TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Informaatikainstituut

Töö kood: IDK70LT

Oskar Liblik 144335

KLASSIFITSEERIMISE ALGORITMI ABIL E-ÕPPE SÜSTEEMIS TUDENGITE VÄLJALANGEMISE ENNETAMINE INFORMAATIKA AINE NÄITEL

Magistritöö

Juhendaja: Jekaterina Tšukrejeva
assistent
Kaasjuhendaja: Kristina Murtazin
lektor

Tallinn 2016

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Oskar Liblik

09.05.2016

Klassifitseerimise algoritmi abil e-õppe süsteemis tudengite väljalangemise ennetamine informaatika aine näitel

Annotatsioon

Antud töö peamiseks eesmärgiks on välja töötada algoritm, mis oskaks ennustada õppuri tulemuste ja kursuse külastatavuse abil tudengi õppeedukust. Tugevamate ja nõrgemate tulemustega õpilased klassifitseeritakse ning neile määratakse riskigrupid.

Esmalt vaadeldakse erinevaid masinõppe algoritme ning leitakse ülesande jaoks kõige sobivam algoritm. Valitud algoritmiks osutub ID3 algoritm, mille abil arvutatakse informatsiooni kasulikkus. Arendatava algoritmi sisenditena kasutatakse vaid kõige suurema kasulikkusega andmeid.

Magistritöö teine eesmärk on dokumenteerida töö nii, et seda oleks võimalik kohaldada kõikidel õppejõududel vastavalt nende kursusega.

Töö tulemusena töötatakse välja algoritm, mille abil on võimalik ennetada tudengite väljalangemist. Algoritm on võimeline tuvastama, millised tudengid kuuluvad riskigruppi ning võimaldab õppejõududel/ülikoolil suunata oma tähelepanu antud riskigrupi õpilastele.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 71 leheküljel, 9 peatükki, 23 joonist ja 2 tabelit.

Predicting students dropout by using classification algorithm on the example of informatics in the e-learning system

Abstract

The main purpose of this work is to develop an algorithm that can predict student success based on their results and activity. Students are classified according to the results and course visits and they are assigned risk groups.

At first, other classification algorithms are analyzed. After that, the most suitable algorithms to solve the task are chosen. Based on the analysys, ID3 algorithm was closest to achive the goals.

The information gain calculations in ID3 algorithm are used to calculate the most relevant attributes from dataset to predict students failure or success. After the information gain is calculated, the most beneficial data is used in algorithm to calculate the risk groups for each student. Algorithm is developed in Java programming language using object oriented principles.

Algorithm is closely related to ID3, as it is also a decision tree algorithm. The decisions are made based on if-else statements for each input.

Another aim of the work is to document the algorithm in a way that allows it to be applied to other datasets. Current dataset is taken from Moodle e-learning system and contains Moodle logs and student grades.

As a result, it is possible to predict students that may dropout from the course based on their risk group that has been assigned by the algorithm. Algorithm is open source and can be used by anyone.

The thesis is written in Estonian and contains 71 pages of text, 9 chapters, 23 figures and 2 tables.

Lühendite ja mõistete sõnastik

Avatud õppija mudel	***Open learner model (OLM)*** "Õppija mudelid sisaldavad ning uuendavad dünaamiliselt õppija õppimise kohta järgnevat informatsiooni: praegused teadmised, oskused, nõrkused, eesmärgid jne." (Dr. M. Kickmeier-Rust 2014)
Kursuse juhtimise süsteem	***Course management system (CMS)*** Rakendused mille abil õppejõud saavad edastada õppuritele informatsiooni, lisada materjale, koostada ülesandeid ning teadmiste kontrolle, algatada arutelusid ning juhtida kaugõppe klasse läbi interneti. (Milani ja Mazza 2004)
Massiivsed avatud internetikursused	***Massive open online courses (MOOC)*** Tasuta veebipõhine kaugõppe programm, mis on mõeldud erinevatest geograafilistest asukohtadest pärit suurtele kuulajaskondadele. (TechTarget 2013)

Sisukord

Jooniste nimekiri

Tabelite nimekiri

1. Sissejuhatus

1.1 Taust ja probleem

Tallinna Tehnikaülikoolis õppis 2015 aastal üle 11000 üliõpilase. Samal aastal on õpingud katkestanud veidi üle 2800 üliõpilase. Katkestanute arv ei ole lähima 10 aasta jooksul nii kõrge olnud. Õpingud katkestab keskmiselt 25% tudengitest. Infotehnoloogia teaduskonnas on 2015 aastal katkestanute protsent lausa 27. (Õppeinfosüsteem 2016)

E-õppe keskkonnad koguvad tudengite kohta erinevaid andmeid. Hetkel ei ole tudengite õppevõimekuse ning saavutuste põhjal järeldusi tehtud. Andmeid analüüsides võib ennetustöö tulemusena väljalangevuse protsenti vähendada.

Töö motivatsiooniks on Kristina Murtazin ja Jekaterina Tšukrejeva doktoritöö, kus nad analüüsivad võimalusi intelligentse e-õppe süsteemi loomiseks, mis suudaks tudengite õppetulemusi semestri jooksul ennustada. Õppetulemuste ennustamiseks on doktorantidel vaja koostada algoritm, mis hindab õppija tegevusi õppetöö ajal.

1.2 Ülesande püstitus

Tudengite väljalangevuse ennetamiseks, tuleb kaardistada ja analüüsida, miks on mõne tudengi tulemused paremad kui teistel ning töötada välja algoritm mis prognoosib tudengite vahetulemuste, testide sooritamise ning õppetööst osavõtu aktiivsuse põhjal tudengi riskigrupi. Samuti eristatakse edukamaid õpilasi ja nendele pakutakse võimalus mahajäänumaid abistada.

Ennetamise algoritmi väljatöötamiseks tuleb esmalt analüüsida olemasolevaid algoritme ning uurida kas neid saab antud töös kasutada. Kui olemasolevatel lahendustel ilmnevad puudused, siis tuleb arendada uus algoritm.

Programm peab leidma riskantsemad tudengid, kes võivad aine katkestada. Algoritm tuleb dokumenteerida, et seda oleks võimalik kõikidel huvilistel kasutada.

1.3 Metoodika

Kuna soovitakse leida kindlasse gruppi kuuluvaid tudengeid, siis tuleb kasutada klassifitseerimise algoritmi. Klassifitseeriv algoritm peab iga tudengi tulemused rühmadesse jagama. Algoritmi tulemuse põhjal on näha, kellele ülesanded probleeme valmistasid ning kus võis tekkida tudengi mahajäävus.

Läbi Moodle õppekeskkonna on andmeid kogutud pikka aega. Küll aga ei ole nende põhjal veel järeldusi tegema hakatud ja antud andmeid ei ole varem analüüsitud. Lõputöö eesmärkide saavutamiseks on vaja esmalt kaardistada ning analüüsida masinõppe algoritme. Masinõppimine on teadusvaldkond, mille eesmärk on välja töötada ennustusi tegevaid algoritme. Seejärel tuleb koostada algoritm, mis õppuri andmed riskigruppidesse klassifitseerib ning analüüsi tulemusel järeldusi teeb.

Moodle keskkonnas on hetkel võimalik õppurite ning kursuse ressursside analüüsimiseks seadistada GISMO blokk. GISMO kuvab näiteks graafiliselt välja, milliseid kursusele lisatud ressursse tudengid vaatavad, millised testid neil tehtud on, mis tulemustele testid on sooritatud, kui tihti tudengid külastavad Moodle keskkonda ja nii edasi. Kuna GISMO blokk kasutab palju serveri mahtu, siis seda saab näha hetkel maksimaalselt kolme kuu kohta. Tulevikus on plaan GISMO üldse sulgeda. See on ka üks põhjus, miks antud algoritmi loomine on väärtuslik ja vajalik.

Klassifitseerimine on mudeli leidmise protsess mida kasutatakse andmete jagamiseks erinevatesse klassidesse kindlate kitsenduste alusel. Teisisõnu võib öelda, et klassifitseerimine on andmete üldistamine sarnasuste ja erinevuste põhjal. (Raj Kumar 2012)

Töös klassifitseeritakse Moodle andmete põhjal tudengite sooritused järgnevate kriteeriumite alusel:

1. Kui tihti tudeng Moodlet külastab (üliõpilase aktiivsus)
2. Millised on õppuri testide tulemused
3. Millised on tudengi kodutööde tulemused

Algoritm koostatakse programmeerimiskeeles Java kasutades objektorienteeritud lähenemist.

1.4 Ülevaade tööst

Töö algab sissejuhatusega probleemi ja selle kirjeldusega; põhjendatakse ära, mis ajendas autorit antud tööd kirjutama. Välja tuuakse osapooled, keda lahendus huvitada võib.

Seejärel seatakse uurimuse eesmärgid ning kirjeldatakse lühidalt, kuidas neid eesmärke plaanitakse saavutada.

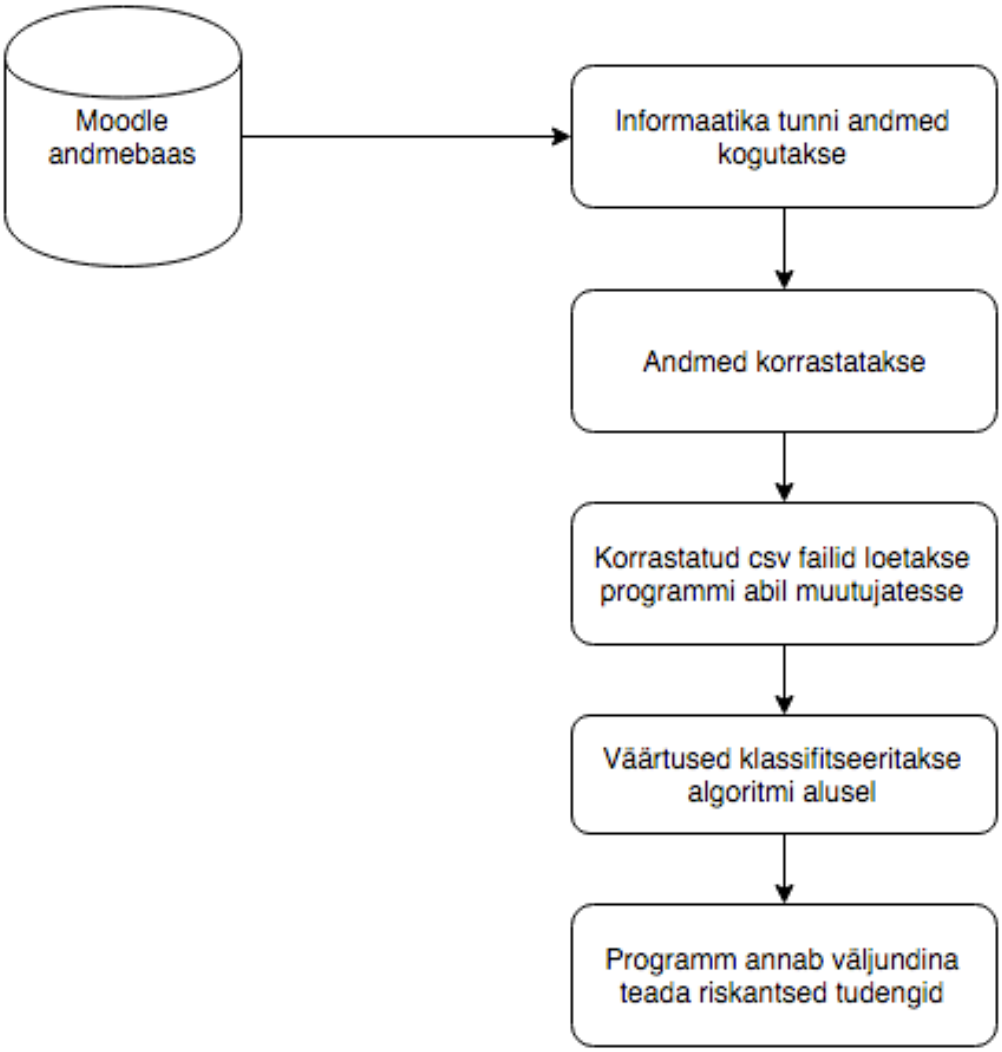
Neljandas peatükis tutvustatakse Moodle keskkonda, kust on võetud ka antud töös analüüsitav valim. Samuti vaadeldakse Moodlele sarnast Soomo süsteemi ning MOOC kursusi üldiselt

Viiendas peatükis grupeeritakse ja analüüsitakse olemasolevaid algoritme. Algoritmide gruppidest valitakse välja sobivaimad algoritmid eelnevalt kirjeldatud eesmärkide saavutamiseks ning kuuendas peatükis kirjutatakse täpsemalt valitud algoritmidest.

Võttes arvesse töö jooksul kogutud informatsiooni ning teoreetilisi aluseid, koostatakse programmi loogikareeglid. Loogikareeglite põhjal on võimalik koostada pseudokood programmi paremaks mõistmiseks. Lõplik algoritm kirjutatakse Java programmeerimiskeeles tuginedes pseudokoodile.

Paljudele olukordadele on lisatud paremaks mõistmiseks ka ekraanipildid. Uurimustöö lõppeb kokkuvõttega, kus tehakse järeldusi saavutatud eesmärkide osas.

Järgneval joonisel (joonis 1) on esitatud algoritmi tööpõhimõte. Jooniselt on näha, et algoritmi sisendandmed tulevad Moodle andmebaasist. Sisendandmeteks on informaatika aine hindamistabel ning Moodle logid. Andmed korrastatakse, ehk kõik ebavajalik, nagu näiteks kommentaarid eemaldatakse tabelist. Seejärel loeb programm andmed sisse, klassifitseerib väärtused ning annab väljundina teada millised tudengid kuuluvad riskigruppi.



Joonis 1 - Üldine ülevaade

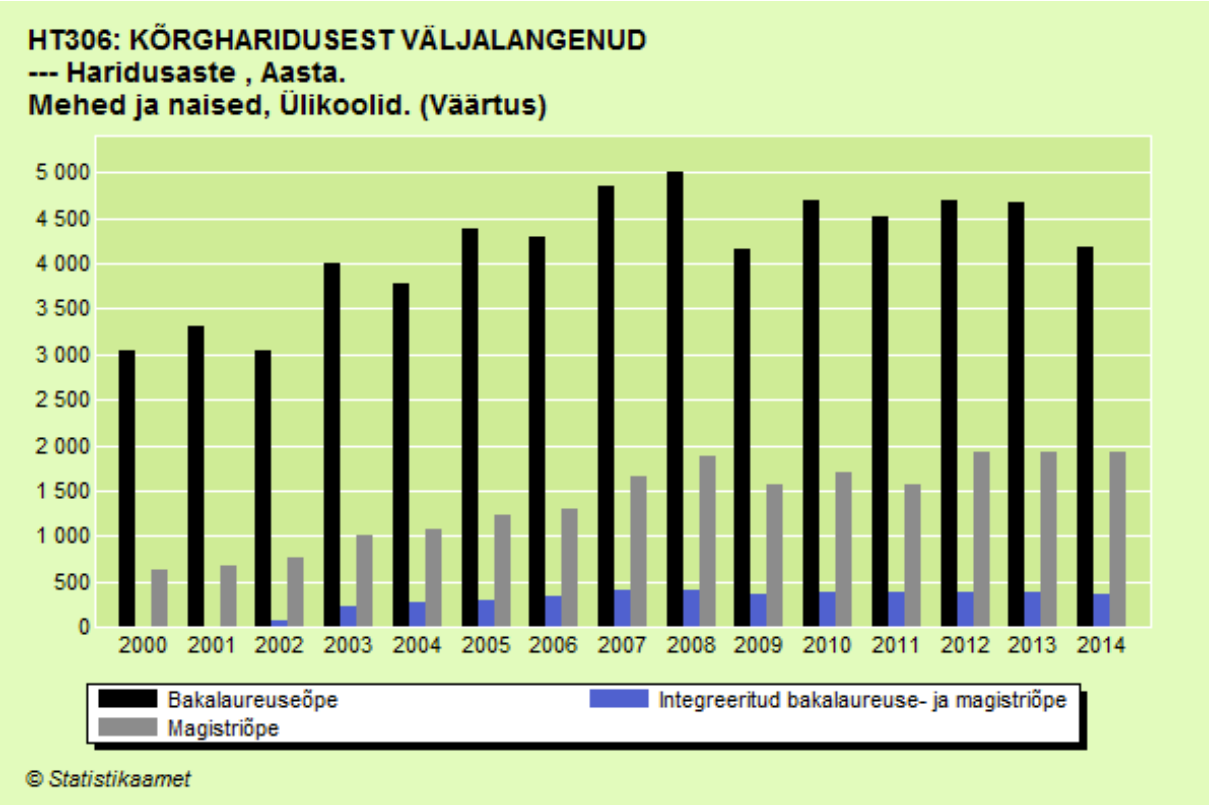
2. Probleemi kirjeldus

Töö aluseks on võetud Carlos Márquez-Vera1, Alberto Cano, Cristobal Romero, Amin Yousef Mohammad Noaman, Habib Mousa Fardoun ning Sebastian Ventura koostatud juhtumiuuring teemal "Andmekaevandamise abil kooli katkestamise ennetamine keskkooli õpilaste näitel". Autorid on seisukohal, et kooli katkestamine on suur probleem ning soovivad seda ennetada kursuse keskel. Õppijate väljalangemist saab ennustada juba 4-6 nädalal pärast õpingute algust. Antud juhtumi puhul analüüsiti õpilaste kohalkäimisi, sotsiaalset käitumist ning korraldati küsitlusi. (Márquez-Vera, et al. 2015)

Töös jõuti järeldusele, et klassifitseerimisalgoritmi põhjal saab ennustada õpilaste väljalangemist õppeprogrammidelt. Algoritmis jagati õpilased teatud kriteeriumite alusel klassidesse. Pärast klassifitseerimist saab õpilaste õppeedukuse kohta hinnanguid anda. Selles töös analüüsiti ka kriteeriume, mida e-õppe süsteemidest ei ole võimalik kätte saada. Näiteks õpilaste alkoholi tarbimist ei suuda Moodle süsteem tuvastada.

Väljalangemise ennetamiseks on autorite sõnul kõige sobivamad meetodid lapsevanemate informeerimine ja kaasamine, tugiisikute leidmine ning personaalsete õppeplaanide koostamine, trahvide või muude sanktsioonide rakendamine ja nii edasi.

