әл-Фараби атындағы Қазақ ұлттық университеті

ӘОЖ 004.45

Қолжазба құқығы негізінде

ДАРКЕНБАЕВ ДАУРЕН КАДЫРОВИЧ

Үлкен өлшемді деректі өңдеуге арналған сандық модельдеу және бағдарламалық қамтама құру

6D075100 – Информатика, есептеу техникасы және басқару

Философия докторы (PhD)
дәрежесін алу үшін дайындалған диссертация

Отандық ғылыми кеңесші
физика-математика
ғылымдарының докторы,
профессор Балакаева Г.Т.

Шетелдік ғылыми кеңесші
Phd философия докторы,
профессор Крис Филлипс

Қазақстан Республикасы
Алматы, 2020

МАЗМҰНЫ

НОРМАТИВТІК СІЛТЕМЕЛЕР

Бұл диссертацияда келесі стандарттарға сәйкес сілтемелер қолданылды:

ҚР МЖМБС 5.04.034–2011 «Қазақстан Республикасының Мемлекеттік жалпыға міндетті білім беру стандарты. Жоғары оқу орнынан кейінгі білім. Докторантура». Негізгі ережелер ҚР Білім және Ғылым министрімен бекітілген. «17» маусым 2011 ж. №261. Астана 2011.

«Диссертацияларды және авторефераттарды рәсімдеу бойынша нұсқаулық», ҚР БҒМ, Жоғары аттестаттау комитеті, Алматы, 2004. МЕСТ 7.1-2003. Библиографиялық жазба.

БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТУЛАР

РДҚБЖ – Реляциялық деректер қорын басқару жүйесі
BigData– үлкен өлшемді деректер
ДҚБЖ– Деректер қорын басқару жүйесі
NoSQL – Not only Structured Query Language
MongoDB – MongoDatabase
ЕКӘ – Ең кіші квадраттар әдісі
IBS – Information Business Systems
IBM – International Business Machines
SNA – Systems Network Architecture
IMDB – Internet Movie Database
RAID – Redundant Array of Independent Disks
MIMD – Multiple Insruction stream, Multiple Data stream
MISD – Multiple Instruction, Single Data
SIMD – Single Instruction, Multiple Data
SISD – Single Instruction , Single Data
LR – Linear Regression
LogR – Logistic Regression
NN – Neural Networks
HDFS – Hadoop Distribution File System
ROC - receiver operating characteristic
DataMining – деректерді қазбалау әдісі
Дефолт – несие бойынша міндеттемелердің орындалмау жағдайы

КІРІСПЕ

Зерттеу тақырыбының өзектілігі. Деректердің өсу қарқыны соңғы онжылдықта айтарлықтай біліне бастады. Зерттеулер көрсеткендей, соңғы екі онжылдықта әр екі жыл сайын деректер көлемі шамамен он есе өсті. Процессорлардың қуатын екі есе көбейтетін Мур заңынанда асып кетті[1]. Әр секунд сайын отыз мың гигабайтқа жуық дерек жинақталады және оларды өңдеу жылдамдықты қажет етеді. Әлеуметтік желілерде қолданушылары видеоларын, фотоларын және хаттарын жүктеулері үлкен өлшемді құрылымданбаған деректердің көптеп жинақталуына әкеледі[2]. Бізге әртүрлі форматтағы деректермен жұмыс істеуге тура келеді және жақсы нәтижелерге қол жеткізу үшін деректерді даярлаумен қатар, өңдеулерді модельдеу қажет[3]. Диссертациялық жұмыста жүргізілген зерттеулерден үлкен өлшемді деректерді өңдеуде модель мен алгоритмдер құру өзекті екенін көрсетіп отыр. Ақпараттық ағындар жыл сайын арта түсетіні және осыған байланысты үлкен өлшемді деректерді сақтау мен өңдеу мәселелерін шешу өзекті болатындығы сөзсіз[4]. Диссертация тақырыбының өзектілігі, қазіргі қоғамның көптеген салаларының цифрлануы және кәсіби қызметтердің онлайн режимге ауысуларының артуына байланысты болып отыр.

Диссертациялық жұмыста үлкен өлшемді деректерді өңдеудің моделін құру, ипотекалық несие беру мәселелерін шешуде Data Mining және машиналық оқыту әдістерін қолдану арқылы ипотекалық несие алатын жеке тұлғалардың төлем қабілеттерін анықтау және болжау бойынша зерттеулердің нәтижелері берілген. Диссертациялық жұмыс банк жүйесінің өзекті мәселелерінің бірі - ипотекалық несиелендіру жұмыстарын шешуге арналған. Негізгі мәселе деректерді интеллектуалды талдау әдісін қолдана отырып, ұзақ мерзімге ипотекалық несие алушылардың төлем қабілеттерін болжау болып отыр. Негізгі міндет - ипотекалық несие алушылардың төлем қабілеттерін анықтайтын жүйе негізінде үлкен өлшемді деректерді өңдеу үдерісін жүзеге асыру. Қазіргі уақытта ұзақ мерзімді ипотекалық несиелендіру қарқыны жыл сайын артып келеді, диссертациялық жұмыста ипотекалық несиелеу жүйесінің моделін құру бойынша зерттеулер өте өзекті және уақытылы жүргізілуі баспана алғысы келетін кез-келген адамның төлем қабілетін нақты болжап, тиісті шешімдер қабылдауға ықпал етеді.

Ипотекалық несиелеу мәселесін таңдау себебіміз, қазіргі кезде Қазақстан Республикасында ипотекалық несиелеу бағдарламалары жүзеге асырылып жатыр, бұл өз кезегінде ұзақ уақыт бойы ипотекалық несие алушылардың төлем қабілеттерін талдау, анықтау және болжау жүйесін дамытуды талап ететіні анық. Жеке тұлғалар туралы деректер мен олардың атрибуттарының көптігі, үлкен өлшемді деректерді өңдеу қажеттілігін туындатып отыр. Диссертациялық жұмыста ипотекалық несиелеу мәселелерін оңтайлы шешу үшін бағдарламалық қамтама құрылды, бағдарламалық қамтамада нейронды желі алгоритмдерінің

қолданылуы және салмақ коэффициенттері уақытылы жаңарып отыруы деректерді талдау жұмыстарын айтарлықтай жеңілдетті. Мысалы, ипотекалық несие беруші қаржы ұйымдарының жұмыстарына құрылған бағдарламалық қамтаманы енгізу қызмет көрсету сапасын арттырады, басқару жүйесін жаңартады, қауіпсіздік және т.б. мәселелердіде толық шешеді.

Көптеген несие беруші ұйымдар, атап айтқанда банктер және шағын несие беруші ұйымдар өз клиенттеріне несие беру немесе бермеу туралы шешім қабылдауда автоматтандырылған жүйе жұмысына жүгінеді, бұл жүйе клиенттердің басындағы сипаттамаларға негізделіп, несиелік балды есептеп сәйкесінше оң немесе теріс нәтиже береді [5]. Бұл жүйе орталық несиелік бюроға сұрау салып, несие алушының деректерін тексеріп, несиелік тарихын тексеру арқылы жүзеге асады. Егер, несие алып уақытында түрлі жағдайларға байланысты қайтарым жасай алмаған клиент үшін ұзақ мерзімге ипотекалық несие алу мүмкіндігі жоқ деп айтуға да болады. Бұл жұмыс ипотекалық несие алып баспаналы болғысы келген адамның арманы мен мүмкіндігін шектегендік болады. Сондықтан, диссертациялық жұмыста кез келген баспаналы болғысы келген адамның төлем қабілетін нақты болжап, сәйкесінше шешім қабылдайтын ипотекалық несие беру жүйесінің моделі құрылды. Әрбір адамның яғни ипотека беруші ұйымдардың клиенттерінің деректерін өңдеу заманауи технология мен сапалы техникалық жабдықтарды талап ететіні анық. Егер осы жүйе біздің елде қолданысқа енсе, көптеген жұмыстар жеңілдемек. Атап айтсақ, ипотекалық несие алушының деректері орталық дерекқордан алынып, нақты уақыт режимінде өңделіп, шешім қабылданатын болады. Тікелей ипотекалық несие беруші ұйымның қызметкері мен тұтынушы арасында байланыс болмағандықтан, түрлі заң бұзушылыққа жол берілмей, шынымен баспаналы болғысы келетін және төлем қабілеті сәйкес келетін азаматтар пәтерге ие болатын еді. Сонымен қатар, қашықтан өтінім беру – клиенттердің үйден шықпай, уақыт үнемдеумен қатар, қазіргі таңда ғасыр дерті болып отырған әлемдік пандемияның таралмауына да өз септігін тигізетіні анық. DataMining әдістерін пайдаланып, құрылатын бағдарламалардың барлығы заман талабынан туындайтын ойлар негізінде жүзеге асады. Ипотекалық несие беруші үлкен ұйымдар клиенттерінің деректері туралы жазбалар миллиондап сақталып тұр, ал жыл сайын клиенттері туралы жаңа деректерді сақтап, өңдейді. Бұл өз кезегінде шынайы деректерге бай жаңа жүйелік модель құруға үлкен септігін тигізеді. Ипотекалық несие алушының деректерін салыстырып өңдеп, нақты нәтиже алуымызға бұл таптырмас құрал деп айтуымызға негіз бар[6].

Жыл сайынға қаржы ұйымдарының арасындағы бәсекелестіктің артуы өздерінің тұтынушыларының деректерін жылдам өңдеп, қысқа уақытта сәйкесінше шешім қабылдауды қажет етіп отыр. Қолданыстағы бар жүйелер қаржы ұйымдарының талаптарын толық қанағаттандырып отырған жоқ. Диссертациялық жұмыста жүргізілген үлкен өлшемді деректерді өңдеуді

моделдеу және ипотекалық несие беру жүйесін құру жұмыстары, осы мәселелерді шешуге бағытталған өзекті зерттеу жұмыстары болып табылады.

Қазіргі қолданыстағы құрылғылар мен жүйелер қаржы беруші ұйымдардың талабын қанағаттандырмайды. Тікелей шетелдік дайын жүйені алып қолдануға болмайды, себебі ол шынайы деректерді талдау негізінде құрылған жүйе емес. Шетелдің өмір сүру деңгейі мен тұрғындарының айлық табысында айтарлықтай алшақтық бар, осының өзі – бізге шетелдің дайын жүйесін қолдана алмайтынымыздың бір дәлелі. Алайда, ол жүйелерді өзімізге икемдеуімізге болады, дегенмен барлық ипотекалық несие беруші ұйымдар ол жүйелерді сатып алуға қауқарсыз. Сатып алып қолданысқа енгізгеннің өзінде де жарты жылдан соң қолданыстан шығып қалу қаупі де басым. Көршілес Ресей Федерациясында ипотекалық несие алушылардың төлем қабілетін анықтайтын жүйе бар, бірақ көп емес, ал Қазақстан Республикасында аталмыш жүйе жоқ. Сол себепті жеке тұлғаның жалақысы туралы анықтама, жұмыс орнынан анықтама талап етеді. Сондықтан алдымызда, ипотекалық несие алушының деректерін жылдам өңдеп, макроэкономикалық ортаның факторларына байланысты өзгеріп отыратын жүйе құру міндеті тұр. Бұл міндеттің бір ғана шешімі бар, бұл – өзіміздің бар деректерімізді өңдеп, талдап, салыстыру нәтижелеріне сүйеніп, автоматтандырылған ипотекалық несие беру жүйесін құру.

Жұмыстың зерттелу деңгейі. Зерттеу жұмысының өзектілігі әлем ғалымдарының зерттеу тақырыбына байланысты жазған еңбектерінен айқындалады. Зерттеу тақырыбына ұқсас зерттеулер жүргізген алыс шетел ғалымдары Chung, H.M., Joyce Jackson, SrinivasanV., KimYong, Henley W. E., Desai V. S., Conway D. G., Crook J. еңбектеріне шолу жасалды. Ресей ғалымдары Н.В. Бабина, А.А. Земцов және Т.Ю. Осипова, В.Расторгуев секілді ғалымдардың еңбектеріне шолу жасалды. Отандық ғалымдар Калимолдаев М.Н., Амиргалиев Е.Н., Балакаева Г.Т., Мамырбаев Ө.Ж. және т.б. отандық ғалымдардыңда еңбектеріне шолу жасалды.

Қазіргі таңда жеке тұлғалардың төлем қабілеттерін анықтайтын, сәйкесінше балдық жүйемен төлем қабілеттерін бағалап, нәтиже беретін бағдарламалар бар. Мысалы, «SAS», «Deductor», «Statistica Data Miner», «Scorto» және т.б. бағдарламаларды айтуға болады. Дайын технологияларды пайдалану біздегі кездесетін қиындықтарды толық шешіп отырған жоқ және зерттеу жұмыстары да сәйкесінше деңгейде емес. Аталмыш бағдарламаларда сыртқы экономикалық факторлар ескерілмей қалған және өлшем коэффициенттері автоматтандырылмаған. Қолданыстағы бар бағдарламалардың математикалық модельдері коммерциялық құпияға байланысты ашылып жазылмайды.

Диссертациялық жұмыстың мақсаты. Үлкен өлшемді деректерді өңдеудің моделі мен алгоритмдерін құру, ипотекалық несие берудің қолданбалы міндеттерін шешуде талдау жүргізу және болжам жасау.

Зерттеудің міндеттері. Алға қойған мақсатқа жету үшін төмендегі міндеттерді шешу қажет:

1. Деректерді өңдеуге арналған әдістер мен өңдеу жүйелеріне сараптама жүргізу;

2. Data Mining әдістері: сызықты регрессия, логистикалық регрессия, көпқабатты нейронды желі негізінде үлкен өлшемді деректерді өңдеудің алгоритмдері мен моделін құру;

3. Үлкен өлшемді деректерді өңдеу жүйесінің жұмыс сапасын бағалау, деректерді тестілеу;

4. Құрылымданбаған үлкен өлшемді деректерді өңдеу негізінде жеке тұлғалардың төлем қабілеттеріне талдау жүргізу және болжам жасау;

Зерттеу нысаны. Ипотекалық несие алушылардың төлем қабілеттеріне талдау жүргізу және болжам жасау үшін үлкен өлшемді деректерді өңдеу жүйесін құру.

Зерттеу пәні. Үлкен өлшемді деректерді өңдеудің әдістері мен алгоритмдері.

Зерттеу әдісі. BigData теориясы және технологиялары, Data Mining әдістері: сызықты регрессия, логистикалық регрессия, нейрондық желілер. NoSQL технологиялары, машиналық оқыту алгоритмдері, бағдарламалық қамтаманы жобалау.

Жұмыстың ғылыми жаңалығы.

1. Үлкен өлшемді құрылымданбаған деректерді өңдеудің алгоритмдері құрылды;

2. Үлкен өлшемді құрылымданбаған деректерді өңдеудің сандық моделі құрылды;

3. Машиналық оқыту алгоритмдеріне шешілетін міндеттің форматына сай модификация жасалып, ипотекалық несие алушы жеке тұлғалардың төлем қабілеттеріне талдау жүргізу және болжам жасау механизмдері құрылды;

Жұмыстың теориялық және практикалық маңызы. Алынған нәтижелерді теориялық және практикалық тұрғыдан ипотекалық несие беруші қаржылық ұйымдардың жұмысын автоматтандыру үшін қолдануға болады.

Дүние жүзінде таралып отырған пандемия кезінде қаржы ұйымының қызметкерлері ипотекалық несие алушы азаматтардың төлем қабілеттерін қашықтан болжап, шешімін айта алады. Сонымен қатар, диссертациялық жұмыста жүргізілген зерттеулер ұзақ мерзімге ипотекалық несие алушыларға ипотекалық несие берілуі немесе берілмеуі туралы талдау жүргізіп, шешім қабылдауда өзінің септігін тигізеді. Құрылған жаңа жүйені азаматтардың төлем қабілеттерін анықтауға ғана емес, ғылымның басқа салаларына да қолдануға болады. Нақтырақ айтқанда, медицинада, геоинформатикада қолдануға болады, сонымен қатар, білім алушылардың оқу үлгерімдеріне болжам жасауға болады.

Қорғауға шығарылған негізгі тұжырым. Заманауи технологиялар (NoSQL, MongoDB) негізінде DataMining әдістерін қолдана отырып, үлкен өлшемді құрылымданбаған деректерді өңдеу жүйесінің моделі құрылды. Жеке тұлғалардың төлем қабілеттерін анықтауда машиналық оқыту алгоритмдеріне модификация жасалды. Дерекқорда тіркелген ипотекалық несие алушы жеке

тұлғалардың деректеріне талдау жүргізіліп, төлем қабілеттеріне болжам жасалды. Үлкен өлшемді құрылымданбаған деректерді өңдеуде көпқабатты нейронды желіні қолдану, жоғары нәтиже беретіні анықталды.

Зерттеушінің жеке үлесі. Ізденуші диссертациялық жұмыстың барлық міндеттерін шешті. DataMining, MongoDB және т.б. технологияларды зерттеді және мәселелерді шешуде қолданды. Ипотекалық несие беру мәселелерін шешуде талдау мен болжам жасау үшін, үлкен өлшемді құрылымданбаған деректерді өңдеу жүйесі құрылды.

Диссертацияның құрылымы мен көлемі. Диссертациялық жұмыс кіріспе, 3 тарау, қорытынды және пайдаланылған әдебиеттерден тұрады. Диссертацияның толық көлемі: 101 бет жазба мәтіні, соның ішінде 34 сурет, 10 кесте, 83 пайдаланылған әдебиеттер тізімінен және 3 қосымшадан тұрады.

Кіріспеде тақырыптың өзектілігі, диссертациялық жұмыстың мақсаты, міндеті, зерттеу әдістері айқын жазылған. Алынған нәтижелері, ғылыми жаңалығы мен маңызы сипатталған. Сонымен қатар аталмыш диссертациялық жұмыс бойынша жазылған мақалалар тізімі берілген.

Диссертациялық жұмыстың **бірінші тарауында** ипотекалық несие алушы жеке тұлғалардың төлем қабілетін анықтау мақсатында ғылыми еңбектерге шолу жасалып, үлкен көлемді деректерге яғни BigData-ға толық анықтама берілді. Сонымен қатар үлкен көлемді деректерді өңдеу және сақтау үшін қолданылып жүрген құрылғыларға шолу жасалды. Құрылымданбаған деректерді сақтауға арналған біріңғай дерекқор ретінде MongoDB дерекқоры таңдалып алынып оның қолданысқа енгізілуі туралы мәселелер қарастырылған.

Екінші тарауда ұзақ мерзімге ипотекалық несие алушы жеке тұлғалардың төлем қабілетін анықтайтын жүйе моделін құру әдістері талқыланады. Негізгі кезеңдер толық қарастырылды. Деректерді өңдеу жүйесінің моделін құруда Data Mining әдістерінің қалай қолданылғаны көрсетілді. Жеке тұлғалардың төлем қабілетін анықтау мақсатындағы жүйе құрудың негізгі міндеттері мен кездесетін қиындықтары анықталды. Диссертациялық жұмыстың жалпылама міндеттері ашылып жазылып, шешу әдістері ұсынылды.

Үшінші тарауда модификация жасалынған машиналық оқыту алгоритмдері жүзеге асырылды және DataMining әдістері негізінде бағдарламалық қамтама құрылған. Диссертациялық жұмыстың эксперименталды бөлімі көрсетілген. Алынған барлық нәтижелер кесте түрінде және суреттермен берілген.

Қорытындыда негізгі нәтижелер мен диссертациялық зерттеулердің қорытындылары берілген.

Зерттеу нәтижелерінің апробациясы. Ғылыми зерттеу жұмысының нәтижелері әл-Фараби атындағы Қазақ ұлттық университетінің информатика кафедрасының ғылыми семинарларында талқыланды және келесі халықаралық конференцияларда баяндамалар жасалды:

– XIV Miedzynarodowej naukowi-praktycznej konferencji «Naukowa przestrzeń Europy – 2018» (Прага, Чехия);

– V Студенттер және жас ғалымдардың «Фараби әлемі» атты халықаралық ғылыми конференциясы (2018, Алматы, Қазақстан);

– Международная конференция «Актуальные проблемы вычислительной и прикладной математики», «Марчуковские чтения – 2019» (Академгородок, Новосибирск, Россия);

– II Халықаралық ғылыми-практикалық интернет конференциясы «Заманауи зерттеулердің өзекті мәселелері» (2019, Нұр-Сұлтан, Қазақстан);

– Ф.К.Бойконың 100 жылдығына арналған «Ф.К.Бойко I мерейтойлық оқулары» атты халықаралық ғылыми-техникалық конференциясы (2020, Павлодар, Қазақстан)

Диссертация тақырыбы бойынша 13 мақала, авторлық куәлік және өндіріске енгізілгені туралы акт алынды:

1. Даркенбаев Д.Қ. Big Data. Үлкен көлемді деректермен жұмыс істеу қағидалары // ҚазҰПУ хабаршысы. – 2017. -№ 3 (59). – Б. 211-214.

2. Balakayeva G.T., Darkenbayev D.K., Chris Phillips. Investigation of technologies of processing of Big Data // Internation Journal of Mathematics and Physics. – 2017. – Vol.8. No.2. – P.13-18.

3. Balakayeva G.T., Darkenbayev D.K. Modeling the processing of a large amount of data// Al-Farabi Kazakh National University. Journal of Mathematics, Mechanics and Computer Science. – 2018. -Vol.1(97). – P.120 – 126.

4. Балақаева Г.Т., Даркенбаев Д.Қ. Үлкен өлшемді деректерді өңдеу үдерісін моделдеу // ҚазҰПУ хабаршысы. – 2018. -№ 1(61). – Б. 248-252.

5. Darkenbayev D.K. Numerical solution of the regression model for analysis and processing of Big Data//Vestnik KazNRTU. – 2018. – № 6(130).–P.132 – 139.

6. Balakayeva G.T., Darkenbayev D.K. Correlation and regression analysis for Big Data processing // Vestnik KazNRTU. – 2019. – № 1(131). – P.338 – 345.

7. Balakayeva G.T., Chris Phillips, Darkenbayev D.K., Turdaliyev M. Using NoSQL for processing unstructured Big Data // News of the National Academy of sciences of the Republic of Kazakhstan. – 2019. –Vol.6.No.438. – P. 12 – 21.

8. G. Balakayeva, D. Darkenbayev. The solution to the problem of processing Big Data using the example of assessing the solvency of borrowers // Journal of Theoritical and Applied Information Technology. – 2020. – Vol.98. No13.– P. 2659-2670. (Scopus).

9. Darkenbayev D.K. Increasing the efficiency of processing large-size data using Big SQL technology//Materialy XIV Miedzynarodowej naukowi-praktycznej konferencji, «Naukowa przestrzeń Europy - 2018».– Vol.10. – P. 50-55.

10. Даркенбаев Д.К. Повышение эффективности и применение новых технологий для обработки больших объемов данных //V Международные Фарабиевские чтения. – Алматы, 2018. – С. 215.

11. D. K. Darkenbayev, G. T.Balakayeva. Modeling big data processing using regression analysis // Марчуковские научные чтения – 2019. – Академгородок, Новосибирск, Россия. – С. 135.

12. Даркенбаев Д.Қ. Үлкен көлемді деректерді сақтау және талдау әдістері //Заманауи зерттеулердің өзекті мәселелері» II Халықаралық ғылыми - практикалық интернет конференциясы. – Нұр-Сұлтан, 2019. – Б.120-124.

13. Darkenbayev D.K. Building a linear regression model for processing Big Data in the definition of solvency of citizens// Материалы международной научно-технической конференции «I юбилейные чтения Бойко Ф. К», посвященной 100-летию Бойко Ф. К. – Павлодар, 2020. – С. 23-29.

Авторлық куәлік және өндіріске енгізілгені туралы акт алынды:

1. ЭВМ-ге арналған бағдарлама «NoSQL технологияларын және нейрондық желілерді қолданып үлкен көлемді деректерді өңдеу» авторлық куәлік № 8459 «28» ақпан 2020 жыл.
2. Диссертациялық жұмыстың нәтижелерінің өндіріске енгізілгені туралы акт.

1 ҮЛКЕН ӨЛШЕМДІ ДЕРЕКТЕРДІ ӨҢДЕУДІҢ ЗАМАНАУИ ЖАҒДАЙЫ ЖӘНЕ МӘСЕЛЕЛЕРІ

1.1 Үлкен өлшемді деректерді өңдеудің рөлі мен маңызы

Үлкен көлемді деректерді құрылымданған және құрылымданбаған деректерді қамтитын жалпыланған деректер жиынтығы деп түсіндіріледі, олар көлемі жағынан маңызды және құрылымы жағынан әр түрлі болады. Әдетте, үлкен өлшемді деректер негізінен құрылымданбаған деректердің алуан түрлерін үнемі жинақтау үдерісін білдіреді [7]. Жылдам өсіп келе жатқан деректер көлемі бізге оны сақтау мен өңдеуде күрделі жаңа міндеттерін жүктейді. Үлкен көлемді деректер дәстүрлі жүйелердің алдына күрделі міндеттерді туындатты. IDC компаниясының болжамы бойынша, 2020 жылғы сандық ғалам 40 зеттабайтты құрады, 2010 жылдан бері деректер көлемі 50 есе өскен[8]:

1. Нью-Йорк қор биржасы күніне терабайт деректер шығарады.

2. Facebook әлеуметтік желісінің деректерді сақтау қоймасының көлемі күніне 500 терабайтқа көбейіп отырады.

3. Internet Archive жобасы 2 петабайт деректерді сақтап отыр, айына ол деректер 20 терабайтқа көбейіп отыр.

4. Тәжірибе көрсеткендей үлкен адронды коллайдер секундына петабайт деректі шығара алады.

IBS компаниясының талдаушылары дүние жүзілік деректердің көлеміне төмендегідей болжам жасады:

2003 ж. — 5 эксабайт дерек (1 ЭБ = 1 млрд гигабайт)

2008 ж. — 0,18 зеттабайт дерек (1 ЗБ = 1024 эксабайт)

2015 ж. — 6,5 зеттабайттан астам

2020 ж. — 40–44 зеттабайт (жыл соңына дейінгі болжамы)

2025 ж. — бұл деректер 10 есе көбейеді.

Бизнес үдерістерді дамыту және ақпараттар ағынын ақылды цифрлік ресурсқа айналдыру жолында үлкен көлемді деректерді өңдеудің ауқымды мәселелері тұр. Үлкен өлшемді деректер ұғымы айтарлықтай қатаң анықтамаларды талап етпейді. Бұл ұғым экспоненциалды түрде өсетін, үлкен, өңделмеген және реляциялық дерекқор әдістерімен талдау үшін құрылымданбаған деректер жиынтығын сипаттайды. Түсіну үшін терабайт немесе петабайт секілді өсу көлемі емес, оны қалай өңдеу мәселесі өзекті болып тұр. «Үлкен деректер» ұғымын 2008 жылдың 3-ші қыркүйегінде «Nature» журналының редакторы Клиффорд Линч енгізді. Ол өзінің «Үлкен өлшемді деректер технологияларының ғылымның болашағына әсері» - деген мақаласында деректердің шамадан тыс өсіп келе жатқанын және оны өңдеудің болашақта қозғалатын өзекті мәселелердің бірі екенін ашып жазды [9].

Үлкен өлшемді деректердің ақпараттық құндылығы айқын. Үлкен өлшемді деректердің ақпараттық ағынын талдау арқылы шешілетін міндеттер келесідей:

- байланыс орталықтарының, техникалық қолдау қызметтері мен веб-сайттың трафигін талдау негізінде клиенттердің қарқынын болжау;

- болжамды модельдер құру;
- нақты уақыт режимінде алаяқтықты анықтау;
- тәуекелдерді талдау;
- жедел аналитикалық өңдеу және т.б. [10].

Аналогты форматтан цифрлі форматқа ауысу бизнес саласындағы деректердің күн сайын өсуіне әкелді. IDC зерттеулеріне қарағанда, 2010 жылы 1 триллион гигабайт дерек тіркелген. Бұл деректер бірнеше миллиардтаған телефондардың қолданылуына, әлеуметтік желілердегі ондаған миллиярд жарияланымдарға, автокөлікте интернетке қосылған датчиктардың санының артуына, сауда терминалдарының өсуіне және т.б. құрылғылардың санының көбеюіне тікелей байланыстыруға болады [11]. Көп жағдайларда, деректерді талдауды үлкен көлемді деректерді өңдеу деп қабылдап жатады. Шын мәнінде егер әртүрлі, құрылымданбаған деректер кездессе оның көлеміне аса мән берудің қажеті жоқ. Бүгінгі таңда кез келген кездесетін жағдайға сай үлкен көлемді деректерді өңдеудің универсалды талдау әдісі немесе алгоритмдері жоқ. Әр кезде өңделмеген шикі деректен қажетті білімімізді алатын кезде әрбір міндет үшін нақты әдісті және алгоритмдерді құру қажеттілігі туындайды. Әлемнің мамандары үлкен көлемді деректерді өңдеу әдістерінің дамуын зерттеп, оның кәсіпорындардың жұмысының болашағына қалай әсер ететінін зерттеуде. Қоймада деректерді ұзақ мерзімге сақтау түрлі қиындықтар туындатып қана қоймай, оларды дұрыс сақтап, өңдеп талдаған кезде кіріс әкелуі де мүмкін. 2020 жылда деректерді Жер шарының әрбір адамына шаққанда, карттар мен сәбилерді қосып есептегенде 5200 ГБ-тан келіп, соның 15% пайызы ғана бұлтты ортаға жазылады деп болжаған. (Digital Universe by Lucas Mearian болжамы).

Деректердің көлемі жыл сайын 2 есе өседі деп болжам жасалуда. IDC компаниясы 2020 жылдағы деректерді талдаса 33% пайызы құнды деректер деп пайымдап отыр. Big Data саласының мәселелерін зерттеп жүрген адамға деректерді өңдеу үшін 3 мәселеге ерекше назар аударуы қажет. Олар деректерді жинау, өңдеу сақтау мәселелері. Сондықтан, Big Data бұл кешенді жүйе деп айтуға болады. Әр жүйесінің өзіндік міндеті бар және олар басқа жүйелермен оңай интеграцияланады.

1.2 Үлкен өлшемді деректерді сақтауға және өңдеуге арналған қолданыста бар өнімдер

Деректерді өңдеу, жинақтау, құрылымдау мәселелерін шешуде Oracle Big Data Appliance қолданылады. Бұл Oracle NoSQL Database-тың алдын ала орнатылған Hadoop кластері және басқа деректер қоймаларымен біріге алатын құрал. Oracle Big Data Appliance міндеті құрылымданбаған немесе аз ғана құрылымданған деректерді өңдеуге, сақтауға негізделген. Бұл Hadoop дерекқорында аталмыш жұмыстардың жақсы орындалатынын көрсетеді[12].

Сурет 1.1 – Oracle Big Data Appliance-та деректерді сақтау үлгісі [13]

Ақпараттық технологиялар саласын зерттеп талдау жасайтын Gartner компаниясы өздерінің мақалаларында үлкен көлемді деректердің ең басты 3 сипаттамасын жазды және оны «үш V» деп белгіледі[14,15]:

- көлем (ағылшын тілінен аударғанда. volume) – сақталған деректердің физикалық көлемі;

- жылдамдық (ағылшын тілінен аударғанда velocity) – деректердің өзгеру жылдамдығы және бұл өзгерістерге артынша талдау жүргізу;

- көп қырлылық (ағылшын тілінен аударғанда variety) – өңделетін деректердің алуан түрлері яғни құрылымданған және құрылымданбаған деректер.

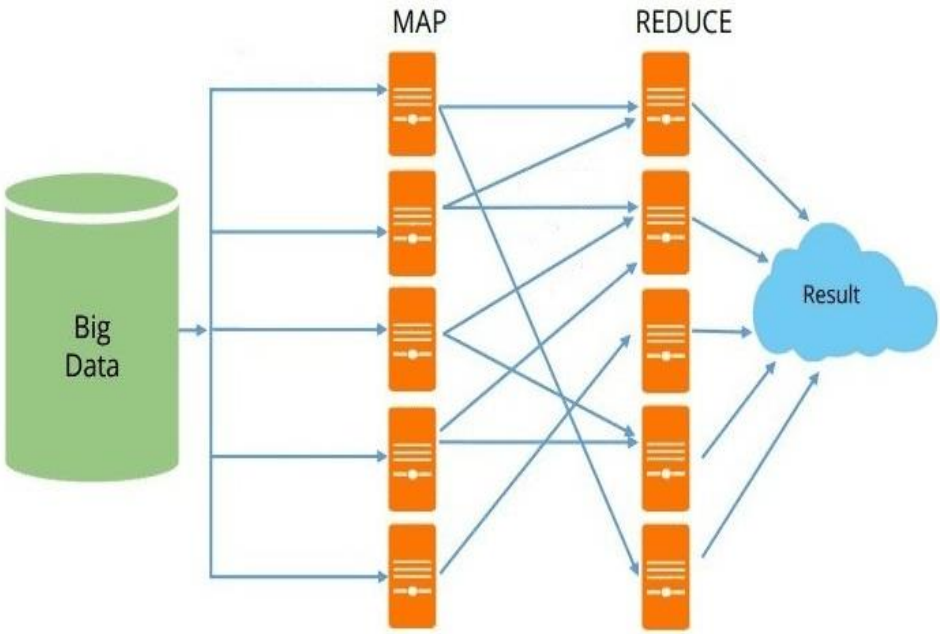
Үлкен өлшемді деректермен жұмыс істейтін моделдің жұмысы күннен күнге танымал болу үстінде. Ол Apache Software Foundation Apache Hadoop жобасында жүзеге асырылған. Apache Hadoop 2 компоненттен тұрады: Hadoop Distributed File System (HDFS) таратылған кластерлік жүйесі және Map Reduce бағдарламалық интерфейсі.

Hadoop - параллельді (MPP) деректерді жаппай өңдеуге арналған таратылған қосымшаларды құруға арналған бағдарламалық платформа. Hadoop платформасында екі негізгі компонентті шартты түрде бөлуге болады:

Hadoop Distribution File System (HDFS) - қолданбалы деректерге жоғары жылдамдықта қол жеткізуді қамтамасыз ететін үлестірілген файлдық жүйе;

MapReduce - бұл есептеу кластерінде үлкен өлшемді деректерді таратып, өңдеуге арналған бағдарламалық платформа [16].

Үлкен өлшемді деректерді өңдеу мәселесін шешу үшін NoSQL деректер қорының ерекше түрі жасалды. Реляциялық деректер базасы мен NoSQL қасиеттерін салыстыру 1.1- кестеде келтірілген.



Сурет 1.2 – Map-Reduce технологиясының картасы [16]

Реляциялық деректер қоры мен NoSQL қасиеттерін салыстырып кестеге түсірілді:

Кесте 1.1 – Дерекқорлар қасиеттерін салыстыру

Реляциялық деректер қоры	NoSQL деректер қоры
Деректердің күрделі қатынастары	Өте қарапайым қатынастар
Сызбалардың орталықтануы	Ерікті сызба: Құрылымданбаған деректер
Масштабталуы	Бөлінген өңдеулер
Статикалық жады	Жады есептеу ресурстарымен бірге масштабталады
Әмбебап функциялар мен қасиеттер	Жүйе қосымшаларға және құрастырушыларға бағытталған.

NoSQL технологиясы (мысалы, Cassandra) реляциялық дерекқорды ауыстыруға арналмаған, керісінше, деректер көлемі тым үлкен болған кезде мәселелерді шешуге көмектеседі. NoSQL көбінесе арзан стандартты серверлердің

кластерлерін пайдаланады. Бұл шешім секундына бір гигабайт құнын бірнеше есе азайтуға мүмкіндік береді [17].

SNA – жекелеген тораптардың өздерінің жадтары диск массивтері және енгізу, шығару құрылғылары бар тәуелсіз таратып есептеу архитектурасы. Бұл архитектураны 1974 жылы IBM жасаған [18].

1.3 Үлкен өлшемді деректерді аналитикалық өңдеу технологиялары

Бүгінгі таңда көптеген ұйымдарда үлкен көлемде деректер жинақталды. Қойылған міндеттерге байланысты бұл деректерді өңдеу үшін түрлі әдістер мен тәсілдерді қолдану қажет. Деректерді сақтауға арналған жабдықтардың кең спектрі бар, олардың кейбіреулері оларды кейінгі өңдеу мен талдаудың ыңғайлылығы үшін арнайы сақтау технологияларын қолданады: мультипроцессорлық жүйелер[19].

Мультипроцессинг (Multiprocessing, Multiprocessing, Eng. Multiprocessing) - бір компьютерлік жүйеде жұп немесе одан да көп физикалық процессорларды қолдану. Сондай-ақ, бұл жүйенің бірнеше процессорларды қолдау қабілеттілігін немесе олардың арасында міндеттерді бөлу қабілеттілігін де білдіреді. Бұл тұжырымдаманың көптеген нұсқалары бар және мультипроцессингтің анықтамасы контекстке байланысты өзгеруі мүмкін, негізінен процессорлар қалай анықталатынына байланысты .

Компьютерлік технологияның даму жылдарында мультипроцессорлық жүйелер дамудың бірқатар кезеңдерінен өтті. Тарихи тұрғыдан SIMD технологиясы бірінші болып игерілді. Алайда, қазір MIMD архитектурасына тұрақты қызығушылық пайда болды. Бұл қызығушылық негізінен екі фактормен анықталады:

SISD-мен өңделу мыналарды білдіреді: бір инструкция ағынымен және бірыңғай деректер ағынымен компьютерде бір процессор нұсқаулықтарды дәйекті түрде өңдейді; әр машинада бір деректер элементі өңделеді.

