Machine Learning per a l'optimització d'un model epidemiològic en tuberculosi

Martí Català Sabaté
Màster en Bioinformàtica i Bioestadística
Programació per a la bioinformàtica

Pau Andrio Balado
Maria Jesús Marco Galindo

2 de gener del 2018

Agraïments

Un cop acabat aquest treball del qual n'estic molt satisfet m'agradaria donar les gràcies a tots aquells que m'han ajudat a dur-lo a terme.
No puc començar per una altre persona que no sigui el meu tutor, el Pau ha estat qui m'ha guiat al llarg d'aquest camí, a vegades, una mica incert.
També donar les gràcies al Dani i a la cristina per haver-me ajudat en la realització d'aquest.
Sense el Bernat i tots aquells que han contribuït en escriure i crear el C simulator aquest treball no hauria estat possible, Clara, Joan Francesc, Júlia...
Per acabar donar les gràcies al meu company Nura per la paciència que ha tingut amb mi i a tota la gent amb la que he conviscut al llarg d'aquest temps: els meus companys de despatx, els meus amics i la meva família!

Moltes gràcies a tots, aquest treball també és vostre!

Martí

FITXA DEL TREBALL

Títol del treball:	*Machine Learning per a l'optimització d'un model epidemiològic en tuberculosi*
Nom de l'autor:	Martí Català Sabaté
Nombre del consultor:	Pau Andrio Balado
Nom del PRA:	Maria Jesús Marco Galindo
Data d'entrega (mm/aaaa):	01/2018
Titulació:	Màster universitari en Bioinformàtica i bioestadística UOC-UB
Area del Treball Final:	*Programació per a la Bioinformàtica*
Idioma del treball:	*Català*
Paraules clau	*Machine Learning, tuberculosis, epidemiology*

Resum del Treball (màxim 250 paraules): *Amb finalitat, context, metodologia, resultats i conclusions del treball.*

La tuberculosi és una malaltia infecciosa que afecta gairebé a un terç de la població mundial. Per a predir l'evolució de la malaltia és molt útil disposar d'eines computacionals que permeten també provar diferents estratègies de contenció. Actualment els models epidemiològics són massa costosos computacionalment per a poder realitzar moltes simulacions consecutives.

El Machine Learning és una branca de la intel·ligència artificial usat per estudiar i fer prediccions d'un conjunt de dades. En concret, una branca del Machine Learning és la regressió que permet fer prediccions a partir d'un conjunt de dades d'entrada i un conjunt de dades de sortida.

En aquest treball s'han usat tècniques de Machine Learning per a aconseguir models de regressió d'un model epidemiològic. Aquest model serveix per predir l'evolució de la tuberculosi en un districte de Barcelona (Ciutat Vella). S'ha aconseguit ajustat el model a dades experimentals dels últims 10 anys.

Per obtenir el conjunt de dades amb el que s'ha treballat s'han hagut de determinar els paràmetres d'entrada i sortida del model. Pels paràmetres d'entrada s'ha establert un rang de valor plausibles a explorar. El mostreig del model s'ha realitzat amb la tècnica de mostreig Latin Hypercube Sample. Les dades obtingudes a partir d'aquest mostreig són les que s'han usat com a paràmetres d'entrada pel model epidemiològic. A partir de les simulacions realitzades amb el model epidemiològic s'ha aconseguit obtenir el conjunt de variables de sortida amb el que s'ha treballat.

Usant els algoritmes: Logistic Regression, Decision tree classifier, Random Forest, KNeighbors Classifier, Linear discriminant Analysis, Gaussian Naive

Bayes classificator i Suport Vector Machine s'han realitzat diferents aproximacions a les dades obtingudes experimentalment. S'han realitzat un total de tres models diferents per aproximar les dades obtingudes del model epidemiològic.

Els errors d'alguns dels models usats estan dins del rang de tolerància acceptable de les dades epidemiològiques. Aquests models poden ser usats en un futur per a una calibració del model epidemiològic de forma automatitzada per a qualsevol conjunt de paràmetres de sortida.

Abstract (in English, 250 words or less):

Tuberculosis is an infectious disease that affects almost a third of world's population. There exist computational tools that allow us to simulate its evolution. These computational models are very useful to try different containment strategies. Existing epidemiological models have a hard computational cost if a lot of evaluations are need.

Machine learning is a branch of artificial intelligence that can be used to achieve regression models from a set of data. These regression models can be evaluated nearly instantaneously.

In this work, Machine Learning techniques are used to achieve regression models of an existing epidemiological model. This model is used to predict tuberculosis evolution in a district of Barcelona (Ciutat Vella). The model was properly adjusted to the last 10 years tuberculosis data.

There were determined the input and output interest parameters of the model. For each input parameter it was determined an intervals of interest. All intervals were explored using Latin Hypercube Sample technique. This sampling was used to compute multiple evaluations of the epidemiologic model. The results of the model and the sample of parameters were used to train Machine Learning models to adjust the regression models.

The Machine Learning algorisms used were: Logistic Regression, Decision tree classifier, Random Forest, KNeighbors Classifier, Linear discriminant Analysis, Gaussian Naive Bayes Classificator and Support Vector Machine.

Using those algorisms there were obtained different regression models that reproduce correctly the epidemiologic model data. These models can be used to properly calibrate the epidemiologic model for any output desired.

Índex

Índex de figures

Índex de Taules

1. Introducció

1.1 Context i justificació del treball

Avui en dia la tuberculosi (TB) és una amenaça per a la població mundial sent la malaltia infecciosa que causa més morts a tot el món. L'any 2015 es van registrar 10,4 milions de nous casos de TB i va causar un total de 1,8 milions de víctimes a tot el món[1]. Encara que la majoria de la tuberculosi ocorre en països amb recursos limitats, també representa una amenaça potencial important als països més avançats. Això es deu a la naturalesa de la forta interacció de la malaltia amb la dinàmica del VIH i la recent aparició soques de tuberculosi resistents als fàrmacs que s'han usat clàssicament.

