TALLINNA TEHNIKAÜLIKOOL
INSENERITEADUSKOND
Virumaa kolledž

IT erialade õpilaste väljalangevuse ennustamine Tallinna Polütehnikumi näitel

Predicting Dropout Rates of IT Students: A Case Study of Tallinn Polytechnic School

ARUKAD SÜSTEEMID JA RAKENDUSINFOTEHNOLOOGIA ÕPPEKAVA
LÕPUTÖÖ

Üliõpilane: Andra Kullerkupp
Üliõpilaskood: 212439EDTR
Juhendaja: Avar Pentel, lektor

Kohtla-Järve, 2025

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

LIHTLITSENTS LÕPUTÖÖ ÜLDSUSELE KÄTTESAADAVAKS TEGEMISEKS JA REPRODUTSEERIMISEKS[1]

Mina, Andra Kullerkupp (25.10.2001)

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „IT erialade õpilaste väljalangevuse ennustamine Tallinna Polütehnikumi näitel", mille juhendaja on Avar Pentel,
   1.1. reprodutseerimiseks säilitamise ja elektroonilise avaldamise eesmärgil, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta kolmandate isikute intellektuaalomandi ega isikuandmete kaitse seadusest ja teistest õigusaktidest tulenevaid õigusi.

---

[1] *Lihtlitsents 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 ühisautori(d) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.*

SISUKORD

JOONISTE LOETELU

EESSÕNA

Rakenduskõrghariduse lõputöö eesmärk on välja töötada ennustusmudelid, mis võimaldavad prognoosida õppivate õpilaste väljalangemise tõenäosust tulevikus.

Rakenduskõrghariduse lõputöös kasutan Tallinna Polütehnikumi poolt väljastatud pseudonümiseeritud andmeid.

Usun, et analüüs ja masinõpe võivad aidata lahendada probleeme ja suurendada mõju haridussektoris.

Soovin tänada Tallinna Polütehnikumi koostöö eest ning enda juhendajat, Avar Pentelit.

LÜHENDITE JA TÄHISTE LOETELU

Python – programmeerimiskeel

JSON – (JavaScript Object Notation) – struktueeritud andmeformaat

WEKA – (*Waikato Environment for Knowledge Analysis*) - masinõppe tarkvarapakett

EHIS – Eesti Hariduse Infosüsteem

ARFF (*Attribute-Relation File Format*)– tekstipõhine failiformaat

TA – Tallinna Polütehnikumi ”Tarkvaraarendaja” eriala lühendnimetus

IT – Tallinna Polütehnikumi ”IT- süsteemide spetsialist” eriala lühendnimetus

Naive Bayes (NB) – tõenäosusmudel

Simple Logistics (SL) – lineaarne klassifitseerija

Random Forest (RF) – otsustuspuude masinõppe algoritm

C4.5 Decesion tree (J48) – otsustuspuu algoritm C4.5 põhjal

Sequential Minimal Optimization (SMO) – optimeerimisalgoritm

Discrete – diskreetsed tunnused

Continuous – arvulised tunnused

String – tekstijada, mis sisaldab tähti, numbreid või muid sümboleid.

Nested – pesastatud andmed

Numeric – numbriline andmetüüp, mis sisaldab arvulisi väärtusi

Dict objekt – andmestruktuur, mis sisaldab võtit ja väärtust

Massiiv – andmete kogum, mis sisaldab mitut elementi järjestatud järjestuses

SISSEJUHATUS

Rakenduskõrghariduse lõputöö „IT erialade õpilaste väljalangevuse ennustamine Tallinna Polütehnikumi näitel" eesmärk on välja töötada ennustusmudelid, mis võimaldavad prognoosida õppivate IT erialade õpilaste väljalangemise tõenäosust tulevikus. Sealhulgas uuritakse, milliste teguritega on väljalangevus seotud. Andmete põhjal selgub, et õpilaste väljalangevus on 49.32%. Andmed on alates aastast 2017, aastatest 2017-2018 on 3- aastased õppekavad, alates 2019 aastast on 4-aastased õppekavad. Õppestaatusega „õpib" on kokku andmestikes 331 õpilast. Katkestanud on 122 õpilast ning lõpetanud on 140 õpilast. Ühtlasi on eesmärgiks on mõista noorte kitsaskohti IT valdkonna õppes Tallinna Polütehnikumis. Töös analüüsin järgmiseid andmeid:

- Sugu
- Vanus
- Nominaalaja lõpp
- Eriala
- EHIS kood
- Õpperühm
- Õppekava kinnitamise kuupäev
- Käskkirja liik
- Käskkirja põhjus
- Elukoht
- Lõpetatud kooli nimi
- Eelnev haridus
- Pseudonüümne ID
- Õppeainete hinded

Andmed on kokku 2 formaadis – JSON formaadis olevad andmed on õpilaste hinded koos õppeainetega ning ülejäänud eelnimetatud 13 liiki andmed saabusid Exceli formaadis. JSON- formaadis olevaid andmeid töödeldakse Pythonis ning masinõppe tarkvaraks on valitud Weka.

Lõputöö esimeses osas antakse ülevaade varasematest uuringutest ning teistest sarnastest töödest ja nende erinevustest ning mittesobivusest. Teises osas antakse ülevaade andmete eeltöötlusest, andmetest ning nende formaadist. Kolmandas osas antakse käsitletakse kasutatavaid tehnoloogiaid ja valideerimisest. Neljandas osas kirjeldatakse tulemusi.

Võtmesõnad: andmed, andmetöötlus, algoritm, masinõpe, rakenduskõrghariduse lõputöö.

1. VARASEMAD UURINGUD

Käesolev peatükk annab lühiülevaate varasemalt tehtud uuringutest, uuringute aktuaalsusest ning ülevaate kasutatud meetoditest.

2015 viidi Taanis läbi uuring, mille eesmärgiks oli uurida ja ennustada, miks õpilased ei omanda soovitud eriala. Selleks analüüsiti suuremahulist andmekogumit, mis sisaldas andmeid õpilaste kohta, kes olid õppinud vähemalt kuus kuud Taani keskkoolis, eesmärgiga ennustada nende väljalangemist järgnevate kolme kuu jooksul. Uuringus osales kokku 36,299 õpilast, kelle andmed olid nii treeningu kui ka testimise jaoks mõeldud masinõppe mudelitele. Erinevaid masinõppe meetodeid katsetades saadi parim tulemus Random Foresti klassifikaatori abil, mis saavutas täpsuse 93,47% ja kõvera alla 0,965. Uuringu tulemused kinnitasid, et masinõpet saab kasutada väljalangemise ennustamiseks haridussektoris, arvestades andmete suurt hulka ja varieeruvust. Saadud tulemused osutuvad üpriski kindlateks ja täpseteks, mis viitab masinõppe potentsiaalile hariduse kontekstis, et aidata prognoosida ja ennetada õpilaste akadeemilisi väljakukkumisi [1].

2019 viidi Nigeerias läbi uuring, mille eesmärgiks oli ennustada väljalangemise põhjuseid, et nende kohta hoiatusi anda ja vähendada õpilaste akadeemilisi väljalangejaid. Uuringu tulemused olid suunatud lõpetajate arvu suurendamisele ja tingimusena oli esitatud vajadus automaatselt töötava mudeli järele. Antud artiklis esitati meetod/protseduur, mis kohandub vastavalt mudelile, et valida parim ennustusalgoritm. Kasutati eksperimentaalseid lähenemisi, et tuvastada parim algoritm adaptatiivsete mudelite jaoks, milleks valiti K-Lähima naabri meetod (K-Nearest Neighbors, KNN). Lisaks katsetati logistilise regressiooni (LR), lineaarse diskriminantanalüüsi (LDA), klassifikatsiooni ja regressioonipuude (CART), Gaussi Naive Bayes (NB) ning toevektorimasinate algoritmi. Andmeanalüüsi jaoks kasutati ajaloolisi andmeid Põhja-Kesk Nigeeria osariigi ülikoolist. Uuringus saadud tulemuste põhjal, mis hõlmasid täpsuse, tundlikkuse ja F1-skoori analüüsi, saadi kõigi atribuutide keskmiseks väärtuseks 0.98, samas kui sisendtunnuste alusel saavutati täpsus 0.90, tundlikkus 0.95 ja F1-skoor 0.92. Need tulemused näitasid, et akadeemiline sooritusvõime on väga oluline põhjus, miks inimesed väljalangemise läbi õppesüsteemist välja langevad. Väljatöötatud mudel on võimeline varakult tuvastama neid õpilasi, kellel on suur kalduvus väljalangemisele, ja annab võimaluse varakult sekkuda, et suurendada õppeasutuste lõpetajate arvu [2].

2021 viidi läbi Tallinna Tehnikaülikooli Virumaa Kolledžis uuring, mille eesmärk on prognoosida esimese kursuse arvutiteaduse tudengite väljalangemist Tallinna Tehnikaülikooli Virumaa Kolledžis ning määrata kindlaks tegurid, mis mõjutavad väljalangemise määra. Uuringus kasutati kahte erinevat andmestikku: (1) andmed TalTechi õppeinfosüsteemist; (2) Virumaa Kolledžis kogutud tudengite ajaloo ja õpitulemuste andmed. Ennustavate mudelite loomiseks rakendati järgmisi masinõppe algoritme: Naive Bayes, otsustuspuud, logistiline regressioon, tugi-vektormasinad ja tehisnärvivõrgud. Uuringu tulemusena hinnati, kuidas muutuvad väljalangemise ennustamise täpsused alates tudengite vastuvõtust kuni esimese semestri lõpuni. Leiti, et enne immatrikuleerimist kättesaadavate andmete põhjal oli võimalik ennustada väljalangemist 70% täpsusega. Esimese semestri jooksul kogutud andmete kasutamine tõstis ennustustäpsuse 90%-ni. Lisaks määratleti tegurid, mis on seotud väljalangemisega, ja need, mis ei ole [3].

