

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Mihkel Keso 222548IAIB

Rakenduse loomine tudengite edenemise jälgimiseks masinõppe meetodite abil

Bakalaureusetöö

Juhendaja: Ago Luberg

Ph.D

Tallinn 2025

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: Mihkel Keso

04.06.2025

Annotatsioon

Käesoleva bakalaureusetöö eesmärk on arendada Tallinna Tehnikaülikoolile süsteem, mis võimaldab (abi)õppejõududel analüüsida tudengite edenemist programmeerimisaines. Loodud lahendus kogub andmeid igapäevastest tagasisideküsimustikest ja hinnetest, töötleb neid, kasutades erinevaid masinõppemeetodeid, ning esitab tulemused visuaalselt selleks, et need oleksid õppejõule kiirelt mõistetavad ja otsustusteks kasutatavad. Süsteemi abil saab õppejõud varakult tuvastada tudengeid, kes võivad vajada lisatuge edasisel õppimisel.

Tudengite tagasiside sisaldab nii valikvastustega kui ka avatud tekstiväljaga küsimusi, mille analüüsiks kasutatakse keelemudeleid. Hinnete põhjal rakendatakse masinõppemeetodeid, mis võimaldavad prognoosida tudengi lõpptulemusi. Süsteem ühendab erinevad andmeallikad ühtseks tulemuseks ja aitab selle abil õppejõul langetada paremaid otsuseid selle kohta, mis tudengi poole pöörduda.

Rakendus koosneb mitmest komponendist ja on loodud nii, et seda oleks võimalik lihtsalt laiendada ja kohendada ka teiste kursuste jaoks tulevikus.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 40 leheküljel, 6 peatükki, 14 joonist, 7 tabelit.

Abstract

Developing an Application for Monitoring Students' Progress Using Machine Learning Methods

The goal of this bachelor's thesis is to develop a system for Tallinn University of Technology that enables (assistant) instructors to analyze students' progress in an introductory programming course. The developed solution collects data from weekly feedback questionnaires and grades, processes it using various machine learning methods, and presents the results visually to ensure they are quickly understandable and actionable for instructors. The system helps identify students who may require additional support early in the course.

The feedback includes both multiple-choice and open-ended text questions, which are analyzed using language models. Based on students' grades, machine learning techniques are used to predict final course outcomes. The system integrates various data sources into a unified view, supporting instructors in making informed decisions about which students may need attention.

The application was tested by using data from previous courses in earlier years. The results show that machine learning models can predict student performance based on grades with increasing accuracy as more data becomes available throughout the semester. Language models analyzing student feedback also demonstrated good accuracy, as students often write openly about their progress and emotions. This accuracy also improved as the course progressed and more feedback was collected. Furthermore, the study confirmed that combining diverse data types can provide strong indications of which students may need support to complete the course successfully.

Some predictions are inaccurate because students often don't put much effort into their feedback responses, leading to incomplete or misleading insights. Additionally, the language model sometimes places too much importance on overly detailed information in the text,

which might not truly reflect the student's overall progress or learning needs. Future work could focus on developing the language model to better distinguish between meaningful and less relevant feedback.

The application consists of multiple components and is designed to be easily extendable and adaptable for other courses in the future.

The thesis is in Estonian and contains 40 pages of text, 6 chapters, 14 figures, 7 tables.

Lühendite ja mõistete sõnastik

AI	Artificial Intelligence, tehisintellekt
Apache Superset	Apache Superset, andmete visualiseerimise tööriist
API	Application Programming Interface, rakendusliides
CI/CD	Continuous Integration / Continuous Deployment, järjepidev integreerimine ja kasutuselevõtt
Docker	Docker, konteineripõhine virtualiseerimise platvorm
GitLab	GitLab, versioonihalduse platvorm
GPT-4o mini	GPT-4o mini, OpenAI keelemudel
kNN	k-Nearest Neighbors, k-lähimate naabrite algoritm
Llama3	LLaMA3, Meta keelemudel
LLM	Large Language Model, suur keelemudel
MAE	Mean Absolute Error, keskmine absoluutne viga
Moodle	Moodle Modular Object-Oriented Dynamic Learning Environment, veebipõhine õpiahaldussüsteem
NLP	Natural Language Processing, loomuliku keele töötlus
NumPy	Numerical Python, Python'i teek arvutusteks massiividega
Pandas	Pandas, Python'i teek tabelandmete töötlemiseks
PostgreSQL	PostgreSQL, relatsiooniline andmebaasisüsteem
R ²	Coefficient of Determination, determinatsioonikordaja
Scikit-learn	Scikit-learn, Python'i masinõppe teek
SMOTE	Synthetic Minority Oversampling Technique, sünteetilise vähemuse üleproovimise tehnika
SVR	Support Vector Regression, tugivektorregressioon
XGBoost	Extreme Gradient Boosting, ansambelõppel põhinev masinõppe algoritm, mis koosneb otsustuspuudest

Sisukord

1	Sissejuhatus.....	11
1.1	Tudengite edenemise jälgimine	11
1.2	Projekti eesmärk	12
2	Olemasolevad lahendused	14
2.1	Näited tehisintellekti kasutusest hariduses.....	14
2.1.1	Õppimisanalüütika ja varajase hoiatamise süsteem	14
2.1.2	Kroonilise puudumise ennetamine masinõppe abil	15
2.1.3	Väljalangemise ennustamine ühe küsimustiku põhjal	15
2.1.4	Meelestatuse analüüsi ja masinõppe kombinatsioon akadeemilise edu ennustamiseks.....	16
2.2	Olemasolevate lahenduste kokkuvõte.....	16
3	Kasutatavad tehnoloogiad.....	18
3.1	Süsteemi arhitektuur.....	18
3.2	Töövahendid ja tehnoloogiad	19
4	Rakenduse arendusprotsess.....	21
4.1	Arendusvõtted	21
4.2	Andmed ja nende eeltöötlemine	21
4.2.1	Tudengid ja kursused	21
4.2.2	Iganädalased tagasisided	22
4.2.3	Hinded.....	24
4.3	Keelemudeli kasutamine	27
4.3.1	LlaMA3	28
4.3.2	GPT-4o mini.....	30
4.3.3	Viiba koostamine	32
4.3.4	Mudelite maksumus	33
4.4	Hinde ennustamine.....	34
4.5	Analüüsiandmete koondamine	36

4.6	Andmete uuendamine	36
4.7	Visualiseerimine	37
5	Tulemused	38
5.1	Hinde ennustus	38
5.2	Keelemudeli ennustus	41
5.3	Ennustuste visualiseerimine	46
5.4	Keelemudeli võrdlus inimesega	47
5.5	Tulemuste kokkuvõte	49
6	Kokkuvõte.....	50
	Kasutatud kirjandus	51
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks.....	53
	Lisa 2 – Keelemudelile etteantav ülesande kirjeldus.....	54
	Lisa 3 – Graafikud Supersetis.....	55

Jooniste loetelu

Joonis 1. Andmevahetuse diagramm.....	18
Joonis 2. Tudengite tagasiside infot hoidvad tabelid Moodle'is.	23
Joonis 3. Tagasisidetabeli struktuur rakenduse andmebaasis.	24
Joonis 4. Keskmine absoluutne viga (MAE) erinevate mudelite puhul.	35
Joonis 5. Hinde ennustamine 1. nädalal.	39
Joonis 6. Hinde ennustamine 5. nädalal.	40
Joonis 7. Keelemudeli ennustuste täpsus - Programmeerimise algkursus 2024/2025.	41
Joonis 8. Keelemudeli ennustuste täpsus - Programmeerimise algkursus 2023/2024.	42
Joonis 9. Keelemudeli ennustustäpsuse võrdlus kahel kursusel.	43
Joonis 10. Keelemudeli valed ennustused, kus piiriks on 3.....	45
Joonis 11. Näidiskoondpaneel.	47
Joonis 12. Erinevused abiõppejõu ja keelemudeli analüüsis.	48
Joonis 13. Tudengite vastused küsimusele aine tempo kohta.....	55
Joonis 14. Tudengite vastused küsimusele kulutatud aja kohta (tundides).....	55

Tabelite loetelu

Tabel 1. Esimese nädala hindeelemendid.	25
Tabel 2. Kasutatud tunnused ja nende kirjeldused.	27
Tabel 3. Näited LLaMA3 analüüsist tudengite tagasiside kohta.....	29
Tabel 4. Näited gpt4-o mini analüüsist tudengite tagasiside kohta.	31
Tabel 5. Andmebaasitabeli analysis_result struktuur.....	36
Tabel 6. API lõpp-punkt tagasiside analüüsi käivitamiseks.	37
Tabel 7. Muutus nädalate kaupa konkreetse tudengi näitel.....	44

1 Sissejuhatus

Üliõpilaste väljalangemine kõrgkoolist on nii Eestis kui ka mujal maailmas jätkuvalt tõsine probleem ja see mõjutab nii inimeste isiklikke elusid kui ka ühiskonna arengut tervikuna. Näiteks Tallinna Tehnikaülikoolis lõpetas oma õpingud 2022/23. õppeaastal nominaalajaga 47% üliõpilastest [1]. Väljalangemisel on mitmeid erinevaid põhjuseid, millest igaüks vajab omaette ja läbimõeldud lähenemist.

Eriti teravalt tuleb probleem esile IT-valdkonnas ning just programmeerimisainetes, kus ained on tihti mahukad, tempokad ja nõuavad tudengilt suurt enesedistsipliini ning järjepidevat pingutust. Paraku on nendes ainetes väljalangevusprotsent kõrge, kuna paljud tudengid, kes võiksid olla edukad, satuvad juba kursuse alguses raskustesse. Sageli ei ole küsimus mitte õpivõimekuses ega motivatsioonis, vaid pigem puudulikus toetamises või oskamatuses raskusi õigel ajal märgata.

Käesolev töö keskendubki sellele kitsaskohale: eesmärk on arendada lahendus, mis aitaks varakult tuvastada need tudengid, kellel võib tekkida raskusi programmeerimiskursuse läbimisel, ning toetada neid enne, kui motivatsioon kaob ja tekib soov õpingud katkestada. Fookus on just neil tudengitel, kellel on potentsiaal edukalt hakkama saada, kuid kes vajavad varajases etapis täiendavat tuge.

1.1 Tudengite edenemise jälgimine

Tudengite edenemise jälgimise viisid erinevad märkimisväärselt sõltuvalt koolist, kursusest ja õppejõust. Õppejõu töökoormust suurendab tihti asjaolu, et kursustel osalejate arv on suur – sageli ulatub see sadadesse. Iganädalaste koolitööde analüüsimine, tudengite edenemise jälgimine ja probleemide tuvastamine on käsitsi tehes äärmiselt aeganõudev ja ebaefektiivne. Selle tagajärjel võib mõni tudeng vajalikust toest õigeaegselt ilma jääda, mis võib viia tema väljalangemiseni.

Kuigi on väljaarendatud tööriistu, mis võimaldavad õppejõul mugavamalt näha visuaalseid

graafikuid tudengite hetkeseisudest, siis on puudus mugavatest lahendustest, mis lisaks ka ennustusi ning aitaks varakult hädasolevaid tudengeid tuvastada. Sellised tudengid ei pruugi ainult hinnete kaudu alati välja paista. Parema sisendi tudengi kohta saaks temaga rääkides, aga nagu mainitud, siis suure hulga inimeste puhul on seda ülimalt keeruline rakendada.

1.2 Projekti eesmärk

Antud projekti raames keskendutakse Tallinna Tehnikaülikoolis õpetatavale kursusele "Programmeerimise algkursus" ning üritatakse arendada rakendus, mis aitaks just selles aines varakult üles leida tudengid, kes rohkem abi vajaksid. Igal aastal osaleb kursusel ligikaudu 500 tudengit. Lisaks põhiõppejõule on kaasatud mitmeid abiõppejõude, mis loob hea eelduse personaalsemaks lähenemiseks – vaja on vaid lahendust, mis hädas olevad tudengid varakult esile toob.

Skoobis oleval kursusel täidavad tudengid igal nädalal tagasisideküsimustikke, mis pakuvad väärtuslikku sisendit õppejõule tudengite hetkeseisu kohta. Et vähendada käsitsi analüüsi koormust ning muuta sekkumine efektiivsemaks, töötatakse välja keelemudelit kasutav rakendus, mis suudab automaatselt analüüsida tudengite tagasisidet ning teha selle põhjal ennustusi võimalike probleemide kohta.

Lõputöö eesmärk on luua töövahend, mis toetab õppejõude tudengite edenemise jälgimisel ning aitab varakult märgata märke võimalikust väljalangevusest. Arendatav rakendus saab integreeruma Moodle'i¹ keskkonnaga, võimaldades koguda ja töödelda andmeid iganädalastest küsimustikest ja hinnetest. See võimaldab luua mitmekesise andmepõhise vaate iga tudengi arengule.

Töös rakendatakse masinõppemeetodeid, sealhulgas loomuliku keele töötlust (NLP), et tuvastada korduvaid mustreid, võtmesõnu ja võimalikke takistusi õppimisel. Samuti kasutatakse traditsioonilisemaid algoritme hinnete analüüsimiseks.

Et tagada lahenduse praktiline kasutatavus, kasutatakse sobivat andmete visualiseerimise tööriista, kus õppejõud saab jälgida tudengite edenemist, analüüsida masinõppe poolt tehtud ennustusi ning üldiseid kursuse trende. Selle tulemusel on lihtne leida õiged tudengid,

¹ <https://moodle.taltech.ee/>

kellele ligineda eesmärgiga neid kursusest läbi aidata.

2 Olemasolevad lahendused

Enne rakenduse arendamist on vaja teha põhjalik uuring projektidest, millel on olnud sarnane eesmärk. See on oluline, kuna nii saab mõista, milliseid lahendusi on varem kasutatud, millised meetodid on osutunud edukaks ning milliseid teid tuleks vältida. Varasemate tööde analüüs aitab selgemalt määratleda käesoleva projekti uuenduslikkust ja vajadust ning mõjutab valikuid, mida tehakse rakenduse arendamisel.

2.1 Näited tehisintellekti kasutusest hariduses

Viimastel aastatel on tehisintellekti (AI) areng toonud inimeste eludesse suuri muutusi. Samuti toimub pidevalt suuri muutusi ka hariduses, kus järk-järgult tehisaru võimekusi aina rohkem kasutusele võetakse [2]. Antud alapeatükis tuuakse mõned näited sellest, mis haridusevaldkonnas saavutatud on.

2.1.1 Õppimisanalüütika ja varajase hoiatamise süsteem

2019. aastal ilmunud artiklis „Using learning analytics to develop early-warning system for at-risk students“ uuriti põhjalikult võimalusi varajase hoiatamise süsteemi loomiseks üliõpilaste jaoks, kellel on oht aine mitte läbida [3]. Artikli autorid soovisid mõista, kuidas erinevad andmed ja meetodid võiksid aidata tuvastada riskis olevaid tudengeid juba kursuse varajastes etappides, et pakkuda vajadusel täiendavat tuge või sekkumist.

