TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Mark Genrich Geller 211679IABM

SARNASTE HOONETE LEIDMINE LOD2 PÕHJAL KASUTADES MASINÕPPE ALGORITME

Magistritöö

Juhendaja: Innar Liiv
PhD

Konsultant: Ergo Pikas
PhD

Tallinn 2023

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Mark Genrich Geller 211679IABM

FINDING SIMILAR BUILDINGS BASED ON LOD2 USING MACHINE LEARNING ALGORITHMS

Master's Thesis

Supervisor: Innar Liiv
PhD

Consultant: Ergo Pikas
PhD

Tallinn 2023

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: Mark Genrich Geller

10.05.2023

Annotatsioon

Sarnaste hoonete leidmine LOD2 põhjal kasutades masinõppe algoritme

Käesoleva magistritöö teemaks on LOD2 põhjal sarnaste hoonete leidmise masinõppe algoritmide väljatöötamine ja testimine. Töö on loodud RESTO uurimisprojekti (toetab Haridus- ja Teadusministeerium ja Euroopa Regionaalarengu Fond; grant 2014-2020.4.01.20-0289) raames ning selle eesmärgiks on viia läbi mitmeid eksperimente ning luua masinõppel põhinevate meetoditega mudelid, mis suudaksid vastavalt hoonete geomeetrilistele andmetele leida kõik omavahel sarnased hooned üle Eesti. Muutes seeläbi massrenoveerimise ressursi planeerimise oluliselt efektiivsemaks.

Töö algul oli püstitatud kaks uurimisküsimust:

- Kas LOD2 taseme põhjal on võimalik leida sarnased hooned ning nad omavahel grupeerida?
- Kui palju võib erineda inimsilma hinnang masinõppe mudeli hinnangust?

Magistrtöö raames viidi läbi mitmeid eksperimente kolmel eri andmestikul kasutades kaheksat erinevat masinõppe mudelit. Töö käigus uuriti mudelite sobivust vastava andmestikuga, erinevate sisendparameeetrite mõju algoritmidele ning võrreldi kõiki kaheksat masinõppe algoritmi. Töö tulemusena treeniti algoritmid, mis suudavad võrdlemisi täpselt jaotada hooned geomeetrilise sarnasuse järgi. Parimaks mudeliks osutus hierarhilise klasterdamise mudel, mis suutis paigutada kõik sarnased hooned ühte klastrisse. Sisendparameetritest osutusid parimateks: ehitusaluse pinna välisnurkade arv, ehitusaluse pinna perimeter, fassaadi pindala, katuse pindala ning pinnasel põranda pindala. Antud mudeli tulemustel leiti ka vastused magistritöö alguses püstitatud küsimustele. LOD2 taseme põhjal on võimalik saada piisavalt palju detailseid arvutusi ning andmeid hoone kohta, et jaotada neid sarnasuse järgi. Lisaks sellele leiti ka, et masinõppe mudeli hinnang ei ole kuigi palju erinev inimsilma hinnangust ning on võimeline tuvastama omavahel visuaalselt sarnased hooned.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 72 leheküljel, 5 peatükki, 28 joonist, 16 tabelit.

Abstract

The topic of this master's thesis is the development and testing of machine learning algorithms for finding similar buildings based on LOD2. The work was created within the RESTO research project (supported by the Estonian Ministry of Research and Education and European Regional Development Fund; grant 2014-2020.4.01.20-0289), and its aim is to conduct several experiments and create models using machine learning methods that can find all similar buildings across Estonia based on their geometric data, making the resource planning for mass renovation much more efficient.

At the beginning of the work, two research questions were posed:

- Is it possible to find similar buildings and group them based on LOD2 level?
- How much can the estimation of the machine learning model differ from human estimation?

To solve the problem, several experiments were conducted on three different datasets using eight different machine learning models. During the work, the suitability of the models with the corresponding datasets was examined, the effect of different input parameters on the algorithms was studied, and results between all eight machine learning algorithms were compared.

As a result of the work, algorithms were trained that can accurately classify buildings based on their geometric similarity. The best model turned out to be the hierarchical clustering model, which was able to place all similar buildings in one cluster. The best input parameters were found to be the external corner count of the construction site surface, the perimeter of the construction site surface, the facade area, the roof area, and the floor area on the ground. The results of this model also answered the research questions posed at the beginning of the master's thesis. Based on LOD2, it is possible to obtain enough detailed calculations and data about a building to classify them according to similarity. Additionally, it was found that the estimation of the machine learning model is not much different from human estimation and is capable of identifying visually similar buildings.

The thesis is written in Estonian and is 72 pages long, including 5 chapters, 28 figures and 16 tables.

Lühendite ja mõistete sõnastik

API	Rakendustarkvara liides (*Application Programming Interface*)
EHR	Ehitisregister, www.ehr.ee
LOD	Detailsuse tase (*Level of Detail*)
RESTO	Renoveerimisstrateegia tööriist (*Renovation Strategy Tool*)
JSON	Andmevahetusvorming, mis põhineb Javascript programmeerimiskeele alamhulgal (*JavaScript Object Notation*)
CityGML	XML keelel põhinev treabemudel (*City Geography Markup Language*)
XML	Märgistuskeel, mille eesmärgiks on struktueeritud info jagamine (*Extensible Markup Language*)
KML	CityGML-iga sarnane teabemudel (*Keyhole Markup Language*)
OGC	CityGML mudeli loonud organisatsioon (*Open Geospatial Consortium*)
ISO	Rahvusvaheline standardiseerimise organisatsioon (*International Organization for Standardization*)
2D	Kahemõõtmeline ruum, koordinaatide määramiseks vaja kahte koordinaati
3D	Kolmemõõtmeline ruum, koordinaatide määramiseks vaja kolme koordinaati
GIS	Geograafiline infosüsteem ()
BFR	Kui suur osa hõlmab vaadeldavast pinnast hoone (*Building footprint ratio*)
FAR	Põranda ja pindala suhe (*Floor area ratio*)
ML	Masinõppe (*Machine learnin*g)
AI	Tehisintellekt (*Artificial Intelligence*)
kNN	: k-lähima naabri algoritm (*k-nearest neighbors algorithm*)
SOM	Iseorganiseeruv kaart(*Self-Organizing Map*)
SVM	Tugivektor-masinad(*Support Vector Machines*)
LASSO	*Least Absolute Shrinkage and Selection Operator*
LARS	*Least-angle regression*

CART	Klassifikatsiooni ja regressiooni puu (*Classification And Regression Tree*)
MLP	Mitmekihilised Percptronid (*Multilayer Perceptrons*)
CNN	Konvolutsiooniline närvivõrk (*Convolutional Neural Network*)
RNN	Rekurrentne närvivõrk (*Recurrent Neural Network*)
PCA	Peakomponentide analüüs (*Principal component analysis*)
GMM	Gaussi segu mudelid (*Gaussian Mixture Models*)

Sisukord

Jooniste loetelu

Tabelite loetelu

1. Sissejuhatus

Käesoleva magistritöö teemaks on masinõppe meetoditel põhinevate sarnaste LOD2 tüüpi hoonete mudelite tuvastamise algoritmide välja töötamine ja testimine. Töö on tehtud RESTO projekti raames, et aidata tuvastada sarnaseid majamudeleid üle Eesti. Töö tegemisel on peamiselt kasutatud nelja erinevat andmestikku - 417 Kredexi poolt toetatud hoonete andmestikku, eelmise aasta magistritöö [1] ning RESTO projekti koostöö raames valminud tüpoloogia andmestikku, tudengite poolt koostatud sarnaste hoonete andmestikku ning ka nõukoguaegsete paneelmajade andmestikku. Lisaks sellele kasutatud ka RESTO projekti raames loodud arvutuste API *endpoint*'i.

1.1 Taust ja probleem

2019. aasta lõpus esitas Euroopa Liit Euroopa rohelist kokkulepet (inglise keeles - *European Union's Green Deal*), mille peamiseks eesmärgiks on vähendada kasvhuhoonegaaside heitkoguseid 55% võrra võrreldes 1990. aasta andmetega ning aastaks 2050 saavutada kliimaneutraalne hoonefond [2]. Vastavalt varasemale uuringule moodustab hoonete energiatarbimine ligi 40% kogu Euroopa energiatarbimisest ning tekitab ligikaudu 39% CO2 heitkogusest, mis seab ehitussektori pea, et suurimaks energiatarbijaks Euroopas [2, 1]. Kuigi tänapäeval ehitatud uued hooned tarbivad poole vähem kui 1980. aastal ehitatud hooned ning peavad alates 2021. aastast olema liginullenergiahooned (NZEB), siis valdav osa praegusest hoonefondist, mis ehitati ilma oluliste energiatõhususe nõueteta, kujutab endast peamist probleemi ning suurt väljakutset ja võimalust kogu Euroopas [2].

Paari aasta tagune TalTechi ehitus- ja arhitektuuriosakonna uuring tõi välja, et Eestis tähendaks see ligi 60% hoonete renoveerimist (ülejäänud on kas juba heal energiatõhususe tasemel või lammutatakse aastaks 2050). Lisaks sellele uuriti välja, et kokku läheks eesmärgini jõudmine maksma ligi 25 miljardit eurot. Samuti oli leitud, et tänase seisuga renoveeritakse Eestis aastas ligikaudu 1% hooneid, kuid eesmärgi saavutamiseks 2050. aastaks peaks renoveerimismaht suurenema vähemalt viis korda [3].

Tänapäeval on hoonete renoveerimisel peamiseks vastutajaks eelkõige hoone omanik. Probleemiks on aga see, et tihtipeale napib enamusel rahalisi vahendeid ja nii teadmisi kui ka tahtmist renoveerimistöid ette võtta, vaatamata sellele, et ka riik on proovinud Kredexi kaudu hoonete renoveerimist läbi aastate toetada. Juhtudel, kus isegi võetakse renoveerimine ette, sageli piirdutakse paraku vaid avariiremondiga, kuid hoone täielik

renoveerimine jääb tegemata. Euroopa Liidu renoveerimisstrateegia näeb ette, et sellistes olukordades peaks kohalikud omavalitsused saama kogu renoveerimisprotsessi eestvedajateks. Eestis on praegu kohalikul omavalitsusel olnud pigem aga kooskõlastaja roll. Samas, kui omavalitsusel tekiks arusaam, kuidas hooneid ja isegi terveid linnaosasid terviklikult ja ratsionaalselt renoveerida energiasäästlikuks ja kliimaneutraalseks, saaksid nad ka hooneomanikele otsuste tegemisel ja rahastamise korraldamisel tuge pakkuda. Selleks, et omavalitsused saaksid sellist teenust osutada, peaks neil olema selge ülevaade konkreetsete hoonete ja linnakvartalite renoveerimise kõige otstarbekamatest viisidest. Veelgi enam, selleks, et omavalitsused saaksid jagada kogu seda infot ka hooneomanikele, tuleks koostada terviklikud linnaosade renoveerimisstrateegiad. Selline strateegia annaks teavet antud piirkonna energiatõhususe ja kliimaneutraalsuse, iga hoone jaoks vajalike investeeringute suuruse ja ka selle kohta, milliseid renoveerimismeetmeid oleks kõige parem kasutada [3].

Tänaseks päevaks on juba mitmeid Euroopa riike, nagu näiteks Soome, Itaalia kui ka Euroopa Liidu väliseid riike nagu Bosnia ja Hertsogoviina, väljatöötamas renoveerimisstrateegiat, mis aitab neil püstitatud eesmärkideni jõuda [4, 2]. Rohelise kokkuleppe suunas liigub ka Eesti. TalTechi ehituse ja arhitektuuri instituudi ning tarkvarateaduse instituut koostöös Võru linna ning Targa linna tippkeskusega FinEst on välja töötamisel renoveerimisstrateegia tööriist (RESTO). Tööriista eesmärk on aidata kohalikel omavalitsustel toetada hooneomanikke terviklike renoveerimisprojektide koostamisel ja elluviimisel, hinnata hoonete ühisrenoveerimisel tehtavate investeeringute mahtu ja mõjusid ning leida hoonete omadustest lähtuvalt parimaid tehnilisi lahendusi [1, 3].

Probleem seisneb aga selles, et hetkel ei ole nimetatud uurimisrühmadel ega ka eraisikust või ettevõtetest renoveerijatel andmebaasi ega ka muud süsteemi, mille järgi oleks võimalik kõik hooned Eestis sarnasuse järgi ära jaotada või leida automaatselt juba sarnaseid teostatud töid. Traditsiooniliselt jaotatakse ekspertide poolt sarnased hooned erinevatesse tüpoloogiatesse, kuid kahe hoone numbrilise sarnasuse leidmine automaatselt ei ole olnud võimalik. Sarnasuse põhjal grupeerimine võimaldaks paremat ajalise ja ka rahalise ressursi planeerimist. See aitaks ka efektiivsemalt viia läbi massrenoveerimise, sest siis oleks paremini võimalik hinnata, kui mitmele hoonele kui palju materjali ning ehitustööjõudu on vastavalt hoone tüübile vaja.

1.2 Motivatsioon

Tänaseks on majade gruppidesse jagamisel ning ühiste omaduste leidmisel tehtud väga palju manuaalset tööd. Näiteks 2022/2023 aasta lõputöös loodi tüpoloogia andmestik, kus vastavalt majade erinevatele mõõtmetele ja omadustele defineeriti kindlad hoonetüübid, mille järgi hooned grupeeriti. Andmestik loodi peamiselt 417 näidismaja põhjal, kus kõik

vajalikud arvutused ja omadused olid kätte saadud. Seda arvesse võttes oli ka üheks peamiseks motivatsiooniks luua algoritm, mis suudaks eelmainitud protsessi automatiseerida ning ühtlasi tulevikus skaleerida kogu riigi tasemele. See võimaldaks nii RESTO projekti raames kui ka hilisemalt eraisikust renoveerijatel palju paremini planeerida rahalist ning ajalist ressurssi. Ühtlasi aitab see ka Eestil jõuda Euroopa Liidu poolt seatud eesmärkideni palju kiiremini ning efektiivsemalt.

1.3 Magistritöö küsimused ja eesmärk

Käesoleva töö eesmärgiks oli kasutades erinevaid masinõppe meetodeid viia läbi eksperimente neljal eri andmestikul ning luua nende põhjal sobivaim algoritm. Algoritmidel põhinev lahendus peaks suutma vastavalt LOD2 maja omadustele ja arvutustele grupeerida kõik omavahel sarnased majad samasse segmenti ning tüpoloogia ja kredexi hoonete andmestike puhul, peaks algoritm suutma kõik EHR'i omadustega majad grupeerida eeldefineeritud tüüpidesse.

Käesoleva magistritöö alguses olid püstititatud järgmised küsimused:

- Kas LOD2 taseme põhjal on võimalik leida geomeetriliselt sarnased hooned ning nad omavahel grupeerida?
- Kui palju võib erineda inimsilma hinnang masinõppe mudeli hinnangust?

Lahenduse loomiseks on eelkõige vaja defineerida, milliste andmetega on tegemist ning millist masinõppe mudelit oleks kõige mõistlikum antud andmestiku puhul kasutada, kas klasterdamist või klassifitseerimist. Järgmine samm oleks andmetest sisendandmete valimine ehk leida, millised parameetrid mängivad kõige suuremat rolli sarnasuse defineerimisel. Juhul, kui tegemist on näiteks klassifikatsiooniga, tuleks luua nii treening- kui ka testandmed, millega mudeleid vastavalt treenida ja testida ning lõpuks saadud tulemusid omavahel võrrelda. Seejärel on vaja katsetada andmestikul läbi mitmeid erinevaid mudeleid ning vaadata, milline on kõige täpsem, näiteks klassifikatsiooni puhul, milline kahest on täpsem, otsustuspuu või juhusliku metsa meetod. Lõpptulemusena peaks olema võimalik defineerida, milline masinõppe mudel, milliste sisendandmetega peaks sobima millisele andmestikule kõige paremini.

1.4 Metoodika ja andmed

Magistrtöö tulemusteni jõudmiseks ning masinõppe mudeli treenimiseks ja testimiseks kasutatakse peamiselt nelja erinevat andmestikku. Nii 417 Kredexi poolt toetatud hoone valim

kui ka selle põhjal loodud tüpoloogia andmestik pärinevad mõlemad RESTO projektist ning on ka käesolevas lõputöös kasutatud koos. Kolmas andmestik, mida kasutatakse, on tudengite poolt loodud sarnaste hoonete nimekiri, ning neljas on nõukoguaegsete paneelmajade andmebaas. Kuna viimase kahe andmestiku puhul on mõningad olulised omadused ja arvutused puudu, kasutatakse ka RESTO API *endpoint*-i, mis tagastab vastavalt hoone EHR-i numbrile kõik vajalikud LOD2 põhjal saadavad lisaandmed. Töös on kokku kasutatud kaheksat erinevat masinõppe algoritmi: K-keskmine (*K-means*), hierarhiline klasterdamine, Gaussi segu mudelid (*Gaussian Mixture Models*), logistiline regressioon, otsustuspuu, juhuslik mets, tugivektor-masinad (SVM - *Support vector machines*), närvivõrgud.

1.5 Uudsus ning töö äriline kasu

Käesoleva magistritöö uudsus seisneb eelkõige selles, et Eesti on üks väheseid riike, kes on võtnud aktiivselt kasutusele LOD2-l põhinevaid maja mudeleid. 3D geoinformatsiooni uurimisrühm [5] on kaardistanud välja ligikaudu 40 linna üle maailma, kes on võtnud kasutusele ükskõik millise neljast LOD tasemest. Nendest ligi 30 on LOD2 põhjal loodud ning Eesti on üks ainukesi, kes on võtnud LOD2 üleriigiliselt kasutusse. Seetõttu pole ka maailmas siiani väga midagi sellist tehtud, eriti mis hõlmaks just geomeetrilise sarnasuse leidmist majade vahel.

Suurimaks kasu saajaks on eelkõige RESTO projekt, mille eesmärk on toetada massrenoveerimise strateegiate loomist ning täita seeläbi Euroopa Liidu püstitatud Rohelise kokkuleppe eesmärk 2050. aastaks. Lisaks sellele on näha ka tõusvat trendi ning vajadust LOD mudelite järgi, sest aina rohkem riike on hakanud minema üle LOD mudelite peale, eriti just LOD2-le. Näiteks 2020. aasta lõpuks sai Visicom valmis projektiga, kus oli loodud 3D LOD2-l põhinevad majamudelid tervele Lähis-Ida piirkonnale [6]. Ka 3D geoinformatsiooni andmetabelis on näha, et viimase paari aastaga on uuema 3D tasemega riiklikul tasemel ühinenud nii Jaapan kui ka Holland [5]. Seetõttu võib antud tööst olla kasu ka kõikidele riikidele, kes äsja LOD2-le üle läinud või alles plaanivad minna, eriti näiteks Euroopa Liidu liikmesriigid, kellel kõigil seisab ees massrenoveerimise strateegia loomine.

Tulevikus võib äriline kasu väljenduda ka muul kujul, näiteks juhul, kui ülemaailmselt hakatakse aktiivsemalt ülevõtma LOD3 mudelit. Sellisel juhul on olemas põhi, mille pealt juba loodud süsteemi edasi arendada. Töö ei tule kasuks mitte ainult massrenoveerimisel, vaid võib aidata ka eraisikust renoveerijal, kes suudaks ressurssi ning materjali kulu paremini ette planeerida. Samuti on sellest kasu ka linna ning teede planeerimisel, sest sarnane lahendus aitaks paremini aru saada piirkonna või linna omadustest ning planeerida ehitusi vastavalt. Veelgi enam, tööd saaksid kasutada ka kindlustusettevõtted selleks, et

paremini hinnata vara riski ning võimaliku kahju suurust.

1.6 Ülevaade tööst

Sissejuhatuses käsitletakse lähemalt üldist tausta ning probleemi, mille lahendamisele antud lõputöö suunatud on. Kuna lõputöö on loodud RESTO projekti raames, kirjeldatakse sissejuhatuses pikemalt renoveerimisstarteegiat, miks seda vaja on, kuidas see aitab ning millega RESTO projekt tegeleb. Samuti kirjeldatakse motivatsiooni ja töö üldist eesmärki.

Esimene osa on fokuserunud rohkem töö teoreetilistele aspektidele ning on jagatud kolmeks alapeatükiks. Esimeses alapeatükis kirjeldatakse lähemalt LOD tüüpe ja nende olemust. Samuti antakse ka ülevaade varasemalt tehtud töödest 2D ja 3D kujunide sarnasuste leidmise teemadel. Viimases alapeatükis kirjeldatakse lähemalt masinõpet ning kirjutatakse lahti, milliseid masinõppe algortime ning mis põhjusel antud töö raames rakendati.

Teises peatükis kirjeldatakse detailsemalt eksperimentide disaini alates andmete kogumisest ning normaliseermisest kuni algortimide kasutamiseni ning tulemuste saamiseni.

Viimases peatükis vaadeldakse detailsemalt eksperimentide käigus saadud tulemusi ning magistritöö lõpuks võetakse kõige olulisem kokku.

2. Teoreetiline taust

Käesolev peatükk annab laiema ülevaate hoonete 3D mudelitest ja nende olulisusest ning lõpuks tuuakse näiteid ka võimalikkest kasutusvaldkondadest. Lisaks sellele analüüsitakse ka varasemalt kirjutatud töid nii sarnaste hoonete leidmisel kui ka näiteks sarnasuse leidmisel 2D ja 3D kujundite puhul.

2.1 Maja 3D mudelid ja nende *Level of Detail* (LOD)

Juba mõned aastad on 3D maja- ja üldiselt linnamudeleid eelistatud visualiseerimisel tavalistele 2D kaartidele, sest nad pakuvad palju rohkem võimalusi ning realistlikumaid kogemusi [7, 8]. Tänapäeval 3D mudelite olulisus kasvanud ning lisaks visualiseerimisele kasutatakse neid ka mitmel muul otstarbel ja mitmes erinevas valdkonnas [7]. Näiteks on neid hakatud rohkem kasutama arhitektuuri ning liikluse planeerimisel, linnaplaneerimisel kui ka keskkonna- ja energia planeerimisel. Üldjuhul oleneb nende rakendusala suuresti väga palju 3D mudeli detailsuse tasemest (LOD) [9]. Selleks, et saada 3D mudelitest maksimaalne kasu ning neid täies mahus ära kasutada, on vaja üldtunnustatud andmemudelit modelleeritud tunnuste geomeetria ja semantika salvestamiseks ja vahetamiseks [10].

