Tartu Ülikool
Loodus- ja täppisteaduste valdkond
Matemaatika ja statistika instituut

Hardi Roosi

Juhumetsa mudel ja selle rakendamine liiklusõnnetuste prognoosimisel

Matemaatilise statistika eriala

Bakalaureusetöö (9 EAP)

Juhendaja Taavi Unt

Tartu 2019

Juhumetsa mudel ja selle rakendamine liiklusõnnetuste prognoosimisel

Bakalaureusetöö

Hardi Roosi

Lühikokkuvõte. Bakalaureusetöös uuritakse, kuidas ilmastiku- ja teeolud mõjutavad liiklusõnnetuste toimumist. Töö esimeses peatükis antakse teoreetiline ülevaade juhumetsa mudelist ning otsustuspuudest. Teises peatükis rakendatakse juhumetsa mudelit, prognoosimaks liiklusõnnetuste esinemist ilmastiku- ja teeoludest tulenevalt.

CERCS teaduseriala: P160 Statistika, operatsioonanalüüs, programmeerimine, finants- ja kindlustusmatemaatika

Märksõnad: Juhumets, klassifitseerimispuu, Tundlikus, spetsiifilisus

Random forest model and its application of predicting road accidents

Bachelor's thesis

Hardi Roosi

Abstract. The purpose of this Bachelor's is to examine how weather and road conditions affect the occurrence of road accidents. The first chapter of the thesis gives a theoretical overview of the random forest algorithm and the decision trees. In the second chapter, the random forest model is applied to predict the occurrence of traffic accidents due to weather and road conditions.

CERCS research specialisation: P160 Statistics, operation research, programming, actuarial mathematics

Keywords: Random forest, classification tree, sensitivity, specificity

Sisukord

Sissejuhatus

Liiklusel ja transpordil on ühiskonnas suur roll. Kuna teedel on liiklustihedus suur, on liiklusõnnetuste juhtumine paratamatu. Õnnetuste vältmine on alati iga liikleja eesmärgiks ning õnnetuste tagajärel saadavad vigastused ja elukaotused on ühiskonnale korvamatu kahju. Liiklusõnnetuste põhjuseid on väga palju, üks neist on teedel esinevad olud. Inimeste hooletusest tingitud õnnetusi on raske vältida, kuid erinevaid ilmastiku mõjul toimunud õnnetusi on võimalik ennetada, kui sõitjal on piisavalt informatsiooni valitsevate olude kohta.

Käesoleva bakalaureusetöö eesmärgiks on uurida, kuidas ilmastik ja teeolud mõjutavad liiklusõnnetuste toimumist. Seejuures vaadeldakse liiklusõnnetuste esinemist või mitteesinemist uuritavates piirkondades ühetunnise ajaakna vältel. Eraldiseisvateks uuritavateks piirkondadeks on üle Eesti paiknevaid teeilmajaamu ümbritsevad viiekilomeetrise raadiusega alad. Liiklusõnnetuste modelleerimiseks on kasutatud juhumetsade mudelit.

Töö põhiosa on jaotatud kahte peatükki. Esimeses peatükis antakse teoreetiline ülevaade juhumetsa mudelist ning otsustuspuudest, millel juhumetsa mudel põhineb. Teises peatükis rakendatakse juhumetsa mudelit Eesti Liikluskindlustuse Fondi ning Maanteeameti teeilmajaamade andmetele, prognoosimaks liiklusõnnetuste esinemist ilmastikust ja teeoludest tulenevalt.

Bakalaureusetöö andmete korrastamiseks ning analüüsiks on kasutatud statistikatarkvara *R*(versioon 3.5.1). Töö on kirjutatud kasutades *LaTeX* veebirakenduse liidest Overleaf.

Bakalaureusetöö autor tänab Taavi Unti ja Annegrete Peeki bakalaureusetööd puudutavate nõuannete, panustatud aja ning paranduste eest.

1 Juhumets klassifitseerimisülesande korral

Kuna juhumetsa mudel, mida tutvustatakse alapunktis 1.2, tugineb otustuspuudel, antakse esmalt neist ülevaade.

1.1 Klassifitseerimispuu

Selles peatükkis on kasutatud allikad James jt 2017, lk 306-314, kui ei ole viidatud teisiti.

Klassifitseerimisülesande, sealhulgas klassifitseerimispuu eesmärk on jaotada vaatlused selgitavate tunnuste abil K klassi, kus K on uuritava tunnuse Y võimalike väärtuste arv, st $y_i \in \{0, ..., K-1\}$ iga $i = 1, ..., n$ korral, kus n on vaatluste arv andmestikus.

Puu koostamisel jagatakse selgitavate tunnuste $X_1, \ldots, X_p$ poolt määratud ruum rekursiivselt binaarsete tükelduste läbi lõikumatuteks piirkondadeks $R_1, R_2, \ldots, R_j$ ehk lehtedeks. Puuharu rekursiivsel moodustamisel valitakse tunnus X_h ja tunnusele vastav lõikepunkt s, mille korral tekkinud tükeldusest $R_{m_v} = \{R_m \mid X_h \leq s\}$ ja $R_{m_p} = \{R_m \mid X_h > s\}$ saadav kasu on suurim. Seejuures tähistab $\{R_m \mid X_h \leq s\}$ piirkonna R_m sellist alampiirkonda, mille korral tunnus $X_h \leq s$.

Üheks võimaluseks tükeldusest saadava kasu mõõtmiseks on kasutada klassifitseerimisviga, mis piirkonna R_m korral on defineeritud kui

$$E_m = 1 - \max_k(\hat{p}_{mk}), \tag{1.1}$$

kus $\hat{p}_{mk}$ tähistab klassi k osakaalu piirkonnas R_m. Klassifitseerimisviga väljendab piirkonda R_m kuuluvate vaatluste osakaalu, mis ei kuulu selle piirkonna enamlevinud klassi.

Lisaks kasutatakse puu koostamisel piirkondade tükeldamiseks Gini indeksit või entroopiat. Gini indeks, mis piirkonna R_m korral on defineeritud kui

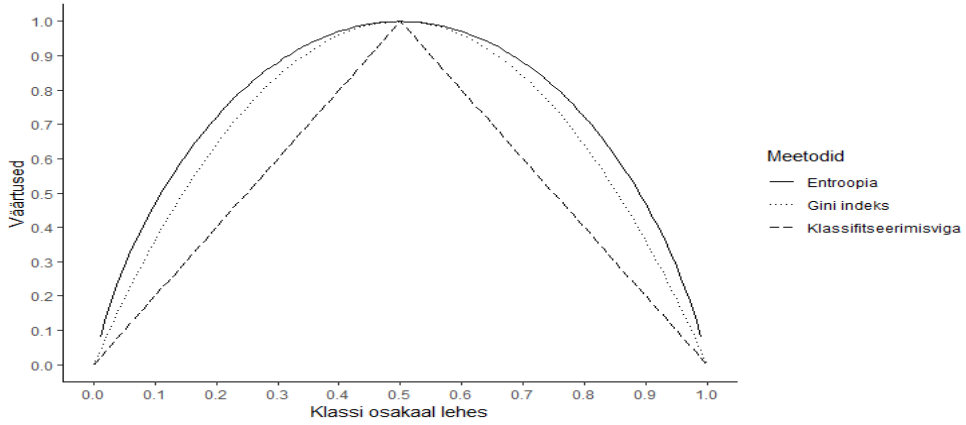
$$G_m = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}). \tag{1.2}$$

Gini indeks mõõdab varieeruvust üle K klassi. Gini indeksile viidetakse kui lehe puhtuse näitajale kuna kui kõik $\hat{p}_{mk}$ on 0 või 1 ümbruses, siis Gini ideksi väärtus tuleb väike. Seega väike Gini indeksi väärtus viitab, et piirkonnas olevad vaatlused on enamus samast klassist.

Entroopia piirkonnas R_m on defineeritud kui

$$D_m = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}. \tag{1.3}$$

Kuna $0 \leq \hat{p}_{mk} \leq 1$, siis $0 \leq -\hat{p}_{mk} \log \hat{p}_{mk}$. Analoogselt Gini indeksile, kui kõik $\hat{p}_{mk}$ väärtused on 0 või 1 ümbruses, siis tuleb entroopia kordaja väike, mis omakorda viitab, et vaatlused on enamasti samast klassist.



Joonis 1. Skaleeritud klassifitseerimisviga, Gini indeks ja entroopia erinevate klassi osakaalude korral binaarsel juhul (Hastie jt 2009, lk 309)

Klassifitseerimisviga ei ole aga piisavalt tundlik puu kasvatamisel. Seda iseloomustab binaarse uuritava tunnuse korral joonis 1, millel on kuvatud ümberskaleeritud

klassifitseerimisvea, Gini indeksi ja entroopia joonised juhul, kui uuritav tunnus on binaarne.

Piirkonna R_m tükeldamisel valitakse selline tunnus X_h ja sellele vastav lõikepunkt s, mille korral tükeldatud piirkondadele R_{m_v} ja R_{m_p} vastavate puhtuse mõõdikute vaatluste arvuga kaalutud summa oleks minimaalne. Näiteks Gini indeksi (1.2) kasutamise korral lahendatakse järgmist optimeerimisülesannet:

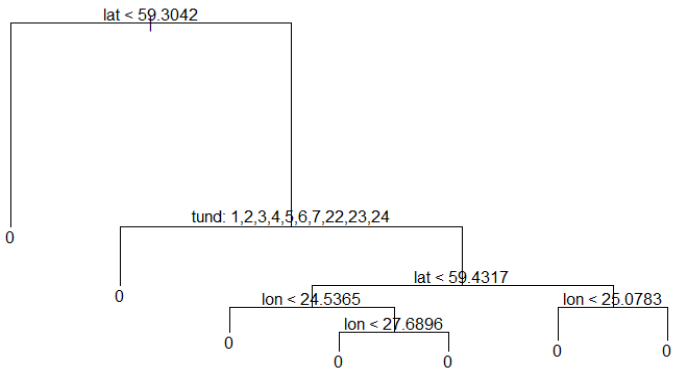
$$\min_{s,h}(N_{m_v} \cdot G_{m_v} + N_{m_p} \cdot G_{m_p}), \tag{1.4}$$

kus N_{m_v} ja N_{m_p} on vastavalt piirkonna R_m vasaku ja parema haru piirkondade suurused ning G_{m_v} ja G_{m_p} on vastavate harude Gini indeksid. (Hastie jt 2009, lk 307-309)

Puu tõlgendamisel on tähtis selgitavate tunnuste asukoht puus. Mida lähemal juurele ehk esimesele tükeldusele on tunnust kasutatud, seda tugevamini mõjutab antud tunnus uuritavat tunnust. Tulemuste üldises interpretatsioonis vaadeldakse prognoositava tunnuse Y klasside osakaalu igas lehes. Tunnuse Y klass, mille osakaal vaadeldavas lehes R_m on suurim, on vastava lehe prognoos $\hat{y}$ tunnusele Y ehk $\hat{y}_i = \mathrm{argmax}_k\, \hat{p}_{mk}$ iga vaatluse i korral, mille selgitavad tunnused kuuluvad piirkonda R_m.

Lihtsamaks arusaamiseks, kuidas puu tulemusi tõlgendada on toodud üks näide. Antud puu korral prognoositakse erinevate tunnuste mõju liiklusõnnetuste toimumisele. Lähemalt saab uurida andmestikku peatükkis 2.1. Antud näite puhul on puu koostamisel kasutatud Gini indeksit (1.2).

Jooniselt 2 on näha, et õnnetuse toimumist mõjutab kõige rohkem tunnus lat ehk laiuskraad. Lisaks mõjutavad liikusõnnetuste toimumist ka tunnus $tund$ ehk kellaaeg ja tunnus lon ehk pikkuskraad. Näiteks võiks mudelit interpreteerida järgnevalt: Kui laiuskraad on üle 59.3042 ning tund on 1 - 7 või 22 - 24, siis ei toimu õnnetust. Lisaks on märgata, et kõikides puu lehtedes prognoositakse sama



Joonis 2. Klassifitseerimispuu tasakaalustamata andmete korral

klassi 0. Küll aga on iga hargnemisega lehed „puhtamaks"muutunud. Sellepärast on puu joonisel hargnemiskohti, mille lehed prognoosivad sama tulemust.

