TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Taavi Aun 143024

TITANICU REISIJATE PÄÄSEMIST ENNUSTAVA MUDELI LOOMINE

Bakalaureusetöö

Juhendaja: Ants Torim
Lektor

Tallinn 2017

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Taavi Aun

22.05.2017

Annotatsioon

Minu töö eesmärk on uurida Titanicu pardal olevate reisijate kohta saadaolevaid andmeid ning nende põhjal koostada erinevaid masinõppe meetodeid kasutades mudelid, mis suudavad ennustada pardal olnud reisija pääsemist. Antud töös on andmekaeve teostamiseks tuginetud CRISP-DM raamistikule.

Töös on täpsemalt välja toodud reisijate andmete atribuudid ning nende kirjeldused. Eraldi on näidatud statistiliselt, kuidas jagunes reisijate pääsemine ja hukkumine atribuutide järgi.

Oma töös teen ma selgeks masinõppe meetodite põhimõtted ja hiljem kasutan neid mudelite loomiseks. Peale mudelite loomist kasutan ristkontrolli, et anda mudelite täpsusele hinnang ning leida, millised mudelid olid parimad.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 44 leheküljel, 9 peatükki, 18 joonist, 0 tabelit.

Abstract

Creation of predictive model for the survival of Titanic passengers

The main aim of this research is to investigate available data about the passengers of Titanic and create models capable of predicting survival of passengers based on the data using different machine learning methods. Data mining in the research has been done using CRISP-DM framework.

Research covers detailed descriptions of the data attributes. Furthermore the research also shows the distribution of survival based on one attribute at the time.

In this research I will explain basic principles of the machine learning methods that have been used for creating predictive models. After the creation of the models I will use cross-validation to assess the accuracy of models to find out the most accurate ones.

The thesis is in Estonian and contains 44 pages of text, 9 chapters, 18 figures, 0 tables.

Lühendite ja mõistete sõnastik

Atribuut	*Attribute* Atribuutideks nimetatakse iga instantsi kohta käivaid tunnusjooni, mille arv ja võimalikud väärtused on teada [1].
Rist-valideerimine	*Cross-validation* Rist-valideerimine on ennustus mudeli hindamise meetod. Rist-valideerimist kasutades ei võeta kõiki andmeid treeningandmeteks vaid osa neist eemaldatakse ja kui mudel on treenitud, kontrollitakse mudeli täpsust „uute“ andmete peal, mida treenimiseks ei kasutatud [3].
CSV	*Comma separated values* Komaeraldusega väärtused Porditav failivorming, kus andmebaasikirjed on üksteisest eraldatud komadega. Selles vormingus on iga rida üks kirje, mille väljad on üksteisest komadega eraldatud. Komade järel võib olla suvaline arv tühikuid ja/või tabeldusmärke (tab character), sest neid ignoreeritakse. Kui väli ise sisaldab koma, siis peab kogu väli olema ümbritsetud jutumärkidega [2].
Instants	*Instance* Instants on näide(rida) andmekogumis, millel on teatud arv atribuute [1].
Aritmeetiline keskmine	*Mean* Aritmeetiline keskmine on võrdne arvude summa jagatuna nende arvude koguarvuga.
Standardhälve	*Standard deviation* Standardhälve iseloomustab vastuste hajuvust keskmise ümber. Standardhälbe saab, kui leida kõigi vastajate vastuste erinevus üldisest keskmisest ning arvutada nende erinevuste keskmine [5].
Andmekogum	*Dataset* Andmekogum on maatriks, milles on kõik andmed ja mis on suurusega instantside arv * atribuutide arv [1].
Ülesobitamine	*Overfitting* Ülesobitamise all mõeldakse olukorda, kus mudel ennustab treeningandmeid liiga hästi, kuid uute andmete ennustamisel on mudeli täpsus madal. Olukord tekib, kui mudel loob andmetes sisalduva müra järgi reegleid, mitte ei üldista. Müraks loetakse juhuslikke kõikumisi treeningandmete seas. Ülesobitamist on võimalik vältida kärpides mudeli täpsust treeninandmetel treenides [24].

Eukleidiline distants — *Euclidean distance*

Eukleidiline distants on geomeetriline kaugus kahe punkti vahel mitmedimensioonilises ruumis. Eukleidilist distantsi punktide *x* ja *y* vahel arvutatakse valemiga: $distants(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$, kus *i* on dimensioonide arv [11].

Keskmine absoluutviga — *Mean absolute error (MAE)*

Keskmine absoluut viga näitab, kui palju erinevad tegelikud väärtused ennustatud väärtustest. [14] Väikseim väärtus, mis ühtlasi näitab ka, et ennustused olid täpsed, on 0. Tulemuseks saavad olla ainult positiivsed väärtused. [15] Keskmine absoluutviga arvutatakse valemiga:

$$MAE = \frac{\sum_{i=1}^{n} |x_i - y_i|}{n}$$

Kus y_i on tegelik väärtus, x_i on ennustatud väärtus ja n on ennustavate väärtuste arv [27].

Sisukord

Jooniste loetelu

1 Sissejuhatus

Titanic oli Briti reisilaev, mis sõitis vastu jäämäge oma esimesel reisil üle Atlandi ookeani. Intsident juhtus 15.04.1912 varahommikul ning selle käigus hukkus umbes 2200 pardal olnud inimesest üle 1500 [29].

Töö annab ülevaate sellest, millistel inimestel oli parim pääsemislootus. Peale lühidat ülevaadet loon erinevaid masinõppe meetodeid kasutades mudelid, mis suudaksid ennustada reisijate pääsemist ja võrdlen loodud mudelite täpsust kasutades ristkontrolli.

Alguses on välja toodud kasutatud tarkvara ja CRISP-DM raamistiku põhimõtted kuidas antud mudeli järgi lahendada andmekaeve probleeme. Järgmisena on välja toodud töös kasutatavate masinõppe meetodite kirjeldused ja loogika, kuidas nad toimivad. Edasi on uuritud algandmeid ja nendega seonduvaid probleeme, näiteks osati puudulikud andmed ja nende lahendamine. Peale andmete parandamist on toodud välja statistika osade atribuutide kohta, kuidas nad mõjutasid reisija pääsemist. Järgmiseks loon masinõppe meetodeid kasutades mudelid, mis suudaksid ennustada reisija pääsemist ja võrdlen loodud mudeleid, et leida täpseimad. Töö käigus toon lühidalt välja ka erinevused teise analüütiku koostatud tööga, kus on algandmeteks antud tööga samad andmed kaggle keskkonnast.

2 Kasutatud tarkvara

Antud töös kasutatakse andmeanalüüsi ja ennustusmudelite koostamiseks Continuum Analyticsi poolt pakutavat vabavara Anaconda3.

2.1 Anaconda3

Antud lõputöös on kasutatud vabavara, Anaconda3-4.3.1, mis sisaldab endas condat, conda-buildi, Pythonit ja üle 150 automaatselt installitud paketi [12].

Esimene põhjus, miks antud tarkvara on valitud, sest sisaldab endas piisavalt võimalusi käesoleva töö läbi viimiseks ning mida on kiidetud ka mitmete andmeanalüütikute poolt [13]. Kuigi antud allikas on kommenteerijad anonüümsed ja kõike ei tasu täielikult uskuda, annab see hea ülevaate, mida inimesed tarkvarast arvavad ning oli ka üks põhjuseid, miks töö koostaja otsustas Anaconda3 kasuks.

Teine põhjus, miks töö tegemisel on kasutatud Anacondat, on mugavus. Windowsi masinale tarkvara installimine on väga lihtne ja kiire ning nagu eespool mainitud, sisaldab Anaconda kõike, mida antud töö raames vaja on.

3 Metoodika

Antud töös on andmekaeve teostamiseks tuginetud suuremas osas CRISP-DM mudelile. Ainsaks erinevuseks on andmete mõistmise ja andmete ettevalmistamise tihe kooslus, sest algandmetel on vanuse atribuut puudulik ning seetõttu koostatakse detailsem analüüs peale puudu olevate väärtuste asendamist.

Sarnaselt antud tööle, tundub, et ka teine andmeanalüütik kasutab CRISP-DM raamistikku, ainsa erinevusena, et tema töö jääb poolikuks vahetult peale andmete uurimist ja puudu olevate andmete asendamist.

