Tartu Ülikool
Loodus- ja täppisteaduste valdkond
Matemaatika ja statistika instituut
Kindlustus- ja finantsmatemaatika õppekava

Hardi Roosi

K-lähinaabri meetodi ja selle modifikatsioonide rakendamise tehnilistest detailidest ja nende võimalikust mõjust tulemuste täpsusele

Magistritöö (30 EAP)

Juhendaja: PhD Raul Kangro

Tartu 2023

K-LÄHINAABRI MEETODI JA SELLE MODIFIKATSIOONIDE RAKENDAMISE TEHNILISTEST DETAILIDEST JA NENDE VÕIMALIKUST MÕJUST TULEMUSTE TÄPSUSELE

Magistritöö

Hardi Roosi

Lühikokkuvõte

Käesoleva magistritöö eesmärk on uurida vaatluste vahelise kauguse arvutamise vorme ning rakendada neid k-lähinaabri algoritmi prognoosil. Töös tutvustatakse esmalt vaatluste vahelise kauguse arvutamise meetodeid ja kuidas käituda numbrilise või nominaalse tunnuse korral. Järgnevalt vaadeltakse uuritava tunnuse prognoosimist lähtudes KNN algoritmi modifikatsioonidest. Lõpuks antakse praktiline näide KNN algoritmi ja KNN algoritmi modifikatsioonide põhjal.

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

Märksõnad: K-lähinaabri algoritm, masinõpe, tõenäosus.

ON THE TECHNICAL DETAILS OF THE IMPLEMENTATION OF THE K-NEAREST NEIGHBOR METHOD AND ITS MODIFICATIONS AND THEIR POSSIBLE EFFECT ON THE ACCURACY OF THE RESULTS

Master thesis

Hardi Roosi

Abstract

The purpose of this master's thesis is to study the forms of calculating the

distances between observations and apply them to the prediction of the k-nearest neighbor algorithm. The paper first introduces the methods of calculating the distance between observations and how to behave in the case of a numerical or nominal characteristic. In the following, the prediction of the investigated characteristic based on the modifications of the KNN algorithm is considered. Finally, a practical example based on KNN algorithm and KNN algorithm modifications is given.

CERCS research specialisation: P160 Statistics, programming, financial and actuarial mathematics.

Key Words: k-nearest neighbours algorithm, machine learning, probability.

Sisukord

Sissejuhatus

Antud magistritöö eesmärk on lähemalt uurida k-lähinaabri algoritmi nüansse. KNN on lihtne juhendatud masinõppe liik. Algoritm põhineb ideel, et üksteisele lähedaste tunnuste väärtustega vaatlused on tavaliselt ka prognoositava tunnuse käitumise mõttes lähedased. Treeningandmetel on igal vaatlusel komplekt seletavate tunnuste väärtuseid ning prognoositava tunnuse väärtus. Uue vaatluse puhul leitakse selle naabrid seletavate tunnuste abil, leides treeningandmetest k lähimat vaatlust. Seejärel toimub uue vaatluse prognoos k lähima naabri väärtuste põhjal.

Töö on jagatud kolme suuremasse ossa. Esmalt uurime vaatluste vaheliste kauguste määramist ning kuidas kuidas arvutada nominaalsete tunnuste vahelist kaugust. Antud teadmisi rakendame k-lähinaabri algorimi realiseerimisel. Töö teises osas uurime k-lähinaabri algoritmi modifikatsioone, milleks on lokaalne lineaarne regressioon ja tuumaga silumise meetod. Viimasena uurime algoritimi töövõimet, kasutades bondora veebisaidilt saadud andmeid(Bondora (2023)). Hindame tõenäosust, et kliendil tekib laenu tagasi maksmisega probleeme. Selleks kasutame järgnevaid seletavaid tunnuseid: sugu, vanus, haridustase, laenusuurus, laenu intressimäär, laenupikkus ning laenaja residentsus.

Magistritöö kirjutamiseks kasutasime Latex veebirakenduse liidest Overleaf. Töös esinevad joonised tegime kasutades tarkvara R(versioon 4.2.3).

1 k-lähinaabri algoritm

Antud peatükis anname ülevaate k-lähinaabri (*k-nearest neighbor*) algoritmi kohta. k-lähinaabri algoritm on mitteparameetriline masinõppe meetod prognoosimaks nominaalseid ning arvulisi tunnuseid. Treeningandmetel on igal vaatlusel komplekt seletavate tunnuste väärtuseid ning prognoositava tunnuse väärtus. Uue vaatluse puhul leitakse selle naabrid seletavate tunnuste abil, leides treeningandmetest k vaatlust, mille seletavad tunnused on võimalikult lähedased uue vaatluse seletavatele tunnustele. Seejärel toimub uue vaatluse prognoos k lähima naabri väärtuste põhjal. (Fix ja Hodges (1952))

Vastavalt prognoositava tunnuse tüübile on ka talitus erinev. Kui prognoositav tunnus on nominaalne, leitakse treeningandmete põhjal k lähimad naabrid ning valitakse nende prognoositava väärtuste seast mood. Arvuliste tunnuste korral leitakse k lähima naabri prognoositava tunnuse väärtuse keskmine.

1.1 Vaatluste vaheline kauguste arvutamine

Andmestikus vektorite $x = (x_1, x_2, \ldots, x_m), x_i \in \mathbb{R}$ ja $y = (y_1, y_2, \ldots, y_m)$, $y_i \in \mathbb{R}$ vahelise kauguse arvutamiseks on mitmeid erinavaid meetodeid. Tuntumateks on eukleidiline kaugus ning Manhattani kaugus. Need leitakse vastavalt valemitega

$$d(x, y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{1.1}$$

ning

$$d(x, y) = \sum_{i=1}^{m} |x_i - y_i|. \tag{1.2}$$

Üldistavalt võime valemid 1.1 ning 1.2 kokku võtta Minkowski kauguse valemiga

$$d(x, y) = \Big(\sum_{i=1}^{m} (x_i - y_i)^p \Big)^{\frac{1}{p}}, \tag{1.3}$$

kus $p \in \mathbb{R}$. Kui mõni seletavatest tunnustest on nominaalne, siis ei saa me valemi 1.3 põhjal vaatluste vahelist kaugust arvutada. Tavaline lahendus sellele probleemile on iga nominaalse tunnuse kodeerimine ühe või mitme arvulise tunnuse abil, mida vaatleme järgnevas alampeatükis.

1.2 Nominaalsete tunnuste kodeerimine kauguste arvutamisel

Nominaalsete tunnuste korral ei ole enamasti mõistlik kohelda sama tunnuse erinevaid tasemeid kauguse arvutamisel erinevalt. Nominaalse tunnuse kodeerime kasutades *one-hot* kodeerimisy (James *et al.* (2013)). *One-hot* kodeerimisel luuakse nominaalse tunnuse igale võimalikule väärtusele vastav uus tunnus, mis on üks juhul kui asendatava tunnuse väärtuseks on vaadeldav väärtus ja null vastasel korral. Juhul kui meil on tegemist järjestustunnusega, siis võib olla mõistlik, et me kohtleme tunnuse väärtuseid erinevalt. Näiteks, kui meil on tunnus „haridustase". Sel juhul võib olla loogiline, et tunnuse väärtus „põhiharidus" on kaugemal tunnuse väärtusest „kõrgharidus", kui seda on „keskhariduse" ja „kõrghariduse" tunnuse väärtuste vaheline kaugus. Illustreerime *one-hot* kodeerimise põhimõtet kasutades andmestikku *iris* (Fisher (1936)) ning valides sealt juhuslikult 10 vaatlust. Saame järgneva andmestiku:

Tabel 1: *One-hot* kodeerimise näide

Id	Liik		Id	Liik:setosa	Liik:versicolor	Liik:virginica
3	setosa		3	1	0	0
17	setosa		17	1	0	0
147	virginica		147	0	0	1
87	versicolor		87	0	1	0
96	versicolor	$\longrightarrow$	96	0	1	0
94	versicolor		94	0	1	0
23	setosa		23	1	0	0
119	virginica		119	0	0	1
32	setosa		32	1	0	0
84	versicolor		84	0	1	0

Antud viisil saame 0/1 väärtustega tunnuste komplekti, kus igal real on üks ja ainult üks nullist erinev number ning iga rea summeerimisel saame väärtuseks 1. Järelikult tekib andmemaatriksi veergude vahel kollineaarsus. Seega pole antud tunnused üksteisest lineaarselt sõltumatud. Paljude standardsete statistikameetodite puhul (näiteks lineaarsete ja üldiste lineaarsete mudelite) on kollineaarsus probleemiks, kuna mudeli kordajad ei ole sel juhul üheselt määratud. Seetõttu jäetakse kodeerimisel sageli üks tase ära (nn baastase) ja nii saame *leave-one-out* kodeeringu. Esitame tabelis 1 toodud andmed sellisel kujul, kus baastasemeks on liik „versicolor".

Tabel 2: *Leave-one-out* kodeering

Id	Liik:setosa	Liik:virginica
3	1	0
17	1	0
147	0	1
87	0	0
96	0	0
94	0	0
23	1	0
119	0	1
32	1	0
84	0	0

Sellisel kujul saame iga liigi korral info kätte. Kui mingi rea puhul on liik „setosa"

= 0 ning liik „virginica” = 0, siis peab vaadeldav vaatlus olema liik „versicolor”.

Arvutame andmete 1 põhjal kaugused id = 3, id = 87 ning id = 119 vahel. Kauguste arvutamisel kasutame Minkowski valemit ning parameetri $p = 1$ korral saame vaatluste vaheliseks kauguseks.

