Big data aroko metodo estatistikoak
Hoteleko prezioen bistaratzea

2019

Organismo Autónomo del

Eustat
EUSKAL ESTATISTIKA ERAKUNDEA
INSTITUTO VASCO DE ESTADÍSTICA
www.eustat.es

EUSKO JAURLARITZA
GOBIERNO VASCO

Lanketa:
EUSTAT
Euskal Estatistika Erakundea
Instituto Vasco de Estadística

Argitalpena:
EUSTAT
Euskal Estatistila Erakundea
Instituto Vasco de Estadística
Donostia-San Sebastián 1,
01010 Vitoria-Gasteiz

Euskal AEko Administrazioa

1. argitalpena:
I/2019

Inprimaketa eta Koadernatzea:
Eusko Jaurlaritzako Inprimategia eta Erreprografia Zerbitzua

ISBN: 978-84-7749-489-8

Lege-Gordailua: VI-154/19

BIG DATA GARAIKO Metodo estadistikoak

Hoteleko prezioen bistaratzea

Ander Juarez Mugarza

EUSKAL ESTATISTIKA ERAKUNDEA

INSTITUTO VASCO DE ESTADISTICA

Donostia-San Sebastián, 1

01010 VITORIA-GASTEIZ

Tel.: 945 01 75 00

Fax.: 945 01 75 01

E-mail: eustat@eustat.eus

www.eustat.eus

Aurkezpena

Azken garaiotan big data gizartearen arlo guztietara zabaldu da. Halaber, estatistika ofizialean, non erronka bat den bere erabilera datu iturri berri bat bezala. Estatistikako institutuetan hainbat proiektu pilotu ari dira egiten big dataren ondorioei heltzeko produkzio estatistikoaren alderdi guztietan.

Eustatek 2016an antolatu zuen Nazioarteko Estatistika Mintegiaren XXIX. edizioa “Big data for Official Statistics” izenaren pean, zeina Peter Struijsek eman baitzuen, Big Dataren programako koordinatzailea Statistics Netherlands (SN) izenekoan eta Big Dataren taldearen koordinatzailea Europar Batasuneko ESSnet (European Statistical System network) izenekoan.

Argitalpen honek ezagutarazi nahi du arlo honetan eginiko ikerketa lana Eustaten formakuntzako eta ikerkuntzako beketariko baten eskutik. Dokumentu honek bi atal ditu. Lehenbizikoan errepaso bat egiten zaie ikaskuntza automatikoko metodo batzuei big datan erabilgarriak eta bigarrenean aztertzen dira Euskal AEko hotel establezimenduetako prezioen eboluzioa, weba eskrapeatuz lortuak, prezioen korrelazio espaziala eta horien bistaratzea bero mapen bidez.

Vitoria-Gasteizen, 2019ko martxoan

Josu Iradi Arrieta

EUSTATeko Zuzendari Nagusia

Indizea

Atarikoa

Euskal Estatistika Erakundeak (Eustat) 2017. urtean estatistika eta matematika metodologietan prestatzeko eta ikertzeko emandako bekari esker Machine Learning inguruan egin den lanaren emaitza da Koaderno Tekniko honetan bildutakoa.

Liburu honek bi helburu nagusi ditu: Gaur egungo *machine learning* munduan erabiltzen diren teknika nagusien azalpen bat ematea eta azken bi urteetan zehar Eustaten, Euskal Estatistika Erakundean, egindako Big Data proiektuaren emaitzak azaltzea. Hori dela eta koadernoa bi atal nagusietan bananduko da non atal bakoitza lau kapituluz osatuta dagoen.

Lehenengo atalari dagokionez, alde teorikoz osatuta egongo da metodo desberdinen adibideak emanez. Esan bezala lau kapituluz osatuta egongo da. Lehenengo bi kapituluetan *machine learning* munduko bi familia nagusiei buruz, *supervised learning* eta *unsupervised learning*, hitz egingo da. Ondoren, hirugarren kapituluan azken urteetan fama handia hartzen ari diren *sare neuronalei* buruz hitz egingo da. Azkenik, laugarren kapituluan erabiltzen diren algoritmoetarako normalean egiten diren datuen *test-balioztatze-entrenamendu* banaketei buruz hitz egingo da.

Bigarren atalari dagokionez, aldiz, lan prozesuan jarraitutako pausuak eta hauei dagokien alde teorikoa ikusiko da. Bigarren ataleko proiektua Euskal Autonomi Erkidegoko hotelen prezioaren urtean zeharreko eboluzioa aztertzen du. Azterketa hau egin ahal izateko, erkidegoko hotelen prezioak web plataformoetatik lortu ziren, Eustat erakundeak garatutako web scraping prozesu baten bitartez. Datu hauek Eustateko turismo direktorioaren *Establezimendu Turistiko Hartzaileen Inkestako* datuekin fusionatu ziren datu base osoago bat lortuz.

Lortutako datu horiekin EAE-ko ostatuen denboran zeharreko prezioen eboluzioa eta hauek beraien ingurunearekin duten erlazioa modu intuitibo batean erakusten duen aplikazio bat garatu da. Aplikazio hau ESRA-k, *European Survey Research Association*-ek, 2018ko azaroan Bartzelonan antolatu zuen *BigSurv18* konferentzian aurkeztu zen.

Halaber, atariko hau baliatu nahi dut Eustateko Metodologia, Berrikuntza eta I+Gko Arloa osatzen duten guztiei, Anjeles Iztueta, Jorge Aramendi, Elena Goni, Inmaculada Gil eta Marina Ayestarán, igaro diren bi urteetan emandako babes eta konfiantza eskertzeko. Modu berean, Eustateko familia osatzen duten langileei eta Asier Badiola bekadunari eskerrak eman nahi dizkiet sortutako lan giro onagatik. Bukatzeko, eskerrak nire familiari, Garaziri eta modu berezi batean nire aitite Alejandro Mugarzari, beti nire alboan egoteagatik.

Machine learning teknikak

Liburuaren lehengo atal honetan gaur egungo *Machine Learning* tekniken errepaso bat egingo da. *Machine Learning* kontzeptu berri bat badirudi ere bere izena 1959-tik dator. Garai artako ikertzaileek datuetatik ikasi eta aurreikuspenak egiteko gai diren algoritmoak garatzea interesgarria zela pentsatu zuten, gaur egungo *Machine Learning* teknikak guztiak garatzeko beharrezkoak izan diren lehenengo pausuak emanez.

Esan bezala *Machine Learning* prozesuaren helburua datuetatik ikastea da eta, hau eginda, etorkizunean aurreikuspenak egitea. Hemen bi familia nagusi bereizi ahalko lirateke: *Supervised Learning* eta *Unsupervised Learning*. Lehenengoaren helburua, *Supervised Learning*, iraganeko datuetatik abiatuz etorkizuneko datuak aurresatea litzateke. Bigarrenarena, aldiz, datuetatik ikastea da, batzuetan ezkutuak dauden edo begi bistaz ikusi ezin diren informazioa eta aztertutako elementuen arteko erlazioak ateraz. Bi familia hauei buruz lehenengo bi kapituluetan hitz egingo da.

Hirugarren kapituluari dagokionez, gaur egun indarra handia hartu duten *Sare Neuronalei* hitz egingo da. *Machine Learning* terminoari gertatzen zaion moduan, *Sare Neuronalak* gaur egungo gauza badirudite ere duela denbora bat definitutako metodoak dira, 1943 urtean alegia. *Sare Neuronalek* hainbat eginkizunetarako balio dutenez, bai *Supervised Learningerako* baita *Unsupervised Learningerako* ere, kapitulu propio bat merezi zuela erabaki da.

Bukatzeko, *Supervised Learning* motako tekniketan (mota honetako problemetarako erabiltzen diren *Sare Neuronalak* barne) aurreikuspenak egin behar diren heinean hauek ere testatua behar dira. Hori dela eta, atal honetako azken kapituluan eskuragarri dauden datuak, algoritmoak garatzeko eta test desberdinak egiteko, banatzeko existitzen diren teknika desberdinei buruz hitz egingo da.

1. Supervised learning

Mota honetako teknikak aztertzen den elementu bakoitzeko irteera balio edo etiketa bat aurresan nahi dugunean aplikatzen dira, hau da, hasierako datu batzuetatik abiatuz irteerako balio batzuk inferitu nahi direnean. Irteera datuak bai kuantitatiboak, *erregresioa*, baita kualitatiboak, *klasifikazioa*, izan daitezke. Helburu nagusia ez da inoiz datuetatik ikastea izango, baizik eta sarrerako balioak irteerako balioekin ondo erlazionatzen dituen erregela bat lortzea. Horrela, etorkizunean datu berriak lortzerakoan, hauei dagokien irteera balioa inferitu ahal izango dira.

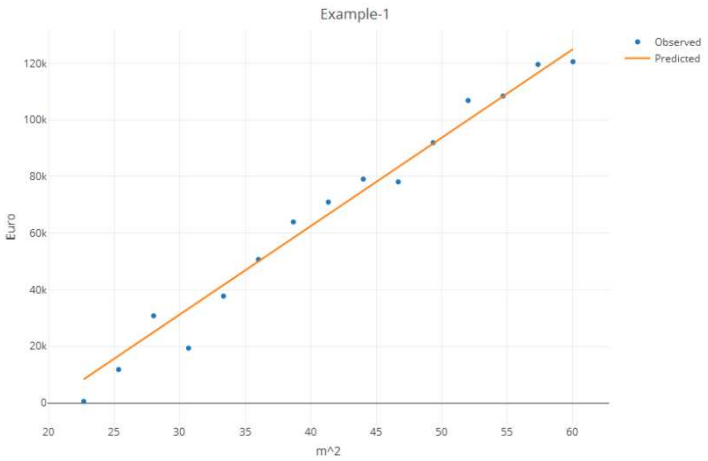
Supervised learning metodoen artean ondokoak aztertuko dira:

1. Erregresio lineala
2. Erregresio logistikoa
3. KNN
4. Perceptron
5. SVM
6. Decision trees

1.1. Erregresio lineala

Erregresio linealak datu baseko $X_1, \dots, X_n$ elementu bakoitzari dagokion $Y_1, .., Y_n$ etiketa kuantitatiboa kontuan hartuz, X' elementu berri bat lortzerakoan, honi dagokion Y'etiketa kuantitatiboa aurresaten saiatzen da.

Demagun etxe desberdinen metro karratuak eta prezioak dakizkigula eta hauen erlazioaren grafikoa egiten dugula, eskumako grafikoaren puntu urdinak lortuz. Ikus daitekeen moduan, lortutako puntuek metro karratuen eta prezioaren arteko erlazioa lineala dela suposa daiteke. Orduan, etxe baten metro karratuak jakinda honen prezioa aurresatea interesgarria litzateke. Erregresio linealak, funtzio lineal baten bitartez, X_i datuen elementu bakoitza dagokion Y etiketarekin egokitzen saiatzen da (lerro laranja).



1.1.1. Algoritmoa

Y' etiketa berriak aurresateko, teknikaren izenetik ondorioztatu daitekeen moduan, $f(X, \theta)$ funtzio *lineala* lortzea da helburua, non $\theta = \{\theta_0, \dots, \theta_p\}$ funtzio linealaren koefizienteak diren eta

$$Y \sim f(X,\theta) \; = \; \theta_0 + \sum_{i=1}^{p} \theta_i X + \epsilon$$

den, ϵ ia beti egongo den errorea izanik.

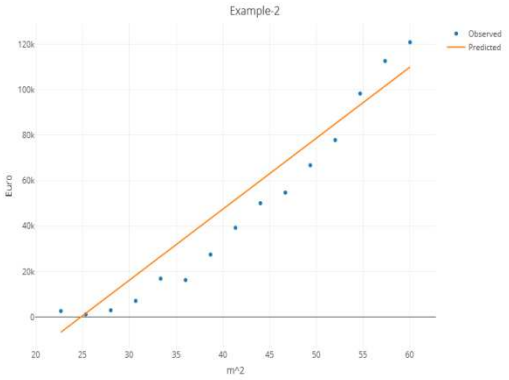
Erregresio linealaren θ parametro optimoak errore minimoa ematen dutenak izango dira. Errorea neurtzeko garaian, normalena $f(X,\theta)$ aurresandako balioaren eta elementuaren Y etiketaren arteko aldea neurtzen da, horretarako distantzia euklidearra erabiliz. Beraz, minimizatu beharreko balio funtzioa ondokoa litzateke:

$Cost(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(f(X_i,\theta) - Y_i)^2$.

Funtzio hau minimizatzeko, *gradient descent* algoritmoa erabili daiteke. Algoritmoa aplikatzerakoan, ondoko pausua behin eta berriz egingo litzateke konbergentzia lortu arte edota aurretik finkatutako iterazio kopuru maximoa igaro arte:

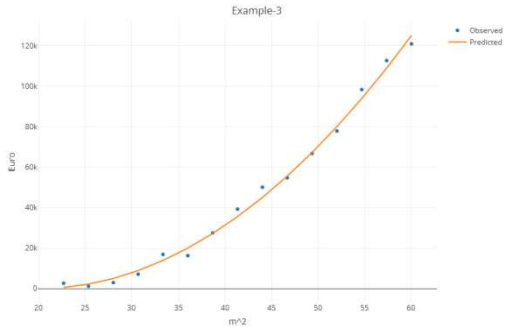
$\theta = \theta - \frac{\alpha}{2n}\sum_{i=1}^{n}\left(X_i^T\theta - Y_i\right)X_i$, $1 \le i \le n$ bakoitzeko.

1.1.2. Erlazio ez linealeko aldagaiak



Batzuetan gerta daiteke aldagaien arteko erlazioa lineala ez izatea, eskumako grafikoan ikus daitekeen moduan. Kasu hauetan ere erregresio lineala aplikatzea posible izango litzateke.

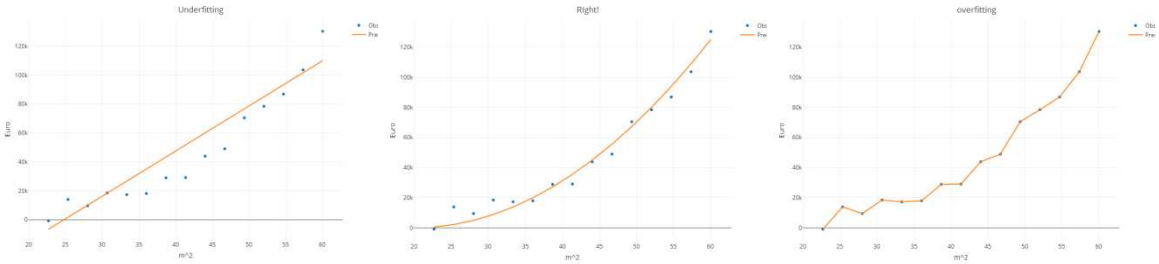
Demagun gure datuek p aldagai desberdin dituztela, orduan aldagai hauen potentziak aldagai berri moduan erabili daitezke erregresio lineala datuei hobeto egokitzeko. Erregresioa linealean θ_i koefizienteak linealak izan behar diren arren eskura dauden datuak egokitu daitezke emaitza hobetzen badu. Adibidez, aurreko kasuan aldagaien lehenengo lau potentzia aldagai moduan gehituz, ezkerrean agertzen den emaitza lortuko genuke. Begi-bistakoa den moduan, hobekuntza nabaria lortzen da. Aldagaiei potentzia desberdinak aplikatzeaz aparte beste hainbat funtzio aplikatu ahal zaizkie, adibidez, logaritmo funtzioa, erro karratua, esponentziala eta abar.



1.1.3. Erregularizazioa

Ereduari aldagai berriak gehitzerako garaian lortutako ereduak gure entrenamendu edo jatorrizko datuak "ikasi" ditzake, etorkizuneko kasu berrietan errore handiagoa emanez. Fenomeno horri *overfitting* deritzo eta saihetsi beharreko gauza bat da. Azpiko argazkian puntako bi kasuak ikus daitezke, bai overfitting bai underfitting kasuak. Bigarren hau dugun eredua entrenamenduko puntuen izaera ondo islatzen ez duenean gertatzen da.

Underfitting **Good aproximation** **Overfitting**

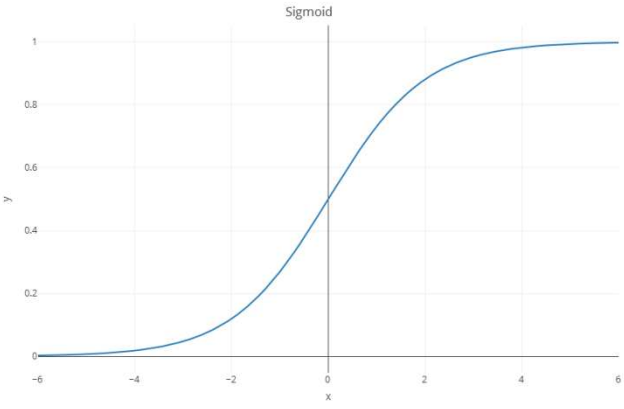


Overfitting-a saihesteko egin daitekeen gauza bat aldagaiak kentzea da. Hortaz aparte, erregularizazio parametro deituriko λ konstantea balio funtzioari gehitu ahal zaio θ parametroen penalizazio modura, ondokoa lortuz.

$Cost(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(f(X_i,\theta) - Y_i)^2 + \lambda\sum_{j=1}^{p}\theta_j$ (ohartu θ_0 penalizaziorik ez duela).

Parametro honek balio funtzioaren balioa handitu egiten du θ_i parametroen balioa handitu ahala. Parametro honen balioa handitu ahala, balio funtzioa minimizatzerako orduan θ_i $(i \neq 0\ izanda)$ elementuen balioa zerorantz joango da eta erregresioak erantzun konstante bat emango du, θ_0. Parametroaren balioa txikitzerakoan, aldiz, lortutako eredua erregularizaziorik gabeko ereduaren antzekoa izango da. Azkenik, $\lambda = 0$ kasuan erregularizazio gabeko ereduaren berdina izanda

1.2. Erregresio logistikoa



Erregresio logistikoa klasifikaziorako tekniken artean erabilienetakoa dela esan daiteke. Klasifikaziorako teknika guztien antzera, erregresio logistikoan ditugun $\{x_1, \dots, x_n\}$ datu bakoitzeko $\{y_1, \dots, y_n\}$ etiketa izango ditugu. Etiketa hauek bitarrak izango dira, hau da, bi aukera izango dituzte (bai/ez, bizi/hil…). Metodoaren helburua, x' elementu berri bakoitzari dagokion y' etiketa aurresatea da. Kasu honetan, estimatu nahi den etiketa, edota etiketa positiboa, 1 moduan ikusiko dugu eta bestea 0 moduan.

Erregresio logistikoa, *funtzio logistikoa* (*sigmoid function*) deituriko funtzioa erabiliz, elementu desberdinen aldagaien eta hauen etiketen arteko erlazioa bilatzen saiatzen da.Erlazio hau jakinda, etiketa desberdineko datuak linealki banantzen ditu. Metodo hau, orokortutako erregresio linealaren kasu berezi moduan ikus daiteke.

1.2.1. Algoritmoa

Demagun datuak X multzoan ditugula. Orduan algoritmo honen helburua θ parametroa, p tamainakoa, estimatzea da non $\sigma(x'^T\theta)$ funtzioak x' elementuak 1 etiketa izateko duen probabilitatea adierazten duen,

$$\sigma(z) = \frac{1}{1+e^{-z}}$$ *sigmoid* funtzioa izanda.

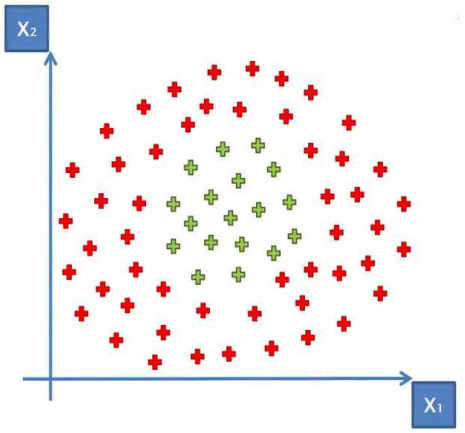
Erregresio logistikorako parametro optimoak lortzeko, ondoko kostu funtzioa minimizatu behar da:

$$Cost(\theta) = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\sigma(x_i^T\theta) + (1-y_i)\log(1-\sigma(x_i^T\theta))) .$$

Funtzio hau minimizatzeko, *gradient descent* erabili daiteke baina baita beste teknika batzuk, adibidez: *conjugate gradient, BFGS, L-BFGS...*

1.2.2. Linealak ez diren mugak

Orain arte ikusitakoarekin, muga linealak dituzten elementu ezberdinak banandu eta aurresateko ahalmena badugu ere, normalean datuak ez dira horrelakoak izango. Eguneroko lanean, ohikoena, linealki banangarriak ez diren mugak aurkitzea da, eskumako adibidean ikus daitekeen moduan. Kasu hauetan, erregresio linealean ikusi dugun moduan, aztertzen diren aldagaien potentziak aldagai berri moduan datuei esleitzea pena mereziko luke. Hau eginda, gure metodoa datuetara egokitu daiteke. Ondoko bideoak datuak 5. mailako potentziari igota gradient descent algoritmoaren pausuen eboluzioa erakusten du.



1.2.3. Erregularizazioa

Hainbat supervised learning-eko metodoekin gertatzen den moduan, erregresio logistikoan ere eredua egiteko erabili ditugun datuak "ikastea" eta hauekiko desberdina den datu berri bat sartzerakoan emaitza txarra ematea gerta daiteke. "*Overfiting*" fenomeno hau ekiditeko erregularizazio parametro bat ereduari gehitzea gomendagarria da. Hau kontuan hartuz, ondoko balio funtzioarekin lan egingo genuke:

$$Cost(\theta) = -\frac{1}{n}\sum_{i=1}^{n} y_i \log\left(\sigma(x_i^T\theta)\right) + (1-y_i)\log\left(1-\sigma(x_i^T\theta)\right)) + \frac{\lambda}{2n}\sum_{j=1}^{p}\theta_j.$$

Parametro honen eraginez eredua egiterakoan errorea handitzen bada ere, datu berrien etiketak aurresateko erabiltzerakoan lortutako emaitzak hobetzen dira. Ondoko bideoan ikus daitekeen moduan, hasieran oso irregularra den muga leuntzen badoa ere λ-ren balioa igo ahala, puntu batetik aurrera eredua sinplifikatzen doa emaitza txarrak emanez. Beraz, kontu handia eduki behar da λ parametroaren balioa aukeratzerakoan.

1.2.4. Klase anitzen klasifikazioa

Bi etiketa desberdin dituzten datuekin lan egin badugu ere, baliteke 3 edo etiketa gehiago dituzten elementuekin lan egitea nahi izatea. Kasu hauetan klase anitzen klasifikazio baten aurrean egongo ginateke.

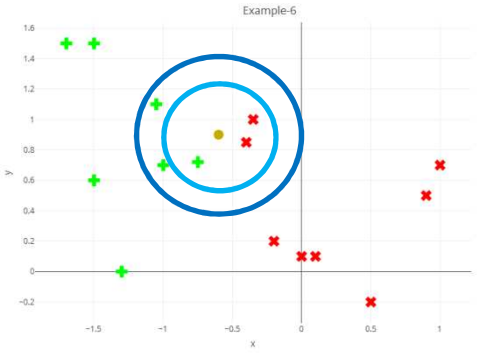
Klase anitzen klasifikazio problema ebazteko aukeretatik intuitiboena *One vs All* deiturikoa litzateke. Demagun k etiketa desberdin daudela, hau da, $y \in \{e_1, \dots, e_k\}$. Orduan, erregresio logistiko bitarra aplikatzen da e_i etiketa 1 kasua moduan hartuz eta beste guztiak batera 0 kasua moduan, $i \in \{1, \dots, k\}$ bakoitzeko, k eredu desberdin lortuz. Azkenik, datu berriei eredu desberdin bakoitzean lortutako probabilitateen artean maximoa duen klasea izango zen elementuari egokitutako klasea, hau da, e_j non $\mathrm{h}_{\theta_j} = \max_{i \in \{1,\dots,k\}}(h_{\theta_i}(x))$ den.

One vs All metodoaz aparte, beste hainbat teknika desberdin existitzen dira. Adibidez, *multimodal logistic regresion* eta *ordinal logistic regresion*. Lehenak, One vs All teknikaren antzeko kontzeptuak lantzen ditu, baina etiketa guztiak bakarka aztertu beharren etiketa bat pibote moduan erabiltzen du eta besteak honekiko aztertzen ditu. Bigarrena aldiz ordenatutako kategorien azterketarako erabiltzen da.

1.3. K-nearest-neighbours

Klasifikazio tekniken artean k nearest neighbors edo KNN deituriko teknika intuitiboenetarikoa dela esan daiteke. Teknika hau klasifikaziorako balio duenez, gure datuak n elementu izango dira bakoitza bere etiketarekin.

Teknika honen ideia antzera etiketatutako elementuak batera egotera joko dutela da. Hori kontuan hartuz, elementu berri bat aztertzerakoan elementu horretatik hurbilen dauden elementuak aztertzen dira. Hau egin ostean, gehien agertzen den etiketa elementu berriaren etiketa izango da.



Adibide moduan, demagun eskumako argazkiko egoeran gaudela. Hemen, bi modutan etiketatutako elementuak ditugu (gurutze berdea eta gurutze gorria) eta datu berri bat lortzean (borobil horia) honi dagokion etiketa asmatu behar dugu. Orduan, $k = 3$ "auzokide" hurbilenak aztertuz gero, elementu berria gurutze gorri moduan etiketatu behar dela lortuko genuke. Aldiz, $k = 5$ kasuaren azterketa egiten bada, gurutze berde berri batean aurkituko ginateke.

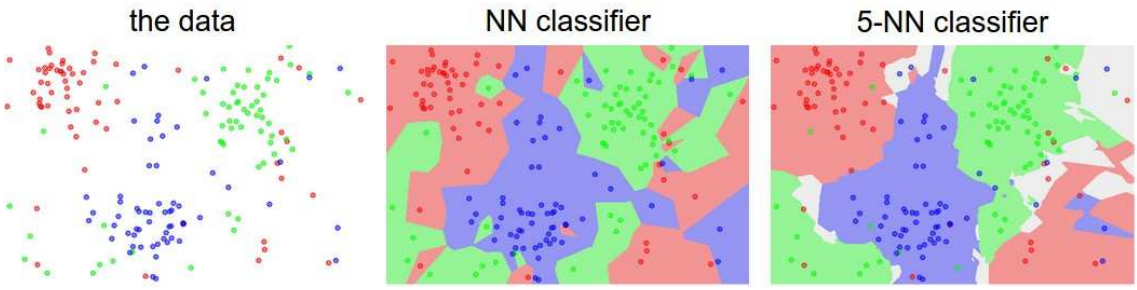
1.3.1. Algoritmoa

Etiketa ezezaguneko elementu berri bat aztertzerako orduan, elementu horren eta etiketa ezaguneko elementuen arteko distantzia, normalean, ohikoa den distantzia euklidearraren bitartez kalkulatzen da. Horretarako, aztertutako elementuak aldagai kuantitatiboak izan behar dira. Hala ere,datuak adierazgarriak diren aldagai kualitatiboak izanez gero, beste distantzia funtzio bat defini daiteke. Adibide moduan, seinaleak aztertzerakoan elementuen arteko

korrelazioa kontuan hartzen da eta karaktere kateak aztertzerakoan, hitz batetik bestera heltzeko zenbat elementu gehitu, aldatu edo ezabatu behar diren zenbatzen da.

Distantziez hitz egiten denez, aldagai guztiak teknikan eragin berdina izateko, egokiena aldagaiak normalizatzea izango dela.