1.2 Juhumets

Antud peatükis on kasutatud allikad James jt. 2017, lk 318-320.

Juhumetsa algoritmis koostatakse ühe puu asemel mitu puud. Algoritmis valitakse iga puu jaoks *bootstrap* meetodiga ehk tagasipanekuga juhuvalimiga n vaatlust, st iga puu kasutab erinevaid vaatlusi. Lisaks ei kasutata puuharude konstrueerimisel kõiki tunnuseid, vaid igas puuharus vaadledakse ainult juhuslikult valitud $l \approx \sqrt{p}$ selgitavat tunnust, kus p on andmestikus olevate selgitavate tunnuste arv. Iga puuharu koostamisel valitakse valitud tunnuste seast parim tunnus koos sellele tunnusele vastava lõikepunktiga. Seega erinevalt üksikutest puudest jäävad enamus andmestikus olevatest selgitavatest tunnustest puuharu tekkimisel arvesse võtmata. Sellest tulenevalt jäävad juhumetsa mudelis olevad puud robustsemad

kui klassifitseerimispuu, mille hindamiseks kasutatakse iga tükelduse korral kõiki selgitavaid tunnuseid.

Juhumetsa prognoosiskooride saamiseks väljastatakse kõige pealt iga vaatluse jaoks klassi kuulumise prognoosid (binaarsel juhul kas 0 või 1) iga puu korral. Juhumetsa prognoosid saadakse puude poolt väljastatud prognooside „hääletuse" tulemusena. Juhumetsa i-nda vaatluse klassi k kuulumise tõenäosushinnanguks on puude osakaal, mis prognoosisid i-nda vaatluse klassi k kuuluvaks. Seejuures juhumetsa poolt prognoositud i-nda vaatluse klassiks on klass, mida puud sellele vaatlusele kõige enam prognoosisid.

Kuna juhumetsa moodustavate puude arv on üldjuhul väga suur, siis ei ole mõistlik uurida hinnatud juhumetsa mudelit üksikute puude kaupa. Selle asemel kasutatakse juhumetsas selgitavate tunnuste tähtsust. Iga tunnuse korral leidakse tunnuse tähtsus kui maksimaalne puhtuse näitaja keskmine muutus, kui puhtuse näitaja arvutamisel vaadeldavat tunnust ei kasutataks. Teisisõnu, mida kõrgem on selgitava tunnuse tähtsuse väärtus, seda tähtsam on selgitav tunnus mudelis.

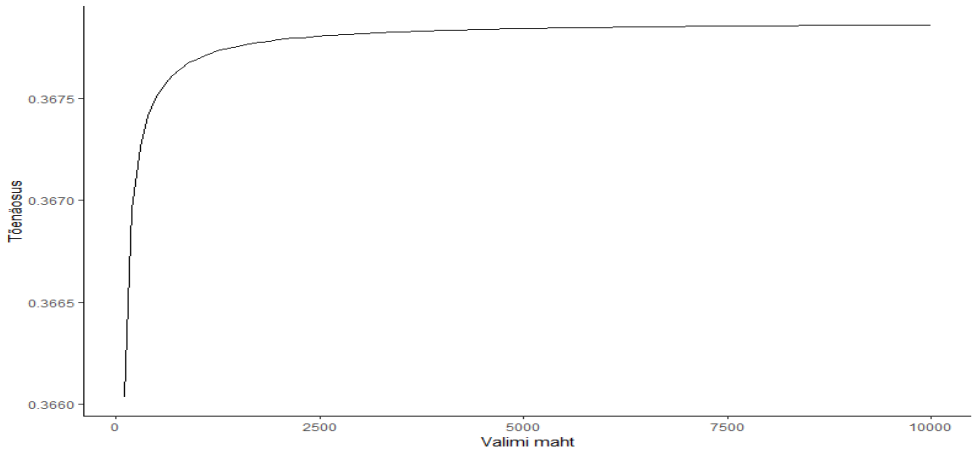
1.3 *Out-of-bag* vaatlused

Antud peatükis on kasutatud allikad Hastie jt 2017, lk 317-318.

Tõenäosus, et i-s vaatlus ei lange *bootstrap* meetodiga võetud valimisse, on.

$$\left(1 - \frac{1}{n}\right)^n$$

Jooniselt 3 on märgata, et *bootstrap* meetodiga valides jääb piisavalt suurte valimitega iga puu korral keskmiselt umbes 1/3 vaatlusi kasutamata. Neid vaatlusi nimetatakse *out-of-bag* (lüh. OOB) ehk valimist väljas olevateks vaatlusteks. Kuna neid vaatlusi ei ole juhumetsa hindamisel kasutatud, saab nende põhjal kontrol-



Joonis 3. Funktsioon $\left(1-\frac{1}{n}\right)^n$

lida hinnatud mudeli prognoosimisvõimet. Selleks vaatleme i-nda vaatluse andmeid, kasutades ainult neid puid, mille koostamistel ei kasutatud i-nda vaatluse andmeid. Prognoosimaks i-ndat vaatluse tulemust, leitakse kõikide eelnevalt mainitud puude prognooside vektor $\hat{\mathbf{y}}_\mathbf{i} = (\hat{y}_{i_1}, \cdots, \hat{y}_{i_O})$, kus O on puude arv, kus i vaatlust puude koostamisel ei kasutatud.

Uuritava vaatluse i klassi k kuulumise OOB tõenäosuse prognoosiks on $\hat{p}_{ik}^{oob} = \frac{1}{O}\sum_{j=1}^{O} I(\hat{y}_{i_j} = k)$ ja vaatluse i klassiprognoosiks on üldiselt vektori $\hat{\mathbf{y}}_\mathbf{i}$ suurima osakaaluga klassi väärtus, st $\hat{y}_i^{oob} = \mathrm{argmax}_k\, \hat{p}_{ik}^{oob}$. Antud viisil prognoosides saaks *out-of-bag* klassifitseerimisviga arvestada kui mudeli üldist viga kuna puude prognoosimiseks kasutati ainult vaatlusi, milleta puud konstrueeriti.

1.4 Mudeli täpsuse hindamine

Selles peatükis on kasutatud allikat Johnson ja Kuhn 2013, lk 256-264.

Kuna puude koostamisel luuakse kaks andmestiku, treening- ja testandmestik, siis saab juhumetsa mudeli täpsust mõõta ka testandmestiku põhjal. Erinevalt *out-of-*

bag meetoditst, saab testandmestiku korral kasutada kõiki juhumetsa hindamiseks treenitud puid, vaatlemaks, kui hästi hinnatud mudel töötab. Algoritmi prognoosivõime hindamiseks binaarsel juhul kasutatakse järgnevat 2x2 sagedustabelit.

Tabel 1. 2x2 sagedustabeli näidis

Prognoos \Reaalsus	Toimus liiklusõnnetus	Ei toimunud liiklusõnnetust
Toimus liiklusõnnetus	õiged positiivsed	valed positiivsed
Ei toimunud liiklusõnnetus	valed negatiivsed	õiged negatiivsed

Tabeli 1 põhjal leitakse mudeli täpsus kui õigesti prognoositud vaatluste ja koguvaatluste arvu suhtena ehk

$$\textit{Täpsus} = \frac{\textit{õiged postiivsed} + \textit{õiged negatiivsed}}{n}. \tag{1.5}$$

Kui uuritava tunnuse mingi klass on tugevalt domineeriv, siis näitab väga täpseid tulemusi ka selline mudel, mis prognoosib kõik vaatlused enamesinevasse klassi kuuluvaks. Näiteks, kui 99% vaatlustest kuuluvad klassi 0 ja 1% klassi 1, siis on mudeli, mis prognoosib kõik vaatlused klassi 0 kuuluvaks, täpsus 0,99. Seetõttu tasub mudeli prognoosivõime uurimisel arvesse võtta ka teisi mudeli prognoosivõimet kirjeldavaid mõõdikuid, näiteks mudeli tundlikkust ja spetsiifilisust, mis on defineeritud järgnevalt:

$$\textit{Tundlikkus} = \frac{\textit{õiged positiivsed}}{\textit{õiged positiivsed + valed negatiivsed}} \tag{1.6}$$

ja

$$\textit{Spetsiifilisus} = \frac{\textit{õiged negatiivsed}}{\textit{õiged negatiivsed} + \textit{valed positiivsed}}. \tag{1.7}$$

Seega tundlikus väljendab tõenäosust, et tegelikult toimunud sündmus on prognoositud mudeli poolt õigesti. Spetsiifilisus väljendab tõenäosust, et mitte toimunud sündmust on mudeli poolt õigesti prognoositud.

Fikseeritud mudeli täpsuse korral tekib kompromiss mudeli spetsiifilisuse ja tundlikkuse vahel. Kui mudeli tundlikkus kasvab alaneb mudeli spetsiifilisus kuna rohkem tunnuseid prognoositakse positiivseteks. Antud kompromissi hindab ROC-kõver. ROC-kõvera loomiseks hinnatakse fikseeritud spetsiifilisuse korral mudeli tundlikkust ning kuvatakse see kõverana graafikule. Mudeli täpsuse hindamiseks kasutatakse AUC parameetrit ehk pindala, mis jääb ROC-kõvera alla.

1.5 Andmestiku tasakaalustamine

Selles peatükis on kasutatud allikat Johnson ja Kuhn 2013, lk 427.

Andmestiku tasakaalustamine seisneb uuritava tunnuse klasside osakaalu ühtlustamises. Kaks laialt levinud lähenemist on alavalimine ja ülevalimine. Kui alavalimise korral võetakse tagasipanekuta juhuvalim vaatlustest, mille uuritava tunnuse väärtus kuulub enamlevinud klassi (n-ö enamusgrupp), siis ülevalimise korral võetakse tagasipanekuga juhuvalim vaatlustest, mille uuritava tunnuse väärtus kuulub vähemlevinud klassi (n-ö vähemusgrupp). Arvutusliku efektiivsuse huvides kasutatakse käesoleva bakalaureusetöö teises peatükis ainult alavalimist.

Andmete tasakaalustamine on oluline, kui prognoosida vähemlevinud klassi tunnuse väärtuseid. Klassifitseerimispuu jaotab vaatlused võimalikult ,,puhtatesse"lehtedesse. Seetõttu väljastab puu algoritm tulemused, mis prognoosib enamusgrupi väärtusi.

2 Liiklusõnnetuste modelleerimine

Antud töös on võetud eeldusteks, et

- teeilmajaamast 5 kilomeetri raadiuses on ilmastikuolud samad,
- ühe tunni jooksul ilmastikuolud oluliselt ei muutu.

2.1 Andmestiku ülevaade

Käesolevas bakalaureusetöös on kasutatud Maanteeameti teeilmajaamade ajaloolisi andmeid[1], mis on Maanteeameti poolt saadetud Tartu Ülikooli infotehnoloogia mõju-uuringute keskusele CITIS. Liiklusõnnetuste andmestik, mille bakalaureusetöö autor on samuti saanud uurimisrühmalt CITIS, sisaldab Eesti Liikluskindlustuse Fondi andmetel toimunud liiklusõnnetusi[2].

Kokku on andmeid 61 teeilmajaama kohta vahemikus 1.01.2012. - 31.12.2014. Ilmajaamade andmestikus on kokku 18 ilma ja teekatte olukorda kirjeldavat tunnust. Need tunnused on järgnevad

- teel oleva jääkihi paksus
- temperatuur millest alates hakkab kaste tekkima
- pikkuskraad
- laiuskraad
- teel oleva lumekihi paksus
- teepinna temperatuur
- maksimaalne tuulekiirus
- keskmine tuulekiirus
- tuulesuund
- nähtavus
- õhuniiskus
- õhutemperatuur
- sajuintensiivsus
- teel olevate soolade kontsentratsioon
- soolade sisaldus teel
- teepinna olukord
- teel oleva veekihi paksus
- teepinna karedus

[1] Mõningaid teeilmajaamade poolt mõõdetavaid tunnuseid saab reaalajas näha veebilehel `https://tarktee.ee/`.