2022 viidi Tallinna Tehnikaülikooli Virumaa Kolledžis läbi infotehnoloogia üliõpilaste varajase väljalangevuse prognoosimise eesmärgil masinõppe juhtumiuuring. TalTech Virumaa Kolledžis on aastaid olnud esmakursuslaste seas suur väljalangevus. Infotehnoloogia üliõpilaste seas on see umbes 40%. Väljalangevuse vähendamiseks korraldatakse kõrgkooli vastuvõtuprotsessis alates 2019/20 õppeaastast üliõpilaskandidaatidega vestlusi ja küsitlusi, et hinnata üliõpilaste motivatsiooni ja valmisolekut õppida ning tagada eriala sobivus. Väljalangemist mõjutavate tegurite väljaselgitamiseks analüüsime sisseastumisprotsessi käigus kogutud andmeid ja rakendame tekstikaevetehnikaid, et analüüsida üliõpilaste esseesid, mis on tehtud esimese aasta õppe lõpus kursuse Sissejuhatus erialasse [4].

2020 viidi läbi uuring, mille eesmärk oli leida olulisemad tegurid, mis mõjutavad riigieksamite lõpptulemust positiivselt ja vastupidi. Eesti gümnaasiumiõpilastel on kolm kohustuslikku riigilõpueksamit - matemaatikas, eesti keeles ja võõrkeeles. Eelmiste hinnete ja demograafiliste andmete põhjal õpilase lõpueksamitulemuste prognoosimiseks kasutasime ühest koolist saadud andmeid. Masinõppepaketti Weka kasutati ennustavate mudelite koostamiseks ja kõigi testitud atribuutide komplektidega saime klassifitseerimisel täpsuse üle 80%. Pidevatel mudelitel saime keskmise absoluutvea 10 lähedale või alla selle, kui testitulemuste vahemik oli 0-100 punkti. Enamik meie atribuute olid ainehinded skaalal 1–5 ja seetõttu, kuna me piirasime oma testimisvalimi ainult ühe kooliga, andis see huvitava ülevaate sellest, kuidas mõned ained ja õpetajad annavad oma panuse lõpptesti tulemustesse. Ja üllatuslikult selgus, et mõne testitulemuse puhul oli kõige olulisem ennustaja tulemusega negatiivses korrelatsioonis [5].

2. METOODIKA

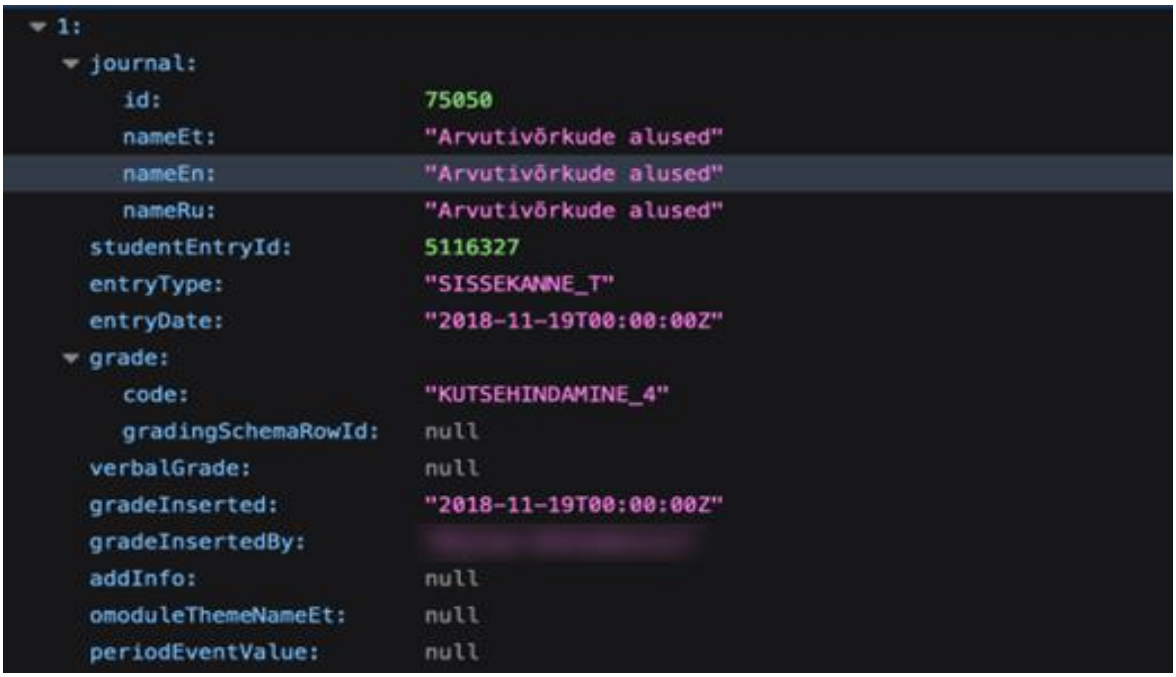
Metoodika peatükis antakse ülevaade saadud andmetest, andmete eeltöötlusest ja andmeanalüüsi protsessist.

2.1 Andmed

Antud töös kasutatud andmed esitati JSON- ja Excel-formaadis. Mõlemad formaadid võimaldavad struktureeritud andmete esitamist ning olid sobilikud edasiseks analüüsiks ja töötlemiseks.

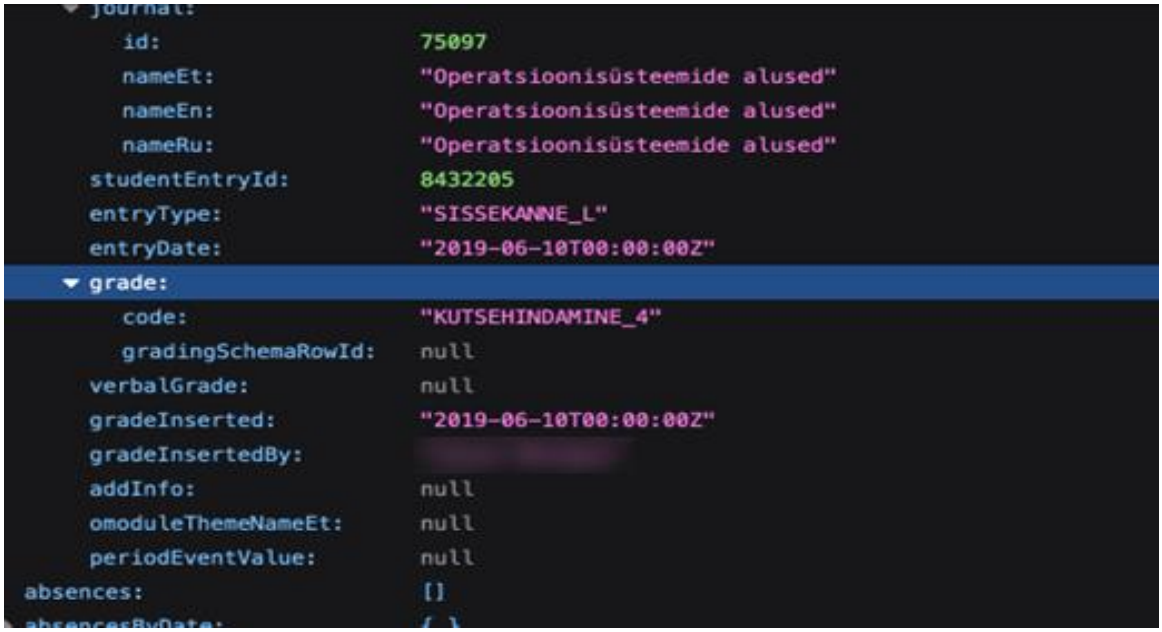
JSON-formaat on pesastatud ja ei ole sobilik mudelite treenimiseks. Seetõttu tuli muuta JSON-formaadis esitatud andmed tasaseks ehk ühe õpilase andmed esitada ühemõõtmelise, mitte pesastatud objektina.

Näide 1 JSON- formaadi andmetes



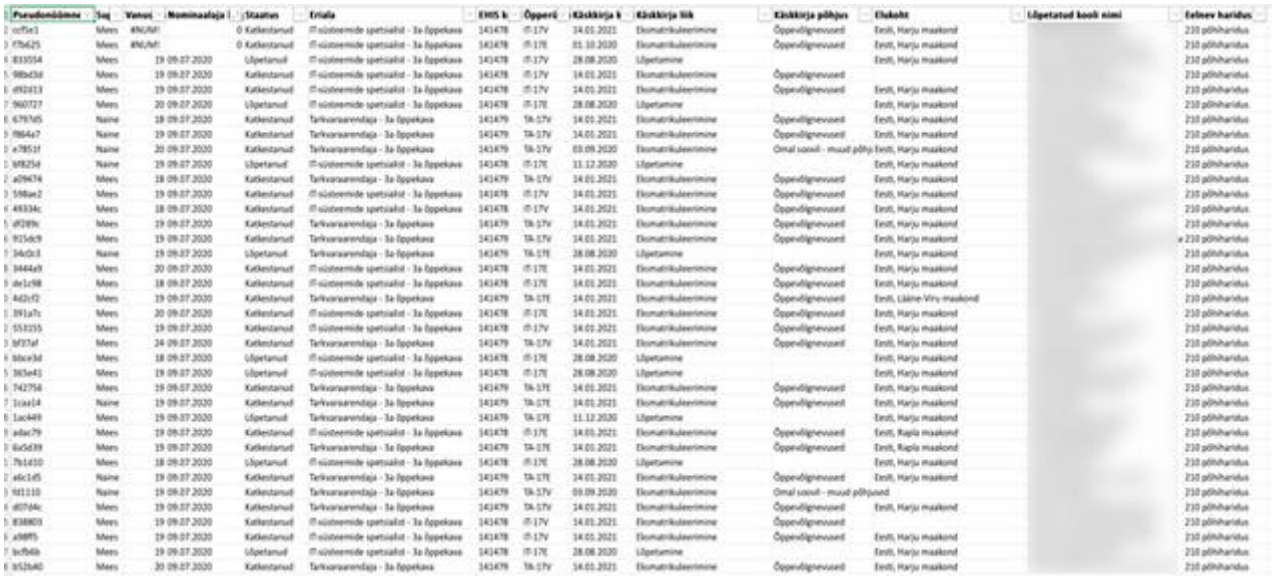
Joonis 2.1 Andmed JSON-formaadis

Näide 2 JSON-formaadi andmetest:



Joonis 2.2 Andmed JSON- formaadis

Näide Exceli formaadi andmetest:



Joonis 2.3 Andmed Excel-i formaadis

2.2 Andmete eeltöötlus

Andmete eeltöötlusel tuli kokku panna andmed erinevatest allikatest - õppeinfosüsteemist ja EHIS-est. Esimeses olid andmed JSON- formaadis failidena, teises Exceli tabelina.