Metodo honen aurrean jartzerako orduan hartu beharreko erabaki bakarra, aztertuko diren "auzokide" kopurua da. Parametro honen balioa erabakitzerakoan ondokoa kontuan hartu behar da: Gero eta baxuagoa izan orduan eta emaitza hobeak lortuko ditu baina gero eta altuagoa izan orduan eta emaitza leunagoa lortuko ditu. Beraz, gomendagarria da ditugun datuak bi multzotan, entrenamendu eta test multzotan, banantzea. Kasu honetan, test multzoko elementuak entrenamendu multzoko elementuekin konparatuko dira k-ren balio desberdinetarako eta gero, lortutako etiketa duten etiketarekin bat datorren ikusiko da. Hau egin ostean, errore txikiena duen k balioa hautatuko da.



1.3.2. Adibideak

KNN oso teknika sinple eta intuitiboa bada ere, klasifikazio problema askotan emaitza onak ematen ditu. Adibidez, eskuz idatzitako digituak aurresateko erabiltzen da edota satelitez lortutako irudien eszenarioa aurresateko ere.

1.3.3. Pros vs cons

Alde batetik teknika honek hainbat aspektu positibo ditu:

- Oso sinple eta eraginkorra da.
- Ez du entrenamendurik behar. Adibide berriak oso erraz gehitzen dira.
- Interpretazio oso sinplea dauka.

Bestalde, bere puntu negatiboak ere ditu:

- Konputazionalki garestia da. Datu basearen elementu eta aldagaiak handitu ahala orduan eta geldoago joango da.
- Eskalarekin kontu handia izan behar da.

1.4. Perceptron

Perceptron algoritmoa klasifikaziorako tekniken artean sinpleenetakoa da. Hemen, aztertuko diren datuak aldagai kuantitatiboak izango dira, bakoitzak duen etiketa izan ezik kualitatiboa izango dena.

Algoritmo hau aplikatu ahal izateko, etiketa desberdindun puntuen multzoak linealki banangarriak izan behar dira. Beste modu batean esanda, elementuek bi etiketa posible badituzte (demagun 1 eta -1 etiketak), -1 etiketadun eta 1 etiketadun elementuak banatzen dituen zuzen edo plano bat existitu behar da.

1.4.1. Algoritmoa

Teknika honek θ parametroa ($p \times 1$ dimentsiodun bektorea) aurkitzen saiatzen da non aztertzen den x_i elementu bakoitzeko, honen etiketa $y_i = sign(x_i\theta)$ izango den. Horretarako, ondoko balio funtzioa minimizatzen saiatzen da:

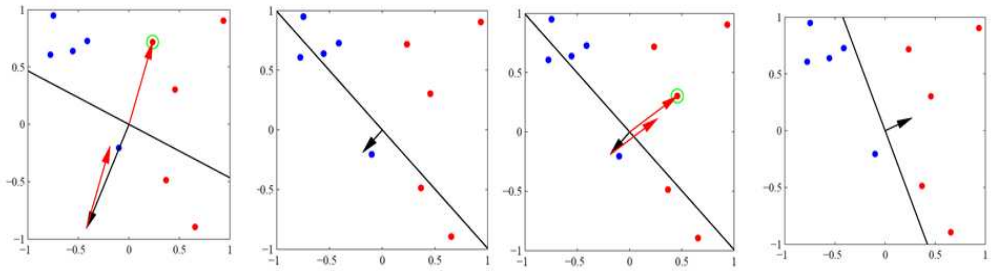
$$L = -\sum_{i \in M} y_i(\theta\, x_i),$$

non M multzoa txarto klasifikatutako, $y_i \neq sign(\theta\, x_i)$, elementuen indizeez osatutako multzoa den.

Hau lortzeko, *stochastic gradient descent* deituriko algoritmoa erabiltzen da. Hemen, elementu guztiek gradientean duten eraginaren batura honen norabide negatiboan eragin beharrean, elementuak banaka aztertzen dira.

Algoritmoak ondoko pausuak jarraitzen ditu:

1. $\theta^1 = 0$ hasieratu.
2. Txarto klasifikatutako (x_i, y_i) elementuak aurkitu ($y_i \neq sign(\theta\, x_i)$ betetzen dutenak).
3. Aurreko pausuan elementuren bat aurkitzen bada θ parametroa ondoko eran eguneratu: $\theta^{t+1} = \theta^t + \rho y_i x_i$.
    a. Bueltatu 2. puntura.
4. Bigarren puntuan ez bada elementurik aurkitzen, algoritmoa bukatu.



1.4.2. Kontuan hartu beharreko gauzak

- Data banangarria denean erantzun posible asko daude eta hasieratze balioen araberako erantzuna emango da.
- Pausu kopuru finitu baten emaitza lor badaiteke ere, pausu kopuru altua izan daiteke.
- Data banangarria ez bada metodoa ez du inoiz konbergituko eta lortutako emaitzak patroi zikliko bat jarraituko dute.

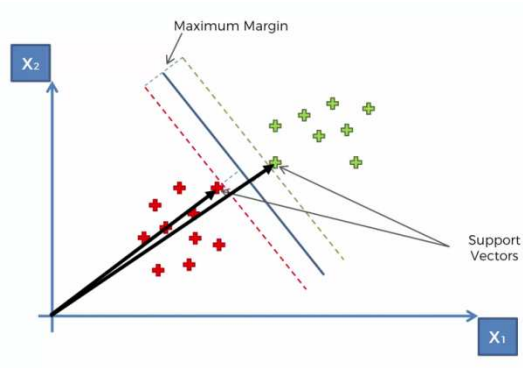
1.5. Support Vector Machine (SVM)

Klasifikaziorako erabiltzen diren algoritmoen artean SVM-a algoritmo indartsu eta nahiko zabaldua da. Algoritmo hau klasifikaziorako erabiltzen den algoritmoa da, hau da,

etiketatutako datu multzo batetik abiatuz, datuen eta etiketen arteko erlazioa ikasten saiatzen da. Horrela, etiketarik gabeko datu berriak lortzerakoan, datu horiek zein multzokoak diren aurresan ahal izateko.

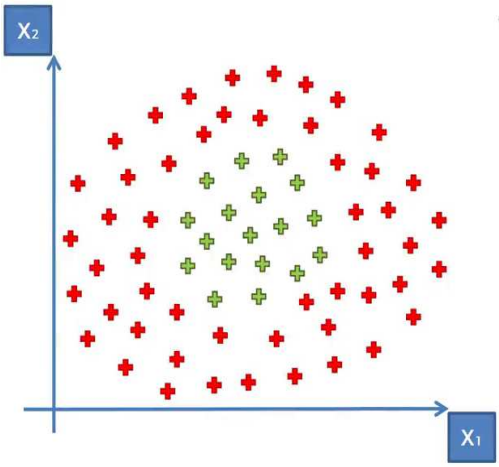
Demagun linealki banangarria den etiketatutako datu multzoa dugula. Algoritmo honek desberdin etiketatutako datu multzoak (demagun bi etiketa daudela) hiperplano batekin banatzen saiatzen da. Horrela, hiperplanoaren alde batean dauden datu guztiak etiketa bat izango dute eta beste aldekoak bestea.

Beste algoritmo batzuek, adib. "Percetron" algoritmoa, desberdin etiketatutako multzoak banatzen dituen muga aurkitzearekin nahikoa badute ere SVM algoritmoa "Large margin Classifier" moduan ezagutzen da. Klasifikazio metodo honek, multzo desberdinak banatzeaz aparte multzoen eta mugen arteko distantzia maximizatzen du.



SVM algoritmoa aplikatzerakoan, sarrera moduan aldagai kuantitatiboak izango ditugu eta, sarrera datu bakoitzeko, hauek etiketatzen dituen aldagai kualitatibo bat izango dugu.

1.5.1. Linealak ez diren mugak



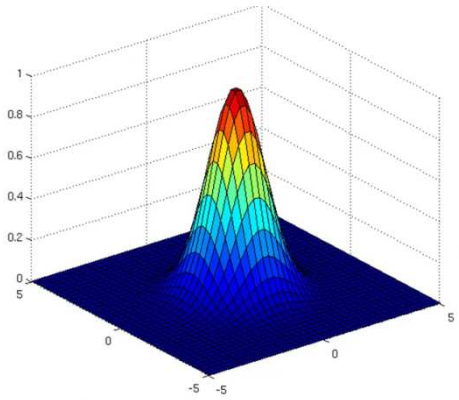
Azaldutako SVM metodoa duen arazo nagusia datuak linealki banangarriak izan behar direla da. Beraz, aurkitu daitezkeen kasurik gehienetan arazoak izango genituzke, eskumako argazkian ikus daitekeen moduan.

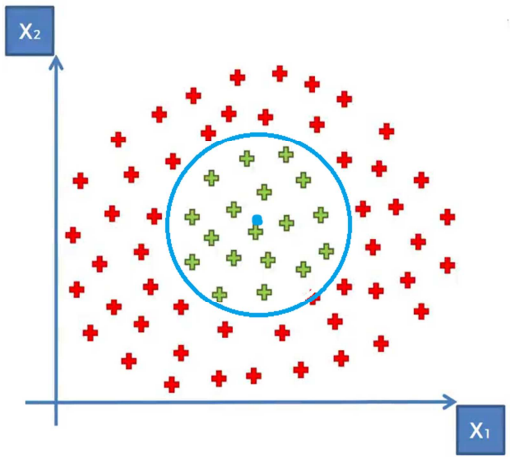
Kasu hauek gainditzeko daukagun datuei Kernel deituriko funtzioak aplikatzen zaizkie. Horrela, linealki banandu ezin diren puntuetatik abiatuz linealki banangarriak diren puntuak lortzen dira, ondoko bideoan ikus daitekeen moduan.

Erabiltzen den kerneletatik ezagunena edo erabiliena Kernel Gaussiarra da, RBF Kernel moduan ere ezagutzen dena: $K(\vec{x}, \vec{l^i}) = e^{-\left(\|\vec{x}+\vec{l^i}\|^2 / 2\sigma^2\right)}$

1.5.1.1. Algoritmoa

Kernel hau aplikatzeko, egin beharreko lehenengo gauza zentro bat, $\vec{l^i}$, eta bariantzarentzako balio bat, σ, aukeratzea da. Hau eginda, funtzio honek zentrotik hurbil dauden puntuen eta urrun dauden puntuen arteko aldea nabarmenduko du, urrun dauden puntuei zero balioa egokituz. Aurreko adibidearekin jarraituz, eta $\vec{l^i}$ eta σ egokiak aukeratuz, ondoren azaltzen den moduko emaitza lortuko genuke:





Demagun algoritmoa entrenatzeko daukagun multzoan n elementu daudela $(\overrightarrow{x_1}, \dots, \overrightarrow{x_n})$. Orduan, $\overrightarrow{l^i}$ balioak aukeratzeko posibilitate bat, elementu bakoitza dagoen toki berdinean $\overrightarrow{l^i}$ puntu bat jartzea litzateke, hau da, $\overrightarrow{l^1} = \overrightarrow{x_1}, \dots, \overrightarrow{l^n} = \overrightarrow{x_n}$. Kasu honetan n zentro eta, beraz, n kernel edukiko genituzke.

Orduan, algoritmoa entrenatzeko ondoko balio funtzioa defini daiteke:

$$C \sum_{i=1}^{n} [y^i cost_1(\theta^T f^{(i)}) + (1 - y^i) cost_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_{j=1}^{n} {\theta_j}^2$$

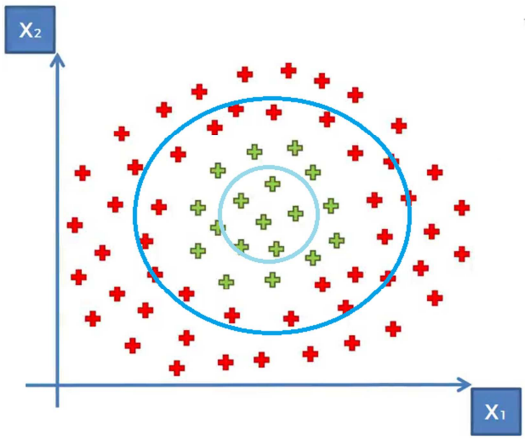
non

$$cost_1(z) = -\log\left(\frac{1}{1+e^{-z}}\right) \; eta \; cost_0 = -\log\left(1 - \frac{1}{1+e^{-z}}\right)$$

eta

$f^{(i)} = \begin{pmatrix} {f_0}^{(i)} = 1 \\ {f_1}^{(i)} = K(\overrightarrow{x_i}, \overrightarrow{l^1}) \\ \vdots \\ \vdots \\ {f_n}^{(i)} = K(\overrightarrow{x_i}, \overrightarrow{l^n}) \end{pmatrix}$ diren eta θ ikasi nahi diren parametroez osatutako bektorea den.

1.5.1.2. C eta σ balioak



RBF Kernel-a aplikatzerako orduan, kontu handia eduki behar da σ balioa aukeratzerakoan. Balio txikiegia hartuz gero algoritmoa oso zorrotza izango da, kasu positibo batzuk kanpoan utziz (zirkulu urdin argia). Bestalde, balio altua hartuz gero kasu positibo moduan puntu behar baino gehiago klasifika ditzake (zirkulu urdin iluna).

C balioa, aldiz, erregulazio parametro bat da. Balio oso altua erabiliz gero, algoritmoa datu baseko adibideak

ikasiko lituzke hauen arteko erlazioak ikasi beharrean eta etorkizuneko adibideetan huts eginez. Bestalde, balio baxua erabiliz gero, algoritmoak iragarpenak egiten ondo ez ikastea gerta daiteke, ez entrenamenduan ezta etorkizuneko kasuetan ere.

1.5.1.3. Beste Kernel batzuk

RBF Kernela erabilienetakoa bada ere, beste Kernel mota batzuk erabili ahal dira, adibidez:

1. Kernel lineala (kernel gabe): y=1 baldin eta soilik baldin $\theta^T X \geq 0$ bada.
2. Polinomikoa: $K(x, l) = (ax\text{^}T\ l + c)^{m}$ non a,c konstanteak diren eta m zenbaki arrunta den.
3. Laplaziarra: $K(x, l) = \exp(-\alpha\|x - l\|)$ non $\alpha > 0$ den.

1.5.2. Noiz erabili SVM (ERREGRESIO LOGISTIKOA VS SVM)

Demagun dugun datu kopurua n dela eta datu bakoitzak p aldagai kuantitatibo dituela, orduan:

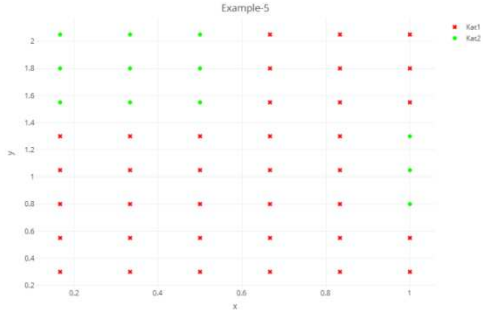
- Aldagai asko baditugu, hau da, p >>> n bada gomendagarria da erregresio logistikoa erabiltzea edota SVM kernelik gabe erabiltzea.
- p nahiko txikia bada (1-1000) eta datu asko baditugu, hau da, n (10-10000) (1:10) tartean badago SVM- Kernel Gaussiar batekin erabiltzea gomendagarria da.
- n oso handia bada (>50000) eta p nahiko txikia (1-1000 tartean) orduan SVM oso astiro joango litzateke. Beraz, aldagai gehiago gehitu edo sortzeaz aparte erregresio logistikoa erabiltzea edo SVM kernel-ik gabe erabiltzea gomendatzen da.

1.6. Decision trees

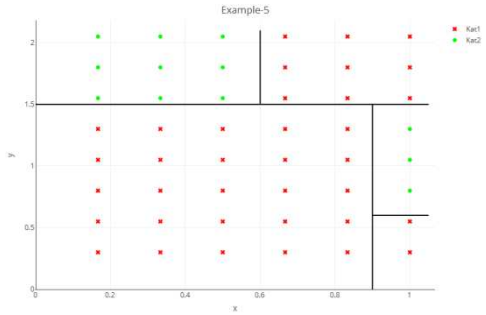
Zuhaitzetan oinarritutako metodoek aztertzen den datuen espazioa banatzen dituzte eta zati bakoitzari etiketa edo balio bat esleitzen dio. Metodo hauek bai erregresiorako baita klasifikaziorako erabili ahal diren arren hemen klasifikaziorako kasua aztertuko da.

Eremu desberdinak definitzerako orduan, mugen interpretazioa errazteko, modu errekurtsibo eta bitar batean definituko dira.

Adibide moduan, demagun aztertzen den datu baseko elementuek bi aldagai kuantitatibo, X eta Y, eta bi balioa hartzen dituen etiketa bat, Z, dituztela. Demagun ere datuak X eta Y aldagaiekiko irudikatzen direla eta puntu bakoitzari etiketaren arabera kolore bat ematen zaiola, eskuman ikus daitekeen grafikoa lortuz.

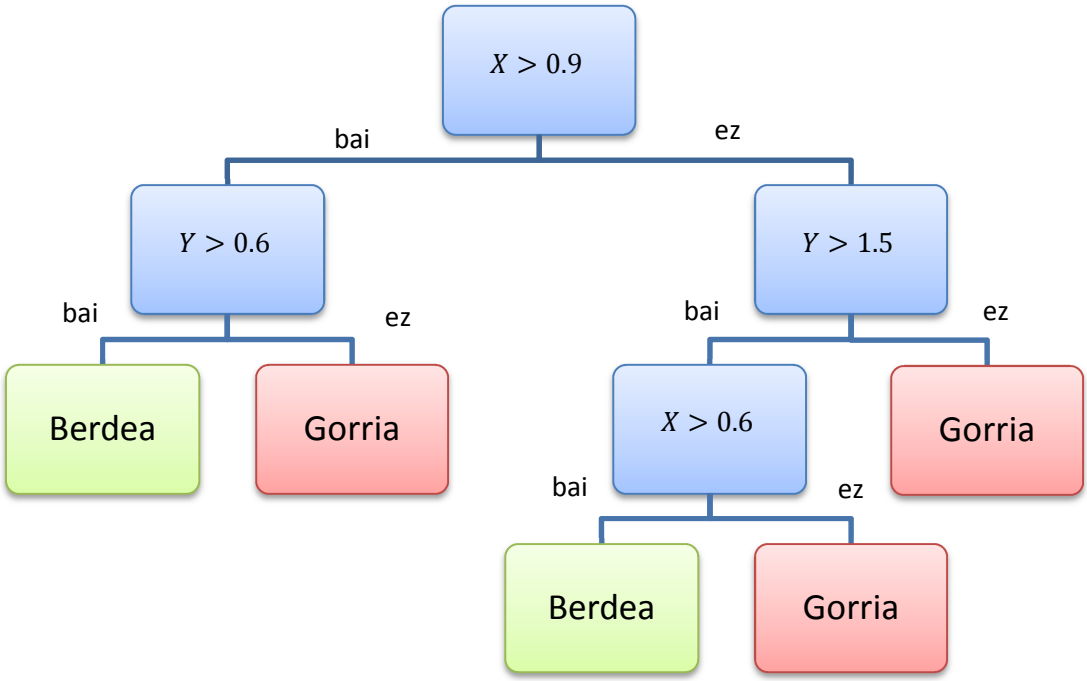


Irudian ikus daitekeen moduan, etiketa berdedun puntuak bi eremu isolatuetan daude, etiketa gorridunak beste espazio guztia betez. Algoritmoa datu multzoa bitan bananduko du X edo Y aldagaiarekiko zati bakoitzean mota bakarreko etiketa isolatzen saiatuz. Lortutako azpimultzoetan prozesu bera jarraituko du, eremu bakoitzean etiketa bakarra egon arte edo lortutako errorea aurretik finkatutako balio bat baino baxuagoa izan arte. Azaldutako



adibidean, ezkerreko grafikoan ikus daitekeen emaitza lortuko zen, ikus daitekeenez, etiketa gorri eta berdedunen arteko mugak oso ondo finkatzen dira.

Etiketa gabeko datu berri bat lortzerakoan, nahikoa litzateke bere Xeta Y aldagaien balioak aztertzea zein eremukoa den jakiteko. Horrela, duen etiketa aurresango zen. Lehenago lortutako emaitza, zuhaitz deituriko grafo batean laburbildu daiteke, era intuitiboago baten lan egin ahal izateko. Ondoren, aztertzen hari garen adibidearen zuhaitza ikus daiteke:



1.6.1. Algoritmoa

Demagun aztertu beharreko datu multzoa n aldagai dituela, hau da, $X_1, \ldots, X_n$. Algoritmoa iterazio bakoitzean bi gauza erabaki beharko ditu: datuak banatzeko erabiliko den j aldagaia eta s ebaketa puntua. Hau lortzeko ez dago erregela finkorik, algoritmoak aldagai desberdinen puntu desberdinak frogatzen ditu eta errore minimizatzen duen t_1 ebaketa puntua hautatzen du. Errorea kalkulatzeko balio desberdinak erabiltzen dira, ondoren hiru neurri definituko dira: *Klasifikazio errorea, Gini indizea* eta *Entropia.* Demagun p_i balioa aztertzen ari garen eremuan i-garren etiketadun elementuen proportzioa adierazten duela, orduan:

1. **Klasifikazio errorea:** $1 - \max_{1 \le i \le k} p_i$.
2. **Gini indizea:** $1 - \sum_{i=1}^{k} p_i^2$.
3. **Entropia:** $-\sum_{i=1}^{k} p_i \ln(p_i)$.

Hauen artean erabilienak Gini indizea eta Entropia dira. Ikus daitekeen moduan, balio guzti hauek 0 izango dira baldin eta soilik baldin aztertzen den eremua etiketa mota bakarra duen eremua bada.

Behin zatiketa puntua aurkituta, datu multzoa bi azpimultzotan banatzen da eta azpimultzo hauetan prozesu bera jarraitzen da. Hala ere, kontua handia izan behar da egiten den ebaketa kopuruarekin. Oso zuhaitz handiek aztertzen den datu base espezifikoa ikasiko (overfiting) lukete eta oso zuhaitz txikiak, aldiz, emaitza txarrak emango lituzke. Hau saihesteko, *Pruning* deituriko teknika jarraitzen da. Hemen, lehenik eta behin zuhaitza hazten da, aurretik finkatutako tamaina bat lortu arte. Ondoren, honen tamaina murrizten joaten da tamaina optimo bat eduki arte. Hasierako zuhaitzaren zein azpi-zuhaitzarekin geratuko garen erabakitzeko, zuhaitzaren erroreari zuhaitzaren tamainaren menpeko pisu bat gehituko zaio eta balio minimoa duen emaitza hautatuko da.

1.6.2. Abantailak vs desabantailak

Metodo honek hainbat abantaila ditu, adibidez:

- Interpretatzeko erraza da.
- Aldagai kuantitatibo zein kualitatiboekin la egiteko aukera ematen du.
- Ez du datuen prozesamendurik behar.
- Datu base handiekin emaitza onak ematen ditu.

Bestalde, baditu bere desabantailak ere:

- Ezegonkorrak izan daitezke, datuen aldaketa txiki bat lortutako emaitzan aldaketa handia eragin dezake.
- Algoritmoa ez da *NP-Complete,* hau da, problemaren soluzioa bat denbora polinomiokoan lortu daiteke baina globalki optimoa den problemaren emaitza lortzea ez da hain erraza.

1.7. Essembled methods

Teorian ikusitako metodo guztiak aplikatzerako orduan, modu eta parametro desberdinekin saiatu arren, batzuetan lortutako emaitzak nahi bezain egokiak ez izatea gerta daiteke. Hau gertatzerakoan, interesgarria litzateke emaitza ez oso onak lortzen dituen tekniketatik abiatuz eredu onak lortzea.

Bereziki hori da ondoren ikusiko diren metodo desberdinen helburua. Teknikak aztertzen hasi baino lehen, eredu desberdinen konbinazioak eredu solteak lorturiko emaitzak hobetzeko, hauen errorea %50 baino txikiagoa izan behar du. Beste modu batean esanda, ausaz lortutako ereduak baino emaitza hobeak eman behar dituzte.

1.7.1. Boostrap

Entrenamendu datu multzo berdinari metodo berdina aplikatzerakoan, logikoa den moduan, emaitza berdinak lortuko genituzke. Beraz, emaitzak hobetzeko eredu desberdinak garatu eta konbinatu nahi baditugu entrenamendu datu multzo desberdinak erabili beharko genituzke. Horretarako, “Boostrap” deituriko teknika erabili daiteke.

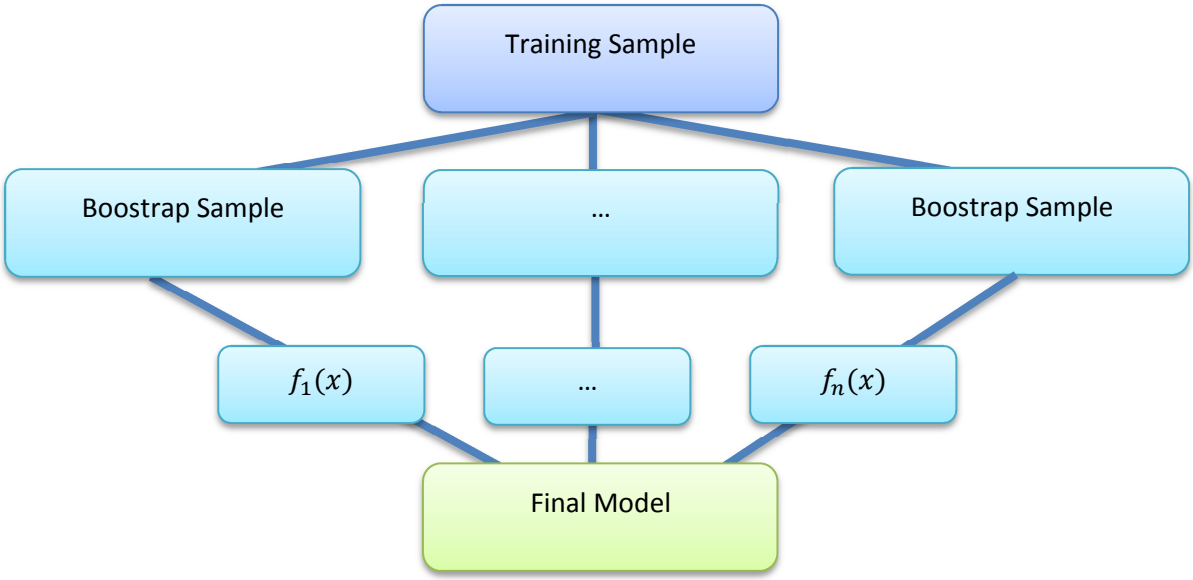
Teknika honek hasierako datuetatik $\{x_1, \dots, x_n\}$, n ausazko aukeraketa egiten ditu, errepikapenekin. Hau da, b multzo entrenatu nahi izanez gero, ausaz b aldiz hasierako multzotik n elementu errepikapenekin aukeratzen dira, $B_1, \dots B_b$, lagin desberdinak lortuz.

Behin entrenamenduko datu multzotik konbinatuko diren lagin desberdinak lortuz, hauek konbinatzeko teknika desberdinak ikusiko ditugu.