3.1 Algandmed

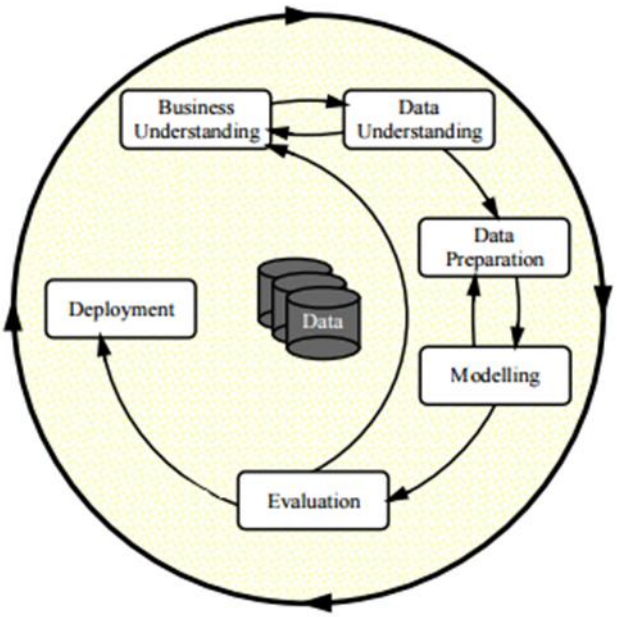
Antud töö jaoks kasutatavad algandmed on vabalt kätte saadavad Kaggle keskkonnast. Kaggle on 2010 aastal asutatud platvorm, mis korraldab andmeanalüüsi võistlusi, milles saavad osaleda statistikud üle maailma, et luua parimaid mudeleid andmete ennustamiseks. Kaggle korraldab ka värbamisvõistlusi, mille käigus analüütikud saavad võistelda ning võitjad kutsutakse interjuule juhtivatese andmeanalüüsi ettevõtetesse, näiteks Facebook, Winton Capital ja Walmart [26].

Algandmed on Kaggle keskkonnas üleval CSV faililaiendiga ning eraldi on ka välja toodud atribuutide kirjeldused ning võimalikud väärtused [30].

3.2 CRISP-DM

CRISP-DM ehk tööstusharude standard protsess andmekaeveks (*Cross Industry Process for Data Mining*) on mudel, mille järgi ka antud töös andmekaevet teostatakse.

Mudel määrab , millises järjekorras erinevad andmekaeve etapid toimuvad [25].



Joonis 1.CRISP-DM mudeli etapid

- Ärinõuete mõistmine (*Business understanding*)

Selles etapis pannakse paika andmekaeve eesmärgid ja nõuded äri perspektiivist [25]. Antud töös on selleks eesmärk, milleni tahetakse jõuda, ehk siis reisijate pääsemist ja hukkumist ennustavate mudelite loomine ja nende võrdlus ning tulemust kõige rohkem mõjutavate atribuutide välja toomine.

- Andmete mõistmine (*Data understanding*)

Andmete mõistmise etapp algab andmete kogumisega ja edasi õpitakse andmeid tundma, et teada saada algandmetega seotud probleeme ning leida hüpoteese peidetud infoks [25]. Antud töös on selleks etapiks algandmete atribuutide kirjeldus ja täpsem atribuutide uurimine, kus uuritakse andmeid ning püstitatakse hüpotees peidetud info kohta, mis kohe välja ei tule.

- Andmete ettevalmistamine (*Data preparation*)

Andmete ettevalmistamine hõlmab endas kõiki tegevusi loomaks lõplikku andmekogumit algandmetest. Andmete ettevalmistamise alla lähevad atribuutide valikud, andmepuhastus, uute atribuutide loomine ning andmete muutmine mudelite loomiseks [25].

- Modelleerimine (*Modelling*)

Modelleerimise käigus luuakse erinevaid mudeleid ja kalibreeritakse nende parameetreid, et saada optimaalseid tulemusi. Andmete ettevalmistamine ja modelleerimine on tihedalt seotud, sest tihti tulevad andmete muutmiseks ideed just selles etapis [25]. Antud töös on modelleerimise etapiks erinevate ennustavate mudelite loomine.

- Hindamine (*Evaluation*)

Hindamise käigus hinnatakse mudeleid ja vaadatakse üle mudelite loomiseks tehtud sammud, et olla kindel, et püstitatud ärinõuded saaksid täidetud. Selle etapi lõpuks peaks selguma, kas andmekaeve tulemusi ka reaalselt rakendatakse [25]. Antud töös on hindamise etapiks erinevate mudelite täpsuste võrdlus ja hindamine, millised mudelid said kõige paremini hakkama.

- Kasutuselevõtt (*Deployment*)

Antud etapiskorrastatakse andmekaeve käigus saadud teadmised ja viiakse need kliendile arusaadavale kujule. Olenevalt nõuetest, võib see etapp sisaldada ainult aruande koostamist või uuesti kogu andmekaeve protsessi kordamist. Paljudel juhtudel otsustab mitte andmeanalüütik vaid klient, kas loodud mudel võetakse kasutusele või mitte [25]. Antud töös on selleks etapiks kirjaliku dokumentatsiooni koostamine.

4 Ennustusmeetodid

Antud töös on reisija pääsemiseks või hukkumiseks loodud seitse erinevat mudelit, kasutades kuute erinevat algoritmi nende loomiseks. Käesolevas peatükis on välja toodud töös kasutatud algoritmide lühike kirjeldus, kuidas nende abil tulemusi ennustatakse.

4.1 Otsustuspuu

Decision tree

Otsustuspuu on masinõppe õppemeetod, mille eesmärgiks on luua mudel, mille abil saab sihtmuutuja väärtust ennustada kasutades lihtsaid reegleid, mis on tuletatud andmeid analüüsides [4].

Antud töös kasutatakse just nimelt klassifitseerimis- ja regressioonipuud, lühidalt CART (Classification and Regression Tree). CART rakendamiseks on vaja ühte prognoositavat muutujat, mis võib olla nii diskreetne (kategooriline), kui ka pidev ja potentsiaalseid riskifaktoreid, mis võivad samuti olla nii pidevat kui ka diskreetsed. Otsustuspuu koostamisel jäetakse välja riskifaktorid, mis prognoositava tulemusele mõju ei avalda [8]. Klassifitseerimis- ja regressioonipuu suurim erinevus on see, milliseid väärtuseid nad prognoosivad. Nimelt saavad klassifitseerimispuu prognoositavateks väärtusteks olla ainult klassid täisarvu (Integer) kujul ning prognoositavad tulemused saavad olla treeningandmete seas olevad sihtmuutuja väärtused [9]. Regressioonipuu suudab prognoosida ka ujukomaga (Float) arve, mida treeningandmete seas ei olnud [10].

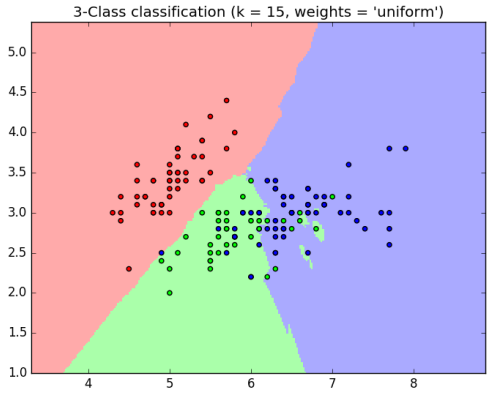
Klassifitseerimispuud kasutatakse töö lõpu poole, kui üritatakse ennustada, kas antud parameetritega reisija pääses või mitte, ehk üritatakse ennustada, kas reisija klassiks on „1“ – reisija pääses või „0“ – reisija hukkus.

Regressioonipuud kasutatakse andmete ennustamisel. Kuna puudu on väga paljude reisijate vanused ja nende vanuste asendamine lihtsalt aritmeetilise keskmisega või muu sarnase lihtsa meetodiga ei sobi, siis oleks vaja meetodit, mis ei viiks lõpliku pääsemise ennustust liiga ebatäpseks.

4.2 Lähima naabri meetodid

Nearest-neighbor methods

Lähima naabri meetodid on eksemplari põhised õppemeetodid. Nad ei loo keskset mudelit, vaid kasutavad lähedal asuvaid treeningandmepunkte, et ennustada sihtmuutuja väärtust. Andmepunktide kaugust enamasti arvutatakse eukleidilise distantsi abil. Lähedal asuvate andmepunktide arv võib olla ette antud või võib oleneda ka andmepunktide tihedusest (raadiusel baseeruv naabri õpe). Sarnaselt otsustuspuule, võib ka lähima naabri meetodeid kasutada nii klassifitseerimiseks, kui ka regressioni probleemide lahendamiseks [6].



Joonis 2. Klassifitseerimine kasutades lähima naabri meetodit

Joonisel 2 on kujutatud ka andmete klassifitseerimist kasutades lähima naabri meetodit. Antud töös seda graafiliselt kujutada ei saa, sest parameetreid, millest reisija pääsemine või hukkumine sõltub, on liiga palju ja seetõttu ei oleks joonis arusaadav ega informatiivne.