La tuberculosi és una malaltia infecciosa crònica transmissible per via respiratòria que afecta als pulmons, tot i que també s'han observat casos de tuberculosi extrapulmonar. La TB pulmonar és una malaltia infecciosa causada pel *Mycobacterium tuberculosis* (Mtb). Quan un malalt esternuda, tus o conversa por infectar a altres individus susceptibles que comparteixen l'entorn, ja que poden inhalar gotes de saliva que contenen el bacteri. Aquesta infecció generalment conduirà a la iniciació d'una resposta immune que pot tenir 3 resultats diferents: (1) l'eliminació del bacteri, (2) una infecció tuberculosa latent, LTBI, o (3) la progressió cap a la malaltia activa.

LTBI es produeix quan la resposta immune de l'hoste conté la infecció del Mtb, però no es capaç d'eliminar-lo completament, ja que els bacils resisteixen la fagocitació dels macròfags. D'altra banda, si la resposta immune de l'hoste no és suficient, suposarà un creixement incontrolat dels bacils que esdevindrà en un cas de malaltia activa. En ambdós casos, la resposta immune conduirà al desenvolupament de granulomes en els pulmons on els bacils queden encapsulats. Els bacils queden físicament continguts i estan immobilitzats per aquests granulomes al llarg de la vida dels hostes. Els pacients que presenten una LTBI no tenen perquè veure afectada la seva vida quotidiana ja que el diàmetre dels granulomes presents en els pulmons és d'uns pocs mil·límetres i no acostuma a afectar el correcte funcionament d'aquests. No obstant això, fins i tot dècades després de la infecció i especialment en el cas d'hostes immunosuprimits poden veure's afectats per la malaltia. De mitjana, aproximadament el 10% de les persones amb LTBI desenvolupen una TB activa durant el transcurs de la infecció. Es calcula que el nombre de persones amb LTBI i, per tant, en risc de desenvolupar la malaltia activa és superior als dos mil milions[1].

Un dels principals problemes al que ens enfrontem és que aquest procés (LTBI) és molt silenciós, causant un retard en el seu diagnòstic. El motius són dos, per una banda no s'han trobat uns símptomes associats a la LTBI i, per tant, és molt difícil d'identificar, per l'altra banda no es coneixen els factors que fan passar a una persona amb LTBI a una malaltia activa. Aquests dos factors fan que sigui molt difícil detectar i poder tractar els individus abans de desenvolupar la malaltia activa. Per tant, hi ha una necessitat d'eines de suport per decidir estratègies de prevenció de la tuberculosi. Segons la coalició europea per la TB[22] la Unió Europea gasta més de cinc mil milions d'euros anualment només en el tractament de la població amb tuberculosi, una suma molt gran de diners que no sempre és ben invertida. És prioritari optimitzar les estratègies d'inversió dels fons destinats a combatre aquesta malaltia de manera que n'obtinguem el màxim benefici per a la salut pública.

Les ciutats europees s'enfronten a un repte important, controlar la infecció i propagació de la tuberculosi. Actualment, existeixen models epidemiològics per a simular l'evolució d'una població. Aquests models són aplicables a la tuberculosi i serveixen per intentar predir l'evolució d'una població en unes condicions concretes.

S'han utilitzat diversos models matemàtics per estimar la dinàmica a llarg termini de les epidèmies de la tuberculosi i l'eficàcia dels programes de control per combatre-la a tot el món. Hi ha una extensa bibliografia sobre models compartimentats, que s'utilitzen per descriure la dinàmica de transmissió de la tuberculosi. Aquests models utilitzen un enfocament d'estratègia de dalt a baix (top-down) on la població es divideix en diferents compartiments (per exemple, susceptibles, exposats, infectats i recuperats en el cas del model SEIR), es fixen fluxos específics entre aquests compartiments i s'utilitzen equacions diferencials no lineals ordinàries per descriure la dinàmica de la malaltia[2].

En aquest treball ens centrarem en un model epidemiològic ja existent[2]. Aquest és molt costós computacionalment. Quan volem calcular l'evolució d'una població usant un determinat conjunt de paràmetres el cost computacional és raonable. Tanmateix fer moltes avaluacions d'aquest mateix model és molt més car. Per exemple, en el cas en el que només coneixem un nombre reduït dels paràmetres i volem imposar un estat final. Resoldre aquest problema necessita moltes simulacions amb moltes configuracions diferents de paràmetres i el cost computacional es dispara. És per aquest motiu que aconseguir un model computacional amb un cost més reduït seria molt útil per a ajustar el model a dades experimentals.

En aquest context és on entra el Machine Learning (aprenentatge automàtic) que és un camp de la intel·ligència artificial que està dedicat al disseny, l'anàlisi i el desenvolupament d'algoritmes i tècniques que permeten entrenar models a partir de les dades disponibles. Es tracta de crear programes capaços de generalitzar comportaments a partir del reconeixement de patrons o classificació.

Dins del Machine Learning, ML, a nosaltres ens interessarà la branca de la regressió on a partir d'un conjunt de dades d'entrada i un conjunt de dades de sortida es construeix un model (que s'avalua quasi instantàniament) que intenta reproduir la informació que conté aquest conjunt de dades. Intentem a partir d'un conjunt de dades poder obtenir un valor aproximat de les variables de sortida usant variables d'entrada que no són iguals a cap de les originals.

1.2 Objectius del treball

L'objectiu principal d'aquest treball és aconseguir usant tècniques de regressió en Machine Learning un model simplificat que ens permeti tenir avaluacions quasi instantànies del nostre model epidemiològic i d'altra banda determinar usant tècniques de classificació en Machine Learning quins són els paràmetres més importants del nostre model.