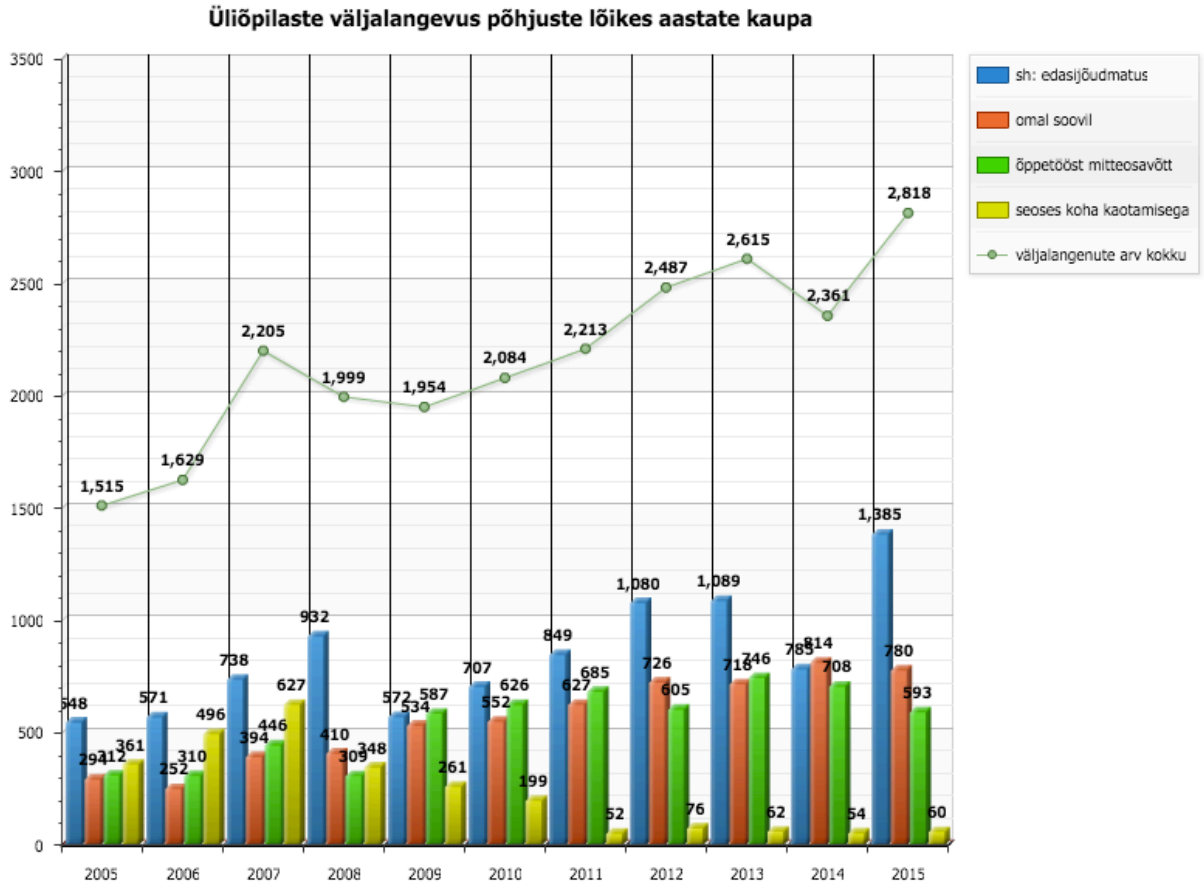
Eesti ülikoolides on väljalangevus suureks probleemiks. Alates aastast 2000 kuni aastani 2008 on väljalangevus pidevalt kasvanud ning seejärel stabiliseerunud. Olukorda illustreerib järgnev diagramm statistikaameti kodulehelt (Joonis 2).



Joonis 2 - Kõrgharidusest väljalangenud (Statistikaamet 2014)

Joonisel 2 on kuvatud kõikidest Eesti ülikoolidest väljalangenud tudengid erinevatel tasemeõpetel. Andmed on esitatud iga aasta kohta ning kõige värskemad andmed on statistikaameti kodulehel 2014 aastani.

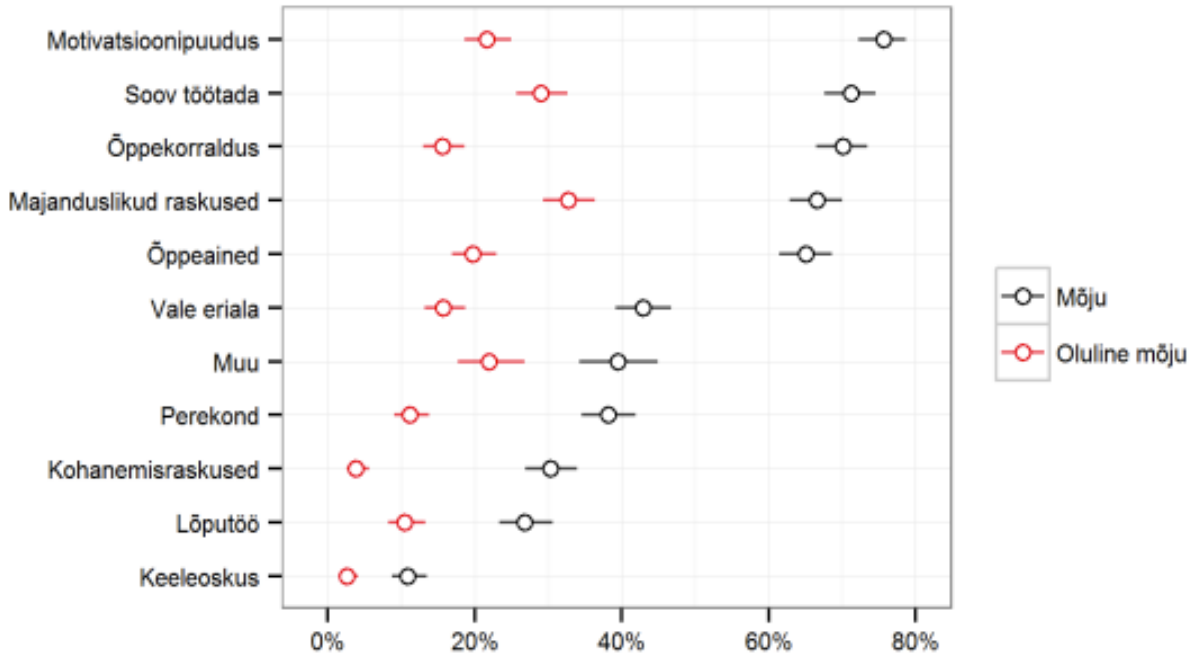
Tudengite suur väljalangevus on probleemiks ka Tallinna Tehnikaülikoolis. Järgneval pildil on näha Tallinna Tehnikaülikooli tudengite väljalangevus põhjuste lõikes aastate kaupa.



Joonis 3 Üliõpilaste väljalangevus põhjuste lõikes aastate kaupa (TTÜ Õppeinfosüsteemi statistika 2016)

Statistikast on näha, et kõige suurem väljalangevuse põhjus on edasijõudmatus. Kui tudengitel on raske õppida siis saab väljalangemist ennetada, pakkudes tudengitele tugiisikuid, mentoreid või suunates nad õppealasele nõustamisele.

Õpingute katkestamist on analüüsitud ka ainult infotehnoloogia erialadel. Tudengite arvates on kõige suurem katkestamise põhjus motivatsioonipuudus. (Eesti rakendusuuringute keskus Centar 2015) Olukorda illustreerib järgnev joonis.



Joonis 4 - Katkestanute osakaal, kelle hinnangul tegur avaldas mõju õpingute katkestamisel (Eesti rakendusuuringute keskus Centar 2015)

Joonisel on näha, et kõige rohkem hindasid infotehnoloogia tudengid katkestamise põhjuseks motivatsioonipuudust ning soovi töötada. Kuna motivatsioonipuudus on kõige kõrgemalt hinnatud, siis seda saab ennetada õppealasel nõustamisel või õppejõuga vesteldes. Seetõttu olekski vaja kaardistada kõige riskantsemad tudengid.

Väljalangemise ennetamise kohta on palju teooriaid, kuid sobivat algoritmi, mille tulemusel oleks võimalik ennetada tudengite väljalangemist, ei ole veel varem Moodle süsteemi jaoks arendatud. Sarnaselt eelpool mainitud juhtumuuringule, proovitakse leida väljalangemise riskiga tudengid. Algoritmi sisenditeks kohandatakse Tallinna Tehnikaülikooli Moodlest võetud informaatika õppeaine logid ning õppeaine tulemused. Järgnevas peatükis on välja toodud olemasolevate lahenduste puudused.

2.1 Olemasolevate lahenduste puudused

Hetkel ei ole võimalus õppejõududel automaatselt analüüsida, kes ja kui tihti kursust külastab. Õppejõul on võimalus tundides kohalkäimist kontrollida, kuid puudub võimalus näha, kes külastavad kursuse materjale e-õppe keskkondades. Eriti oleks seda vaja teada kaugõppe õppurite juhendajatele.

Sarnaseid andmeanalüüse on välisülikoolides koostatud ajalooliste andmete põhjal mis ongi nende üheks kõige suuremaks puuduseks. Ennetamiseks oleks vaja teada jooksva semestri käigus õppurite aktiivsust. Mitteaktiivsete tudengite staatus saab kaardistada juba hetkel, kui tudengi õppetööst osavõtt muutub harvemaks. Seda on võimalik saavutada siis, kui anda jooksval semestril programmile sisendiks Moodle logid. Siis on reaalajas näha tudengid, kes kursust kõige vähem külastanud on.

Hetkel on Moodle keskkonnas võimalik kasutada GISMO blokki. GISMO bloki peamiseks puuduseks on see, et andmeid on võimalik vaadata maksimaalselt kolme kuu kohta. Tulevikus on oht, et GISMO blokk eemaldatakse Moodle lisade alt üldse ära. Pärast GISMO kaotamist oleks eriti vajalik, kui õppejõud või akadeemilised nõustajad saaksid infot tudengite kohta, kes on kursusel väheaktiivsed ning kellel on aine katkestamise oht. Selle tulemusena võib ennustada ka ülikoolist väljalangejaid.

2.2 Kellele on lahendus suunatud

Doktorandid Kristina Murtazin ja Jekaterina Tšukrejeva analüüsivad enda doktoritöö raames võimalusi luua intelligentne e-kursuste juhtimise süsteem. Doktorandid uurivad kuidas analüüsida ning teha järeldusi tudengite õppeprotsessi käigus saadud tulemuste põhjal. Töö tulemusena soovitakse kaardistada sisendandmed, need analüüsida ning koostada algoritm, mis graafilisel kujul kaasaegsel kasutajaliidesel kuvab välja tudengitele nende õppeprotsessi. Antud magistritööst on doktorandid väga huvitatud, kuna selle tulemusel valmib algoritm, mis tudengite õppetulemusi ennustada suudab.
Tulevikus on lahendusest kasu nii tudengile kui ka õppejõududele. Lisaks võib lahendus huvi pakkuda Moodle arendajatele.

Õppejõud saavad vaadata kellel tudengitest esineb nende kursusel probleeme. Eriti raske on jälgida kaugõppurite tööd. Tudengite riskigrupi põhjal saavad õppejõud pakkuda õppuritele järeleaitamise tunde. Veel kuvatakse õppejõududele kursuse kõige edukamad õpilased.

Selleks, et vähendada tudengite väljalangevust, võiksid edukamad õpilased õppejõudu abistada ning nende töökoormust vähendada, aidates mahajäänumaid tudengeid. Tudengitel, kes ei ole nii edukad, oleks võimalus saada abi ning järeleaitamise tunde kas õppejõududelt või kursusel edukamatelt tudengitelt.

Samuti võib algoritmile ligipääsu võimaldada ka õppenõustajatele, kes saavad tudengeid algoritmi tulemuste põhjal nõustamisele kutsuda. Psühholoogiline nõustamine on tasuta teenus üliõpilaste toetamiseks õpingute käigus tekkivate raskuste korral.

Moodle arendajatele võib huvi pakkuda lahenduse integreerimine Moodle keskkonda, et õppejõududel oleks seda veelgi lihtsam ja mugavam kasutada. Hetkel kasutuses olev GISMO blokk vajab liiga palju kõvaketta ruumi graafikute salvestamiseks ning loodav algoritm lahendaks selle probleemi.

3. Töö eesmärgid

Antud magistritööl on kokku kolm eesmärki. Esimene nendest on peaeesmärk, milleni jõuab läbi kahe kõrvaleesmärgi.

Õppeainete katkestamine on tudengite seas suureks probleemiks. Töö peamine eesmärk on ennetada tudengite väljalangevust informaatika aine tulemusi analüüsides. Informaatika aine asub Moodle keskkonnas, kus asuvad ka algoritmiks kasutatavad sisendandmed - hinded ja logid.

Antud tööl on kaks täiendavat lisaeesmärki. Mõlemad eesmärgid on rakendusliku iseloomuga ning nendest on kasu nii tudengitele kui ka õppejõududele. Kõrvaleesmärgid toetavad peaeesmärgi täitmist ning on välja toodud järgnevates peatükkides.

3.1 Õppeainete katkestamise ennetamine

Töö peamine eesmärk on vähendada õpingute katkestanute arvu läbi õppeainete katkestamise ennetamise. Selle saavutamiseks on vaja semestri kestel õppetulemusi hinnata ning riskantsemad tudengid klassifitseerida. Samuti klassifitseeritakse edukamad tudengid, kes võivad õppejõudusid aitama hakata. Õppurite aktiivsuse põhjal saab õppejõule riskantsemate tudengite kohta jooksvalt infot anda. Semestri keskel saadud informatsiooni põhjal on õppejõul võimalus ennetada tudengi väljalangemist kursuselt.

Eriti kasulik on lähenemine kaugõppurite õppejõududele, kuna neil puudub ülevaade kes nende kursusel aktiivsemad on. Tihti ei pruugi kaugõppe õppejõud enne semestri lõppu teadagi, et mõni tudeng on nende ainel õppimise katkestanud.

Õpingute katkestamise ennetamine on ühiskonnas oluline probleem, sest õpingutega kaasnevad kulud nii riigile kui ka tudengile. Üheks enamlevinud põhjuseks miks paljud tudengid ülikooli pooleli jätavad on see, et tööturg ei väärtusta kõrgharidust tõendavat diplomit piisavalt. Selle tulemusena kaob tudengitel motivatsioon õppida ning nad ei pruugi enam kursustelt osa võtta. Kui selle probleemiga ei tegeleta, siis raiskab riik asjatult raha tasuta kõrghariduse toetamiseks. Antud töös saab lahendada just kursusest osavõtu probleemi ning motiveerida algoritmi väljundist lähtuvalt tudengeid.

3.2 Algoritmi väljatöötamine

Selleks, et lahendada peamine eesmärk, on vaja välja töötada algoritm, mis oskab lugeda Moodle õppurite aktiivsust ja tulemusi ning selle põhjal tudengeid klassifitseerida. See on ka magistritöö teine eesmärk - algoritmi väljatöötamine.

Algoritm kirjutatakse programmeerimiskeeles Java ning see sarnaneb otsustuspuu ülesehitusele (kui-siis reeglid). Algoritmi saab käivitada kõikidel platvormidel. Sisendiks tuleb programmile ette anda jooksva semestri hinnete tabel ning Moodle logid.

3.3 Algoritmi dokumenteerimine

Töö kolmas eesmärk on algoritm piisaval detailsusastmega dokumenteerida, et seda oleks lihtne kasutada ja erinevatele valimitele rakendada. Algoritmi lähtekood jäetakse avatuks. Algoritmi lähtekood asub Github keskkonnas ning sellele on viidatud töö lõpus lisades (Lisa 1).

Tulenevalt sellest on kõigil huvilistel võimalik antud algoritmi kasutada ning soovi korral edasi arendada või kohandada. Programmi võib tulevikus liidendada Moodle keskkonnaga, et seda oleks mugavam kasutada. Töö lähtekoodi on lisatud kommentaarid programmi paremaks mõistmiseks.

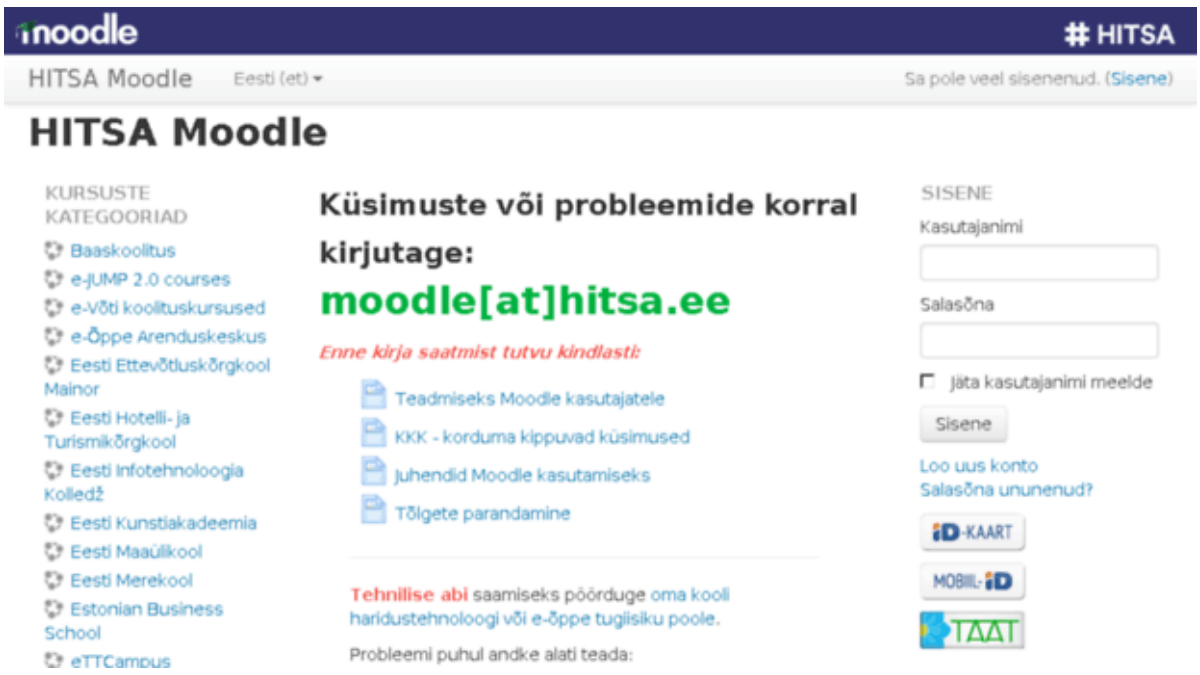
4. E-õppekeskkondade ülevaade

Kursuse juhtimissüsteemid (i.k. course management systems) on rakendused mille abil õppejõud saavad edastada õppuritele informatsiooni, lisada materjale, koostada ülesandeid ning teadmiste kontrolle, algatada arutelusid ning juhtida kaugõppe klasse läbi interneti. (Milani ja Mazza 2004)

Kursuse juhtimissüsteeme võib kutsuda ka e-õppekeskkondadeks. Järgnevates alapunktides on välja toodud erinevad e-õppekeskkonnad ning kirjeldatud nende statistika koostamise tehnikaid tudengite andmete põhjal.