Id = 3 ja Id = 87	Id = 3 ja Id = 119	Id = 87 ja Id = 119
$d = 2$	$d = 2$	$d = 2$

Näeme, et kui kodeerida tabel 1 toodud meetodil, saame tulemuseks, et liikide vahelised erinevused (kaugused) on samad. Leiame tunnuste vahelised kaugused vaadeldes tabelit 2.

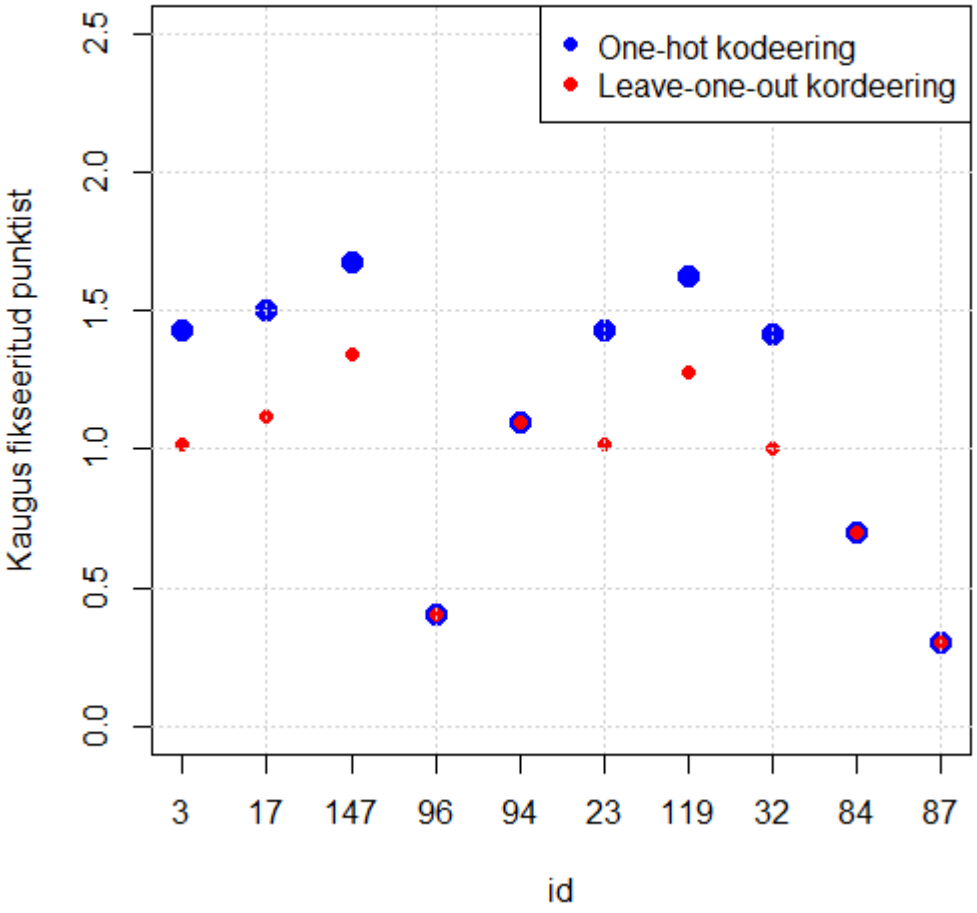
Id = 3 ja Id = 87	Id = 3 ja Id = 119	Id = 87 ja Id = 119
$d = 1$	$d = 2$	$d = 1$

Näeme, et id = 87 (liik „versicolor”) on lähemal teistele liikidele vastavatele vaatlustele, kui teistele liikidele vastavad vaatlused omavahel. Probleemi illustreerimiseks lisame tabelis 1 olevatele liikidele juurde taime tupplehe laiuse (*sepal width*) sentimeetrites.

Tabel 3: Iirise liigid ning nende vaatluste tupplehe laiused

Id	Liik	Tupplehe laius
3	setosa	3,2
17	setosa	3,9
147	virginica	2,5
87	versicolor	3,1
96	versicolor	3,0
94	versicolor	2,3
23	setosa	3,6
119	virginica	2,6
32	setosa	3,4
84	versicolor	2,7

Fikseerime uue vaatluse, mille liik on "versicolor"ning tupplehe laius on 3,4 sentimeetrit. Lisame joonise, kus on näidatud fikseeritud vaatluse kaugused ülejäänud vaatlustest. Nominaalset tunnust "liik"dekodeerime kasutades tabelis 2 näidatud viisil. Kauguste arvutamiseks on kasutatud valemit (1.1).



Joonis 1: Id = 87 kaugus teistest tabelis 3 toodud väärtustest.

Vaatleme fikseeritud punkti kaugust vaatlustest, kus $id = 23$ ja $id = 94$. Kasutades *one-hot* kodeerimist on $id = 94$ lähemal fikseeritud vaatlusele. Kasutades *leave-one-out* meetodit on $id = 23$ lähemal fikseeritud punktile. Näeme, et erinevate

kodeerimismeetodite korral saame vastuseks erinevad lähimad vaatlused.

Lähinaabrite korral pole oluline, et tunnused on omavahel lineaarselt sõltuvad (James *et al.*, 2013). Kollineaarsus on probleem näiteks regresioonimudelite korral, kus ei saa kollineaarse andmemaatriksi korral leida täpseid parameetreid prognoosi leidmiseks, kuna üks andmeveerg on teisega sõltuvuses (James *et al.*, 2013).

Tabeli 1 korral tuleb maksimaalne tasemete vaheline kaugus 2, aga kroonlehe laiuse puhul näeme oluliselt väiksemaid kauguste vahesid. Sellega seoses muutub taime liik oluliselt suuremaks faktoriks kauguse arvutamisel, kui seda on kroonlehe laius. Selle probleemiga tegeleme järgnevas alapeatükis.

1.3 Andmete skaleerimine

1.3.1 Arvuliste tunnuste skaleerimine

Eelnevalt toodud näite 2 juures nägime, et taime tupplehe laius on väiksema mõjuga distantsi arvutamise juures, kui seda oli taime liik. Selle probleemi saame lahendada kui skaleerime numbrilise tunnuse. Antud juhul taime tupplehe laiuse. Skaleerimise eesmärk on viia vektori väärtused kindlale skaalale. Kui me viime kõik numbrilised väärtused samale skaalale, pole kauguste arvutamisel ükski tunnus eelistatuim. Lisaks vabaneme sääraselt käitudes tunnuse mõõtühikust.

Idee illustreerimiseks vaatleme tabeli 3 andmete põhjal, kuidas tunnuse mõõtühik mõjutab vaatluse vahelist kauguse arvutamist. Teisendame tunnuse „tupplehe laius" sentimeetritelt mikromeetritele. Sarnaselt talitleme ka fikseeritud punktiga. Saame testandmetele järgneva tulemuse.

Tabel 4: Iirise liigid ning nende vaatluste tupplehe laiused mikromeetrites

Id	Liik	Tupplehe laius
3	setosa	32 000
17	setosa	39 000
147	virginica	25 000
87	versicolor	31 000
96	versicolor	30 000
94	versicolor	23 000
23	setosa	36 000
119	virginica	26 000
32	setosa	34 000
84	versicolor	27 000

Kodeerime andmed viisil 1 näidatud meetodil ning arvutame kaugused valemi (1.1) põhjal. Leiame kaugused eelmises peatükis fikseeritud vaatlusega. Saame viieks lähimaks naabriks vaatlused identifikaatoritega 3, 96, 23, 32 ja 87.

Antud tulemus erineb alapeatükis 1.1 leitud tulemusest. Tupplehe laius mikromeetrites mõjutab suuremal määral vaatluse kaugust fikseeritud punktist, kui seda mõ-

jutas tuppehe laius sentimeetrites. Probleemi lahendamiseks skaleerime tupplehe laiuste väärtused.

Skaleerimiseks on mitmeid erinevaid meetodeid. Üks nendest on normaliseerimine. Normaliseerimise käigus leiame iga vääruse kauguse miinimaalsest väärtusest ja jagame selle tunnuse maksimum ja miinimum väärtuse vahega. Valemi kujul on see järgnev

$$X = \frac{X - X_{min}}{X_{max} - X_{min}}. \tag{1.4}$$

Niiviisi saame vektori X, mille väärtused $X_i \in [0; 1]$. Tunnust saab ka viia vahemikku $[a; b]$. Seda saab teha järgnevalt