En concret els objectius d'aquest treball són:

1. Estudiar el model epidemiològic ja existent.

    1.1 Modificar el model epidemiològic permetent un canvi de paràmetres a partir d'un fitxer extern.

    1.2 Determinar els paràmetres d'entrada lliures del model i en quin rang s'avaluaran.

    1.3 Determinar les variables de sortida d'interès del model.

    1.4 Fer un mostrejat extens del model usant la tècnica de mostreig Latin Hypercube Sampling.

2. Aplicar algoritmes de Machine Learning per a l'optimització del model.

    2.1 Obtenir nous models (simplificacions de l'inicial) usant tècniques de Machine Learning.

    2.2 Determinar la configuració més adient per cada algoritme de Machine Learning determinant quin és el model més adequat per al nostre problema.

    2.3 Determinar la importància dels diferents paràmetres d'entrada del model epidemiològic i eliminar, si s'escau, els menys importants.

    2.4 Veure l'ajust del model en funció del nombre de dades usades.

1.3 Enfocament i mètode seguit

El primer que s'ha decidit fer és estudiar el model abans de començar a aplicar els algoritmes de ML ja que coneixent el model es podran plantejar optimitzacions prèvies al ML que serviran per tenir un guany computacional. A més, podrem determinar quines són les entrades i sortides que volem estudiar en el nostre model.

Els algoritmes de ML que farem servir estan inclosos en la llibreria Scikit-learn del llenguatge de programació Python [3].

S'han usat un total de 3 llenguatges de programació. En primer lloc C que és el llenguatge amb el que està implementat el *C simulator* (el simulador epidemiològic que es vol optimitzar) i s'ha fet servir per fer-li les modificacions corresponents. En segon lloc Python que s'ha fet servir per implementar els models de ML. Finalment, s'ha fer servir Matlab per crear la majoria de les figures d'aquest projecte.

1.4 Breu descripció del productes obtinguts

Els productes obtinguts són per una banda:

- **Memòria**: és aquest document, on es recull la part principal de la feina feta al llarg d'aquest projecte.
- **Codis**: recull de tots els codis realitzats per dur a terme aquest projecte, es poden trobar en un GitHub: **https://github.com/marticatalasabate/CodisTFM** juntament amb un document de text 'Readme.md' que explica què conté cada document.
- **Models resultants**: models computacionals resultat dels diferents algoritmes de Machine Learning usats, es calculen a partir dels codis explicats anteriorment.
- **Results.xlsx**: el document resultat de l'escombrat de dades que s'ha fet i que ha servit per a entrenar els models de Machine Learing. Està present també en el GitHub dels codis.

En breus se li afegiran un document d'autoavaluació i la presentació virtual.

1.5 Breu descripció dels capítols de la memòria

El primer capítol és la introducció on s'expliquen les motivacions que han dut a fer aquest treball i quins són els objectius d'aquest.

En el segon capítol s'explica el model epidemiològic que volem optimitzar amb tècniques de ML. Quina és la seva utilitat i què simula, quins són els seus paràmetres i perquè s'han decidit explorar uns i no uns altres.

El tercer capítol tracta dels set algoritmes de ML que es faran servir al llarg d'aquest treball. Se'n fa una breu explicació i al final veiem un petit exemple didàctic per veure en què consisteixen alguns d'ells.

El capítol quatre és el capítol on s'expliquen tots els resultats del treball, els diferents models obtinguts, els errors obtinguts i els resultats desitjables que haurien hagut de tenir els models.

Finalment, el capítol cinc és on veiem les principals conclusions a les que s'ha arribat amb aquest treball i quina és la feina futura que es pot fer a partir d'ara.

Tancant la memòria trobem els apartats del glossari i la bibliografia.

2. Un IBM per Ciutat Vella

A continuació s'explicarà el model amb el que treballarem. És un model creat per simular la dinàmica d'una població tuberculosa usant les dades de Ciutat Vella, un districte de Barcelona.
És un model basat en individus (IBM) per predir l'evolució de la tuberculosi a Barcelona. Inicialment va ser creat per Prats et al [4][2] i després va ser modificat per Vila [5] per tenir en compte la dependència de la tuberculosi amb factors com el gènere, l'edat o l'origen dels individus. Tots aquests models es van implementar en NetLogo. Recentment Puig [6] n'ha fet una adaptació a C obtenint els mateixos resultats. Aquesta adaptació a C és el que anomenem el *C simulator*.
El *C simulator* permet salvar una gran barrera computacional que tenia el NetLogo, mentre l'avaluació d'una simulació a 10 anys vista en NetLogo podia trigar mitja hora usant el *C simulator* podem obtenir els mateixos resultats en menys d'un minut.

2.1 Descripció ODD del *C simulator*

Els models basats en l'individu s'han usat en múltiples camps de la ciència. Com a conseqüència d'aquesta diversitat s'han establert al llarg dels anys moltes maneres diferents per descriure'ls. Per intentar ajustar totes les descripcions dels models IBM es va crear el protocol ODD (Overview, Design concepts and Details) [7]. A continuació podrem veure una descripció del *C simulator* usant el protocol ODD, es poden trobar més detalls en Puig 2017[6]:

Visió general (*Overview*):

Propòsit: l'objectiu d'aquest model és simular l'evolució de la tuberculosi pulmonar en una comunitat, en concret, les dades que s'han utilitzat són les del barri de Ciutat Vella, Barcelona. L'objectiu final d'aquest model serà comprovar i comparar estratègies de prevenció de la tuberculosi pulmonar.

Entitats, variables d'estat i escalars: l'individu d'aquest model són les persones (no sanes). Les persones poden estar en 5 estats diferents: sanes, infectades, malaltes, en tractament o recuperant-se. En la figura 1 es poden veure els 5 estats i les possibles transicions que es consideren. Les persones no sanes (infectades, malaltes, en tractament i susceptibles) són els individus d'aquest model. Les persones sanes són propietats del model que es mantenen constants al llarg del temps. Només calcularem les característiques del nostre individu un cop hagin

entrat en el cicle infecciós. Aquesta simplificació va suposar un reducció dràstica del cost computacional que ha estat testejada i no altera el resultat [2]. Això és degut a que la fracció de població sana és molt gran comparada amb la resta de la població.