MIMD архитектурасы үлкен икемділікті қамтамасыз етеді: тиісті аппараттық және бағдарламалық қамтамасыз етумен MIMD бір қолданушы үшін, бір қосымшаның мәліметтерін жоғары өнімділікпен өңдеуді, параллельді түрде көптеген тапсырмаларды орындайтын көп бағдарламалы машина ретінде және осы мүмкіндіктердің кейбір үйлесімі ретінде жұмыс істей алады [20]. MIMD архитектурасы шығындар мен өнімділіктің қатаң талаптары негізінде, қазіргі заманғы микропроцессорлық технологияларды толықтай қолдана алады. Іс жүзінде қазіргі заманғы мультипроцессорлық жүйелердің барлығы дерлік жеке компьютерлерде, жұмыс станцияларында және шағын бір процессорлы серверлерде болатын бірдей микропроцессорларға салынған [21].

Компьютерде SIMD-мен бір уақытта бірнеше ағынмен және бірнеше ағынмен жұмыс істегенде, бір процессор нұсқаулар ағынын өңдейді, олардың әрқайсысы мәліметтер жиынында параллель есептеулер жүргізе алады. Бұл өңдеу компьютерлік модельдеуде кеңінен қолданылады, бірақ оны бизнес үдеріс үшін аз

пайдаланылады және ұсынылмайды. Архитектураның барлық ұсынылған мүмкіндіктерін пайдалана алу үшін бағдарламалар арнайы жазылып, әр міндет үшін жеке анықталуы керек.

Үшінші қолданыстағы мультипроцессорлық архитектура – MISD. Ол бірнеше нұсқаулық ағындарымен және бірыңғай мәліметтер ағынымен өңделетін процессор көбінесе артықшылықты сақтауды ұсынады, өйткені мультипрограммалау модульдері бірдей мәліметтер бойынша бір тапсырманы орындайды, егер модульдердің біреуі сәтсіз болса, нәтиже алу мүмкіндігін азайтады. MISD архитектурасы ақауды анықтау үшін есептеулер нәтижелерін салыстыруға мүмкіндік береді.

MIMD көп процессорлы өңдеу архитектурасы әртүрлі мәліметтер жиынтығына қатысты командаларды толығымен тәуелсіз және параллель орындауды жүзеге асыратын көптеген әртүрлі тапсырмалар үшін жарамды. Осы себепті және оны жүзеге асыру оңай болғандықтан, MIMD көп процестерде басым болады. Сыйымдылығы жоғары жедел жады, үлкен өлшемді деректерді өңдеуші құрылғылар қатарына жатады.

RAID - бірнеше құрылғылардан тұратын дисктер массиві (яғни күрделі) - қатты дискілер. Бұл массив деректерді сақтаудың сенімділігін арттыруға, ақпаратты оқу, жазу жылдамдығын арттыруға қызмет етеді [22]. Деректер қорлары шеңберінде үлкен көлемде деректермен жұмыс жасау өзекті мәселе болып табылады. Деректердің мөлшері өскен сайын, қатты дискілерімізде біршама көлемде қиындықтар туындауы мүмкін және ең бастысы, қажетті деректерге қол жеткізу уақыты болуы керек. Кэшті қолдануға болады, бірақ бұл, сайып келгенде, көмектеспейді. Деректер қорын бөлуге болады, әр сыныптың ақпаратын өз деректер қорына орналастыру. Деректер көлемі өскен сайын жүйенің жылдамдығы айтарлықтай төмендейді. Деректерге қол жеткізу уақытын қысқартудың бір әдісі - дерекқорды жедел жадқа орналастыру. Бұл әдіс жылдамдықты 100 есеге дейін арттыруға мүмкіндік береді [23]. Жадтағы дерекқорлар - IMDB - деректерді сақтау үшін компьютердің жедел жадын пайдаланатын деректер қоры. Оперативті жады мұндай жүйелердегі негізгі деректер қоймасы. Жадтың құны күн сайын азайып келе жатқандықтан, оны сақтау ретінде пайдалану мәліметтерді өңдеу жылдамдығын арттыру үшін тиімді болады.Үлкен өлшемді деректермен жұмыс істеуге арналған деректер қорларының жаңа түрлері бар, мысалы, аналитикалық деректер базасы. Бүгінгі таңда барлық дерлік мәліметтер базасында бұл тұжырымдама қолданылады [24]. Алайда, Terradata әзірлеушілері бірінші болып дерекқорға ендірілген талдауды жасады [25].

Сонымен қатар, мәліметтер базасының бір түрі - бағандық деректерді сақтау. Соңғы жылдары MonetDB [26,27] және C-Store [28] - ны қосқанда, бірқатар бағандық деректер қоры пайда болды. Бұл жүйелерді жасаушылар бұл тәсіл кейбір жүктемелерге, әсіресе деректер қоймасындағы қосымшаларға ұқсас

мәліметтерді оқуға арналған көп сұранысы бар аналитикалық жұмыс жүктемелерінің нәтижелерін арттырады деп мәлімдейді [29].

Data Mining алгоритмдерінің жиынтығы мен дерекқордың айырмашылығы, аналитикалық платформалар бастапқыда деректерді талдауға бағдарланған және дайын аналитикалық шешімдер жасауға арналған.

Аналитикалық платформа - бұл ақпараттық аналитикалық жүйе, сонымен қатар «шикі» деректерден шаблондар алу үдерісін жүргізуге арналған барлық құралдарды қамтитын мамандандырылған бағдарламалық шешім, бүкіл деректер массивінен кейбір шаблондарды алу үдерісі жүзеге асырылады: ақпаратты бір көзде шоғырландыру құралдары, деректерді алу, түрлендіру, сақтау, ақпаратты алу алгоритмдері, визуализация, қарапайым және күрделі әдістер мен моделдердің ауысуы [30]. Үлкен өлшемді деректерді дұрыс өңдеу үшін бірыңғай әдіс немесе алгоритм жоқ. Сәйкесінше, әр тапсырма өзінің орындалу алгоритмін, деректерді талдау алгоритмін көздейді. Үлкен өлшемді деректерді өңдеуге арналған алгоритмдер көп, алайда белгілі бір мақсаттарға жету үшін жаңа алгоритмдер жасау қажет.

2.4 NoSQL дерекқорының түрлері

NoSQL - деректер қорын басқару тетіктерін іске асыруға бағытталған, SQL тілі арқылы ақпаратқа қол жетімділіктің әдеттегі модельдерінен едәуір ерекшеленетін жалпы сипаттама. NoSQL икемді күйімен сипатталады, ол уақыт бойынша өзгере алады және әр сұраныс үшін қол жетімді. NoSQL деректер қорын барлық ақпараттық модельдерді - мәтін, графика, құжаттарды кілттер мәнінің жұбын қолдана отырып пайдалануға болады. NoSQL термині бойынша әр түрлі деректер қорларын табуға болады, бірақ барлығына бірдей сипаттамалар бар. Деректермен жұмыс істеу сипатына қарай дерекқорларды қалауымызша таңдап жұмыс жасауға болады.

Құжаттар дерекқоры әр кілтті құжат деп аталатын күрделі мәліметтер құрылымымен байланыстырады. Құжат кілт – мән жұптарының жиынтығы. MongoDB – бұл құжаттар қоймасының деректер қорының мысалы. MongoDB құжаттар тобы коллекция деп аталады. Бұл ДҚБЖ кестесінің баламасы.

Графикалық қоймалар әлеуметтік байланыстар сияқты деректер желілері туралы ақпаратты сақтау үшін қолданылады. Графикалық дүкендерге Neo4J және Giraph кіреді.

Деректер базасының "кілт мәні" қоймасы әр жеке элементті дерекқорда кілт ретінде, оның мәнімен бірге сақтайды. "Кілт-мән" сақтау мысалдары: Riak және Berkeley DB. Redis сияқты кілттер мен мәндердің кейбір қоймалары әр мәнге функционалдылықты қосатын бүтін сан сияқты түрге ие болуға мүмкіндік береді.

Cassandra және HBase сияқты кең бағандары бар қоймалар үлкен деректер жиынтығына сұрау салу үшін оңтайландырылған және олар жолдар емес, деректер бағандарын бірге сақтайды[31].

1.5 Үлкен өлшемді деректерді сақтауда MongoDB қолданылуы

MongoDB – NoSQL технологиясының C++ бағдарламалау тілінде жазылған алдыңғы қатарлы дерекқоры. Python тілінде жазылған кез келген бағдарламалық қосымшалармен интергацияланады. SQL мен NoSQL айырмашылығына тоқтала кетелік**.** SQL дерекқорлары деректерді анықтау және манипуляциялау үшін құрылымдық сұраныстар тілін (SQL) қолданады. SQL-ді қолданған кезде бізге SQL Server, MySQL Server немесе MS Access сияқты мәліметтер қорын реляциялық басқару жүйесі (ДҚБЖ) қажет болады. ДҚБЖ-да деректер кесте деп аталатын деректер қорының объектілерінде сақталады [32]. Кесте дегеніміз - бағандар мен жолдардан құралған байланысты деректер жазбаларының жиынтығы.

NoSQL дерекқорының құрылмданбаған деректерге арналған динамикалық схемасы бар. NoSQL – де деректер бірнеше әдіспен сақталады: олар бағанға, құжатқа, графтарға немесе кілт және мән сақтауға негізделуі мүмкін. NoSQL дерекқорының келесідей басымдылықтары бар:

– құжаттарды құрылымын алдын ала анықтаусыз-ақ құра беруге болады;
– әрбір құжаттың өзінің бірегей құрылымы болуы мүмкін;
– деректер қорының синтаксисінде басқа деректер қорынан айырмашылық болуы мүмкін;
– үлкен көлемді құрылымданған, жартылай құрылымданған және құрылымданбаған деректерді сақтай алады;
– объектіге бағытталған бағдарламалау қолдануға жеңіл әрі ықшамды;
– горизантальді масштабталуы;

MongoDB-де қолданылатын терминдер туралы толық түсінік алу үшін біз оларды РДҚБЖ-дағы баламамен салыстырамыз:

Кесте 1.2 – РДҚБЖ мен MongoDB - ді салыстыру

РДҚБЖ	MongoDB
Деректер қоры	Деректер қоры
Үстел	Коллекция
Жол	Құжат
Баған	Өріс
Негізгі кілт	Негізгі кілт
Кестеге қосылу	Кірістірілген құжаттар

MongoDB – де деректер қорын құру үшін біз MongoClient данасын қолданамыз және деректер қорының қорның атын көрсеткеннен кейін MongoDB деректер қорын құрып жұмыс жасайды:

db=client[datacampdb]

Айта кететін жайттардың бірі, деректер қорымен мен коллекциялар MongoDB – де баяу құрылады. Бұл коллекциялар мен деректер қоры бірінші құжат енгізілген кезде бірақ жасалады дегенді білдіреді. MongoDB деректері JSON стиліндегі құжаттарды қолдана отырып ұсынылған және сақталған . PyMongo-да біз құжаттарды ұсыну үшін сөздіктерді қолданамыз. Төменде PyMongo құжатының мысалын көрсетейік:

dissertation ={“author”: “Darkenbayev Dauren”,
“about”: “MongoDB and Python”; “tags”:[“mongodb”, “python”, “pymongo”]}

MongoDB дерекқорын басқаруда коллекцияға ат қойып қосымша мәліметтер қосумен қатар өшіруге де болады. Қауіпсіздік мәселесі де толық қамтылған үлкен көлемді құрылымданбаған деректерімізді реляциялық емес дерекқор MongoDB – де толық басқарып қолдана аламыз.

Бірінші тарау бойынша қорытынды

Бұл тарауда үлкен өлшемді деректерді өңдеудің өзекті мәселелері және ақпараттық технологиялар саласындағы алпауыт компаниялардың деректерді сақтау және өңдеу технологиялары зерттелді. Шетелдік және отандық жетекші ғалымдардың еңбектеріне шолу жасалды, мәселенің өзектілігі ашылып жазылды. Дүние жүзінде адам санының өсуімен қатар деректер көлемі де қарқынды өсу үстінде. Деректер форматы әртүрлі болғандықтан, оларды өңдеу қиындықтар туғызатыны анық. Құрылымданбаған деректерді сақтау үшін реляциялық емес деректер қорын пайдалану өңдеу үдерісін шешуде оң нәтиже көрсетті. Шетелдік ғылыми кеңесшім Ньюкасл университетінің профессоры (Ұлыбритания) Крис Филлипстың пайымдауынша, үлкен өлшемді деректерді өңдеудің өзіндік қағидасы қалыптасқан жоқ, сондықтан қалай өңдеу керек екені туралы түрлі пікірталастар бар. Мәселені шешуде машиналық оқыту алгоритмдерін қолдану жақсы нәтижелерге қол жеткізуге болатынын түсіндірген болатын. Отандық ғылыми кеңесшім ф.-м.ғ.д., профессор Г.Т. Балакаевамен бірге DataMining және NoSQL технологияларына зерттеулер жүргізілді. Алдағы тарауларда технологиялардың интеграцияланғаны және олардың ұзақ мерзімге ипотекалық несие алушы жеке тұлғалардың төлем қабілеттерін бағалауда және болжам жасауда қолданылғаны толық ашылып жазылған.

2 DATA MINING ӘДІСТЕРІНІҢ ҰЗАҚ МЕРЗІМГЕ ИПОТЕКАЛЫҚ НЕСИЕ АЛУШЫ ЖЕКЕ ТҰЛҒАЛАРДЫҢ ҚҰРЫЛЫМДАНБАҒАН ДЕРЕКТЕРІН ӨҢДЕУДЕ ҚОЛДАНЫЛУЫ

2.1Ұзақ мерзімге ипотекалық несие беру жүйесінің математикалық моделі

DataMining – бұл деректерді қазбалау, талдау, белгісіз деректерді табу технологиясы. Адам өміріндегі түрлі салаларда шешім қабылдауда өз ықпалын тигізеді. Басты міндеттерінің бірі-деректерді классификациялау, болжам жасау болып табылады.

Қазіргі кезде ипотекалық несиені қаржы ұйымдары тұтынушыларға кеңінен ұсынып, тұрғын үй мәселесін едәуір шешуде. Алайда қаржы ұйымы осы ипотекалық несиені ұсыну барысында кейбір келеңсіздіктермен ұшырасуда. Бұл несие тәуекелділік деген ұғыммен ұштасады. Банктер қаржы беру барысында делдал болып табылады, сондықтан ол өзіне несие тәуекелділігін қабылдайды. Несиенің өз уақытында қайтарылмауы банктің зиян шегуіне немесе банкроттыққа алып келеді. Несиелік тәуекел – бұл қарыз алушының қаржылық жағдайының төмендеуімен немесе банкроттылығымен және несиенің толық немесе бір бөлігін, олар бойынша пайыздар сомасын қайтара алмаумен байланысты тәуекел. Ол ішкі ортаның әсеріне қатысты болады. Банк қызметкерлерінің клиенттердің төлем қабілеттерін дұрыс анықтап болжай алмауы салдарынан несие қайтару қабілеттілігі жоқ клиентке несие берілу себептері туындайды[33]. Осының бәрі несиенің бір түрі болғандықтан, ипотекалық несиеге де тікелей байланысты. Ипотекалық несие тәуекелі өте жоғары болып келеді, өйткені ол жерде көп сомма қаражат беріледі және оның пайызы да жоғары болады. Ең негізгісі несиенің ұзақ мерзімге берілгендіктен қайтарылу тәуекелділігі өте жоғары. Бұл дегеніміз қаржылық ұйымдардың өз тұтынушыларының төлем қабілетін нақты анықтап, болжам жасау жұмыстарын жоғарғы деңгейге көтеруін талап ететіні анық.

Макроэкономикалық көрсеткіштер немесе экономикалық, әлеуметтік, демографиялық факторлар өзгерген кезде қарыз алушылардың төлем қабілетсіздігі қаупі туындауы мүмкін, сондай-ақ қарыз алушы табысының өзгеру қаупі артады. Бұл жағдай транзакциялық тәуекел болып табылады. Қарыздардың үлкен топтарын ұқсастық, бірдей өнімдер, бірдей кепілдік және т.б. принципі бойынша бір «үлкен несиеге» біріктіруге болады. Бұл топ портфолио деп аталады. Несиелерді бір топқа біріктіру қажеттілігі, атап айтқанда портфель басқару шығындарын азайту қажеттілігімен байланысты: сәйкесінше, бір портфельді алу нәтижесінде портфельді бір үлкен несие ретінде басқаруға болады деп болжанады[34]. Бірақ содан кейін мұндай «үлкен несие» өзіне тән тәуекелді – портфелдік тәуекелді бағалауға мүмкіндік беретін параметрлермен сипатталуы керек. Портфолиоға тәуекелдің біркелкі факторларының әсерінен туындаған несиелер кіреді, олардың арасында экономикалық (мысалы, саладағы сұраныс жағдайы) және әлеуметтік факторлары бар. Иерархияның үшінші деңгейі деп бөлу несиелік тәуекелі деп аталады - бұл банк активтерін сала, қатысу аймағы

және банк өнімдері бойынша бөлу кезінде туындайтын тәуекел. Дамудың әр түрлі динамикасы және аймақтық экономика, секторлардың әр түрлі жағдайлары, мысалы, банк несиелерінің түрлеріне сұраныс банк қалыптастырған несие портфельдерінің сапасының өзгергіштігін анықтайды [35]. Жеке тұлғаның төлем қабілетін анықтайтын жүйе - транзакциялық несиелік бағалау жүйесін және басқаруды қамтамасыз етуге мүмкіндік беретін әдіс.

Жеке тұлғаның төлем қабілетін анықтайтын жүйе – бұл математикалық немесе статистикалық модель, оның көмегімен банктің қызметін пайдаланған клиенттердің несиелік тарихына сүйене отырып, соңғысы клиенттің несиені уақтылы қайтару ықтималдығы неде екенін анықтауға тырысады, яғни, бұл ықтимал қарыз алушының оны несиелеу туралы мәселені қарастыру кезінде банкроттыққа ұшырау ықтималдығының диагнозы [36]. Аталмыш жүйені қолдана отырып, клиентті бағалау нәтижесі – кешіктіруге рұқсаты бар клиентке белгілі бір рейтинг беру.

Ипотекалық несие беру кезіндегі жеке тұлғаның төлем қабілетін анықтайтын жүйенің негізгі құралы – тұлғаның жеке картасы. Қарыз алушының сипаттамаларын сандық мәндермен салыстыруға және сайып келгенде, ұпай рейтингін алуға мүмкіндік беретін математикалық модель. Үлкен өлшемді деректерді өңдеуді қолданудың жарқын мысалдарының бірі - банк жүйесінің әрекетінің тәуекелділігіне талдау үшін негіз болатын тұтынушының төлем қабілетін айқындайтын карта.

Банктің міндеттері үлкен мөлшерде деректерді өңдеу алгоритмдерін қамтитын банктік тәуекелдерді басқару жүйесінің скорингтік модельдерін жасаудың негізгі әдістері мен технологиялары болып табылады. Бұл міндеттер банк бөлімшелеріндегі өтінімдердің құжат айналымын оңтайландыру және қолданыстағы технологияларды қолдана отырып, дұрыс және барабар скоринг моделін құру және жаңа алгоритмдерді құру арқылы шешіледі.

Жеке тұлғалардың төлем қабілеттерін анықтайтын жүйе модельдерінің құрылысын автоматтандыру маңызды рөл атқарады. Ықтимал ипоткалық несие алушыға, жеке немесе заңды тұлғаға несие беру туралы шешім қабылдағанға дейін сенімділікті тексеріп, қаржылық жағдайды бағалаудың бір бағыты - скорингтік бағалау. «Scoring» термині ағылшынша Score сөзінен шыққан, ол ұпай, ойындағы ұпай, қарыз сомасы, себеп сияқты мағыналарға ие[37].

Ықтимал ипотекалық несие алушы азаматтар туралы ақпараттың түріне байланысты олардың төлем қабілетін анықтау келесі түрлері бөлінеді [38]:

- Ипотекалық несие алу үшін өтінім беру, енгізген деректеріне сәйкес жаңа несие беру туралы шешім қабылдау;

- Ипотекалық несие алушы азаматтардың жеке басының мінез-құлықын бағалау яғни олардың төлем қабілеттіліктерін анықтау, шоттарындағы операциялардың тарихы (қарызды өтеу кестесі, ағымдағы шоттардағы айналым, жаңа қарыздарға сұраныс) туралы мәліметтер негізінде динамикалық бағалау. Бағалау нәтижелері бойынша ипоткалық несие алушы азаматтар үшін ағымдағы

несие лимитін анықтауға болады. Төлемдер кешіктірілген жағдайда қабылданған шаралар клиентке кері әсерін тигізуі мүмкін. Төлем қабілетін анықтау жүйесін несиелік өнімді сату кезеңінде ғана емес, сонымен қатар оны жобалау кезінде де қолдануға болады, оның негізінде өнім жобаланған және тәуекелді төмендетуге ықпал ететін ипотекалық несие алушылардың негізгі қасиеттерін атап, негізгі маркетингтік күш-жігерді бағыттауға болады.

Банк жүйесінде адам ипоткалық несие алуға жүгінген кезде, банк талдау үшін келесі ақпаратқа ие болуы мүмкін:

- қарыз алушы толтырған өтініш нысаны;
- несие бюросынан осы әрбір несие алушы туралы ақпарат;
- шоттың қозғалысы туралы мәліметтер

Ең жеңілдетілген түрдегі төлем қабілетті анықтау моделі белгілі бір сипаттамалардың өлшенген қосындысы болып табылады. Нәтиже интегралды көрсеткіш – балл болып табылады, ол соғұрлым жоғары болған сайын, клиенттің сенімділігі соғұрлым жоғары болады, ал банк өз клиенттерін несие қабілеттілігінің өсу дәрежесі бойынша сұрыптай алады. Әрбір клиенттің интегралды көрсеткіші белгілі бір сандық шекпен немесе бөлу сызығымен салыстырылады, ол мәні бойынша үзіліс сызығы болып табылады және бір борышкерден шығынды өтеу үшін уақытында төлейтін орташа клиенттердің қатынасы негізінде есептеледі. Несиелер осы сызықтан жоғары интегралды индикаторы бар клиенттерге беріледі, бірақ индикаторы осы сызықтан төмен клиенттерге берілмейді. Біздің зерттеулерімізде ұзақ мерзімге ипотекалық несие алушы азаматтардың төлем қабілетін бағалауда екі топқа бөлінді, несие беруге болатын және несие беруге болмайтын. Клиенттерді классификацияланды.

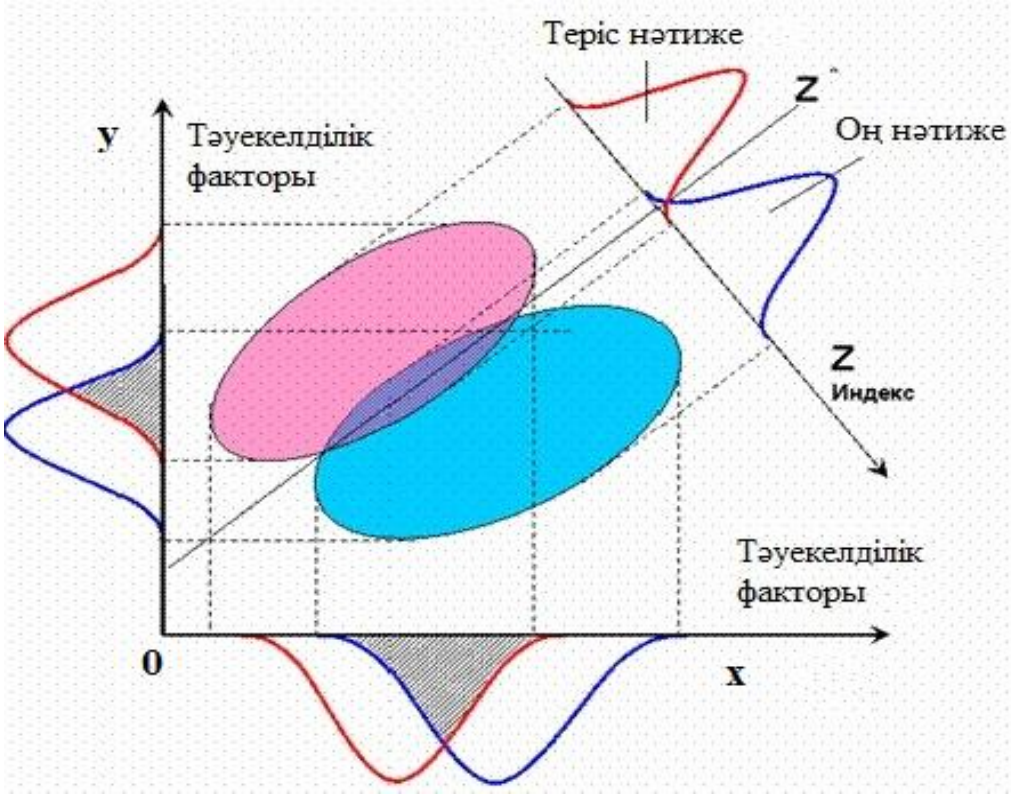
Несиелік тәуекелді бағалау үшін келесі сипаттамалар жиі қолданылады:

- Жасы
- Балаларының саны
- Мамандағы
- Жұбайының мамандығы
- Айлық табысы
- Жұбайының айлық табысы
- Тұрғылықты мекен-жайы
- Тұрғын үйдің бағасы
- Байланыс телефонының номері

Ипотекалық несие алушы азаматтардың төлем қабілетін анықтайтын жүйе - бұл қол жетімді ақпаратқа сүйене отырып, клиенттердің төлем қабілеттерін «теріс» және «оң» деп нақты бөліп есептеп классификациялық есепті шешеді.

Жеке тұлғаның төлем қабілетін бағалау - бұл біз үшін барлық ипотекалық несие алушы азаматтарды әртүрлі топтарға жіктеуге мүмкіндік береді. 2.1 - суретте көрсетілгендей ипоткалық несиелерді клиенттерінің нәтижелері бойынша «оң» және «теріс» нәтижелі деп екіге бөлінген. Оң нәтижелі клиенттердің төлем қабілеттері 0-ге тең болады, ал теріс нәтижелі клиенттердің төлем қабілеттері 1-ге

тең болады. Клиент несиені қайтаратынын немесе қайтара алмауын алдын-ала білмейміз, бірақ клиенттің қай топқа бөліну керектігін анықтауға көмектесетін басқа да фактілерді білеміз. Статистикада популяцияны топтарға бөлу идеяларын Фишер 1936 жылы өсімдіктерді мысал ретінде қолдана отырып жасаған[39]. Дәл осы әдісті алғаш рет 1941 жылы Дэвид Дюран несие бойынша кешіктірулердің бар, жоғына байланысты, яғни несиелерді «теріс» және «оң» нәтижелі деп жіктеген[40].



Сурет 2.1 – Дэвид Дюран ұсынған несие алушы азаматтардың төлем қабілеттерінің геометриялық интерпретациясы

Суретте екі сыныпқа жататын ипотекалық несие алушылар жұмыртқа тәрізді сопақшамен бөлініп тұр. Жоғарғы сопақшада теріс нәтижелі, ал төменгі сопақша оң нәтижелі ипоткалық несие алушы азаматтарды көрсетілген. Графикте остер бойымен несиелік тәуекелділік факторлары X тәуелсіз және Y тәуелді айнымалылары берілген. Бұрын өңделген ипотекалық несиелер статистикасын қолдана отырып, төлем қабілетті анықтау моделі кеңістіктегі деректерді тәуекелділік факторы ретінде іздейді (2.1 – суретте бұл кеңістік екі өлшемді, ал тұтастай алғанда ол көп өлшемді), сондықтан әртүрлі көзқарастар тұрғысынан әр түрлі кластардың объектілері бір-бірінен мүмкіндігінше ерекшеленеді. Суретте көру бұрышы екі сопақша арасынан өткен түзу сызықпен сызылған. Бұл сызыққа перпендикуляр сызылған үзік сызықтар ипоткалық несие алушыларды оң нәтижелі және теріс нәтижелі екенін бірден көруімізге мүмкіндік береді. Әр түрлі кластардағы ипотекалық несие алушылардың тығыздық функциясы Z бағалаушы

оске бағытталған кезде бір-бірінен өзгеше болады. Тәуекелділік факторының моделін бағалайтын коэффициенттердің сандық мәні пайда болады. Бұл коэффициенттер оқыту процедурасының нәтижесі болып табылады, оған моделді құру үшін қолда бар статистикалық мәліметтер ұсынылған кезде ол ипотекалық несие алушылардың санаттарын танудың дәлдігі максималды болатындай коэффициенттерді таңдайды.

Жеке тұлғаларды несиелеуде төлем қабілетті анықтау маңызды және бұл бірнеше міндеттерді атқаратын жүйе. Классикалық нұсқада оған келесі элементтер кіреді: анкеталарды қашықтықтан толтыруға арналған интерфейс, құжат айналымы схемасы, бағалау, күзет және несие инспекторы үшін жұмыс орындары, құжаттар пакетін автоматты түрде құру және бухгалтерлік есеп жүйесімен біріктіру. Дәл осы тізбектегі барлық сілтемелерді жүзеге асыру ипотекалық несие алушы азаматтардың төлем қабілеттерін анықтайтын жүйенің тиімді шешімін құруға мүмкіндік береді, бірақ бұл технология бөлек іске аспайды. Сонымен қатар, несие өтініштерін қарау барысында пайда болатын бизнес-процестердің кесімді сипаты өтініштерді қабылдау уақыты банк бөлімшелерінің өзара іс-қимылына өте тәуелді екендігіне әкеледі. Сондықтан жүйелік тәсілді қолдана отырып, несие берудегі шешімін қолдану, бизнес-үдерістерді қайта үйлестіру, басқаруға технологиялық көзқарас күрделі және шұғыл міндет болып табылады.

Жоғарыда аталған барлық элементтерді қамтитын тұтынушылық төлем қабілеттеріне қарай несиелеу жүйесінің дамуын талдау. Төлем қабілетін анықтайтын жүйе моделдері экономикалық тәжірибеде жеке және заңды тұлғалардың несиелік қабілеттілігін, банкроттық тәуекелін және басқа да мәселелерді бағалау кезінде қолданылады. Жалпы алғанда, жүйенің математикалық моделі келесідей:

$$\overline{Y_i} = p_1 t_1 + p_2 t_2 + ... + p_n t_n \qquad (2.1)$$

мұндағы $\overline{Y_i}$ - объектінің жалпыланған бағалауының мәні; t_1, t_2, t_n – бағаланатын объектінің талданатын сипаттамасына әсер ететін факторлардың нормаланған мәні, $p_1, p_2, ... p_n$ – сарапшылар үшін тиісті факторлардың маңыздылығын сипаттайтын салмақтары. Төлем қабілетті анықтайтын жүйенің моделді қарапайым әртүрлі сипаттамаларының өлшенген қосынды болып табылады, нәтижесінде алынған интегралды көрсеткіш таңдалған шекті мәнмен салыстырылады, соның негізінде несие беру немесе бермеу туралы шешім қабылданады. Бұл кімге ипоткалық несие бер немесе бермеу туралы шешімді шығарады. Сонымен, аталмыш жүйенің моделінің мәні қарапайым болғанымен сыртқы қарапайымдылықтың артында бірқатар қиындықтар жатыр. Төлем қабілетті анықтау жүйесінің бірінші мәселесі – моделге қандай сипаттамаларды енгізу керек және олардың салмағы оларға сәйкес келу керектігін анықтау қиын. Қорытынды бағалаудың сапасы және, сайып келгенде, тәуекелдерді бағалау

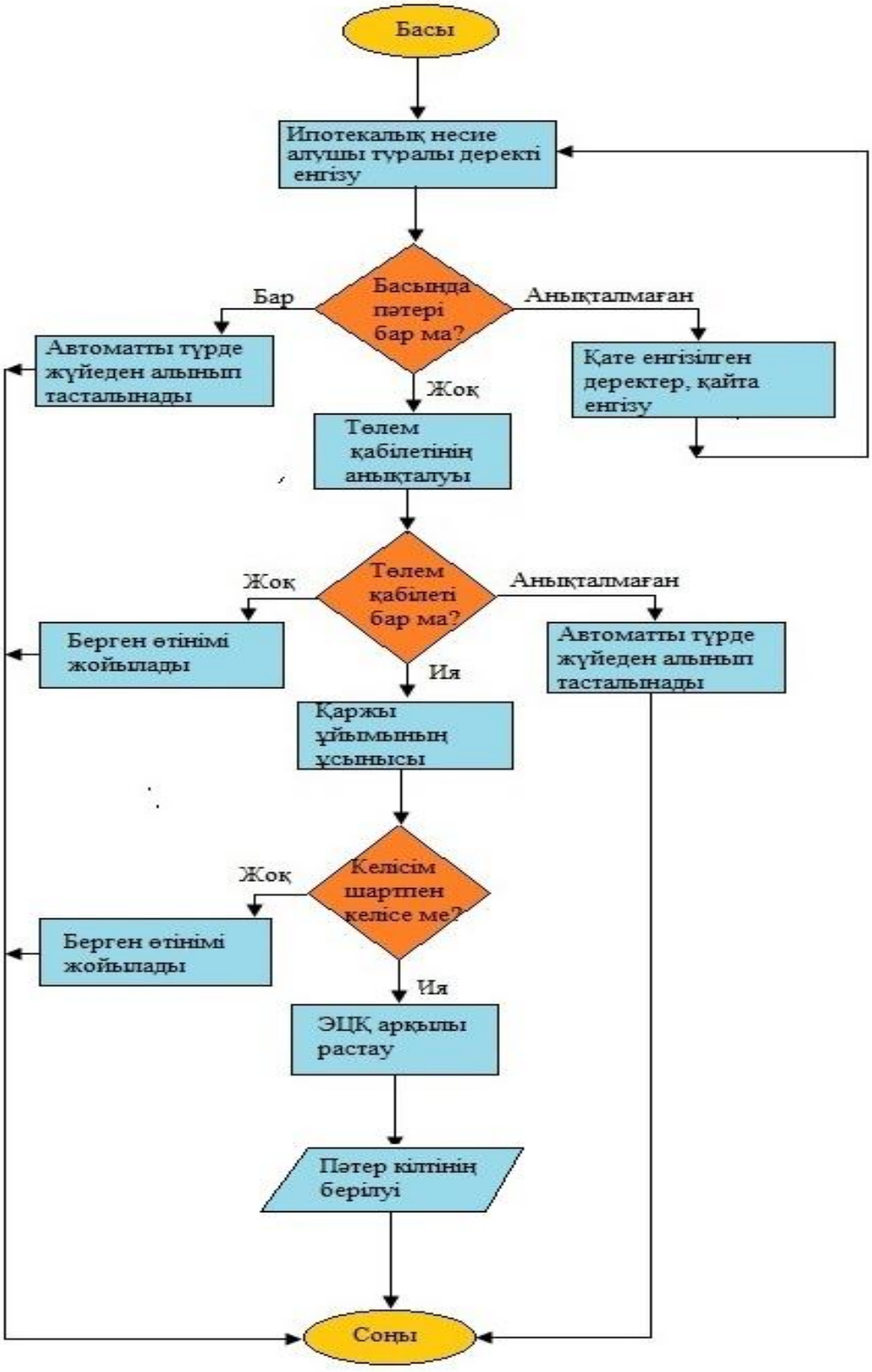
тиімділігі мен несие портфелінің кірістілігі көп жағдайда бастапқы деректерді таңдауға байланысты болады. Бұл мәселеге бірнеше көзқарастар бар, классикалық тәсіл, әрине, клиенттерді оқыту үлгісі, олар бұрыннан белгілі, олар жақсы қарызгер екендіктерін немесе жоқ екендіктерін дәлелдеді. Іріктеме мөлшері Батыс елдерінде мәселе емес, бірақ біздің елімізде шынымен тиімді жүйені құру үшін бізге берілген несиелер туралы тарихи мәліметтер қажет, ол үшін несиелер беру керек, ол үшін төлем қабілетті анықтау жүйесі қажет. Төлем қабілетті анықтау жүйесінің қалыпты жұмыс істеуі үшін қарыз алушының несиелік қабілеттілігі туралы қорытынды жасауға болатын белгілі бір үлгіні алу қажет. Бірақ бұл үлгі өз кезегінде ұпай жинау жүйесінен алынады. Сондықтан, біраз уақыт жұмсау керек және сынақтар мен қателіктер арқылы ақиқатқа бейімделген төлем қабілетті анықтау жүйесінің негізін құрайтын деректер жинала бастайды. Қазіргі уақытта несие бюроларында клиенттер туралы көптеген мәліметтер жинақталған, бұл ақпаратты банктің дерекқорында тиісті ақпарат болмаған кезде төлем қабілетті анықтау үшін және үлгілерді де алу үшін де қолдануға болады [41].

Теориялық тұрғыдан алғанда, төлем қабілетті бағалау жүйесі шынымен тиімді оны құру бірнеше жыл бойына белгілі нәтижелерімен (несиені өтеген немесе өтемеген) және жүйені «оқытумен» қарыз алушылардың айтарлықтай іріктелуін талап етеді, бірақ көптеген банктерде мұндай статистика жоқ. Жоғарыда айтылғандарды ескере отырып, «дайын» батыс жүйелерін қолдану тиімді емес[42,43].

Кесте 2.1 – Несие алушы азаматтардың деректер картасының үлгісі

Ұзақ мерзімге ипотекалық несие алушы туралы деректер	№	Мәні	Нәтижесі
Басында пәтері барма?	1	Бар	1
	2	Жоқ	0
	3	Анықталмаған	1
Несие алушының жасы	1	52 жастан жоғары	1
	2	46 жаспен 51жас арлығы	0
	3	36 жаспен 45 жас аралығы	0

Мемлекеттік тапсырыс аясында жүзеге асырылатын мемлекеттік тұрғын үй бағдарламалары шынымен мұқтаж адамдарды қамтуы тиіс болғандықтан, төлем қабілеті мен қатар басында үй болмағаны біріңғай дерекқордан тексеріледі, қалған қадамдар төменде блок схема түрінде 2.2-суретте көрсетілген.