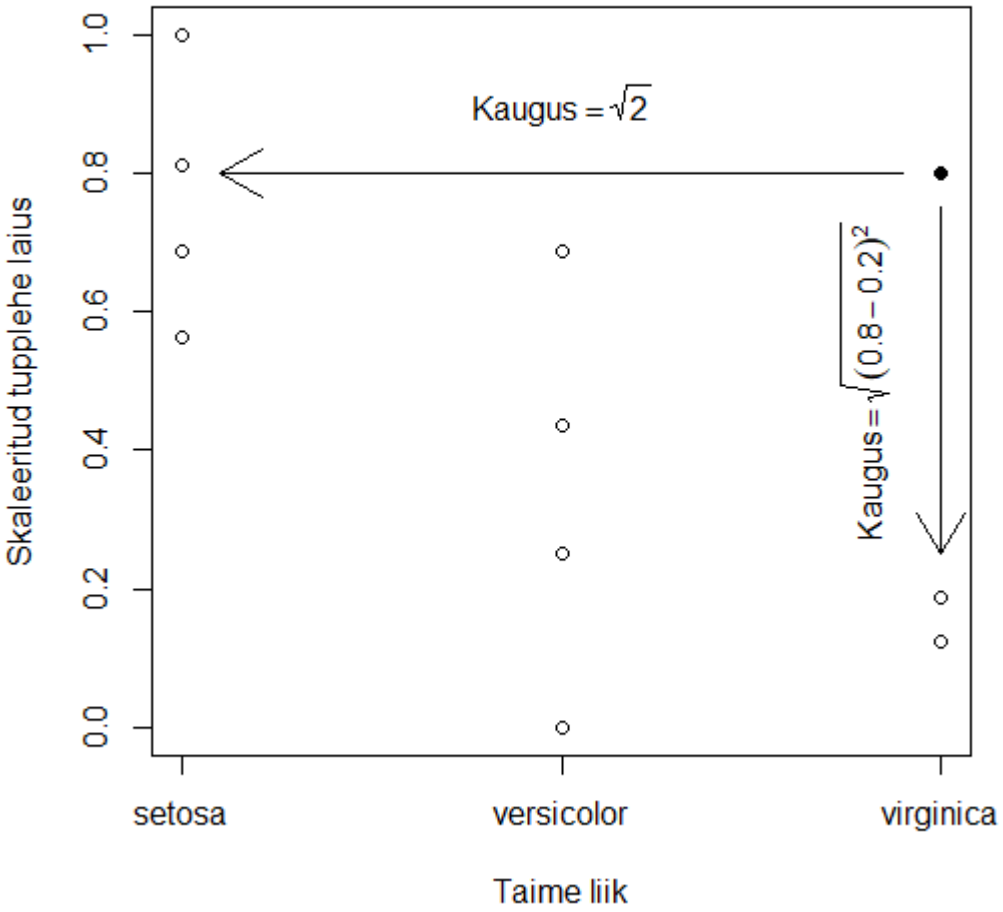
$$X_{[a;b]} = a + \frac{(X - X_{min})(b - a)}{X_{max} - X_{min}}. \tag{1.5}$$

Teine tuntud meetod on väärtuste standardiseerimine ehk z skoori leidmine.

$$z = \frac{X - \mu}{\sigma}, \tag{1.6}$$

kus μ on tunnuse X keskmine ning σ on tunnuse X standardhälve. (Spiegel ja Stephens (2008))

Normaliseerime näites 2 toodud andmed. Fikseerime uue vaatluse, milles tupplehe normaliseeritud laius on 0,8 ning taime liigiks on „virginica". Leiame uue vaatluse ja joonisel märgitud vaatluste vahelise kauguse.

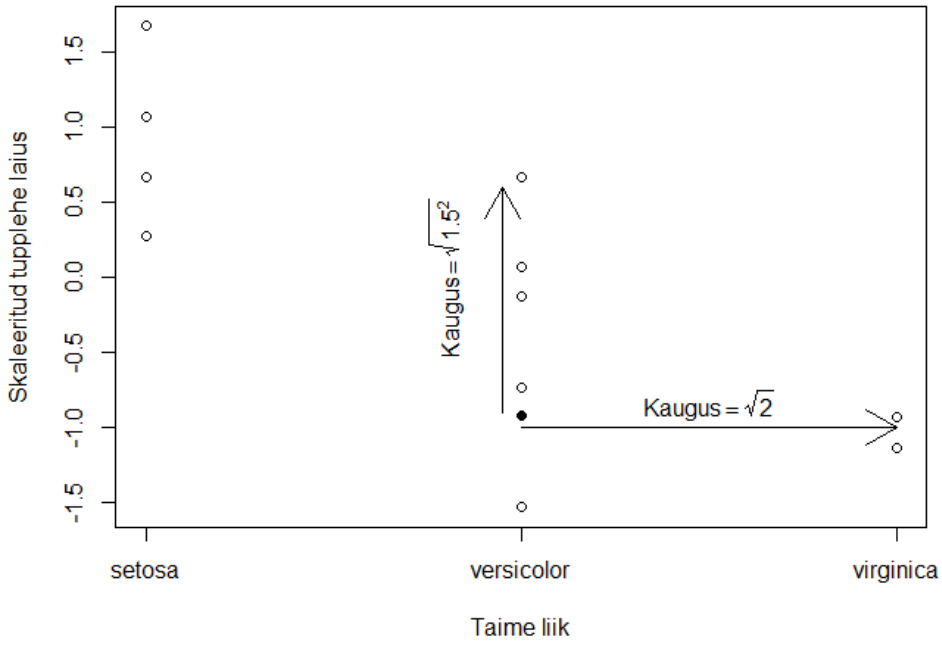


Joonis 2: Nominaalse tunnuse ning normaliseeritud numbrilise tunnuse mõju kauguse arvutamisel

Näeme, et taime liigil on kauguse arvutamisel suurem mõju kui tupplehe laiusel. Näitasime alapeatükis 1.2, et nominaalse tunnuse erinevate klasside vaheline kaugus on $\sqrt{2}$. Viies tupplehe laiuse valemis (1.4) näidatud kujule on vaatluse erinevus nominaalse tunnuse väärtusest alati suurema mõjuga kauguse arvutamisel, kui seda on erinevus tupelehe laiuse normaliseeritud väärtustest, mis on alati lõigus $[0; 1]$.

Vaatleme kas sama olukord tekib, kui taime tupplehe laiusi hoopis standardisee-

rida. Fikseerime uue vaatluse, mille taime liik on „versicolor" ning taime tupplehe standardiseeritud laius on -1. Leiame fikseeritud vaatluse kauguse vaatluste vahel, kus taime liigiks on „versicolor", mille skaleeritud tupplehe laius on 0.5, ning fikseeritud vaatluse ja vaatluse kus taime liigiks on „virginica", mille skaleeritud tupplehe laius on -1.



Joonis 3: Nominaalse tunnuse ning standardiseeritud numbrilise tunnuse mõju kauguse arvutamisel

Näeme, et taime liik ei mõjuta vaatluse kaugust nii suurel määral, kui eelmises näites. Seda seetõttu, et standardiseeritud tupplehe laius on arv, mis jääb tavaliselt lõiku $x \in [-3; 3]$. See võimaldab vaatluste vahel suuremat kaugust kui $\sqrt{2}$ ehk kauguste arvutamisel pole lähimad punktid nominaalse tunnusega fikseeritud.

Tekib loomulik küsimus, kas kasutada skaleerimisel meetodit (1.4) või (1.6). Mõlemad meetodid on korrektsed ja meetodi valik sõltub, kui suurt tähtsust omab nominaalne tunnus prognoositava tunnuse määramisel.

1.3.2 Nominaalse tunnuse skaleerimine

Tekib küsimus, kas tuleks skaleerida ka nominaalse tunnuse kodeerimisel tekkivad binaarsed tunnused. Näeme, et meil pole mõtet normaliseerida nominaalset tunnust, kuna väärtuse $x_i = 0$ korral on vektori liikme normaliseeritud väärtus

$$x_i = \frac{0-0}{1-0} = 0$$

ning väärtuse ($x_i = 1$) korral on ta normaliseeritud väärtus

$$x_i = \frac{1-0}{1-0} = 1$$

ehk nominaalse tunnuse korral on tunnus võrdne selle normaliseeritud kujuga.

Vaatleme nominaalse tunnuse korral, mis sellele vastavate binaarsete tunnustega juhtub, kui neid standardiseerida. Kui me valime juhuslikult nominaalse tunnuse väärtuse andmestikus olevate vaatluste hulgast, saame suuruse Bernoulli jaotusega parameetriga p, mis on ühtede osakaal väärtuste hulgas. Seega saame keskmise ja standardhälbe arvutamisel kasutada tõenäosusteooriast tuntud valemeid

$$\hat{\mu} = EX = p \tag{1.7}$$

ning σ saab hinnangulise väärtuse

$$E[X^2] = p \cdot 1^2 + (1-p) \cdot 0^2 = p. \tag{1.8}$$

Seega

$$\hat{\sigma} = \sqrt{DX} = \sqrt{E[X^2] - E[X]^2} = \sqrt{p - p^2}. \tag{1.9}$$

Seega leitakse nominaalse tunnuse korral standardiseeritud väärtus järgnevalt:

$$z = \frac{x - p}{\sqrt{p - p^2}}. \tag{1.10}$$

Fikseerime nominaalse tunnuse esinemise ning standardiseerime tunnuse kolmel erineval juhul. Olgu tunnuse esinemistõenäosus on $p = 0,1$, siis tunnuse standardiseeritud väärtus on

$$\frac{1 - 0.1}{\sqrt{0.1 - 0.1^2}} = \frac{0.9}{0.3} = 3. \tag{1.11}$$

Olgu tunnuse esinemistõenäosus on $p = 0,5$, siis tunnuse standardiseeritud väärtus on

$$\frac{1 - 0.5}{\sqrt{0.5 - 0.5^2}} = \frac{0.5}{0.5} = 1. \tag{1.12}$$

Olgu tunnuse esinemistõenäosus on $p = 0,9$, siis tunnuse standardiseeritud väärtus on

$$\frac{1 - 0.9}{\sqrt{0.9 - 0.9^2}} = \frac{0.1}{0.3} = \frac{1}{3}. \tag{1.13}$$

Näeme et vähem esinevad nominaalsed tunnused saavad standardiseerides suurema väärtuse. Suurem standardiseeritud väärtus viib kauguste arvutamisel suurema kauguseni. Väiksem standardiseeritud väärtus muudab tunnuse esinemise kauguste arvutamisel vähem tähtsamaks.