[2] Andmed on olnud üleval LKF-i kaardirakenduses `http://kindlustus.maps.arcgis.com/apps/Viewer/index.html?appid=abd977aeea074631845cc67bfc3da87d`.

Antud tunnuseid käsitletakse pidevatena, väljaarvatud teepinna karedus ja tuulesuund, mis on töö autori poolt diskreetseks kodeeritud. Teepinna karedus ehk hõõrdetegur on faktortunnus, kuna antud tunnusel oli 1 284 202 puuduvat väärtust ning teepinna hõõrdetegurit ei saa arvestada teiste jaamade tulemuste keskmisena. Puuduvad väärtused on kodeeritud seejuures eraldi klassina. Tuulesuund on faktortunnus, mis on jaotatud 8 klassiks.

Algselt oli ilmaandmeid mõõdetud 10-minutilise itervalliga. Andmemahu kokkuhoimise huvides agregeeriti andmed tunnipõhiselt. Valdavalt võeti tunnustest aritmeetiline keskmine, välja arvatud nimetatud diskreetsete tunnuste korral ja maksimaalse tuulekiiruse korral, mil võeti tunni jooksul esinenud vaatluste maksimaalne väärtus.

Kuna sageli ei pruugi õnnetuse põhjuseks olla valitsevad ilmastikuolud vaid nende muutus, siis lisati andmestikku tunnused, mis kirjeldavad õhutemperatuuri ja sademete muutust võrreldes tunni aja eest eksisteerinud ilmaoludega.

Lisaks osad ilmajaamad ei salvestanud kõikide tunnuste väärtuseid, millest tulenevalt oli algselt andmestikus palju puuduvaid väärtusi. Puuduvad väärtusted asendati fikseeritud päeva ja tunni korral teiste teeilmajaamade kõikide mitte puuduvate vastavate tunnuste väärtuste keskmisena.

Liikluskindlustusfondi andmetest vaadeldi kõiki õnnetusi, mis toimusid aastatel 2012 - 2014. Nendest omakorda valiti välja liiklusõnnetused, mida võis peale muude faktorite mõjutada ka ilmastik. Töös vaadeldi õnnetusi, mis toimusid 5 kilomeetri raadiuses teeilmajaamast. Kui üks õnnetus toimus enam kui ühe teeilmajaama suhtes 5 kilomeetri raadiuses, siis valimisse läks ilmajaama andmed, mis oli õnnetuspaigale kõige lähemal.

Igal teeilmajaamal oli oma unikaalne ID ning ilmajaama asukoha koordinaadid. Liiklusõnnetuste ja teeilmajaamade andmete ühendamiseks kasutati

- õnnetuse toimumise aega (tunni täpsusega)
- õnnetuse toimumiskoha koordinaate.

Antud töös on uuritud ainult mootorsõidukitega seonduvaid õnnetusi ehk vaatluse alt on välja jäänud jalakäijate ja jalgratturitega juhtunud õnnetused. Välja on jäänud ka kõik parkimisega seonduvad ja ristmikutel toimunud õnnetused. Säärased insitente ei arvestatud mudeli koostamisel, kuna arvatavasti ei ole nende tekkepõhjus niivõrd seotud ilmastiku või liiklustihedusega.

Võib eeldada, et liiklusõnnetuste esinemist mõjutab liiklustihedus, mida kirjeldavat tunnust paraku kasutada ei olnud. Küll aga kirjeldavad liiklustihedusest tingitud efekte kaudselt kellaaeg ja teeilmajaamade asukoht.

Lisaks nendele tulemustele lisati andmestiku binaarne tunnus õnnetus, mille väärtus on 1, kui õnnetus toimus vaadeldavas piirkonnas ühe tunni jooksul ning 0, kui õnnetust ei toimunud.

Tabel 2 kirjeldab, palju on andmestikus vaatlusi, mil toimus liiklusõnnetus ja palju on vaatlusi, mil ei toimunud.

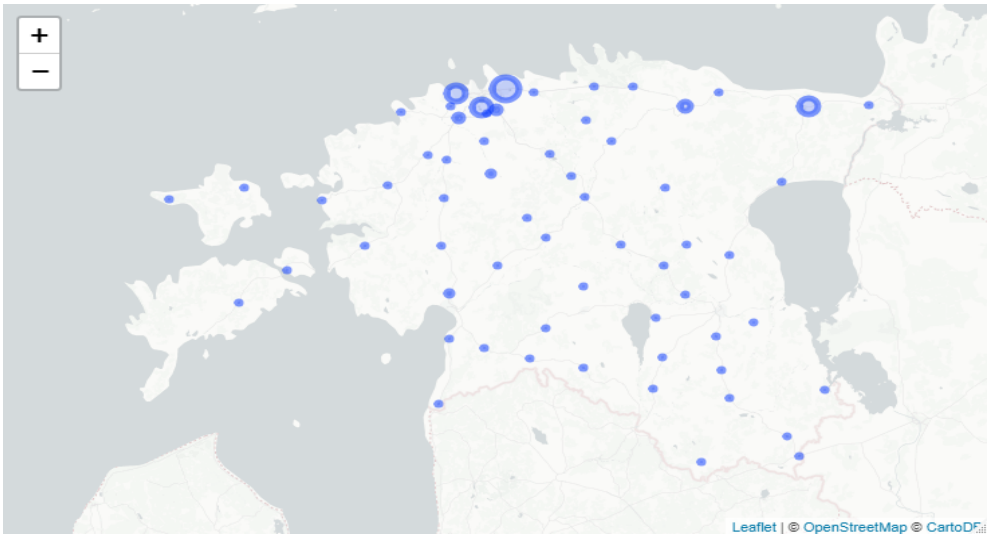
Tabel 2. Liiklusõnnetuste toimused.

Ei toimunud liiklusõnnetust	Toimus liiklusõnnetus	Kokku
1432062	4552	1436614

Seega vaadeldakse antud töös 1432062 vaatlust, millest 4552 korral toimus liiklusõnnetus. Seega on andmestikus õnnetuste toimumise ja mittetoimumise suhe $\frac{1393013}{4572} \approx 304$.

Lisaks õnnetuste arvule on kindlasti oluline õnnetuste toimumise piirkond. Kuna

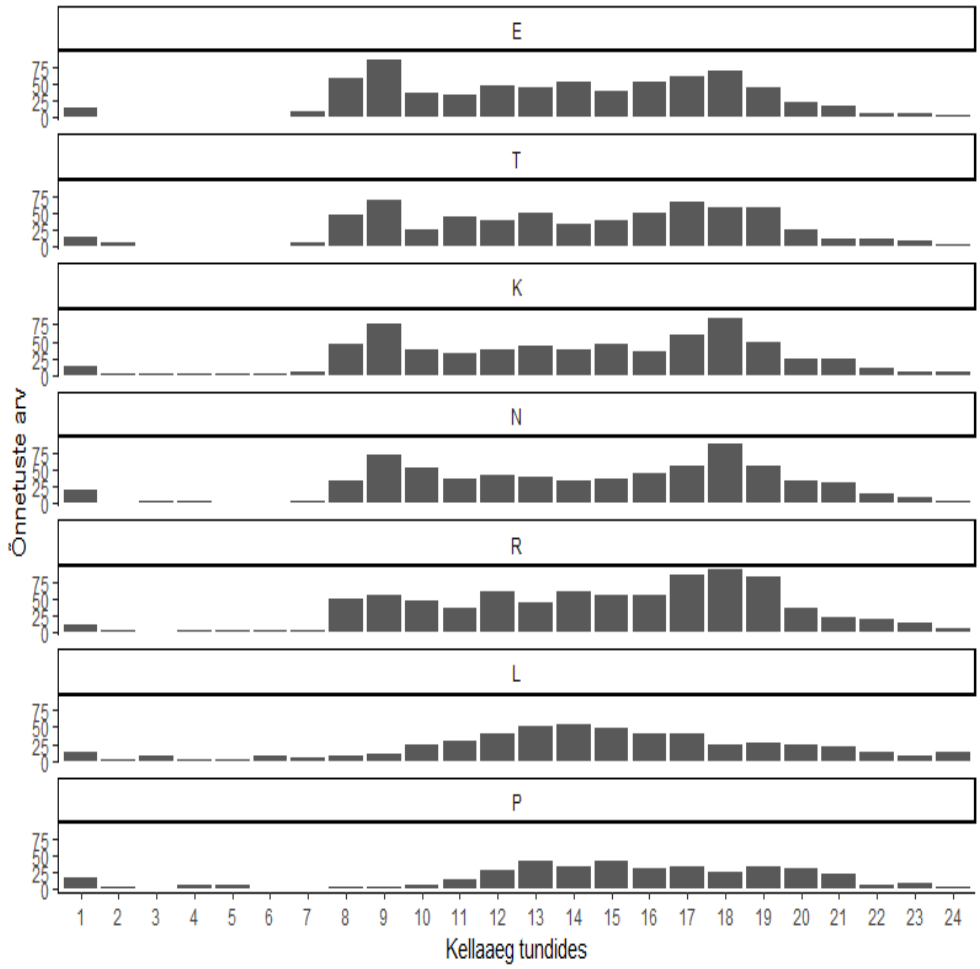
antud töös on fikseeritud õnnetuste toimumiskoht suhteliselt täpselt, siis vaatleme pigem palju on õnnetusi toimunud ilmakaamerate juures.



Joonis 4. Ilmakaamerate juures toimunud õnnetuste arv.

Mida suurem on ring joonisel 4, seda rohkem on selle teeilmajaama piirkonnas toimunud õnnetusi. Kõige rohkem õnnetusi toimub Tallinna ning selle lähiümbruses. Rakvere ja Jõhvi läheduses on samuti toimunud rohkem õnnetusi. Lisaks ei ole jooniselt näha, et Eesti põhimaanteedel, näiteks Tallinn - Tartu maanteel, oleks toimunud rohkem õnnetusi, kui muudel maanteedel.

Joonisel 5 on märgata, et tööpäevadel on liiklusõnnetuste arv suurem kui puhkepäevadel. Lisaks on märgata, et tööpäevadel on liiklusõnnetuste arv tavapärasest suurem kella 8-9 ajal ning 17-19 ajal. Antud tulemused on igati eeldatavad kuna enamus inimesi liikleb enim just nendel aegadel.



Joonis 5. Õnnetuste arv vastavalt nädalapäevadele ja tundidele

2.2 Juhumetsa rakendamine

Juhumetsa algoritmi rakendamisel on kasutatud paketti *randomForest*[3].

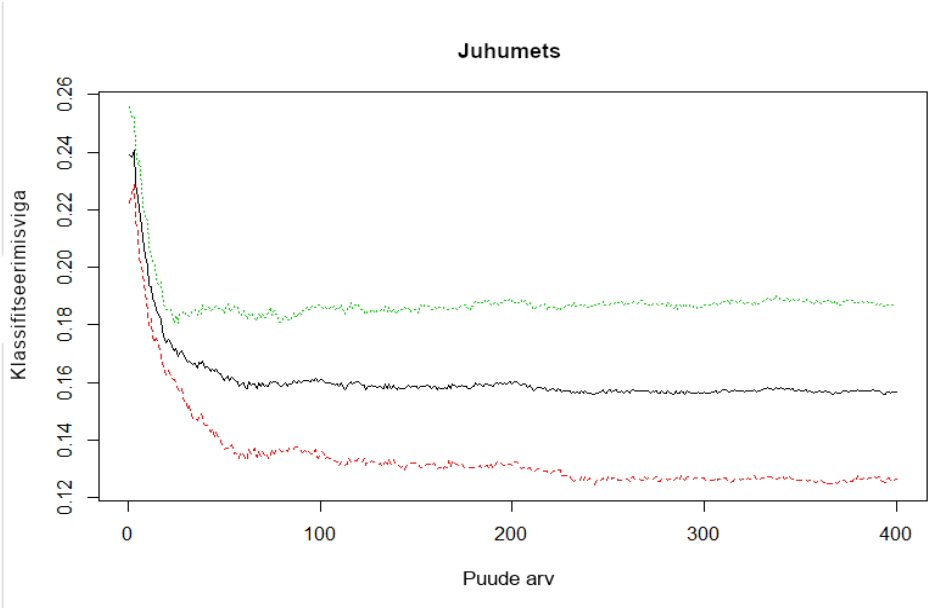
Mudelite koostamisel luuakse algsest valimist kaks lõikumatut valimit, kus treeningandmestikus on 70% algse andmestiku vaatlustest ja testandmestikus on 30% algse andmestiku vaatlustest. Seejärel tasakaalustatakse treeningandmestik liik-

[3]Pakett „randomForest" `https://cran.r-project.org/web/packages/randomForest/index.html` [07.05.2019]

lusõnnetuste toimumise järgi. Juhumetsa algoritmiga luuakse mudel kolmele treeningandmestikule, kus vastavalt on tasakaalustatud suhted õnnetuse toimumise ja mitte toimumise vahel

- 1:1,
- 1:5,
- 1:25.