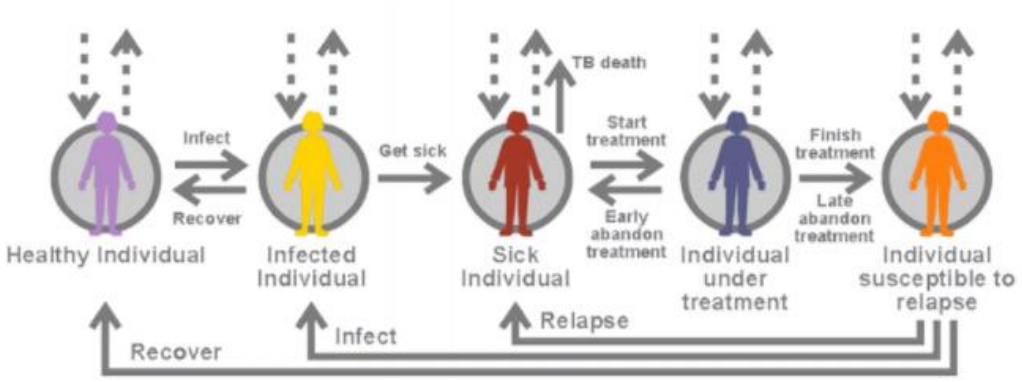

Figura 1, diagrama d'estats del model IBM, les fletxes grises contínues signifiquen un flux de persones entre els diferents estats i les fletxes grises puntejades són les morts. [4]

Les variables d'estat fan referència al seu estat en la infecció tuberculosa, el temps que ha transcorregut en cada estat i el temps de diagnòstic. Altres variables d'estat fan referència a l'individu: edat, origen, sexe i els diferents factors de risc (fumar, diabetis i HIV). Un cop la persona ha estat infectada també s'ha de considerar la presencia o no de cavitació. La població és de 105123 persones que representa la població total del barri de Ciutat Vella (2012).

La simulació té lloc en un espai, però no representa l'espai real del barri. Les simulacions tenen lloc en una xarxa quadrada 501x501, cada espai representa un punt on dues persones es poden trobar, però no tenen un significat explícit. Els individus es mouen aleatòriament al llarg de la xarxa. La unitat de temps és un dia i les simulacions poden durar anys. En el nostre cas les simulacions seran de 10 anys.

Visió general i planificació: inicialment la població és generada i distribuïda aleatòriament d'acord a les distribucions dels paràmetres inicials. Cada dia tots els individus fan una sèrie d'accions que poden variar les seves variables instantàniament: envellir, moure's, infectar-se, emmalaltir, ser diagnosticat, començar un tractament, acabar o abandonar un tractament, recuperar-se i morir. Cada acció es duu a terme o no en funció de l'estat de l'individu. Quan un individu es mor apareix un nou individu dins de la població de sans.

Conceptes de disseny (*Design concepts*):

Principis basics: el model es basa en el coneixement de la història natural i dades empíriques de la TB. Les consideracions més importants del model són: un individu infectat no té perquè desenvolupar la malaltia activa (només un 10% l'acaba desenvolupant), de fet, una persona pot desenvolupar la tuberculosi uns anys després de ser infectada, tot i que la probabilitat de desenvolupar la malaltia decau amb el temps [8][9]. La població infectada només es detecta mitjançant controls a l'atzar, per tant, la gran majoria d'infectats no estaran identificats. D'altra banda, només els malalts de tuberculosi poden infectar a altres individus estenent així la infecció. La probabilitat d'infectar augmenta si aquest individu presenta cavitació. Un cop una persona ha estat detectada el seu tractament triga 6 mesos. La probabilitat de tornar a caure malalts després del tractament és de l'1% en els següents 2 anys.
La gent infectada també pot ser tractada si és detectada, en el seu cas és un tractament que dura uns 9 mesos i serveix per prevenir el desenvolupament d'una malaltia activa [1]. Tot i que el tractament no és 100% efectiu i existeix una probabilitat de recaure i tornar a l'estat infectat.

Emergències: els fenòmens emergents es relacionen principalment amb la dinàmica a llarg termini de la infecció en el nivell de la població. D'una banda, només les persones no tractades que pateixen una TB activa poden contagiar la malaltia. Per tant, el temps de diagnòstic és un paràmetre essencial per a la prevalença de la malaltia. D'altra banda, les persones infectades poden desenvolupar una tuberculosi activa pocs anys després de la infecció. Per tant, es poden detectar conseqüències globals de determinades condicions en un moment precís alguns anys més tard.

Interaccions: les interaccions locals entre individus sans i malalts són explícitament modelats i crucials per a la dinàmica del sistema. Es refereixen a la reunió de dues persones afavorides per la proximitat espacial entre elles i la possibilitat que una d'aquestes persones amb TB activa pugui infectar a l'altra persona.

Estocasticitat: L'aleatorietat s'introdueix en tots els nivells de la simulació. Des de la distribució inicial de les propietats individuals al moviment. A més a més, cada acció està associada a una certa probabilitat i així s'executa d'acord amb un nombre estocàstic. Per tant, un factor molt important del nostre simulador és que els resultats tenen una part aleatòria i el mateix conjunt de paràmetres pot donar lloc a dos resultats una mica diferents.

Col·lectius: es diferencien varis col·lectius dins de la simulació, per origen, per gènere i per edat. Els diferents col·lectius poden tenir temps de diagnòstics diferents, probabilitats d'emmalaltir diferents i també determinen els patrons d'interacció. Els individus es relacionen els uns amb els altres en funció del col·lectiu al que pertanyen.

Observacions: les dades de sortida mostren l'evolució anual del nombre (o prevalença) de persones sanes, persones infectades, persones malaltes, persones en tractament i persones que ja han estat tractades. També mostra el nombre de casos per any (o incidència).

Detalls (*Details*):

Inicialització: per a aquest estudi en particular, la majoria dels paràmetres d'entrada es van obtenir dels informes oficials fets per l'Ajuntament de Barcelona [20][21].