Mõtte illustreerimiseks standardiseerime näites 3 nominaalse tunnuse "taime liik". Selleks leiame iga liigi korral tema keskväärtuse ning standardhälbe. Liigi "setosa"ning liigi "versicolor"korral on keskväärtus 0,4 ning standardhälve 0,516. Standardiseerime liigi "setosa"tunnused. "Setosa"esinemisel on standardiseeritud väärtus

$$z = \frac{1 - 0,4}{0,516} = 1,16$$

liigi mitte esinemise korral on väärtuseks

$$z = \frac{0 - 0,4}{0,516} = -0,77.$$

Liigi "versicolor"korral on tulemused samad. Nüüd standardiseerime väärtused liigi "virginica"korral. Liigi "virginica"korral on keskväärtus 0,2 ning standardhälve 0,422. Liigi "virginica"esinemise korral on standardiseeritud väärtus

$$z = \frac{1 - 0,2}{0,422} = 1,90$$

ning liigi "virginica"mitte esinemise korral on väärtuseks

$$z = \frac{0 - 0,2}{0,422} = -0,47.$$

Standardiseerisime tabelis 3 toodud nominaalse tunnuse "taime liik". Näeme, et tulemused on erinevad. Selliselt talitledes muutub taime liik "virginica"suuremaks faktoriks kauguste arvutamisel, kui seda on ülejäänud liigid.

Juhul kui me ei soovi nominaalse tunnuse kindlat väärtust eelistada, siis pole meil ka mõistlik antud tunnust standardiseerida.

1.4 Kaalutud kauguste leidmine

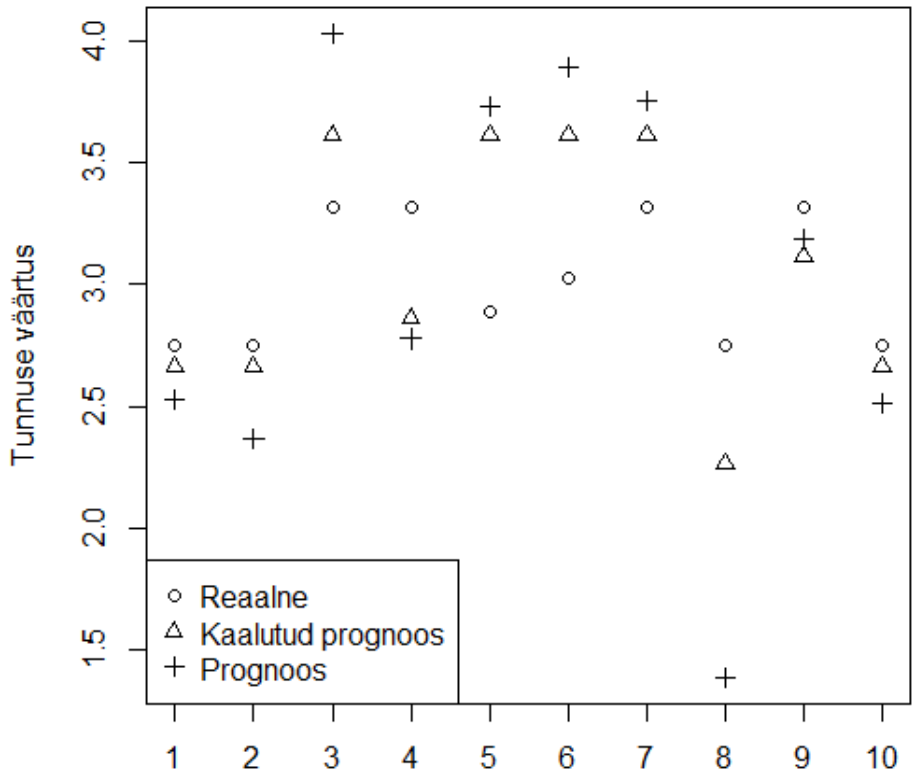
Vaatluste vaheliste kauguste leidmiseks on eelnevalt mainitud mitmeid erinevad meetodeid. Antud viisid ei anna aga aimu, kuidas mingi seletav tunnus võib mõjutada prognoositavat tulemust. Selleks me kasutame kaalutud Minkowski kauguse valemit, mis arvestab määratud tunnuse seost prognoositava tunnusega. Valemi esitame kujul

$$d(X_i, Y) = (\sum_{i=1}^{m} \omega_i^p |x_i - y_i|^p)^{\frac{1}{p}}, \tag{1.14}$$

kus $\omega = (\omega_1, \omega_2, \ldots, \omega_m)$ on kaalude vektor, kus antud olukorras $\omega_i = \frac{|\rho(X_i,Y)|}{\sigma_{X_i}}$ ehk Pearsoni kordaja prognoositava tunnuse ja i-nda andmeveeru vektori vahel ning see on jagatud i-nda andmeveeru standardhälbega. Paneme tähele, et selliste kaalude kasutamine on samaväärne korrelatsioonidega kaalutud standardiseeritud tunnuste väärtuste vahelise kauguse kasutamisega (Hechenbichler ja Schliep (2004)).

Näitamaks, kuidas valem (1.14) aitab prognoosimisel, genereerime 100 vaatlust treeningandmestikku, kus esimene veerg on $x_1 \sim N(0; 1)$ ja teine veerg on $x_2 \sim Exp(0, 5)$. Prognoositav veerg on $y = 0, 9 \cdot x_1 + 3$. Fikseerime 10 uut vaatlust, kus x_1 ning x_2 väärtused on teada. Prognoosime tunnuse y väärtusi. Fikseerime $k = 5$. Antud näites leiame vaatluste vahelised kaugused kahel viisil, kasutades valemeid 1.3 ning 1.14. Kauguste arvutamiseks fikseerime $p = 2$. Kuna y väärtused on x_1 kaudu leitavad, leiame ka reaalsed y väärtused.

Saame järgneva tulemuse:



Joonis 4: Tunnuse väärtuste prognoosimine 2 erineval viisil.

Näeme, et prognoos, kus on naabrite leidmisel kasutatud prognoositava tunnuse ja i-nda andmeveeru vektori vahelist korrelatsiooni, prognoosib tunnuse y väärtusi paremini. Selline tulemus on oodatav, sest tunnused x_1 ja y on omavahel tugevalt korreleeritud

$$cor(x_1, y) = cor(x_1; 0,9 \cdot x_1 + 3) = cor(x_1; 0,9 \cdot x_1) = 1.$$

Samas korrelatsioon tunnuste x_2 ja y vahel on teoreetiliselt

$$cor(x_2, y) = \frac{cov(x_2, y)}{\sigma_{x_2}\sigma_y} = \frac{E(x_2 y) - E(x_2)E(y)}{\sigma_{x_2}\sigma_y} = \frac{E(x_2)E(y) - E(x_2)E(y)}{\sigma_{x_2}\sigma_y} = 0.$$

Andmete põhjal arvutades saame kasutada andmevektorite vahelist empiirilist korrelatsiooni, mis ei ole täpselt 0. Seega saame treeningandmete põhjal leida ka tunnuste x_2 ning y korrelatsioon.

Joonis 4 korral on korrelatsioonide absoluutväärtused: $|cor(x_1, y)| = 1$ ning $|cor(x_2, y)| = 0,22$.

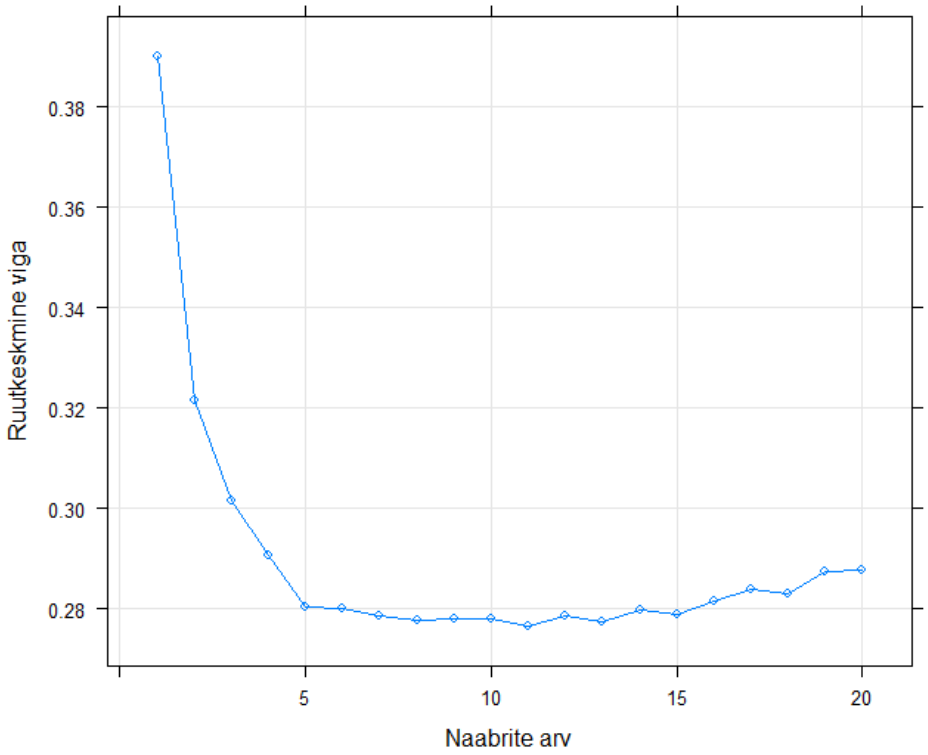
1.5 Parameetri k väärtuse arvutamine

Antud alampeatüki tulemused pärinevad raamatust Friedman, Hastie ja Tibshirani (2016) kui ei ole viidatud teisiti.