4.1 Moodle

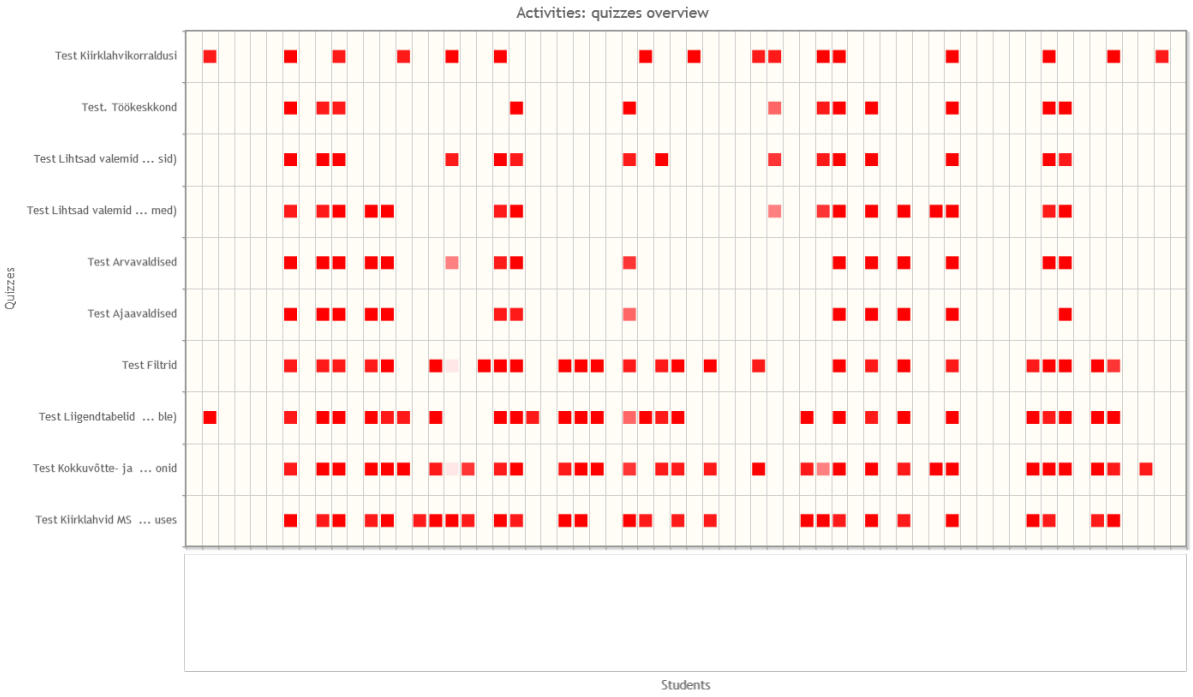
MOODLE (i.k. Modular Object-Oriented Dynamic Learning Environment) defineeritakse kui kursuse juhtimissüsteemi (CMS). (FORMATEX 2006) Kuna Tallinna Tehnikaülikoolis on Moodle üks kasutatuimaid kursuse juhtimissüsteeme, siis kasutatakse antud töös Moodlest saadud andmeid. Näidisandmed, mida kasutatakse algoritmi testimiseks, on informaatika kursuse 2015 aasta sügissemestri tulemused (valimis 62 inimest). Järgneval joonisel on näha Moodle ekraanipilt.



Joonis 5 Moodle ekraanipilt (Moodle hitsa 2016)

Avatud õppija mudelit (OLM) võib vaadelda kui õppija mudelit, mis kuvab kasutajale välja süsteemi poolt kogutud andmed inimloetavas keeles. (Kay ja Bull 2015)

Hetkel koostab Moodle keskkond GISMO blokki kasutades avatud õppija mudeli, kus on erinevaid andmeid tudengi aktiivsuse kohta. Järgneval pildil on välja toodud üks näidis GISMO bloki mudelist.



Joonis 6 GISMO blokk (Moodle hitsa 2016)

Joonisel on näha GISMO poolt koostatud graafiline kujutis, kus vasakul on testide nimed ning all on tudengite nimed (anonüümsuse huvides on nimed pildilt kustutatud). Iga tudengi juures on näha milline test tal sooritatud on ning mida tumedam kastike nime juures, seda parem on tulemus.

Mudeli miinuseks on see, et andmeid saab vaadelda vaid kolme kuu lõikes. Heaks analüüsiks oleks vaja terve semestri aineid.

Mudelit analüüsides on võimalik kindlaks teha õppija:

- nõrkused
- tugevused

- oskused

Antud magistritöös koostatakse õppija mudel Moodle logide ning tudengi tulemuste põhjal informaatika aines. Avatud õppija mudelisse saadakse andmed Moodlest ning hinnetelehelt. Kõiki andmeid ei ole võimalik Moodle kaudu kätte saada, kuna õppimine on suures osas interaktiivne tegevus. Töös koostatud avatud õppija mudel ei ole graafiline, nagu GISMO blokk.

Seoses e-õppevormi kasutamisega teistes ülikoolides, on ka mujal välja arendatud erinevaid algoritme õppurite aktiivsuse hindamiseks ning väljalangevuse ennetamiseks. Näiteks on Soomo e-õppe süsteemis sarnaselt Moodle GISMO blokile võimalik näha interaktiivseid jooniseid tudengite õppeedukuse kohta. Soomo on kasutusel Ameerika Ühendriikide kõrgkoolides. Sarnast lähenemist on kasutanud ka MOOC süsteemid. Mõlemast lähenemisest tuleb antud peatükis pikemalt juttu.

4.2 MOOC süsteemides väljalangemise ennustamine

Massiivsed avatud internetikursused ehk MOOC (i.k. Massive Open Online Courses) on viimasel ajal väga palju populaarsust kogunud. MOOC platvormid on näiteks Coursera, edX ning Udacity. Need platvormid sisaldavad endas maailma suurimaid e-kursusi ning täiendkoolitusi. Süsteemi vahendusel on õppejõududel võimalus enda e-kursusi lisada kõigile tasuta vaatamiseks. MOOC on loonud ka enda mudeli, kuidas ennustada õppureid, kes kursuse katkestada võivad. (openeducation 2014)

Õppuritel on erinevad eesmärgid ja kavatsused mis muutuvad ajas. Otsuse õpingud pooleli jätta võivad tudengis esile kutsuda erinevad faktorid tema elus. Laialt võib faktorid jagada kaheks: sisemised motivatsioonifaktorid (õppuri soov ja tahe) ning välimised motivatsioonifaktorid (õppetöö välised faktorid nt. kodune elu). Väliseid motivatsioonifaktoreid on peaaegu võimatu ennustada. MOOC süsteemis toimubki ennustamine seetõttu sisemiste faktorite põhjal. (openeducation 2014)

Kui tudeng ei külasta kursuse materjale siis tema riskigrupp suureneb. Süsteem on üles ehitatud nii, et kui õppur on katkestamise ohus, siis tekib tema nime kõrvale punane märge.

Ennustusalgoritm otsustab, kellele punane märge tekib, järgnevalt:

1. Arvutatakse kindlate tegevuste põhjal punktid igale õppurile. Näiteks vaadeldakse testide tulemusi ning õppurite videote vaatamise harjumusi. Kui õppur vaatab videoid mitmeid kordi või kerib vaatamise ajal videot pidevalt tagasi, siis on ta suuremas riskigrupis.

2. Iga tegevuse põhjal otsustatakse, kas punktid on piisavad. Etteantud punktidele mittevastavatele tudengitele tehakse punane märge. Punktidele mittevastavus võib tekkida näiteks kursuse lehe harva külastamisest.

3. Kui mõne indikaatori punktid ei ole piisavad, lisatakse õppurile punane märge. Näiteks kui tudeng ei ole külastanud kursuse lehte kolm nädalat. (openeducation 2014)

Kõige olulisem kriteerium väljalangevuse hindamiseks on kursuse külastamise aktiivsus mida mõõdetakse kursuse lehe külastamisega või ressursside hankimisega kursuse lehelt. (openeducation 2014) Antud mudel suudab 2 nädalat enne tudengite väljalangemist kindlaks teha 60% väljalangejatest. E-õppe puhul mõeldakse väljalangemise all tudengeid, kes ei ole kursuse materjale külastanud üle ühe kuu ja kes on vaadanud alla poolte kursuse videotest.

Kuna MOOC kursused koosnevad videoloengutest, siis on väga oluline jälgida tudengite käitumist video vaatamisel. Suuremad MOOC süsteemid nagu näiteks Coursera ning edX logivad video vaatamisel väga palju andmeid. Näiteks logitakse õppurite video vaatamise aeg, pauside tegemise aeg, mängimise kiirus ja nii edasi.

Hästi populaarsed on ka foorumite kasutamine õppetööks. Õppuritel on võimalus postitada küsimusi ning saada vastusi läbi foorumi. Massiivsetel kursustel on õppejõududel raske kõikide õppuritega personaalselt kontakteeruda ning seepärast ongi üks parimaid info saamise kanaleid MOOC süsteemides foorum.

Moodles ei ole foorumid veel nii laialt levinud. Samuti ei kasutata videomaterjale päevases õppes (tihtipeale isegi mitte kaugõppes). Seetõttu on raske antud lähenemist üks ühele kohandada Moodle jaoks vastavaks.

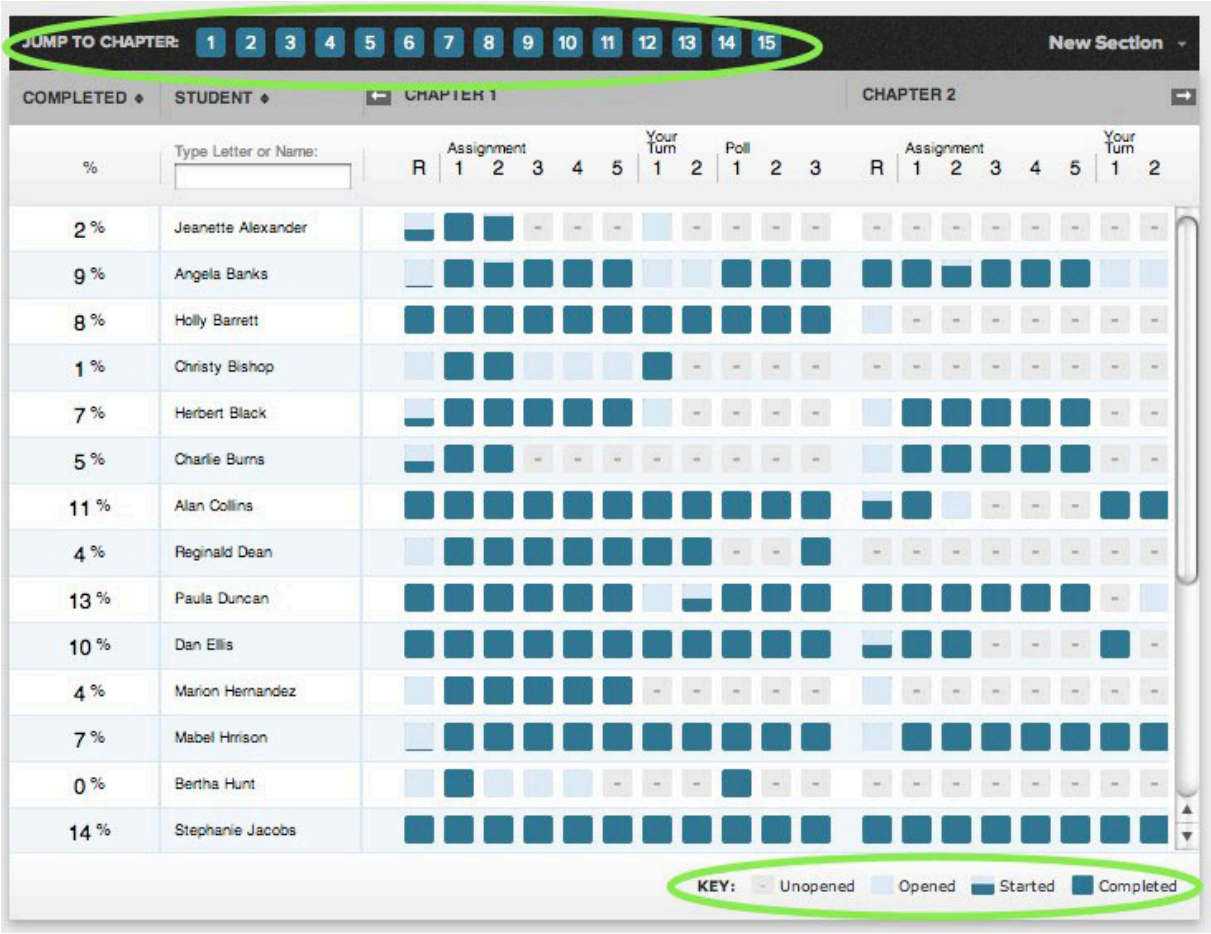
Antud ennustusmudel on süsteemi sisse ehitatud ning ei ole avatud lähtekoodiga. Seega on antud süsteemi Moodles kasutamine võimatu. Küll aga andis antud lähenemisega tutvumine kinnitust, et sellist ennustusalgoritmi oleks vaja ka Moodles kasutada.

4.3 Soomo näitel e-õppe riskifaktorite analüüs

Kolumbia ülikooli teadlased on seisukohal, et tudengitel kes õpivad e-kursustel võib olla raskem hakkama saada, kuna neil puudub võimalus näost näkku tundides informatsiooni hankimisele. Samuti puudub juhendajatel kursusel osalejate probleemidest ülevaade. Kaugõppurite katkestamise protsent on suurem kui täiskoormusega õppuritel. Seetõttu on hakatud e-õppe riskifaktoreid koguma ja hindama. (Baker, et al. 2015)

Kõige rohkem huvitavad uurijaid külastatavuse arv, foorumis osalemise aktiivsus, ülesannete esitamise arv ning testide lahendamise aeg. Uuring viiakse läbi Soomo e-õppekeskkonnas. Soomot kasutab üle 100 ülikooli. (Baker, et al. 2015) Soomo keskkond sarnaneb Moodle omale.

Soomo keskkonnas on olemas väga head tudengeid jälgivad blokid, mis sarnanevad Moodles kasutatavatele GISMO blokkidele. Kursuse analüüsi tulemused kuvatakse juhendajatele ning see on esitatud järgneval ekraanitõmmisel (Joonis 7).



Joonis 7 Soomo analüüsi tulemused (Soomo learning 2016)

Joonisel on kujutatud Soomo süsteemi tudengite nimekiri ning iga tudengi juures on testide sooritamise kohta informatsioon. Valgete kastidega märgitakse tudengid, kes ei ole testi sooritanud, sinistega need kes on testi sooritanud ning sinise valgega märgitakse tudengid kellel on testi sooritamine pooleli. See annab juhendajatele hea ülevaate kursusel toimuvast.

Kõige olulisem on hinnata tudengite edasijõudmist kohe kursuse alguses. Uurimustöö autorid on arvamusel, et mahajäänud tudengeid on võimalik järele aidata, kui nende probleemid tehakse kindlaks esimeses semestri pooles. (Baker, et al. 2015) Seega teine pool semestrist on võimalik nendega veel tegeleda.

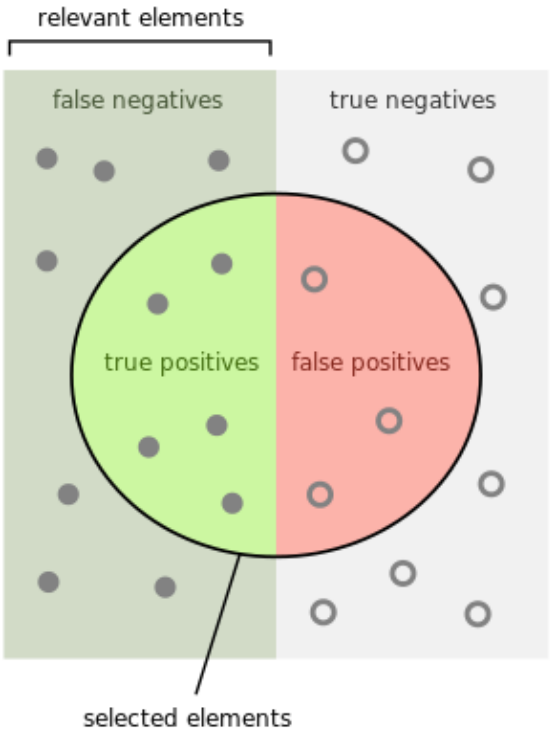
Selleks, et analüüsida andmeid ja identifitseerida õpilased, kellega tuleks tegelema hakata koostatakse Soomo keskkonna andmetele põhinedes täpsuse-täielikkuse graafik, kus päevade arv, mil tudengid kursuse materjale külastasid, on N. Kanooniline päevade N arv on 7. Riskantseteks hinnatakse tudengeid, kes said tulemuse alla 73%. Päevade ja tulemuste suhte kohta on koostatud graafik (Joonis 8). Täpsus ja täielikkus (i.k. precision and recall) on arvutatud järgnevalt:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

(Baker, et al. 2015)

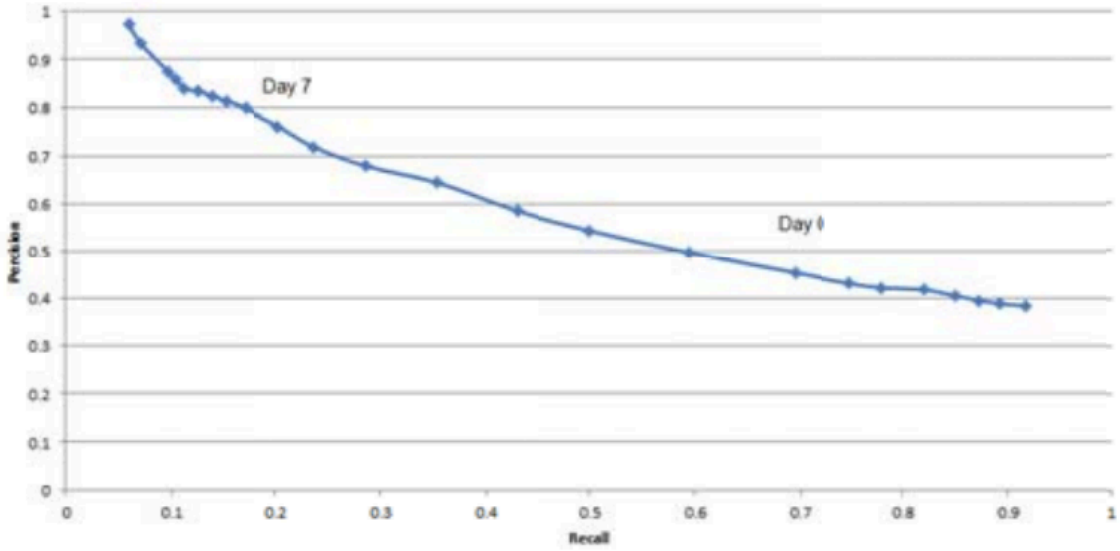
Valemis täpsus ehk "Precision" on teiste sõnadega vaste sellele, kui kasulikud olid tulemused ning täielikkus ehk "Recall" näitab kui täielikud andmed olid. Valemeid illustreerib paremini järgnev joonis.