1.7.2. Bagging (Boostrap aggregation)

Metodo hau ikusiko direnen artean sinpleenetarikoa dela esan daiteke. Kasu honetan lortutako entrenamendu multzo bakoitzetik eredu bat lortzen da, f_b, eta gero datu berri bat, x_0 , lortzerakoan honi loturiko etiketa edo balioa ondoko eran kalkulatuko litzateke:

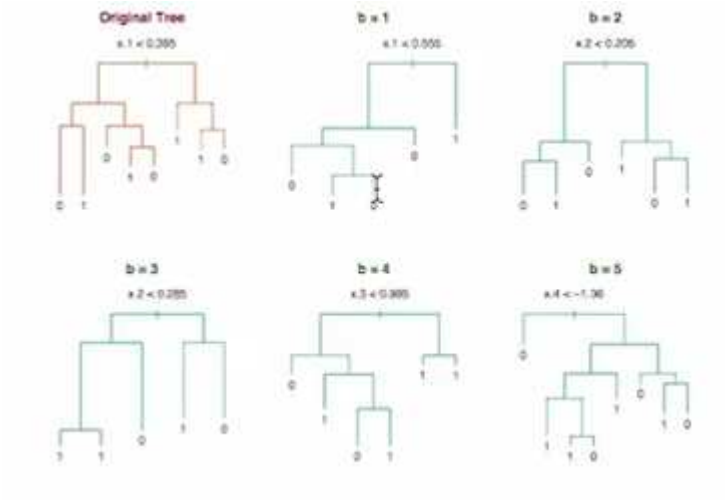
- Eredua erregresio eredua bada, emaitza eredu desberdinen emaitzen media izango litzateke, hau da, $f(x_0) = \frac{1}{b}\sum_{i=1}^{b} f_i(x_0)$ kalkulatuko litzateke.
- Eredua klasifikaziorako bada, lortutako b ereduak aplikatu ostean kasu gehienetan lorturiko etiketa egokituko zaio.



1.7.3. Random forest

Erabaki zuhaitzak garatzerako orduan Bagging teknikak espero baino emaitza txarragoak ematen ditu. Honen arrazoirik nagusiena Bagging teknikarekin lortutako zuhaitz desberdinen artean oso korrelazio altua dagoela da. Hori dela eta, lortutako emaitzak bateratu arren antzeko emaitzak lortuko genituzke.

Arazo hauek gainditzeko Ramdom forest teknika garatu zen. Nahiko sinplea den eta Bagging teknikarekin antzeko ezaugarriak mantentzen dituen teknika bada ere, emaitza hobeagoak lortzen dituela ikus daiteke. Bagging eta Random forest algoritmoen arteko desberdintasuna, Boostrap teknika aplikatu ostean lortutako lagineko datuetatik $m < p$ ausazko aldagai aukeratzen dituela da, lagin bakoitzeko m aldagaien aukeraketa desberdina izanda. Normalean, $m \sim \sqrt{p}$ erabiltzen da baina edozein balio aukeratu daiteke.



1.7.4. Boosting

Bukatzeko, boosting teknikan ere eredu ahul desberdinak emaitza hobeagoak lortzeko konbinatzen diren arren, aurreko teknikekiko desberdintasun nabari bat dauka. Kasu honetan, ez dira entrenamendurako multzo desberdinak boostrap-en bidez sortu behar. Hemen, dugun entrenamendurako datu multzotik abiatuz eta elementu bakoitzari "pisu" (α_i) bat emanez entrenatuko den multzo berria aukeratzen da. Eredu guztiak kalkulatu ostean, hauek entrenatzeko erabili diren pisuek bakoitzak duen eragina adieraziko dute. Beraz, eredu finala ondoko moduan kalkulatuko litzateke:

$$f(x_0) = sign(\sum_{i=1}^{b} \alpha_i f_i(x_0))$$

1.7.4.1. Adaboost algoritmoa

Boosting-a erabiltzen duen algoritmo ezagunenetako bat AdaBoost algoritmo da. Demagun n elementu ditugula gure entrenamendu multzoan eta elementu bakoitzeko etiketa bat dugula (1 edo -1), hau da, $(X_1, y_1), \dots, (X_n, y_n)$ tuplak ditugula X_i elementu bat izanda eta y_i honen etiketa. Hasiera batean lehengo ereduan parte hartzeko elementu bakoitzaren probabilitatea $p_{0,i} = \frac{1}{n}$ izango da. Orduan Adaboost algoritmoak ondoko pausuak jarraitzen ditu $t = 1, \dots, T$ iterazio bakoitzeko:

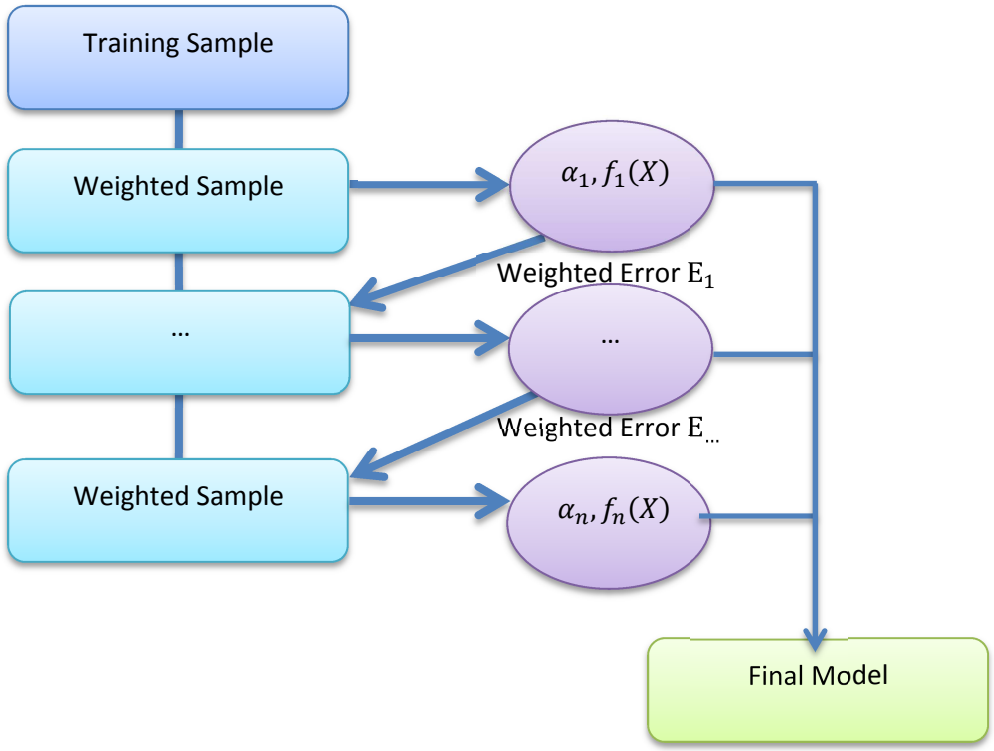
1. Boostrap teknikarekin X_i elementu bakoitzak $p_{t,i}$ aukeratzeko probabilitatea edukita, n tamainako B_t multzoa aukeratzen da.
2. f_t klasifikazio eredua ikasten da B_t multzoarekin.
3. Lortutako ereduaren erroreak eta pisuak ondoko eran kalkulatzen dira :
    a. $e_t = \sum_{i=1}^{n} p_{t,i} 1\{y_i \neq f_t(X_i)\}$.
    b. $\alpha_t = \ln(\frac{1-e_t}{e_t})$.
4. Probabilitate berriak bi pausutan kalkulatzen dira:
    a. Elementu bakoitzaren eragina kalkulatzen da txarto klasifikatuei garrantzi handiagoa emanez:
        i. $\hat{p}_{t+1,i} = p_{t,i}\, e^{-\alpha_t\, y_i\, f_t(X_i)}$

b. Elementu guztien eragina kontuan hartuz, bakoitzak hurrengo pausuan kontuan hartzeko duen probabilitatea kalkulatzen da.

   i. $p_{t+1,i} = \frac{\hat{p}_{t+1,i}}{\sum_j \hat{p}_{t+1,j}}$

Bukatzeko elementu berri bakoitzaren etiketa aurresateko lortutako eredu desberdinak konbinatzen dira. Hau egiterakoan eredu bakoitzaren eragina aurretik kalkulatutako pisuak izango dira:

$f_{Boost}(x_{new}) = sign\,(\sum \alpha_t\ f_t(x_{new}))$.



Ondoko estekan algoritmoaren eboluzioa ikus daiteke 300 iterazioetan zehar.

2. Unsupervised learning

Unsupervised learning motatako tekniketan datuek ez dute inongo etiketa edo baliorik esleituta izango. Hemen, ditugun datuei buruz "ezkutatutako" informazio lortzea izango da helburu nagusia. Teknika hauei esker antzeko ezaugarriak dituzten elementuak batu, bi pertsonen gustuak jakinda errekomendazioak egin (Netflix edo Amazon egiten duten moduan), informazio minimoa galduz datuen dimentsioa laburtu edota elementuen arteko menpekotasuna aztertu moduko gauzak egin daitezke.

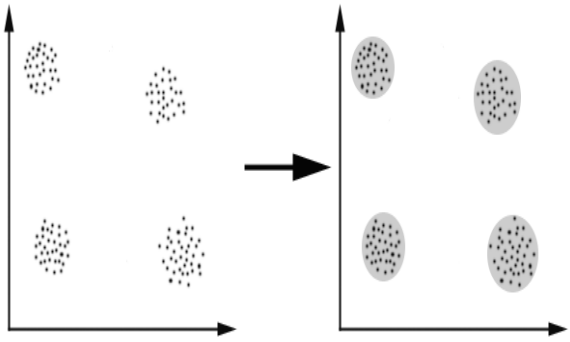
Unsupervised learning metodoen artean ondokoak aztertuko dira:

1. Kluster
2. PCA
3. Asociation rules
4. Content based filtering eta collaborative filtering
5. Markov Models

2.1. Kluster

Kluster teknika aztertzen den datu multzoan dauden egiturak aurkitzen eta datuak multzo desberdinetan batzen saiatzen den "unsupervised" metodo barruan dagoen teknika bat da.

Esan bezala, ditugun datuen elementuak multzokatzea ("kluster" desberdinetan sartzea) da metodo honen helburua. Horretarako, bereziki bi ideia kontuan hartu beharko dira. Alde batetik, multzo berdineko elementuak oso antzekoak izan behar dira. Bestetik, multzo desberdineko elementuak beraien artean ahalik eta desberdinenak izan behar dira.



2.1.1. K-means algoritmoa

Klusterrak kalkulatzerako garaian teknikarik ezagunenetariko bat k-means algoritmoa da. Algoritmo hau aplikatzerako garaian, datuak zenbat k multzotan banandu nahi diren finkatu beharko da. Hau eginda, datuen multzoko elementuen artean k aukeratzen dira, $(\mu_1, \dots, \mu_k)$, klusterren *zentroideak* izango direnak. Ondoren, algoritmoak ondoko pausuak errepikatzen ditu:

1. x_i elementu bakoitzari hurbilen duen c_i *klusterra* egokitzen zaio. Horretarako, $\arg\min_k \|\mu_k - x_i\|_2$ kalkulatzen da.
2. n_k balioa k klusterrean dauden elementuen kopurua adierazten badu, *zentroideak* berritu egiten dira ondoko eran: $\mu_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x_i$ non x_i elementuak k klusterrean dauden.

Prozesu berdina behin eta berriz errepikatzen da konbergitu arte edota aurretik finkatutako iterazio kopurua igaro arte. Ondoko bideoan algoritmoaren adibide intuitibo bat ikus daiteke.

2.1.2. Klusterren aukeraketa

K-means teknikaren berezitasun bat ditugun datuak zenbat kluster desberdinetan bananduko diren hasieratik erabaki behar dela da. Hala ere, gomendagarria da algoritmoa k-ren balio desberdinetarako aplikatzea eta lortutako emaitzak konparatzea. Emaitzak konparatu ahal izateko kostu funtzio bat definitu behar da. Adibide bezala, gure X datu baseak n elementu baditu, ondokoa kostu funtzio posible bat litzateke:

$$f(X, \mu_1, \dots, \mu_k) = \frac{1}{n}\sum_{j=1}^{k}\sum_{i=1}^{n_j}\left\|x_{ji} - \mu_j\right\|_2$$

non x_{ji} elementua j klusterraren i elementua den.

Ohartu gero eta kluster gehiago definitu orduan eta kostu txikiagoa izango duela, n elementurako n kluster definitzerakoan zero kostua lortuz. Aldagai honen balioa helburuaren arabera aukeratzen da. Adibidez, aztertzen den datu basea enpresa baten bezeroak badira eta bezeroak k langileen artean banandu behar badira, langile bakoitzari egokitutako bezeroak ahal eta antzekoenak izanda, k kluster egitea interesgarria litzateke.

2.1.3. Ohiko arazoa

k-means algoritmoa aplikatzerakoan, hasieran egiten den ausazko zentroideen aukeraketaren arabera algoritmoa emaitza optimo lokal batean gera daiteke. Kasu honetan, emaitza honek koste baxua izan arren koste baxuagoko emaitza bat existituko litzateke. Egoera hau saihesteko gomendagarria litzateke algoritmoa hasieratze desberdinekin aplikatzea eta kostu minimoa duen emaitzarekin geratzea. Ondoko bideoan algoritmoa hasieratze desberdineko emandako emaitzak ikus daitezke.

2.1.4. K-medoids

K-means funtzioaren kasuan bi elementuen arteko distantzia edo antzekotasuna norma euklidearraren bitartez neurtzen da. Beraz, algoritmo hori aplikatzeko nahi eta nahi ez datuak kuantitatiboak izan behar dira. Gainera, distantzia euklidearra erabiltzen den heinean, distantzia altuko elementuak eragin handia izango dute metodoan. Arazo hauek gainditzeko, k-medoids metodoa definitzen da. Algoritmo honen eta k-means algoritmoaren arteko desberdintasun bakarra algoritmoaren lehenengo atalean dago. Hemen, elementuak kluster desberdinei egokitzeko garaian hauen eta kluster desberdinen arteko konparaketa beste desberdintasun D funtzio bat erabiltzen da. Beraz, algoritmoaren lehenengo pausua ondokoa litzateke:

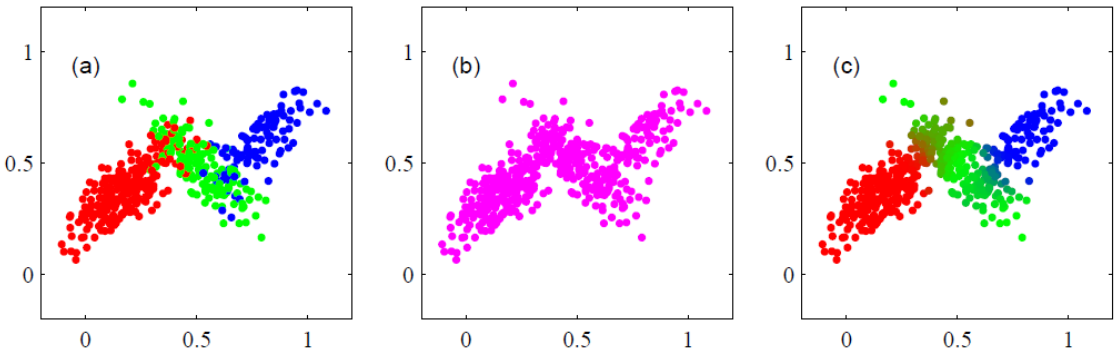
1. x_i elementu bakoitzari hurbilen duen c_i *klusterra* egokitzen zaio. Horretarako, $\arg\min_k D(\mu_k, x_i)$ kalkulatzen da.

Kmedoids metodoaren kasuan, kmeans metodoan ez bezala, Klusterren zentro moduan datu baseko elementuak aukeratzen dira.

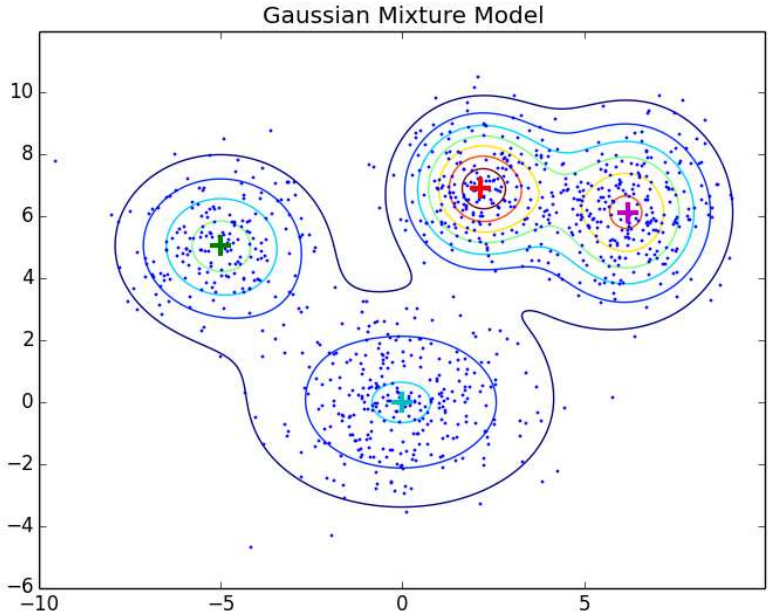
2.1.5. Mixture models

Kluster bat egiterakoan bi motatako teknika daudela esan daiteke: *hard clustering* eta *soft clustering*. Lehenengoaren kasuan, elementu bakoitzari kluster bat egokitzen zaio eta bigarrenean, aldiz, elementu bakoitzari kluster bakoitzean egoteko probabilitatea egokitzen

zaio. Lehen ikusitako k-means algoritmoa lehenengo teknika multzoari dagokio. Ondorengo irudian datu base bati k-means (3 kluster) algoritmoa lortutako klusterrak (a), aztertzen den datua basea (b) eta *soft clustering* familiako teknika batek (c) emandako emaitzak ikus daitezke.



Bigarren metodo probabilistiko hauetan M*ixture models* deituriko teknikak erabiltzen dira. Hemen *Gaussian Mixture Models* (*GMM*) deituriko teknika aztertuko ditugu. Teknika honek, k-means algoritmoa moduan, aztertzen diren datuak *k* klusterretan banatzen saiatzen da. Horretarako, sarrerako datuen artean parametro desberdineko k banaketa normal daudela suposatzen da. Ereduak elementu bakoitza banaketa bati egokituko dio. Banaketa normalen bariantza zerorantz joan ahala, lortutako emaitza k-means algoritmoak ematen dituen emaitza berdinak lortuko dira.

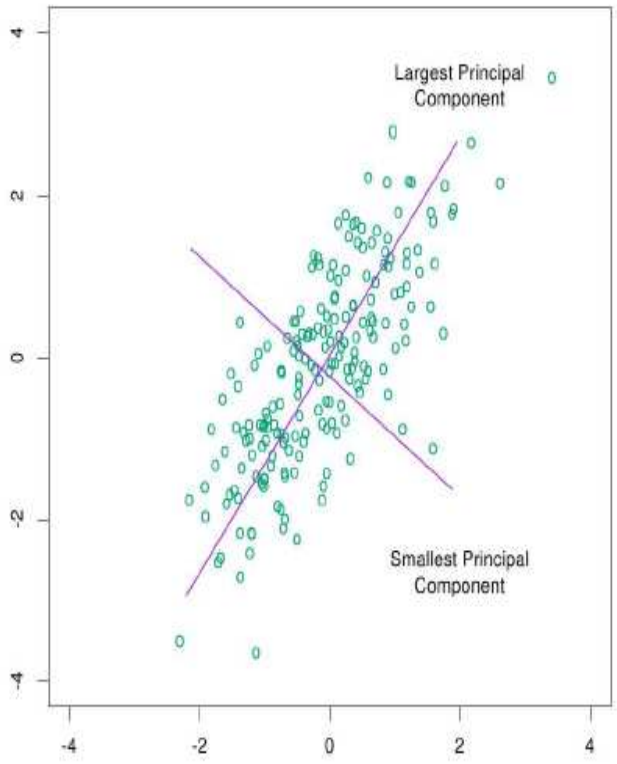


Eredu hauen abantailarik nagusiena *hard clustering* teknikak baino emaitza zehatzagoak ematen dituztela da. Hala ere, datuen banaketa aurretik finkatzen denez, kluster desberdinak itxura finko batekoak izango dira, Gaussian Mixture Models kasuan, elipsoideak. K-means algoritmoa aldiz, oso algoritmo azkarra da, datu kopurua handiarekin lan egin ahal duena.

2.2. PCA

Big data-ri buruz hitz egiterakoan datu kantitate erraldoi bati buruz hitz egiten ari gara. Hori dela eta, analisi desberdinak egiteko, erabiliko diren teknikak datu kopuru altu batekin borrokatu beharko dira. Aldagai askorekin lan egingo dugula kontuan hartuz, interesgarria litzateke aldagaien informazio guztia edo ia guztia aldagai kopuru txikiago batekin ordezkatzea. PCA teknikaren helburua hori litzateke, aztertzen den datu basearen aldagaiak kontuan hartuz, hauen informazioa dimentsio txikiago batera bidaltzea.

Demagun gure datu baseak n elementu dituela, bakoitza d aldagaiekin. Orduan, gure datuak d dimentsioko espazio batean "bizi" direla esan daiteke. PCA-ren lana dimentsio horretako zein norabideetan datuen bariantzarik altuena dagoen ikustea da, hau da, zein norabideetan dagoen elementuen aldaketa handiena. Norabide horiek jakinda, PCA teknikak dimentsio baxuagoko espazio batean gure datuen hurbilpenik onena lortzen du.



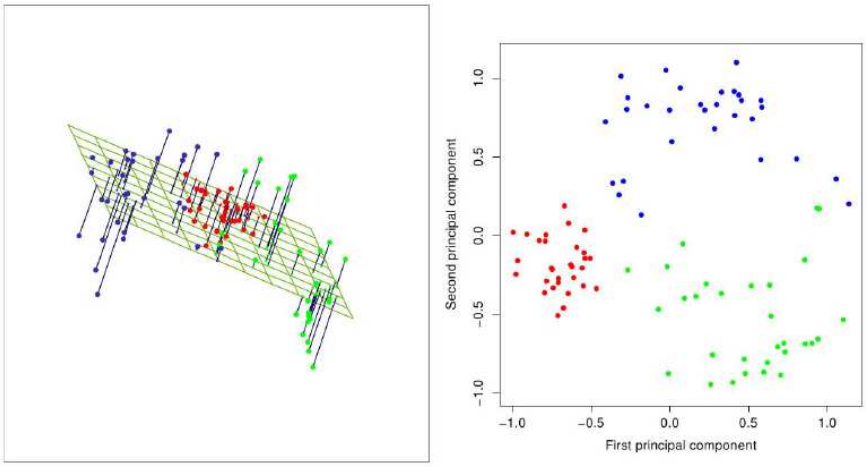
2.2.1. d=2 adibidea

Kontzeptua hobeto ulertzeko, ondorengo irudian $d = 2$ den adibide bat ikusiko dugu. Adibidean gure datuen puntu grafikoa ikusteaz aparte, geroago definituko ditugun datuen elementu nagusiak (principal component) ikusten dira. Hauek informazioaren aldaketa handiena zein norabideetan gertatzen den adierazten dute.

Kasu zehatz honetan gure datuak bi dimentsiodun espazio batean daudenez, hauek espazio txikiago batean aztertu nahi izanez gero bat dimentsioko espazio batera eramatea aukera bakarra da. Horretarako, datu guztiak gure elementu nagusietako handienera proiektatuko ditugu, datu guztiak bat dimentsioko zuzenean utziz.

2.2.2. d=3 adibidea

Ondoren, adibide modura, $d = 3$ den adibide bat ikusten da non datuak hiru dimentsioko espaziotik abiatuz bi dimentsioko espazio batera proiektatzen diren.



2.2.3 Alde teorikoa

Ditugun datuak dimentsio baxuago batera proiektatzen ditugunean informazio galera bat egongo dela dakigu. Beraz, logikoa den moduan, proiekzioa egiterakoan dagoen informazio galera txikiena bermatzen duten norabideak aukeratuko dira. Hasieran komentatu bezala, aldagai desberdinen arteko bariantza aztertuko da hauen arteko erlazioa ikusteko. Hau guztia kontuan hartuz, froga daiteke kobariantza matrizearen balio singularren deskonposizioan (SVD) lortutako norabidea bilatzen ari garen norabideak direla. Beraz, $Z = \frac{1}{p}XX^T$ datuen kobariantza matrizea bada, $Z = US^2U^T$ honen SVD-a litzateke non U matrize ortonormala $p \times p$ dimentsiokoa den eta S^2 matrize diagonala den, Z-ren $\lambda_1 > \lambda_2 > \cdots > \lambda_d > 0$ balio propioez osatua dagoen eta $d \leq m$ den.

Orduan, transformaziorako behar diren bektoreak U matrizearen lehenengo $q \leq p$ zutabeak izango dira. Bektore hauek U_q proiekzio matrizea osatuko lukete. Ohartu, $q = p$ kasuan dimentsioa ez dela murriztuko, koordenatu aldaketa bat soilik izango bailitzateke.

Bukatzeko, gure datuak $\{x_1, \dots, x_n\}$ badira, PCA aplikatu ostean lortutako datuak $\{\tilde{x}_i\}_{i\in\{1,\dots,n\}}$ lirateke non $\tilde{x}_i = U_q x_i$ diren.

2.2.4 Dimentsioen aukeraketa

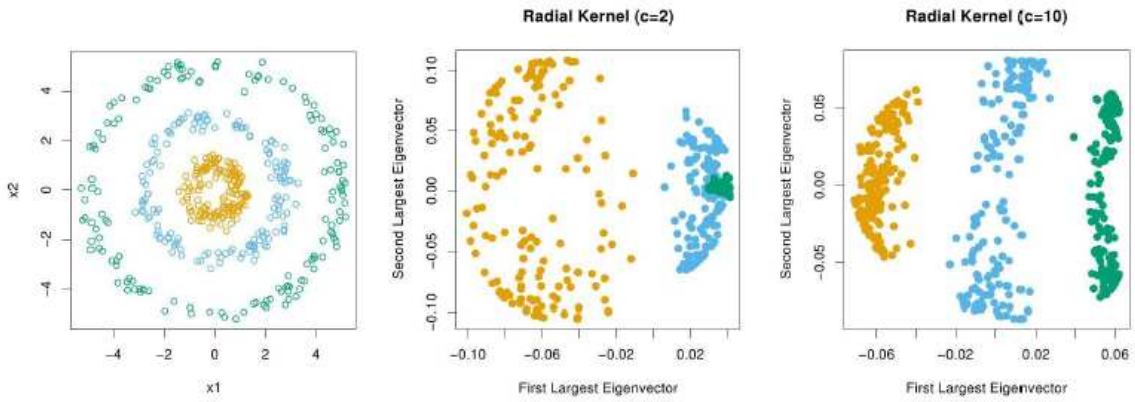
Behin datu multzoaren proiekzio matrizea kalkulatuta erabiliko diren dimentsioak aukeratu behar dira. Horretarako, dimentsio bakoitzak ordezkatzen duen hasierako datuen bariantzaren ehunekoa zein den aztertzen da eta egokia den kopurua aukeratzen da. Hau egiteko ez dago lege edo erregela finkorik eta bakoitzaren beharren araberako aukeraketa egiten da. Demagun $\lambda_1, \dots, \lambda_d$lortutako balio propioak direla, orduan lehenengo $k < d$ aldagai nagusiak ordezten duen bariantza ondokoa litzateke:

$$\frac{\sum_{i=1}^{i=k} \lambda_i}{\sum_{j=1}^{j=d} \lambda_j} * 100.$$

2.2.5 Kernel PCA

Beste teknika batzuekin gertatzen den moduan hemen ere datuen linealtasuna aurresuposatzen da baina hori ez da beti zertan gertatu behar. Kasu hauetan, teknika aplikatzen hasi baino lehen datuak dimentsio altuago batera eramatea beharrezkoa litzateke, datuen linealtasuna bermatzeko. Horretarako, kernel desberdinak erabili daitezke, kernel gaussiarra esaterako.

Ondoren, etiketatutako datu multzo bati PCA kernel gaussiar desberdinak aplikatuz lortutako emaitzak ikusten dira.



Ikusten denez, Kernel Gaussiarra $c = 10$ balioarekin aplikatu ostean, lortutako emaitza datuen benetako natura askoz hobeago erakusten du.