Сурет 2.2 – Ипотекалық несие алушы тұлғаларға несие берілу алгоритмінің блок-схемасы

Балдық жүйе емес нақты мәндер яғни [0,1] аралығында мәндер жуықталып алынады. Бұл өз кезегінде төлем қабілеттерін тез әрі нақты анықтауға септігін тигізеді. Белгілі бір алгоритм бойынша «оқыту» керек, ал егер таңдау аз болса, қол жетімді мәліметтер негізінде осы алгоритмді «тарататын» немесе ұқсас моделдерді қолдануға болады. Бұл жүйені нақты деректердегідей тиімді оқыту үшін нейрондық желіні қолданамыз, бұл тәсіл дайын жүйелерге қарағанда әлдеқайда тиімді болатыны анық. Ипотекалық несие алушы азаматтарға сіз «жақсы», ал сіз «жаман» клиентсіз деп айту этикаға жатпайтын болғандықтан несие «оң» және «теріс» нәтиже деп алдық. Нәтижесі «1» болған клиенттер теріс нәтижелі деп есептеліп ұзақ мерзімге ипотекалық несие берілмейді, ал нәтижесі «0» болған клиенттер оң нәтижелі болып оларға ұзақ мерзімге ипотекалық несие беріледі.

Жеке тұлғалардың төлем қабілетін анықтайтын жүйені банктер тәжірибесінде енгізу банктердің өздері үшін де, қарыз алушының несиені қайтару сенімділігі үшін де қажет.

Ипотекалық несие алушы азаматтардың төлем қабілетін анықтайтын жүйесі қарыз алушының несиелік қабілетін талдау мен бағалауды ғана емес, сонымен бірге бөлшек несиелендіру нарығының қатысушылары үшін қазіргі уақытта маңызды болып келе жатқан бірқатар мәселелерді шешетіндігін атап өткен жөн. Атап айтқанда:

- ақпарат ағындарының көбеюі
- шешім қабылдау уақытын қысқарту қажеттілігі
- әр клиентке жеке көзқарас талаптары
- шешім қабылдауды автоматтандыру
- өзгермелі нарықтық жағдайларға тез бейімделу.

Data Mining әдістерін: сызықты регрессияны, логистикалық регрессияны, көпқабатты нейрондық желілерді қолдана отырып, салмақ есептеу жеке тұлғалардың төлем қабілетін анықтайтын жүйенің моделі мен алгоритмдерін құру. Ипотекалық несие беру жүйесінің моделін құру кезінде туындайтын мәселелердің бірі ол уақыт өте келе модельдердің өзгеруі. Ипотекалық несиелік алушылардың төлем қабілеттерін жүйе жаңа ипоткалық несиеге пәтер алушы үміткерлердің төлем қабілетін болжау үшін «оң» немесе «теріс» нәтижелер санатына жатқызылған бұрынғы өтінім берушілердің сипаттамаларын қолдануға болатындығын білдіреді. Кейде сипаттамалардың таралу үдерісі уақыт өте тез өзгеріп отырады, сондықтан ол өзектілігін сақтау үшін жеке тұлғалалардың төлем қабілеттерін анықтайтын жүйенің моделін үнемі жаңартып отыруды қажет етеді [44].

2.2 Сызықты регрессия әдісінің қолданылуы

Data Mining әдістерінің бірі регрессиялық талдау. Регрессиялық талдаудың негізгі мақсаты кейбір сипаттамалар арасындағы байланысты анықтау. Y айнымалы тәуелді айнымалы деп аталады, ал әсер етуші айнымалылар $x_1, x_2, ..., x_n$ факторлар (регрессорлар) деп аталады. Тәуелділік сипатын анықтау, регрессия моделін (теңдеуін) таңдау және олардың өлшемдерін бағалау регрессиялық талдаудың міндеттері болып табылады[45].

Регрессиялық талдауда мына түрдегі теңдеу зерттеледі:

$$Y = \varphi(X) + \varepsilon \qquad (2.2)$$

бұл жерде Y – нәтижені береді, (жауап беруші, кездейсоқ тәуелді айнымалы); X – фактор (кездейсоқ емес, тәуелсіз айнымалы); ε – кездейсоқ айнымалы, ол X – факторының регрессия сызығынан ауытқуын сипаттайды (қалдық айнымалы).

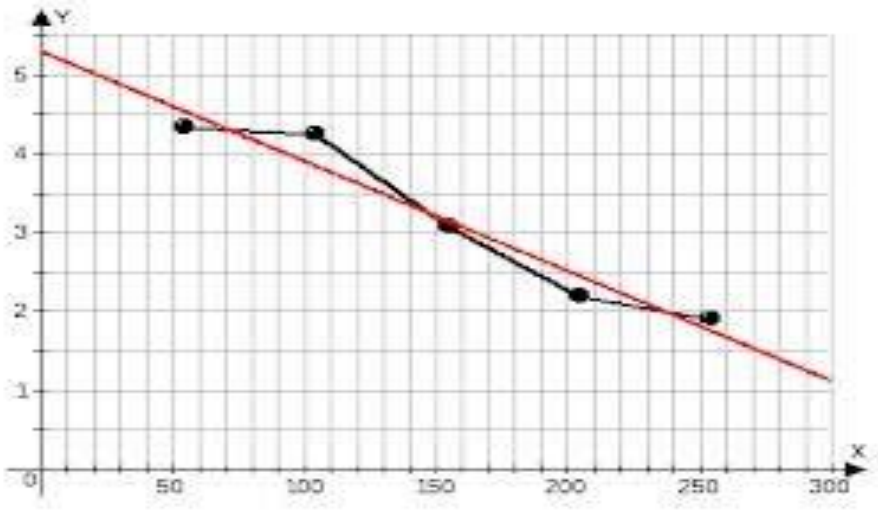
Регрессия теңдеуі мына түрде жазылады:

$$y_x = \varphi(x, b_0, b_1, ..., b_p) \qquad (2.3)$$

бұл жерде x – X шаманың мәні. $y_x = M_x(Y)$; $b_0, b_1, ..., b_p$ – φ регрессия функциясының өлшемдері. Сондықтан, регрессиялық талдау есебі функция мен оның өлшемдерін анықтаудан тұрады.

Таңдап алынған теңдеудің типіне қарай сызықты немесе сызықты емес деп бөлінеді. (соңғы жағдайда одан әрі нақтылауға болады: квадраттық, экспоненциалдық, логарифмдік және т.б.) Өзара байланысу белгілерінің санына қарай жұп және көпше регрессия деп бөлінеді. Егер, екі белгі арасындағы байланыс зерттелетін болса, (нәтиже және фактор), онда регрессия жұп деп аталады. Егер, үш белгі немесе оданда көп белгілер арасындағы байланыс зерттелетін болса, онда регрессия көпше (көпфакторлы) деп аталады.

Регрессиялық талдаудың алғашқы кезеңінде, өңделетін деректер графикалық түрде беріледі.



Сурет 2.3 – Регрессияның эмпирикалық сызығы

X және Y айнымалыларының арасындағы тәуелділік координаталық жазықтықтағы (x, y) нүктелермен белгіленген және оларды сынық сызық жалғап тұр. Бұл қисық график, регрессияның эмпирикалық сызығы деп аталады. Регрессияның эмпирикалық сызығының түріне қарай, Y айнымалының X айнымалыдан тәуелділігінің түрін болжауға болады. Бұл жағдайда, сызықты байланыс екені көрініп тұр.

Егер, φ функциясының түрі регрессия теңдеуінде таңдалса, онда белгісіз өлшемдерді $b_0, b_1, ..., b_p$ бағалау үшін ең кіші квадраттар әдісі (ЕКӘ). Әдіске сай, функцияның белгісіз өлшемдері тәжірибелік (эмпирикалық) мәндердің y_i – дің олардың есептелген (теориялық) мәндерінен ауытқу квадраттарының қосындысы минималды болатындай етіп таңдалады.

$$S = \sum_{i=1}^{n} \left(y_{i\,эксп} - y_i^p \right)^2 = \sum_{i=1}^{n} \left(y_{i\,эксп} - \varphi\left(x_i, b_0, b_1, ...b_p\right) \right)^2 \to \min \qquad (2.5)$$

y_i^p – регрессиялық теңдеумен есептелген мән; $y_i - y_i^p = \varepsilon$ – ауытқу (қателік, қалдық); n – кіріс деректерінің саны.

Жұпталған сызықты регрессия моделі. Жұпталған сызықты регрессия моделі екі айнымалының арасындағы регрессия функциясы $\varphi(x)$ сызықты болатын байланысты қарастырамыз. Y – тің шартты орташа мәнін y_x деп белгілейміз, жалпы жиынтығының X айнымалысының белгіленген x мәні.

Регрессия теңдеуі мына түрге келеді:

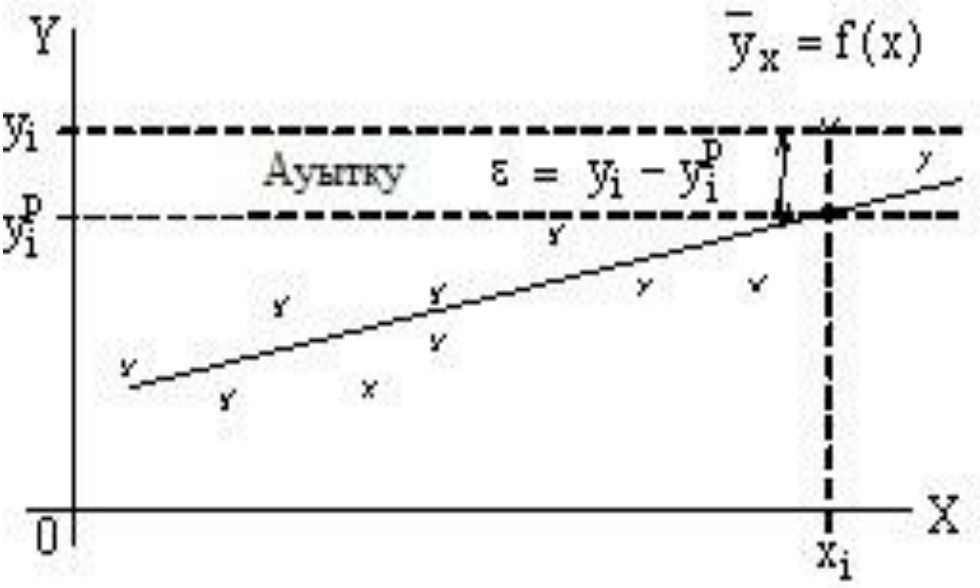
$$y_x = ax + b \qquad (2.6)$$

бұл жерде a – регрессия коэффициенті (сызықты регрессия сызығының иілу көрсеткіші). Регрессия коэффициенті X айнымалысының бір бірлікке өзгергендегі Y айнымалысының неше орташа бірлікке өзгеретінін көрсетеді. Ең кіші квадраттар әдісінің көмегімен сызықты регрессияның өлшемдерін есептейтін формулалар алынады. Формулалар 2.2 – кестеде берілген.

Кесте 2.2 – Сызықты регрессия өлшемдерін есептеуге арналған формулалар

Бос мүше b	Регрессии коэффициенті a	Детерминация Коэффициенті
$b = \frac{\overline{y} \cdot \overline{x^2} - \overline{x} \cdot \overline{xy}}{\overline{x^2} - (\overline{x})^2}$	$a = \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - (\overline{x})^2}$	$R^2 = \frac{\Sigma(y_i^p - \overline{y})^2}{\Sigma(y_i - \overline{y})^2}$
Регрессия теңдеуінің маңыздылығы туралы гипотезаны тексеру		
H_0: $R^2 = 0$	H_1: $R^2 > 0$	$F = \frac{R^2}{1 - R^2} \cdot \frac{n - p - 1}{p}$
$F(\alpha; k_1; k_2)$, $k_1 = p, k_2 = n - p - 1$, (сызықты регрессия үшін $p = 1$)		

Айнымалылар арасындағы байланыс бағыты регрессия коэффициентінің белгісі негізінде анықталады. Егер, регрессия коэффициенті оң болса тәуелді айнымалы мен тәуелсіз айнымалы байланысы оң болады. Егер, регрессия коэффициенті теріс болса, тәуелді айнымалы мен тәуелсіз айнымалы байланысы теріс болады (кері).



Сурет 2.4 – Ауытқу тұжырымдамасы (сызықты регрессия)

Сурет 2.5 – Детерминация коэффициентінің графикалық Интерпретациясы (сызықты регрессия)

Регрессия теңдеуінің жалпы сапасына талдау жасау үшін детерминация коэффициентін R^2 қолданады, оны көпше корреляция коэффициентінің квадраты

деп атауға да болады. Детерминация коэффициенті әрдайым [0;1] аралығында болады. Егер R^2 мәні бірге жақын болса, бұл құрылған модел сәйкес айнымалылардың толық өзгерулерін түсіндіретінін білдіреді немесе R^2 мәні нөлге жақын болса, құрылған моделдің сапасының сапасының төмендігін білдіреді[46].

Детерминация коэффициенті R^2 табылған регрессия функциясының X пен Y бастапқы мәндерінің арасындағы байланысты неше пайыз $\left(R^2\right)\cdot 100\%$ екенін көрсетеді. 2.4 – суретте $\left(y_i^p - \bar{y}\right)$ – регрессиялық моделмен түсіндірілетін вариация және $\left(y_i - \bar{y}\right)$ – жалпы вариация. Әлбетте, $\left(1 - R^2\right)\cdot 100\%$ мәні Y параметрінің өзгеруінің қанша пайызы регрессия моделіне кірмеген факторларға байланысты екенін көрсетеді.

Детерминация коэффициентінің жоғарғы мәнінде $\left(R^2 \geq 75\%\right)$, нақты x^* мәніне бастапқы деректер ауқымында $y^* = f\left(x^*\right)$ болжам жасауға болады. Бастапқы деректер ауқымына кірмейтін мәндерге болжам жасау кезінде, алынған моделдің заңды екеніне кепілдік беруге болмайды. Бұл, модель ескермеген жаңа факторлардың болуымен түсіндіріледі.

Регрессия теңдеуінің маңыздылығын бағалау Фишер критериі арқылы жүзеге асады (2.2 – кестеде көрсетілген.) Нөлдік гипотеза дұрыс болған жағдайда, критериде $k_1 = p, k_2 = n - p - 1$ еркіндік дәрежесі бар Фишер үлестірімі болады (жұпталған сызықтық регрессия үшін $p = 1$). Егер нөлдік гипотеза кері қайтарылса, регрессия теңдеуі статистикалық маңызды болып саналады. Егер нөлдік гипотеза кері қайтарылмаса, регрессия теңдеуі статистикалық маңызды емес әрі сенімсіз болып саналады.

Жүйені моделдеу үшін мысал ретінде мынадай есеп шығарылды. Несие беруші қаржы ұйымында пәтерлер құнын есептеп өтінім берушілердің төлем қабілетіне талдау жүргізді. Белгілі болғанындай, жеке тұлғаларға несиені өтеуде қайтару мерзімі мен айына қайтаратын қаржы шығындары есепке алынды. Маңызды фактор ретінде несие мерзімі мен оны өтеп бітіруге әсер ететін айлық кірістері таңдалып алынды. Қарызды өтеп бітетіндерін болжау үшін деректерге регрессиялық талдау жүргіземіз:

Кесте 2.3 – Несие туралы деректер картасы

Несиені қайтару жылдары	*3,5*	*2,4*	*4,9*	*4,2*	*3,0*	*1,3*	*1,0*	*3,0*	*1,5*	*4,1*
Айына қайтаратын төлемі (млн.теңге)	*16*	*13*	*19*	*18*	*12*	*11*	*8*	*14*	*9*	*16*

Регрессиялық талдау жүргізу үшін[47]:

1. Бастапқы деректерді графигін сызу керек, тәуелділік сипатын шамамен анықтау керек;

2. Регрессия функциясының түрін таңдау керек және моделдің сандық коэффициенттерін ең кіші квадраттар әдісімен анықтау қажет;

3. Детерминация коэффициентінің көмегімен регрессиялық тәуелділіктің шамасын бағалау қажет;

4. Регрессия теңдеуінің маңыздылығын бағалау;

5. Қабылданған модель бойынша 2 жылға болжам жасау (немесе болжам жасау мүмкін еместігі туралы қорытынды жасау);

Әрі қарай бастапқы деректер бойынша талдауды жалғастыра береміз:

1. Сызылған нүктелер сызықтың бойында емес, төлем мерзімі мен айлық кірісінен басқа отбасындағы адамдардың саны, өмірде кездесетін түрлі жағдайлар әсерін беретіні анық. Бірақ, нүктелер сызықтың жанына жақын орналасқандықтан өлшемдердің өзара оң байланысқандығы байқалады.

2. Сызықты регрессия коэффициенттерін есептейміз және детерминация коэффициентін R^2 есептейміз:

Кесте 2.4 – Регрессия коэффициенттерінің есептелуі

№	x_i	y_i	x_i^2	$x_i y_i$	y_i^p	$\left(y_i^p - \overline{y}\right)$	$\left(y_i - \overline{y}\right)^2$
1	3,5	16	12,25	56,00	15,22	2,63	5,76
2	2,4	13	5,76	31,20	12,30	1,70	0,36
3	4,9	19	24,01	93,10	18,95	28,59	29,16
4	4,2	18	17,64	75,60	17,09	12,15	19,36
5	3,0	12	9,00	36,00	13,89	0,08	2,56
6	1,3	11	1,69	14,30	9,37	17,88	6,76
7	1,0	8	1,00	8,00	8,57	25,27	31,36
8	3,0	14	9,00	42,00	13,89	0,09	0,16
9	1,5	9	2,25	13,50	9,90	13,67	21,16
10	4,1	16	16,81	65,60	16,82	10,36	5,76
Σ	28,9	136	99,41	435,30	–	112,42	122,40

$$\bar{x} = \frac{\sum n_i x_i}{n} = 2{,}89;\ \bar{y} = \frac{\sum n_i y_i}{n} = 13{,}6;$$

$$b = \frac{13{,}6 \cdot 9{,}941 - 2{,}89 \cdot 43{,}53}{9{,}941 - 2{,}89^2} = 5{,}91;\ a = \frac{43{,}53 - 2{,}89 \cdot 13{,}6}{9{,}941 - 2{,}89^2} = 2{,}66;$$

Регрессиялық тәуелділік мына түрге келеді: $y^p = 2{,}66x + 5{,}91$. Айнымалылар арасындағы байланысу бағытын анықтаймыз: регрессия коэффициентінің белгісі оң, сондықтан байланыста оң болады.

3. Детерминация коэффициентін есептейміз: $R^2 = \frac{112{,}42}{122{,}40} = 0{,}92$ немесе 92% Осылайша, сызықты модель, 92% пайыз төлемнің вариациясын түсіндіреді, бұл дегеніміз факторларды таңдаудың дұрыстығын білдіреді. 8% пайыз вариация түсіндірілмейді, себебі ол басқа факторлардың әсері болып саналады және ол сызықты регрессия моделіне кірмеген.

4. Регрессия теңдеуінің маңыздылығын тексереміз:

$$F_{бақылау} = \frac{0{,}92^2}{1 - 0{,}92^2} \cdot \frac{10 - 1 - 1}{1} = 44{,}1$$

$F_{бақылау} = 44{,}1 \succ F_{кисык}(0{,}05; 1; 10 - 1 - 1) = 5{,}32$ болғандықтан, регрессия теңдеуі статистикалық маңызды болып саналады.

5. Болжам жасау есебін шешеміз. Детерминация коэффициенті R^2 жеткілікті жоғарғы мәнге ие болғандықтан 2 жылға жасалатын болжам бастапқа деректердің диапазонына кіреді, ендеше болжам жасауға болады:

$y^*(x = 2жыл) = 2{,}66 \cdot 2 + 5{,}913 = 11{,}2$ млн.

2.3. Логистикалық регрессия әдісінің қолданылуы

Бұл бөлімде сызықты регрессия функциясының кері логит түрлендірудегі теориялық аспектілерді қарастыратын боламыз. Қарапайым тілмен айтқанда, логистикалық жауап функциясы деп айтуымызға болады. Одан кейін максималды шынайылық әдісінің арсеналын қолдана отырып, логистикалық регрессия моделімен сәйкесінше Logistic Loss шығын функциясын табамыз, түсінікті болуы үшін, логистикалық регрессия моделінде салмақтардың векторының параметрлері таңдалатын функцияны анықтаймыз.

Логистикалық регрессия - сызықтық жіктеуге жататын модельдердің бірі. Қарапайым сөзбен айтқанда, сызықтық классификатордың міндеті X айнымалыларынан (регрессорлар) y-тің мақсатты мәндерін болжау. Бұл жағдайда X сипаттамалары мен y үшін мақсатты мәндер арасындағы байланыс сызықты деп саналады. Осыдан классификатордың нақты атауы - сызықтық. Шын мәнінде жалпылау үшін логистикалық регрессия моделі X белгілері мен мақсатты y мәндері арасында сызықтық байланыс бар деген болжамға негізделген[48].

2.3.1 Регрессия теңдеуін түрлендірудің қажеттілігі.

Жоғарыдағы бөлімдерде айтып кеткеніміздей, ұзақ мерзімге ипотекалық несиені алған потенциалды клиент үшін оны уақытылы қайтара алуын бірнеше факторларға сүйене отырып анықтау қаржы ұйымы үшін бастапқы міндеттердің бірі. Бұл мәселені шешу үшін екі факторды, яғни айлық табысы мен ипотекалық несиені ай сайынғы қайтару мөлшерін ескеріп болжамдарды жасап көруге болады. Есеп өте шартты, бірақ бұл мысалда біз оны шешу үшін сызықтық регрессия функциясын неге қолдану жеткіліксіз екенін түсініп, функциямен қандай түрлендірулер қажет екенін білетін боламыз [49].

Жоғарыда келтірілген тәуелділіктердің негіздеріне сүйене отырып, неғұрлым жалақысы жоғары болса, соғұрлым оның ипотекалық несиені қайтару мүмкіндігі де жоғары екенін білеміз. Сонымен қатар, жалақының белгілі бір ауқымы үшін бұл тәуелділік айтарлықтай сызықтық болады. Мысалы, 60 000 теңге мен 200 000 мың теңге жалақы алатын азаматтардың диапазонын алсақ, айлық төлемнің жалақыға сызықты тәуелді екенін көреміз. Егер жалақының көрсетілген диапазоны үшін жалақының төлемдер арасындағы арақатынасы 3-тен төмен түсе алмайтын болса, қарыз алушының қалтасында 5000 мың теңге болуы керек деп есептесек, сызықты регрессия теңдеуі мына түрге келеді:

$$f(w, x_i) = w_0 + w_1 x_{i1} + w_2 x_{i2} \qquad (2.7)$$

бұл жерде, $w_0 = -5000$, $w_1 = 1$, $w_2 = -3$, x_{i1} i-ші ипотекалық несие алушының жалақысы, x_{i2} i-ші ипотекалық несие алушының ай сайынғы төлемі. Жалақыны және несие төлемін белгіленген параметрлермен $\vec{w}$ теңгерімге ауыстыра отырып, сіз несие беру немесе бас тарту туралы шешім қабылдауға болады. Болашаққа көз жүгіртсек, берілген $\vec{w}$ параметрлері үшін логистикалық белсендіру функциясында қолданылатын сызықтық регрессия функциясы несиенің қайтарылу ықтималдығын анықтау үшін есептеулер жүргізуді қиындататын үлкен мәндерді шығаратынын ескереміз. Сондықтан біздің коэффициенттерімізді мысалы, 25000 есе төмендету ұсынылады. Осы өзгерістен бастап несие беру туралы шешім өзгермейді. Түсінікті болуы үшін біз үш ипотекалық несие алушының жағдайын қарастырамыз. Есептеулердің толық нәтижелері 3-ші тарауда берілген.

Кесте 2.5 – Потенциалды ипотекалық несие алушылар кестесі

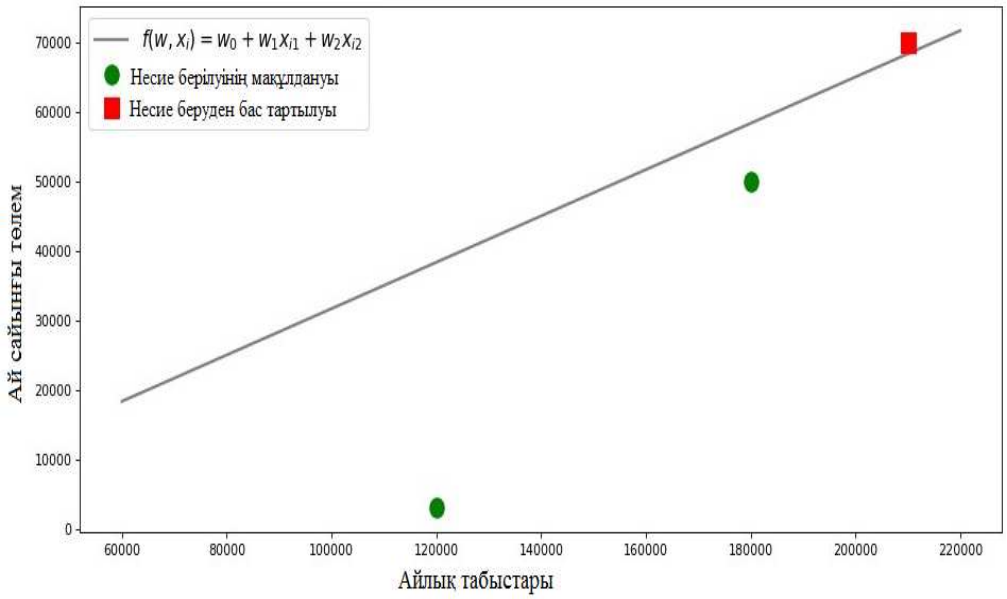
№	Ипотекалық несие алушы	Айлық табысы, мың теңге	Ай сайынғы төлемі, мың теңге	$f(w, x)$	Шешім
1	Асанов Ерлан	120 000	3 000	4.24	Мақұлданған
2	Раисов Казбек	180 000	50 000	1.00	Мақұлданған
3	Сайтбаев Мұрат	210 000	70 000	-0.20	Бас тартылған

Кестедегі деректерде көріп отырғанымыздай, Асанов Ерланның, 120 000 теңге айлығымен қайтарымы 3000 болатын несие алғысы келеді.

Анықталғанындай, Асанов Ерланның қалауы мақұлдануы үшін ай сайын қайтаратын ақшасынан 3 есе асуы қажет және өзінің қалтасында 5 000 теңгедей қалуы керек. Бұл шартты Асанов Ерлан қанағаттандырады:

120000-3*3000-5000= 106000.

Тіпті 106000 теңге қалады екен. $f(w, x_i)$-ді есептеу кезінде $w \to 25000$ есе азайтқанымызға қарамастан, сол кредит мақұлданып кетуі мүмкін. Раисов Казбек те ипотекалық несие алады. Ал, Сайтбаев Мұраттың жалақысы ең жоғары болса да оған несие берілуден бас тартылады. Осы жағдайды график түрінде көрсетсек болады.



Сурет 2.6 – Несие алушылардың классификациясы

Сонымен, $f(w, x_i) = w_0 + w_1 x_{i1} + w_2 x_{i2}$ функциясы негізінде сызылған түзуіміз, потенциалды клиенттерімізді оң нәтижелі және теріс нәтижелі деп екіге бөледі. Қалауы мүмкіндігіне сәйкес келмеген потенциалды несие алушылар түзу сызықтың үстінде қалады, ал қалауы мүмкіндігіне сай келген несие алушылар түзі сызықтың астында қалады. Демек, түзу сызық потенциалды клиенттерді екіге бөледі. Бірінші класқа ипотекалық несиені қайтара алады деп +1, яғни оң нәтижелі клиенттерді жатқызсақ, екінші класқа несиені қайтара алмайды деп теріс, -1 немесе 0 нәтижелі клиенттерді жатқызамыз.

Осы келтірілген мысалға қарап қорытынды жасаймыз. $M(x_1, x_2)$ нүктесін алып, нүктенің координаталарын $f(w, x_i) = w_0 + w_1 x_{i1} + w_2 x_{i2}$ түзулерінің сәйкес теңдеуіне қойып, үш нұсқаны қарастырамыз:

1. Егер нүкте түзу сызықтың астында болса, оны біз оң, яғни «+1» класқа жатқызамыз. $f(w, x_i) = w_0 + w_1 x_{i1} + w_2 x_{i2}$ функциясының мәні оң болады, яғни 0-ден $+\infty$ дейін. Демек, біз ипотекалық несиені қайтару ықтималдығы (0.5,1] аралығында деп тұжырымдасақ болады. Функцияның мәні неғұрлым жоғары

болса, соғұрлым оның ықтималдығы жоғары болады.

2. Егер нүкте түзу сызықтың астында болса, оны біз теріс, яғни «-1» немесе «0» класқа жатқызамыз. Функцияның мәні 0-ден $-\infty$ болады, яғни теріс болады. Демек, біз ипотекалық несиені қайтару ықтималдығы [0,0.5) аралығында болады және модул бойынша функция мәні неғұрлым жоғары болса, соғұрлым сенімділігіміз жоғары болады.

3. Егер нүкте түзу сызықтың бойында немесе екі кластың айналасында болса, $f(w,x_i)=w_0+w_1x_{i1}+w_2x_{i2}$ функциясының мәні 0-ге тең және несиені төлей алу ықтималдығы 0.5-ке тең.

Диссертациялық жұмыстың зерттеу тақырыбына сай, бізде екі ғана фактор емес миллиондаған адамдардың төлем қабілетін анықтау мәселесі тұр деп есептейміз. Ендеше түзу сызықтың орнына біз m өлшемді жазықтық болады және w коэффициенттері кездейсоқ емес барлық ережеге сай несиені қайтарған немес қайтармаған туралы деректерге сүйеніп алынады. Байқағандарыңыздай, несие алушылар w коэффициенттері анықталған кезде таңдалынып жатыр. Іс жүзінде логистикалық регрессия моделінің міндеті Logistic Loss шығын функциясының мәні минимумге ұмтылған кезде w параметрін анықтаудан тұрады. Дегенмен, $w\to$ векторы қалай есептелетініне кейінірек тоқталатын боламыз[50].

$f(w,x_i)=w_0+w_1x_{i1}+w_2x_{i2}$ функциясының көмегімен, біз кімге несие беру керек ал кімге бермеу керек екенін білдік. Ипотекалық несиені, беру немесе бермеу туралы шешіммен қаржы ұйымының басшылығына тікелей бара алмаймыз, себебі біз қайтара алу мүмкіндігін болжауымыз қажет болатын. Әрі қарай не істеуіміз қажет екендігі туралы ойлана келе, мәні $(-\infty,+\infty)$ арасында жататын $f(w,x_i)=w_0+w_1x_{i1}+w_2x_{i2}$ функциясын мәні [0,1] аралығында жататын функцияға түрлендіру қажеттілігі туындайды. Ондай функция бар, логистикалық жауап немесе кері логит түрлендіру деп атайды:

$$\delta=\left(\vec{w}^{T}\ \vec{x}_i\right)=\frac{1}{1+e^{-w^T\vec{x}_i}} \qquad (2.8)$$

Әрі қарай, қадамдар бойынша логистикалық жауап функциясы қалай алынатынын көреміз. Кері бағытты қарастырамыз, яғни 0 мен 1 арасындағы ықималдықтың мәні белгілі деп есептеп, $-\infty$ -тен $+\infty$ -ке дейінгі сандардың барлық ауқымы үшін бұл мәнді айналдырып есептей береміз.

Логистикалық белсендіру функциясын алу.

Ықтималдықтың мәндерін $[0,+\infty)$ диапазонына ауыстырамыз. $f(w,x_i)=w_0+w_1x_{i1}+w_2x_{i2}$ функциясын логистикалық жауап функциясына $\delta=\left(\vec{w}^{T}\ \vec{x}_i\right)=\frac{1}{1+e^{-w^T\vec{x}_i}}$ трансформациялау кезінде, несиелік аналитиканы қоя тұрып, бухмекерлік кеңсенің ұтыс ойынына тігілген бәс секілді 4 мүмкіндіктің 1-еуі келуі мүмкін деген ықтималдықты қарастырамыз. Бәс тіккен ойыншылардың ұтып алу мүмкіндігі сәттіліктің сәтсіздікке қатынасын көрсетеді. Ықтималдықтар жағынан

қарасақ, мүмкіндіктер болатын оқиғаның болмауына қатынасын білдіреді. Оқиға $(odds_+)$ болу мүмкіндігінің формуласын мына түрде жазамыз[51]:

$$odds_+ = p_+ 1 - p_+ \tag{2.9}$$

p_+ оқиғаның орындалу ықтималдығы, $(1-p_+)$ -оқиғаның орындалмау ықтималдығы. Мысалы, жас жігіттің жоғарғы қабатқа егде жастағы қариямен салыстырғанда баспалдақпен жылдам көтерілу ықтималдығы 0.8-ге тең. Оның мүмкіндігі $\frac{1}{4}\left(\frac{0.8}{(1-0.8)}\right)$, және керісінше мүмкіндіктерді біле тұра p_+ ықтималдығын есептеу қиындық тудырмайды:

$$\frac{p_+}{1-p_+} = 4 \Rightarrow p_+ = 4(1-p_+) \Rightarrow 5p_+ = 4 \Rightarrow p_+ = 0.8$$

Осылайша, 0-ден $+\infty$ арасындағы мәндерді қыбылдайтын ықтималдықты мүмкіндікке айналдыруды үйренеміз. Осы қарқынмен, әрі қарай қадам жасай отыра ықтималдықты $-\infty$-тен $+\infty$ аралығындағы сандық түзуге айналдыруды да толық үйренеміз.

Ықтималдық мәндерін $-\infty$ пен $+\infty$ диапазонына айналдырайық. Бұл қадам өте қарапайым - мүмкіндікті Эйлер ℓ санына негіздеп, мыналарды аламыз:

$$f(w, x_i) = \vec{w}^T \vec{x} = \ln(odds_+) \tag{2.10}$$

Енді біз, $p_+ = 0.8$ болатын болса, $f(w, x_i)$-дің мәнін есептеу ыңғайлы болады, сонымен қатар ол оң болуы керек:

$$f(w, x_i) = \ln(odds_+) = \ln\left(\frac{0.8}{0.2}\right) = \ln(4) \approx +1.38629.$$

Оң болғанына көз жеткізілді. Сенімді болу үшін, егер $p_+ = 0.2$ деп есептесек $f(w, x_i)$ функциясының теріс мәні шығады деп күтеміз. Тексеріп көреміз: $f(w, x_i) = \ln\left(\frac{0.2}{0.8}\right) = \ln(0.25) \approx -1.38629$. Бәрі дұрыс, күтілген мән шықты.

Енді біз барлық сандық жолда ықтималдықтың мәнін 0-ден 1-ге $-\infty$-ден $+\infty$-ге қалай ауыстыру керектігін білеміз. Келесі қадамда біз керісінше жасаймыз. Қазірше, логарифм ережелеріне сәйкес, $f(w, x_i)$ функциясының мәнін біле отырып, мүмкіндіктерді есептеуге болады:

$$odds_+ = e^{f(w, x_i)} = e^{\vec{w}T\vec{x}} \tag{2.11}$$

Бұл мүмкіндікті анықтайтын әдіс алдағы жұмыстарымызда да қажет болады.

Формуланы p_+-ді анықтау үшін шығару. Сонымен, p_+-ді анықтап $f(w, x_i)$ функциясының мәнін табуды үйрендік десекте болады. Шын мәнінде біз керісінше $f(w, x_i)$ функциясының мәнін біле тұра p_+-ді табуды толық блуіміз қажет. Ол үшін, мүмкіндіктің кері функциясы деген ұғымға сүйене отырып анықтаймыз:

$$p_+ = \frac{odds_+}{1 + odds_+} \tag{2.12}$$

Диссертациялық жұмыста жалпылама формуланы есептемейміз, дегенмен жоғарыдағы сандармен салыстырып тексереміз. Білетініміз, мүмкіндік $\frac{1}{4}(odds_+ = 4)$ болғанда оқиғаның болу ықтималдығы 0.8 $(p_+ = 0.8)$-ге тең. Орындарына қойып есептеп көрелік, $p_+ = \frac{4}{1+4} = 0.8$. Бұл алдыңғы жүргізген есептеу нәтижелерімен сәйкес келеді.