Joonis 8 - Täielikkus ja täpsus (Walber 2014)

Täpsus näitab kui palju selekteeritud objektidest on asjakohased. Täielikkus näitab, kui palju asjakohastest objektidest on selekteeritud. Joonisel 8 on asjakohased elemendid märgitud vasakul pool. Selekteeritud elemendid on keskel ringi sees.

Joonis 9 näitab kui täpne oli tudengi hinde ennustamise tulemus. Nagu eelpool mainitud, otsiti tudengeid, kelle tulemus on alla 73% ning vaadeldi kas nad on kursuse materjale N päeva jooksul vaadanud. Vasakul skaalal on täpsus ning paremal täielikkus.



Joonis 9 - Alla 73% tulemuse ennustamise täpsus (Baker, et al. 2015)

Uurimustöös jõutakse järeldusele, et kursuse tihe külastamine on korrelatsioonis õppurite edukusega. Samuti mängib rolli testide õigeks ajaks esitamine ning testide tulemused. Kõiki neid parameetreid saab kasutada ka uue algoritmi ülesehitamisel.

Uurimuse suurim miinus on see, et kasutatakse e-õppesüsteemi Soomo, mis ei ole Eestis populaarsust kogunud. Kuna ei ole täpselt teada milliseid andmeid on võimalik Soomo kaudu kätte saada, siis ei ole võimalik sama algoritmi rakendada Moodle keskkonnale.

Soomo süsteemi algoritm on teoreetiliselt Moodlele väga sarnase lähenemisega, kuid see on arendatud Soomo e-õppe süsteemi jaoks. Antud algoritmi kohaldamine võib minna keerukamaks kui uue algoritmi arendamine spetsiaalselt Moodle jaoks.

5. Masinõppe algoritmide kirjeldus

Antud peatükk defineerib masinõppe ning vaatleb millised algoritmid on masinõppe algoritmid. Peatükis on lühidalt kirjeldatud masinõppe algoritme ning need gruppidesse jagatud. Kuuendas peatükis analüüsitakse valitud masinõppe algoritmide käitumist lähemalt.

Masinõpet võib defineerida kui programmi võimet õppida ilma, et teda oleks spetsiaalselt selle jaoks programmeeritud. Üldiselt toimivad masinõppe algoritmid seoste loomise ning üldistamise põhjal. Teisisõnu suudavad masinõppe algoritmid võtta sisendandmed, need üldistada ning selle põhjal järeldusi teha.

Masinõppe algoritmid suudavad lahendada täpselt neid probleeme, millele lõputöös lahendust otsitakse. Masinõppe algoritme kasutatakse klassifitseerimise ülesannete lahendamiseks. Mingite sisendandmete korral suudavad masinõppe algoritmid leida sarnaseid mustreid ning nende põhjal tulemused klassidesse jagada. Klassidesse jagatakse andmeid erinevate algoritmide korral erinevalt. Seetõttu on antud peatükis analüüsitud erinevaid algoritme ning nad funktsionaalsuse järgi gruppidesse jagatud.

Magistritöö eesmärkide saavutamiseks on oluline leida sobiv algoritm. Erinevaid loodud algoritme on väga palju. Kuigi mõni algoritm võib sobida mitmesse gruppi, on ta lisatud kõige sobivamasse. Iga algoritmi grupi juurde on lisatud pilt algoritmi rühma üldise tööpõhimõtte illustreerimiseks ning välja on toodud paljud gruppide esindajaid.

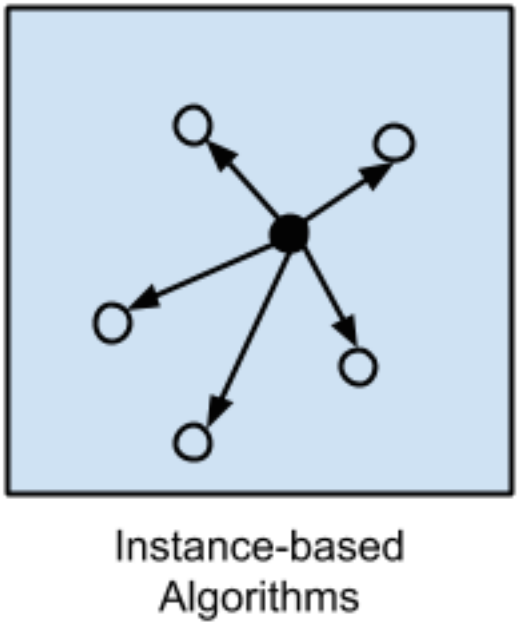
5.1 Juhtumil põhinevad algoritmid

Juhtumil põhinevaid algoritme võib nimetada ka eelkogemusel põhinevateks algoritmideks. Seda just seetõttu, et andmete üldistamise asemel võrdlevad algoritmid uusi sisendeid treenimise jooksul kasutatud sisenditega ning teevad selle põhjal järeldusi. Treenimise all peetakse silmas programmi õppimist sisendandmete põhjal. Teisisõnu püstitatakse hüpotees treeningandmete põhjal. Andmeid eelnevatega võrreldes leitakse kõige sarnasem ning antakse see tulemuseks.

Populaarseimad juhtumil põhinevad algoritmid on:

- k- lähima naabri algoritm (kNN)

- Ise organiseeruv kaart (SOM)
- Kohapeal kaalutud õppimine (LWL)
- Õppevektori kvantimine (LVQ)



Joonis 10 - Juhtumil põhinev algoritm (Brownlee 2013)

Joonisel on näha, et objekti klassifitseerimisel vaadatakse läbi kõik lähedalolevad andmed. Algoritmid liigutavad vaba objekti temale kõige sarnasema objekti juurde, ehk omadustelt kõige lähema atribuudi (naabri) juurde. Nii töötavad kõik eelpool mainitud algoritmid.

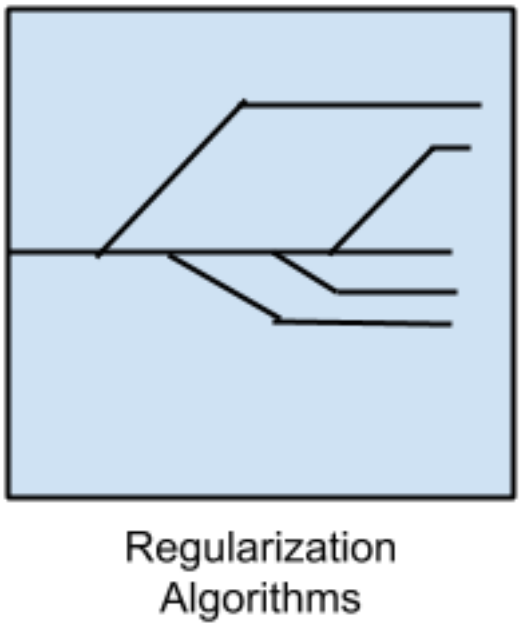
5.2 Seaduspärasuse algoritmid

Need on algoritmid mida võib vaadelda ka kui laiendusi teistele mudelitele (tüüpiliselt regressiooni mudelitele). Seaduspärasuse algoritmid vaatlevad lihtsaimaid mudeleid, mis suudavad paremini andmeid üldistada. Keerukamad mudelid jäetakse tahaplaanile.

Populaarseimad seaduspärasuse algoritmid on:

- Tipu regressioon (i.k. Ridge Regression)
- Elastne võrk
- LASSO

- Väikseima nurga regressioon (LARS)



Joonis 11 - Seaduspärasuse algoritm (Brownlee 2013)

Antud joonisel on näha kuidas andmed selekteeritakse ning reguleeritakse kindlate reeglite alusel, et ennustada tulemusi. Kuna klassifitseerimine toimub varajases faasis, ning andmed ei liigu enam peasiinile tagasi, siis võib tekkida probleem, et olulised andmed klassifitseeritakse valesti. Näiteks tudengite korral võib esimesest testist mitte osa võtnud tudeng sattuda riskigruppi ning sealt enam mitte välja saada. Seega antud algoritmi ei oleks otstarbekas kasutada. Vastasel korral võib riskigruppi sattunud tudengite arv kasvada liiga suureks.

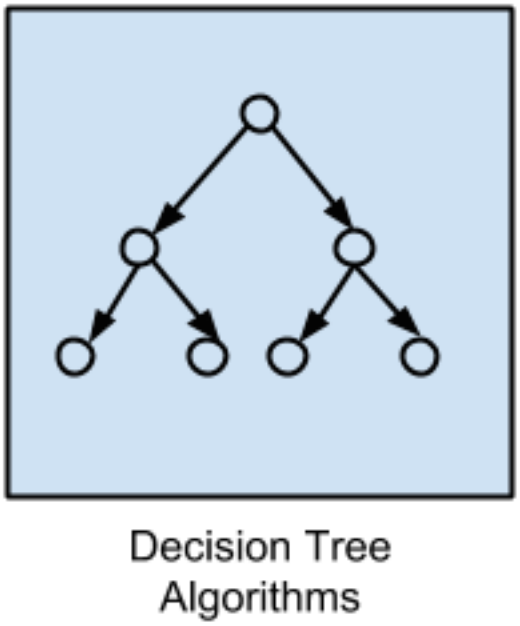
5.3 Otsustuspuu algoritmid

Antud algoritmid koostavad sisendite põhjal otsustuspuu. Otsustuspuud on spetsiaalselt treenitud andmete klassifitseerimiseks ning regressiooni probleemide lahendamiseks. Masinaõppes on otsustuspuu algoritmid ühed kõige kasutatuimad tänu kiiretele ning täpsetele tulemustele.

Populaarseimad otsustuspuu algoritmid on:

- Tingimuslikud otsustuspuud
- Iterative Dichotomiser 3 (ID3)

- C4.5 ja C5 (ID3 algoritmi edasiarendused)
- Hii-ruudu automaatse vastastikmõju avastamine (CHAID)
- Otsustuskänd ehk üheastmeline otsustuspuu (i.k. Decision Stump)
- M5



Joonis 12 - Otsustuspuu algoritm (Brownlee 2013)

Joonisel on näha otsustuspuu. Otsustuspuu algoritmid teevad otsuse ning liiguvad edasi tehtud otsuse suunas. Mööda hargnenud harusid liikudes jõutakse tulemuseni. Otsustuspuu algoritmid on need, mida antud magistritöös oleks vaja kasutada, kuna iga sisendi põhjal on vaja eraldi otsus teha. Nii saab testitulemuste ning muude atribuutide põhjal hinnata, kas tudeng võib antud kursuse pooleli jätta. Otsustuspuu väljundiks on kas jah või ei väärtus.

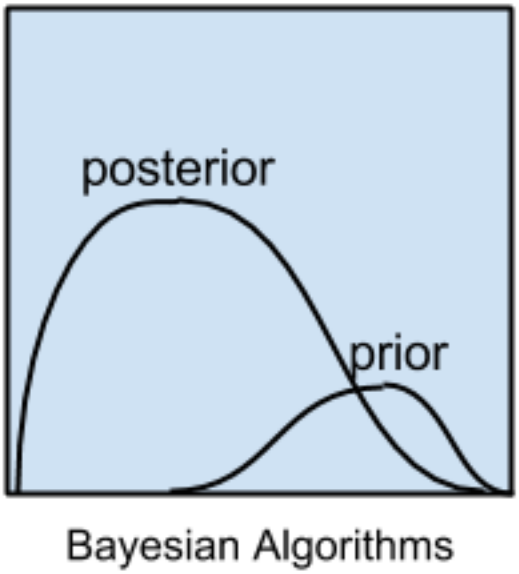
Otsustuspuu algoritmidest kõige populaarsem on ID3 ning selle edasiarendus C4.5. Peatükis 6 analüüsitakse täpsemalt ID3 algoritmi tööpõhimõtet ning tehakse järeldus, kas algoritm sobib antud probleemi lahendamiseks.

5.4 Bayesi algoritmid

Bayesi algoritmid rakendavad klassifitseerimise ning regressiooni probleemide lahendamisel Bayesi teoreemi. Bayesi algoritme oleks mõistlik rakendada, kui sisendandmeid on väga palju. Vaatamata algoritmi lihtsusele suudab see saavutada eesmärgid paremini, kui mõni teine algoritm.

Populaarseimad Bayesi algoritmid on:

- Naiivne Bayes
- Gaussi naiivne Bayes
- Multinominaalne naiivne Bayes
- Bayesi võrk (BN)
- Keskmist ühest sõltuvust ennustavad (AODE)



Joonis 13 - Bayesi algoritm (Brownlee 2013)

Joonisel "Prior" tähendab eelteadmisi. Olgu teada, et tudeng sai eelmise testi viie, siis eelteadmiste põhjal võib järeldada, et ka järgmine test on viiele sooritatud. Vastupidiselt

väidab "Posterior", et kuna tudeng sai juba viie, siis ta enam viite ei saa. Teisisõnu on kõik atribuudid on teineteisest sõltumatud ning mõlemad juhud vaadeldakse läbi.

Kuna ka tudengite testitulemused on üksteisest sõltumatud, siis tundub antud algoritm esialgu hea lähenemisena. Seetõttu analüüsitakse kuuendas peatükis täpsemalt algoritmi plusse ja miinuseid.

5.5 Klasterdamise algoritmid

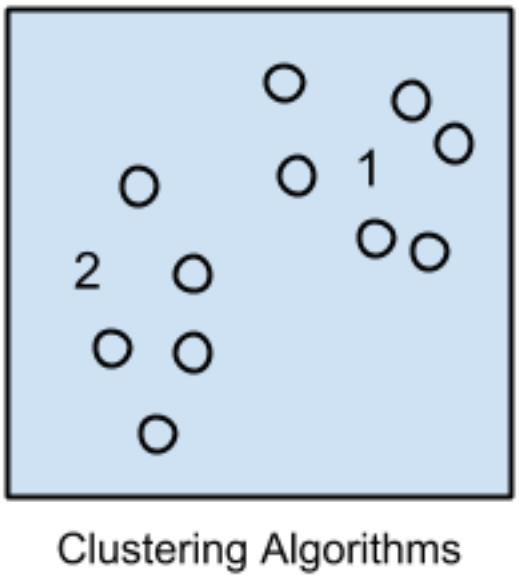
Klasterdamise algoritmid sarnanevad olemuselt regressiooni algoritmidele. Klasterdamise algoritmidele tuleb sisendiks anda mingid punktid, mis algoritm grupeerib klassidesse järgides järgnevaid reegleid:

1) Igas klassis olevad punktid on teine teisega sarnased

2) Erinevates klassides olevad punktid on erinevad.

Populaarseimad klasterdamise algoritmid on:

- k-Keskmine
- k- Mediaani
- Eelduste maksimeerimise algoritm (EM)
- Hierarhiline klasterdamine



Joonis 14 - Klasterdamise algoritm (Brownlee 2013)

Klasterdamise algoritmid grupeerivad sarnased andmed klassidesse. Sarnased atribuudid sattuvad samasse klassi. Erinevates väljaannetes kritiseeritakse klasterdamise algoritmigruppi ning väidetakse, et see ei tohiks olla masinõppe algoritm, kuna algoritm ei õpi treeningandmete põhjal. (Stackexchange 2014) Algoritm võib iga kord käivitades anda erineva tulemusi, kuna hakkab klasterdamisel peale suvalistest elementidest (ilma kindla järjekorrata).

Eelneva põhjal võib välistada antud algoritmi kasutamise töös, kuna oleks vaja saada täpsemaid tulemusi. Eelkõige oleks oluline, et algoritm teeks iga kord sama otsuse.

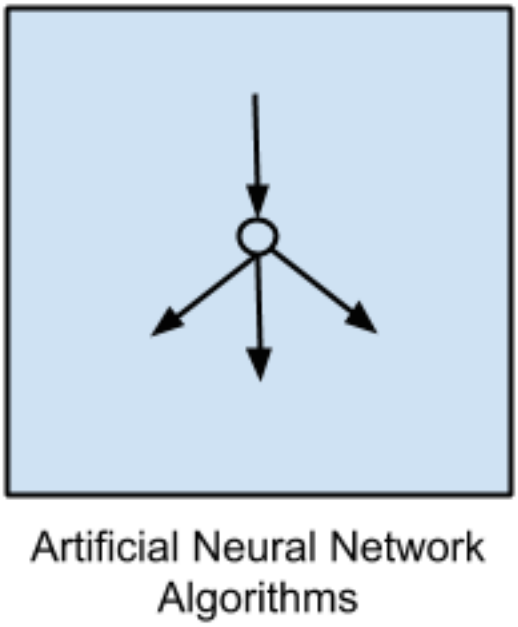
5.6 Tehislikud närvivõrkude algoritmid

Närvivõrkude struktuur sarnaneb bioloogiliste närvivõrkudega. Närvivõrkude klass on üks suurimaid, sisaldades väga palju erinevaid algoritme. Närvivõrkude algoritmid suudavad suurte tundmatute sisendandmete korral funktsioone koostada. Lisaks väljundi arvutamisele suudab närvivõrk õppimise käigus ka seoste kaale muuta. Tihtipeale esitatakse närvivõrkude jooniseid neuronite abil, mis on üksteisega seotud ning vahetavad informatsiooni.

Populaarseimad närvivõrkude algoritmid on:

- Perceptron
- Tagasi-levitamine

- Hopfield võrk
- Radiaalsel põhimõttel toimiv võrk (RBFN)



Joonis 15 - Tehis närvivõrgu algoritm (Brownlee 2013)

Joonisel on näha, et sisendväärtust võrreldakse läviväärtusega ning väljastatakse funktsioonid. See kirjeldab tehisnärvivõrkude peamist tööpõhimõtet. Tehisnärvivõrkude peamiseks puuduseks on kitsendused sisendandmetele. Nimelt suudavad närvivõrgu algoritmid analüüsida ainult numbrilisi sisendeid. Kuna Moodle logidest tuleb väga palju lausetena teksti, mida on vaja sorteerida, siis närvivõrkude algoritm langeb valikust välja.