Parameeter k määrab lähimate naabrite arvu, mida algoritm arvestab prognoosimisel. Naabrite arvu valik võib oluliselt mõjutada algoritmi prognoosimisvõimet. Valides liiga väikse parameetri väärtuse on algoritm tundlik müra suhtes. Kui parameetri k väärtus on liiga suur, siis kasutatakse hinnangu leidmisel vaadeldavast tunnuste komplektist kaugel olevaid vaatluseid prognoosi arvutamisel ning see võib kaasa tuua suure nihke.

Parima parameetri k väärtuse leidmiseks jagame andmed treening ja testandmeteks. Kasutades treeningandmestikku, modelleerime algoritmi iga parameetri $k \in \mathbb{N}, a \leq k \leq b$ väärtuse korral. Mudeli täpsust hindame testandmestiku põhjal. Mudeli täpsuse hindamiseks kasutame regressiooniülesande korral prognooside ruutkeskmist viga. Valime parameetri k väärtuse, mille korral on mudeli ruutkeskmine viga kõige väiksem.

Ristvalideerimise illustreerimiseks leiame parameetri k väärtuse *iris* andmestiku põhjal. Prognoosime taime tupplehe laiust kasutades taimeliiki, tupplehe pikkust ning taime kroonlehe pikkust ning laiust. Mudeli treenime kasutades 70% vaatlustest ning mudeli täpsuse hindamiseks kasutame 30% vaatlustest.



Joonis 5: RMSE arvutamine erinevate parameetri k väärtuste korral.

Näeme, et väikseim ruutkeskmise vea väärtus tuleb parameetri väärtuse $k = 11$ korral. Uute sisendandmete prognoosimisel treenime mudeli kasutades kõiki teadaolevaid andmeid ning fikseerime parameetri väärtuse $k = 11$.

Sarnaselt saame ka käituda, kui prognoositav tunnus on nominaalne. Nominaalse tunnuse korral arvutame iga parameetri $k, k \in [a; b], a, b \in \mathbb{N}$ väärtuse korral saadud mudeli täpsuse. Mudeli täpsuse hindamiseks jagame mudeli poolt õigesti hinnatud vaatluste arvu kõikide prognoosi saanud vaatluste arvuga.(Kuhn ja Johnson (2013))

2 Prognooside arvutamise modifikatsioonid

2.1 Naabritele kaalude omistamine (*weighted kNN*)

Antud alampeatüki tulemused pärinevad raamatust Hechenbichler ja Schliep (2004) kui ei ole viidatud teisiti.

Oleme eelnevalt arutanud, kuidas prognoosimisel valime k lähimat naabrit ja arvestame nende puhul teadaolevaid prognoositava tunnuse väärtuseid võrdselt. Antud alapeatükis arutleme viise, kuidas me saame määrata kaale punktide vahelisele kaugusele nii, et lähemal olevad vaatlused määravad lõpp-tulemust rohkem, kui kaugemal olevad punktid.

Tähistame vaatluste vaheliste kauguste kaale funktsiooniga $T(d)$. Vaatluste vaheliste kauguste kaalude omistamiseks on mitmeid erinevaid meetodeid, kuid neil on järgmised omadused

- $T(d) \geq 0, \forall d \in \mathbb{R}$
- $T(0) = 1$
- $T(d)$ väheneb monotoonselt $d \to \infty$.

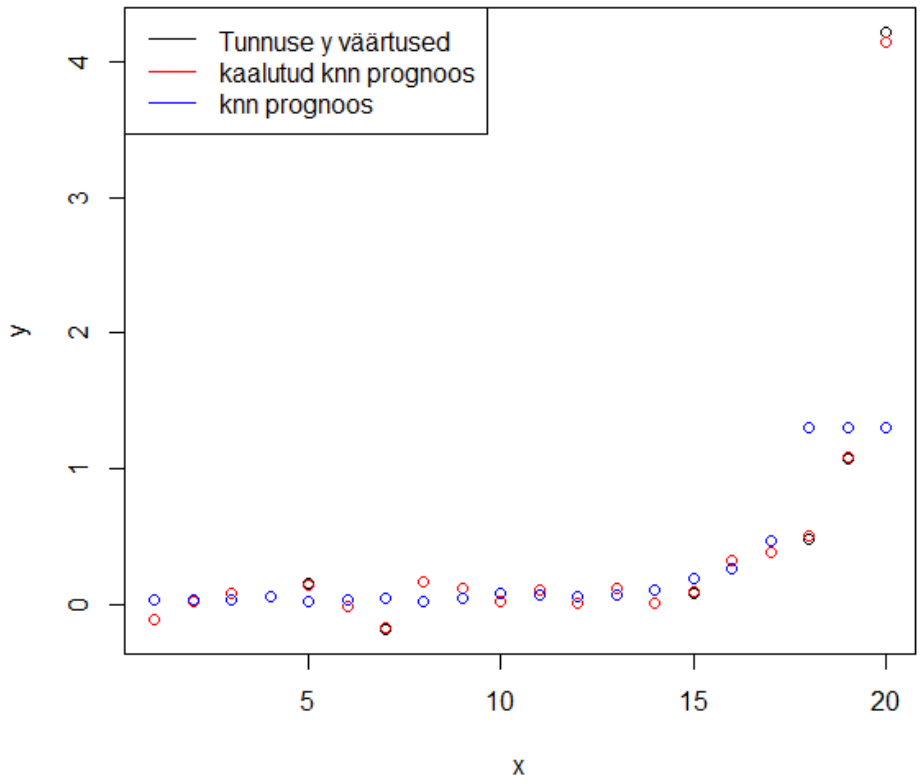
Kaalutud K-lähinaabri algoritmi korral, leiame hinnangu $\hat{y}$ järgnevalt.

$$\hat{y} = \frac{\sum_i^{N_k(x)} T(d_i) y_i}{\sum_i^N T(d_i)}, \tag{2.1}$$

kus $N_k(x)$ on k lähimat vaatlust vaadeldavast tunnusest x. Kui prognoositav tunnus y on nominaalne, siis käitume nominaalse tunnuse kodeerimisel tekkivate binaarsete tunnustega sarnaselt valemiga 2.1.

Olgu meil seletavad andmed $x_i = i, i = 1, \ldots, 20$ ning prognoositav tunnus $y_i = \frac{6}{1{,}43(x_i - max(x) + 0.01)^2}$. Prognoosime k-lähinaabri algoritmi ning kaalutud k-lähinaabri

algoritmi põhjal väärtusi tunnusele y, kasutades tunnust x. Kaalutud k-lähinaabri algoritmi korral on kasutatud kaalude arvutamiseks valemit $T(d) = \frac{1}{d}$. Kaugused on arvutatud kasutades valemit 1.3 parameetri $p = 1$ väärtusel. Parameetri K väärtuseks valime $K = 7$.



Joonis 6: Tunnuse y prognoos kasutades k-lähinaabri algoritmi ning kaalutud k-lähinaabri algoritmi

Näeme, et prognoositav tunnus käitub mõlema algoritmi korral sarnaselt. Erinevus tuleb antud juhul tunnuse y ekstreemsel väärtusel, kus kaalutud k-lähinaabri funktsioon käitub paremini.

2.2 Tuumaga silumine numbrilise tunnuse korral

Antud alampeatüki tulemused pärinevad raamatust Friedman, Hastie ja Tibshirani (2016) kui ei ole viidatud teisiti.

Eelnevalt oleme arutanud numbriliste tunnuste prognoosimist valemiga

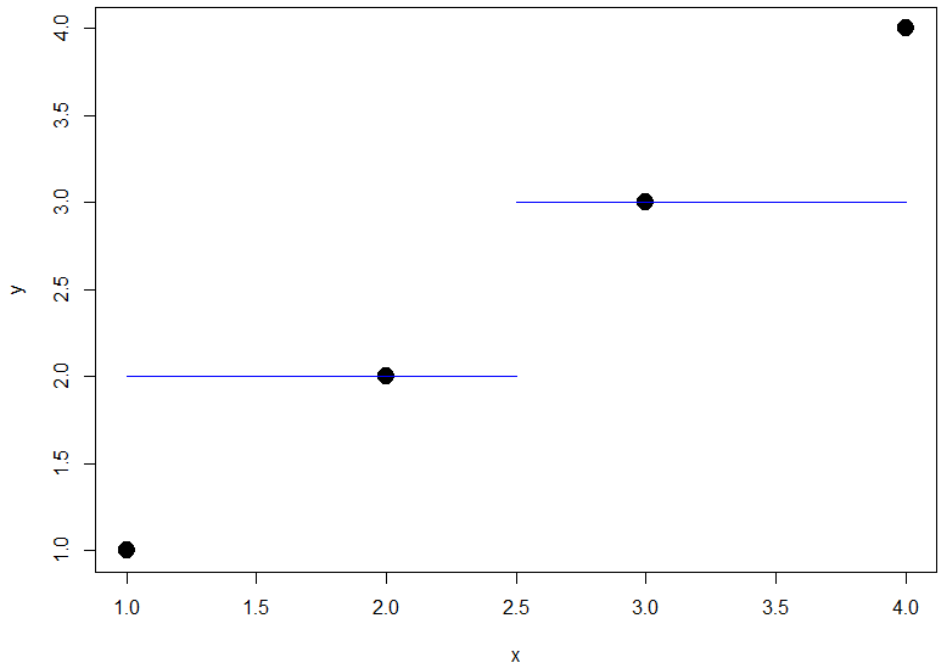
$$\hat{f}(x) = Ave(y_i|x_i \in N_k(x)), \tag{2.2}$$

kus Ave tähistab keskmise võtmist ning $N_k(x)$ tähistab k lähimat vaatlust vaatlusest x. Antud viisil prognoosides saame funktsiooni, mis ei ole pidev kogu oma määramispiirkonna ulatuses. Vaatleme tunnuse y prognoosimist Tabelis 5 toodud andmete põhjal.