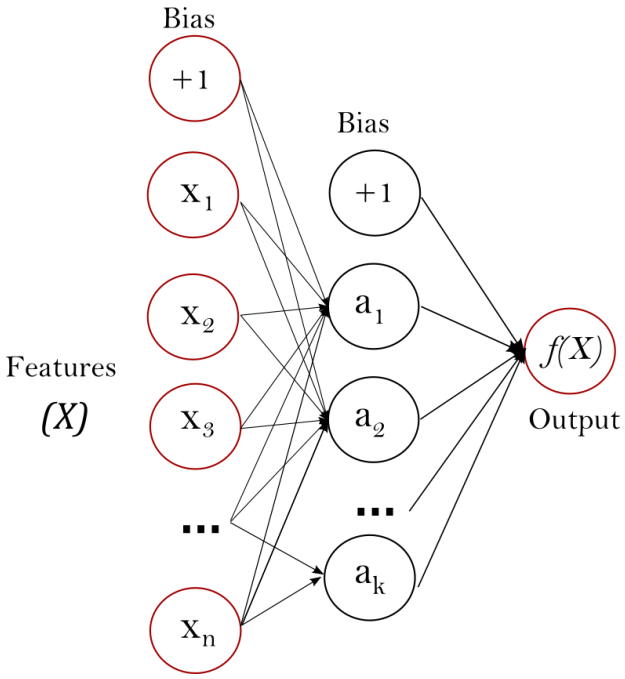
Antud töös kasutatakse lähima naabri meetodit töö lõpu poole klasifitseerimiseks, täpsemalt, reisija pääsemise või hukkumise ennustamiseks. Klassifitseerimiseks raadiusel põhinevat naabriõpet ei kasutata, sest meetodi rakendamiseks, tuleb antud andmete puhul raadius viia liiga suureks ja seeläbi kannatab ennustuse täpsus ning seetõttu tuginetakse ainult lähimatele naabritele, hoolimata nende kaugusest. Küll aga antakse ette, mitut lähedalt asuvat andmepunkti kasutatakse. Antud meetodit rakendatakse kaks korda, kasutades ühel korra kolme lähimat naabrit ja teisel korral viite lähimat naabrit.

4.3 Tehisnärvivõrgud

Artificial Neural Nets (ANN)

Tehisnärvivõrk on informatsiooni töötlev paradigma, mis on inpireeritud bioloogilistest närvisüsteemidest, näiteks aju. See koosneb mitmest omavahel tihedalt ühendatud töötluselementidest, neuronitest, mis töötavad koos, et lahendada konkreetne probleem. Tehisnärvivõrgud õpivad sarnaselt inimestele läbi kogemuse ja neid saab kasutada nii klassifitseerimiseks kui ka regressiooni probleemide lahendamiseks [16].

Antud töös kasutatakse MLP (Multi-layer Perceptron) algoritmi, et koostada mudel, mis ennustab, kas reisija pääseb või hukkub. Algoritm õpib andmetel treenides funktsiooni f(·) R^m→R^o, kus m on sisendi dimensioonide arv ja o on väljundi dimensioonide arv. Andes algoritmile atribuutide hulga X = x1, x2,...,xm ja sihtmuutuja y, õpib see mittelineaarse funktsiooni klassifitseerimiseks või regressiooni probleemi lahendamiseks. See on erinev logistilisest regressioonist seetõttu, et sisend- ja väljundkihi vahel võib olla mitu peidetud mittelineaarset kihti. Joonisel 3 on kujutatud MLP, millel on üks peidetud kiht.



Joonis 3. Ühe peidetud kihiga MLP närvivõrk

Kõige vasakpoolsem kiht, mis on ühtlasi ka sisendkiht, koosneb neuronitest {x_i |x_1,x_2,…,x_m}, mis tähistavad sisendatribuute. Iga peidetud kihi neuron muudab väärtused eelmisest kihist lineaarse kaalutud summaga w_1 x_1+w_2 x_2+...+w_m x_m,

millele järgneb mittelineaarne aktiveerimisfunktsioon g(·):R→R, näiteks hüperboolne tangens funktsioon. Väljundkiht võtab sisendiks viimase peidetud kihi väljundid ja muudab need väljundväärtusteks [17].

4.4 Naiivne Bayes

Naive Bayes

Naiivse bayesi meetodid on õppealgoritmid, mis kasutavad Bayesi teoreemi eeldusega, et iga atribuudi paari vahel puuduvad omavahelised seosed. Andes sihtmuutuja y ja atribuutide vektori x1 kuni xn, kus n on atribuutide arv, saame Bayesi teoreemi kasutades järgneva sõltuvuse:

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)}$$

Kasutades naiivset sõltumatuse eeldust, et

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y)$$

Kõikide *i*-de korral saab seda seost lihtsustada järgnevalt:

$$P(y|x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \dots, x_n)}$$

Kuna $P(x_1, \dots, x_n)$ on konstant, saame kasutada järgmist reeglit:

$$P(y|x_1, \dots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

$$\Downarrow$$

$$\hat{y} = arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i|y)$$

Ja me saame kasutada maksimaalse A tagantjärele (Maximum A Posteriori) ennustamist, et ennustada P(y) ja P(x_i |y).

Naiivse Bayesi klassifitseerijate peamine erinevus sõltubki P(x_i |y) jaotuses [18].

Antud töös kasutatakse Gaussi ehk normaaljaotusega naiivse Bayesi algoritmi reisijate pääsemist või hukkumist ennustava mudeli loomiseks.

4.5 Juhusliku metsa algoritm

Random forest algorithm

Juhusliku metsa algoritmi saab kasutada nii klassifitseerimiseks kui ka regressiooniks. Antud töös kasutatakse algoritmi, et luua mudel, mis ennustaks reisija pääsemist või hukkumist.

Juhusliku metsa algoritm on oma olemuselt otsustuspuu edasiarendus. Nimelt koostatakse palju otsustuspuid ja selle jaoks, et mingit sihtmuutujat ennustada, vaadatakse kõigi puude ennustusi. Klassifitseerimise puhul võetakse ennustuste mood ning regressiooni puhul loetakse metsa lõplikuks ennustuseks puude ennustuste keskmine [19].

Nii regressiooni kui klassifitseerimise puhul koostatakse mudel järgnevalt:

1. Kui treeningandmetes on N instantsi, valitakse juhuslikult N instantsi, kuid muudatustega algandmetest. Selles valimis olevatest andmetest saavad treeningandmed puu koostamiseks.

2. Valitakse juhuslikult m atribuuti, mille abil iga puu sõlm lahkneb. Kogu puu koostamise juures on m konstantne iga sõlme puhul. Kusjuures m<<M, kus M on kõigi atribuutide arv.

3. Iga puu kasvatatakse maksimumini seda pügamata [20].

4.6 Kohanemisvõimeline tõukealgoritm

Adaptive boost algorithm (AdaBoost)

Algoritm loob klassifitseerija kasutades paljusid nõrku klassifitseerijaid. Seda tehakse luues mudel kasutades treeningandmeid ja seejärel luues teise mudeli, mis üritab esimese mudeli vigu parandada. Mudeleid luuakse, kuni treeningandmeid suudetakse ennustada perfektselt või jõutakse loodud mudelite maksimum arvuni. AdaBoost oli esimene tõeliselt edukas tõukealgoritm, mis oli arendatud binaarseks klassifitseerimiseks [22].

Algoritm on oma olemuselt sarnane juhusliku metsa algoritmiga ning nõrkade mudelite loomiseks võibki kasutada otsustuspuid [23]. AdaBoost algoritmi võtsid esimesena kasutusele Yoav Freund ja Robert Schapire 1995 aastal [21].

Antud töös kasutatakse algoritmi, et luua mudel, mis ennustab reisijate pääsemist ja hukkumist ning nõrkade mudelite piirarvuks on seatud 50.

5 Algandmete atribuudid

Antud peatükis on välja toodud algandmete atribuudid, nende täpsemad kirjeldused ja võimalikud väärtused.

Survival	Pääsemine Võimalikud väärtused: 0 – reisija hukkus 1 – reisija pääses
Pclass	Sotsiaalne klass (*SES – socioeconomic status)* Võimalikud väärtused: 1 – ülemklass 2 – keskklass 3 – alamklass
Sex	Reisija sugu Võimalikud väärtused: Male – mees Female – naine
Age	Reisija vanus Vanus on murdosaline, kui reisija oli alla 1 aasta vana
Sibsp	Reisija vendade, õdede, abikaasade arv pardal kokku liidetuna Arvesse on võetud ainult vendi, õdesid, poolvendi, poolõdesid ja abikaasasid (kihlatuid ja armukesi see muutuja ei kajasta)
Parch	Reisija laste ja vanemate arv pardal kokku liidetuna Vanem – ema või isa

	Laps – poeg, tütar, kasupoeg, kasutütar
Ticket	Reisija pileti kood
Fare	Reisija pileti hind
Cabin	Reisija kajuti number
Embarked	Teekonna alustamise sadam Võimalikud väärtused: C – Cherbourg Q – Queenstown S - Southampton

6 Andmete ettevalmistamine

Antud peatükis on käsitletud algandmetega seotud probleeme. Põhiliseks probleemiks olid puudulikud algandmed, mille asendamist käesolevas peatükis ka kirjeldatakse. Lisaks luuakse üks lisa atribuut kahe algatribuudi baasil.

Võrdluses teise analüütiku koostatud tööle, on kõige suuremad erinevused just siin osas. Nimelt töötleb tema puudu olevaid andmeid hoopis teisiti ning nime atribuudist, mida antud töös pole mudelite loomiseks üldse kasutatud, toob tema välja atribuudi, mis näitab reisija tiitlit. Samuti kasutab ta ära kajuti atribuuti, et luua uus atribuut, mis näitab, mitu kajutit igal reisijal oli [31]. Kuna kajutite andmed on samuti puudulikud 687 reisijal, ei ole antud töös otsustatud neid andmeid kasutada ennustava mudeli loomiseks.