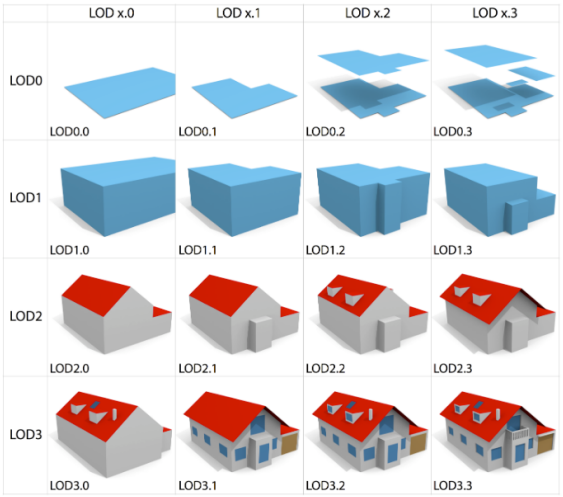
2.1.1 CityGML

CityGML (inglise keeles - *The City Geography Markup Language*) on kõigile kättesaadava avalik XML keelel põhinev teabemudel, mille eesmärk on eelnevalt välja toodud 3D linnamudelite nii geomeetriliste kui ka semantiliste andmete esitamine, vahetamine ning salvestamine. Andmemudel on loodud *Open Geospatial Consortium* (OGC) organisatsiooni poolt, mida ühtlasi peetakse ka rahvusvahelise standardiseerimise organisatsiooni (ISO) järel üheks kõige olulisemaks ettevõteks antud valdkonnas [10]. Erinevalt teisest sellele sarnanevast mudelist, näiteks *Keyhole markup language* (KML), mida kasutatakse Google Earth-is, Collada-s ning ka X3D-s, eristab CityGML reaalmaailma omadusi pakkudes kokku 98 klassi koos 372 täpselt määratletud atribuudiga. Seetõttu on võimalik CityGML-i lisaks tavalisele visualiseerimisele kasutada ka muudel otstarbetel, nagu näiteks energiahinnagute määramiseks või erinevate simulatsioonide läbiviimiseks [10]. Selle teeb võimalikuks CityGML-i kõige olulisem moodul, ehituse moodul. Läbi selle saab andmemudelil kujutada hoonete, hooneosade ja paigaldiste temaatilisi ja ruumilisi aspekte viiel detailsuse tasemel (LOD), mida kirjeldab autor ka järgnevates alapeatükkides [9].

2.1.2 Detailsuse tasemed

CityGML andmemudelis on üldjuhul defineeritud viis põhilist detailsuse taset (vt Joonis 1), mis kannavad lühendeid: LOD0, LOD1, LOD2, LOD3 ning LOD4. Sageli on välja toodud üksnes neli taset viiest, näiteks LOD4, jääb välja toomata. Vahel jaotatakse need viis või neli põhilist taset veel omakorda alatasemeteks (vt Joonis 2), kuid üldjuhul arvestatakse siiski viie põhitasemega [11].

Joonis 1. Viis põhilist LOD taset [12].



Joonis 2. LOD tasemed koos nende alamtasemetega. Puudub ka interjööri tase [12].

Null detailsuse tase - LOD0

Esimese detailsuse tase, ehk tegelikult null detailsuse tase, on ainus tase, mis on 2D, mitte 3D tase. LOD0 puhul on hoone kujutatud kas selle põranda- või katusekontuuri järgi. Vahel on antud detailsuse taseme puhul tegemist ka 2.5D kujutisega, sest kujutatud on nii põranda kui ka katusekontuurid. Üldjuhul on LOD0 tase sarnane tavalise linnakaardi puhul kasutatava kujutisega. Seda kasutatakse näiteks kas lihtsa maaomandi visualiseerimise puhul või siis näiteks ka kauguse või tiheduse arvutamisel tulekahjude ettevaatusabinõude jaoks [10, 13].

Esimene detailsuse tase - LOD1

LOD1 tasemest alates on tegemist juba 3D mudelitega. Sellisel tasemel on hoone üldjuhul kujutatud plokkmudelina ning visualiseerib minimaalselt maja lisaomadusi. Põhimõtteliselt luuakse LOD0 taseme põranda- ja katusekontuuride vahele tahud, mis teevad mudeli

kolmemõõtmeliseks. LOD1 tasemel ei ole näiteks võimalik eristada hooneid katuste järgi, sest kõik katused on kujutatud ühtlase pinnana. Samuti ei erista antud tase igasuguseid väiksemaid detaile nagu näiteks aknad, korstnad jne [13]. Sarnaselt eelnevale tasemele on ka sellel mudelil lisaks visualiseerimisele mitmeid muid otstarbeid. Näiteks kasutatakse antud taset müra kaardistamise meetodite puhul või ka näiteks üleujutustasandite tegeliku mahu hindamiseks üleujutuste vältimiseks tulevikus. Vahel piisab LOD1 tasemest ka mobiilsidevõrkude modelleerimiseks, seni kuni pole vaja kasutada peegeldusomadusi [10].

Teine detailsuse tase - LOD2

Teine detailsuse tase on jällegi üks samm edasi eelnevast tasemest. LOD2 puhul on majad juba üksteisest eristatavad ning ei ole kujutatud samasuguste või väga sarnaste plokkmudelitena. Antud tasemel on defineeritud katus, samuti üksikutel juhtudel võivad olla defineeritud ka mitmesugused väiksemad detailid, nagu näiteks korstnad ja juurde ehitised [10, 13]. Eestis on loodud ka näiteks 3D maa-ameti kaart, kus kõik hooned on kujutatud LOD2 tasemel. Ka käesoleva magistrtöö eksperimendid on loodud LOD2 põhjal ning kasutades antud detailsuse taseme andmeid. Nagu ka eelpool mainitud, on magistrtöö eesmärgiks leida, kas LOD2 hooneid, mis on esimene 3D tase, kus majad on väga selgelt üksteisest eristatavad, on võimalik sarnaste omaduste järgi grupeerida. LOD2 hoone esitamise taset on võimalik kasutada ka näiteks päikeseenergia potentsiaali analüüsimiseks ning seinapindade kogupinna põhjal võib planeerida ehitise soojaisolatsiooni [10].

Kolmas detailsuse tase - LOD3

Lisades Teisele tasemele juurde aknad, uksed, rõdud ning väiksemaid detaile, on tulemuseks LOD3. Kolmas tase on hetkel välisstruktuuri järgi kõige detailsem mudel koos, LOD4-ga [13]. See on kõige ligilähedasem päris hoonele ning seetõttu suureneb ka antud taseme kasutusala veelgi. Näiteks tänu uste olemasolule on võimalik analüüsida erinevaid evakuatsioonistsenaariume ning politseioperatsioonide jaoks juurdepääsu tagamist hoonetele [10].

Neljas detailsuse tase - LOD4

Välisomaduste poolest ei erine LOD4 enam suuremal määral LOD3-st, mistõttu on nad mõlemad välimuse poolest kõige detailsemad hoone mudelid. Küll aga lisanduvad LOD4 puhul lisaks välistele ka sisemised omadused ning antud tasemel on kujutatud näiteks ka hoone interjööri. Tänu täiendavale detailsusele on võimalik veelgi ulatuslikumalt analüüsida hoone energiasäästlikust, -tarbimist ning ka erinevaid õhu- ja soojusvooluga seotud andmeid [10, 11, 13].

2.1.3 Kasutusalad

Käesolev alapeatükk keskendub põhjalikumalt paarile spetsiifilisemale valdkonnale, kus antud 3D mudeleid kasutatakse, ning kus tulevikus võib vaja minna sarnaste hoonete segmenteerimist, mille abil suudaksid ehitajad või renoveerijad paremini materjaliga arvestada.

Üheks põhiliseks valdkonnaks, eriti just mudelitel alates LOD2 tasemest, kus 3D maja mudeleid kasutatakse, on hoonete insolatsiooni ehk päikeselt saabuva otsese kiirgusvoo hindamine. Oluliseks on see lisaks päikesepaneelide paigaldusele ka seetõttu, et järjest enam on kohalikud omavalitsused hakanud elamute ehitamisel ja rekonstrueerimisel nõudma erinevate insolatsiooni nõuete täitmist [7, 14]. Päikesepaneelide puhul on võimalik näiteks juba teisest detailsuse tasemest alates hinnata, kas hoone on päikese käes sellises ulatuses, et katusele päikesepaneelide paigaldamine on tasuv või mitte. Seda on võimalik saavutada just seetõttu, et 3D maja mudel pakub üldjuhul vajalikku geomeetrilist teavet, nagu näiteks katuse kalle, orientatsioon ja ka pindala, mida on võimalik võtta arvesse näiteks ressursi planeerimisel [7].

Järgmine kasutusala on mõneti sarnane eelnevaga ning samuti toetab magistritöö uurimisprobleemi lahendamist osas, mis puudutab RESTO projekti ning Euroopa rohelist kokkulepet. Nimelt kasutatakse 3D mudeleid ka energianõudluse hinnangu tegemisel. Näiteks Saksamaal on teadlased kasutanud hoone mudeleid, et kombineerida hoonete mahu, korruste arvu, hoone tüübi ja muude näitajate andmeid selliselt, et on võimalik ennustada hoonete energiavajadust [7]. Energiavajaduse hindamine ja ennustamine on oluline näiteks energiatõhusa moderniseerimise kasulikkuse hindamisel ning materjalide planeerimisel. Eespool mainitud otstrabe puhul oleks masinõppe kasutamine eriti kasulik, sest see võimaldaks planeerida konkreetseid materjale, võttes arvesse kõiki linnas või koguni riigis asuvaid sarnaseid hooneid.

Lisaks eelpool väljatoodule täidavad 3D mudelid ka visualiseerimisega seonduvaid eesmärke. Näiteks võidakse 3D linna- või majamudeleid kasutada selliste analüüside tulemuste esitamise täiustamiseks, mis ei pruugi olla GIS-i (*Geographic information system*) või otseselt 3D hoonetega isegi seotud, nagu majandustegevused, tsunamianalüüsid ja tuuleparkide kavandamised [7].

2.1.4 Probleem ning seos lõputööga

3D hoone ja isegi linnamudelite kasutamine muutub üha olulisemaks, sest need pakuvad realistlikumaid kogemusi kui tavalised 2D kaardid. Sellele vaatamata ei liigu antud valdkonnas areng veel nii kiiresti kui oleks võimalik, ning esineb väiksemaid takistusi. Üheks põhilisemaks on see, et kuna nõudlus üksikasjaliku väljenduse järele aina kasvab, kasvavad ka 3D ehitusmudelite andmemahud. Vaatamata sellele, et arvutid on viimaste aastatega muutunud palju võimsamaks, jääb maksimaalse detailsusega mudelite kuvamine, eriti terve linna või riigi korraga kuvamine, liiga raskeks ning võib omakorda põhjustada halva kasutajakogemuse. Antud probleem oli ka üks nendest põhjustest, miks loodi erinevad LOD tasemed. Paljusid probleeme on võimalik lahendada ka kasutades vähemdetailsemaid mudeleid, kuid kõrgemate ning detailsemate tasemete puhul võib see olla keerulisem [8]. Sellel põhjusel on ka oluline luua algoritm, mis suudaks leida sarnaseid hooneid LOD2 taseme põhjal, sest tegu on esimese 3D mudeli tasemega, mille järgi on võimalik hooneid üksteisest selgelt eristada. Lisaks sellele on tegu ka palju vähem keerulisema ning kiiremini laetava mudeliga kui LOD3 ja LOD4, mistõttu ei kannata ka näiteks kasutajamugavus.

2.2 Kujude sarnasus

Sarnasusmõõtmised võimaldavad kvalifitseerida, kui sarnased on kaks või enam etteantud objekti. Seetõttu on seda valdkonda ka laialdaselt uuritud ning tänaseks kasutatakse neid mitmes kohas, alates tehisintellektist (AI) ja kujutiste otsimisest kuni protseduurilise sisu loomise ja tehisnägemiseni [15]. Täna on ka sarnasuse leidmisel loodud mitmeid alagruppe ning üks nendest on geomeetriline sarnasus, millega puutub käesolev magistrtöö tihedamalt kokku. Geomeetriline sarnasus mõõdab eelkõige geomeetriliste objektide sarnasust. Käesoleva peatüki eesmärgiks on anda ülevaade varasemalt tehtud töödest, mis käsitlevad laiemas mõistes sarnasuse leidmist, peamiselt 2D ja 3D kujude puhul.

2.2.1 2D kujud ja mudelid

Kuigi antud magistritöö raames oli peamiselt tegeletud 3D mudelitega, oli siiski oluline saada ülevaade ka võimalikest viisidest, kuidas leitakse sarnasused 2D kujundite vahel. 2D mudelite sarnasuse leidmine on tänaseks hästi uuritud ning selle kohta leidub palju erinevat kirjandustööd [16]. Seni koostatud Uurimistööd on peamiselt keskendunud tehisnägemise ja robootika valdkondadele ning põhiliselt on lähteandmetena kasutatud piltkujutisi, kus alguses tuvastatakse vajalik objekt ning hiljem leitakse sarnased objektid teistelt piltidelt [17].

2D kujundite puhul on enamlevinud võimalused geomeetrilise sarnasuse leidmisel sellised meetodid nagu:

- Hausdorff-i kaugus;
- Fréchet kaugus;
- Sümmeetrilise erinevuse pindala;
- *Earth mover*-i kaugus;
- Graafiline rekonstruktsioon.

Hausdorff-i kauguse puhul mõõdetakse kahe punktikomplekti vahelist kaugust, mis võiksid kujutada kahe kujundi piiri. See arvutab ühe komplekti punkti maksimaalse kauguse teise komplekti lähima punktini ja vastupidi. Mida väiksem on kaugus, seda sarnasemad on kujundid omavahel. Antud meetod on kõige efektiivsem, kui kujundid on oma struktuurilt sarnased, kuid võivad skaala või orientatsiooni poolest erineda. Fréchet-i kaugus on väga sarnane eelnevale kaugusele, kuid erinevuseks on see, et see meetod mõõdab kahe kõvera sarnasust, leides kahte kõverat ühendava köie lühima võimaliku pikkuse. Sarnaselt on ka selle kauguse puhul oluline see, et mida väiksem on kaugus, seda sarnasemad on omavahel objektid. *Earth mover*-i kaugus on eelnevast kahest kaugusest natukene erinev. See meetod arvutab iga punkti ühest kujundist teise teisaldamise maksumuse ja leiab optimaalse lahenduse, mis minimeerib kogukulusid. Kokkuvõttes mõõdab *Earth mover* minimaalset tööd, mis on kujundi teisendamiseks vajalik. Sümmeetrilise erinevuse pindala puhul mõõdetakse kahe kuju erinevust, arvutades nende sümmeetrilise erinevuse pindala. Sümmeetriline erinevus on tasapinna piirkond, mis kuulub ühele kahest kujunditest, kuid mitte mõlemale. Meetod võib kõige paremini sobida näiteks juhul, kui kujude proportsioonid on sarnased, kuid sisemised omadused on erinevad. Viies meetod, mis on üldkasutatav geograafilise sarnasuse leidmisel 2D kujundite puhul, on kujundi graafiku kujule rekonstrueerimine ning kahe graafiku omavaheline võrdlemine [15].

Küll aga ei ole need ainukesed meetodid kujundite sarnasuse leidmiseks ning läbi aastate on teadlased töötanud välja mitmeid erinevaid algoritme ja nende kombinatsioone, mis nii geomeetrilise sarnasuse kui ka üldise sarnasuse leidmise ära suudaksid lahendada. Näiteks üheks meetodiks oli pakutud välja kasutada 2D kuju histogrammi, kus kujutatakse 2D joonise kuju kasutades kauguse jaotust kahe juhuslikult valitud punkti vahel. Teiseks oli *Spherical Harmonics Transformation* ehk sfääriline teisendus, kus alguses teisendatakse kujud 3D ruumi ning seejärel kasutatakse sfäärilist teisendust, et saada pöörlemise invariantne deskriptor [17].

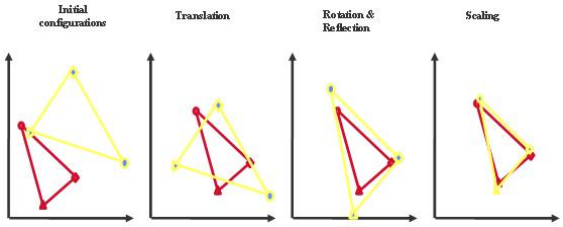
2.2.2 3D kujud ja mudelid

Maailm liigub aina rohkem kolmemõõtmelise maailma peale ning üha rohkem hakatakse erinevatest valdkondades töötama välja 3D mudeleid. Aina rohkem kasutatakse kolmemõõtmelist ruumi näiteks mängudes, kunstis, biokeemias, meditsiinis ning loomulikult ka ehituses ja arhitektuuris [18]. Tõusev trend on viimase aastatega suurendanud ka kirjandustööde tekkimist antud teemal ning üha rohkem töötatakse välja erinevaid meetodeid näiteks 3D mudelite geomeetrilise sarnasuse leidmisel [19].

3D mudelite puhul on loodud ning katsetatud mitmeid viise sarnasue leidmiseks. Tihti kasutatakse antud probleemi lahendamisel geomeetriapõhist lähenemist, mis jaguneb kaheks: kujupõhine ja tüpoloogiapõhine sobitamine. Kujupõhisel sarnasuse leidmisel kasutatakse tippude või hulknurkade jaotust, et hinnata 3D mudelite sarnasust. Tüpoloogia lähenemise puhul kasutatakse mudeli tüpoloogilisi struktuure. Mõlema lähenemise implementeerimisel on omad väljakutsed, probleemid ja kasutusalad ning tihti on vajalik kasutada nende omavahelist kombinatsiooni. Näiteks nii sõrm kui ka õlg inimese mudelil on osa tema kehast. Kuigi tüpoloogia vaatenurgast on tegu täiesti erinevate objektidega, siis mudelite kuju poolest on nad siiski sarnased [19].

Kolmemõõtmeliste mudelite sarnasuste leidmiseks kasutatakse tihtipeale tehisnägemise lähenemist, kus kujud skaleeritakse samale tasemele, pööratakse samaks ning seejärel viiakse koordinaatteljestikul võimalikult samale tasemele (vt Joonis 3) [19]. Nagu eelnevalt mainitud, siis 2D mudelite sarnasuse leidmisega on tegeletud palju rohkem ning on välja töötatud suur osa erinevaid lahendusi. Seetõttu on teadlased proovinud minna seda ka kolmemõõtmeliste mudelite puhul seda teed, et jaotada 3D objekt ühes suunas lõigates mitmeks väiksemaks 2D tükiks ning võrrelda hoopis tekkinud 2D kujusid omavahel. Kuigi selline lähenemine on võimeline leidma sarnasusi väiksemate mudelite seas, ei ole see siiski alati piisavalt täpne ning suurtemate mahtudega läheb maksimaalse täpsuse saavutamine veelgi raskemaks [18]. Ohbuchi and Takei [20] pakkusid enda artiklis välja uut lähenemisviisi hulknurksete kujundite mudelite sarnasuse mõõtmiseks, mis hõlmab endas 3D alfa kujundite (*alpha-shapes*) kasutamist. Meetod teisendab 3D hulknurga mudeli punktikomplekti mudeliks ning seejärel rekonstrueerib 3D alfa-kujude komplekti kasutades mitut ettenatud parameetrit. Seejärel tuletatakse deskriptor. Artikli tulemusel leiti, et uus lähenemisviis on palju täpsem kui varasemad sarnased meetodid, kuid on selle arvelt ka palju aeglasem. Laga, Takahashi ja Nakijama [21] tutvustasid enda artiklis tehnikat, mis lahendab 3D objektide sarnasuse hindamise probleemi kasutades geomeetrilisi kujutisi. Lühidalt, geomeetria kujutised on kujutised, mis jäädvustavad 3D-objekti geomeetrilisi omadusi ühel pildil, mis omakorda muudab need kompaktseks ja tõlkimiseks vastupidavaks. Siiski leiti, et algoritmi täiuslikuks testimiseks, efektiivsuse mõõtmiseks ning määramiseks

on vajalik läbi viia täiendavaid katseid.

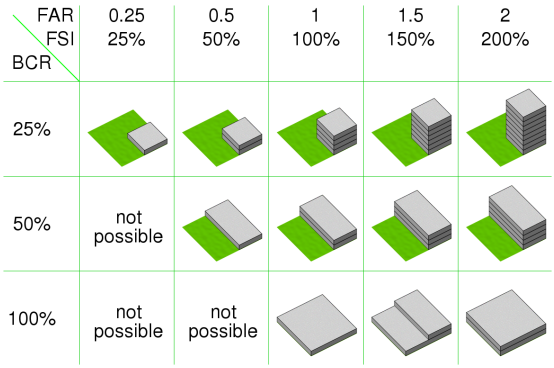


Joonis 3. Tehisnägemise ja pilditöötluse puhul tehtavad sammud sarnasuse leidmisel: skaleerimine, keeramine ning viiakse umbes samale tasemele

3D hoone ja linnamudelite sarnasus

Vaatamata sellele, et artikleid, mis käsitlesid sarnaste hoonete leidmist 3D mudelite baasil oli üpris vähe, õnnestus töö autoril leida siiski üks, mis oli käesoleva tööga väga sarnane. Hiina teadlaste tiim [22] töötas välja uue meetodi 3D linnamorfoloogia ja ruumilise struktuuri analüüsimiseks. Meetod põhineb samuti 3D hoonetel. Uuringus valiti teatud hoone atribuudid nende tugeva statistilise korrelatsiooni tõttu teiste näitajatega ja tuvastati neli kõige efektiivsemat omadust:

- Bfr (inglise keeles *Building footprint ratio*) ehk kui suur osa hõlmab vaadeldavast pinnast hoone [23]
- Hoone ainulaadsus [23]
- Hoonete tihedus selekteeritud alal [23]
- Põranda pindala suhe (FAR) hoone kasutatava põranda kogupinna suhte (ehk kõik korrused kokku) ja selle krundi kogupindala vahel, millel hoone asub (vt Joonis 4) [23]



Joonis 4. Põrandapinna suhte (FAR) ja hoone katvuse suhte (BCR *Building coverage ratio*) võrdlus

Neid hoone atribuute kasutatakse selleks, et jaotada Shanghai linna piirkonnad vastavalt nende omadustele erinevatesse gruppidesse, kasutades klasterdamise meetodit. Samuti ka, et defineerida selle baasil tüpoloogia, ehk näiteks piirkonnas number üks on enamik maju

x põranda pindala suhtega ning piirkonnas number kaks on keskmine põranda pindala suhe y. Töö autorid arvavad, et antud mudelit võib tulevikus kasutada linnaruumilise struktuuri analüüsimisel ning sellel on universaalne rakendatavus ning tähtsus linnamorfoloogia uuringutes.

