

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

Батырбекқызы Маралым

«Дәріханалар бойынша мобильдік қолданба-агрегаторға Data Science
шешімдерін әзірлеу»

Дипломдық жобаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

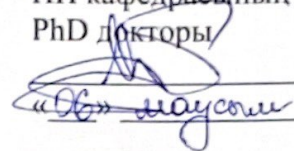
СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

ҚОРҒАУҒА ЖІБЕРІЛДІ

ПИ кафедрасының меңгерушісі,
PhD докторы

 М. Тұрдалыұлы
«06» маусым 2021 ж.

Дипломдық жобаға
ТҮСІНІКТЕМЕЛІК ЖАЗБА


Тақырыбы: «Дәріханалар бойынша мобильдік қолданба-агрегаторға Data Science шешімдерін әзірлеу»

5B070400 – «Есептеу техникасы және бағдарламалық қамтамасыз ету»
мамандығы

Орындаған:

Батырбекқызы М.

Ғылыми жетекші, техн. ғыл.
магистрі, лектор

 Д.А. Баймбетов
«04» маусым 2021 ж.

Алматы 2021

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

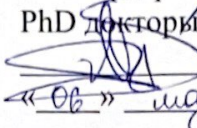
СӘТБАЕВ УНИВЕРСИТЕТІ

Кибернетика және ақпараттық технологиялар институты

«Программалық инженерия» кафедрасы

БЕКІТЕМІН

ПИ кафедрасының меңгерушісі,
PhD докторы

 М. Тұрдалыұлы
«06» маусым 2021 ж.

Дипломдық жобаны орындауға

ТАПСЫРМА

Білім алушыға Батырбекқызы Маралым

Тақырып: «Дәріханалар бойынша мобильдік қолданба-агрегатор» бойынша Data Science шешімдерін әзірлеу

Университет ректоры бұйрығының №2131-б «24» қараша 2021 ж. шешімімен бекітілген.

Орындалған жобаның өткізу мерзімі: «08» маусым 2021 ж.

Дипломдық жобаның бастапқы мәліметтері: Жобаның төлқұжаты, технология бойынша техникалық құжаттама, техникалық құжаттаманы әзірлеу кезінде құрылған жобаның UML диаграммалары, техникалық тапсырма.

Есеп – түсініктеме жазбаның талқылауға берілген сұрақтардың тізімі:

- а) ұсыныс жүйелерін жалпы сипаттау;
- б) жобаны модельдеу және UML диаграммаларын сызу;
- в) Data Science шешімдерін жобалау;
- г) тесттерден өткізу, кітапханалармен жұмыс жасау, бірнеше модельді салыстыру;


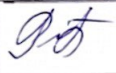
Графикалық материалдар тізімі (міндетті суреттердің нақты көрсетілуімен): презентацияның 20 слайдпен берілген құжат түрінде ұсынылған.

Ұсынылған негізгі әдебиеттер: 9 пайдаланылған әдебиеттер тізімінен.

Дипломдық жобаны орындау
КЕСТЕСІ

Бөлімдердің атаулары, зерттелген мәселелердің тізімі	Ғылыми жетекшіге және кеңесшілерге ұсыну мерзімі	Ескерту
1. Диплом жұмысының жоспарын құру.	11.01.2021	жоқ
2. Тапсырма қойылымы және бағдарламау ортасын таңдау.	15.01.2021	жоқ
3. Зерттеу тақырыбы бойынша ғылыми - теориялық материалдарды жинау және негізгі бөлім бойынша есеп беру жазбасын дайындау.	01.02.2021	жоқ
4. Жобалау бөліміне сызбаларды дайындау.	15.02.2021	жоқ
5. Data Science модельдерін құру, тесттен өткізу.	22.03.2021	жоқ
6. Дипломдық жобаға түсіндірме жазба жазу.	26.04.2021	жоқ

Дипломдық жұмыс бөлімдерінің кеңесшілерінің аяқталған жұмысқа қойылған қойған қолтаңбалары

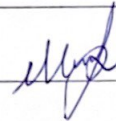
Бөлімдер атауы	Кеңес берушілер (аты-жөні, тегі, ғылыми дәрежесі, атағы)	Қолтаңба қойылған мерзімі	Қолы
Нормалық бақылаушы	Қ. Марғұлан, лектор, техн. ғыл. магистрі	06.06.2021	
Бағдарламалық бөлім	А.Б. Рамазан, ассистент, техн. ғыл. магистрі	31.05.2021	

Ғылыми жетекші



Д.А. Баймбетов

Тапсырманы орындауға қабылдап алған студент



М. Батырбекқызы

Күні

«06» маусым 2021ж.

АҢДАТПА

XXI ғасыр, клиентке бағытталған дәуірде, ұсыныс жүйелерін пайдаланбайтын жоғары сапалы бағдарламалық қолданбаны елестету қиын. Бұл дипломдық жоба дәріханалар агрегатор қолданбасына арналған ұсыныс жүйелерін қамтиды.

Ұсыныс жүйелері қолданушылардың өнімді бағалау тарихына, қолданушының талғамына, элементтердің ұқсастығы мен қолданушылардың ұқсастығына негізделіп, соған байланысты қажетті өнімді қажетті уақытта ұсынады. Ұсыныс жүйелерін жасау арқылы дәріхана қолданба-агрегатордың кірістерін, таңдау рейтингісін және тіпті клиенттердің қанағаттанушылығын арттыруға болады.

АННОТАЦИЯ

В XXI веке, в эпохе клиентоориентированности, сложно представить себе качественное программное приложение, не использующее системы рекомендаций. Данный дипломный проект содержит рекомендательные системы для применения аптечных агрегаторов.

Системы рекомендаций основаны на истории оценки продукта пользователями, вкусах пользователя, сходстве элементов и пользователей, и в зависимости от этого, предоставляют необходимый продукт в нужное время. Создавая системы рекомендаций, можно повысить доход аптек в аптечном приложении-агрегатора, рейтинг выбора и даже удовлетворенность клиентов.

ABSTACT

In the 21st century, in the era of customer focus, it is difficult to imagine a high-quality software application that does not use a recommendation system. This diploma project contains recommendation systems for the use of pharmacy aggregators.

Recommendation systems are based on the history of user evaluation of the product, the user's tastes, the similarity of elements and users, and depending on this, provide the necessary product at the right time. By creating recommendation systems, you can increase the revenue of pharmacies in the pharmacy aggregator app, the selection rating, and even customer satisfaction.

МАЗМҰНЫ

	Кіріспе	7
1	Негізгі бөлімі	8
1.1	Ұсыныс жүйелерінің сипаттамасы	8
1.2	Дерек көздері	9
1.3	Ұсыну жүйелерінің түрлері	10
1.4	Ұсыныс жүйелерінде кездесетін қиындықтар	11
1.5	Бағалау көрсеткіштері	12
2	Жобалау бөлімі	14
2.1	UML тілі	14
2.2	Бағдарламалау ортасы	17
3	Зерттеу бөлімі	19
3.1	Pandas	19
3.2	Numpy	20
3.3	Дерек қоры	20
3.4	TF-IDF алгоритмі	22
3.5	Cosine Similarity	22
3.6	Бағалау метрикалары	24
3.7	Content Based Recommender System кодына шолу	25
3.8	Item Based Collaborative Recommender System кодына шолу	26
3.9	User Based Collaborative Recommender System кодына шолу	28
	Қорытынды	30
	Пайдаланылған әдебиеттер тізімі	31
	А Қосымшасы. Техникалық тапсырма	32
	Б Қосымшасы. Бағдарлама мәтіні	34
	Сипаттізім	40

КІРІСПЕ

Ұсыныс жүйелері – қолданушыға пайдалы болуы мүмкін элементтер бойынша кеңес беретін программалық құрылғы. Көбінесе ұсыныс жүйелері әуен, кітап, фильм, сатылатын тауарлар контентінде қолданылады.

Қазіргі таңда технологиялар қарқынды даму үстінде, әсіресе ақпараттық. Соған байланысты желіде өте ауқымды мөлшерде қолжетімді таңдау бар. Оның әрқайсысын зерттеу мүмкін емес.

Егер қолданушы өзінің бір әйгілі электрондық коммерция веб-сайтында қол жетімді бір миллиард өнімдердің каталогын ақтаруға бір секундтан құртса, олардың бүкіл каталогын көру үшін шамамен 30 жыл – яғни, өмірінің жартысы қажет болады. Осы орайда, ұсыныс жүйелері соңғы жылдарда өзін шамадан тыс ақпараттың шешімі болып бой көрсетті.

Ең танымал ұсыныс жүйелерін Amazon, Netflix, Youtube, IMDB секілді танымал сайттарда көруге болады.

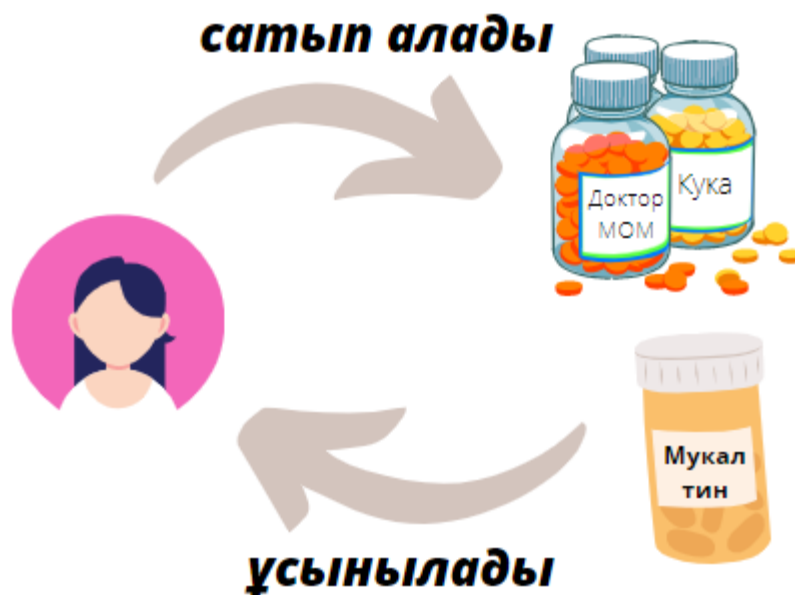
Ұсыныс жүйелерін мобильдік қолданба-агрегаторға енгізудің мақсаты: қолданушылар үшін сапалы ұсыныс жасау арқылы пайдалы ақпаратпен қамтамасыз ету.

Ұсыныс жүйелерін мобильдік қолданба-агрегаторға енгізудің өзектілігі: қолданушылар мен дәріханалар үшін де пайдалы өніммен қайтамасыз ету. Ол қолданушылар үшін қажетті болуы мүмкін дәрі-дәрмекті тауып, ұсыну арқылы қолданушының іздеу және таңдау қиындығынан босатып, дәріханадан ұсынылған дәрілерді сатып алу және қолданбамен үнемі қолдану ықтималдығын жоғарылатады.