6.1 Embarked

Puudu olid kahe reisija teekonna alustamise sadamad 891 reisijast.

Ülejäänud reisijate alustamis kohad jagunesid järgnevalt: 8,64% reisijatest alustasid teekonda Queenstownist, 18.86% Cherbourgist ja 72,28% Southamptonist. Kuna puudu oli nii väheste reisijate andmed, eeldasin, et nad alustasid oma teekonda Southamptonist, sest see oli kõige populaarsem stardi koht. Peale asendamist alustas Southamptonist teekonda 72,5% reisijatest.

Võrdluseks teise analüütiku tööga võib mainida, et tema puuduvaid väärtusi selle atribuudi puhul ei töötlenudki, vaid jättis nad lihtsalt tühjaks. Selline lähenemine on antud atribuudi puhul täiesti sobilik, sest puudu on ainult 2 reisija andmed ning üldist statistikat need ilmselt oluliselt ei muudaks [31].

6.2 Age

Vanuse väärtus oli puudu 19,9% reisijatest. Kuna andmeid on niigi vähe, ei ole võimalik lihtsalt kustutada reisijate andmeid, kellel pole vanuse muutujat antud. Puudu olevate vanuste leidmiseks kasutasin regressioonipuud. Peale lühikest võrdlus teise analüütiku tööga, toon ka välja regressioonipuu treenimise.

Vanuse atribuudi muutuja juures on ka kõige suurem erinevus teise analüütiku tööga. Nimelt on teine analüütik puuduvad väärtused lihtsalt asendanud vanuse väärtustega, mis on juba varem esinenud [31]. Seetõttu peaks ka kannatama tema analüüsi täpsus ning antud töös loodud mudelid võiksid teoreetiliselt olla täpsemad ning tulla paremini toime uue infoga.

Esmalt treenisin mudeli. Selleks võtsin kõikide reisijate andmed, kellel on vanus antud. Mudeli treenimiseks on vaja, et kõik andmed on antud numbriliste väärtustega.

Mudeli treenimine:

1. Viisin kõikide atribuutide väärtused numbrilisele kujule.

Soo atribuut (*Sex*):

Male -> 0

Female ->1

Reisi alguskoha atribuut (*Embarked*):

S (Southampton) -> 0

C (Cherbourg) -> 1

Q (Queenstown) -> 2

2. Atribuutide valik mudeli jaoks:

    - *PassengerId:* Atribuut on kõigi reisijate puhul erinev ja näitab ainult reisijate järjekorra numbrit antud andmebaasis. Mudeli treenimiseks see ei sobi, kuna tekiks *overfittingu* probleem: mudel määraks reisija vanuse ainult tema järjekorra numbri järgi.

    - *Survival*: Atribuudil on kõik väärtused kõigil reisijatel olemas ning võimalikeks väärtusteks ainult „1“ ja „0“, sobib.

- *PcClass:* Atribuudil on kõik väärtused kõigil reisijatel olemas ning võimalikeks väärtusteks ainult „1“, „2“ ja „3“, sobib väga hästi mudeli treenimiseks.

- *Name:* Atribuut on olemas kõigil reisijatel, kuid tekib sarnane probleem, mis *PassengerId* puhulgi, samuti ei ole võimalik seda atribuuti viia nummbrilisele kujule, mudeli treenimiseks seda ei kasuta.

- *Sex:* Atribuudil on kõik väärtused olemas ja peale asendamist on võimalikeks väärtusteks ainult „0“ ja „1“, sobib mudeli treenimiseks.

- *SibSp:* Atribuudil on kõik väärtused olemas, sobib mudelitreenimiseks.

- *Parch:* Atribuudil on kõik väärtused olemas, sobib mudeli treenimiseks.

- *Ticket:* Atribuut on küll olemas kõigil reisijatel, kuid ühtib ainult osadel reisijatel. Reisija vanuse ennustamiseks see ei sobi, sest kõigil reisijatel on erinev pilet.

- *Fare:* Atribuut on olemas kõigil reisijatel ning juba viidud numbrilisele kujule. Mudeli treenimiseks sobib.

- *Cabin:* Atribuut on puudu 687 reisijal, mudeli treenimiseks nii puudulikud andmed ei sobi. Antud atribuut ei sobiks isegi juhul, kui kõigi reisijate andmed oleksid olemas, sest ühes kajutis olevate reisijate vanused võivad olla väga erinevad.

- *Embarked*: Peale puudu olevate väärtuste asendamist ja andmete ümber viimist numbrilisele kujule, on võimalikeks väärtusteks „0“, „1“ ja „2“, sobib vanuse ennustamiseks.

- *FamilySize*: Atribuut on olemas kõigil reisijatel ja sobib mudeli treenimiseks.

3. Seejärel treenisin mudeli. Enne mudeli treenimist eraldasin train_test_split() funktsiooniga testandmed, mida mudeli treenimiseks ei kasutata, et saaks kontrollida mudeli täpsust. Mudelid treenisin kasutades fit() meetodit. Kokku treenisin kolm mudelit, mille erinevus seisnes puu maksimaalses sügavuses. Puude sügavuseks olid 2, 5, 8 ja piiramatu sügavusega puu.

4. Ennustuste täpsuse hindamiseks kasutasin mean_absolute_error() funktsiooni. Kõige täpsemini ennustas antud funktsiooni hinnangul puu, mille sügavuseks oli 5, seda puud ma aga ennustamiseks ei valinud. Nimelt ennustasid kõik puud peale piiramatu sügavusega puu väga konkreetset vanust, näiteks arvas puu sügavusega 5, et enamike puudu olevate vanustega reisijate vanuseks oli 28. Piiramatu sügavusega puu valisingi just nimelt sel põhjusel, et ennustuste kõrge täpsus ei tuleks ainult tänu keskmise ennustamisele.

6.3 Family size

Family size – perekonna suurus, on ise loodud atribuut, mis näitab reisija pardal oleva pere suurust, see on leitud liites kokku reisija Sibsp ja Parch, et paremat ülevaadet reisija pardal olevate sugulaste kohta saada. Antud atribuudi on samamoodi loonud ka teine analüütik [31].

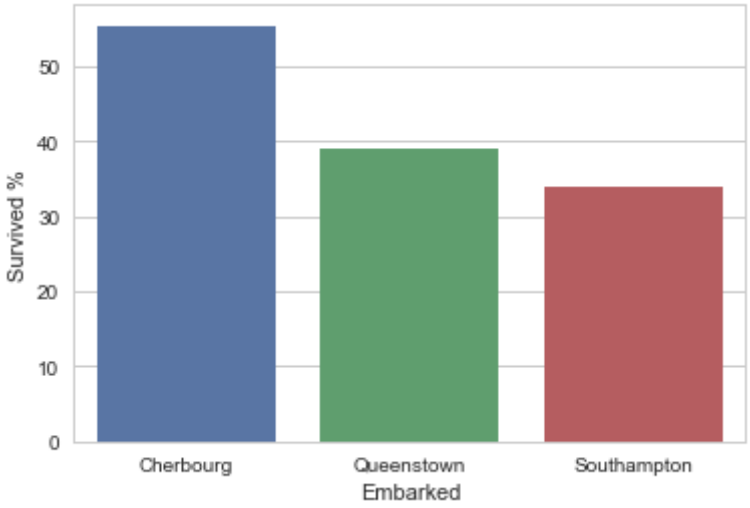
7 Atribuutide uurimine

Antud peatükis uurin lähemalt üksikuid atribuute ning lisan ka graafikud, et paremini illustreerida nende mõju reisijate ellujäämisele.

Ka atribuutide uurimine on mõlemas töös veidi erinev, nimelt kasutab teine analüütik graafikutel tihti reisijate arvu, mitte ei kirjelda reisijate pääsemist ja hukkumist protsentuaalselt. Protsentuaalne info on enamasti siiski välja toodud graafiku all tabeli kujul [31].

7.1 Embarked

Atribuut näitab, kust alustasid reisijad teekonda Titanicu pardal. Alustamissadamaid oli kolm: Cherbourg, Queenstown ja Southampton.