2.3 Masinõpe

Masinõpe (ML, *Machine Learning*) on tehisintellekti (AI, *Artificial Intelligence*) alamvaldkond, mis keskendub algoritmide ja mudelite loomisele. Läbi selle võimaldab masinõpe masinatel erinevatest andmetest õppida ning selle põhjal ennustusi ja otsuseid teha. Andmete põhjal õppivate masinate kontseptsioon on tegelikult olnud kasutusel juba vähemalt 1950. aastatest saati, kuid hiljutised edusammud andmete kogumise, arvutivõimsuse ja täiustatud õppealgoritmide vallas on õhutanud uut huvilainet selle valdkonna vastu [24].

Oma põhimõttelt hõlmab masinõpe arvutiprogrammi koolitamist andmete mustrite tuvastamiseks. Tavaliselt on see tehtud nii, et programmile söödetakse suur hulk märgistatud või märgistamata andmeid. Näiteks märgistatud andmeteks võib olla piltide komplekt, millel on küljes silt, mis ütleb, kas tegemist on koera või kassiga. Seejärel toimub nende andmete põhjal mudeli treenimine. Treenimise tulemusel peaks mudel suutma täpselt klassifitseerida (kui tegemist on märgistatud andmetega) uusi temale tundmatuid märgistamata andmeid [25].

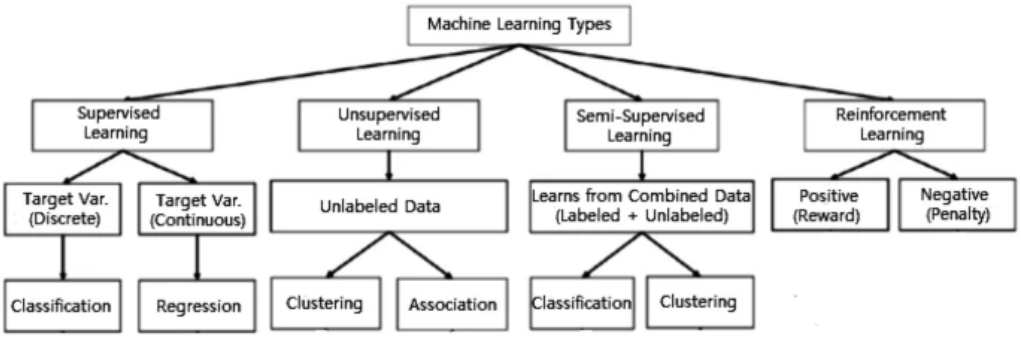
Aastatega on loodud mitmeid erinevaid masinõppe algoritme ja mudeleid. Nende kasutusala on aga erinev ning oleneb sellest, kas eesmärgiks on lahendada näiteks klassifitseerimise, regressiooni, klasterdamise või mingi muu probleem. Lisaks sellele oleneb ka, millist õppeviisi on vaja andmestikul rakendada [26].

Masinõppe algoritmidel võib olla üldjuhul neli erinevat õppeviisi [26]:

- Juhitud õpe (*supervised learning*): Mudeli treenimise puhul on sisendandmed üldjuhul märgistatud. Mudeli eesmärgiks on ennustada uute märgistamata andmete väljundit vastavalt treenitud andmetele [27, 24].
- Juhtimata õpe (*unsupervised learning*): Sisendandmed ei pruugi olla alati tulemi osas märgistatud ning eesmärgiks on leida või tuletada sarnased mustrid ja/või struktuurid ning luua üldistavad reeglid [27, 24]
- Pooleldi juhitud õpe (*semi-supervised learning*): Kombinatsioon juhitud ja juhtimata õppeviisidest ehk sisendandmed võivad olla nii märgistatud kui ka märgistamata. Mudeli eesmärgiks on nii ennustada väljundit kui ka leida erinevaid struktuure ning reegleid [27].

- Kinnitustega õppimine (*reinforcement learning*): Oma olemuselt on tegu katse-eksitus meetodiga, kus agent õpib keskkonnaga suheldes ning teeb järeldused vastavalt oma tegevustele saadud tagasisidele [28].

Ülaltoodud definitsioonid on ülevaatlikult kirjeldatud ka joonisel 5.



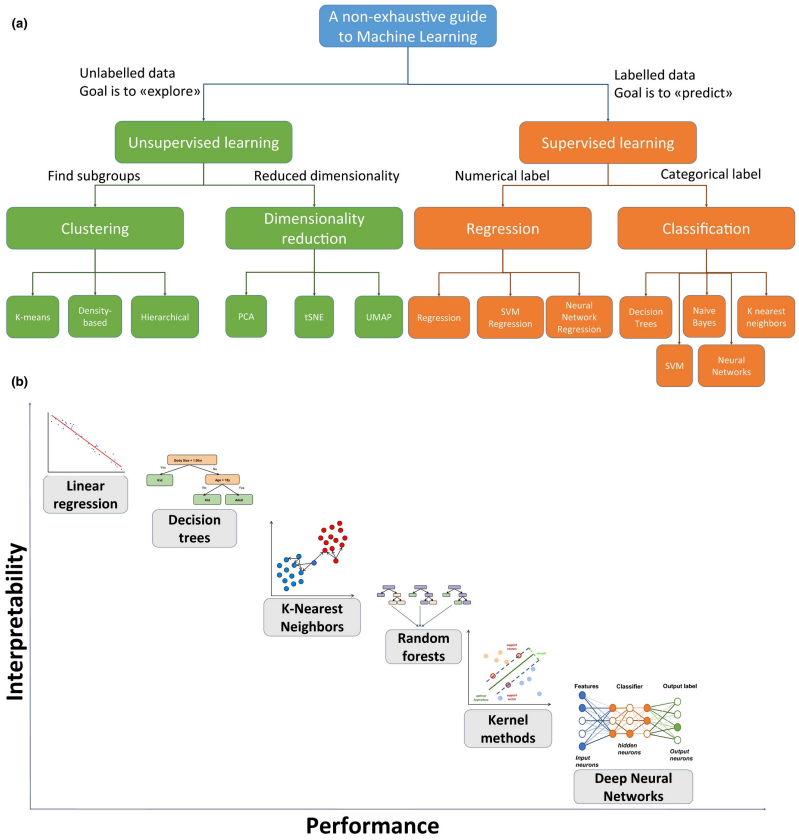
Joonis 5. Õppeviisid ja nende alla kuuluvad algoritmi kategooriad [29]

Mis puudutab aga algoritmide grupeerimist, siis Brownlee [30] on toonud välja 11 kategooriat kõige populaarsematest masinõppe algoritmidest:

- Regressiooni algoritmid: lineaarregressioon, logistiline regressioon;
- Juhtumil põhinevad algortimid: k-lähima naabri algoritm (kNN *k-Nearest Neighbor*), ise organiseeruv kaart (SOM *Self-Organizing Map*), tugivektor masin (SVM *Support Vector Machines*);
- Seaduspärasuse algoritmid: Ridge regressioon, LASSO, LARS;
- Otsustuspuu algoritmid: klassifikatsiooni ja regressiooni puu (CART), tingimuspõhised otsustuspuu;
- Bayesi algortimid: Naiivne Bayes, Gaussi naiivne bayes;
- Klasterdamise algortimid: k-keskmine, k-mediaan, hierarhiline klasterdamine;
- Assotsiatsioonireeglite õppimise algortimid: Apriori algoritm, Esclat algoritm;
- Tehislikud närvivõrkude algortimid: Percpetron, Mitmekihilised Percptronid (MLP *Multilayer Perceptrons*);
- Sügavõppe algoritmid: konvolutsiooniline närvivõrk (CNN *Convolutional Neural Network*), Rekurrentne närvivõrk (RNN *Recurrent Neural Network*);
- Mõõtmete vähendamise algoritmid: Peakomponentide analüüs (PCA *Principal component analysis*;
- Ansambelõppe algoritmid: AdaBoost, juhuslik mets.

Väike hulk ülaltoodud kategooriatest ning algoritmidest on illustreeritud koos õppimisviisidega ka joonisel 6.

Lisaks ülalmainitule on kasutusel ka veel mitmeid erinevaid juhupõhiseid kategooriad, mis on sobivad ainult kindla ülesande või probleemi lahendamiseks. Näiteks erinevad soovitussüsteemide ja tehisnägemisega seotud algoritmid [30].



Joonis 6. Visuaalne kujutis kuhu alla kuuluvad mõnigad algortimid [24]

Masinõpet kasutatakse tänapäeval mitmel erineval otstarbel, alates pildi- ja kõnetuvastusest kuni pettuste tuvastamise ja soovituse süsteemideni. Masinõpe on kasutusel ka väga mitmes valdkonnas, sealhulgas robootika, arvutimängud, virtuaalsed isiklikud assitendid, liikluse ennustamine, meditsiiniline diagnostika jne.

Järgnevalt annab autor detailsema ülevaate masinõppe algoritmidest, mida on antud töö raames kasutatud.

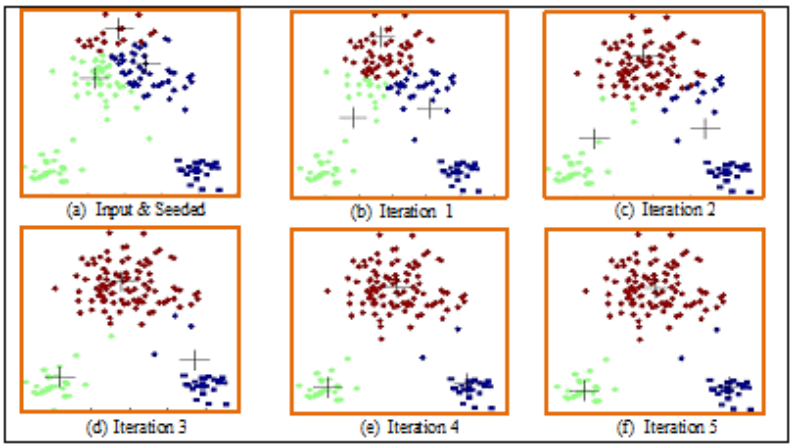
2.3.1 Valitud masinõppe algoritmid

Eelneva alapeatüki põhjal on võimalik näha, et tänaseks päevaks on loodud suur hulk erinevaid masinõppe algoritme, mida laialdaselt kasutatakse. Sellele vaatamata on käesoleva töö eksperimentide jaoks valitud vaid kaheksa algoritmi. Valik on tehtud nii nende tööpõhimõte, eesmärgi kui ka populaarsuse põhjal. Masinõppe mudelid, mida antud töös kasutatakse ning millest tuleb järgnevalt ka põhjalikumalt juttu on: k-keskmine, Hierarhiline klasterdamine, Gaussi segu mudelid (*Gaussian Mixture models*), Logistiline regressioon,

Otsustuspuu, juhuslik mets, närvivõrgud ning tugivektor masinad (SVM).

K-Keskmine

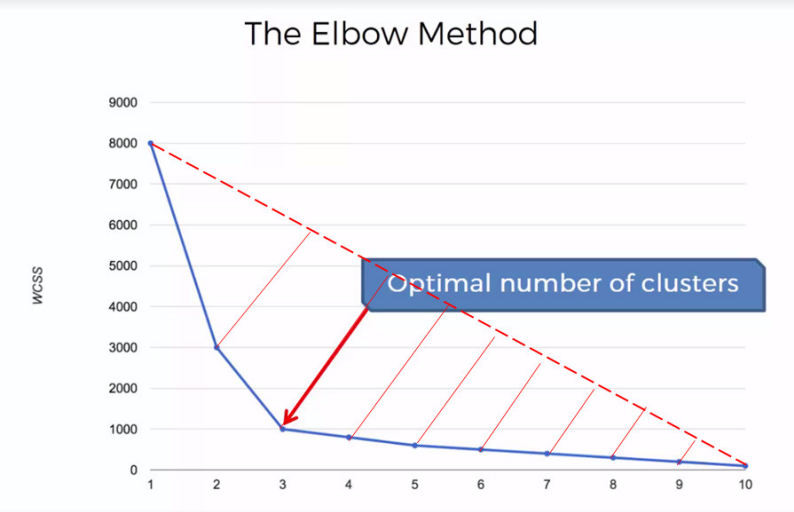
K-keskmise (*K-Means*) meetod on üks populaarsemaid klasterdamise masinõppe algoritme, mida kasutatakse mitmes erinevas valdkonnas, olgu selleks tehisnägemine või sarnase mustri leidmine. Mudeli eesmärk on jagada andmepunktide komplekt nende sarnasuse alusel K erinevaks klastriks. Alguses valib mudel juhuslikult klastrite K tsentroidi, mis toimivad klastrite niinimetatud algkeskustena. Seejäärel määrab algoritm iteratiivselt iga andmepunkti klastrile, mille keskpunkt on sellele kõige lähemal. Selleks kasutatakse kauguse mõõdikut, üldjuhul on selleks Eukleidiline kaugus. Pärast seda kui kõik punktid on klastrite vahel ära jaotatud, arvutatakse iga tsentroid uuesti ümber, kuid seekord vastavalt andmepunktide keskmisele. Klastritele andmepunktide määramise ja tsentroidide ümberarvutamise protsessi korratakse seni, kuni tsentroidid enam oluliselt ei muutu või on saavutatud määratud maksimaalne iteratsioonide arv [31, 32]. Kirjeldatud k-keskmise protsessi etappe on kujutatud ka joonisel 7.



Joonis 7. K-keskmise etapid korrektsetesse klastritesse jõudmiseni [33]

Vaatamata sellele, et klasterdamise meetodid suudavad andmeid ise rühmitada ning leida mingisuguseid reegleid ja struktuure andmete vahel, peab üldjuhul klastrite arvu, mille vahel nad andmeidd hakkavad jagama, eeldefineerima. See on üks suurimaid väljakutseid, mille ees seisab nii k-keskmine kui ka paljud teised klasterdamise algoritmid. Aastate jooksul on antud probleemi üritatud mitut moodi lahendada ning tänaseks üks levinumaid meetodeid selle lahendamiseks on kasutada küünarnuki meetodit. Meetodi põhiidee on joonistada andmepunktide ja neile määratud tsentroidide vahelise ruudu summa (tuntud ka kui klastrisisese ruutude summa WCSS *Within Cluster Sum of Squares*) ja klastrite arv K. Seejärel tuleb graafikut visuaalselt kontrollida, et tuvastada nö "küünarnuki"lõikepunkt (vt Joonis 8), mis näitab head kompromissi klastrite täpsuse ja keerukuse vahel. Tavaliselt mis juhtub on see, et klastrite suurenemise puhul hakkab WCSS vähenema, kuid mingil hetkel muutub selle vähenemine vähemoluliseks ning rohkemate klastrite lisamine võib hoopiski

põhjustada ülepaigutamist [31, 32].

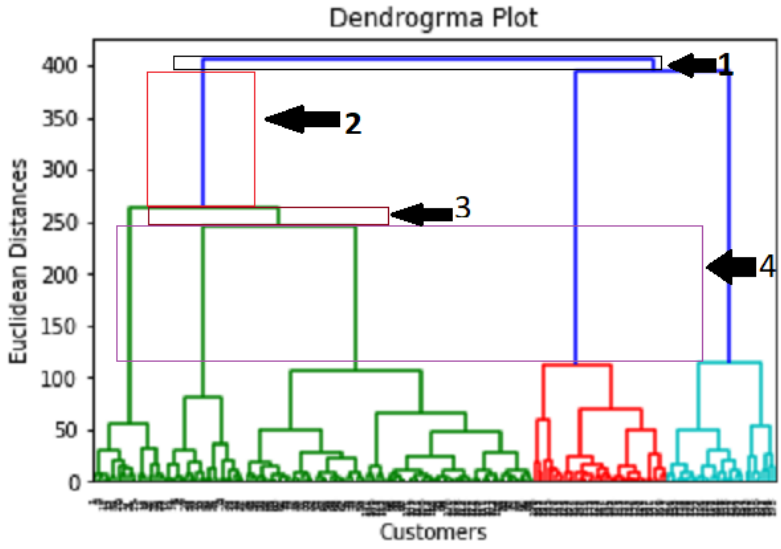


Joonis 8. Küünarnuki meetodi abil optimaalse klastrite arvu leidmine. Y telg on WCSS ning X teljel on klastrite arv [34]

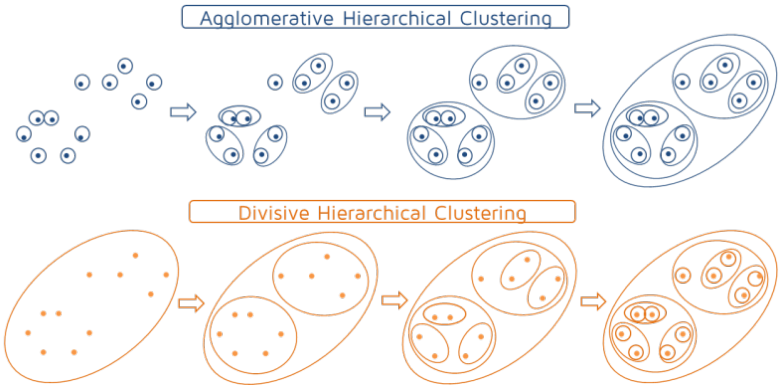
Hierarhiline klasterdamine

Hierarhiline klasterdamine on järgmine populaarne masinõppe mudel, mis tegeleb andmete klasterdamisega. Hierarhilise üks põhilisi erinevusi ning samaaegselt ka eeliseid k-keskmise ees on see, et selle jaoks ei pea eeldefineerima klastrite numbrit. Alguses käsitleb mudel igat andmepunkti eraldi iseseisva klastrina. Seejärel otsitakse kõige lähemal olev klaster (kõige sarnasem) ning liidetakse kaks klastrit omavahel kokku. Lõpuks arvutatakse uued kaugused tekkinud klastri ning vanemate klastrite vahel. Eelkirjeldatud protsess toimub niikaua, kuni pole tekkinud üks klaster, kuhu kuuluvad kõik andmepunktid [35]. Hierarhilise klastri moodustamisel on tulemuseks puutaolise diagrammiga sarnane dendrogramm (vt Joonis 9), mis näitab klastrite ühinemise järjekorda ja ka klastrite vahelist kaugust. Juhul, kui peaks olema soov eeldefineerida klastrite number, siis üheks võimaluseks on visualiseerida tekkinud dendrogramm ning selle põhjal leida sobivaim arv klastreid. Üldjuhul vaadatakse kaugust, mille horisontaalsel teljel ei toimu ühtegi uue klastri tekkimist [36].

Hierarhilist klastrit on peamiselt kahte tüüpi: aglomeratiivne ja jagunev. Algomeratiivne on kõige enam levinud tüüp ning selle tööpõhimõte on tegelikult täpselt sama nagu eelnevalt oli kirjeldatud. Jagunev hierarhiline klasterdamine on palju vähem kasutatud ning üldjoontes on selle tööpõhimõte vastupidine algomertiivsele. Selle puhul kogub mudel alguses kõik andmepunktid ühte klastrisse ning seejärel hakkab neid tükkhaaval sealt välja võtma, kuni iga andmepunkt on jaotatud eraldi klastrisse [37]. Joonisel 10 on illustreeritud mõlema tüübi etapid.



Joonis 9. Hierarhilise klasterdamise tulemusel tekkinud Dendrogramm. Antud juhul on optimaalne klastrite arv 5 [36]



Joonis 10. Algomeratiivse ja jaguneva hierarhilise klasterdamise töö etapid [38]

Sarnaselt k-keskmisele on hierarhilise puhul kasutusel erinevad mõõdikud kauguse arvutamiseks. Kõige levinum on samuti Eukleidiline kaugus, kuid tihti kasutatakse ka Manhattani kaugust või korrelatsioonikordajat, olenevalt sellest, milliste andmetega on tegemist [39].

Gaussian Mixture models

Kolmas ning viimane klasterdamise masinõppe meetod, mida käesoleva töö puhul on kasutatud, on Gaussi segu mudelid (GMM *Gaussian Mixture Models*). Tegemist on tõenäosusmudeli tüübiga, mida tihti kasutatakse nii klastrite moodustamiseks kui ka tiheduse hindamiseks. Antud mudelit võib nimetada ka tõenäousel põhinevak k-keskmiseks. Põhjuseks on see, et sarnaselt k-keskmisele on mõlema treeningprotsess ning lähtepunkt samad, kuid k-keskmine kasutab vahemaapõhist lähenemist, aga Gaussi segu mudelid kasutavad selle asemel tõenäosusliku lähenemist [40]. GMM-i põhiidee on esitada andmeid mitme muutujaga Gaussi jaotuse seguna, millest igaühel on oma keskmine ja kovariatsiooni-

maatriks. Seejärel hindab mudel, tavaliselt kasutades maksimaalset tõenäosust või Bayesi järeldust, jaotuste parameetreid andmete põhjal [41]. Sarnaselt eelnevatele mudelitele saab klastrite arvu fikseerida või arvutada välja optimaalne klastrite arv, kasutades selliseid meetodeid nagu näiteks Bayesi teabekriteerium [40].