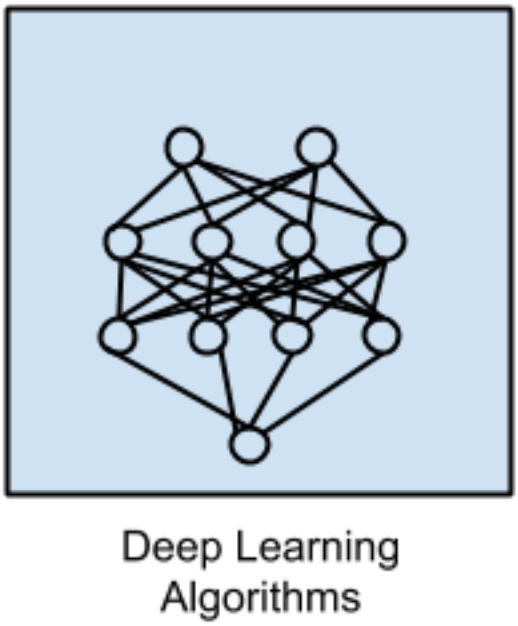
5.7 Süvanärvivõrkude algoritmid

Süvanärvivõrkude algoritmid on kaasaegsemad ning nad ehitavad kordades suuremad ning keerukamad võrgud. Süvanärvivõrkude korral rakendatakse õppimist keerukate etteantud struktuuridega (võivad puududa tähistatud andmed).

Populaarseimad süvanärvivõrkude algoritmid on:

- Süva Boltzmanni Masin (DBM)
- Sügava veendumuse võrgud (DBN)

- Edasiarendatud närvivõrgud (CNN)
- Kuhjatud automaatkodeerijad



Joonis 16 - Süvanärvivõrgu algoritm (Brownlee 2013)

Joonisel on näha keerukamad närvivõrgud, mis on kombineeritud teiste meetoditega. Pärast närvivõrkude treenimist on peaaegu võimatu näha, kuidas algoritm lahenduseni jõudis. Treeningandmete põhjal probleemi lahenduseni jõudmist võib kutsuda ka mustaks kastiks. Seega on võimatu muuta algoritmi. Ainus võimalus on muuta algoritmi sisendit. Nagu närvivõrkude juures mainitud, sobib ka süvanärvivõrgu algoritm vaid numbriliste sisendandmete korral. Seega ei sobi ka antud algoritmigrupp magistritöö eesmärkide saavutamiseks.

5.8 Kokkuvõte

Antud peatükis tutvustati paljusid masinõppe algoritme ning esitati lihtsamaks mõistmiseks ka joonised. Arvestades magistritöö eesmärke tuleb leida parimad algoritmid, mis sobivad tudengite ainelt väljalangemise ennetamise probleemi lahendamiseks. Väga paljudel algoritmidel esines puudusi, mistõttu neid edaspidi süvitsi ei analüüsita. Küll aga leiti kaks head algoritmigruppi, mida vaadeldakse peatükis 6.

Vaja on leida klassifitseerimise algoritm, mis suudab sisendandmete põhjal tudengid klassifitseerida gruppidesse. Kuna eelnevalt on teada, et sisendandmeid on väga palju, siis ühe valikuna jääb sõelale kindlasti Bayesi algoritmigrupp, millest järgnevas peatükis rohkem kirjutatakse.

Teiseks on teada tudengite tulemused. Riskigruppi oleks hea ennustada siis-kui reeglite abil. Kuna otsustuspuu algoritmid sobivad siis-kui reeglite põhjal järeldusi tegema, siis on teine valik otsustuspuu algoritmid. Otsustuspuu algoritmide grupist analüüsitakse ID3 algoritmi.

Järgmises peatükis hakatakse rohkem keskenduma eelpool mainitud kahele algoritmigrupile: Bayesi algoritmid ning otsustuspuud.

6. Valitud algoritmide analüüs

Järgnevalt on analüüsitud mõningaid algoritme näidete põhjal, mis on proovinud objekte klassifitseerida või otseselt tudengite väljalangemist hinnata. Välja on toodud nende lähenemiste plussid ja miinused.

Kuna uurimustöö eesmärk on ennetada tudengite väljalangevust siis on selle jaoks vaja leida parim meetod. Eelmises peatükis vaadeldud algoritmidest keskendutakse nüüd täpsemalt Bayesi algoritmide ning otsustuspuude uurimisele.

6.1 Bayesi algoritmid

Bayesi staatilised klassifikaatorid ennustavad objekti klassi kuuluvust tõenäosuse abil nõnda, et sisendandmed jaotatakse ära klassi kuulumise tõenäosuse põhjal. Arendatud on mitmeid erinevaid Bayesi algoritme millest tuntuimad on Bayesi võrgud ning naiivne Bayes.

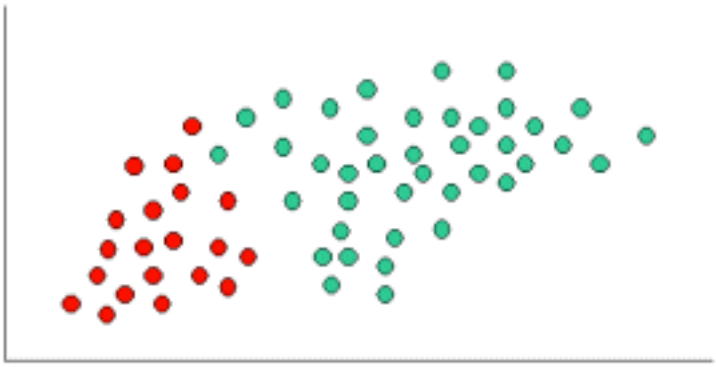
Naiivse Bayesi algoritmid eeldavad, et atribuutide efektid klassidele on sõltumatud teistest atribuutidest. Reaalsuses tavaliselt siiski esinevad seosed atribuutide vahel.

Bayesi võrgud on aga graafi mudelid, mis kirjeldavad ühise tingimuse tõenäosusjaotust. Bayesi algoritme kasutatakse tihtipeale andmete klassifitseerimisel tänu nende lihtsusele, arvutuse efektiivsusele, ning adekvaatsetele väljunditele. Lisaks loetakse Bayesi mudelite olulisteks eelisteks ka kiiret treenimist, tulemuste hindamist ja väljundi asjakohasust erinevates valdkondades. (Kabakchieva 2013)

Kõik masinõppe algoritmid vajavad treenimist, kui soovida, et nad väljastaksid klassifitseeritud ning ennustavad andmed. Treenimine tähendab kindlate sisendite etteandmist programmile, mille tulemusel saab hiljem anda programmile ette tundmatuid andmeid(selliseid mida programm varem töödelnud ei ole). Õppimise tulemusel suudavad algoritmid ennetada või klassifitseerida andmeid. Nõnda on üles ehitatud enamus masinõppe tehnikaid, sealhulgas ka Bayesi teoreemid. (Bittlingmayer 2016)

Naiivse Bayesi näide:

Järgnevas näites klassifitseeritakse kahte tüüpi objekte. Objektideks on punased ja rohelised ringid. Ülesandeks on klassifitseerida uued objektid nende tekkimise ajal arvestades eelteadmisena olemasolevaid objekte. Näidet on illustreeritud järgneval joonisel.



Joonis 17 - Klassifitseeritud objektid (Bittlingmayer 2016)

Joonisel on näha klassifitseeritud objektid. Objektid on klassifitseeritud värvide alusel punasteks ning rohelisteks. Punaseid objekte on 20 ning rohelisi 40.

Eelteadmisena on teada, et rohelisi objekte on punastest 2 korda rohkem. Kui lisada massiivi üks uus objekt siis lähtuvalt Bayesi analüüsi eeltõenäosusest on 2 korda tõenäolisem, et see objekt on roheline. Eeltõenäosus põhineb eelneval kogemusel- antud juhul punaste ja roheliste objektide osakaalust. Tulenevalt eelpool mainitust saab ennustada tulemusi enne kui need juhtuvad. (Bittlingmayer 2016) Öeldust järeldub, et eeltõenäosus roheliste objektide saamiseks avaldub:

$$Rohelise\ objekti\ tulek = \frac{roheliste\ objektide\ arv}{objektide\ koguarv}$$

Analoogselt avaldub ka punaste objektide tõenäosus:

$$Punaste\ objekti\ tulek = \frac{punaste\ objektide\ arv}{objektide\ koguarv}$$

Asendades arvud valemisse saab roheliste objektide tuleku tõenäosuseks 40/60=0.66(6) ning punaste objektide tuleku tõenäosuseks 20/60=0.33(3). Nagu eelnevalt mainitud, siis järeldub siit, et tekkinud objekti liik sõltub eelteadmistest.

Antud algoritmi teooriat saaks analoogselt rakendada ka tudengi hinnete analüüsimiseks. Selle jaoks tuleks võtta tudengi eelnevad hinded ning nende abil saaks ennustada ka tulevasi hindeid vastavalt enim saadud hinnetele.

Algoritmi kõige suuremaks miinuseks on ennustamise täpsus. Bayesi klassifitseerimise täpsuseks loetakse ligi 60%, mis on madalam kui otsustuspuu korral. (Kabakchieva 2013) Seetõttu vaadeldakse järgmises peatükis ühte otsustuspuu tuntuimat algoritmi.

6.2 Otsustuspuu algoritm ID3

Otsustuspuude peamiseks eelisteks loetakse nende arusaadavust. Otsustuspuud esitavad reegleid, mida kasutajatel on lihtne mõista. Otsustuspuud ei nõua keeruliste andmete ettevalmistamist ning toimivad hästi nii numbriliste kui kategoriseeritud väärtuste korral. (Kabakchieva 2013)

Induktiivse masinõppe algoritm ID3 (Iterative Dichotomiser 3) on üks tuntuimaid algoritme andmete klassifitseerimiseks. See koostab sisendiks antud otsustustabeli põhjal otsustuspuu minimaalsete tippude arvuga. Antud algoritmil on küll väike veaprotsent, kuid see töötab aeglaselt.

ID3 algoritmi väljundiks on paaride komplekt väärtustega 0 või 1. Väärtus 0 on väljundiks anomaalia korral ning väärtus 1 on väljundiks normaalolukorras. (Khedr, Idrees ja Seddawy 2016)

Otsustuspuu algoritmide kasutamise peamised küsimused :

1. Otsuse tegemisel arvestatavad atribuudid
2. Millises järjekorras andmeid kasutada
3. Kui sügav ja kui lai puu peab olema (Adhatrao, et al. 2013)

Eelnevatele punktidele vastamisel tuleb eelkõike lähtuda sellest, millised atribuudid toovad kõige suurema kasu informatsiooni analüüsimisel, ehk teisisõnu kõige relevantsemad

atribuudid tulemuse hindamiseks. (Dankel 1997) Samuti tuleb tähelepanu pöörata andmete kasutamise järjekorrale. Algoritm jõuab esimesena puu ülemistesse harudesse. Kui sinna on pandud kriteerium, mis väga palju andmeid välja sorteerib, siis võib tulemus olla mittetäpne.

Näidiseks võetakse lihtsuse huvides kaks atribuuti: tudengite hinne ja kohalkäimine. Atribuudid ning nende väärtused tuleks kirja panna järgnevalt:

hinne = {1, 2, 3, 4, 5}

kohalkäimine = {harva, tihti}.

Informatsioonist saadav kasu tuleb arvutada kõikide atribuutide kohta. Kuna meil on 2 atribuuti, siis nende põhjal tuleb teha otsus kas tudeng jätkab või katkestab ülikooli. Järgnevas tabelis on lihtsustatud näide illustreeritud.

Tabel 1 - Sisendandmed tudengi kohta

Tudeng	Hinded	Kohalkäimised	Katkestamine
113114	5	tihti	Ei
112349	4	tihti	Ei
113214	5	harva	Ei
113225	2	harva	Jah
113953	1	harva	Jah

Info kasulikkuseks nimetatakse alguse ning lõpptulemuse entroopiate vahet. Teisisõnu hinnatakse kui palju ebakindlust andmetest eemaldatakse mingi atribuudi arvestamisega. Informatsioonist saadav kasulikkus arvutatakse järgneva valemiga:

$$Kasu(S, kohalkäimised) =$$
$$= Entroopia(S) - \left(\frac{2}{5}\right) * Entroopia(S\ tihti) - \left(\frac{3}{5}\right) * Entroopia(S\ harva)$$

Valemis on S tudengite arv, tihti ja harva on kohalkäimise võimalikud väärtused. Entroopia on ebakindlate andmete osakaal sisendandmetes.

Entroopia arvutatakse omakorda järgneva valemiga:

$$Entroopia(S) = -\left(\frac{2}{5}\right)Log2\left(\frac{2}{5}\right) - \left(\frac{3}{5}\right)Log2\left(\frac{3}{5}\right) = 0.970$$

Valemis on 5 ridade arv tabelis, 2 on "jah" vastuse saanud tudengid ning 3 on "ei" vastuse saanud tudengid tabelist 1.

Arvutama peab ka mõlemale sündmusele eraldi entroopiad:

$$Entroopia(S\ tihti) = -\left(\frac{0}{2}\right)Log2\left(\frac{0}{2}\right) - \left(\frac{0}{2}\right)Log2\left(\frac{0}{2}\right) = 0$$

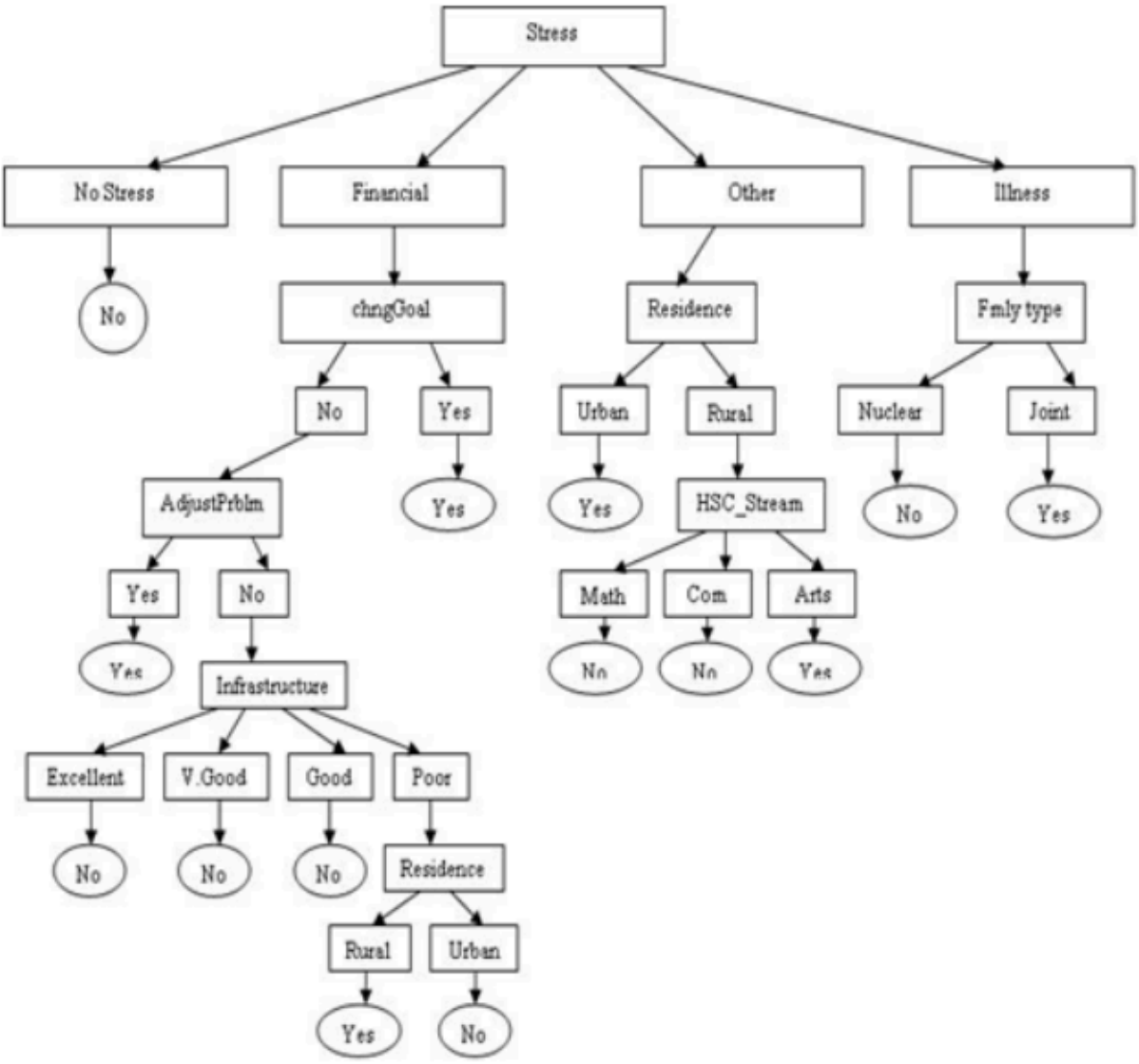
$$Entroopia(S\ harva) = -\left(\frac{1}{3}\right)Log2\left(\frac{1}{3}\right) - \left(\frac{2}{3}\right)Log2\left(\frac{2}{3}\right) = 0.918$$

Asendades arvud valemisse, saab kohalkäimiste atribuudi informatsiooni kasulikkuseks: 0.970 - (2/5) * 0 - (3/5) * 0.918 = 0.419, mis on päris kõrge kasulikkuse näitaja. Kasulikkuse näitajate võimalikud väärtused on vahemikus 0 kuni 1, kus 0 on kõige madalam kasulikkuse näitaja võimalik väärtus. Siit järeldub, et kohalkäimine on väga oluline parameeter hindamaks tudengi jätkamist õppetöös.

Järgmise näitena vaadeldakse ID3 algoritmi Botswana teadlaste poolt loodud otsustuspuu algoritmi edasiarendust. Antud töös on kasutatud aastal 1979 arendatud ID3 otsustuspuu algoritmi edasiarendust kasutades Rényi entroopiat.

