5: Резултати според метриката за оценка на представянето

Техниките, базирани на потребител взимат редовете, а техниките базирани на продукт взимат колоните, за да се изчислят подобието между всеки два потребителя (респективно продукта). След това подобието между потребители (респективно продукти) се използва, за да се изчислят препоръките, т.е. да се попълнят липсващите рейтинги в матрицата потребител-продукт. Техниката базирана на потребител има тенденцията да се представя много добре по отношение на метрики като *Precision* и *Recall*. Тя обаче е тежка от изчислителна гледна точка и не се мащабира добре. Техниката, базирана на продукт е по-лека от изчислителна гледна точка, но като цяло формира по-лоши препоръки по отношение на метриките *Precision* и *Recall*. Накрая, може да се каже, че корелационното подобие на Пиърсън дава по-добри резултати от подобието с Лог-Правдоподобност и при двете техники (потребител базирана и продукт базирана).

Изводи към Глава 4.

- Огромни количества данни от потребители се генерират всеки ден в множество области. Количеството данни расте драстично поради увеличаването на сайтовете за електронна търговия. В този момент, става актуален въпросът, как може да съхраняваме всичките тези данни и как да ги разбираме и тълкуваме. Големите данни е едно от най-добрите решения за съхраняване на тези данни и мотивира създаването на Системи за Препоръки, които помагат за разбирането и тълкуването им. Препоръките помагат за взимането на по-добри решения по-бързо. Те са в помощ както на клиенти така и на бизнеси.
- Обикновено, системите за препоръки са софтуерни системи, разработени за решаване на задачи за предсказване на потребителския рейтинг или предпочитание към продукти, които потребителя все още не е виждал. Системите за Препоръки са мощна

нова технология за извличане на допълнителна информация за предприятия и бизнеси. Тези системи помагат на клиентите да намерят бързо продуктите, които харесват и искат за закупят.

- Разгледани са различни техники на Системите за Препоръки и са разяснени техните силни и слаби страни, както и предизвикателствата свързани с хибридните системи създавани за повишаване качеството на препоръките. Дискутирани са различни алгоритми за обучение. Разгледани са мерки за оценка на представянето на алгоритмите и качеството на направените препоръки. Тези материали могат да подпомогнат бъдеща изследователска дейност в областта на хибридните системи за препоръки. Те могат да служат като отправна точка на изследователи, целящи подобрение на съвременните техники за препоръки.
- Разработени са нови алгоритми и Техники за Съвместно Филтриране (базирани на потребител и базирани на продукт) за формиране (генериране) на препоръки при наличие на големи количества структурирани данни с използване на Apache Mahout на Платформа Hadoop. Направено е сравнение между различните техники за определяне на качеството на формираните препоръки с цел подобряване на работата на системите чрез включване на предимствата и отстраняване на недостатъците на всяка една от техниките.

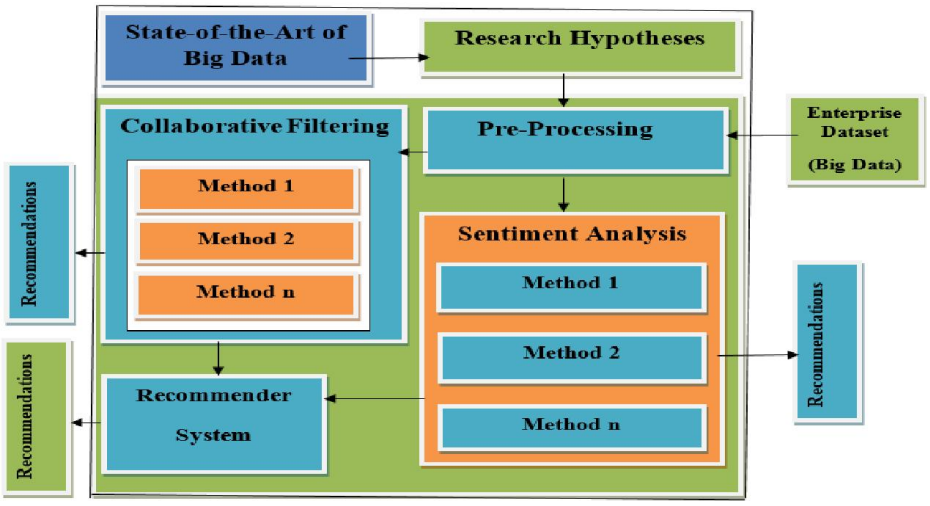
Глава 5. Методология и експерименти:

Тази глава описва използваната методология и експериментите, разяснява предложената Платформа за Облачно Базирана Хибридна Система за Препоръки (*Framework for Cloud Based Hybrid Recommender System* (FCHRS)) за извличане на информация от големи масиви от данни и дискутира реализацията на всички методи и алгоритми използвани в хибридната система за препоръки. Първо са алгоритмите за съвместно филтриране, след това алгоритмите за анализ на мнение, т.е. използването на обработка на естествен език, и анализ на текст. Платформата комбинира резултатите от алгоритмите. Комбинацията от резултатите на два алгоритъма дават по-полезна информация за бизнес интелигентност. Накрая са проведени експерименти за оценяване на работоспособността на всички алгоритми, използвани в платформата за избрания масив от данни с цел установяване на качеството на предложените модели.

Методология (предложената система):

Предприятията и бизнесите съхраняват данни „на склад“, които могат да използват за формиране на препоръки. Но за провеждане на широкообхватен бизнес анализ е

необходимо събиране на информация за поведението на клиентите и за продуктите от социалните медии, където хората изказват своето мнение. Там има налични данни в големи количества под формата на рейтинги, рецензии (отзиви), мнения, оплаквания, забележки, обратна връзка (feedback), и коментари относно продукти (стоки, събития, услуги, лица и т.н.). Така, за компаниите е от особена важност възможността за извличането на информация от социалните медии (големи данни) с цел формиране на препоръки. Настоящата работа предлага Платформа за Облачно Базирана Хибридна Система за Препоръки за Големи данни . Фигура 5.1 показва архитектурата за извличане на данни за формиране на препоръки. Тази платформа приема големи данни за вход. Системата за Препоръки използва един или повече алгоритъма за препоръки и формира (генерира) препоръките. Системите за препоръки събират информация на базата на предпочитанията на потребителите на продукти (филми, новини, видео, стоки и т.н.) или всеки друг атрибут. Тук ще бъдат използвани алгоритмите за генериране на препоръки. Първо се използва нов алгоритъм за съвместно филтриране на големи данни за формиране на препоръки. За тази цел се използват главно релационни данни. След това се използва нов алгоритъм за анализ на мнение за формиране на препоръки от големи данни. За тази цел се използват текстови данни. Първият алгоритъм може да работи с исторически данни на дадено предприятие (компания). Те са под формата на структурирани таблици. Вторият алгоритъм може да използва данни от социални медии, свързани с предприятието (компанията) и неговите продукти. Тези данни са неструктурирани.



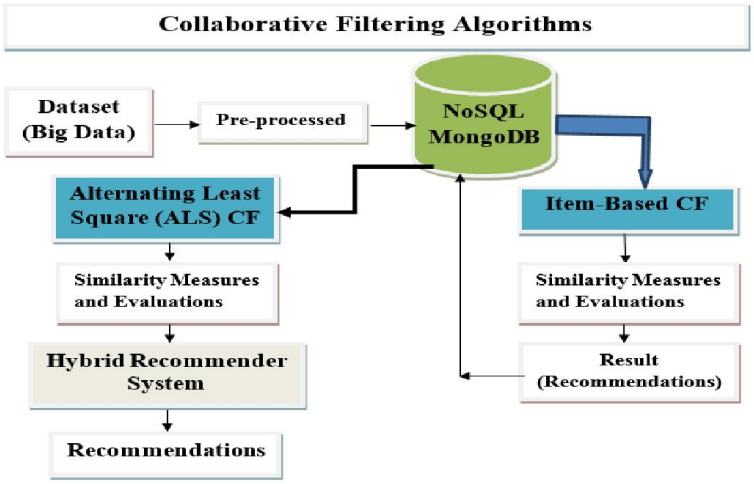
Фиг. 5.1: Схема на Предложената Архитектура за Извличане на Големи Данни

Накрая, гореописаната платформа комбинира резултатите от алгоритмите, и прилага две категории хибридни препоръчители. Комбинацията на резултатите от два алгоритъма е по-полезна за бизнес интелигентността и анализа. Тази комбинация е нова,

неизследвана територия и е обект на изследване в настоящата дисертация. Получените резултати могат да помогнат на предприятията и бизнесите да подобрят работата си и повишат конурентноспособността. В допълнение към изложеното ще бъдат приложени и алгоритми за машинно обучение и библиотеки за машинно обучение с големи данни.

Алгоритми за Съвместно Филтриране

Тази секция ще представи алгоритмите за съвместно филтриране. Подходите за съвестно филтриране изграждат модел на базата на поведението на потребителя в миналото (закупени или избрани продукти и/или числен рейтинг, даден на тези продукти), а също така подобни решения взети от други потребители. След това моделът се използва, за да предскаже продуктите (или рейтинга) на продуктите, от които потребителят може да се интересува. Методите за съвместно филтриране страдат от проблеми като студен старт (*cold start*), мащабируемост (*scalability*), и разреденост (*sparsity*). Фигура 5.2 показва структурата на Хибридната Система за Препоръки използваща Алгоритми за Съвместно Филтриране за формиране на препоръки при големи данни. Прилага се каскадна хибридизация. Това е една от категориите за хибридни препоръки, при която един препоръчител подобрява (refines) резултатите (препоръките) на друг такъв.



Фиг. 5.2: Хибридна Система за формиране на препоръки при големи данни.

За провеждането на тази хибридизация са използвани две техники за формиране на препоръки. Първата използва алгоритъм за съвместно филтриране **базиран на памет** (Memory-based), който прави предсказания за рейтинга за неоценени досега (без рейтинг от дадения потребител) продукти на базата на предходни оценки (рейтинги), поставени от този потребител на други продукти. Втората техника използва алгоритъм за съвместно филтриране **базиран на модел** (**Model-based)**, който използва колекция от оценки (рейтинги) за да обучи модел да предсказва липсващи оценки (рейтинги). Методите

базирани на модел стават популярни с появата на високоскоростните компютри, тъй като изискват сложни изчисления. За реализация на тази хибридна система за препоръки са използвани библиотеките Apache Mahout и Spark.