Uuring viidi läbi 76 tudengiga, kes osalesid arvutiriistvara kursusel. Selleks koguti kursuse käigus andmeid nii tudengite varasemate õpitulemuste kui ka jooksva tegevuse (näiteks osalemise ja kodutööde esitamise) kohta. Uurijad katsetasid erinevaid masinõppemeetodeid, sealhulgas logistilist regressiooni ja k-lähimate naabrite algoritmi (kNN), et ennustada tudengite lõpptulemust kursusel.

Analüüsi tulemustest selgus, et k-lähimate naabrite algoritm osutus kõige täpsemaks, võimaldades teha täpseid ennustusi juba kursuse kolmandal nädalal. Selleks ajaks oli

võimalik ligikaudu 74% täpsusega prognoosida, millised tudengid saavad aine edukalt läbitud ja millised mitte. Kursuse edenedes suurenes mudelite täpsus veelgi, mis näitab, et tudengi tegevuste ja soorituste jälgimine kursuse jooksul annab väärtuslikku infot nende lõpptulemuse ennustamiseks.

2.1.2 Kroonilise puudumise ennetamine masinõppe abil

Tehisintellekti kasutamine hariduses ei piirdu ainult lõpptulemuste ennustamisega. 2024. aastal ilmunud artiklis „Leveraging Modern Machine Learning to Improve Early Warning Systems and Reduce Chronic Absenteeism in Early Childhood“ uuriti, kuidas masinõpe võiks aidata ennetada kroonilist puudumist ning vähendada selle negatiivset mõju õpitulemustele [4].

Uuringus rakendati masinõppemeetodeid, nagu XGBoost ja SMOTE, mis pakkusid kuni 52% võrra täpsemat ennustust võrreldes traditsiooniliste meetoditega. See näitab, et ka varasemalt keeruliseks peetud probleemide (nt puudumiste ennustamine) puhul võib masinõpe pakkuda uusi ja kvaliteetseid lahendusi.

Kuigi puudumiste ennetamise uuringud on eriti olulised üldhariduses ja varasemas lapseas, on teema aktuaalne ka ülikoolides, kus puudumised võivad olla seotud väljalangemise või madala õpitulemusega.

2.1.3 Väljalangemise ennustamine ühe küsimustiku põhjal

2024. aastal Tallinna Tehnikaülikoolis kaitstud bakalaureusetöös „Väljalangemise ennustamise tööriista arendamine programmeerimisaine küsimustike põhjal“ uuriti, kas ühe küsimustiku põhjal oleks võimalik edukalt ennustada väljalangemist [5].

Töö keskendus semestri alguses tehtud küsimustikule, mille tulemuste põhjal ennustati tudengite edukust kursusel. Ennustuse täpsus oli 62%, mis on kõrge tulemus, arvestades andmeid, mis selleks hetkeks saadaval on, kuid edasiseid andmeid, mis kursuse jooksul tekkisid, enam arvesse ei võetud. See tähendab, et tudengid, kelle raskused võivad ilmnedas alles kursuse hilisemas osas, võivad jääda tähelepanuta.

Lisaks nõudis tehtud tööriist, et õppejõud laadiks ise vajalikud failid üles, mis lisas käsitsi tehtavat tööd ja vähendas lahenduse kasutamise mugavust. Käesolev projektis üritatakse

neist piirangutest üle saada, luues süsteemi, mis kogub andmeid automaatselt ning võtab arvesse ka ajas muutuvat infot.

2.1.4 Meelestatuse analüüsi ja masinõppe kombinatsioon akadeemilise edu ennustamiseks

2023. aastal Tallinna Tehnikaülikoolis kaitstud magistritöös „Integrating Sentiment Analysis and Machine Learning to Predict Students’ Academic Performances in an Introductory Programming Course“ püüti ühendada masinõpe ja meelestatuse analüüs, et ennustada tudengite edukust [6].

Töö keskendus programmeerimise kursusele ja kasutas iganädalasi küsimustikke, kuid analüüsiti eelkõige numbrilisi välju, jättes tekstilised vastused tagaplaanile. Tekstiväljad, kus tudengid saavad oma muresid ja emotsioone väljendada, võivad aga sisaldada olulist teavet probleemide varajaseks avastamiseks. Ennustustäpsus oli 70%, kuid mudel ei kasutanud varasemate nädalate andmeid, vaid ainult parajasti sisse antud vastuseid, mis võis tulemuste kvaliteeti oluliselt piirata.

Lisaks oli ka selles töös probleemiks, et andmed tuli käsitsi üles laadida, mis vähendas kasutusmugavust. Käesolevas projektis on plaanis kaasata ka tekstiväljade analüüs ja arvestada tagasiside analüüsimisel ka varasemate nädalate vastuseid selleks, et mõista tudengi arengut ja selle abil saavutada täpsemad tulemused.

2.2 Olemasolevate lahenduste kokkuvõte

Analüüsitud varasemad tööd näitavad, et tehisintellektil ja masinõppel on potentsiaali üliõpilaste õppeedukuse ja väljalangemisohu ennustamisel. Uuringutes kasutatakse väga palju erinevaid meetodeid, mille tulemused on samuti väga erinevad ning igal probleemil võib olla selle optimaalseimaks lahenduseks täiesti erinev algoritm, mis võtab arvesse just selle probleemi spetsiifilisi detaile. Näiteks analüüsitud tööde seast k-lähimate naabrite (kNN), XGBoosti ja SMOTE'i algoritmid saavutasid juba semestri varajases faasis üsna kõrge ennustustäpsuse. Üks oluline järeldus on see, et mida varem andmeid kogutakse ja analüüsitakse, seda kiiremini saab sihipärast tuge pakkuda väljalangemisohus olevatele tudengitele.

Igal lahendusel leiduvad omad puudused. Näiteks kursuse "Programmeerimise algkur-

sus" jaoks tehtud tööd ei kasutanud kas üldse mitte või täielikult ära tekstivälju, mille kaudu tudengid oma muresid väljendavad. Tekstiliste väljade kaudu väljendavad tudengid ka positiivseid aspekte kursuse kohta, mis võib õppejõule väga kasulik olla tulevaste kursuste planeerimisel ja korraldamisel. Samuti esines tehnilisi kitsaskohti, nagu vajadus andmete käsitsi üleslaadimiseks. See vähendab lahenduste kasutatavust praktikas ning on kindlasti asi, mida on võimalik automatiseerida.

Töödest ilmneb selgelt, et kõige edukamad lahendused on need, mis suudavad kombineerida mitmesuguseid andmeallikaid (nt küsimustikud, puudumised, varasemad tulemused/hin- ded), võtavad arvesse ajas muutuvat infot ning seejärel suudavad teha kvaliteetseid ennustusi. Käesolev projekt eristub varasematest lahendustest sellega, et eesmärgiks on luua auto- maatselt andmeid koguv, tekstianalüüsi kaasav ning jooksvalt ennustusi uuendav süsteem, mis vähendab käsitsi sekkumise vajadust ja pakub täpsemat ennustust/tuge probleemide varajaseks tuvastamiseks.

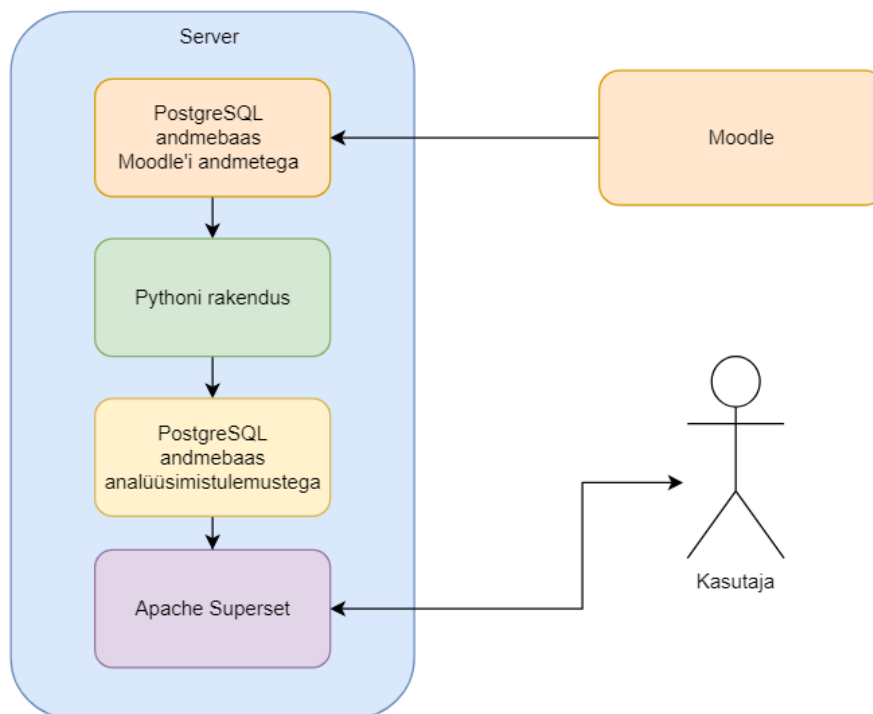
3 Kasutatavad tehnoloogiad

Käesolevas peatükis antakse ülevaade projektis kasutatavatest tehnoloogiatest ning üldisest arhitektuurist.

3.1 Süsteemi arhitektuur

Rakendus on üles ehitatud mitmekihilise arhitektuuriga. Selles sisalduvad andmetöötlus- ja analüüsimoodulid, keskeks andmebaasiks olev PostgreSQL ning visualiseerimiskiht Apache Superseti näol. Lisaks on rakendus integreeritud samas serveris asuva andmebaasiga, mis Moodle'ist igapäevaselt tudengite andmeid pärib ning seega ajakohast informatsiooni annab nii hinnete kui ka tagasiside kohta. Süsteemi komponentide vahel toimub andmevahetus lokaalses Dockeri võrgus. Rakendus on üles ehitatud kogu arhitektuuri silmas pidades, et tagada selle lihtne laiendatavus tulevikus.

Andmevahetus süsteemi erinevate osade vahel on kujutatud joonisel 1.



Joonis 1. Andmevahetuse diagramm.

3.2 Töövahendid ja tehnoloogiad

Käesolevas projektis kasutatakse mitmeid töövahendeid ja -tehnoloogiaid, mis toetavad andmete töötlemist, masinõppe mudelite rakendamist ja tulemuste visualiseerimist. Peamiseks programmeerimiskeeleks on valitud Python, kuna see on 2025. aasta seisuga enim kasutatav programmeerimiskeel ning seetõttu pakub see laialdast tuge [7]. Samuti pakub Python suurepärase mugavust, kvaliteeti ja kiirust andmetöötlusel, mistõttu sobib see hästi ka masinõppe rakendusteks [8].

Andmete analüüsimisel kasutatakse Pandas teeki, mis võimaldab tabelkujul andmeid mugavalt töödelda ja analüüsida [9]. Keerukamate arvutuste ja massiividega töötamiseks kasutatakse NumPy teeki, mis on kasulik näiteks tudengite hinnete analüüsimisel ja muudes matemaatilistes protsessides [10]. Masinõppemudelite treenimiseks ja hindamiseks kasutatakse Scikit-learn teeki, mis pakub laia valikut meetodeid ja võimaldab kiiresti katsetada erinevaid algoritme [11].

Lisaks numbrilistele analüüsidele kasutatakse tekstianalüüsi jaoks OpenAI GPT mudeleid, mis on näidanud häid tulemusi näiteks sentimentide tuvastamisel ja tekstide klassifitseerimisel [12].

Projektis hoitakse tudengite andmeid ja analüüsitulemusi PostgreSQL andmebaasis, mis on üks maailma populaarsemaid relatsioonilisi andmebaase ja pakub usaldusväärset viisi andmete hoiustamiseks [13]. Visualiseerimiseks kasutatakse Apache Superset tööriista, mis võimaldab kiiresti ja mugavalt luua graafikuid ja aruandeid, pakkudes õppejõule lihtsat ligipääsu tudengite edenemise andmetele [14].

Projekti mugavamaks haldamiseks ja skaleeritavuse tõstmiseks kasutatakse Dockerit, mis võimaldab süsteemi erinevaid osi konteinerites käivitada nii, et igas konteineris on isoleeritud keskkond [15]. See võimaldab konteineritel töötada ühtselt erinevatel platvormidel.

Rakenduse arendamiseks ning automatiseerimiseks kasutatakse GitLab CI/CD (Continuous Integration and Continuous Development) tööriistu. Rakenduse kood paikneb GitLabis ning igal muudatusel käivitatakse automaatne töövoog, mis tagab värskema versiooni jõudmise serverisse.

Kõigi nende tööriistade ja tehnoloogiate kombineerimine võimaldab rakenduse töökindlat toimimist ja lihtsat haldamist. Nii tekib tehniline alus, millele saab arendada masinõppe rakenduse tudengite edukuse analüüsimiseks

4 Rakenduse arendusprotsess

Käesolev peatükk annab ülevaate süsteemi arendamisest, sellega kaasnenud probleemidest ning kasutatud lahendustest tekkinud murede lahendamiseks.

4.1 Arendusvõtted

Projekti haldamiseks kasutati Tallinna Tehnikaülikooli hallatavat GitLab-i. Arendusprotsess jagunes verstapostideks (ingl. k *milestone*), mis omasid pileteid (ingl. k *tickets*), mis perioodi lõpuks tehtud pidid saama. Iga pilet sisaldas dokumentatsiooni ehk täpset kirjeldust, mida tegema peab.

Lisaks on mugava arenduse saavutamiseks kasutatud CI/CD lahendust, tänu millele kaob vajadus käsitsi koodi uuendada.

4.2 Andmed ja nende eeltöötlamine

Käesolev alampeatükk annab ülevaate andmete, mida projektis kasutatakse, struktuurist ning töötlemisest, mis on vaja enne analüüsimist teha.

Kõik andmed päritakse Moodle'ist kooliserveris asuvasse andmebaasi. See protsess ei ole implementeeritud selle projekti raames. Tänu sellele, et andmed samas serveris asuvad, on rakendusel väga lihtsalt ja kiiresti võimalik need kätte saada, mis muudab analüüsimise palju efektiivsemaks.

4.2.1 Tudengid ja kursused

Kõige aluseks on andmed tudengite ja kursuste kohta, mis võimaldavad tulemusi siduda konkreetsete õppijate ja ainetega. Iga tudeng on seotud vähemalt ühe kursusega ning analüüside läbiviimiseks on oluline teada, millisel kursusel ta osaleb ja millised on tema varasemad tulemused.

Terve arendusprotsessi ajal järgiti seda, et andmed oleks kaitstud ning ei leviks kuhugi väljapoole. Keelemudeliga vesteldes eemaldati päringust kõik tudengi andmed. Andmetele pääsevad ligi ainult õppejõud.