Joonis 4. Reisijate pääsemine teekonna alustuskoha järgi

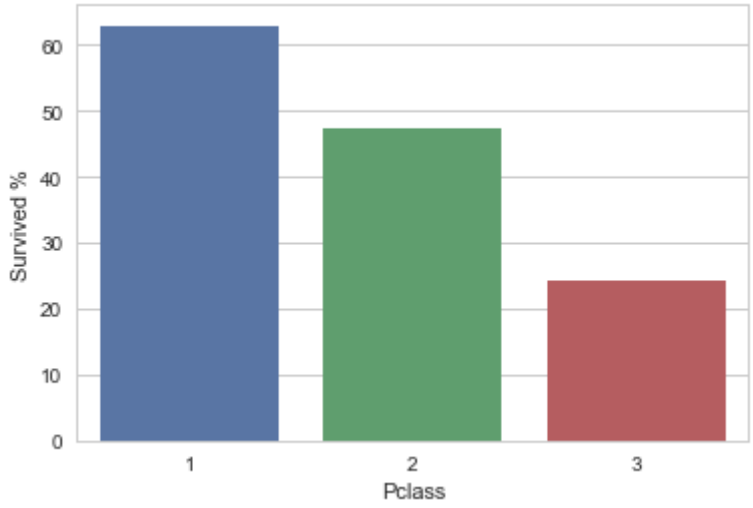
Cherbourg – pääses 55,36% reisijatest

Queenstown – pääses 38,96% reisijatest

Southampton – pääses 33,9% reisijatest

7.2 Pclass

Atribuut toob väga hästi välja, et ellujäämine sõltus üsnagi palju reisija majanduslikust ja sotsiaalsest seisusest:



Joonis 5. Reisijate pääsemine sotsiaalse klassi järgi

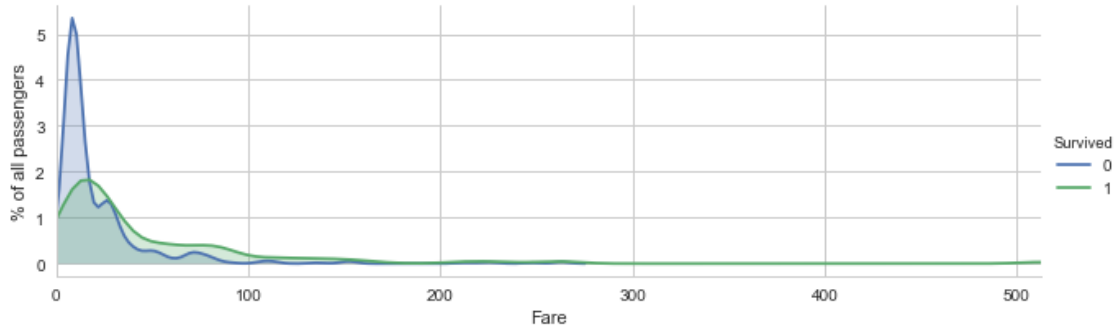
1 – ülemklass: pääses 63% reisijates

2 – keskklass: pääses 47% reisijatest

3 – alamklass: pääses 24% reisijatest

7.3 Fare

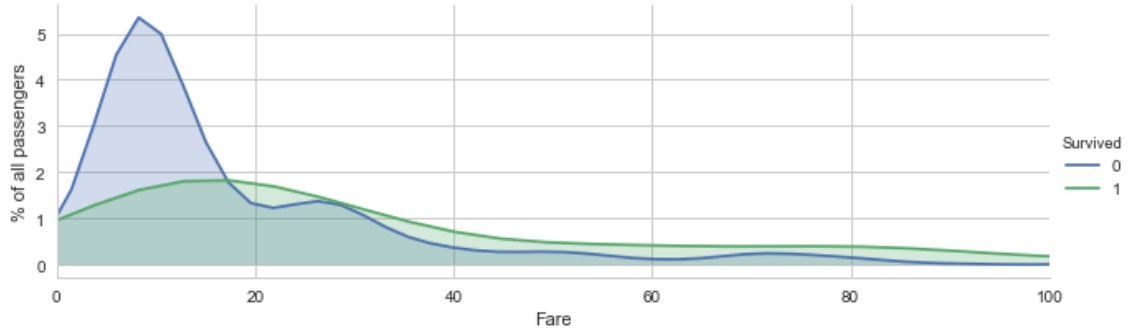
Selle jaoks, et uurida, kuidas mõjutas reisijate ellujäämist pileti hind. Eeldan ka seost Pclass atribuudiga, millest võis selgelt välja lugeda, et jõukamatel reisijatel oli suurem pääsemislootus.



Joonis 6. Reisijate pääsemine piletihinna järgi

Antud graafiku y-telg näitab, kui palju reisijaid konkreetse pileti hinnaga hukkus või pääses reisijate koguarvust protsentuaalselt. Graafikul tähistatud sinise joonega on reisijad, kes hukkusid ja rohelisega reisijad, kes pääsesid. Kuna väga kõrge piletihinnaga

reisijaid on väga vähe, koostan ka graafiku, mille maksimum piletihinnaks on 100 dollarit, et detailsemalt näidata, kuidas mõjutas piletihind reisija pääsemist.



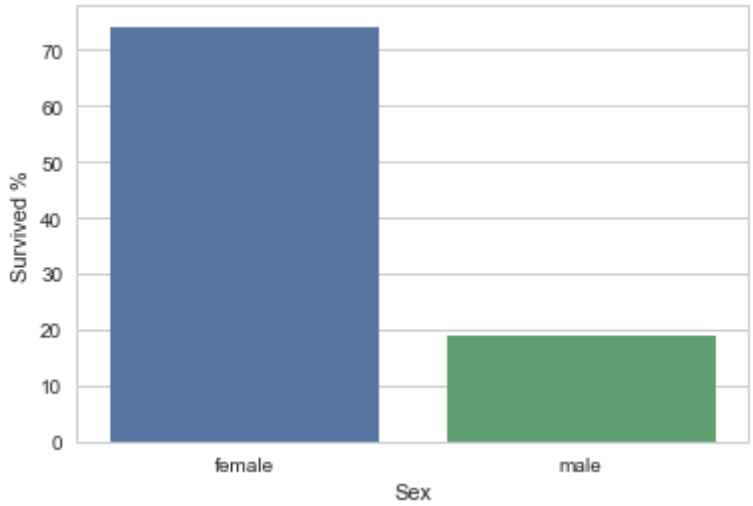
Joonis 7. Reisijate pääsemine pileti hinna järgi, max pileti hind 100 dollarit

Graafikud kinnitavad ka hüpoteesi, et kui odavama pileti hinnaga reisijatest hukkus enamus, siis reisijatest, kelle piletihind oli vähemalt 20 dollarit või suurem, pääses enamus.

Eraldi mainimist tasub ka asjaolu, et 15 reisija piletihinnaks oli „0“ dollarit.

7.4 Sex

Järgnevalt uurin reisija pääsemist soo järgi, et näha, kuidas mõjutas reisija sugu tema pääsemist.

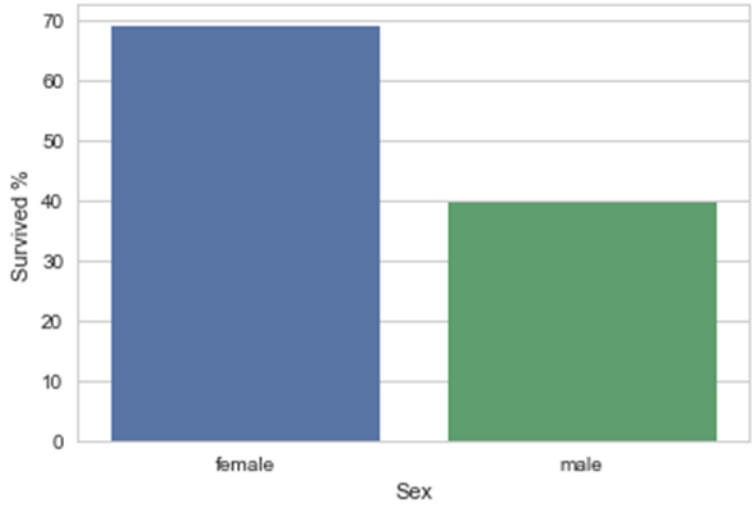


Joonis 8. Reisijate pääsemine reisija soo järgi

Joonist 8 vaadates, on väga hästi näha, et just eriti hea pääsemislootus on naistel, sest pääses 74% naistest ning kõigest 19% meestest.

Lisaks uurin ka laste ja täiskasvanute pääsemist eraldi soo järgi. Lasteks loen reisijaid vastavalt Eesti Vabariigi seadusele, mis sätestab, et laps on kuni 18 aastane inimene [7]. Täisealised on inimesed, kes on vähemalt 18 aastat vanad.

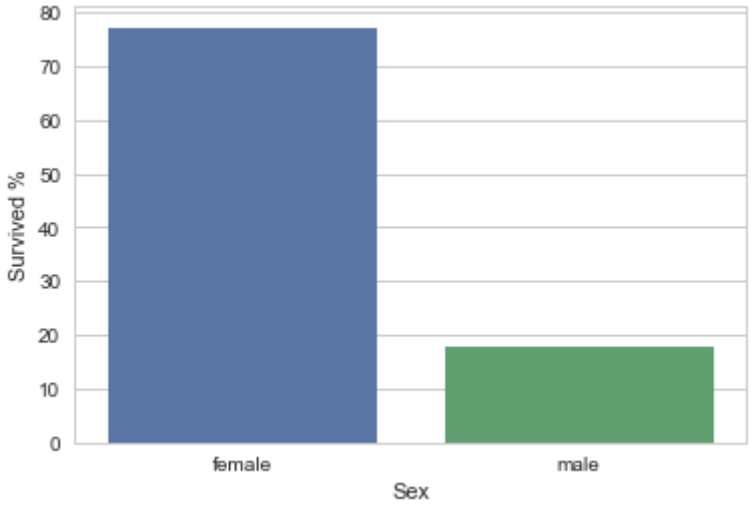
Alaealiste seas üldiselt pääses 54% reisijatest. Paremaks illustratsiooniks toon välja ka graafiku, mis näitab eraldi pääsemist alaealiste seas vastavalt reisija soole.