Töös kasutatakse sisendandmetena ajaloolisi andmeid õppuri kohta nagu näiteks vanemate haridus, vanemate töökogemus, akadeemilised saavutused jne. Sisendandmeid koguti 31 valdkonna koha. Andmekogumit vähendati märkimisväärselt võttes arvesse vaid andmeid, mis on korrelatsioonis õpingute katkestamisega. Pärast ID3 algoritmi edasiarendust leiti, et vaid 12 muutujat andmehulgast mõjutavad otseselt õppuri õppeedukust. (Sivakumar, Venkataraman ja Selvaraj 2016)

Kuna algoritmi sisendiks oli palju andmeid siis olid ka tulemused väga täpsed. Töö autorid hindasid kas tudengid jätkavad ülikoolis õppimist või katkestavad ülikooli. Andmete põhjal koostati otsustuspuu (Joonis 18). Tavalise ID3 algoritmi täpsuseks arvutasid autorid enda töös 92.50%, kuid täiendatud algoritmi täpsuseks arvutati 97.50%. (Sivakumar, Venkataraman ja Selvaraj 2016)



Joonis 18 - Otsustuspuu (Sivakumar, Venkataraman ja Selvaraj 2016)

Joonisel nähtavat otsustuspuud kasutades, tehakse tudengi väljalangemise kohta hinnang. Otsustuspuu uurib, kas stressist põhjustatud faktorid mõjutavad tudengi katkestamist.

Töö tulemusel jõudsid autori järeldusele, et kõige suurem väljalangevus on põhjustatud perekondlikest põhjustest (10.25%). Teisel kohal on ülikoolilinnaku halb keskkond (7.58%). (Sivakumar, Venkataraman ja Selvaraj 2016)

Antud edasiarendus algoritmile on väga hea täpsusega (97.5%) ning seda võiks kohaldada ka Eesti ülikoolide jaoks. Selle põhjal saaks teha statistikat ning avastada peamised tegurid õpingute katkestamisel mingi aine näitel.

Algoritmi miinusteks on see, et andmeid saab analüüsida vaid tagantjärgi ning nende tulemustega enam otseselt kedagi väljalangemisest päästa ei saa. Antud töö eemärk on aga arendada välja algoritm, mis õppetöö käigus suudaks jooksvalt identifitseerida tudengeid, kellel on suurem väljalangemise risk.

Lisaks on antud töö puudus ka see, et tulenevalt Eesti ja India erinevast ühiskonnast ei pruugi seosed olla samasugused ja seega tuleks enne antud metodoloogia kasutamist hinnata kriitiliselt otsustuspuu sobivust võttes arvesse siinset keskkonda ja eripärasid.

6.3 Teiste algoritmide peamised puudused

Eelnevatel lahendustel on piisavalt miinuseid või kitsaskohti, mis on järgnevalt välja toodud.

Otsustuspuu algoritmi peamiseks puuduseks on, et see analüüsib ajaloolisi andmeid. Nende põhjal saaks teha statistikat, küll aga mitte ennetustööd tudengite väljalangemisel. Ometigi on võimalik kasutada otsustuspuu algoritmi puhul info kasulikkuse hindamist, mille abil määratakse kõige vajalikumad kriteeriumid tudengi riskigrupi määramisel.

Moodle logid sisaldavad väga vähe struktureeritud andmeid. Struktureeritud andmete põhjal suudavad algoritmid teha täpsemaid otsuseid. Logidest peab väärtused välja sorteerima abstraktsetest lausetest. Laused on pikad ning seal on väga palju ebavajalikku informatsiooni.

Teised e-õppe jaoks loodud algoritmid keskenduvad peamiselt videote vaatamistele ning foorumi aktiivsusele. MOOC süsteemides on need parameetrid kõige olulisemad näitajad. Moodles aga ei ole foorumi kasutamine ning videoloengute üleslaadimine veel väga populaarne. Seega antud parameetreid on raske hinnata.

Siit võib järeldada, et kui soovitakse töö alguses püstitatud eesmärke saavutada, siis tuleb arendada uus algoritm, mis suudab õppetöö kestel piisava täpsusega tudengite õppeedukust

prognoosida. Selle saavutamiseks peab kasutama andmeid, mida Moodlest tudengite kohta kätte saab. Näiteks peab hindama tudengite testide hindeid, testide lahendamise aegasid ning kursuse külastatavust.

6.3.1 Miks oleks vaja uut algoritmi?

Eelnevalt vaadeldud algoritmid andsid klassifitseerimisel üpris täpseid tulemusi. Analüüsitud algoritmidest kõige täpsemini suutis tulemusi ennustada ID3 algoritm. Küll aga ei pruugi antud lõputöö probleemile lähenemine ID3 algoritmi kasutades sama täpseid tulemusi anda.

ID3 algoritm sobiks andmete klassifitseerimiseks kõige paremini siis, kui sisendandmete atribuudid oleksid kujul jah-ei. Siis suudab algoritm kõige täpsema tulemuse saavutada, kuna kasutab tulemuste arvutamisel 0-1 süsteemi. Algoritmi väljundiks on 0 anomaalia korral ning 1 normaalsete (oodatud) väärtuste korral. Tudengite tulemusi ei saa nii ühekülgselt vaadata. ID3 algoritmi kasutades võrdsustataks tudengid, kelle tulemus on 51 punkti, tudengitega, kelle tulemus on 100 punkti. Mõlemad on aines testi läbinud ning ID3 algoritmi põhjal saaksid nad tulemuseks 1, ehk oleksid võrdsed.

Seetõttu oleks vaja kirjutada algoritm, mis kindlatele tulemustele seab vastavusse kindlad punktid. Teisisõnu on vaja koostada tudengi profiil ning igale tudengile seada vastavusse tema riskigrupp. Vastavalt tudengi õppeedukusele ning muudele Moodle logidest saadavatele parameetritele tuleb koostada algoritm, mis annab iga tulemuse põhjal tudengitele kindlaid punkte. Nii välditakse olukorda, kus algoritm väljastab ebatäpseid tulemusi ning ei suuda hoiatada õppejõudu tudengi probleemidest piisavalt varakult.

Algoritmi aluseks on mõistlik võtta ID3 algoritmi informatsioonist saadava kasu hindamine. Selle meetodi abil on võimalik määrata, kas ja millistest andmetest tudengi tulemuste ennustamisel üldse kasu on.

7. Algoritmi arendamine

Järgnevas peatükis tutvustatakse informaatika aine hindamiskriteeriume. Informaatika I ainekoodiga IDK0091 on matemaatika ja loodusteaduskonna õppeaine (Y-teaduskond). Informaatika aine on valitud seetõttu, et seal esineb väga palju erinevaid tegevusi Moodle keskkonnas.

Pärast aine tutvustamist arvutatakse info kasulikkus testide ning ülesannete põhjal ning kasutatakse valitud parameetreid loogikareeglites ning pseudokoodis. Leitud parameetreid kasutatakse ka algoritmi arendamisel ning nende põhjal ennustatakse, kas tudeng võib õppeaine katkestada.

Lõpuks arvutatakse välja algoritmi täpsus, kõrvutades algoritmi väljundandmeid tegelike kursuselt välja langenud õpilaste arvuga.

7.1 Informaatika aine kriteeriumid

Enne algoritmi koostamist on vaja teada informaatika aine hindamiskriteeriume. Õppeinfosüsteemis oleva informatsiooni põhjal on teada, et informaatika aines hinnatakse tudengeid mitteeristuvas hindamissüsteemis, kus tulemusteks on arvestatud või mittearvestatud (A/MA). Arvestuse saavad kõik tudengid kellel on:

- Esitatud ja aktsepteeritud kõik kodutööd
- Tehtud kõik praktikumide ülesanded ja harjutused.

Kodutööde aktsepteerimine on arvestusele pääsu üheks eelduseks. Mitteaktsepteeritud töid on võimalik parandada. Tööd aktsepteeritakse siis kui :

- Kõik ülesanded on lahendatud
- Ükski lahendus ei sisalda olulisi vigu
- Vormistus on korrektne ja vastab nõuetele.

(TTÜ Õppeinfosüsteem 2016)

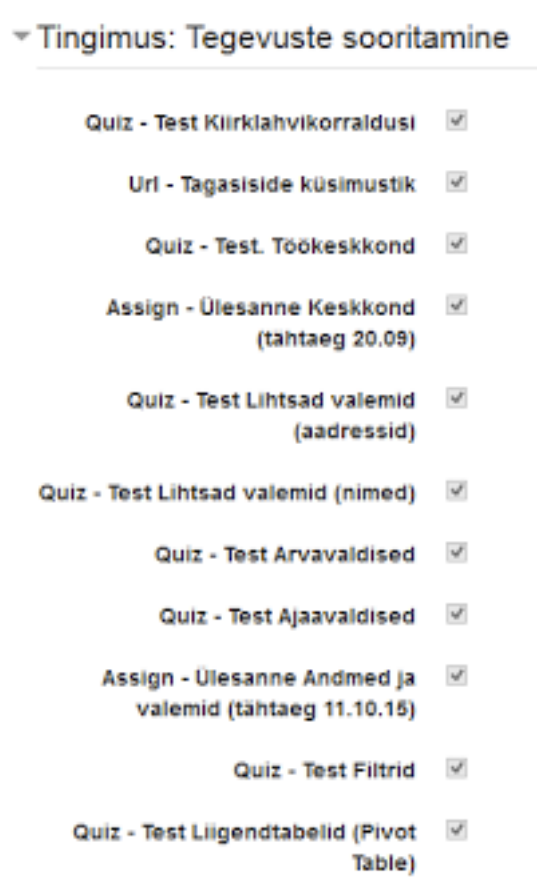
Tuleb arvestada, et kõik testid peavad olema sooritatud vähemalt 90 protsendile maksimumist. Teste saab sooritada mitu korda. Testid on mõeldud lihtsate algoritmide tutvustamiseks, sagedaste vigade näitamiseks ning asjakohaste nuputusülesannete jaoks.

Informaatika I aine IDK0091 (Y-teaduskonnale) käigus õpitakse rakenduste loomist tabelprogrammide keskkonnas. Kursusel käsitletakse rakenduste loomise üldiseid põhimõtteid, meetodeid, vahendeid ja põhifaase. Vaadeldavatest meetoditest ja vahenditest on kesksel kohal objektorienteeritud modelleerimine. (TTÜ Õppeinfosüsteem 2016)

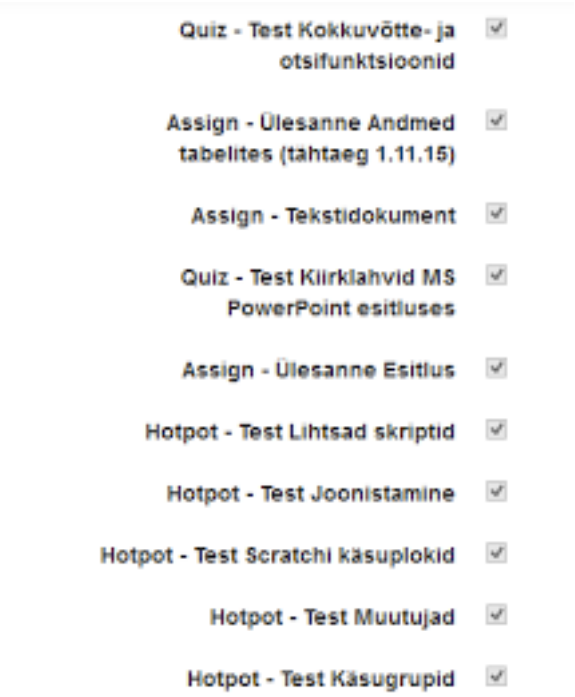
Informaatika aines õpitakse ka sissejuhatust programmeerimisse ja arendussüsteemi Visual Basic. Tutvutakse programmide ja protseduuride tüüpidega, lihtsamate objektide ja skalaarandmete kasutamisega, programmide sisestamisega, redigeerimisega, silumise ja käivitamisega. (TTÜ Õppeinfosüsteem 2016)

Algoritmi üheks sisendiks on informaatika aine hinneteleht. Järgnevalt on kirjeldatud parameetreid, mis algoritmile sisendiks antakse ning millisele kujule on hinneteleht vaja kohaldada kõikidel, kes soovivad algoritmi kasutada. Hinnetelehe andmed on tabelis, mille esimesed 2 veergu on tudengi eesnimi ning perekonnanimi. Neljas kuni kaheksas veerg sisaldavad testide tulemusi skaalal, kus 5 on maksimumtulemus. Veerud 9-11 sisaldavad testide tulemusi skaalal, kus 10 on maksimumtulemus. Edasi on kolme ülesande tulemused mille võimalikeks väärtusteks on "arvestatud", "lubatud kaitsmisele" ja "parandada kaitsmiseks". Edasi on veergudes 13 ja 15 ülesannete lahendused, kus kasutatakse samu parameetreid nagu eelnevate ülesannete korral. Veerus 14 ja 17 on testide tulemused 5 punkti skaalal. Kaheksateistkümnendas veerus on test mille maksimaalseks tulemuseks on 2 punkti. 19 ning 21 veergudes on taas testid, mille maksimaalne võimalik skoor on 5 punkti. Ning viimased testid on 20 ja 22 veergudes maksimaalsete punktidega 10.

Eelnevat lõiku illustreerivad järgnevad 2 joonist, kus on näidatud millised sooritused on kohustuslikud aine läbimiseks.



Joonis 19 Aine sooritamiseks vajalikud testid ja ülesanded 1 (Moodle hitsa 2016)



Joonis 20 Aine sooritamiseks vajalikud testid ja ülesanded 2 (Moodle hitsa 2016)

Eelnevatelt joonistelt ei ole vaja arvestada tagasisidega seotud küsimustikke, kuna nende tulemused ei kajastu sisendiks võetud hinnetelehel. Seega ülejäänud atribuute arvesse võttes saab neid kasutada teoreetilistes alustes ning algoritmi koostamisel. Kõigil kes soovivad antud algoritmi hinnete alusel riskigrupi määramiseks kasutada, peavad algoritmile samas formaadis sisendi andma. Kui tudengil veel mõne testi eest tulemust ei ole ning tähtaeg ei ole veel möödas, siis võib väljad tühjaks jätta.

Teiseks sisendiks on Moodle logid. Moodle logides on erinevatel kellaaegadel tehtud tegevused. Igas logi reas kuvatakse tudengi nimi, tegevus mida tudeng sooritas, tegevuse kirjeldus ning seade, millelt süsteemi külastati. Tegevuste nimede järgi saab kokku arvutada mitu korda iga tudeng süsteemi külastas. Kui tegevuse nimi on *course viewed* siis tuleb antud sündmused iga tudengi kohta kokku arvutada.

7.2 Teoreetilised alused

Reeglitepõhised klassifitseerimisetehnoloogiad on ühed täpseimad meetodid hindamaks akadeemilisi saavutusi. Tulemuste ennustamise algoritmi täpsus Fadhilah Ahmad, Nur Hafieza Ismail ja Azwa Abdul Aziz uurimustöös oli 71.3%. (Ahmad, Ismail ja Aziz 2015)

"Teadmiste esitamisel reeglitena on võimalik formaliseerida ühte elus üsna tihti esinevat järeldamise moodust. Selline esitus lubab ka suhteliselt kergesti selgitada ja põhjendada esitatavaid küsimusi ja tehtud otsust. Lisaks on reeglibaasis suhteliselt lihtne teha reeglite lisamist, muutmist ja kustutamist. Kuna muutujaid säilitatakse eraldi reeglitest, saab ka muutujaid enamasti muuta reeglitest sõltumatult." (Tepandi 2016)

"Lihtsamal juhul kasutatakse reegleid tingimuslike seoste kirjeldamiseks. Nad esitatakse kujul: Kui (tingimus) Siis (mingi muutuja on tõene) {Muidu (mingi muutuja on väär)}." (Tepandi 2016)

Kuna teadmiste esitamisel reeglitena on lihtne reeglibaasis vajadusel muudatusi teha, on ka antud töös otsustatud kasutada tudengite akadeemiliste saavutuste hindamiseks klassifitseerivat reeglitepõhist algoritmi. Peatüki 6.3.1 põhjenduse põhjal võib järeldada, et tuleb koostada algoritm, mis seab tudengi tulemustele vastavusse kindlad punktid.

Enne algoritmi koostamist tuleb hinnata, millised sisendandmed annavad meile kõige täpsemad tulemused ehk kasu informatsioonist (i.k. Information Gain). Olgu antud sisendina

tudengite hinded ning kohalkäimised loengutes. Selle põhjal peab leidma kummast parameetrist on lõpptulemuseni jõudmisel rohkem kasu. Lõpptulemuseks on tudengeite aine läbimine ning atribuuti väärtus võib olla kas jah või ei.

Info kasulikkust saaks kõige paremini hinnata ID3 algoritmi info kasulikkuse hindamisega. Enne ID3 algoritmi otsustuspuu koostamise etappi hindab antud lähenemine ära kõige kasulikumad atribuudid. Hindamiseks kasutatavad valemid on välja toodud peatükis 5.2.1. Nende valemite põhjal arvutatakse järgmises peatükis (7.2) info kasulikkus.

Pärast info kasulikkuse leidmist võetakse kasutusele vähemalt 3 parameetrit, mille abil algoritm hakkab tudengitele vastavusse seadma punkte. Kui kõik sisendparameetrid annavad ennustamiseks piisavalt informatsiooni siis hakatakse kõiki algoritmis kasutama. Mida väiksemad on tudengi punktid, seda suurem on ta risk informaatika aine katkestada või üldse ülikool pooleli jätta.

7.3 Info kasulikkus ID3 algoritmi põhjal

Info kasulikkuse jaoks analüüsitakse järgnevaid parameetreid. Parameetrite järel loogelistes sulgudes on toodud välja parameetrite võimalikud väärtused.

- Testide hinded {-, 1, 2, 3, 4, 5}
- Ülesannete hinded {-, arvestatud (A), mittearvestatud (MA), parandatud (P)}
- Vabatahtlike ülesannete hinded {-, 1, 2, 3, 4, 5}
- Testide lahendamise kiirus {kiire, aeglane}
- Moodle külastamise aktiivsus {tihti, harva}

Parameetrid on arvutatud järgnevate reeglite abil:

1. Kui tudeng külastab Moodlet vähemalt nädalas ühel korral, siis on külastamise aktiivsus "tihti". Vastasel korral on külastamise aktiivsuse väärtuseks "harva".
2. Kui tudeng lahendab testi alla nelja minuti, siis on tema testide lahendamise kiirus "kiire". Vastasel juhul on testide lahendamise kiirus "aeglane".

3. Hinnete ning ülesannete puhul "-" tähendab testile mitteilmumist. Testi või ülesande sooritamise eest antakse tudengile kas hinne või arvestus.

Järgnevas tabelis on lihtsuse huvides toodud välja viis tudengit. Tegelikult on informaatika hinnete tabelis 62 õpilast ning info kasulikkuse arvutus koostatakse kõikide tudengite tulemusi arvesse võttes. Terviktabel on leitav Lisa 1 olevalt aadressilt.