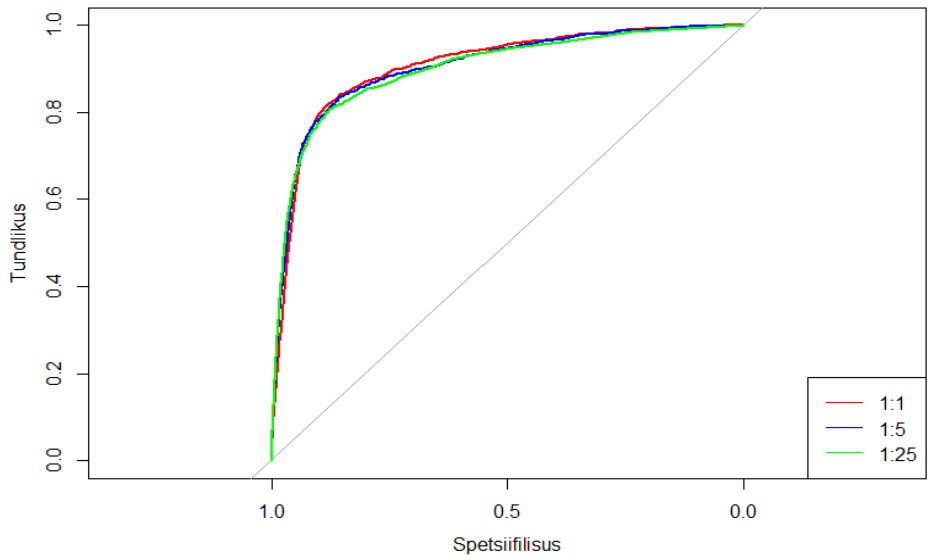
Treeningandmestikule suhtega 1:1 vastav juhumetsa veagraafik on järgnev:



Joonis 6. Juhumetsa vea graafik erineva puude arvu korral

Jooniselt 6 on punase värviga kujutatud liiklusõnnetuse mittetoimumise prognoosimise veamäära. Musta värviga kujutatud üleüldine veamäär ning rohelise värviga kujutatud liiklusõnnetuse toimumise prognoosimise veamäär.

Antud valimimahtude korral tuleb ROC-kõver järgnev



Joonis 7. ROC-kõverate graafikud testandmestiku pealt arvutatuna

Antud ROC-kõveratele vastavad AUC väärtused on järgmised:

- kui õnnetuste suhe on 1:1, siis AUC = 0,9049,
- kui õnnetuste suhe on 1:5, siis AUC = 0,9035,
- kui õnnetuste suhe on 1:25, siis AUC = 0,8992.

Kuigi tulemused on väga sarnased, on suurim AUC näitaja juhul, kui õnnetuste toimumiste ja mitte toimumiste suhe on 1:1. Antud töös tuuakse 2x2 sagedustabeli näited juhul, kui juhumetsa algoritmi koostamisel on treeningandmestikus võrdselt õnnetuste toimumisi ja mitte toimumisi. Lisaks on sagedustabelite rakendamisel lähtutud otsustuspiirist 0,5, st prognoositakse seda klassi, mida enamus hinnatud juhumetsa puudest prognoosib.

Kui andmestikus on liikusõnnetuste toimumiste ja mitte toimumiste arv võrdne, siis mõlemad klasse esindab treeningvalimis 3169 vaatlust. Kokku on valimis $2 \cdot 3169 = 6338$ vaatlust.

Tabel 3. Treeningandmestiku põhjal koostatud prognooside ja tegelike tulemuste sagedustabel

Prognoos \Reaalsus	Toimus	Ei toimunud
Toimus	3169	0
Ei toimunud	0	3169

Nagu näha, on treeningandmestiku pealt väljastatud prognoosid on tugevalt ülesobitatud (tabel 3), kuna juhumetsa algoritmi korral kasvatakase iga puu nii suureks, et igas lehes on üks vaatlus. Seega antud olukorras vaadeldakse, kas i-s tunnus langeb i tunnuse jaoks loodud lehte. See juhtub alati ning sellest tingituna saadakse alati ideaalne olukord.

Tabelis 4 on kajastatud tulemused, kui prognoosimiseks on kasutatud küll treeningandmestikku, kuid väljastatakse OOB prognoose. Saadud 2x2 sagedustabel näeb välja järgnev.

Tabel 4. OOB prognooside ja tegelike tulemuste sagedustabel

Prognoos \Reaalsus	Toimus	Ei toimunud
Toimus	2577	401
Ei toimunud	592	2768

Out-of-bag meetodiga prognoositi õigesti $\frac{2768+2577}{2768+592+407+2577} \approx 84,3\%$ juhtudest. Erinevalt vahetult treeningandmestiku põhjal prognoosimisest, kasutati antud olukorras vaatluste prognoosimiseks puid, mis vastavaid vaatlusi ei ole treenimiseks

kasutanud. Seetõttu ei toimu ka tulemuste ülesobitamist. Küll aga ei väljenda tabelis 4 esitatud väärtused tegelikkust selles mõttes, et prognoosimiseks kasutatud andmestikus on mõlema klassi osakaalud võrdsed.

Tegelikkusele vastavaid tulemusi väljendab aga tabel 5, mille korral on prognoositud testandmestikku kasutades.

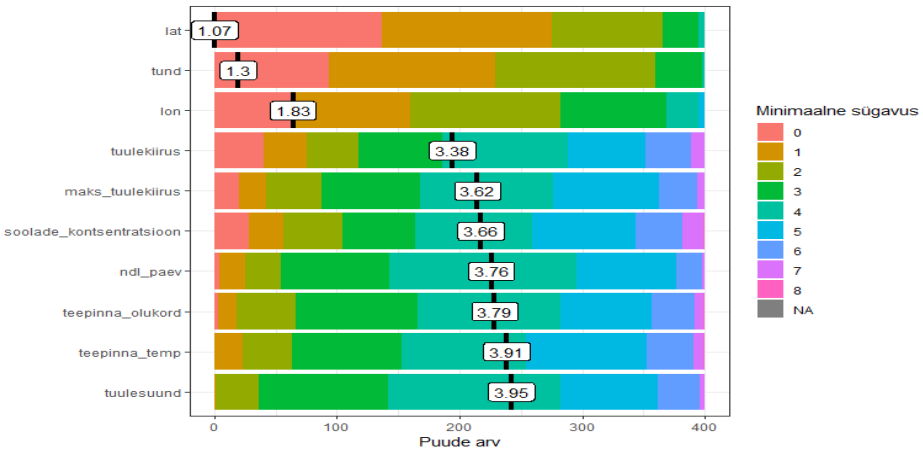
Tabel 5. Testandmestiku põhjal koostatud prognooside ja tegelike tulemuste sagedustabel

Prognoos \Reaalsus	Toimus	Ei toimunud
Toimus	1165	54842
Ei toimunud	218	374760

Algoritm prognoosis õigesti $374760 + 1165 = 374876$ vaatlust. Valepositiivseid vaatlusi on 54842 ning valenegatiivseid vaatlusi on 218. Seega on antud juhumetsa prognoosi täpsus $\frac{374760+1165}{374760+1165+54842+218} \approx 87,2\%$.

Eelnev pole otseselt välja selgitanud, mis tegurid võiksid liiklusõnnetuste tõenäosust suurendada. Selle selgitamiseks kasutatakse statistika tarkvara R paketti „randomForestExplainer"[4]. Järgneval joonisel väljastatakse 10 kõige mõjukamat tunnust liiklusõnnetuse esinemisele.

[4]Pakett „randomForestExplainer" `https://cran.r-project.org/web/packages/randomForestExplainer/randomForestExplainer.pdf`



Joonis 8. 10 kõige mõjukamat tunnust liiklusõnnetuste esinemisele.

Joonisel 8 märgib värv tunnuse minimaalset sügavust juhusliku metsa algoritmis koostatud puudes. Tunnuse keskmist sügavust puudes määrab tunnuste nimede juures olev number.

Mida lähemale on selgitav tunnus juurele, seda tähtsam see tunnus kirjeldamaks prognoositavat tunnust. Seejuures juureks nimetatakse puu esimest lõike kohta. Joonis 8 põhjal on liiklusõnnetuse toimumise prognoosimisel kõige tähtsamad tunnused laiuskraad, tund ning pikkuskraad ehk liiklustihedust kaudselt hõlmavad tunnused. Antud tulemus on oodatav kuna joonis 4 põhjal on näha, et asukohal oli mõju õnnetuse toimumisel. Samuti viitab joonis 5, et liiklustihedus mõjutab õnnetuste arvu.

Seega ei mõjuta liiklusõnnetuse toimumist mitte niivõrd ilmastikuolud ja teeolud, vaid liiklustihedus.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli uurida, kuidas ilmastik ja teeolud mõjutavad liiklusõnnetuste esinemist.

Töö esimeses osa anti ülevaade otsutuspuu ja juhumetsa meetodite olemusest. Töö teises osas anti ülevaade andmestikust ning rakendati juhumetsa algoritmi antud andmestikule.

Töö tulemusena selgus, et liiklusõnnetusi mõjutavad enim sõiduki asukoht ning kellaeg, millal sõidetakse, mitte niivõrd ilmastik ja teeolud.

Tulevikus on võimalik tööd edasi arendada, hindamaks juhumetsa ka ülevalimise korral. Antud töös seda arvutusressursist tulenevalt ei käsitletud.