4.2.2 Iganädalased tagasisided

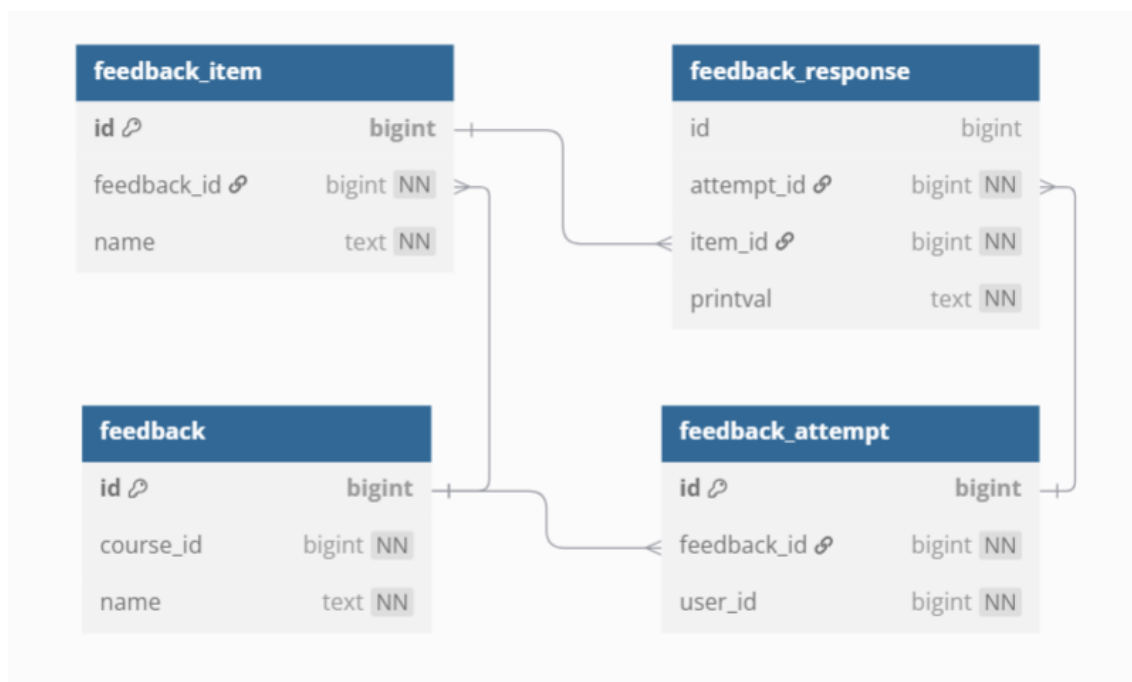
Kursuse "Programmeerimise algkursus" käigus jagatakse igal nädalal tudengitele tagasisidevorm, mida soovi korral täita saab. Tagasisidevorm, mida kasutati 2024. aasta sügissemestril, koosnes järgmistest küsimustest:

1. Küsimus: Milline oli Su enesetunne seoses selle algkursuse teemaga?
Valikvastused: "Väga positiivne", "Pigem positiivne", "Neutraalne", "Pigem negatiivne", "Väga negatiivne".
2. Küsimus: Kas selle teema raames õppisid programmeerimise kohta midagi kasulikku?
Valikvastused: "Õppisin väga palju", "Õppisin natuke", "Ei õppinud üldse", "Ei oska öelda".
3. Küsimus: Milline on aine tempo Sinu jaoks?
Valikvastused: "Liiga kiire – liiga rasked ülesanded", "Liiga kiire – liiga suur ülesannete hulk", "Paras", "Pisut aeglane".
4. Küsimus: Kui kaua Sul selle teema ülesannete lahendamiseks aega läks (tundides)?
Vastus sisestatakse arvuna vahemikus 0–168.
5. Küsimus: Kuidas hindaksid selle teema ülesannet 10-palli skaalal?
Valik: täisarv vahemikus 1 kuni 10.
6. Küsimus: Positiivsed mõtted ja emotsioonid seoses selle teema ja ülesandega.
Avatud vastus.
7. Küsimus: Negatiivsed mõtted ja emotsioonid seoses selle teema ja ülesandega.
Avatud vastus.
8. Küsimus: Kas käisid jooksva nädalal loengus või praktikumides kohal?
Valikvastused: "Jah", "Ei, sest ei leidnud aega", "Ei, sest ei olnud vaja".
9. Küsimus: Tagasiside praktikumidele ja abistamisele.
Avatud vastus.
10. Küsimus: Siin saad soovi korral abiõppejõudusid nimeliselt kiita (vali kuni 3 nime).
Abiõppejõudude nimekiri, kust saab valida.

Enamik küsimusi on valikvastustega, mis teeb ennustamise struktuuri poolest lihtsamaks. Siiski leidub ka avatud vastustega küsimusi, millele tudeng võib vastata, aga ei pruugi. Isegi kui tudeng vastust ei kirjuta, salvestatakse vastus siiski rakenduse andmebaasi tühja väärtusena. Sellest saab keelemudel hiljem järeldada, et tudeng ei soovinud või ei osanud oma mõtteid väljendada, mis omakorda võib viia mingisuguste järeldusteni.

Tekstilised vastused pakuvad ennustamisel lisaväärtust, kuna need pakuvad tudengile võimalusi väljendada positiivseid ja negatiivseid asju just sellisel viisil, nagu tema soovib. Keelemudelid suudavad nendes ära tunda korduvaid mustreid, hoiakuid ning seostada neid õppimise edukusega. Selline info loob parema pildi tudengi seisust ja täiendab numbriliste vastuste põhjal tehtavat analüüsi.

Andmeid tagasisideküsimustike kohta hoitakse Moodle'i andmebaasis, kasutades paljusid tabelleid. Antud projektis oli tagaside jaoks vaja infot kätte saada neljast erinevast tabelist, mille struktuur on näha joonisel 2.



Joonis 2. Tudengite tagasiside infot hoidvad tabelid Moodle'is.

Tagasiside analüüsi hõlbustamiseks koondati erinevatest allikatest saadud andmed ühte tabelisse, mida hoitakse rakenduse andmebaasis. Lõpptulemuses esindab iga rida tabelis ühe tudengi vastust ühele konkreetsele küsimusele. Tabeli struktuuri kujutab joonis 3.

Veeru nimi	Kirjeldus
feedback_id	Vastuse unikaalne identifikaator
course_id	Kursuse identifikaator, mille kohta tagasiside anti
student_id	Tudengi identifikaator, kes vastuse andis
question_id	Küsimuse identifikaator, millele vastati
answer_text	Tudengi vastus tekstina
week	Nädala number, millal tagasiside anti

Joonis 3. Tagasisidetabeli struktuur rakenduse andmebaasis.

Kuna küsimustikud võivad varieeruda ning ühele struktuurile põhinedes ei oleks lahendus väga jätkusuutlik, arendati rakenduses loogika, mis võimaldab analüüsida igasugust küsimustikku. Selleks loodi eraldi tabel, mis hoiab erinevaid küsimusi ja nende identifikaatoreid. Sarnane tabel esines ka Moodle'i andmebaasis, aga seal oli iga küsimus seotud konkreetse küsimustikuga. See tähendab, et isegi kui kahes erinevas küsimustikus esitati täpselt sama küsimus, loodi sellele iga küsimustikus eraldi kirje koos unikaalse identifikaatoriga. Selle tulemusel esines sama sisuga küsimusi Moodle'i andmebaasis palju kordi, igaüks erineva ID-ga. See muudab analüüsi keerukaks juhul, kui tekib vajadus analüüsida konkreetse küsimuse tulemusi erinevatel nädalatel.

4.2.3 Hinded

Kursuse "Programmeerimise algkursus" Moodle'i hindersaamat koosneb väga paljudest erinevatest elementidest, mis kajastavad tudengite edenemist. Üldiselt saavad tudengid igal nädalal punkte erinevate teooriatestide eest kui ka praktiliste programmeerimisülesannete eest. Tabel 1 kujutab näitena ette kõik esimese nädala hindeelemendid.

Kuna tudengitel on teooriateste võimalik sooritada lõpmatu arv kordi, tähendab see tavaliselt seda, et nende eest saadakse maksimaalsed punktid. See omakorda tähendab, et tudeng, kes on maksimaalsest skoorist näiteks 0,1 punkti vähem saanud, võib hiljem masinõppe abil tehtud ennustustes paigutada tagumisse poolde. Selline väike erinevus ei pruugi peegeldada tegelikku erinevust tudengite teadmiste ja pingutuse vahel, kuid võib mõjutada mudeli hinnangut nende edasisele edukusele. Selline oletus leidis kinnitust, uurides 2024. sügissemestri andmeid, kus autor võrdles teooriatestide punkte kursuse lõpptulemustega.

Kursusel on ka suur hulk erinevat tüüpi programmeerimisülesandeid, mis kajastuvad eraldi väljadena ka hinderaamatus.

Pärast süvitsi andmete uurimist ning katsetusi tehes otsustas autor esimese katsetuse käigus kasutada edasisel analüüsil vaid kokkuvõtvaid tulemusi iga nädala kohta. See omab endas nii teooriatestide, programmeerimisülesannete kui ka muude asjade eest saadud punkte. Nii on kursuse käigus hea analüüsida, kuidas tudeng edeneb.

Tabel 1. Esimese nädala hindeelemendid.

Kategooria	Hindeelement	Maks. punktid
Teooriatestid (Moodle)	Test 1.1 – arv	3
	Test 1.2 – sõne	3
	Test 1.3 – muutuja	3
	Test 1.4 – print()	3
	Test 1.5 – input()	3
Charoni ülesanded	EX00 - hello.py	0
	EX01 – hello.py	3
	EX01 – poem.py	3
	EX01 – operators_task.py	3
	EX01 – maths.py	3
	EX01 – atm.py	3
	EX01 – Style	1
	EX01 – Defense	1
Kokkuvõtvad punktid	EX00 - Hello kokku	0
	EX01 – Beginning kokku	15
	1. nädal kokku	30

Lisaks nädalate hinnetele ja punktidele leidub antud kursuse hinderaamatus ka väga palju välja eksamite kohta. Aines korraldatakse üldiselt rohkem kui viis eksamit, kus iga eksami iga ülesanne kajastub iseseisvalt hinderaamatus. Kuna eksamitulemus on kursuse raames väga suur osa lõpptulemusest (antud kursuse puhul 40% osakaaluga), tähendab see seda, et kursuse lõpliku punktisumma ennustamisel mängib see asjaolu suurt rolli ja võib ennustuse kvaliteeti ja täpsust mõjutada. Hinderaamatus on olemas ka väli, mis esindab

tudengi lõpptulemust ilma eksamipunktideta. Just see väli sai uuringu käigus valitud ennustatavaks elemendiks, kuna niimoodi saab vähendada eksamitulemustest tingitud ennustuse kallutatust.

Tulemused, mis saavutati, kui kasutati ainult kokkuvõtvaid andmeid nädalate kohta, näitasid erinevate aastate kursuste andmete katsetamisel paljulubavaid märke. Probleem sellise lahendusviisi juures peitub selles, et hinnete saamisel ei arvestata ajaga, millal tudeng tulemuse on saanud. See tähendab seda, et kui tudeng on varajaste nädalate ülesanded alles kursuse lõpus teinud, siis mudeli jaoks tähendab see sama, kui see, et ülesanne oleks õigel nädalal juba tehtud. Selline viga on lubamatu, kuna niimoodi pole võimalik õiget ettekujutust saada sellest, kuidas algoritm päriselt kursuse jooksul toimiks.

Edasisel katsetamisel ning uurimisel selgus, et Moodle'i admebaasist on võimalik iga hinde puhul saada kätte ka ajahetk, millal hinne saadud on. See andis suurepärase võimaluse teha algoritm just nii, et iga nädala kohta arvestatakse vaid nende tudengi hinnetega, mis selleks hetkeks saadud olid.

Kui esimese variandina katsetas autor ennustamist, mille käigus arvestati vaid iga nädala kogupunktisummaga ning ennustatavaks väärtuseks oli kursuse punktide arv ilma eksami tulemuseta, siis analüüsimise käigus osutus selline lähenemisviis väga spetsifiliseks ja sobis vaid ühe konkreetse kursuse andmetele. Lähtudes soovist arendada rakendust nii, et seda oleks tulevikus soovi korral võimalikult lihtne ka teistele kursustele laiendada, otsustas autor teistmoodi probleemile läheneda.

Selleks, et antud algoritm sobituks ka teistele kursustele, oli esimese asjana vaja hinded normaliseerida. See tähendas seda, et iga tudengi hinne teisendati edasiseks analüüsiks vahemikku 0-1, kus 0 tähistas madalaimat ja 1 kõrgeimat võimalikku tulemust antud ülesande puhul. Selline teisendus võimaldas hinnata tudengi sooritust sõltumata erinevate kursuste ja ülesannete hindamisskaalast.

Selleks, et olemasolevatest hinnetest veelgi rohkem väärtust kätte saada, otsustas autor kasutada tunnuste väljatöötamise tehnikat (ingl. k. *feature engineering*), mis võimaldab lisaks põhiaandmetele (hinnetele) tuletada uusi tunnuseid. Need aitavad õigete tunnuste valimise puhul mudeli täpsust suurel hulgal parandada [16].

Tabel 2 näitab hinnete ennustamise jaoks valitud lisatunnuseid, mis olemasoleva andmestiku kaudu tuletati.

Tabel 2. Kasutatud tunnused ja nende kirjeldused.

Tunnus	Tüüp	Kirjeldus
total_points	Arvuline	Tudengi seni teenitud punktide summa.
num_submissions	Arvuline	Sooritatud hindamisüksuste arv.
mean_normalized	Arvuline	Hinnete keskmine normaliseeritud väärtus (skaalal 0–1).
std_points	Arvuline	Hinnete standardhälve.
last_submission_days_ago	Arvuline	Päevade arv viimasest tulemusest.
current_progress	Suhtarv	Hetkel teenitud punktid jagatud maksimaalse võimalikuga (ennustuse hetkeks).
completion_rate	Suhtarv	Tehtud ülesannete arv jagatud mõõdnud nädalate arvuga (ennustuse hetkeks).
consistency	Arvuline	Keskmine tehtud ülesannete arv nädalas.
weighted_points	Arvuline	Punktide summa, kus hilisematele sooritustele on antud suurem kaal.
weekly_trend	Arvuline	Näitab hinnete muutumise suunda ajas ehk kui kiiresti ja mis suunas tudengi tulemused nädalate jooksul muutuvad (lineaarne trend).

4.3 Keelemudeli kasutamine

Väga oluline osa projekti käigus oli keelemudeli kasutamine. Eesmärk oli leida keelemudel, kes saaks aru eesti keelest ning mõistaks tudengite kirjutatud lauseid nii, et selle järgi mõistlikuid ennustusi teha.

Valitud keelemudel pidi suutma analüüsida tudengite tagasiside vastuseid, mis sageli sisaldasid lühikesi, kõnekeelseid ja kirjavigadega kirjutatud lauseid. Lisaks oli oluline terve tagasiside kontekstist aru saada - näiteks, kui tudeng on andnud märku, et ülesannetele kulus üle 20 tunni (mis on ühe nädala kohta keskmisest oluliselt rohkem), siis peaks keelemudel aru saama, et tudengil võis raske olla ning sellest järeldusi tegema.