Submodels

Envelliment: tots els individus envelleixen un dia en cada pas de temps.

Moviment: tots els individus es poden moure a un espai contigu.

Infectar-se: si hi ha un individu en una de les celes contigües (o en la seva) un individu no infectat es pot infectar.

Emmalaltir: un cop un individu ha estat infectat aquest pot desenvolupar la malaltia en els 7 anys posteriors a la infecció d'acord a una sèrie de probabilitats[8][9]. La probabilitat d'emmalaltir després del setè any és negligible. Aquesta probabilitat com ja hem dit abans serà diferent en funció de les característiques de l'individu. Hi ha certs factors de risc que poden fer augmentar encara més aquesta probabilitat com és el fet de fumar, ser un individu immunosuprimit o ser diabètic.

Ser diagnosticat i tractat: cada individu té un temps de diagnòstic que se li assigna al emmalaltir, aquest temps és el més perillós perquè és quan pot infectar a la resta d'individus amb els que està en contacte. Un cop diagnosticat comença automàticament el tractament.

Abandonar el tractament: hi ha un probabilitat que un individu decideixi abandonar el tractament abans de finalitzar-lo. La probabilitat de recaure si abandona el tractament depèn del punt on l'individu abandona el tractament i va des del 100% si abandona el tractament durant als primers 15 dies al 1% si l'abandona al cap de 180, la relació és lineal.

Recuperar-se: quan un individu és diagnosticat i completa el tractament (de 180 dies) es recupera de la malaltia, tot i que té una probabilitat de recaure del 1%.

Morir: tots els individus poden morir en qualsevol moment de la simulació, la probabilitat depèn de l'edat, en el cas dels individus malalts és molt més gran i també depèn d'altres característiques de l'individu.

2.2 Paràmetres

Un cop explicat el funcionament del model a continuació explicarem quines són les variables i paràmetres principals del model. El nostre model com acabem de veure és complex i dependrà de moltes variables, les hem separat en 3 tipus segons el seu interès: els paràmetres interns, els paràmetres d'entrada i les variables de sortida.

2.2.1 Paràmetres interns

Els paràmetres interns són aquells paràmetres del model que hem considerat no variar i mantenir constants. Generalment són paràmetres amb una justificació mèdica o demogràfica al darrere i que, per tant, poc sentit té saber quina és la sensibilitat del model a elles o com s'hi ajusta si a la pràctica el seu valor és una dada que no podem canviar.

És el cas de dades com la mida de la xarxa (la xarxa és de 501 per 501 en totes les simulacions).
També de les dades demogràfiques. S'han separat les persones per sexe, origen i grup d'edat per veure com es relacionen entre elles. Per exemple, la probabilitat de morir canviarà en cada cas en funció de les característiques de l'individu, la mitjana distribució de l'edat, les proporcions entre els dos sexes... Dins de les dades demogràfiques també hi incloem les condicions inicials, número de infectats, número de pacients tractats, número de malalts, número de pacients en tractament... Totes aquestes dades s'han agafat dels informes de salut pública que fa la ciutat de Barcelona [20][21] i es poden veure explicades en més detall en el treball fet per Puig [6]. L'única dada que és una estimació és la quantitat d'infectats ja que els infectats no estan identificats i és molt difícil detectar-los.
En aquest grup també hi incloem les dades mèdiques com ara el temps màxim d'infecció, el temps mínim de tractament, la durada dels diferents tractaments, la probabilitat d'emmalaltir en funció de l'estat d'infecció.

En resum els paràmetres interns són aquells que al llarg d'aquest treball romandran constants sense alteració. Per nosaltres són dades que formen part del model.

2.2.2 Paràmetres d'entrada

Els paràmetres d'entrada són aquells paràmetres del model el valor dels quals no és conegut o només se'n coneix una estimació. Per tant, són aquells paràmetres que tenen un interès per nosaltres, ja sigui per determinar si el valor aproximat és correcte com per relacionar-los amb dades experimentals i per veure'n quin és el seu significat.
Els seus valors inicials són els que es van ajustar per reproduir correctament les dades experimentals.
En gran part són els paràmetres sobre els que la salut pública pot incidir i poden canviar al llarg del temps o paràmetres amb un valor desconegut difícil d'estimar.

En total s'han determinat un total d'11 paràmetres d'entrada:

Temps de diagnòstic mitjà per a autòctons i per a estrangers, *diagnose_mean*, és un vector de dimensió 2, la primera component fa referència a la població local i la segona als estrangers. Els seu valors per defecte són 46 i 33 respectivament. Per simplicitat ens referirem a la primera component com a *authocton* i a la segona com a *foreign*.

Probabilitat d'abandonar el tractament, *p_abandon*, s'estima que en un 10% dels casos s'abandona el tractament, i tot i que és una dada coneguda volem veure com afecta el fet que més o menys gent abandoni el tractament. El seu valor per defecte és, per tant, 0.1.

Probabilitat d'infectar per contacte, *p_infect*, és la probabilitat d'infectar a una persona amb la qual estàs en contacte (estar en la mateixa cel·la o en una de les cel·les del costat), aquesta probabilitat es va ajustar al valor 0.06 per reproduir aproximadament els comportament observats experimentalment.

Probabilitat d'emmalaltir, es coneix part de la distribució de la probabilitat d'emmalaltir en funció del temps transcorregut des de la infecció. Se sap que aproximadament un 5% dels pacients recauen en els 2 primers anys i un 5% més al llarg dels 5 següents[8][9].
Aquesta probabilitat la representem en dos variables:
p_sicken_all: és un terme que multiplica a tota la distribució de probabilitats i que, per tant, augmenta o disminueix la probabilitat total. El seu valor per defecte és 1.
f_sicken: és una variable discreta que pren els valors 1, 2, 3 o 4; segons la distribució de probabilitats. Considerem 4 distribucions de probabilitats possibles: parabòlica (1), constant (2), exponencial (3) i mixte (4). En la figura 2 es poden veure les diferents distribucions.
La probabilitat es manté constant al llarg de l'any i només varia quan entren en un nou any d'infecció tal i com es pot veure en la figura 2. El valor per defecte és f_sicken=1.
Aquesta part ha estat introduïda en aquest treball ja que el *C simulator* només contemplava la probabilitat d'emmalaltir com una funció parabòlica amb una probabilitat del 10%, és a dir, el primer cas.