1 Негізгі бөлім

1.1 Ұсыныс жүйелерінің сипаттамасы

Ұсыныс жүйелері – қолданушыларға өздеріне әртүрлі ықпал әсерінен ұсынылған жаңа өнімдерді ашуға арналған жүйелер [1]. Біраз уақыт қолданба немесе сайтпен қолдана келе, адамды қызықтыратын өнімдер өздері іздеп алдынан шыға бастайды. Мысалын төмендегі 1.1-суретте көре аламыз:



1.1-сурет – Ұсыныс жүйесіне мысал

Ұсыныс жүйесі келген ақпараттың үлкен көлемімен жұмыс істейді, пайдаланушыдан түскен мәліметтерге және пайдаланушының қалауы мен мүдделерін ескеретін басқа факторларға негізделген ең маңызды ақпараттарды сүзеді. Ол пайдаланушы мен элемент арасындағы сәйкестікті анықтайды және ұсыныс үшін пайдаланушылар мен элементтер арасындағы ұқсастығын есептейді.

Ұсыныс жүйелерін қолданатын компаниялар жекелендірілген ұсыныстар мен клиенттерге қызмет көрсету сапасын жақсарту арқылы сатылымды арттырады [8]. Пайдаланушының не қалайтынын біле отырып, компания бәсекелестік артықшылыққа ие болады және клиентті бәсекелестің пайдасына жоғалтып алу қаупі азаяды.

Netflix өз пайдаланушыларына фильмдер мен сериалдарды ұсыну үшін ұсыныстар жүйесін қолданады. Сол сияқты, YouTube әртүрлі бейнелерді ұсынады. Бүгінгі таңда кеңінен қолданылатын ұсыныс жүйелерінің көптеген мысалдары бар.

Қорытындыласақ, ұсыныс жүйелері өнімді бағалау немесе тұтынушының қалауын болжай алатын машинаны оқыту алгоритмдерінде жұмыс істейтін

ақпаратты сүзудің тиімді жүйесі және пайдасы жағынан таптырмас құрал, келесі 5 пунктте көрсетілген:

- 1) пайдаланушыларға өздерін қызықтыратын заттарды табуға көмектеседі;
- 2) өнім жеткізушілеріне оларды қажетті пайдаланушыға жеткізуге көмектеседі;
- 3) пайдаланушылар үшін ең маңызды сәйкестендіру өнімдері;
- 4) жеке мазмұн;
- 5) веб-сайттарға пайдаланушылармен қарым-қатынасты жақсартуға көмектеседі.

1.2 Дерек көздері

Ұсыныс жүйелері ұсыныс жасау үшін әртүрлі деректерді белсенді түрде жинайтын ақпаратты өңдеу жүйелері. Деректер, ең алдымен, не ұсынылатынына және осы ұсыныстарды алатын пайдаланушыларға қатысты. Бірақ ұсыныс жүйелері үшін қолжетімді деректер мен білім көздері әр түрлі болуы мүмкін болғандықтан, оларды қолдануға болады ма, жоқ па, ұсыныс техникасына байланысты болады.

Жалпы жіктеу ретінде ұсыныс жүйелері пайдаланатын мәліметтерге элементтер, пайдаланушылар мен транзакциялар, яғни пайдаланушылар мен заттар арасындағы қатынастар жатады.

Элементтер – бұл ұсынылатын нысандар. Элементтерді күрделілігі мен құндылығы немесе пайдалылығы бойынша сипаттауға болады. Ұсыныс жүйелері негізгі технологиясына сәйкес, элементтердің бірқатар қасиеттері мен функцияларын қолдана алады.

Пайдаланушылар. Олар әртүрлі мақсаттар мен сипаттамаларға ие болуы мүмкін. Адамның ұсыныстары мен өзара әрекеттесуін жекелендіру үшін ұсыныс жүйелері пайдаланушылар туралы әртүрлі ақпаратты пайдаланады. Бұл ақпаратты басқаша құрылымдауға болады және қай ақпаратты модельдеу керектігін таңдау, ұсыныс техникасына байланысты болады.

Мәмілелер. Транзакциялар – бұл адам мен компьютердің өзара әрекеттесуі кезінде жасалған және пайдалы болатын маңызды ақпаратты сақтайтын журнал түріндегі мәліметтер. Мысалы, транзакция журналы пайдаланушы таңдаған элементке сілтеме жасай. Егер қол жетімді болса, транзакция пайдаланушы берген нақты кері байланыс болуы мүмкін, мысалы, таңдалған элементтің рейтингі.

Ұсыныс жүйелерін құру үшін деректерді жинаудың екі әдісі бар — анық және жасырын.

Нақты кері байланыс деректері — пайдаланушының өнімге берген нақты саны. Айқын кері байланыстың кейбір мысалдары-Netflix-тегі пайдаланушылардың фильм рейтингтері, Amazon-дағы пайдаланушылардың өнім рейтингтері. Нақты кері байланыс пайдаланушының өнімді қалай ұнатқаны

немесе ұнатпағаны туралы ақпаратты ескереді. Нақты кері байланыс деректері сандық бағалауға жатады.

Жасырын кері байланыс деректері — пайдаланушының мүдделерін тікелей көрсетпейді, бірақ ол пайдаланушының мүдделері үшін прокси ретінде әрекет етеді. Мысалы, қолданушының шолу тарихы, сілтемелерді басу, ән ойнату саны, веб-бетті айналдыру пайызы немесе тінтуірдің қозғалысы жатады.

1.3 Ұсыну жүйелерінің түрлері

1.3.1 Popularity Based Recommender System (Танымалдыққа негізделген ұсыныстар жүйесі)

Бұл танымалдық принципі бойынша жұмыс істейтін ұсыныс жүйесінің түрі. Бұл жүйелер қазіргі уақытта қандай өнімдер немесе фильмдер трендте екенін немесе пайдаланушылар арасында ең танымал екенін тексереді және оларды тікелей ұсынады.

Артықшылықтары:

- а) ол суық іске қосу проблемаларынан зардап шекпейді, яғни жұмыстың бірінші күнінде ол әртүрлі сүзгілері бар өнімдерді ұсына алады;
- б) пайдаланушының тарихи деректері қажет емес.

Кемшіліктері:

- а) жекелендірілмеген;
- б) жүйе барлық басқа пайдаланушылар үшін танымалдыққа негізделген бірдей өнімдерді / фильмдерді ұсынады.

1.3.2 Content Based Recommender System(Мазмұнға негізделген ұсыныстар жүйесі)

Бұл ұсыныс жүйесі қолданушының ұнатқан элементтеріне ұқсас элементтерді ұсынып үйренеді. Элементтердің ұқсастығы олардың қасиеттерінің ұқсастығын өлшеу арқылы анықталады.

Артықшылықтары:

- а) пайдаланушы бойынша көп дерек қажет емес;
- б) суық іске қосудан зардап шекпейді.

Кемшіліктері:

- а) тауарлар туралы деректер жақсы көлемде болуы тиіс;
- б) ұқсастықты есептеу үшін функциялар қол жетімді болуы керек.

1.3.3 Collaborative Filtering Recommender System (Бірлескен сұзу ұсыныс жүйесі)

Белгілі бір пайдаланушының өзара әрекеттесуін ескеретін мазмұнға негізделген ұсыныс жүйесінен айырмашылығы, бірлескен сұзу әдісі жетілдірілген тәсілге сәйкес келеді және пайдаланушы мен элементтің өзара әрекеттесуіне негізделген ұқсас пайдаланушыларды анықтайды. Оның екі түрі бар:

а) пайдаланушы негізінде жақын көршіні бірлесіп сұзу:

Жүйе талғамы бірдей пайдаланушыларды анықтайды және олардың арасындағы ұқсастық мінез-құлқы негізінде есептеледі.

б) элементтер негізінде жақын көршілерді бірлесіп сұзу:

Жүйе пайдаланушы сатып алған тауарларға ұқсас тауарларды тексереді. Әр түрлі элементтер арасындағы ұқсастық болжау үшін пайдаланушылар емес, элементтер негізінде есептеледі.

Артықшылықтары:

а) өнімді сипаттауға қойылатын талаптар жоқ.

Кемшіліктері:

а) жаңа пайдаланушыларды ұсыну қиын және көптеген шолулары бар танымал өнімдерге артықшылық беруге бейім;

б) жаңа шығарылымдарды ұсыну қиын, өйткені олардың пікірлері аз.

1.3.4 Hybrid Recommender System (Гибридті ұсыныс жүйелері)

Гибридті ұсыныс жүйелері – бұл ұсыныстар үшін мазмұнды да, бірлескен сүзгілердің концепцияларын қолданатын ұсыныс беруші. Жүйе алдымен мазмұнды ұсынуды орындайды, өйткені пайдаланушы деректері жоқ, содан кейін жүйені қолданғаннан кейін ұқсас пайдаланушылармен пайдаланушы қалаулары орнатылады.

1.4 Ұсыныс жүйелерінде кездесетін қиындықтар

Қазіргі ұсыныс жүйелері қазірдің өзінде өте жақсы жұмыс істейді. Дегенмен, кейбір шектеулер бар. Құжатта біз негізгі мәселелерді шешу үшін Recommender System өнімділігін өлшеу үшін қолданылатын өлшемдер туралы түсінік береміз, соның ішінде суық іске қосу, дәлдік, деректердің жұқаруы, масштабталуы және әртүрлілігі секілді негізгі қиындықтарға шолу жасаймыз.

1.4.1. Суық іске қосу

"Суық іске қосу" қолжетімді ақпарат немесе метадеректер жеткіліксіз болған кезде, ұсыныс жүйелері дұрыс жұмыс жасамайды. Суық іске қосуды екі бөлек кіші топқа бөлуге болады: өнімнің суық іске қосылуы және пайдаланушының суық іске қосылуы. Электрондық коммерция веб-сайтында жаңа өнім пайда болған сайын, ол өнім суық іске қосылады, бұл пайдаланушының өзара әрекеттесуінің болмауына байланысты кері байланыс жоқ дегенді білдіреді. Сол себепті, ұсыныс жүйелері осы өнімге қатысты жарнамаларды қашан көрсету керектігін білмейді.

1.4.2 Сирек деректер

Деректер сирек болуы пайдаланушылар шектеулі элементтерді ғана бағалау әсерінен кездесуі мүмкін. Осылайша, ұсыныс жүйелері кері байланыс немесе баға бермейтіндерге негізсіз ұсыныстар бере алады.

1.4.3 Ауқымдылығы

Электрондық коммерция сайттарының тез өсуіне байланысты масштабтау мәселелері едәуір артты. Ұсыныс жүйелері нақты уақыт режимінде көптеген ықтимал көршілерді іздей алады, бірақ қазіргі заманғы электрондық коммерция сайттарының талаптары олардан көбірек көршілерді іздеуді талап етеді. Миллиондаған пайдаланушылар мен өнімдері бар платформа үшін масштабтау маңызды мәселе болып табылады.

1.5 Бағалау көрсеткіштері

Әдетте біз өз ұсыныстарымыздың сапасын олардың біздің қалауымызға қаншалықты сәйкес келетіндігіне және қаншалықты түсініксіз, бірақ қызықты болуына қарай бағалаймыз. Бірінші жағдайда біз релеванттықты еске түсіру және дәлдік сияқты көрсеткіштер арқылы бағалаймыз, ал екінші жағдайда әртүрлілік, қол жетімділік, айқындық және жаңашылдық сияқты көрсеткіштер арқылы бағалаймыз. Біз жиынтықтағы соңғы өлшемдерді көрнекі өлшемдер деп атаймыз. Ұсыныс жүйелерінің өнімділігін бағалауға арналған типтік өлшемдер – recall, precision, accuracy, ROC curves, F-measure қолданылады.