CF алгоритъм, базиран на памет и базиран на продукт

Ще бъде използван алгоритъм за препоръки базиран на продукт, като се използва методологията за Съвместно Филтриране CF. CF базиран на продукт използва подобието в рейтингите. Ако два продукта са едновременно харесвани (или едновременно нехаресвани) от даден потребител, и тази тенденция се запазва при повечето потребители, който са оценили двата продукта, то тогава двата продукта се очаква да бъдат подобни, понеже всеки потребител се очаква да има подобни предпочитания за подобни продукти. Тези подобия се използват за намиране и препоръчване на продукти, които са подобни или свързани с целевия потребител (target user). Този препоръчител, вместо да използва само потребители за мярка на близост в предпочитанията, използва и подобието на избраните продукти.

Алгоритъм с алтернативна регресия на най-малките квадрати , базиран на модел (ALS)

Алтернативната регресия на най-малките квадрати е известен алгоритъм за Съвместно Филтриране (CF). Тази техника цели запълването на липсващите елементи (рейтинги) в матрицата потребител-продукт. ALS алгоритъмът е базиран на матрична факторизация (декомпозиция). Факторизацията на матрици е стар алгебричен метод целящ разлагането на матрицата на произведение от матрици. ALS е оптимизационна техника за решаване на задачи за матрична факторизация. Тази техника е мощна, дава добри резултати, и е относително лесна за паралелна реализация. Библиотеката за съвместно филтриране Mllib е фокусирана върху препоръки, базирани на потребител, и използва алтернативна регресия на най-малки квадрати. Библиотеката Spark ML по настоящем поддържа филтриращи техники базирани на модел, при които потребителите и продуктите се описват чрез малко множество от латентни фактори, които могат да бъдат използвани да предскажат липсващите елементи (рейтинги). Spark ML използва алтернативна регресия на най-малките квадрати, за да получи тези латентни фактори.

Реализация на алгоритмите чрез библиотеките Apache Mahout и Spark:

В тази секция се описва реализация на предложения подход чрез използване на библиотеките на Apache Mahout и Spark. Всички алгоритми са написани на език за програмиране Java. След изпълнение на предварителни стъпки с файловете на множеството от данни, се въвеждат файловете на рейтингите (PreprocessedMovieRatings.csv) и файловете на филмите (PreprocessedMovies.csv) в NoSQL базата данни MongoDB. Въведените файлове се зареждат като входни данни за прилагане

на първия Алгоритъм (Препоръчител Базиран на Продукт) за генериране на резултати (препоръки) и след това изходните данни се въвеждат в MongoDB. От там това ново множество от данни и се подава като вход за ALS алгоритъм за прилагане на нов Хибриден Препоръчител. След въвеждане на множеството от данни в системата преди прилагането на моделите за оценка, множеството от данни се разделя по случаен принцип на **80 %** тренировъчно множество и **20 %** тестово множество. Тренировъчното множество се използва, за да обучи модела да прави предсказания/препоръки, а тестовото множество се използва за оценка на представянето на обучения модел. Псевдокодовете на алгоритмите са дадени по-долу и са обяснени подробно в дисертационната работа.

Hybrid Recommender pseudo code:

procedure Item-Based CF Using Apache Mahout.

Input: Rating File (PreprocessedMovieRatings.csv) [UserId, MovieId, Rating] from (MongoDB)

Output: Recommendations (ItemBasedRecommenderReults.csv) [UserId, MovieId, Rating].

1. Load data from MongoDB into MongoDBDataModel ←load (PreprocessedMovieRatings.csv)
2. **for all** $item_i$ which $user_u$ has no preference **do**
3. **for all** $user_v$ which has a preference for $item_i$ **do**
4. Compute a similarity **s** between $user_u$ and $user_v$, Perform PCC Similarity
5. **end for**
6. Add $user_v$**'s** preference for $item_i$ **,** weighted by **s**, to a running average
7. **end for**
8. **return** top items, ranked by weighted average
9. randomSplit() the dataset into training (80%) and test (20%),
10. Regression Evaluator for performance evaluation
evaluator← Evaluate Metrics by using RMSE.

end procedure

procedure Alternating Least Square CF using Apache Spark.

Input: (ItemBasedRecommenderReults.csv) [UserId, MovieId, Rating], MovieFile (PreprocessedMovies.csv) [MovieId, Title], from (MongoDB)

Output: UserId, MovieId, Title, Ratings, and Predictions.

1. Load data from MongoDB into Dataset<Row>. presentDS← MongoSpark.load (ItemBasedRecommenderReults.csv), and .movies ← (PreprocessedMovies.csv).
2. Split the data based on comma (',') and return Dataset of Rating class and Movie class objects ← Emit < ItemBasedRecommenderReults (UserId, MovieId, Rating)>and <PreprocessedMovies (MovieId, Title)>
3. Store the Dataset's data in memory using cache() , or
dataFrame.registerTempTable("ratings")
4. randomSplit() the (presentDS) into trainingDataset (80%) and tesDataset (20%).
5. ALS als← new ALS() .setMaxIter(10).setRegParam(0.01)
.setUserCol("UesrId").setItemCol("MovieId").setRatingCol("Rating");
6. ALSModel model ←als.fit(trainDataset)
7. predict← model.trasform(testDataset)
8. Regression Evaluator for performance evaluation
evaluator← Evaluate Metrics by using RMSE on predict.
9. return <UserId, MovieId, Title, Ratings, and Predictions> for the specified user.
10. Store the results in MongoDB or Result file (FinalResults.csv)

end procedure

Експерименти

Множество от данни:

За провеждане на експериментите е използвано моделното множество от данни MovieLens. Данните са събрани от GroupLens Изследователски Проект към Университета на Минесота. Тези масиви от данни са често-използвани за тестване на алгоритми за съвместно филтриране и системи за препоръки. Използваното множество от данни MovieLens 100k фокусира главно върху таблицата с рейтинги. Състои се от 100,000 рейтинги от 1 до 5 на филми, т.е. оценки на потребители за определени филми. Оценките са поставени от 943 потребителя, а общия брой на филмите е 1682, като всеки потребител е оценил (т.е. дал е рейтинг на) поне 20 филма.

NoSQL база данни MongoDB:

MongoDB е бинарен JSON (BSON) формат документен модел и водеща NoSQL база данни. MongoDB е най-често използваната NoSQL база данни. Тук MongoDB е интегрирана с Apache Mahout и Spark за създаване на колекция, получаване на колекция или списък от колекции, добавяне на документ или множество от документи към колекция, намиране на документ или множество от документи, и се използва за съхранение, зареждане, търсене, и обновяване на базата данни. Най-полулярните MongoDB графични потребителски интерфейси като Management Tools, Robo 3T (Robomongo) и Studio 3T (MongoChef) са използвани за вход и изход на масивите от данни с MongoDB. MongoDB може да бъде използван за структурирани и неструктурирани данни, и принадлежи към документно ориентираните бази данни, разработени за съхранение, извличане, и обработка на документно-ориентирана информация.

Мерки (критерии) за подобие

Мерките за подобие се използват от системите за препоръки и от системите за изкуствен интелект, такива като системите базирани на метода на прецедентите (Case-Based Reasoning), за определяне на подобието между продукти и/или потребители на системата. Мерките за подобие се използват често и за оценъчна метрика на системите за препоръки. За настоящата система се използват коефициента на корелация на Пиърсън за подобие.

Тренировъчно и тестово множество

За да оценим грешката, разделяме данните на две множества от данни: тренировъчни данни и тестови данни. След зареждане на множеството от данни в система, преди прилагането на моделите за оценка, данните се разделят на **80 %** тренировъчни и **20 %** тестови по случаен принцип. Тренировъчното множество се използва в процеса на

изграждане на модела, тоест за трениране (обучение) на модела за правене на предсказания и издаване (формиране) на препоръки. Тестовото множеството от данни е независимо от тренировъчното множество и не се използва в процеса на изграждане на модел. Тестовото множество се използва за оценка на представянето, тоест за изчисляване на това колко точни са предсказанията/препоръките. За оценъчни метрики на представянето се използват средно-квадратична грешка (RMSE) и други.

РЕЗУЛТАТИ

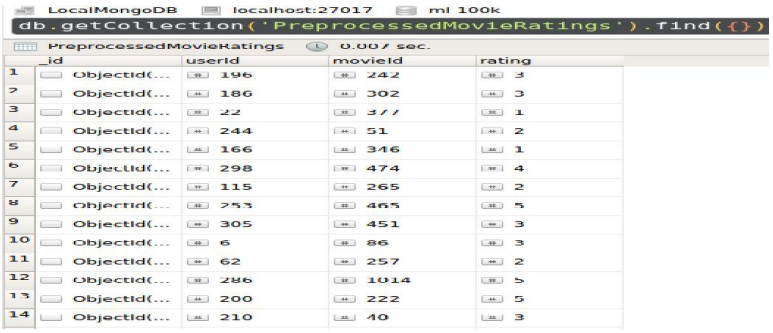
1. Базово множество от данни: Множеството от данни е обработено предварително преди въвеждането в MongoDB, с цел изчистване и премахване на ненужни атрибути, преобразуване на символите за интервали в запетая, и преобразуване формата на множеството от данни в (CSV) формат. Предварителната обработка се извършва върху базовите файлове като при това се генерират два файла: PreprocessedMovies, съдържащ филмите, и PreprocessedMovieRatings, съдържащ рейтинги на тези филми от разни потребители. На фигура 5.3 е показано част от съдържанието на PreprocessedMovies (movieId, title), а на фигура 5.4 част от съдържанието на PreprocessedMovieRatings (_id, userId, movieId, rating). Тези файлове са въведени в MongoDB като базови данни. Фигура 5.3 показва съдържанието и JSON формата на предварително обработения филмов файл в MongoDB. Множеството от данни се използва и от двете техники за съвместно филтриране.

```
+-------+--------------------------------------------------------+
|movieId|title                                                   |
+-------+--------------------------------------------------------+
|1      |Toy Story (1995)                                        |
|2      |GoldenEye (1995)                                        |
|3      |Four Rooms (1995)                                       |
|4      |Get Shorty (1995)                                       |
|5      |Copycat (1995)                                          |
|6      |Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)    |
|7      |Twelve Monkeys (1995)                                   |
|8      |Babe (1995)                                             |
|9      |Dead Man Walking (1995)                                 |
|10     |Richard III (1995)                                      |
|11     |Seven (Se7en) (1995)                                    |
|12     |Usual Suspects                                          |
|13     |Mighty Aphrodite (1995)                                 |
|14     |Postino                                                 |
|15     |Mr. Holland's Opus (1995)                               |
|16     |French Twist (Gazon maudit) (1995)                      |
|17     |From Dusk Till Dawn (1996)                              |
|18     |White Balloon                                           |
|19     |Antonia's Line (1995)                                   |
|20     |Angels and Insects (1995)                               |
+-------+--------------------------------------------------------+
only showing top 20 rows
```

```
LocalMongoDB   localhost:27017   ml-100k
db.getCollection('PreprocessedMovies').find({})
PreprocessedMovies   0.118 sec.
/* 1 */
{
    "_id" : ObjectId("5ace6d33fce12b2c7cfcb0dc"),
    "movieId" : 1,
    "title" : "Toy Story (1995)"
}

/* 2 */
{
    "_id" : ObjectId("5ace6d33fce12b2c7cfcb0dd"),
    "movieId" : 2,
    "title" : "GoldenEye (1995)"
}

/* 3 */
{
    "_id" : ObjectId("5ace6d33fce12b2c7cfcb0de"),
    "movieId" : 3,
    "title" : "Four Rooms (1995)"
}

/* 4 */
```

Фиг. 5.3 Предварително обработени данни с филми в MongoDB

Фигура 5.4 показва част от съдържанието на предварително обработеното множество с Рейтинги на Филми въведено в MongoDB. Това множество е основното, базово множество. То се зарежда и използва и от двете техники (Базирана на продукт и ALS) за съвместно филтриране, използвайки Mahout и Spark библиотеки за машинно обучение.