Kasutatud kirjandus

[1] James, G., Witten, D., Hastie, T., Tibshirani, R.(2017) *An Introduction to Statistical Learning with Applications in R*, Springer, `http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Seventh%20Printing.pdf`, Vaadatud, [06.05.2019]

[2] Kuhn, M.,Johnson, K.(2013), Applied Predictive Modeling, Springer

[3] Hastie, T., Tibshirani, R., Friedman, J. (2009) The Elements of Statistical Learning Data Mining, Inference, and Prediction Second Edition https://web.stanford.edu/ hastie/Papers/ESLII.pdf

Lisad

Lisa 1. Ilmastiku ja liikluskindlustusfondi andmestiku parandamine.

```
    #Library

       #
#
   ##########################################################################


library(lubridate)
library(magrittr)
library(dplyr)
library(ggplot2)
library(data.table)
library(tidyverse)
library(stringr)
library(data.tree)
library(forcats)
library(zoo)
library(readxl)
library(geosphere)
library(readr)
library(gtools)
library(randomForest)
library(caret)
library(tree)
library(pROC)
library(randomForestExplainer)
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

#LIIKLUSKINDLUSTUSE andmete sisse lugemine ja filtreerimine
```

```
    juhtumi j rgi.

liiklusKindlustuseAndmed <- fread("lkf.csv", sep = ";",
   encoding = "UTF-8")
liiklusKindlustuseAndmed <- liiklusKindlustuseAndmed %>%
  mutate(time = as.POSIXct(time, format = "%Y-%m-%d␣%H:%M:%
     OS")) %>%
  filter(year(time) %in% c(2012, 2013, 2014))

huvipakkuvad_onnetused <- c(
  "Liiklus nnetused␣teel␣ja␣ristmikul:␣Muud:␣Kokkup rge␣
     vastutuleva␣s idukiga.␣v hemalt␣ ks ␣ei␣mahu␣enda␣
     suunav ndisse",
  "Liiklus nnetused␣teel␣ja␣ristmikul:␣M das it.␣
     reastumine.␣k rvalekaldumine:␣K rvalritta␣kaldumine␣
     reastumissoovita.␣kokkup rge␣seal␣liikujaga",
  "Liiklus nnetused␣teel␣ja␣ristmikul:␣M das it.␣
     reastumine.␣k rvalekaldumine:␣M das it.␣
     kokkup rge␣samas-␣v i␣vastassuunas␣liikujaga",
  "Liiklus nnetused␣teel␣ja␣ristmikul:␣M das it.␣
     reastumine.␣k rvalekaldumine:␣Reastumine.␣kokkup rge
     ␣k rvalreas␣liikujaga",
  "Liiklus nnetused␣teel␣ja␣ristmikul:␣M das it.␣
     reastumine.␣k rvalekaldumine:␣Vastassuunda␣kaldumine␣
     m das idusoovita.␣kokkup rge␣seal␣liikujaga",
  "Liiklus nnetused␣teel␣ja␣ristmikul:␣Ristmiku␣ letamine
     ␣ja␣p rded:␣Kokkup rge␣ristuval␣teel␣liikujaga",
  "Liiklus nnetused␣teel␣ja␣ristmikul:␣Ristmiku␣ letamine
     ␣ja␣p rded:␣P rdel.␣kokkup rge␣taganttulijaga",
  "Liiklus nnetused␣teel␣ja␣ristmikul:␣Ristmiku␣ letamine
     ␣ja␣p rded:␣P rdel.␣kokkup rge␣vastutulijaga",
  "Liiklus nnetused␣teel␣ja␣ristmikul:␣Ristmiku␣ letamine
     ␣ja␣p rded:␣Tagasip rdel.␣kokkup rge␣samas-␣
     v i␣vastassuunas␣liikujaga",
  "Liiklus nnetused␣teel␣ja␣ristmikul:␣Tagant␣otsas it:␣
```

```
   Tagant␣otsas it␣ees␣liikuvale␣v i␣peatunud␣
   s idukile",
"MUUD:Juhitavuse␣kaotus␣teel:",
"MUUD:Otsas it␣ohutussaarele:",
"MUUD:Muud␣(mujal␣kirjeldamata␣L ␣situatsioon):",
#"MUUD:Otsas it paremale teeservale pargitud s idukile
   :",
#"MUUD:Otsas it vasakule teepervele pargitud s idukile
   :",

"RISTUVAD␣S IDUSUUNAD:OTSE:Kokkup rge␣ristmikul",
"RISTUVAD␣S IDUSUUNAD:OTSE:Kokkup rge␣rongiga",
"RISTUVAD␣S IDUSUUNAD:P  RDEL:Kokkup rge␣
   paremp rdel␣vastutulijaga",
"RISTUVAD␣S IDUSUUNAD:P  RDEL:Paremp rdel␣ette␣
   keeramine␣tagant␣tulijale",
"RISTUVAD␣S IDUSUUNAD:P  RDEL:Vasakp rdel␣ei␣anna␣
   teed",
"RISTUVAD␣S IDUSUUNAD:P  RDEL:Vasakp rdel␣tagant␣
   tulijale␣ette␣keeramine",
"RISTUVAD␣S IDUSUUNAD:P  RDEL: heaegne ␣vasakp re",
"SAMA␣S IDUSUUND:OTSE:Kokkup rge␣s itu␣alustavaga",
"SAMA␣S IDUSUUND:OTSE:K lgkokkupuude",
"SAMA␣S IDUSUUND:OTSE:M  das it",
"SAMA␣S IDUSUUND:OTSE:Reavahetus␣paremale",
"SAMA␣S IDUSUUND:OTSE:Reavahetus␣vasakule",
"SAMA␣S IDUSUUND:OTSE:Tagant␣otsas it␣ees␣liikuvale",
"SAMA␣S IDUSUUND:OTSE:Tagant␣otsas it␣ees␣pidurdajale",
"SAMA␣S IDUSUUND:OTSE:Tagant␣otsas it␣takistuse␣ees␣
   peatunud␣s idukile",
"SAMA␣S IDUSUUND:OTSE:Topeltm das it",
"SAMA␣S IDUSUUND:RISTMIKUL:Muu␣kokkup rge␣vasak␣
   p  rdel",
"SAMA␣S IDUSUUND:RISTMIKUL:Muu␣ nnetuse ␣p  rdel␣
   paremale",
```

```
"SAMA␣S IDUSUUND:RISTMIKUL:Tagant␣otsas it␣paremale␣
   keerajale",
"SAMA␣S IDUSUUND:RISTMIKUL:Tagant␣otsas it␣vasakule␣
   p  rdel",
"SAMA␣S IDUSUUND:RISTMIKUL:Tagasip  re␣tagant␣tuleva␣
   s iduki␣ette",
"TEELT␣V LJAS IT:Teelt␣v lja␣paremale:",
"TEELT␣V LJAS IT:Teelt␣v lja␣vasakule:",
"TEELT␣V LJAS IT:Teelt␣v ljas it␣paremale␣paremas␣
   kurvis:",
"TEELT␣V LJAS IT:Teelt␣v ljas it␣paremale␣vasakkurvis
   :",
"TEELT␣V LJAS IT:Teelt␣v ljas it␣ristmikul:",
"TEELT␣V LJAS IT:Teelt␣v ljas it␣vasakule␣paremkurvis
   :",
"TEELT␣V LJAS IT:Teelt␣v ljas it␣vasakule␣vasakkurvis
   :",
"VASTASSUUNAS:OTSE:Kokkup rge␣kurvis",
"VASTASSUUNAS:OTSE:Kokkup rge␣m  das idul␣kurvis",
"VASTASSUUNAS:OTSE:Kokkup rge␣m  das idul␣sirgel",
"VASTASSUUNAS:OTSE:Kokkup rge␣sirgel",
"VASTASSUUNAS:OTSE:Teelt␣v ljas it␣kokkup rke␣
   v ltimiseks",
"VASTASSUUNAS:RISTMIKUL:Kokkup rge␣erisuunalistel␣
   p  retel",
"VASTASSUUNAS:RISTMIKUL:Kokkup rge␣p  rdel␣samasse␣
   suunda",
"VASTASSUUNAS:RISTMIKUL:Muu␣kokkup rge␣vastutulijaga␣
   paremp  rdel",
"VASTASSUUNAS:RISTMIKUL:Tagasip  re␣vastutulija␣ette",
"VASTASSUUNAS:RISTMIKUL:Vasakule␣p  rdel␣kokkup rge␣
   vastuliikujaga",
"  hesidukinnetused  ␣Asja.␣ehitise␣v i␣rajatise␣
   kahjustamine:␣Muu␣asja␣kahjustamine",
```

```
  "   hesidukinnetused   ␣Asja.␣ehitise␣v i␣rajatise␣
     kahjustamine:␣Muu␣ehitise␣v i␣rajatise␣kahjustamine",
  "   hesidukinnetused   ␣Asja.␣ehitise␣v i␣rajatise␣
     kahjustamine:␣Tee␣v i␣teerajatise␣kahjustamine",
  "   hesidukinnetused   ␣Muud:␣Kokkup rge␣rongiga",
  "   hesidukinnetused   ␣Muud:␣Otsas it␣teel␣olnud␣
     takistusele␣ja/v i␣s iduki␣ mberpaiskumine ␣teel",
  "   hesidukinnetused   ␣Muud:␣Reisijale␣kahju␣tekitamine.
     ␣kokkup rge␣v i␣s iduki␣ mberpaiskumine ␣puudub",
  "   hesidukinnetused   ␣Muud:␣Teelt␣v ljas it"
)


liiklusKindlustuseAndmed <- liiklusKindlustuseAndmed %>%
  filter(situatsioon %in% huvipakkuvad_onnetused)
#Siin on liiga paju  nnetusi
liiklusKindlustuseAndmed <- liiklusKindlustuseAndmed %>%
  mutate(time2 = strftime(time, format = "%H:%M:%OS")) %>%
  filter(time2 != "03:33:00")



####################################

#Ilmajaamade sisselugemine
########################################
#Ilmajaamade koordinaadid
Ilmajaamad <- as.data.frame(read_excel("Ilmajaamad/HO␣
   seadmete␣nimekiri␣2016␣-␣GPS␣koordinaadid.xlsx") %>%
   filter(!is.na(name)))
#Ilmajaamade ID ja koht
Ilmajaamade_id <- as.data.frame(read_delim("Ilmajaamad.csv"
   ,
                                           ";", escape_
                                              double =
```

```
                                                    FALSE,
                                                locale = locale(
                                                    encoding = "
                                                    WINDOWS-1252"
                                                    ),
                                                trim_ws = TRUE,
                                                    col_names = c
                                                    ("id", "koht"
                                                    )))

View(Ilmajaamad)
Ilmajaamade_id <- Ilmajaamade_id %>%
  mutate(id = stringr::str_trim(id),
         koht = stringr::str_trim(koht))

Ilmajaamade_id$id = as.numeric(substr(Ilmajaamade_id$id,
   start = 5, stop = 6))

for(row in 1:nrow(Ilmajaamad)){
  praegu = Ilmajaamad[row,]
  a = unlist(strsplit(praegu$name, split = '␣'))
  #Kangrus olid numbriga indikaatorid ka.
  if(length(a) > 3){
    lisatav = a[c(1,2)]
    paste(lisatav,collapse="")
    Ilmajaamad$koht[row] = paste(lisatav,collapse="")
  }else {
    Ilmajaamad$koht[row] = a[1]
  }
}

Ilmajaamad$koht[Ilmajaamad$koht == "V ike-Rakke"] = "V-
   Rakke"

Ilmajaamad <- left_join(Ilmajaamad, Ilmajaamade_id, by = "
```

```
   koht")

#Eemaldan NA

Ilmajaam <- Ilmajaamad %>%
  filter(!is.na(id)) %>%
  select(lat, lon, id)

# summary(Ilmajaam)
# colnames(Ilmajaam)[3] <- "kaameraId"
# View(Ilmajaam)
# table(Ilmajaam$kaameraId)
# Ilmajaam$kaameraId <- as.factor(Ilmajaam$kaameraId)
#
# andmestik1 <- left_join(andmestik, Ilmajaam, by = "
   kaameraId")
# andmestik1$kaameraId <- as.factor(andmestik1$kaameraId)
# table(andmestik$kaameraId)
# andmestik <- select(andmestik1, -c(V1, kaameraId, paev,
   kuu))
# summary(andmestik1)

#Funktsioon kaugusKaamerast: Leiab onnetuse ja kaamera
   vahelise kauguse. Tagastab onnetused, mis on 5km
   raadiuses
###################################

kaugusKaamerast <- function(id, piirKaugus) {
  indeks <- which(Ilmajaam$id %in% id)
  lon <- Ilmajaam$lon[indeks]
  lat <- Ilmajaam$lat[indeks]
  oigedOnnetused <- liiklusKindlustuseAndmed
  oigedOnnetused$kaameraId <- id
  oigedOnnetused$kaugus <- apply(liiklusKindlustuseAndmed[,
     c("lon","lat")], 1, function(x) distm(c(x["lon"], x["
```

```
      lat"]), c(lon,lat), fun =  distHaversine))
  return(oigedOnnetused[oigedOnnetused$kaugus < piirKaugus
      ,])
}


onnetused <- list()
for (i in 1:length(Ilmajaam$id)) {
  onnetused[[i]] <- kaugusKaamerast(Ilmajaam$id[i], 5000)
  print(i)
}

onnetusedKaameraJuures <- bind_rows(onnetused)
# kui mitu kaamerat 5 km sees, siis valime l hima

onnetusedKaameraJuures <- onnetusedKaameraJuures %>%
  group_by(juhtum) %>%
  top_n(-1, kaugus) %>%
  ungroup()

summary(onnetusedKaameraJuures)

#Kontroll
table(as.data.frame(table(onnetusedKaameraJuures$juhtum))$
    Freq)

onnetusedKaameraJuures <- onnetusedKaameraJuures %>%
  select(-kahju_kokku, -kahju_liik, -kulu_asja, -kulu_isiku
      )

#Et saaks hiljem ilmastiku andmestikuga yhendada.
valid_column_names <- make.names(names=names(
    onnetusedKaameraJuures), unique=TRUE, allow_ = TRUE)
names(onnetusedKaameraJuures) <- valid_column_names
```

```
save(onnetusedKaameraJuures, file = "onnetusedKaameraJuures
   _tmp")
load("onnetusedKaameraJuures_tmp")


# Kui samal ajal samas kohas mitu juhtumit, siis v tame
   arvesse vaid  he

onnetusedKaameraJuures <- onnetusedKaameraJuures %>%
  group_by(time, lon, lat) %>%
  mutate(N_juhtum = n()) %>%
  slice(1) %>%
  ungroup() %>%
  select(-N_juhtum)

write.csv(onnetusedKaameraJuures, file = "
   onnetusedKaameraJuures_v2.csv", row.names = F)
#onnetusedKaameraJuures <- read.csv("onnetusedKaameraJuures
   _v2.csv")
###################################

#Ilmade sisse lugemine
#
   ##########################################################################


#Formaat
formaat <- fread("formaat.csv")

formaadi_veerunimed <- make.names(names=names(formaat),