Алдыңғы қадамдарда, $odds_+ = e^{\vec{w}^T \vec{x}}$ екенін шығарылған болатын, демек мүмкіндіктің кері функциясын алмастыруға болады:

$$p_+ = \frac{e^{\vec{w}^T \vec{x}}}{1 + e^{\vec{w}^T \vec{x}}} \tag{2.13}$$

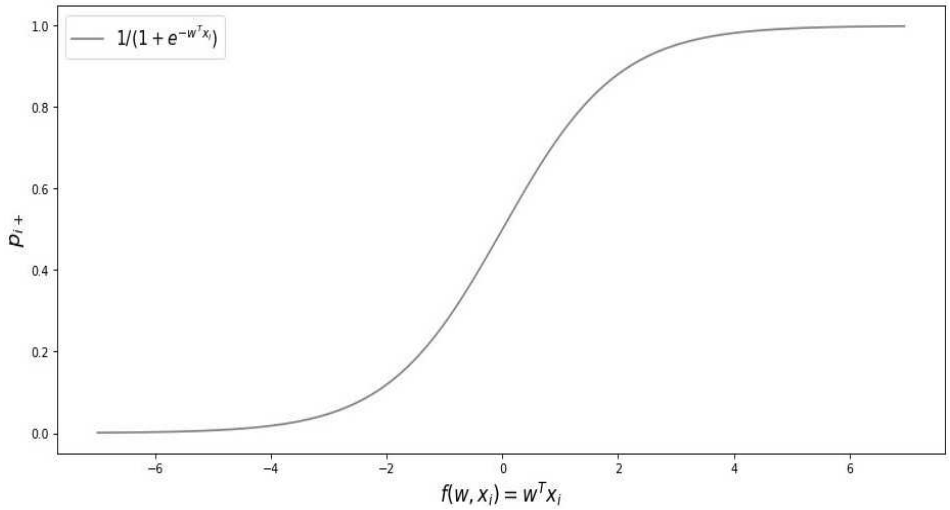
Алымы мен бөлімін $e^{\vec{w}^T \vec{x}}$ бөліп жіберіп мына түрге келеміз:

$$p_+ = \frac{1}{1 + e^{-\vec{w}^T \vec{x}}} = \delta\left(\vec{w}^T \vec{x}\right) \tag{2.14}$$

Ешқандай қателік кетпегеніне көз жеткізу үшін тағы бір кішігірім тексеру жүргіземіз. Алдыңғы қадамдардың бірінде $p_+ = 0.8$ үшін $f(w, x_i) \approx +1.38629$ екені анықталған болатын. Ендеше $f(w, x_i)$ мәнін логистикалық жауап функциясының орнына қойып $p_+ = 0.8$ болатынын күтеміз. Орындарына қойылғаннан кейін мынадай нәтиже алынады:

$$p_+ = \frac{1}{1 + e^{-1.38629}} = 0.8$$

Логистикалық белсендіру функциясын тестілеп, дәл нәтиже алынды. Функцияның графигін көрелік.



Сурет 2.7 – Логистикалық белсендіру функциясы

Кейбір әдебиеттерде бұл функцияны сигмоид-функция деген атаумен кездеседі. Графиктен классқа жататын нысанның ықтималдылығының негізгі

өзгерісі $f(w, x_i)$ шамасында -4-тен +4-ке дейінгі жерде болатындығын анық көруге болады.

Ипотекалық несие берудегі талдау жұмыстарына қайта оралсақ, 2.5-кестедегі деректерге сай есептеулерді жалғастырамыз. Сонымен, ипотекалық несиені қайтару ықтималдығын анықтадық. 1-ші номерлі несие алушы 120 мың теңге жалақысымен ай сайын қаржы ұйымына 3000 мың теңге қайтара алу ықтималдығы 100%. Қайтару ықтималдығы 0.3 болса, қаржы ұйымының саясатына байланысты 3-ші номерлі клиентке де ипотекалық несие беруге болады. Бұл жағдайда қаржы ұйымы тәуекелге баратыны анық. Жалақының айлық төлемге қатынасы 3-тен төмен болмауы және ипотекалық несиені қайтарушының төлемнен басқа өзінде 5000 теңге қалуы деген талаптар қаржы ұйымының өздерінің ойлап тапқандары деп айтуға болады[52]. Сондықтан, салмақ векторын $\vec{w} = (-5000, -1, -3)$ алғашқы түрде қолданбау керек. Талап етілгендей, коэффициенттерді едәуір төмендету керек және бұл жағдайда біз әр коэффициент 25000 мыңға бөлініп, яки іс жүзінде нәтиже түзелді. Бірақ бұл бастапқы кезеңде материалды түсінуді жеңілдету үшін жасалды. Өмірде коэффициенттерді ойлап тауып, түзетудің қажеті жоқ, шынайы деректерді дерекқорлардан алып қолдануға болады.

2.4 Логистикалық белсендіру функциясының салмақ векторын анықтауда ең кіші квадраттар әдісінің қолданылуы.

Қолданыста, $\vec{w}$ салмақ векторын таңдау әдісі бар екені белгілі, ол ең кіші квадраттар әдісі деп аталады. Бұл әдісті бинарлы классификация есептерінде қолдануға болады. Шынымменде, ең кіші квадраттар әдісін қолдануға еш кедергі жоқ. Бірақ, бұл әдіс классификация есебінде дәл нәтижені азырақ береді. Бұған теориялық негіздеме бар. Бір кішігірім мысалды қарастырайық. Біздің моделдеріміз (MSE және Logistic Loss қолдана отырып) салмақ векторын таңдауға кірісіп кетті делік және есептеуді бір сатыда тоқтатты делік. Ортасында, соңында немесе басында ма, маңызды емес, бастысы - бізде салмақ векторының кейбір мәндері бар және осы қадамда екі модель үшін де салмақ векторларының айырмашылығы жоқ деп айтуға болады. Содан кейін алынған салмақтарды алып, оларды логистикалық белсендіру функциясына $\left(\frac{1}{1+e^{-w^T \vec{x}_i}}\right)$, +1 класына жататын кейбір ныасандарға ауыстырамыз. Таңдалған салмақ векторына сәйкес екі жағдайды зерттей келе модель қателеседі және керісінше нысан +1 класқа жатады деп топшылайды. Logistic Loss және ең кіші квадраттар әдісі қолданылады. Алынған салмақтарды логистикалық белсендіру $\left(\frac{1}{1+e^{-w^T \vec{x}_i}}\right)$ функциясының орнына кез келген +1 класқа жататын нысан үшін қойылады [67].

Өрескел қателік жағдайы - модель нысанды +1 класқа 0.001 ықтималдықпен жатқызады.

Ең кіші квадраттар әдісі қолданылғандағы қателік:

$$MSE(y - p_{+}) = (1 - 0.01)^2 = 0.9801$$

Logistic Loss қолданылғандағы қателік:

$$LogLoss = \log_e\left(1 + e^{-yf(w,x)}\right) = \log_e\left(1 + e^{-1(-4.595...)}\right) \approx 4.605$$

Өте сенімділік жағдайы - модель нысанды +1 класқа 0.99 ықтималдықпен жатқызады.

Ең кіші квадраттар әдісі қолданылғандағы қателік:

$$MSE = (1 - 0.99)^2 = 0.0001$$

Logistic Loss қолданылғандағы қателік:

$$LogLoss = \log_e\left(1 + e^{-4.595...}\right) \approx 0.01$$

Бұл мысалда өрескел қателікте жоғалту функциясы LogLoss ең кіші квадраттар әдісінен қарағанда моделге тиімсіз екенін көруге болады.

2.4.1 Максималды ықтималдылық және логистикалық регрессия

Қаржы ұйымынан ұзақ мерзімге ипотекалық несие алу үшін жоғарыда келтірілген мысалға қайта ораламыз. Біз қазірге кезде көптеген мысалдар мен және көлемі үлкен деректермен жұмыс істеуіміз мүмкін. Бұл мысалдарда адамға тез түсінікті болу үшін цифрлер аз алынған еді. Қаржы ұйымының басшылығы ешнәрсеге қарамастан барлық өтінім иелеріне ипотекалық несие беремін деп шешім қабылдады делік, және бір сәтке көз алдымызға өтінім иелері туралы деректер жойылып кетті деп елестетейік. Ипотекалық несиені қайтару ықтималдығы әрбір үшінші адам несиені қайтармайды деп болжауымызға тура келеді. Басқаша айтқанда түйсігімізбен пайымдаған болжам $p = \frac{2}{3}$. Бұл түйсіктік болжамымызға теориялық негіз бар. Бұл максималды шынайылық әдісі, көптеген кітаптарда бұл әдісті максималды шынайылық принципі депте айтады[53].

Іріктеменің ықтималдығы – нәтижелерге және бақылауларға негізделген ықтималдық. Шынайылық функциясы шынайылықты іріктемелер тарату параметрлерінің мәндерімен байланыстырады. Біздің жағдайда жаттығулар жиынтығы жалпыланған Бернулли схемасы болып табылады, онда кездейсоқ шамасы тек екі мәнді алады: 1 немесе 0. Демек, үлгінің ықтималдығын p параметрінің ықтималдық функциясы ретінде келесі түрде жазуға болады:

$$P\left(\vec{y}|p\right) = \prod_{i=1}^{3} p^{y_i}(1-p)^{(1-y_i)} = p^1(1-p)^{1-1} * p^1(1-p)^{1-1} * p^0(1-p)^{1-0} = p * p * (1-p) = p^2(1-p) \quad (2.15)$$

2.5 – кестеде келтірілген 3 потенциалды несие алушылардың бірінші және екіншісінің несиені қайтару ықтмалдығы $p * p = p^2$, ал үшіншісінің несиені қайтару ықтималдығы $1 - p$. Үш оқиғаның ықтималдығы $p^2(1-p)$-ге тең.

Максималды шынайылық әдісі – бұл белгісіз параметрді бағалау әдісі және ол шынайылық функцияны максимизациялау жолымен жүзеге асады. Мына $P\left(\vec{y}\middle|p\right)=p^2(1-p)$ жағдайдайында, p-ның мәні макисмумға жетуі қажет. Идеяның шығу тегі, іріктеме біз үшін қол жетімді жалпы халық туралы білімнің жалғыз қайнар көзі болып табылады. Іріктеуде жалпы халық туралы білетініміздің бәрі көрсетілген. Сондықтан, біз таңдауға болатын үлгі - бұл халықтың қолы жететін ең дәл ұсынуы. Сондықтан, қол жетімді үлгі ең ықтимал болатындай параметрді табу керек.

Біз оңтайландыру мәселесімен айналысатынымыз анық, онда функцияның экстремум нүктесін табу қажет. Экстремум нүктесін табу үшін бірінші ретті шартты қарастыру керек, яғни функцияның туындысын нөлге теңестіріп, керекті параметрге қатысты теңдеуді шешу керек. Алайда көптеген факторлардың көбейтіндісінің туындысын іздеу ұзаққа созылған зат болуы мүмкін, бұған жол бермеу үшін арнайы әдіс қажет - ықтималдық функциясының логарифміне көшу. Назар аударатын мәселенің бірі бұл жерде $P\left(\vec{y}\middle|p\right)$ экстремум функциясын іздеп отырған жоқпыз, экстремум нүктесін іздеп отырмыз яғни $P\left(\vec{y}\middle|p\right)$ максимумға жеткендегі p белгісіз параметрінің мәнін іздеп отырмыз. Логарифмге өту кезінде, экстремум нүктесі өзгермейді, себебі логариф монотонды функция.

Жоғарғы жақта талқыланған мәселеге яғни ипотекалық несие беруге оралсақ. Алдымен шынайылықтың логарифмдік функциясына тоқталамыз[54]:

$$\log P\left(\vec{y}\middle|p\right)=\log p^2(1-p)=2\log p+\log(1-p) \qquad (2.16)$$

Енді өрнекті p бойынша дифференциалдай аламыз:

$$\frac{\partial \log P\left(\vec{y}\middle|p\right)}{\partial p}=\frac{\partial}{\partial p}\left(2\log p\right)+\log(1-p))=\frac{2}{p}-\frac{1}{1-p} \qquad (2.17)$$

Сонымен, туындыны нөлге теңестіріп бірінші шартты қарастырамыз:

$$\frac{2}{p}-\frac{1}{1-p}=0\Rightarrow\frac{2}{p}=\frac{1}{1-p}\Rightarrow 2(1-p)=p\Rightarrow p=\frac{2}{3}.$$

Түйсікке сеніп болжаған болжаған ипотекалық несиені қайтару ықтималдығын бағалаудағы теориялық негіздемеміз $p=\frac{2}{3}$ тең болды.

Тек түйсікке сүйеніп әрбір несие алушы қарызды қайтармайды деп ойлауға болмайды. Несиені қайтару ықтималдығы $p=\frac{2}{3}$ дей келе мынадай факторлар ескерілмеген: потенциалды клиенттің жалақысы және ай сайынғы несие бойынша төлемінің мөлшері. Жоғарыда несиені қайтару ықтималдығы әр клиент үшін осы факторлар ескеріліп есептелген еді. Алынған ықтималдықтар $\frac{2}{3}$-ке тең болатын тұрақтыдан айырмашылығы қисынды.

Іріктеудің шынайылығын анықтау қажет. Іріктеудің шынайылығы $p = \frac{2}{3}$ тұрақты мәнге тең болған кезде:

$$P\left(\vec{y}\middle|p\right) = p^2(1-p) = \frac{2}{3}^2\left(1-\frac{2}{3}\right) \approx 0.148$$

$\vec{x}$ факторларын ескере отырып, несиені өтеу ықтималдығын есептеу кезінде іріктеудің шанайылығы:

$$P\left(\vec{y}\middle|p\right) = \prod_{i=1}^{3} p^{y_i}(1-p)^{(1-y_i)} = p_1^1(1-p_1)^{1-1} * p_2^1(1-p_2)^{1-1} * p_3^0(1-p_3)^{1-0} = p_1 * p_2 * (1-p_3) =$$

$$= 0.99 * 0.73 * (1-0.45) \approx 0.397$$

Факторларға байланысты есептелген ықтималдылықпен сынаманың ықтималдығының тұрақты мәні кезінде жоғары болды. Бұл нені білдіреді? Бұл факторларды білу әрбір клиент үшін несиені қайтару мүмкіндігін дәл анықтауға септігін тигізеді. Сондықтан, келесі несие берген кезде, қарыздың қайтарылу ықтималдығын қолданған дұрыс болады[55]. Егер іріктеу шынайылығының функциясын максималдасақ белгілі алгоритмдерді қолдануымыз қажет болады. Ол аталмыш ипотекалық несие алушылар үшін 0.99 және 0.01 ықтималдықтарды есептеу үшін қажет. Мүмкін бұл алгоритм іріктеулерді оқыту кезінде жақсы нәтиже көрсетуі мүмкін. Бұл алгоритм сызықты болмайтыны белгілі. Егер факторлар белгілі болса, несиені қайтару ықтималдығы бірдей болады ма? Әрине жоқ. 2.5 – ші кестедегі деректерге сай бірінші номерлі клиент жалақысының 2.5% пайызын несиені қайтаруға жұмсайды. Ал екінші номердегі клиент 27.8% жұмсайды. Сонымен қатар 2.6-шы суретте «Клиенттер классификациясы» графигінен көретініміздей бірінші номерлі клиент екінші номерлі клиентке қарағанда класқа бөлетін сызықтан алыс орналасқан. $f(w,x) = w_0 + w_1x_1 + w_2x_2$ функциясы бірінші және екінші клиенттер үшін әртүрлі мәндерді қабылдайды: бірінші клиент үшін 4.24, ал екінші клиент үшін 1.0. Көріп отырғанымыздай екінші клиент ипотекалық несиені азырақ сұрағанда немесе жалақысы көбірек болғанда екеуінің несиені қайтару ықтималдығы бірдей болуы мүмкін еді. Сызықты тәуелділікті алдау мүмкін емес. Егер шынымен w коэффициентті ойша ала салмай есептеген болсақ, әрбір клиент үшін несиені қайтару ықтималдығын есептеуде жақсы нәтиже көрсетеді деп ауыз толтырып мәлімдеуге болар еді. Қойылған шарт бойынша w коэффициенті есептелетін болған соң, есептелген коэффициент дәл нәтиже береді деп тұжырымдаймыз[56].

Енді, әрбір ипотека алушы жеке тұлғалардың несиені қайтару ықтималдығын болжау үшін салмақ векторы қалай анықталатыны талқыланады[57]:

1. Нәтижеге әсер ететін бүтін айнымалы мен фактордың арасындағы тәуелділік сызықты. Осыған байланысты, $f(w,x) = \vec{w}^T X$ түріндегі клиенттерді +1 немесе -1 мен 0 класына бөлетін сызықты регрессияның функциясы қолданылады. Аталмыш жағдайда, теңдеу мына (2.15) түрінде болады:

$$f(w,x)=w_0+w_1x_1+w_2x_2$$

2. Мына түрдегі $p_+=\frac{1}{1+e^{-\vec{w}^T\vec{x}}}=\delta\left(\vec{w}^T\vec{x}\right)$ кері логит түрлендіру функциясын +1 класқа жататын нысанның ықтималдығын анықтау үшін қолданамыз.

3. Іріктеп оқытуды Бернуллидің жалпыланған схемасын іске асыру деп есептейміз, яғни кез-келген нысан үшін кездейсоқ шама құрылады, оның ықтималдығының мәні 1 және $(1-p)-0$ ықтималдығы болады.

4. Қол жетімді үлгі ең қолайлы болуы үшін біз алынған факторларды ескере отырып, үлгінің ықтималдылық функциясын барынша арттыруымыз керек екенін білеміз. Басқаша айтқанда, үлгі неғұрлым қолайлы болатын параметрлерді таңдау керек. Біздің жағдайда таңдалған параметр - несиені өтеу мүмкіндігі p, бұл өз кезегінде белгісіз w коэффициенттерге байланысты. Бұл бізге салмақтың $\vec{w}$ векторын табуымыз керек дегенді білдіреді, онда іріктеу ықтималдығы максимум болады.

5. Іріктеудің ықтималдылық функциясын максимумға келтіру үшін максималды ықтималдылық әдісін қолдануға болатынын білеміз. Бұл әдіспен жұмыс жасаудың барлық әдіс-тәсілдері белгілі.

Енді бөлімнің басында біз логикалық жоғалту функциясының нысан сыныптарының қалай бөлінгеніне байланысты екі түрін алғымыз келгенін есімізге түсірейік. Екі сыныпты жіктеуде проблемалар туындаған кезде сыныптар +1 және 0 немесе -1 деп белгіленді. Белгілеуге байланысты нәтижесінде тиісті шығын функциясы болады.

Нысанды +1 және 0 класына бөлу.

Жоғарыда, іріктеудің шынайылығын анықтауда және несие алушының несиені қайтару ықтималдығын есептеулер факторлар мен енгізілген w коэффициенттерді анықтауда мына формуланы қолданылған[58]:

$$P\left(\vec{y}\middle|p\right)=\prod_{i=1}^{3}p^{y_i}(1-p)^{(1-y_i)} \qquad (2.18)$$

Шын мәнінде, p_i - логистикалық шығын функциясының $p_+=\frac{1}{1+e^{-\vec{w}^T\vec{x}}}=\delta\left(\vec{w}^T\vec{x}\right)$ берілген салмақ $\vec{w}$ векторының мәні. Ендеше, іріктеудің шынайылық функциясын былай жазуға ешқандай кедергі жоқ:

$$P\left(\vec{y}\middle|\delta\left(\vec{w}^TX\right)\right)=\prod_{i=1}^{n}\delta\left(\vec{w}^T\vec{x}_i\right)^{y_i}\ 1-\delta\left(\vec{w}^T\vec{x}_i\right)^{(1-y_i)}\to\max \qquad (2.19)$$

Кейде кейбір тәжірибесіз сарапшыларға бұл функцияның қалай жұмыс істейтінін бірден түсіну қиынға соғады. Барлығын анықтайтын 4 қысқа мысалды қарастырайық:

1. Егер $y_i = +1$ (оқыту үлгісіне сәйкес нысан +1 класына жататын болса) және біздің $\delta\left(\vec{w}^T X\right)$ алгоритміміз объектіні +1 класына 0,9-ға тең ықтималдығын анықтайтын болса, онда іріктеу ықтималдығының бұл бөлігі келесідей есептеледі:

$$0.9^1 * (1-0.9)^{(1-1)} = 0.9^1 * 0.1^0 = 0.9$$

2.Егер, $y_i = +1$, ал $\delta\left(\vec{w}^T X\right) = 0.1$ болса есептеу келесідей болады:

$$0.1^1 * (1-0.1)^{(1-1)} = 0.1^1 * 0.9^0 = 0.1$$

3.Егер, $y_i = 0$, ал $\delta\left(\vec{w}^T X\right) = 0.1$ болса есептеу келесідей болады:

$$0.1^0 * (1-0.1)^{(1-0)} = 0.1^0 * 0.9^1 = 0.9$$

4. Егер, $y_i = 0$, ал $\delta\left(\vec{w}^T X\right) = 0.9$ болса есептеу келесідей болады:

$$0.9^0 * (1-0.9)^{(1-0)} = 0.9^0 * 0.1^1 = 0.1$$

Шынайылық функциясы 1 және 3 жағдайларда, немесе жалпы жағдайда - объектіні +1 класына тағайындау ықтималдығының дұрыс болжанған мәндерімен максималды болатыны анық. Нысанды +1 класына жатқызу ықтималдығын анықтау кезінде біз w коэффициенттерін ғана білмейтіндігімізге байланысты оларды іздейміз. Жоғарыда айтылғандай, бұл оңтайландыру мәселесі, онда алдымен салмақтың w векторына қатысты ықтималдылық функциясының туындысын табу керек. Алайда, алдымен тапсырманы жеңілдетудің мәні бар: біз туынды шынайылық функциясының логарифмінен іздейміз[59]:

$$L_{\log}\left(X, \vec{y}, \vec{w}\right) = \sum_{i=1}^{n}\left(-y_i \log_e \delta\left(\vec{w}^T \vec{x}_i\right) - (1-y_i)\log_e\left(1-\delta\left(\vec{w}^T \vec{x}_i\right)\right)\right) \to \min \qquad (2.20)$$

Логарифмдегеннен кейін логистикалық қателік функциясында «+» таңбасының «-» таңбасына ауысу себебі, модельдің сапасын бағалауда функцияның мәнін азайту әдеттегідей, содан кейін біз өрнектің оң жағын көбейтіп, сәйкесінше максималдаудың орнына қазір функцияны азайтамыз[60]. Енді коэффициенттерді табу үшін бізге логистикалық қателік функциясының туындысын табу керек, содан кейін градиенттік түсу немесе стохастикалық градиентті түсіру сияқты оңтайландырудың сандық әдістерін қолдана отырып, ең оңтайлы w коэффициенттерді таңдау керек.

Нысанды +1 және -1 класына бөлу.

Мұнда тәсіл 1 және 0 сыныптардағыдай болады, бірақ Logistic Loss логистикалық шығын функциясын жоғалту жолы көрсетіледі. Біз ықтималдылық функциясы үшін «егер ... онда ...» операторын қолданамыз. Егер i-ші нысан +1 класына жататын болса, онда іріктеудің шынайылығын есептеу үшін p ықтималдығын қолданамыз, егер нысан -1 класына жататын болса, онда біз шынайылық функциясына $(1-p)$ қойылады. Шынайылық функциясы келесідей:

$$P\left(\vec{y}\middle|\delta\left(\vec{w}^{T}X\right)\right)=\prod_{i=1}^{n}\delta\left(\vec{w}^{T}\vec{x}_i\right)^{[y_i=+1]}\left(1-\delta\left(\vec{w}^{T}\vec{x}_i\right)\right)^{[y_i=-1]}\to\max \qquad (2.21)$$

Бұл қалай жұмыс істейтінін білу үшін 4 жағдайды қарастырамыз:

1. Егер, $y_i=+1$ және $\delta\left(\vec{w}^{T}\vec{x}_i\right)=0.9$ болса іріктеудің шынайылығы 0.9 болады.

2. Егер, $y_i=+1$ және $\delta\left(\vec{w}^{T}\vec{x}_i\right)=0.1$ болса іріктеудің шынайылығы 0.1 болады.

3.Егер, $y_i=-1$ және $\delta\left(\vec{w}^{T}\vec{x}_i\right)=0.1$ болса іріктеудің шынайылығы $1-0.1=0.9$ болады.

4. Егер, $y_i=-1$ және $\delta\left(\vec{w}^{T}\vec{x}_i\right)=0.9$ болса іріктеудің шынайылығы $1-0.9=0.1$ болады.

Әлбетте, 1 және 3 жағдайларда, ықтималдықтар алгоритммен дұрыс анықталған кезде, шынайылық функциясы барынша максималданады, күтетін нәтиже дәл осы еді. Алайда, бұл тәсіл өте қиын және төменде біз ықшамды жазба қарастырылады. Алдымен, шынайылық функциясын белгінің өзгеруімен логарифмделеді, өйткені ол азайтылады[61].

$$L_{\log}\left(X,\vec{y},\vec{w}\right)=\sum_{i=1}^{n}\left(-[y_i=+1]\log_e\delta\left(\vec{w}^{T}\vec{x}_i\right)-[y_i=-1]\log_e\left(1-\delta\left(\vec{w}^{T}\vec{x}_i\right)\right)\right)\to\min \qquad (2.22)$$

$\delta\left(\vec{w}^{T}\vec{x}_i\right)$ -тің орнына $\dfrac{1}{1+e^{-\vec{w}^{T}\vec{x}_i}}$ өрнекті қоямыз:

$$L_{\log}\left(X,\vec{y},\vec{w}\right)=\sum_{i=1}^{n}\left(-[y_i=+1]\log_e\left(\frac{1}{1+e^{-\vec{w}\vec{x}}}\right)-[y_i=-1]\log_e\left(1-\frac{1}{1+e^{-\vec{w}\vec{x}}}\right)\right)\to\min \qquad (2.23)$$

Оң жақтағы қосындыны азайтып қарапайым арифметикалық амалдарды орындап мына түрге келеміз:

$$L_{\log}\left(X,\vec{y},\vec{w}\right)=\sum_{i=1}^{n}\left(-[y_i=+1]\log_e\left(\frac{1}{1+e^{-\vec{w}\vec{x}}}\right)-[y_i=-1]\log_e\left(\frac{1}{1+e^{-\vec{w}\vec{x}}}\right)\right)\to\min \qquad (2.24)$$

Әрі қарай, «егер,...онда»деген операторларды алып тасталынады. Байқайтынымыздай, y_i нысаны +1 класына жатқанда логорифмдік өрнектің e бөліміне e-нің дәрежесіне $-\vec{w}^{T}\vec{x}_i$ жазылады. Егер, нысан -1 класына жатса, e-нің дәрежесіне $+\vec{w}^{T}\vec{x}_i$ жазылады. Дәрежесіндегі жазбаларды ықшамдап, екі жағдайды біріктіреміз: $-y_i\vec{w}^{T}\vec{x}_i$. Логистикалық функцияның қателігі келесідей түрге келеді:

$$L_{\log}\left(X,\vec{y},\vec{w}\right)=\sum_{i=1}^{n}-\log_e\left(1-\frac{1}{1+e^{-y_i\vec{w}^{T}\vec{x}}}\right)\to\min \qquad (2.25)$$

Логарифм ережелеріне сәйкес біз бөлшекті айналдырып, «-» (минус) таңбасын алып, логарифмнен кейін қоя отырып келесідей түрге қол жеткіземіз:

$$L_{\log}\left(X, \vec{y}, \vec{w}\right) = \sum_{i=1}^{n} \log_e\left(1 + e^{-y_i \vec{w}^T \vec{x}}\right) \to \min \qquad (2.26)$$

Бұл іріктеулер оқыту кезінде пайдаланылатын +1 және -1 класына жататын шығын функциясы Logistic Loss.

Қолданылатын логистикалық регрессия әдісінің ең басты ерекшеліктерінің бірі белгілі сипаттамалық мәндермен тәуелді айнымалы мәнінің математикалық күту мәнін табу, яғни $E(Y|x)$, мұндағы Y және x тәуелді айнымалы мен сипаттамалардың векторын білдіреді [62].

Мынадай белгілеулер енгіземіз:

$p(x_i) = E(Y|x_i)$ i-ші несие үшін дефолт болу ықтималдығы.

Ендеше, логистикалық регрессия формуласы келесі түрде ұсынылады:

$$p(x_i) = G(x_i, \omega) = \frac{e^{\omega_0 + \omega_1 x_{i1} + \omega_2 x_{i2} + \ldots \omega_p x_{ip}}}{1 + e^{\omega_0 + \omega_1 x_{i1} + \omega_2 x_{i2} + \ldots \omega_p x_{ip}}} = \frac{e^{x_i \omega}}{1 + e^{x_i \omega}} \qquad (2.27)$$

Логоримфдік түрлендіру мынадай түрге келеді:

$$\ln\left(\frac{p(x_i)}{1 - p(x_i)}\right) = \omega_0 + \omega_1 x_{i1} + \omega_2 x_{i2} + \ldots + \omega_p x_{ip} + \varepsilon_i = x_i W + \varepsilon_i \qquad (2.28)$$

Диссертациялық жұмыста логистикалық регрессияның коэффициенттерін есептеу үшін ықтималдылықтың максималды әдісі қолданылады, бұл коэффициенттер ықтималдылық функциясын барынша көбейту арқылы есептеледі. Ықтималдылық функциясын алу үшін y_i оқиғасының ықтималдық функциясын ұсынамыз. Ықтималдық функциясы i-ші несие үшін келесі түрде жазылады[63]:

$$p(x_i)^{y_i} (1 - p(x_i))^{1 - y_i} \qquad (2.29)$$

Ипотекалық несие алушылардың төлем қабілетін анықтау моделін құру мәселесін шешу үшін сипаттамалардың мәндері дербес бөлінеді, сондықтан ықтималдылық функциясы былайша жазылады:

$$L(\omega) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1 - y_i} \qquad (2.30)$$

(2.30) функциясының көмегімен есептеулерді орындау үшін біз логарифмдеу операциясын орындаймыз және мына түрге келеміз:

$$l(\omega) = \ln(L(\omega)) = \sum_{i=1}^{n} \{y_i \ln(p(x_i)) + (1 - y_i) \ln(1 - p(x_i))\} \qquad (2.31)$$

(2.28) теңдеудің функциясын максимумға көбейтетін коэффициенттерді табу үшін (2.31) теңдеуді ω бойынша дифференциялдап мына түрге келеміз:

$$\frac{dl(\omega)}{d\omega_0} = \sum_{i=1}^{n} (y_i - p(x_i)) = 0 \qquad (2.32)$$

$$\frac{dl(\omega)}{d\omega_j} = \sum_{i=1}^{n} (y_i - p(x_i))x_{ij} = 0, \quad j = 1,2,...p \qquad (2.33)$$

(2.32), (2.33) теңдеулерінің ω_i бойынша сызықты емес екендігіне байланысты Гаусс-Ньютон әдісі қолданылады, оның көмегімен алынған теңдеулер жүйелерінен табылған салмақ коэффициенттері бойынша есептеулер жүргізілген[64].

2.5 ROC талдау

Логистикалық регрессия моделінің сапасын тексеруде жақсы құралдардың бірі ол – ROC талдау.

ROC қисық (Receiver Operator Characteristic) машиналық оқыту кезінде екілік классификация нәтижелерін көрсету үшін жиі қолданылатын қисық. Атау сигналдарды өңдеу жүйелерінен шыққан[65]. Екі класс болғандықтан, олардың бірі оң нәтижелері бар класс деп аталады, екіншісі – теріс нәтижелері бар класс. ROC қисығы дұрыс жіктелген жағымды мысалдар санының қате жіктелген теріс мысалдар санына тәуелділігін көрсетеді. ROC талдауының терминологиясында біріншісі шынайы оң, ал екіншісі жалған теріс деп аталады. Бұл жағдайда классификатордың белгілі бір параметрі болады деп болжанады, оның өзгеруіне байланысты біз бір немесе басқа сыныпты екі класқа бөлеміз. Бұл параметр көбінесе шекті немесе кесу мәні деп аталады. Оған байланысты I және II типтегі қателіктердің әртүрлі мәндері алынады. Логистикалық регрессияда кесу шегі 0 – ден 1 – ге дейін өзгереді – бұл регрессия теңдеуінің есептелген мәні. Мұны рейтинг деп аталады [66].

I және II типтегі қателіктердің мәнін түсіну үшін модель бойынша жіктеу нәтижелері мен мысалдардың кластарға нақты (объективті) тиістілігі негізінде құрылған төрт өрісті шатастыру матрицасын қарастырылады:

Кесте 2.6 – Моделді классификациялау

Модель	Іс жүзінде оң	Іс жүзінде теріс
Оң	TP	FP
Теріс	FN	TN

– TP (True Positives) – дәл классификацияланған оң мысалдар (шынайы оң жағдайлар).

– TN (True Negatives) – дәл классификацияланған теріс мысалдар (шынайы теріс жағдайлар).

– FN (False Negatives) – теріс классификацияланған, оң мысалдар (I текті қателік). Қажетті оқиға қателіктің негізінде анықталмаған жағдайда бұл – «жалған өту» деп аталады.

– FP (False Positive) – оң классификацияланған, теріс мысалдар (II текті қателік). Бұл жалған анықтау, оқиғаның анықталмағанына қарамастан оның бар екендігі туралы шешімнің шығуы (жалған оң жағдайлар) .

Қайсысы оң оқиға немесе қайсысы теріс оқиға екені нақты есепке байланысты анықталады. Мысалы, егер ипотекалық несие алушы ааматтардың төлем қабілетін болжаймыз. Оң класқа «төлем қабілеті жоқ» ал теріс класқа «төлем қабілеті бар» бөлінеді, немесе потенциалды клиенттің төлем қабілеті бар болуының ықтималдығын ғана анықтағымыз келсе оң класқа «төлем қабілеті бар» деп жатқызуға болады. Талдау кезінде олар көбінесе абсолютті көрсеткіштермен емес, пайызбен көрсетілген салыстырмалы үлестермен (rates) жұмыс істейді[67]:

– Шынайы оң мысалдардың үлесі (True Positive Rate): $TPR = \frac{TP}{TP + FN} \cdot 100\%$

– Жалған оң мысалдардың үлесі (False Positive Rate): $FPR = \frac{FP}{TN + FP} \cdot 100\%$

Әрі қарай екі анықтама енгізіледі: модельдің сезімталдығы мен ерекшелігі. Бинарлы классификатордың объективті құндылығы модельдің сезімталдығы мен ерекшелігіне қарай анықталады.

Сезімталдық (Stnsitivity) – шынайы оң жағдайлардың үлесі:

$$S_e = TRP = \frac{TP}{TP + FN} \cdot 100\% \qquad (2.34)$$

Ерекшелік (Specificity) – модельмен дұрыс идентификацияланған шынайы теріс жағдайлардың үлесі:

$$S_p = \frac{TN}{TN + FP} \cdot 100\% \qquad (2.35)$$

$$FPR = 100 - S_p \qquad (2.36)$$

Сезімталдығы жоғары модель әдетте оң жағдайларда дәл нәтиже береді (оң мысалдарды анықтайды). Ал ерекшелігі жоғары модель әдетте теріс жағдайлар болғанда дәл нәтие береді (теріс мысалдарды анықтайды). Қаржы ұйымында ипотекалық несие беруде экономика тілімен айтқанда дефолт болмауы үшін потенциалды клиенттерді теріс нәтижелі және оң нәтижелі деп жіктеу моделі жұмысы ерекше.

Сезімталдық – төлем қабілеті жоқ клиенттерді максималды түрде анықтап қаржы ұйымында дефолт болуына тосқауыл болады. Ерекшелік – потенциалды клиенттердің төлем қабілетін анықтауда белгілі бір нәтижеге қол жеткізуге септігін тигізеді.

ROC қисығы келесідей жолмен алынады:

– d_x қадаммен 0-ден 1-ге дейін өзгеретін әрбір кесу шегінің мәні үшін (мысалы 0.01), S_e сезімталдық мәні және ерекшелік мәні S_p есептеледі. Сонымен қатар, шектің альтернативі ретінде іріктеу мысалдарындағы мәндері болуы мүмкін.

– Тәуелділік графигі сызылады: Y осі бойынша S_e сезімталдық шектеледі, ал X осі бойынша жалған оң жағдайлардың үлесі [68]:

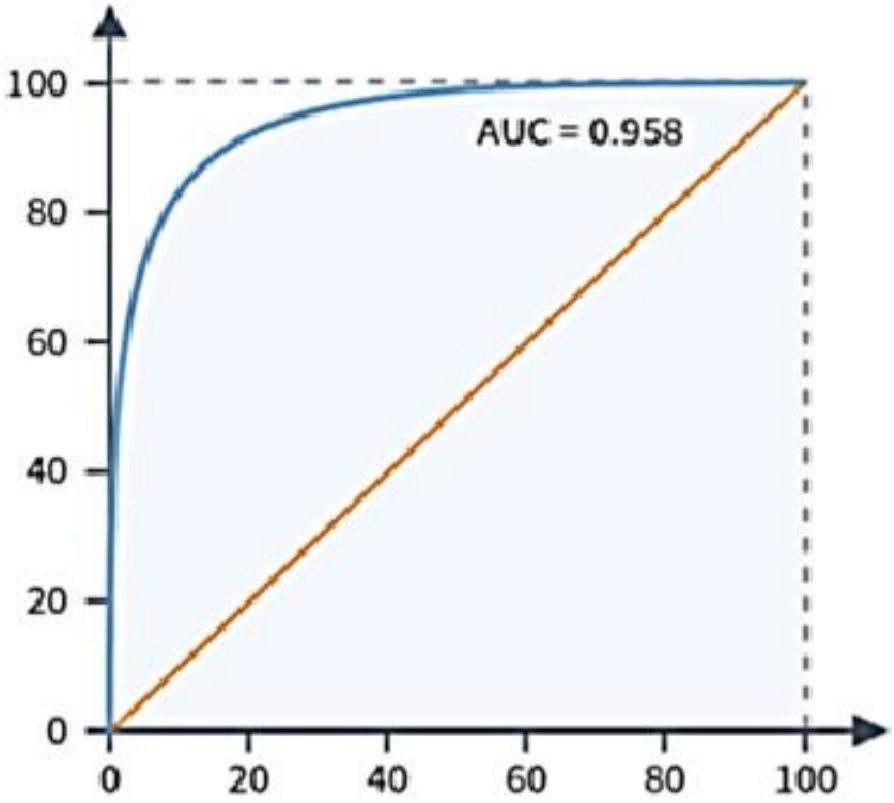
$$FPR = 100 - S_p \quad (2.37)$$

2.5.1 ROC қисықтың канондық алгоритмін құру

Кірістер: L – мысалдар жиыны $f[i]$ - алынған модел рейтингі немесе i -ші мысал шығысы оң нәтижелі дегенді білдіреді; min және max – оралатын f -тің максималды және минималды мәні; d_x - қадам; P және N сәйкесінше оң және теріс мысалдар мөлшері.