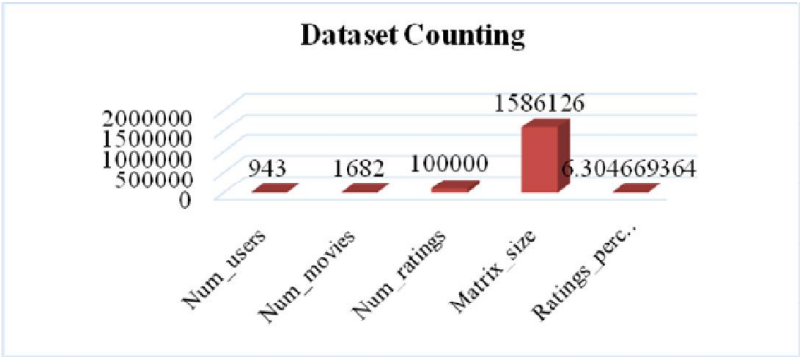


Фиг. 5.4 Предварително обработени данни с рейтинги на филми в MongoDB

Съдържанието на основното, базово множество от данни (предварително обработени рейтинги на филми) е показано на фигура 5.5 и графичен вид в таблица 5.2

Num_users	Num_movies	Num_ratings	Matrix_size	Ratings_percentage
943	1682	100000	1586126	6.30467

Таблица 5.2: Съдържание на базовото множество от данни (предварително обработено множество от рейтинги на филми).



Фиг. 5.5: Графично представяне на таблица 5.2

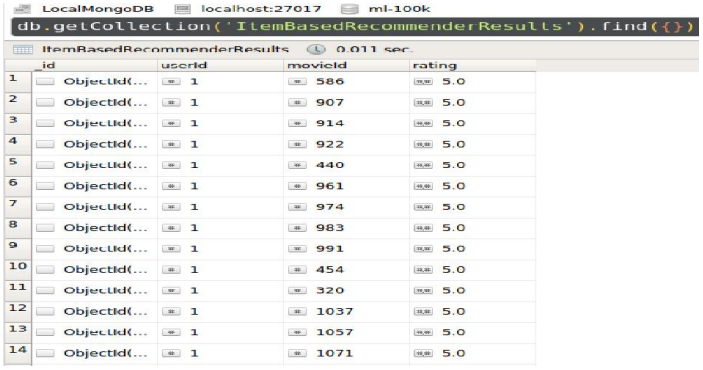
Фигура 5.6 показва резултатите от ALS метода (за изчисляване на липсващите рейтинги), приложен към базовото множество от данни PreprocessedMovieRatings, използвайки Spark ML. Тук са показани предсказания за рейтингите на филмите само за един потребител, а именно (userId =100).

```
+----------------------------+-------+------+------+----------+
|title                       |movieId|rating|userId|prediction|
+----------------------------+-------+------+------+----------+
|Good Will Hunting (1997)    |272    |4.0   |100   |4.4753714 |
|Kull the Conqueror (1997)   |266    |2.0   |100   |3.1227744 |
|Liar Liar (1997)            |294    |4.0   |100   |3.0400925 |
|Tomorrow Never Dies (1997)  |751    |4.0   |100   |2.9792948 |
|Boogie Nights (1997)        |340    |3.0   |100   |2.4169753 |
|Chairman of the Board (1998)|1234   |1.0   |100   |2.3016608 |
|Anna Karenina (1997)        |990    |3.0   |100   |2.2135324 |
|Rosewood (1997)             |292    |2.0   |100   |2.211881  |
+----------------------------+-------+------+------+----------+
```

Фиг. 5.6: Резултати (предсказания) за активен потребител (userId =100)

2. Ново множество от данни: Фигура 5.7 показва съдържанието на новото множество от данни - резултати от препоръчител, базиран на продукт (Item Based Recommender Results) с полета (_id, userId, movieId, and rating). Тези резултати са генерирани на базата на

базовото множество (PreprocessedMovieRatings), чрез прилагане на Техника Базирана на Потребител, използвайки Mahout библиотека за машинно обучение. След това, това ново множество е въведено в MongoDB и заредено като вход за ALS техниката за генериране на нова Хибридна Препоръка, използвайки Spark ML библиотеката за машинно обучение и за генериране на препоръки чрез Каскадна Хибридизация.



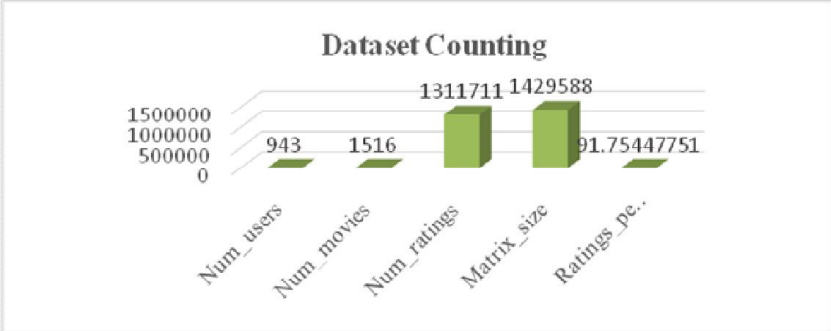
	_id	userId	movieId	rating
1	ObjectId(...	1	586	5.0
2	ObjectId(...	1	907	5.0
3	ObjectId(...	1	914	5.0
4	ObjectId(...	1	922	5.0
5	ObjectId(...	1	440	5.0
6	ObjectId(...	1	961	5.0
7	ObjectId(...	1	974	5.0
8	ObjectId(...	1	983	5.0
9	ObjectId(...	1	991	5.0
10	ObjectId(...	1	454	5.0
11	ObjectId(...	1	320	5.0
12	ObjectId(...	1	1037	5.0
13	ObjectId(...	1	1057	5.0
14	ObjectId(...	1	1071	5.0

Фиг. 5.7: Новото множество от данни (Item Based Recommender Results) в MongoDB

Съдържанието на новото множество (Item Based Recommender Results) е показано на таблица 5.3 и фигура 5.8 съответно.

Num_users	Num_movies	Num_ratings	Matrix_size	Ratings_percentage
943	1516	1311711	1429588	91.7545

Таблица 5.3 Преобразуване на новото множество от данни (Item Based Recommender Results)



Фиг. 5.8: Преобразуване на новото множество от данни (Item Based Recommender Results)

Фигура 5.9 показва резултатите получени от Хибридния Препоръчител чрез прилагане на ALS техниката върху новото множество от данни (Item Based Recommender Results), използвайки Spark ML библиотека за машинно обучение. Показани са само резултатите (предсказанията) за определен потребител (userId =100).

title	movieId	rating	userId	prediction
I Like It Like That (1994)	1424	5.0	100	4.286269
Panther (1995)	1438	4.157659	100	4.2395153
Land Before Time III: The Time of the Great Giving (1995) (V)	1412	3.1875868	100	4.004975
Last Time I Saw Paris	1123	4.5	100	3.977834
Kim (1950)	1200	4.474762	100	3.9504516
Two Bits (1995)	920	3.950424	100	3.93657
Sudden Manhattan (1996)	1644	3.0	100	3.9270477
Germinal (1993)	1488	3.8834848	100	3.8510914
Mark of Zorro	1116	4.1974688	100	3.8326356
Aparajito (1956)	1558	3.6222653	100	3.8191123
Fluke (1995)	726	4.0275965	100	3.7966263
Secret Agent	1205	3.220234	100	3.785085
Paris Is Burning (1990)	645	5.0	100	3.7462249
Bad Girls (1994)	1247	4.1173735	100	3.7309618
Walking and Talking (1996)	1262	4.237982	100	3.704977
Clean Slate (Coup de Torchon) (1981)	1560	3.4404387	100	3.6877959
Ruby in Paradise (1993)	962	3.5899367	100	3.6876113
Guilty as Sin (1993)	1213	4.06692	100	3.6680613
Swan Princess	1409	5.0	100	3.6666005
Secret Adventures of Tom Thumb	1555	3.3228757	100	3.6460786

only showing top 20 rows

Фиг. 5.9: Крайни резултати (предсказания) за активен потребител (userId =100), получени от новото множество от данни

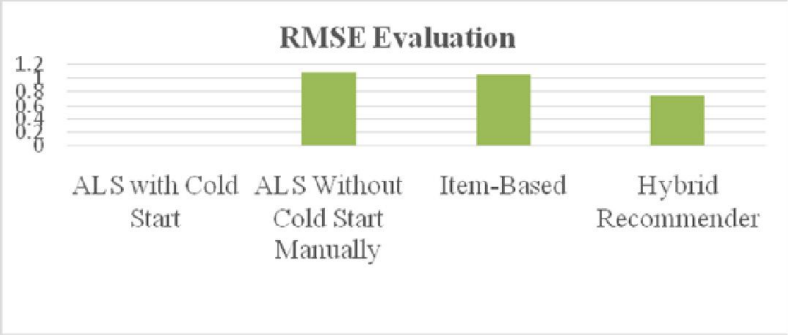
Метрики за оценка на представянето

Оценката на модел в Системите за Препоръки се осъществява чрез използване на различни метрики, но най-популярната е Средноквадратичната Грешка (RMSE). Средноквадратичната грешка е корен квадратен от средната стойност на квадратите на разликите между предсказаната стойност и действително наблюдаваната стойност. Метриката (RMSE) се използва за оценка на регресионни алгоритми и други. Тук са приложени различни модели за препоръки чрез Техники за Съвместно Филтриране към тренировъчното множество от данни, и след това е направена оценка на представянето на различните модели като е използвано тестовото множество от данни за изчисляване на средноквадратичната грешка (RMSE).