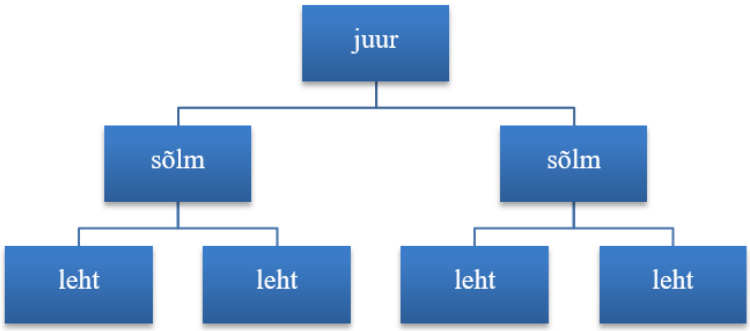
Logistiline regressioon

Järgnevalt kirjeldatakse erinevaid klassifitseerimise mudeleid. Esimeseks on võetud logistiline regressioon. Üldjuhul annab antud mudel binoomtulemuse, sest vastavalt sisendandmete väärtustele, annab see tõenäosuse, kas mingisugune sündmus toimub või ei. Näiteks selliste juhtude alla kuulub kasvaja prognoosimine (kas on pahaloomuline või ei) või ka kirja rämpsposti saatmine (e-kiri on rämpspost või ei ole) [26]. Tänaseks on loodud mitu erinevat logistilise regressiooni tüüpi, mistõttu on mudeli kasutusvaldkond läinud palju laiemaks. Antud meetodi tüübid on järgmised [42]:

- Binaarne logistiline regressioon - Klassikaline mudel, kus on ainult kaks võimaliku järeldus;
- Multinominaalne logistiline regressioon - Mudel, mille tulemuseks võib olla enam kui kaks võimaliku diskreetset väärtust. Tegemist on väärtustega, mis ei ole sisulistel järjestatud. Näiteks kas on taimetoitlane, mitte taimetoitlane või vegan;
- Järjestatud logistiline regressioon - Sarnane multinominaalsega, kui tegemist on järjestatud väärtustega. Näiteks prognoosimine, mis on filmi hinne ühest viieni.

Otsustuspuu

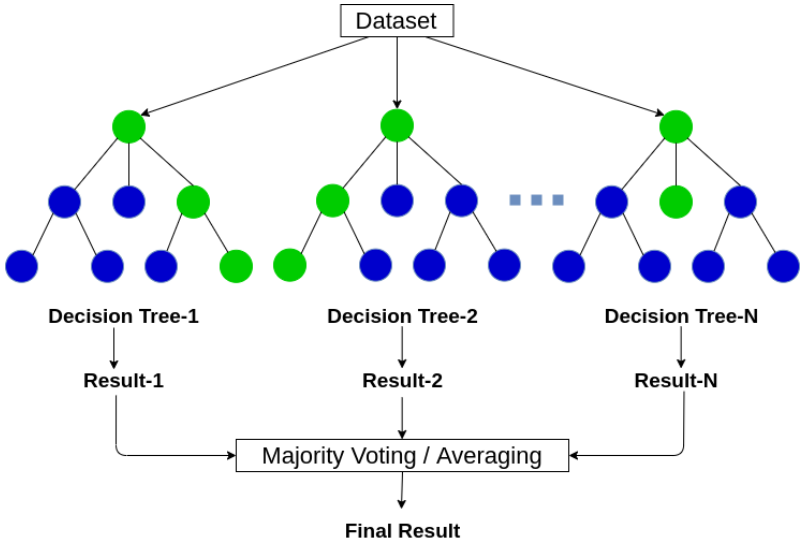
Otsustuspuu on populaarne masinõppe algoritm, mida kasutatakse nii klassifitseerimise kui ka regressiooni ülesannete jaoks. Otsustuspuu põhiidee on andmete jaotamine väiksemateks alamhulkadeks sisendfunktsiooni väärtuste põhjal. Seda andmete jagamise protsessi juhivad erinevad reeglid ehk "jagamiskriteeriumid", mille eesmärk on maksimeerida teabe või puhtuse suurenemist igal etapil. Tulemuseks on tavaliselt puustruktuur, mida saab esitada "kui-siis"(*if-else*) lausete seeriana, kus otsused on lehtedes ning andmed on jagatud sõlmedesse (vt Joonis 11). Antud mudel saab hakkama nii kategooriliste kui ka pidevate sisendfunktsioonidega ning on võimeline õppima keerulisi mittelineaarseid seoseid sisend- ja väljundmuutujate vahel. Otsustuspuud võivad aga väga kergelt kannatada ülepaigutamise (*overfitting*) all, mistõttu kasutatakse nende puhul tihti ka ansambelõpet, et mudeleid laiendada teiste masinõppe algoritmidega, nagu näiteks juhuslikud metsad, ning ühtlasi parandada täpsust ja järjepidavust [26].



Joonis 11. Otsustuspuu struktuur [43]

Juhuslik mets

Juhuslik mets on ansambelõppe algoritm, mis kombineerib mitu erinevat otsustuspuud, et parandada mudeli täpsust, järjepidevust ja efektiivsust [44]. Antud masinõppe meetodit on võimalik kasutada nii klassifitseermise kui ka regressiooni ülesannete lahendamisel ning need saavad hästi hakkama ka väga suurte andmehulkade ja suure koguse parameetritega. Mudeli põhiidee on treenida erinevaid otsustuspuid erinevatel andmehulkadel ja parameetritel ning koondada nende ennustused häälteenamuse või kaalutud keskmise abil (vt Joonis 12). Juhuse ja liitmisprotsess aitab vähendada mudeli ülepaigutamist ning ühtlasi ka suurendada selle stabiilsust [45].



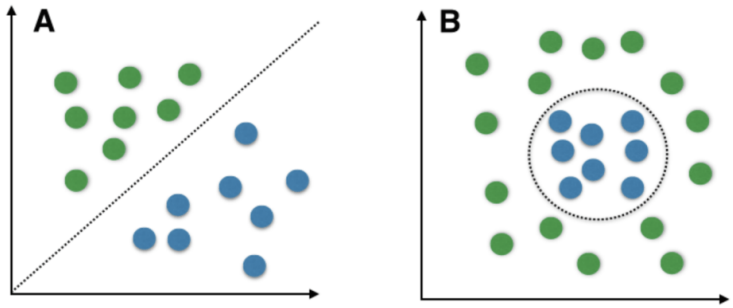
Joonis 12. Juhusliku metsa tööpõhimõte [46]

Viimase 10 aastaga on juhuslik mets väga palju ning väga kiiresti arenenud, mistõttu on kasvanud ka selle kasutusala ja valdkonnad. Juhuslik mets on kasutusel näiteks bioinformaatikas, meditsiinis, juhtimisteaduses ning ka majanduses. Meditsiinis kasutatakse mudelit näiteks kopsusõlmede automaatse tuvastamise hõlbustamiseks. Majandusjuhtimises on aga kõige laialdasem kasutusala näiteks klientide kahjumäära ennustamine. Samuti on aastatega tehtud ka mitmeid uuringuid, kus on võrreldud juhusliku metsa mudelit paljude

teiste klassifikatsiooni mudelitega ning mitmel korral, olenevalt valdkonnast muidugi, on eelistatud juhuslikku metsa [45].

Tugivektor-masinad SVM

Tugivektor-masinate mudeli ehk SVM-i puhul on tegemist võrdlemisi uue algoritmiga, mis on samuti viimaste aastatega populaarsust kogunud. Tegemist on mudeliga, mis on sobilik nii klassifikatsiooni kui ka regressiooni ülesannete lahendamiseks. Selle tööpõhimõte on leida hüpertasand, mis eraldab sisendandmed erinevatesse klassidesse, kus marginaal määratletakse kaugusena hüpertasandi ja lähimate andmepunktide vahel mõlemal küljel. Hüpertasand on valitud nii, et see maksimeeriks marginaali ning minimeeriks treeningandmete klassifitseerimisel tekkinud vea [47]. Juhul, kui andmed ei ole lineaarselt eraldatavad, kasutab SVM teatud tehnikat, mida kutsutakse ka "kerneli trikiks"(*"Kenel trick"*), et muuta sisendparameetrid kõrgema mõõtmega ruumiks, kus andmed muutuvad eraldatavaks, ning seejärel leida uues ruumis hüpertasand. Seetõttu on SVM-i hea kasutada nii lineaarsete kui ka mittelineaarsete andmete puhul [48]. Joonisel 13 on kujutatud hüpertasandiga jagamine andmed kahte klassi.



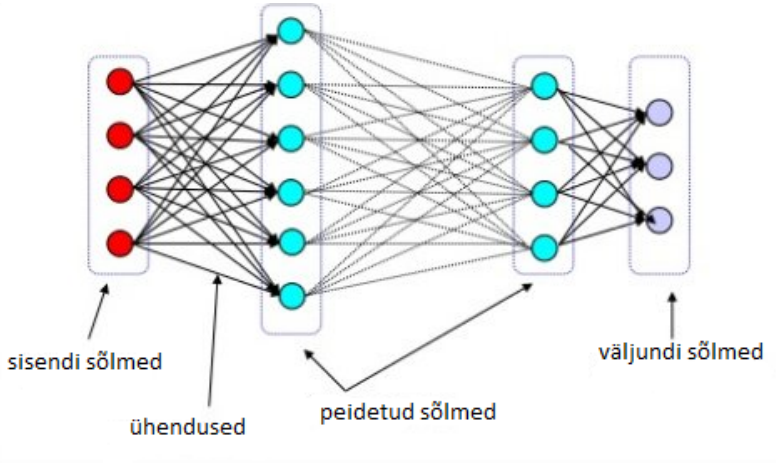
Joonis 13. Vasakul pool on hüpertasandiga jaotatud lineaarsed andmed. Paremal pool on tegemist aga mittelineaarsete andmetege [49]

Tugivektor-masinad on aastatega kogunud suurel määral populaarsust, ning sarnaselt ka paljudele teistele algoritmidele, on sellest saanud üpris populaarne ning laialdaselt kasutatav algoritm. Mõned kohad, kus SVM on näidanud väga häid tulemusi, on näiteks käsikirjas kirjutatud numbri tuvastamine, teksti klassifikatsioon ning ka pildilt objektide tuvastamine [50].

Närvivõrgud

Närvivõrgud, tuntud ka kui tehisnärvivõrgud, on masinõppe algoritmide klass, mis on inspireeritud aju bioloogiliste neuronite struktuurist ja funktsionaalsust. Närvivõrkude tööpõhimõte on modelleerida keerulisi sisend- ja väljundsuhteid üksteisega ühendatud tehisneuronite kihtide (vahel kutsutakse ka sõlmedeks) abil. Närvivõrkudes on üldjuhul kolme tüüpi kihte: sisendkiht, peidetud kiht ning väljundi kiht (vt Joonis 14), kuid mudeli

kirjeldamisel tavaliselt arvestatakse ainult peidetud kihiga ehk kui mudelil peaks olema sisendkiht, peidetud kiht ja väljund kiht, siis võib öelda, et tegu on ühekihilise mudeliga. Kihid sooritavad tavaliselt sisendandmetega lihtsaid toiminguid ning edastavad tulemused järgmisesse kihti. Iga närvivõrgu neuronit iseloomustab hulk õpitavaid parameetreid, nagu näiteks kaalud, mis määravad vastava neuroni kihi käitumist [51].



Joonis 14. Närvivõrgu ülesehitus [43]

Närvivõrgud võib tavaliselt jaotada kolmeks tüübiks: lihtsamad närvivõrgud, mis koosnevad ühest või maksimaaslelt kahest peidetud kihist (suurem osa närvivõrke), keerulisemad närvivõrgu mudelid, mis võivad koosneda rohkematest kihtidest ning tegeleda mahukamate ja keerulisemate andmestikega. Kolmas on konvolutsioonilised närvivõrgud, mida peamiselt kasutatakse tehisnägemises kujutiste ja piltide töötlemisel [51].

2.3.2 Mudelite tulemuste mõõtmine

Olenevalt valitud õppeviisist ning vastavast ülesandest on oluline valida ka õige tulemuste mõõdik, et saada vastused sellistele küsimustele nagu näiteks [52]:

- Kui hea on mudeli jõudlus ja ennustuse täpsus?
- Kui suur mudeli kasu?

Kui klasterdamise ülesannete puhul on tulemuste leidmine üldjuhul seotud pigem visuaalse kontrolli ning seoste leidmisega, siis klassifikatsiooni puhul kasutatakse mitmeid erinevaid mõõdikuid, et luua ühine arusaam mudeli sooritusest. Järgnevalt on detailsemalt lahtikirjutatud, millised mõõdikud on kõige tihedamini kasutatud ning mida need mudelite puhul näitavad.

Klassifikatsiooni mõõdikud

Klassifikatsiooni mudelite tulemuste mõõtmisel on kõige esimeseks etapiks andmete jagamine treening ning testandmeteks. Suuremate andmekogumite puhul on soovitatav luua ka valideerimis andmestik. Suhe andmestike vahel peaks olema üldjuhul, kas 80% treeningandmed ning 20% testandmed või kui on võimalus siis isegi 60% treeningandmed, 20% testandmed ning 20% valideerimisandmed. Testandmestiku põhjal toimub üldjuhul tulemuste testimine ning valideerimine [52].

Kõige tavapärasem mõõdik mida klassifikatsiooni mudelite puhul kasutatakse on mudeli täpsus ehk kui suur osa kõikidest ennustustes olid õiged [52].

$$\text{Täpsus} = \frac{\text{õiged ennustused}}{\text{kõik ennustused}}$$

Klassifitseerimise puhul esineb tavaliselt nelja tüüpi tulemusi: õige positiivne, vale negatiivne, vale positiivne ja õige negatiivne. Kõik need tulemuse tüübid on kujutatud eksimismaatriksil [53] (vt Tabel 1).

Tabel 1. Eksimismaatriks

	Ennustatud P	**Ennustatud N**
Tegelik P	Õige Positiivne	Vale Negatiivne
Tegelik N	Vale Positiivne	Õige Negatiivne

Lisaks mainitud tulemuste tüüpidele, on eksimismaatriksit kasutades võimalik tuletada ka teisi mõõdikuid, mis on klassifitseerimise mudeli hindamisel väga laialt kasutatavad. Üks neist on mudeli positiivne ennustusvõime *precision*. See näitab kui suur osa ennustatud positiivsetest olid tegelikult ka positiivsed [53].

$$\text{Positiivne ennustusvõime} = \frac{\text{Õige positiivne}}{\text{Õige positiivne} + \text{Vale positiivne}}$$

Järgmine maatriksist tuletatud mõõdik on õige-positiivsete määr ehk tundlikkus, mis on tuntud ka kui täielikkus (*Recall*). See näitab kui suurt osa tegelikest positiivsetest ennustas mudel positiivseteks [53].

$$\text{Tundlikkus} = \frac{\text{Õige positiivne}}{\text{Õige positiivne} + \text{Vale negatiivne}}$$

Lõpuks kombineeritakse mõlemad väärtused üheks mõõdikuks, mis kannab nime F1 skoor. Selle puhul on oluline jälgida, mida lähemal on selle väärtus ühele, seda parem ning stabiilsem on mudeli ennustusvõime [53].

$$\text{F1} = 2 \times \frac{\text{Positiivne ennustusvõime} \times \text{Tundlikkus}}{\text{Positiivne ennustusvõime} + \text{Tundlikkus}}$$

3. Eksperimendid

Käesolevas peatükis on detailsemalt kirjeldatud, kuidas eksperimendid läbi viidi, millised olid andmed, milliseid masinõppe mudeleid kasutati ning milliste tulemusteni jõuti.

3.1 Eksperimendi disain

Töö raames läbiviidud eksperimendid koosnesid peamiselt neljast suuremast etapist, mis on alltoodud punktides lühidalt kirjeldatud.

1. Andmete kogumine ja arvutuste tegemine

Eksperimentide aluseks olid neli erinevat andmestiku: 417 Kredexi hoone andmestik, Tüpoloogia andmestik, tudengite poolt koostatud andmestik sarnaste hoonete kohta ning nõukogudeaegsete paneelmajade andmebaas. Andmed olid peamiselt saadud kas RESTO projektilt või koostatud koostöös juhendajaga. Viimases kahes andmestikus puudusid vaikimisi lisaomadused ja arvutused, mistõttu pidi nende peale rakendama ka eraldi API *endpoint*-i, et kõik vajalikud matemaatilised andmed oleksid samuti iga maja juures olemas.

2. Õige mudeli valik vastavalt andmestikule

Kui andmestikud olid valmis, oli järgmiseks etapiks korrektse masinõppe mudeli valimine. Iga andmestik oli üksteisest mõneti erinev ning seetõttu oli tarvis rakendada erinevaid mudeleid erinevate andmestike peal. Näiteks klassifikatsiooni mudelite puhul ei olnud mõistlik kasutada tudengite hoonete valimit, sest puudus kindel grupp/sihtväärtus, mille järgi hooneid klassifitseerida/ennustada.

3. Parameetrite valik, teisendamine ja agregeerimine

Mudeli valiku tegemisele järgnes andmete normaliseerimine, ühtsele skaalale viimine ning parameetrite valimine. Kõikide masinõppe algoritmide puhul oli oluline valida korrektsed sisendandmed, kuid näiteks klassifitseerimise ülesande lahendamisel pidi defineerima ka sihtandmed ning klasterdamise puhul oli oluline leida kõige optimaalsem klastrite arv.

4. Mudelite treenimine ja testimine

Viimases etapis rakendati andmestikule vastavaid masinõppe algoritme. Klassifikatsiooni puhul jaotati andmestikud omakroda ka treeninbaasiks ning testimisebaasiks, et ennustamise täpsust hiljem testida.

Järgnevates alapeatükkides on detailsemalt selgitatud igat ülaltoodud etappi ning mida täpselt tehti, et tulemused kätte saada.

3.1.1 Andmete kogumine ning arvutuste tegemine

Käesoleva töö raames läbiviidud eksperimentide puhul oli kasutuses neli eri andmestikku: 417 Kredexi poolt toetatud hoone andmestik, RESTO projekti ja eelneva aasta lõputöö koostöös loodud tüpoloogia andmestik, tudengite poolt koostatud sarnaste hoonete andmestik ning nõukogudeaegsete paneelmajade andmestik. Esimesed kaks valimit olid saadud RESTO projekti poolselt konsultandilt. Tudengite sarnaste majade nimekiri oli koostatud ühe õppeaine raames äriinfotehnoloogia tudengite poolt kasutades 3D maa-ameti kaarti. Viimane ehk paneelmajade andmebaas on avalikult kättesaadav andmestik, kust on võimalik saada Eestis rajatud paneelmajade aadressi ning selle projekti ja/või seerianumbri. EHR-i kood antud hoonetele sai lisatud juurde kasutades 3D maa-ameti kaarti.

Kõige suuremad ning detailsemad andmestikud, mida antud töö puhul sai kasutada, olid tüpoloogia andmestik, mis valmis 2023 aasta lõpus tudengi magistrtöö tulemusel, ning kredexi poolt toetatud 417 hoonete andmestik, mis oli tüpoloogia koostamisel samuti kasutusel. Käesoleva töö eksperimentide puhul olid mõlemad andmestikud kasutuses koos, kuid natukene erineva otstarbega. Tüpoloogia andmestiku puhul olid olemas juba grupid, mis olid manuaalselt koostatud ning hooned vastavalt omadustele gruppidesse jaotatud. Samuti oli selles andmestikus toodud välja ka mitmed tüüphooned. Kredexi hoonete andmestik aga kujutas endast valimit 417 hoonest, millel oli suur hulk erinevaid omadusi ning arvutusi juba juures, mille järgi oleks maju võimalik gruppeerida. Eksperimendid olid jooksutatud Kredexi hoonete andmete peal ning Tüpoloogia andmestik oli kasutuses pigem mudelite testimise otstarbeks. Antud andmestike kombinatsioonil oli võimalik jooksutada ka kõige rohkem eksperimente, sest võis läheneda sellele nii juhtimata kui ka juhitud õppeviisiga ehk kasutada nii klasterdamist kui ka klassifitseerimist. Näiteks sai jooksutada masinõppe klasterdamise mudeleid ning vaadata, kas algoritm suudaks jagada majad samadesse gruppidesse nagu seda oli tehtud tüpoloogi andmestiku koostamisel. Teisisõnu oli tegemist rohkem tüpoloogia andmestiku valideerimisega ja prooviga saavutada sarnased tulemused automatiseerimise teel. Teiselt poolt aga oli võimalik jooksutada ka klassifitseerimise mudeleid, kus sisendandmeteks on omadused ja arvutused ning sihtväärtuseks on varasemalt manuaalselt loodud tüüp ning seejärel testida, kui suur hulk hooneid sattus õigesse gruppi. Näide tüpoloogia andmestikust on toodud välja lisas 2.

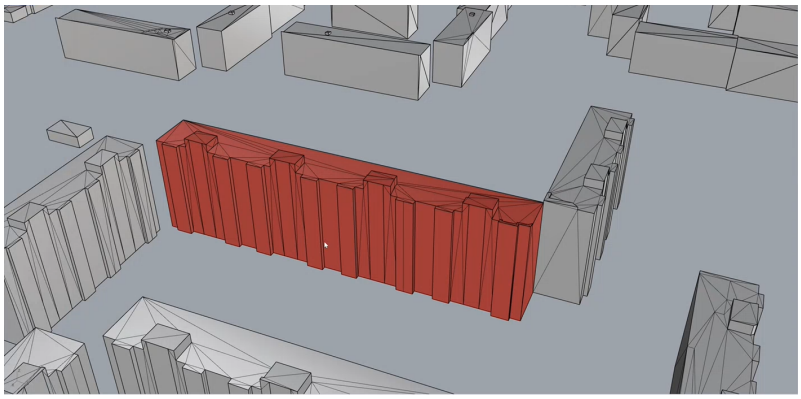
Teiste andmestikke puhul oli algselt tegemist vaid andmetabelitega, mis sisaldasid vaid paari hoone sisendit nagu näiteks aadress, EHR-i kood ning võib olla ka mingisugune väärtus, mis näitas, et tegu peaks olema samat tüüpi hoonega. Selleks aga, et saada kõik vajalikud arvutused ning lisaomadused maja kohta teada, oli võimalik kasutada RESTO projekti raames loodud API *endpoint*i. Arvutuste kättesaamiseks oli vaja teha HTTP päring antud endpointile ning anda kaasa maja EHR-i kood. Vastuseks tuli LOD2 põhjal tehtud arvutused, mis olid pakitud JSON kujule. Hiljem olid need arvutused lisatud andmestikele juurde, mis muutsid andmestikud piisavalt heaks, et jooksutada masinõppe mooduleid.