Lisaks pidi keelemudel suutma kokkuvõtet genereerida, et õppejõud saaks kiiresti ülevaate suurest hulgast tagasisidest ilma iga vastust käsitsi läbi lugemata. See funktsionaalsus oli mõeldud eelkõige õppejõu töö lihtsustamiseks ja aja säästmiseks, võimaldades tal kiiresti tuvastada korduvad teemad, mured või positiivsed aspektid tudengite tagasisides. Mudel pidi eristama olulisi märksõnu ja mustreid ning esitama need lühidalt ja arusaadavalt, säilitades samal ajal tagasiside sisulise mõtte. Eesmärk oli pakkuda automaatselt loodud kokkuvõtteid, mis oleksid piisavalt usaldusväärsed, et neid saaks kasutada tulevikus kursuse kohandamiseks või probleemide varajaseks märkamiseks.

Samuti testiti, kui hästi mudel suudab üldistada erineva stiiliga ja keelekasutusega vastuseid, säilitades olulise info ka siis, kui tagasiside oli napisõnaline.

4.3.1 LLaMA3

Üks esimesi katseid oli LLaMA3 mudeli käivitamine lokaalselt arvutis, kasutades 8b versiooni, mis viitab mudeli suurusele, sisaldades 8 miljardit parameetrit. Kuigi mudel suutis genereerida mõningaid tulemusi, ei olnud need piisavalt täpsed ega kvaliteetsed. Lisaks ei saanud LLaMA3 korralikult aru eesti keelest, kuna mudel pole piisavalt hästi selle jaoks treenitud.

Näide tudengi tagasisidest: "Üllataval kombel ei tekkinudki seekord täielikku segadust." - tagasiside viitab, et tavaliselt tudengil tekib ülesandeid tehes suur segadus, mis ei ole kindlasti kõige positiivsem. Llama3 võttis tagasisidet aga sellisena: "*In a surprising way, there was no complete chaos this time either.*", mis eesti keelde tõlgituna tähendab "Üllataval kombel ei tekkinud ka seekord täielikku kaost". Kahe lause võrdlemisel võib aru saada, et mõte on täiesti erinev, mis põhjustaks edasistes ennustustes ebakorrektsed tulemusi.

Probleemi lahendamiseks kasutas autor MarianMTModel'it, mis võimaldab tõlkida teksti eesti keelest inglise keelde enne, kui vastavad andmed edasiseks analüüsiks sisestati. See lähenemine parandas oluliselt LLaMA3-mudeli arusaamist, kuna vastav keelemudel on treenitud eelkõige ingliskeelse sisuga. Siiski ei tulnud mudel hästi toime tagasisidega, kus tudengid ei väljenda otsesõnu, kas mingi osa kursusest oli raske või lihtne, vaid teevad seda kaudselt.

Tabel 3 toob välja näited tudengite tagasisidest ning LLaMA3 analüüsitulemused. Viip (ingl.

k *prompt*), mida kasutati seletamaks keelemudelile ülesannet, oli järgmine: Änalüüsi ühe lausega järgnevat tudengi tagasisidet ja ennusta tõenäosus skaalal 1-10, et tudeng läbib edukalt kursuse."

Tabel 3. Näited LLaMA3 analüüsist tudengite tagasisidede kohta.

Tudengi tagasiside	Ennustus (1–10)	Kokkuvõte keelemudeli poolt
Tegin kõik ülesanded ära, aga see ei tähenda, et need arusaadavad oleksid olnud - pigem oli tunne, nagu lahendaks ristsõna võõras keeles. Õpetaja abi polnud küll tihti vaja, aga peamiselt sellepärast, et ma ei osanud küsida midagi.	9	Tudeng on edukalt lahendanud ära kõik ülesanded, kusjuures õpetaja abi pole vaja läinud. See viitab, et tudeng on iseseisev ja pädev.
Üllataval kombel ei tekkinudki seekord täielikku segadust.	8	Tudeng mainib, et teda üllatas kaose puudumine, mis viitab sellele, et tema ootused olid madalad või ta oli valmis ootamatusteks.

Tulemused, mis katsetamise käigus tehti (nähtaval tabelis 3) ei olnud rahuldavad. Esimese tagasiside puhul ennustas mudel väga kõrget kursuseläbimise tõenäosust (9) ja seda selle pärast, et tudeng kõik ülesanded ära on lahendanud ilma abi küsimata. Kui tegelikult tagasisidest üritada aru saada, võib järeldada, et tudeng pole tegelikult paljudest tehtud ülesannetest aru saanud. Seda tõestab ka lause selle kohta, et õpetajalt pole osatud midagi küsida. Autori arvates, kasutades inimaju lause analüüsimiseks, peaks ennustus kindlasti väiksem olema kui 9.

Teise tagasiside puhul ennustas keelemudel jällegi üsna kõrget tõenäosust tudengile. Tudengi tagasisidet inimajuga uurides väidab autor, et ennustus peaks jällegi madalam olema, kuna jääb mulje, et tudengil on eelnevate ülesannetega suur segadus olnud ning seekord oli see segadus lihtsalt väiksem.

Suur probleem antud mudeliga oli ka see, et mudel ei osanud genereerida vastust sellisel

kujul, nagu nõutud oli. Mudeli vastusest pidi olema võimalik välja võtta number (ennustuse tõenäosus) ning vajadusel midagi muud. See osutus väga raskeks, kuna vastuste genereerimisel ei võetud arvesse viibas seletatud struktuuri.

4.3.2 GPT-4o mini

Järgmine mudel, mida projektis katsetati, oli OpenAI mudel GPT4o-mini. Selle kasutamiseks loodi ühendus OpenAI ametliku API kaudu, mis tähendas, et teenuse eest tuli tasuda. Mudel valiti aga just seetõttu, et see on üks soodsamaid valikuid OpenAI valikus, pakkudes samal ajal väga head kvaliteeti ja suurt tuge erinevatele keeltele, sealhulgas ka eesti keelele, mis projekti skoobi all oleval kursusel põhikeeleks on. Mudeli kasutamine andis hea ülevaate tudengite emotsioonidest ning hoiaku muutustest erinevate küsimuste põhjal. Kui LLaMA3 piirdus tihtipeale spetsiifiliste osadega lausest, siis GPT-4o mini suutis paremini aru saada tervest lausest ja selle kontekstist.

Antud OpenAI mudel suutis lisaks eesti keelest aru saamisele ka suurepäraseid vastuseid anda just eesti keeles. See on terve protsessi juures positiivne, kuna sellisel juhul ei ole vaja ühtegi tõlkimist teha, kus lause mõte võib järsult kaotsi minna.

Tabelis 4 on toodud välja mõned näited GPT-4o mini analüüsist tudengite tagasiside kohta. Samuti tuuakse välja mudeli ennustused kursuse edukalt läbimise tõenäosuse osas.

Tabel 4. Näited gpt4-o mini analüüsist tudengite tagasiside kohta.

Tudengi tagasiside	Ennustus (1–10)	Kokkuvõte keelemudeli poolt
Tegin kõik ülesanded ära, aga see ei tähenda, et need arusaadavad oleksid olnud - pigem oli tunne, nagu lahendaks ristsõna võõras keeles. Õpetaja abi polnud küll tihti vaja, aga peamiselt sellepärast, et ma ei osanud küsida midagi.	4	Tudengi tagasiside viitab, et ta on järginud ülesandeid, kuid arusaamatused takistavad sügavat mõistmist, mis võib põhjustada edasisi raskusi.
Üllataval kombel ei tekkinudki seekord täielikku segadust.	7	Tudengi tagasiside viitab sellele, et ta on suutnud kursuse sisu mõista paremini kui varasematel kordadel, mis on positiivne märk edusammudest.

Tulemused GPT-4o mini mudeli katsetamisel olid oluliselt paremad võrreldes LLaMA3 mudeliga. Esimeses tagasisides mõistis keelemudel, et tudeng, olenemata sellest, et ülesanded on tehtud, ei oma sügavat arusaama ning seetõttu võib tekkida edasisi raskusi. Ennustuseks on 4, mis võiks viidata, et (abi)õppejõud peaks tudengiga tegelema. Vaadates tudengi tagasisidet, oleks see ka mõistlik, kuna tudeng ise ei oska tihtipeale ligineda õppejõule küsimustega.

Ka teise tagasiside puhul mõistis GPT-4o mini paremini tudengit, ennustades tõenäosuseks 7. Kuigi ennustuse number ei erine LLaMA3 mudeli ennustusest palju, siis mudeli kokkuvõttest saab lugeda, et GPT-4o mini on aru saanud tudengi raskustest varasematel nädalatel ning jäeldab, et nüüd on tudeng tegemas edusamme.

Kui LLaMA3 puhul oli suur probleem vastuste struktuuri eiramine, mis raskendas mudeli tulemuste hindamist, siis GPT-4o mini suutis väga korrektselt genereerida vastuseid. Vastused struktureeriti täpselt sellisel kujul, nagu viibas palutud oli ning see teeb edasise töötlemise palju lihtsamaks.

Kokkuvõttes on GPT-4o mini mudel selgelt täpsem ja usaldusväärsem lahendus kui LLaMA3. Arusaam tervest kontekstist on ülioluline tudengite edukuse ennustamisel ning seega otsustas autor edasises projektis just GPT-4o mini mudeli kasuks.

4.3.3 Viiba koostamine

Selleks, et keelemudelilt saada võimalikult kvaliteetne ja täpne ennustus, on vaja selget juhust ehk viipa (ingl. k. *prompt*). Hästi koostatud viip aitab keelemudelil mõista ülesande sisu ja konteksti ja suurendab mudeli vastuse täpsust.

Hea viip koosneb tavaliselt järgmistest osadest: juhust, mis annab mudelile selge ülevaate ülesandest, mida on vaja lahendada; kontekst, mis aitab mudelil mõista ülesande tausta; sisendandmed ehk konkreetset andmed (antud juhul tudengi tagasiside), mida soovid, et mudel analüüsiks; ning väljundi täpsustav juhend, mis näitab mudelile seda, mis kujul vastust tahetakse saada [17].

Et koostada võimalikult edukas viip tudengi tagasiside analüüsimiseks ja selle abil ennustuste tegemiseks, jälgiti eelnevalt välja toodud printsiipe.

- **Kontekst.** Selleks, et keelemudel juhusest täpselt aru saaks, tuuakse viibas esimesena välja kontekst: *“Ma esitan sulle tudengi tagasiside. Tagasiside on antud programmeerimise kursuse nädala kohta ja võib hõlmata endas küsimusi tempo, raskuse ja mõtete kohta.”*. Kuna viibas esitatakse ka ennustus hinnete põhjal, mis varem tehakse, siis antakse selle kohta teada lausega: *“Ma võin lisaks tagasisidele anda ka masinõppemudeli tehtud ennustuse tudengi lõppskoori kohta. Ennustus on ujukomaarv vahemikus 0 kuni 1, mis tähistab punktide kogusumma protsentuaalset osa (nt 0.87 tähendab 87%).”*.
- **Juhust.** Peale konteksti selgituse on kirjas üldine juhust: *“Sinu ülesanne on ennustada, kui tõenäoliselt tudeng kursuse läbib, skaalal 1 kuni 10 (kus 10 = kõrgeim tõenäosus).”*. Juhust ütleb konkreetset ära peamise ülesande, mida mudel tegema peab ja mis skaalal hinnangut anda. Seejärel kirjeldatakse täpsemaid juhuseid ehk kuidas tõenäosust määrata. Näiteks: *“Kui tudengi emotsioonid on negatiivsed ja ta mainib suuri raskusi, määra madalam tõenäosus (1–4).”*. Samuti on keelemudel ülesandeks võtmesõnade genereerimine, mis annaksid ülevaate tudengi seisust.
- **Väljundi täpsustus.** Väga oluline osa viibast on väljundi täpsustus ehk kirjeldus, mis

formaadis peaks mudeli vastus olema. See aitab hiljem vastusest vajaliku info kerge vaevaga kätte saada. Antud juhul on see välja toodud nii: “*Vastuse vorming (järgige seda rangelt): Probability: {{number}}, Keywords: {{fraas1}}, {{fraas2}}, {{fraas3}}, ...*”. See tagab struktureeritud väljundi, mis on analüüsi ja hilisema töötlemise jaoks sobiv. Samuti on oluline keelemudelile tuua näiteid selle kohta, milline väljund võiks olla. Selleks, et saada soovitud formaadis võtmesõnu, on viibas lause: “*Kasuta selgeid ja lühikesi fraase (nt "raskused rekursiooniga", "enesekindel koodi silumisel", "madal motivatsioon").*”

- **Sisendandmed.** Viip eeldab, et mudelile antakse tudengi tagasiside ja võimalusel ka hinnete põhjal tehtud ennustus, mida mudel peab analüüsima. Tagasiside antakse kujul, kus on välja toodud iga küsimuse tekst ja tudengi vastus küsimusele. Sellele lisatakse iga eelneva nädala kokkuvõtte võtmesõnadest, mis keelemudel genereerinud on.

4.3.4 Mudelite maksumus

Soovides seda, et rakendus ka tulevikus kasutust leiaks, on oluline arvestada mudelite kasutamise kuludega.

LlaMA3 mudeli puhul, mis töötas lokaalselt, oli üheks suurimaks eeliseks see, et teenuse eest polnud vaja maksta. See võimaldas rakenduse pidevat kasutamist ilma täiendavate kuludeta.

GPT-4o mini puhul tuleb umbkaudu hinnata, kui palju ning kui suuri tagasisidesid kursuse jooksul analüüsitakse. 2025. aasta mai seisuga on mudeli sisendsõnede (ingl. k *input tokens*) hind 0,15 USA dollarit miljoni sõne kohta ning väljundsõnede (ingl. k *output tokens*) hind 0,60 USA dollarit miljoni sõne kohta.

2024. aasta sügissemestril jäi tudengite arv aines "Programmeerimise algkursus" 400-500 vahele. On oluline ka mainida, et kursuse möödudes jääb iga nädalaga tagasiside arv väiksemaks, aga praeguseks arvutuseks oletatakse, et semestris on nädalaid kokku 16 ja seega tagasisidesid $16 \times 500 = 8000$. Erinevatel nädalatel on rakenduse tehtud päring keelemudelile erineva suurusega, kuna hilisematel nädalatel lisatakse viiba sisse ka varasemate nädalate analüüsimise tulemused. Seega, ümardades üles, oletatakse hetkel, et keskmiselt on ühe sisendi sõnede arv 1000 ja väljundi sõnede arv 70. See tähendab, et

terve kursuse jooksul õpilaste analüüsimiseks antud keelemudeliga on sisendsõnede hind maksimaalselt $8000 \times 1000 \times 0.15 / 1000000 = 1.20$ (USA dollarit) ja väljundsõnede hind $8000 \times 70 \times 0.60 / 1000000 = 0.34$ (USA dollarit) ehk kokku on hind 1.20 USA dollarit + 0.34 USA dollarit = 1.54 USA dollarit.