Tabel 2 - Hinnatavad kriteeriumid

Tudeng	Testid	Ülesanded	Vabatahtlikud ülesanded	Kiirus	Moodle külastamine	Katkestamine
142982	5	A	-	aeglane	tihti	Ei
155346	4	A	-	kiire	tihti	Ei
142280	5	A	-	kiire	harva	Ei
144335	2	P	-	kiire	harva	Jah
144336	1	MA	-	aeglane	harva	Jah
...	...	...	...	...	...	...

Info kasulikkus on arvutatud valemiga, mida tutvustati punktis 5.2.1 vastavalt tabelis toodud parameetritele. Valemites on tähistatud entroopia tähega E.

Lihtsustamaks antud lähenemist, on antud skaalal arvesse võetud tudengite keskmist hinnet, mitte iga sooritust eraldi. Hindele 5 on vastavusse seatud suurepärane, hinnetele 4-1 on vastavuses hea ning läbi kukkunud hinnetele on pandud vastavusse kasin.

Kõigepealt arvutatakse entroopia kõikide parameetrite kohta. Meil on teada, et kursuse katkestas 8 tudengit ning kursuse läbis edukalt 54 tudengit.

$$Entroopia(S) = -\left(\frac{8}{62}\right) Log2\left(\frac{8}{62}\right) - \left(\frac{54}{62}\right) Log2\left(\frac{54}{62}\right) = 0.555$$

7.3.1 Informatsiooni kasulikkus testide põhjal

Nüüd tuleb arvutada kasu iga parameetri kohta. Esmalt võetakse testid. Järgnevalt esitatakse testidest kasu arvutamise valem.

$$Kasu(S, Testid) = \\ = E(S) - \left(\frac{50}{62}\right) * E(S\ suurepärane) - \left(\frac{4}{62}\right) * E(S\ hea) - \left(\frac{8}{62}\right) \\ * E(S\ kasin)$$

Et antud valemit kasutada saaks peab arvutama entroopiad iga tulemuse kohta:

$$Entroopia(S\ suurepärane) = -\left(\frac{50}{62}\right) Log2\ \left(\frac{50}{62}\right) = 0.250$$

$$Entroopia(S\ hea) = -\left(\frac{4}{62}\right) Log2\ \left(\frac{4}{62}\right) = 0.255$$

$$Entroopia(S\ kasin) = -\left(\frac{8}{62}\right) Log2\ \left(\frac{8}{62}\right) = 0.381$$

Seega asendades tulemused valemisse "Kasu(S, testid)", saab tulemuseks:

Kasu (S, testid) = 0.555 - (50/62 * 0.250) - (4/62 * 0.255) - (8/62 * 0.381) = 0.288

7.3.2 Informatsiooni kasulikkus ülesannete põhjal

Järgnevalt esitatakse ülesannetest kasu arvutamise valem.

$$Kasu(S, Ülesanded) == E(S) - \left(\frac{52}{62}\right) * E(S\ A) - \left(\frac{3}{62}\right) * E(S\ P) - \left(\frac{7}{62}\right) * E(S\ MA)$$

Entroopiad arvutatakse iga parameetri kohta:

$$Entroopia(S\ A) = -\left(\frac{52}{62}\right) Log2\ \left(\frac{52}{62}\right) = 0.213$$

$$Entroopia(S\ P) = -\left(\frac{2}{62}\right) Log2\ \left(\frac{2}{62}\right) - \left(\frac{1}{62}\right) Log2\ \left(\frac{1}{62}\right) = 0.256$$

$$Entroopia(S\ MA) = -\left(\frac{7}{62}\right) Log2\ \left(\frac{7}{62}\right) = 0.355$$

Seega asendades tulemused valemisse "Kasu(S, ülesanded)", saab tulemuseks:

Kasu (S, ülesanded) = 0.555 - (52/62 * 0.213) - (3/62 * 0.256) - (7/62 * 0.355) = 0.324

7.3.3 Informatsiooni kasulikkus vabatahtlike ülesannete põhjal

Kuna vabatahtlike ülesandeid semestri jooksul keegi ei lahendanud, siis antud infost tudengi väljalangemise ennustamisel mingit kasu ei ole. Kõikidel on tulemuseks märgitud mõttekriips ning valemisse asendades tekib nulliga jagamine, mistõttu on info kasulikkuseks 0.

7.3.4 Informatsiooni kasulikkus testide lahendamise kiiruse põhjal

Eelnevalt tuleb kokku leppida mida vaadelda kiire ning mida aeglase testi sooritusе all. Testid on suhteliselt lühikesed ning loetakse kiiresti testi sooritanuks tudengi, kes on testi edastanud nelja minuti jooksul alates selle alustamisest. Aeglaselt edastanuks loetakse üle nelja minuti testi sooritanud tudengid.

$$Kasu(S, kiirus) == E(S) - \left(\frac{31}{62}\right) * E(S\ kiire) - \left(\frac{31}{62}\right) * E(S\ aeglane)$$

Nüüd arvutatakse entroopiad mõlema parameetri kohta:

$$Entroopia(S\ kiire) = -\left(\frac{31}{62}\right) Log2\ \left(\frac{31}{62}\right) = 0.5$$

$$Entroopia(S\ aeglane) = -\left(\frac{31}{62}\right) Log2\ \left(\frac{31}{62}\right) = 0.5$$

Seega asendades tulemused valemisse "Kasu(S, kiirus)", saab tulemuseks:

Kasu (S, kiirus) = 0.555 - (31/62 * 1/2) - (31/62 * 1/2) = 0.055

Kuna antud kasu on väga madal (ligi 0), siis seda parameetrit ei arvesta algoritmi väljatöötamisel.

7.3.5 Informatsiooni kasulikkus Moodle külastamise põhjal

Eelnevalt tuleb kokku leppida, et tudengid, kes külastavad Moodlet vähemalt 3 kord nädalas võib lugeda tihti külastajateks. Ülejäänutele tuleb teha märge harva külastajad.

$$Kasu(S, Moodle\ külastamine) == E(S) - \left(\frac{55}{62}\right) * E(S\ tihti) - \left(\frac{7}{62}\right) * E(S\ harva)$$

Nüüd arvutatakse entroopiad mõlema parameetri kohta:

$$Entroopia(S\ tihti) = -\left(\frac{55}{62}\right) Log2\ \left(\frac{55}{62}\right) - \left(\frac{55}{62}\right) Log2\ \left(\frac{55}{62}\right) = 0.153$$

$$Entroopia(S\ harva) = -\left(\frac{7}{62}\right) Log2\ \left(\frac{7}{62}\right) 0.355$$

Seega asendades tulemused valemisse "Kasu(S, Moodle külastamine)", saab tulemuseks:

Kasu (S, Moodle külastamine) =0.555 - ((55/62 * 0.153) - (7/62 * 0.355)) = 0.459

7.3.6 Järeldused

Eelnevate arvutuste põhjal on näha, millistest sisendandmetest on tudengi väljalangevuse ennustamisel kõige suurem abi. Kuna kõik tulemused peale kahe ühe on üle nulli, siis on kõikidest ülejäänud parameetritest kasu. Tulemuste ennustamisel ei olnud kasu vabatahtlike ülesannete sooritamisest ning ülesannete lahendamise ajast.

Vabatahtlike ülesannete sooritamisest oleks olnud kasu siis, kui vähemalt üks tudeng oleks vähemalt ühe ülesande sooritanud. Praeguste informaatika aine sisendandmete korral ei ole vabatahtlike ülesandeid vaja arvestada, kuna see ennustamise tulemust ei muudaks.

Loogiline on ka, et ülesannete lahendamise aeg ei ole korrelatsioonis tudengi katkestamisega. Selle parameetri kasulikkuse kohta võiks tegelikult kirjutada eraldi uurimustöö, kus vaadeldakse parameetrit pikemate testide korral. Antud testid on nii lühikesed, et tulemus võis muutuda seetõttu ebaadekvaatseks. Algoritmi koostamisel testide lahendamise aega seega ei arvestata.

Kasulikkust ei ole enam tulevikus vaja uuesti arvutada, kui ei teki uusi parameetreid. Info kasulikkus ei ole erinevate sisendandmetega oluliselt muutuv parameeter. Seega võib arvestada vajalikuks infoks kõiki eelpool toodud parameetreid ka järgnevatel aastatel ning teistel kursustel.

7.4 Funktsionaalsed nõuded

Esmalt tuleb kaardistada funktsionaalsed nõuded algoritmile. Järgnevalt on välja toodud süsteemile esitatavad nõuded, mida süsteem kindlasti peab täitma.

1. Leida tudengid kes sooritasid testid alla 50% .
2. Suurendada iga testi eest alla 50% saanud tudengite riskigruppi.
3. Leida tudengid, kes külastavad süsteemi kõige harvemini.
4. Lisada 15 kõige harvemini kursust külastavat tudengit riskigruppi.
5. Algoritm peab hindama tingimusi formaadis "kui-siis".
6. Algoritm peab hindama kõiki hinnetelehel olevaid tulemusi.
7. Algoritm peab väljundina andma õppejõule konsooli tulemused, kus on riskantsete õpilaste nimed ning riskigrupp.

Kõik eelnevad sammud on vajalikud, et ennetada tudengite väljalangemist. Järgnevas peatükis on täpsemalt välja toodud töös kasutatavad loogikareeglid.

7.5 Loogikareeglid tulemuste kohta:

Loogikareeglites on kasutatud punast ja rohelist värvi parema mõistmise huvides. Rohelised tegevused langetavad tudengi riskigrupi punkte, punased aga suurendavad seda. Hinnetelehel on kokku 9 testi, 5 ülesande ning 5 *hotpotatoes* testi tulemust.

- IF test1 = A THEN riskPoints-- ELSE riskPoints ++

- IF test2 = A THEN riskPoints-- ELSE riskPoints ++
- IF test3 = A THEN riskPoints-- ELSE riskPoints ++
- IF test4 = A THEN riskPoints-- ELSE riskPoints ++
- IF test5 = A THEN riskPoints-- ELSE riskPoints ++
- IF test6 = A THEN riskPoints-- ELSE riskPoints ++
- IF test7 = A THEN riskPoints-- ELSE riskPoints ++
- IF test8 = A THEN riskPoints-- ELSE riskPoints ++
- IF test9 = A THEN riskPoints-- ELSE riskPoints ++
- IF ylesanne1 = A THEN riskPoints-- ELSE riskPoints ++
- IF ylesanne2 = A THEN riskPoints-- ELSE riskPoints ++
- IF ylesanne3 = A THEN riskPoints-- ELSE riskPoints ++
- IF ylesanne4 = A THEN riskPoints-- ELSE riskPoints ++
- IF ylesanne5 = A THEN riskPoints-- ELSE riskPoints ++
- IF HPtest1 = A THEN riskPoints-- ELSE riskPoints ++
- IF HPtest2 = A THEN riskPoints-- ELSE riskPoints ++
- IF HPtest3 = A THEN riskPoints-- ELSE riskPoints ++
- IF HPtest4 = A THEN riskPoints-- ELSE riskPoints ++
- IF HPtest5 = A THEN riskPoints-- ELSE riskPoints ++

7.6 Loogikareeglid logide kohta:

Logidest tuleb vaadata millised tudengid kõige vähem süsteemi külastavad. Selle jaoks kasutatakse Java *Comparator* klassi ning võrreldakse tudengi süsteemi külastamist teiste

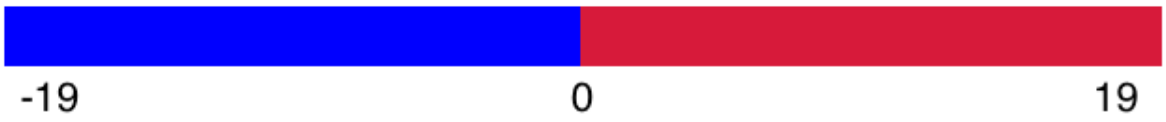
tudengitega. Kõige harvemini süsteemi külastanud 15 tudengit lisatakse riskantsete tudengite nimekirja.

- IF lastLogin > otherStudents lastLogin THEN addStudentToRiskList

7.7 Punktijaotus

Töös koostatakse algoritm, mille põhjal hinnatakse tudengi kursuselt väljakukkumise tõenäosust. Tudengid klassifitseeritakse punktide alusel kahte grupp: riskantsed ja mitteriskantsed. Iga testi tulemus mõjutab tudengi riskigruppi. Logide alusel punkte ei jaotata, vaid kuvatakse välja 15 kõige vähem süsteemi külastanud tudengid.

Informaatika aine hinnetetabelis on kokku 15 erinevat testi tulemust ning 4 ülesande tulemust. Iga testi või ülesande korral on märgitud tulemused. Kui tudengil on saadud tulemus testi või ülesande eest, siis tudengi riskigrupp väheneb ühe punkti võrra. Kui tudeng ei ole sooritanud testi või ülesannet siis tema riskigrupp suureneb. Programm hakkab välja kuvama kõiki tudengeid, kelle riskigrupp on positiivne. Mudeli skaala jääb vahemikku -19 kuni +19.



Joonis 21- Riskipunktide skaala hinnete alusel

Informaatika aine Moodle logides on kokku üle 47000 rea ühe semestri kohta. Logid koosnevad kõikidest tegevustest mida süsteemis on võimalik teha. Näiteks on logis kirjas süsteemi külastamise ajad, testide alustamise ning lõpetamise ajad. Sealt on võimalik välja lugeda kui tihti inimesed kursust külastasid. Kursuse külastajate hulgast on vaja välja leida kursust kõige vähem külastanud tudengid. Seega programm loeb sisse kõik logid ning arvutab igale tudengile süsteemi külastamise arvu. Riskigruppi kuuluvad süsteemi 15 kõige vähem külastanud tudengit.

Andmekaevandamise tehnoloogiaid kasutatakse, et avastada andmetes teatud mustreid. Andmete kaevandamise ülesandeks on mingi mustri alusel andmeid analüüsida. Kõige populaarsemad andmed, mida kaevandatakse on seosed, klassifitseerimised, klasterdamised ning võõrväärtuste kindlakstegemine. (El-Halees 2009) Antud töös käsitletakse tudengite tulemusi punases ja sinises alas võõrväärtustena (ehk väärtustena mis esinevad keskmisest kõige enam).

Punases alas olevate tudengite probleeme tuleb analüüsida ning võimaluse korral neid lahendada. Õppejõud saab tudengitele vastu tulla ning vajadusel korraldada nendele järeleaitamise tunde. Samuti on võimalik saata õpingutega raskustes olevad tudengid nõustamisele. Sinises alas olevad õpilased on aga edukamad. Antud tudengeid võib õppejõud soovitada ettevõtetele töötajateks või praktikantideks. Lisaks võivad antud tudengid mahajäänud tudengeid juhendada ning aidata.

Antud meetodi eelis on see, et madalaid hindeid on näha töö käigus. Näiteks saab juhendaja ennetada tudengite läbikukkumist enne semestri lõppu ja nendele järeleaitamisi pakkuda. Tähtis on teada, et klassifitseerimise reeglid on erinevad assotsiatsiooni reeglitest. Assotsiatsioonireeglid on olemuselt iseloomustavad reeglid (kirjeldades hetke olukorda), kuid klassifitseerimise reeglid on olemuselt ennustavad reeglid (kirjeldavad tuleviku olukorda). (El-Halees 2009)

Eelnevat põhjendust arvestades saab arendada algoritmi, mis juba tudengi õpingute ajal suudaks ennustada tudengi õppeedukust. Kuna algoritmi peab kasutama õpingute alguses siis on valitud analüüsitavaks õppeaineks informaatika, kus tudengid hakkavad tulemusi saama juba semestri alguses. Teiseks põhjuseks miks informaatika valiti on, et selle kohta leidub nii Moodles kui ka hinnetelehel kõige rohkem andmeid.

Algoritm koostatakse Java programmeerimiskeeles. Algoritmi sisendiks on informaatika tunni hinneteleht Excelis ning Moodle logi sama aine kohta CSV formaadis. Algoritm võtab arvesse punktis 7.3 leitud parameetrid, mille info kasulikkus on kõige suurem.

7.8 Pseudokood

Enne algoritmi realiseerimist tuleb see parema mõistmise huvides kirja panna pseudokoodis. Pseudokood kirjeldab inimesele arusaadavamas keeles programmi tööpõhimõtte ning tegevuste järjekorra. Lisaks näidatakse ära mis andmed programmi sisendis on. Tudengite tulemusi analüüsiva algoritmi pseudokood on esitatud järgnevalt:

```
ALGORITHM studentRiskGroup

    INPUT1 (from logs): aeg, nimi, event, komponent, sündmuseNimi,
    kirjeldus

    INPUT2 (from grades): eesnimi, perekonnanimi test1, test2,
    test3, test4, test5, test6, test7, test8, test9, ülesanne1,
    ülesanne2, ülesanne3, ülesanne4, test9, ülesanne5, test10,
    test11, test 12, test13, test14

    WHILE logsScanner HAS NEXT

        IF komponent =  " Course viewed "

            student.addToHashmap
            FOR EACH student in Hashmap

                calculate visits

    WHILE gradesScanner HAS NEXT

        FOR EACH INPUT2 calculatePoints

            IF result  = passed

                SET riskGroup--

            ELSE

                SET riskGroup ++
```

7.9 Klassifitseerimise algoritmi väljatöötamine

Algoritm on arendatud java programmeerimiskeeles ning kasutatud on objektorienteeritud lähenemist. Algoritm on viies klassis. Sisendiks on projekti ülemkausta kopeeritud CSV formaadis Moodle logid ning informaatika aine tulemused.

Algoritmi kood tervikuna asub Githubi keskkonnas ning sellele on viidatud lisade all. Järgnevalt on toodud olulisemad lõigud algoritmist ning seletatud lõikude tööpõhimõte.