Tabel 5: k-lähinaabri algoritmi näite andmed

x	1	2	3	4
y	1	2	3	4

Määrame $k = 3$ ja kauguste arvutamiseks kasutame valemit (1.1).



Joonis 7: K-lähinaabri prognoos piirkonnas $x \in [1; 4]$ $(k = 3)$

Näeme, et tunnuse y väärtusi prognoosiv funktsioon ei ole pidev punktis $x = 2,5$. Prognoosi pidevuse probleemi saame lahendada, kui anname igale naabruskonna vaatlusele erineva kaalu.

Üks võimalikest meetoditest on tuumaga silutud akende kasutamine. Tuumaga silutud akna korral saavad treeningandmestikus suurema kaalu väärtused, mis on sisendandmete seletavatele tunnustele lähemal.

Tuumaga silutud meetod (*Nadaraya–Watson kernel-weighted average*) leidmaks hinnangu $\hat{f}(x_0)$ väärtust punktis x_0

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{n} T_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{n} T_\lambda(x_0, x_i)}, \tag{2.3}$$

kus

$$T_\lambda(x_0, x) = D\Big(\frac{|x - x_0|}{\lambda}\Big) \tag{2.4}$$

ning

$$D(t) = \begin{cases} \frac{3}{4}(1-t^2) & \text{kui } |t| \leq 1 \\ 0 & \text{vastasel juhul.} \end{cases} \tag{2.5}$$

Antud tuumaga silumise akent (valem 2.5) nimetatakse Epanechnikovi tuumaks. Sääraselt lähenedes saame prognoosiva funktsiooni $\hat{f}(x)$, mis on pidev kogu x määramispiirkonnas.

Valem 2.5 on ainult üks näidetest. Valemi võib ka defineerida kujul

$$D(t) = \begin{cases} (1-|t|^3)^3 & \text{kui } |t| \leq 1 \\ 0 & \text{vastasel juhul.} \end{cases} \tag{2.6}$$

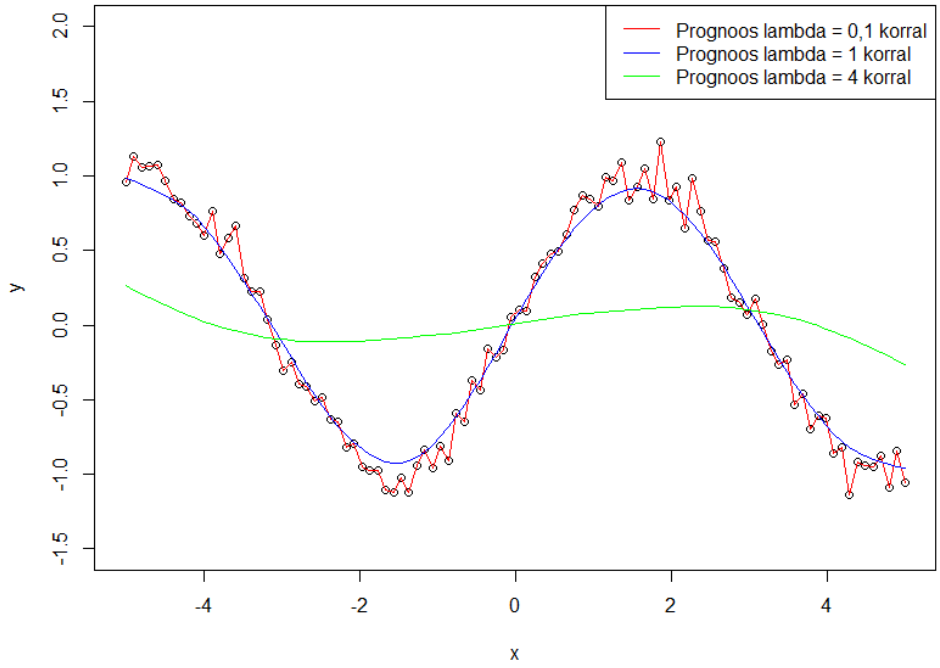
või

$$D(t) = \phi(t) \tag{2.7}$$

kus $\phi(t)$ on normaaljaotuse tihedusfunktsioon. Funktsiooni $D(t)$ valik sõltub prognoositava andmestiku omadustest.

Peale kaalude saame valemis 2.3 defineerida ka parameetri λ väärtuse. Toome näite, kuidas akna laiuse väärtus mõjutab tuumaga silumise saadud väärtust. Parameetrit λ kasutame naabruskonna suuruse määramiseks.

Näitamaks λ väärtuse mõju prognoosile defineerime $y = \sin(x) + N(0; 0, 1)$. Prognoosime $\hat{y}$ väärtusi piirkonnas $x \in [-5; 5]$. Fikseerime $\lambda = \{0, 1; 1; 4\}$ ja leiame prognoosi funktsiooni iga λ väärtuse korral. Saame tulemuseks



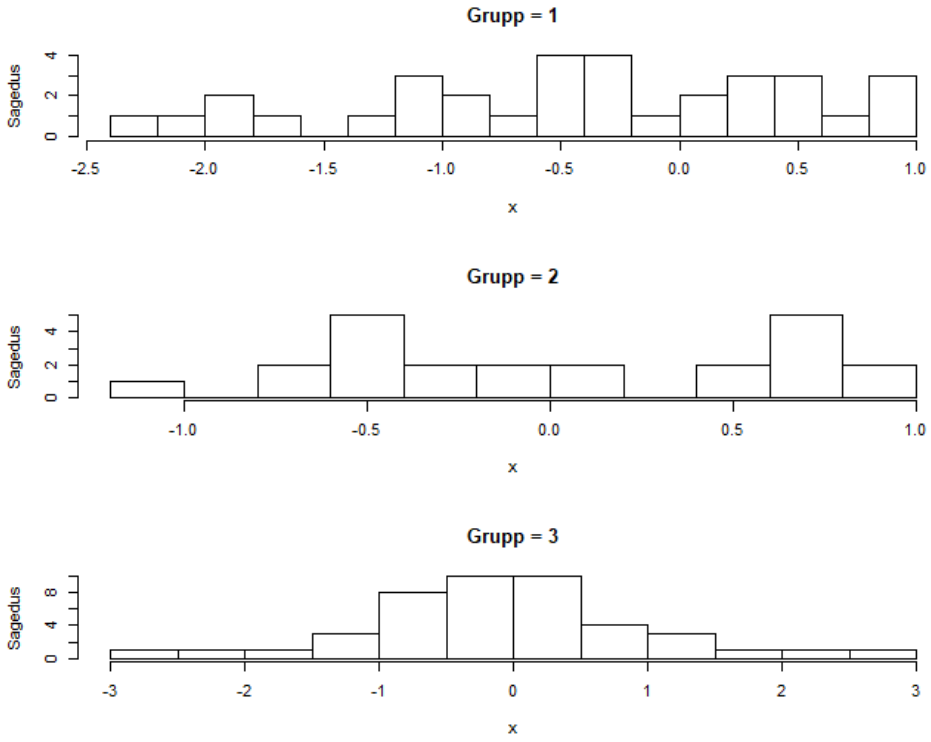
Joonis 8: Hinnang väärtusele erinevate λ väärtuste korral.

Näeme, et prognoosi funktsioonid on pidevad kogu oma määramispiirkonnas. Väiksem parameetri ($\lambda = 0,1$) väärtus on väga tundlik müra suhtes. Samas, suure parameetri ($\lambda = 4$) väärtuse korral on prognoosi funktsioon liiga sile.

2.3 Tuumaga silumine nominaalse tunnuse korral

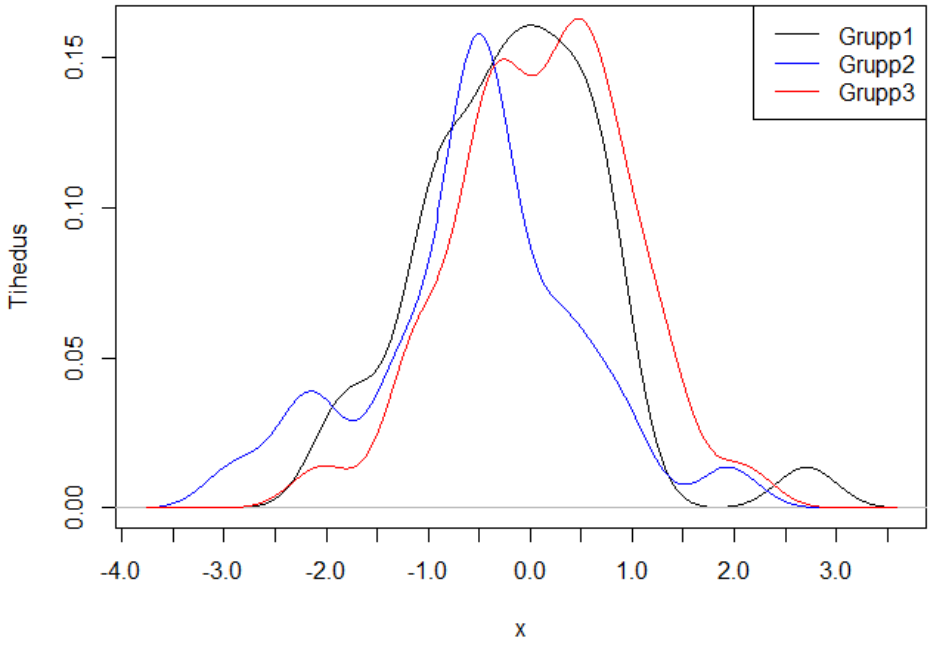
Sarnaselt alapeatükkis 2.2 mainitule saame käituda ka nominaalse tunnuse kodeerimisel tekkivate binaarsete tunnuste korral.