Mõlemas andmestikus olid sama õpilase andmed identifitseeritavad unikaalse ID-ga. Lõppeesmärgiks oli saada kogu andmestik tabeli kujule, kus üks rida oleks ühe õpilase kohta ja veerud õpilast iseloomustavate atribuutide kohta.

Õppeinfosüsteemi JSON-faili formaat oli pesastatud (nested) ja sisaldas eritüübilisi andmeid, massiive, objekte, numbreid ja stringe. Seetõttu sai kirjutatud Pythoni skript, mis pesastatud andmed tasandas.

Samuti selgus õppeinfosüsteemi JSON-failide uurimisel, et ühe õppeaine päevikus saadud hinnete puhul ei ole võimalik luua igast üksikust hindest unikaalset atribuuti, kuna puudub info selle kohta, millise konkreetse ülesande eest on see hinne pandud ja ainus lisainfo on hinde sisestamise daatum. Samuti võis olla erinevatel õpilastel erinev arv hindeid samas õppeaines, mistõttu osutus vajalikuks rakendada andmete agregatsiooni. Nii arvestati iga õppeaine keskmine hinne ja loendati puudulikud ja tegemata tööd ning arvestatud tööd ja neid andmeid kasutati iga õppeaine progressi iseloomustavate atribuutidena.

Lõpuks saadi list, kus iga õpilase kohta oli üks lame (vastandina pesastatud) dict objekt, millest loodi kolm ARFF [6] tüüpi faili - üks, kus on mõlema õppekava õpilaste andmed koos ja teised kaks erinevate õppekavade kohta.

Samuti eemaldati kõik andmed, mis võisid olla otseselt seotud ennustatava väärtusega - õpilase väljalangemise või lõpetamisega. Nii näiteks eemaldati andmetest lõpuks unikaalsed ID-d, käskkirjade kuupäevad, käskkirja põhjendused ja muud õpilase õppestaatusega seotud andmed, mis olid otseses seos ennustava atribuudiga.

Joonis 2.4 eesmärk on võtta tekstifailist sisse loetud ja puhastatud andmed ning ümber korraldada need kasutamiseks struktureeritud sõnastikuna, kus õpilase ID on võtmena ja kõik andmed tema kohta on väärtustena.
Muud_andmed funktsioon loeb sisse Exceli faili, kus on õpilaste demograafilised andmed ja õppestaatusega seotud andmed. Fail on salvestatud CSV-tüüpi, eraldatud failina (.txt). Kui faili lugemisel tekib viga, tõstetakse vastav erind.

Kõik väärtused muudetakse stringideks ja teisendatakse vajalike stringivormingute (näiteks , ja #NUM! eemaldamine) kaudu.

Saadud andmeid salvestatakse sõnastiku kujul *{"õpilase nr 1 id": {"atribuut1": väärtus, "atribuut2": väärtus, ...}}.* Funktsioon loob õpilase ID alusel seotud atribuudid ja väärtused, lisades vastavalt nendele määratud tingimustele ja loogikale, ning tagastab sõnastiku.

Lõpuks tagastatakse sõnastik, mis sisaldab kõikide õppijate andmeid ja nendega seotud atribuutide väärtusi.

```
def muud_andmed():
    ## Exceli fail õppijate demograafiliste andmete ja õppestaatusega salvestatud csv tüüpi tab delimited failiks.
    ## Selles funktsioonis teisendatakse see dict objektiks kujul {"õpilase nr 1 id":{"atr1":value,"atr2":value,"etc":value}}
    try:
        with open("6ppijate-andmed-p2eva6pe-2017-2023.txt",'r',encoding="ANSI") as f:
            kkp = f.read()
    except:
        raise Exception(f"Lugedes faili, tekkis viga")
    op = {}
    csv = []
    s=kkp.strip().split('\n')
    c = 0
    for x in s:
        temp = x.split('\t')
        csv.append(temp)
        kood = temp[0].strip()
        if temp[9]=="Eksmatrikuleerimine" or temp[9]=="Lõpetamine" or temp[9]=="Koolist väljaarvamine":
            op[kood]={}
            if c > 0:
                for i in range(len(temp)):
                    val = csv[c][i].replace(',','').replace(' ','-').replace('#NUM!','21')

                    # kuna kõik väärtused on siin stringi tüüpi, siis teisendame õpilase vanuse numbriliseks
                    if val.isnumeric() and len(val) == 2:
                        val = int(val)

                    # Võtame õpperühma koodist vaid viimase tähe, mis vastas alguses õppekeelele
                    if csv[0][i]=="Õpperühm":
                        val = val[-1]
                    if csv[0][i]=="Nominaalaja lõpp":
                        # Kui nominaalaja lõpp oli nominaalne, siis osutusid mingid aastad olulisteks atribuutideks,
                        # mis tundus olevat seotud koroona perioodiga. Seega, Nominaalaja lõpu numbriliseks muutmise võib ka välja kommenteerida.
                        tykid=val.split(".")
                        val=int(tykid[-1])

                    if val == "":
                        val = None

                    op[kood][csv[0][i]] = val

        c=c+1
    return op
```

Joonis 2.4. Skripti osa, mille eesmärk on võtta tekstifailist sisse loetud ja puhastatud andmed.

Joonis 2.5 eesmärk on lugeda ja analüüsida JSON-andmeid. *Read_json* funktsioon loeb JSON-faili ja teisendab selle sisu Pythoni sõnastikuks. Antud funktsioon käsitleb vigu, mis võivad faililugemise ajal tekkida, faili mitte leidmine või kodeerimisprobleemid, ning tõstab välja osa, kui viga tekib.

Flatten_json funktsioon võtab vastu pesastatud JSON-objekti (mis võib sisaldada sõnastikke ja loendeid) ning teisendab selle lameks sõnastikuks, kus iga võti esindab teed pesastatud väärtusele (kasutades punkte kui eraldajaid). Funktsiooni eesmärk on ühtlasi aidata keerulisi JSON-struktuure muuta lihtsamaks, mis on lihtsamini töödeldav.

Json_to_arff funktsioon kasutab tasandatud andmeid, et genereerida ARFF (Attribute-Relation File Format) fail. Antud funktsioon tõmbab tasandatud JSON-andmetest välja unikaalsed atribuudid, eemaldab spetsiifilised atribuudid, mis ei ole analüüsi jaoks olulised ("Staatus") ning seejärel kirjutab need atribuudid koos nende väärtustega ARFF-faili. Funktsioon aitab teisendada struktureeritud JSON-andmed vormingusse, mis on sobivam analüüsiks ja masinõppemudelite jaoks.

```python
def read_json(filename: str) -> dict:
    try:
        with open(filename, "r", encoding="UTF-8") as f:
            data = json.loads(f.read())
    except:
        raise Exception(f"Lugedes {filename} faili, tekkis viga")
    return data



def flatten_json(nested_json, parent_key='', sep='.'):
    """
    Recursiivselt tasandame pesastatud JSON faili, mis sisaldab dicte ja liste.
    """
    items = []
    if isinstance(nested_json, dict):
        for k, v in nested_json.items():
            new_key = f"{parent_key}{sep}{k}" if parent_key else k
            items.extend(flatten_json(v, new_key, sep=sep).items())
    elif isinstance(nested_json, list):
        for i, item in enumerate(nested_json):
            new_key = f"{parent_key}[{i}]"
            items.extend(flatten_json(item, new_key, sep=sep).items())
    else:
        items.append((parent_key, nested_json))
    return dict(items)

def json_to_arff(json_data, relation_name='dataset', output_file='output.arff'):
    """
    Konverteerime JSON faili ARFF failiks.
    """
    # Rasandame JSONi
    flattened_data = [flatten_json(record) for record in json_data]

    # Unikaalsed atribuudid
    attributes = sorted(set(k for record in flattened_data for k in record.keys()))

    #Kõrvaldame sorteeritud attributes listist ennustatava atribuudi "Staatus", et lisada see viimaseks
    attributes.remove("Staatus")

    #kõrvaldame atribuudid, mis on otseselt seotud ennustatav atribuudiga
    attributes.remove("status")
    attributes.remove('Käskkirja kinnitamise kp')
    attributes.remove('Käskkirja liik')
    attributes.remove('Käskkirja põhjus')
```

Joonis 2.5 . Skripti osa, mille eesmärk on teisendada JSON- andmeid.

Joonis 2.6 eesmärk on teisendada andmed ARFF-failiks, mida saab kasutada masinõppe platvormil Weka. Vaja on eemaldada üleliigsed atribuudid: õpilase ID-d ("id" ja "Pseudonüümne ID") eemaldatakse, kuna need pole analüüsi jaoks olulised. Lisaks on vaja määrata sihtmuutuja, sest ennustatav atribuut "Staatus" lisatakse nimekirja viimaseks, sest see on analüüsi eesmärk.

Seejärel genereeritakse ARFF-fail. Faili algusesse kirjutatakse andmestiku nimi ja atribuutide nimed ning tüübid. Numbrilistele, tekstilistele ja kategoorilistele andmetele määratakse automaatselt sobiv tüüp. Andmete sektsiooni kirjutatakse iga rida vastavalt atribuutide väärtustele. Vaja on käsitleda ka puuduvaid väärtusi. Kui väärtus puudub, märgitakse see küsimärgiga „?“.