Joonis 9. Alaealiste reisijate pääsemine reisija soo järgi

Pääses 69,1% alaealistest neidudest ja 39,7% noormeestest. Kui võrrelda antud graafikut joonisega 8, kus on lihtsalt arvestatud reisija sugu, on näha, et alaealiste reisijate seas mängib reisija sugu väiksemat rolli, kuigi endiselt on parem pääsemise lootus naistel.

Täisealistest reisijatest pääses 38,1%, mis näitab ka seda, et parem pääsemislootus oli alaealisel reisijal. Ka täisealiste reisijate puhul toon paremaks illustratsiooniks välja graafiku, mis näitab täisealiste reisijate pääsemist vastavalt reisija soole.

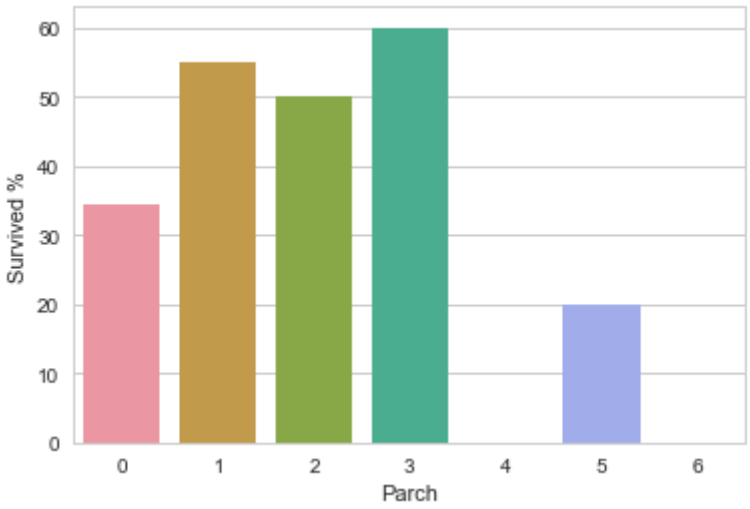


Joonis 10. Täisealiste reisijate pääsemine reisija soo järgi

Täisealiste reisijate seas oli sool väga suur kaal. Nimelt pääses 77,2% naistest ja kõigest 17,7% meestest.

7.5 Parch

Atribuut näitab, kui palju on konkreetsel reisijal pardal lapsi ja vanemaid summaarselt.

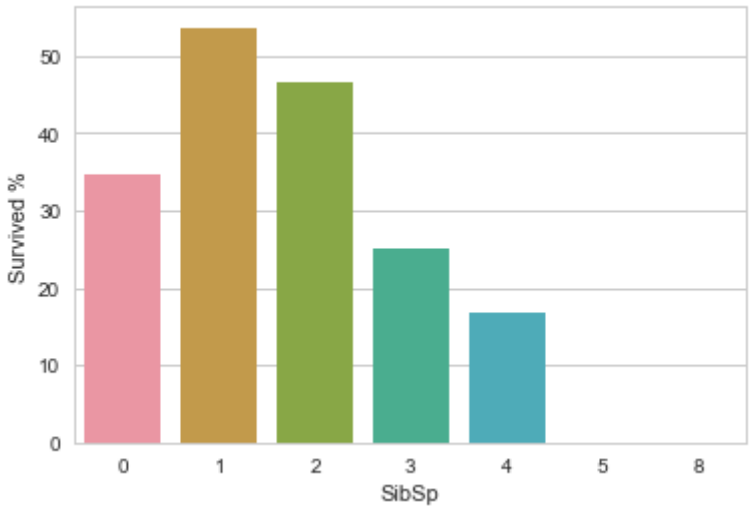


Joonis 11. Reisijate pääsemine pardal olevate laste ja vanemate summa järgi

Joonist 11 vaadates tundub, et parim pääsemislootus on reisijatel, kellel on kaasas mõni laps või vanem. Võib oletada, et see tuleb sellest, et esimesena lasti päästepaatidesse naisi ja lapsi, kellest suurem osa olid kas ise vanemad või kellel olid vanemad reisil kaasas. Antud joonist vaadates esitan ka hüpoteesi: parim pääsemislootus on väikestel peredel, liikme arvuga 1-3.

7.6 Sibsp

Atribuut näitab, kas reisijal oli abikaasa ning kui palju oli tal pardal õdesid ja vendi. Antud atribuut sobib ka Parch atribuudi all oleva hüpoteesi, et väikestel peredel on parim pääsemislootus, kontrollimiseks.

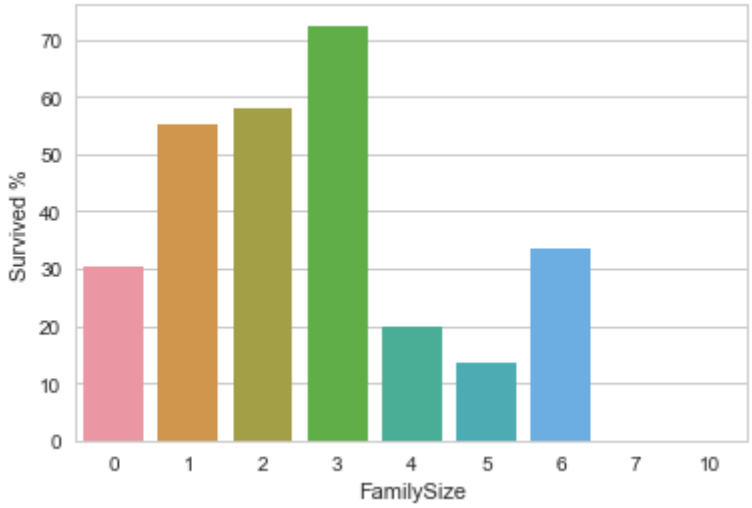


Joonis 12. Reisijate pääsemine pardal olevate õdede, vendade ja abikaasa summa järgi

Võib väita, et joonis 12 kinnitab osaliselt eelpool püstitaud hüpoteesi, sest 3 liikmelisi peresid pääses vähe, kõigest veerand neist.

7.7 FamilySize

Antud atribuut on loodud kahe eelneva atribuudi (Parch ja SibSp) summana ning peaks väikeste perede pääsemise hüpoteesi kas lõplikult kinnitama või ümber lükkama.



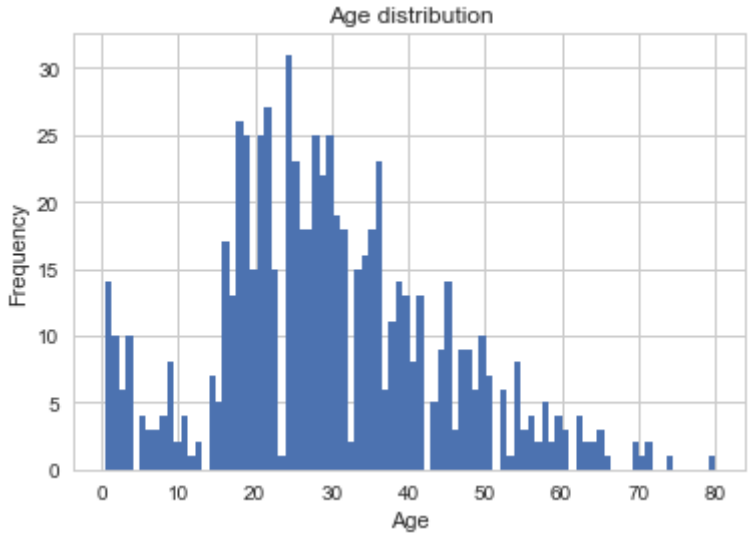
Joonis 13. Reisija pääsemine pardal oleva pere suuruse järgi

Joonis 13 kinnitab varem loodud hüpoteesi väga hästi, sest arvesse on võetud nii õdesid, vendi, vanemaid, lapsi kui ka abikaasat. Antud atribuut toob välja ka ühe fakti: nimelt üksi reisivatest inimestest pääses vaid 30%, mis on üllatav töö koostaja jaoks, sest võiks arvata, et kui reisija kellegi eest ei vastuta, on tal lihtsam enda eest seista ning pääseda.

7.8 Age

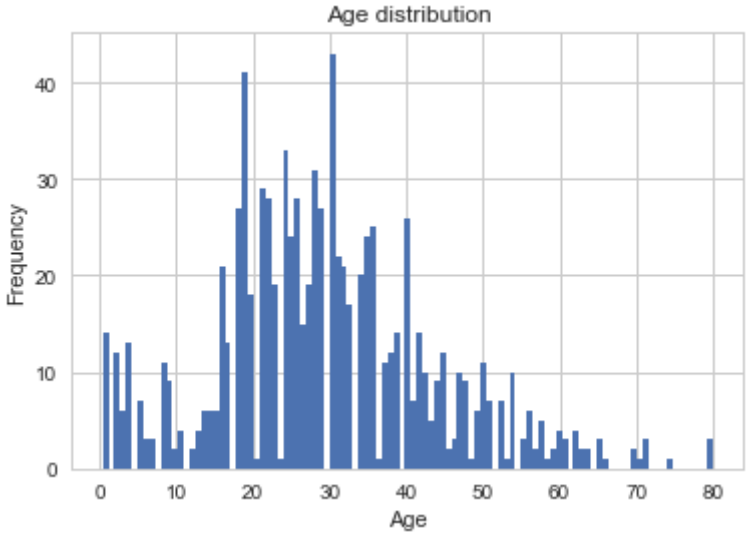
Atribuut näitab reisija vanust. Vanuse puhul toon välja graafiku enne puudu olevate vanuste ennustamist ja graafiku peale vanuste ennustamist.