4.4 Hinde ennustamine

Lisaks keelemudeli kasutamisele katsetati antud projektis ka teisi masinõppe meetodeid ning seda tehti eelkõige hinnete ennustamisel. Nagu on kirjeldatud peatükis 4.2.3, kasutas autor esimeses ennustusmodelis iga nädala kogupunktisummat, kus üritatati ennustada kursuse lõpptulemust ilma eksami eest saadud punktideta. Mudelit arendati paremaks nii, et see võtab sisendina kõik hinded normaliseeritud kujul, kuhu juurde lisatakse erinevad tunnused. Kui alguses arendatud mudel ennustas täpset punktisummat, mis ei olnud hea lahendus, kuna punktisummad võivad erineda kursustel, siis lõplik mudel ennustab lõpptulemust vahemikus 0-1. Minimaalne väärtus 0 tähendab kõige kehvemat võimalikku tulemust ja 1 maksimaalset võimalikku tulemust.

Antud töö raames katsetati mitmeid erinevaid juhendatud (ingl. k. *supervised*) masinõppe mudeleid, mille eesmärk oli ennustada tudengi kursuse lõpptulemust lähtuvalt erinevate ülesannete punktidest. Kasutatud mudelid olid:

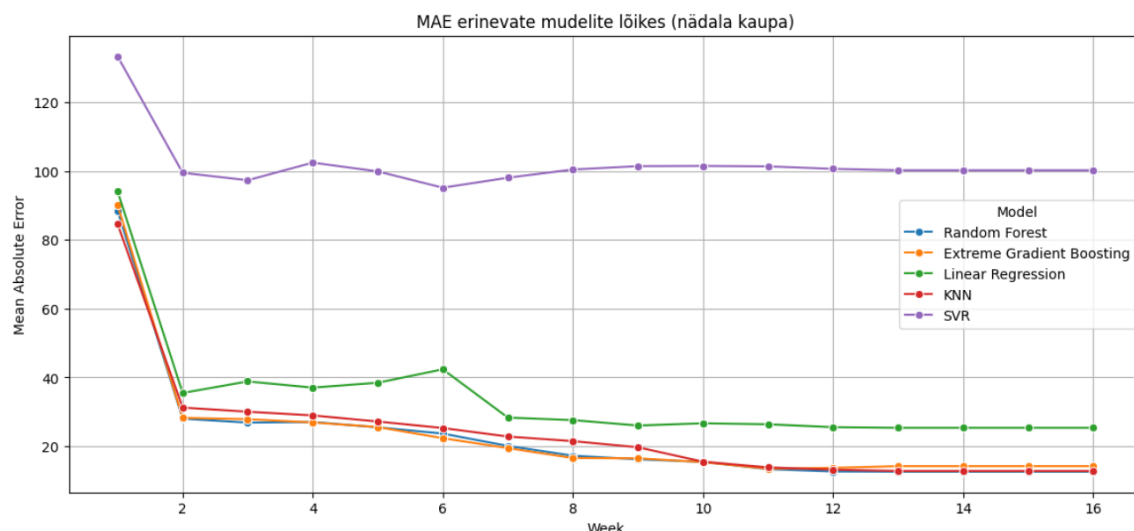
- **Lineaarne regression (ingl. k. *Linear Regression*)** - lihtne ja hästi tõlgendatav baasvõrdlus, mis aitab hinnata lineaarsete seoste olemasolu.
- **Juhuslik mets (ingl. k. *Random Forest Regression*)** - ansamblipõhine meetod, mis suudab paremini aru saada mittelineaarsetest seostest ja aitab vältida üleõppimist.
- **Ekstreemne gradiendiga võimendamine (ingl. k. *Extreme Gradient Boosting Regression*)** - iteratiivne mudel, sobib hästi väiksema müra ja keerukama struktuuriga andmestike puhul.
- **K-lähimate naabrite algoritm (ingl. k. *K-Nearest Neighbors*)** - mudel, mis tugineb sarnaste näidete leidmisele.
- **Tugivektorklassifitseerija (ingl. k. *Support Vector Regression (SVR)*)** - sobib hästi väiksematele andmestikele ning suudab arvestada keerukamate seostega, kui kasutada sobivat tuumfunktsiooni.

Kõik mudelid treeniti ja testiti samadel andmetel, milleks oli aine "Programmeerimise algkursus" 2024. aasta sügissemestri hindaamat. Sisenditena kasutati esimestel nädalatel saadud punkte erinevatest ülesannetest (nt 1.–5. nädalani saadud punktid), millele lisati nendest arvutatud tunnused.

Tulemuste hindamisel kasutati peamiselt kahte mõõdikut:

- **R^2** (determinatsioonikordaja) – mõõdab, kui hästi mudel saab hakkama andmete varieeruvusega.
- **MAE** (keskmise absoluutne viga) – annab aimu, kui suur on mudeli ennustuste keskmine kõrvalekalle tegelikest väärtustest.

Mudeleid võrreldi iga nädala lõikes, et hinnata, mis nädalast alates hakkab mudel andma piisavalt täpseid ennustusi. Lisaks võimaldas see hinnata, kui vara kursuse jooksul saab usaldusväärselt prognoosida tudengi edukust. Mudelite võrdlus ja visualiseerimine andis hea ülevaate, millised lähenemised sobivad taolise ennustamise jaoks kõige paremini. Joonisel 4 on näha graafikut, mis võrdleb mudeleid keskmise absoluutse vea järgi. Samuti on näha, kuidas mudelite viga nädalate suurenedes muutub.



Joonis 4. Keskmise absoluutne viga (MAE) erinevate mudelite puhul.

Pärast mudelite võrdlust selgus, et suurima täpsusega selle konkreetse ülesande puhul on juhusliku metsa ja XGBoost algoritmid. Pea iga mudeli tulemustest võis ka näha, et 1. nädalal on väga keeruline täpseid ennustusi lõpptulemuse kohta teha. Põhjuseid selleks on kindlasti mitu: näiteks see, et ülesanded on väga väikesed ja lihtsad, kui ka see, et paljud

tudengid alles katsetavad, kas kursust edasi teha.

4.5 Analüüsiandmete koondamine

Kuna tudengite analüüsimiseks tehakse keelemudelile palju päringuid, mis maksavad ning võivad võtta ka palju aega, otsustati analüüsiandmed koondada ühte tabelisse andmebaasis. Nii on neid ka hiljem vajadusel lihtne ja mugav edasiseks tegevuseks välja võtta.

Iga keelemudeli vastus, mis genereeritakse analüüsisides tudengi tagasisidet, teisendatakse rakenduses üheks andmebaasi kirjeks, mis sisaldab lisaks analüüsi tulemustele viidet tudengi ja kursuse tabelile. Samuti on lisatud nädala number. Selline andmestruktuur võimaldab hõlpsasti pärida analüüsi konkreetse nädala, tudengi või kursuse lõikes. Andmebaasitabeli struktuur on kujutatud tabelis 5.

Tabel 5. Andmebaasitabeli `analysis_result` struktuur.

Veerg	Andmetüüp	Kirjeldus
<code>analysis_id</code>	INTEGER	Unikaalne analüüsisimise ID
<code>student_id</code>	INTEGER	Tudengi ID
<code>course_id</code>	INTEGER	Kursuse ID
<code>week</code>	INTEGER	Nädala number
<code>prediction</code>	INTEGER	Ennustus (1-10)
<code>keywords</code>	TEXT	Võtmesõnad tudengi edukusest

Samuti on andmebaasitabelis nende andmete hoidmine vajalik selleks, et hilisemate nädalalate analüüsimises saaks kasutada varasemate nädalate analüüside tulemusi. Nii mõistab keelemudel, mis seis on tudengil senini kursusel olnud ning oskab ennustust tehes arvesse võtta trendi ehk seda, kas tudengi edukus on kursuse jooksul suurenemas või vähenemas.

4.6 Andmete uuendamine

Andmete uuendamiseks lõi autor teenuse, mis on realiseeritud **FastAPI** raamistikus. Teenuse eesmärk on pakkuda API lõpp-punkte (ingl. *endpoints*), mille kaudu on võimalik rakenduse andmebaasis olevaid andmeid uuendada või täiendada.

Lähtudes rakenduse põhifunktsionaalsusest, on tähtsaim roll analüüside ja ennustuste genereerimisel. Selleks loodi lõpp-punkt `/api/analyse-feedback`, mis võtab sisendiks päringuparameetrid (ingl. *query parameters*), nagu kursuse identifikaator ja nädala number (semestris). Päringu tulemusena kogub rakendus kokku vastava kursuse ja nädala tagasiside ning edastab selle keelemudelile, et saada analüüsi ja ennustusi. Analüüsitulemused salvestatakse seejärel andmebaasi.

Tabel 6 kujutab loodud API lõpp-punkti struktuuri.

Tabel 6. API lõpp-punkt tagasiside analüüsi käivitamiseks.

Väli	Kirjeldus
URL	<code>/api/analyse-feedback</code>
Meetod	GET
Päringuparameetrid	<code>course_id</code> - kursuse identifikaator <code>week</code> - nädala number
Kirjeldus	Käivitab tagasiside analüüside ja ennustuste genereerimise

4.7 Visualiseerimine

Selleks, et analüüsitulemused oleksid kasutatavad ning võimalikult arusaadaval kujul, on vajalik tulemusi kuidagi visualiseerida.

Antud projektis sai visualiseerimistööriistaks valitud Apache Superset, kuna see võimaldab väga mugavalt ühenduda andmebaasiga, kust seejärel andmeid erineval viisil pärida ning eriilmelisi graafikuid koostada.

Apache Superset võimaldab luua koondpaneele (ingl. k. *dashboard*), mille abil saab õppejõud kiiresti hinnata tudengite üldist edenemist. Samuti saab rakendada tervele juhtpaneelile filtri, milleks antud juhul võiks olla näiteks kursus ja nädal. Filtri rakendamisel muutuvad kõik paneelil olevad graafikud, milles filter sees on. See teeb kasutajale soovitud andmete vaatamise mugavaks.

5 Tulemused

Käesolevas peatükis tuuakse välja põhilised projekti tulemused. Esmalt vaadeldakse tudengite hinnete ennustustäpsust valitud masinõppemudeliga, seejärel hinnatakse keelemudeli põhjal tehtud analüüside kvaliteeti ning lõpuks rakenduse toimivust ja kasutatavust.

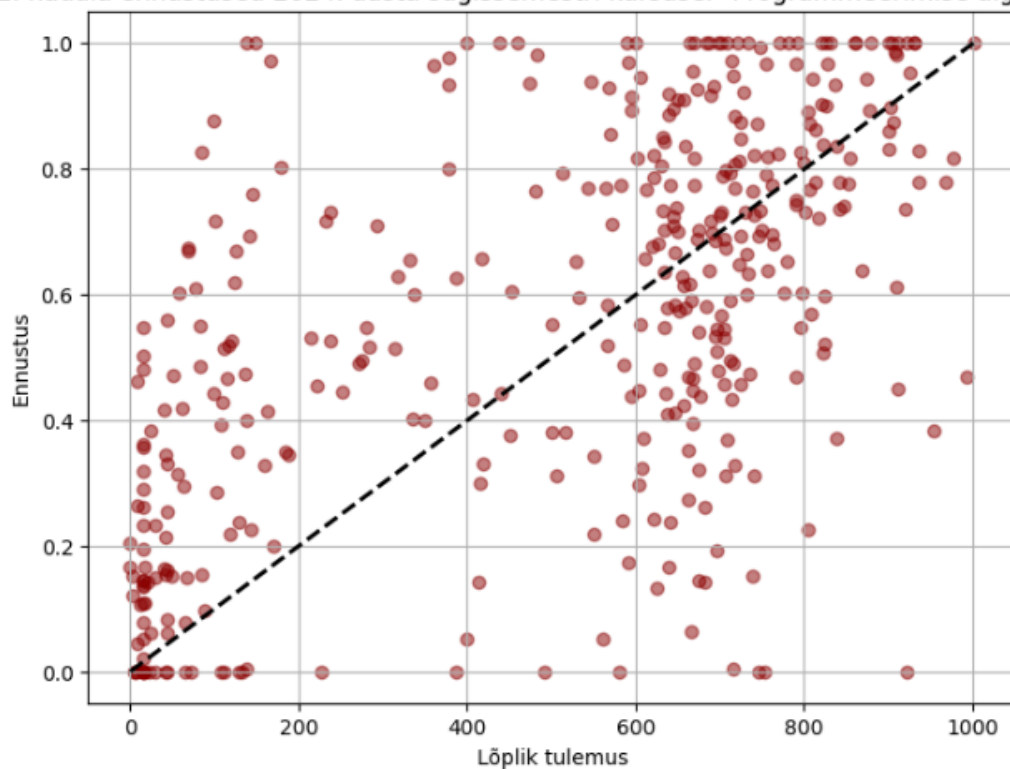
5.1 Hinde ennustus

Nagu on kirjeldatud peatükis 4.4, siis erinevate masinõppe mudelite võrdluse tagajärjel valiti lõplikuks hinnete ennustamiseks XGBoost algoritm. Antud mudeliga katsetati ennustamist 2024. aasta "Programmeerimise algkursuse" andmete peal ning seda tehti iga nädala kohta. Nii oli võimalik hea vaatepilt saada sellest, kuidas algoritm alguses väheste andmetega ja hiljem rohkematega hakkama saab.

Kursusel oli võimalik kokku saada umbes 1000 punkti. See koosnes põhiliselt kohustuslikest ja vabatahtlikest programmeerimisülesannetest, teooriatestidest ja 400-punktilisest eksamist.

Joonis 5 näitab hinde ennustamise tulemusi 1. nädalal. Joonis 6 näitab tulemusi aga 5. õppenädalal. Testid tehti kõigi 2024. aasta "Programmeerimise algkursuse" tudengite andmete peal.

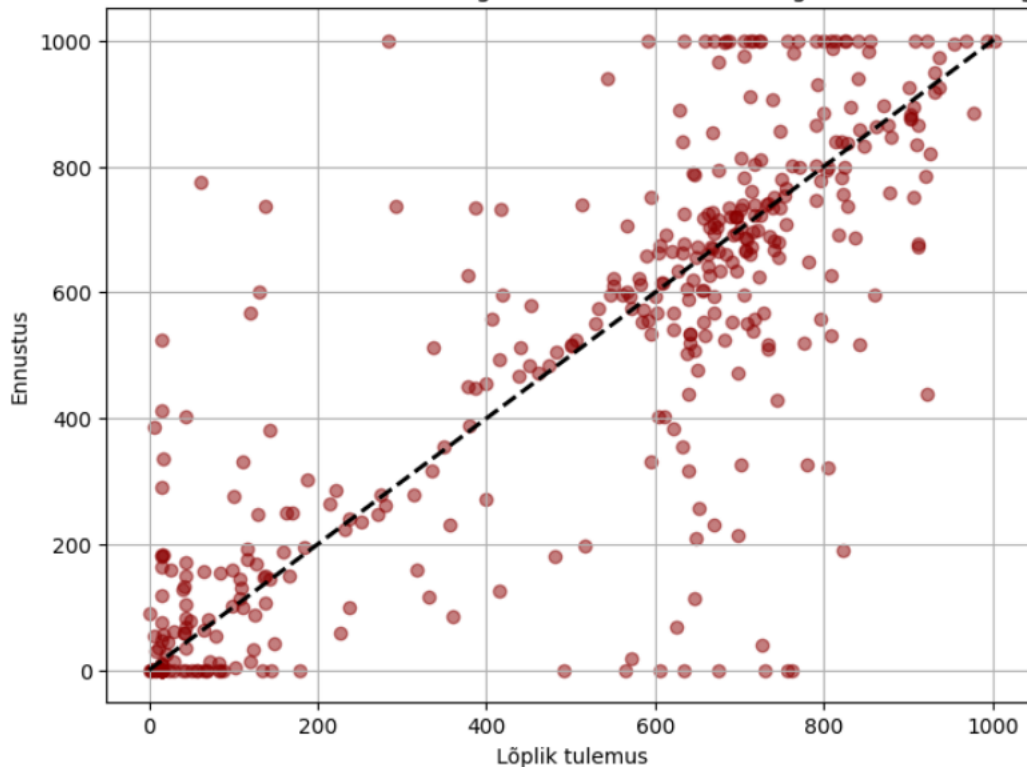
1. nädala ennustused 2024. aasta sügissemestri kursusel "Programmeerimise algkursus"



Joonis 5. Hinde ennustamine 1. nädalal.