Таблица 5.4 и фиг. 5.10 показват средноквадратичната грешка, т.е. RMSE оценка на представянето на всички приложените техники. Представянето на Хибридния Препоръчител е най-добро тъй като има най-ниска стойност на средноквадратичната грешка.

Techniques with Evaluation Metrics	RMSE Evaluation
ALS with Cold Start	NaN
ALS Without Cold Start Manually	1.08727
Item-Based	1.06101
Hybrid Recommender	0.73724

Таблица 5.4: Средноквадратична грешка (RMSE) на Техниките за Съвместно Филтриране



Фиг. 5.10: Средноквадратична грешка (RMSE) на Техниките за Съвместно Филтриране

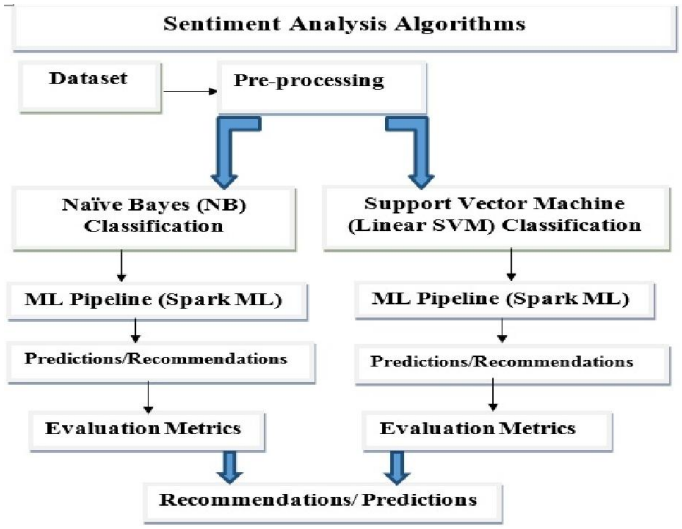
Анализ на резултатите

Резултатите, както е споменато в предходната секция, са получени чрез прилагане на Техники за Съвместно Филтриране върху голямо множество от данни на платформа базирана на Каскадна Хибридна Система за Препоръки, която реализира новия хибриден препоръчител използвайки Mahout и Spark библиотеки за машинно обучение. Първо, препоръчителят базиран на продукт е приложен над базовото множество от данни, което има налични само **(6.30 %)** от всички елементи (рейтинги) в матрицата потребител-продукт (таблица 5.2) и е приложена Пиърсън корелационна мярка за подобие за генериране на препоръки/предсказания. След това резултатът от предметно базирания препоръчител (за пояснение виж предходната секция) е използван за ново множество от данни, което вече е с по-висок брой налични елементи (рейтинги) **(91.75 %)** в матрицата потребител-продукт (таблица 5.3). Това ново множество се зарежда като вход за ALS техниката. Това е последната стъпка на новата Хибридна Система за формиране (генериране) на препоръки.

Оценката на работата на двете техники (i) Базирана на потребител и (ii) на алтернативна регресия на най-малките квадрати (ALS) по отделно, както и на съвместната им работа в Хибридния Препоръчител е направена чрез изчисляване на средноквадратичната грешка (RMSE). В таблица 5.4 и фигура 5.10 са показани резултати за (RMSE). Когато само ALS техниката е приложена към базовото множество от данни резултатът е **Nan** (липса на предсказание) поради проблемът със студения старт (cold start), при който повечето предсказания на са налични. Spark ML обаче ни позволява да отстраним проблема със студения старт като „ръчно“ премахнем липсващите предсказанията от резултатите. Тогава средноквадратичната грешка е **(1.08727)**.Когато към базовото множество от данни е приложена само Техниката Базирана на Продукт (Item-Based) се генерира ново множество с по-висок брой елементи (рейтинги) като при това средноквадратична грешка е **(1.06101)**. Проблемът със студения старт (cold start) и разредеността на матрицата (sparsity) е премахнат, но остава проблемът с мащабируемостта (scalability). Най-накрая, когато Хибридният Препоръчител е приложен към новото множество, използвайки ALS, средноквадратичната грешка е **(0.73724)**, проблемът с мащабируемостта (scalability) е решен, и са формирани по-голям брой препоръки. Както се вижда, средноквадратичната грешка (RMSE) е намаляла с около **40 %**, което означава, че формираните (генерирани) препоръки са с по-висока точност. Можем да заключим, че Хибридния Препоръчител е по-добър от единичните (отделни) техники за генериране на препоръки.

Алгоритми за анализ на мнение (Sentiment Analysis Algorithms)

Анализът на мнение или за извличане на мнение, се дефинирана като задачата за откриване и тълкуване на мнения за специфични продукти и други. Софтуерните приложения, свързани с наблюдение на социални медии, както и компании и бизнеси, използват анализа на мнение и машинното обучение, за да изградят по-точна представа за определени продукти, марки, и т.н. Анализът на мнение реферира към използването на обработка на естествен език, анализ на текст, и изчислителна лингвистика за идентифициране и извличане на субективна информация от различни източници. Интернет е богато място за събиране на мнение. Анализът на мнение на отзиви (рецензии) е процесът на изследване на отзиви (например в интернет) за определен продукт с цел съставяне на по-пълно мнение. Анализът на мнение може да се разглежда като класификационна задача, тъй като той класифицира даден текст като или положителен или отрицателен. Машинното обучение е един от широко използваните подходи за анализ на мнение. Фигура 5.11 показва структура на алгоритми за анализ на мнение с цел формиране на препоръки за големи масиви от данни, базирана на една от категориите Хибридни препоръчители. Тук е приложена смесена хибридизация, при която „препоръки от няколко различни препоръчителя се представят едновременно“. Една от формите на текстовия анализ е анализа на мнение. Ако имате откъс от текст и искате да разберете какъв вид емоция изразява, например, любов, омраза, положителна, отрицателна, и т.н. може да използвате техниките на анализа на мнение. Настоящата работа представя експерименти направени с използването на софтуерен инструмент с отворен код за машинно обучение и извличане на данни. Използвани са два класификационни алгоритъма при реализацията на инструмента за Анализ на Мнение.



Фиг. 5.11. Алгоритми за Анализ на мнение приложени към големи данни

Обработка на естествен език (Natural Language Processing (NLP)):

При решаването на проблемите от анализа на мнение са използвани някои техники и концепции от обработката на естествен език. Те са свързани със следните атрибути и характеристики са:

Характеристика: - представлява атрибут или свойство на наблюдението. Нарича се също променлива. Характеристиката представлява независима променлива. В таблично множество от данни, редът представлява наблюдение, а колоната характеристика.

Екстрактори на Характеристики (Feature Extractors)**:**

- **Честота на появяване на термин – обратна честота на документа** (Term frequency-inverse document frequency) (**TF-IDF**): е векторизационнен метод, широко използван в извличането на информация от текст, който оценява важността на даден термин за даден документ. MLlib разделя метода на: TF и IDF. **TF**: Както Hashing TF,така и Count Vectorizer могат да бъдат използвани за генериране на векторите на честота на появяване. **HashingTF** е преобразувател, който взима множества от термини и ги превръща в характеризиращи вектори с фиксирана дължина. В обработката на текст, „множество от термини“, може да е чанта с думи. **IDF** е оценител, който съставя IDF модел. IDF моделът взима характеризиращи вектори и мащабира всяка характеристика.

Чанта-с-Думи (Bag-of-Words **BoW):** е репрезентация на текст, описваща появата на думи в документ. Този метод разглежда всяко преброяване на думи като характеристика. Тъй като алгоритмите за машинно обучение не могат да работят директно с текст, текстът трябва да бъде преобразуван в числа, по-точно вектори от числа.

Преобразуватели на Характеристики (Feature Transformers)**:**

- **Tokenization** е преобразувател, който конвертира входен текст в малки букви (долен регистър (lowercase)) и го разделя на думи като използва интервалите (whitespaces) като разделители.
- **Отстранител на Стоп Думи** (Stop Words Remover): Стоп думите са думи, които трябва да бъдат изключени от текста понеже се появяват често и не носят много информация (смисъл) Например думи като „и, от, на , за” и др..

Naïve Bayes класификационен алгоритъм:

Класификацията на Bayes представлява метод за контролирано машинно обучение както и статистически метод за класификации. Той може да решава диагностични задачи и задачи за предсказване (прогнози). Той е прост мултикласов алгоритъм за класификация, базиран на приложение на теоремата на Bayes от теория на вероятностите. Naïve Bayes е

вероятностен модел, който прави прогнози чрез изчисляване на определени вероятности. Библиотеката Spark по настоящем поддържа многономинален (multinomial) Naïve Bayes. Naïve Bayes се използва в много приложения от практиката, такива като например анализ на мнение в текст с цел класификация на емоции, позитивни или негативни. Този алгоритъм се тренира (обучава) бързо и се тества лесно. Използва се за „предсказване на сценарии“ в реално време, тоест бързи предсказания на събития, базирани на данни, генерирани в реално време. Използва се в много системи за препоръки за издаването на полезни предложения към потребителите.

Алгоритъм на машини с поддържащи вектори (SVM):-

Това е друг популярен алгоритъм от алгоритмите за контролирано машинно обучение, използвани в много практически приложения като категоризация, класификация на образи, анализ на мнение, и разпознаване на ръкописни знаци (цифри). SVM е популярна и мощна техника за разпознаване и класификация. За разлика от Naïve Bayes това не е вероятностен модел, а предсказва класове на базата на това дали оценката на модела е положителна или отрицателна. Spark има реализация на линейния SVM, който е бинарен класификатор. В Spark ML1ib има пълен линеен SVM алгоритъм и LinearSVC модел, които са свързани със Spark библиотеката за машинно обучение. LinearSVC в Spark ML поддържат бинарна класификация. SVM се използва за класификация на текстове като положителни или отрицателни. Той работи добре за класификация на текстове поради някои свои предимства, такива като възможност за справяне с големи характеристики.