1) $t = min$
2) *repeat*
3) $FP = TP = 0$
4) *for*all examples i belongs to$\boldsymbol{L}$ {
5) *if* $f[i] \geq t$ then // this exampleis beyond the threshold
6) *if*positiveexample, then
7) { $TP = TP + 1$ }
8) *else* // this is a negativeexample
9) { $FP = FP + 1$ }
10) } // exit from thecondition
11) $Se = \frac{TP}{P} * 100$
12) $point = FP/N$ //calculation $(100 - Sp)$
13) Add thepoint $(point, Se)$in ROC-curve
14) $t = t + d_x$
15) *while*$(t > max)$.

Сурет 2.8 – ROC қисығының канондық алгоритмі



Сурет 2.9 – ROC қисық

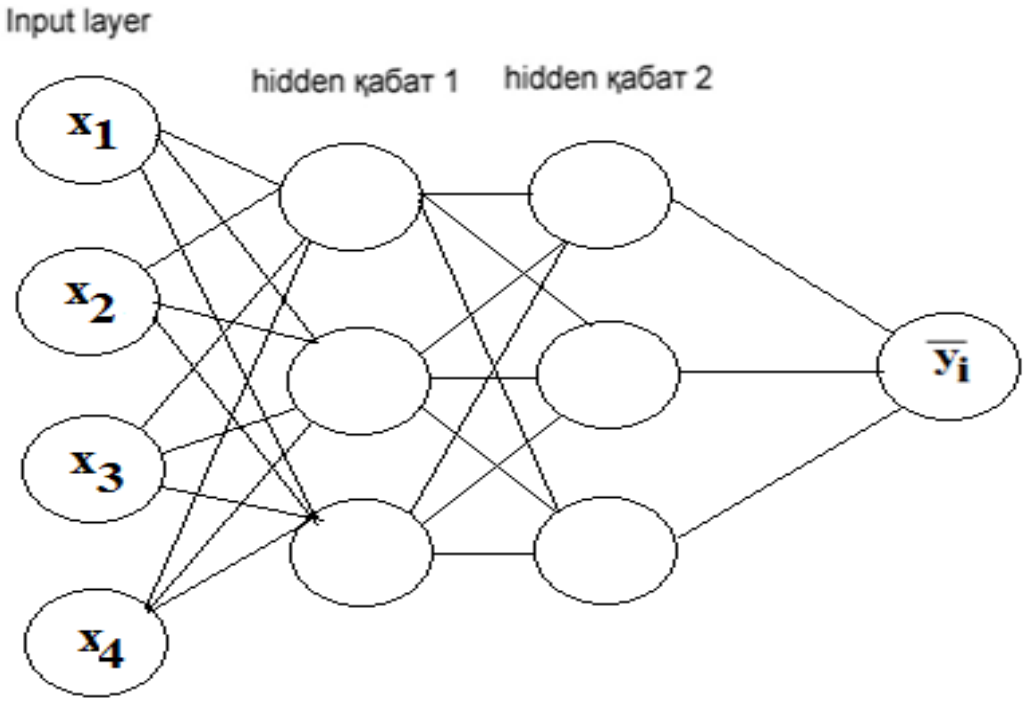
График көбінесе $y = x$ түзу сызығымен толықтырылады. Байқалатынындай, жоғарыда есептелген әдістерге қарағанда ROC қисығының нүктелерін есептеудің үнемді әдісі бар. Оның есептеу күрделілігі сызықты емес және ол $O(n^2)$-қа тең. Әрбір шек үшін жазбаларды «өту» және әрдайым TP және FP есептеу қажет. Егер классификатордың шығыс өрісінің кему ретімен сұрыпталған мәліметтер базасына төмен түссек (рейтинг), онда TP және FP мәндерін біртіндеп жаңартып, ROC қисық сызығының барлық нүктелерінің мәндерін бір жолға есептей аламыз[69].

2.6 Көпқабатты нейрондық желілерді қолданып ипотекалық несие алушылардың төлем қабілеттерін анықтау

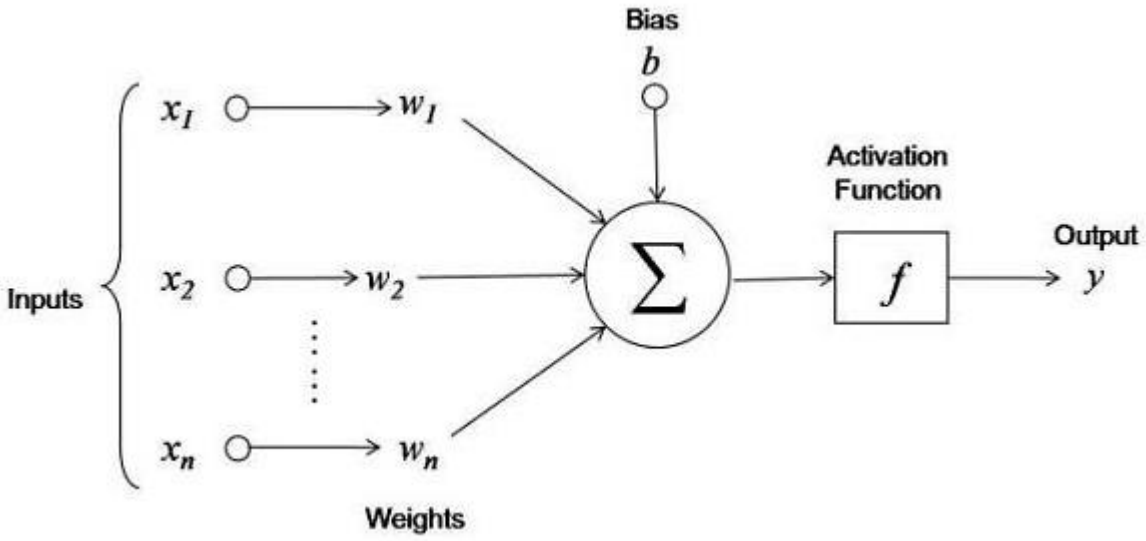
Data Mining-тің тағы бір технологияларының бірі - бұл әр түрлі деректерді талдау міндеттерін шешу үшін транзиттік фазадағы бар деректерді құрастырғаннан кейін жасалған нейрондық желілер. Осы әдістерді қолдану кезінде, ең алдымен, олардың әрқайсысындағы қабаттар саны мен нейрондардың санын таңдау туралы сұрақ туындайды. Содан кейін салынған желі оқу үдерісі деп аталады. Бұл кезеңде желі нейрондары енгізілген деректерді итеративті түрде өңдейді және олардың салмақтарын реттейді, осылайша желі енгізілген деректерді жақсы болжайды. Қолда бар деректер тасымалданғаннан кейін, желі пайдалануға дайын болады және оны болжам жасау үшін пайдалануға болады[70].

Ең кіші нейрон желісі тәуелсіз айнымалы ретінде бір түйіннен және бір шығысы бар тәуелді айнымалы ретінде бір мақсаттан тұрады. Оның кешені жасырын қабаттар мен қосымша айнымалылардың кеңеюіне байланысты ұлғайды (2.10, 2.11 – суреттер). Жасырын қабаттарсыз нейрон желісін зерттеу нәтижелері регрессия нәтижелеріне ұқсайды. Енгізген әр айнымалы жасырын қабаттағы әр вариациямен байланысты, ал әр жабық немесе жабық өзгергіштік әр өзгеретін нәтижемен байланысты. Жасырын қабаттар оңтайлы нәтижеге қол жеткізу және қолдану функцияларын біріктіреді. Көлеңкелі қабаттар әдетте жақын емес [71].

Нейрондық желілер – бұл статистикалық модель, ол көптеген нейрондар жиынынан тұрады және қабаттарға топтастырылып желі құрайды. Әрбір нейрон ол – бірлік функциясымен берілген өңдеуші элемент. Нейрондар арасындағы байланыстар желіні құрайды және олар жекелеген деректердің арасындағы байланысты анықтауға мүмкіндік береді [72]. Алға қойған міндетті шешу үшін көпқабатты нейронның 2 қабаты пайдаланылды.



Сурет 2.10 – Қолданылған көп қабатты нейронды желі



Сурет 2.11 – Нейрондық желі схемасы

$$h_i^{(1)} = f^{(1)}\left(\sum \omega_i^{(1)} x_i^{(1)} + b_i^{(1)}\right) \quad (2.38)$$

$$h_i^{(2)} = f^{(2)}\left(\sum \omega_i^{(1)} h_i^{(1)} + b_i^{(2)}\right) \quad (2.39)$$

$$\overline{y_i} = f^{3}\left(\sum \omega_i^{(3)} h_i^{(2)} + b_i^{(3)}\right) \quad (2.40)$$

$$f^{(1)}(x) = f^{(2)} = x^{+} = sigmoid(x) \quad (2.41)$$

$$f^{3}(x)_i = \frac{1}{1+e^{-x}} \quad (2.42)$$

x_i-кіріс сигналдары;

$\overline{y_i}$ - моделдің шығысы;

$\theta\left\{\omega_i^{(1)}, \omega_i^{(2)}, \omega_i^{(3)}, b_i^{(1)}, b_i^{(2)}, b_i^{(3)}\right\}$-модель параметрлері;

$f^{(3)}(x)$- белсендіру функциясы сигмойда.

α-белсендіру жылдамдығын арттыратын мән;

$$S = \sum_{i=1}^{n} \omega_i x_i \quad (2.43)$$

n – нейронның кіріс саны

x_i – нейронның i-ші кірісінің мәні

w_i – і-ші синапс салмағы

S – нейрон

Нейронның аксонының мәні мына формуламен анықталады:

$$Y = f(S) \quad (2.44)$$

Сигмойда белсендіру функциясының басты артықшылығы - ол бүкіл абцисса осінде дифференциалданатын және өте қарапайым туындыға ие [73]:

$$f(x) = \alpha f(x)(1 - f(x)) \tag{2.45}$$

Нейрондық желілерді қолданып модел құру үшін кері тарату алгоритмі қолданылады. Нейрондық желілерді зерттеу барысында кері таратудағы нейрондық желілер бірнеше қабатты нейроннан тұратынын, әрбір нейронның і қабаты әрбір $i+1$ нейронның қабатымен байланысты екенін байқадым. Диссертациялық жұмыстағы нейрондық желілердің міндеті - $Y = F(X)$, тәуелділігін табу, мұндағы X - кіріс, Y - шығу векторлары. Жалпы жағдайда, кіріспе мәліметтердің шектеулі жиынтығымен мұндай мәселе шексіз шешімге ие болады. Жұмыста оқыту кезінде іздеу кеңістігін шектеуде нейрондық желілердің қателіктерінің мақсатты функциясын минимизациялау міндеті қойылды. Ол ең кіші квадраттар әдісімен табылады:

$$E_i = \frac{1}{2}\sum_{j=1}^{p}\left(y_j - d_j\right)^2 \tag{2.46}$$

Бұл жерде E_i - минимизациялау функциясын бағалау, y_j , j - ші нейрожелінің шығыс мәні

d_j, j-ші шығыстың мақсатты мәні

p - шығыс қабатындағы нейрондардың саны

Салмақты коэффициенттер әрбір итерация кезінде формула бойынша өзгереді

$$\Delta w_{ij} = -\mu \frac{\partial E}{\partial w_{ij}} \tag{2.47}$$

μ - оқытудың жылдамдығын анықтайтын параметр

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} * \frac{\partial y_j}{\partial S_j} * \frac{\partial S_j}{\partial w_{ij}} \tag{2.48}$$

y_j, j-ші нейронның шығыс мәні.

S_j – формула бойынша анықталған кіріс сигналдарының өлшенген қосындысы:

$$S_j = \sum_{i=1}^{n} \omega_i x_{ij} \tag{2.49}$$

мұндағы:

$$\frac{\partial S_j}{\partial w_{ij}} \equiv x_i \tag{2.50}$$

x_i, нейронның i -ші кірісінің мәні

(2.47) Формуланың бірінші көбейткішінің анықтамасы

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} * \frac{\partial y_k}{\partial S_K} * \frac{\partial S_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} * \frac{\partial y_k}{\partial S_K} * w_{jk}^{(n+1)} \tag{2.51}$$

Бұл жерде k, нейронның $n+1$ қабатындағы саны. Төмендегідей тұжырымға келсек,

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} * \frac{\partial y_j}{\partial S_j} \quad (2.52)$$

Әрі қарай n-ші қабатты анықтау үшін рекурсивті формуланы нақтылауымыз қажет, егер келесі $(n+1)$-ші қабат белгілі болса:

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} * w_{jk}^{(n+1)}\right] * \frac{dy_j}{dS_j} \quad (2.54)$$

Соңғы қабат келесідей анықталады:

$$\delta_j^{(N)} = \left(y_i^{(N)} - d_i\right) * \frac{dy_i}{dS_i} \quad (2.55)$$

Нейрондық желіні оқытудың келесідей алгоритмі қолданылды [74]:

1. Нейрондық желінің кірісін және шығыс анықтау.
2. Шығыс қабатын (2.51) формула бойынша есептеу және шығыс N қабатының салмағының өзгеруін (2.52) формула бойынша есептеу.
3. Нейрондық желінің қалған қабаттарын n = N-1..1 сәйкесінше (2.54) және (2.55) формулалармен есептеу.
4. Нейрондық желінің салмағын реттеу үшін:

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t) \quad (2.54)$$

5.Егер қателік айтарлықтай болса 1 кезеңге өту.

Екінші тарау бойынша қорытынды

Бұл тарауда DataMining әдістерін қолданып мәселені шешудің негізгі әдістері көрсетілген. Логистикалық регрессия, сызықтық регрессия, нейрондық желілері қолданылып деректер көлемдері бойынша өңдеу нәтижелеріне салыстырулар жүргізілген. Үлкен өлшемді деректерді өңдеу негізінде ипотекалық несие алушылардың төлем қабілетін анықтаудың сандық моделі көрсетілген. Үлгі өлшеміне негізделген есептеулер ұсынылған. Жеке тұлғалардың төлем қабілеттерін бағалайтын жүйе моделін құру міндеттері мен әдістері жан-жақты зерттелді.

3. ҮЛКЕН ӨЛШЕМДІ ҚҰРЫЛЫМДАНБАҒАН ДЕРЕКТЕРДІ ӨҢДЕУДІҢ САНДЫҚ МОДЕЛІН ҚҰРУ

3.1 Бағдарламалау тілінің көмегімен бағдарламалық қамтаманы құру

Компьютерлік модель – бұл жекелеген компьютерде немесе суперкомпьютерде жұмыс істеуге арналған бағдарлама.

Диссертациялық жұмыста жеке тұлғалардың төлем қабілетін анықтау үшін құрылған жүйенің моделін толық автоматтандыру үшін шынайы деректер талданды. Нәтиже бойынша жеке тұлғаға ипотекалық несие беру немесе бермеу туралы шешімді шығару жұмыстары да автоматтандырылған. Тәуелсіз айнымалылар тұтынушының жеке басы туралы енгізетін деректер болып табылады. Енгізілген деректер NoSQL технологиялары көмегімен біріңғай дерекқорымыз MongoDB-де сақталады және несие алушылардың деректері аталмыш MongoDB дерекқорынан алынып өңделеді. Бағдарламалық қамтама Python 3.8 бағдарламалау тілінде жазылған. NumPy - Python бағдарламалау тілі үшін ашық бастапқы кеңейтілген кітапхана, көпөлшемді массивтермен жұмыс істеуге арналған жоғары деңгейлі математикалық функциялармен жұмыс жасай алады[75]. Үлкен өлшемді құрылымданбаған деректерді ипотекалық несие алушылардың төлем қабілетін анықтау негізінде өңдеу үшін құрылған бағдарламалық кешеннің интерфейсі 3.5 – суретте көрсетілген. Суреттен көретініміздей, тұлғалардың жеке сипаттамалары яғни олар туралы деректер енгізуімізге болады. Бұл өз кезегінде деректерді өңдеу үшін қажет және енгізілген деректерді тәуелсіз айнымал депте атауымызға болады. Ал, айнымалымыз бағдарламаның нәтижесін көру батырмасын басқанда экранға берілетін болады. DataMining алгоритмдерін пайдалану[76,77] арқылы үлкен өлшемді деректерді өңдеп сәйкесінше нәтиже алатын боламыз. Бізде деректер әр түрлі форматта кездеседі[78,79], статистикалық деректер көбінесе Microsoft exel бағдарламасында болғандықтан деректерді өзіміздің біріңғай деректер қорымызға трансформациялау мәселесі бойынша жұмыс жасаймыз.

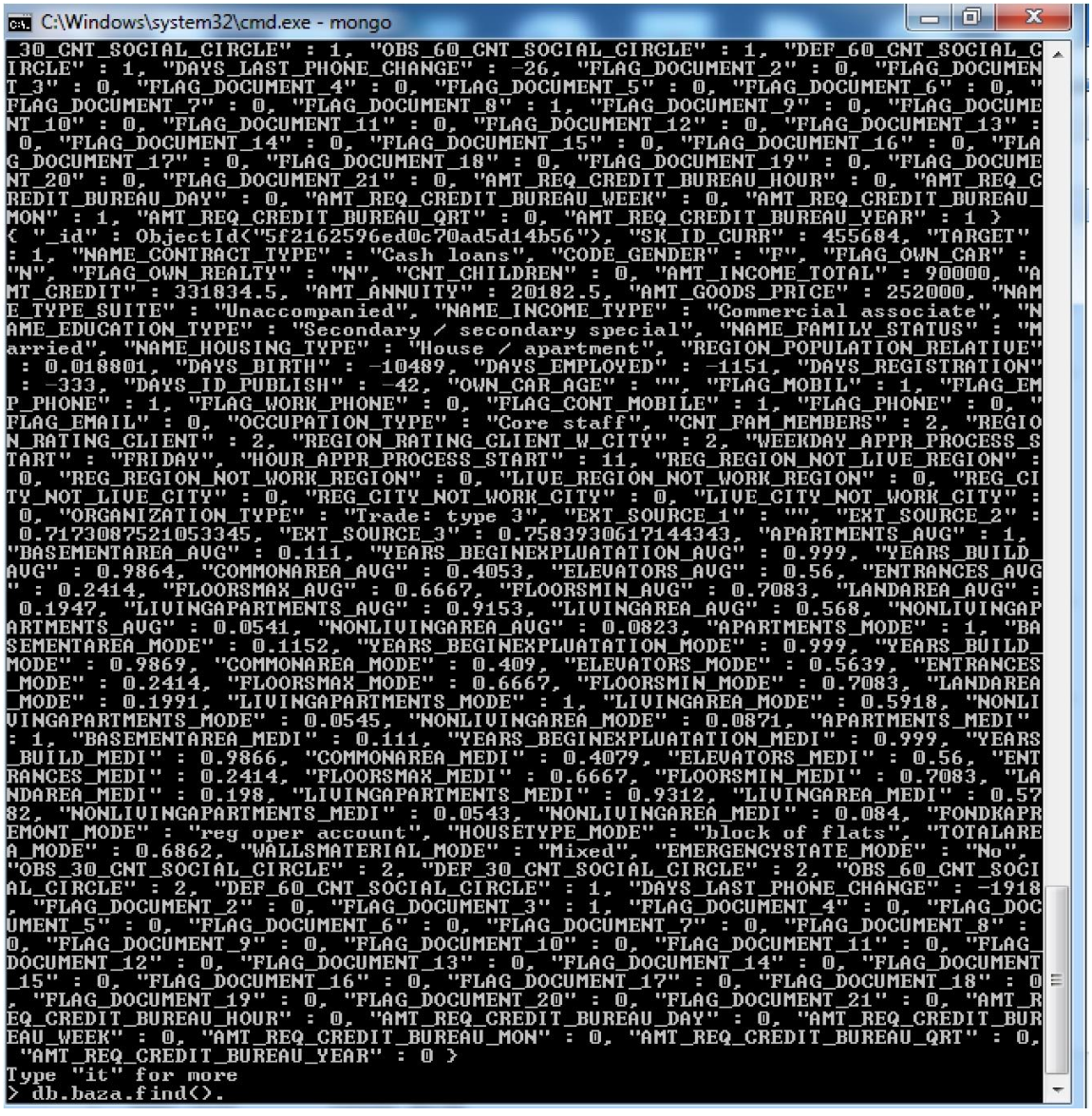
Негізгі мақсат csv форматтағы ғана деректер емес жалпы басқа форматтағы да деректерді өңдеуге мүмкіншілік туындап отыр. Жоғарыдағы бөлімдерде айтып өткендей әлеуметтік желі қолданушылары күнделікті өздерінің фото, видеоларын және хаттарын жүктеуі серверлердің жұмысына біршама жүктемені көбейтті. Құрылымданбаған үлкен өлшемді деректерді өңдеу мәселесінде түрлі пікірталастар бар. Зерттеушілердің пікірлері әртүрлі біреулері жаңа технологияларды қолдануды ұсынса, біреулері жүктемені бөліп өңдеулерді ұсынып отыр. Деректерді даярлауда, нормализациялап алып өңдеуде кейбір алгоритмдер толық дұрыс нәтиже бермейді. Диссертациялық жұмыстың міндетіне сай үш машиналық оқыту алгоритмі таңдалып отыр. Әдістердің өздері емес олардың нәтижелері салыстырылып, шешілеттің міндеттің форматына сай модификацияланды және ипотекалық несие алушы азаматтардың төлем қабілеттерін болжауда жоғарғы нәтиже көрсетті.

3.1.1 Деректерді даярлау

Деректер қорын бағдарламалық кешенге салып өңдеу үшін алдымен өзге форматтағы деректерді біріңғай дерекқорымыз MongoDB-ге трансформациялап аламыз.

C:\ mongo db\bin\mongo import- -db Dauren--collection application_test –type csv --file E: Base\application_test.csv – headerline

C:\ mongo db\bin\mongoimport –db Dauren-collection application_train –type csv—file E:Base\application_train.csv --headerline.



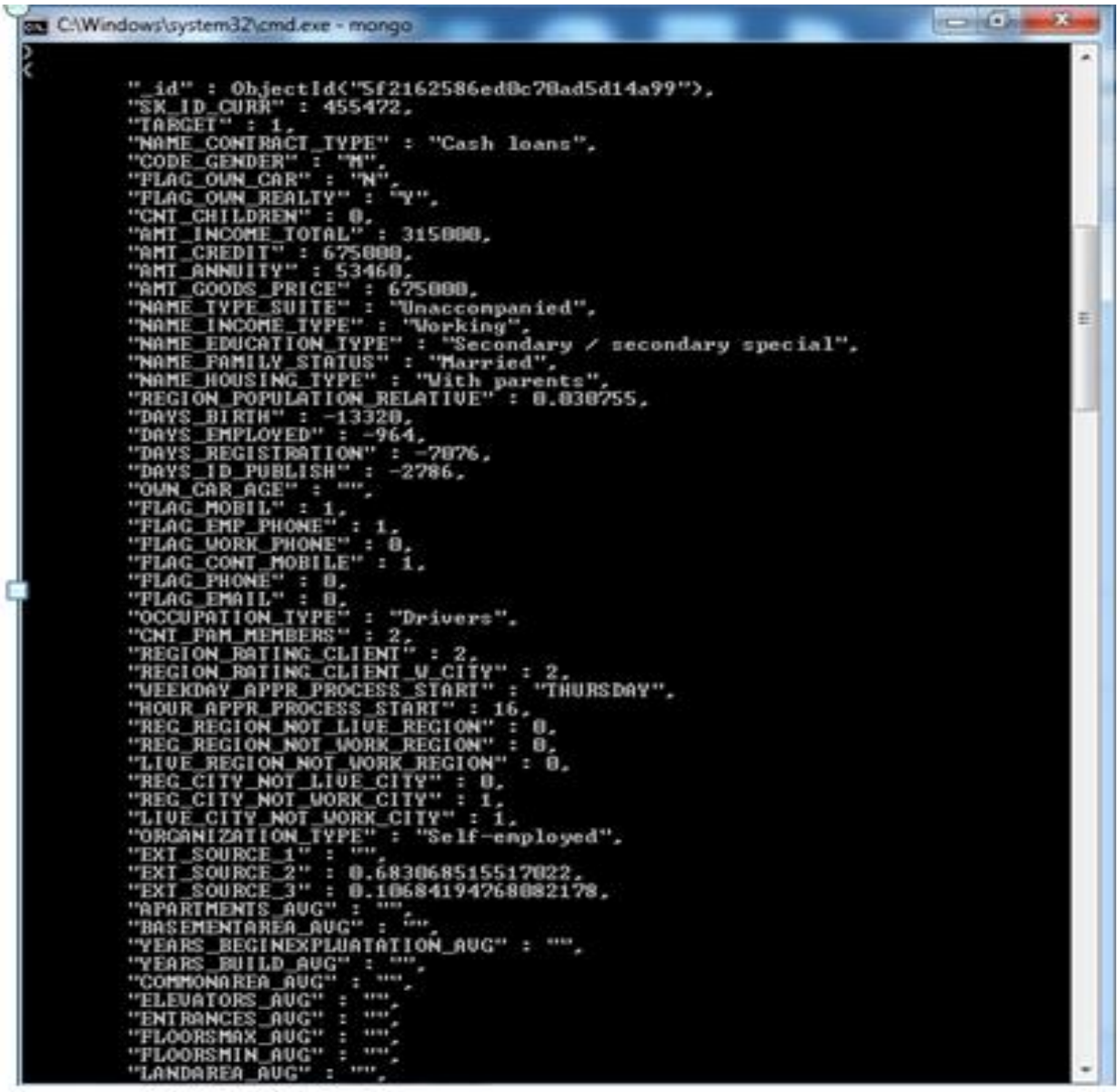
Сурет 3.1 – Деректердің MongoDB дерекқорына трансформациялануы

Mongo DB-дегі дерекқорымызды тексеру үшін:

C: mongodb\bin\mongodb

Show dbs

Use Dauren
Show collections
Db collections
Db.application_test немесе db.application_train



Сурет 3.2 – MongoDB дерекқорындағы деректер

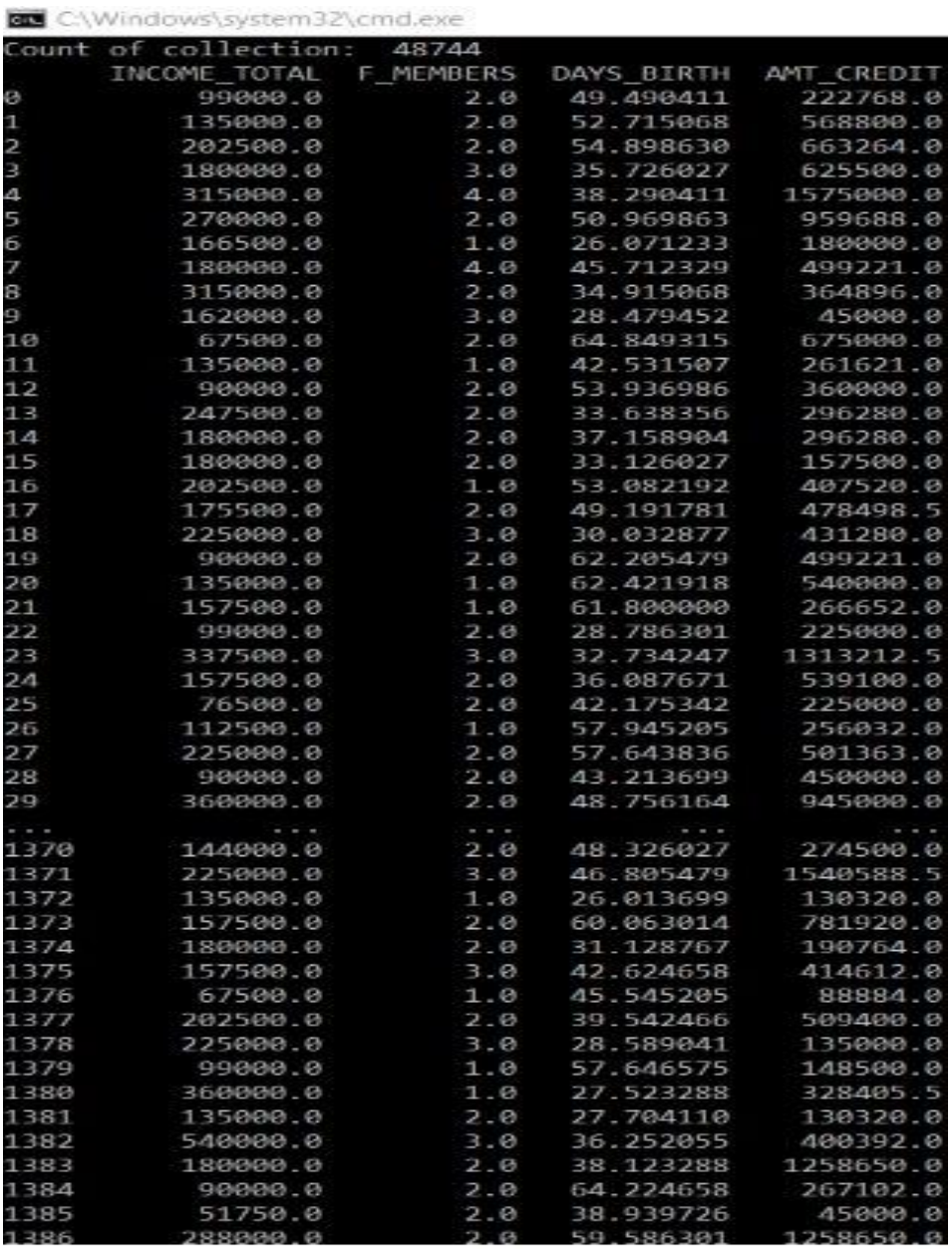
3.1.2 Деректерді нормализациялау

Нормализациялау - бұл барлық мәндер 0-ден 1-ге дейін болатындай етіп деректерді бастапқы ауқымынан қалпына келтіру[80].

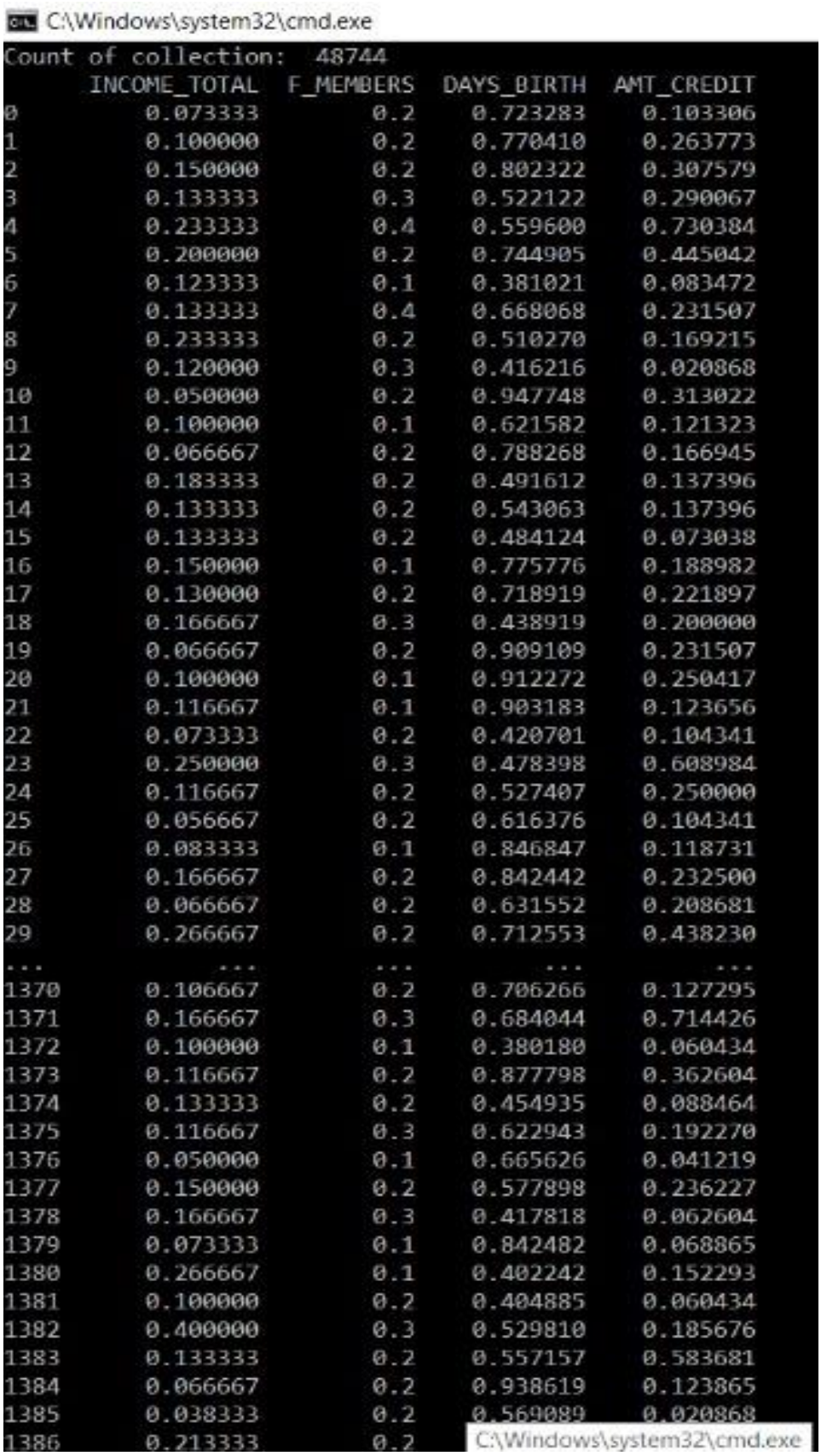
Нормализациялау машиналық оқытудың кейбір алгоритмдеріне пайдалы болуы мүмкін, себебі кейбір жағдайларда уақытша қатарлардың деректерінің кіріс мәндері көлемдері әртүрлі болады. Бұл k-Nearest көрші алгоритмдеріне қажет,