1.5.1 Recall және Precision (шолу және дәлдік)

Дәлдік барлық пайдаланушыға ұсынылатын элементтердің ішінен тиісті элементтерінің үлесін көрсетеді, ал кері байланыс – бұл ұсынылуға тиісті элементтердің жалпы элементтер санына қатысты саны. Тиісті элемент – бұл пайдаланушы тартымды деп санайтын элемент.

Жақсы ұсыну жүйесі екі метриканы бір уақытта оңтайландыруға тырысады. Мысалы, ол максималды қамтуды алу үшін көптеген өнімдерді пайдаланушыға ұсына алады.

1.5.2 Accuracy (Дәлдік)

Ұсыныс жүйесінің дәлдігін анықтау оңай емес, өйткені ұсыныс дәл ме, жоқ па, соны анықтайтын нақты әдіс жоқ. Оның дәлдігін бағалау үшін жеке тексеруді қолдана отырып, төмен болжамды қателерді іздеу керек.

1.5.3 ROC Curve

Қабылдағыштың жұмыс сипаттамаларын талдау (ROC) дәлдік пен шолуды алмастырады. Дәлдік неғұрлым жоғары болса, еске түсіру мәні соғұрлым төмен болады. ROC-талдау релевантты емес элементтерді алып тастамай-ақ, релевантты элементтерді алуға бағытталған.

Егер барлық тиісті элементтер маңызды емес элементтердің алдында көрсетілсе, онда идеалды ROC қисығы алынады.

1.5.4 F-өлшем

F-өлшем – бұл дәлдік пен шолудың тағы бір өлшемі және осы көрсеткіштерінің әрекетін көрсетеді. Ықтималдық тұрғысынан алғанда, F өлшемі – бұл бірінші сәтсіз сынақты анықтау үшін жасалатын сынақтардың саны.

2 Жобалау бөлімі

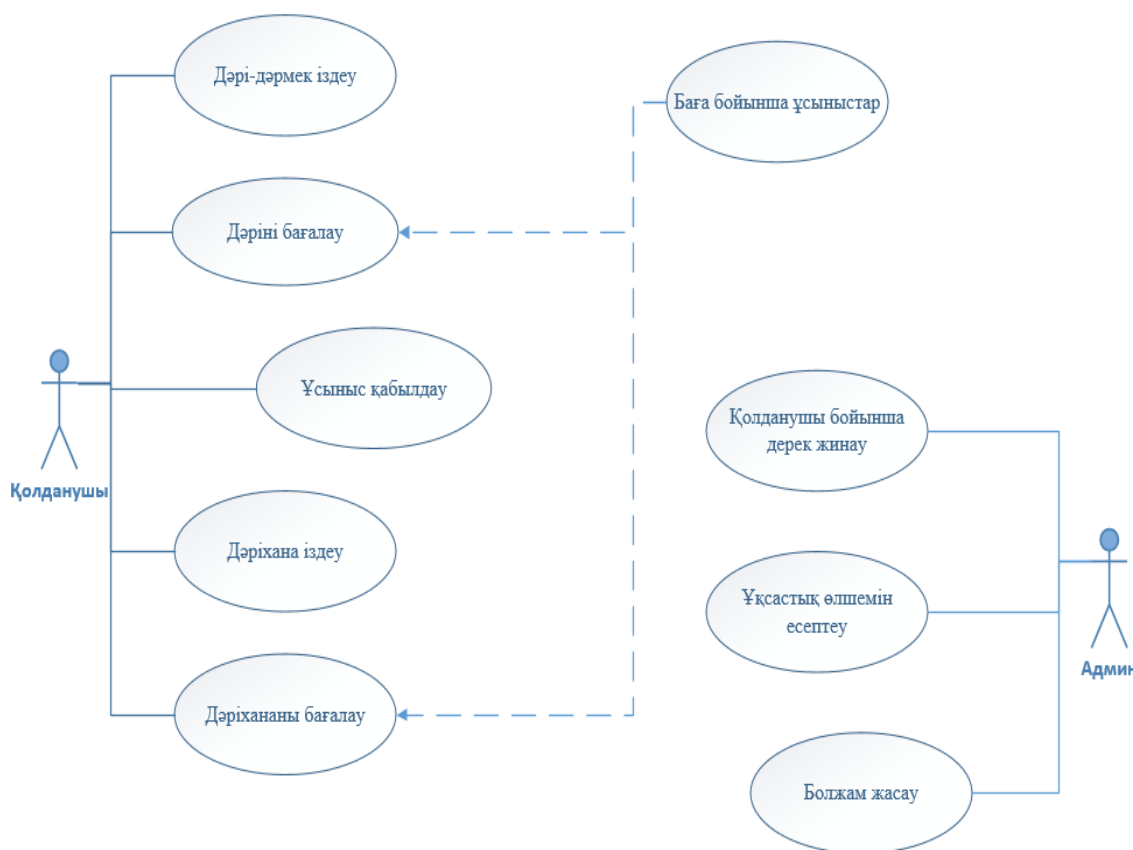
2.1 UML тілі

UML – бағдарламалық шешімдерді, қолданбалы құрылымдарды, жүйелік мінез-құлықты және бизнес-процестерді модельдеуге арналған бірыңғай модельдеу тілі. Негізінен, UML диаграммалар арқылы бағдарламалық жасақтаманы визуализациялайды [7].

2.1.1 Use case диаграммасы

Прецеденттер диаграммасының мақсаты – пайдаланушының жүйемен өзара әрекеттесуінің әртүрлі тәсілдерін көру болып табылады. Прецедент диаграммасының әр нұсқасы жүйе мен актер арасындағы іс-әрекеттерді анықтайды. Диаграмма жүйенің контекстін және талаптарын алу, архитектурасын тексеру үшін қажет.

Менің дипломдық жобамның прецеденттер диаграммасы төмендегі 2.1-суретте көрсетілген. Сонымен қатар, оқиғалар 2.1-кестеге салынған.



2.1-сурет – Use case диаграмма

2.1-кесте – Use case шарттары

Актерлер	Оқиға	Мазмұны
Қолданушы	Дәрі іздеу	Жүйеден сұрау
	Дәріні бағалау	Дәріге пікір қалдыру
	Дәріхана іздеу	Жүйеден сұрау
	Дәріхананы бағалау	Дәріханаға пікір қалдыру
	Ұсыныс қабылдау	Ұсынылған дәрі алу
Админ	Дерек жинау	Қолданушы іс-әрекеттері
	Ұқсастық ұпайын есептеу	Корреляция қолдану
	Болжам жасау	Дәрі ұсынады

2.1.2 Күй диаграммасы

Күй диаграммасы объектінің өмірлік циклі бойында тұра алатын бірқатар жағдайларды, сонымен қатар күйдің өзгеру себептерін көрсетеді. Орындалатын жағдайларға байланысты объектінің күйін модельдеуге болады. Диаграммада бір күйден екінші күйге ауысуға ерекше мән беріледі. Осылайша, объектіні инициализациядан шығаруға дейін модельдеуге болады.

Күй – бұл қандай-да бір шартты қанағаттандыратын, қандай-да бір әрекетті орындайтын немесе қандай-да бір оқиғаны күтетін объектінің өміріндегі жағдай немесе жағдай.

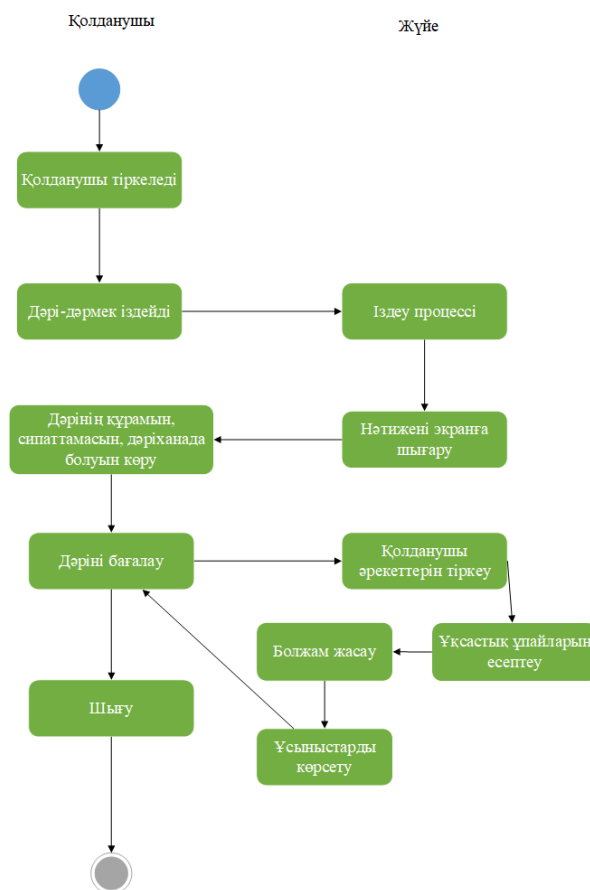
Оқиға – бұл уақыт пен кеңістікте орын алатын маңызды оқиғаның сипаттамасы. Мемлекеттік машиналар жағдайында оқиға-бұл мемлекеттің ауысуына әкелуі мүмкін ынталандырудың пайда болуы.

Қорғаныс күйі ауысу үшін триггер оқиғасы болғаннан кейін бағаланады. Егер қорғаныс шарттары бір-біріне сәйкес келмесе, бірдей бастапқы күйден және оқиғаның бірдей триггерінен бірнеше ауысулар болуы мүмкін. Қорғау шарты оқиға туындаған кезде ауысу үшін тек бір рет бағаланады. Логикалық өрнек объектінің күйіне сілтеме жасай алады.

2.1.3 Әрекеттер диаграммасы

Әрекеттер диаграммасы әрекеттің жұмыс процессін көрсететін диаграмма. Ол процедуралардың логикасын, бизнес-процесі мен жұмыс ағынын сипаттайтын технология. Бұл диаграмма басқару ағынының бір әрекеттен екінші әрекетке ауысуын көрсететін блок-схема.

Менің дипломдық жұмысымның әрекеттер диаграммасы төмендегі 2.2-суретте көрсетілген.



2.2-сурет – Әрекеттер(Activity) диаграммасы

2.1.4 Тізбек диаграммасы

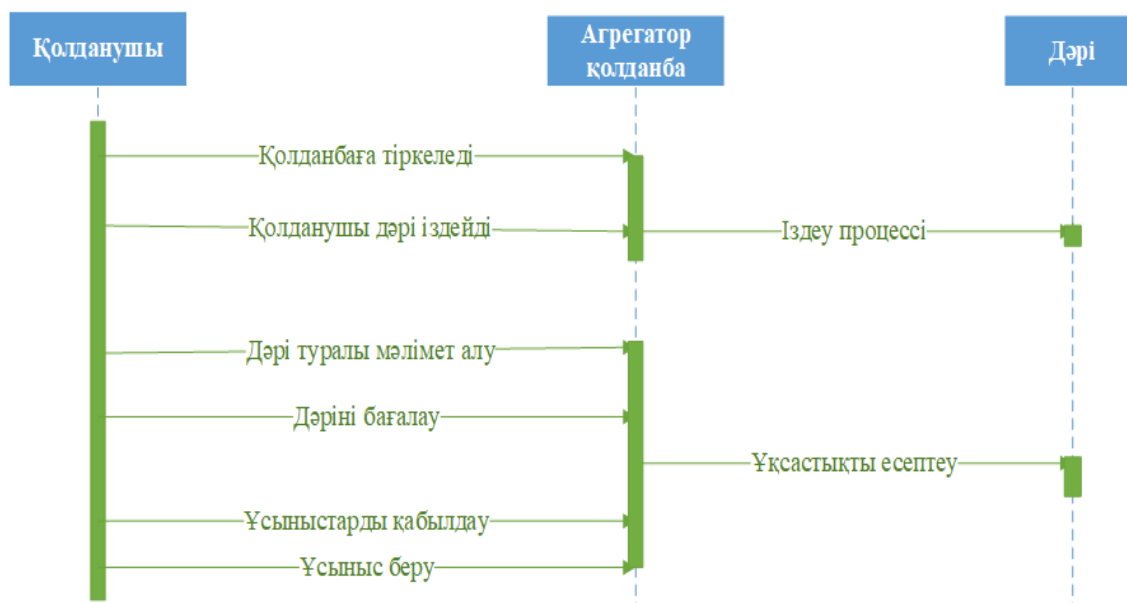
Тізбек диаграммасы жүйенің бөліктері қажетті функцияны атқару үшін бір-бірімен қалай әрекеттесетінін және орындау кезінде олардың тәртібін көрсетеді. Бұл жүйенің қалай әрекет ететінін болжауға және жаңа жүйені модельдеу процесінде класста туындауы мүмкін міндеттерді анықтауға көмектеседі.