Esialgsete andmete põhjal joonistatud graafik:



Joonis 14. Reisijate vanuse esialgne jaotus

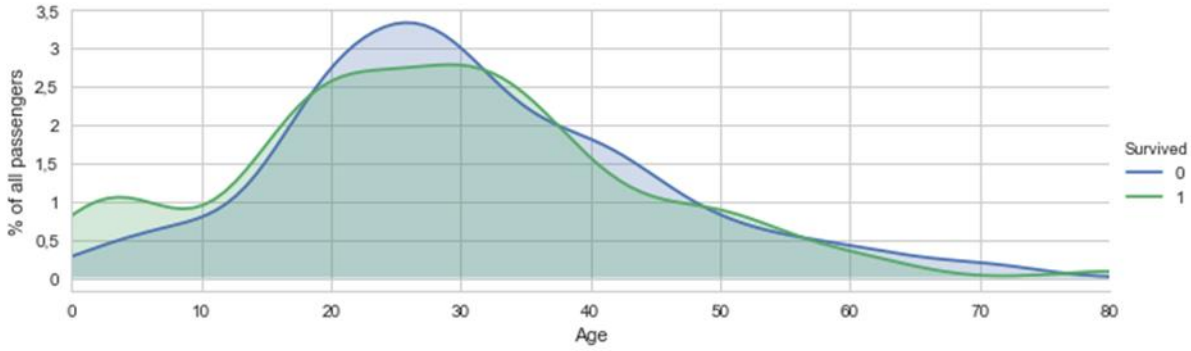
Peale puudu olevate andmete ennustamist joonistatud graafik:



Joonis 15. Reisija vanuse jaotus peale puudu olevate andmete ennustamist

Vanuste jaotuse osas on küll väike erinevus silmaga nähtav, kuid üldjoontes võib ennustatud tulemustega rahule jääda ja nende põhjal tööd edasi teha.

Reisijate pääsemist vanuse järgi uurin peale vanuse ennustamist.



Joonis 16. Reisijate pääsemine reisija vanuse järgi

Antud graafiku y-telg näitab, kui palju reisijaid konkreetses vanuses hukkus või pääses reisijate koguarvust protsentuaalselt. Graafikul tähistatud sinise joonega on reisijad, kes hukkusid ja rohelisega reisijad, kes pääsesid.

8 Modelleerimine

Antud peatükis toon välja iga loodud mudeli kohta detailsemalt info ja kirjeldan ka, millised atribuudid valisin mudelite loomiseks.

8.1 Valitud atribuudid

Pääsemist ennustavate mudelite loomiseks ei saa kasutada kõiki atribuute ning seetõttu on valitud ainult osad, enamasti samad atribuudid, mida kasutati puudu olevate vanuste ennustamiseks treenitud regressioonipuu puhul. Ainsateks erinevusteks on *Age* atribuut, mida nüüd kasutan treenimiseks ja *Survival* atribuut, mida nüüd üritan ennustada.

8.2 Mudelite parameetrid

Antud peatükis kirjeldan iga loodud mudelit veidi detailsemalt. Nimelt toon välja iga mudeliga tema loomiseks seotud parameetrid, mida antud töös on kasutatud.

8.2.1 Otsustuspuu

Antud töös pääsemise ennustamiseks loodud otsustuspuu on piiramata sügavusega klassifitseerimispuu, mis tähendab, et see suudab ennustada väärtusi ainult varem nähtud väärtuste seas. See omakorda tähendab, et ennustatava atribuudi, antud juhul *Survived*, kõik võimalikud väärtused peavad olema esindatud treeningandmete seas.

8.2.2 Lähima naabri meetodid

Lähima nääbri meetodit on kasutatud antud töös kaks korda ning mõlemal korral on meetodit piiratud lähimate naabrite arvuga. Ühe mudeli lähimate naabrite arvuks on seatud kolm naabrit ja teisel viis naabrit. Raadiusel põhinevat lähima naabri meetodit kasutatud ei ole.

8.2.3 Tehisnärvivõrk

Tehisnärvivõrgul on parameetritena kasutatud vaikimisi väärtusi. Selle peamiseks põhjuseks on mudeli suur keerukus ja töö koostaja vähene kogemus antud mudeliga.

8.2.4 Naiivse Bayesi mudel

Antud mudelile parameetreid ette antud ei ole, sest mudel töötab väga konkreetse matemaatilise valemi järgi. Töösse ongi antud meetod valitud tema lihtsakoelisuse tõttu ning töö koostaja huvist konkreetset meetodit ise kasutada. Nimelt on kirjanduses antud meetod tihti välja toodud kui üsna heade tulemustega meetod, mis tundub veidi uskumatu, vaadates teiste meetodite keerukust.

8.2.5 Juhusliku metsa mudel

Antud mudelile on koostaja poolt seatud piiranguks igale puule, millest mets koosneb, maksimum sügavus, milleks on valitud viis. Antud piirang on seatud seetõttu, et vältida ülesobitamise probleemi, mis tekkis, kui lasta puudel lõpmata sügavaks minna.

8.2.6 Kohanemisvõimeline tõukealgoritm

Antud mudelile töö koostaja poolt seatud piiranguid pole, küll aga on vaikimisi seatud piirang, et üldine ennustus tehakse viiekümne nõrga mudeli ennustuste keskmisena.

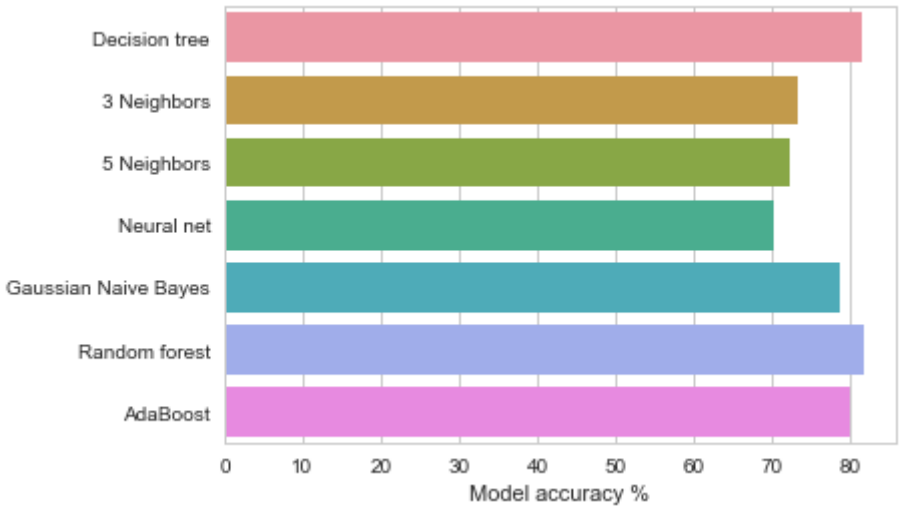
9 Mudelite täpsuse hindamine

Mudelite täpsuse hindamiseks kasutatakse antud tööd kümne kordset rist-valideerimist. See tähendab, et mudel treenitakse osade andmete peal ja tema täpsust hinnatakse „uute“ andmete peal, mida mudeli treenimiseks kasutatud ei ole. Seejärel arvutatakse tema täpsus kasutades järgmist valemit:

$$Täpsus = \frac{\sum_{i=1}^{n} x_i = y_i}{n} * 100\%$$

Kus x on ennustatud väärtus, y on õige väärtus ja n on ennustatavate väärtuste kogus. Seda protsessi sooritatakse iga mudeli korral kümme korda ning mudeli üldine täpsus leitakse täpsuste aritmeetilise keskmisega.