Lõpuks luuakse valmis ARFF-fail, mida saab kasutada masinõppe mudelite treenimiseks ja testimiseks.

```
    #kõrvaldame üleliigsed õpilase ID-ga seotud atribuudid
    attributes.remove("id")
    attributes.remove("Pseudonüümne ID ")

    #lisame listi lõppu ennustatava atribuudi "Staatus"
    attributes.append("Staatus")

    # Kirjutame ARFF faili
    with open(output_file, 'w', encoding="ANSI") as f:
        # Kirjutame relation nime
        f.write(f"@relation {relation_name}\n\n")

        # Kirjutame atribuudid
        for attr in attributes:
            tyyp = atr_tyyp(attr,flattened_data)
            f.write(f"@attribute '{attr}' {tyyp}\n")


        f.write("\n@data\n")

        # Kirjutame andmed
        for record in flattened_data:
        #    row = [f"{record.get(attr, '?')}" for attr in attributes]
            row =[str_v_num(record.get(attr, '?')) for attr in attributes]
            f.write(f"{','.join(row)}\n")



def atr_tyyp(atr, data):
    tyyp = "string"
    h = set()
    for ob in data:
        if atr in ob:
            if(type(ob[atr]) is int or type(ob[atr]) is float):
                tyyp="numeric"
            elif(type(ob[atr]) is str):
                h.add(ob[atr])
    if(len(h)>0):
        sep=", "
        tyyp = "{"+sep.join(h)+"}"
    return tyyp
```

Joonis 2.6. Skripti osa, mille eesmärk on teisendada andmed ARFF- failiks

Joonis 2.7 eesmärk on teisendad õpilase päevikuandmed ja demograafilised andmed üheks koondatud ja lihtsustatud andmeobjektiks. Selle eesmärk on luua tasandatud (lamendatud) andmestikku, mis sisaldab päeviku põhjal arvutatud andmeid ning lisada neile eraldi failist pärit demograafilised ja staatusega seotud andmed.

Kui õpilasel on päevikus erinev arv hindeid ja teisi seotud andmeid, siis koondatakse need näiteks keskmisteks hinneteks ja loetakse kokku puudulike või tegemata tööde arv. Lisaks sellele lisatakse demograafilised andmed, nagu vanus, sugu ja õppija staatus (nt kas lõpetanud, eksmatrikuleeritud või väljaarvatud).

Kood kontrollib esmalt, kas vastava õpilase ID on demograafilistes andmetes olemas. Kui ID-d ei leita, siis tagastatakse tühi objekt. Kui andmed on olemas, kogutakse päevikust vajalikke näitajaid (nt puudumiste ja hilinemiste arv, puudumiste protsent) ja ühendatakse need demograafiliste andmetega.

```python
def str_v_num(v):
    if type(v) is str and v != '?' and v!='#NUM!':
        return f"'{v}'"
    elif v is None or v == '#NUM!':
        return f"?"
    else:
        return f"{v}"


# Ühe õpilase andmete teisendamine tasaseks või lamendatud (flatten vs nested, ehk lame vs pesastatud) dict objektiks
# siin toimub ka päeviku andmete agregatsioon, kuna õpilastel võib olla erinev arv hindeid samades õppeainetes,
# siis arvutatakse keskmised hinded ja loetakse kokku puudulike/tegemata/arvestamata tööde arv igas õppeaines
# lõpus lisatakse sellele objektile andmed eraldiseisvast exceli failist, kus on natuke demograafilist infot ning õppija staatus,
# mida käesolevas töös ennustatakse
def yks_dict(ob):
    """
    # laeme exceli tabelis õpilaste demograafilised andmed ja staatuse dict objektiks teisendatuna
    # muud_andmed() ei väljasta hetkel veel õppivate või akadeemilisel puhkusel olevate õppurite andmeid.

    Parameetrid:
    ob (dict): Pesastatud dict õpilase päevikute andmetega.

    Tagastab:
    dict: Tasandatud dict õpilase andmetega või tühi dict, kui vastavat demograafilist andmestikku ei leidu.
    """
    # Laeme demograafilised andmed
    mandmed = muud_andmed()

    student_id = ob.get("id")
    if student_id not in mandmed:
        return {}

    temp_p = {
        key: ob.get(key) for key in [
            "id", "status", "totalAbsences", "withoutReasonAbsences",
            "withReasonAbsences", "beingLate", "journalEntryLessons",
            "lessonAbsencePercentage", "withoutReasonAbsencesPercentage"

        ]
    }
```

Joonis 2.7. Skripti osa, mille eesmärk on teisendada õpilase päevikuandmed ja demograafilised andmed koondatud ja lihtsustatud andmeobjektiks.

Joonis 2.8 töötleb õpilaste päevikuandmed ja demograafilise teabe masinõppeks sobivasse vormingusse, salvestades tulemused ARFF-failidena. Kood analüüsib iga õpilase päevikuandmeid nii:

- Keskmine hinne (kui hindeid on).
- Arvestatud tööde arv.
- Mittearvestatud tööde arv.
- Kokku esitatud tööde arv.

Iga näitaja seostatakse konkreetse õppeaine nimega, näiteks „Matemaatika_keskmine_hinne”.

Päevikute põhjal töödeldud andmed ühendatakse eraldi failist saadud demograafiliste andmetega, mis sisaldavad näiteks õpilase staatust (kas lõpetatud, katkestatud) ja muid isikuandmeid. Kui demograafilisi andmeid pole, jäetakse õpilase rida välja.

Funktsioon loeb etteantud JSON-failidest õpilaste andmed, teisendab need lamedaks vormiks (kus kõik pesastatud andmed on lihtsamini kasutatavas tabelivormis) ja salvestab need ARFF-formaadis faili. Töödeldakse erinevate mustrite järgi sorteeritud JSON-faile, näiteks üldised failid, IT eriala failid ja TA eriala failid ning igaüks salvestatakse eraldi ARFF-faili.

Kood genereerib kolm ARFF-faili: *output.arff*, *IT-output.arff* ja *TA-output.arff*. Need sisaldavad lamedaks tehtud andmeid õpilaste kohta, ühendades päeviku- ja demograafilised andmed.

```
    # Töötleme päevikuid
    for result in ob.get("resultColumns", []):
        journal_result = result.get("journalResult")
        if isinstance(journal_result, dict):
            temp_p["journal_id"] = journal_result.get("id", "")
            p_nimi = ""
            hinne_sum = hinne_count = kokku_k = hinne_a = hinne_x = 0

            for sk in journal_result.get("results", []):
                p_nimi = sk["journal"]["nameEt"].replace(",", "")
                grade_code = sk["grade"]["code"].split("_")

                kokku_k += 1
                if grade_code[1] == "A":
                    hinne_a += 1
                elif grade_code[1] in {"MA", "X"}:
                    hinne_x += 1
                else:
                    hinne_sum += int(grade_code[1])
                    hinne_count += 1

            temp_p[f"{p_nimi}_keskmine_hinne"] = hinne_sum / hinne_count if hinne_count else 0
            temp_p[f"{p_nimi}_arvestatud"] = hinne_a
            temp_p[f"{p_nimi}_mittearvestatud"] = hinne_x
            temp_p[f"{p_nimi}_hindeid_kokku"] = kokku_k

    # Ühendame õpilase demograafilise andmestikuga
    return {**temp_p, **mandmed[student_id]}



def arff_faili_loomine(json_allikas, rel, arff_output_fail):
    uus_json=[]
    for f_name in glob(json_allikas):
        nested_json = read_json(filename=f_name)
        for obj in nested_json:
            if("id" in yks_dict(obj)):
                uus_json.append(yks_dict(obj))

    json_to_arff(uus_json, relation_name=rel, output_file=arff_output_fail)


arff_faili_loomine('data/*.json','dataset','output.arff')
arff_faili_loomine('data/IT*.json','IT-dataset','IT-output.arff')
arff_faili_loomine('data/TA*.json','TA-dataset','TA-output.arff')
```

Joonis 2.8. Skripti osa, mille eesmärk on töödelda õpilaste päevikuandmeid ja ühendab demograafilise teabe masinõppeks sobivasse vormingusse.

Atribuutide arvud on järgmised:

- IT-TA: 954
- IT: 630
- TA: 598

2.3 Masinõpe

Antud peatükis kirjeldatakse lähemalt masinõpet, töös kasutatud masinõppe algoritme, mille abil ennustati lõplikusse valimisse jäänud andmeid.

Masinõpe (machine learning) on tehnoloogia, mis põhineb andmete analüüsil ja kasutab erinevaid algoritme, et luua matemaatilisi mudeleid ning teha ennustusi. Prognooside kvaliteet sõltub suuresti andmete hulgast – mida rohkem andmeid, seda täpsemad

mudelid saab luua. Selle termini võttis esimesena kasutusele Arthur Samuel 1959. aastal. Masinõpe on osa tehisintellekti (artificial intelligence) valdkonnast, mis keskendub algoritmide loomisele, võimaldades arvutitel õppida andmete ja varasemate kogemuste põhjal iseseisvalt järeldusi tegema [7].

Masinõppes liigitatakse tunnused peamiselt kahte tüüpi: diskreetsed tunnused (*discrete*), mis on tekstipõhised ja kuuluvad piiratud väärtuste hulka, ning pidevad tunnused (*continuous*), mis on arvulised ja võivad võtta mis tahes väärtusi teatud vahemikus. Lisaks võib tunnuseid käsitleda ka binaarsetena, mis tähistavad kahte võimalikku olekut, näiteks "õige" või "vale" (0 või 1) või "kuulub klassi" või "ei kuulu klassi" [8].
Antud töös on väljalangevust ennustavate mudelite treenimiseks ja testimiseks kasutati masinõppe programmi nimega Weka [9]. Programm sisaldab masinõppe algoritmide kogumit, mis on loodud spetsiaalselt andmekaevandamise ülesannete täitmiseks.
Weka programmis on võimalik eeltöödeldud andmeid muuta, kasutades algoritme ja saada treenitud andmete baasil mudeleid, mis ennustavad õpilase väljalangevust. Esimeseks sammuks on vaja valida *preprocess-is* eeltöödeldud andmete fail. Vaikimis eeldatakse ARFF- formaadis faili, aga sobib ka csv. ning mitmed muud formaadid. Seejärel on võimalik näha, kui palju on andmestikus atribuute ja isendeid. Edasi saab liikuda *Classify* vaheaknasse, kus saab erinevate algoritmide abil hakata ennustusmudelit treenima ja loodud mudelite täpsust testima.

2.4 Kasutatud masinõppe meetodid

Antud alapeatükis on tutvustatud neid masinõppe algoritme, mida antud töös väljalangevuse ennustamise mudelite treenimiseks kasutati. Algoritmide valik põhineb autori õpi- ja katsetuskogemusest.