2.2.6 Aplikatzerakoan kontuan hartu beharreko puntuak

- Balio galduekin kontu berezia eduki behar da. Metodo honen helburua informazioa laburtzea da eta horretarako korrelazio edo kobariantza matrizea erabiltzen du. Hori dela eta, matrize hau ondo definituta egotea beharrezkoa da.
- Aldagaien arteko bariantza neurtzen denez, nahitaezkoa da hauek eskala berdinean egotea. Horretarako aldagaiak estandarizatzea gomendagarria litzateke metodoarekin hasi baino lehen.
- Balio propioak hautatzerakoan, gero eta gehiago hautatu, orduan eta bariantza orokorraren portzentajea altuagoa azalduko da. Hala ere, batzuetan n balio propio hautatzetik $n+1$ hautatzera dagoen aldea oso txikia da. Beraz, balio propioen kopurua ondo aukeratu behar da, azaldutako bariantza eta aldagai berrien kopuruaren arteko balantza mantenduz.
- Kontuan hartu kalkulatutako U_q matrizea entrenamendurako multzoarekin kalkulatzen dela. Matrize hori bai entrenamendurako balidaziorako eta baita test-erako erabiliko da.

2.3 Association rules

Asoziazio analisia datu multzo baten barruan batera azaltzen diren gertakizunak aurkitzen saiatzen da. Teknika honek, bereziki, aldagai bitarrak dituen datu base komertzialak aztertzen

ditu eta “market basket” analisi moduan ezagutzen ohi da. Testuinguru honetan, aldagai bakoitza produktu desberdin bati egokituko litzateke, adibidez, $\{X_1, \dots, X_d\}$ elementu badaude, i elementu bakoitzeko X_j aldagaiak bi balio desberdin hartuko lituzke: $x_{ij} = 1$ balioa, i obserbazioan j produktua agertzen dela adieraziko luke eta $x_{ij} = 0$ balioa aldiz, j produktuaren falta adieraziko luke. Asoziazio legeak aldiz produktu edo elementu baten edo multzo baten presentziak beste elementu desberdin baten presentzian duen eragina aztertzen du.

Esan bezala, normalean testuinguru komertzial batean erabiltzen den teknika da. Bereziki, denden apalen antolamenduan, produktuen promozioen marketing-a prestatzeko, katalogoen diseinuan eta erosketa patroietan oinarritutako erosleen segmentazioan erabili ohi da.

Adibide moduan, demagun janari denda baten jabeak bere dendatik igaro izan diren azkenengo bost erosleen erosketen zerrendak dituela:

Eroslea	**Produktuak**
1	{ogia, esnea}
2	{ogia, haur-oihala, garagardoa, arrautzak}
3	{esnea, haur-oihala, kola, garagardoa}
4	{ogia, esnea, haur-oihala, garagardoa}
5	{ogia, esnea, haur-oihala, kola}

Orduan asoziazio legeek bermatuko liokete erosketen patroiak aurkitzea, adibidez, haur-oihalak erosten dituzten bezeroek garagardoa ere erosteko aukera altua dela. Asoziazio analisiak aldiz, ogia eta esnea normalean batera erosten direla azalduko luke.

2.3.1 Asoziazio analisia

Algoritmoa azaldu baino lehen, erabiliko diren pare bat kontzeptu ikusiko dira. Izan bedi $\mathcal{K} \subset \{1, \dots, d\}$ azpimultzoa $X_1 = 1, \dots, X_d = 1$ elementuen presentzia adierazten duena eta izan bitez $A, B \subset \mathcal{K}$ non $A \cap B = \emptyset$ eta $A \cup B = \mathcal{K}$ diren. Ondoko kontzeptuak kontuan hartu behar dira:

- $\mathcal{P}(\mathcal{K}) = \mathcal{P}(A, B)$; $\mathcal{K}$ multzoko elementuen f*rekuentzia.* Zein elementuen konbinazioa askotan agertzen den ikustea interesgarria litzateke,
- $\mathcal{P}(A|B) = \frac{\mathcal{P}(\mathcal{K})}{\mathcal{P}(B)}$; B elementua dagoela jakinda zein den A elementua egoteko *konfiantza*. Normalean $A \Longrightarrow B$ *lege* bat dagoela adierazteko erabiltzen da.
- $\mathcal{L}(A, B) = \frac{\mathcal{P}(A,B)}{\mathcal{P}(A)\mathcal{P}(B)} = \frac{\mathcal{P}(B|A)}{\mathcal{P}(B)}$; $A \Longrightarrow B$ erregelaren “L*ift*” deiturikoa. A elementua dagoela jakinda B elementua egoteko konfiantzan dagoen segurtasuna zein den neurtzen du.

2.3.2 Apriori algoritmoa

Algoritmo honen helburua aurredefinitutako t “threshold” bat baino frekuentzia altuagoa duten $\mathcal{K} \subset \{1, \dots, d\}$ azpimultzoak bilatzea da. Helburu hau lortzen laguntzeko ondoko bi propietateak erabiltzen dira:

- Aztertutako $\mathcal{K}$ azpimultzoaren frekuentzia nahikoa ez bada, $\mathcal{K}$ barruan duen edozein multzoaren frekuentzia ez da nahikoa izango. Matematikoki baliokidea dena, $P(\mathcal{K}) < t$ bada eta $\mathcal{K}' = \mathcal{K} \cup A$ bada $A \subset \{1, \dots, d\}$ izanda, $P(\mathcal{K}') < t$ izango da.
- Aztertutako $\mathcal{K}$ azpimultzoaren frekuentzia nahiko altua bada eta $A \subset \mathcal{K}$ bada, orduan A multzoaren frekuentzia nahiko altua izango da. Baliokidea dena, $P(\mathcal{K}) > t$ eta $A \subset \mathcal{K}$ badira, orduan $P(A) > t$ izango da.

Propietate hauen laguntzaz *apriori* algoritmoaren bertsio sinple bat ondokoa litzateke:

1. $0 < t < 1$ "threshold" bat finkatu.
2. Elementu bakarreko, $|\mathcal{K}| = 1$ azpimultzoetatik, t bainoa frekuentzia altuagoa dituztenak gorde.
3. Aurreko pausuan gordetako elementuak konbinatuz lortzen diren bi elementuetako $|\mathcal{K}| = 2$ azpimultzoetatik, t bainoa frekuentzia altuagoa dituztenak gorde.
4. Pausua errepikatzen da $\mathcal{K}$ multzoaren elementu kopurua, $|\mathcal{K}| = k$, handituz $k \leq p$ elementu dituen multzoetatik t bainoa frekuentzia altuagoko bat ere ez egon arte.

Aurreko propietateak kontuan hartuz begi-bistakoa da k handitu ahala baztertzen diren azpimultzoen kopurua handituko dela. Horri esker, konbinazio guztiak ez dira aztertu beharko. Gainera, oso probablea da $k < p$ bat existitzea non bertatik aurrera dauden azpimultzo guztiak baztertzen diren.

2.3.3 Algoritmoa hobetzen

Emandako algoritmoa *apriori* algoritmoaren inplementazio basikoa besterik ez da eta suposa daitekeen moduan modu desberdinetan hobe daiteke:

- $|\mathcal{K}| = k - 1$ elementuko multzo baten frekuentzia t baino altuagoa dela aurrez aldetik jakinez gero, multzo horren elementuez edota hauen konbinazioez osatutako azpimultzoen frekuentzia t baino altuagoa dela baiezta daiteke. Beraz, azpimultzo hauek ez dira zertan aztertu behar.
- $\{a, b\} \cup \{c\} = \{c\} \cup \{a, b\}$ denez, errepikapenak saihesteko indizeak erabiltzea lagungarria da.

2.3.4 Asoziazio legeak

Behin batera agertzeko joera dituzten elementuak aurkituta, hauen artean kausalitate erlazioren bat dagoen aztertzen da. Horretarako, bigarren t_2 "threshold" bat finkatzen da eta aurreko algoritmotik lortutako $\mathcal{K}$ azpimultzoak aztertzen dira. Demagun $A \cup B = \mathcal{K}$ betetzen dela orduan $A \Longrightarrow B$ beteko da baldin eta $\mathcal{P}(A|B) > t_2$ bada. Gogoratu *apriori* algoritmoan $P(A)$ eta $P(B)$ kalkulatzen direla eta $\mathcal{P}(A|B) = \frac{\mathcal{P}(\mathcal{K})}{\mathcal{P}(B)}$ dela, eta, beraz, ez dela berriz ere probabilitateen kalkulua egin beharko. Adibidea

2.3.4 Lift-a

Aurreko 2.3.1 atalean definitutako kontzeptuetatik *Lift* elementua zertarako balio duen ikustea falta zaigu. Lehen esan bezala, *Lift*-a multzo baten A elementua dagoela jakinda multzo horretan B elementua egoteko konfiantzan dagoen segurtasuna zein den neurtzen du. Elementu honen balioa aztertzerakoan hiru kasu bereizi beharko dira: *Lift* < 1 kasua, *Lift* = 1

kasua eta *Lift* > 1 kasua. Aztertzen den kasuaren *Lift*-a hartzen duen balioa 1 baino baxuagoa bada aztertutako elementuetako baten agerpenak bigarrenaren agerpenean eragin negatiboa duela esan nahiko du. Bestalde, *Lift*-a 1 balioa hartzen badu elementuen presentzia independentea dela esan nahiko du. Azkenik, 1 baino balio altuagoa hartuz gero elementuak batera azaltzeko joera dutela esango du.

2.4 Content-based-filtering eta Collaborative filtering

Teknika hauek objektu edo produktu batekiko erabiltzaileek duten preferentzia edo balorazioa aurresaten saiatzen diren teknikak dira. Mota honetako teknikak, gaur egun oso ezagunak diren *Amazon, Netflix* edo *Youtube* moduko web-orrialdetan erabiltzen dira erabiltzaileen gustuak jakinda produktu, serie edo bideo berriak gomendatzeko.

Sistema hauek bi adar nagusitan banandu daitezke: *Content-based filtering* eta *Collaborative filtering*.

2.4.1 Content-based filtering

Metodo mota hauek produktuen deskripzioetan eta erabiltzaileen gustuetan oinarritzen dira. Hemen hitz gakoak produktu desberdinak deskribatzeko erabiltzen dira eta erabiltzaileen gustuak hitz gako hauen arabera ezartzen dira. Horrela, erabiltzaileak produktu bat ondo baloratzen badu (edo erosten badu), algoritmoak produktu horren antzeko ezaugarriak dituzten beste batzuk gomendatuko ditu.

Content-based filtering moduko algoritmoak garatzeko hainbat modu desberdin dauden arren, ondoren problema modu analitiko batean ebazteko algoritmo bat azalduko da. Hala ere, mota honetako problemak *bayesian classifier, cluster analisis, decision trees* edota *sare neuronal*-en bitartez ebatz daitezke, emaitza sinpleagoak lortuz.

2.4.1.1 Algoritmoa

Demagun ondoko taulan erabiltzaile desberdinek hainbat pelikulei emandako balorazioak ditugula:

Izena\pelikula	Eraztunen Jauna	Harry Potter	Star Wars	American pie	Deadpool
Ane	0	?	3	?	4
Leire	2	0	?	5	?
Aitor	?	5	3	2	5
Jon	5	4	4	?	4

Ohartu taulak elementu galduak dituela, arraroa baita pertsona guztiek pelikula guztiak ikustea. Hortaz aparte, demagun ondoko taulan pelikulen ezaugarriak laburtzeko erabilitako parametroak ditugula:

Pelikula\Ezaugarria	Akzioa	Fantasia	komedia
Eraztunen Jauna	0.8	0.9	0.2
Harry Potter	0.6	0.9	0.3
Star Wars	0.6	0.8	0.1

American pie	0.01	0	0.9
Deadpool	0.8	0.	0.8

Demagun N_1 erabiltzaile eta N_2 pelikula desberdinez osatutako datu basea aztertu behar dugula. Orduan bi matrize izango ditugu. Alde batetik, $N_1 \times N_2$ dimentsioko matrize bat izango dugu, i erabiltzaile bakoitzak j pelikula bakoitzari emandako y_{ij} balorazioarekin. Bestetik, $N_2 \times d$ matrizea izango dugu film bakoitzaren x_j ezaugarriekin, $d > 0$ balio arbitrario bat izanda.

Algoritmo honen helburua i erabiltzaile bakoitzari dagokion θ_i gustuak deskribatuko dituen bektorea lortzea da. Behin bektore hauek lortuta erabiltzaileek pelikulei emandako balorazioa $(\theta_i)^T x_j$ moduan berreskuratu ahalko da.

Pertsona bakoitza modelatzen duen θ_i bektoreak lortzeko, beste metodoetan egin den moduan, errorea kalkulatzen duen balio funtzio bat minimizatu beharko da:

$Cost(\theta) = \frac{1}{2}\sum_{i=1}^{N_1}\sum_{j:r(i,j)=1}^{N_2}\left((\theta_i)^T x_j - y_{ij}\right)^2$ non i erabiltzaileak j pelikulari balorazioa eman badio $r(i,j) = 1$ izango den. Hemen ere, ereduak aztertzen dituen datuak ez ikasteko erregularizazioa parametro bat esleituko zaio, ondoko funtzioa lortuz:

$Cost(\theta) = \frac{1}{2}\sum_{i=1}^{N_1}\sum_{j:r(i,j)=1}^{d}\left((\theta_i)^T x_j - y_{ij}\right)^2 + \frac{\lambda}{2}\sum_{1}^{N_1}\sum_{k=1}^{d}\theta_{ik}^2.$

Balio funtzio honen minimoa aurkitzeko lehen azaldutako *gradient descent* teknika erabili daiteke.

2.4.2 Collaborative filtering

Content-based filtering metodoen kasuan ez bezala, mota honetako teknikek erabiltzaileen aktibitatea, preferentziak eta testuingurua kontuan hartzen dituzte, hauen arteko antzekotasunak bilatuz. Hau eginda, antzekoak diren erabiltzaileak antzeko produktuak gustuko dituztela suposatzen da. Metodo mota hauetan, lehenago ikusitakoekin ez bezala, produktuen ezaugarriak ez dira aztertzen eta, beraz, pelikulak bezalako objektu konplexuak gomendatzeko gai dira hauen ezaugarriak jakin gabe.

2.4.2.1 Algoritmoa

Kasu honetan, aurreko tauletatik lehenengoarekin nahikoa izango litzateke, hau da:

Izena\pelikula	Eraztunen Jauna	Harry Potter	Star Wars	American pie	Deadpool
Ane	0	?	3	?	4
Leire	2	0	?	5	?
Aitor	?	5	3	2	5
Jon	5	4	4	?	4

Mota honetako tekniketan, pertsona bakoitzaren θ_i gustuez aparte, pelikula desberdinen x_j ezaugarriak ere modelatuko dira. Horretarako, ondoko balio funtzioa minimizatu beharko da:

$Cost(\theta, x) = \frac{1}{2}\sum_{i=1}^{N_1}\sum_{j:r(i,j)=1}^{N_2}\left((\theta_i)^T x_j - y_{ij}\right)^2$ non i erabiltzaileak j pelikulari balorazioa eman badio $r(i,j) = 1$ izango den. Lehenago ikusi den moduan, ereduak aztertzen diren datuak ez ikasteko, erregularizazioa parametro bat esleituko zaio, ondoko funtzioa lortuz:

$$Cost(\theta, x) = \frac{1}{2}\sum_{i=1}^{N_1}\sum_{j:r(i,j)=1}^{d}\left((\theta_i)^T x_j - y_{ij}\right)^2 + \frac{\lambda}{2}\sum_{1}^{N_1}\sum_{k=1}^{d}\theta_{ik}^2 + \frac{\lambda}{2}\sum_{j=1}^{N_2}\sum_{k=1}^{d}x_{jk}.$$

Aurreko kasuan bezala, hemen ere minimoa *gradient descent* teknikaren bitartez lor daiteke.

Pelikulen adibidearekin jarraituz, gomendioak egiteko garaian bi kasu desberdin bereziko ditugu. Alde batetik, θ_i elementu desberdinak konparatuz antzeko gustuak dituzten erabiltzaileak aurki daitezke. Demagun i_1 eta i_2 erabiltzaileak antzeko gustuak dituztela ($\left\|\theta_{i_1} - \theta_{i_2}\right\|_2 \sim 0$), orduan logikoa litzateke i_1-ek gustuko dituen pelikulak i_2-ri ere gustatzea eta alderantziz. Bestetik, pelikulen ezaugarriak jakinda eta i_1 erabiltzaileak j_1 pelikula gustuko duela jakinda, pelikula horren antzerako pelikulak ere gustuko dituela suposa daiteke. Beraz, edozein j pelikula j_1-en antzerakoa bada gomenda daiteke, hau da, $\left\|x_j - x_{j_1}\right\|_2 \sim 0$ betetzen duen edozein pelikula i_1 erabiltzaileari gomenda dakioke.

Gure sistema duen plataformara erabiltzaile berri bat sartzerakoan, honen gustuen daturik ez da egongo eta, beraz, ezin izango dugu beste erabiltzaileekin konparatu. Adibidez, aurreko adibideekin jarraituz, erabiltzaile berri bat sartzerakoan demagun ondoko egoeran gaudela:

Izena\pelikula	Eraztunen Jauna	Harry Potter	Star Wars	American pie	Deadpool
Ane	0	?	3	?	4
Leire	2	0	?	5	?
Aitor	?	5	3	2	5
Jon	5	4	4	?	4
Jone	?	?	?	?	?

Kasu honetan, ezinezkoa da erabiltzaile berriaren gustuak inferitzea, informaziorik ez baitugu. Mota honetako oztopoak gainditzeko aukera posibleetatik bi aipatuko ditugu. Lehena eta egiten sinpleena, pelikulen balorazioen batez bestekoak erabiltzaile berriaren balorazio moduan kontuan hartzea da. Hau eginda, hasiera batean erabiltzaile gehienek gustuko dituzten produktuak gomendatuko dira. Bezero berriak produktuak baloratzen doan heinean, gomendioak pertsonalagoak izango dira. Bigarrena, erabiltzaile berri bat sartzerakoan, produktu sorta baten balorazioa ematea da. Horrela, bere gustuak modelizatzeko beharrezko hasierako datuak izango ditugu.

2.4.3 Teknikak hobetzen: Metodo hibridoak

Teknika mota biak bakarka emaitza onak eman arren, metodo hauek konbinatuz lortutako emaitzak hobetzen direla ikusi da.

2.4.4 Kontuan hartu beharreko puntuak

- Aztertuko diren datuek gehienbat datu galduak izango dituzte, oso arraroa da erabiltzaile guztiek produktu guztiak aztertu izana.
- Gomendioak egiterako garaian beharrezkoa da erabiltzaileen artean korrelazioa egotea.

- Erabiltzaileen balorazioez aparte beste datu batzuk kontuan har daitezke: adina, nazionalitatea…

2.5 Markov Models

Orain arte ikusitako ereduak eratzeko aztertutako aldagaiak I.I.D (askeak eta berdinki banatuak) zirela onartzen zen. Hala ere, bizitza errealean badaude hainbat kasu non I.I.D hipotesia ezin izango den bete, adibidez, orduko egongo diren prezipitazioak, grabatutako audio baten ezaugarri akustikoak edo eguneko moneta aldaketa ratioak modelatzerakoan, argi ikusten da hurrengo balio aurreko balioen araberako izango dela.

Kasu hauetan, ezingo lirateke aurretik ikusitako teknikak erabili. Mota honetako aldagaiekin lan egiteko *Markov Models* deituriko ereduak erabiliko ditugu. Ondoren, *Marko Chains* eta *Hidden Markov Models* aztertuko dira.

2.5.1 Markov chain

Probabilitate teorian Markov-en prozesua Markov-en propietatea betetzen duen prozesua da. Propietate hau "memorylessness" moduan ere ezagutzen da eta gertakizun kate batean etorkizuneko gertakizunak iraganeko gertakizunekiko independenteak direla esan nahi du. Horrela, etorkizuneko momentu bat modelatzeko, momentuko egoera bakarrik aztertu beharko da.

Modu matematikoago batean esanda, $(x_1, \dots, x_n)$ egoera desberdinen segida badugu, s_t elementuaren egoera s_{t-1} elementuaren egoeraren menpekoa bakarrik izango da, hau da:

$$P(x_t|x_1, x_2, \dots, x_{t-1}) = P(x_t|x_{t-1})\ \forall i \in \{1, \dots, n\}$$

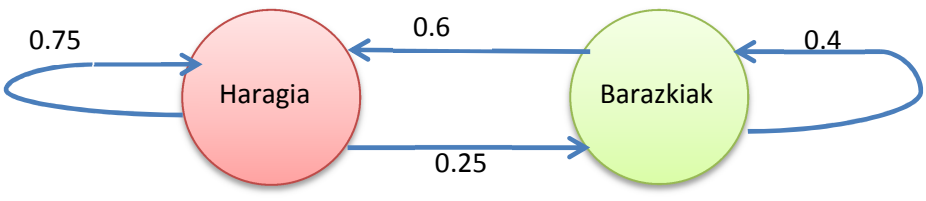
izango da.

Markov kateen adibide ezagun bat "mozkorraren ibilera" moduan ezagutzen da. Demagun zenbaki arrunten (0,1,2,….,n) zuzenetik dabilela eta dagoen zenbakitik hurrengo zenbakira joateko probabilitate eta aurrekora joatekoa 0.5-eko dela. Orduan, 5 zenbakitik 4 zenbakira joateko probabilitatea 0.5-ekoa izango litzateke, 5 zenbakitik 6 zenbakira joateko probabilitatea bezala. Ohartu probabilitate hauek aurreko egoerekiko independentea dela, hau da, ez du inporta 5 zenbakira 6 edo 4 zenbakietatik ailegatu den.

Beste adibide sinple moduan demagun animali omniboro baten dieta aztertzen gaudela. Animalia hurrengo egunean haragia edo barazkiak jango dituen jakin nahiko genuke. Demagun ere ondoko probabilitateen taula dugula:

Gaur/Bihar	Haragia	Barazkiak
Haragia	0.75	0.25
Barazkiak	0.6	0.4

Orduan ondoko Markov-en Katea edukiko genuke:



Egoera batetik bestera pasatzeko probabilitateek *trantsizio matrizea* deituriko matrizea osatzen dute. Kasu honetan ondoko matrizea edukiko genuke:

$$M = \begin{pmatrix} 0.75 & 0.25 \\ 0.6 & 0.4 \end{pmatrix}.$$

Matrize honen M_{ij} elementu bakoitzak i egoeratik j egoerara pasatzeko probabilitatea adierazten du. Kasu orokorrago batean, demagun 6 egoera desberdin daudela, orduan ondoko matrizea izango genuke:

$$M = \begin{pmatrix} p_{11} & p_{12} & p_{13} & P_{14} & p_{15} & p_{16} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & p_{26} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} & p_{36} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} & p_{46} \\ p_{51} & p_{52} & p_{53} & p_{54} & p_{55} & p_{56} \\ p_{61} & p_{62} & p_{63} & p_{64} & p_{65} & p_{66} \end{pmatrix}.$$

Ohartu matrize honen lerro desberdinen elementuen batura beti 1 izango dela.

Matrize hau hasiera batetik definituta etor daiteke, aurreko adibidean bezala, edo bakoitzak definitu beharko du. Horretarako, egiantza handieneko metodoa erabil daiteke. Demagun, n elementuko segida bat dugula eta i egoeratik j egoerara pasatzeko probabilitatea (M_{ij} izango dena) jakin nahi dugula. Orduan M_{ij} elementurako, ditugun n elementuetatik i egoeratik j egoerara zenbatetan igarotzen da zati i egoeran dauden elementu kopurua har dezakegu, hau da:

$$M_{ij} = \frac{\sum_{k=0}^{k=n-1} \mathbb{I}(x_k = i, x_{k+1} = j)}{\sum_{k=0}^{k=n-1} \mathbb{I}(x_k = i)}.$$

Matrize hau oso garrantzitsua da eredu osoa definitzen baitu. Aztertutako elementua w_t egoera batean badago eta hurrengo momentuan zein w_{t+1} egoeratan egongo den jakin nahi badugu nahikoa izango da $w_{t+1} = w_t M$ kalkulatzea. Kontzeptu hau orokortuz, hasierako w_0 egoera batetik abiatuz t momentu pasa eta gero elementua duen egoera jakin nahi izanez gero, nahikoa izango da $w_t = w_0 M^t$ kalkulatzea.

Aplikazio pare baten adibideak ikusi baino lehen hauek ulertzeko garrantzitsua izango den azkeneko kontzeptu bat ikusiko dugu, *Stationary Distribution. Stationary Distribution* deiturikoa hasierako egoera batetik infinitu aldiz mugitu ostean lortzen den egoera da, hau da, $w_\infty = \lim_{t\to\infty} w_t$. *Stationary Distribution*-a existituko da baldin eta soilik baldin ondoko bi baldintzak betetzen dira:

1. Edozein egoeratik edozein egoeratara heldu daiteke.
2. Egoera sekuentziek ez dituzte inolako buklerik.

Ohartu orain arte Markov kateen artean sinpleenetariko bati buruz hitz egin dugula, *first order Markov chain* deituriko. Hemen, egoera berriak inferitzeko aurreko egoera bakarrik kontuan hartzen da. Hala ere badaude eredu konplexuagoak, egoera gehiago kontuan hartzen dituztenak *m-order-Markov-chain*.

2.5.1.1 Aplikazioen adibidea: Ranking-ak

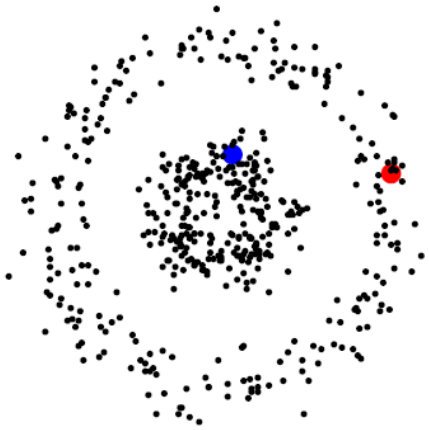
Markov-en kateek objektuen rankingak egiteko balio dute. Hemen, aztertuko diren datuak objektuen arteko konparaketak izango dira. Adibide moduan, kirol talde edo kirolarien rankingak egin daitezke. Kasu hauetan, helburua objektuak “onenetatik” “txarrenetara” ordenatzea izango litzateke.

Rankingak egiteko trantsizio matrize bat eratuko da non egoera bakoitza talde/kirolari bat den. Orduan, S*tationary distribution*-a lortzerakoan bilatzen ari garen emaitza izango dugu.

Matrizea eraikitzeko momentuan galtzen duten taldeetatik irabazten duten taldeetara joatea bermatuko da. Hau da, A taldeak B taldeari irabazten badio, B->A probabilitatea A->B probabilitatea baino altuagoa izango da. Lotura hau gogorragoa izan daiteke partidaren markagailuaren arabera. Bukatzeko, matrizearen lerroak normalizatzen dira hauen batura 1 izateko. Hau eginda, Markov-en katearen *stationary distribution*-a bilatu egiten da eta lortutako bektorearen balioak taldeen arteko rankinga emango digu. Probabilitate handien dituzten egoerak talderik onenak izango dira eta probabilitate baxuenekoak, aldiz, txarrenak.

2.5.1.2 Aplikazioen adibidea: Klasifikazioa

Demagun datu asko baina datuen kopuru txiki baten etiketak bakarrik ditugula eta datuak duten egituraz baliatuz etiketarik gabeko elementuen etiketak jakin nahi ditugula. Datuak egituratuta badaude, Markov-en kateen bitartez elementuen etiketa inferitu daiteke. Adibidez, eskumako irudian datu kopuru handia dugu (puntu beltzak) etiketaturiko bi punturekin (elementu urdina eta gorria). Kasu honetan, badirudi datuek egitura finko bat dutela (puntu urdinak zentroan eta gorria eraztunean), beraz, Markov-en kateez balia gaitezke.