1. nädala tulemustest on näha, et mudelil on keeruline ennustusi teha olnud. Kuna ülesanded olid väga väikesed ning andsid vähe punkte, siis tegid väga paljud tudengid maksimaalse soorituse, mis tähendas kõrget ennustust paljude inimeste puhul. Keskmine absoluutne viga 1. nädala ennustustulemustega oli 210.52.

5. nädala ennustused 2024. aasta sügissemestri kursusel "Programmeerimise algkursus"



Joonis 6. Hinde ennustamine 5. nädalal.

5. nädalaks kogunenud andmetega saab mudel juba väga hästi hakkama, kuna punktierinevused erinevate tudengite vahel olid selgemad. Selle tulemusel paranesid ennustuste täpsusnäitajad märgatavalt – keskmine absoluutne viga vähenes 125.71 peale.

Vaadates põhjalikumalt ennustusi, mille puhul algoritm suurelt eksis, selgusid kaks peamist põhjust. Esiteks oli kursusel palju tudengeid, kellel raskused punktide järgi ilmusid alles semestri teises pooles. See tähendab, et hilisemate nädalate ülesannete eest saadi vähe punkte või ei saadud üldse. Esimeste nädalate ülesanded olid väiksema keerukusastmega ja seetõttu saadi palju punkte, mistõttu mudel ei suutnud näha märke sellest, et hiljem tudengil raskem läheb.

Teiseks leidis tudengeid, kes said suure osa punktidest lisaülesannetest. Lisaülesandeid sai teha terve kursuse vältel ning paljudel tudengitel jäi nende tegemine viimastesse nädalatesse. Tavaülesannete punktisummade põhjal polnud selliste tudengite puhul aru saada, et lõpptulemus võiks kõrge olla, mistõttu tekkisid madalamad ennustused.

Kokkuvõtvalt võib siiski järeldada, et puhtalt varajaste hinnete kaudu on võimalik väga palju

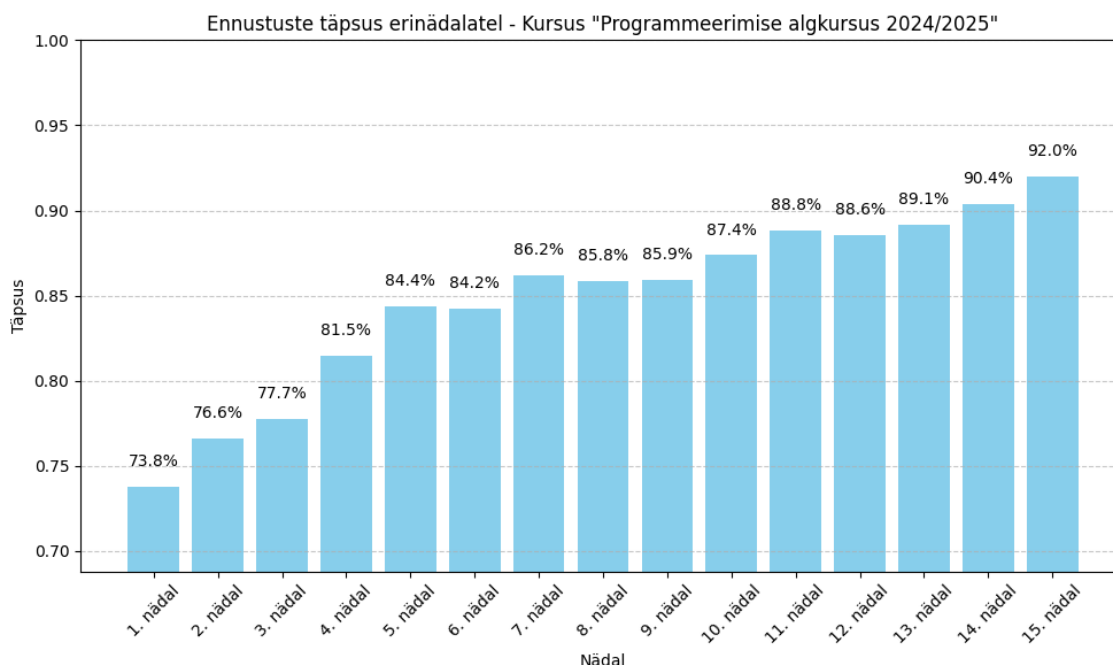
tudengi edasise edukuse kohta õelda. Kes teevad alguses kõik ülesanded maksimaalsetele punktidele, saavad ka hiljem kõrgemad tulemused.

5.2 Keelemudeli ennustus

Selles peatükis on vaatluse all keelemudel ja selle ennustused. Keelemudelile antakse iga nädala kohta ette tudengi tagasiside, eelnevate nädalate tagasiside kokkuvõte (mis on samuti keelemudeli enda poolt tehtud) ning sellele kõigele liidetakse veel hinde ennustustulemus, mis sai lahatud peatükis 5.1.

Keelemudeli ülesandeks oli tudengile määrata tõenäosus skaalal 1-10 selle kohta, et vastav tudeng läbib kursuse edukalt (Lisa 3). Keelemudeli tagastatavatest numbritest tekkis aga küsimus, et milline number on selleks piiriks, mis eraldab omavahel kursuse läbikukkujad ning edukad läbijad. Kõige suurem täpsus oli mudelil siis, kui piiriks valida 3. See tähendab, et kui ennustus on 3 või alla selle, on suur tõenäosus, et tudeng ei läbi kursust.

Joonis 7 näitab keelemudeli ennustuste täpsust kursusel "Programmeerimise algkursus 2024/2025", kui valida piiriks 3.

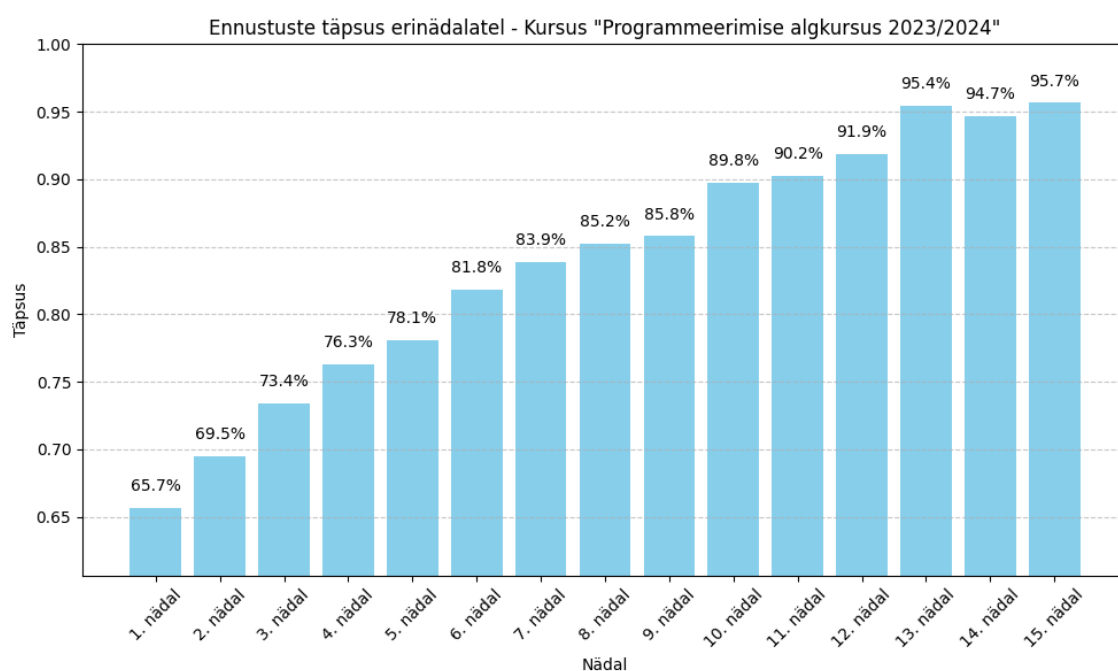


Joonis 7. Keelemudeli ennustuste täpsus - Programmeerimise algkursus 2024/2025.

Ennustuste tulemusest on näha, et esimeste nädalatega kasvab täpsus üha suuremaks. Juba 1.

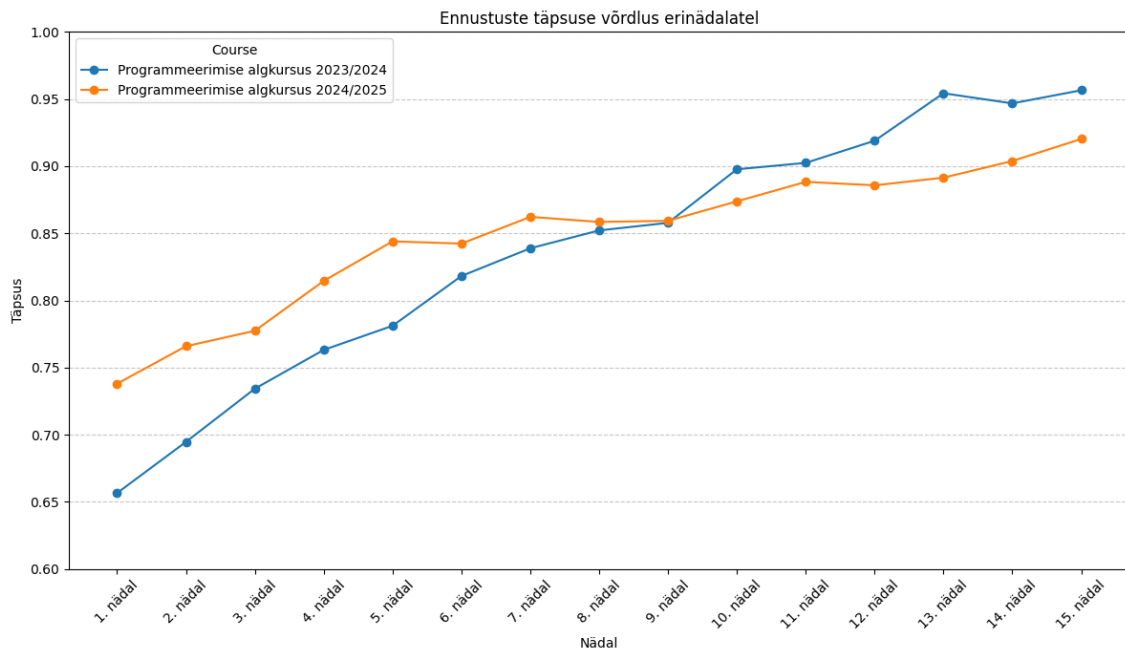
nädalal ennustas keelemudel 73.8% täpsusega tudengite kursuse läbimist. 5. nädalaks oli see juba 84.4% ja täpsus tõusis kursuse jooksul aina suuremaks. See tõestab, et keelemudel loeb hästi tagasisidest välja punktid, mis viitavad tudengi edasisele edukusele. Tagasisidesid käsitsi uurides leidis autor, et üldiselt kirjutavad tudengid kursuse teises pooles oma probleemidest ja muredest rohkem, mis kursuse keerukust ja tempot arvestades on igati loogiline. Kuna keelemudelile antakse ette ka eelnevate nädalate tagasisided, siis saab see palju parema üldpildi tudengi edukusest.

Et veenduda keelemudeli täpsuses, testis autor sellega ka 2023. aasta sügissemestri kursuse "Programmeerimise algkursus" andmete peal algoritmi. Tulemused on näha joonisel 8.



Joonis 8. Keelemudeli ennustuste täpsus - Programmeerimise algkursus 2023/2024.

2023. aasta sügissemestri kursuse ennustuste tulemusi vaadates saab veelgi kinnitust, et keelemudel on täpne. Täpsuse kasv nädalate lõikes on suur, kusjuures 1. nädala täpsus on juba 65.7%. 6. nädalal on täpsus 81.8%. Joonis 9 näitab kahe testitud kursuse ennustuste täpsuse võrdlust.



Joonis 9. Keelemudeli ennustustäpsuse võrdlus kahel kursusel.

Selleks, et aru saada, miks mõnede tudengite puhul ennustus valeks osutus, uuris autor põhjalikumalt konkreetseid ennustusi mudeli poolt. Ühe konkreetse ebakorrekse ennustuse näitena saab välja tuua tudengi, kelle kohta mudel ennustas 5. nädalal kursuse läbimiseks tõenäosuse 3 ehk väga madala arvu. Tegelikult oli antud tudengi lõppskoor 780 punkti (78% maksimaalsest tulemustest), mis tähendab, et kursus läbiti väga kindlalt. Mõned tagasiside väljad erinevatel nädalatel on antud tudengi kohta toodud välja tabelis 7. Uurides tagasisidesid, mis tudeng esimestel nädalatel andis, selgub, et kuni 4. nädalani olid vastused väga positiivsed ning mingisuguseid märke selle kohta, et midagi halvasti oleks, ei olnud. Samuti jäid keelemudeli poolt ennustatud väärtused kõrgeks. Suure muutuse tõi sisse 5. nädal, kus tudengil tekkisid raskused ühe konkreetse ülesandega. Tagasiside vastustest tuli välja järgnev:

"Jäin kohutavalt hätta remove_unnecessary_chars ülesandega, ilmselt kulutasin 10-15 h selle peale, ja selle tõttu ei jõudnud korralikult üle vaadata kas teised funktsioonid korras. Ausalt oli kogu kursuse osas allaandmise plaan."

Kuna tudengi tagasisides oli väga selgelt kirjas, et tal oli tekkinud soov kursusest loobuda, võttis keelemudel seda tugevalt arvesse. Kuigi tegelikult võis tegu olla lühiajalise motivatsiooni langusega, mis oli seotud ühe konkreetse keeruka ülesandega, ei suutnud keelemudel

sedas arvesse võtta. Mudel tõlgendas sellist lauset kui tugevat viidet väljalangemisele ega suutnud mõista hetkelist frustratsiooni.