Біз актерлерді әртүрлі рөлдерді, соның ішінде пайдаланушылар мен басқа да сыртқы тақырыптарды бейнелеу үшін қолданамыз. Реттілік диаграммасында бізде бірнеше кейіпкерлер болуы мүмкін.

Өмір жолы-бұл реттілік диаграммасында жеке қатысушыны бейнелейтін аталған элемент. Тізбектік диаграммадағы әр көшірме өмір сызығымен ұсынылған. Өмір сызығының элементтері тізбектік диаграмманың жоғарғы жағында орналасқан. Өмір жолын атауға арналған UML стандарты келесі форматқа сәйкес келеді – дананың аты: сынып атауы.

Мысалы, мұнда орынды брондау жүйесіндегі пайдаланушы, егер ол жүйеден тыс болса және жүйенің бөлігі болмаса, субъект ретінде көрсетіледі.

Дипломдық жұмыстың тізбек диаграммасының көрінісі 2.3-суретте көрсетілген.



2.3-сурет – Тізбек диаграммасы

2.2 Бағдарламалау ортасы

Бағдарламалау ортасын таңдауда мен басты назар аударған пункттер:

1) танымал:

Мен алдымда кездескен қиындықтар, сұрақтарға жауап оңай табу үшін;

2) үйренуге жеңіл:

Мақсатым – пайдалы кодты тез құру.

2.2.1 Python

Есеп беру немесе визуализация арқылы деректерді аналитикалық деректерге айналдыра алатын әдістерді қолдану үшін бізге ең жақсы құралдар қажет. Деректерге мән беру үшін C, C++, Java және Javascript сияқты танымал тілдер бар. Бірақ, R және Python сияқты танымал тілдер деректер ғылымы мен машиналық оқыту саласындағы тапсырмаларды сәтті орындау үшін өте пайдалы.

Python әмбебап және жоғары деңгейлі бағдарламалау тілі [2]. Оны таңдауымның себептері келесі пункттерде көрсетілген:

- 1) оңай, әрі түсінікті тіл;
- 2) икемді;
- 3) қолжетімді кітапханалар тізімі;

- 4) визуализация құралдары;
- 5) ауқымдылық.

2.2.2 Jupyter Notebook

Jupyter Notebook – бұл пайдаланушыларға кодтарды, теңдеулерді, визуализацияларды және мәтінді құруға және бөлісуге мүмкіндік беретін клиенттің интерактивті веб-қосымшасы [4]. Ол 40-тан астам бағдарламалау тілдерін қолдайтын көп тілді интерактивті есептеу ортасы болып саналады.

Jupyter Notebook-тің міндеттері:

- деректерді дайындау;
- статистикалық модельдеу;
- модельдерді оқыту;
- деректерді визуализациялау.

3 Зерттеу бөлімі

3.1 Pandas

Деректер жиынтығын жүктеу үшін және онымен жұмыс істеу үшін DataFrame ішіндегі pandas кітапханасын пайдалану керек.

Pandas – бұл Python-дағы ашық көзі, ол деректерді өңдеу және талдау үшін, сонымен қатар машиналық оқыту тапсырмалары үшін кеңінен қолданылады [4]. Ол NumPy деп аталатын басқа пакеттің негізінде салынған, ол көпөлшемді массивтерге қолдау көрсетеді. Ең танымал деректер пакеттерінің бірі ретінде Pandas Python экожүйесіндегі көптеген басқа деректерді өңдеу модульдерімен жақсы жұмыс істейді.

Ол сіздің деректеріңізді жол-баған форматында ұсынады. Pandas табиғи тілдерді өңдеуге арналған сандық кестелерді, уақыт қатарларын, суреттер мен мәліметтер жиынтығын басқаруға арналған дайын мәліметтер құрылымы мен амалдарын ұсынады. Негізінен pandas бұл кестелер ретінде оңай ұсынылатын мәліметтер жиынтығы үшін пайдалы.

Pandas көмегімен бірқатар амалдарды орындауға болады:

- деректерді тазалау;
- деректерді толтыру;
- деректерді нормалды түрге келтіру;
- біріктіру және қосу;
- деректерді визуалдау;
- деректерді тексеру;
- деректерді жүктеу, сақтау.

Pandas-ты қолдану мысалы 3.1-суретте көрсетілген:

```
B [1]: import numpy as np
import pandas as pd

import os
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

import json
from pandas.io.json import json_normalize

import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings; warnings.simplefilter('ignore')
```

```
B [2]: f = open(r'C:\Users\Maralym\Desktop\фото лекарств\DIPLOM\agregator.json', encoding='utf-8')
data = json.load(f)
df = pd.io.json.json_normalize(data)
data
```

```
Out[2]: [{"model": "users.user",
          "pk": 1,
          "fields": {"password": "pbkdf2_sha256$216000$6fCSMT01PV8D$0SDLJtqSL1DTQh56Shj3Rt6Lf3IOktCucHUf66CwZBM=",
                    "last_login": "2021-04-17T05:44:08.578Z",
                    "is_superuser": True,
                    "username": "admin",
                    "first_name": "",
                    "last_name": "",
                    "email": "admin@mail.ru",
                    "is_staff": True,
                    "is_active": True,
                    "date_joined": "2021-03-28T15:13:28.207Z",
                    "address": "",
                    "groups": [],
                    "user_permissions": []}]
```

3.1-сурет – Деректерді pandas кітапханасы арқылы жүктеу

3.2 Numpy

NumPy – бұл көпөлшемді массивтер объектілерінен және осы массивтерді өңдеуге арналған процедуралар жиынтығынан тұратын кітапхана [6]. NumPy көмегімен массивтерде математикалық және логикалық операцияларды орындауға болады.

NumPy әртүрлі функциялар орындайды, соның ішінде келесі маңызды функциялар бар:

- N-өлшемді массивтің нысаны;
- күрделі функциялар;
- C/C++ және Fortran кодтарын біріктіруге арналған құралдар;
- сызықтық алгебраның, Фурье түрлендіруінің және кездейсоқ сандардың пайдалы мүмкіндіктері.

3.3 Дерек қоры

Деректер қоры дәріханалар бойынша агрегатордың деректерін сақтайтын json форматындағы файлдан алынды. Алғашқыда деректер қоры құрылымданбаған күйде сақталды, кейін ондағы деректер нормалауға келтірілді.

Мен қолданған деректер қорының көрінісі 3.2-суретте бейнеленген.

```
f = open(r'C:\Users\Maralym\Desktop\фото_лекарств\DIPLON\agregator.json', encoding='UTF8')
data = json.load(f)
df = pd.io.json.json_normalize(data)
data
```

```
[{'model': 'users.user',
  'pk': 1,
  'fields': {'password': 'pbkdf2_sha256$216000$6fC5MT01PV8D$05DLJtqSL1DQhS65hj3Rt6LfJIOktCuchUF66CwzBh=',
            'last_login': '2021-04-17T05:44:08.578Z',
            'is_superuser': True,
            'username': 'admin',
            'first_name': '',
            'last_name': '',
            'email': 'admin@mail.ru',
            'is_staff': True,
            'is_active': True,
            'date_joined': '2021-03-28T15:13:28.207Z',
            'address': '',
            'groups': [],
            'user_permissions': []}},
 {'model': 'users.user',
  'pk': 2,
  'fields': {'password': 'pbkdf2_sha256$216000$MCaWNB097JgIS2NPagm04L6LJejQJlNxnHzFKsATXkABFrdzTJzETr05BM=',
            'last_login': None,
            'is_superuser': False,
            'username': 'molya'}}
```

```
df.head()
```

	model	pk	fields.password	fields.last_login	fields.is_superuser	fields.username	fields.first_name	fields.last_name
0	users.user	1	pbkdf2_sha256\$216000\$6fC5MT01PV8D\$05DLJtqSL1DQhS65hj3Rt6LfJIOktCuchUF66CwzBh=	2021-04-17T05:44:08.578Z	True	admin		admin
1	users.user	2	pbkdf2_sha256\$216000\$MCaWNB097JgIS2NPagm04L6LJejQJlNxnHzFKsATXkABFrdzTJzETr05BM=	None	False	molya		molyadm99
2	users.user	3	pbkdf2_sha256\$216000\$E0ZHHLeSt3K\$1WbgrXOnUJ4O...	None	False	maralym		maralym.bb
3	users.user	4	pbkdf2_sha256\$216000\$3ZleKily2WW1\$NLZVPCLen2Vd...	None	False	maralym.bb@gmail.com		
4	users.user	5	qwerty	2021-03-31T03:59:28Z	False	aizhan01	Айжан	Байхакоса

5 rows x 41 columns

3.2-сурет – Деректерді оқу

Деректер қоры дәрі және дәріханалар бойынша ақпарат сақтайтын 10 кестеден тұрады. Соның ішінде, ұсыныс жүйелерін құруда 4 кесте қолданылды:

- 1) product.product – дәрілер жайлы ақпарат,
- 2) product.reviewproduct – дәрілер бойынша пікір,
- 3) categories.category – дәрі-дәрмектің түрін,
- 4) users.user – қолданушылар бойынша ақпарат сақтайды.