   unique=TRUE, allow_ = TRUE)

eemalda_veerud <- c("Time.Dev", "Closed", "V15", "V17", "
```

```
   Relays.on.off",
                    "V20", "V21", "V22", "Air.temperature.
                       trend",
                    "V24", "Configurable.measurement.2",
                    "Configurable.measurement.3", "V30", "V
                       31", "Freezing.point..Tr",
                    "T.base", "V40", "Coverage.value.low",
                       "Coverage.value.high",
                    "Surface.temperature.1", "Ground.
                       temperature.1",
                    "Conductivity.1", "Surface.signal.1",
                    "Black.ice.frequency.1", "Freezing.
                       point..Tr.1",
                    "Surface.status.1", "V54", "V55",
                    "Concentration.1", "Amount.of.chemical.
                       1",
                    "Freezing.point.1", "Water.thickness.1"
                       ,
                    "Coverage.value.high.1", "Surface.
                       temperature.2",
                    "V63", "V64", "V65", "V66", "V67",
                    "Surface.status.2", "V69", "Level.of.
                       grip",
                    "V71", "V72", "V73",
                    "Amount.of.water", "Amount.of.ice", "
                       Amount.of.snow",
                    paste0("V",77:91),
                    "NWS.codes", "Housekeeping.status",
                    "Rain.off.on", "Configurable.
                       measurement", "Freezing.point")


#Ilmajaamade failid
```

```
Ilmaandmete_kleepija_v2 <- function(ilmaFailid){
  datalist = list()
  D = matrix(nrow = length(ilmaFailid), ncol = 2)
  for (i in 1:length(ilmaFailid)) {
    kaameraId <- as.numeric(substr(ilmaFailid[i], start = 8
       , stop = 9))
    if (kaameraId %in% Ilmajaam$id){
      ilm <- fread(ilmaFailid[i], fill = T)
      names(ilm) = formaadi_veerunimed[1:ncol(ilm)]
      ilm$paev = as.Date(substr(ilmaFailid[i], start = 1,
         stop = 6), format = "%y%m%d")
      ilm$kaameraId = kaameraId
      datalist[[i]] = ilm
      D[i,] = dim(ilm)
    }

    if (i %% 500 == 0) print(i)
  }
  IlmaAndmed <- bind_rows(datalist)
  return(IlmaAndmed)
}

#Ilmaandmete sisselugemine

wd_yldine <- getwd()
setwd(wd_yldine)
ilmaFailid2012 <- list.files("2012\\teeilm␣1␣(TTK)",
   recursive = T)
setwd(paste0(wd_yldine, "//2012//teeilm␣1␣(TTK)"))
Ilm2012 <- Ilmaandmete_kleepija_v2(ilmaFailid2012)
#apply(Ilm2012, 2, function(x) sum(is.na(x)))

#2013
setwd(wd_yldine)
ilmaFailid2013 <- list.files("2013\\teeilm␣1␣(TTK)",
```

```
   recursive = T)
setwd(paste0(wd_yldine, "//2013//teeilm␣1␣(TTK)"))
Ilm2013 <-Ilmaandmete_kleepija_v2(ilmaFailid2013)

#2014
setwd(wd_yldine)
ilmaFailid2014 <- list.files("2014\\teeilm␣1␣(TTK)",
   recursive = T)
setwd(paste0(wd_yldine, "//2014//teeilm␣1␣(TTK)"))
Ilm2014 <-Ilmaandmete_kleepija_v2(ilmaFailid2014)
setwd(wd_yldine)

# Andmete  hendamine  (10 minutilised andmed)
Ilm_10min <- bind_rows(Ilm2012, Ilm2013, Ilm2014)


# lisame tunnuse tund

Ilm_10min <- Ilm_10min %>%
  mutate(tund = as.numeric(substr(Time, 1,2)))

# Andmete tunni peale agregeerimine
Ilm <- Ilm_10min %>%
  group_by(kaameraId, paev, tund) %>%
  summarise(ohutemp = mean(Air.temperature, na.rm = TRUE),
            ohuniiskus = mean(Humidity, na.rm = TRUE),
            kastepunkt = mean(Dew.point, na.rm = TRUE),
            #sajab = max(Rain.off.on, na.rm = TRUE),
            tuulekiirus = mean(Wind.speed.10.min.avg, na.rm
                = TRUE),
            tuulesuund = mean(Wind.direction.10.min.avg, na
               .rm = TRUE),
            sajuhulk = mean(Precipitation.sum, na.rm = TRUE
               ),
            sajuintensiivus = mean(Rain.intensity, na.rm =
```

```
    TRUE),
lumekorgus = mean(Snow.height, na.rm = TRUE),
n2htavus = mean(Visibility, na.rm = TRUE),
#ohurohk = mean(Configurable.measurement, na.rm
    = TRUE),
j22kihi_paksus = mean(Configurable.measurement.
    1, na.rm = TRUE),
lumekihi_paksus = mean(General.status, na.rm =
    TRUE),
saju_klass = median(Rain.class, na.rm = TRUE),
maks_tuulekiirus = max(Wind.speed.max...10.min
    ., na.rm = TRUE),
karedus = mean(Wind.dir..max...10.min., na.rm =
    TRUE),
teepinna_temp = mean(Surface.temperature, na.rm
    = TRUE),
teekatte_temp = mean(Ground.temperature, na.rm
    = TRUE),
teepinna_juhtivus = mean(Conductivity, na.rm =
    TRUE),
teepinna_signaal = mean(Surface.signal, na.rm =
    TRUE),
mustaj22_sagedus = mean(Black.ice.frequency, na
    .rm = TRUE),
teepinna_olukord = median(Surface.status, na.rm
    = TRUE),
soolade_kontsentratsioon = mean(Concentration,
    na.rm = TRUE),
soolade_sisaldus = mean(Amount.of.chemical, na.
    rm = TRUE),
#kylmumistemperatuur = mean(Freezing.point, na.
    rm = TRUE),
veekihi_paksus = mean(Water.thickness, na.rm =
    TRUE),
kattekiht_ohuke = median(Coverage.value.low.1,
```

```
                na.rm = TRUE)
  ) %>%
  ungroup()


#Ilmaandmete ja  nnetuste  kokkusidumine
onnetusedKaameraJuures <- as.data.frame(
   onnetusedKaameraJuures)
onnetusedKaameraJuures <- onnetusedKaameraJuures %>%
  mutate(paev = as.Date(time),
         tund = hour(time),
         onnetus = 1) %>%
  select(paev, tund, kaameraId, onnetus)

andmestik <- left_join(Ilm, onnetusedKaameraJuures, by = c(
   "paev", "tund", "kaameraId"))

andmestik$onnetus[is.na(andmestik$onnetus)] = 0
andmestik$onnetus = as.factor(andmestik$onnetus)
write.csv(andmestik, file = "yhendatud_andmestik.csv", row.
   names = F)

andmestik$kuu <- as.factor(lubridate::month(andmestik$paev)
   )
andmestik$ndl_paev <- as.factor(weekdays(as.Date(andmestik$
   paev), abbreviate = T))

andmestik <- select(andmestik, -c(teekatte_temp, teepinna_
   juhtivus, teepinna_signaal,
                                  kattekiht_ohuke, sajuhulk
                                     , saju_klass, mustaj22
                                     _sagedus))

#Andmestiku parandamine
andmestik$ohuniiskus[andmestik$ohuniiskus<0] <- NA
```

```
andmestik$ohuniiskus[andmestik$ohuniiskus>100] <- NA

#Soolade kontsentratsioon  le  0
andmestik$soolade_kontsentratsioon[andmestik$soolade_
   kontsentratsioon<0] <- NA
andmestik$sajuintensiivus[andmestik$sajuintensiivus < 0] <-
    NA
#Eestis m  detud maksimaalne temperatuur on 35,6. Eeldame
   , et 2012-2014 v is p ikse k es ka nii soe olla
andmestik$ohutemp[andmestik$ohutemp>35] <- NA
#Kui lumekihti ei m  deta , siis on suvi ja suvel on lume
   ja  j    paksus 0 ehk NA -> 0
andmestik$lumekihi_paksus[is.na(andmestik$lumekihi_paksus)]
    <- NA
andmestik$j22kihi_paksus[is.na(andmestik$j22kihi_paksus)]
   <- NA
#Teepinna temp
andmestik$teepinna_temp[andmestik$teepinna_temp>50] <- NA
andmestik$teepinna_temp[andmestik$teepinna_temp< -29] <- NA
#Juhtivus
#andmestik$teepinna_juhtivus[andmestik$teepinna_juhtivus>12
   ] <- NA
#Osadel oli suund suht metsas, v tasin t isp  rded maha
andmestik$tuulesuund[andmestik$tuulesuund>360] <- NA
#-inf -> NA
andmestik$maks_tuulekiirus[andmestik$maks_tuulekiirus<0] <-
    NA
andmestik$kastepunkt[andmestik$kastepunkt < -20] <- NA
andmestik$kastepunkt[andmestik$kastepunkt > 35] <- NA

#Soolad
andmestik$soolade_kontsentratsioon[andmestik$soolade_
   kontsentratsioon < 0] <- NA
andmestik$karedus <- cut(andmestik$karedus, seq(0,0.9, 0.1)
   )
```

```
andmestik$karedus <- as.character(andmestik$karedus)
andmestik$karedus[is.na(andmestik$karedus)] <- "puudu"
andmestik$karedus <- as.factor(andmestik$karedus)
andmestik$paev <- as.Date(andmestik$paev)

# Lisame juurde teeilmajaama koordinaadid

andmestik <- left_join(andmestik, Ilmajaam, by = c("
   kaameraId" = "id"))

# Eemaldame duplikaatread
andmestik <- andmestik %>%
  group_by(kaameraId, paev, tund) %>%
  slice(1) %>%
  ungroup()

andmestik <- as.data.frame(andmestik)

andmestik_pikk <- andmestik %>%
  select(-karedus) %>%
  gather(key = tunnus, value = vaartus, -kuu, -ndl_paev, -
     onnetus, -kaameraId, -paev, -tund, -lat, -lon)

andmestik_pikk <- andmestik_pikk %>%
  arrange(kaameraId, paev, tund)

# NA-de asendamine
andmestik_pikk$vaartus <- as.numeric(andmestik_pikk$vaartus
   )

andmestik_pikk <- andmestik_pikk %>%
  filter(tunnus != "karedus") %>%
  group_by(tunnus, paev, tund) %>%
  mutate(tunnuse_keskmine = mean(vaartus, na.rm = TRUE))
     %>%
```

```
  ungroup()

andmestik_pikk <- andmestik_pikk %>%
  mutate(vaartus_imput = ifelse(is.na(vaartus), tunnuse_
     keskmine, vaartus))


andmestik_imput <- andmestik_pikk %>%
  select(-vaartus, -tunnuse_keskmine, -onnetus, -kuu, -ndl_
     paev, -lat, -lon) %>%
  spread(key = tunnus, value = vaartus_imput)

andmestik_imput <- left_join(andmestik_imput,
                             andmestik %>% select(kaameraId
                                , paev, tund,
                                                  karedus,
                                                     onnetus
                                                     , kuu,
                                                      ndl_
                                                     paev,
                                                     lat,
                                                     lon),
                             by = c("kaameraId", "paev", "
                                tund"))

andmestik_imput <- na.omit(andmestik_imput)

# Nimetame uuesti andmestikuks
andmestik <- andmestik_imput %>%
  arrange(kaameraId, paev, tund) %>%
  group_by(kaameraId) %>%
  mutate(temp_erinevus = c(0,diff(ohutemp)),
         sade_erinevus = c(0,diff(sajuintensiivus))) %>%
  ungroup()
```

```
andmestik$temp_erinevus[andmestik$temp_erinevus > 10] <- 10
andmestik$temp_erinevus[andmestik$temp_erinevus < -10] <- -
   10
andmestik$sade_erinevus[andmestik$sade_erinevus > 100] <- 1
   00
andmestik$sade_erinevus[andmestik$sade_erinevus < -100] <-
   -100
andmestik$karedus <- as.factor(andmestik$karedus)
andmestik$ndl_paev <- as.factor(andmestik$ndl_paev)
andmestik$tund <- as.factor(andmestik$tund)
andmestik$tuulesuund <- cut(andmestik$tuulesuund, seq(0,360
   ,45))
andmestik$tuulesuund <- as.character(andmestik$tuulesuund)
andmestik$tuulesuund[is.na(andmestik$tuulesuund)] = "puudub
   "
andmestik$tuulesuund <- as.factor(andmestik$tuulesuund)
andmestik$onnetus <- as.factor(andmestik$onnetus)
andmestik$kuu <- as.factor(andmestik$kuu)
andmestik$kaameraId <- as.factor(andmestik$kaameraId)
andmestik$tund <- as.numeric(andmestik$tund)
andmestik$tund <- as.factor(andmestik$tund)

andmestik <- select(andmestik, -c(kaameraId, paev, kuu))

save(andmestik, file = "andmestik_mudelisse")

%\end{verbatim}
```