ML Pipelines (конвейер)

В машинното обучение често се практикува пускането на поредица от алгоритми за обработка и обучение от данни. Например, при обработката на прост текст работният процес може да включва няколко етапа: Разделяне на всеки документ на отделни думи, превръщане на съвкупността от думи в числен вектор на характеристиките, обучаване на модел за предсказания, използвайки вектора на характеристиките и етикети. MLlib представя такъв работен процес като Pipeline, който се състои от поредица от етапи (преобразувания и оценки) с цел по-лесно комбиниране на множество от алгоритми в един работен процес, който да бъде изпълняван в определена последователност. ML Pipelines предоставя множество от високо-качествени потребителски интерфейси (API), надградени върху рамки от данни, които помагат на потребителите да създават и настройват практични конвейерни машини за обучение.

Реализация на алгоритми за анализ на мнение чрез Apache Spark

Тази секция описва реализацията на анализ на мнение в етикирани текстови множества от данни чрез използването на Алгоритми и Библиотеки за Машинно

Обучение, такива като класификационни алгоритми (Naïve Bayes и Линеен SVM) и библиотеката Spark ML. В началото масивите от данни се зареждат като вход за системата. След зареждане преди съставяне на оценки множеството от данни се разделя по случаен принцип на две части: **80 %** тренировъчно множество от данни и **20 %** тестово множество. След това върху тренировъчното множество започва предварителна обработка, използвайки концепции от NLP за анализ на мнение, такива като Екстрактори на Характеристики и Преобразуватели на Характеристики. До тук всички стъпки за прилагане на двата алгоритъма са еднакви, но при стъпка (10) изпълнението на алгоритъм зависи от избора на алгоритъм. След това всички следващи стъпки от тази процедура са същите и за двата алгоритъма до края на процедурата. Ето защо е написана само една процедура и за двата алгоритъма. Изпълнението е чрез ML Pipeline (Spark ML). Стъпките на процедурата по реализация на алгоритъмите са показани и обяснени подробно в дисертационната работа.

ЕКСПЕРИМЕНТИ

Множество от Данни: Множествата от данни (data sets) са данни, предварително снабдени с етикети и подготвени за използване от алгоритми за контролирано машинно обучение за анализ на мнение. Множествата от данни с етикирани сентиментни изречения (изречения съдържащи сентимент, емоция, мнение) са налични на https://archive.ics.uci.edu/ml/datasets/. Това множество от данни съдържа образец от отзиви (рецензии) от три уебсайта, отзиви за филми (imdb.com), отзиви на Amazon за мобилни телефони (amazon.com), отзиив за ресторанти (yelp.com). Включени са, случайно избрани, **500 положителни** и **500 отрицателни** отзива от всеки сайт. Отзив с отрицателно мнение е маркиран с етикет **0**, а отзив с положително мнение е маркиран с **1**. Отзивът се отделя от етикета си с табулатор (tab). Тук са използвани отзивите и рецензиите на amazon.com и yelp.com. Те се намират във файловете amazon_cell_labelled.txt и yelp_labelled.txt. Целта е моделът да се обучи (тренира) за правене на предсказания и да предсказва дали дадено изречение има положителен или отрицателен сентимент (мнение, отношение, емоция). По конкретно, ще бъде обучен мултикласови класификатор използвайки файловете amazon_cell_labelled.txt файла и yelp_labelled.txt.

На фигури 5.13 и 5.14 са дадени образци от съдържанието на Amazon и yelp базите данни, съответно, и техните схеми.

Тренировъчни и тестови множества

За да се направи оценка на грешката, множествата от данни се разделят на две части: тренировъчно (обучително) множество и тестово множество. След зареждане на

данните в системата, преди извършване на оценките, данните се разделят случайно на **80 %** тренировъчно множество и **20 %** тестово множество. Тренировъчното множество се използва за да обучи модела да прави предсказания. Тестовото множество, което е независимо от тренировъчното и не се използва в процеса на обучение, се използва за тестване (оценка) на представянето на обучения модел. За оценка на точността на предсказанията се използва *accuracy score*.

```
+-----+--------------------------------------------------------------------------------------------------------------+
|label|text                                                                                                          |
+-----+--------------------------------------------------------------------------------------------------------------+
|0.0  |So there is no way for me to plug it in here in the US unless I go by a converter                             |
|1.0  |Good case Excellent value                                                                                     |
|1.0  |Great for the jawbone                                                                                         |
|0.0  |Tied to charger for conversations lasting more than 45 minutesMAJOR PROBLEMS!!                               |
|1.0  |The mic is great                                                                                              |
|0.0  |I have to jiggle the plug to get it to line up right to get decent volume                                     |
|0.0  |If you have several dozen or several hundred contacts then imagine the fun of sending each of them one by one|
|1.0  |If you are Razr owneryou must have this!                                                                      |
|0.0  |Needless to say I wasted my money                                                                             |
|0.0  |What a waste of money and time!                                                                               |
|1.0  |And the sound quality is great                                                                                |
|1.0  |He was very impressed when going from the original battery to the extended battery                            |
|0.0  |If the two were seperated by a mere 5+ ft I started to notice excessive static and garbled sound from the headset|
|1.0  |Very good quality though                                                                                      |
|0.0  |The design is very odd as the ear clip is not very comfortable at all                                         |
|1.0  |Highly recommend for any one who has a blue tooth phone                                                       |
|0.0  |I advise EVERYONE DO NOT BE FOOLED!                                                                           |
|1.0  |So Far So Good!                                                                                               |
|1.0  |Works great!                                                                                                  |
|0.0  |It clicks into place in a way that makes you wonder how long that mechanism would last                       |
+-----+--------------------------------------------------------------------------------------------------------------+
only showing top 20 rows

root
 |-- label: double (nullable = true)
 |-- text: string (nullable = true)
```

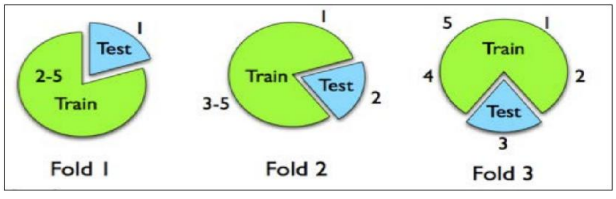
Фиг. 5.13: Множество от Данни на Amazon с неговата Схема

```
+-----+--------------------------------------------------------------------------------------------------------------+
|label|text                                                                                                          |
+-----+--------------------------------------------------------------------------------------------------------------+
|1.0  |Wow Loved this place                                                                                          |
|0.0  |Crust is not good                                                                                             |
|0.0  |Not tasty and the texture was just nasty                                                                      |
|1.0  |Stopped by during the late May bank holiday off Rick Steve recommendation and loved it                        |
|1.0  |The selection on the menu was great and so were the prices                                                    |
|0.0  |Now I am getting angry and I want my damn pho                                                                 |
|0.0  |Honeslty it didn't taste THAT fresh)                                                                          |
|0.0  |The potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer|
|1.0  |The fries were great too                                                                                      |
|1.0  |A great touch                                                                                                 |
|1.0  |Service was very prompt                                                                                       |
|0.0  |Would not go back                                                                                             |
|0.0  |The cashier had no care what so ever on what I had to say it still ended up being wayyy overpriced           |
|1.0  |I tried the Cape Cod ravoli chickenwith cranberrymmmm!                                                        |
|0.0  |I was disgusted because I was pretty sure that was human hair                                                 |
|0.0  |I was shocked because no signs indicate cash only                                                             |
|1.0  |Highly recommended                                                                                            |
|0.0  |Waitress was a little slow in service                                                                         |
|0.0  |This place is not worth your time let alone Vegas                                                             |
|0.0  |did not like at all                                                                                           |
+-----+--------------------------------------------------------------------------------------------------------------+
only showing top 20 rows

root
 |-- label: double (nullable = true)
 |-- text: string (nullable = true)
```

Фиг. 5.14: Множество от Данни на Yelp с неговата Схема

Пресечено-потвърждение (Cross-validation)

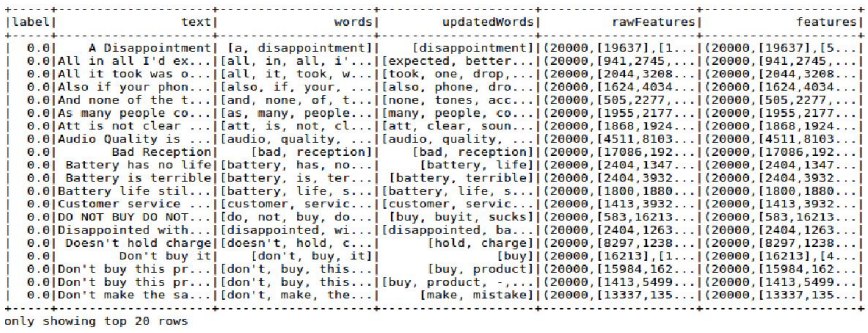
Пресеченото-потвърждение разделя множеството на k подмножества с почти еднакви размери, например петте подмножества показани на фигура 5.15. Първо подмножествата 2-5 се използват за обучение на модела, а подмножество 1 за тестване на обучения модел. Процедурата се повтаря пет пъти, като всеки път едната част се оставя за тестване. Накрая грешката се осреднява по петте повторения.