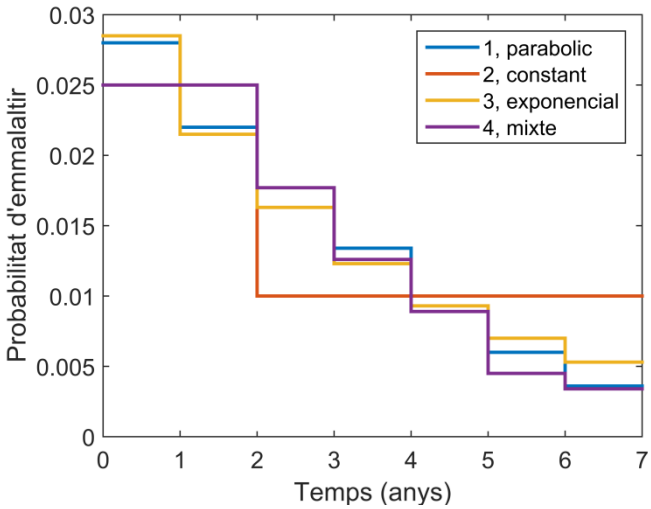

Figura 2, Probabilitat que un individu passi d'una tuberculosi latent a un cas de malaltia activa al llarg de la infecció. Podem veure les 4 possibles distribucions que s'han considerat.

A continuació, explicarem els 5 paràmetres restants que es tracten de factors que fan augmentar la probabilitat d'emmalaltir en funció del col·lectiu al que forma part l'individu.

Factor multiplicador pels nens petits, *f_sicken_child*, és el factor que fa augmentar la probabilitat d'emmalaltir dels nens petits (0-5 anys). Inicialment està estimat en 16.

Factor multiplicador pels joves, *f_sicken_young*, és el factor que fa augmentar la probabilitat d'emmalaltir dels joves (5-15 anys). Inicialment està estimat en 5.

Factor multiplicador pels diabètics, *f_diabetes*, és el factor que fa augmentar la probabilitat d'emmalaltir dels diabetics. Inicialment està estimat en 2.

Factor multiplicador pels immunosuprimits, *f_HIV*, és el factor que fa augmentar la probabilitat d'emmalaltir dels immunosuprimits. Inicialment està estimat en 18.5.

Factor multiplicador pels fumadors, *f_smoking*, és el factor que fa augmentar la probabilitat d'emmalaltir dels individus fumadors. Inicialment està estimat en 1.2.

Aquests 11 paràmetres els explorarem entre la meitat del seu valor per defecte i el doble. Hi ha dos casos que se sortiran d'aquesta norma: és el cas de *f_smoking* ja que la meitat del seu valor és 0.6 cosa que voldria dir que el col·lectiu de fumadors és afavorit per superar la tuberculosi i aniria en contra de les observacions experimentals, per tant, es fixa el seu llindar inferior en 1.0. En el cas de la variable *p_sicken_all* és una variable que augmenta molt la probabilitat d'emmalaltir i per no simular casos irreals es fixa el seu valor entre 0.7 i 1.3. En la taula 1 es poden veure els valors per defecte de cada variable i els intervals d'exploració.

Taula 1, Valors per defecte i intervals d'exploració pels 10 paràmetres que s'exploren en el model IBM estudiat.

Variable	Valor per defecte	Interval		Variable	Valor per defecte	Interval	
		Mínim	Màxim			Mínim	Màxim
authocton	46	23	92	*f_sicekn_child*	16	8	32
foreign	33	16.5	66	*f_sicekn_young*	5	2.5	10
p_infect	0.06	0.03	0.12	*f_HIV*	18.5	9.25	37
p_abandon	0.01	0.005	0.02	*f_diabetes*	2	1	4
p_sicken_all	1	0.7	1.3	*f_smoking*	1.2	1	2.4

2.2.3 Variables de sortida

Estudiarem un total de 7 variables de sortida per a cada any de simulació. Les nostres simulacions seran a 10 anys vista, per tant tindrem un total de 70 valors de sortida per cada simulació, 10 valors per cada variable.

Nombre de malalts, *n_new_sick*: és el nombre de persones que han emmalaltit durant aquell any i han desenvolupat una malaltia activa.

Nombre de infectats, *n_new_infected*: és el nombre de persones que han estat infectades al llarg de l'any, però que no han desenvolupat una malaltia activa. Són les que diem que pateixen una LTBI.

Nombre de curats, *n_heal_infected*: nombre de persones que han passat d'estar infectades a estar curades al llarg de l'any.

Nombre de persones mortes a causa de la tuberculosi, *n_dead_infected*: nombre de persones que han mort al llarg de l'any degut a la malaltia.

Nombre de tractats nous, *n_new_treatement*: nombre de persones que han entrat en tractament al llarg d'aquell any.

Nombre de tractats, *n_new_treated*: nombre de persones que han finalitzat el seu tractament satisfactòriament al llarg de l'any.

Nombre de persones recuperades, *n_recovered*: nombre de persones recuperades.

3. Algoritmes de Machine Learning

En aquest capítol s'explicaran breument els 7 mètodes de ML usats, l'objectiu d'aquest treball no és teòric és per això que no s'entrarà gaire en detall en cap d'ells. La idea és poder-nos fer una idea de quines són les diferències i les semblances entre ells i què és el que fan. Després veurem un exemple fent servir alguns dels mètodes explicats perquè acabi de quedar clar com funcionen els diferents algoritmes.