Lisa 2. Statistikatarkvara R Liiklusõnnetuste prognoosimise kood

```
    library(randomForestExplainer)
library(lubridate)
library(magrittr)
library(dplyr)
library(ggplot2)
```

```
library(data.table)
library(tidyverse)
library(stringr)
library(data.tree)
library(forcats)
library(zoo)
library(readxl)
library(geosphere)
library(readr)
library(gtools)
library(randomForest)
library(caret)
library(tree)
library(pROC)

setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
load("andmestik_mudelisse")


puu <- tree(onnetus ~., data = andmestik)
plot(puu)
text(puu, pretty = 0)

train <- sample(nrow(andmestik), floor(0.7*nrow(andmestik))
   , replace = FALSE)
TrainSet <- andmestik[train,]
ValidSet<- andmestik[-train,]

alavalim <- downSample(TrainSet, TrainSet$onnetus)
alavalim <- select(alavalim, -Class)
set.seed(1)
rf <- randomForest(onnetus~., data = alavalim, ntree = 400,
   mtry = 5,importance = T)
plot(rf)
pred.train <- predict(rf, newdata = alavalim)
```

```
table(pred.train, alavalim$onnetus)
pred.oob <- predict(rf)
table(pred.oob, alavalim$onnetus)
pred.valid <- predict(rf, newdata = ValidSet, type = "class
   ")
table(pred.valid, ValidSet$onnetus)

#SEE HILJEM VAJALIK
# pl <- plot_min_depth_distribution(rf)
# pl + ylab("Puude arv") + xlab("") +
#   labs(title = NULL) +
#   guides(fill=guide_legend(title="Minimaalne s gavus"))

#ROC
valid.pred_prob <- predict(rf, newdata = ValidSet, type = "
   prob")
oob.pred_prob <- predict(rf, type = "prob")
roc1 <- roc(ValidSet$onnetus, valid.pred_prob[,2])
roc1.oob <- roc(alavalim$onnetus,oob.pred_prob[,2])

plot(roc(ValidSet$onnetus, valid.pred_prob[,2]))
plot(roc(alavalim$onnetus,oob.pred_prob[,2]))

#5x
toimus <- TrainSet %>% filter(onnetus == 1)
eitoimunud <- TrainSet %>% filter(onnetus == 0)
train <- sample(nrow(eitoimunud), size =  floor(5*nrow(
   toimus)), replace = FALSE )
TrainSet1 <- rbind(toimus, eitoimunud[train,])
set.seed(1)
rf1 <- randomForest(onnetus~., data = TrainSet1, ntree = 40
   0,mtry = 5,importance = T)
plot(rf)
plot(rf1)
valid.pred_prob1 <- predict(rf1, newdata = ValidSet, type =
```

```
    "prob")
oob.pred_prob1 <- predict(rf1, type = "prob")
roc2 <- roc(ValidSet$onnetus, valid.pred_prob1[,2])

#25x
train <- sample(nrow(eitoimunud), size =  25*nrow(toimus),
   replace = FALSE )
TrainSet2 <- rbind(toimus, eitoimunud[train,])

set.seed(1)
rf3 <- randomForest(onnetus~., data = TrainSet2, ntree = 40
   0,mtry = 8,importance = T)

plot(rf3)
valid.pred_prob3 <- predict(rf3, newdata = ValidSet, type =
    "prob")
roc25 <- roc(ValidSet$onnetus, valid.pred_prob3[,2])
plot(roc25)

set.seed(1)
rf2 <- randomForest(onnetus~., data = TrainSet2, ntree = 40
   0,mtry = 8,importance = T)

plot(roc1, col = "red", xlab = "Spetsiifilisus", ylab = "
   Tundlikus")
plot(roc2, col = "blue", add = T)
plot(roc25, col = "green", add = T)

roc1
roc2
roc25

legend("bottomright", c("1:1", "1:5", "1:25"), col = c("red
   ", "blue", "green"), lty = 1)
valid.pred_prob2 <- predict(rf1, newdata = ValidSet, type =
```

```
    "prob")
roc(ValidSet$onnetus, valid.pred_prob2[,2])

dim(andmestik)
table(andmestik$onnetus)


pred.train <- predict(rf1, newdata = alavalim)
table(pred.train, alavalim$onnetus)
pred.oob <- predict(rf)
table(pred.oob, alavalim$onnetus)
pred.valid <- predict(rf, newdata = ValidSet, type = "class
   ")
table(pred.valid, ValidSet$onnetus)


pl <- plot_min_depth_distribution(rf)
pl + ylab("Puude arv") + xlab("") +
  labs(title = NULL) +
  guides(fill=guide_legend(title="Minimaalne s gavus"))




mean(pred.valid == ValidSet$onnetus)
abi <- data.frame(ValidSet$onnetus, valid.pred_prob[,2])
colnames(abi) = c("onnetus", "toenaosus")
ggplot(abi, aes(x = toenaosus, fill = onnetus)) + geom_
   density(alpha = 0.4)

a = matrix(NA, nrow = 8, ncol = 3)

#Vaatamaks milline on antud olukorras parimad juhumetsa
   algoritmi parameetrid
```

```
for(i in 1:8){
  for (j in 1:3){
    set.seed(1)
    mudel <- randomForest(onnetus~., data = alavalim, ntree
        = 400 * j, mtry = i+3, importance = T)
    predValid <- predict(mudel, newdata = ValidSet, type =
       "class")
    a[i,j] = mean(predValid == ValidSet$onnetus)

  }
}

plot(roc1,xlab = "Spetsiifilisus", ylab = "Tundlikus")

rf_suhe1 <- randomForest(onnetus~., data = alavalim, mtry =
    8,ntree = 1000, importance = T)
summary(alavalim)

pred.train <- predict(rf_suhe1, newdata = alavalim, type =
   "class")
pred.oob <- predict(rf_suhe1)
pred.valid <- predict(rf_suhe1, newdata = ValidSet, type =
   "class")


table(pred.valid, ValidSet$onnetus)
table(pred.oob, alavalim$onnetus)

mean(pred.valid == ValidSet$onnetus)

mean(pred.oob == TrainSet$onnetus)
varImpPlot(rf_suhe1)
length(valid.pred_prob)
2*length(ValidSet$onnetus)
```

```
length(ValidSet$onnetus)
length(pred.valid)
length(alavalim$onnetus)
table(pred.train, alavalim$onnetus)
a <- order(pred.valid)
pred.train[a]


#ROC
valid.pred_prob <- predict(rf_suhe1, newdata = ValidSet,
   type = "prob")
oob.pred_prob <- predict(rf_suhe1, type = "prob")
plot(roc(ValidSet$onnetus, valid.pred_prob[,2]))
plot(roc(alavalim$onnetus,oob.pred_prob[,2]))

View(oob.pred_prob)
length(oob.pred_prob[,2])
length()
roc(TrainSet$onnetus, pred.train[order(pred.train)],
   direction="<")


length(a)
length(TrainSet$tund)
table(pred.train, alavalim$onnetus)
table(pred.oob, alavalim$onnetus)
table(pred.train, alavalim$onnetus)

plot(alavalim2$onnetus)

Train
vek <- andmestik$onnetus == 1
length(vek[vek==T])
alavalim2 <- andmestik[vek,]
```

```
mitteToim <- andmestik[!vek,]
table(mitteToim$onnetus)


train <- sample(x = andmestik$onnetus, size = 2,
                replace = FALSE, prob = c(0.8, 0.2))
table(andmestik$onnetus)
size =  floor(5.03*nrow(alavalim2))
mitte2 <- sample(nrow(mitteToim),  replace = FALSE)

TrainSet <- andmestik[mitte2,]
TrainSet <- rbind(TrainSet, alavalim2)
ValidSet <- mitteToim[-mitte2,]
summary(andmestik)
table(TrainSet$onnetus)
5*4646
table(andmestik$onnetus)

rf_suhe1 <- randomForest(onnetus~., data = TrainSet, ntree
   = 800, mtry = 8, importance = T)

pred.train <- predict(rf_suhe1, data = alavalim)
pred.oob <- predict(rf_suhe1)
pred.valid <- predict(rf_suhe1, data = ValidSet)
length(pred.valid)

table(pred.train, TrainSet$onnetus)
mean(length(pred.valid) == length(ValidSet$onnetus))
table(pred.oob, alavalim$onnetus)
table(pred.train, alavalim$onnetus)

onnetusT <- andmestik$onnetus == 1
toimus <- andmestik[onnetusT,]
```

```
class(alavalim)
summary(rf_suhe1)
table(andmestik$ndl_paev)
#as.factor viskab errori prst
levels(andmestik$tund)

colnames(andmestik)
andmestik <- select(andmestik, -c(V1,kaameraId, paev, kuu))
summary(head(andmestik))
set.seed(1)
train <- sample(nrow(andmestik), floor(0.7*nrow(andmestik))
   , replace = FALSE)
TrainSet <- andmestik[train,]
ValidSet<- andmestik[-train,]
colnames(andmestik)
TrainSet <- as.matrix(TrainSet)


alavalim <- downSample(andmestik, andmestik$onnetus)
#Veel mingite sammudega, nt 1, 5, 10
#Sampliga x korda rohkem nulle
#K igepealt treeninguks ja siis tasakaalusta.
alavalim <- select(alavalim, -Class)
train <- sample(nrow(alavalim), floor(0.7*nrow(alavalim)),
   replace = FALSE)
TrainSet <- alavalim[train,]
ValidSet<- alavalim[-train,]
summary(alavalim)

eiToimu <- andmestik[!onnetusT,]
length(table(eiToimu$lon))

#Esmalt puu
TrainSet$onnetus <- as.factor(TrainSet$onnetus)
```

```
puu <- rpart(onnetus~., data = TrainSet)
install.packages("party")
library(party)
summary(puu)
Train1 <- select(TrainSet, )
puu1 <- tree(onnetus~., data = TrainSet)
par(mfrow = c(1,2), xpd = NA)
plot(puu)
text(puu, use.n = T)
puu.pr <- predict(puu, data = ValidSet)
summary(puu.pr)

View(puu.pr)

summary(TrainSet)
#Random Forest


model <- randomForest(onnetus~., data = TrainSet, ntree = 5
   00, mtry = 11, type = "class")
model
#Predictions
valid.pred <- predict(model, ValidSet, type = "class")
test.pred <- predict(model, TrainSet, type = "class")
oob.pred <- predict(model, type = "class")
table(test.pred, TrainSet$onnetus)
table(valid.pred, ValidSet$onnetus)
table(oob.pred, TrainSet$onnetus)

mean(valid.pred == ValidSet$onnetus)
table(test.pred)
table(oob.pred, TrainSet$onnetus)
mean(oob.pred == TrainSet$onnetus)
mean(valid.pred == ValidSet$onnetus)
mean(oob.pred == TrainSet$onnetus)
```

```
#Valid
table(valid.pred, ValidSet$onnetus)
mean(predValid == ValidSet$onnetus)



pred.rf <- predict(model, ValidSet, type = "class")
mean(pred.rf == ValidSet$onnetus)
rf.tabel <- table(pred.rf, ValidSet$onnetus)
prop.table(rf.tabel,1)
t2htsus <- importance(model)
t2htsuse_joonis <- varImpPlot(model)


pl <- plot_min_depth_distribution(rf_suhe1)
pl + ylab("Puude␣arv") + xlab("") + labs(title = NULL) +
   guides(fill=guide_legend(title="Minimaalne␣s gavus"))


t2htsus.rf <- measure_importance(model)

plot_multi_way_importance(t2htsus.rf, size_measure = "no_of
   _nodes") +
  xlab("keskmine␣minimaalne␣puu␣s gavus") +
  ylab("Puude␣arv") + theme_classic()


for (i in 1:length(colnames(andmestik))) {
  print("\item␣"+ colnames[i])
}
length(colnames(andmestik))

(1-1/20)**20
#oob
oob.pr <- predict(mudel, type = "class")
```

```
mean(oob.pr == TrainSet$onnetus)
table(oob.pr, TrainSet$onnetus)

#Trainset
train.pr <- predict(mudel, TrainSet, type = "class")
mean(train.pr == TrainSet$onnetus)
table(train.pr, TrainSet$onnetus)



model1 <- randomForest(onnetus ~ ., data = TrainSet,
   importance = TRUE)
model1

plot(model)
model2 <- randomForest(onnetus ~ ., data = TrainSet, ntree
   = 100, mtry = 6, importance = TRUE)
model2
plot(model2)

# Predicting on train set
predTrain <- predict(model2, TrainSet, type = "class")
# Checking classification accuracy
table(predTrain, TrainSet$onnetus)

#
oob.pr <- predict(model2, type = "class")
table(oob.pr, TrainSet$onnetus)
# Predicting on Validation set
predValid <- predict(model2, ValidSet, type = "class")
table(predValid, ValidSet$onnetus)
?randomForest
# Checking classification accuracy
mean(predValid == ValidSet$onnetus)
```

```
table(predValid,ValidSet$onnetus)
prop.table(table(predValid, ValidSet$onnetus),1)

importance(model2)
varImpPlot(model2)
model1

plot(table(test1$tund, test1$onnetus)[,2])
a=c()




table(test1$onnetus)
plot(rf.test)

distribution
distribution <- plot_min_depth_distribution(rf.test)

View(distribution)

plot_min_depth_distribution(rf.test, title(main = "
   Minimaalse␣s gavuse␣varieeruvus␣ja␣selle␣keskv  rtus"
   ),
                                              xlab = "Puude␣
                                                 arv", ylab =
                                                 "Tunnus",
                                                 mean_sample =
                                                  "relevant_
                                                 trees")


min_sygavus <- min_depth_distribution(rf)
min_sygavus
```

```
importance_frame <- measure_importance(rf.test)
save(importance_frame, file = "importance_frame.rda")
load("importance_frame.rda")

(importance_frame)

plot_multi_way_importance(importance_frame, size_measure =
   "no_of_nodes")

plot_importance_ggpairs(importance_frame)

plot_importance_rankings(importance_frame)

vars <- important_variables(importance_frame, k = 5,
                            measures = c("mean_min_depth",
                                "no_of_trees"))
vars
interactions_frame <- min_depth_interactions(rf.test, vars)
View(interactions_frame)
save(interactions_frame, file = "interactions_frame.rda")
load("interactions_frame.rda")
(interactions_frame[order(interactions_frame$occurrences,
   decreasing = TRUE), ])

table(a)
table(andmestik$onnetus)
plot_min_depth_interactions(interactions_frame)
puu <- tree(rf.test)
plot(puu)
text(puu)
```

```
summary(andmestik)



table(andmestik$kaameraId,andmestik$onnetus)

#oof, kui mudelit pole
plot(model1)
#out of bag -
#

# write.csv(andmestik, "data1.csv")
# andmestik <- fread("data1.csv")
# summary(andmestik)
# table(andmestik$tund, andmestik$onnetus)
# ggplot(andmestik, aes(x = tund, y = )) + geom_point()
#
# andmestik$paev <- as.Date(andmestik$paev)
# andmestik <- andmestik %>%
#   arrange(kaameraId, paev, tund) %>%
#   group_by(kaameraId) %>%
#   mutate(temp_erinevus = c(0,diff(ohutemp)),
#          sade_erinevus = c(0,diff(sajuintensiivus))) %>%
#   ungroup()
#
# andmestik$sade_erinevus[andmestik$sade_erinevus > 100] <-
    50
# andmestik$sade_erinevus[andmestik$sade_erinevus < -100]
   <- 50
# andmestik$karedus <- as.factor(andmestik$karedus)
# andmestik$ndl_paev <- as.factor(andmestik$ndl_paev)
# andmestik$tuulesuund <- cut(andmestik$tuulesuund, breaks
   = seq(0,360, 45))
# andmestik$tuulesuund <- as.character(andmestik$tuulesuund
   )
```

```
#
# andmestik$tuulesuund[is.na(andmestik$tuulesuund)] <- "
   puudub"
# andmestik$tuulesuund <- as.factor(andmestik$tuulesuund)
# andmestik$tuulesuund <- as.factor(andmestik$tuulesuund)
#
# andmestik$onnetus <- as.factor(andmestik$onnetus)
# andmestik$kuu <- as.factor(andmestik$kuu)
# andmestik$kaameraId <- as.factor(andmestik$kaameraId)
# andmestik$tund <- as.numeric(andmestik$tund)
# andmestik$tund <- as.factor(andmestik$tund)
# andmestik$ndl_paev <- factor(weekdays(as.Date(andmestik$
   paev), abbreviate = T),
#                              levels = c("E", "T", "K", "N
   ", "R", "L", "P"))
# andmestik <- select(andmestik, -c(V1,kaameraId, paev, kuu
   ))


%\end{verbatim}
```