Nädal	Tundide arv	Enesetunne	Tempo	Keelemudeli ennustus
1	2	Väga positiivne	Paras	8
2	6	Väga positiivne	Paras	8
3	6	Pigem positiivne	Paras	7
4	10	Neutraalne	Liiga kiire – liiga rasked ülesanded	6
5	20	Neutraalne	Liiga kiire – liiga suur ülesannete hulk	3

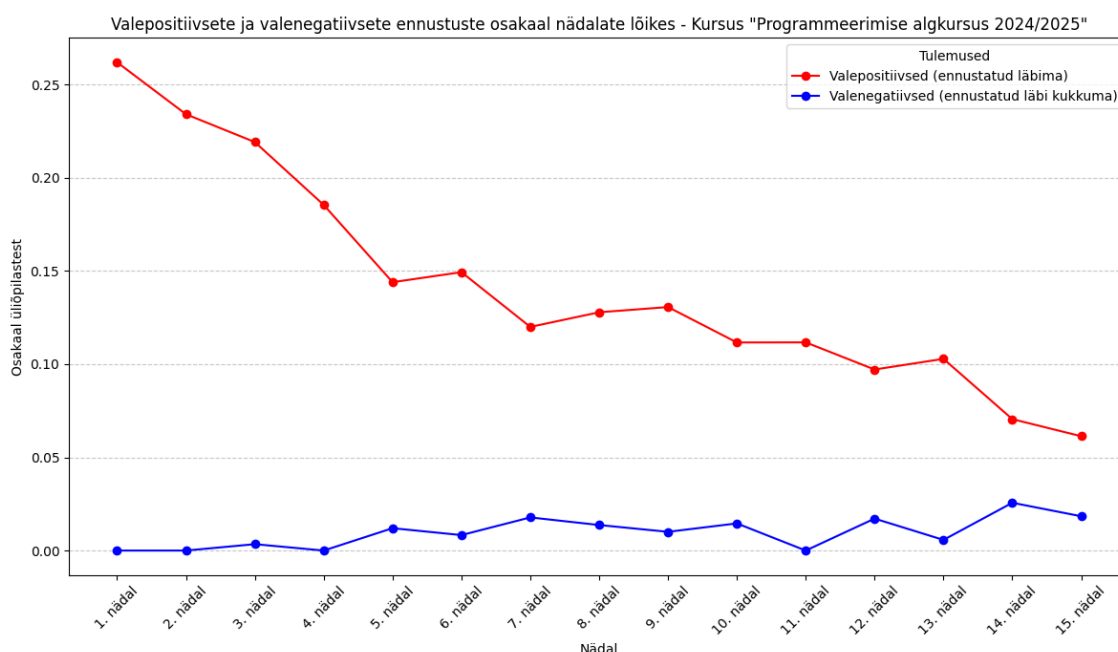
Tabel 7. Muutus nädalate kaupa konkreetse tudengi näitel.

Just selliste näidete puhul võib öelda, et keelemudelil puudub piisav võimekus mõista üldist konteksti ja eelkõige inimeste tundeid ja emotsioone. Antud tudeng jätkas kursusel osalemist, saades häid tulemusi ning lõpetas aine edukalt, kuigi mudel oli ennustanud vastupidist.

Sellel põhjal ei tee halba asjaolu, et keelemudel just madala ennustuse andis, sest see võimaldab varakult tähelepanu pöörata tudengitele, kelle puhul mingigi märk võimaliku väljalangemise kohta esineb. Liigne abi pole kindlasti halb, aga suurem probleem võib tekkida juhul, kui mudel ennustab raskustes oleva tudengi läbimise tõenäosust suureks ning selle tagajärjel tudeng tähelepanuta jääb. Ka sellist tüüpi viga esines keelemudeli ennustustes. Ühele tudengile ennustati läbisaamise tõenäosuseks 5. nädalal 8, aga lõpptulemuseks kujunes tudengil vaid 120 punkti (500 on vaja kursuse lõpetamiseks). Antud tudengi põhjalikumast analüüsist selgus, et tagasisides olevad kohustuslikud väljad, mis on valikvastustega, on täidetud positiivsete vastustega, aga avatud tekstiväljadega küsimused, mis ei ole kohustuslikud, on jäänud vastamata. Selline juhtum võib viidata asjaolule, et keelemudel peaks rohkem välja tooma tudengeid, kes küsimustele mingil põhjusel ei vasta. Sellisel juhul on aga raske mõista, kas vastamata jäänud küsimused tähendavad, et tudengil on kõik hästi ning probleeme ei ole, või hoopiski vastupidist.

Kuigi ennustuste täpsust võib pidada heaks, siis parema ülevaate saab juhul, kui vaadata

valepositiivseid ja valenegatiivseid ennustusi. Valepositiivne ennustus tähendab siinkohal juhtu, kus mudel on ennustanud, et tudeng läbib kursuse, kuigi tegelikkuses nii ei olnud. Just neid juhtusid on antud projekti eesmärki vaadates tähtis vältida, kuna nii võivad abivajavad tudengid märkamata jääda. Valenegatiivne tulemus tähendab juhtu, kus mudel ennustab tudengi läbikukkumist kursusel, kuigi tegelikult tudeng sooritas kursuse edukalt. Joonis 10 näitab "Programmeerimise algkursus 2024/2025" andmete põhjal mudeli valesti tehtud ennustuste jaotust valepositiivseteks ja valenegatiivseteks juhtudeks. Antud joonisel on kasutatud läbimise ja läbikukkumise klassifitseerimise lävena (ingl. k. *threshold*) arvu 3, kuna see andis varem tehtud testides parima tulemuse.



Joonis 10. Keelemudeli valed ennustused, kus piiriks on 3.

Valepositiivsete ja -negatiivsete ennustuste hulka analüüsidest selgub, et valepositiivsete arv on kordades suurem. Valenegatiivsete ennustuste arv on pea olematu, aga selle põhjus peitneb ilmselt madalalt määratud lävest, mida tudengite klassifitseerimisel kasutatud sai (3 või väiksem arv tähendab läbikukkumist). Suurem läve korral väheneb küll valenegatiivsete ennustuste arv, aga tõuseb suurel hulgal ka valepositiivsete arv, mis viib selleni, et programm toob kokku välja palju rohkem abivajavaid tudengeid. Nii on suurem tõenäosus, et raskustes olevad tudengid on selles hulgas, kuigi neid võib olla raske leida tudengite seast, kes samuti ennustuse järgi hulka sattusid, aga tegelikult nii suurt abi ei vaja.

Nädalate möödudes valepositiivsete ennustuste hulk kahaneb ja 7. nädalal on see umbes

12%. Kahjuks tähendab see seda, et on olemas mingisugune hulk tudengeid, kelle kohta mudel ei oska kindlalt väita läbikukkumist. Nagu varem mainitud, peitub põhjus tihti tühjades tagasisideväljades, kui ka tudengi eraelulistest põhjustest, mis ei pruugi kuskil kursusega seotud andmetes kajastuda. Näiteks, mõned tudengid täitsid kohustuslikud valikvastustega väljad positiivselt, kuid jätsid avatud tekstiväljad vastamata. See võib viidata madalale motivatsioonile või soovile kiiresti küsimustik ära täita ning seda ei ole keelemudel suutnud piisavalt hästi tuvastada.

Valenegatiivsete ennustuste analüüs näitas, et keelemudel tõlgendab ajutisi raskusi, näiteks tudengi probleeme ühe konkreetse ülesandega, liiga tõsiselt, ennustades kursuselt läbikukkumist, kuigi tudeng tegelikkuses aine edukalt on lõpetanud. Andmestiku uurimine tõi välja, et kuigi enamik tudengeid suudab ajutisest raskusest üle saada, esineb üksikuid juhtumeid, kus tudeng kukub kursuse konkreetse ülesande pärast läbi. Seega on madala ennustuse andmine õigustatud, kuna see võimaldab varakult tuvastada potentsiaalseid riske, isegi kui see vähendab mudeli üldist täpsust.

5.3 Ennustuste visualiseerimine

Selleks, et kõik tulemused nähtavad oleks, kasutatakse projektis Apache Superset tööriista, mis võimaldab andmebaasidega ühendada ning seejärel graafikuid teha, mida omakorda koondpaneelidele (ingl. k. *dashboard*) panna saab. Koondpaneelil on võimalik kõikidele seal olevatele graafikutele ühine filter panna ning antud projektis on see hea võimalus tulemusi kas kursuse või nädala kaupa filtreerida. Joonis 11 kujutab näidiskoondpaneeli, kus peal on graafik, mis toob välja abivajavad tudengid. Graafik on filtreeritud konkreetse kursusega ja nädalaga ning eespool on väiksema ennustuse saanud tudengid.

Filters

ADD/EDIT FILTERS

Nädal

3

Kursus

IT10102 Programmeerimise...

APPLY FILTERS

Programmeerimisained

Draft

Superset Admin

a day ago

EDIT DASHBOARD

Abivajavad tudengid

prediction	week	student_name	email	course	keywords
2	3		@taltech.ee	IT10102 Programmeerimise algkursus (2023/2024)	liiga kiire aine tempo, väga negatiivne enesetunne, 100 tundi probleemide lahendamiseks, abiõppejõudude ebasobiv käitumine
2	3		@taltech.ee	IT10102 Programmeerimise algkursus (2023/2024)	liiga kiire tempo, pigem negatiivne, midagi ei sujunud, 20 tundi probleemilahendust, ei sujunud midagi
3	3		@taltech.ee	IT10102 Programmeerimise algkursus (2023/2024)	liiga kiire tempo, väga raske % kasutamine, Caesari ülesanne keeruline, neutraalne enesetunne, abivalmis abiõppejõud
3	3		@taltech.ee	IT10102 Programmeerimise algkursus (2023/2024)	liiga kiire tempo, õppisin väga palju, pigem negatiivne, 14 tundi probleemide lahendamiseks, sain vastused küsimustele
3	3		@taltech.ee	IT10102 Programmeerimise algkursus (2023/2024)	negatiivne enesetunne, liiga kiire tempo, palju abi, videoid võiks olla rohkem
3	3		@taltech.ee	IT10102 Programmeerimise algkursus (2023/2024)	liiga kiire tempo, negatiivne enesetunne, aeganõudev aine, abivalmid õppejõud, vähe õppimist
3	3		@taltech.ee	IT10102 Programmeerimise algkursus (2023/2024)	raskustaseme hüppe, koodivihi aitas, palju õppimist
3	3		@taltech.ee	IT10102 Programmeerimise algkursus (2023/2024)	liiga kiire tempo, ajaga on kitsas, positiivne enesetunne
3	3		@hotmail.com	IT10102 Programmeerimise algkursus (2023/2024)	liiga kiire tempo, hakan vaikselt maha jääma, negatiivne enesetunne, 20 tundi probleemide lahendamiseks, teadmiste puudumine ülesannetes
3	3		@taltech.ee	IT10102 Programmeerimise algkursus (2023/2024)	raskused loopidega, liiga kiire aine tempo, negatiivne enesetunne, abi vihvideost, kulus 48 tundi

Joonis 11. Näidiskoondpaneel.

Samuti on võimalik koondpaneelile lisada erinevaid graafikuid tagasiside kokkuvõtetest, mis annavad võimaluse analüüsida konkreetsete küsimuste tulemusi (Lisa 3). Tagasisideid on projekti andmebaasis koondatud tabelisse feedback.

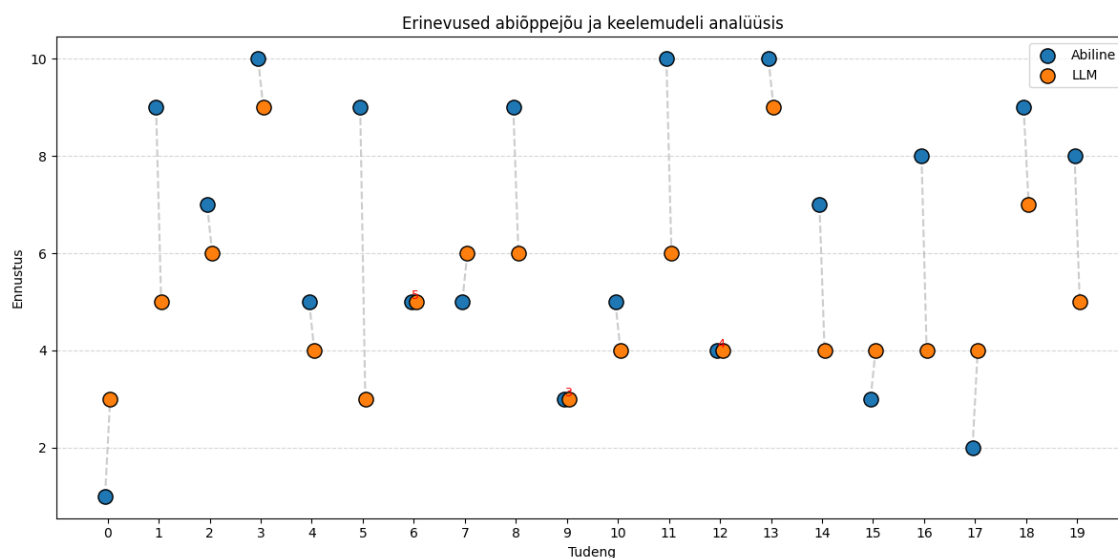
Kuigi antud projekti eesmärgiks on täpsete ennustuste tegemine selleks, et abivajavad tudengid välja tuua, ja kasutajaliidese tegemine on teisejärguline, anti ühele abiõppejõule võimalus programmeerimisainete jaoks tehtud koondpaneeli uurida ning seda ise muuta. Tagasisidest selgus, et graafikud olid arusaadavad ning vajaliku informatsiooni kättesaamine oli väga lihtne ja otsekohene. Kui abiõppejõud üritas ise uusi graafikuid teha, et mõne tagasisideküsimuse tulemusi visualiseerida, tekkis esialgu raskusi. Graafikute tegemiseks on vajalik päringukeele SQL oskus ning samuti on vaja mõista erinevaid graafikutüüpe, et vastavalt olukorrale parim valida. Samuti mainis abiõppejõud, et kõigil õppejõududel võiks olemas olla võimalus teha märkmeid näiteks tudengite kohta, kes abi on juba saanud. Seda võiks keelemudel edasistel analüüsimistel arvesse võtta ning seejärel uusi arvestusi teha. Samuti annaksid sellised märkmed parema ülevaate õppejõule, et näha tudengeid, kes veel abi pole saanud.

5.4 Keelemudeli võrdlus inimesega

Selleks, et ülevaade saada, kui erinevalt mõistab inimese aju tudengi tagasiside tekste võrreldes keelemudeliga, viis autor läbi eksperimendi, mille käigus määras skoobi all oleva kursuse abiõppejõud 20 tagasisidele väärtuse vahemikus 1-10. Väärtus paluti iga tagasiside

47

puhul valida selle põhjal, kui tõenäoliselt antud tudeng kursust läbib, võttes arvesse nii vastuste tooni kui ka mainitud raskusi. Samu tagasisidesid analüüsi ka projektis kasutatava keelemudeliga. Erinevusi mudeli ja inimese vahel kujutab joonis 12.