Tabel 2. RESTO API-st saadud arvutustega JSON objekti parameetrid ning tähendused

JSON võti	**Tähendus**
L1	Katuse tüüp
L2	Kõrgus
L3	Ehitusaluse pinna perimeeter
L4	Tuletatud maht
L5	Fassaadi pindala
L6	Katuse pindala
L8	Pööningu vahelae pindala
L9	Pinnasel põranda pindala
L10	Kütmata keldrilae pindala
L11	N (Põhi) fassaadi pindala
L12	NE (Kirre) fassaadi pindala
L13	E (Ida) fassaadi pindala
L14	SE (Kagu) fassaadi pindala
L15	S (Lõuna) fassaadi pindala
L16	SW (Edel) fassaadi pindala
L17	W (Lääs) fassaadi pindala
L18	NW (Loe) fassaadi pindala
L19	Ehitusaluse pinna välisnurkade arv
L20	Ehitusaluse pinna sisenurkade arv
L21	Välisseinte vaheline liitekoha pikkus
L22	VS KL liitekoha pikkus
L23	VS pööningu VL liitekohapikkus
L24	VS PP liitekoha pikkus
L25	VS keldri VL liitekoha pikkus
L26	VS VL liitekoha pikkus

RESTO arvutuste *endpoint* teeb aga kõik arvutused ja järeldused omakorda E-ehituse

API põhjal. Tegemist on antud valdkonnas ning RESTO projekti puhul väga laialdaselt kasutatud *endpoint*-iga, mida kasutab ka näiteks 3D maa-ameti kaart. LOD2 maja mudeli andmed tulevad osakeste jadana, mis on JSON objekti kujul. Iga osake on täisnurkne kolmnurk, mille põhjal on kõik hooned üles ehitatud. Ühe sellise osakese JSON objekt sisaldab endas täisnurkse kolmnurga tipu koordinaate, kolmnurga pindala ning kolm normaali. Normaalide kaudu on võimalik leida, kas antud osake pärineb seinast, põrandast või katusest. Joonisel 15 on kujutatud, kuidas paiknevad maja mudelil osakesed, mida päringu tulemusel saadakse. Tabelis 3 on kujutatud tagastatud JSON-i parameetrid ning nende kirjeldused.



Joonis 15. LOD2 põhjal maja mudel, kus on näha osakeste paiknemist

Tabel 3. E-ehituse API-st saadud JSON objekti parameetrid ning kirjeldused

JSON võti	**Tähendus**	**LOD2 geomeetria**
etak	Topograafia infosüsteemis määratud unikaalse nähtuse kood	ETAK kood
ehr	Ehitsregistri kood	EHR kood
particles	Hoone kolmnurkadeks jagatud tahkude loend	Hoone kolmnurgad ja näitajad
area	EHR registri poolt tehtud pindala arvutus antud kolmnurgale	Kolmnurga pindala
x0	Kolmnurga esimese tipu x koordinaat	x0 koordinaat
y0	Kolmnurga esimese tipu y koordinaat	y0 koordinaat
z0	Kolmnurga esimese tipu z koordinaat	z0 koordinaat
x1	Kolmnurga teise tipu x koordinaat	x1 koordinaat
y1	Kolmnurga teise tipu y koordinaat	y1 koordinaat
z1	Kolmnurga teise tipu z koordinaat	z1 koordinaat
x2	Kolmnurga kolmanda tipu x koordinaat	x2 koordinaat

Jätkub...

Tabel 3 – *Jätkub...*

JSON võti	**Tähendus**	**LOD2 geomeetria**
y2	Kolmnurga kolmanda tipu y koordinaat	y2 koordinaat
z2	Kolmnurga kolmanda tipu z koordinaat	z2 koordinaat
nx	Kolmnurga pinnaga risti olev vektor, mis määrab pinna positiivse ja negatiivse poole	Kolmnurga normaali x komponent
ny	Kolmnurga pinnaga risti olev vektor, mis määrab pinna positiivse ja negatiivse poole	Kolmnurga normaali y komponent
nz	Kolmnurga pinnaga risti olev vektor, mis määrab pinna positiivse ja negatiivse poole	Kolmnurga normaali z komponent

Kolmandas andmestikus olid TalTech tudengid välja toonud 433 hoonet, kus iga kolmene grupp koosnes omavahel sarnastest LOD2 maja mudelitest. Andmestik oli loodud kasutades 3D maa-ameti kaarti. Nagu ka eelnevalt mainitud, koosnes valim algul vaid aadressist ning EHR-i koodist, kuid hiljem kasutades RESTO *endpoint*-i, oli võimalik andmetele lisada hoone spetsiifilisi omadusi ja arvutusi juurde. Antud andmete puhul oli tehtud järeldus, et tegu on nö. juhendamata õppeviisiga (inglise keeles *unsupervised learning*), sest kuigi oli enam vähem teada, millised majad on omavahel sarnased, ei olnud neid siiski võimalik jagada eeldefineeritud gruppidesse. Käesoleva töö raames tähendas see seda, et antud olukorras oli võimalik rakendada ainult klasterdamise masinõppe mudeleid.

Viimane andmestik, mida antud töö tegemisel kasutati, oli veneaegsete paneelmajade andmekogum. Tegemist on andmebaasiga, mis sisaldab endas kõiki nõukogudeaegsete ehitiste seeria ning projekti numbreid. Sarnaselt esimestele andmevalimitele, võis ka selle puhul kasutada nii klasterdamise kui ka klassifikatsiooni masinõppe mudeleid. Näiteks klasterdamise puhul oli võimalik kontrollida, kas mudel suudab jagada sama seeria või projekti numbriga hooned samasse klastrisse. Klassifikatsiooni puhul sai testida, kas masinõppe algoritm on võimeline õigesti ennustama vastava seeria või projektinumbri vastavalt sisendparameetritele. Sarnaselt tudengi andmestikule, oli kas siin kasutatud RESTO API-d, et kõikvõimalikud omadused ja arvutused kätte saada.

3.1.2 Mudeli valik vastavalt andmestikule

Parimate masinõppe mudelite valimisel lähtuti sellest, millise andmestikuga on tegemist ning millise õppeviisi lähenemist oleks kõige mõistlikum kasutada, kas juhitud või juhtimata. Vastavalt sellele tehti otsus, kas klassifikatsiooni või klasterdamise mudelite kasuks.

Tudengite sarnaste hoonete andmestiku puhul oli selgelt tegemist juhtimata õppeviisiga. Andmetel puudus kindel grupp kuhu hooned peaksid kuuluma ning ei olnud võimalik ka eeldada, et iga uus hoone sobituks eeldefineeritud gruppi. Seetõttu oli selle andmestiku puhul võimalik rakendada vaid klasterdamise algoritme ehk eksperimendid viidi läbi peamiselt kasutades mudeleid nagu: K-keskmine, hierarhiline klasterdamine ning Gaussi segu mudelid.

Nõukogude liidu valimis olid hooned juba jagatud eraldi seeria ja projekti järgi seega nende puhul võis arvestada, et tegu on eeldefineeritud gruppidega. See tähendas seda, et selle valimi puhul oli võimalik jooksutada nii klassifitseerimise kui ka klasterdamise algoritme. Klassifitseerimise puhul oli võimalik kasutada vastavaid seeria ja projekti numbreid sihtparameetritena, mida hoonete puhul proovitakse ennustada vastavalt arvutustele. Klasterdamise puhul olid vastavad numbrid rohkem testimise otstarbeks ning aitasid valideerida, kas antud arvutuste järgi on võimalik majad gruppeerida samadesse projektidesse nagu andmekogumis oli etteantud. Kasutati kõiki mudeleid: K-keskmine, hierarhiline klasterdamine, Gaussi segu mudelid, logistiline regressioon, otsustuspuu, juhuslik mets, SVM ning närvivõrgud.

Sarnaselt paneelmajade andmestikule oli ka Kredexi hoone valimiga võimalik eksperimenteerida mõlema masinõppe tüübiga. Andmestikuga oli kaasas ka selle tüpoloogia andmekogum, kus hooned olid juba mingil määral jaotatud erinevatesse gruppidesse vastavalt kindlatele parameetritele. Ka siin oli klasterdamise lähenemisel võimalik testida, kas moodustunud sarnaste hoonete klastrid olid sarnased hoonete tüpoloogiaga ning ühtlasi ka valideerida koostatud tüpoloogiat. Klassifitseerimise puhul oli eeldefineeritud grupp võetud sihtparameetriks, mida mudel pidi igale hoonele suutma ennustada. Ka selle andmestiku puhul olid kasutuses kõik kaheksa masinõppe mudelit.

3.1.3 Parameetrite valik, teisendamine ja agregeerimine

Mudeli valikule järgnes parameetrite valik ja andmete teisendamine ning normaliseerimine. Masinõppe algoritmid üldjuhul ei mõista sõne tüüpi väärtuseid, mistõttu on oluline teisendada sellised väärtused alguses numbriteks. Erand on ainult juhul, kui klassifitseerimise ülesande lahendamisel on sellised väärtused kasutusel sihtparameetrina, sellisel juhul suudab algoritm neid ise automaatselt hallata. Normaliseerimise eesmärk on muuta andmestiku numbriliste veergude väärtused ühisele skaalale, ilma et see moonutaks väärtusvahemike erinevusi. Üldjuhul teisendatakse väärtused nulli ja ühe vahelisele skaalale. Ühisele skaalale viimine toimus kõigi andmestike puhul kasutades samat algoritmi, MinMax skaleerimise algoritmi (valem toodud allpool), mille tööpõhimõte seisneb selles, et kõik numbrilised väärtused viiakse vahemikku nullist üheni. Parameetrite leidmine ning

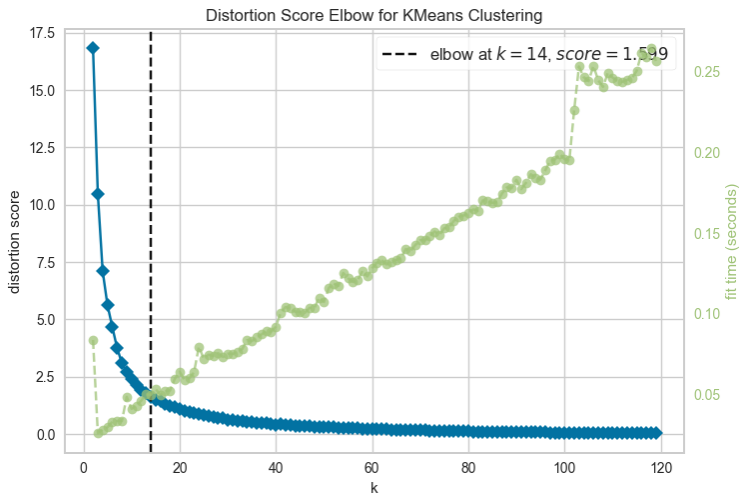
andmete teisendamine aga oli tehtud iga andmestiku puhul natukene erinevalt, olenevalt millise masinõppe ülesandega oli tegemist.

$$Xsc = \frac{X - Xmin}{Xmax - Xmin}$$

Tudengite poolt koostatud sarnaste hoonete andmestik

Tudengite sarnaste hoonete andmestiku puhul oli andmete hulk natuke väiksem ning oli ka vähem sõne tüüpi ehk nö kategoorilisi väärtusi. Ainus väärtus, mida oli vaja teisendada, oli hoone katuse tüüp, mis oli üldjuhul esitatud väärtustena: viilkatus ja lamekatus. Teisenduse tagajärjel teisendati need numbrilisteks väärtusteks, üks ja null.

Antud andmekogumi puhul oli tegemist klasterdamise ülesande lahendamisega, mistõttu oli lisaks parameetri valikule oluline leida ka kõige optimaalsem klastrite arv. K-keskmise kui ka teiste jaoks oli kasutusele võetud "küünarnuki meetod". Selle jaoks jooksutati küünarnuki meetodi funktsioon, mis võttis sisendiks klastrite arvu vahemiku ning jooksutas mudelit iga sisseantud klastrite arvuga. Tulemuseks kuvas see graafiku, kus võis välja valida kõige optimaalsem klastrite arv (vt joonis 16). Number võis vastavalt sisendparameetritele varieeruda.



Joonis 16. K-keskmisel klastrite arvu valimine vahemikust 2 kuni 120. Antud juhul on näha, et kõige optimaalsem klastrite arv on 14

Parameetrite valikul arvestati sellega, et andmed ei oleks liiga üldised. Näiteks katuse tüübi puhul võis arvata, et tegu võiks olla ühe võimaliku parameetriga, kuid hiljem leiti, et valimis on see esindatud vaid kahe väärtusena, mistõttu võib see osutuda liiga üldiseks parameetriks sarnasuse leidmisel. Parameetrite seas olid ka arvutused, mis olid tingitud hoone asukohast kaardil ning sõltusid ilmakaarest, mis tähendas, et isegi omavahel identsed

hooned olid mudeli silmis erinevad. Ka neid ei võetud arvesse. Lisaks olid mõne parameetri väärtused omavahel liiga sarnased, mistõttu arvestati vaid ühega. Kõiki neid tähelepanekuid arvestades ning mitme erineva katsetus tulemusel leiti, et kõige paremad parameetrid olid:

- Ehitusaluse pinna välisnurkade arv - Näitas enam-vähem ära milline on maja kuju;
- Ehitusaluse pinna perimeeter;
- Fassaadi pindala;
- Katuse pindala;
- Pinnasel põranda pindala.

Nõukogudeaegsete paneelmajade andmekogu

Käesolev andmestik oli oma loomult väga sarnane tudengite andmestikuga, sest kõik saadud arvutused olid samad. Selle tõttu olid ka valitud parameetrid klasterdamisel identsed eelneva andmestikuga. Erinevus oli klassifikatsiooni puhul. Eelkõige oli antud valimi puhul võimalik valida sihtparameeter, mida mudel vastavalt sisendparameetritele ennustama pidi. Selleks oli valitud seeria ja projekti number, mis oli teisendatud täisarvu kujule sõne asemel. Sisendiks kasutati nii samu parameetreid, mis klasterdamise puhul kui ka kõik ülejäänud 24 arvutust, et võrrelda mudeli erinevust.

Tüpoloogia ja kredexi hoonete andmestik

Kredexi hoonete puhul oli andmeid palju rohkem ning seetõttu oli ka andmete töötlemine palju põhjalikum ning aeganõudvam. Esiteks oli oluline lisada tüpoloogiale vastav välisseina grupp ning teisendada see numbriks. Samamoodi oli tehtud ka ajavahemikuga mis näitas hoone ehitusaastat. Järgmine etapp oli andmete normaliseerimine ning mitteoluliste andmete eemaldamine. Klasterdamise puhul võeti sisendparametriteks hoone atribuudid, millega olid loodud ka tüpoloogia grupid. Nendeks olid:

- Ajastu ehk vahemik millal hoone ehitati;
- Hoone välisseina liik;
- Trepikodade arv;
- Maksimaalne korruste arv;
- Lisati ka akna ja seina suhe.

Selline valik oli tehtud seetõttu, et antud andmestiku puhul oli põhiliseks eesmärgiks tüpoloogia valideerimine ning sarnaste gruppide loomine. Klassifikatsiooni korral oli sihtparameetriks võetud hoonete tüpoloogiate grupid, mida mudel pidi ennustama ning sisendiks oli võetud kõik Kredexi hoone andmestikus olevad parameetrid. See tegi kokku ligikaudu 34 atribuuti.

3.1.4 Treening- ja testbaasid ning tulemuste valideerimine

Treening- ja testbaasi jaotamine, mis on oluline protsess eriti klassifikatsiooni puhul, oli tehtud hea tava põhjal. Algbaas oli juhuslikult jaotatud kaheks osaks. 80% algbaasist oli kasutatud treeningbaasina ning ülejäänud 20% oli kasutatud testbaasina. Klasterdamise ülesannete jaoks oli valitud nö eeldefineeritud grupp, mille järgi tulemusi valideeriti ja analüüsiti. Paneelmajade andmekogumi puhul oli selleks seerianumber ja kredexi hoonete puhul oli tüpoloogia grupp. Tulemuste võrdlemisel vaadeldi, kas tekkinud klastrite struktuur klappis eeldefineeritud gruppide omaga. Tudengite poolt valitud sarnaste hoonete andmestiku puhul oli teada, et iga hoonete kolmik alustades algusest, koosneb omavahel visuaalselt sarnastest majadest. See oli võetud arvesse ka andmete töötluse jooksul ning selle põhjal oli loodud uus veerg, kus kõik hooned olid paigutatud kolme kaupa gruppidesse. Selle põhjal valideeriti klasterdamise tulemusi viimase valimi puhul.

3.1.5 Mudelite treenimine ja testimine

Masinõppe mudelite koostamiseks kasutati Pythoni programmeerimiskeelt. Selle põhjuseks oli see, et tegu on üsna laialdaselt kasutatud keelega antud valdkonnas ning ka töö autori varasem kogemus sarnastel teemadel põhineb samuti Pythonil. Lisaks on sellel keelel mitmeid erinevaid lisateeke, mis sarnaste probleemide lahendamise teeb palju kergemaks ja kiiremaks. Põhilised teegid, mis olid antud töö käigus kasutusel:

- Numpy - Andmemassiivide haldamiseks ja töötlemiseks;
- Pandas - Sarnaselt Numpy-le andmemassiivide haldamiseks ja töötlemiseks mõeldud teek;
- Scikit-learn - Üks populaarsemaid Pythoni masinõppe "raamatukogusid". Sisaldab endas suurt kogust erinevaid masinõppe meetodeid ja andmestikke.

Masinõppe algoritmide treenimise ja testimise peamised etapid olid järgmised:

- Andmekogumi laadimine Pythonisse;
- Vajadusel lisaarvutuste lisamine;
- Andmete eeltöötlus;
- Sisend- ja sihtparameteetrite defineerimine;
- Andmete jagamine treening- ja testandmeteks;
- Kõigi algoritmide treenimine ja vajadusel sisendite ning mudeli parameetrite korrigeerimine;
- Mudelite valideerimine testandmetel ning tulemuste graafikute ja tabelite loomine.

3.2 Eksperimendi tulemused

Käesolev alapeatükk annab detailsema ülevaate erinevatest eksperimendi tulemustest ning on omakorda jaotatud andmestike järgi järgmisteks alaosadeks:

- Tudengite sarnaste hoonete andmestiku klasterdamise tulemused;
- Nõukogudeaegsete paneelmajade klasterdamise ja klassifikatsiooni tulemused;
- Kredexi hoonete ja tüpoloogia andmestiku klasterdamise ja klassifikatsiooni tulemused.

3.2.1 Tudengite sarnaste hoonete andmestik

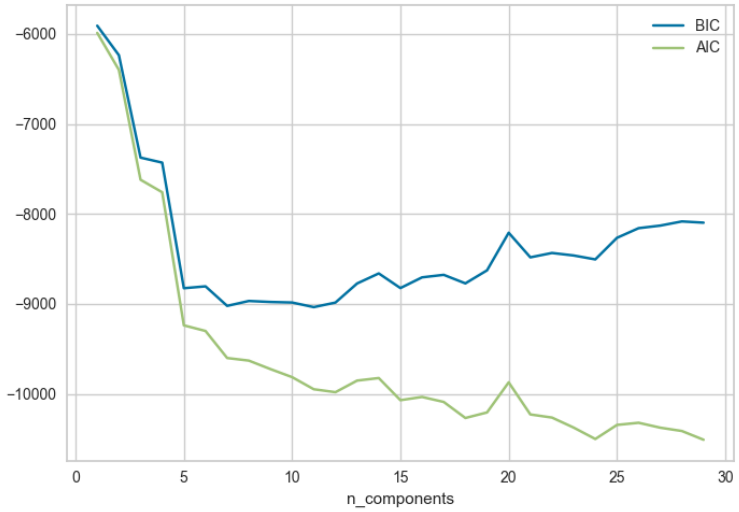
Tudengite poolt koostatud sarnaste andmestikuga oli jooksutatud ainult klasterdamise eksperimente. Tulemuste võrdlemiseks ja/või kontrollimiseks oli jooksutatud kood, mis jagas iga tudengi 12 hoonet omakorda kolmestesse gruppidesse. Selle eesmärk oli luua vähemalt mingisugune indikaator sarnaste majade tuvastamiseks, ilma et peaks iga hoone korral kontrollima 3D maa-ameti kaarti, mille põhjal andmestik koostatud oli. Sellegi poolest oli oluline valideerida osad tulemused ka kasutades kaarti.

Kõik mudelid olid jooksutatud kasutades mitmeid erinevaid atribuute ning parimateks kõigi kolme puhul osutusid juba eelnevalt mainitud:

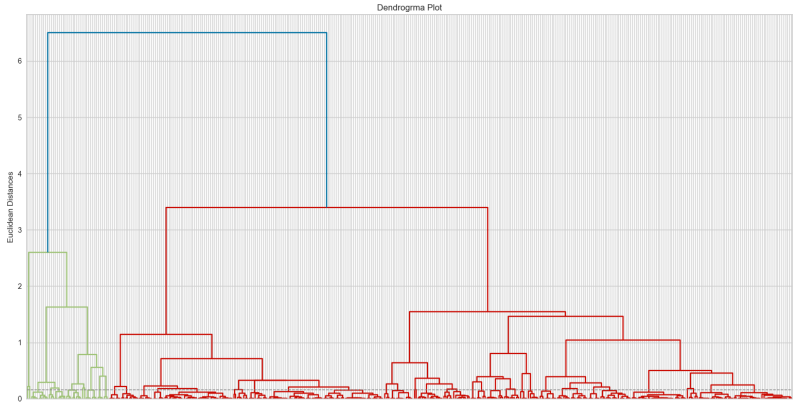
- Ehitusaluse pinna sisenurkade arv;
- Ehitusaluse pinna perimeeter;
- Fassaadi pindala;
- Katuse pindala;
- Pinnasel põranda pindala.