Lisa 3. Jooniste tegemise kood statistikatarkvaras R

```
library(lubridate)
library(magrittr)
library(dplyr)
library(ggplot2)
library(data.table)
library(tidyverse)
library(stringr)
library(data.tree)
library(forcats)
library(zoo)
library(readxl)
library(geosphere)
```

```
library(readr)
library(gtools)
library(randomForest)
library(caret)
library(tree)
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

#ndl + tund
andmestik$ndl_paev = factor(andmestik$ndl_paev, levels = c(
   "E", "T", "K", "N", "R", "L", "P"))
ggplot(andmestik %>% filter(onnetus == 1), aes(x = tund)) +
  geom_bar() +
  facet_wrap(~ndl_paev, ncol = 1)

summary(andmestik1)
andmestik <- fread("data.csv")
andmestik$onnetus <- as.factor(andmestik$onnetus)
table(andmestik$onnetus)
################################
#Gini
gini <- function(x){x*(1-x) * 4}
#Entroopia
ent(0.5)
ent <- function(x){(-1*(x*log(x) + (1-x)*log(1-x)))/0.69314
   72}
#KlassifitseerimisViga
e <- function(x){2*x*(x<=0.5) + -2*(x-1)*(x>0.5)}

abi <- function(x){(1 - 1/x)**x}
#Viide:
2**3
#http://t-redactyl.io/blog/2016/03/creating-plots-in-r-
   using-ggplot2-part-9-function-plots.html
joonis <- ggplot(data.frame(x = c(0, 1)), aes(x = x)) +
  stat_function(fun = abi) + scale_x_continuous(name = "p␣
```

```
      v  rtus",
                                                breaks = seq(
                                                   10,10000))
                                                    + geom_
                                                   point()


joonis
tkeldamiseMeetodid <- ggplot(data.frame(x = c(0, 1)), aes
    (x = x)) +
  stat_function(fun = gini,
                aes(linetype = "Gini indeks")) +
  stat_function(fun = ent,
                aes(linetype = "Entroopia")) +
  stat_function(fun = e,
                aes(linetype = "Klassifitseerimisviga")) +


  scale_x_continuous(name = "Klassi osakaal lehes",
                     breaks = seq(0, 1, 0.1),
                     limits=c(0, 1)) +
  scale_y_continuous(name = "V  rtused", breaks = seq(0,1
     ,0.1)) +
  ggtitle("") +
  scale_linetype_manual("Meetodid", values = c("solid", "
     dotted", "longdash")) +
  theme_classic()
tkeldamiseMeetodid

funktsioon <- ggplot(data.frame(x = c(100, 10000)), aes(x =
    x)) +
  stat_function(fun = abi,
                aes()) + theme_classic() + ylab("
                   T en osus") + xlab("Valimi maht")
funktsioon
```

```
# mberskaleeritud . Entroopia k ige tundlikum

#MAP
#Viide
#https://rstudio.github.io/leaflet/markers.html
summary(andmestik)
a <- which(andmestik$onnetus %in% 1)
abi <- andmestik[a,]
library(leaflet)
m <- leaflet(data = a) %>%
  addTiles() %>%  # Add default OpenStreetMap map tiles
  addMarkers(lng=andmestik$lon, lat=andmestik$lat, popup="
     The␣birthplace␣of␣R")
m  # Print the map
onnetused <- andmestik$onnetus == 1
onnetusedKaameraJuures <- andmestik[onnetused, ]



onnetusedKaameraJuures_agr <- onnetusedKaameraJuures %>%
  count(kaameraId) %>%

onnetusedKaameraJuures_agr$lon <- onnetusedKaameraJuures$
   lon


onnetusedKaameraJuures_agr <- left_join(
   onnetusedKaameraJuures_agr,
                                        Ilmajaam,
                                        by = c("kaameraId"
                                           = "id"))

onnetusedKaameraJuures_agr$
leaflet(onnetusedKaameraJuures_agr) %>%
  fitBounds(lng1 = ~min(lon), l
```

```
  at1 = ~min(lat), lng2 = ~max(lon), lat2 = ~max(lat)) %>%
  addProviderTiles(providers$CartoDB.PositronNoLabels) %>%
  addCircleMarkers(~lon, ~lat, radius = ~ceiling(n/200))


onnetus <- leaflet() %>% addTiles() %>% addMarkers(
  clusterOptions = markerClusterOptions(),
  lng = abi$lon,
  lat = abi$lat

)

%\end{verbatim}
```

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Hardi Roosi,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Juhumetsa mudel ja selle rakendamine liiklusõnnetuste prognoosimisel", mille juhendaja Taavi Unt, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktis 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Hardi Roosi
08.05.2019