Hemen ere trantsizio matrizea eratu egin behar da, hurbilen dauden puntuen artean mugitzeko probabilitatea handiagoa emanez. Kasu honetan, etiketaturiko elementu batera ailegatzean bertan geratuko da, hau da, x_i etiketatutako elementu bat bada $M_{ii} = 1$ izango da. Puntu hauek absortzio egoerak deituko dira.

Azkenik, etiketa gabeko puntuen etiketa jakiteko puntu horietatik abiatuz eta trantsizio matrizea aplikatuz lortzen den lehenengo absortzio egoerak aztertuko da. Bukatzeko, egoera horrek duen puntuaren etiketa hasierako puntuaren etiketa izango da.

2.5.2 Hidden Markov Models

Orain arte ikusitako adibideetan behaketak neurgarriak ziren. Hala ere, hori ez da zertan beti gertatu behar. Batzuetan baliteke aztertuko diren egoerak neurgarriak ez izatea edo hauek neurtzeko aukera ez izatea. Kasu horietan lagungarria lirateke *Hidden Markov Models* erabiltzea. Hemen, neurgarriak diren obserbazio sekuentzia batetik abiatuz $\{x_1, \dots, x_n\}$ aztertuko diren egoerak lortzen dira.

Adibide teoriko moduan, demagun Ane eta Jon elkarrengandik urrun bizi diren bi lagun direla. Demagun ere telefonoz egunero hitz egiten dutela. Jonek eguraldiaren arabera hiru ekintza nagusi egiten ditu arratsaldeetan, kirola, erosketak edo ordenagailuan ibili. Demagun Jonek egun bakoitzean egindako ekintza Aneri kontatzen diola. Datu horiekin, Anek egun bakoitzean egindako eguraldia jakin nahiko luke, bi gertaeren artean erlazioa argia ikusten baitu. Hau *Hidden Markov Model* aplikatzeko egoera aproposa litzateke. Anek bere lagunak egindako ekintzak dakizkien arren ez dauka egun bakoitzeko eguraldiaren informazioa. Demagun, Anek, Jonen herriaren eguraldiaren joera ondoko probabilitate banaketa jarraitzen duela dakiela:

Gaur/Bihar	**Eguzki**	**Euri**
Eguzki	0.6	0.4
Euri	0.3	0.7

Eguzki	**0.4**
Euri	0.6

Gainera, Jonek ondoko ohiturak dituela ere badakiela:

Gaur/Bihar	**Kirola**	**Erosketak**	**Ordenagailua**
Eguzkia	0.6	0.3	0.1
Euri	0.1	0.2	0.7

Datu guzti hauekin Anek Jonek egun bakoitzean egindako aktibitate jakinda egun horretako eguraldiaren probabilitatea jakiteko informazioa izango du.

Ondoko bi artikuluetan pare bat adibide praktiko ikusi ahalko dira. Lehenengoan, eredu hauek medikuntza arlora aplikatuta lortutako emaitzak ikus daitezke. Bigarrenean, seinu hizkuntzaren keinuak duten esanahia inferitzen saiatzen dira.

Metodo honen kasuan hiru osagai nagusi izango genituzke:

1. M_t trantsizio matrizea, egoera batetik bestera aldatzeko probabilitateak dituena.
2. M_e emisio matrizea, elementu bakoitza egoera bakoitzetik sortua izateko probabilitatea adierazten duena.
3. π hasierako egoeren probabilitateen banaketa.

Aurreko adibidean ondoko matrizeak izango genituzke:

$$M_t = \begin{pmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{pmatrix}; M_e = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.2 & 0.7 \end{pmatrix}; \pi = (0.4 \quad 0.6).$$

Datu hauek aurretik emanda egon daitezke, adibidean ikusi den moduan adibidez. Hala ere, beste kasu batzuetan inferitu behar izango dira. Aztertutako kasu hau, bereziki, *Hidden Markov Model diskretua* izango litzateke, egoerak diskretuak baitira. Bukatzeko, teknika honek hiru egoera nagusien aurrean erabili ohi da:

1. Obserbazio sekuentzia bat $\{x_1, \dots, x_n\}$ eta $\{\pi, M_t, M_e\}$ elementuetatik abiatuz $\{s_1, \dots, s_n\}$ egoera bakoitza gertatzeko probabilitatea jakin nahi denean. Horretarako, *forward-backward* algoritmoa erabiltzen da.

2. Obserbazio sekuentzia bat $\{x_1, \dots, x_n\}$ eta $\{\pi, M_t, M_e\}$ elementuetatik abiatuz, probableen den egoera sekuentzia $\{s_1, \dots, s_n\}$ lortu nahi denenan. Horretarako *Viterbi algoritmoa* erabiltzen da.
3. Obserbazio sekuentzia $\{x_1, \dots, x_n\}$ batetik abiatuz ereduaren $\{\pi, M_t, M_e\}$ elementuak lortu nahi denean. Horretarako, *egiantza handieneko metodoa* erabiltzen da.

Bukatzeko, hemen azaldutako eredua diskretua bada ere, badago metodo hau egoera espazio jarraitu batean aplikatzea, adibidez, egoera denbora izanda.

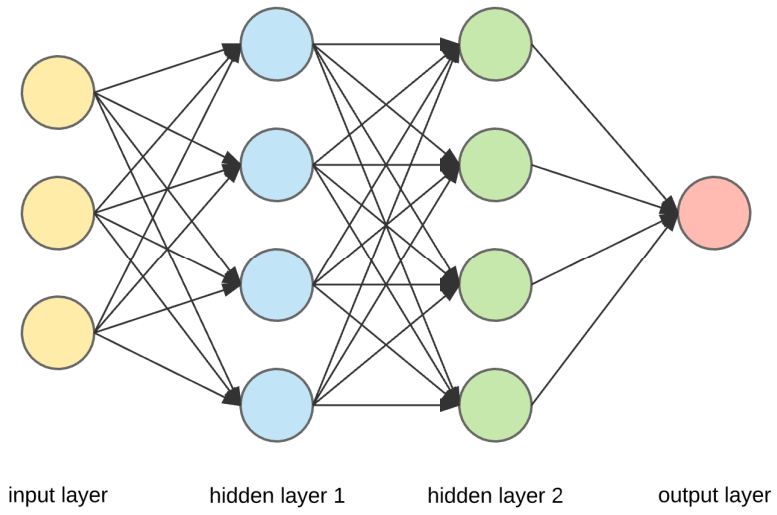
3.Sare neuronalak

Sare neuronalak animalien neuronen sarearen funtzionamendua imitatzen saiatzen diren teknikak dira. Sare hauek, naturan gertatzen den moduan, neurona deituriko interkonektatutako nodo desberdinez osatuta daude. Teknika hauek gehienbat "supervised learning" motatako eginkizunetarako erabiltzen badira ere, badaude "unsupervised learning" motatako lanak egiteko gai direnak. Hortaz aparte, robotikaren munduan ere erabili ohi dira, kotxe autonomoen garapenean adibidez.

Sare neuronalen egitura, normalean, hiru zati nagusietan banatzen da:

1. *Sarrerako geruza*: "Input layer" deiturikoa, hemengo nodoetan hasierako datuak sartuko dira (balio kategorikoak ,jarraituak, argazkiak, testua, etab.).
2. *Ezkutuko geruza*: "Hidden layers" deiturikoa, hemen sarrerako datuetatik habiatuz beharrezko konbinazioak egin eta funtzio desberdinak aplikatzen dira.
3. *Irteera geruza*: "Output layer" deiturikoa, irteerako balioak dituzten neuronez osatuta dago (kategoriak, balio jarraituak, argazkiak, etab.). Helburuaren arabera aurreko ezkutatutako kapen konbinazio lineala edo honi aplikatutako funtzio bat izango du.

Oinarrizko sare neuronaletan, geruza batetik besterako bidean, aurreko geruzaren neuronen konbinazio lineal desberdinak egiten dira, *pisu* deituriko balio eskalarrak erabiliz. Ondoren, nodo bakoitzean *step-function* edo *activation function* deituriko funtzio bat konbinazioan lortutako balioari aplikatzen zaio, neurona bakoitza aktibatzen den edo ez edo neurona bakoitzaren eragina jakiteko. Ondoren oinarrizko sare neuronal baten egitura ikus daiteke:

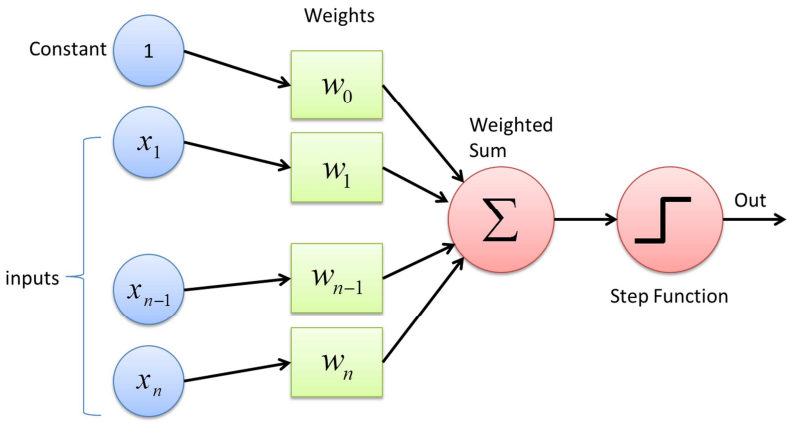


Aurreko irudiko sare neuronalak ondorengo prozesua jarraituko luke:

1. Hasierako kapan gure datuen lehenengo elementuaren aldagaiak sartuko lirateke $X_1 = (x_{11}, x_{12}, x_{13})$.
2. Lehenengo ezkutuko geruzaren $j \in \{1, \dots, 4\}$ nodo bakoitzeko, w_{j1}, w_{j2} eta w_{j3} pisuak egongo lirateke.
3. Nodo bakoitzean $y_{1j} = f_j(w_{j1}x_{11} + w_{j2}x_{12} + w_{j3}x_{13})$ aurreko geruzaren balioen konbinazio lineal bati f_j*aktibazio funtzioa* aplikatuko litzaioke.

4. Bigarren ezkutuko geruzan aurreko geruzaren prozesu berdina jarraituko zen, k neurona bakoitzeko w_{k1}, w_{k2} eta w_{k3} pisu eta $f_k$aktibazio funtzio berriekin, y_{2k} balioak lortuz.
5. Azkenik, aurreko kasuen moduan, irteera geruzara azkeneko ezkutuko geruzaren balioen konbinazio lineal bati azkeneko aktibazio funtzioa aplikatuz lortutako balioa esleituko litzaioke, w_1, w_2, w_3 pisu eta f aktibazio funtzioak erabiliz eta y irteera balioa lortuz.

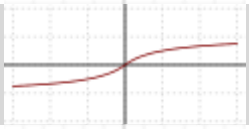
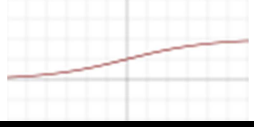
Sare neuronal sinpleenetarikoen artean 1. atalean ikusitako *Perceptron* algoritmoa,E*rregresio Lineal Sinplea* edota *Erregresio Logistiko Sinplea* egongo lirateke. Hauek, ondorengo irudiko egitura izango lukete, aktibazio funtzio desberdina aplikatuz.



Perceptron-en kasurako seinu funtzioa erabiliko genuke. E*rregresio Lineal Sinple*-rako, aldiz, identitate funtzioarekin nahikoa izango genuke edota geroago ikusiko den *ReLU* funtzioarekin (gure etiketatutako datuek balio negatiboak ez dituztenean). Azkenik, *Erregresio Logistiko Sinple*-rako, *sigmoide* funtzioa erabiltzearekin nahikoa genuke.

3.1 Aktibazio funtzioak

Ikusi den moduan, aktibazio funtzioaren aukeraketak bukaerako ereduan eragin handia dauka. Hain zuzen ere, klasifikaziorako balio duen eredu batetik erregresiorako balio duen eredu batera pasatzea ahalbidetzen gaitu. Ondoren, funtzio nagusi batzuk aipatuko dira:

Funtzioaren izena	*Funtzioa*	Grafikoa
Identity	$f(x) = x$	
Binary step	$f(x) = \begin{cases} 0 \; non \; x < 0 \\ 1 \; non \; x \geq 0 \end{cases}$	
Softsign	$f(x) = \frac{x}{1 + \lvert x \rvert}$	
Sigmoid (Logistic)	$f(x) = \frac{1}{1 + e^{-x}}$	

ReLU(Rectified linear unit)	$f(x) = \begin{cases} 0 \; non \; x < 0 \\ x \; non \; x \geq 0 \end{cases}$ 

Adibide guzti hauek nodo bateri bakarrik eragiten diete, baina badira geruza berdineko nodo guztietara sartzen den informazioa kontuan hartzen duten funtzioak; adibidez:

Funtzioaren izena	*Funtzioa*
Softmax	$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{k=1}^{n} e^{x_k}}$

Azken funtzio hau besteak beste klase anizkoitzen klasifikaziorako erabiltzen da.

Aktibazio funtzioekin bukatzeko, ohartu hemen azaldutako funtzioak existitzen diren funtzioen portzentaje txiki bat direla. Hortaz aparte, posiblea litzateke funtzio propioak erabiltzea, bai asmatutakoak, baita ikusitako baten bariazioak ere.

3.2 Sarrerako eta irteerako datuak

Esan dugun moduan sare neuronalek zenbakiekin lan egiteaz gain, aldagai kategorikoekin, argazkiekin edota testuarekin lan egiteko gai dira. Hala ere, kalkuluak egin ahal izateko zenbakiak behar direnez, aldagai mota guzti hauek transformatu beharko dira.

Aldagaiak kategorikoak badira, bi kategoria daudenean hauei 1 eta 0 balioak eman dakieke. Bestalde, n badaude, n aldagai berri sortzearekin nahikoa litzateke 1 edo 0 balioekin. Aldagai bakoitza kategoria bat definituz.

Aldagaiak argazkiak direnean bi kasu bereiz genituzke. Alde batetik, kolorea garrantzitsua ez denean argazkia gris eskalara pasa daiteke. Horrela, argazkia pixelez osatutako $n * m$-ko matrize bat izango litzateke. Orduan, matrizea nm elementuko bektore bihurtuko zen, elementu bakoitza aldagai bat izanda. Bestetik, kolorea garrantzitsua denean aurreko gauza bera egin daiteke baina, kasu honetan, $n * m * 3$ dimentsioko matrize bat izango genuke eta lortutako hiru bektoreak bata bestearen atzean jarriko genituzke.

Azkenik, testuarekin lan egiterakoan hainbat aukera egongo lirateke. Alde batetik, agertzen den hitz bakoitzeko aldagai bat sor daiteke. Hemen, aztertzen diren esaldietan, hitz bakoitza zenbat aldiz agertzen den aztertuko litzateke. Bestetik, agertzen diren hitzen sustraiak atera (batetik -> bat esaterako) daitezke. Hau egiterakoan, sustrai bakoitzeko aldagai bat egitearekin nahikoa litzateke, 1 edo 0 balioa emanez.

3.3 Sareak entrenatzen

Sare neuronal bat eratzerako orduan, lehenik eta behin, honen egitura aukeratu behar da. Hau da, zenbat geruza eta zenbat nodo erabiliko diren aukeratu behar da. Honetarako ez dago erantzun finkorik. Helburuaren arabera baliteke aurretik finkatutako konfigurazio batzuk aztertuta jada egotea baina gehienetan froga desberdinak egin beharko dira egitura egokia aurkitu arte. Estruktura bat hautatu eta gero erabiliko diren pisuak definitu behar dira. Horretarako, erabiliena den metodoa 1. ataleko metodo batzuetan erabilitako *Gradient*

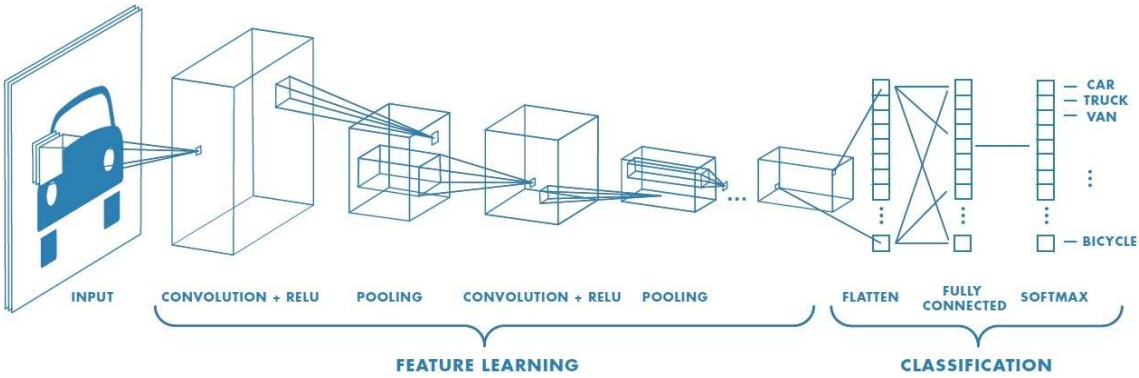
Descent teknikaren oso antzekoa da, *Backpropagation* deiturikoa. Teknika honetan hasieran pisuak ausaz aukeratzen dira eta datuei sarea aplikatzen zaie. Ondoren, lortutako emaitza (hasiera baten seguruenik txarra izango dena) emaitza errealarekin konparatu egiten da eta, errorearen arabera, pisuak egokitzen dira. Metodoa modu sakonago eta bisualago batean ondoko estekan ikus daiteke.

3.4 Beste sare neuronal mota batzuk

Orain arte aztertutako sareak oinarrizkoenak lirateke, *Multilayer FeedForward fully conected Neural Networks* moduan ezagutzen direnak. Hala ere, badira beste sare mota asko. Ondoren beste 4 sare familien adibideak ikusiko dira.

3.4.1 Convolutional Neural Networks: CNN

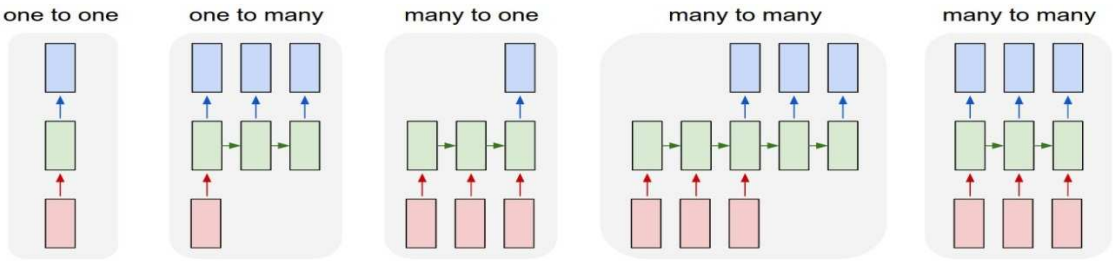
Argazkiekin lan egiterakoan, klasifikaziorako normalean, interesgarria da argazkien ezaugarriak detektatzeko gai den eredu bat izatea, hau da, aurpegiak detektatzen duen eredu bat nahi izatekotan, ereduak belarriak, begiak, ezpainak etab. dauden edo ez jakitea baliagarria litzateke. Hori da hain zuzen ere mota honetako sareen eginkizuna.



Sare mota hauek aurretik aipatutako geruza motez aparte beste bi mota berri erabiltzen ditu: *Convolutional layers* eta *Pooling layers*. Lehenengoak argazkiei filtro desberdinak pasatzen die hauen ezaugarriak isolatzeko asmoz eta bigarrenak aztertzen diren ezaugarrien dimentsioa txikitzen du gero eta zehatzagoa izan dadin. Normalean, mota honetako sareetan aipatutako bi geruza horiek erabiltzen dira ezaugarrien eragina ateratzeko eta gero, klasifikazioa egiteko helburuarekin, hasieran azaldutako sareak erabiltzen dira.

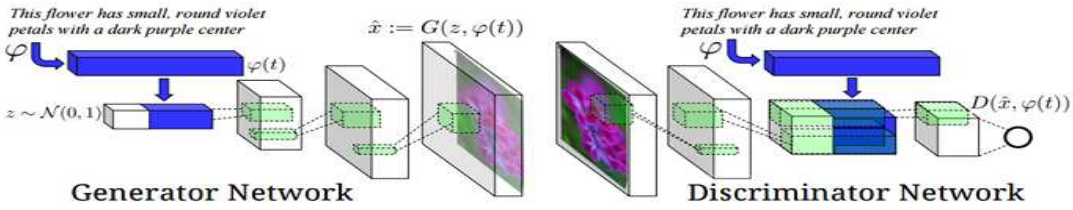
3.4.2 Recurrent Neural Networks

Batzuetan ereduak “memoria” izatea interesatzen da, hau da, ereduak lehenago lortutako emaitzak hurrengo datuetarako kontuan izatea. Kasu horietan mota honetako sareak erabiltzen dira. Normalean, testua edo soinua sortzeko edo aztertzeko erabiltzen diren sareetan erabili ohi dira, hemen testuinguruak garrantzi handia baitu.



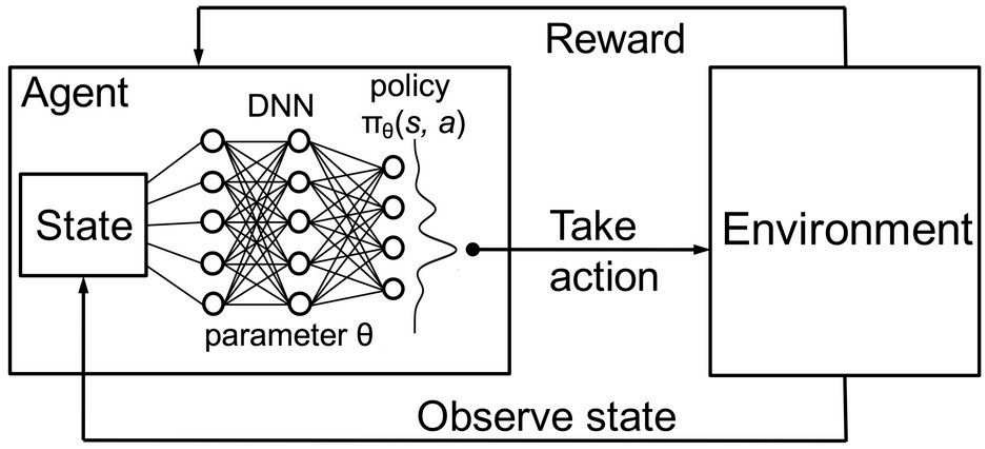
3.4.3 Generative Adversarial Nertworks

Bere izenak dioen moduan, mota honetako sareek lehiaketaren bitartez sortzen diren sareak dira. Normalean, bi sare desberdinez osatuta daude, *Generative* eta D*iscriminative* deiturikoak. Hauen helburua, kasu askotan, data sortzea izaten da. Lehenengo motako sareek data sortzen ikasten dute eta bigarren motakoek data benetakoa den edo asmatutakoa den bereizten saiatzen dira. Horrela, datuak gero eta hobeago sortzerakoan *discriminative* den sareak gero eta emaitza txarragoak emango ditu, eta beraz, zuzenketa gogorragoak izango ditu bere pisuetan. Bestela, justu kontrakoa gertatuko litzateke eta *generative* dena zuzenketa nabarmenagoak izango lituzke.



3.4.4 Deep Reinforcement Learning

Azken mota honetako sareek ingurumenetik ikasten dute. Normalean robotikan erabili ohi dira eta esperientziatik hobetzen duten ereduak dira. Hasiera batean ausazko portaera bat izango dute eta gero eta saiakera gehiago egin orduan eta emaitza hobeagoak izango dituzte.



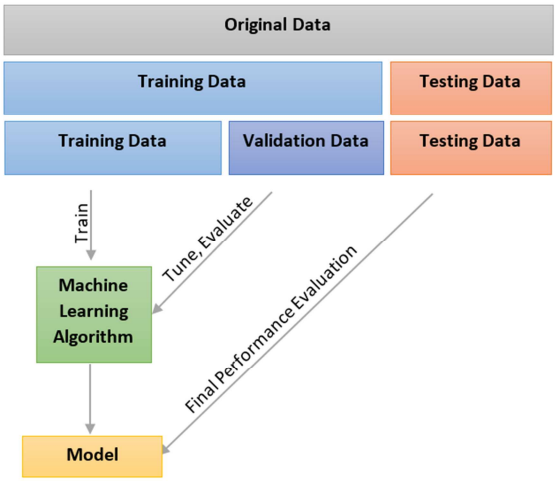
Ondorengo bideoetan aipatutako Convolutional Neural Networks, Recurrent Neural Networks, Generative Adversarial Netwroks eta Deep reinfocement Learning metodoen adibideak ikus daitezke.

3.5 Abantailak vs Desabantailak

Bukatzeko, sareen abantailarik handiena ia edozer egiteko gai direla da. Gainera berrerabili daitezke, hau da, demagun katuak detektatzen dituen sare bat dugula baina txakurrak detektatu nahi ditugula. Orduan, sare berri bat zerotik hastea ez da beharrezkoa izango. Katuetarako erabiltzen den saretik abiatuz denbora gutxiagoan txakurrak detektatzen dituen eredu bat lortuko dugu. Bestalde, interpretatzen oso zailak dira, normalean ehunka edo milaka neurona erabiltzen dira geruza bakoitzean eta bakoitzak zer egiten duen jakitea zaila izaten da.

4. Entrenamendu, test eta balioztatze multzoak

Supervised Learning eta *Sare Neuronalen* kasuetan, ereduak garatzeaz aparte hauen kalitatea ere neurtu beharko da. Eginkizun honetarako ereduak garatzeko erabilitako datu berdinak erabili ezkero, kasu erreal batean (eredurako ezezagunak diren datuekin) baino emaitza hobeagoak lortzea gerta daiteke. Hori saihesteko, normalean, eskuragarri dauden datuak hiru multzotan banatzen dira: Entrenamendurako multzoa, balioztatze multzoa et test multzoa.



Hasteko, entrenamendurako multzoak eredua ikasteko erabiliko diren datuak izango ditu. Hauek aurreko orrietan azaldutako parametroak, normalean pisuak, ikasteko erabiltzen dira. Entrenamendurako multzoarekin batera balioztatzeko multzoa ere aukeratzen da. Bigarren multzo honen helburua ereduen *hyperparametroak* aukeratzea eta entrenamenduaren overfiting-a kontrolatzea izango da. Multzo hau, adibidez, sare neuronalen geruza kopurua edo aktibazio funtzioak aukeratzeko erabiltzen da. Sare neuronalen kasuan, sare mota desberdinak definituko lirateke, entrenamendurako multzoa erabiliz hauen pisuak kalkulatuz. Hau egin ostean, lortutako ereduak balioztatze multzoko datuetan aplikatu eta lortutako emaitzak aztertuko dira, hyperparametro hoberenak dituen sarea aukeratuz. Azkenik, test multzoa eredu finalaren kalitatea neurtzeko erabiltzen da. Datu hauetan eredua aplikatuko zen, aurretik finkatutako emaitzekin konparatuz.

Hiru multzo hauen erabileraren adibide moduan, sare neuronal baten kasuaren prozesua ondokoa litzateke :

1. Entrenamendu multzoko datuekin eredu finalerako hautagaiak izango diren sare desberdinak sortzen dira .
2. Balioztatzeko multzoko elementuak lortutako ereduetan aplikatzen dira.
3. Lortutako emaitzak aztertzen dira, hyperparametro egokienak dituen sarea aukeratuz. Hemen, eredua balioztatze multzoko datuetarako egokituta egotea gerta daiteke.
4. Balioztatze prozesuko ereduak test multzoan lortzen dituen emaitzak aztertzen dira.
   4.1. Accuracy, sensitivity, specificity, F-measure moduko neurriak erabiliz.
5. Lortutako emaitzak egokiak ez badira.
   5.1. Eredu ez da egokia izango.
   5.2. Eredu berriak garatu beharko dira.
6. Lortutako emaitzak onak badira.
   6.1. Eredu finala lortu da, kasu errealetan testatzeko prest dago.