K-keskmise puhul leiti, et kõige optimaalsem klastrite arv on ligikaudu 14 (vt joonis 16). Gaussi segu mudelite puhul leiti, et kõige optimaalsem komponentide arv on 12 (vt joonis 17). See oli arvutatud visualiseerides kahte graafikut, Akaike teabekriteeriumit (AIC *Akaike Information Criterion*) ja Bayesi teabekriteeriumit (BIC *Bayes Information Criterion*) ning leides nende kahe kõige madalama punkti kombinatsiooni. Hierarhilise puhul oli loodud dendrogram (vt joonis 18), mille põhjal oli vaadatud hoonete jagunemist erinevatesse gruppidesse ning vastavalt sellele võetud sobiv klastrite arv eksperimenteerimiseks.

Valideerimine toimus kolmikute põhjal, mille moodustasid tudengite poolt valitud kolm järjestiku paigutatud hoonet. Selle puhul oli oluline tähele panna , et omavahel sarnased



Joonis 17. Optimaalsete komponentide leidmine Gaussi segu mudelite jaoks



Joonis 18. Hierarhilise klasterdamise käigus tekkinud dendrogram

hooned võisid tegelikult asuda täiesti erinevates gruppides (näiteks juhul kui kaks erinevat tudengit olid pannud lähestikku asuvad sarnased hooned). See oli ka hea viis klasterdamise mudeli testimiseks, et vaadelda, kas kahe erineva tudengi hoonegrupid, mis on visuaalselt sarnased, paigutatakse samasse klastrisse.

Hierarhiline klasterdamine

Hierarhilise klasterdamise puhul valideeriti tulemusi jooksutades mudelit kolme erineva klastri numbriga: K-keskmise optimaalse klastrite arvuga ehk 14, Gaussi optimaalse komponentide arvuga ehk 12 ning dendrogrammi vaatluse tulemusel otsustati jooksutada ka suurema klastrite arvuga, milleks oli 57 klastrit ehk suhteliselt maksimaalne klastrite arv, mida dendrogrammilt võis välja lugeda. Lisaks oli suurema klastrite arvuga võimalik detailsemalt valideerida tudengite valitud sarnaseid hooneid.

Kaheteistkümne klastri puhul paigutas hierarhiline mudel valesti üheksa hoonet. See

tähendas seda, et kolmest hoonest, mis tudengi silmis olid visuaalselt sarnased, üks ei olnud mudeli jaoks sarnane. Kokku tekkis selline olukord üheksal korral. Sellest võib järeldada, et üldpildis oli mudel klasterdanud 390-st hoonest üheksa valesti. Neljateistkümne ehk K-keskmise optimaalse klastrite arvuga oli selliseid vigaselt paigutatud hooneid samuti üheksa. 57 klastri puhul oli vigade arv kaks korda suurem, keskmiselt 18 kolmiku korral ei sattunud üks hoone samasse gruppi, kuhu kaks ülejäänud sarnast hoonet. Tulemused on toodud välja tabelis 4.

Tabel 4. Hierarhilise klasterdamise eksperimendi tulemused. Vigade arv vastavalt klastrite arvule

Klastrite arv	**Vigade arv**
12	9
14	9
57	18

Gaussi segu mudelid

Gaussi segu mudelitele läheneti samade klastrite arvuga ning oli püstitatud ka hüpotees, et 12 klastri puhul on antud mudelil kõige paremad tulemused, sest tegu oli meetodi optimaalse komponendi arvuga. Sellele vaatamata oli mudel iga komponentide numbri korral teisel kohal. Kaheteist klastri puhul oli mudel paigutanud kümnel korral ühe hoone ülejäänud kahest eraldi klastrisse. Neljateistkümne klastri puhul oli vigade arvuks 14 ning palju suurema klastrite arvu puhul ehk 57 puhul, oli vigaselt paigutatud hooneid 22. Gaussi mudeli tulemused on näidatud tabelis 5.

Tabel 5. Gaussi segu mudelite eksperimendi tulemused. Vigade arv vastavalt klastrite arvule

Klastrite arv	**Vigade arv**
12	10
14	14
57	22

K-keskmine

Eksperimendi alguses oli püstitatud hüpotees, et k-keskmise puhul on tulemused keskmiselt kõige paremad, kuid see oli peamiselt tingitud mudeli populaarsusest ning autori varasemast kogemusest mudeliga. Teiseks hüpoteesiks, sarnaselt Gaussi segu mudelitele, oli see, et 14 klastri puhul on k-keskmise tulemused kõige paremad, sest tegu on algoritmi optimaalse klastrite arvuga. Reaalsuses olid mudeli tulemused kõige kehvemad kõigil

kolmel korral. Gaussi optimaalse klastrite arvu puhul oli K-keskmine paigutanud üheteistkümnel korral valesti. Neljateist klastri puhul oli vigade arv 12 ning 57 klastri korral oli 25 viga. Tabelis 6 on võimalik näha K-keskmise tulemusi.

Tabel 6. K-keskmise eksperimendi tulemused. Vigade arv vastavalt klastrite arvule

Klastrite arv	**Vigade arv**
12	11
14	12
57	25

Kokkuvõtlikult võib järeldada, et klasterdamise masinõpe mudelid on võimelised hooneid kindlate parameetrite järgi sarnasuse põhjal grupeerima. Samuti oli leitud, et vähemalt valitud kolme mudeli põhjal, saab hierarhiline klasterdamine selle ülesandega kõige paremini hakkama.

3.2.2 Nõukogudeaegsete paneelmajade andmestik

NSVL paneelmajade andmestikul oli jooksutatud nii klasterdamise kui ka klassifikatsioone eksperimente. Tulemuste võrdluseks oli kasutatud hoonete projekti ja seerianumbreid. Klasterdamise puhul võeti klastrite arvuks projektide ja seeriate arvu, milleks oli 11, sest eesmärk oli näha, kas mudel suudab jaotada hooned samamoodi klastritesse. Klassifitseerimisel oli eeldefineeritud projekti number võetud sihtparameetriks, mille mudel pidi hoonele ennustama.

Klasterdamine

Käesoleva andmestiku puhul oli kõikide mudelite puhul võetud klastrite arvuks projekti numbrite arv ehk 11. Eesmärk oli näha, kas mudel suudab jaotada hooned samamoodi gruppidesse. Ka parameetriteks olid valitud mitmed erinevad kombinatsioonid, kuid tulemuste vahel suuri erinevusi ei olnud.

Kõik kolm mudelit tagastasid enam vähem sarnased tulemused. Kõik kolm suutsid paigutada hooned, millel oli spetsiifiline projekti number olemas, näiteks 1-464A-12, samasse klastrisse. Küll aga üldisema projekti numbri puhul, näiteks 1-464, vastavat klastrit ei tekkinud ning selle grupi hooned olid keskmiselt jaotatud kolme klastri vahel. Hiljem andmete ajaloo uurimisel ja RESTO projekti poolse kaasjuhendajaga konsulteerimisel selgus, et sellised erinevused võisid olla tingitud sellest, et tegu on hoonete eri generatsioonidega. Vanematel hoonetel oli rajamise ajal olemas vaid seerianumber, mille järgi neid

paigutati, näiteks 1-464. Järgmise generatsiooniga muutus seerianumber spetsiifilisemaks ning tekkis konkreetsem projekti number, näiteks 1-464D. Lõpuks kolmanda generatsiooniga läksid projekti numbrid veelgi spetsiifilisemaks ning tekkis näiteks 1-464A-17. Ka visuaalse valideerimise käigus oli näha, et üldisema grupi puhul võisid hooned olla vägagai erinevad, kuid mida spetsiifilisemaks läks projekti number, seda rohkem sarnanesid hooned omavahel visuaalselt. See võis seletada ka antud klasterdamise tulemusi.

tyyp	aadress	k_means...	Hierarchial_cl...	gaussian_...
1-464A-17	Keskuse, 12	8	8	8
1-464A-17	Keskuse, 14	8	8	8
1-464A-17	Keskuse, 22	8	8	8
1-464A-17	Keskuse, 20	8	8	8
1-464A-14	Eduard Vilde tee, 104	7	4	4
1-464A-14	Mustamäe tee, 102	7	4	4
1-464A-14	Puhangu, 53	7	4	4
1-464A-13	Tedre, 77	5	2	10
1-464A-13	Tedre, 81	5	2	10
1-464A-13	Tedre, 85	5	2	10
1-464A-13	Valdeku, 116	9	2	1
1-464A-13	Koskla, 3	5	2	10
1-464A-13	Kuldnoka, 8	5	2	10
1-464A-13	Kuldnoka, 9	5	2	10
111-121	Tallinn,Arbu 13	9	2	1
1-404-3	Kopli tänav 69E Tallin...	3	0	9
1-432	Endla tänav 88	5	2	1
1-432	Endla tänav 90	5	2	1
1-432	Jakobi tänav 19	0	5	7
1-432	Juurdeveo tänav 10	9	2	1
1-432	Kiisa tänav 7	9	2	1
1-432	Kiisa tänav 9	9	2	1
1-432	Pae tänav 31	0	5	7
1-432	Pae tänav 33	0	5	7
1-432	Pae tänav 48	0	5	7
1-432	Pae tänav 50	0	5	7
1-432	Pae tänav 52	0	5	7

Joonis 19. NSVL paneelmajade klasterdamise tulemus

Joonis 19 illustreerib osa tulemustest, kus esimene tulp on hoone tüüp, mille väärtuseks on seeria või projekti number. Selle järel aadress ning seejärel tulevad kolme meetodi loodud klastrid. Sealt saab näha, et esimese kolme tüübi puhul on suutnud mudel paigutada kõik hooned vastavalt kolme klastrisse. Küll aga üldisema seeria puhul, ei ole mudel suutnud leida sellele vastava kindla klastri ning on hooned jaotanud kolme eri klastri vahel.

Klassifitseerimine

Klassifitseerimisel oli sihtparameetriks valitud hoone seerianumber, eesmärgil, et mudel on võimeline korrektselt hoone projekti ennustama. Sisendiks olid valitud kõik ülejäänud parameetrid, sest sellisel moel näitasid mudelid kõige paremaid tulemusi.

Iga mudeli treenimisel oli jooksutatud ka eraldi funktsioon, mis aitas leida, millised peaksid olema mudeli enda parameetrid, et see tagastaks kõige paremad tulemused.

Logistilise regressiooni puhul oli mudeli parim keskmine ennustuse täpsus 61% (+/- 19.8%). Üle kõikide sihtgruppide, oli mudeli keskmine positiivne ennustusvõime 0.7 (vt table 7). See näitas, kui suur osa kõikidest hoonetest, millele mudel ennustas grupi X,

päriselt ka kuulusid gruppi X. Tundlikkus ehk õige positiivsete määr oli keskmiselt 0.65. See omakorda näitas, kui suurele osale kõikidest hoonetest, mis päriselt pidid kuuluma gruppi X, suutis mudel ennustada korrektse grupi. Näiteks grupi 0 puhul oli tundlikkus 100%, kuid positiivne ennustusvõime vaid 67%. Sellest võime järeldada, et mudel ennustas üle ehk ta paigutas gruppi 0 kõik need hooned, mis päriselt kuulusid sinna ning ka need, mis sinna tegelikult ei pidanud kuuluma. Viimane näitaja, mida vaadeldi oli F1 skoor, mis antud mudeli puhul oli keskmiselt 0.61. Selle järgi võime järeldada, et tegu ei ole ei halva ega ka hea mudeliga ning üldjuhul suudab mudel näidata paremat tulemust kui juhuslikult klassifitseerimine.

Tabel 7. Logistilise regressiooni tulemused

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
0	0.67	1.00	0.80
2	1.00	0.60	0.75
3	0.60	0.67	0.63
4	0.33	1.00	0.50
5	0.00	0.00	0.00
7	1.00	0.40	0.57
8	1.00	0.50	0.67
9	1.00	1.00	1.00

Otsustuspuu puhul oli täpsus kõrgem, kui logistilise regressiooni mudeli puhul. Kõrgeim keskmine mudeli ennustustäpsus oli 74% (+/-25%). Mudeli keskmine positiivne ennustusvõime (vt table 8) oli 0.64 ehk 64% ning tundlikkus 67%. F1 skoor oli sellel mudelil 0.61 ehk täpselt samasugune, mis ka logistilise regressiooni puhul.

Tabel 8. Otsustuspuu tulemused

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
0	0.50	1.00	0.67
2	1.00	0.40	0.57
3	0.80	0.44	0.57
4	0.50	1.00	0.67
5	0.00	0.00	0.00
7	0.83	1.00	0.91
8	0.50	0.50	0.50

Jätkub...

Tabel 8 – *Jätkub...*

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
9	1.00	1.00	1.00

Juhuslik mets oli samuti väga lähedal otsustuspuu tulemustele ning kohati oli isegi parem (vt table 9). Kõrgeim keskmine täpsus ennustamisel oli selle algoritmi puhul 73.3% (+/- 21%). Keskmiselt 68% mudeli ennustustest klappisid päris andmetega ning mudeli keskmine tundlikkus oli 66%. F1 skoor oli sellel meetodil 0.66, mis näitab, et juhuslik mets peaks olema võrreldes ülejäänud mudelitega kõige parem ning stabiilsem grupi ennustamisel.

Tabel 9. Juhusliku metsa tulemused

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
0	0.67	1.00	0.80
2	1.00	0.60	0.75
3	0.75	0.67	0.71
4	1.00	1.00	1.00
5	0.00	0.00	0.00
7	1.00	1.00	1.00
8	0.00	0.00	0.00
9	1.00	1.00	1.00

SVM ehk tugivektor-masinate mudeli tulemused olid peaaegu halvimad ning kohati sarnanesid logistilise regressiooni tulemustega. Kõrgeim keskmine täpsus oli 68% (+/- 14%). Keskmine positiivne ennustusvõime sealjuures vaid 0.36, tundlikkus 0.42 ning F1 skoor oli 0.38. Nende näitajate järgi oli SVM kõige halvem mudel antud sihtparameetri ennustamiseks ning üleüldiselt antud andmestiku jaoks. Sellest võib teha ka järelduse, et tihtipeale võib juhuslik ennustamine olla isegi täpsem kui antud mudeli kasutamine.

Tabel 10. SVM tulemused

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
0	0.50	1.00	0.67
2	1.00	0.80	0.89

Jätkub...

Tabel 10 – *Jätkub...*

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
3	0.38	0.56	0.45
4	0.00	0.00	0.00
5	0.00	0.00	0.00
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	1.00	1.00	1.00

Viimane mudel, mille peal jooksutati eksperimente, olid närvivõrgud. Närvivõrkude kõrgeim keskmine ennustustäpsus oli umbes samal tasemel, mis otsustuspuu ja juhusliku metsa mudelitel, 74% (+/- 17%). Ülejäänud näitajate järgi jäi mudel teiste seas pigem keskele. Positiivne ennustusvõime oli keskmiselt 49%, tundlikkus jäi 55% juurde ning F1 skoor oli täpselt 0.5. Selle põhjal võib järeldada, et mudel on täpselt keskel oma ennustusvõime poolest ning on tõenäoline, et võib võib jääda alla juhuslikule klassifitseerimisele. Detailsemad tulemused on toodud välja tabelis 11.

Tabel 11. Närvivõrkude tulemused

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
0	0.43	0.75	0.55
2	0.75	0.60	0.67
3	0.50	0.44	0.47
4	0.50	1.00	0.67
5	0.00	0.00	0.00
7	0.75	0.60	0.67
8	0.00	0.00	0.00
9	1.00	1.00	1.00

Kokkuvõteks võib järeldada, et sarnaselt klasterdamisele, mängis ka klassifikatsiooni puhul tõenäoliselt väga suurt rolli see, kuidas ajalooliselt olid need hooned projektidesse jaotatud. Lisaks sellele oli tegemist liiga väikse andmestikuga (treeningbaas - umbes 120, testbaas - umbes 30), et kasutada ära mudelite kogu potentsiaali. Samas on näha, et ka väiksemate andmekoguste puhul on kõik mudelid võimelised mingil määral hooned õigesti ära jaotama.

3.2.3 Kredexi hooned ja tüpoloogia

Kredexi hoonete andmestiku peal jooksutati nii klasterdamise meetodeid kui ka klassifitseerimist. Antud andmestiku põhjal loodud tüpoloogia tabel oli peamiselt kasutusel mudeli täpsuse valideerimiseks.

Klasterdamine

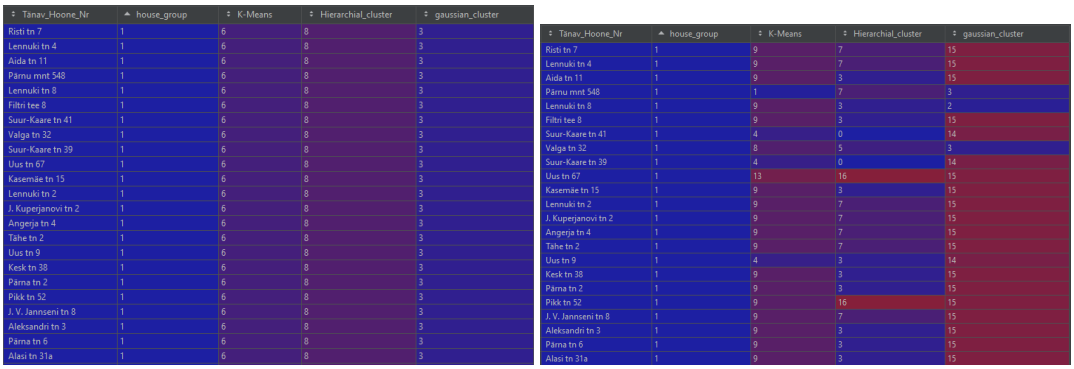
Klasterdamise puhul oli eesmärgiks kasutada kõiki samu parameetreid, mille põhjal oli loodud hoonete tüpoloogia. Atribuutideks olid: ajastu ehk vahemik, millal hoone ehitati, hoone välisseina liik, trepikodade arv, maksimaalne korruste arv ning juurde oli lisatud ka akna ja seina suhe, mis hüpoteesi alusel pidi olema hea parameeter hoone sarnasuse hindamisel. Tüpoloogia andmestikus oli kokku loodud 21 erinevat gruppi, mis ühtlasi määrati ka klastrite arvuks.

Esimese korraga olid tulemused väga erinevad võrreldes sellega, mida näitas tüpoloogia andmestik ning väga vähesed referentshooned klappisid määratud uue klastriga. Kiiresti jõuti ka järeldusele, et probleem võib olla tingitud sellest, et ajastu ja hoone välisseina liigi parameetrite puhul, on tegemist nominaalatribuutidega ehk nende puhul puudub kindel järjestus. Seetõttu oli tehtud teine katsetus, kuid seekord oli lisatud üks lisaetapp andmete teisendamisele ning igale nominaalkategooria jaoks oma atribuut väärtusega null või üks. Võtame näiteks välisseina liigi, mida oli kokku neli. Selle põhjal oli tekitatud neli lisa veergu ning iga rea väärtus näitas kas hoone välissein on tüüpi X (väärtuseks 1) või ei ole (väärtuseks 0). Ja nii iga liigi korral.

Uue teisendamise järel olid tulemused palju täpsemad. Kui enne ei suutnud mudel ühtegi eeldefineeritud grupile vastavat klastrit luua, siis teisel korral oli seis parem (võrdlus kahe tulemuse vahel on näha joonisel 20). Mudel suutis defineerida neli samasugust klastrit ning kõik teised hooned, mis pidid kuuluma ühte gruppi, olid üldjuhul jaotatud kahte klastrisse. Näiteks joonise 21 põhjal on näha kuidas hoonete eeldefineeritud grupile number neli oli loodud kõigi kolme mudeli poolt vastav klaster. Samas on näha ka seda, et grupis number viis olevad hooned, olid masinõpe mudeli tulemusel jaotatud hoopis kahe klastri vahel.

Klassifitseerimine

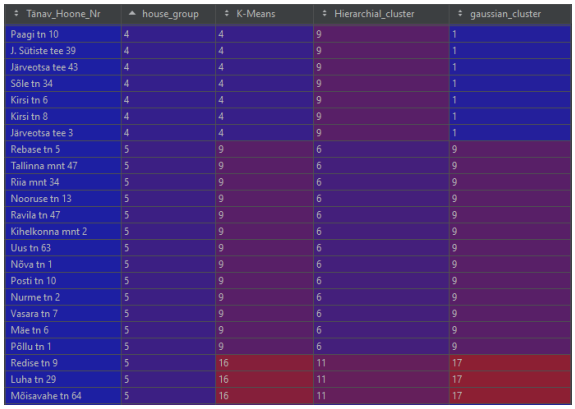
Klassifitseerimise käigus oli sihtparameetriks valitud hoonete tüpoloogia grupid ning siendandmeteks 34 hoone atribuuti. Eesmärgiks oli vastavalt sisendile ennustada hoone korrektne tüpoloogia grupp ning leida, milline mudel saab sellega kõige paremini hakkama. Ka antud eksperimendi puhul olid mudelid jooksutatud erinevate mudeli parameetritega ning tulemused on saadud kasutades kõige optimaalseid muutujaid.