Genereerime 100 tunnuste paari (x,Grupp), kus $x \sim N(0; 1)$ ning tunnusel Grupp on 3 võimalikku nominaalset väärtust. Visualiseerime saadud andmed histogrammil.



Joonis 9: Nominaalse tunnuse hajuvus üle numbrilise tunnuse x.

Tuumaga silumisel kasutame silumisparameetrit $\lambda = 1$ ning kasutame tuumaga silumisel valemit (2.5).



Joonis 10: Tuumaga silumine diskreetse tunnuse korral.

Uue vaatluse $x_0 \in [-5; 5]$ väärtuse korral määrame prognoositava tunnuse "Grupp"väärtuseks kategooria, millele vastava binaarse tunnuse hinnatud tihedus on vaadeldavas punktis x_0 maksimaalne.

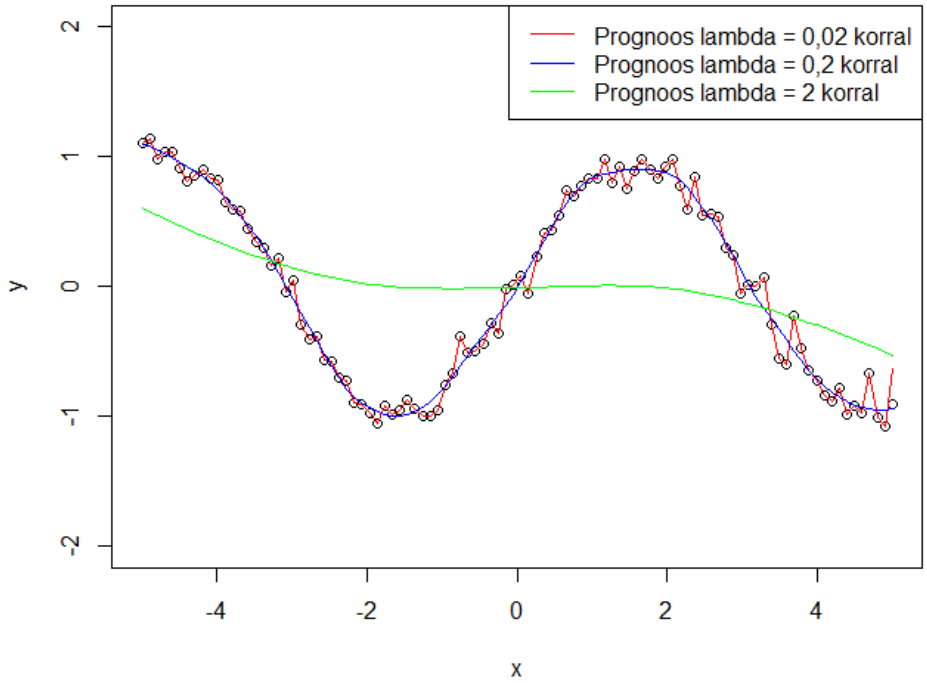
2.4 Lokaalse regressiooni mudel

Antud alapeatüki tugineb raamatule Cleveland (1979) kui ei ole viidatud teisiti.

Lokaalne regressiooni mudel on mitteparameetriline meetod hindamaks prognoositavat väärtust üle treeningandmestiku. Siingi on tulemuseks funktsioon, mis on sile kogu oma määramispiirkonnas. Olgu meil sisendandmestik X, millele tahame prognoosida tunnuse y väärtusi.

Olgu meil prognoositav vaatlus $x_0 \in X$. Prognoosimaks vaatlusele x_0 prognoosi y, kus alamhulk valitakse kauguste põhjal. Silumisparameetriga λ määratakse maksimaalne kaugus vaatlusest x_0. Saadud alamhulga peale luuakse lokaalne lineaarse regressiooni mudel $\hat{y}_i = f(x_i), x_i \in N(x_0)$, kasutades vähimruutude meetodit. Selle abil arvutatakse prognoos x_0 jaoks kujul $\hat{y} = f(x_0)$. Antud meetodi miinuseks on suurem arvutusvõimsus võrreldes teiste töös mainitud meetoditega.

Näitame kuidas silumisparameetri λ valik mõjutab lokaalse regressiooni mudeli korral tunnuse $\hat{y}$ hinnangut. Olgu meil $x \in [-5; 5]$ ning $y = sin(x) + N(0; 0, 1)$. Fikseerime silumisparameetri $\lambda \in \{0, 02; 0, 2; 2\}$



Joonis 11: Parameetri λ mõju lokaalse regressiooni mudeli prognoosi hindamisel.

Näeme, et prognoosi funktsioonid on pidevad kogu oma määramispiirkonnas. Väiksem silumisparameetri ($\lambda = 0,02$) väärtus on väga tundlik müra suhtes. Samas suure parameetri ($\lambda = 2$) väärtuse korral on prognoosifunktsioon liiga sile.

3 Numbrilised tulemused

Uurime k-lähinaabri algoritmi kasutates Bondora laenude pankrotistumise prognoosimist (Bondora (2023)). Mudel programmeeriti programmeerimiskeeles R. Mudeli ehitamiseks kasutame järgnevaid tunnuseid:

- pankroti langemise kuupäev
- haridustase
- vanus
- sugu
- laenatud summa
- intressimäär
- laenupikkus kuudes
- igakuise makse suurus
- laenaja residentsus

Tunnus „laenaja residentsus" omab 5 võimaliku väärtust: Eesti(EE), Hispaania(ES), Soome(FI), Holland(NL) ja Slovakia(SK). Kuna meil ei ole kuigi head põhjendust, miks peaks üks riik teisega sarnasem olema, siis kodeerime tunnuse *One-hot* meetodiga. Tunnuse „haridustase" korral käitume vastupidiselt ja eeldame, et saame neid järjestada. Tunnusel „haridustase" on järgnevad võimalikud väärtused: 1 - alghariidus, 2 - põhiharidus, 3 - kutseharidus, 4 - keskharidus, 5 - kõrgharidus. Tunnuse „sugu" väärtusi käsitleme binaarse tunnusena. Ülejäänud tunnuseid käsitletakse pideva arvtunnusena.

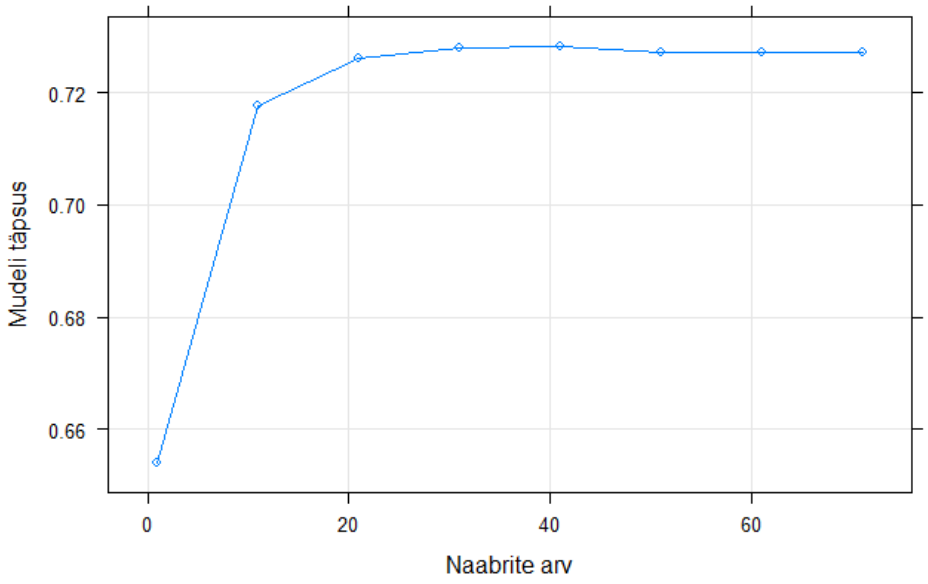
Pankrotti langemise kuupäeva abil saame väärtused tunnusele "pankrot". Kui andmestikus pole laenul pankrotti langemise kuupäeva, pole kliendil laenu tagasi maksimisel probleeme tekkinud. Enne mudeli ehitamist eemaldame laenud, mille laenutähtaeg on tulevikus ja mis pole pankrotistunud. Seda seetõttu, et nende puhul pole teada, kas pankrotistumine toimub või mitte. Mudelite koostamisel loome algsest valimist kaks lõikumatut valimit, kus treeningandmestikus on 70% algse andmestiku vaatlustest ja testandmestikus on 30% algse andmestiku vaatlustest. Treeningandmetes on meil 86 363 vaatlust, millest 60 847 on olnud pankroti seisundis ning

25 516 pole olnud pankroti seisundis. Testandmetes on meil 37 013 vaatlust, millest 26 178 on olnud pankroti seisundis ning 10 835 pole olnud pankroti seisundis.