Фиг. 5.15: Пресеченото-потвърждение K-Fold

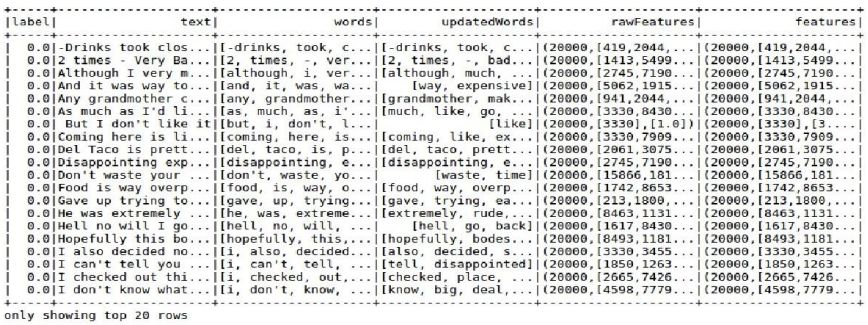
Предварително обработени множества от данни:

В началото, множествата се зареждат в системата от местната файлова система и се разделят случайно на **80 %** тренировъчни данни и **20 %** тестови данни. След това над тренировъчното множество се изпълнява предварителна обработка или почистване чрез **NLP** концепции (екскрактори на характеристики и преобразуватели). Входния низ (стринг) се преобразува в малки букви и се разделя на думи (tokens) чрез използването на интервалите като за разделител. Елиминират се ненужните думи. Понеже данните все още не са в подходящ формат за използване от алгоритмите за машинно обучение е нужно да се създаде Вектор на Характеристиките за всяко изречение в множеството. За тази цел се използва TF-IDF, който се прилага над чантата-с-думи. След това се прилагат алгоритми за анализ на мнение, използвайки чантата-с-думи и конвейерната конфигурация ML pipeline. Резултатите от стъпките с предварителната обработка от всички концепции и за двете множества от данни са показани съответно на фигура 5.16 и 5.17.



```
+-----+--------------------+--------------------+--------------------+--------------------+--------------------+
|label|                text|               words|        updatedWords|         rawFeatures|            features|
+-----+--------------------+--------------------+--------------------+--------------------+--------------------+
|  0.0|    A Disappointment| [a, disappointment]|    [disappointment]|(20000,[19637],[1...|(20000,[19637],[5...|
|  0.0|All in all I'd ex...|[all, in, all, i'...|[expected, better...|(20000,[941,2745,...|(20000,[941,2745,...|
|  0.0|All it took was o...|[all, it, took, w...|[took, one, drop,...|(20000,[2044,3208...|(20000,[2044,3208...|
|  0.0|Also if your phon...|[also, if, your, ...|[also, phone, dro...|(20000,[1624,4034...|(20000,[1624,4034...|
|  0.0|And none of the t...|[and, none, of, t...|[none, tones, acc...|(20000,[505,2277,...|(20000,[505,2277,...|
|  0.0|As many people co...|[as, many, people...|[many, people, co...|(20000,[1955,2177...|(20000,[1955,2177...|
|  0.0|Att is not clear ...|[att, is, not, cl...|[att, clear, soun...|(20000,[1868,1924...|(20000,[1868,1924...|
|  0.0|Audio Quality is ...|[audio, quality, ...|[audio, quality, ...|(20000,[4511,8103...|(20000,[4511,8103...|
|  0.0|       Bad Reception|    [bad, reception]|    [bad, reception]|(20000,[17086,192...|(20000,[17086,192...|
|  0.0| Battery has no life|[battery, has, no...|    [battery, life]|(20000,[2404,1347...|(20000,[2404,1347...|
|  0.0| Battery is terrible|[battery, is, ter...| [battery, terrible]|(20000,[2404,3932...|(20000,[2404,3932...|
|  0.0|Battery life stil...|[battery, life, s...|[battery, life, s...|(20000,[1800,1880...|(20000,[1800,1880...|
|  0.0|Customer service ...|[customer, servic...|[customer, servic...|(20000,[1413,3932...|(20000,[1413,3932...|
|  0.0|DO NOT BUY DO NOT...|[do, not, buy, do...| [buy, buyit, sucks]|(20000,[583,16213...|(20000,[583,16213...|
|  0.0|Disappointed with...|[disappointed, wi...|[disappointed, ba...|(20000,[2404,1263...|(20000,[2404,1263...|
|  0.0| Doesn't hold charge|[doesn't, hold, c...|      [hold, charge]|(20000,[8297,1238...|(20000,[8297,1238...|
|  0.0|        Don't buy it|    [don't, buy, it]|             [buy]|(20000,[16213],[1...|(20000,[16213],[4...|
|  0.0|Don't buy this pr...|[don't, buy, this...|     [buy, product]|(20000,[15984,162...|(20000,[15984,162...|
|  0.0|Don't buy this pr...|[don't, buy, this...|[buy, product, -,...|(20000,[1413,5499...|(20000,[1413,5499...|
|  0.0|Don't make the sa...|[don't, make, the...|     [make, mistake]|(20000,[13337,135...|(20000,[13337,135...|
+-----+--------------------+--------------------+--------------------+--------------------+--------------------+
only showing top 20 rows
```

Фиг. 5.16: Резултати от предварителната обработка на множеството от данни на Amazon



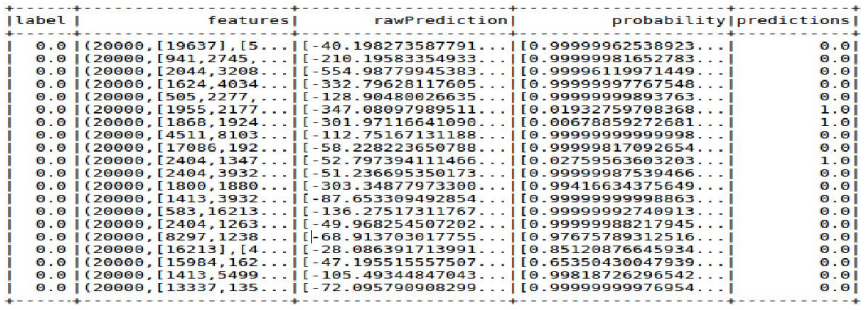
```
+-----+--------------------+--------------------+--------------------+--------------------+--------------------+
|label|                text|               words|        updatedWords|         rawFeatures|            features|
+-----+--------------------+--------------------+--------------------+--------------------+--------------------+
|  0.0|-Drinks took clos...|[-drinks, took, c...|[-drinks, took, c...|(20000,[419,2044,...|(20000,[419,2044,...|
|  0.0|2 times - Very Ba...|[2, times, -, ver...|[2, times, -, bad...|(20000,[1413,5499...|(20000,[1413,5499...|
|  0.0|Although I very m...|[although, i, ver...|[although, much, ...|(20000,[2745,7190...|(20000,[2745,7190...|
|  0.0|And it was way to...|[and, it, was, wa...|    [way, expensive]|(20000,[5062,1915...|(20000,[5062,1915...|
|  0.0|Any grandmother c...|[any, grandmother...|[grandmother, mak...|(20000,[941,2044,...|(20000,[941,2044,...|
|  0.0|As much as I'd li...|[as, much, as, i'...|[much, like, go, ...|(20000,[3330,8430...|(20000,[3330,8430...|
|  0.0| But I don't like it|[but, i, don't, l...|              [like]|(20000,[3330],[1.0])|(20000,[3330],[3....|
|  0.0|Coming here is li...|[coming, here, is...|[coming, like, ex...|(20000,[3330,7909...|(20000,[3330,7909...|
|  0.0|Del Taco is prett...|[del, taco, is, p...|[del, taco, prett...|(20000,[2061,3075...|(20000,[2061,3075...|
|  0.0|Disappointing exp...|[disappointing, e...|[disappointing, e...|(20000,[2745,7190...|(20000,[2745,7190...|
|  0.0|Don't waste your ...|[don't, waste, yo...|      [waste, time]|(20000,[15866,181...|(20000,[15866,181...|
|  0.0|Food is way overp...|[food, is, way, o...|[food, way, overp...|(20000,[1742,8653...|(20000,[1742,8653...|
|  0.0|Gave up trying to...|[gave, up, trying...|[gave, trying, ea...|(20000,[213,1800,...|(20000,[213,1800,...|
|  0.0|He was extremely ...|[he, was, extreme...|[extremely, rude,...|(20000,[8463,1131...|(20000,[8463,1131...|
|  0.0|Hell no will I go...|[hell, no, will, ...|    [hell, go, back]|(20000,[1617,8430...|(20000,[1617,8430...|
|  0.0|Hopefully this bo...|[hopefully, this,...|[hopefully, bodes...|(20000,[8493,1181...|(20000,[8493,1181...|
|  0.0|I also decided no...|[i, also, decided...|[also, decided, s...|(20000,[3330,3455...|(20000,[3330,3455...|
|  0.0|I can't tell you ...|[i, can't, tell, ...|[tell, disappointed]|(20000,[1850,1263...|(20000,[1850,1263...|
|  0.0|I checked out thi...|[i, checked, out,...|[checked, place, ...|(20000,[2665,7426...|(20000,[2665,7426...|
|  0.0|I don't know what...|[i, don't, know, ...|[know, big, deal,...|(20000,[4598,7779...|(20000,[4598,7779...|
+-----+--------------------+--------------------+--------------------+--------------------+--------------------+
only showing top 20 rows
```

Фиг. 5.17: Резултати от предварителната обработка на множеството от данни на Yelp

Резултати

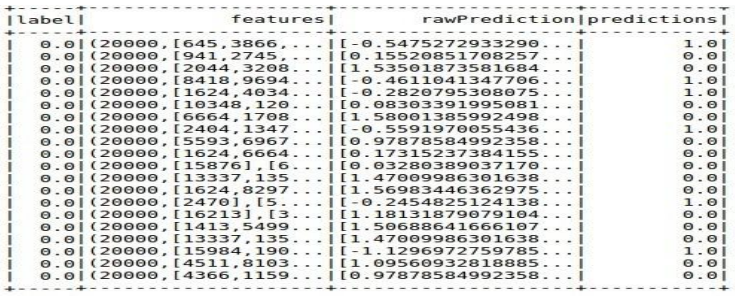
1. Етикирано множество от данни на Amazon: Резултатите от двата алгоритъма за машинно обучение за текст анализ на множеството от данни amazon_cell_labelled са показани на фигура **5.18** и **5.19**, съответно. Тези резултати са крайните резултати от тестването на обучените модели с тестовото множество. За обучаване на моделите е използвано тренировъчното множество. Моделът Naïve Bayes изчислява вероятностите и генерира предсказания. Моделът LinearSVM генерира само предсказания. Дали дадено

предсказание е вярно или не, може лесно да се провери като се сравни стойността му с тази на етикета (label).



label	features	rawPrediction	probability	predictions
0.0	(20000,[19637],[5...	[-40.198273587791...	[0.99999962538923...	0.0
0.0	(20000,[941,2745,...	[-210.19583354933...	[0.99999981652783...	0.0
0.0	(20000,[2044,3208...	[-554.98779945383...	[0.99996119971449...	0.0
0.0	(20000,[1624,4034...	[-332.79628117605...	[0.99999997767548...	0.0
0.0	(20000,[505,2277,...	[-128.90480026635...	[0.99999999893763...	0.0
0.0	(20000,[1955,2177...	[-347.08097895511...	[0.01932759708369...	1.0
0.0	(20000,[1868,1924...	[-301.97116641090...	[0.00678859272681...	1.0
0.0	(20000,[4511,8103...	[-112.75167131188...	[0.99999999999998...	0.0
0.0	(20000,[17086,192...	[-58.228223650788...	[0.99999817092654...	0.0
0.0	(20000,[2404,1347...	[-52.797394111466...	[0.02759563603203...	1.0
0.0	(20000,[2404,3932...	[-51.236695350173...	[0.99999987539466...	0.0
0.0	(20000,[1800,1880...	[-303.34877973300...	[0.99416634375649...	0.0
0.0	(20000,[1413,3932...	[-87.653309492854...	[0.99999999998863...	0.0
0.0	(20000,[583,16213...	[-136.27517311767...	[0.99999992740913...	0.0
0.0	(20000,[2404,1263...	[-49.968254507202...	[0.99999988217945...	0.0
0.0	(20000,[8297,1238...	[-68.913703017755...	[0.97675789312516...	0.0
0.0	(20000,[16213],[4...	[-28.086391713991...	[0.85120876645934...	0.0
0.0	(20000,[15984,162...	[-47.195515557507...	[0.65350430047939...	0.0
0.0	(20000,[1413,5499...	[-105.49344847043...	[0.99818726296542...	0.0
0.0	(20000,[13337,135...	[-72.095790908299...	[0.99999999976954...	0.0