себебі ол ара қашықтықты есептеуде және сызықты регрессия, жасанды нейрондық желілердің салмақ мәндерін есептеуде қолданылады. Нормализация есептеуіміздегі минималды және максималды мәндерді нақты бағалауды қажет етеді. Егер уақытша қатар жоғарылау немесе төмендеу тенденциясына ие болса күтілетін мәндерді анықтау қиындық туғызады және бұл жағдайда нормализациялау мәселені шешудегі оңтайлы әдіс болып саналмайды. Мән келесідей нормализацияланады[81]:

$$y = (x - \min)/(\max - \min) \qquad (3.1)$$



Сурет 3.3 – Нормализацияланбаған деректер

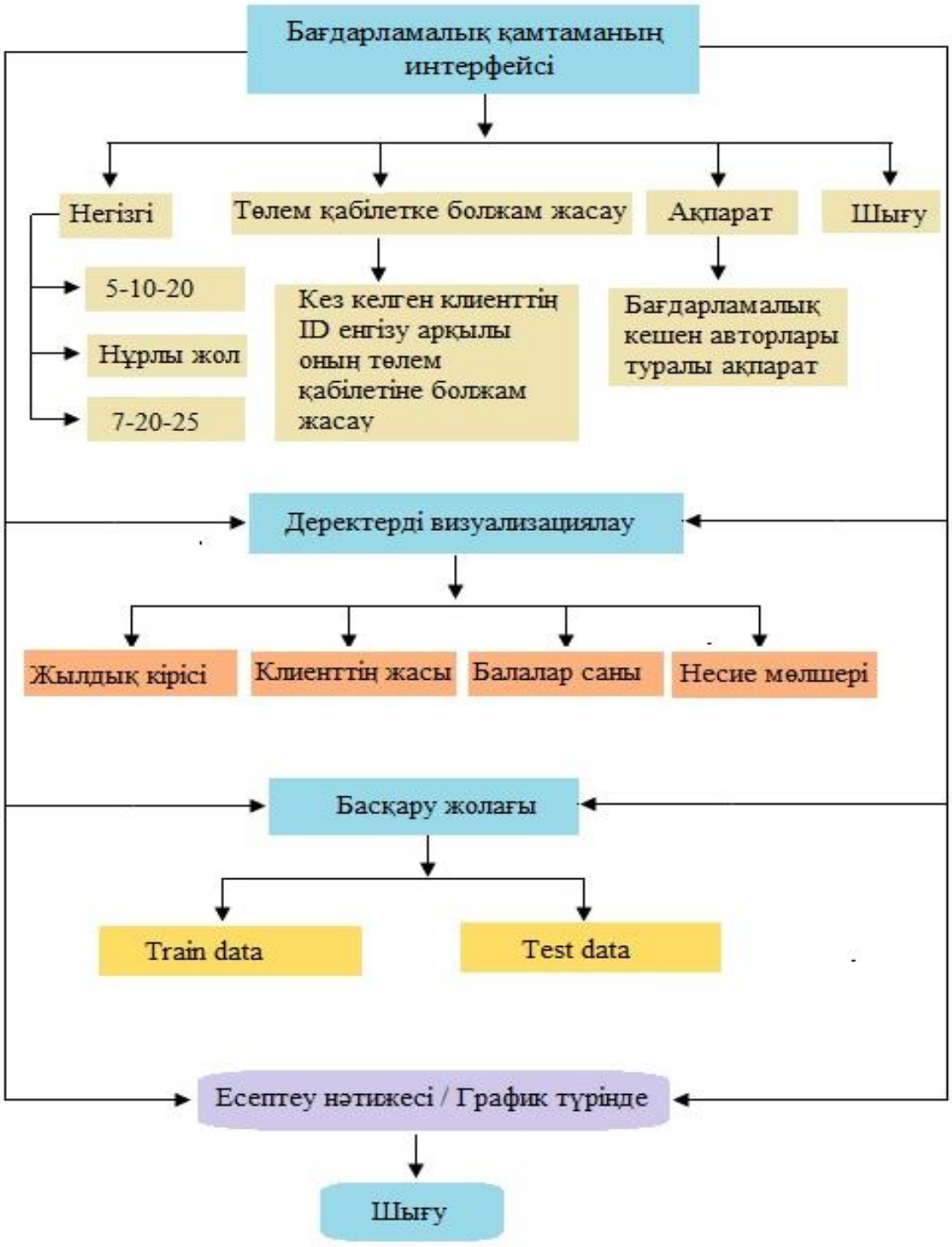


Сурет 3.4 – Нормализацияланған деректер

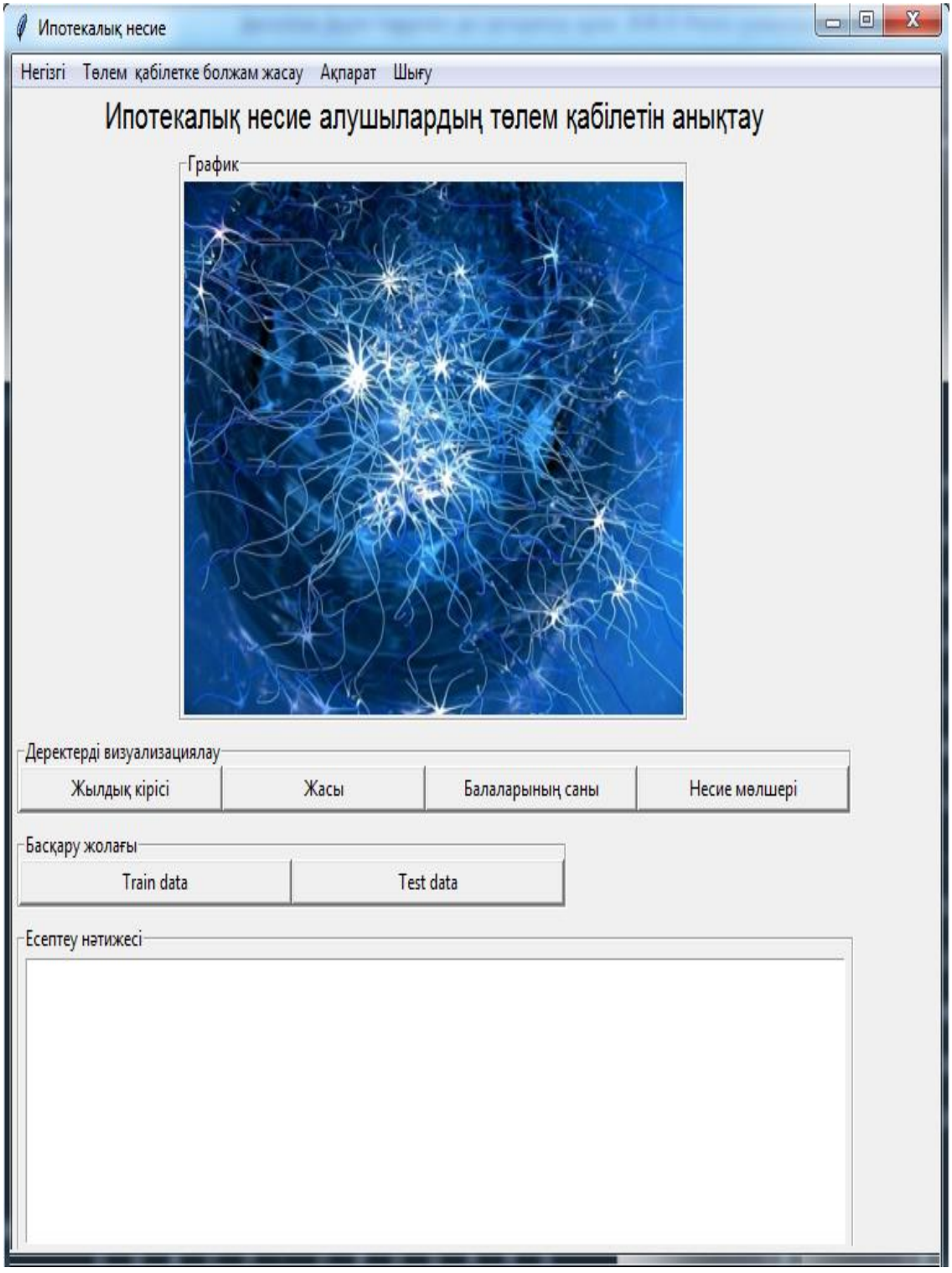
3.2 Бағдарламалық қамтаманың архитектурасы

Бағдарламалық қамтаманың жұмыс істеуінің сипаттамасы қарапайым. Төменде берілген 3.1 – суретте Python тілінің компьютерлік инженерия саласында жүрген ғалымдарға арналған өте қуатты Spider ортасында құрылған ұзақ мерзімге ипотекалық несие алушы тұлғалардың төлем қабілетін анықтау жүйесінің жұмыс істеу процесінің сызбасы берілді. Ол бойынша қолданушы алдымен ұзақ ипотекалық несие алушының төлем қабілетін анықтауға негізделген бағдарламаның негізгі 4 мәзіріне өтеді. Олар: «**Негізгі»**, «**Төлем қабілетке болжам жасау»**, «**Ақпарат»**, «**Шығу»** деп аталады. Негізгі мәзірінде мемлекеттік тұрғын үй бағдарламалары туралы ақпарат берілген. Төлем қабілетке болжам жасау батырмасын басып ипотекалық несие алушы азаматтардың ID номерін енгізіп төлем қабілетін болжауға болады. Ақпарат батырмасын басу арқылы бағдарламалық кешеннің авторлары туралы толық ақпарат алуға болады. Сонымен қатар, бағдарламалық қамтамадан шығуға да болады. Төлем қабілетті есептеуде диссертациялық жұмыстың нәтижелерін беретін үш әдіс берілген. Олар DataMining әдістері бойынша, яғни сызықты регрессия, логистикалық регрессия және нейрондық желілерді[82] қолданып, ұзақ мерзімге ипотекалық несие алушы жеке тұлғалардың төлем қабілеттерін анықтап беретін есептеулер. Бағдарламалық кешен шынайы деректерді ғана өңдеуге негізделген. Себебі құрылатын бірыңғай MongoDB деректер қорында азаматтардың толық деректері сақталынатын болады және қауіпсіздік мақсатында оларға ID номер беріледі. Егер өтінім берушілер туралы бірыңғай дерекқордан табылмаса жүйе автоматты түрде өтінімді күмәнді деректер қатарына қосып өшіріп тастайды. Себебі, өтінім беруші өзі туралы шынайы деректерді бермеуі мүмкін немесе басқа мемлекеттің азаматы болуы да мүмкін. Бағдарламалық қамтама тек біріңғай дерекқорда деректері бар Қазақстан Республикасының азаматтарына ғана арналған. Сонымен қатар, уақыт пен ипотекалық несие мерзімін дұрыс көрсетпесе қате хабарлама беріледі. Бағдарламалық қамтаманың интерфейсінде деректерді визуализациялайтын «**Жылдық кірісі», «Клиенттің жасы», «Балаларының саны», «Несие мөлшері»** сияқты батырмалары бар. Сонымен қатар, басқару жолағында деректерді «Train data», «Test data» бөліп экранға график түрінде нәтижелерді шығаратын батырмаларда бар.

Data Mining әдістері қолданылып есептелген нәтижелер бойынша қателіктерде бағдарламалық қамтамада қамтылған. Нәтижелер график түрінде де экранға шығады және қай алгоритм үлкен өлшемді құрылымданбаған деректерді өңдеуде жоғарғы дәлдікті көрсеткені анық беріледі.



Сурет 3.5 – Бағдарламалық қамтаманың архитектурасы



Сурет 3.6 – Бағдарламалық қамтаманың интерфейсі

3.3 Деректерді өңдеу алгоритмдерінің жүзеге асырылуы және өңдеу моделінің құрылуы

1.Сызықты регрессияны пайдаланып құрылған алгоритм.

1. x_{ij}-кіріс деректері; $\overline{y_i}$ -орташа шығыс мән; ε, N;
2. for epoch 0 do N epoch
3. $\overline{y}$ -шығыс мәнді есептеу; (3.2)
4. b_i -регрессия коэффициентін табу; (регрессия сызығының иілу көрсеткіші) (3.3)
5. a_i -бос мүшені табу (3.4)
6. шығын функциясын есептеу; (3.5)
7. if $(E_i \prec \varepsilon)$ онда 2 қадамға бару немесе есептеуді тоқтату;

$$\overline{y}_i = a_i + b_i x_{ij} \tag{3.2}$$

$$b_i = \frac{\sum\left(\left(x_i - \overline{x}\right)\cdot\left(y_i - \overline{y}\right)\right)}{\sum\left(x_i - \overline{x}\right)^2} \tag{3.3}$$

$$a_i = \overline{y}_i - b_i \cdot \overline{x} \tag{3.4}$$

$$E_i = \frac{1}{n}\sum_{i=1}^{n}\left(\overline{y_i} - y_i\right)^2 \tag{3.5}$$

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.6}$$

2**.** Логистикалық регрессияны пайдаланып құрылған алгоритм.

1. x_{ij}-кіріс деректері; θ_j-бастапқы салмақ коэффициенті; ε , N;
2. for epoch 0 do N epoch
3. $\overline{z_i}$ - шығыс мәнді есептеу; (3.7)
4. θ_j - салмақ коэффициентін есептеу; (3.8)
5. b_i -бос мүшені есептеу (3.9)
6. шығын функциясын есептеу; (3.10)
7. if $(E_i < \varepsilon)$ онда 2 қадамға бару немесе есептеуді тоқтату;

$$\overline{z_i} = \sum_{i=1}^{n}\theta_j x_{ij} + b_i \tag{3.7}$$

$$\theta_j = \theta_j - \alpha\frac{\partial E_i}{\partial \theta_j} \tag{3.8}$$

$$b_i = b_i - \alpha\frac{\partial E_i}{\partial b_i} \tag{3.9}$$

$$E_i = \frac{1}{n}\sum_{i=1}^{n}\left(\overline{z_i} - z_i\right)^2 \tag{3.10}$$

$$\frac{\partial E_i}{\partial \theta_j} = \left(\overline{z_i} - z_i\right)x_i \tag{3.11}$$

$$E_i\left(\overline{z_i} - z_i\right) \tag{3.12}$$

$$\frac{\partial E_i}{\partial b_i} = \left(\overline{z_i} - z_i\right) \tag{3.13}$$

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.14}$$

3. Көпқабатты нейрондық желіні пайдаланып құрылған алгоритм.

1. x_{ij}-кіріс деректері; ω_j-бастапқы салмақ коэффициенті; ε, N ;
2. for epoch 0 do N epoch
3. $\overline{y_i}$-шығысты есептеу ; (3.15)
4. салмақ коэффициенттерін есептеу; (3.16)
5. шығын функциясын есептеу; (3.17)
6. if $(E_i \succ \varepsilon)$ онда 2 қадамға бару немесе есептеуді тоқтату;

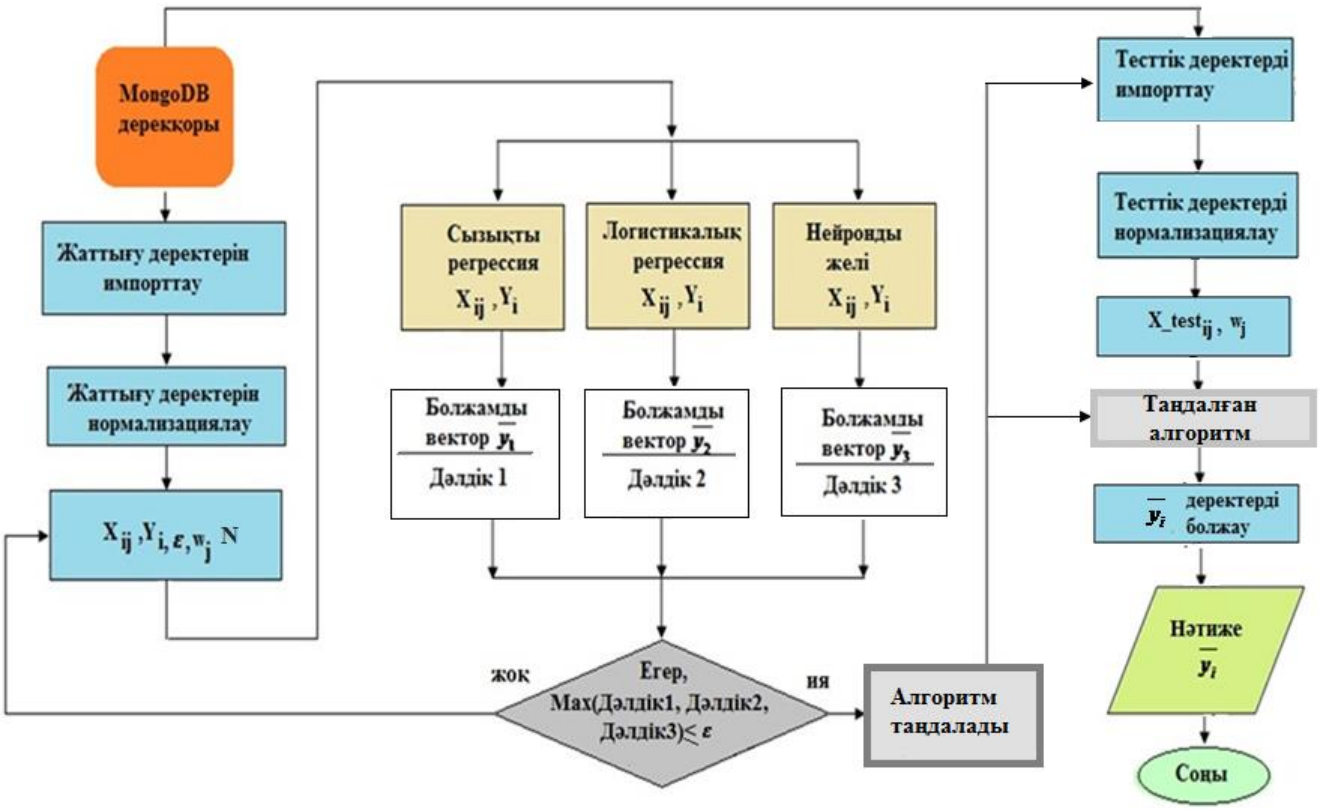
$$\overline{y_i} = f\left(\sum_{j}^{n} x_{ij}\cdot\omega_j + b_i\right) \tag{3.15}$$

$$\Delta\omega_{ij} = -\mu\frac{\partial E_i}{\partial \omega_{ij}} \tag{3.16}$$

$$E_i = \frac{1}{2}\sum_{i=1}^{n}\left(\overline{y_i} - y_i\right)^2 \tag{3.17}$$

$$\frac{\partial E_i}{\partial \omega_{ij}} = \left(\overline{y} - y_i\right)x_i \tag{3.18}$$

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.19}$$



Сурет 3.7 – Құрылған жүйенің моделі

Компьютерлік модельде MongoDB дерекқоры құрылымданбаған үлкен өлшемді деректерді сақтау үшін қолданылып отыр. Дерекқордың ерекшелігі түрлі бағдарламалау тілдерімен жеңіл интеграцияланады, яғни деректерді трансформациялауға, импорттауға, нормализациялауға ыңғайлы. Бұл жердегі X_{ij}-матрица түріндегі деректер, Y_i-бізге белгілі деректерді жаттықтыруға арналған (train)мән. ε-қолдан енгізілетін бастапқы қателік, ω_j-салмақ коэффициенті, N-эпоха. Машиналық оқытудың үш алгоритмімен 100 эпоха беріп, және қателікті қолдан енгізу арқылы күткен нәтижеге жетеміз. Қолданылған үш алгоритм дәлдіктері салыстырылып, жоғарғы нәтиже көрсеткен алгоритм таңдалады. Әрі қарай MongoDB дерекқорымыздан тесттік деректерді өңдеу үдерісі жоғарғы нәтижелі дәлдік көрсеткен алгоритммен өңделеді. Моделді (сандық моделді) құруымыздағы басты міндеттің бірі – ол жеке тұлғалардың құрылымданбаған үлкен өлшемді деректерін өңдеу арқылы, олардың төлем қабілеттерін бағалау. Егер, MongoDB дерекқорын ипотекалық несие беруші ұйымдар үшін біріңғай дерекқор етіп таңдап алсақ, көптеген мәселелер оңтайлы шешілетіні анық. Клиенттердің деректері бірден дерекқордан алынып өңделсе, пәтер үлестірудегі түйткілді мәселелер оң шешімін табар еді.

3.4 Ипотекалық несие алушылардың төлем қабілеттерін анықтауда DataMining әдістерін қолданып эксперимент жүргізу.

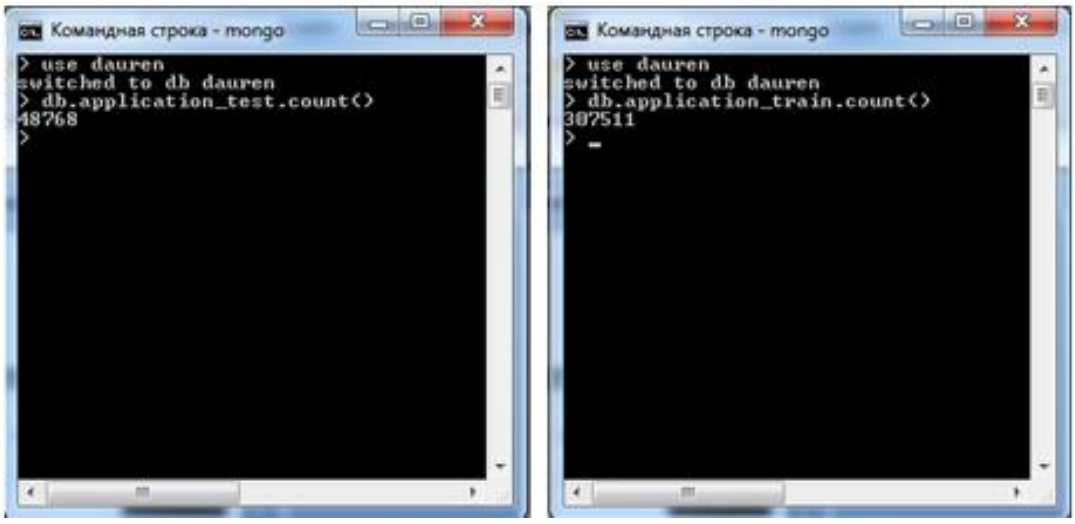
Отандық ғылыми кеңесшім физика-математика ғылымдарының докторы, профессор Г.Т.Балакаевамен бірге ипотекалық несие алушылардың төлем қабілетін анықтау негізінде үлкен көлемді деректерді өңдеуде DataMinig әдістерін қолданып эксперименттер топтамасын жүргіздік. Эксперименттер нейрондық желілерді оқыту үдерісін бағалау үшін жүргізілді.

Ипотекалық несие алушылардың төлем қабілетін анықтау үшін 3.8-суретте MongoDB дерекқорымыздан алынған статистикалық деректер берілген. Төлем қабілетті анықтауда жүйе моделінің жұмысын тексеру үшін барлығы 356 256 адамның оқу тестілік деректері қолданылды. Олардың ішінде 48744 адамның дерегі тесттік жаттығу жиынын құрады. Деректерді нақты болжауымыз үшін 100% деректің 25% тестке, қалған 75% жаттығуға бөлінуі қажет. Жалпы бағдарламалық қамтамамен 356 279 жеке тұлғалардың 1425116 жазба деректері өңделді.

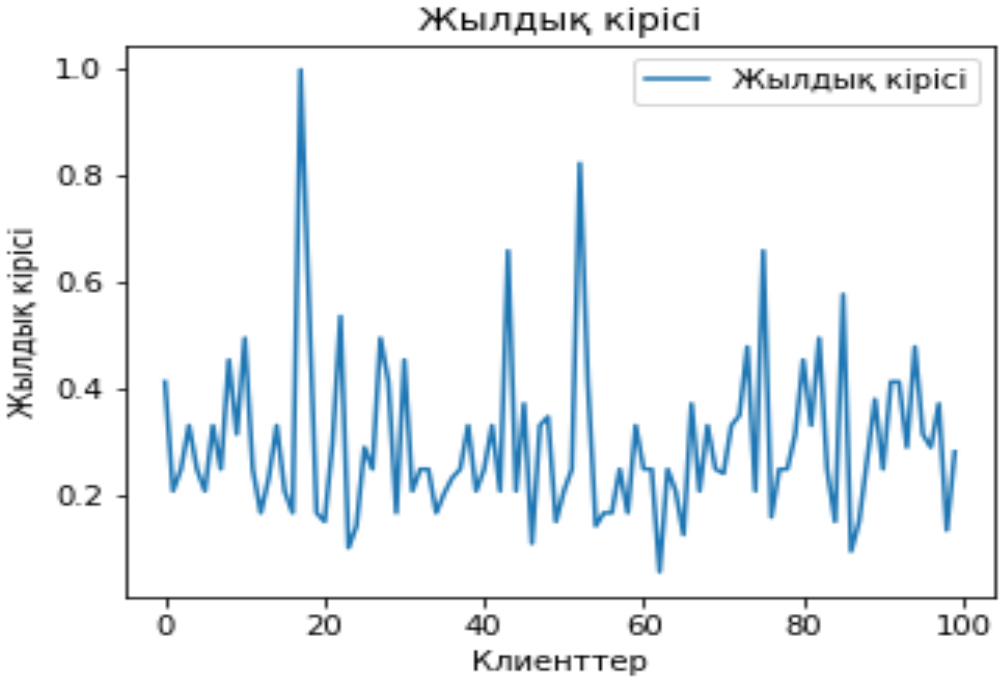
Жеке тұлғалардың төлем қабілеттерін анықтауда жүйені модельдеуге қолданылған жиыны төменде кестеде көрсетілді.

Кесте 3.1 – Эксперимент үшін қолданылған деректер жиыны

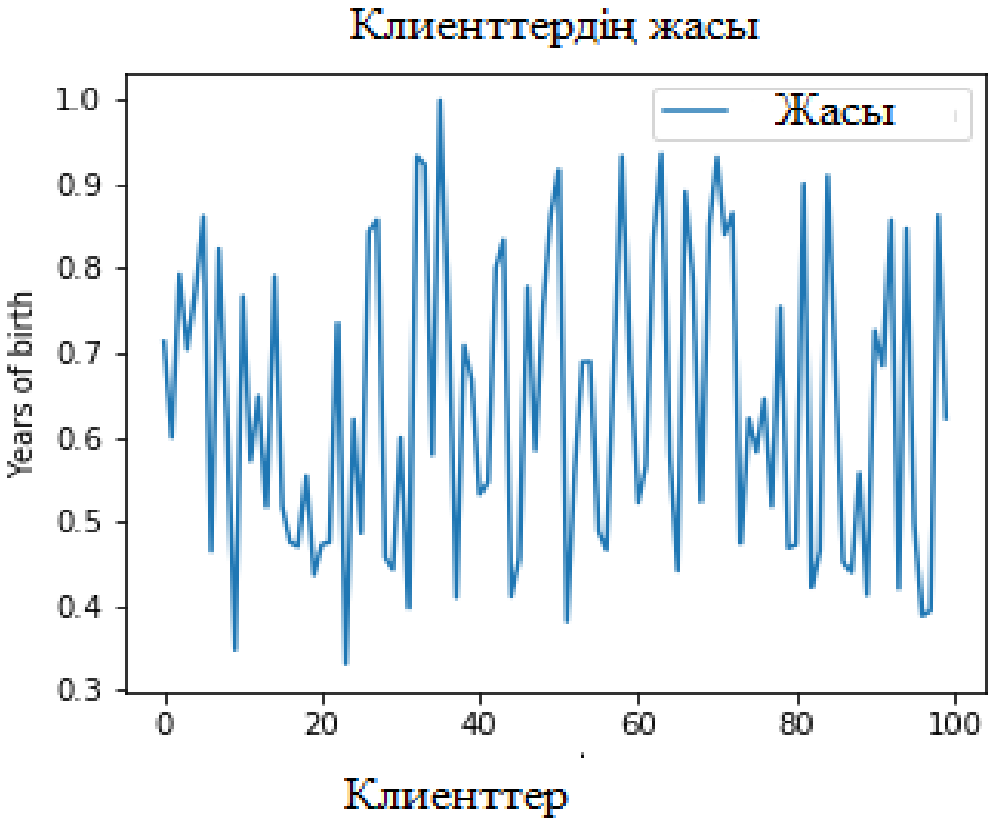
Деректер жиыны (Data sets)	Жалпы саны	Барлығы (Total)
Оқыту (Train)	307 511	356 279
Тестілеу (Test)	48 768	



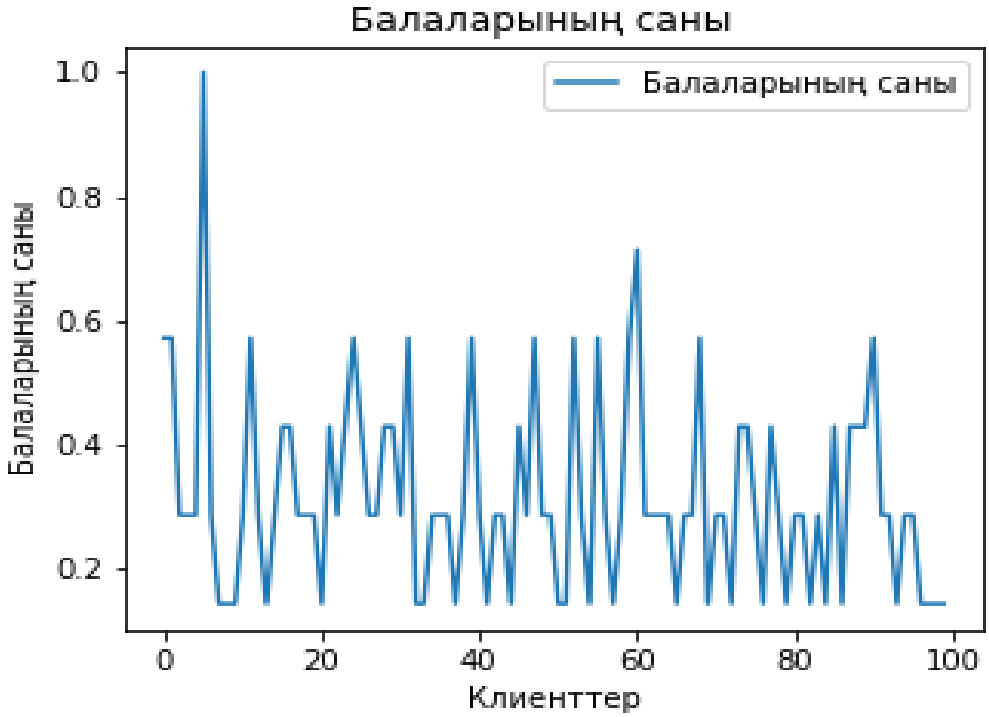
Сурет 3.8 – MongoDB деректер қорындағы деректер



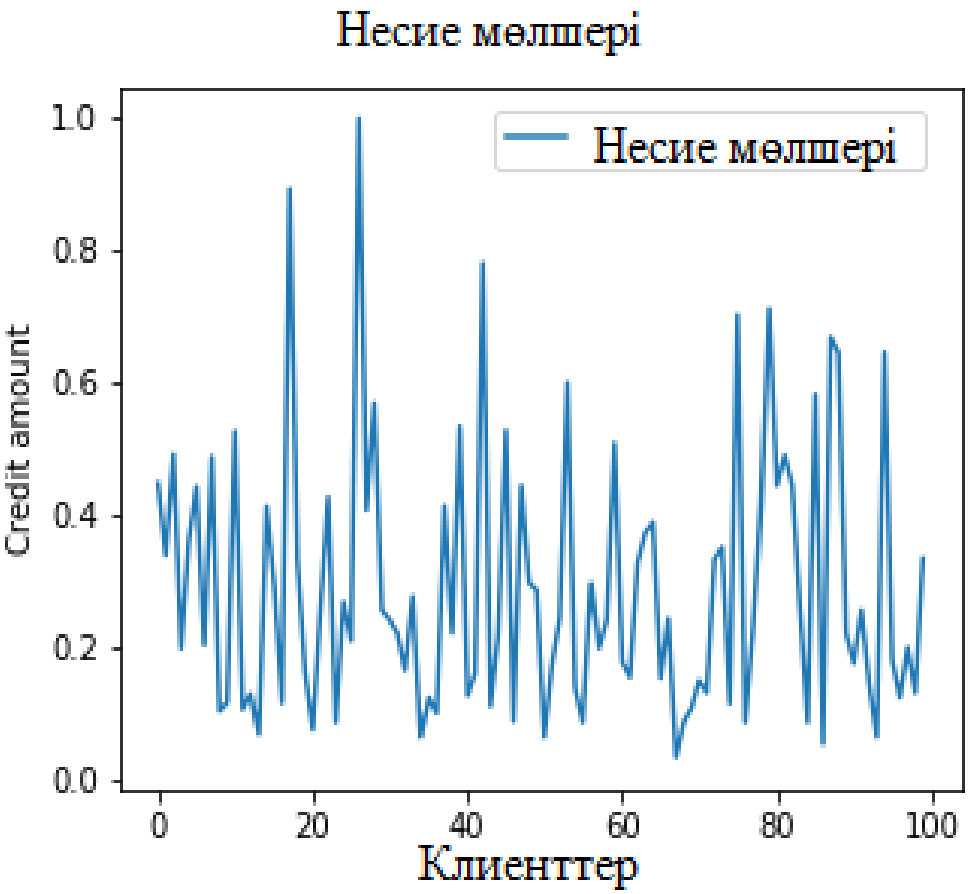
Сурет 3.9 – Клиенттердің жылдық кірісі



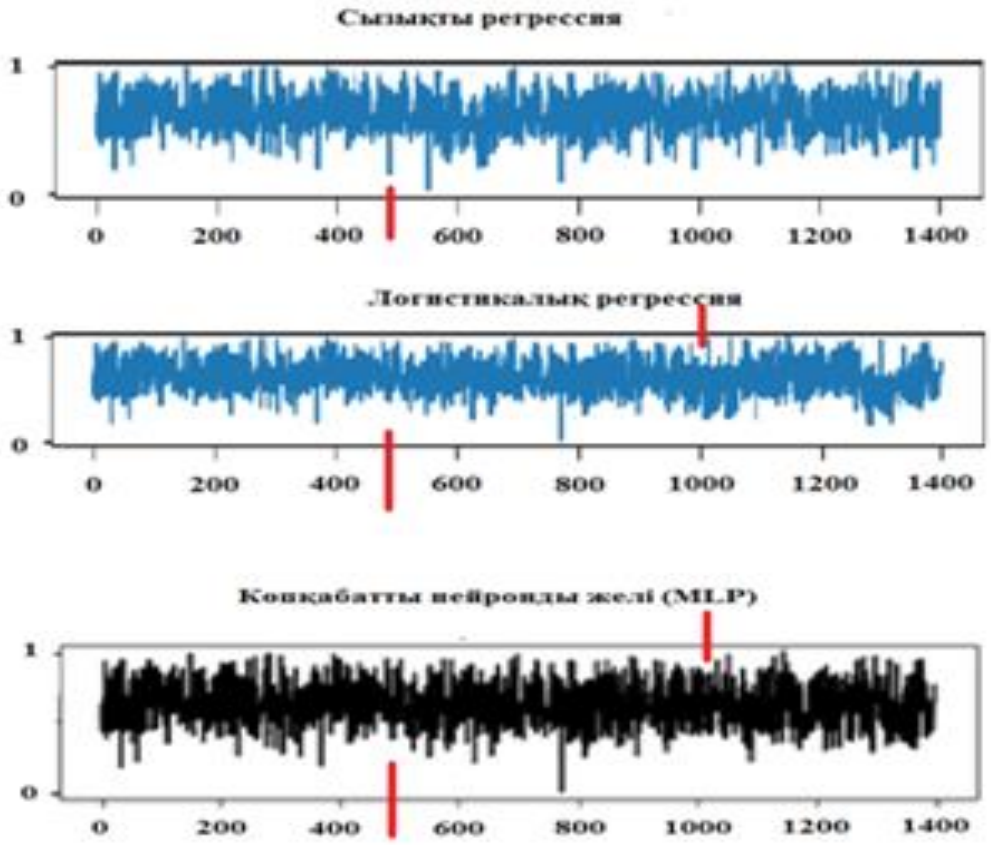
Сурет 3.10 – Клиенттердің жасы



Сурет 3.11 – Балаларының саны

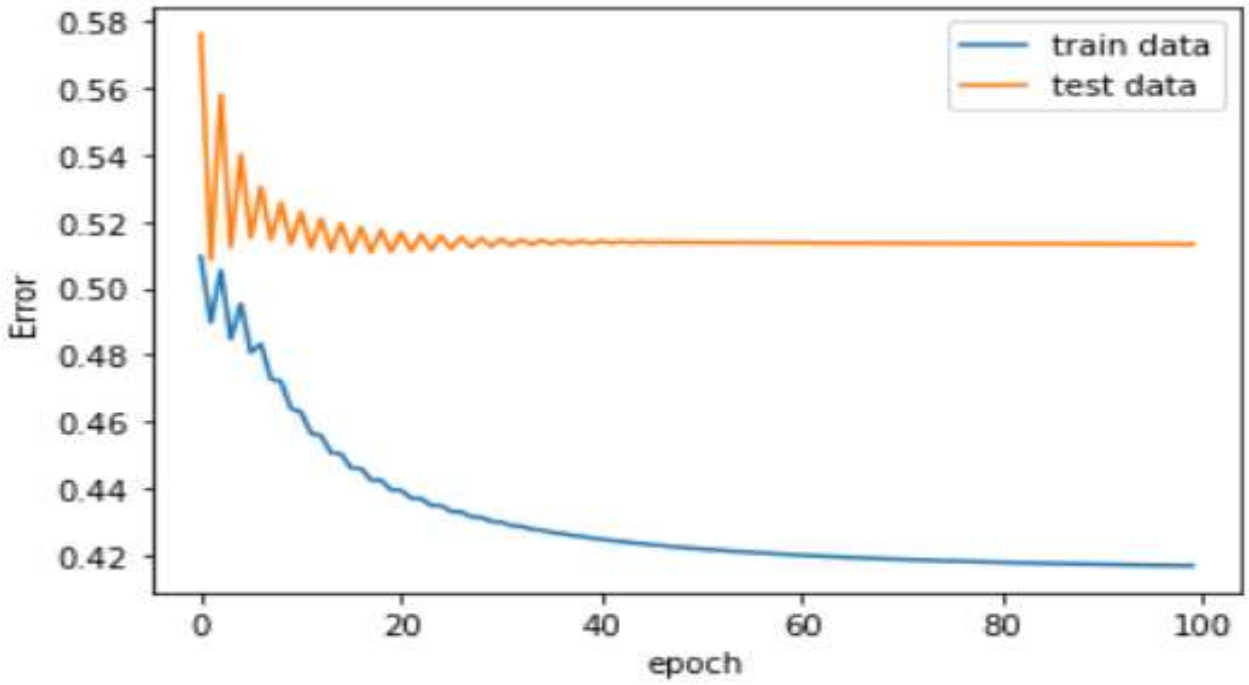


Сурет 3.12 – Несие мөлшері

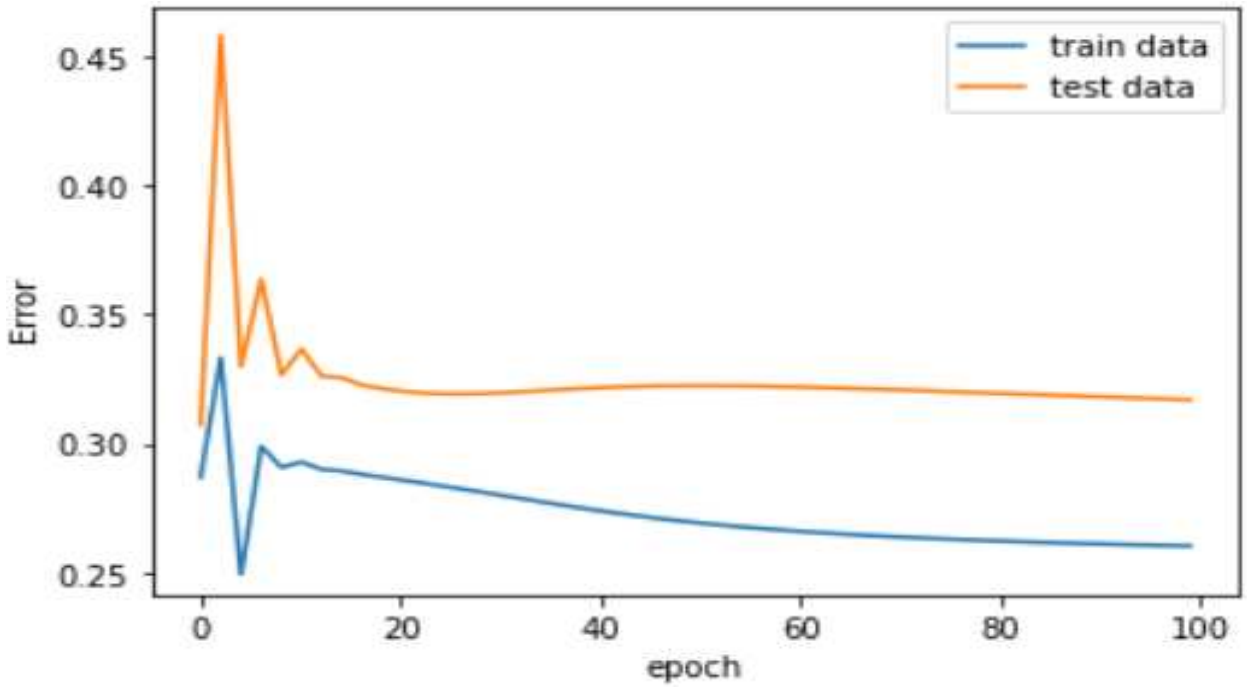


Сурет 3.13 – Жаңа деректерді болжау кезінен үзінді

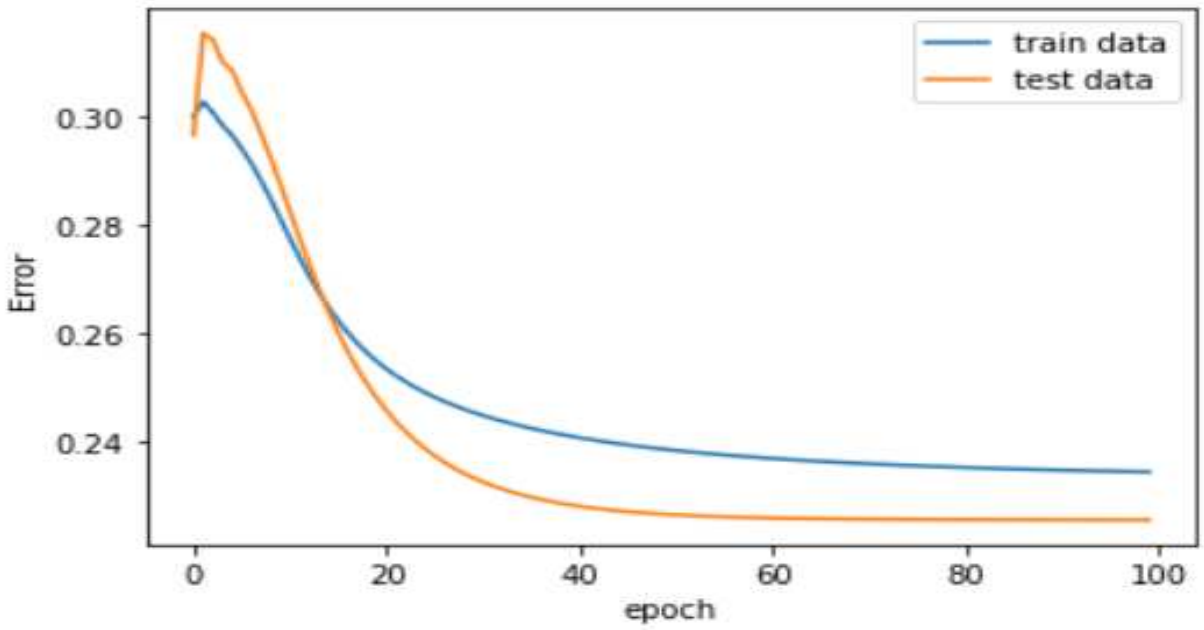
3.13 – суретте, алгоритмдерді тестілеу нәтижелері көрсетілген. Графикте, қызыл сызықпен өңдеу бойынша алгоритмдердің нәтижелері сызылып көрсетілген. Сызықты регрессия аз көлемді деректерді логистикалық регрессия орта көлемдегі деректерді, яғни сызықты регрессиядан қарағанда 2 есе көп деректерді, ал көп қабатты нейронды желі үлкен өлшемді деректерді өңдегенде жоғарғы нәтиже көрсетіп отыр. 3.13 – суретте байқағанымыздай, таңдалған үш алгоритмнің графиктеріндегі өзгешелікті, яғни деректер санының өсуіне байланысты өзгергенін байқауға болады. Бұл өз кезегінде көпқабатты нейронды желіні қолдану үлкен өлшемді деректерді өңдеудегі нәтиженің жоғары болатынына тағыда дәлел. Төменде үш алгоритмнің деректерді өңдеу бойынша нәтижелерінің қателіктері графикпен беріліп отыр.