C4.5 decision tree (J48) on algoritm, mida kasutatakse otsustuspuu loomiseks. Klassifitseerimiseks saab kasutada C4.5 mgenereeritud otsustuspuid ja seetõttu nimetatakse C4.5 sageli statistiliseks klassifikaatoriks. C4.5 valib puu igas sõlmes andmete atribuudi, mis kõige tõhusamalt jagab selle valimikomplekti ühes või teises klassis rikastatud alamhulkadeks. Jagamiskriteeriumiks on normaliseeritud infovõimendus. Otsuse tegemiseks valitakse atribuut, millel on suurim normaliseeritud teabevõimendus. Seejärel kordub C4.5 algoritm partitsioonidega alamloendites [10] [11].

Random forest on masinõppe meetod, mis toimib mitme otsustuspõhise mudeli (otsustuspuu) kombineerimise põhimõttel. Koolitusfaasis luuakse iga otsustuspuu juhuslikult valitud andme- ja tunnuste alamhulkade põhjal. Selline juhuslikkus suurendab puude vahelist varieeruvust, vähendades üleõppimise ohtu ning parandades mudeli üldist prognoosimisvõimet. Juhuslike metsade peamine tugevus seisneb nende võimes käsitleda keerukaid andmeid ja andmestikus leiduvat müra. Klassifitseerimisülesannetes määratakse lõplik ennustus puude hääletuse põhjal, kus enim hääli saanud klass valitakse lõpptulemuseks [12].

Gaussian tüüpi **Naive Bayes** on meetod, kus arvestatakse pidevaid atribuute ja andmefunktsioonid järgivad kogu andmestiku Gaussi jaotust. Gaussian Naive Bayes on teatud tüüpi klassifitseerimisalgoritm, mis töötab pidevatel normaalselt jaotatud funktsioonidel, mis põhineb Naive Bayesi algoritmil [13].

Simple Logistics on üks enim kasutatud masinõppe algoritme binaarseks klassifitseerimiseks. See on masinõppe algoritm, mis rakendab logistilist regressiooni kuid sisaldab optimeerimisi ja täiustusi, mis muudavad selle hästi sobivaks andmekaevandamise ja klassifitseerimise ülesannete jaoks. Näiteks võib Simple Logistics sisaldada funktsioone, nagu tunnuste automaatne valik, et parandada mudeli üldistust [14].

Sequential Minimal Optimization (***SMO***) on algoritm, mida kasutatakse tugivektorite masinate (SVM, *Support Vector Machines*) efektiivseks treenimiseks. See on eriti oluline suurte andmekogumite ja keerukate klassifitseerimisprobleemide korral, kuna SMO vähendab arvutuslikku keerukust. SMO lahendab SVM-i jaoks kvadraatse optimeerimisprobleemi, mis on tavaliselt keeruline ja ressursimahukas. Selle asemel, et lahendada suur optimeerimisprobleem korraga, jagab SMO selle väiksemateks osadeks [15].

2.5 Tulemuste valideerimine

Tulemuste testimiseks on valitud 10-kordne ristvalideerimine.

Ristvalideerimine (*Cross-Validation*) on meetod mida kasutatakse algoritmide mudelite hindamiseks ja valideerimiseks. See aitab vältida algoritmi ala- või ületreenimist. Antud töös määrasin ristsobivuse väärtuseks 10, mis omakorda jagas andmed kümneks osaks. Ristsobivus on hea meetod ainult siis kui tegemist on väikest sorti andmekogudega. Suurte puhul võib see nõuda palju aega ja tulemused ei pruugi tulla täpsed. Protsess toimib nii, et andmed jagatakse X võrdseks osaks (X on määratud osade ehk foldide arv), seejärel treenitakse mudelit ühe osaga ja teise osaga testitakse. Antud protsess korratakse täpselt nii palju kui on määratud X ja iga kord valitakse täiesti erinevad osad treenimiseks ja testimiseks. Lõpus arvutatakse saadud tulemuste baasil keskmised ja selle baasil saab teada alogritmi mudeli täpsust [16].

3. TULEMUSED

Antud peatükis antakse ülevaade Tallinna Polütehnikumi IT õpilaste väljalangevuse ennustamise tulemustest. Tulemused esitatakse loetletud masinõppe algoritmide rakendamisel loodud mudelite 10-kordse ristvalideerimise keskmiste tulemustena.

3.1 Ennustamise täpsused

Ennustamisel kasutasin masinõppe algoritme, mille tulemused on välja toodud alljärgnevates tabelites järgmiste lühenditena:

- **NV** – *Naive Bayes*
- **RF** – Random Forest
- **SL** – *Simple Logistics*
- **J48** - *Decesion tree J48*
- **SMO** - *Sequential Minimal Optimization*

Klassifitseerimise tulemuste võrdluseks on tabelis 5 masinõppe algoritmi – *Naive Bayes* (NB), *Simple Logistic* (SL), *Random Forest* (RF), *Sequential Minimal Optimization* (SMO) ja J48 ennustustulemused.

Need tulemused on välja toodud F1-skooridena, mis mõõdavad klassifitseerimise täpsuse ja leitavuse tasakaalu. Lisaks on igale algoritmile arvutatud ka F1-skooride kaalutud keskmine, et anda ülevaade mudelite jõudlusest, arvestades klasside jaotust. F1-skoor on mõõdik, mis ühendab täpsuse (kui palju ennustatud positiivsetest vastustest on õiged) ja määratavuse (kui palju tegelikke positiivseid vastuseid suudeti õigesti leida). Täpsus (precision) - kui palju ennustatud positiivsetest juhtumitest on õiged ning määratavus (recall) - kui palju tegelikest positiivsetest juhtumitest suudeti õigesti tuvastada [17].

3.2 F1-skooride analüüs masinõppe algoritmide poolt lõpetanute ja katkestanute klassifitseerimisel

Tabelist (1) ilmneb, et lõpetanute klassis saavutavad algoritmid järjepidevalt kõrgemaid F1-skoore võrreldes katkestanute klassiga. See viitab tasakaalustamatusele andmestikus, kus katkestanute klass võib olla vähem varieeruv. IT andmestik annab üldiselt paremaid tulemusi võrreldes TA andmestikuga.

IT ja TA andmestike ühendamine toob esile ennustustäpsuse langust, eriti RF ja J48 algoritmide puhul. Kaalutud keskmised F1-skoorid näitavad, et SMO ja NB algoritmid paistavad silma oma stabiilsuse poolest, pakkudes häid tulemusi nii lõpetanute kui katkestanute klassi ennustamisel.

SMO ja NB on parimad kandidaadid antud klassifitseerimisülesande lahendamiseks, näidates lõpetanute kui katkestanute klassis häid tulemusi. RF algoritmi madalamad F1-skoorid ja ebastabiilsem jõudlus viitavad, et see algoritm ei pruugi antud andmestikus sobida. Tabeli põhjal saab teha esialgse järelduse, et SMO ja NB algoritmide optimeerimine võiks viia veelgi paremate tulemusteni.

Tabel 1. Klassifitseerimise tulemused F-skoori järgi

Andmestik	Klass	NB	SL	RF	SMO	J48
IT	Lõpetanud	0.912	0.884	0.865	0.901	0.879
	Katkestanud	0.899	0.868	0.796	0.880	0.840
	Kaalutud keskmine	0.906	0.877	0.835	0.891	0.860
TA	Lõpetanud	0.892	0.897	0.852	0.897	0.886
	Katkestanud	0.860	0.851	0.883	0.851	0.849
	Kaalutud keskmine	0.878	0.877	0.868	0.877	0.870
IT ja TA	Lõpetanud	0.898	0.868	0.837	0.896	0.828
	Katkestanud	0.878	0.817	0.769	0.861	0.740
	Kaalutud keskmine	0.889	0.845	0.806	0.880	0.788

3.3 Eksimismaatriksid

Tabel 2 esitab eksimismaatriksi TA andmestiku jaoks, kus A tähistab katkestanud õpilasi ja B lõpetanud õpilasi.

Eksimismaatriks näitab, kui täpselt need algoritmid suudavad tulemusi ennustada.

Naive Bayes (NB) ennustas 46 väljalangenut õigesti ja 7 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 62 õigesti, kuid 8 liigitas valesti katkestanuteks.

Sequential Minimal Optimization (SMO) ennustas 43 väljalangenut õigesti ja 10 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 65 õigesti, kuid 5 liigitas valesti katkestanuteks.

Random Forest (RF) ennustas 39 väljalangenut õigesti ja 14 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 64 õigesti, kuid 6 liigitas valesti katkestanuteks.

J48 - ennustas 45 väljalangenut õigesti ja 8 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 62 õigesti, kuid 8 liigitas valesti katkestanuteks.

Simple Logistics ennustas 43 väljalangenut õigesti ja 10 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 65 õigesti, kuid 5 liigitas valesti katkestanuteks.

Parim algoritm on Naive Bayes, kuna see suudab ennustada edukalt nii väljalangenuid kui ka lõpetanuid. J48 on hea alternatiiv, kuid jääb täpsuselt veidi alla. Random Forest ja SMO ei sobi, kuna nende veamäärad on kõrged, eriti lõpetanute puhul.

Tabel 2. TA eksimismaatriks

Algoritm / Klass	NB		SL		RF		SMO		J48	
	A	B	A	B	A	B	A	B	A	B
Katkestanud	46	7	43	10	39	14	43	10	45	8
Lõpetanud	8	62	5	65	6	64	5	65	8	62

Tabel 3 esitab eksimismaatriksi IT andmestiku jaoks, kus A tähistab katkestanud õpilasi ja B lõpetanud õpilasi.

Naive Bayes (NB) ennustas 58 väljalangenut õigesti ja 8 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 67 õigesti, kuid 5 liigitas valesti katkestanuteks.

Sequential Minimal Optimization (SMO) ennustas 55 väljalangenut õigesti ja 11 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 68 õigesti, kuid 4 liigitas valesti katkestanuteks.

Random Forest (RF) ennustas 52 väljalangenut õigesti ja 14 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 6 õigesti, kuid 4 liigitas valesti katkestanuteks.