4.1. Multzoak definitzen

Machine Learning-aren munduko egoera askotan gertatzen den moduan, multzoak aztertzen den problemaren arabera eta eskura dauden datuen arabera aukeratuko dira. Garatuko den

eredua hyperparametro gutxi baditu, adibidez, balioztatze multzo txiki batekin nahikoa litzateke. Kontrako kasuan, hauen kopurua handia bada, balioztatze multzoa handitzea gomendagarria litzateke. Bestalde, eskura dauden datuen kopurua txikia bada, baliteke datu guztiak eredua garatzeko behar izatea, kasu optimoa ez bada ere.

Hau kontuan hartuta, kasu idealean hasierako multzoa bitan banatzen da, eredua garatzeko multzoa eta test-erako multzoa. Hemen, ereduarentzako multzoa test multzoa baino handiagoa izatea gomendatzen da. Hasierako banaketa egin eta gero, ereduaren multzotik entrenamendurako eta balioztatzerako erabiliko diren multzoak lortzen dira. Bigarren pausu honetarako *Cross-validation* edo *balioztatze-gurutzatua* deituriko teknikak erabili ohi dira. Balioztatze-gurutzatu mota desberdinak daude eta hauek bi multzotan banan daitezke: b*alioztatze gurutzatu sakonak* eta *balioztatze gurutzatu ez sakonak*. Ondoren bi multzo hauetako metodo batzuk aipatu eta azalduko dira:

1. Balioztatze gurutzatu sakonak:
   1.1. Leave-one-out balioztatze gurutzatua
   1.2. Leave-p-out balioztatze gurutzatua
2. Balizotatze gurutzatu ez sakonak:
   2.1. K-fold edo k-hosto balioztatze gurutzatua
   2.2. Holdout metodoa
   2.3. Monte Carlo balioztatze gurutzatua

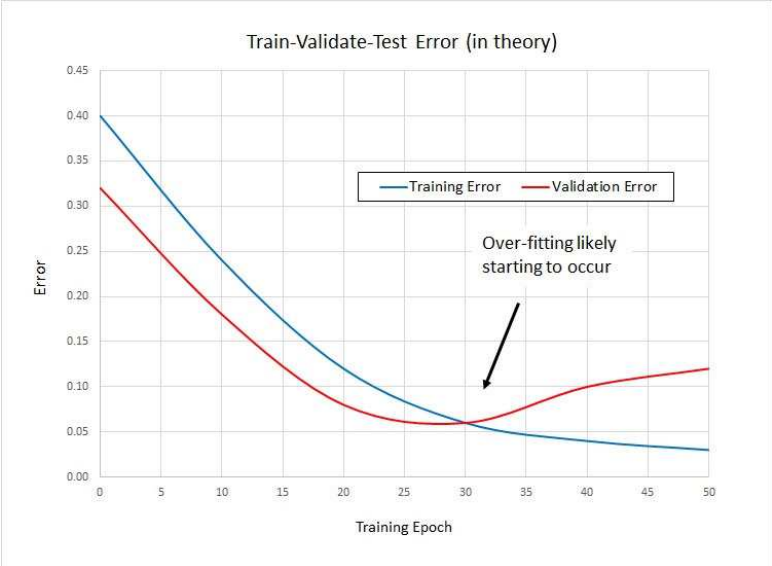
Balioztatze gurutzatu sakonek multzoan existitzen diren konbinazio guztiak kontuan hartzen dituzte. Leave-one-out kasuan, adibidez, eredurako erabiliko den multzoa n elementu baditu entrenamendurako $n-1$ hartuko ditu, faltatzen den elementua balioztatze prozesurako utziz. Gauza bera n aldiz egingo litzateke bakoitzean elementu desberdin bat balioztatzerako erabiliz. Leave-p-out metodoa, aldiz, aurreko kasuaren orokortasun bat da. Hemen, iterazio bakoitzean $n-p$ elementu entrenamendurako erabiliko dira p kasu balioztatze prozesurako utziz. Ohartu lehenengoaren kasuan n elementurako n konbinazio posible bakarrik dauden arren leave-p-out metodoaren kasuan konbinazio kopurua (n elementurako C_p^n konbinazio) askoz handitzen dela. Adibidez, demagun $n=50$ eta $p=2$ ditugula, orduan leave-one-out kasuan 50 konbinazio konprobatuko litzateke eta leave-p-out-ren kasuan, aldiz, $C_2^{50}=1225$.

Bestalde, balioztatze gurutzatu ez sakonen kasuan ez dira zertan aukera posible guztiak aztertu behar. Hemen, k-hosto balioztatze gurutzatuan ereduarentzako aukeratutako multzoa ausaz k azpimultzotan banatzen da. Ondoren, azpimultzo bat balioztatzerako gordeko da, beste $k-1$ azpimultzoak entrenamendurako erabiliz. Bukatzeko, leave-one-out-en gertatzen zen moduan, prozesua k aldiz errepikatuko da iterazio bakoitzean balioztatzeko azpimultzo desberdin bat erabiliz. Holdout metodoari dagokionez, aldiz, eredurako erabiliko den multzo bi azpimultzotan ausaz banantzen da, aurretik finkatutako proportzio batean (kontuan hartu, normalean, entrenamendurako azpimultzoa balioztatzeko azpimultzo baino handiagoa izatea gomendatzen dela). Kasu bi hauetan azpimultzo bakoitzean dauden elementuek antzeko banaketa izan behar dute, hau da, azpimultzo guztiek aztertzen diren elementu mota guztietako adibideak izan behar dituzte. Bukatzeko, Monte Carlo balioztatze gurutzatua dugu. Kasu honetan portzentaje bat aurretik finkatzen da, demagun %80. Orduan, ausaz datuen %80-

a entrenamendurako hartuko dira eta beste %20-a balioztatzerako. Hau egin ostean, prozesu berdina hainbat aldiz errepikatuko lirateke.

4.2. Overfitting-a antzematen

Lehenago aipatu den moduan, balioztatze multzoa *overfiting-a* antzemateko ere erabiltzen da. Horretarako, algoritmoak aplikatzen diren heinean, iterazio bakoitzean lortutako erroreak aztertzen dira, bai entrenamenduko multzokoa baita balioztatze multzokoa ere . Ondoren, erroreen bilakaera modu grafiko batean ikus daiteke:



Grafikoan ikus daitekeen moduan entrenamenduko errorea beti txikitu edo konstante mantenduko da. Hau beti gertatuko da, elementu berdinak algoritmoa entrenatzeko erabiltzen direnez iterazioak pasa ahala datu berdinekin gero eta emaitza hobeagoak emango baititu. Bestalde, 25. iteraziotik aurrera balioztatze datuen errorea txikitzen gelditu eta handitzen hasten da. Puntu horretan algoritmoak informazio berria ikasteari utzi eta entrenamenduko datuen emaitzak ikasten hasiko da eta, beraz, kasu honetan lor daitekeen modelorik onena iterazio kopuru horrekin izango da.

Autokorrelazio espaziala eta bero mapak

Egin den Euskal Autonomi Erkidegoko hotel eta pentsioen azterketarako erabilitako datuak bi iturri nagusitik etorri dira: web plataformetatik eta Euskal Estatistika Erakundearen, Eustat, turismoko direktorioetatik.

Eustat erakundeari dagokionez, Establezimendu Turistiko Hartzaileen Inkestako datuak erabili dira. Hauek, hoteli eta pentsioei buruzko informazio gehigarria dituzte: Kategoria, Okupazioa, logela kopurua, estratoa, koordenatu geografikoak…

Web plataformei dagokienez, *Eustat* erakundeak garatutako web scraping teknika pertsonalizatu baten bitartez EAE-ko hotel eta pentsioen prezioak lortu dira. Gainera, prezioek denboran bariazioak eduki ditzaketenez, aztertzen den eguna baino 120 egun lehenagotik egun horretako hotelaren prezioa egunero hartu egin dira, egoerarik optimoenean hotel eta egun bakoitzeko 120 prezio lortuz. Behin hotel eta egun bakoitzeko 120 datu inguru edukita horiek laburbildu egin dira, kasu bakoitzeko prezio bakarra izateko. Helburu horrekin, lortutako elementu guztien mediana kalkulatu da. Estatistiko honek muturreko balioen eragina ezabatzen baitu eta, beraz, hasierako aurreprozesatze moduan balio dezake.

Pentsa daitekeen moduan *Eustat* erakundeko datu baseko hotel guztiak ez dira web plataformetan agertu. Hala ere, Erkidegoko hotel eta pentsioen %80 inguruko kobertura lortu egin da.

Behin bi iturrietako informazioa fusionatu eta gero ondoko pausuak jarraitu dira:

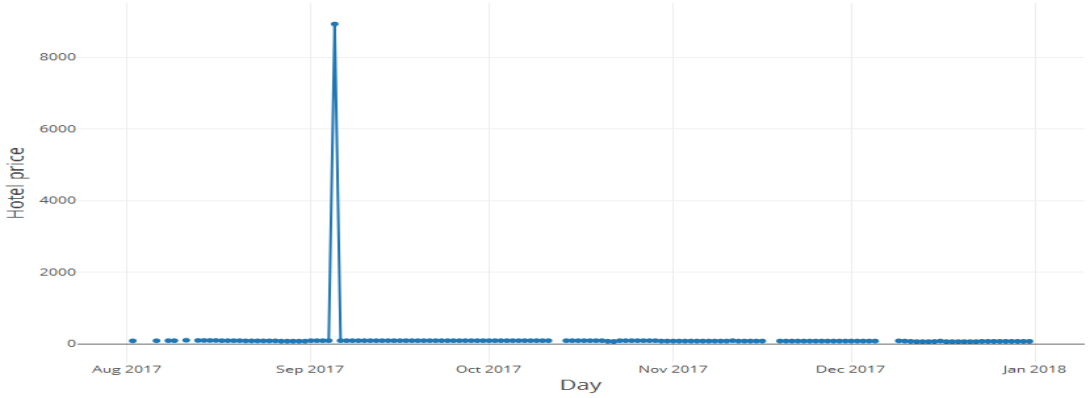
1. Outlier azterketa.
2. Balio galduen inputazioa.
3. Hotelen analisi espaziala.
4. Bero mapa.

Egindako analisi guztiak *R* programazio lengoaian egin dira. Programazio lengoaia hau estatistikaren munduan erabilia izateaz gain, azken urteetan *machine learning* munduan indarra hartu du. *R*-n programatzeko *Rstudio* Garapen Ingurune Integratua (IDL ingelesez) erabili da, duen kode editoreaz, debugging tresnez eta ikustarazte tresnez baliatzeko.

5. Outlier azterketa

Lan egiteko erabiltzen diren datuak modu automatiko baten lortzen direnez hauek txarto hartzeko aukera existitzen da (scraping-aren prozesuan arazoren bat egon delako, web orrialdearen arazo batengatik…). Txarto hartutako balioak, txarto egonda ere, denboran hurbil hartutako datuak bezalakoak badira etorkizuneko analisietan ez dute zertan eragin handia izan behar. Bestalde, errore hauek muturreko balioak edo oso arraroak badira, etorkizunean eragin negatiboa izan dezakete.

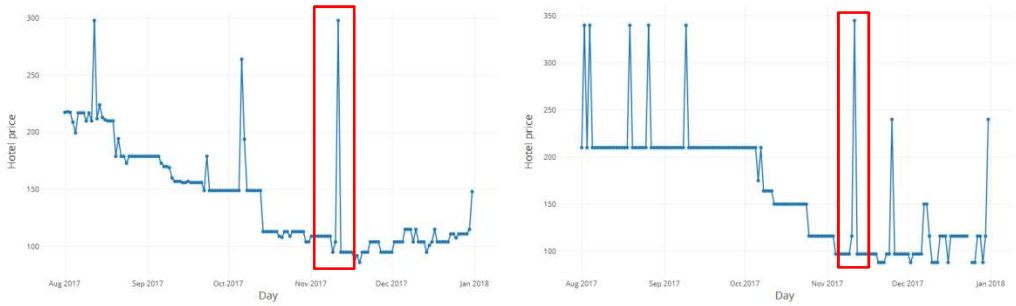
Scraping-a egin eta gero egun bakoitzerako lortutako datu guztien mediana kalkulatzen denez prozesuan ateratako balio arraro asko jada desagertzen dira. Hala ere, honek ez ditu muturreko balio guztiak ezabatzen eta azterketa sakonago bat egitea beharrezkoa egiten da:



Balio arraroen artean hiru mota nagusi daudela esan daiteke:

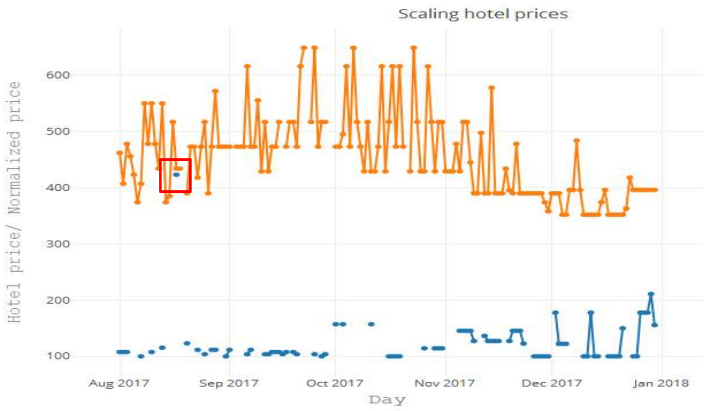
1. *Outlier*: Datuen kopuru txiki bat, bata bestearengandik eta gehiengoaren multzotik bananduta daudenak.
2. *Anomalia*: Datuen kopuru txiki bat, kategoriaduna normalean, bata bestearengandik hurbil baina gehiengoaren multzotik urrun daudenak.
3. "*Novelty*": Ezezaguna den kategoria berri bat osatzen duten puntuak.

Gure kasuan lehenengo bi motatako balioak eduki ditzakegu. Alde batetik, outlierrak aurreko irudian ditugun puntua bezalakoak diren datuak izango dira. Hemen, "ezinezkoa" den balio baten aurrean baikaude. Bestetik, anomaliak egun bereziak izango lirateke, zubiak, jai egunak, asteburuak... Ondoko irudian ikusten den moduan, azaroaren 11-an hainbat hotelek balio arraroak dituzte. Hala ere, fenomeno hau hainbat kasutan gertatzen da, ez da hotel bakarreko gauza.

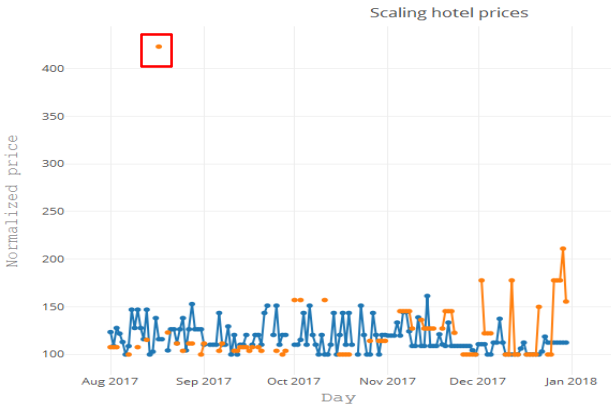


5.1. Hotelak konparagarriak egiten

Hotel desberdinei irizpidea berdina aplikatzeko hauek eskala berdinean egotea beharrezkoa da. Hau da, 5 izarreko hotel baten prezioa ez da inoiz izar bateko pentsio baten prezioarekin konparagarria izango. Modu berean, Donostiako pentsioak, orokorrean, ez dituzte Gasteizeko pentsioen prezio eremu berdinak. Ondorengo grafikoan 5 izarreko hotel baten eta pentsio baten prezioak ikus daitezke.



Karratu gorri batez markatutako balioa outlier bat dela badirudien arren bi hotelak batera aztertzerakoan gerta daiteke puntu hau bost izarreko balioen artean ez nabarmentzea. Arazo hori saihesteko hotelak eskala berdinera pasatu dira. Horretarako, hotelen prezioa aztertu beharrean prezioen bariazioa aztertu da, hilabeteka. Hilabeteko balio minimoak oso egonkorrak zirela ikusita, hilabete bakoitzeko minimoari 100 balioa esleitu zaio. Hori eginda, beste balioei minimoarekiko duten igoerarekiko proportzionala den balioa esleitu zaie. Hau da, hilabeteko minimoa 200€ bada eta hilabete berdinean 600€-ko balio bat badago, minioa duen puntuari 100 balioa esleituko zaio eta 600 duenari 300. Hau eginda, lehen aztertutako hotel berdinak aztertuz ondoko grafikoa edukiko genuke:



Ikus daitekeen moduan, hasiera baten outlierra zela pentsatzen genuen puntua kasu honetan argi eta garbi nabarmentzen da.

5.2. Grubbs-en metodoa

Grubbs-en testa outlier bakarra aurkitzen duen testa da. Honek aztertutako datuen maximoa eta minimoa aztertzen ditu eta hauetako bat outlierra den esaten digu. Horretarako,

aztertutako elementuen eta hauen batez bestekoaren arteko distantzia maximoa duen elementua aztertzen du desbiderapen estandarra kontuan hartuz:

$$G = max_{i=1,..,N} \frac{|Y_i - \bar{Y}|}{s}.$$

Test hau R softwar-aren *outlier* paketean eskuragarri dago *grubbs.test()* funtzioaren bitartez. Funtzio honek aztertutako elementuen maximoa edo minimoa balio estremoak diren esaten digu. Horretarako, $0 \leq p - balio \leq 1$ estadistiko bat bueltatzen digu. Balio hori, guk finkatutako dugun muga batekin batera, elementu bat outlier-a den edo ez jakiteko balioko digu.

5.3. Hotelak aztertu

Egindako outlier azterketan, lehenik eta behin hotelak bakarka aztertu dira. Konturatu puntu honetan outlierrak diren baino puntu gehiago aterako direla, hemen anomaliak ere agertuko baitira. Prozesu hau hilabeteka egin da, horrela, hile berri bateko datuak lortzerakoan ez dira aurreko hilabeteko datuak beharko.

5.4. Egunak aztertu

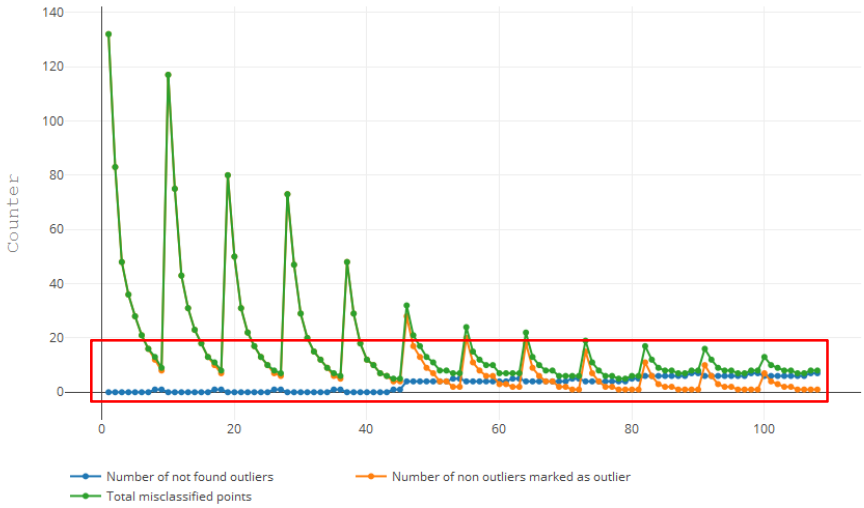
Hotelak aztertzeaz gain, egunak ere aztertu dira. Hau eginda, egun bateko gertaera berezi batengatik (festak, oporrak, ekitaldiak…) hotelek prezio igoera orokor bat izan badute puntu horiek ez dira outlier moduan hartuko. Azterketa hau lurraldeka egin da, lurralde bakoitzak bere jai propioak baititu.

5.5. Emaitzak bateratu

Azkenik, gure datuen outlier moduan aztertutako kasu bietan outlier moduan agertzen diren puntuak bakarrik hartu dira. Beste modu batean esanda, balio bat outlierra izango da baldin eta soilik baldin bere hotelaren balioen artean outlier-a bada eta bere egunean outlierra bada.

5.6. p-balioa

Bukatzeko, lehen komentatu den moduan grubbs-en testak $p - balio$ bat bueltatzen du eta muga bat definitu behar da non $p < muga$ bada aztertutako puntua outleirra izango den. Horretarako, ditugun datuen outleir-ak eskuz markatu dira eta $p - balio$ desberdinen konbinazioak (bat hotelen azterketarako eta beste bat egunen azterketarako) lortutako emaitzak eskuz markatutako emaitzekin konparatu dira. Ondoko grafikoan lortutako balioen konparazioa dago non puntu desberdinak $p - balio$ bikote desberdin bat adierazten duten. Lerro urdinak markatutako outlier-etatik aurkitu ez direnak dira, laranjak aldiz normalak diren eta outlier moduan markatu diren puntuak dira eta berdeak, azken bi motako erroreen batura da:



Hau eginda, $p - balio$-aren muga definitzeko orduan ondo irizpidea jarraitu da:

1. Errore kopuru totala (marra berdea) minimizatzen dituzten $p - balio$-ak aztertu.
2. Outlier gehien aurkitzen dituen $p - balio$-ak aukeratu (marra urdina minimizatu).

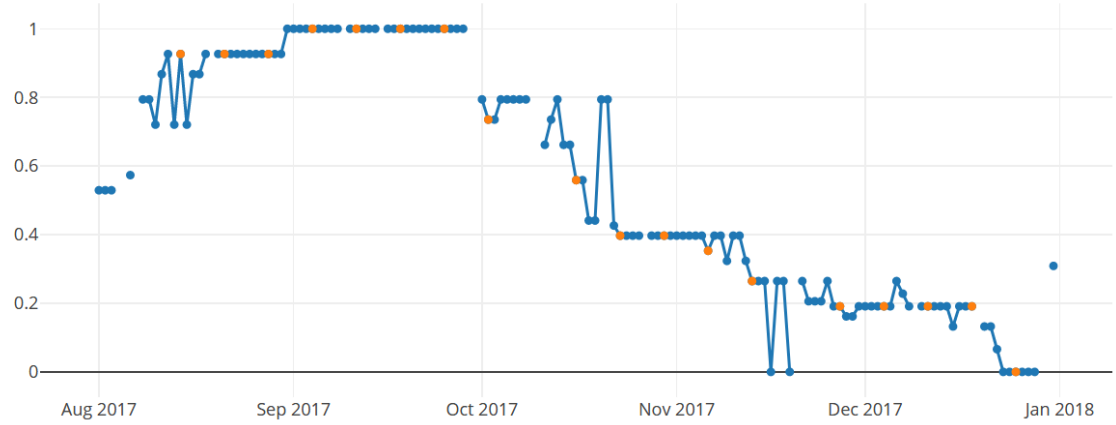
6. Balio galduen inputazioa

Outlierrak detektatu eta gero hauekin zer egingo den erabaki behar da. Gure kasuan balio horiek ezabatu eta egun horretako datuak inputatzea erabaki da, balio galduekin batera. Horretarako, *R*-k baliatzen duen *imputeTS* paketea erabili da. Pakete honek denbora serieen inputaziorako hainbat metodo ditu: *na.seasplit, na.seadec, na.interpolation, na.kalman…* Gure kasu partikularrean serieak aldizkakotasuna dutela esan daiteke, hau da, normalean 7 eguneko patroiak ikus daitezke non ostiral eta larunbatetan prezioa gora doan..

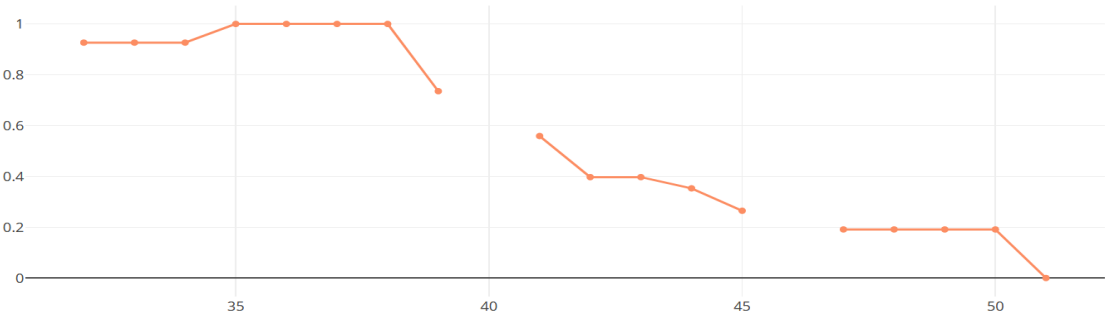
Inputazio metodoa aukeratze irizpideak eta emaitzak azaldu baino lehen aldizkakotasuna dute serieekin lan egiteko gomendatzen diren funtzio nagusiak ikusiko ditugu: *na.seasplit, na.seadec, na.kalma.*

6.1 na.seasplit

Ikusiko dugun lehengo funtzio honetarako egin beharreko lehenengo pausua aztertuko den seriearen aldizkakotasuna adieraztea da. Horretara, gure *x* hotel baten prezioaren serieari denbora serie formatua, *R*-n, emango diogu: `x<-ts(x, frequency=7)`. Hau egiterakoan 7 eguneko aldizkakotasuna dagoela adierazten dugu. Hau eginda, metodoak serie originaletik 7 serie desberdin aterako ditu (bat asteko egun bakoitzeko). Adibidez demagun hotel baten prezio normalizatuen serie bat dugula non astelehenak laranjaz nabarmendu diren:



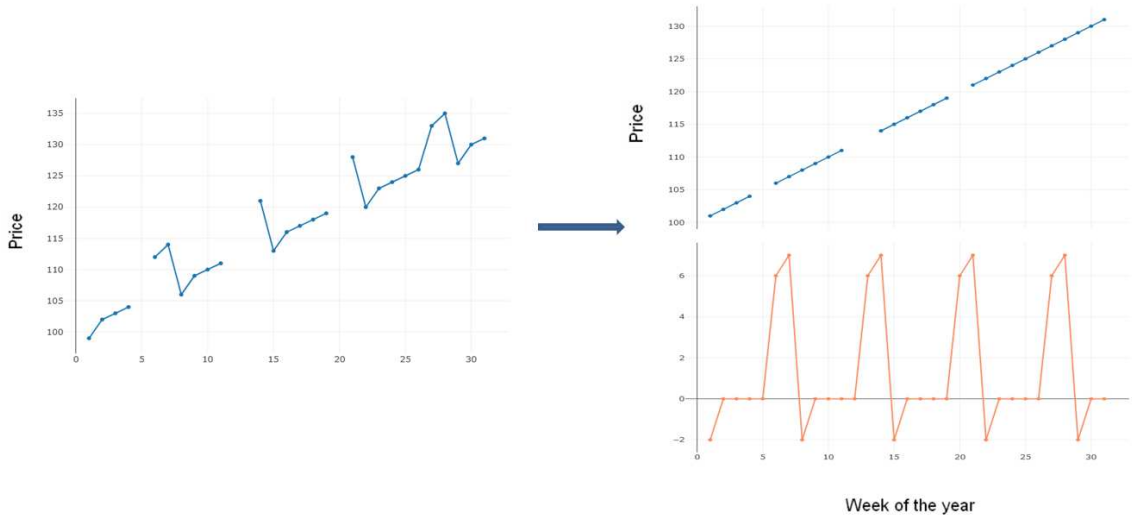
Orduan metodoak ondorengo argazkian ikus daitekeen seriearen moduko beste 7 aterako lituzke:



Hau egin eta gero serie bakoitzari nahi den inputazio metodoa aplikatzen zaio, *imputeTS* paketeko aukera posibleetatik: *ARIMA, inerpolazioa, Basic Estructural Models…*

6.2 na.seadec

Aurreko kasuan gertatzen zen bezala, hemen ere gure seriea 7 eguneko aldizkakotasuna duela adierazi beharko dugu. Hau egin ostean serieari aldizkakotasuna kentzen zaio geratzen den seriea inputatzeko. Orduan, inputatutako serie sinpleari aldizkakotasuna gehituko litzaioke. Ondorengo irudietan ikus daitekeen moduan:



Hemen, ezkerrean hotel baten prezioaren eboluzioa ikus daiteke, hainbat balio galduekin. Orduan, eskumako zatian ikusten den moduan seriearen aldizkakotasuna (serie laranja) baztertuko litzateke. Hau eginda, geratzen den serieari (urdina) nahi den inputazio metodoa aplikatuko litzaioke (*ARIMA, inerpolazioa, Basic Estructural Models…*) eta lortutako inputatutako serieari aldizkakotasuna berriro gehituko litzaioke.