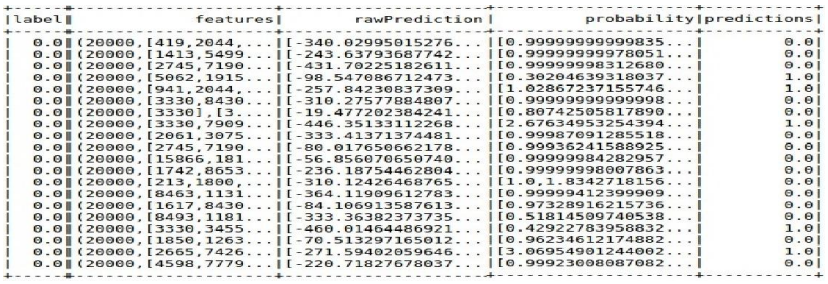
Фиг. 5.18: Резултати от Анализ на Мнение с Naïve Bayes върху данните на Amazon



label	features	rawPrediction	predictions
0.0	(20000,[645,3866,...	[-0.5475272933290...	1.0
0.0	(20000,[941,2745,...	[0.15520851708257...	0.0
0.0	(20000,[2044,3208...	[1.53501873581684...	0.0
0.0	(20000,[8418,9694...	[-0.4611041347706...	1.0
0.0	(20000,[1624,4034...	[-0.2820795308075...	1.0
0.0	(20000,[10348,120...	[0.08303391995081...	0.0
0.0	(20000,[6664,1708...	[1.58001385992498...	0.0
0.0	(20000,[2404,1347...	[-0.5591970055436...	1.0
0.0	(20000,[5593,6967...	[0.97878584992358...	0.0
0.0	(20000,[1624,6664...	[0.17315237384155...	0.0
0.0	(20000,[15876],[6...	[0.03280389037170...	0.0
0.0	(20000,[13337,135...	[1.47009986301638...	0.0
0.0	(20000,[1624,8297...	[1.56983446362975...	0.0
0.0	(20000,[2470],[5...	[-0.2454825124138...	1.0
0.0	(20000,[16213],[3...	[1.18131879079104...	0.0
0.0	(20000,[1413,5499...	[1.50688641666107...	0.0
0.0	(20000,[13337,135...	[1.47009986301638...	0.0
0.0	(20000,[15984,190...	[-1.1296972759785...	1.0
0.0	(20000,[4511,8103...	[1.09560932818885...	0.0
0.0	(20000,[4366,1159...	[0.97878584992358...	0.0

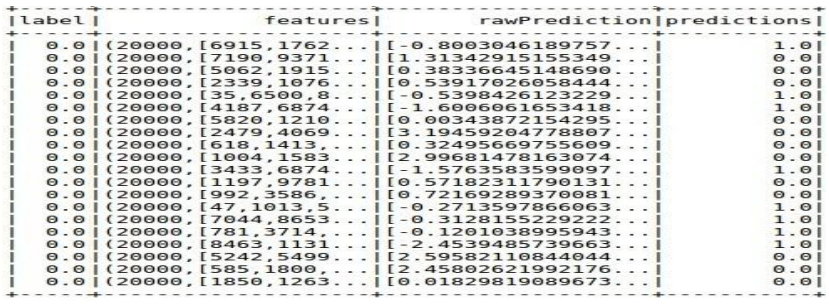
Фиг. 5.19: Резултати от Анализ на Мнение с **Линеен SVM** върху данните на Amazon

2. Етикирано множество от данни на Yelp: Резултатите от двата алгоритъма за машинно обучение за текст анализ на множеството от данни yelp_labelled са показани на фигура **5.20** и **5.21**, съответно.



label	features	rawPrediction	probability	predictions
0.0	(20000,[419,2044,...	[-340.02995015276...	[0.99999999999835...	0.0
0.0	(20000,[1413,5499...	[-243.63793687742...	[0.99999999978051...	0.0
0.0	(20000,[2745,7190...	[-431.70225182611...	[0.99999998312680...	0.0
0.0	(20000,[5062,1915...	[-98.547086712473...	[0.30204639318037...	1.0
0.0	(20000,[941,2044,...	[-257.84230837309...	[1.02867237155746...	1.0
0.0	(20000,[3330,8430...	[-310.27577884807...	[0.99999999999998...	0.0
0.0	(20000,[3330],[3,...	[-19.477202384241...	[0.80742595817890...	0.0
0.0	(20000,[3330,7909...	[-446.35133112268...	[2.67634953254394...	1.0
0.0	(20000,[2061,3075...	[-333.41371374481...	[0.99987091285518...	0.0
0.0	(20000,[2745,7190...	[-80.017650662178...	[0.99936241588925...	0.0
0.0	(20000,[15866,181...	[-56.856070650740...	[0.99999984282957...	0.0
0.0	(20000,[1742,8653...	[-236.18754462804...	[0.99999998007863...	0.0
0.0	(20000,[213,1800,...	[-310.12426468765...	[1.0,1.8342718156...	0.0
0.0	(20000,[8463,1131...	[-364.11909612783...	[0.99999412399909...	0.0
0.0	(20000,[1617,8430...	[-84.106913587613...	[0.97328916215736...	0.0
0.0	(20000,[8493,1181...	[-333.36382373735...	[0.51814509740538...	0.0
0.0	(20000,[3330,3455...	[-460.01464486921...	[0.42922783958832...	1.0
0.0	(20000,[1850,1263...	[-70.513297165012...	[0.96234612174882...	0.0
0.0	(20000,[2665,7426...	[-271.59402059646...	[3.06954901244002...	1.0
0.0	(20000,[4598,7779...	[-220.71827678037...	[0.99923008087082...	0.0

Фиг. 5.20: Резултати от Анализ на мнение с Naïve Bayes върху данните на Yelp



label	features	rawPrediction	predictions
0.0	(20000,[6915,1762...	[-0.8003046189757...	1.0
0.0	(20000,[7190,9371...	[1.313429155155349...	0.0
0.0	(20000,[5062,1915...	[0.383366451486900...	0.0
0.0	(20000,[2339,1076...	[0.539170260584440...	0.0
0.0	(20000,[35,6500,8...	[-0.539842612322900...	1.0
0.0	(20000,[4187,6874...	[-1.600606616534180...	1.0
0.0	(20000,[5820,1210...	[0.003438721542950...	0.0
0.0	(20000,[7479,4069...	[3.194559204778880...	0.0
0.0	(20000,[618,1413,...	[0.324956697556090...	0.0
0.0	(20000,[1004,1583...	[2.996814781630740...	0.0
0.0	(20000,[3433,6874...	[-1.576358359909700...	1.0
0.0	(20000,[1197,9781...	[0.571823117901310...	0.0
0.0	(20000,[992,3586,...	[0.721692893700810...	0.0
0.0	(20000,[47,1013,5...	[-0.271359786606300...	1.0
0.0	(20000,[7044,8653...	[-0.312815522922200...	1.0
0.0	(20000,[781,3714,...	[-0.120103899959430...	1.0
0.0	(20000,[8463,1131...	[-2.453948573966300...	1.0
0.0	(20000,[5242,5499...	[2.595821108440440...	0.0
0.0	(20000,[585,1800,...	[2.458026219921760...	0.0
0.0	(20000,[1850,1263...	[0.018298190896730...	0.0

Фиг. 5.21: Резултати от Анализ на Мнение с **Линеен SVM** върху данните Yelp

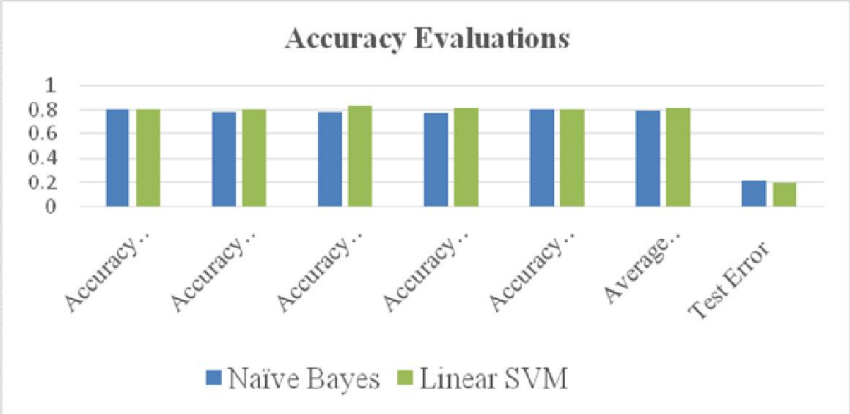
Оценка на Моделите

Обикновено, метриката за оценка на представянето на модела зависи от задачата за машинно обучение. Различни метрики се използват за регресия, класификация, групирани (clustering), и формиране на препоръки. В настоящата работа е използван оценител за многокласов класификатор, който оценява точността на предсказанията и ефективността на модела. Този оценител изисква като вход рамка с данни, а връща скалар. Рамката с данни, която се подава като аргумент трябва да има колони означени *етикет* (label) и *предсказание* (prediction). Точността е прост оценителен метод. Използва се като оценителна метрика за различните модели и се дефинира като процента на успешно предсказаните етикети. Например, ако тестово множество от данни има 100 наблюдения и моделът правилно предсказва 85 от тях, точността му е 85 %.