J48 - ennustas 50 väljalangenut õigesti ja 16 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 69 õigesti, kuid 3 liigitas valesti katkestanuteks.

Simple Logistics ennustas 56 väljalangenut õigesti ja 10 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 65 õigesti, kuid 7 liigitas valesti katkestanuteks.

Eksimismaatriks näitab, et masinõppe mudelid, eriti *Random Forest*, suudavad hästi ennustada õpilaste väljalangemist IT eriala andmete põhjal.

Tabel 3. IT eksimismaatriks

Algoritm / Klass	NB		SL		RF		SMO		J48	
	A	B	A	B	A	B	A	B	A	B
Katkestanud	58	8	56	10	52	14	55	11	50	16
Lõpetanud	5	67	7	65	4	68	4	68	3	69

Kombineeritud IT ja TA andmestiku eksimismaatriks aitab hinnata masinõppe algoritmide täpsust, kui prognoositakse õpilaste väljalangemist kahe eriala ühisandmestiku põhjal.

Naive Bayes (NB) - ennustas 104 väljalangenut õigesti ja 15 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 128 õigesti, kuid 14 liigitas valesti katkestanuteks.

Simple Logistics (SL) - ennustas 89 väljalangenut õigesti ja 30 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 132 õigesti, kuid 10 liigitas valesti katkestanuteks.

Random Forest (RF) - ennustas 83 väljalangenut õigesti ja 36 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 128 õigesti, kuid 14 liigitas valesti katkestanuteks.

Sequential Minimal Optimization (SMO) - ennustas 96 väljalangenut õigesti ja 23 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 134 õigesti, kuid 8 liigitas valesti katkestanuteks.

J48 - ennustas 77 väljalangenut õigesti ja 42 väljalangenut liigitas valesti lõpetanute klassi. Lõpetanute klassist liigitas 12 õigesti, kuid 130 liigitas valesti katkestanuteks.

Tabel 4. Kombineeritud IT ja TA eksimismaatriks

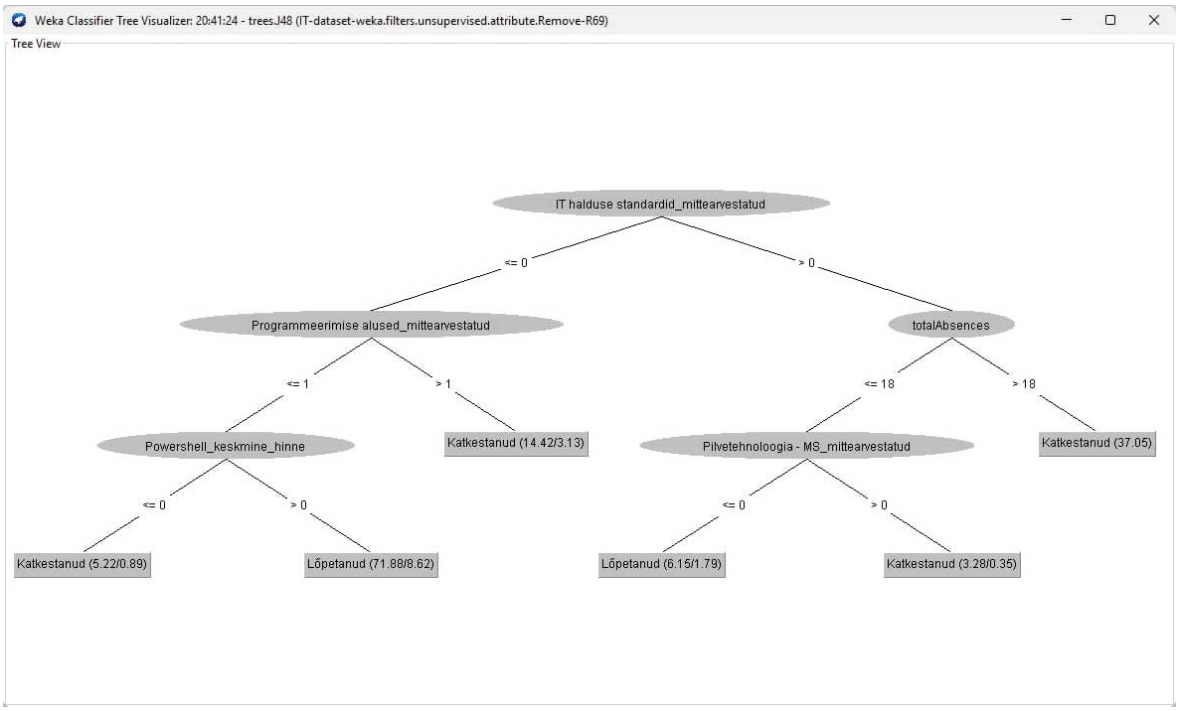
Algoritm / Klass	NB		SL		RF		SMO		J48	
	A	B	A	B	A	B	A	B	A	B
Katkestanud	104	15	89	30	83	36	96	23	77	42
Lõpetanud	14	128	10	132	14	128	8	134	12	130

3.4 Visualiseerimine

Joonis 4.1 otsustuspuu eesmärk on prognoosida, kas IT õppekava õpilane lõpetab õpingud või katkestab. See põhineb andmetel, mis sisaldavad infot näiteks õppeainete läbimise, hinnete ja puudumiste kohta.

Kui õpilane on läbinud aine "IT halduse standardid" ja tema teised tulemused (puudumiste arv ja hinnete keskmine) on head, siis ennustab puu suure tõenäosusega, et ta lõpetab õpingud.

Kui "Powershell" õppeaine keskmine hinne on üle nulli ja puudumisi on vähe, siis õpilane klassifitseeritakse lõpetanuks. Kui õpilasel on mittearvestatud aineid ("IT halduse standardid" või "Pilvetehnoloogia – MS") või kui tal on palju puudumisi (rohkem kui 18 tundi), siis klassifitseeritakse õppija suure tõenäosusega katkestanuks.

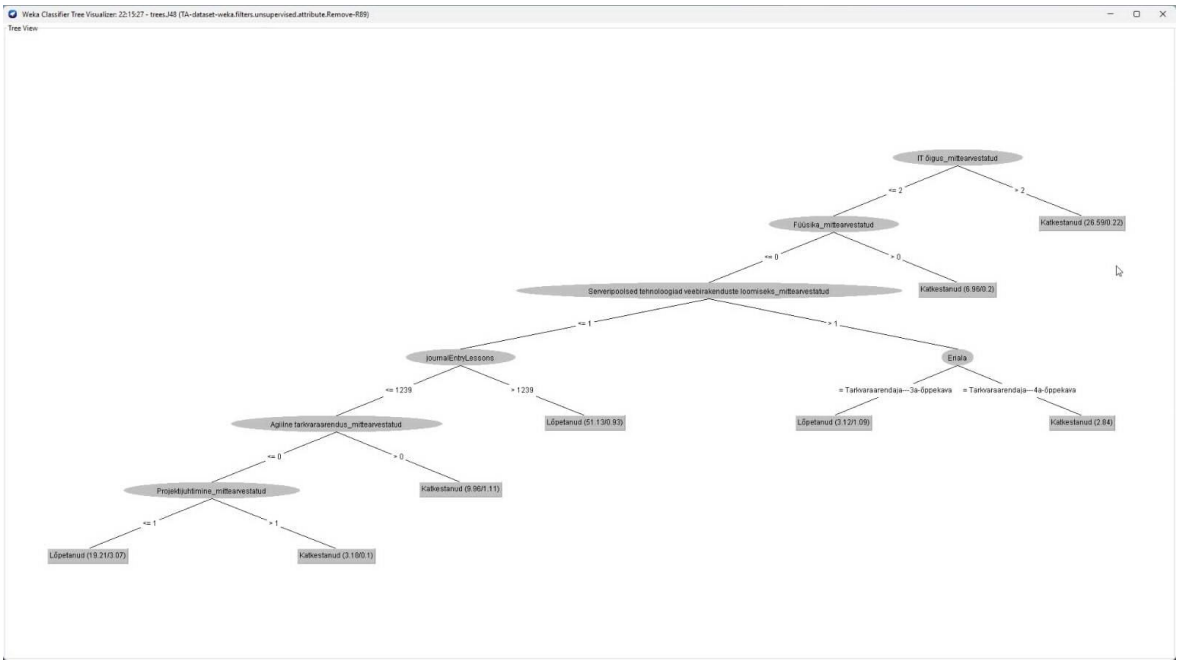


Joonis 3.1. IT otsustuspuu

Joonis 4.2 otsustuspuu eesmärk on prognoosida, kas õpilased katkestavad õpingud või lõpetavad edukalt, toetudes akadeemilistele ja käitumuslikele näitajatele. Otsustusprotsessi alguses mängib olulist rolli õppeaine "IT õigus", kus aine läbimata jätmine suurendab oluliselt katkestamise tõenäosust. Teiseks oluliseks teguriks on "Füüsika" tulemused - madal sooritus ja puudulik arvestamine suunavad otsustuspuu katkestamise harusse. Lisaks mõjutab õpingute lõpetamist oluliselt osalus aine "Serveripoolsed tehnoloogiad veebirakenduste loomiseks", kus vähene aktiivsus on seotud katkestamise riskiga.

Kui õpilased ületavad varasemad kriteeriumid, saab otsustavaks teguriks nagu "Agile tarkvaraarendus" ja "Projektijuhtimine" arvestuse puudumine. Ilmneb, et madalad tulemused või tundides mitte kohal käimine nendes ainetes suurendavad katkestamise tõenäosust, samas kui positiivsed sooritused viivad lõpetamiseni. Lisaks on oluline ka *journalEntryLessons* näitaja, kus väiksem arv tulemusi on seotud katkestamisega ja suurem arv lõpetamisega.

Olulist rolli mängib ka õppekava tüüp. 3-aastase tarkvaraarenduse õppekava õpilased lõpetavad suurema tõenäosusega, samas 4-aastase programmi õpilastel on katkestamise risk kõrgem.