3.1 Mudeli ehitamine k-lähinaabri algoritmiga

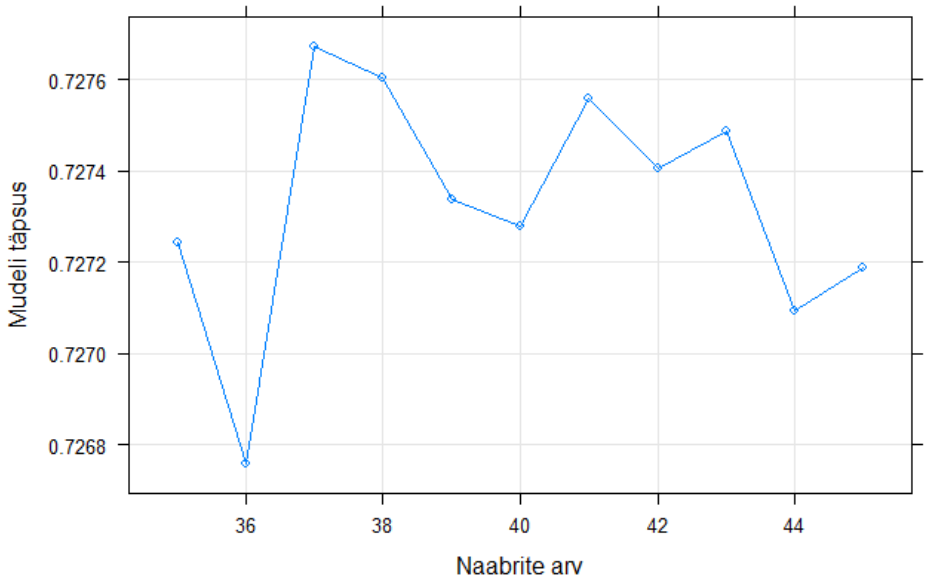
Enne mudeli ehitamist leiame parima parameetri k väärtuse. Seda teeme sarnaselt alapeatükis 1.5 näidatule. Mudeli treenimisel kasutasime valemit (1.3) parameetriga $p = 2$.

Parima parameetri k arvutamiseks sobitame kõigepealt viis mudelit, kus $k \in \{1, 11, 21, 31, 41, 51\}$



Joonis 12: Parima parameetri k väärtuse leidmine.

Näeme jooniselt, et parim parameetri k väärtus on väärtuste $k = 30$ ning $k = 50$ vahel. Ehitame uued mudelid, kus parameetri väärtused on $k \in \{35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45\}$.



Joonis 13: Parima parameetri k väärtuse leidmine.

Näeme jooniselt, et parim parameetri väärtus on $k = 37$.

3.2 Mudeli ehitamine kaalutud k-lähinaabri algoritmiga

Rakendame saadud parameetri $k = 37$ põhjal uue mudeli. Enne mudeli ehitamist standardiseerime treening- ja testandmestikus kõik selgitavad tunnused. Saadud andmete pealt ehitame mudeli ning saame mudeli täpsuse hinnanguks $\frac{2702+24461}{37013} = 0.7339$.

Realiseerime uue mudeli. Enne mudeli ehitamist standardiseerime treening- ja testandmetest ainult numbrilised seletavad tunnused. Binaarseid seletavaid tunnuseid me ei standardiseeri. Ehitame saadud andmete peale mudeli. Mudeli täpsuse hinnanguks saame $\frac{2752+24450}{37013} = 0.7349$, mis on veidi väiksem, kui eelmisel mudelil.

Realiseerime samadest põhimõtetest lähtuvalt kaks uut mudelit. Mudelil arvestame ka seekord tuumafunktsiooniga ning anname vaatlustele juurde kaalud. Enne mudeli ehitamist, standardiseerime numbrilised tunnused. Mudeli ehitamisel anname

kaaludeks valemid (2.5) ning (2.7). Epanechnikovi tuumaga silumise korral saame mudeli täpsuse hinnanguks $\frac{2986+24146}{37013} = 0.7321$. Kui võtame aluseks normaaljaotuse tihedusfunktsiooni tuumaga silutud funktsiooni, siis saame saadud mudeli täpsuse hinnanguks $\frac{2981+24165}{37013} = 0.7334$.

Iga ehitatud mudeli korral tuleb täpsuse hinnanguks 73%. Antud juhul on parima täpsuse hinnanguga mudel, kus standardiseerisime ainult numbrilise väärtusega tunnused.

Mudelid ehitasime kasutades tarkvaraprogrammi R. Mudelite ehitamiseks kasutasime tarkvaraprogrammi R funktsiooni kknn (Schliep, Hechenbichler ja Lizee (2016)). Leitud mudelite ehitamise protsess on välja toodud Lisas 1.

Kokkuvõte

Magistritöö eesmärk oli uurida k-lähima naabri algoritmi ning selle modifikatsioone.

Töö on jagatud kolme suuremasse ossa. Esmalt uurisime vaatluste vaheliste kauguste määramist ning kuidas kuidas arvutada nominaalsete tunnuste vahelist kaugust. Antud teadmisi rakendasime k-lähinaabri algorimi ehitamisel. Töö teises osas uurisime k-lähinaabri algoritmi modifikatsioone. Uurisime tuumaga silutud funktsioone ning nende mõju prognoosi hinnangule. Viimasena uurisime algoritimi töövõimet, kasutades bondora veebisaidilt saadud andmeid. Ehitasime mudelid, mis hindasid tõenäosust, et kliendil tekib laenu tagasi maksmisega probleeme. Mudelite koostamisel lõime algsest valimist kaks lõikumatut valimit, kus treeningandmestikus on 70% algse andmestiku vaatlustest ja testandmestikus on 30% algse andmestiku vaatlustest. Andsime omapoolse hinnangu, milline mudel võiks olla parim. Otsuse tegemiseks valisime suurima väärtusega mudeli täpsuse hinnangu. Parimaks osutus mudel, kus skaleeriti kõik numbriliste väärtustega seletavad tunnused.

Kasutatud allikad

Bondora (2023). *LoanData*. URL: https://www.bondora.com/et/public-reports (vaadatud 12.05.2023).

Cleveland, William S. (1979). *Robust Locally Weighted Regression and Smoothing Scatterplots. with Applications in R*. Journal of the American Statistical Association.

Fisher, Ronald Aylmer (1936). *The use of multiple measurements in taxonomic problems*. Czechoslovak Mathematics Journal.

Fix, Evelyn ja Joseph Lawson Hodges (1952). *Discriminatory Analysis - Nonparametric Discrimination: Small Sample Performance*. California University Berkley.

Friedman, Jerome, Trevor Hastie ja Robert Tibshirani (2016). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. 2. väljaanne. Springer.

Hechenbichler, Klaus ja Klaus Schliep (2004). *Weighted k-Nearest-Neighbor Techniques and Ordinal Classification*.

James, Gareth, Daniela Witten, Trevor Hastie ja Robert Tibshirani (2013). *An Introduction to Statistical Learning. with Applications in R*. Springer.

Kuhn, Max ja Kjell Johnson (2013). *Applied Predictive Modeling*. Springer.

Schliep, Klaus, Klaus Hechenbichler ja Antoine Lizee (2016). *Weighted k-Nearest Neighbors*. URL: https://cran.r-project.org/web/packages/kknn/kknn.pdf (vaadatud 16.05.2023).

Spiegel, Murray R ja Larry J Stephens (2008). *Schaum's Outline of Statistics*. 4. väljaanne. Spiegel.

Lisa 1. k-lähinaabri algoritmi ehitamine programmis R

```
library(kknn)
# Kaalumata kNN mudel
m <- kknn(Defaulted~., train, test, k = 37,
    distance = 2, kernel = "rectangular", scale = T)
# Mudeli tapsuse hindamine
t <- table(m$fitted.values, test$Defaulted)
t
(sum(diag(t)))/sum(t) # 0.7339

# Kaalumata kNN mudel,
# kus ainult numbrilised tunnused on skaleeritud
m1 <- kknn(Defaulted~., train_data1, test_data1, k = 37,
    distance = 2, kernel = "rectangular", scale = F)
# Mudeli tapsuse hindamine
t1 <- table(m1$fitted.values, test$Defaulted)
t1
(sum(diag(t1)))/sum(t1) # 0.7349

# Kaalud kNN mudel,
# kus ainult numbrilised tunnused on skaleeritud
# epanechnikovi aken
m2 <- kknn(Defaulted~., train_data1, test_data1, k = 37,
    distance = 2, kernel = "epanechnikov", scale = F)

# Mudeli tapsuse hindamine
t2 <- table(m2$fitted.values, test$Defaulted)
```

```
t2
(sum(diag(t2)))/sum(t2) # 0.7321

# Kaalud kNN mudel,
# kus ainult numbrilised tunnused on skaleeritud
# normaaljaotuse aken
m3 <- kknn(Defaulted~., train_data1, test_data1, k = 37,
    distance = 2, kernel = "gaussian", scale = F)

# Mudeli tapsuse hindamine
t3 <- table(m3$fitted.values, test$Defaulted)
t3
(sum(diag(t2)))/sum(t2) # 0.7334
```

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Hardi Roosi,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „K-lähinaabri meetodi ja selle modifikatsioonide rakendamise tehnilistest detailidest ja nen-de võimalikust mõjust tulemuste täpsusele", mille juhendaja on Raul Kangro, 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 4.0, 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 punktides 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
16.05.2023