Ұсыныс жүйелері модельдері барлығы 81 дәріге жазылған 421 пікір бойынша оқытылды. Көрінісі төмендегі 3.3-суретте.

```
new_df=df[['model','pk', 'fields.author', 'fields.rating', 'fields.text', 'fields.product']]
new_df=new_df.rename(columns={'fields.author':'authorId', 'fields.rating':'Rating', 'fields.text':'Review', 'fields.product':'productId'})
new_df=new_df.loc[new_df['model'] == 'product.reviewproduct']
new_df
```

	model	pk	authorId	Rating	Review	productId
358	product.reviewproduct	8	5.0	5.0	Препарат очень эффективно помогает для детей и...	35.0
359	product.reviewproduct	7	8.0	5.0	Один из самых лучших препаратов необходимых в ...	41.0
360	product.reviewproduct	8	9.0	5.0	хорошее средство от поориза 🍌🍌🍌\!прекондую...	1.0
361	product.reviewproduct	9	7.0	5.0	С тенотеном без истерик пролечили дочке зубы, ...	48.0
362	product.reviewproduct	10	10.0	5.0	У дочки переходный период начался, начала осор...	48.0
...
774	product.reviewproduct	426	101.0	1.0	Ничего особенного	98.0
775	product.reviewproduct	427	88.0	4.0	Хорошие витамины	97.0
776	product.reviewproduct	428	84.0	4.0	Мой любимый чаек, очень хорошо помогает при пе...	83.0
777	product.reviewproduct	429	88.0	3.0	Не понравился	83.0
778	product.reviewproduct	430	109.0	4.0	хороший чай, читайте инструкцию	83.0

421 rows x 6 columns

```
new_df.productId.unique()
```

```
array([ 35., 41., 1., 48., 49., 43., 44., 72., 73., 74., 2.,
        5., 66., 52., 67., 68., 36., 27., 4., 38., 7., 9.,
        10., 11., 20., 12., 18., 16., 8., 14., 15., 3., 6.,
        70., 69., 32., 24., 22., 23., 90., 40., 37., 38., 33.,
        54., 42., 47., 56., 71., 19., 57., 17., 59., 60., 50.,
        21., 28., 26., 45., 98., 88., 99., 89., 105., 75., 101.,
        76., 77., 104., 100., 92., 78., 79., 80., 94., 82., 102.,
        85., 97., 96., 83.]
```

3.3-сурет – Деректерге шолу

Дәрі-дәрмекке қойылған ең төмен ұпай саны – 1, ең жоғарғысы – 5, төмендегі 3.4-суреттен көруге болады:

```
print('Минималды рейтинг: %d' %(new_df.Rating.min()))
Минималды рейтинг: 1
```

```
print('Максималды рейтинг: %d' %(new_df.Rating.max()))
Максималды рейтинг: 5
```

3.4-сурет – Дәрі ұпайлары

3.4 TF-IDF алгоритмі

TF-IDF алгоритмі кез-келген құжаттағы кілт сөзді өлшеу үшін қолданылады және құжатта қанша рет пайда болғанына байланысты осы кілт сөздің мағынасын береді. Қарапайым тілмен айтқанда, TF-IDF неғұрлым жоғары болса, термин соғұрлым аз және маңызды болады және керісінше.

Әр сөз немесе термин тиісті TF және IDF көрсеткішіне сәйкес келеді. TF және IDF терминдерінің бағалары осы терминнің TF-IDF салмағы деп аталады.

TF (термин жиілігі) сөз құжатта пайда болған саны болып табылады. Мұны білген кезде, сіз терминді тым жиі немесе сирек қолданғанды көреміз. (3.1 - формула)

TF (t) = (құжатта кездесетін t термин саны) / (құжаттағы терминдердің жалпы саны), (3.1)

IDF (терминнің маңыздылығы) есептеу кезінде барлық терминдер бірдей маңызды болып саналады. Алайда, "бар", "туралы" және "сол" сияқты белгілі бір терминдер бірнеше рет пайда болуы мүмкін, бірақ маңызды емес. (3.2 - формула)

IDF (t) = \log_e (құжаттардың жалпы саны / ондағы t терминімен құжаттар саны). (3.2)

TF-IDF артықшылықтары:

а) бұл сұрауға қатысты құжаттармен сұраудағы сөздерді салыстырудың тиімді және қарапайым алгоритмі;

б) TF-IDF нақты сұранысқа тікелей қатысты құжаттарды қайтарады;

в) TF-IDF күрделі алгоритмдер мен сұрау іздеу жүйелері үшін негіз құруға өте ыңғайлы.

3.5 Cosine Similarity

Косинус ұқсастығы – бұл элементтердің көлеміне қарамастан қаншалықты ұқсас екенін анықтау үшін қолданылатын көрсеткіш.

Математикалық тұрғыдан ол көп өлшемді кеңістікке жобаланған екі вектордың арасындағы бұрыштың косинусын өлшейді. Бұрыш неғұрлым аз болса, косинусидтық ұқсастық соғұрлым үлкен болады. Евклид қашықтығын өлшеуден айырмашылығы, косинус ұқсастығы шаманы емес, құжаттардың бағытын анықтайды.

Косинус ұқсастығын өлшеу екі нөлдік емес вектордың косинусын табудан басталады (3.3 формула):

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}, \quad (3.3)$$

мұнда A және B – векторлар,

A_i – A векторының компоненті,

B_i – B векторының компоненті.

Нәтижесі -1-ден 1-ге дейінгі диапазондағы мән болады, онда -1 – ұқсастық емес, 0 – ортогональды (перпендикуляр), ал 1 – толық ұқсастық.

Косинус ұқсастығында абстрактілі математикадан тыс қосымшалар бар. Өлшеу деректерді іздеу, ақпаратты іздеу және мәтінді сәйкестендіру процестерінде қолданылады. Векторлар айнымалы қасиеттерге тағайындалғаннан кейін, өлшеу объектілер арасындағы ұқсастықты түсінудің құнды құралына айналды.

Машиналық оқыту деректерді іздеу және ақпаратты іздеу сияқты қосымшаларда косинус ұқсастығын қолданады. Мысалы, құжаттар базасын әр терминге құжатта осы терминнің жиілігіне сәйкес келетін өлшем және байланысты вектор тағайындалатындай етіп өңдеуге болады. Бұл косинус ұқсастығын өлшеуге, құжаттарды бір-бірімен олардың ұқсастықтары мен тақырыптарының сәйкес келуіне қарай ажыратуға және салыстыруға мүмкіндік береді.

Ұқсастықты есептеу көптеген мәтінді іздеу қосымшаларының негізгі компоненті болып табылады. Мысалы, егер бізде екі мәтіндік сегменттің ұқсастығын бағалаудың тамаша әдісі болса, олар идеалды ақпараттық-іздеу жүйесін құра алар еді. Соңғы жылдары көптеген метрикалар пайда болды – Евклидке негізделген қашықтық метрикасы, косинус, Джаккард, сүйектер, Йенсеншаннонның дивергенцияға негізделген метрикасы әртүрлі түрлерді шешу үшін ұсынылды.

Python scikit-learn кітапханасы косинус ұқсастығын есептеу мүмкіндігін ұсынады. Scikit-learn әдісі екі вектордың орнына екі матрицаны параметр ретінде қабылдайды және екі параметр арасындағы векторлардың әр жұбы арасындағы косинус ұқсастығын есептейді және нәтижені матрица ретінде (3.5-сурет) қайтарады.

```
B [12]: from sklearn.metrics.pairwise import cosine_similarity
import operator
def similar_users(authorId, matrix, k=3):
    user = matrix[matrix.index == authorId]

    other_users = matrix[matrix.index != authorId]

    similarities = cosine_similarity(user, other_users)[0].tolist()

    indices = other_users.index.tolist()

    index_similarity = dict(zip(indices, similarities))

    index_similarity_sorted = sorted(index_similarity.items(), key=operator.itemgetter(1))
    index_similarity_sorted.reverse()

    top_users_similarities = index_similarity_sorted[:k]
    users = [u[0] for u in top_users_similarities]

    return users

B [13]: current_user = 9.0
similar_user_indices = similar_users(current_user, rating_matrix)
print(similar_user_indices)

[19.0, 35.0, 25.0]
```

3.5-сурет – Ұсыныс жүйеде қолданған cosine similarity

3.6 Бағалау метрикалары

Орташа квадраттық қате (RMSE) (3.4 формуласы) – бұл болжамды рейтингтердің дәлдігін бағалауда ең танымал көрсеткіш. Ол Евклид қашықтығын қолдана отырып өлшенген шынайы мәндердің ауытқу болжамдарының қаншалықты екенін көрсетеді:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N \|y(i) - \hat{y}(i)\|^2}{N}}, \quad (3.4)$$

мұнда N-деректер нүктелерінің саны,

$y(i)$ – i -өлшеу,

$\hat{y}(i)$ – оның тиісті болжамы.

RMSE есептеу үшін әр дерек нүктесі үшін болжау мен шындық арасындағы айырмасын есептейміз, сосын қалдықтардың орташа мәнін есептеп және осы орташа мәннен квадрат түбірін алу қажет. RMSE әдетте бақыланатын оқытуға арналған қосымшаларда қолданылады, өйткені RMSE әр болжамды деректер нүктесінде дәл өлшеуді қолданады және қажет етеді.

Орташа абсолютті қате (MAE) – машиналық оқыту контекстінде абсолютті қате байқаудың болжамдары мен осы байқаудың шынайы мәні арасындағы айырмашылық (3.5-формула). MAE болжамдар мен бақылаулар тобы үшін абсолютті қателіктердің орташа мәнін бүкіл топ үшін қателіктердің өлшемі ретінде қабылдайды. MAE-ні L1 жоғалту функциясы деп те атауға болады.

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n}, \quad (3.5)$$

мұнда N-деректер нүктелерінің саны,

$y(i)$ – i -өлшеу,

$x(i)$ – оның тиісті болжамы.

Қалыпқа келтірілген қашықтыққа негізделген тиімділік көрсеткіші (NDPM) – бізде пайдаланушының салыстырмалы рейтингі туралы ақпарат жоқ кезде қолайлы. (3.6-формула)

$$NDPM = \frac{C^- + 0.5C^{u0}}{C^u}, \quad (3.6)$$

C^- – бұл пайдаланушы жауап берген элементтердің ұсынылған жұптарының саны (жоқ жауабы). C^+ – бұл пайдаланушы жауап берген элементтердің ұсынылған жұптарының саны (иә жауабы). C^u – бұл

элементтердің барлық жұптарының саны, онда пайдаланушының жауабы басқаша болды.

3.7 Content Based Recommender System кодына шолу

Мазмұнға негізделген ұсыныс жүйесінде деректер қорынан қажет деректерден витрина жинап аламын. Витринаны 3.6-суреттен көруге болады:

authorid	Rating	Review	productid	title	compound	desc
0	5.0	5.0	35	ГРОПРИНОСИН 500 МГ №50 ТАБЛ.	Одна таблетка содержит активное вещество - ино...	Характеристики:\r\nАртикул:\r\n1001468\r\n\r\n...
1	6.0	5.0	41	КРЕОН 10000 150 МГ №20 КАПС.С МК/СФЕР.	Одна капсула содержит активное вещество - панкре...	Характеристики:\r\nАртикул:\r\n1001759\r\n\r\n...
2	9.0	5.0	1	DUCRAY KERTYOL PSO Бальзам увлажняющий 200мл	Состав:\r\nЦелестрол® - Оказывает противово...	Артикул:\r\n203995\r\n\r\nШтрихкод:\r\n32827...
3	7.0	5.0	48	ТЕНОТЕН ДЕТСКИЙ №40 ТАБЛ. ГОМЕОП.	Одна таблетка содержит активное вещество: Анти...	Характеристики:\r\nАртикул:\r\n1000177\r\n\r\n\r\n...
4	10.0	5.0	48	ТЕНОТЕН ДЕТСКИЙ №40 ТАБЛ. ГОМЕОП.	Одна таблетка содержит активное вещество: Анти...	Характеристики:\r\nАртикул:\r\n1000177\r\n\r\n\r\n...
...
416	101.0	1.0	96	ВИТАМИННО-МИНЕРАЛЬНЫЙ КОМПЛЕКС ОТ А ДО ZN №30 ...	Содержание активных веществ в одной таблетке (...)	Характеристики:\r\nАртикул:\r\n0258960\r\n\r\n\r\n...
417	88.0	4.0	97	ВИТАМИННО-МИНЕРАЛЬНЫЙ КОМПЛЕКС ОТ А ДО ZN №30 ...	Содержание активных веществ в одной таблетке (...)	Характеристики:\r\nАртикул:\r\n0258960\r\n\r\n\r\n...
418	84.0	4.0	83	ИМБИРНЫЙ ЧАЙ КЛАССИЧЕСКИЙ 1,5 Г №20 ФИТОЧАЙ ПА...	Состав: корневища с корнями имбиря лекарствен...	Характеристики:\r\nАртикул:\r\n20005885\r\n\r\n\r\n...
419	66.0	3.0	83	ИМБИРНЫЙ ЧАЙ КЛАССИЧЕСКИЙ 1,5 Г №20 ФИТОЧАЙ ПА...	Состав: корневища с корнями имбиря лекарствен...	Характеристики:\r\nАртикул:\r\n20005885\r\n\r\n\r\n...
420	109.0	4.0	83	ИМБИРНЫЙ ЧАЙ КЛАССИЧЕСКИЙ 1,5 Г №20 ФИТОЧАЙ ПА...	Состав: корневища с корнями имбиря лекарствен...	Характеристики:\r\nАртикул:\r\n20005885\r\n\r\n\r\n...