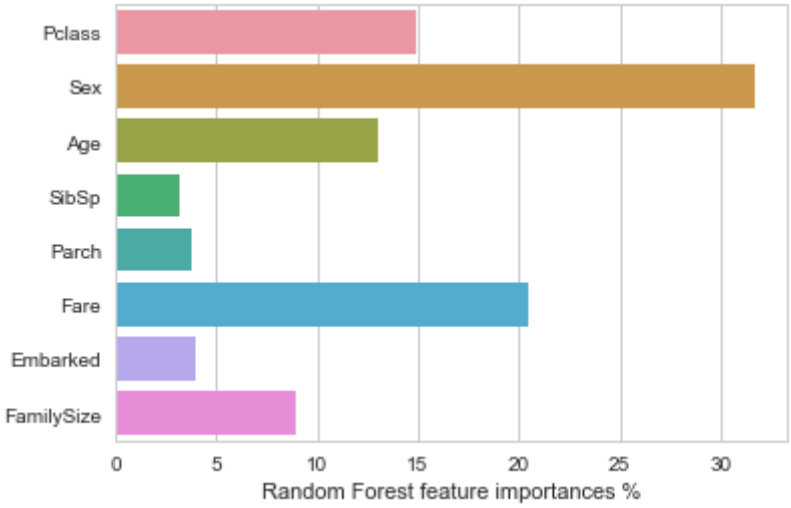
Antud töös kasutatakse kümne kordseks rist-valideerimiseks scikit learni poolt pakutavat funktsiooni cross_val_score(), millele antakse ette parameetritena klassifitseerija, andmed ilma sihtmuutujata, sihtmuutuja ning täisarv, mitu korda protsessi korratakse, antud juhul kümme. Funktsioon tagastab mudeli täpsused massiivina (Array), mille väärtused jäävad vahemikku [0:1] [28]. Peale massiivi tagastamist, leitakse nende aritmeetiline keskmine, mis korrutatakse 100-ga, et esitada neid paremini protsentuaalselt.



Joonis 17. Ennustusmudelite täpsused

Joonisel on kuvatud erinevate ennustusmudelite täpsused ning üsna selgelt on näha neli paremat. Kõige täpsemini suutis ennustada juhusliku metsa algoritm, mille täpsus oli

81,72% ja kõige kehvemini suutis reisijate pääsemist või hukkumist ennustada tehisnärvivõrk. Tehisnärvivõrgu kehva ennustuse põhjuseks võib osaliselt olla ka töö koostaja vähene varasem kokkupuude närvivõrkudega.



Joonis 18. Juhusliku metsa algoritmi abil loodud mudeli atribuutide tähtsused

Joonisel on välja toodud juhusliku metsa algoritmiga loodud mudeli erinevate atribuutide tähtsused reisijate pääsemise ennustamisel. Nagu näha, on väga ülekaalukalt tähtsaim reisija sugu ning seejärel tulevad pileti hind ning reisija sotsiaalset ning majanduselikku klassi näitav atribuut, mis on omavahel seotud. Väga hea on näha, et ise loodud muutuja, perekonna suurus pardal, on samuti esindatud.

Mudelite täpsust oleks tõenäoliselt saanud parandada kasutades kombinatsiooni nii antud töös, kui ka teise analüütiku poolt tehtud tööga. Nimelt usun ma, et veidi oleks täpsust lisanud reisijate nimedest nende tiitlite eraldamine ja kajutite info parem ära kasutamine. Veel oleks saanud erinevaid atribuute luua kasutades teadmisi pardal olevate pereliikmete kohta saada olevaid andmeid, näiteks uurides, kas vanematel või lastel, kellel oli kaasas mõni vanem, on parem lootus pääseda ja seeläbi veel parandada mudelite täpsust.

Kokkuvõte

Titanicu uppumine ja sellega kaasnenud hukkunute arv oli üks suurimaid omalaadseid ning sellest õnnetusest on võimalik väga palju õppida.

Antud töös loodud analüüs annab üsna hea ülevaate sellest, et parimad pääsemislootused olid kahe kuni nelja liikmeliste perede naistel, mis on tegelikult ka arusaadav, sest nemad olid esimesed, kes päästeti.

Pääsemist ennustavatest mudelitest täpseimateks osutusid otsustuspuid kasutavad meetodid: juhusliku metsa algoritm ja piiramatu sügavusega otsustuspuu.

Antud tööga oleks võimalik minna veel rohkemate detailsusteni, mida antud juhul piiras veidi ette antud töö maht ja koostaja kogenematus andmekaeve vallas. Ka lühike võrdlus teise analüütiku poolt koostatud tööga tõi välja, et mõlemal tööl on omad tugevused ja nõrkused ning parim tulemus oleks võimalik saavutada neid kombineerides.

Lõpuks võib öelda, et jäin ülesande valikuga rahule ja leidsin andmekaeve näol uue valdkonna, millel on väga palju potentsiaali ja millega kavatsen ka edaspidi tegelemist jätkata.

Summary

The sinking of Titanic and the amount of casualties are one of the biggest disasters of their kind and there is much to be learned from it.

The analysis in this research shows that women in families with two to four members had the best chance of survival which is self-explanatory because they were he first to be evacuated.

The most accurate predictive models were the ones that used decision trees to make predictions: random forest algorythm and decision tree with unlimited depth.

This research could go into more detail but in this case it was limited by the given volume and the researchers inexperience in data mining. The brief comparison with another data miners research also brought out that both researches have theirs strengths and weaknesses and that the best result could be achieved by combining them.

In conclusion I can say that I am satisfied with the given subject and found a new field in the face of data mining that has a lot of potential which I am going to keep working on.

Kasutatud kirjandus

1. Ian H. Witten, Eibe Frank, „Data mining: Practical machine learning tools and techniques“, 2nd edition, San Francisco: Morgan Kaufmann Publishers, 2005
2. H. Vallaste, „CSV (Comma Separated Values)“ http://vallaste.ee/index.htm?Type=UserId&otsing=6107 [01.04.2017]
3. J. Schneider „Cross Validation“ https://www.cs.cmu.edu/~schneide/tut5/node42.html [01.04.2017]
4. Scikit learn „Decision Trees“ http://scikit-learn.org/stable/modules/tree.html [01.04.2017]
5. K. Rootalu „Kirjeldav statistika“ http://samm.ut.ee/kirjeldav-statistika [01.04.2017]
6. Scikit learn „Nearest Neighbors“ http://scikit-learn.org/stable/modules/neighbors.html [01.04.2017]
7. Eesti Vabariigi põhiseadus https://www.riigiteataja.ee/akt/12850781 [19.04.2017]
8. Tanel Kaart „Matemaatilise statistika koolitus“ http://www.eau.ee/~ktanel/mmstatistika_koolitus_EMYs_2016/MMSkoolitus%20-%20loeng5.pdf [25.04.2017]
9. Scikit learn „Tree Classification“ http://scikit-learn.org/stable/modules/tree.html#tree-classification [19.04.2017]
10. Scikit learn „Tree Regression“ http://scikit-learn.org/stable/modules/tree.html#tree-regression [19.04.2017]
11. „Cluster Analysis“ http://www.statsoft.com/Textbook/Cluster-Analysis#d [19.04.2017]
12. „Conda docs“ https://conda.io/docs/ [03.05.2017]
13. „Disadvantages of continuum analytics anaconda/conda“ https://www.reddit.com/r/Python/comments/4uj41m/disadvantages_of_continuum_analytics_anacondaconda/ [03.05.2017]
14. „Mean absolute error“ https://www.kaggle.com/wiki/MeanAbsoluteError [03.05.2017]

15. „Sklearn mean_absolute_error()“ http://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html [03.05.2017]
16. C. Stergiou ja Dimitrios Siganos „Neural Networks“ https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#What is a Neural Network [11.05.2017]
17. Scikit learn „Neural network models (supervised)“ http://scikit-learn.org/stable/modules/neural_networks_supervised.html#multi-layer-perceptron [11.05.2017]
18. Scikit learn „Naive Bayes“ http://scikit-learn.org/stable/modules/naive_bayes.html#naive-bayes [11.05.2017]
19. Andy Liaw ja Matthew Wiener „Classification and Regression by randomForest“ http://ai2-s2-pdfs.s3.amazonaws.com/6e63/3b41d93051375ef9135102d54fa097dc8cf8.pdf [11.05.2017]
20. Leo Breiman ja Adele Cutler „Random Forests“ https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#overview [11.05.2017]
21. Scikit lean „AdaBoost“ http://scikit-learn.org/stable/modules/ensemble.html#adaboost [12.05.2017]
22. Jason Brownlee „Boostin and AdaBoost for Machine Learning“ http://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/ [12.05.2107]
23. Dragos D. Margincantu ja Thomas G. Dietterich „Pruning Adaptive Boosting“ https://pdfs.semanticscholar.org/b25f/615fc139fbdeccc3bcf4462f908d7f8e37f9.pdf [12.05.2017]
24. Jason Brownlee „Overfitting and Underfitting With Machine Learning Algorithms“ http://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/ [12.05.2017]
25. Rüdiger Wirth ja Jochen Hipp „CRISP-DM: Towards a Standard Process Model for Data Mining“ http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.198.5133&rep=rep1&type=pdf [12.05.2017]
26. Wikipeedia „Kaggle“ https://en.wikipedia.org/wiki/Kaggle [13.05.2017]

27. Scikit learn „Model Evaluation“ http://scikit-learn.org/stable/modules/model_evaluation.html#mean-absolute-error [13.05.2017]
28. Scikit learn „cross_val_score“ http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html#sklearn-model-selection-cross-val-score [13.05.2017]
29. Wikipedia „Titanic“ https://en.wikipedia.org/wiki/RMS_Titanic [16.05.2017]
30. Kaggle „Titanic: Machine Learning from Disaster“ https://www.kaggle.com/c/titanic/data [18.05.2017]
31. Xiaojun Yu „Predictive Analysis of Survival Rate on Titanic“ https://www.kaggle.com/tony9090/predictive-analysis-of-survival-rate-on-titanic/notebook [18.05.2017]