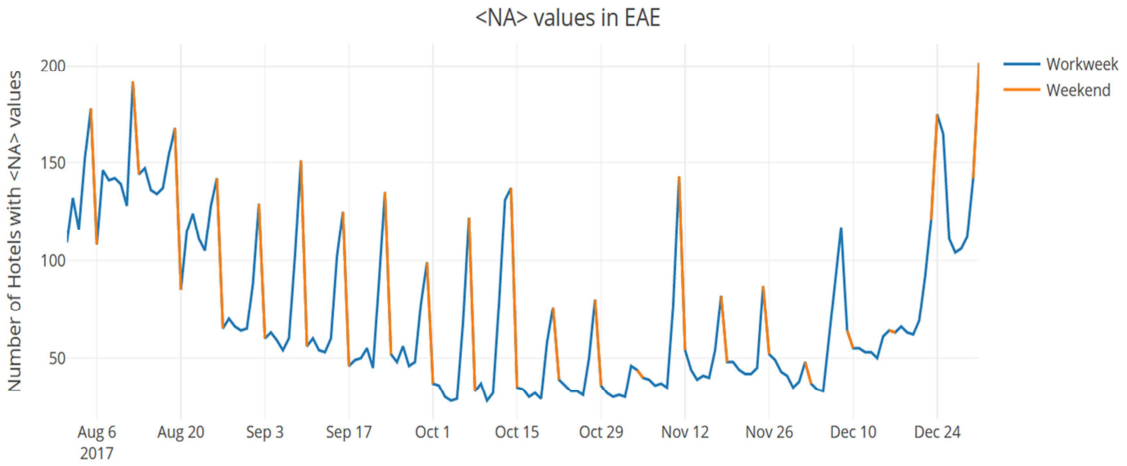
6.3 na.kalman

Funtzio honek aztertzen den seriearen modelizazioan oinarritzen da. Funtzioari seriearen edozein eredu pasa lekioke, kasuaren beharren arabera, baina funtzioak berak ere baditu 2 modelizazio metodo integraturik: *auto.arima* eta *StructTS*. Funtzio honen oinarria *kalmanen filtroetan* dago.

Kalmanen filtroak, Linear Quadratic Estimation (LQE) moduan ere ezagutzen direnak, ezezagunak diren balioak estimatzeko denboran zehar hartutako balioak erabiltzen ditu. Horretarako denbora leiho bakoitzeko bildura probabilitate banaketa bat estimatzen du. Eredu hau Hidden Markov Models tekniken antzekoa dela esan daiteke, egoera espazioa jarraitua izanda eta aldagaiak banaketa Gaussiarra edukiz.

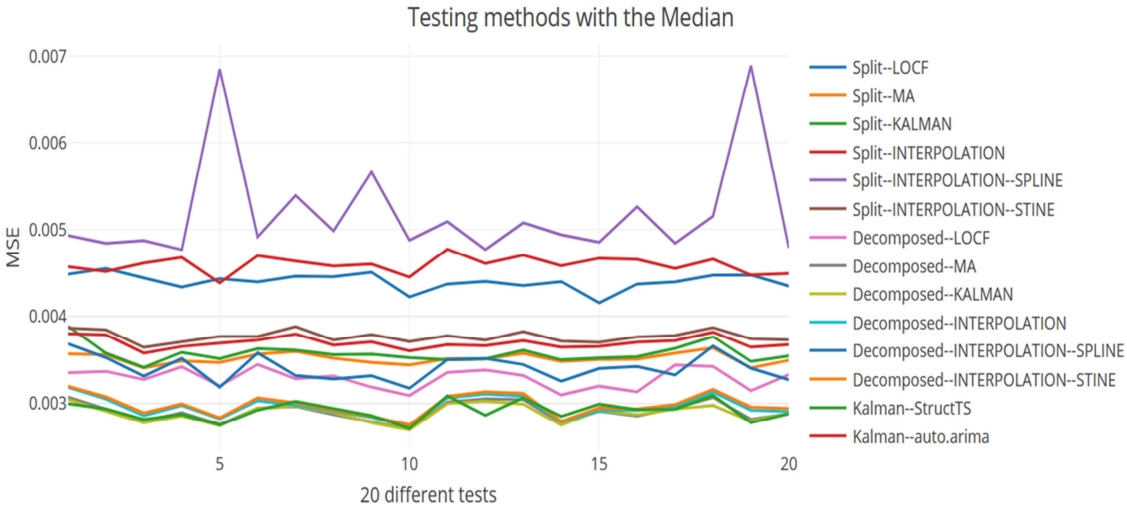
6.4 Inputazio optimoaren aukeraketa

Inputazio metodo finala aukeratzeko garaian, outlierrekin jarraitutako antzeko prozesu bat jarraitu da. Hemen, osoak eta outlier barik (150 inguru) zeuden serieak hartu dira eta ausaz 15-20 puntu tartean kendu zaizkie. Ondoren, metodo desberdinak serie hauetan aplikatu eta lortutako emaitzak jatorrizko serie osoekin konparatu dira. Serieen balio galduak gehienbat asteburuetan pilatzen direla antzeman denez (ikusi hurrengo irudia), serie osoetatik asteburuetako balioa baztertzeko probabilitatea handitu egin da.

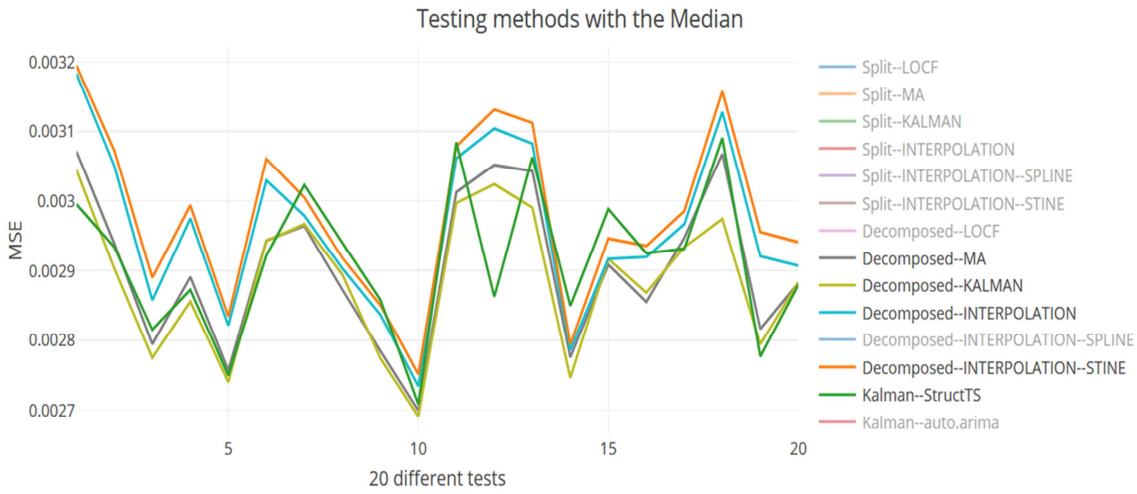


Inputatutako seriearen eta serie originalaren arteko aldea neurtzeko batez besteko errore koadratikoa erabili da.

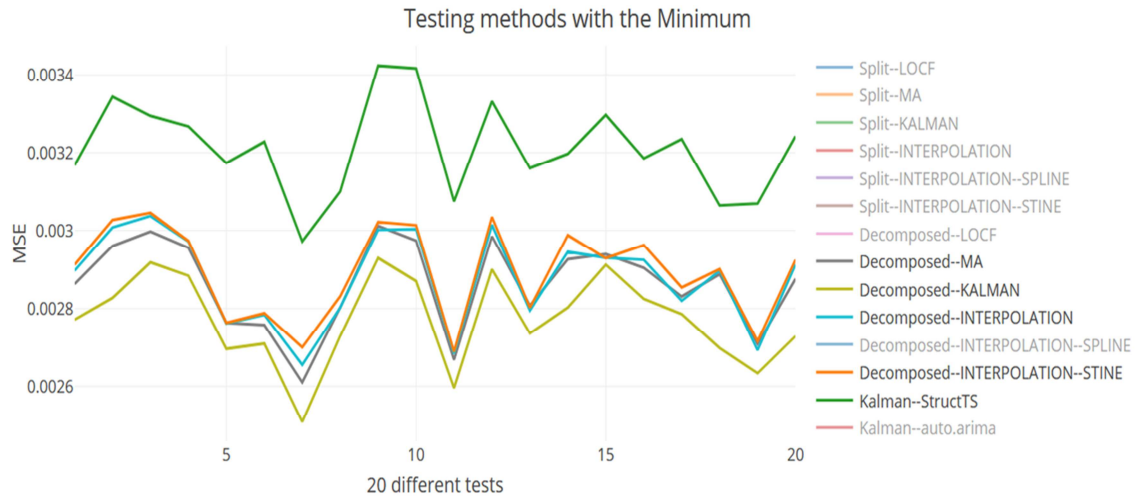
Prozesu hau 20 aldiz egin da *ImputTS* paketeko funtzio bakoitzeko eta gauza bera egin da hotelen eguneko prezio minimoekin ere (prezio hauek egonkorragoak dira eta aldizkakotasuna nabariagoa dute). Hurrengo argazkian aldi bakoitzean eta metodo bakoitzean lotutako errorea ikus daiteke:



Hemen ikus daitekeen moduan errore minimoa dutenak *na.seadec* funtzioa *MA,kalman, interpolation* eta *stine-interpolation* eta *na.kalman* StructTS *ereduarekin dira:*



Eta gauza bera gertatzen da prezio minimoak aztertzerakoan:



Datu guzti hauek kontuan hartuz, *na.seadec* funtzioa *kalman* filtroarekin erabiltzea erabaki da.

7. Autokorrelazio espaziala

Izan bedi $\Omega = \{w_i : i = 1, \dots, n\}$ espazialki egituratutako multzoa (aztertzen ari garen hotelen multzoa adibidez) orduan, autokorrelazioa espaziala Ω maparen antolaketa eredua aztertzean datza. Puntu batean aztertutako egituraren magnitudearen balioa erlatiboki altua (baxua) bada, gure kasuan hotelen prezioa izango zena, eta bere inguruneko magnitudeen balioak ere altuak (baxuak) badira, orduan autokorrelazioa positiboa (altua) izango genuke. Aldiz, aztertutako kokapen zehatz batean magnitudearen balioa erlatiboki altua (baxua) bada eta magnitude horren balio baxuz (altuz) inguratuta badago, autokorrelazio negatiboa (baxua) dagoela esan daiteke. Aldagaiek beraien inguruko aldagaiekiko balio antzekoak edo desberdinak hartzeari dependentzia espaziala deritzo eta antzekotasun edo desberdintasun hori neurtu ahal izateko autokorrelazio espazialaren indizeak erabiltzen dira.

7.1 Oinarrizko definizioak

Korrelazio indize hauek kalkulatzeko orduan badaude beharrezkoak diren bi elementu: *Pisuen matrizea* eta *egitura matrizea*. Pisuen matrizea, $P = [p_{ij}]$, elementuen balioen menpekoa da, matrize karratua da eta bere balioak absolutuan hartzen baditugu simetrikoa izango litzateke. Bestalde, egitura matrizeak, $A = [a_{ij}]$, elementuek espazioan duten egitura adierazten du eta simetrikoa da.

Egitura matrizea definitzeko modu asko daude, adibidez, elementuen arteko distantzia d_{ij} kontuan har daiteke $a_{ij} = \frac{1}{d_{ij}}$ moduan definituz. Kasu honetan kontu handia izan behar da $d_{ij} = 0$ denean. Egitura matrizeko elementuei ere 1 balioa eman ahal zaie ondoz ondoko elementuak badira eta 0 kontrako kasuan. Hotelen kasuan, ondoz-ondokotasun kontzeptu hau estrato, herri edota lurralde historiko berdinean egotea adieraz dezake. Bukatzeko, normalean a_{ii} osagaiari 0 balioa ematen zaio.

Behin matrize hauek definituta autokorrelazio indize bat ondoko itxura edukiko luke:

$$\Gamma = \lambda \sum_i \sum_j a_{ij} p_{ij}.$$

Korrelazio indizeak definitu baino lehen erabiltzen diren estatistiko espazial desberdinak definituko dira. Hasteko, w_i elementu bakoitzak A_i pisu espaziala izango du, ondoko eran definitzen dena:

$$A_i = \sum_j a_{ij}.$$

Hauek jakinda multzoko pisu totala ere definitu daiteke:

$$A_{Tot} = \sum_i A_i = \sum_i \sum_j a_{ij}.$$

Konturatu$a_{ii} = 0$ denean A_{Tot} bikoitia izango dela, $a_{ij} = a_{ji}$ baita. Behin elementu bakoitzaren pisua eta pisu totala definituta, hauekin *egokitutako media* eta *bariantza* definitzen dira. Gure kausan w_i hotel bakoitzaren prezioa x_i moduan adieraziko dugu. Hasteko, egokitutako media ondoko eran definitzen da:

$$\overline{x_A} = \frac{1}{A_{Tot}} \sum_i A_i x_i.$$

Media berri honen ezaugarri nagusia bere inguruan elementu gehien (gutxien) dituzten elementuek balio finalean eragin handiagoa (txikiagoa) izango dutela da. Hotelen kasuan, adibidez, Donostia moduko hiri baten hotelek pisu handiagoa izango dute Galdakaon egon daitekeen hotel batekin konparatuz. Ohartu media honek ohikoak betetzen dituen propietateak ere betetzen dituela:

$$\frac{1}{A_{Tot}} \sum_i A_i (x_i - \overline{x_A}) = \frac{1}{A_{Tot}} \sum_i A_i x_i - \frac{1}{A_{Tot}} A_{Tot} \overline{x_A} = 0.$$

Egokitutako bariantza modu baliokide batean definitzen da:

$$s_A^2 = \frac{1}{A_{Tot}} \sum_i A_i (x_i - \overline{x_A})^2.$$

Mediaren kasuan gertatzen zen moduan, elementu asko dituzten inguruneetan dauden elementuek eragin handiagoa izango dute isolatutako elementuekin konparatuz. Behin estatistiko pare hau definituta, hurrengo pausu logikoa estandarizatutako aldagaiak definitzea litzateke. Aldagai berri hauen egokitutako media 0 eta egokitutako bariantza 1 izango litzateke:

$$z_i = \frac{x_i - \overline{x_A}}{s_A^2}.$$

Hortaz aparte, hotel bat bere ingurunearen bitartez karakteriza daiteke. Horrela, $w_i$elementuaren ingurunea ondoko eran laburbildu daiteke:

$$\mathrm{y_i} = \frac{1}{\mathrm{A_i}} \sum_{\mathrm{j}} \mathrm{a_{ij} x_j}.$$

Modu honetan lortzen diren elementuek x_i elementuen media mantentzen dute eta hauek osatzen duten multzoa hasierako multzoaren leunketa baten modukoa izango litzateke.

7.2 Indize globalak

Behin oinarrizko kontzeptuak definituta korrelazio indizeak ikusteko ordua iritzi da. Lehen esan bezala, indizeak normalean ondoko itxura dute:

$$\Gamma = \lambda \sum_i \sum_j a_{ij} p_{ij}.$$

Ikusiko dugun lehen indizea *Moran*-ena da. Honek pisu moduan estandarizatutako elementuen biderketak hartzen ditu, $p_{ij} = \frac{(x_i - \bar{x})}{s} \frac{(x_j - \bar{x})}{s}$. Gure kasuan alde espaziala aztertzen gaudenez egokitutako media eta bariantza erabiliko ditugu, $p_{ij} = z_i z_j$. Beraz, *Moranen egokitutako indizea* ondokoa litzateke.

$$I^* = \frac{1}{A_{Tot}} \sum_i \sum_j a_{ij} z_i z_j = \frac{1}{A_{Tot}} \sum_i \sum_j a_{ij} \left(\frac{x_i - \overline{x_A}}{s_A^2}\right)\left(\frac{x_j - \overline{x_A}}{s_A^2}\right).$$

Indize honek gero eta balioa altuago eduki orduan eta korrelazio altuagoa adieraziko du. Modu berean, gero eta balio negatiboagoa hartzerakoa orduan eta korrelazioa baxuagoa izango du.

Beste indize bat *Geary*-ren indizea litzateke. Honek, pisu matrizeko elementu moduan estandarizatutako elementuen kenketak ditu. Beraz, *egokitutako Gearyren indizearen* kasuan $p_{ij} = z_i - z_j$ edukiko genuke:

$$C^* = \frac{1}{2}\frac{1}{A_{Tot}} \sum_i \sum_j a_{ij}(z_i - z_j) = \frac{1}{2}\frac{1}{A_{Tot}} \sum_i \sum_j a_{ij} \frac{\left(x_i - x_j\right)^2}{s_A^2}.$$

Geary-ren indizearen kasuan balio positiboak hartuko ditu. Hemen, gero eta zerotik hurbilago egon orduan eta korrelazio altuagoa dagoela adieraziko da.Gainera, gero eta balio altuagoa hartu orduan eta korrelazio negatiboagoa dagoela adieraziko da.

Hirugarren indizea *Lebart*-en indizea litzateke. Kasu honetan, *egokitutako Lebarten indizea* pisu moduan elementuen eta hauen ingurunearen arteko kenketaren karratua, $p_{ij} = \frac{(x_i - y_i)^2}{s_A^2}$, hartuko luke:

$$L^* = \frac{1}{A_{Tot}} \sum_i \sum_j a_{ij} \frac{(x_i - y_i)^2}{s_A^2} = \frac{1}{A_{Tot}} \sum_i A_i \frac{(x_i - y_i)^2}{s_A^2}.$$

Gearyren kasuan gertatzen den moduan, hemen ere indizea gero eta 0 baliotik hurbilago egon orduan eta korrelazio positiboagoa egon da. Aldiz, gero eta balio altuagoa eduki orduan eta korrelazio negatiboagoa adieraziko du.

Laugarren indizea [11] lanean definitutako η_1 indizea da. Honek pisu matrizeko elementu moduan elementuen ingurunearen eta egokitutako mediaren arteko kenketen karratuak $p_{ij} = (y_i - \overline{x_A})^2$ ditu:

$$\eta_1^* = \frac{1}{A_{Tot}} \sum_i \sum_j \frac{a_{ij}(y_i - \overline{x_A})^2}{s_A^2} = \frac{1}{A_{Tot}} \sum_i A_i \frac{(y_i - \overline{x_A})^2}{s_A^2}.$$

Bukatzeko, aipatutako lan berdinean agertutako beste indize bat, η_2 indizea, definituko da. Honen kasuan, $p_{ij} = \left(y_i - x_j\right)^2$ elementuaren ingurunea beste elementuekin konparatzen da kenketaren bitartez eta karratua kalkulatzen da:

$$\eta_2^* = \frac{1}{A_{Tot}} \sum_i \sum_j a_{ij} \frac{\left(y_i - x_j\right)^2}{s_A^2}\ .$$

Azkeneko bi indize hauek ere 0 baliotik hurbil daudenean autokorrelazio positiboa adierazten dute eta gero eta balio handiagoa izan orduan eta korrelazio negatiboagoa izango du.

7.3 Indize lokalak

Indize globalek multzo osoaren korrelazio espaziala neurtzen duten arren, aztertutako elementu bakoitzak korrelazioa espazial totalean duen eragina ere ikustea interesgarria litzateke. Horretarako, korrelazio indize lokalak definitzen dira. Indize hauek Anselin [2] proposatutako irizpidea jarraituz definitzen dira:

1. Indize lokalek, elementuek indize globalean duten eragina adierazten dute.
2. Elementu guztiei dagokien indize lokalen batura indize globalarekiko proportzionala da.

Irizpide hori jarraituz aurreko atalean azaldutako indizeen zati lokalak defini daitezke:

1. **Moran:** $I_i^* = \frac{1}{A_i}\sum_{j=0}^{n} a_{ij}(\frac{x_i-\overline{x_A}}{S_A})(\frac{x_j-\overline{x_A}}{S_A})$.
2. **Geary:** $C_i^* = \frac{1}{2}\frac{1}{A_i}\sum_{j=0}^{n} a_{ij}\left(\frac{x_i-x_j}{S_A}\right)^2$.
3. **Lebart:** $L_i^* = \left(\frac{x_i-y_i}{S_A}\right)^2$.
4. $\boldsymbol{\eta_1}$: $\eta_{2i}^* = \left(\frac{y_i-\overline{x_A}}{S_A}\right)^2$.
5. $\boldsymbol{\eta_2}$: $\eta_{2i}^* = \frac{1}{A_i}\sum_{j=0}^{n} a_{ij}\left(\frac{y_i-x_j}{S_A}\right)^2$.

Hemengo indize lokal bakoitzak w_i hotelak indize globalean duen eragina adierazten du. Gainera, indize lokalen konbinazio linealetatik indize globalak lor daitezke. Demagun Γ indize globala dela eta Γ_i bere aldagai lokala dela, orduan:

$$\Gamma = \frac{1}{A_{Tot}}\sum_i A_i\Gamma_i.$$

Hau da, indize orokor guztiek elementu guztien indize lokalen baturarekiko proportzionalak dira, bereziki, indize orokor guztiek elementu guztien indize lokalen *batazbestekoak* dira. Hori dela eta, hasieran finkatutako irizpide guztiak betetzen direla baiezta daiteke eta, beraz, indize lokalak ondo definituta daudela ere.

7.4. Indizeak hoteletan

Behin indizeak definituta eta hauek aplikatzen hasi baino lehen pare bat gauza konpontzea faltako zen. Alde batetik, egitura matrizea definitu beharko litzateke. Bestetik, hotelen prezioetan hauen kokapena ez-ezik kategoria ere eragin handia du. Hori dela eta, kategoriaren eragina indargabetu beharko litzateke. Arazo hau kontuan hartzen ez bada egingo den analisiak emaitza okerrak eman ditzake, adibidez, kategori baxuko hotelez inguratutako kategoria altuko hotelak kategoriak duen prezioaren igoeraren eraginagatik nabarmentzea gerta daiteke eta ez kokapenaren eraginagatik.

Kategoriaren eragina saihesteko asmoz hiru analisi desberdin jarraitu dira. Hauek hobeto ulertzeko demagun ondorengo datuak ditugula:

Hotela	Kategoria	Data	Prezioa
1	H3	2039-08-19	75
1	H3	2039-11-15	56
2	H5	2039-08-19	500
2	H5	2039-11-15	300
3	P1	2039-08-19	30
3	P1	2039-11-15	15
4	H3	2039-08-19	90
4	H3	2039-11-15	60
4	H3	2039-07-15	75

Alde batetik, lehenengo konparazioan hotelen prezioak 0 eta 100 arteko balioetara pasatu dira bere kategorian duten prezioaren arabera. Beste modu batean esanda, kategoria bateko prezio maximoari 100 balioa eman zaio eta besteak modu proportzional batean 0-100 tartera pasatu dira.

Hotela	Kategoria	Data	Balioa_1
1	H3	2039-08-19	55
1	H3	2039-11-15	0
2	H5	2039-08-19	100
2	H5	2039-11-15	0
3	P1	2039-08-19	100
3	P1	2039-11-15	0
4	H3	2039-08-19	100
4	H3	2039-11-15	11
4	H3	2039-07-15	55

Bestetik, hotelen azterketa kategoriaka egin da. Hau da, aztertzen den hotela bere kategoria berdina duten hotelekin bakarrik konparatu da.

Hotela	Kategoria	Data	Balioa_2
1	H3	2039-08-19	75
1	H3	2039-11-15	56
2	H5	2039-08-19	500
2	H5	2039-11-15	300
3	P1	2039-08-19	30
3	P1	2039-11-15	15
4	H3	2039-08-19	90
4	H3	2039-11-15	60
4	H3	2039-07-15	75

Azkenik, hotelen prezioa aztertu beharrean prezioen tendentzia aztertu da. Hemen hotel bakoitza banaka aztertu da eta bere prezioak 0 eta 100 tartera pasatu dira. Kasu honetan, hotelak bere prezio maximo historiko lortzen duen egunari 100 balioa, minimoa duen egunaria

0 balio eta beste egunei balio proportzionalak esleitu zaizkie. Gainera, prezio konstantean duten hotelei 50 balioa esleitu zaizkie.

Hotela	Kategoria	Data	Balioa_3
1	H3	2039-08-19	100
1	H3	2039-11-15	0
2	H5	2039-08-19	100
2	H5	2039-11-15	0
3	P1	2039-08-19	100
3	P1	2039-11-15	0
4	H3	2039-08-19	100
4	H3	2039-11-15	0
4	H3	2039-07-15	50

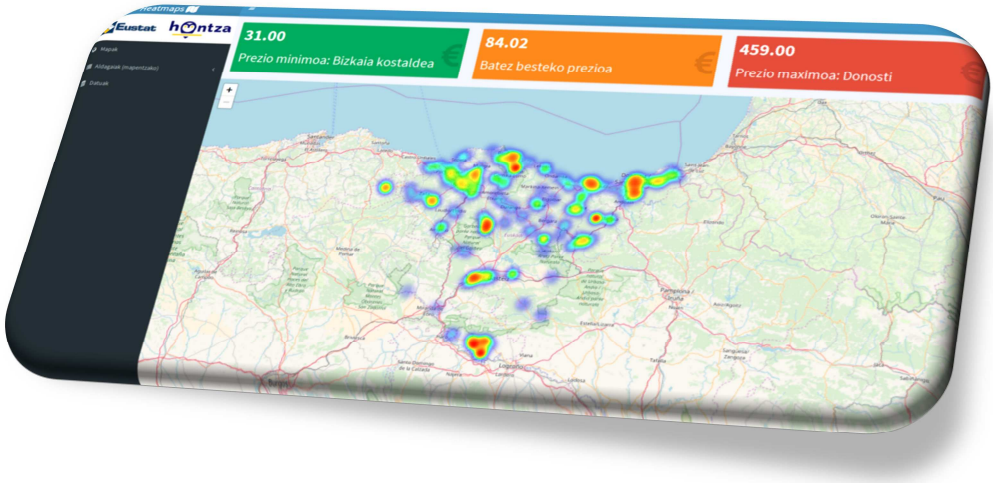
Behin kategoriaren eragina minimizatuta egitura funtzioaren aukeraketaren ordua heldu da. Hemen ere hiru kasu desberdin aztertu dira, guztietan $a_{ii} = 0$ izanik. Hasteko, w_i hotela aztertzerakoan $a_{ij} = 1$ moduan hartu da baldin eta w_j hotela hasierako hotelaren estrato berdinean badago. Bigarren eta hirugarren kasuetarako hotelen arteko distantziak kalkulatu dira. Horretarako *R* softwar-aren *geosphere* paketeko *distHaversine()* funtzioa erabili da. Funtzio honek bi puntuen arteko distantzia neurtzen du, Lurraren kurbadura kontuan hartuz. Bigarren kasuan $a_{ij} = 1$ izango da baldin eta soilik baldin w_j hotela w_i hotelik, gehienez, aurretik finkatutako r distantzia batera badago. Egindako azterketaren kasuan $r = 3km$ eta $r = 5km$ distantziak erabili dira. Azkenik, hirugarren kasurako hotel guztien eragina aztertu nahi izan da. Horretarako, $a_{ij} = \frac{1}{d_{ij}}$ moduan definitu da non d_{ij}-ren balioa w_i a w_j hotelen arteko distantzia den. Horrela, w_i hotel bakoitzaren azterketan beste hotel guztiak kontuan hartuko dira, hurbilen daudenei pisu handiagoa emanez.