421 rows x 7 columns

3.6-сурет – Оқытылатын деректер

Деректерді алған соң, мен дәрі-дәрмектің сипаттамаларын TF-IDF көмегімен кодтаймын (3.7-сурет):

```

In [9]: tf = TfidfVectorizer(analyzer='word', ngram_range=(1, 3), min_df=0)
tfidf_matrix = tf.fit_transform(result['desc'])
tf

```

3.7-сурет – TF-IDF кодтау

Мазмұн ұсынысы пайдаланушының ұсынылған дәріні ұнатуы ықтималдығын болжауды қамтиды.

Мазмұн ұсынысын косинус ұқсастығы (3.8-сурет) арқылы алдым.

```
In [8]: cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
results = {}
for idx, row in result.iterrows():
    uksas_index = cosine_similarities[idx].argsort()[::-100:-1]
    uksas_onim = [(cosine_similarities[idx][i], result['productId'][i]) for i in uksas_index]
    results[row['productId']] = uksas_onim[1:]
```

3.8-сурет – Косинус ұқсастығын мазмұн ұсыныс жүйесіне қолдану

3.8 Item Based Collaborative Recommender System кодына шолу

Бірлескен сүзу – бұл ұқсас дәрілер әсері негізінде пайдаланушыға ұнауы мүмкін элементтерді сүзуге болатын әдіс.

Екі элементтің ұқсастығын есептеу үшін біз мақсатты элемент үшін бағаланған элементтер жиынтығын зерттейміз және олардың мақсатты і элементіне қаншалықты ұқсас екенін есептейміз, содан кейін ең ұқсас элементтерді таңдаймыз. Екі элементтің ұқсастығы екі элементті де бағалаған пайдаланушыларды бағалау арқылы есептеледі, содан кейін косинус ұқсастығын функциясын қолдану қажет.

Алдымен деректер қорынан витрина құрып, оны TF-IDF көмегімен кодтаймыз. Әр қолданушының бағалаған дәрілерін бөліп аламыз (3.9-сурет):

```
In [16]: missing_pivot = result.pivot_table(values='Rating', index='authorId', columns=
rate={}
rows_indexes={}
for i, row in missing_pivot.iterrows():
    rows = [x for x in range(0, len(missing_pivot.columns))]
    combine = list(zip(row.index, row.values, rows))
    rated = [(x,y,z) for x,y,z in combine if str(y) != 'nan']
    index = [i[1] for i in rated]
    row_names = [i[0] for i in rated]
    rows_indexes[i] = index
    rate[i] = row_names

In [17]: rate
Out[17]: {5.0: ['MILANI Палитра теней GILDED NUDE 89198A',
'VICHY AQUALIA THERMAL Бальзам Пробуждающий для контура глаз 15мл',
'ГРОПРИНОСИН 500 МГ №50 ТАБЛ.',
'MIP AVENE XERACALM Крем для детей и взрослых 200мл',
'УШАСТЫЙ НЯНЬ Порошок для стирки детского белья 400 гр'],
6.0: ['ГРИППО №100 ТАБЛ.',
'КМ-ПОЧЕЧНЫЙ 1,5 Г №20 ФИТОЧАЙ (БАД)',
'КРЕОН 10000 150 МГ №20 КАПС.С МК/СФЕР.',
'УШАСТЫЙ НЯНЬ отбеливатель для детского белья, 750мл'],
7.0: ['SPLAT professional Паста зубная отбеливающая, 100мл.',
'ДИАБЕТИЧЕСКИЙ 1,5 Г №20 ФИТОЧАЙ ПАК. "АЛТАЙСКИЕ ТРАВЫ"',
'КАРДИОАКТИВ ВИТАМИНЫ ДЛЯ СЕРДЦА 0,25 Г №30 КАПС.',
'MIP AVENE XERACALM Крем для детей и взрослых 200мл',
'ТЕНОТЕН ДЕТСКИЙ №40 ТАБЛ. ГОМЕОП.',
'УШАСТЫЙ НЯНЬ Порошок для стирки детского белья 400 гр',
'Я САМАЯ Палочки ватные, 30шт упаковка'],
8.0: ['URIMAGE Вода мицеллярная для нормальной и сухой кожи, 100мл.',
'ДИАБЕТИЧЕСКИЙ 1,5 Г №20 ФИТОЧАЙ ПАК. "АЛТАЙСКИЕ ТРАВЫ"',
'ТЕРМОМЕТР бесконтактный РАК-F103',
'Я САМАЯ Палочки ватные. 30шт упаковка']
```

3.9-сурет – Қолданушылар бағалаған дәрілер

Кейін әр қолданушы үшін бағаланбаған дәрілерді шығарамыз (3.10-сурет):

```

pivot_table = result.pivot_table(values= 'Rating', index='authorId', columns=
pivot_table = pivot_table.apply(np.sign)
notrated={}
notrated_indexes={}
for i,row in pivot_table.iterrows():
rows=[x for x in range (0,len(missing_pivot.columns))]
combine= list(zip(row.index, row.values, rows))
idx_row = [(idx,col) for idx,val,col in combine if not val>0]
indices = [i[i] for i in idx_row]
row_names = [i[0] for i in idx_row]
notrated_indexes[i] = indices
notrated[i] = row_names

```

notrated

```

{5.0: ['AVENE CLEANANCE WOMAN Сыворотка 30 мл',
'AVENE EAU THERMALE Вода термальная 300мл',
'DUCRAY KERTYOL PSO Бальзам увлажняющий 200мл',
'HUGGIES Подгузники Elite Soft 1, 3-5кг 25шт',
'MATERNEA Масло для упругости кожи 100мл',
'MILANI Румяна для щек мини версия ROMANTIC ROSE 901',
'MILANI Хайлайтер с галаграфическим эффектом luster light 03',
'SPLAT professional Паста зубная Отбеливающая, 100мл.',
'URIAGE Вода мицеллярная для нормальной и сухой кожи, 100мл.',
'VICHY BODY Дезодорант-Антиперспирант шариковый, 50мл.',
'VICHY LIFTACTIV Крем FLEXILIFT TEINT тональный от морщин SPF25 25тон 30м
л',
'VIVIENNE SAVO Тушь для ресниц Artistique 01 D215003101',
'YOKO SUN PREMIUM Подгузники-трусики XL 12-20кг 38шт',
'YOKO SUN Салфетки влажные детские 0+ 100шт',
'АНАФЕРОН №20 ТАБЛ./РАССАС. ГОМЕОП.',
'АСКОРБИНОВАЯ КИСЛОТА 250 МГ №200 ДРАЖЕ (БАД)',
'АСКОРУТИН 0,33 Г №50 ТАБЛ. (БАД)',
'АЩ ХОТ ДРИНК 600 МГ №6 ПОР./Д/Р-РА ДЛЯ ПРИЕМА ВНУТРЬ ПАК.',
'БАНЫКА АГАФИИ Маска для лица омолаживающая 100мл.',
'БЕЛЫЙ КОСМОС 100 мл. крем']

```

3.10-сурет – Қолданушы бағаламаған дәрілер

Элементке негізделген тәсіл үшін талаптарға сәйкес келетін алгоритм – kNN, өйткені алгоритм центрленген косинустың ұқсастық формуласына өте жақын. KNN дәріге қатысты болжам жасаған кезде, ол мақсатты дәрі мен кез-келген басқа дәрі арасындағы "қашықтықты" есептейді. Содан кейін ол өзінің қашықтықтарын бағалайды және ең жақын көршілердің ең жақсы дәрілерін ең ұқсас дәрілер ретінде қайтарады. Код бөлігі 3.11-суретте бейнеленген:

```

n=5
cosine_knn = NearestNeighbors(n_neighbors=n, algorithm='brute',
metric='cosine')
item_cosine_knn_fit = cosine_knn.fit(pivot_table.T.values)
item_distances, item_indices =
item_cosine_knn_fit.kneighbors(pivot_table.T.values)

```

3.11-сурет – kNN қолданылуы

Соңында ұсыныс беру төмендегі код (3.12-сурет) негізінде іске асады:

```

def getrecommendations(user, number_of_recs = 30):
if user> len(pivot_table.index):
print('Тым көп {} қолданушы, тағы
көрініз!'.format(len(pivot_table.index)))
else:
print("Сіз өткенде бұрында көрген өнім:
\n\n{}".format('\n'.join(rate[user])))
print()
print("Мына өнімдерді де қараңыз:\n")
for k,v in topRecs.items():
if user == k:
for i in v[:number_of_recs]:
print('{} ұқсастық: {:.4f}'.format(i[0], 1-i[1]))

```

3.12-сурет – Коллаборативті ұсыныс жүйесі

3.9 User Based Collaborative System кодына шолу

Бұл әдіс сұралған пайдаланушыға ұқсас пайдаланушыларды анықтайды және қалаған рейтингті сол ұқсас пайдаланушылардың бағалауларының орташа өлшенген мәні ретінде бағалайды.