Сурет 3.14 – Сызықты регрессия көрсеткен қателік



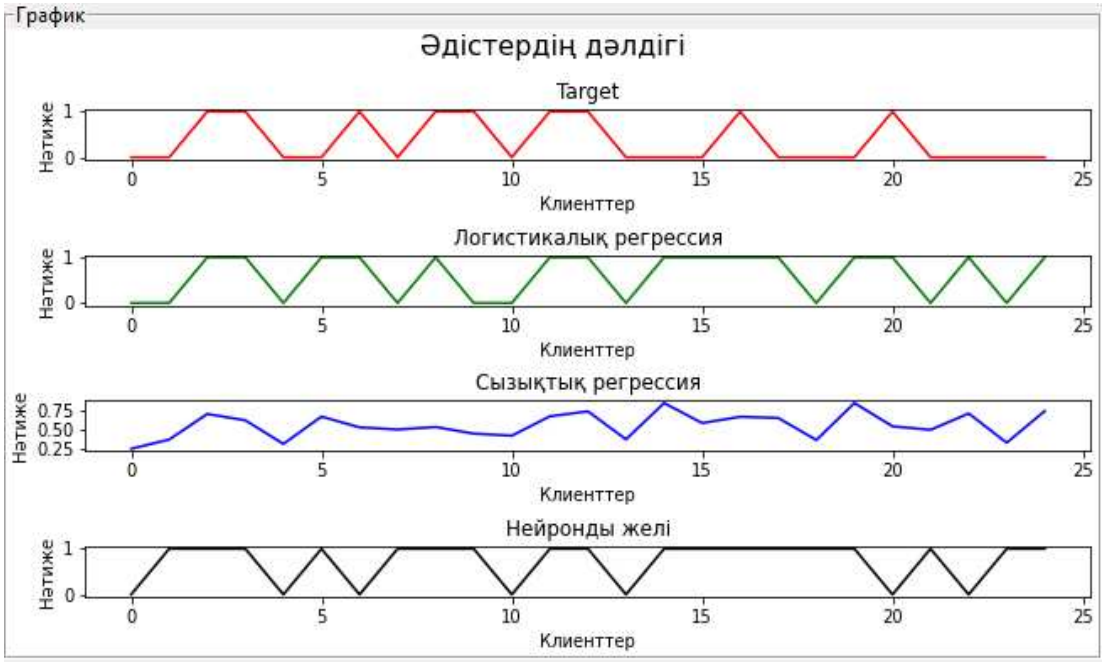
Сурет 3.15 – Логистикалық регрессия көрсеткен қателік



Сурет 3.16 – Көпқабатты нейронды желі көрсеткен қателік

3.5.Қолданылған әдістердің нәтижелерін салыстыру

Диссертациялық жұмыста DataMining әдістерін қолданып, қайсы әдіс үлкен өлшемді құрылымданбаған деректерді өңдеу негізінде ипотекалық несие алушы тұлғалардың төлем қабілетін анықтауда тиімді әрі нақтылық жағынан жоғарғы пайызды көрсеткенін анықтадық.



Сурет 3.17 – Әдістердің дәлдігі

Сызықты регрессияны қолданудағы басымдылықты көрсететін факторлар төмендегідей:

- сызықты регрессия үшін деректер өте тиімді қолданылады.
- жақсы нәтижелер көлемі аз деректерді өңдеуде шығады.
- сызықты регрессия теориясы түсінікті және қарапайым.
- сызықты регрессия әдісі ипотекалық несие алушы азаматтардың төлем қабілетін анықтау жүйесін моделдеуде көп жұмысын жеңілдетті.

Сызықты регрессияның негізгі кемшіл тұстары:

- шығыс параметрлері, тәуелсіз айнымалылар [0,1] аралығынан асып кетуі.
- алынған нәтижелерде белгісіз параметрлердің мәндері жиі кездеседі.

Логистикалық регрессия әдісін қолданып алынған нәтижелердің басымдылықтары:

- орта көлемдегі деректерге болжамның нақты жасалуы, яғни өңделетін дерек көлемі сызықты регрессия қолданылып өңделетін деректен 2 есе көп деп айтуға болады.

Көпқабатты нейрондық желілерді қолдану әдісі диссертациялық жұмыста қолданылған үлкен өлшемді құрылымданбаған деректерді өңдеуде өте жақсы нәтиже көрсетті. Шешіліп отырған есеп үшін нейрондарды оқыту және есептеу жылдамдығы өте жақсы нәтиже көрсетіп отыр. Әдістерді салыстыру төменде 10-кестеде көрсетілген.

Кесте 3.2 – Әдіс нәтижелерін салыстыру кестесі

Әдіс/Method	Деректер көлемі	Дәлдік (%)	Қателік/ Error (+/-)
Сызықты регрессия/ Linear regression	$n \approx 1.43 \cdot 10^6$	54,62	45,38
Логистикалық регрессия/ Logistic Regression	$n \approx 1.43 \cdot 10^6$	68,3	31,7
Нейрондық желілер/ Neural networks	$n \approx 1.43 \cdot 10^6$	78,44	21,56

3.2 – кестеде көрсетілген нәтижелердің салыстырмасынан көрсеткендей DataMining әдістері өте жақсы нәтиже берді. Сызықты регрессияның деректердің көлемінің аз болғанында нақты нәтиже беретінін, ал логистикалық регрессия орта көлемдегі деректерде қолдану жақсы нәтиже беретінін көрдік.

Нейрондық желілерді қолдану үлкен өлшемді $n \approx 1.43 \cdot 10^6$ болатын деректер жазбасын өңдеуде жоғарғы нәтиже көрсетті. 3.2 – кестеде кестеде құрылған модельдің есептеу нәтижелері берілген.

3.6 Деректерді өңдеу нәтижелері.

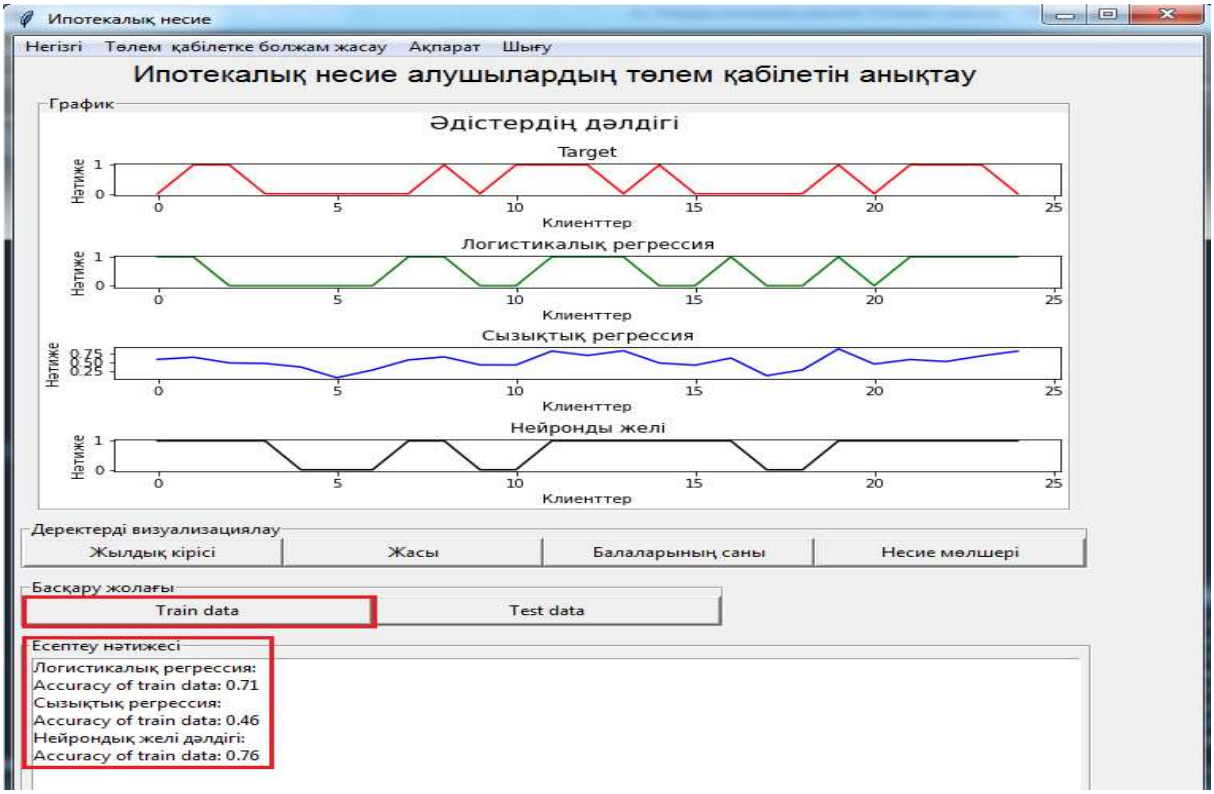
Деректерді өңдеу кезінде құрылымданбаған деректер міндетті түрде кездеседі.

Құрылымданбаған деректер дегеніміз – форматы әртүрлі деректер. Олар өңдеу барысында қиындықтар тудыратыны анық. Әлеуметтік желілер қолданушылары да күнделікті деректерді желігі жүктеуі серверлердің жүктемесін қиындатады. Диссертациялық жұмыста да кездесетін деректер көлемі үлкен және құрылымданбаған. Деректерді трансформациялау, нормализациялау, жүйелерді интеграциялау жұмыстары жүргізілді.

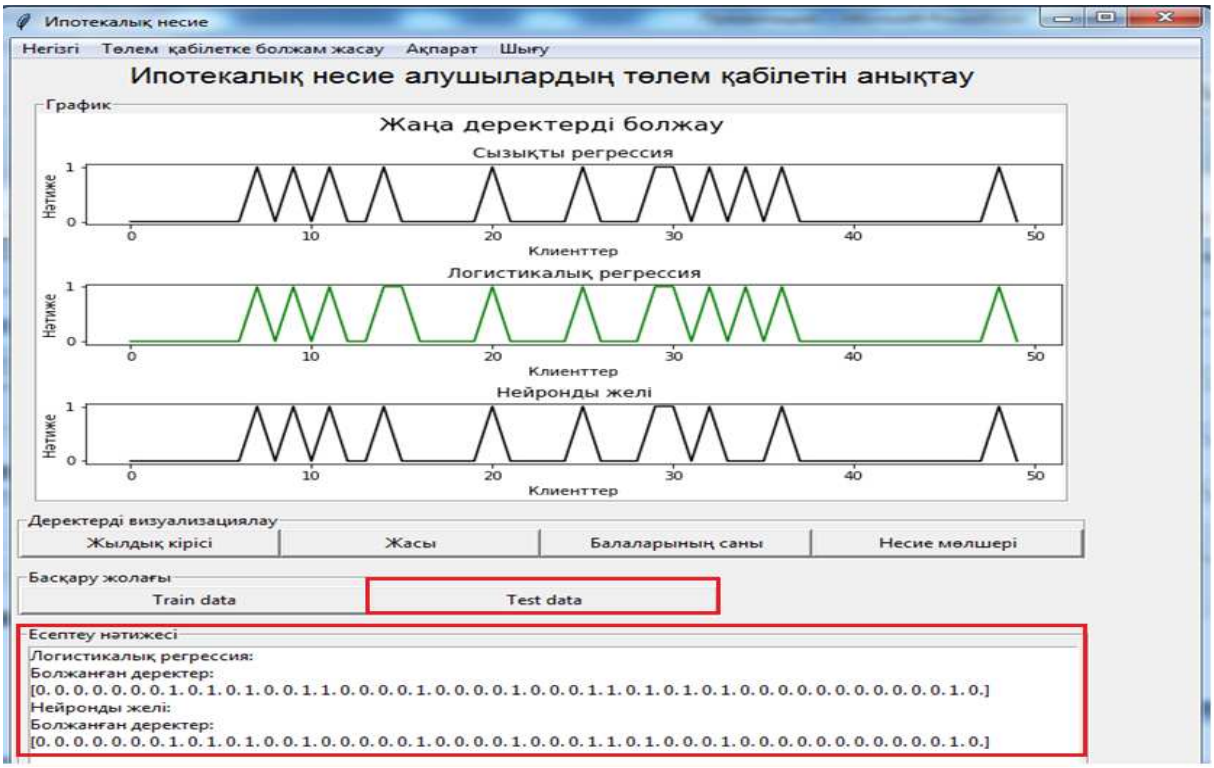
MongoDB - форматы әртүрлі яғни құрылымданбаған деректерді сақтауда тиімділігін көрсетті, құрылған өңдеу жүйесімен жеңіл интеграцияланды. Дерекқордағы жүйені моделдеп тексеруіміз үшін қолданылған шынайы деректеріміз 356 279 адамды құрады. Олардың сипаттамалырының саны 200-ге жуық. Атрибут санын көбейту арқылы қажетімізге қарай қолдануымызға болады. Диссертациялық жұмыста, бізге қажетті атрибуттар алып өңделді. Бізге ең қажетті деректер олар:

- Клиенттің жылдық кірісі;
- Балаларының саны;
- Алынатын несие мөлшері;
- Клиенттің жас мөлшері;

Осы төрт атрибут бойынша 356279 адамның 1425116 жазба дерегі оқытылып, өңделді. Клиенттің төлем қабілетін дәл анықтау үшін бірінші деректерді жаттықтырып алып “train” батырмасын басамыз, одан кейін деректер жаттыққаны бойынша дәлдікке көзімізді жеткізген соң “test” бастырмасын басып өңдеу нәтижелерін көреміз. Егер ипотекалық несие алушы жеке тұлғалар осы бағдарламаны қолданатын ипотекалық несие беру ұйымының менеджеріне келсе, менеджер қалаған атрибутты таңдап жеке тұлғаның төлем қабілетін болжап, қажетінше шешім қабылдай алады.

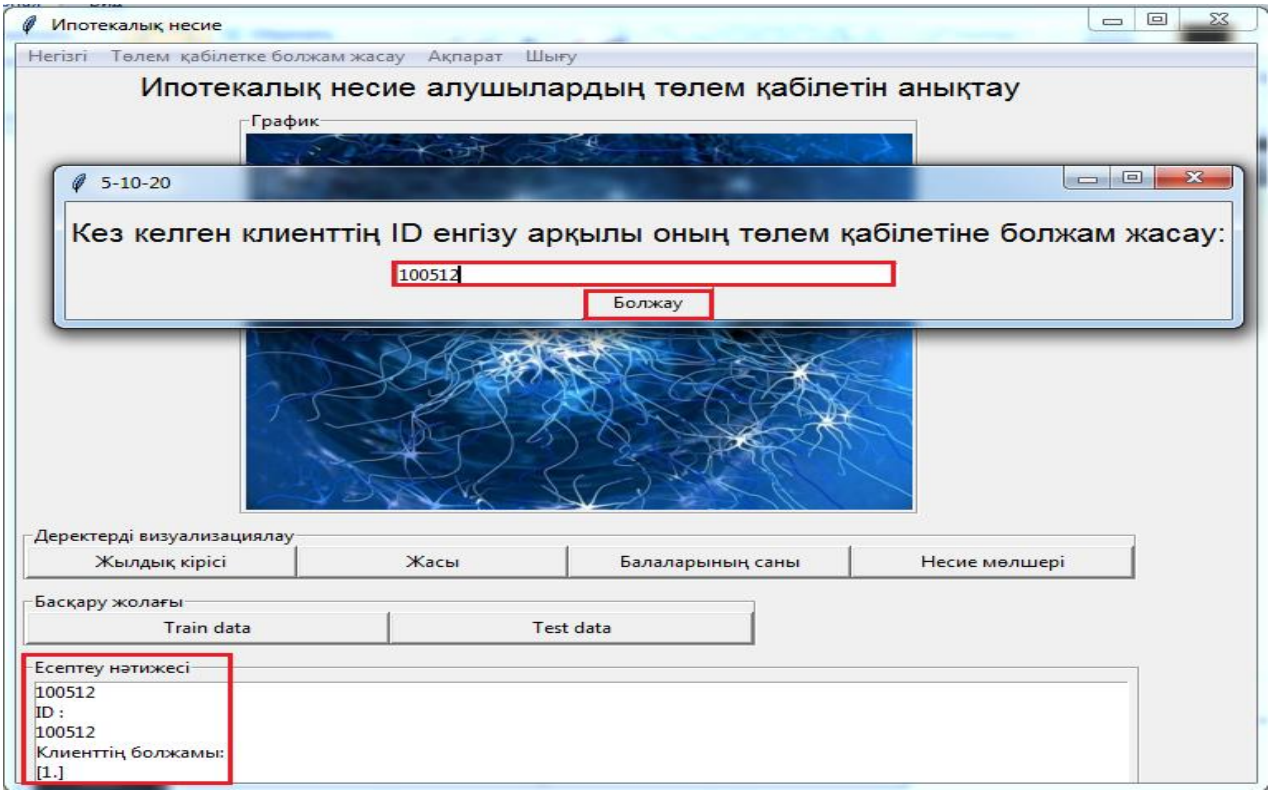


Сурет 3.18 – Деректерді жаттықтыру үдерісі

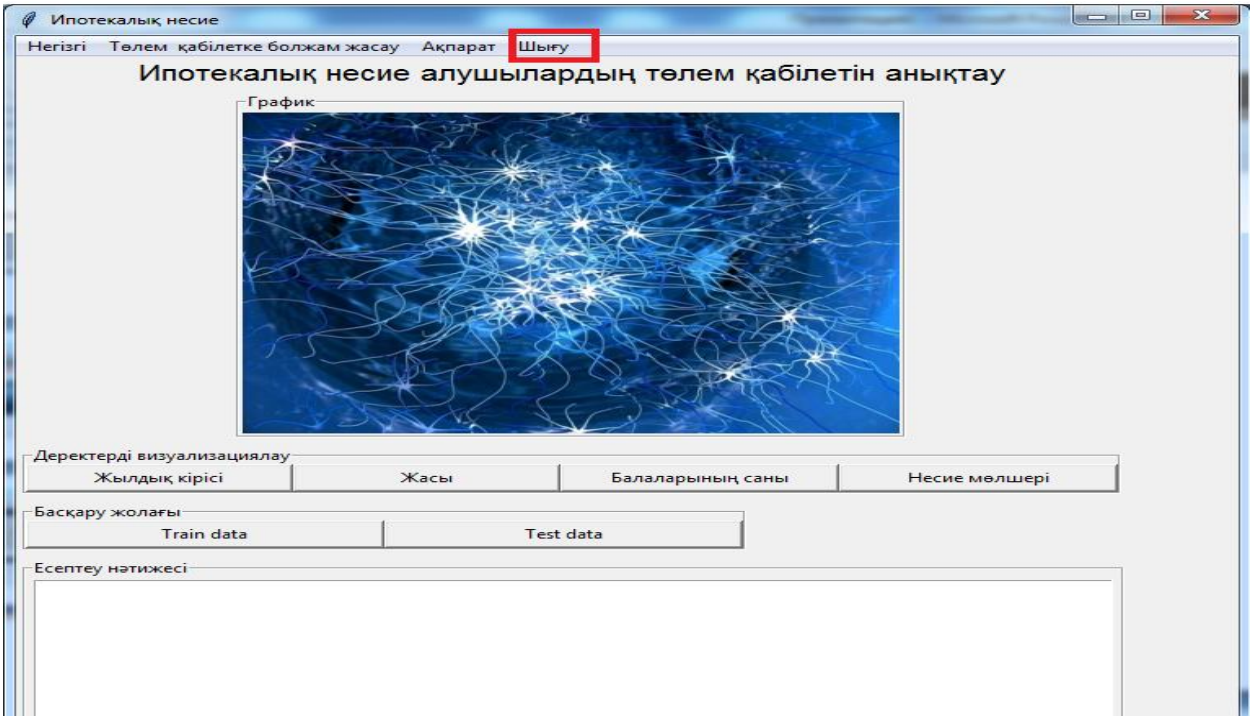


Сурет 3.19 – Тесттік деректерді өңдеу нәтижесі

Жекелеген клиенттердің де ID номерлерін енгізу арқылы төлем қабілеттерін анықтауға болады.



Сурет 3.20 – Жеке тұлғаның төлем қабілетін болжау нәтижесі



Сурет 3.21 – Бағдарламалық қамтамадан шығу

ҚОРЫТЫНДЫ

Диссертациялық жұмыста үлкен өлшемді құрылымданбаған деректерді өңдеу моделі мен бағдарламалық қамтама құрылды. Келесідей міндеттер шешілді:

1. Деректерді өңдеуге арналған әдістер мен өңдеу жүйелеріне сараптама жүргізілді;

2. Data Mining әдістері: сызықты регрессия, логистикалық регрессия, көпқабатты нейронды желі негізінде үлкен өлшемді деректерді өңдеудің алгоритмдері мен моделі құрылды;

3. Үлкен өлшемді деректерді өңдеу жүйесінің жұмыс сапасын бағаланып, деректер тестілеуден өтті;

4. Құрылымданбаған үлкен өлшемді деректерді өңдеу негізінде жеке тұлғалардың төлем қабілеттеріне болжам жасалды, бағдарламалық қамтама құрылды, ипотекалық несие беру ұйымының шешім шығару жүйесіне енгізілді. Енгізілгені туралы акт алынды.

Бағдарламалық қамтама келесідей алгоритмдермен жүзеге асырылды:

1. Деректерді даярлау:

- csv форматтағы үлкен өлшемді құрылымданбаған деректерді MongoDB деректер қорына трансформациялау жүргізілді.

2. Оқыту деректер импортталады.

3. Импортталған деректерді нормализациялау жүргізілді.

4. Өңделетін деректер алдын ала белгілі жауаппен салыстырылды.(future matrix, target y_i)

5. Өңделетін деректер DataMining әдістері: сызықты регрессия, логистикалық регрессия, көп қабатты нейрондық желілер бойынша өңделіп, нәтижелердің дәлдіктері анықталды.

6. Егер модель көп қателіктерді анықтаса, цикл қайталанып MongoDB дерекқорынан қашанғы қателік пайызы азайғанша деректерді алып өңдеуін тоқтатпайды. Қателік азайғанға дейін цикл қайталанып салмақ коэффициенті өзгеріп отырады. Қателікті қолмен енгізумен қатар, машиналық оқыту алгоритмдері сызықты регрессияны, логистикалық регрессияны, көпқабатты нейронды желіні қолдану кезінде қайта оқыту болмауын қадағалауымыз керек, ол үшін әрбір итерация үшін график алып бақылауда ұсауымыз маңызды. Егер қайта оқыту болған жағдайда ол модельдің шынайылығы күмән туғызатын болады және оны жарамсыз деп айтуымызға негіз бар.

Бұл құрылған жүйе қаржы ұйымының ипотекалық несие беру кезіндегі жұмыстарда тестілеуден сәтті өтті. Өндіріске сәтті енгізілгені туралы А қосымшасында акт берілген. Үлкен өлшемді құрылымданбаған деректерді өңдеуге арналған бағдарламалық қамтама жеке тұлғалардың төлем қабілеттерін анықтауға, болжам жасауға үлкен септігін тигізеді. Барлық шынайы деректер қаржы ұйымдарынан алынды, құрылған модель сапасының жоғары екеніне көз жеткізілді.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1 Хашковский В. В., Шкурко А. Н. Современные подходы в организации систем обработки больших объемов данных. Известия Южного федерального университета. Технические науки. 2014. № 8 (157). С. 241–250.

2 Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Технологии анализа данных. Data Mining, Visual Mining, Text Mining, OLAP. 2 изд. СПб.: БХВ-Петербург, 2007.- 384 с.

3 Balakayeva G.T., Darkenbayev D.K. Modeling the processing of a large amount of data. Al-Farabi Kazakh National University. Journal of Mathematics, Mechanics and Computer Science №1 (97) 2018.- P.120-126.

4 Даркенбаев Д.Қ. Үлкен көлемді деректерді сақтау және талдау әдістері. «Заманауи зерттеулердің өзекті мәселелері» II Халықаралық ғылыми-практикалық интернет конференциясы. Нұр-Сұлтан, 2019. Б.120-124. ISBN 978-601-275-886-4.

5 Thomas L. C., EdelmanD. B., Crook, J. N. Credit Scoring and Its Applications // SIAM Monographs on Mathematical Modeling and Computation. – Philadelphia: SIAM, 2002. – 263 p.

6 G. Balakayeva, D.Darkenbayev. The solution to the problem of processing Big Data using the example of assessing the solvency of borrowers. Journal of Theoretical and Applied Information Technology-2020. Vol.98. No 13. P.-2659-2670. ISSN: 1992-8645.

7 Большие Данные // Толковый словарь на Академике.2014. [Электронный ресурс]. URL: https://dic.academic.ru/dic.nsf/ruwiki/1422719 (дата обращения: 04.04.2019).

8 Рубанов В.А. Между стандартами управления и информационной стихией // Технологический Прогноз.- 2010.-№ 3.

9 Tom White Hadoop: The Definitive Guide, 3rd Edition. O'Reilly Media, 2012, 688 p.

10 Большие данные // [Электронный ресурс].URL:https://ru.wikipedia.org. 20.07.2010.

11 Нурлыбаева К.К., Балакаева Г.Т. Алгоритмизация процесса построения скоринговых моделей // Вестник КазНТУ. Серия Технические науки - 2014.- №6(106). - С.195-200.

12 Артемов С. Big Data: новые возможности для растущего бизнеса // Инфосистемы Джет [Электронный ресурс]. URL: http://www.pcweek.ru. 20.08.2008.

13 Doug L. 3D Data Management: Controlling Data Volume, Velocity and Variety // Meta Delta. - 2001. - P.949-951.

14 Pettey C., Goasduff L. Gartner Says Solving Big Data Challenge Involves More Than Just Managing Volumes of Data [Электронный ресурс]. URL: http://www.gartner.com.27.06.2011.

15 Doug L. 3D Data Management: Controlling Data Volume, Velocity and Variety // Meta Delta. - 2001. - P.949-951.

16 Pettey C., Goasduff L. Gartner Says Solving Big Data Challenge Involves More Than Just Managing Volumes of Data [Электронный ресурс].URL: http://www.gartner.com.27.06.2011.

17 Петухов Д. Big Data. Проблема и решения [Электронный ресурс].URL: http://www.codeinstinct.pro. 11.08.2012.

18 HDFS Architecture Guide [Электронный ресурс]. URL: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html (дата обращения: 15.04.2019)

19 Apache Cassandra [Электронный ресурс]. URL: http://cassandra.apache.org (дата обращения: 15.04.2019).

20 Семенов Ю.А. Большие объемы данных (big data) [Электронный ресурс].URL: http://book.itep.ru.21.04.2013.

21 Jacobs A.The pathologies of big data // Communications of the ACM. 2009. Vol. 52. Iss. 8. P. 36–44.

22 Tsvetkov V. Ya., Lobanov A. A.Big Data as Information Barrier // European researcher. Series A. 2014. Vol. 78. Iss. 7-1. P. 1237-1242.

23 Баженов Р.И. Интеллектуальные информационные технологии. Биробиджан: ПГУ им. Шолом-Алейхема, 2011.-176 с.

24 Lockwood G.K. Conceptual Overview of Map/Reduce and Hadoop [Электронный ресурс]: http://www.glennklockwood.com. 28.06.2014.

25 АншинаМ. Методы работы с большими данными и их эффективность // Конференция Big Data: возможность или необходимость, 26 марта. – Москва, 2013.-312 с.

26 ШипуновА. Б., Балдин Е. М. Анализ данных с R [Электронный ресурс]. URL: http://www.inp.nsk.su.05.01.2008.

27 Abadi D.J., Madden S., Hachem N. ColumnStores vs. RowStores: How Different Are They Really? // Proceedings of the ACM SIGMOD International Conference on Management of Data. - Vancouver.– 2008.-P.475.

28 Гаврилов Д. Аналитическая платформа – что это такое? [Электронный ресурс].URL: http://www.abc.org.ru. 28.12.2006.

29 Fernández A, Río S, López V, Bawakid A, del Jesus M, Benítez J, Herrera F (2014) Big data with cloud computing: an insight on the computing environment, MapReduce and programming framework.WIREs Data Min Knowl Discov 4(5) P. 380–409.

30 И.В. Котенко, И.Б. Саенко. Построение системы интеллектуальных сервисов для защиты информации в условиях кибернетического противоборства // Труды СПИИРАН. 2012. № 3(22). С. 84–100.

31 G. T. Balakayeva, C. Phillips, D. K. Darkenbayev, M. Turdaliyev. Using NoSQL for processing unstructured Big Data. News of the National Academy of sciences of the Republic of Kazakhstan. ISSN 2224-5278 Volume 6, Number 438, 2019.- P.12 – 21.

32 G.T. Balakayeva, D.K. Darkenbayev, Chris Phillips. Investigation of technologies of processing of Big Data. International Journal of Mathematics and Physics 8, №2, (13) 2017.-P.13-18.

33 Зобова Е. В., Самойлова С. С. Управление кредитным риском в коммерческих банках // Социально-экономические явления и процессы. Тамбов, 2012. № 12 (046). С. 74-81.

34 Международная конвергенция измерения ка-питала и стандартов капитала: уточненные рамочные подходы (Базель 2). М., 2004.

35 Черкашенко В.Н.Управление рисками кредитования малого и среднего бизнеса. [Электронный ресурс].URL: http://bankir.ru. 31.07.2012.

36 Ворошилова И. В.,Сурина И. В.К вопросу о совершенствовании механизма оценки кредитоспособности индивидуальных заемщиков [Электронный ресурс].URL: http://ej.kubagro.ru. 03.08.2005

37 Усачев С. Кредитный скоринг // Банки и технологии. № 04. 2008. С.128.

38 Усачёв С. Кредитный скоринг: решения desktop или enterprise// Банки и технологии.–2008. - №04. - С.50-54.

39 D. Evgueni Solojentsev, Scenario Logic and Probalilistic Management of Risk in Business and Engineering. Springer Science Business Media Inc 2005. Printed in USA springeronline.com

40 HAH Abdou, J. Pointon, Credit scoring, statistical techniques and evaluation criteria: Are view of the literature This version is available at: [Электронный ресурс].URL: http://usir.salfor d.ac.uk/id/e print/16518/ 2011

41 Буракова В.А. Проблемы применения скоринга в российской банковской практике [Электронный ресурс].URL: http://bizness-gruppa.ru. 28.01.2009.

42 Нурлыбаева К.К., Балакаева Г.Т. Анализ больших объемов данных для принятия решения // Сборник тезисов Международной конференции студентов и молодых ученых. – Алматы, 2014. -С.133.

43 Скажи, кто твой заемщик, и я скажу кто ты [Электронный ресурс].URL: http://www.credits.ru. 05.11.2003.

44 Darkenbayev D.K. Building a linear regression model for processing Big Data in the definition of solvency of citizens. Материалы международной научно-технической конференции «I юбилейные чтения Бойко Ф. К.», посвященной 100-летию Бойко Ф. К. Павлодар-2020. С.23-29.

45 Darkenbayev D.K. Numerical solution of the reggression model for analysis and processing of Big Data. Vestnik KazNRTU, Technical sciences series №6 (130) 2018.-P.132-139.

46 G.T.Balakayeva, D.K.Darkenbayev Correlation and regression analysis for Big Data processing. Vestnik KazNRTU, Technical sciences series №1(131) 2019.- P.338-345.

47 Прикладной регрессионный анализ / Н. Дрейпер, Г. Смит - 2-е изд. - М.: Финансы и статистика, (перевод с английского) 1986.-296 с.

48 Теория вероятностей и математическая статистика / В.Е. Гмурман - 9-е изд. -М.: Высшая школа, 2003.-126 с.

49 Теория вероятностей / Н.И. Чернова - Новосибирск: Новосибирский государственный университет, 2007.-318 с.

50 Бизнес-аналитика: от данных к знаниям / Паклин Н. Б., Орешков В. И.- 2-е изд. - Санкт-Петербург: Питер, 2013.-312 с.

51 Data Science Наука о данных с нуля / Джоэл Грас - Санкт-Петербург: БХВ Петербург, 2017.-308 с.

52 Практическая статистика для специалистов Data Science / П.Брюс, Э.Брюс Санкт-Петербург: БХВ Петербург, 2018.-127 с.

53 Барсегян А.А., Куприянов М.С. и др. Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004. – 116 с.

54 Data Mining в системе управления знаниями [Электронный ресурс]. – Режим доступа: http://www.smart-edu.com/upravlenieznaniyami/izvlechenie-znaniy-data-mining-vsisteme-upravleniya-znaniyami

55 Барсегян А.А.Анализ данных и процессов: учеб. пособие – 3-е изд. - СПб.: БХВ-Петербург, 2009.-121 с.

56 BersonA, SmithS. J. DataWarehousing, DataMining&OLAP. McGrawHill, 1997. – P. 15-16

57 Беляков Д. Е., Решение Задач Классификации с Помощью Деревьев Решений - УДК 519.6 Москва, 2016.- С. 473.

58 J. E.Aronson, T.-P. Leing, E. Turban, "Neural Network in Data Mining," in Decission Support System, (Prentice-Hall of India Private Limited, 2007) – 18 p.

59 J. A. Anderson, J. W. Silverstein, S. A. Ritzand R. S. Jones, “Distinctive features, categoricalperception, and probability learning: Some applications of a neural model”, vol. 84, 1977 – 413 p.

60 Fawcett T. ROC Graphs: Notes and Practical Considerations for Researchers // Kluwer Academic Publishers. 2004.- 127 p.

61 Zweig M.H., Campbell G. ROC Plots: A Fundamental Evaluation Tool in Clinical Medicine // Clinical Chemistry, Vol. 39, No. 4, 1993.-P.1227-1233.