3.1 Breu explicació dels algoritmes

Logistic Regression (LR) [10][11]

La regressió logística és un model on intentem aproximar la relació entre dues variables amb una corba logística del tipus:

$$y = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1)}}$$

El paràmetres β_0 i β_1 són calculats per a cada paràmetre d'entrada. Finalment la variable de sortida ve determinada per la suma de les contribucions de cada paràmetre d'entrada. Els coeficients de cada paràmetre d'entrada es determinen minimitzant l'error entre l'estimador (y) i el seu valor real a partir d'un procés iteratiu.

Decision tree classifier (DTC) [10][12] **i Random Forest (RF)** [10][13]

Són dos models molt semblants, per això, els explicarem junts.

L'aproximació basada en arbres és un mètode de ML molt usat, però que normalment necessita de moltes dades per a obtenir uns bons resultats. Es creen arbres que són diagrames de construccions lògiques que serveixen per agrupar els diferents conjunts de paràmetres d'entrada amb altres conjunts que determinin variables de sortida semblants.
Aquests arbres després poden servir per aproximar la variable de sortida en funció dels paràmetres d'entrada que hem obtingut i en quin grup de conjunt de paràmetres estaria col·locada la nostra variable de sortida.
La diferència entre els dos algoritmes és el mètode com es creen els arbres.

KNeighborsClassifier (KNC) [10][14]

El KNeighbors classifier és un algoritme que intenta agrupar els paràmetres que tenen unes característiques semblants en l'espai de paràmetres. A diferència del classificador per arbres que acabem de veure on es classificaven els conjunts de paràmetres en funció de la seva variable de sortida en aquest cas classificaríem els conjunts en funció d'ells mateixos, la classificació només depèn dels paràmetres d'entrada que formen el conjunt i no de les variables de sortida que determinen.
Aleshores quan tenim un nou conjunt de paràmetres d'entrada del qual no en coneixem la variable de sortida busquem per associació altres conjunts de paràmetres semblants i en podem aproximar el valor de la variable de sortida.

Linear discriminant Analysis (LDA) [10][15]

El Linear discriminant Analysis és un mètode de regressió lineal que a més incorpora la separació en classes.
Aquest mètode consisteix en ajuntar els conjunts de paràmetres d'entrada que determinen una variable de sortida semblant entre ells i se'n creen classes (grups de conjunts de paràmetres semblants), aleshores dins d'una mateixa classe la variable de sortida s'aproxima usant una regressió lineal.

$$y = \sum_i \alpha_i x_i + \alpha_0$$

Donat un nou conjunt de paràmetres d'entrada la variable de sortida s'aproxima en dos passos. Primer de tot es determina la classe de la qual forma part aquest conjunt i després s'usa la recta de regressió específica per aquell conjunt per determinar la variable de sortida.

Gaussian Naive Bayes classificator (GNB) [10][16]

És un predictor basat en el teorema de Bayes i en la hipòtesis que els paràmetres predictores són independentment entre ells, d'aquesta hipòtesis és d'on ve el nom Naive (ingenu).
El que fem és separar primer els conjunts de paràmetres predictors en classes diferents. Per a cada classe suposem que els diferents paràmetres d'entrada segueixen una distribució gaussiana i en calculem la seva mitjana i desviació estàndard. Coneixent aquestes dades podem determinar en quina de les classes col·locaríem un nou conjunt de paràmetres d'entrada. Un cop tenim la classe podem calcular una aproximació de la variable de sortida.

Suport Vector Machine (SVM) [10][17]

Support Vector Machine és una tècnica que necessita els conjunts de paràmetres d'entrada separats en dos grups, com que en el nostre cas no és així, el mateix algoritme fa servir un algoritme de *clustering* (agrupació) per determinar dos grups de conjunts de paràmetres d'entrada. Aleshores fa servir una classificació binària probabilística per determinar on aniran a parar els nous conjunts de paràmetres d'entrada i en funció de la classe o grup que se'ls hi assigni en determina la variable de sortida.

3.2 Exemple

A continuació posarem un exemple didàctic que ens servirà per veure com s'usen alguns dels mètodes explicats. Considerem un model amb 2 paràmetres d'entrada (A i B) i una variable de sortida C. D'aquest model en tenim 4 conjunts de dades i volem predir per la variable de sortida per un cinquè conjunt de paràmetres d'entrada. En la taula 2 podem veure les dades d'aquest model.

Taula 2, conjunts de paràmetres d'entrada d'un model desconegut. Els conjunts de paràmetres 1, 2, 3 i 4 s'usaran per fer una aproximació del 5é conjunt de paràmetres.

Conjunt	Entrada		Sortida
	A	B	C
1	15	6	2.25
2	12	4	2.15
3	-5	-4	1.12
4	13	-6	1.18
5	11	5	?

Mètodes dels arbres (RF i DTC)

Com que és un cas amb molt pocs paràmetres d'entrada l'arbre que obtindrem serà el mateix pels dos mètodes, però cal recordar que es tracten de dos mètodes diferents i que, per tant, els resultats que obtindrem usant els dos mètodes normalment seran diferents.
Podem agrupar els conjunts de paràmetres 1 i 2 dins d'una mateix sac, de la mateixa manera podem agrupar els conjunts 3 i 4 en un altre sac. Això és perquè els conjunts 1 i 2 tenen una variable de sortida molt semblant i el mateix passa pels conjunts 3 i 4. Fent aquesta classificació l'arbre generat és el que es pot veure en la figura 3.

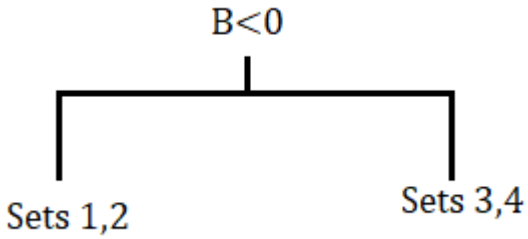

Figura 3, arbre aproximat que podríem obtenir al analitzar les dades de la taula 2 mitjançant un algoritme de creació d'arbres com podria ser DTC o RF.