Деректер алған соң, бірнеше Nan мәндері бар сирек матрицаны аламыз. Ол үшін кем дегенде 5 бағалауы бар қолданушылардың матрицасын аламыз (3.13-сурет).

```
rating_per_user_df = pd.DataFrame (ratings_per_user)
filter_ratings_per_user_df = rating_per_user_df [rating_per_user_df.Rating>=5]
prolific_users = filter_ratings_per_user_df.index.tolist ()

filtered_ratings = rating_df[rating_df.productId.isin(Popular_prod)]
filtered_ratings = rating_df[rating_df.authorId.isin(prolific_users)]
len(filtered_ratings)
rating_matrix = filtered_ratings.pivot_table(index='authorId', columns='productId', values='Rating')
rating_matrix = rating_matrix.fillna(0)
rating_matrix.head()
```

productId	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	...	89.0	90.0	94.0	98.0	99.0	100.0	101.0	102.0	104.0	105.0	
authorId																						
5.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	...	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
9.0	5.0	5.0	3.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	
10.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	...	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	3.0	0.0	
11.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

5 rows x 67 columns

3.13-сурет – Қолданушылардың сирек матрицасы

Косинус ұқсастығы бойынша ұқсас қолданушыларды топтаймыз (3.14-сурет):

```
from sklearn.metrics.pairwise import cosine_similarity
import operator
def similar_users(authorId, matrix, k=3):
    user = matrix[matrix.index == authorId]

    other_users = matrix[matrix.index != authorId]

    similarities = cosine_similarity(user,other_users)[0].tolist()

    indices = other_users.index.tolist()

    index_similarity = dict(zip(indices, similarities))

    index_similarity_sorted = sorted(index_similarity.items(), key=operator.itemgetter(1))
    index_similarity_sorted.reverse()

    top_users_similarities = index_similarity_sorted[:k]
    users = [u[0] for u in top_users_similarities]

    return users

current_user = 9.0
similar_user_indices = similar_users(current_user, rating_matrix)
print(similar_user_indices)

[19.0, 35.0, 25.0]
```

3.14-сурет – Ұқсас қолданушылар

Ұқсас қолданушылар дәрілері орташа ұқсастық мәні бойынша көрсетіледі (3.15-сурет):

```
: def recommend_item(user_index, similar_user_indices, matrix, items=5):  
  
    similar_users = matrix[matrix.index.isin(similar_user_indices)]  
    similar_users = similar_users.mean(axis=0)  
    similar_users_df = pd.DataFrame(similar_users, columns=['mean'])  
  
    user_df = matrix[matrix.index == user_index]  
    user_df_transposed = user_df.transpose()  
    user_df_transposed.columns = ['Rating']  
    user_df_transposed = user_df_transposed[user_df_transposed['Rating']!=0]  
    products_unseen = user_df_transposed.index.tolist()  
  
    similar_users_df_filtered = similar_users_df[similar_users_df.index.isin(products_unseen)]  
    similar_users_df_ordered = similar_users_df.sort_values(by=['mean'], ascending=False)  
    top_n_products = similar_users_df_ordered.head(items)  
    top_n_products_indices = top_n_products.index.tolist()  
    product_information = desc_df[desc_df['productId'].isin(top_n_products_indices)]  
  
    return product_information |
```

3.15-сурет – Қолданушыға негізделген ұсыныс жүйесі

ҚОРЫТЫНДЫ

Қол жетімді ақпарат пен деректердің үнемі өсіп келе жатқан көлемінің аясында қандай ақпаратты іздеу керектігін және оны қайдан іздеу керектігін түсіну қиын. Іздеу процесін жеңілдету үшін көптеген компьютерлік әдістер жасалды, осы әдістердің бірі – пайдаланушыларға ең маңызды ақпаратты іздеу және бөлектеу арқылы қол жетімді ақпаратты зерттеуге бағыт беретін ұсыныс жүйесі.

Ұсыныс жүйелері әртүрлі зерттеу салаларында, соның ішінде ақпаратты іздеу, ақпаратты сүзу, мәтінді жіктеу және т.б. қолданылады. Олар машиналық оқыту және деректерді іздеу сияқты әдістерді, сонымен қатар алгоритмдер, бірлескен және гибриді тәсілдер мен бағалау әдістерін қамтитын бірқатар тұжырымдамаларды қолданады.

Ұсыныс жүйелері ақпаратты іздеудің және сүзудің жаңа мүмкіндіктерін ашты. Ұсыныс жүйесі арқасында интернет-дүкендер пайдасын көбейтті, музыка және кино әуесқойлары өздері үшін жаңаларын тапты, туристтер жаңа қызықты жерлерді іздеп, келе алды. Осы мүмкіндіктердің барлығына ие болу қолданушылар уақытын үнемдейді [3].

Бұл дипломдық жұмыс «Дәріханалар бойынша мобильдік қолданба агрегатор» үшін жасалған ұсыныс жүйелерін қамтиды. Ұсыныс жүйелері қолданушыларға арнап оларды қызықтыруы мүмкін дәрілерді ұсынады. Бұл дипломдық жұмыс ұсыныс жүйелеріне және оларды қолданудың әртүрлі тәсілдеріне бағытталған.

Дипломдық жұмыстың нәтижесінде келесі тапсырмалар орындалды:

- мазмұнға негізделген ұсыныс жүйесі құрастырылды;
- элементке негізделген коллаборативті ұсыныс жүйесі жасалды;
- қолданушыға негізделген коллаборативті ұсыныс жүйесі дайындалды.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Ұсыныс жүйелері// Электрондық нұсқа https://en.wikipedia.org/wiki/Recommender_system.
- 2 Python тілі бойынша нұсқаулық// Электрондық нұсқа <https://tproger.ru/tag/python/>.
- 3 Ұсыныс жүйелері Google негізінде// Электрондық нұсқа <https://developers.google.com/machine-learning/recommendation>.
- 4 Pandas кітапханасы // Электрондық нұсқа <https://pandas.pydata.org/>.
- 5 Jupyter Notebook бағдарламалау ортасы // Электрондық нұсқа <https://jupyter.org/>.
- 6 Numpy кітапханасы // Электрондық нұсқа <https://numpy.org/>.
- 7 UML диаграммалары// Электрондық нұсқа <https://ru.wikipedia.org/wiki/UML>.
- 8 Бабин, В. Е. Анализ рекомендательной системы по подбору музыки с точки зрения экономии времени / В. Е. Бабин. – Текст : электронный // Наука без границ. – 2021. – № 3 (55). – С. 26-39.
- 9 Filipyev, A.V. Item-based recommender system with statistical learning for unauthorized customers // Программные продукты и системы / Software & Systems. – 2019. – №2 (32). – С. 221-226.

А Қосымшасы (міндетті)

Техникалық тапсырма

А.1.1 Data Science шешімдерін әзірлеуге арналған техникалық тапсырма

Бұл техникалық тапсырма дәріханалардың мобильдік қолданба-агрегаторы үшін Data Science шешімдерін әзірлеу мақсатында қолданылады. Бұл Data Science шешімдер дәріханалар бойынша қолданбалар иелері, қарапайым тұтынушылар қолдана алады. Шешімдер дәрі-дәрмек сатып алушылар мен дәріхана иелері арасындағы тез, пайдалы қызмет көрсетуді жақсартуға, дәлірек айтқанда қолданушыларға келесі алуға болатын дәрілерді алдын-ала көріп отыруға мүмкіндік береді.

А.1.1.1 Мақсаты

«Дәріханалар бойынша мобильдік қолданба агрегаторы» үшін Data Science шешімдері қолданба қолданушыларының ұнатуы мүмкін дәрілерді алып отыруға және дәріхана иелеріне кеңес беру жағынан уақыт үнемдеуге, жұмысын жеңілдетуге арналған.

А.1.1.2 Кодтың сыртқы түріне қойылатын талаптар

Кодтың жазылған мазмұны интуитивті түрде түсінікті және ыңғайлы болуы қажет.

А.1.1.3 Функционалдық сипаттамаларына қойылатын талаптар

Пайдаланушыларға:

- жаңа пайдаланушылар үшін қазіргі күнде танымал дәрілерді ұсыну;
- пайдаланушылар тарихына негізделген дәрілерді ұсыну;
- іздеген дәріге ұқсас дәрілер бойынша ұсыну;
- өзіне ұқсас пайдаланушылардың ұнатқан дәрілерін ұсыну.

А.1.1.4 Техникалық талаптар

Жүйе IBM үйлесімді дербес компьютерлерде жұмыс істеуі керек. Ең минималды конфигурация:

- процессор түрі – Pentium және одан жоғары;
- жад сыйымдылығы – 64 МБ және одан жоғары.

А.1.1.5 Бағдарламалық үйлесімділік талаптары

Жүйе Win32 API-ді іске асыратын Windows операциялық жүйелерінің бақылауында жұмыс істеуі керек.

Б Қосымшасы (міндетті)

Бағдарлама коды

Recommender System.ipynb қосымшаның негізгі бөлігі:

```
import numpy as np
import pandas as pd
import json
import matplotlib.pyplot as plt
import statistics
import operator
from pandas.io.json import json_normalize
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.metrics.pairwise import linear_kernel, cosine_similarity
from sklearn.neighbors import NearestNeighbors
from sklearn.cluster import KMeans
from sklearn.metrics import adjusted_rand_score, mean_squared_error
from rake_nltk import Rake
from math import sqrt

f = open(r'C:\Users\123\Desktop\DIPLOM\agregator.json',encoding='UTF8')
data = json.load(f)
df = pd.io.json.json_normalize(data)

rating_df=df[['model','pk', 'fields.author', 'fields.rating', 'fields.text',
'fields.product']]
rating_df=rating_df.rename(columns={'fields.author':'authorId',
'fields.rating':'Rating', 'fields.text':'Review', 'fields.product':'productId'},
inplace=False)
rating_df=rating_df.loc[rating_df['model'] == 'product.reviewproduct']

desc_df=df[['model', 'pk', 'fields.name', 'fields.composition',
'fields.description']]
desc_df=desc_df.loc[desc_df['model'] == 'product.product']
desc_df=desc_df.rename(columns={'pk':'productId', 'fields.name':'title',
'fields.composition': 'compound', 'fields.description':'desc'}, inplace=False)

result = pd.merge(rating_df, desc_df, on="productId", how="left")
result=result.drop(columns=['pk','model_x','model_y'])
result.head()

results = { }
```

Б қосымшаның жалғасы

```
tf = TfidfVectorizer(analyzer='word', ngram_range=(1, 3), min_df=0)
tfidf_matrix = tf.fit_transform(result['desc'])
cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
for idx, row in result.iterrows():
    uksas_index = cosine_similarities[idx].argsort()[:-100:-1]
    uksas_onim = [(cosine_similarities[idx][i], result['productId'][i]) for i in
uksas_index]
    uksas_onim = list( dict.fromkeys(uksas_onim) )
    results[row['productId']] = uksas_onim[1:]

def item(id):
    return result.loc[result['productId'] == id]['title'].tolist()[0].split(' - ')[0]
def recommend(item_id, num):
    print("Ұсынылушы " + str(num) + " өнім ұқсас " + item(item_id) + "...")
    print("-----")
    recs = results[item_id][:num]
    for rec in recs:
        print("Ұсынылады: " + item(rec[1]) + " (мәні: " + '{:.4f}'.format(rec[0]) +
)")")
recommend(8, 6)

item=[]
recommended=[]
def allrecommend(item_id, num=10):
    recs = results[item_id][:num]
    for rec in recs:
        item.append(item_id)
        recommended.append(rec[1])
for i in result.productId:
    allrecommend(i)

content_dariler= pd.DataFrame({'itemId':item, 'recommended':recommended})
content_dariler

json=content_dariler.to_json
(r'C:\Users\bmaralym\Desktop\DIPLOM\json\ContentBRS.json', orient='records',
force_ascii=False)
json

rate={ }
rows_indexes={ }
```

```
missing_pivot = result.pivot_table(values='Rating', index='authorId',
columns='title')
for i, row in missing_pivot.iterrows():
    rows= [x for x in range(0,len(missing_pivot.columns))]
    combine= list(zip(row.index, row.values, rows))
    rated= [(x,z) for x,y,z in combine if str(y) !='nan']
    index = [i[1] for i in rated]
    row_names = [i[0] for i in rated]
    rows_indexes[i] = index
    rate[i] = row_names

notrated={}
notrated_indexes={}
pivot_table = result.pivot_table(values= 'Rating', index='authorId',
columns='title').fillna(0)
pivot_table = pivot_table.apply(np.sign)
for i,row in pivot_table.iterrows():
    rows=[x for x in range (0,len(missing_pivot.columns))]
    combine= list(zip(row.index, row.values, rows))
    idx_row = [(idx,col) for idx,val,col in combine if not val>0]
    indices = [i[1] for i in idx_row]
    row_names = [i[0] for i in idx_row]
    notrated_indexes[i] = indices
    notrated[i] = row_names

n=5
items_dic = {}
topRecs = {}
cosine_knn = NearestNeighbors(n_neighbors=n, algorithm='brute',
metric='cosine')
item_cosine_knn_fit = cosine_knn.fit(pivot_table.T.values)
item_distances, item_indices =
item_cosine_knn_fit.kneighbors(pivot_table.T.values)
for i in range(len(pivot_table.T.index)):
    item_idx = item_indices[i]
    col_names = pivot_table.T.index[item_idx].tolist()
    items_dic[pivot_table.T.index[i]] = col_names
for k,v in rows_indexes.items():
    item_idx = [j for i in item_indices[v] for j in i]
    item_dist = [j for i in item_distances[v] for j in i]
    combine = list(zip(item_dist,item_idx))
    diction = {i:d for d,i in combine if i not in v}
```