Joonis 12. Erinevused abiõppejõu ja keelemudeli analüüsis.

Kui võrrelda inimest ja keelemudelit, võib näha, et leidub palju tagasisidesid, mille puhul mõlemad osapooled ennustasid kas täpselt sama või väga väikese erinevusega väärtused. Siiski leidub ka tagasisidesid, mille puhul on erinevus suur. Üldiselt võib märgata, et inimene on taoliste tagasiside puhul kõrgema hinnangu andnud, kui keelemudel. Autor uuris abiõppejõult põhjuseid, miks need tagasisided nii kõrge väärtuse said ning selgus, et üldiselt peitus põhjus selles, et tudengil polnud ühtegi mure, ülesannetele kulunud tundide arv oli väike ning aine tempo oli tema jaoks kas paras või aeglane. Uurides keelemudeli pandud väärtuseid, selgus, et mudel võttis arvesse detaile, mis inimesele antud kontekstis tähtis ei paistnud. Näiteks kirjeldas üks tudeng, kuidas tal terve nädal oli raskusi arvuti seadistamisega konkreetse ülesande jaoks ning see tekitas pahameelt. Kuigi ülejäänud osa tagasisidest oli positiivne, võttis keelemudel arvuti seadistusprobleeme tõsiselt arvesse, arvates, et see segab tudengi edasist õppimist ja edenemist. Kokkuvõtlikult võib öelda, et inimene ja keelemudel tõlgendavad tagasisidet tihti erinevalt. Inimene keskendub üldisele muljele, aga keelemudel analüüsib detaile, mis võivad muuta ennustuse tulemust nii halvemaks kui ka paremaks.

5.5 Tulemuste kokkuvõte

Analüüsisides nii hinnete kui tagasisideade ennustamise tulemusi, võib öelda, et tudengite kohta on väga vara võimalik järeldusi teha. Kui esimestel nädalatel on puhtalt hinnete põhjal veel raske suuri järeldusi teha, siis lisades sellele tagasiside ehk tudengite enda mõtted, on võimalik saada oluliselt selgem ülevaade olukorrast.

Alates 4.-5. nädalast muutusid ennustused väga täpseks ning sellest hetkest alates enam suurt täpsuse kasvamist edasiste nädalate puhul ei tekkinud. See tähendab, et antud projekti käigus arendatud rakendusega on võimalik juba siis hädasolevad tudengid üles leida, kui veerand semestrist läbi on, mis jätab palju aega selleks, et tudeng edukalt kursusest siiski läbi aidata.

Töö kinnitas, et mitmesuguste andmete kombineerimine võimaldab keelemudelil tuvastada mustreid, mida ühest allikast oleks keeruline tuvastada. Näiteks olukorrad, kus tudengid on saanud häid hindeid, kuid tagasisidest ilmneb motivatsiooni- või arusaamise puudumine, viitasid võimalikele probleemidele, mis hilisemas kursuse faasis võisid tekkida. Sellised märgid võimaldavad varajast sekkumist.

6 Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli arendada süsteem, mis võimaldaks (abi)õppejõududel mugavalt analüüsida tudengite edenemist Tallinna Tehnikaülikooli programmeerimisaines "Programmeerimise algkursus", tuvastades varakult need tudengid, kes võivad vajada lisatuge. Eesmärgiks oli ka vähendada käsitsi tehtava töö mahtu ning pakkuda andmepõhist tuge otsustamiseks.

Töö raames loodi rakendus, mis koondab andmed iganädalastest tagasisideküsimustikest ja Moodle'i hinderaamatust, analüüsib neid ning esitab tulemused visuaalsel kujul. Avatud tekstiväljade sisu analüüsimiseks kasutati keelemudeleid, mis võimaldavad tuvastada korduvaid mustreid ja hinnata tudengi mõtteid. Hinnete ennustamiseks rakendati erinevaid masinõppemeetodeid, et ennustada tudengi lõpptulemust juba varajases kursuse faasis.

Rakendusel töötati välja erinevad osad – andmete kogumine, nende töötlemine, analüüs ja visualiseerimine – eesmärgiga tagada süsteemi laiendatavus ja kohandatavus ka teiste ainete jaoks. Arenduse tulemusena valmis lahendus, mis võimaldab õppejõul saada kiiresti ülevaade tudengite hetkeseisust ning sellest, kes abi vajavad. Andmete liikumine Moodle'ist on automatiseeritud ehk sellega vähendati õppejõu käsitsi tehtavat tööd. Ennustustulemused näitasid, et rakendus on usaldusväärne ning see oleks suureks abiks lahendamaks väljalangevuse probleemi IT-valdkonnas.

Kasutatud kirjandus

- [1] Hendrik Voll. „Väljalangevuse anatoomia“. *Mente et Manu* 1898 (2024). URL: https://haldus.taltech.ee/sites/default/files/2024-03/MM_2024-1.pdf.
- [2] Aman Paul. *AI-Powered Adaptive Learning: Ushering In A New Era Of Education In 2025*. Kättesaadav: <https://elearningindustry.com/ai-powered-adaptive-learning-ushering-in-a-new-era-of-education-in-2025>. 2025. URL: <https://elearningindustry.com/ai-powered-adaptive-learning-ushering-in-a-new-era-of-education-in-2025>.
- [3] Gökhan Akçapınar, Arif Altun ja Petek Aşkar. „Using learning analytics to develop early-warning system for at-risk students“. *International Journal of Educational Technology in Higher Education* 16.1 (2019), lk. 40. URL: <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-019-0172-z>.
- [4] Tiffany Wu ja Christina Weiland. *Leveraging Modern Machine Learning to Improve Early Warning Systems and Reduce Chronic Absenteeism in Early Childhood*. EdWorkingPaper 24-1081. Annenberg Institute at Brown University, 2024. URL: <https://edworkingpapers.com/sites/default/files/ai24-1081.pdf>.
- [5] Viktorija Mištšenko. „Väljalangemise ennustamise tööriista arendamine programmeerimise küsimustike põhjal“. TalTech digikogu. Magistritöö. Tallinna Tehnikaülikool, 2024. URL: <https://digikogu.taltech.ee/en/Item/7196f40a-c792-4189-8a6a-2f8829f1b65a>.
- [6] Ütt Hendrik. „Integrating sentiment analysis and machine learning to predict students' academic performances in an introductory programming course“. TalTech digikogu. Magistritöö. Tallinna Tehnikaülikool, 2023. URL: <https://digikogu.taltech.ee/et/item/dcda4b21-268c-431c-a7c9-ae5d063d617d>.
- [7] Index.dev. *12 Most In-Demand Programming Languages to Learn in 2025*. Kättesaadav: <https://www.index.dev/blog/most-in-demand-programming-languages>. 2023. URL: <https://www.index.dev/blog/most-in-demand-programming-languages>.
- [8] Matthew Urwin. *Python Machine Learning Tutorial: Learn ML With Python*. <https://builtin.com/machine-learning/python-machine-learning>. 2023. URL: <https://builtin.com/machine-learning/python-machine-learning> (vaadatud 19.05.2025).
- [9] NVIDIA Corporation. *What is Pandas in Python?* 2024. URL: <https://www.nvidia.com/en-eu/glossary/pandas-python/> (vaadatud 19.05.2025).
- [10] NumPy Developers. *What is NumPy?* 2024. URL: <https://numpy.org/doc/stable/user/whatisnumpy.html> (vaadatud 19.05.2025).

- [11] scikit-learn developers. *User Guide — scikit-learn*. 2024. URL: https://scikit-learn.org/stable/user_guide.html (vaadatud 24.05.2025).
- [12] Mostafa M. Amin *et al.* „A Wide Evaluation of ChatGPT on Affective Computing Tasks“. *arXiv preprint arXiv:2308.13911* (2023). URL: <https://arxiv.org/abs/2308.13911>.
- [13] DB-Engines. *Popularity Ranking of Database Management Systems*. Mai 2025. URL: <https://db-engines.com/en/ranking> (vaadatud 19.05.2025).
- [14] Apache Software Foundation. *Introduction — Apache Superset Documentation*. 2024. URL: <https://superset.apache.org/docs/intro> (vaadatud 24.05.2025).
- [15] IBM Corporation. *What is Docker?* 2023. URL: <https://www.ibm.com/think/topics/docker> (vaadatud 19.05.2025).
- [16] DataCamp. *Feature Engineering in Machine Learning: A Practical Guide*. 2024. URL: <https://www.datacamp.com/tutorial/feature-engineering> (vaadatud 19.05.2025).
- [17] Pranab Sahoo *et al.* „A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications“. *arXiv preprint arXiv:2402.07927* (2025). URL: <https://arxiv.org/abs/2402.07927>.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Mihkel Keso

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Rakenduse loomine tudengite edenemise jälgimiseks masinõppe meetodite abil”, mille juhendaja on Ago Luberg
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

04.06.2025

¹Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Keelemudelile etteantav ülesande kirjeldus

Ma esitan sulle tudengi tagasiside. Tagasiside on antud programmeerimise kursuse nädala kohta ja võib hõlmata endas küsimusi tempo, raskuse ja mõtete kohta.

Sinu ülesanne on ennustada, kui tõenäoliselt tudeng kursuse läbib, skaalal 1 kuni 10 (kus **10 = kõrgeim tõenäosus**).

Ma võin lisaks tagasisidele anda ka masinõppemudeli tehtud ennustuse tudengi lõppskoori kohta. Ennustus on ujukomaarv vahemikus 0 kuni 1, mis tähistab punktide kogusumma protsentuaalset osa (nt 0.87 tähendab 87%).

****Sinu ülesanded:****

1. ****Ennusta läbimise tõenäosust**** (skaalal 1-10):

- Kui tudengi emotsioonid on ****negatiivsed**** ja ta mainib suuri raskusi, määra madalam tõenäosus (1-4).
- Kui emotsioonid on ****neutraalsed**** määra keskmine tõenäosus (5-6).
- Kui emotsioonid on ****positiivsed**** ja tudeng tunneb end enesekindlalt, määra kõrge tõenäosus (7-10).
- Kui tudeng on kulutanud ****rohkem kui 10 tundi probleemide lahendamisele**** kaalu madalamat tõenäosust.
- Arvesta ennustamisel ka hinde ennustust. Kursuse läbimiseks on vajalik saada vähemalt 50% (0.5) kõigist võimalikest punktidest.

2. ****Tuvasta võtmetähelepanekud lühikeste fraasidega:****

- Too eraldi välja ****nii väljakutsed kui ka tugevused****.
- Kasuta selgeid ja lühikesi fraase (nt "raskused rekursiooniga", "enesekindel koodi silumisel", "madal motivatsioon"). Need peaksid andma sisukaid tähelepanekuid tulevaseks tagasiside analüüsiks.
- Maksimaalselt ****3-5 võtmefraasi**** vastuse kohta.

****Vastuse vorming (järgige seda rangelt):****

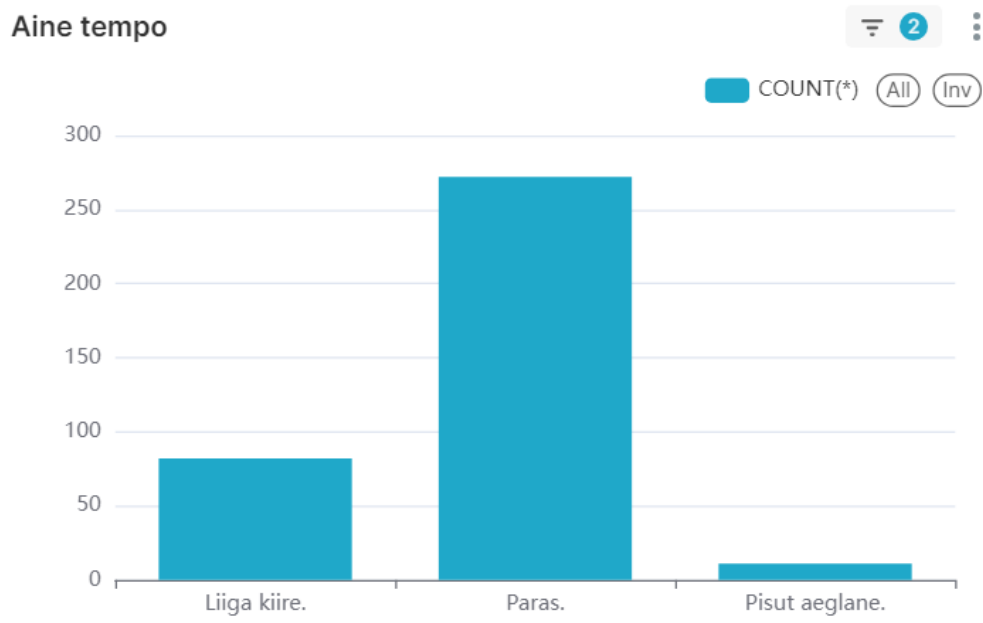
‘ ‘ ‘

Probability: {{number}}

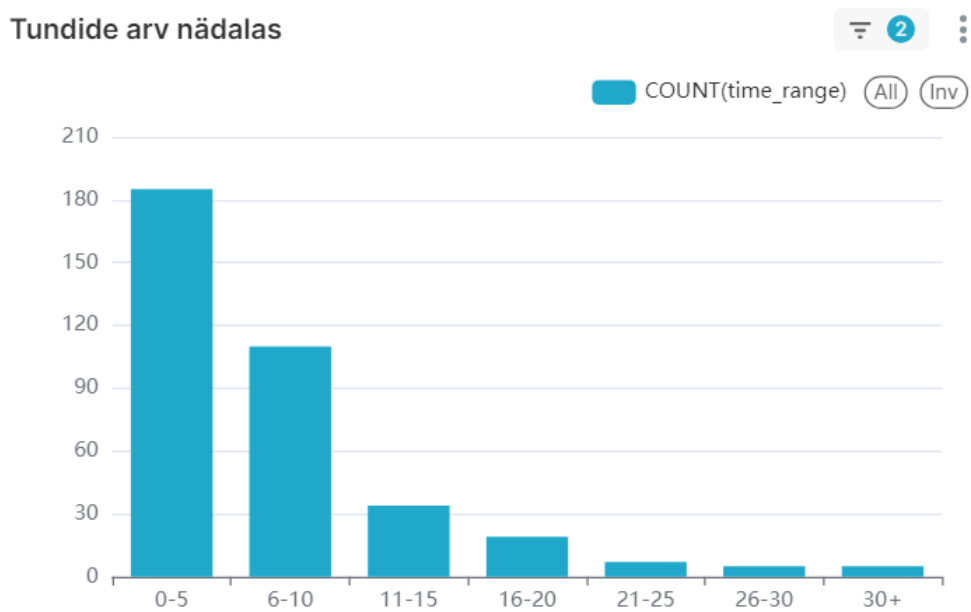
Keywords: {{fraas1}}, {{fraas2}}, {{fraas3}}, ...

‘ ‘ ‘

Lisa 3 – Graafikud Supersetis



Joonis 13. Tudengite vastused küsimusele aine tempo kohta.



Joonis 14. Tudengite vastused küsimusele kulutatud aja kohta (tundides).