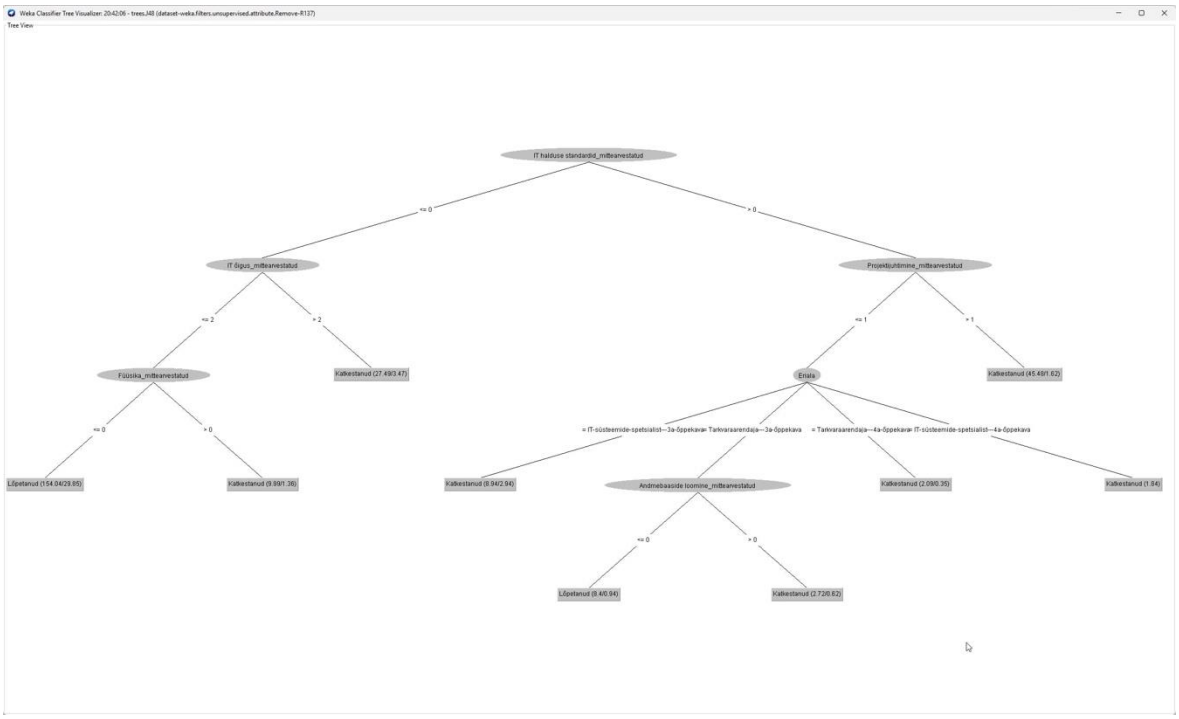


Joonis 3.2. TA otsustuspuu

Joonis 4.3 otsustuspuus mängivad olulist rolli õppeained “IT/halduse standardid”, “IT-õigus”, “Füüsika” ja “Projektijuhtimine”. Kui nende ainete sooritus on puudulik, suureneb katkestamise risk.

Eriala lõikes ilmneb, et 3-aastase tarkvaraarenduse õppekava õpilased lõpetavad edukamalt, samas kui 4-aastase programmi õpilastel on kõrgem katkestamise tõenäosus. Samuti on oluline aine “Andmebaaside loomine”, kus aktiivsus ja arvestuse olemasolu toetavad lõpetamist, samas kui tegevusetus suunab katkestamise suunas.

Otsustuspuu näitab, et edukus sõltub nii konkreetsete ainete sooritamisest kui ka õppekava kestusest, kus IT-valdkonna õppekavade läbimine on suurema riskiga, eriti kui olulised ained jäävad arvestamata.



Joonis 3.3. Kombineeritud IT ja TA otsutuspuu

3.5 Simple logistic mudeli analüüs: tunnuste mõju positiivsele ja negatiivsele lõpetamisele

Antud peatükis antakse ülevaade IT ja TA normaliseeritud andmestiku põhjal ülevaade, millised õppeained mõjutavad positiivselt lõpetamist ja millised õppeained mõjutavad kooli lõpetamist negatiivselt.

IT:

```
Class Lõpetanud :
-3.75 +
[Arvutivõrgud ja võrguseadmed_mittearvestatud] * -1.21 +
[Digitaaltehnika ja mikroprotsessortehnika_mittearvestatud] * -2.63 +
[Füüsika_keskmine_hinne] * 2.99 +
[IT halduse standardid_hindeid_kokku] * 2.62 +
[IT halduse standardid_keskmine_hinne] * 1.82 +
[IT halduse standardid_mittearvestatud] * -2.58 +
[Infoturve_hindeid_kokku] * 0.9  +
[Klienditeenindus_hindeid_kokku] * -1.42 +
[Kutse-eetika_mittearvestatud] * -2.14 +
[Linux/BSD operatsioonisüsteemide diagnostika_hindeid_kokku] * -1.06 +
[Linux/BSD operatsioonisüsteemide paigaldus_mittearvestatud] * -2.5 +
[Lõpetatud kooli nimi=Tallinna-Tehnikagümnaasium] * 1.47 +
[Lõpetatud kooli nimi=Kohtla-Järve-Tammiku-Põhikool] * -1.29 +
[Operatsioonisüsteemid_mittearvestatud] * -1.87 +
[Operatsioonisüsteemide sidumine_hindeid_kokku] * 0.98 +
[Pilveteenuste planeerimine_mittearvestatud] * -0.9 +
[Pilvetehnoloogia - Google_keskmine_hinne] * 1.6  +
[Powershell_arvestatud] * -0.91 +
[Powershell_keskmine_hinne] * 2.35 +
[Raster- ja vektorgraafika alused_keskmine_hinne] * 0.87 +
[SQL_mittearvestatud] * -1.58 +
[Töötamise õiguslikud alused töötervishoid ja tööohutus_mittearvestatud] * -1.69 +
[Vene keel_arvestatud] * -2.12 +
[beingLate] * -1.73
```

Joonis 3.4. IT SL mudel

Positiivse mõjuga atribuudid (klass "Lõpetanud"):

Füüsika keskmine hinne: (+2.99)

IT halduse standardid hinnete kokku: (+2.62)

Infoturve hinnete kokku: (+0.9)

Negatiivse mõjuga atribuudid (klass "Lõpetanud"):

Digitaaltehnika ja mikroprotsessoritehnika mittearvestatud: (-2.63)

SQL mittearvestatud: (-1.58)

TA:

```
Class Lõpetanud :
-0.82 +
[Agiilne tarkvaraarendus_hindeid_kokku] * -0.54 +
[Agiilne tarkvaraarendus_keskmine_hinne] * 4.67 +
[Agiilne tarkvaraarendus_mittearvestatud] * -3.85 +
[Agiilsed tarkvaraarenduse metoodikad_mittearvestatud] * -2.38 +
[Algoritmid_hindeid_kokku] * 4.22 +
[Andmebaaside alused_hindeid_kokku] * -0.99 +
[Eesti keel ja kirjandus_keskmine_hinne] * -2.65 +
[Eesti keel_hindeid_kokku] * 1.16 +
[Ettevõtluse alused_keskmine_hinne] * 0.95 +
[HTML CSS_mittearvestatud] * -1.83 +
[IT halduse standardid_hindeid_kokku] * 0.87 +
[IT juhtimise standardid_arvestatud] * 7.9  +
[IT õigus_arvestatud] * 1.65 +
[IT õigus_hindeid_kokku] * -0.93 +
[IT õigus_mittearvestatud] * -2.79 +
[Inglise keel_mittearvestatud] * -2.23 +
[Karjääri planeerimine_keskmine_hinne] * 1.96 +
[Kasutajaliides_mittearvestatud] * -1.47 +
[Küberturbe alused_hindeid_kokku] * -0.95 +
[Lihtsamate rakenduste loomine konkreetsele platvormile_keskmine_hinne] * 2.14 +
[Lõpetatud kooli nimi=Tallinna-Ühisgümnaasium] * -1.47 +
[Matemaatika_keskmine_hinne] * -1.63 +
[Microsofti platvormil baseeruvate mobiilsete rakenduste loomine_mittearvestatud] * -1.79 +
[Muusika_mittearvestatud] * -1.61 +
[Programmeerimine II_keskmine_hinne] * 2.51 +
[Programmeerimise alused praktilised tööd_keskmine_hinne] * -2.69 +
[Projektijuhtimine_mittearvestatud] * -2.04 +
[Rakendusmatemaatika_hindeid_kokku] * -2.77 +
[Rakendusserverite alused_hindeid_kokku] * -1.67 +
[Riigikeel_hindeid_kokku] * -2.23 +
[Serveripoolsed tehnoloogiad veebirakenduste loomiseks_mittearvestatud] * -3.84 +
[Veebirakenduste alused_mittearvestatud] * -1.31 +
[Vene keel ja kirjandus_arvestatud] * 1.47 +
[Versioonihalduse alused_keskmine_hinne] * 0.71 +
[journalEntryLessons] * 1.27 +
[withReasonAbsences] * 0.76 +
[Ühiskonnaõpetus_keskmine_hinne] * -1.09
```

Joonis 3.5. TA SL mudel

Positiivse mõjuga atribuudid (klass "Lõpetanud"):

Agiilne tarkvaraarenduse keskmine hinne: (+4.67)

Programmeerimine hinnete kokku: (+2.69)

IT halduse standardid arvestatud: (+1.79)

Negatiivse mõjuga atribuudid (klass "Lõpetanud"):

HTML CSS mittearvestatud: (-1.83)

Lihtsamate rakenduste loomine mittearvestatud: (-1.79)

IT ja TA õppekavade mudelite analüüs toob esile erinevusi, mis tulenevad nende erinevast fookusest ja õppeainete ülesehitusest. IT õppekava mudel rõhutas sügavalt tehniliste oskuste ja teadmiste tähtsust. Positiivse mõjuga tegurid, nagu füüsika keskmine hinne ja IT halduse standardid hinnete kokku, näitavad, et tugevad teadmised keerukates tehnilistes ainetes on lõpetamise võtmetegurid. Samas oli negatiivse mõjuga atribuute, nagu Digitaaltehnika mittearvestatud ja SQL mittearvestatud, mis näitab, et nende aine mitte läbimine suurendab katkestamise riski märkimisväärselt.