Б қосымшаның жалғасы

```
zipped = list(zip(diction.keys(), diction.values()))
sort = sorted(zipped, key = lambda x: x[1])
recommendations = [(pivot_table.columns[i],d) for i,d in sort]
topRecs[k] = recommendations

def getrecommendations(user, number_of_recs = 30):
    if user > len(pivot_table.index):
        print('Тым көп {} қолданушы, тағы
көріңіз!'.format(len(pivot_table.index)))
    else:
        print("Сіз өткенде бұрында көрген өнім:
\n\n{}".format('\n'.join(rate[user])))
        print()
        print("Мына өнімдерді де қараңыз:\n")
        for k,v in topRecs.items():
            if user == k:
                for i in v[:number_of_recs]:
                    print('{} ұқсастық: {:.4f}'.format(i[0], 1-i[1]))
getrecommendations(42)

author=[]
values=[]

def allrecommendations(user, number_of_recs = 10):
    for k,v in topRecs.items():
        if user == k:
            for i in v[:number_of_recs]:
                author.append(str(k))
                values.append(i[0])

for i in result.authorId:
    allrecommendations(i)

recom_dariler= pd.DataFrame({'authorId':author, 'dari':values})
recom_dariler

json=recom_dariler.to_json
(r'C:\Users\bmaralym\Desktop\DIPLOM\json\ItemBRS.json', orient='records',
force_ascii=False)
json
```

```
    dari_boyinsha_rating =
pd.DataFrame(result.groupby('productId')['Rating'].count())
    dari_boyinsha_rating_filter = dari_boyinsha_rating
[dari_boyinsha_rating.Rating>=2]
    tanymal_dari = dari_boyinsha_rating_filter.index.tolist ()

    user_boyinsha_rating =
pd.DataFrame(result.groupby('authorId')['Rating'].count())
    user_boyinsha_rating_filter = user_boyinsha_rating
[user_boyinsha_rating.Rating>=5]
    userler = user_boyinsha_rating_filter.index.tolist ()
    rating_filtremen = result[result.productId.isin(tanymal_dari)]
    rating_filtremen = result[result.authorId.isin(userler)]
    rating_matrix = rating_filtremen.pivot_table(index='authorId',
columns='productId', values='Rating').fillna(0)
def uksas_users(authorId, matrix, k=3):
    user = matrix[matrix.index == authorId]
    baska_user = matrix[matrix.index != authorId]
    uksastyk = cosine_similarity(user, baska_user)[0].tolist()
    indices = baska_user.index.tolist()
    index_uksastyk = dict(zip(indices, uksastyk))
    index_uksastyk_sorted = sorted(index_uksastyk.items(),
key=operator.itemgetter(1))
    index_uksastyk_sorted.reverse()
    top_uksas_users = index_uksastyk_sorted[:k]
    users = [u[0] for u in top_uksas_users]
    return users
current_user = 9.0
uksas_user_indices = uksas_users(current_user, rating_matrix)

def recommend_item(user_index, uksas_user_indices, matrix, items=5):
    uksas_users = matrix[matrix.index.isin(uksas_user_indices)]
    uksas_users = uksas_users.mean(axis=0)
    uksas_users_df = pd.DataFrame(uksas_users, columns=['mean'])

    user_df = matrix[matrix.index == user_index]
    user_df_transposed = user_df.transpose()
    user_df_transposed.columns = ['Rating']
    user_df_transposed = user_df_transposed[user_df_transposed['Rating']==0]
    kormegen_dari = user_df_transposed.index.tolist()
```


Б қосымшаның жалғасы

```
uksas_users_df_filter =
uksas_users_df[uksas_users_df.index.isin(kormegen_dari)]
    uksas_users_df_ordered = uksas_users_df.sort_values(by=['mean'],
ascending=False)
    top_n_dari = uksas_users_df_ordered.head(items)
    top_n_dari_indices = top_n_dari.index.tolist()
    product_information =
desc_df[desc_df['productId'].isin(top_n_dari_indices)]
    return product_information
    json=product_information.to_json
(r'C:\Users\bmaralym\Desktop\DIPLOM\json\UserBRS.json', orient='records',
force_ascii=False)

recommend_item(10.0, uksas_user_indices, rating_matrix)
```

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ
МИНИСТРЛІГІ

СӘТБАЕВ УНИВЕРСИТЕТІ

5B070400 – «Есептеуіш техника және программамен қамтамасыз ету»

Батырбекқызы Маралым

Тақырыбы: «Дәріханалар бойынша мобильдік қолданба-агрегаторға
Data Science шешімдерін әзірлеу»

ҒЫЛЫМИ ЖЕТЕКШІНІҢ СЫН-ШІКІРІ

Студент Батырбекқызы М. алдына дәріханалар бойынша мобильдік қолданба үшін ұсыныс жүйелерін Data Science негізінде жасау міндеті қойылды. Бұл тақырып бүгінгі таңда маңызды, әрі өзекті тақырып болып табылады.

Дипломдық жобанда барлық қойылған тапсырмалар толықтай орындалды. Батырбекқызы М. замануи технологияларды қолдана отырып, жұмысын өз бетінше орындап шықты.

Дипломдық жұмыс кіріспе, 3 бөлімнен, қорытынды, қолданылған әдебиеттер тізімі мен қосымшалардан тұрады. Бірінші бөлімде жұмыстың мақсаттары, ұсыныс жүйелерінің жалпы сипаттамасы және анализ келтірілген. Екінші бөлімде ұсыныс жүйелерін құру үшін жобалау жұмыстары сипатталған. Қажетті бағдарламалау орталары салыстырыла отырып таңдалған. Үшінші бөлім дипломдық жұмыстың технологиялық бөлігі болып табылады. Код бөліктеріне шолу жасалынып, көрсетілген.

Дипломдық жоба барлық талаптарға сай орындалып, тақырыбы толықтай ашылған. Жұмысы теориялық және практикалық жағынан жақсы деңгейде орындалған.

Ғылыми жетекшісі ретінде дипломдық жобаны 5B070400 – «Есептеуіш техника және программамен қамтамасыз ету» мамандығының дипломдық жобаларының талаптарына сай деп есептей келе, Батырбекқызы М. дипломдық жобаны қорғауға ұсынамын және 5B070400 – «Есептеуіш техника және программамен қамтамасыз ету» мамандығы бойынша «Бакалавр» академиялық деңгейін тағайындауға лайық деп есептеймін.

Ғылыми жетекші: «Программалық инженерия» кафедрасының техн. ғыл.

магистрі, лектор  Д.А. Баймбетов

«04» маусым 2021 ж.



Метаданные

Название

Батырбекқызы_Маралым_дипломдық_жұмыс.docx

Автор

Батырбекқызы Маралым

Научный руководитель






Даулет Байымбетов

Подразделение

ИКИИТ

Список возможных попыток манипуляций с текстом

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

Замена букв		0
Интервалы		0
Микропробелы		0
Белые знаки		0
Парафразы (SmartMarks)		24

Объем найденных подобиий

Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



KPI1

25

Длина фразы для коэффициента подобия 2



KPI2

5350

Количество слов



KC

42521

Количество символов

Подобия по списку источников

Просмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и обратите внимание на то, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

10 самых длинных фраз

Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	https://towardsdatascience.com/build-a-user-based-collaborative-filtering-recommendation-engine-for-anime-92d35921f304	26	0.49 %
2	https://towardsdatascience.com/build-a-user-based-collaborative-filtering-recommendation-engine-for-anime-92d35921f304	21	0.39 %
3	https://towardsdatascience.com/build-a-user-based-collaborative-filtering-recommendation-engine-for-anime-92d35921f304	20	0.37 %
4	KazNAU/1080_2d7c3a30ebe55f4727dcc798f7cd79cf.docx KazNAU 5/19/2021 Kazakh National Agrarian University (KazHAY)	15	0.28 %

5	https://gist.github.com/venkarafa/0da815727f1ee098b201c371b60b2d72	15	0.28 %
6	https://towardsdatascience.com/build-a-user-based-collaborative-filtering-recommendation-engine-for-anime-92d35921f304	14	0.26 %
7	https://gist.github.com/venkarafa/0da815727f1ee098b201c371b60b2d72	14	0.26 %
8	https://predictivehacks.com/how-to-find-similar-documents-using-n-grams-and-word-embeddings/	13	0.24 %
9	https://towardsdatascience.com/build-a-user-based-collaborative-filtering-recommendation-engine-for-anime-92d35921f304	13	0.24 %
10	https://towardsdatascience.com/build-a-user-based-collaborative-filtering-recommendation-engine-for-anime-92d35921f304	12	0.22 %

из базы данных RefBooks (0.00 %)



ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из домашней базы данных (0.00 %)



ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из программы обмена базами данных (0.82 %)



ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	KazNAU/1080_2d7c3a30ebe55f4727dcc798f7cd79cf.docx KazNAU 5/19/2021 Kazakh National Agrarian University (KazHAY)	44 (5) 0.82 %

из интернета (6.75 %)



ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	https://towardsdatascience.com/build-a-user-based-collaborative-filtering-recommendation-engine-for-anime-92d35921f304	254 (26) 4.75 %
2	https://predictivehacks.com/how-to-find-similar-documents-using-n-grams-and-word-embeddings/	67 (8) 1.25 %
3	https://gist.github.com/venkarafa/0da815727f1ee098b201c371b60b2d72	40 (3) 0.75 %

Список принятых фрагментов (нет принятых фрагментов)

ПОРЯДКОВЫЙ НОМЕР	СОДЕРЖАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	------------	---

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Батырбекқызы М.

Название: Дәріханалар бойынша мобильдік қолданба-агрегаторға Data Science шешімдерін әзірлеу

Координатор: Сейтбекова Е.С.

Коэффициент подобия 1: 7.57

Коэффициент подобия 2: 0.49

Замена букв: 0

Интервалы: 0

Микропробелы: 0

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....

Дата 27.05.2021г

Подпись Научного руководителя



Форматы	Зонасы	Поз-сы.	Белгілеулер	Аталулары	Саны	Қоры-тынды	
			Дипломдық жоба				Бет
			40				
Өлш	Бет	Құжат №	қолы	күні			