В тази работа, за да оценим как моделът се представя на тренировъчното и на тестовото множество, първо се получават предсказанията за всяко наблюдение в тренировъчното и тестовото множество. След това се изчислява точността на модела за тестовото множество. В допълнение, за по-добра оценка на представянето , се извършва k-кратно пресечено-потвърждение, описано по-горе, с k=5. Накрая се изчислява средната стойност на точността на предсказанията. Както се вижда от таблици 5.5, 5.6 и фигури 5.22, 5.23, за конкретните две множества от данни (на Amazon и на Yelp), Линейния SVM се представя малко по-добре от към точност на предсказанията от Naïve Bayes (точност 1 означава 100% правилно предсказани етикета).

Техники:	**Naïve Bayes**	**Линеен SVM**
Точност на Предсказанието, Измерване 1	0.8	0.79803
Точност на Предсказанието, Измерване 2	0.78421	0.7973
Точност на Предсказанието, Измерване 3	0.77941	0.83333
Точност на Предсказанието, Измерване 4	0.77251	0.80729
Точност на Предсказанието, Измерване 5	0.80303	0.8
Стедан Точност на Предсказанието	0.78783	0.80719
Грешка	0.21217	0.19281

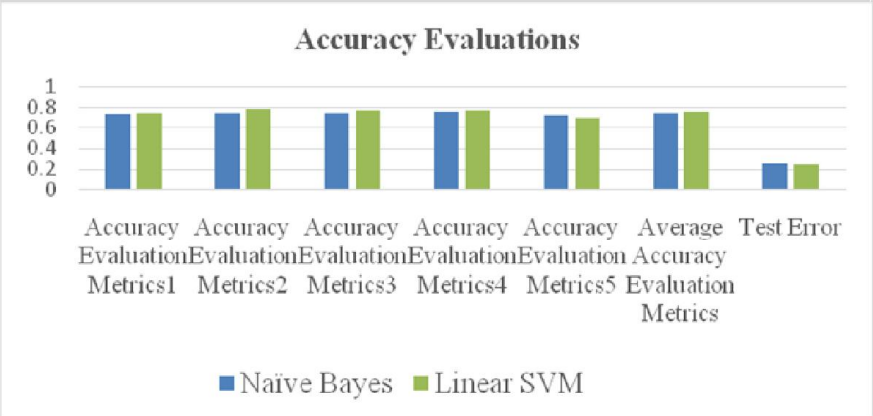
Таблица 5.5 Точност на двете техники за данните на Amazon



Фиг. 5.22: Точност на двете техники за данните на Amazon

Техники:	**Naïve Bayes**	**Линеен SVM**
Точност на Предсказанието, Измерване 1	0.72959	0.73869
Точност на Предсказанието, Измерване 2	0.75	0.78218
Точност на Предсказанието, Измерване 3	0.74257	0.76705
Точност на Предсказанието, Измерване 4	0.75122	0.7732
Точност на Предсказанието, Измерване 5	0.72414	0.70233
Стедан Точност на Предсказанието	0.7395	0.75269
Грешка	0.2605	0.24731

Таблица 5.6 Точност на двете техники за данните на Yelp



Фиг. 5.23 Точност на двете техники за данните на Yelp

Изводи към Глава 5

- Предложена е софтуерна рамка за облачно базирана хибридна система за препоръки, (Framework for Cloud Based Hybrid Recommender System) за извличане на информация от големи данни и са дискутирани методите и алгоритмите използвани в системата. Първо са използвани алгоритми за съвместно филтриране, което традиционно е най-често срещания подход, а след това е алгоритъмът за Анализ на Мнение, което касае обработка на естествен (натурален) език и текстови анализ за извличане на информация от различни материали и източници. Извличането на информация от социални медии се превръща в много важна част от работата на компании и бизнеси в процеса на формиране на препоръки.
- В предложената рамка, различни подходи са комбинирани за формиране на две категории от хибридните системи за препоръки. Комбинацията на резултати от два алгоритъма, по начина предложен в настоящата работа, дава по-точни предсказания, което е предпоставка за по-добро бизнес разузнаване. Тази комбинация на методи е нова и неизследвана област и може да помогне за повишаване на конкурентноспособността на компании и предприятия.

- Анализ на мнение от големи данни е една от най-интересните техники за откриване на мнението на потребителите за дадени продукти. Предложената и оценена в тази глава система може да извършва анализ на мнение на големи масиви от данни с отзиви с висока скорост във време близко до реално-време. Работата на предложения модел минава през следните етапи: предварителна обработка, генериране на характеристики, обучаване на класификаторите, и Pipeline етап..
- Направена е експериментална оценка на представянето на всички алгоритми (методи) на платформата върху избрани множества от данни. С реализирането на алгоритъма за съвместно филтриране са преодолени трите главни предизвикателства на системите за препоръки при големи множества от данни: студен старт (cold start), разреденост (sparsity) и мащабируемост (scalability). Демонстрирано е, че анализът на мнение се изпълнява достатъчно бързо, така че големите данни да могат да бъдат обработвани ефективно. Експериментално е потвърдено, използвайки крос-валидация, че точността на класификация е добра.
- Резултатите показани от алгоритмите за контролирано машинно обучение върху избрани множества от данни са добри и показват, че алгоритмите могат успешно да бъдат използвани за правене на предсказания и издаване на препоръки, които да помогнат на компании и бизнеси при взимането на по-добри решения и така да повишат конкурентноспособност им.
- Всички алгоритми за тази платформа са написани на език за програмиране Java. Самата платформа е реализирана чрез използване на библиотеки за машинно обучение, такива като Mahout на Apache Hadoop и Spark ML на Apache Spark.

Препоръки за Бъдеща Работа

Алгоритмите за Контролирано Машинно Обучение се реализират чрез библиотеки за машинно обучение и помагат на компаниите да извличат знания от големи обеми данни. В настоящата работа е използван единичен възел (машина) за прилагане на предложената платформа към избрани множества от данни (рейтинги и етикетирани отзиви), тъй като разпределената среда е скъпа. С други думи, настоящата работа е реализирана на единичен възел, и въпреки, че предвижданията са, че платформата ще работи много по-добре на многовъзлова (многомашинна) конфигурация от нивото на използваните от компании и бизнеси, работата и трябва да се тества и в такава среда с много по-големи множества от данни. Това е препоръка за бъдещо развитие на настоящия изследователски проект.

ПРИНОСИ КЪМ ДИСЕРТАЦИЯТА

1. Анализирани са извличането на информация от големи данни и различни предизвикателства и проблеми свързани с големи данни. Изяснени са техниките за обработка и съхранение на големи данни и предизвикателствата, свързани с тях. Анализирани са възможностите на известните платформи за обработка и извличане на информация от големите данни, както и на алгоритмите за машинно обучение и библиотеките за машинно обучение, ориентирани към работа с големи данни.
2. Разработени са алгоритми за съвместно филтриране за генериране на препоръки от големи масиви с данни на основата на най-популярните техники за съвместно филтриране, а именно: базирани на потребител и базирани на продукт от техниките, базирани на памет и ALS от техниките базирани на модел.
3. Разработени са алгоритми за анализ на мнение за текстов анализ и предсказване на тенденции от големи етикирани множества от данни. За целта са използвани и адаптирани два от най-известните класификационни алгоритми: Naïve Bayes и Support Vector Machine.
4. Разработени са два типа Хибридни Системи за Препоръки, които са нови и различни от съществуващите хибридни системи. В първия тип са използвани само алгоритми за съвместно филтриране, а системата за препоръки е базирана на каскадна хибридизация. Във втория тип са използвани алгоритми за анализ на мнение, а системата за препоръки е базирана на смесена хибридизация.
5. Разработените по-горе нови алгоритми за извличане на знания от големи данни са обединени в софтуерна платформа, наречена облачно базирана хибридна система за препоръки (Cloud Based Hybrid Recommender System) за големи данни. Алгоритмите на предложената платформа са реализирани чрез използването на мащабируемите библиотеки за машинно обучение Apache Mahout и Apache Spark.
6. Проведени са експерименти с цел оценяване представянето на всички алгоритми и модели на предложената платформа. За оценка на работата на алгоритмите са използвани средноквадратична грешка (RMSE) и известни критерии (крос-валидация и др.) за оценка на различните модели и системи за препоръки. Резултатите от оценката, базирани на известни комплекти от данни (с известни етикирани данни) потвърждават, че разработените хибридни системи дават по-точни препоръки.
7. Преодоляни са трите главни предизвикателства при системите за препоръки, а именно: студен старт (cold start), разреденост (sparsity), и мащабируемост (scalability). За по-добри резултати са комбинирани различни подходи. Комбинирането на резултатите на два от алгоритмите, по начина предложен в настоящата работа, подобрява точността

на крайните резултати и следователно води до по-добра бизнес интелигентност. Предложената комбинация е новаторска и не е изследвана до сега. По-точните резултати са предпоставка за повишаване конкурентноспособността на предприятията.

8. Настоящата работа хвърля светлина върху нуждата от извличане на знания от големи данни. Използването на знания, извлечени от големи данни с помощта на разработената софтуерна платформа би имало значим ефект върху работата на компании и предприятия по отношение на взимане на бързи и адекватни решения, оптимизация на работата, спестяване на ресурси, пари и време и др., и като цяло би довело до, повишаване на тяхната конкурентноспособност.

Авторски публикации по дисертацията:

1. Al-Barznji Kamal, Atanassov Atanas, Comparison of memory based filtering techniques for generating recommendations on large data, Journal "Engineering and Automation Problems", IF 0.133, Volume 1. pp. 44-50, Moskva, Russia, **2018**, ISSN 0234-6206
2. K. Al-Barznji, A. Atanassov, Review Of Big Data And Big Data Mining For Adding Big Value To Enterprises, Science, Engineering & Education,2, (1), UCTM-Sofia, Bulgaria, pp. 50-57, **2017**.
3. А. Атанасов, K. Al-Barznji, Особености и предизвикателства при извличането на големи данни, приета за печат в списание Автоматика и Информатика, **2017**, София **,** ISSN 0861-7562.
4. K. Al-Barznji, A.V. Atanassov, A Framework for Cloud based Hybrid Recommender System (FCHRS) for Big Data Mining, XXV Международен симпозиум управление на енергийни, индустриални и екологични системи, 11-12 май **2017** г, Банкя, Bulgaria, Proceedings pp.87-91.
5. K. Al-Barznji, Atanassov A. A Mapreduce Solution For Handling Large Data Efficiently, 13th INTERNATIONAL CONGRESS "MACHINES, TECHNOLOGIES, MATERIALS"14-17.09.2016, Varna, Bulgaria. Published in an International virtual journal for science, techniques and innovation for the industry MTM, YEAR X, Issue 12 **/2016**, pp. 20- 23, ISSN 1313-0226.