62 Linear Regression Using Least Squares [Электронный ресурс].URL: https://towardsdatascience.com/linear-regression-using-least-squares-a4c3456e8570.

63 Solutions of problems on the topic: The Least Square Method (Processing Experimental Data). MatBuro – Solving problems in mathematics, statistics, economics and programming [Электронный ресурс].URL: www.matburo.ru

64 Newsom. I. Data Analysis II: Logistic Regression, 2015. P.573-580

65 Chitra K., and B.Subashini. "Data Mining Techniquesandits Applicationsin Banking Sector." International Journal of Emerging Technologyand Advanced Engineering 3.8, 2013. P. 219-226

66 Логистическая регрессия и ROC-анализ – математический аппарат [Электронный ресурс].URL: https://loginom.ru/blog/logistic-regression-roc-auc

67 C. Drummond, R.C. Holte, “Costcurves: Animproved method for visualizing classifier performance”, Mach. Learn. 65, 2006 – P. 96-98.

68 Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: классификация и снижение размерности. - М.: Финансы и статистика, 1989.- С. 42-60.

69 David W. Hosmer, Stanley Lemeshow. Applied Logistic Regression, 2nd ed. New York, Chichester, Wiley. 2002.- P. 392. ISBN 0-471-35632-8.

70 Fausett Laurene, Fundamentals of Neural Networks: Architectures, Algorithmsand Applications. Prentice-Hall, NewJersey, USA. Int., Inc., 1994 – P. 264-265

71 J. E.Aronson, T.-P. Leing, E. Turban, "Neural Networkin DataMining," inDecissionSupportSystem, (Prentice-HallofIndiaPrivateLimited, 2007) – P. 18

72 Ковалев М., Корженевская В. Методика построения банковской скоринговой модели для оценки кредитоспособности физических лиц // Банки Казахстана. - 2008. - №1.- С.43-48

73 BaseGroupLabsНейронные сети. // http://www.basegroup.ru. 21.02.2012

74 J.A. Anderson, J. W. Silverstein, S. A. RitzandR. S. Jones, “Distinctive features, categorical perception, and probability learning: Some applications of a neural model”, vol. 84, 1977 – P. 413

75 Boyle M., Crook J. N., Hamilton R., Thomas L. C. Methods applied to slow payers. - Oxford: Oxford University Press,1992. - № 65 (1). – P.58–70.

76 Henley W. E. Statistical Aspects of Credit Scoring. -Milton Keynes: Open University, 1995. - №1 (45). -P.77-95.

77 Desai V. S., Conway D. G., Crook J., Overstreet G. Credit-scoring models in the credit union environment using neural networks and genetic algorithms // IMA Journal of Mathematics Applied in Business and Industry. -1997. -№ 8(4). - P.323-346.

78 West D. Neural Network Credit Scoring Models // Computers and Operations Research. – 2000. -№27. – P.1131-1152.

79 Baesens B. Developing Intelligent Systems for Credit Scoring Using Machine Learning Techniques: Thesis Ph.D., Katholieke Universiteit. - Leuven,2003.- 180 p.

80 Srinivasan V., KimYong H.Credit Granting: A Comparative Analysis. of Classification Procedures // The Journal of Finance. - 1987. – №42 (3). –P. 665–681.

81 Yobas M.B., Crook J.N., Ross P. Credit Scoring Using Evolutionary Techniques // IMA Journal of Mathematics Applied in Business and Industry. – 2000. - №11. - P.111-125.

82 Malhotra R., Malhotra D. K. Evaluating consumer loans using Neural Networks //Omega the International Journal of Management Science. – 2003. -№31 (2). – P.83-96.

83 Lee T.S., Chen N.J. Investigating the Information Content of Non-Cash-Trading Index Futures Using Neural Networks // Expert Systems with Applications. – 2002. - № 22 (3). - P. 225-234.

Қосымша А

Бағдарламалық кодтың толық нұсқасы төмендегідей:

```
From pymongo import MongoClient
 #import pprint
 import numpy as np
 from sklearn import preprocessing
 #from sklearn.preprocessing import StandardScaler
#from pandas import DataFrame
import pandas as pd
client = MongoClient(port=27017)
db=client.Dauren
 collection = db.baza
 #collection = db.train_data
#pprint.pprint(collection.find_one())
print("Count of collection: ",collection.count_documents({}))
INCOME_TOTAL0=[]
INCOME_TOTAL=[]
F_MEMBERS0=[]
F_MEMBERS=[]
DAYS_BIRTH0=[]
DAYS_BIRTH=[]
AMT_GOODS_PRICE=[]
AMT_CREDIT0=[]
AMT_CREDIT=[]
REZULT=[]
CODE_GENDER=[]
FLAG_OWN_CAR=[]
NAME_EDUCATION_TYPE=[]
NAME_INCOME_TYPE=[]
DATA=[]
synaptic_weights = np.random.random((4,1))
 #print("synaptic_weights",synaptic_weights)
 for post in collection.find().limit(48744):
#INCOME_TOTAL.append(post[0])
 x1 = post.get("AMT_INCOME_TOTAL")
x2 = post.get("CNT_FAM_MEMBERS")
x3 = post.get("DAYS_BIRTH")
 x4 = post.get("AMT_GOODS_PRICE")
 x5 = post.get("AMT_CREDIT")
 x6 = post.get("CODE_GENDER")
```

```
x7 = post.get("FLAG_OWN_CAR")
x8 = post.get("NAME_EDUCATION_TYPE")
 x9 = post.get("NAME_INCOME_TYPE")
 y = post.get("TARGET")
INCOME_TOTAL0.append(x1)
F_MEMBERS.append(x2)
DAYS_BIRTH.append(x3)
AMT_GOODS_PRICE.append(x4)
AMT_CREDIT.append(x5)
CODE_GENDER.append(x6)
FLAG_OWN_CAR.append(x7)
 NAME_EDUCATION_TYPE.append(x8)
 NAME_INCOME_TYPE.append(x9)
 REZULT.append(y)
#import test data from mongoDB    application_test
#print(DAYS_BIRTH)
 #pprint.pprint(post)
#REZULT=np.array([REZULT]).T
DAYS_BIRTH[:] = [i*-1 for i in DAYS_BIRTH]
DAYS_BIRTH[:] = [j / 365 for j in DAYS_BIRTH]
GENDER=["M","F"]
encoder1 = preprocessing.LabelEncoder()
encoder1.fit(GENDER)
encoded_GENDER = encoder1.transform(CODE_GENDER)
OWN_CAR=["N","Y"]
encoder2 = preprocessing.LabelEncoder()
encoder2.fit(OWN_CAR)
encoded_OWN_CAR = encoder2.transform(FLAG_OWN_CAR)
EDUCATION_TYP=["Secondary / secondary special","Higher education","Incomplete
higher","Lower secondary","Academic degree"]
encoder3 = preprocessing.LabelEncoder()
encoder3.fit(EDUCATION_TYP)
encoded_EDUCATION_TYPE= encoder3.transform(NAME_EDUCATION_TYPE
INCOME_TYPE=["Working","State servant","Pensioner","Commercial
associate","Unemployed"]
encoder4 = preprocessing.LabelEncoder()
encoder4.fit(INCOME_TYPE)
encoded_INCOME_TYPE = encoder4.transform(NAME_INCOME_TYPE)
F_MEMBERS = np.asarray(F_MEMBERS)
F_MEMBERS = (F_MEMBERS) / F_MEMBERS.max(axis=0)
#print(INCOME_TOTAL)
INCOME_TOTAL = np.asarray(INCOME_TOTAL0)
```

```
INCOME_TOTAL = (INCOME_TOTAL) / INCOME_TOTAL.max(axis=0)
DAYS_BIRTH = np.asarray(DAYS_BIRTH)
DAYS_BIRTH = (DAYS_BIRTH) / DAYS_BIRTH.max(axis=0)
AMT_GOODS_PRICE = np.asarray(AMT_GOODS_PRICE)
AMT_GOODS_PRICE = (AMT_GOODS_PRICE) /
AMT_GOODS_PRICE.max(axis=0)
#print(AMT_CREDIT)
AMT_CREDIT = np.asarray(AMT_CREDIT)
AMT_CREDIT = (AMT_CREDIT) / AMT_CREDIT.max(axis=0)
#print(AMT_CREDIT)
encoded_EDUCATION_TYPE = np.asarray(encoded_EDUCATION_TYPE)
encoded_EDUCATION_TYPE = (encoded_EDUCATION_TYPE) /
encoded_EDUCATION_TYPE.max(axis=0)
encoded_INCOME_TYPE = np.asarray(encoded_INCOME_TYPE)
INCOME_TYPE = (encoded_INCOME_TYPE) /
encoded_INCOME_TYPE.max(axis=0)
print(REZULT)
DATA=list(map(list,zip(INCOME_TOTAL,F_MEMBERS,DAYS_BIRTH,AMT_CRE
DIT,REZULT)))
DATA = np.array(DATA)
#print (DATA)
df = pd.DataFrame(DATA,columns= ['INCOME_TOTAL',
'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT','REZULT'])
X = df[['INCOME_TOTAL', 'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT']]
#print(X)
y = df['REZULT']
print(y)
from pymongo import MongoClient
import numpy as np
from sklearn import preprocessing
import pandas as pd
client = MongoClient(port=27017)
db=client.dauren
collection = db.application_test
#collection = db.train_data
#pprint.pprint(collection.find_one())
print("Count of collection: ",collection.count_documents({}))
INCOME_TOTAL=[]
F_MEMBERS=[]
DAYS_BIRTH=[]
AMT_GOODS_PRICE=[]
AMT_CREDIT=[]
```

```
DATA2=[]
synaptic_weights = np.random.random((4,1))
#print("synaptic_weights",synaptic_weights)
for post in collection.find().limit(356256):
 #INCOME_TOTAL.append(post[0])
 x1 = post.get("AMT_INCOME_TOTAL")
 x2 = post.get("CNT_FAM_MEMBERS")
 x3 = post.get("DAYS_BIRTH")
 x4 = post.get("AMT_GOODS_PRICE")
 x5 = post.get("AMT_CREDIT")
 x6 = post.get("CODE_GENDER")
 x7 = post.get("FLAG_OWN_CAR")
 x8 = post.get("NAME_EDUCATION_TYPE")
 x9 = post.get("NAME_INCOME_TYPE")
 INCOME_TOTAL.append(x1)
 F_MEMBERS.append(x2)
 DAYS_BIRTH.append(x3)
 AMT_GOODS_PRICE.append(x4)
 AMT_CREDIT.append(x5)
#import test data from mongoDB   application_test
#print(DAYS_BIRTH)
 #pprint.pprint(post)
#REZULT=np.array([REZULT]).T
DAYS_BIRTH[:] = [i*-1 for i in DAYS_BIRTH]
DAYS_BIRTH[:] = [j / 365 for j in DAYS_BIRTH]
F_MEMBERS = np.asarray(F_MEMBERS)
F_MEMBERS = (F_MEMBERS)/F_MEMBERS.max(axis=0)
#print(INCOME_TOTAL)
INCOME_TOTAL = np.asarray(INCOME_TOTAL)
INCOME_TOTAL = (INCOME_TOTAL) / INCOME_TOTAL.max(axis=0)
DAYS_BIRTH = np.asarray(DAYS_BIRTH)
DAYS_BIRTH = (DAYS_BIRTH) / DAYS_BIRTH.max(axis=0)
AMT_GOODS_PRICE = np.asarray(AMT_GOODS_PRICE)
AMT_GOODS_PRICE = (AMT_GOODS_PRICE) /
AMT_GOODS_PRICE.max(axis=0)
AMT_CREDIT = np.asarray(AMT_CREDIT)
AMT_CREDIT = (AMT_CREDIT) / AMT_CREDIT.max(axis=0)
#print(AMT_CREDIT)
DATA2=list(map(list,zip(INCOME_TOTAL,F_MEMBERS,DAYS_BIRTH,AMT_CR
EDIT)))
DATA2 = np.array(DATA2)
#print (DATA)
```

```
df = pd.DataFrame(DATA2,columns= ['INCOME_TOTAL',
'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT'])
TEST = df[['INCOME_TOTAL', 'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT']]
print(TEST)
```

Бағдарламалық қамтама интерфейсінің коды

```
# -*- coding: utf-8 -*-
Created on Wed Aug 26 13:27:31 2020
@author: Dauren"""
 import tkinter as tk
 from tkinter import *
 from tkinter import scrolledtext
 from tkinter import messagebox
import matplotlib.pyplot as plt
from matplotlib.animation import ArtistAnimation
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import numpy as np
from PIL import Image, ImageTk
from mpl_toolkits.mplot3d import Axes3D
import os.path
from matplotlib import cm
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import MinMaxScaler
import seaborn as sn
#import load_data_from_Mongo
from load_data_from_Mongo import DATA
from load_test_data_from_Mongo import DATA2
from load_data_from_Mongo import X
from pymongo import MongoClient
#from load_data_from_Mongo import y
class OOP():
 def __init__(self):
 self.createWidgets()
 def createWidgets(self):
  root =  Tk()
  root.title('Ипотекалық несие')
  root.geometry('750x750')
menu = tk.Menu(root)
```

```
root.config(menu=menu)
file_menu1 = tk.Menu(menu, tearoff=0)
file_menu3 = tk.Menu(menu, tearoff=0)
file_menu4 = tk.Menu(menu, tearoff=0)
 file_menu5 = tk.Menu(menu, tearoff=0)
 menu.add_cascade(label='Негізгі', menu=file_menu1)
 file_menu1.add_command(label='5-10-20', command=self.m_1)
 file_menu1.add_command(label='Нұрлы жол', command=self.m_2)
 file_menu1.add_command(label='7-20-25', command=self.m_3)
 menu.add_cascade(label='Төлем  қабілетке болжам жасау', menu=file_menu3)
 file_menu3.add_command(label="Төлем  қабілетке болжам жасау",
command=self.m_4
menu.add_cascade(label='Ақпарат', menu=file_menu4)
file_menu4.add_command(label="Ақпарат", command=self.helf
 menu.add_cascade(label='Шығу', menu=file_menu5)
file_menu5.add_command(label="Шығу", command=root.destroy
 label1=tk.Label(root, text="Ипотекалық несие алушылардың төлем қабілетін
анықтау",width=60,font=("Helvetica", 15))
label1.grid(row=0, column=0,columnspan=4,pady=0)
 self.monty1 = tk.LabelFrame(root, text='График')
 self.monty1.grid(row=1,  column=0,  columnspan=4, padx=4,  pady=4
 #self.monty7.grid(row=1,column=6, columnspan=2, padx=4,  pady=4, rowspan=5)
 self.c = tk.Canvas(self.monty1, width = 750, height = 300, bg = 'white')
 self.c.grid(row=1, column=2, columnspan=2)
 load = Image.open("ANN.jpg")
 render = ImageTk.PhotoImage(load)
 img = tk.Label(self.c, image=render)
 img.image = render
 img.grid(row=0, column=2,columnspan=2)
 self.monty2 = tk.LabelFrame(root, text='Деректерді визуализациялау')
 self.monty2.grid( row=2, column=0, columnspan=4, padx=4,  pady=4,sticky="W")
 tk.Button(self.monty2,text="Жылдық кірісі",
width=22,command=self.func_1).grid(row=2, column=0)
 tk.Button(self.monty2, text='Клиенттің жасы', width=22,
command=self.func_2).grid(row=2, column=1)
 tk.Button(self.monty2, text='Балаларының саны', width=23,
command=self.func_3).grid(row=2, column=2)
 tk.Button(self.monty2, text='Несие мөлшері', width=23,
command=self.func_4).grid(row=2, column=3)
self.monty3 = tk.LabelFrame(root, text='Басқару жолағы')
self.monty3.grid(row=3, column=0, columnspan=4, padx=4,  pady=4,sticky="W")
 tk.Button(self.monty3,text="Train data", width=30,
```

```
command=self.func_5).grid(row=3, column=0)
tk.Button(self.monty3, text='Test data', width=30,
 command=self.func_6).grid(row=3, column=1)
tk.Button(self.monty3, text='', width=32,
command=self.func_6).grid(row=3, column=2
 self.monty4 = tk.LabelFrame(root, text='Есептеу нәтижесі')
 self.monty4.grid(row=4, column=0,  columnspan=6, padx=4,  pady=4,sticky="W")
 scrollb = tk.Scrollbar(self.monty4)
 scrollb.grid(row=5, column=0, columnspan=4, pady=4)
 self.listbox = tk.Listbox(self.monty4,width=110,height=10)
 self.listbox.grid(row=5, column=0, columnspan=4, pady=2, padx=4)
 self.listbox.config(yscrollcommand=scrollb.set)
 scrollb.config(command=self.listbox.yview
# menuBar = Menu(tab1)
# self.win.config(menu=menuBar)
# fileMenu = Menu(menuBar, tearoff=0)
# menuBar.add_cascade(label="File", menu=fileMenu)
# helpMenu = Menu(menuBar, tearoff=0)
# menuBar.add_cascade(label="Help", menu=helpMenu
    def m_3(msg):
print("7-20-25")
popup = tk.Tk()
popup.wm_title("7-20-25")
 msg="""«7-20-25.
 label = tk.Label(popup, text=msg, font=("Helvetica", 15))
 label.pack(side="top", fill="x", pady=10)
 popup.mainloop()
 def m_2(msg):
 print("Нұрлы жол")
 popup = tk.Tk()
 popup.wm_title("Нұрлы жол")
 msg="""
 label = tk.Label(popup, text=msg, font=("Helvetica", 15))
 label.pack(side="top", fill="x", pady=10)
 popup.mainloop(
 def m_1(msg):
 print("5-10-20")
 popup = tk.Tk()
popup.wm_title("5-10-20")
 msg="""
 label = tk.Label(popup, text=msg, font=("Helvetica", 15))
label.pack(side="top", fill="x", pady=10)
```

```
popup.mainloop()
def m_4(self):
print("m_4")
 popup = tk.Tk()
 popup.wm_title("5-10-20")
 msg="""Кез келген клиенттің ID енгізу арқылы оның төлем қабілетіне болжам
жасау:"""
label = tk.Label(popup, text=msg, font=("Helvetica", 15))
label.pack(side="top", fill="x", pady=10)
e = tk.Entry(popup, width=50)
e.pack()
def predict():
print("predict new client")
print("Entry:",e.get())
 print(type(e.get()))
 kl=e.get()
 print("kl:",kl)
 print(type(kl))
 ID=int(kl)
 print("ID: ", ID)
 print(type(ID))
 client = MongoClient(port=27017)
 db=client.dauren
 collection = db.application_test
 INCOME_TOTAL_n=[]
 F_MEMBERS_n=[]
 DAYS_BIRTH_n=[]
 AMT_CREDIT_n=[]
 DATA_n=[]
 for post in collection.find().limit(1400):
  #print(type(post.get("SK_ID_CURR")))
  if (ID== post.get("SK_ID_CURR")):
  x1 = post.get("AMT_INCOME_TOTAL")
  x2 = post.get("CNT_FAM_MEMBERS")
  x3 = post.get("DAYS_BIRTH")
  x5 = post.get("AMT_CREDIT"
  INCOME_TOTAL_n.append(x1)
  F_MEMBERS_n.append(x2)
  DAYS_BIRTH_n.append(x3)
  AMT_CREDIT_n.append(x5)
  print(INCOME_TOTAL_n)
print(F_MEMBERS_n)
```

```
print(DAYS_BIRTH_n)
print(AMT_CREDIT_n)
#DAYS_BIRTH_n[:] = [i*-1 for i in DAYS_BIRTH_n]
 #DAYS_BIRTH_n[:] = [j / 365 for j in DAYS_BIRTH_n]
#F_MEMBERS_n = np.asarray(F_MEMBERS_n)
#F_MEMBERS_n = (F_MEMBERS_n) / F_MEMBERS_n.max(axis=0)
 #print(INCOME_TOTAL)
 #INCOME_TOTAL_n = np.asarray(INCOME_TOTAL_n)
 #INCOME_TOTAL_n = (INCOME_TOTAL_n) / INCOME_TOTAL_n.max(axis=0)
 #DAYS_BIRTH_n = np.asarray(DAYS_BIRTH_n)
 #DAYS_BIRTH_n = (DAYS_BIRTH_n) / DAYS_BIRTH_n.max(axis=0)
 #print(AMT_CREDIT)
 #AMT_CREDIT_n = np.asarray(AMT_CREDIT_n)
 #AMT_CREDIT_n = (AMT_CREDIT_n) / AMT_CREDIT_n.max(axis=0)
 #print(AMT_CREDIT)
df1 = pd.DataFrame(DATA,columns= ['INCOME_TOTAL',
'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT','REZULT'])
 X = df1[['INCOME_TOTAL',
'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT']]
 #print(X)
y = df1['REZULT']
DATA_n=list(map(list,zip(INCOME_TOTAL_n,F_MEMBERS_n,DAYS_BIRTH_n,A
MT_CREDIT_n)))
 DATA_n = np.array(DATA_n)
 print (DATA_n)
 scaler = MinMaxScaler()
 DATA_n = scaler.fit_transform(DATA_n)
 df = pd.DataFrame(DATA_n,columns= ['INCOME_TOTAL',
'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT']
 TEST_n = df[['INCOME_TOTAL',
 'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT']]
 print (TEST_n)
 mlp = MLPClassifier(max_iter=1000,random_state=0)
 mlp.fit(X,y)
 Mnn_pred=mlp.predict(TEST_n)
 print(Mnn_pred)
 self.listbox.insert(tk.END,ID, "ID :", ID)
 self.listbox.insert(tk.END,"Клиенттің болжамы:", Mnn_pred)
 b = tk.Button(popup, text="Болжау", width=10, command=predict)
 b.pack(
 popup.mainloop()
def helf(msg):
```

```
 popup = tk.Tk()
 popup.wm_title("Бағдарламалық кешен авторлары туралы ақпарат")
 msg="""
 label = tk.Label(popup, text=msg, font=("Helvetica", 15))
label.pack(side="top", fill="x", pady=10)
popup.mainloop(
def func_1(self):
 print("func_1")
fig2 = plt.Figure(figsize=(5, 4))
ax=fig2.add_subplot(1,1,1)
ax.plot(X['INCOME_TOTAL'], label='Жылдық кірісі')
ax.set_title('Жылдық кірісі')
ax.legend(['Жылдық кірісі'])
ax.set_xlabel('Клиенттер')
ax.set_ylabel('Жылдық кірісі')
canvas = FigureCanvasTkAgg(fig2, master=self.monty1)  # A tk.DrawingArea.
canvas.get_tk_widget().grid(row=1, column=2, columnspan=2)
def func_2(self):
print("func_2")
 fig2 = plt.Figure(figsize=(5, 4))
 ax=fig2.add_subplot(1,1,1)
 ax.plot(X['DAYS_BIRTH'], label='Years of birth')
 ax.set_title('Years of birth')
 ax.legend(['Years of birth'])
 ax.set_xlabel('Clients')
 ax.set_ylabel('Years of birth')
 canvas = FigureCanvasTkAgg(fig2, master=self.monty1)
 canvas.get_tk_widget().grid(row=1, column=2, columnspan=2
 def func_3(self):
 print("func_3")
 fig2 = plt.Figure(figsize=(5, 4))
 ax=fig2.add_subplot(1,1,1)
 ax.plot(X['F_MEMBERS'], label='Балаларының саны')
ax.set_title('Балаларының санып')
ax.legend(['Балаларының саны'])
ax.set_xlabel('Клиенттер')
ax.set_ylabel('Балаларының саны'
canvas = FigureCanvasTkAgg(fig2, master=self.monty1)
canvas.get_tk_widget().grid(row=1, column=2, columnspan=2
def func_4(self):
print("func_4")
fig2 = plt.Figure(figsize=(5, 4))
```

```
ax=fig2.add_subplot(1,1,1)
ax.plot(X['AMT_CREDIT'], label='Credit amount')
ax.set_title('Credit amount')
ax.legend(['Credit amount'])
ax.set_xlabel('Clients')
ax.set_ylabel('Credit amount'
canvas = FigureCanvasTkAgg(fig2, master=self.monty1)
canvas.get_tk_widget().grid(row=1, column=2, columnspan=2)
def func_5(self):
print("func_5")
df = pd.DataFrame(DATA,columns= ['INCOME_TOTAL',
'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT','REZULT'])
 X = df[['INCOME_TOTAL', 'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT']]
 print(X)
 y = df['REZULT']
 print("Y: ",y)
 X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.25)
 print('target x_pred:', y_train)
 print("форма массива X_train: {}".format(X_train.shape))
 print("форма массива y_train: {}".format(y_train.shape))
 print("форма массива X_test: {}".format(X_test.shape))
 print("форма массива y_test: {}".format(y_test.shape)
 print("X_test ",X_test)
 print("y_test ",y_test)
 logistic_regression= LogisticRegression(C=100).fit(X_train,y_train)
 y_pred=logistic_regression.predict(X_test)
 print("Логистикалық регрессияның дәлдігі:")
 print("Accurqcy of train data: {:.3f}".format(logistic_regression.score(X_train,
y_train)))
 print("Accurqcy of test data: {:.3f}".format(logistic_regression.score(X_test, y_test)))
 y_test = np.array(y_test)
 y_pred = np.array(y_pred)
confusion_matrix = pd.crosstab(y_test, y_pred, rownames=['Нақты'],
colnames=['Болжанған'])
 sn.heatmap(confusion_matrix, annot=True)
 #print('Accuracy: ',metrics.accuracy_score(y_test, y_pred))
 linreg = LinearRegression().fit(X_train, y_train)
 lin_pred=linreg.predict(X_test)
#confusion_matrix2 = pd.crosstab(y_test, lin_pred, rownames=['Нақты2'],
colnames=['Болжанған2'])
 #sn.heatmap(confusion_matrix2, annot=True
 print("\n Сызықтық регрессияның дәлдігі: ")
```

```
 print("Accurqcy of train data: {:.3f}".format(linreg.score(X_train, y_train)))
 print("Accurqcy of test data: {:.3f}".format(linreg.score(X_test, y_test))
 mlp = MLPClassifier(max_iter=99,random_state=0)
 mlp.fit(X_train, y_train)
 Mnn_pred=mlp.predict(X_test)
#confusion_matrix = pd.crosstab(y_test, Mnn_pred, rownames=['Нақты'],
colnames=['Болжанған2'])
#sn.heatmap(confusion_matrix, annot=Tru
 print("Көпқабатты нейронның дәлдігі: ")
 print("Accurqcy of train data: {:.2f}".format(mlp.score(X_train, y_train)))
 print("Accurqcy of test data: {:.2f}".format(mlp.score(X_test, y_test)))
 fig = plt.Figure(figsize=(5, 4))
 ax=fig.add_subplot(1,1,1)
 ax.plot(y_test,label='Target')
 ax.plot(y_pred, label='Логистикалық регрессия')
 ax.plot(lin_pred, label='Сызықтық регрессия')
 ax.plot(Mnn_pred, label='Нейронды желі')
 ax.set_title('Әдістердің дәлдігі')
ax.set_xlabel('Клиенттер')
ax.set_ylabel('Дәлдігі')
ax.legend()
canvas = FigureCanvasTkAgg(fig, master=self.monty1)
canvas.get_tk_widget().grid(row=1, column=2, columnspan=2
self.listbox.insert(tk.END,"Логистикалық регрессия:")
self.listbox.insert(tk.END,"Accurqcy of train data:
{:.2f}".format(logistic_regression.score(X_train, y_train)))
 self.listbox.insert(tk.END,"Accurqcy of test data:
{:.2f}".format(logistic_regression.score(X_test, y_test)))
 self.listbox.insert(tk.END,"Сызықтық регрессия:")
 self.listbox.insert(tk.END,"Accurqcy of train data:
 {:.2f}".format(linreg.score(X_train, y_train)))
 self.listbox.insert(tk.END,"Accurqcy of test data:
 {:.2f}".format(linreg.score(X_test, y_test)))
 self.listbox.insert(tk.END,"Нейронды желі:")
 self.listbox.insert(tk.END,"Accurqcy of train data:
{:.2f}".format(mlp.score(X_train, y_train)))
 self.listbox.insert(tk.END,"Accurqcy of test data:
{:.2f}".format(mlp.score(X_test, y_test)))
 def func_6(self):
 print("testdata")
 df1 = pd.DataFrame(DATA,columns= ['INCOME_TOTAL',
'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT','REZULT'])
```

```
 X = df1[['INCOME_TOTAL', 'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT']]
 print(X)
 y = df1['REZULT']
 df = pd.DataFrame(DATA2,columns= ['INCOME_TOTAL',
'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT'])
 X2 = df[['INCOME_TOTAL', 'F_MEMBERS','DAYS_BIRTH','AMT_CREDIT']]
 print(X2)
 model = LinearRegression().fit(X, y)
 r_sq = model.score(X, y)
 y_pred1 = model.predict(X2)
 self.listbox.insert(tk.END,"Сызықтық регрессия:")
 self.listbox.insert(tk.END,"Болжанған деректер:", y_pred1)
 #self.listbox.insert(tk.END, r_sq,model.intercept_)
 #y_pred1 = np.round_(y_pred)
 #print("Accuracy:",metrics.accuracy_score(DATA, y_pred))
 print(np.sum(y))
 print(np.sum(y_pred1))
 y = np.array(y)
 print('target:', y, sep='\n')
 print('predicted response:', y_pred1, sep='\n'
 fig, axs = plt.subplots(3, 1, constrained_layout=True)
 axs[0].plot(y_pred1, label='Result')
 axs[0].set_title('Сызықтық регрессия')
 axs[0].set_xlabel('Клиенттер')
 axs[0].set_ylabel('Нәтиже')
 fig.suptitle('Жаңа деректерді болжау', fontsize=16)
 logreg = LogisticRegression(solver='liblinear', random_state=0)
 logreg.fit(X,y)
 y_pred2=logreg.predict(X2)
 self.listbox.insert(tk.END,"Логикалық регрессия:")
 self.listbox.insert(tk.END,"Болжанған деректер:", y_pred2)
 axs[1].plot(y_pred2, label='Result')
 axs[1].set_xlabel('Клиенттер')
 axs[1].set_title('Логистикалық регрессия')
 axs[1].set_ylabel('Нәтиже')
 mlp = MLPClassifier(max_iter=99,random_state=0)
 mlp.fit(X,y)
 Mnn_pred=mlp.predict(X2
 axs[2].plot(Mnn_pred, label='Result')
 axs[2].set_xlabel('Клиенттер')
axs[2].set_title('Нейронды желі')
axs[2].set_ylabel('Нәтиже')
```

```
plt.show()
oop = OOP()
mainloop()
#print(X)
#import tkinter as tk
#from tkinter import ttk
#from tkinter import scrolledtext
#import matplotlib.pyplot as plt
#from matplotlib.animation import ArtistAnimation
#from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
#import numpy as np
#from PIL import Image, ImageTk
#from mpl_toolkits.mplot3d import Axes3D
#import os.path
#from matplotlib import cm
#class OOP():
#def __init__(self):
#self.win = tk.Tk()
#self.win.title("Scoring calculator")
#self.win.geometry('800x700')
#self.createWidgets()
#def createWidgets(self):
# tabControl = ttk.Notebook(self.win)
# tab1 = ttk.Frame(tabControl)
 tabControl.add(tab1, text='')
 tabControl.pack(expand=1, fill="both")
 menu = tk.Menu(self.win)
 file_menu1 = tk.Menu(menu, tearoff=0)
 file_menu2 = tk.Menu(menu, tearoff=0)
file_menu3 = tk.Menu(menu, tearoff=0)
# file_menu4 = tk.Menu(menu, tearoff=0)
# file_menu5 = tk.Menu(menu, tearoff=0)
# file_menu1.add_command(label='Государственные жилищные программы РК',)
#file_menu1.add_command(label='Нурлы жол',)
#file_menu1.add_command(label='7-20-25',)
#file_menu1.add_command(label='Жилищные программы жилстрйсбербанк',)
# file_menu1.add_separator()
# file_menu1.add_command(label='Выход' )
# menu.add_cascade(label='Главная', menu=file_menu1)
# menu.add_cascade(label='Подача заявка', menu=file_menu2)
# menu.add_cascade(label='Результать', menu=file_menu3)
# menu.add_cascade(label='Помощь', menu=file_menu4)
```

```
#menu.add_cascade(label='Выход', menu=file_menu5)
# label1=ttk.Label(tab1, text="Scoring calculation using an artificial neural network and
machine learing",width=60,font=("Helvetica", 15))
# label1.grid(row=0, column=0,columnspan=4,pady=0)
# self.monty1 = ttk.LabelFrame(tab1, text='Graphics')
# self.monty1.grid(row=1,  column=0,  columnspan=4, padx=4,  pady=4
# #self.monty7.grid(row=1,column=6, columnspan=2, padx=4,  pady=4, rowspan=5)
# self.c = tk.Canvas(self.monty1, width = 800, height = 300, bg = 'white')
# self.c.grid(row=1, column=2, columnspan=2)
# load = Image.open("ANN.jpg")
#render = ImageTk.PhotoImage(load)
# img = ttk.Label(self.c, image=render)
# img.image = render
# img.grid(row=0, column=2,columnspan=2)
# self.monty2 = ttk.LabelFrame(tab1, text='Data Visualization')
# self.monty2.grid( row=2, column=0, columnspan=4, padx=4,  pady=4,sticky="W")
# ttk.Button(self.monty2,text="Income",
#width=25,command=self.func_1).grid(row=2, column=0)
# ttk.Button(self.monty2, text='Goods price', width=25,
command=self.func_2).grid(row=2, column=1)
#ttk.Button(self.monty2, text='Number of children', width=25,
command=self.func_3).grid(row=2, column=2)
# ttk.Button(self.monty2, text='Credit amount', width=25,
command=self.func_4).grid(row=2, column=3)
# self.monty3 = ttk.LabelFrame(tab1, text='Control Bar')
# self.monty3.grid(row=3, column=0, columnspan=4, padx=4,  pady=4,sticky="W")
# ttk.Button(self.monty3,text="Linear Regression",
width=25,command=self.func_5).grid(row=3, column=0)
# ttk.Button(self.monty3, text='Logical REgression', width=25,
command=self.func_6).grid(row=3, column=1)
# ttk.Button(self.monty3, text='Artificial Neural Network', width=25,
command=self.func_7).grid(row=3, column=2)
# self.monty4 = ttk.LabelFrame(tab1, text='Results calculation')
#self.monty4.grid(row=4, column=0,  columnspan=6, padx=4,  pady=4,sticky="W")
# scrollb = ttk.Scrollbar(self.monty4)
# scrollb.grid(row=5, column=0, columnspan=4, pady=4)
# self.listbox = tk.Listbox(self.monty4,width=110,height=10)
# self.listbox.grid(row=5, column=0, columnspan=4, pady=2, padx=4)
# self.listbox.config(yscrollcommand=scrollb.set)
# scrollb.config(command=self.listbox.yview)
## menuBar = Menu(tab1)
## self.win.config(menu=menuBar)
```

```
##fileMenu = Menu(menuBar, tearoff=0)
## menuBar.add_cascade(label="File", menu=fileMenu)
## helpMenu = Menu(menuBar, tearoff=0)
## menuBar.add_cascade(label="Help", menu=helpMenu)
#def func_1(self):
#print("func_1")
#def func_2(self):
# print("func_2"
#def func_3(self):
# print("func_3")
#def func_4(self):
#print("func_4")
#def func_5(self):
#print("func_5")
# def func_6(self):
# print("func_6")
#def func_7(self):
# print("func_7")
#oop = OOP()
#oop.win.mainloop()
```

Қосымша Ә

Акт о внедрении

результатов диссертационной работы
Даркенбаева Даурена Кадыровича,
представленной на соискание ученой степени
доктора философии (PhD) по специальности
6D075100 – Информатика, вычислительная техника и управление

Настоящим подтверждаем, что результаты диссертационного исследования Даркенбаева Даурена Кадыровича на тему «Численное моделирование и разработка программного комплекса для обработки большого объема данных» обладают актуальностью, представляют практический интерес и были использованы в виде программного комплекса компанией ТОО «WINFIN Kazakhstan» для внедрения в банковскую систему бизнес-процесса принятия решения по ипотечному кредитованию физических лиц. Полученные результаты позволили точно определить платежеспособность физических лиц, что позволило повысить уровень надежности функционирования процесса принятия решения. Результаты протестированы и апробированы. В связи с банковской тайной наименование банка не раскрывается.

ТОО «WINFIN Kazakhstan» выражает признательность Даркенбаеву Даурену Кадыровичу за предоставленную возможность практического применения столь полезных результатов диссертационного исследования и надеется на активное продолжение его работ и нашего сотрудничества.

Генеральный директор
ТОО «WINFIN Kazakhstan» Терекбаева Ж.М.

Қосымша Б

ЭЕМ-ге арналған бағдарламалық авторлық құқық куәлігі

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК

2020 жылғы «28» ақпан № 8459

Автордың (авторлардың) жөні, аты, әкесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе):
ДАРКЕНБАЕВ ДАУРЕН КАДЫРОВИЧ

Авторлық құқық объектісі: ЭЕМ-ге арналған бағдарлама

Объектінің атауы: NOSQL ТЕХНОЛОГИЯЛАРЫН ЖӘНЕ НЕЙРОНДЫҚ ЖЕЛІЛЕРДІ ҚОЛДАНЫП ҮЛКЕН КӨЛЕМДІ ДЕРЕКТЕРДІ ӨҢДЕУ

Объектінің жасаған күні: 23.02.2020

Құжат түпнұсқасын https://www.kazpatent.kz/ru сайтының "Авторлық құқық" бөлімінде тексеруге болады https://copyright.kazpatent.kz

Подлинность документа возможно проверить на сайте kazpatent.kz в разделе «Авторское право» https://copyright.kazpatent.kz

Подписано ЭЦП

Абулкаиров Н.А.