TA õppekava mudel keskendus laiemale oskuste spektrile. Positiivse mõjuga olid sellised atribuudid nagu „Agiilne tarkvaraarenduse keskmine hinne" ja „Programmeerimise hinnete kokku". Kuid negatiivse mõjuga atribuudid olid, näiteks „HTML CSS mittearvestatud" ja „Lihtsamate rakenduste loomine mittearvestatud".

ARUTELU

Kaalutud keskmise F-skoori alusel oli väljalangevuse ennustamisel kõige edukam Naive Bayes algoritm. IT andmestikus saavutas NB kõrgeima kaalutud keskmise F-skoori 0.906, mis näitab, et mudel suudab antud andmestikus kõige täpsemalt eristada katkestajaid lõpetajatest.

TA andmestikus olid tulemused NB ja SMO tulemued väga sarnased. NB F-skoor oli 0.878, samas kui SMO saavutas F-skoori 0.877. Mõlemad mudelid suudavad TA andmestikus pakkuda usaldusväärseid tulemusi.

IT ja TA kombineeritud andmestikus jäi NB jällegi parimaks, saavutades F-skoori 0.889. Kuid väga lähedal oli ka SMO F-skooriga 0.880, mis näitab, et ka SMO suudab selles andmestikus väljalangevust hästi ennustada.

Otsustuspuu analüüsi käigus selgus, et aine "IT-õigus" läbimata jätmine mängib otsustavat rolli, suurendades katkestamise tõenäosust. Sama kehtib ka "Füüsika" aine kohta, kus madal sooritus ja puudulik arvestamine suunavad õpilasi otsustuspuu katkestamise harusse. Samuti ilmnes, et aine "Serveripoolsed tehnoloogiad veebirakenduste loomiseks" mõjutab olulisel määral tulemusi – vähene aktiivsus antud aines suurendab katkestamise riski.

Kui õpilastel õnnestub eelnimetatud kriitilised ained läbida, saavad otsustavaks teguriteks "Projektijuhtimine" ja "Agiilne tarkvaraarendus", kus madalad tulemused või tundides mitte osalemine soodustavad samuti väljalangemist. Samas positiivne sooritus eelnimetatud ainetes suurendab lõpetamise tõenäosust. Tähelepanu väärib ka *journalEntryLessons* näitaja, mis viitab, et väiksem arv päevikusse kantud tulemusi on seotud katkestamisega ja suurem arv lõpetamisega.

Simple Logistic mudelis selgus, et IT õppekava lõpetamiseks mängivad olulist rolli õppeained "Füüsika", "IT halduse standardid" ja "Infoturve" – head õpitulemused nendes ainetes tõstavad lõpetamise tõenäosust. TA õppekava lõpetamiseks mängivad olulist rolli õppeained "Agiilne tarkvaraarendus", "Programmeerimine" ja "IT halduse standardid arvestatud" - head õpitulemused nendes ainetes tõstavad lõpetamise tõenäosust.

Masinõppe algoritmide tulemuste võrdlusest selgus, et *Sequential Minimal Optimization (SMO)* ja *Naive Bayes (NB)* olid kõige täpsemad mudelid, mis pakkus usaldusväärsemaid tulemusi nii lõpetanute kui katkestanute klassis.

Analüüsis mängib olulist rolli ka õppekava tüüp. Tulemustest selgus, et 3-aastase tarkvaraarenduse õppekava õpilased lõpetavad suurema tõenäosusega edukalt, samas kui 4-aastase õppekava õpilastel on katkestamise risk kõrgem. See tulemus võib olla seotud õppekoormuse ja ajapikendusega, mis pikema õppekava puhul võib vähendada motivatsiooni.

IT- ja TA-erialade ühendatud analüüs tõi esile, et kahe õppekava kombineerimine vähendab veidi masinõppe algoritmide täpsust, mis viitab erialade spetsiifilistele erinevustele.

KOKKUVÕTE

Antud lõputöö eesmärk on välja töötada ennustusmudelid, mis võimaldavad prognoosida õppivate tudengite väljalangemise tõenäosust tulevikus.

Tulemustest selgus, et Naive Bayes (NB) on kõige edukam masinõppe algoritm väljalangevuse ennustamisel, saavutades parimaid F-skoore IT, TA ja kombineeritud andmestikes. Otsustuspuu analüüs tõi esile, et kriitilised ained, nagu "IT-õigus", "Füüsika" ja "Serveripoolsed tehnoloogiad veebirakenduste loomiseks", suurendavad katkestamise tõenäosust, kui nendes aineid ei sooritata edukalt. Samuti mõjutavad tulemusi ained, nagu "Projektijuhtimine" ja "Agiilne tarkvaraarendus", kus kehv hinne viib suurema väljalangemisriskini.

Lisaks näitas analüüs, et 3-aastase õppekava õpilased lõpetavad suurema tõenäosusega edukalt, samas kui 4-aastase õppekava puhul on katkestamise risk kõrgem. IT- ja TA-erialade kombineerimisel vähenes algoritmide täpsus, mis viitab nende õppekavade spetsiifilistele erinevustele.

Rakenduskõrghariduse lõputöö täitis minu akadeemilisi ja isiklikke eesmärke, pakkudes häid teoreetilisi ja praktilisi teadmisi.

SUMMARY

The objective of this thesis is to develop predictive models to forecast the likelihood of current students dropping out in the future. The study is based on data from two curricula at Tallinn Polytechnic School: IT Systems Specialist and Software Development. The data, collected from academic information systems and the Estonian Education Information System (EHIS), included academic and demographic indicators supplemented with journal entries.

For data preprocessing, JSON and Excel formats were flattened, and predictive models were trained using the Weka platform with algorithms such as Naive Bayes, Random Forest, Simple Logistic, Sequential Minimal Optimization (SMO), and decision trees (J48). Model evaluation utilized 10-fold cross-validation, with results measured through F1 scores and confusion matrices.

Naive Bayes was the most accurate algorithm, achieving a weighted average F1 score of 0.906 in the IT dataset, 0.878 in the Software Development dataset, and 0.889 in the combined dataset. Analysis with Simple Logistic identified key subjects that influence graduation or dropout, highlighting the importance of strong academic performance in core areas while poor results increased dropout risk.

KASUTATUD KIRJANDUS

[1] R. H. C. I. S. A. Nicolae-Bodan Sara, „High-School Dropout Prediction Usin Machine Learning: A Danish Large-scale Study," [Võrgumaterjal]. Available: https://books.google.ee/books?hl=en&lr=&id=USGLCgAAQBAJ&oi=fnd&pg=PA319&dq=student+dropout+prediction+machine+learning&ots=FufcgoIUUP&sig=9-HUBlg1ykrWEBvgHzkVWAZB5ro&redir_esc=y#v=onepage&q=student%20dropout%20prediction%20machine%20learning&f=false. [Kasutatud 2024].

[2] R. M. Isiaka, R. S. Babatunde, F. J. Ajao ja S. O. Abdusalam, 28 February 2019. [Võrgumaterjal]. Available: https://www.proquest.com/openview/f52ff16f39d0891d83fded6aa7773e8b/1?pq-origsite=gscholar&cbl=2028729. [Kasutatud 2024].

[3] M. Natalja, A. Pentel ja O. Dunajeva, „Predicting First-Year Computer Science Students Drop-Out with Machine Learning Methods: A Case Study," 14 Märts 2021. [Võrgumaterjal]. Available: https://link.springer.com/chapter/10.1007/978-3-030-68201-9_70. [Kasutatud 2024].

[4] N. Maksimova, A. Pentel ja O. Dunajeva, „Computer Science Students Early Drop-Out Prediction Using Machine Learning: A Case Study," 13 september 2022. [Võrgumaterjal]. Available: https://link.springer.com/chapter/10.1007/978-3-031-04286-7_25. [Kasutatud 2024].

[5] L.-L. Kaiva ja A. Pentel, „Predicting Students' State Examination Results based on Previous Grades and Demographics," 2020. [Võrgumaterjal]. Available: https://ieeexplore.ieee.org/document/9284401. [Kasutatud 2024].

[6] Waikato, „Arff stable - Weka Wiki," [Võrgumaterjal]. Available: https://waikato.github.io/weka-wiki/formats_and_processing/arff_stable/. [Kasutatud 2024].

[7] Javatpoint, „Machine Learning Tutorial," Javatpoint, [Võrgumaterjal]. Available: https://www.javatpoint.com/machine-learning. [Kasutatud 2024].

[8] R. Sirel, 10 06 2015. [Võrgumaterjal]. Available: https://slideplayer.com/slide/14581461/. [Kasutatud 2024].

[9] E. Frank, M. A. Hall ja I. H. Witten, 2016. [Võrgumaterjal]. Available: https://ml.cms.waikato.ac.nz/weka/Witten_et_al_2016_appendix.pdf. [Kasutatud 2024].

[10] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, 1993.

[11] I. H. Witten, E. Frank ja M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 2011: The Morgan Kaufmann.

[12] Geeks for Geeks, 11 Detsember 2024. [Võrgumaterjal]. Available: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/. [Kasutatud 2024].

[13] Geeks for Geeks, „Gaussian Naive Bayes," 13 november 2023. [Võrgumaterjal]. Available: https://www.geeksforgeeks.org/gaussian-naive-bayes/. [Kasutatud 2024].

[14] N. Donges, „The Logistic Regression Algorithm," 25 august 2023. [Võrgumaterjal]. Available: https://resources.experfy.com/bigdata-cloud/the-logistic-regression-algorithm/. [Kasutatud 2024].

[15] C. D. P. Jr., „SVM by Sequential Minimal Optimization (SMO)," [Võrgumaterjal]. Available: https://pages.cs.wisc.edu/~dpage/cs760/SMOlecture.pdf. [Kasutatud 2024].

[16] Geeks for Geeks, „Cross Validation in Machine Learning," 7 august 2024. [Võrgumaterjal]. Available: https://www.geeksforgeeks.org/cross-validation-machine-learning/. [Kasutatud 2024].

[17] Geeks for Geeks, „F1 Score in Machine Learning," Geeks for Geeks, 17 oktoober 2024. [Võrgumaterjal]. Available: https://www.geeksforgeeks.org/f1-score-in-machine-learning/.