Tänav_Hoone_Nr	house_group	K-Means	Hierarchial_cluster	gaussian_cluster
Risti tn 7	1	6	8	3
Lennuki tn 4	1	6	8	3
Aida tn 11	1	6	8	3
Pärnu mnt 548	1	6	8	3
Lennuki tn 8	1	6	8	3
Filtri tee 8	1	6	8	3
Suur-Kaare tn 41	1	6	8	3
Valga tn 32	1	6	8	3
Suur-Kaare tn 39	1	6	8	3
Uus tn 67	1	6	8	3
Kasemäe tn 15	1	6	8	3
Lennuki tn 2	1	6	8	3
J. Kuperjanovi tn 2	1	6	8	3
Angerja tn 4	1	6	8	3
Tähe tn 2	1	6	8	3
Uus tn 9	1	6	8	3
Kesk tn 38	1	6	8	3
Pärna tn 2	1	6	8	3
Pikk tn 52	1	6	8	3
J. V. Jannseni tn 8	1	6	8	3
Aleksandri tn 3	1	6	8	3
Pärna tn 6	1	6	8	3
Alasi tn 31a	1	6	8	3

Tänav_Hoone_Nr	house_group	K-Means	Hierarchial_cluster	gaussian_cluster
Risti tn 7	1	9	7	15
Lennuki tn 4	1	9	7	15
Aida tn 11	1	9	3	15
Pärnu mnt 548	1	1	7	3
Lennuki tn 8	1	9	3	2
Filtri tee 8	1	9	3	15
Suur-Kaare tn 41	1	4	0	14
Valga tn 32	1	8	5	3
Suur-Kaare tn 39	1	4	0	14
Uus tn 67	1	13	16	15
Kasemäe tn 15	1	9	3	15
Lennuki tn 2	1	9	7	15
J. Kuperjanovi tn 2	1	9	7	15
Angerja tn 4	1	9	7	15
Tähe tn 2	1	9	7	15
Uus tn 9	1	4	3	14
Kesk tn 38	1	9	3	15
Pärna tn 2	1	9	3	15
Pikk tn 52	1	9	16	15
J. V. Jannseni tn 8	1	9	7	15
Aleksandri tn 3	1	9	3	15
Pärna tn 6	1	9	3	15
Alasi tn 31a	1	9	3	15

Joonis 20. Vasakul on tulemus kui nominaalatribuudid on teisendatud. Paremal on tulemused kui nominaalatribuudid ei ole teisendatud



Tänav_Hoone_Nr	house_group	K-Means	Hierarchial_cluster	gaussian_cluster
Paagi tn 10	4	4	9	1
J. Sütiste tee 39	4	4	9	1
Järveotsa tee 43	4	4	9	1
Sõle tn 34	4	4	9	1
Kirsi tn 6	4	4	9	1
Kirsi tn 8	4	4	9	1
Järveotsa tee 3	4	4	9	1
Rebase tn 5	5	9	6	9
Tallinna mnt 47	5	9	6	9
Riia mnt 34	5	9	6	9
Nooruse tn 13	5	9	6	9
Ravila tn 47	5	9	6	9
Kihelkonna mnt 2	5	9	6	9
Uus tn 63	5	9	6	9
Nõva tn 1	5	9	6	9
Posti tn 10	5	9	6	9
Nurme tn 2	5	9	6	9
Vasara tn 7	5	9	6	9
Mäe tn 6	5	9	6	9
Põllu tn 1	5	9	6	9
Redise tn 9	5	16	11	17
Luha tn 29	5	16	11	17
Mõisavahe tn 64	5	16	11	17

Joonis 21. Kredexi hoonete klasterdamise tulemus

Logistilise regressiooni eksperimendi käigus selgus, et mudeli kõige suurimaks keskmiseks täpsuseks oli 85% (+/-15%). Antud algoritmi keskmine positiivne ennustusvõime oli 0.61 ehk keskmiselt kuulus kõikidest ennustatud hoonetest ainult 61% ennustatud gruppi. Tundlikkus oli mudelil 60% ning F1 skoor 0.59. Selle põhjal võime järeldada, et mudel oli pigem keskmise võimsusega ning ei olnud juhuslikust paigutamisest palju parem. Tabelis 12 on näha tulemusi gruppide põhjal.

Tabel 12. Logistilise regressiooni tulemused

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
1	0.86	0.75	0.80
2	0.93	0.97	0.95
3	0.93	0.88	0.90
5	0.75	1.00	0.86
6	0.00	0.00	0.00
7	1.00	1.00	1.00

Jätkub...

Tabel 12 – *Jätkub...*

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
8	0.00	0.00	0.00
9	0.00	0.00	0.00
10	1.00	1.00	1.00
11	0.80	1.00	0.89
12	0.95	0.95	0.95
13	0.50	0.50	0.50
14	0.00	0.00	0.00
15	1.00	0.50	0.67
16	1.00	1.00	1.00
18	0.00	0.00	0.00

Tabelis 13 on toodud välja tulemused otsustuspuu eksperimentidest. See mudel oli üks täpsemaid ning näitas peaagu, et parimaid tulemusi, kuid jäi oma stabiilsuse poolest juhuslikule metsale alla. Otsustuspuu suurim keskmine täpsus oli 95% (+/- 9%). Keskmine positiivne ennustusvõime oli 0.73, tundlikkus oli 0.7 ning F1 skoor näitas hindeks 0.71. Selle järgi võib järeldada, et mudelist on mingil määral kasu ning päris suure tõenäosusega suudab mudel hoonet õigesti liigitada.

Tabel 13. Otsustuspuu tulemused

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	1.00	1.00	1.00
5	1.00	1.00	1.00
6	0.00	0.00	0.00
7	0.00	0.00	0.00
8	1.00	1.00	1.00
10	0.00	0.00	0.00
11	1.00	0.50	0.67
12	0.93	1.00	0.96
13	1.00	1.00	1.00

Jätkub...

Tabel 13 – *Jätkub...*

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
15	0.00	0.00	0.00
16	1.00	0.50	0.67
18	1.00	1.00	1.00

Juhuslik mets näitas otsustuspuuga kõige paremaid tulemusi, kuid oli oma tulemuste poolest siiski natukene stabiilsem. Seetõttu võis järeldada, et juhuslik mets oli kõige parem mudel antud eksperimentide seast. Meetodi suurim keskmine täpsus oli 94% (+/- 4%). Keskmine positiivne ennustusvõime oli 0.82. Tundlikkus oli ligikaudu 0.86 ning F1 skoor oli 0.84. Nende tulemuste põhjal võis järeldada, et antud mudelitest kõige edukamaks osutus juhuslik mets, sest algoritmi täpsus ei langenud kunagi alla 90%. Lisaks sellele näitas ka F1 skoor väga kõrget tulemust. Detailsemad tulemused toodud välja tabelis 14.

Tabel 14. Juhusliku metsa tulemused

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	1.00	1.00	1.00
5	1.00	1.00	1.00
6	0.00	0.00	0.00
7	1.00	1.00	1.00
8	1.00	1.00	1.00
10	1.00	1.00	1.00
11	0.60	1.00	0.75
12	0.92	1.00	0.96
13	1.00	1.00	1.00
15	0.00	0.00	0.00
16	1.00	0.50	0.67

SVM mudel oli parem kui logistiline regressioon, kuid siiski jäi otsustuspuule ja juhuslikule metsale oma tulemuste poolest alla. Suurim keskmine täpsus antud mudeli puhul 89% (+/- 12%). Mudeli keskmine positiivne ennustusvõime oli 0.69 ning tundlikkus samuti 0.69%. F1 skoor oli 0.69. Selle järgi on näha, et mudel on tegelikult väga lähedal oma stabiilsuse,

täpsuse ning jõudluse poolest otsustuspuule ja juhuslikule metsale, kuid selgelt parem, kui logistiline regressioon. Detailsemad tulemused antud eksperimendist on toodud välja tabelis 15.

Tabel 15. SVM tulemused

Tüüp	**Positiivne ennus-tusvõime**	**Tundlikkus**	**F1**
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	0.80	1.00	0.89
4	1.00	0.50	0.67
5	1.00	0.71	0.83
6	0.00	0.00	0.00
7	0.00	0.00	0.00
8	1.00	1.00	1.00
10	0.00	0.00	0.00
11	0.60	1.00	0.75
12	0.96	1.00	0.98
13	1.00	1.00	1.00
15	0.33	0.50	0.40
16	1.00	1.00	1.00

Närvivõrkude tulemused olid väga sarnased SVM tulemustele ning kohati olid isegi paremad, kuid siiski jäi otsustuspuule ja juhuslikule metsale oma täpsuse ja jõudluse poolest alla. Suurim keskmine ennustustäpsus oli 85% (+/- 12%). Keskmine positiivne ennustusvõime oli 0.77. Tundlikkus oli madalam, keskmiselt 0.67. Nende põhjal arvutatud F1 skoor oli 0.7. Sellest võib järeldada, et närvivõrgud olid juhusliku metsa ja SVM-i vahepeal. Üksikud tulemused iga ennustatud grupi kohta on toodud välja tabelis 16.

Tabel 16. Närvivõrkude tulemused

Tüüp	**Positiivne ennus-tusvõime**	**Tundlikkus**	**F1**
1	1.00	0.67	0.80
2	0.98	1.00	0.99
3	1.00	1.00	1.00
4	0.75	1.00	0.86

Jätkub...

Tabel 16 – *Jätkub...*

Tüüp	**Positiivne ennustusvõime**	**Tundlikkus**	**F1**
5	1.00	0.33	0.50
6	0.00	0.00	0.00
9	0.00	0.00	0.00
10	1.00	1.00	1.00
11	1.00	1.00	1.00
12	0.80	1.00	0.89
13	1.00	0.60	0.75
14	1.00	0.50	0.67
15	0.00	0.00	0.00
16	1.00	1.00	1.00
18	1.00	1.00	1.00

Eksperimentide tulemuste pealt on näha, et sarnaselt NSVL andmekogule, näitas ka tüpoloogia ennustamise puhul kõige paremaid tulemusi otsustuspuu ja juhuslik mets. Seekord oli erinevuseks see, et juhuslik mets oli teistest palju parem tüpoloogia ennustamisel ning ülejäänud mudelid peale logistilise regressiooni näitasid enam vähem sarnaseid tulemusi. Siiski oli märgata ka seda, et andmestik ei olnud paraku piisavalt suur, et võimalikult täpselt kõiki mudeleid hinnata. Seda oli näha näiteks selles, et testandmed ei jagunenud piisavalt hästi, näiteks mingisugusesse gruppi kuuluvaid hooneid oli palju rohkem kui teistes gruppides. Lisaks sellele viitas liigselt väiksele andmestikule ka see, et mõningad grupid puudusid tulemuste tabelist. See tähendas seda, et genereeritud testbaas ei sisaldanud endas ühtegi sellist gruppi. Näiteks tabelil 16 oli näha, et puuduvad grupid 8 ja 7.

4. Tulemused

Käesoleva magistritöö raames oli läbi viidud mitmeid erinevaid eksperimente erinevatel andmestikel kasutades kaheksat masinõpe mudelit. Peamine eesmärk oli leida vastus järgnevatele küsimustele:

- Kas LOD2 põhjal on võimalik leida geomeetrilise sarnasuse järgi omavahel sarnased majad?
- Kui palju erineb inimese hinnang masinõpe mudeli hinnangust?

Mõlemale küsimusele aitasid kõige paremini vastata tulemused, mis olid saadud kasutades tudengite sarnaste hoonete andmestiku. See kasutas LOD2 andmeid ning oli juba mingil määral tudengite poolt paigutatud omavahel sarnastesse kolmikutesse. Nende eksperimentide käigus selgus, et hooneid on võimalik LOD2 põhjal grupeerida geomeetrilise sarnasuse järgi ning mudel on võimeline eristama omavahel sarnaseid ehitisi. Siiski olenevad tulemused väga palju valitud parameetritest ning üleüldiselt arvutustest, mida LOD2 põhjal saadakse. Sama kehtib ka teise küsimuse puhul, kus järelduseks oli see, et mudeli hinnang võib teatud juhtudel väga hästi korreleeruda inimeste hinnanguga, kuid oleneb väga palju hoonetest ning ka talle antud parameetritest.

Sarnaste hoonete valimi puhul oli leitud, et kõige paremini saab hoonete sarnasuse järgi klasterdamisega hakkama hierarhiline klasterdamine. Lisaks suudab antud mudel luua ka dendrogrammi, millega on palju kergem ning mugavam analüüsida tekkinud klastreid ning gruppeerida üldisemaks tüübiks. Üheks märkimisväärsemaks tulemuseks oli näiteks, kuidas mudel suutis paigutada kolme tudengi poolt toodud sarnased, üpriski spetsiifilise välimusega, kolmikud ühte teatud gruppi, millega ükski teine mudel hakkama ei saanud (vt joonis 22). Joonisel 23 on toodud välja hooned aadressil Sütiste tee 35, 41 ja 43 ning

Joonis 22. Tammepõllu hooned. Kõik paigutatud ühte kindlasse klastrisse

Mustamäe tee 137, 147 ja Sääse tänav 1 ning on näha, et kõik hooned on visuaalselt omavahel vägagi sarnased. Joonise põhjal on näha, et mudelid on visuaalselt väga sarnased ning samuti on teada, et hooned ei ole üksteisele nii lähedal, kui näiteks eelneva näite

puhul. Hierarhilise klasterdamise mudel oli võimeline ka need hooned lisaks paarile teisele sarnasele ehitisele paigutama samasse klastrisse.

Joonis 23. Vasakul on kujutatud Mustamäe tee ja Sääse hooned ning paremal Sütiste tee hooned. Paigutatud ühte klastrisse

Lisaks sellele oli mudel võimeline grupeerima samad hooned üle kogu riigi. Näiteks joonisel 24 olevad hooned asuvad kolmes eri kohas, Tallinnas, Kundas ja Tartus, kuid sellele vaatamata, suutis mudel need ikka paigutada sarnasuse järgi täpselt samasse klastrisse.

Joonis 24. Kolm sarnast tüüpi hooned Tallinnas (kõige vasakpoolsem), Kundas (keskmine), Tartus (kõige parempoolsem). Paigutatud ühte klastrisse

Eelnevate tulemuste ja näidiste põhjal võib taaskord järeldada, et küsimused said vastatud. LOD2 andmete põhjal on siiski mingil määral võimalik geomeetrilise sarnasuse järgi hooneid grupeerida. Lisaks sellele ei pruugi paljudel juhtudel tulemused inimsilma tulemustest väga palju erineda.

NSVL paneelmajade andmekogumi puhul olid klasterdamise mudelite tulemused enam vähem sarnased, kuid siiski kõige täpsemaks osutus taaskord hierarhiline klasterdamine. Sarnaselt eelnevatele tulemustele, suutis mudel ka antud andmestikuga leida ja klaster-

dada kõik omavahel sarnased hooned kokku. Probleem tekkis andmekogu tüüpide vastu valideerimisel. Mudel suutis tekitada klastrid hoonetele, mis olid samasuguse spetsiifilise projekti numbriga. Näiteks vastavalt projekti numbrile 1-464A-17, kuhu kuulusid näiteks sellised hooned nagu E.Vilde tee 91 ja Keskuse 22 (vt joonis 25), loodi identne klaster, kuhu paigutati kõik vastava projekti numbriga hooned.

Joonis 25. Projekt 1-464A-17 hooned. Paigutatud samasse klastrisse

Siiski, mida üldisemaks läks projekti ja/või seerinumber, seda vähem klappisid mudeli tulemused päris tüüpidega. Näiteks andmebaasist tuli välja, et sama seeria number, 1-464, on kolmel hoonel: E.Vilde tee 65, A.H. Tammsaare tee 103 ning Akadeemia tee 58. LOD2 põhjal aga on need hooned visuaalselt erinevad (vt joonis 26). Üks hoone on teistest selgelt suurem, teine väiksem ning kolmas on keskmise suurusega ning erineva katuse kujuga. Sarnaselt inimsilmale, olid need hooned erinevad ka masinõpe mudeli jaoks, mistõttu ei suudetud hoonete jaoks vastavalt samale klastrite arvule tekitada samasugust segmenti.

Joonis 26. Seeria/projekt 1-464 hooned. Paigutatud erinevatesse klastritesse

Siiski võib pidada saadud tulemusi samuti positiivseteks, sest anomaalia oli tekkinud pigem andmebaasi ning ajaloolise määramise tõttu kui mudeli. Mudel suutis eristada omavahel sarnased hooned ning paigutada need eraldi klastritesse.

Klassifitseerimise puhul näitas kõige paremat ja stabiilsemat tulemust juhusliku metsa algoritm, mis suutis ennustada korrektselt keskmiselt 74% kordadest. Siiski oli tehtud järeldus, et andmestik ei olnud klassifitseerimise meetodite jaoks piisavalt suur ning mõlemad nii test- kui ka treeningbaas jäid liiga väikesteks, et tulemusi korralikult valideerida.

Kredexi hoonete peal tehtud klasterdamise eksperimentide tulemused olid valideeritud tüpoloogia andmestikus defineeritud hoone tüüpide järgi. Selgus, et mudel ei ole võimeline antud 400 hoone põhjal looma täpselt tüüpidele vastavad klastrid. Meetodid suutsid

keskmiselt defineerida vaid neli klastrit 21-st, mis kattusid defineeritud hoone tüüpidega. Ülejäänud tüübid olid üldjuhul jaotatud kahe klastri kaupa. Sellegipoolest leiti, et väga oluline oli antud mudelite ja andmestiku puhul nominaalatribuutide teisendamine. Ilma ei suutnud mudel defineerida pea ühtegi kattuvat klastrit.

Klassifitseerimise eksperimentide käigus leiti, et kõige paremaid tulemusi näitasid otsustuspuu ja juhusliku metsa algoritmid. Mõlema puhul oli keskmine ennsutuse täpsus ligikaudu 95%. Sellegipoolest vaadates F1 skoori, tuli võitjaks juhuslik mets, mille kõrgeim keskmine F1 skoor oli 0.84. Otsustuspuul oli selleks skooriks 0.71. Ka selle andmestiku puhul oli siiski leitud, et oleks vaja teha täiendavaid uuringuid rohkemate andmetega. Selle põhjuseks oli peamiselt ebaproportisonaalne jaotus test- ja treeningbaasi vahel osade hoonetüüpide puhul. Näiteks oli leitud, et paneelmaju oli palju rohkem võrreldes puitmajadega ning, kui paneelmajade puhul oli jaotus enam vähem proportsionaalne ja mudeli ennustustäpsuses võis olla kindel, siis sama ei saanud kindlusega öelda puitmajade kohta. Sellele vaatamata ei olnud antud andmestikul tehtud eksperimendid kõige olulisemad, sest tegemist ei olnud ainult LOD2 andmetega.

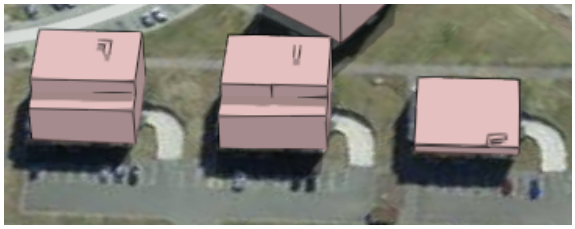
4.1 Peamised tähelepanekud ja edasised tegevused

Eksperimentide käigus oli leitud iga andmestiku puhul ka mitut erinevat tähelepanekut, mis võis suuremal või väiksemal määral antud tulemusi mõjutada negatiivselt. Tudengite sarnaste hoonete valimi puhul oli leitud, et mõningad vead olid tekkinud tähelepanuvigadest. Näiteks tekkis paar olukorda, kus oli kogemata lisatud andmetabelisse vale hoone koos vale ehitisregistrikoodiga. Joonisel 27 on toodud välja 4 hoonet, millest kaks vasakul olid lisatud andmetabelisse ning tõenäoliselt oli plaanis lisada ka kolmas samasugune, kuid kogemata lisati selle asemel hoopis selle kõrval olev maja, mis on selgelt erineva teistest ning seetõttu mõjutas ka tulemusi.

Joonis 27. Valesti lisatud hoone näidis

Paar korda oli märgatud ka seda, et mõned hooned ei ole päris 100% sarnased, näiteks kahel hoonel on lamekatus, kuid ühel on kolmnurkne, kuid sellele vaatamata oli see lisatud andmetabelisse, kui ülejäänud kahega sarnane hoone. Mõndade vigaselt paigutatud hoonete puhul oli leitud ka seda, et RESTO API-lt tagastatud andmed võisid samuti kolmel visuaalselt üpris sarnasel hoonel erineda, kuid ka seda juhtus pigem harva. Ühe

kolmiku puhul oli tekkinud ka selline olukord, kus hooned oli üksteisest väga erinevad igas mõttes ning ühtegi lähedal olevat hoonet ka ei olnud, mistõttu ei saanud see olla seostatud ka esimese näitega. Konsulteerides RESTO poolse konsultandiga, oli jõutud järeldusele, et sellel ajal kui tudengid paigutasid hooned sarnasuse järgi tabelisse, olid kõik hooned selekteeritud kolmikus ehituse all. Eksperimentide jooksutamise hetkeks olid kahe hoone andmed uuenenud, sest hooned olid lõplikult valmis saanud ning ehitisregistrisse korrektselt kantud. Seetõttu muutus ka mõlema hoone visuaalne representatsioon. Käesolev näide on toodud välja joonisel 28.



Joonis 28. Näide hoonetest, millest kaks said valmis ning üks on ikka ehituses

Nagu sai ka eelnevalt üldiste tulemuste juures mainitud, siis NSVL paneelmajade andmebaasi puhul oli peamiseks probleemiks hoonete ajalooline paigutus projekti numbri järgi. Mida uuem hoone, seda detailsem projekti number ning seda sarnasemad on kõik ühe projekti alla kuuluvad hooned. Vanemate puhul oli defineeritud üldjuhul ainult seerianumber ning kõik hooned ei pruukinud olla omavahel üks-ühele sarnased. Lisaks sellele oli leitud, et andmete hulk oli klassifitseerimise jaoks siiski liiga väike. Treening- ja testbaasideks jagamisel oli sihtparameetriteks määratud projekti numbrid jaotatud ebaühtlaselt, mistõttu ei olnud mudelil mõnda projekti isegi võimalik valideerida.

Tüpoloogia puhul oli üldpildis sarnane probleem, mis ka NSVL paneelmajade eksperimentide puhul. Sihtparameetrite jaotamisel treening- ja testbaasiks tekkis hoonetüüpide vahel ebaproportsionaalne jaotus, sest näiteks valdav enamus hooneid oli paneelmajad ning ainult väike osa neist olid puitmajad. Seetõttu võis mudelil olla liiga vähe andmeid, et korrektselt ennustada just puitmaju.

Kokkuvõteks võib öelda, et kuigi küsimustele saadi vastused ning üldpildis olid tulemused positiivsed, vajab töö siiski edasist uurimist, peamiselt just suuremal andmete hulgal. Sellisel juhul oleks võimalik mudeli efektiivsus ja täpsust veelgi paremini valideerida ning võimalusel parandada ka parameetrite valikut.