Hinnetelehe alusel riskigrupi määramiseks loetakse esialgu hinded programmi sisse *scanneri* abil ning eraldatakse komade abil väljade väärtused. Kui sisendandmete lugemisel tekib viga, siis kuvatakse kasutajale välja viga *catch* blokis (kui faili ei leita). Väärtused kogutakse *values[]* massiivi.

```
try {
    Scanner hindeScanner = new Scanner(new File("IDK0091_Y Hinded.xlsx -
Hinded.csv"));
    hindeScanner.useDelimiter("/n");

    while (hindeScanner.hasNext()) {
        Hinded hinded = new Hinded();
        String data = hindeScanner.nextLine();
        data = data.replace('"', ' ');
        String values[] = data.split(",");

    }
}
    hindeScanner.close();
} catch (FileNotFoundException e) {
    System.out.println("File " + e.getMessage().replace("(No such file or
directory)", "") + "not found! ");
}
```

Nüüd kui väljad on loetud massiivi, saab iga rea põhjal midagi teha. Kõigepealt loetakse sisse tudengi eesnimi ja perekonnanimi massivi kohtadelt [1] ja [2]. Seejärel võetakse kolmandast veerust tudengi esimese testi tulemus. Kui esimese testi tulemus on saadud, siis langetatakse riskipunkte meetodiga *decreaseRiskPoints()*. Vastasel korral tudengi riskipunkte tõstetakse

meetodiga *increaseRiskPoints()*.

```
hinded.setName(values[0] + " " + values[1]);
if (!values[3].startsWith("T") & !values[3].startsWith("-")) {
    hinded.setTest1(Double.parseDouble(values[3]));
    hinded.decreaseRiskPoints();
} else {
    hinded.increaseRiskPoints();
}
```

Analoogselt eelnevaga arvutatakse riskipunktid iga ülesande kohta. Ülejäänud arvutused on näha Githubis Main klassis (Lisa 1), ning lihtsustamise mõttes ei ole sarnaseid reegleid siia kopeeritud. Ülesannete korral arvutatakse riskipunktid järgnevalt.

```
if (values[11].equals("arvestatud")) {
    hinded.setYlesanne1(4.0);
    hinded.decreaseRiskPoints();
} else if (values[11].equals("lubatud kaitsmisele")) {
    hinded.decreaseRiskPoints();
    hinded.setYlesanne1(3.0);
} else if (values[11].equals("parandada kaitsmiseks")) {
    hinded.setYlesanne1(2.0);
    hinded.increaseRiskPoints();
}
else if (values[11].equals("-")) {
    hinded.setYlesanne1(0);
    hinded.increaseRiskPoints();
}
```

Kui kõik testid ja ülesanded on läbi käidud ning riskigrupid määratud, kutsutakse välja meetod *getRiskPoints()* mis tagastab konsooli kõik riskantsemad tudengid (kelle riskigrupp oli üle nulli).

```
if (hinded.getRiskPoints() > 0) {
    System.out.println("Kõrged riskipunktid: " + hinded.getRiskPoints() + "
" + hinded.getName());
}
```

Analoogselt leitakse ka kõige tublimad tudengid. Skaala järgi oli minimaalne riskipunktide arv -19. Seega järgnev lõik kuvab välja tudengid, kellel oli kõige madalam risk kursuselt välja langeda.

```
if (hinded.getRiskPoints() <=-19) {
```

```
    System.out.println("Madalad riskipunktid: " + hinded.getRiskPoints() +
" " + hinded.getName());

 }
```

Logide skänner töötab samamoodi, ainuke erinevus on selles, et logide puhul ei hakata riskigruppe määrama. Andmed loetakse programmi ning tudengi nimi ning süsteemi külastamine lisatakse *hashmap* andmestruktuuri. Hashmap on andmestruktuur, kus igal sisestatud elemendil on võti ja väärtus. Selle abil on lihtne tudengite süsteemi külastamist kokku arvutada järgneva programmilõiguga.

```
if (student.getEventName().equals(" Course viewed ")) {
    studentCountList.add(studentID);
    //Scanneri andmed hashmapi (viimane väärtus kirjutatakse üle)
    hm.put(student.getName(), ((CountItemsList<String>)
studentCountList).getCount(studentID));
}
```

Antud lõigus loeb *getCount()* meetod kokku kui palju tudeng on Moodle keskkonda külastanud. Logidest otsitakse kõiki sündmusi kus sündmuse nimi on " Course viewed ", ehk kursust vaadati.

7.10 Algoritmi väljund ja tulemused

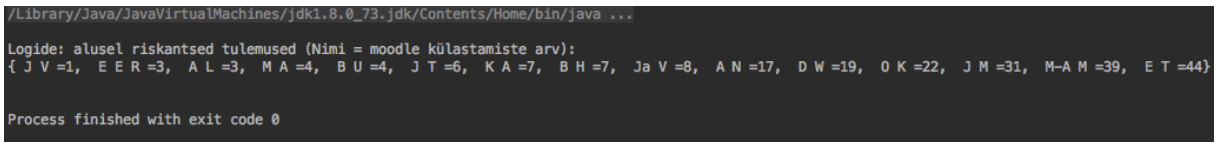
Informaatika kursusele osales 62 tudengit. Algoritm hindas neid tudengeid Moodle logide ja hinnete tabeli põhjal. Järgneval ekraanipildil on näha programmi väljund hinnete tabeli põhjal.

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_73.jdk/Contents/Home/bin/java ...

Hinnete alusel riskantsed tudengid:
Kõrged riskipunktid: 14.0 Eesnimi Perekonnanimi
Kõrged riskipunktid: 19.0 K A
Kõrged riskipunktid: 13.0 R J
Kõrged riskipunktid: 9.0 O K
Kõrged riskipunktid: 15.0 M-A M
Kõrged riskipunktid: 3.0 J M
Kõrged riskipunktid: 5.0 S O
Kõrged riskipunktid: 17.0 E T
Kõrged riskipunktid: 19.0 D W

Process finished with exit code 0
```

Joonis 22 - Algoritmi tulemus hinnete põhjal

Siit on näha, et väljalangemise ohus on 8 tudengit. Anonüümsuse huvides on tudengite nimed asendatud nime esitähtede initsiaalidega. Väljundilt on näha ka tudengite riskigrupp. Maksimaalne riskigrupp antud juhul on 19.

Logide põhjal on nimekirjas näha 15 riskantset tudengit. Üldiselt nimed korduvad hinnete tulemusega, mis annab kinnitust, et info kasulikkuse hindamine ID3 algoritmiga on tulemust andnud. Järgneval pildil on näha logide analüüsil tekkinud tulemuse väljund.



Joonis 23 - Algoritmi tulemus logide põhjal

Jooniselt 23 on näha, et kõige riskantsemate tudengite juurde on kirjutatud nende kursuse külastamise arv. Logide tulemustes on arvestatud samuti asjaoluga, et tudengite tulemused on delikaatsed isikuandmed. Seetõttu on nimed anonüümsuse huvides asendatud esitähtede initsiaalidega.

Algoritm suutis hinnete alusel tuvastada 100% tudengitest kes on väljalangemise ohus. Logide abil oli algoritm võimeline tuvastama 40% riskantsetest tudengitest. Seega algoritm on efektiivsem hinnete analüüsimisel.

8. Ettepanekud edaspidiseks ning algoritmi rakendamine

Algoritmi saavad enda õppetöös kasutada õppejõud, kelle kursus on Moodles ning kes on valmis seda vastavalt enda ainele ümber kohandama. Algoritmi sisendiks tuleb anda CSV formaadis Moodle logid ning õppeaine tulemused. Samuti tuleks jälgida, et tulemused oleksid samas formaadis nagu antud töös. Sisendina kasutatud tabelid on saadaval Lisa 1 oleval lingil. Vajadusel tuleb hinnetelehte korrigeerida töös kasutatud CSV faili sarnaseks, kuid logid on standardsed. Kui mõnes aines on teste või teadmiste kontrolle vähem, kui informaatika aines, siis tuleb hinnete väljad jätta tühjaks. Algoritm arvutab seejärel välja kõige mahajäänumad õppurid antud kursusel ning annab õppejõududele nende kohta infot, et väljalangevust ennetada. Selle tulemusena saavad õppejõud informatsiooni riskantsetest tudengitest ning neil on võimalus ennetada tudengite väljalangevust.

Edasi võiks hakata üle vaatama Moodle logide koostamise programmi. Võimaluse korral oleks hea, kui Moodle logisid modifitseeritaks nii, et igal real on konkreetne info. Hetkel kuvatakse mõnel real välja täispikkasid lauseid ning nende sorteerimine on asjatu töö. Lisaks on Moodle logides osad veerud inglisekeelsed, teised jällegi eestikeelsed. Kui Moodle logid on standardiseeritud, siis on palju lihtsam sarnaseid algoritme koostada või olemasolevat edasi arendada. Samuti võiksid logid eristada õppejõude ning tudengeid.

Pärast eelnevaid samme võib algoritmi kasutusele võtta mõnes e-õppe keskkonnas tudengite aine katkestamise ennetamiseks. Programmile oleks soovitatav arendada ka graafiline kasutajaliides, mis teeb väljundi kuvamise kaasaegsemaks. Algoritmi on võimalik kohandada ka nii, et väljakukkumise riskist teavitatakse tudengit ennast.

9. Kokkuvõte

Töö peaeesmärk oli klassifitseeriva algoritmi abil ennetada õppurite väljalangemist informaatika ainelt. Kõrvaleesmärkideks oli arendada algoritm ning see dokumenteerida piisavalt, et seda saaks ka teistes ainetes kasutada.

Töös analüüsiti erinevaid klassifitseerimise- ning ennustusalgoritme. Jõuti järeldusele, et olemasolevate algoritmide abil on raske ennustada tudengite püsimajäämist ülikoolis. Seetõttu oli vaja arendada uus algoritm, mis ennustaks tudengite hinnete ning Moodle logide abil õppurite võimalikku katkestamist.

Selle saavutamiseks otsustati ID3 algoritmi info kasulikkuse põhimõtte järgi millised andmed on relevantsed tudengi õppeedukuse määramiseks. Antud andmete atribuudid võeti Java programmis algoritmi sisendiks ning nende põhjal tegi algoritm järelduse, millised tudengid võivad informaatika aine läbi kukkuda.

Informaatika aines osales kokku 62 õpilast, kellest 8 katkestasid aine sellel semestril. Algoritm suutis hinnetena antud sisendandmete põhjal nendest õpilastest 8 riskigruppi määrata, mis teeb algoritmi täpsuseks hinnetetabeli põhjal 100%.

Logide põhjal suutis algoritm leida 15 riskantset inimest, kellest 6 kuulusid aine katkestanute hulka. Tulemus on 40% , kuid arvesse peab võtma, et logid ei erista õppejõude ega tudengeid. Seega 15 riskantse inimese hulka sattusid ka õppejõud. Kui Moodle suudaks logides eristada tudengeid õppejõududest, oleks algoritmi täpsus suurem. Tegelikult see ei ole probleem, kuna enamus riskantseid tudengeid siiski leiti. Hetkel tuleb õppejõududel tulemustest välja sorteerida tudengid.

Kahe sisendfaili analüüsimise põhjal võib algoritmi keskmiseks täpsuseks hinnata 70%, mis on üpris tugev tulemus.

Summary

The main objective of this work was to predict students dropout from the course of informatics with classification algorithm. Two side objectives were to develop an algorithm and document it in the way it is also usable in other courses .

In this thesis different classification and prediction algorithms were analyzed. The conclusion was made that it is hard if not impossible to predict students dropout with existing algorithms. Therefore it was necessary to develop a new algorithm that could predict student dropout from the course based on their grades and Moodle logs.

The most relevant data to predict student dropout was found based on ID3 algorithms information gain principle. The attributes of this data was taken as input in Java program. The conclusion was made which students may drop out based on the results of the algorithm.

There were 62 students enroled to informatics course of whom 8 dropped out. The developed algorithm was able to predict 8 of them, which makes the accuracy of the algorithm 100%.

Algorithm was able to identify 15 risky students based on the logs, of whom 6 actually dropped out of the course. The accuracy is 40%. It should be take into count that students are not distinguished from students. Therefore there were teachers in the peoples data set. If Moodle was able to distinguish students from teachers the accuracy would have been higher. It is not actually a problem, because it is possible to sort out the students manually.

Based on the two input files, we can say that the average accuracy of the algorithm is 70%, which is pretty good result.

Kasutatud kirjandus

Adhatrao, Kalpesh, Aditya Gaykar, Amiraj Dhawan, Rohit Jha, ja Vipul Honrao. „PREDICTING STUDENTS' PERFORMANCE USING ID3 AND C4.5 CLASSIFICATION ALGORITHMS." *International Journal of Data Mining & Knowledge Management Process*, 2013: 41-42.

Ahmad, Fadhilah, Nur Hafieza Ismail, ja Azwa Abdul Aziz. *Hikari Ltd.* 02. 11 2015. a. http://www.m-hikari.com/ams/ams-2015/ams-129-130-2015/p/ahmadAMS129-130-2015-2.pdf (kasutatud 07. 04 2016. a.).

Baker, Ryan S., David Lindrum, Mary Jane Lindrum, ja David Perkowski. *Columbia University in the City of New York.* 2015. http://www.columbia.edu/~rsb2162/2015paper41.pdf (kasutatud 17. 04 2016. a.).

Bittlingmayer, A. M. *stackoverflow.com.* 2016. http://stackoverflow.com/questions/10059594/a-simple-explanation-of-naive-bayes-classification (kasutatud 23. 04 2016. a.).

Brownlee, Jason. *A Tour of Machine Learning Algorithms.* 25. 11 2013. a. http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ (kasutatud 24. 04 2016. a.).

Dankel, Dr. Douglas. *The ID3 Algorithm* . 1997. http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm (kasutatud 23. 04 2016. a.).

Eesti rakendusuuringute keskus Centar. *Tudengite õpingute katkestamise põhjused IKT erialadel.* 2015. http://www.centar.ee/uus/wp-content/uploads/2015/07/IKT-katkestajate-uuringu-l%C3%B5ppraport-veebi.pdf (kasutatud 05. 05 2016 . a.).

El-Halees, Alaa Mustafa. *Researchgate.* 2009. https://www.researchgate.net/profile/Alaa_El-Halees/publication/228571634_Mining_Students_Data_to_Analyze_Learning_Behavior_A_Case_Study/links/5611898108ae4833751ba666.pdf (kasutatud 08. 03 2016. a.).

FORMATEX. *Current Developments in Technology-Assisted Education.* 2006. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.119.2854&rep=rep1&type=pdf (kasutatud 08. 03 2016. a.).

Instituut, Eesti Keele. *IT terministandardi sõnastik.* http://eki.ee/dict/its/ (kasutatud 09. 02 2016. a.).

Kabakchieva, Dorina. „Predicting Student Performance by Using Data Mining Methods for Classification." rmt: *Cybernetics and Information Technologies, Volume 13, Issue 1*, – Dorina Kabakchieva, 61–72. Sofia: Sofia University, 2013.

Kay, Judy, ja Susan Bull. „New Opportunities with Open Learner Models and Visual Learning Analytics." rmt: *Artificial Intelligence in Education*, monteerinud: Cristina Conati, Neil Heffernan, Antonija Mitrovic ja M. Felisa Verdejo, 666. Madrid: Springer International Publishing, 2015.

Khedr, Ayman E., Amira M. Idrees, ja Ahmed I. El Seddawy. *Wiley Online Library.* 2016. http://onlinelibrary.wiley.com/doi/10.1002/widm.1177/epdf (kasutatud 23. 04 2016. a.).

Kickmeier-Rust, Dr. Michael. *Lea's box.* 2014. http://css-kmi.tugraz.at/mkrwww/leas-box/olm.html (kasutatud 29. 02 2016. a.).

Márquez-Vera, Carlos, Alberto Cano, Cristobal Romero, Amin Yousef Mohammad Noaman, Habib Mousa Fardoun, ja Sebastian Ventura. *Wiley Online Library.* 16. 11 2015. a. http://onlinelibrary.wiley.com/doi/10.1111/exsy.12135/full (kasutatud 09. 04 2016. a.).

Milani, Christian, ja Riccardo Mazza. *Researchgate.* 2004. https://www.researchgate.net/profile/Christian_Milani/publication/228708439_Gismo_a_grap

hical_interactive_student_monitoring_tool_for_course_management_systems/links/0c960522f2507ce803000000.pdf (kasutatud 07. 03 2016. a.).
Moodle hitsa. 2016. https://moodle.hitsa.ee/ (kasutatud 01. 03 2016. a.).
openeducation. *Moocsandco.* 03 2014. a. http://www.moocsandco.com/sites/default/files/elearning%2037.pdf#page=7 (kasutatud 14. 04 2016. a.).
Rüga, Rauno. „Tallinna Tehnikaülikooli Raamatukogu Digikogu.“ *TTÜ Raamatukogu.* 06 2014. a. http://digi.lib.ttu.ee/i/?2001 (kasutatud 15. 04 2016. a.).
Raj Kumar, Dr. Rajesh Verma. *Academia.edu.* 08 2012. a. http://www.academia.edu/4863068/Classification_Algorithms_for_Data_Mining_A_Survey (kasutatud 10. 04 2016. a.).
Sivakumar, Subitha, Sivakumar Venkataraman, ja Rajalakshmi Selvaraj. „Predictive Modeling of Student Dropout Indicators in Educational Data Mining using Improved Decision Tree.“ *Indian Journal of Science and Technology*, 01 2016: 1-5.
Soomo learning. *Soomo learning.* 2016. http://www.soomolearning.com/blogs/posts/updated-course-analytics-dashboard (kasutatud 29. 04 2016. a.).
Stackexchange. *Stackexchange.* 03 2014. a. http://stats.stackexchange.com/questions/52216/pros-and-cons-of-clustering-algorithms (kasutatud 30. 04 2016. a.).
Statistikaamet. *Eesti statistika.* 2014. http://pub.stat.ee/px-web.2001/igraph/MakeGraph.asp?onpx=y&pxfile=HT3062016415444833.px&PLanguage=2&menu=y&gr_type=1 (kasutatud 15. 04 2016. a.).
TechTarget. *TechTarget IT encyclopedia.* 08 2013. a. http://whatis.techtarget.com/definition/massively-open-online-course-MOOC (kasutatud 01. 05 2016. a.).
Tepandi, Jaak. *Loengud, juhendamised, tööd* . 23. 03 2016. a. http://tepandi.ee/is-loeng.pdf (kasutatud 08. 04 2016. a.).
TTÜ. *Õppeinfosüsteem.* https://ois.ttu.ee/pls/apex/f?p=1000:32:3093580948747101::::: (kasutatud 10. 02 2016. a.).
TTÜ Õppeinfosüsteem. *Tallinna Tehnikaülikooli Õppeinfosüsteem.* 2016. https://ois.ttu.ee/ois2/docs/HKRIT.100101/IDK0091_hindamine_est.pdf (kasutatud 27. 04 2016. a.).
TTÜ Õppeinfosüsteemi statistika. *Tallinna Tehnikaülikooli õppeinfosüsteem.* 2016. https://ois.ttu.ee/pls/apex/f?p=1000:8:279498014993701::NO (kasutatud 30. 04 2016. a.).
Vallaste, Heikki. *e-Teatmik: IT ja sidetehnika seletav sõnaraamat.* 2000. http://www.vallaste.ee (kasutatud 09. 02 2016. a.).
Walber. „creativecommons.“ 2014. https://creativecommons.org/licenses/by-sa/4.0/deed.en (kasutatud 17. 04 2016. a.).

Lisa 1. Rakenduse lähtekood

Rakenduse lähtekood ja dokumentatsioon asub aadressil: https://github.com/oskar202/Moodle