Ohartu erabiltzen diren indizeen arabera emaitza desberdinak lortuko direla. Geary-ren indizeak, adibidez, aztertutako hotelaren prezioa beste prezioekin konparatuko ditu baina $\eta_1$indizeak, aldiz, aztertutako hotela ez du kontuan hartuko, aztertutako hotelaren ingurunea eta hotelen multzoko bataz bestekoa konparatzen baititu. Hori dela eta, kontuan hartu diren indizeetatik gure kasu partikularrerako erabilgarrienetarikoak zeintzuk diren ikustea egin den lehengo gauza izan da.

Hasteko, Moran-en eta Geary-ren indizeak baliokideak direla froga daiteke [11] beraz, Moran-ena baztertzea erabaki da, balio positiboekin bakarrik lan egiteko. Gainera, egindako proiektuaren helburua hotelen prezioa aztertzea denez η_1 eta η_2 ez dira azterketan sartuko. Hauek ez dute aztertzen den hotelaren prezioa kontuan hartzen, honen ingurunekoak baizik .

Hau guztia kontuan hartuz geratzen diren Geary-ren eta Lebart-en indizeakin lortutako emaitzak aztertu dira, aurretik azaldutako lau egitura matrize mota desberdinekin.

8. Bero mapa



8.1. Erabilitako software-a

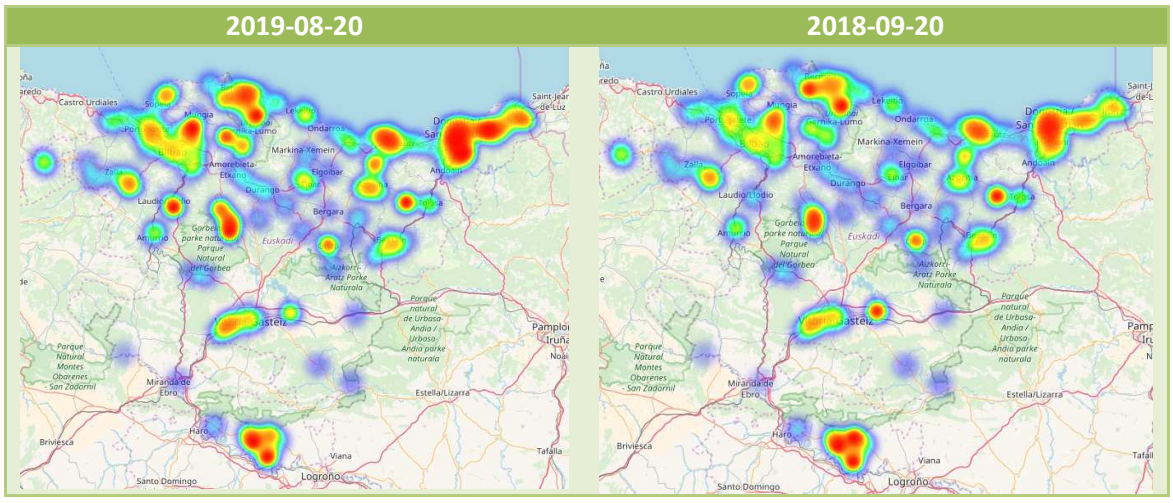
Bero mapa garatzeko *R* software-a erabili da. Hemen, *shiny* paketea erabiliz aplikazio grafiko bat garatu da non, besteak beste, *leaflet* eta *leaflet.extras* paketeen laguntzaz mapa interaktibo bat garatu den.

8.2. Bistaratzen diren datuak

Garatutako bero mapa ditugun hotelen prezioen informazioa modu desberdinean erakusten da. Alde batetik, prezio hutsen bero mapa, bestetik, prezioen tendentzia eta, bukatzeko, autokorrelazio espazialaren mapa ikus daiteke. Gainera, zoomaren arabera hotelen balioa edota aztertzen den zonaldearen balioa adieraziko du.

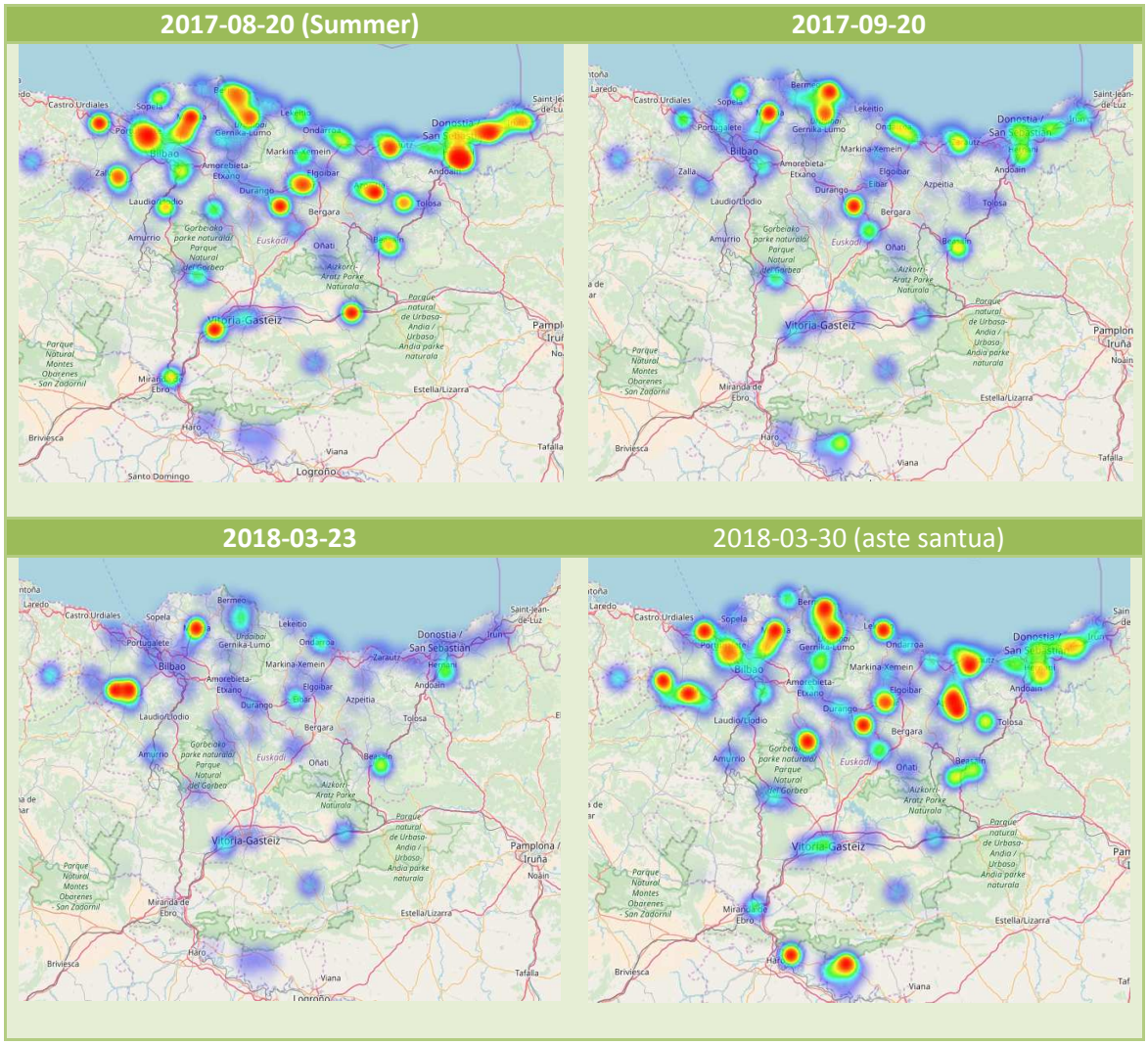
8.2.1. Prezioen bero mapa

Mapa hau sinpleena da eta, suposa daitekeen moduan, hotelen prezioaren eboluzioa erakusten du denboran zehar. Prezioa gero eta altuagoa denean gero eta kolore gorriagoa izango du. gero eta prezio baxuagoa denean, aldiz, urdinagoa. Gorri intentsitate maximoa aztertzen den hotelaren edo zonaldearen balioa erabiltzaileak finkatzen duen balio baten berdina edo altuago denean gertatuko da.



8.2.2. Prezioen tendentzia

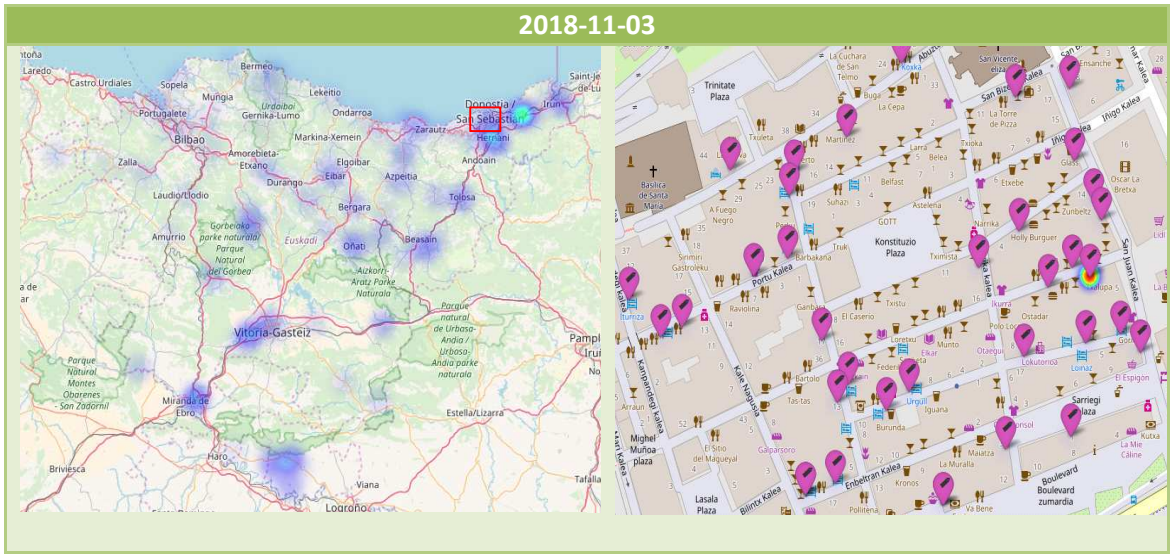
Azaldutako lehenengo modalitatearen arazo nagusiena prezioen eboluzioa bistaratzea kasu batzuetan zaila dela da. Adibidez, Euskal Autonomi Erkidegoari dagokionez, Donostia aldeko prezioa Gasteizkoak baino altuagoak izango dira ia beti eta, beraz, hauek kasu askotan gorri intentsitate handia izango dute. Hori saihesteko asmoz, bigarren bistaratze modalitate bat garatu da. Hemen, hotelen prezioa azaldu beharrean prezioen tendentzia ikusten da. Hotel bakoitza banaka aztertu ondoren honen prezioa 0-100 tartera pasatu da. Hau eginda, prezioen eboluzio orokorra ikus daiteke. Adibidez, udan edo aste santuan hotelek prezio igoerak dituztela ikus daiteke edota Irailaren erdialdera jaisten hasten direla era.



8.2.3. Korrelazio espaziala

Bukatzeko korrelazio espazialeko indizeak ematen diguten informazioa ere bistaratu daiteke. Ikusi dugun moduan, autokorrelazioak, aztertzen ditugun datuak beraien artean antzekoak diren edo nabarmentzen den baten-bat dagoen adierazten digu. Ohartu kasu honetan detailera joan behar dela informazioa lortzeko zeren eta, orokorrean, Euskal Autonomi Erkidegoko hotel guztiak bere inguruko hotelen antzekoak baitira. Adibidez, hurrengo argazkietan ikus daitekeen moduan, azaroaren 3-an zoom urrun batetik nabarmentzen den hotelik ez dagoela pentsa daiteke, hala ere, Donostialdean zoom-a egiterakoan alde zaharrean

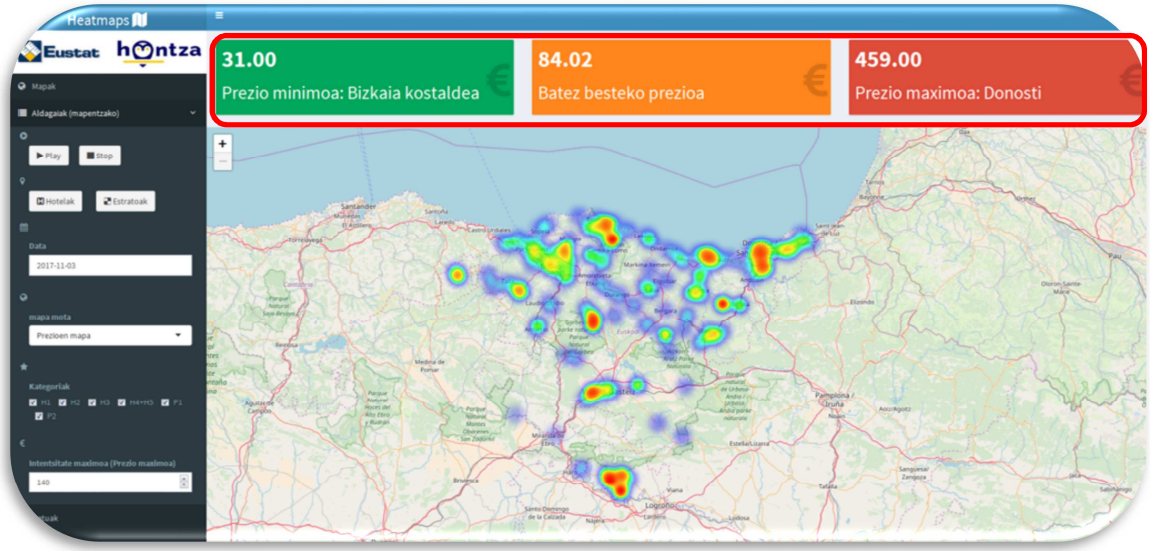
nabarmentzen den pentsio bat dagoela ikus daiteke. Kasu honetan, gaua 360 eurora dago bere inguruko pentsioek 50-100 tarteko prezioa dutelarik.



8.3.Funtzionalitate gehigarriak

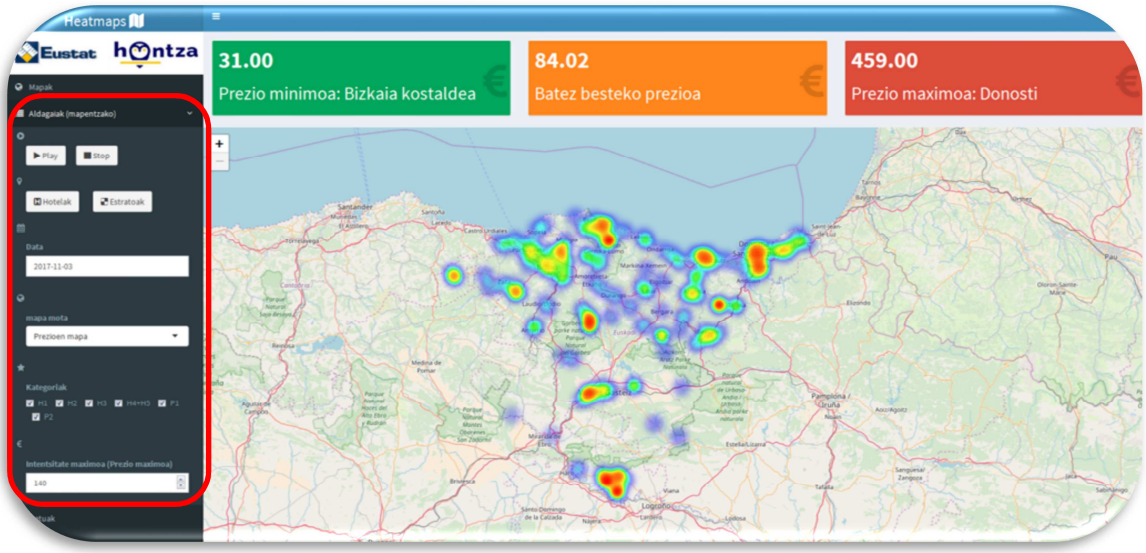
Aplikazioari informazio gehigarria gehitzeko asmoz bero mapari funtzionalitate gehigarri batzuk gehitu zaizkio.

8.3.1. Egunaren datu orokorrak



Aplikazioa zabaltzerakoan bistara datorren lehenengo gauza maparen gainean agertzen den kutxa multzoa da. Kutxa hauetan bistaratzen den egunaren datu orokor batzuk ikus daitezke: prezio minimoa, batez besteko prezioa eta prezio maximoa. Kutxa hauek kolorez aldatuko dira erakusten duten balioen arabera. Gogoratu bero maparen gorri intentsitate maximoa markatzen duen balioa erabiltzaileek aukeratzen dutela. Balio horren arabera ere kutxen koloreak hautatuko dira. Balioa maximoa baino altuagoa edo berdina bada kutxa gorria izango. Bestalde, balio maximoa eta hauen erdiaren arteko balioak baditu, kutxa laranja izango da. Bukatzeko, balio honen erdia baino baxuagoa bada, kutxa berdea izango da.

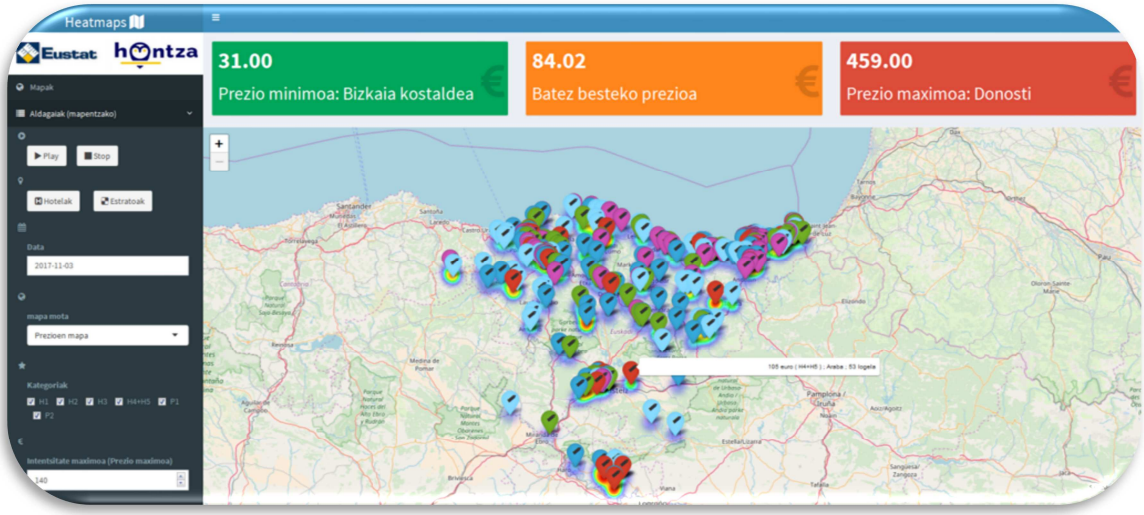
4.3.2. Mapa kontrolatzeko menua



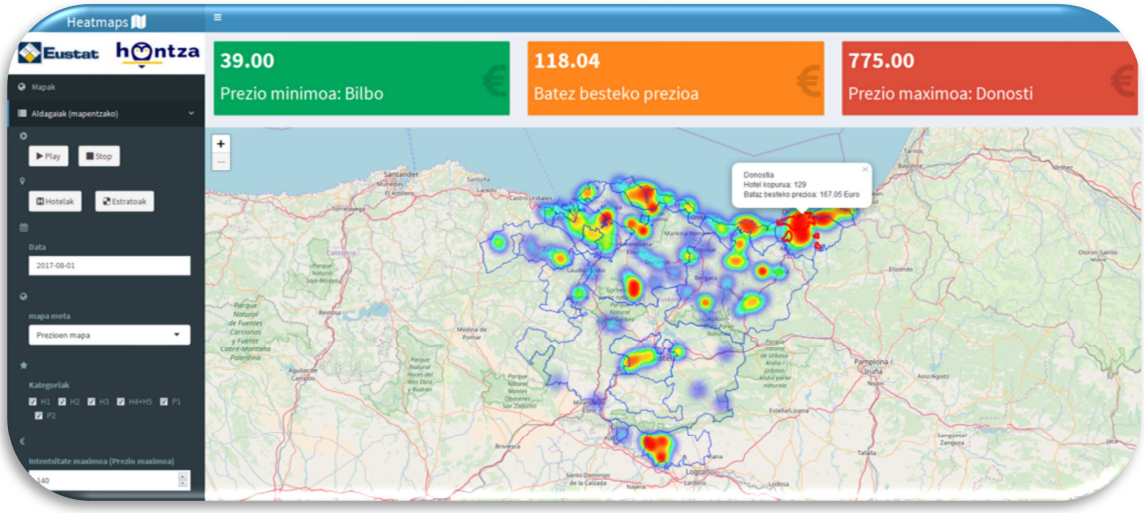
Aplikazioak ere badu atal bat mapa modu interaktiboan kontrolatzeko. Hasteko, play eta stop botoien bitartez grafikoari mugimendua eman ahal zaio. Mapa mugitzen hasterakoan, maparen egoera eguneratzen den ahala kutxetako balioak ere berristen joango dira.

Honen azpian *Hotelak* eta *Estratoak* botoiak ditugu. Hauek, suposa daitekeen moduan, hotelen eta estratoen informazio gehigarria emango dute.

Hotelak botoiari dagokienez, sakatzerakoan hotel bakoitzaren gainean puntu bat agertuko da honen informazioa emanez. Hurrengo argazkian ikus daitekeen moduan puntuak kolore desberdinekoak izango dira hotelaren kategoriaren arabera. Adibidez, pentsioak kolore morea izango dute, hiru izarreko hotelak urdin argia eta lau edo bost izarrekoak gorria. Gainera, sagua puntuen gainetik pasatzerakoan hotelaren informazioa azalduko da. Hemen, besteak beste hotelaren izena,prezioa, kategoria, probintzia eta logela kopurua adieraziko dira.

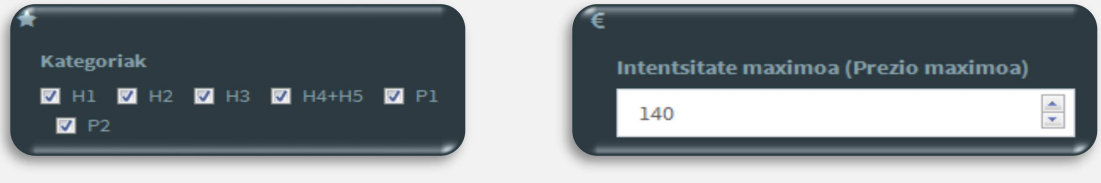


Bestalde, *Estratoak* botoia sakatzerakoan erkidegoko estrato desberdinak banatzen dituzten mugak bistaratuko dira. Hauetan klik egiterakoan fitxa txiki bat azalduko da aztertzen den eguneko informazioarekin: estratoaren izenak, hotel kopurua eta bataz besteko prezioa.



Bukatzeko beste lau kutxa daudela ikus daitezke: *data*, *mapa mota*, *kategoriak* eta *intentsitate maximoa*. Datak aztertzen den eguna eskuz aldatzeko ahalmena ematen du. Mapa mota atalak, aldiz, ditugun bistaratze aukera desberdinak aukeratzen uzten du, bai bero mapa, bai tendentziaren mapa eta baita korrelazio espazialarena ere. Modalitate gehigarri moduan mapa hutsa ikustea ere posiblea izango da.

Gainera, Kategoriak atalak hotelak kategoriaka filtratzeko aukera ematen du, erabiltzaileari interesatzen zaionak bakarrik bistaratzeko. Azkenik, Intentsitate maximoa atalak gorri intentsitate maximoa finkatzen duen balioa aukeratzeko balio du.



Bibliografia

[1] Ander Juarez Mugarza. *Autokorrelazio espazialeko indizeetan oinarritutako ertz detekziorako metodoak*. (master tesia). Euskal Herriko Unibertsitatea/Uniersidad del Pais Vasco, Donostia/San Sebastian.

[2] Anselin, L. (1995). Local indicators of spatial association—LISA. Geographical analysis, 27(2), 93-115.

[3] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[4] Canavos, G. *Probabilidad y Estadística*. McGraw Hill, 1994.

[5] Capp_e, O., Moulines, E., and Ryden, T. *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[6] Casella, G., and Berger, R. *Statistical Inference*. Duxbury Resource Center, June 2001.

[7] Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. *shiny: Web Application Framework for R*, 2017. R package version 1.0.5.

[8] Chen, E. Winning the Netix Prize: A Summary. http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/.

[9] Cheng, J., Karambelkar, B., and Xie, Y. *leaflet: Create Interactive Web Maps with the JavaScript 'Leaet' Library*, 2018. R package version 2.0.0.

[10] Concepción, Morales, Eduardo René (2010). *Contribuciones a la segmentación de imágenes mediante algoritmos de clasificación automática*. (Tesis doctoral). Euskal Herriko Unibertsitatea/Uniersidad del Pais Vasco, Donostia/San Sebastian.

[11] de Lacalle, J. L. *tsoutliers: Detection of Outliers in Time Series*, 2017. R package version 0.6-6.

[12] Devore, J. L. *Probability and Statistics for Engineering and the Sciences*, 8th ed. Brooks/Cole, January 2011. ISBN-13: 978-0-538-73352-6.

[13] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[14] Grubbs, F. E. *Sample criteria for testing outlying observations*. Ann. Math. Statist. 21, 1 (03 1950), 27-58.

[15] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[16] Hyndman, R. J. *forecast: Forecasting functions for time series and linear models*, 2017. R package version 8.2.

[17] James, G., Witten, D., Hastie, T., and Tibshirani, R. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[18] Komsta, L. *outliers: Tests for outliers*, 2011. R package version 0.14.

[19] Koren, Y., Bell, R., and Volinsky, C. *Matrix factorization techniques for recommender systems*. Computer 42, 8 (Aug 2009), 30-37.

[20] Karambelkar,B. and Schloerke, B. (2018). *leaflet.extras: Extra Functionality for 'leaflet' Package*. R package version 1.0.0. https://CRAN.R-project.org/package=leaflet.extras

[21] Kriesel, D. *A Brief Introduction to Neural Networks*. 2007.

[22] LESART, L. *Analyse statistique de la contiguïté*.1969.

[23] Moran, P. A. (1950). *Notes on continuous stochastic phenomena*. Biometrika, 37(1/2), 17-23.

[24] Moritz, S. *imputeTS: Time Series Missing Value Imputation*, 2017. R package version 2.5.

[25] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[26] Rabiner, L. R. *Readings in speech recognition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990, ch. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pp. 267-296.

[27] Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. *Recommender systems handbook*. Springer, New York; London, 2011.

[28] Robert J. Hijmans (2017). *geosphere: Spherical Trigonometry*. R package version 1.5-7.https://CRAN.R-project.org/package=geosphere

[29] Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., and Despouy, P. *plotly: Create Interactive Web Graphics via 'plotly.js'*, 2017. R package version 4.7.1.

[30] Strang, G. *Linear algebra and its applications*. Thomson, Brooks/Cole, Belmont, CA, 2006.

[31] Wickham, H. *Reshaping data with the reshape package*. Journal of Statistical Software 21, 12 (2007), 1-20.