5. Kokkuvõte

Käesoleva magistrtöö eesmärgiks oli masinõppel põhinevate meetoditega luua mudel, mis suudaks leida sarnased hooned nende LOD2 andmete põhjal. Samuti pidi see vastama töö alguses püstitatud küsimustele:

- Kas LOD2 taseme põhjal on võimalik leida sarnased majad ning nad omavahel grupeerida?
- Kui palju võib erineda inimsilma hinnang masinõppe mudeli hinnangust?

Probleemi lahendamiseks ning küsimustele vastamiseks oli läbi viidud mitmeid eksperimente, kasutades kaheksat erinevat masinõppe mudelit neljal eri andmestikul. Töö sissejuhatavas osas rääkis töö autor lähemalt töö vajalikkusest ning selle üldisest taustast ning teoreetilises osas tutvustati, kuidas on siiani sarnaseid ülesandeid lahendatud ning milliseid lähenemisviise kasutatud.

Peamine rõhk oli antud töös juhtimata õppel, sest andmed olid üldjuhul kas sildistamata või eeldefineeritud tüübid ei olnud piisavalt universaalsed, et kõiki Eestis olevaid maju ära katta. Seda arvestades oli leitud, et kõige paremini suutis hooned sarnasuse järgi ära jagada hierarhiline klasterdamine.

Juhitud õppeviisi puhul näitas kõige paremaid tulemusi juhusliku metsa algoritm. NSVL paneelmajade andmestiku puhul oli selle keskmine ennustustäpsus ligikaudud 73%. Tüpoloogia ja Kredexi hoonete puhul, kus ei olnud tegemist ainult LOD2 andmetega, oli mudeli keskmine ennustustäpsus ligikaudu 94%, mis küll oli võrreldav otsustuspuu tõenäosusega, kuid F1 koori järgi oli näha, et juhuslik mets on siiski palju stabiilsem ja parem mudel.

Kokkuvõttes said magistritöös püstitatud eesmärgid täidetud ning küsimused vastatud. Tulemuste põhjal oli leitud, et LOD2 tase on piisavalt detailne, et leida omavahel geomeetriliselt sarnased hooned ning grupeerida neid sarnasuse järgi. Lisaks sellele jõuti ka järeldusele, et üldjuhul ei pruugi mudeli hinnang inimsilma hinnangust väga palju erineda, aga seda juhul, kui on valitud piisavalt suur klastrite arv, et hooned võimalikult detailselt liigitada. Sellegipoolest vajab töö edasisi uuringuid ning ka täiendavat valideerimist suuremate andmestike peal, mis suudaks katta suuremat osa Eesti hooneid.

Kasutatud kirjandus

[1] Elisa Iliste. „Ehitisregistri andmete alusel elamupiirkonna energiatõhususe hindamise alused“. Magistritöö. Tallinna tehnikaülikool, 2023.

[2] Gianluca Ruggieri, Francesca Andreolli ja Paolo Zangheri. *A Policy Roadmap for the Energy Renovation of the Residential and Educational Building Stock in Italy*. 2023.

[3] Einari Kisel *et al. How can we implement a seamless Renovation Wave in Estonia?* [Accessed: 22-02-2023]. URL: https://www.finestcentre.eu/article-implementing-renovation-wave.

[4] T. Vaino ja E. Nippala. *Long-term renovation strategy for 2020-2050: assessment from a low-carbon perspective—case Finland*. 2022.

[5] 3D geoinformation. *Cities/regions around the world with open datasets*. [Accessed: 14.04.2023]. URL: https://3d.bk.tudelft.nl/opendata/opencities/.

[6] Visicom. *3D Data for Middle East*. [WWW]. 2020. URL: https://visicomdata.com/news/3d_data_for_middle_east.

[7] Filip Biljecki *et al. Applications of 3D City Models: State of the Art Review*. 2015.

[8] Biao Wang *et al. A Topology-Preserving Simplification Method for 3D Building Models*. 2021.

[9] Hongchao Fan ja Liqiu Meng. *Automatic Derivation of Different Levels of Detail for 3D Buildings Modeled by CityGML*. 2009.

[10] Marc-O. Löwner *et al.* „New Concepts for Structuring 3D City Models – An Extended Level of Detail Concept for CityGML Buildings“. Teoses: *Computational Science and Its Applications – ICCSA 2013*. 2009.

[11] Open Geospatial Consortium Inc. *Candidate OpenGIS® CityGML Implementation Specification (City Geography Markup Language)*. Editors: Gerhard Gröger and Thomas H. Kolbe and Angela Czerwinski. 2006.

[12] Filip Biljecki, Hugo Ledoux ja Jantien Stoter. „An improved LOD specification for 3D building models“. Teoses: *Computers, Environment and Urban Systems*. 2016.

[13] Gerhard Gröger ja Lutz Plümer. *CityGML – Interoperable semantic 3D city models*. 2012.

[14] TIB OÜ - Kütte- ja ventilatsiooniprojektid. *Insolatsioon*. [Accessed: 22-02-2023]. URL: https://tib.ee/insolatsioon/.

[15] Jordi Vermaulen. „Geometric similarity measures and their applications“. Doktoritöö. 2023.

[16] Cyrus Hillsman, Yan Wang ja Dima Nazzal. *A semi-automatic mold cost estimation framework based upon geometry similarity*. 2013.

[17] Jiantao Pu ja Karthik Ramani. *On visual similarity based 2D drawing retrieval*. 2005.

[18] Pu Jiantao *et al. 3D Model Retrieval Based on 2D Slice Similarity Measurements*. 2004.

[19] Ding-Yun Chen *et al. On Visual Similarity Based 3D Model Retrieval*. 2003.

[20] Ryutarou Ohbuchi ja Tsuyoshi Takei. *Shape-Similarity Comparison of 3D Models Using Alpha Shapes*. 2003.

[21] Hamid Laga, Hiroki Takahashi ja Masayuki Nakajima. *Geometry Image Matching for Similarity Estimation of 3D Shapes*. 2004.

[22] Ling Yang *et al. Urban morphological regionalization based on 3D building blocks—A case in the central area of Chengdu, China*. 2022.

[23] Duo Xu *et al. Field measurement study on the impacts of urban spatial indicators on urban climate in a Chinese basin and static-wind city*. 2018.

[24] Solveig Badillo *et al. An Introduction to Machine Learning*. 2020.

[25] Ethem Alpaydin. *Introduction to Machine Learning, fourth edition*. The MIT Press, 2020.

[26] Susmita Ray. *A Quick Review of Machine Learning Algorithms*. 2019.

[27] Taiwo Oladipupo Ayodele. „Types of Machine Learning Algorithms“. Teoses: *New Advances in Machine Learning*. BoD - Books on Demand, 2010.

[28] Masashi Sugiyama. *Statistical Reinforcement Learning: Modern Machine Learning Approaches*. CRC Press, 2015.

[29] Iqbal H. Sarker. *Machine Learning: Algorithms, Real-World Applications and Research Directions*. 2021.

[30] Jason Brownlee. *A Tour of Machine Learning Algorithms*. [Accessed: 22-02-2023]. 2019. URL: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms.

[31] Mengyao Cui. *Introduction to the K-Means Clustering Algorithm Based on the Elbow Method*. 2020.

[32] Shraddha Shukla ja Naganna S. *A Review ON K-means DATA Clustering APPROACH*. 2014.

[33] Abbas Hanon AlAsadi *et al. HYBRID K-MEANS CLUSTERING FOR COLOR IMAGE SEGMENTATION*. 2015.

[34] Asanka Perera. *Finding the optimal number of clusters for K-Means through Elbow method using a mathematical approach compared to graphical approach*. [Accessed: 22-02-2023]. 2017. URL: https://www.linkedin.com/pulse/finding-optimal-number-clusters-k-means-through-elbow-asanka-perera/.

[35] Stephen P. Borgatti. „How to Explain Hierarchical Clustering“. Teoses: *Connections*. 1994.

[36] JavatPoint. *Hierarchical Clustering in Machine Learning*. [Accessed: 24-02-2023]. URL: https://www.javatpoint.com/hierarchical-clustering-in-machine-learning.

[37] Fatih Karabiber. *Hierarchical Clustering*. [Accessed: 24-02-2023]. URL: https://www.learndatasci.com/glossary/hierarchical-clustering/.

[38] T. Fuertes. *Hierarchical clustering, using it to invest*. [Accessed: 24-02-2023]. 2016. URL: https://quantdare.com/hierarchical-clustering/.

[39] Fionn Murtagh ja Pedro Contrera. *Algorithms for hierarchical clustering: an overview*. 2011.

[40] Ransaka Ravihara. *Gaussian Mixture Model Clearly Explained*. [WWW]. 2023. URL: https://towardsdatascience.com/gaussian-mixture-model-clearly-explained-115010f7d4cf.

[41] Yupeng Li *et al. Clustering Analysis in the Wireless Propagation Channel with a Variational Gaussian Mixture Model*. 2018.

[42] Saishruthi Swaminathan. *Logistic Regression — Detailed Overview*. [WWW]. 2018. URL: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc.

[43] Indrek Arandi. *Masinõppimine*. [WWW]. (25.02.2023). URL: https://masinope.ee/masinoppimine/.

[44] Jehad Ali *et al.* „Random Forests and Decision Trees“. Teoses: *International Journal of Computer Science Issues*. 2012.

[45] Yanli Liu, Yourong Wang ja Jian Zhang. „New Machine Learning Algorithm: Random Forest“. Teoses: *Information Computing and Applications*. Toim. Baoxiang Liu, Maode Ma ja Jincai Chang. 2012.

[46] Chirag Goyal. *Bagging- 25 Questions to Test Your Skills on Random Forest Algorithm*. [WWW]. 2021. URL: https://www.analyticsvidhya.com/blog/2021/05/bagging-25-questions-to-test-your-skills-on-random-forest-algorithm/.

[47] Dustin Boswell. *Introduction to Support Vector Machines*. 2002.

[48] Vikramaditya Jakkula. *Tutorial on Support Vector Machine (SVM)*. 2006.

[49] Geeks for Geeks. *Introduction to Support Vector Machines (SVM)*. [WWW]. 2023. URL: https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/.

[50] Ting-ting Dai ja Yan-shou Dong. „Introduction of SVM Related Theory and Its Application Research“. Teoses: *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*. 2020.

[51] Kevin Gurney. *An Introduction to Neural Networks*. CRC Press, 1997.

[52] Jeremy Jordan. *Evaluating a machine learning model*. [WWW]. 2017. URL: https://www.jeremyjordan.me/evaluating-a-machine-learning-model/.

[53] Sebastian Raschka. *Python Machine Learning*. Packt Publishing, 2015.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Mark Genrich Geller

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose“Sarnaste hoonete leidmine LOD2 põhjal kasutades masinõppe algoritme”, mille juhendaja on Innar Liiv
   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 veebikeskkonna 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.

10.05.2023

[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 ühisautor(id) 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.

Lisa 2 - Tüpoloogia andmestik

Kategooria	Nimetus	T2	Tüpoloogia välisseina lii	silikaltsiit ja kergbetoon		suurpaneel			tellis										Kivi					
		kirjeldav	Asukoht	Kogu Eesti		Harjumaa		Harjumaa,	Kogu															
		E8	Trepikodade arv	alates 1	alates 1	alates 1	alates 1	alates 1	1	1	1	1	1	2+	2+	2+	2+	2+	1	1	2+	2+	2+	2+
		E10 põhjal	Korruste vahemik	2-5	2-5	2-5	6-9	2+	2-5	2-5	2-5	2-5	6+	2-5	2-5	2-5	2-5	6-9	2-5	6+	2-5	2-5	6+	6+
		E12 põhjal	Ajastu	<1970	>1970	1950-1980	1950-1980	1980+	1980+	1960-1980	1950-1960	<1950	>1950	1980+	1960-1980	1950-1960	<1950	>1950	>1950	>1950	<1980	1980+	<1980	1980+
ID	Tüpoloogia kood	T1	-	A211	A212	B211	B221	B212	C112.3	C112.2	C112.1	C111	C122	C212.3	C212.2	C212.1	C211	C222	D112	D122	D211	D212	D221	D222
	Referentshoone	T61	Kaldkatusega	104016899	116017286				-	1.1E+08	1.1E+08	1E+08		1.2E+08	1.1E+08	1.1E+08	1E+08							
	Referentshoone	T62	Lamekatusega	-	108011906	101011258	101013333	104019098	1.1E+08	1E+08	1.1E+08		1.1E+08	1E+08	1.2E+08	-	-	1E+08						
Üldandmed	Hoonete arv valimis	T32	-	26	169	41	8	19	9	9	9	3	4	116	116	116	4	6	13	4	159	101	9	3
	Trepikoja ehitusalune pindala	T33	m2	200.8	200.5	186.5	293.0	204.9	284.6	284.6	284.6	211.3	353.8	199.2	199.2	199.2	242.3	355.1	268.4	353.8	193.1	209.6	316.1	332.5
	Hooneosade arv trepikojas korruse kohta	T34	-	3	2	3	4	3	4	4	4	3	4	3	3	3	4	3	3	4	3	3	4	3
	Ruumide arv vannitoata trepikojas korruse k	T35	-	11	9	9	13	9	12	12	12	11	12	9	9	9	9	10	12	12	9	9	13	9
	Trepikoja suletud netopindala korruse kohta	T36	m2	188.4	189.7	191.3	266.1	204.7	243.9	243.9	243.9	200.4	253.1	179.5	179.5	179.5	203.3	235.2	236.3	253.1	182.3	196.3	262.9	216.2
	Trepikoja köetav pindala korruse kohta	T37	m2	155.5	151.8	157.1	233.3	169.4	201.2	201.2	201.2	153.8	221.7	149.4	149.4	149.4	175.3	205.0	195.6	221.7	148.6	160.1	228.3	194.8
Arhitektuurne vorm	Kald katuse osakaal	T38	%	73.1	39.1	14.6	0.0	15.8	44.4	44.4	44.4	100.0	0.0	57.8	57.8	57.8	100.0	33.3	53.8	0.0	49.4	34.6	0.0	66.7
	Lodzade osakaal	T39	%	0.0	56.8	14.6	87.5	94.7	33.3	33.3	33.3	0.0	75.0	27.6	27.6	27.6	25.0	50.0	23.1	75.0	79.7	81.8	33.3	19.5
	Keldri osakaal	T40	%	96.2	95.3	100.0	100.0	100.0	66.7	66.7	66.7	100.0	75.0	94.8	94.8	94.8	100.0	33.3	76.9	75.0	95.9	96.2	81.8	33.3
	Keskmine akna ja fassaadi suhe	T41	-	0.26	0.22	0.26	0.22	0.20	0.19	0.19	0.19	0.19	0.19	0.23	0.23	0.23	0.21	0.25	0.20	0.19	0.24	0.21	0.22	0.26
	Otsaakende olemasolu osakaal	T42	%	73.1	15.4	68.3	25.0	5.3	88.9	88.9	88.9	66.7	75.0	66.4	66.4	66.4	100.0	100.0	84.6	75.0	53.9	18.0	45.5	100.0
Geomeeria näitajad	Keskmine korruse kõrgus	T43	m	3.5	3.7	3.2	3.1	3.3	3.9	3.9	3.9	4.7	3.3	3.8	3.8	3.8	4.6	3.3	4.0	3.3	3.7	3.6	3.1	3.3
	Tüüpne ruumi kõrgus	T6	m	2.5	2.5	2.5	2.5	2.5	2.5	2.5	3.2	3.2	2.5	2.5	2.5	3.2	3.2	2.5	2.5	2.5	2.9	2.5	2.5	2.5
	Tüüpne keldri ruumi kõrgus	T44	m	2.20	2.20	2.20	2.20	2.20	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3
	Hoone keskmine laius	T45	m	11.5	11.8	12.4	13.1	13.0	14.1	14.1	14.1	9.9	11.4	12.1	12.1	12.1	12.9	14.0	13.3	11.4	11.8	12.3	13.2	14.2
	Keskmine trepikoja pikkus	T46	m	17.7	16.9	15.4	22.1	17.0	20.7	20.7	20.7	20.3	23.5	16.9	16.9	16.9	16.2	21.9	20.6	23.5	16.7	16.9	22.3	21.0
	Köetav kõrgus	T7	m	2.5	2.5	2.5	2.5	2.5	2.5	2.5	3.2	3.2	2.5	2.5	2.5	3.2	3.2	2.5	2.6	2.5	2.9	2.5	2.5	2.5
	Köetav maht	T47	m3	502.0	501.2	466.3	732.5	512.3	711.4	711.4	910.6	676.3	884.4	497.9	497.9	637.3	775.3	887.7	667.3	884.4	647.1	555.7	886.0	886.0
Piirdetarindite soojusläbivused	Välisseina soojusläbivus	T12	W/(m2K)	0.83	0.83	1.05	1.05	0.80	1.15	1.15	1.38	1.38	1.01	1.15	1.15	1.38	1.38	1.01	1.07	1.01	1.21	0.98	1.01	1.01
	Katuslae soojusläbivus	T13	W/(m2K)	0.80	0.80	1.10	1.10	0.90	0.70	1.00	1.10	1.10	0.80	0.70	1.00	1.10	1.10	0.80	0.88	0.80	1.04	0.78	0.80	0.80
	Pinnasel põranda soojusläbivus	T15	W/(m2K)	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70		0.70	0.70	0.70			
	Kütmata keldri VL soojusläbivus	T16	W/(m2K)	0.35	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
	Keldriseina U	T48	W/(m2K)	2.60	2.99	1.05	1.05	0.80	2.78	2.44	2.06	2.06	2.44	2.78	2.44	2.06	2.06	2.44	2.57	2.44	2.10	2.34	2.44	2.44
	Välisuste soojusläbivus	T17	W/(m2K)	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	Akende soojusläbivus	T18	W/(m2K)	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90	2.90
Piirdetarindite liitekohtade joonsoojusläbivused ja õhuleke	VS-VS liitekoha joonsoojusläbivus	T19	W/(mK)	0.26	0.26	1.17	1.17	1.17	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.37	0.49	0.26	0.26
	VS-KL liitekoha joonsoojusläbivus	T20	W/(mK)	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49
	VS-PööninguVL liitekoha	T21	W/(mK)	0.41	0.41	0.49	0.49	0.49	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
	VS-PP liitekoha joonsoojusläbivus	T22	W/(mK)	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
	VS-KeldriVL liitekoha joonsoojusläbivus	T23	W/(mK)	0,2	0,2	0.5	0.5	0.5	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.13	0.13	0.13	0.13	0.13	0.13
	VS-VL liitekoha joonsoojusläbivus	T24	W/(mK)	0.01	0.01	0.77	0.77	0.77	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.16	0.16	0.16	0.16	0.16	0.16
	VS-SS liitekoha joonsoojusläbivus	T25	W/(mK)	0	0	1.03	1.03	1.03	0	0	0	0	0	0	0	0	0	0	0	0	0.12875	0.2575	0	0
	Akna- ja ukse seinakinnituse joonsoojusläbivus	T26	W/(mK)	0.42	0.42	0.07	0.07	0.07	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.37625	0.3325	0.42	0.42
	Õhulekke arv	T27	$m^3/(h \cdot m^2)$	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
Konstruktsioonide paksused	Välisseina tüüpne paksus	T49	m	0.300	0.300	0.275	0.275	0.275	0.430	0.430	0.570	0.570	0.560	0.430	0.430	0.570	0.570	0.560	0.406	0.560	0.464	0.359	0.560	0.560
	Katuslae tüüpne paksus	T50	m	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
	Külma pööninguga lae tüüpne paksus	T51	m	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.4
	Keldriseina tüüpne paksus	T52	m	0.45	0.35	0.28	0.28	0.28	0.4	0.5	0.7	0.7	0.5	0.4	0.5	0.7	0.7	0.5	0.5	0.5	0.5	0.4	0.5	0.5
	Vahelae tüüpne paksus	T53	m	0.3	0.3	0.2	0.2	0.2	0.3	0.3	0.2	0.2	0.3	0.3	0.3	0.2	0.2	0.3	0.3	0.3	0.2	0.3	0.3	0.3
Avatäidete mõõdud	Ukse tüüpne laius	T54	m	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8
	Ukse tüüpne kõrgus	T55	m	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
	Akna tüüpne kõrgus	T56	m	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
	Akna tüüpne laius	T57	m	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6
	Trepikoja akna tüüpne kõrgus	T58	m	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8

Lisa 3 – E-Ehituse API-lt saadud jada osakestest JSONi kujul


```
[
    {
        "area": 13.837,
        "x0": 548439.07,
        "y0": 6598609.21,
        "z0": 33.19,
        "x1": 548432.2,
        "y1": 6598613.53,
        "z1": 36.76,
        "x2": 548439.07,
        "y2": 6598609.21,
        "z2": 36.6,
        "nx": 0.532,
        "ny": 0.847,
        "nz": 0.0
    },
    {
        "area": 30.049,
        "x0": 548439.07,
        "y0": 6598609.21,
        "z0": 36.6,
        "x1": 548432.2,
        "y1": 6598613.53,
        "z1": 36.76,
        "x2": 548428.31,
        "y2": 6598607.23,
        "z2": 36.81,
        "nx": 0.02,
        "ny": -0.005,
        "nz": 1.0
    },
    ...
]
```

Lisa 4 – RESTO API-lt saadud LOD2 põhjal tehtud arvutused JSONi kujul


```
{
  "L1": "pitched roof",
  "L10": 873.4449999999999,
  "L11": 1338.6779999999999,
  "L12": 0,
  "L13": 201.83499999999998,
  "L14": 0,
  "L15": 1344.859,
  "L16": 0,
  "L17": 204.55700000000002,
  "L18": 0,
  "L19": 0,
  "L2": 17.919999999999998,
  "L20": 4,
  "L21": 77.23000000000002,
  "L22": 185.91548894334264,
  "L23": 0,
  "L24": 178.2345139918138,
  "L25": 0,
  "L26": 178.2345139918138,
  "L3": 178.2345139918138,
  "L4": 15652.134399999997,
  "L5": 3089.928999999999,
  "L6": 880.4260000000002,
  "L8": 873.4449999999999,
  "L9": 873.4449999999999,
  "errors": {}
}
```