En un cas real el nombre de paràmetres d'entrada i sobretot el de conjunts és molt més gran i aleshores els arbres observats tindran moltes més bifurcacions.
Aquests arbres també ens donen una idea de quines són les variables més importants a l'hora de determinar la variable de sortida, per exemple, en el nostre model fictici podríem concloure que el paràmetre d'entrada B té un efecte major en la variable de sortida que no pas A.

En el cas d'exemple que estem tractant diríem que el conjunt 5 forma part del grup on hi ha els conjunts 1 i 2, ja que els seus paràmetres d'entrada són semblants. Per tant, podem aproximar la variable de sortida com a: C=2.20.

Mètode de clustering (KNC)

Observant les nostres dades en l'espai de paràmetres s'observa que els conjunts de paràmetres 1 i 2 són molt propers, en canvi, els conjunts 3 i 4 estan allunyats entre ells i respecte l'1 i el 2. És per això que es pot considerar que els conjunts de paràmetres 1 i 2 formen un grup i els altres dos també el formen per si sols:

Grup α: conjunts 1 i 2
Grup β: conjunt 3
Grup γ: conjunt 4

Observant aquest mateix espai es veu que el conjunt de paràmetres 5 és molt proper als conjunts que formen part del grup α. Per determinar la seva variable de sortida es considera que forma part d'aquest grup i es pot aproximar per 2.20.
Cal recordar que aquest és només un exemple didàctic i que en un cas real tindríem molts més conjunts de paràmetres. Normalment en aquest algoritme es limita el nombre de grups que es poden crear al igual que el nombre mínim i màxim de conjunts que en poden formar part.

Regressió lineal per classes (LDA)

L'algoritme primer separa els conjunts de paràmetres en grups (classes). S'ha decidit separar en els mateixos grups que s'ha observat en el cas dels algoritmes basats en arbres (RF i DTC). Per cada grup s'estima la variable de sortida, C, com a:

$$C = \alpha_1 A + \alpha_2 B + \alpha_0$$

Com que hi ha 3 paràmetres lliures en cada cas i el nombre de conjunts de variables és inferior (2) s'ha decidit imposar $\alpha_0 = 0$ per a que el sistema sigui compatible determinat. Cal remarcar que en un cas real aquesta imposició no s'hauria de fer ja que es disposen de dades suficients per aproximar el valor de tots els paràmetres per cada grup. Substituint els diferents conjunts de paràmetres en cada grup s'ha obtingut que:

$$C_{grup\ 1} = +0.3225\ A - 0.4375\ B$$
$$C_{grup\ 2} = -0.0244\ A - 0.2495\ B$$

Les dues rectes de regressió són diferents en funció del conjunt de paràmetres al que formen part. El conjunt de paràmetres 5 forma part del grup 1, substituint en la formula s'obté que la variable de sortida es pot aproximar com a: C=1.36.

En la taula 3 es pot veure un resum dels resultats obtinguts usant els diferents algoritmes de ML. Els diferents algoritmes no donen els mateixos resultats normalment perquè cada algoritme usa unes propietats diferents per aproximar les variables de sortida. En funció del conjunts de dades usat s'obtindran millors resultats amb un algoritme o amb un altre. No hi ha un algoritme que es pugui considerar millor que la resta .

Taula 3, Resultats al aplicar els algoritmes RF, DTC, KNC i LDA en el conjunt de dades que es pot observar en la taula 2.

Mètode	C
RF i DTC	2.20
KNC	2.20
LDA	1.36

4. Implementació i resultats

4.1 Implementació

Tota la part referent al codi epidemiològic està escrita en C ja que és el llenguatge usat en el *C simulator*.
Es va modificar el fitxer principal <main.c>[23] perquè llegeixi els 11 paràmetres d'entrada del fitxer <input.txt> i escrigui les variables de sortida en el fitxer <output.txt>.
Els conjunts de paràmetres usats s'han obtingut a partir del fitxer <LHS.c>[23] encarregat de crear el mostreig en un fitxer de text <sample.txt>.
Finalment, s'ha creat un script en python <simulacions.py>[23] que llegeix el fitxer <sample.txt> i escriu cada conjunt de paràmetres en <input.txt>, executa el codi epidemiològic <a.out> (compilació del fitxer <main.c>[23]) i en llegeix les variables de sortida en <output.txt>. Aquest procés el repeteix per tots els conjunts de paràmetres que hi ha en <sample.txt>. Per acabar escriu totes les dades (paràmetres d'entrada i variables de sortida) en un excel <results.xlsx>[23].

Aquest procés s'ha automatitzat amb la creació d'un makefile <executable>[23] que conté totes les instruccions: compila el codi en c pel mostreig <LHS.c>[23], compila el codi en c del IBM <main.c>[23], executa el mostreig <LHS.o> i executa el codi en python que acaba el procés <simulacions.py>[23].

A continuació el makefile que executem per obtenir l'excel de resultats:

```
1 #!/bin/bash
2 mkdir -p ./data
3 gcc LHS.c -o LHS.o
4 ./LHS.o
5 gcc main.c -lm
6 python simulacions.py
```

Tots els codis de Machine Learning estan en python usant la llibreria sklearn [3].

Es poden trobar tots els codis en el GitHub:
https://github.com/marticatalasabate/CodisTFM

Juntament amb un fitxer Readme.md que conté una explicació del que fa cada codi.

4.2 Mostreig de dades

El primer que necessitem per aplicar els algoritmes de Machine Learning és un mostreig de dades amb el què poder treballar. Com s'ha vist en el capítol 2 en aquest projecte es volen explorar uns intervals determinats per a cada paràmetre. Aquests intervals han estat determinats per poder reproduir un conjunt de paràmetres realista i en un rang de valors raonable.

Per fer el mostreig s'ha decidit apostar pel Latin Hypercube Sample (LHS)[18] en comptes d'usar un Monte Carlo (MC) [19] per garantir unes millors propietats d'ocupació de l'espai. El mètode de MC escull un espai a l'atzar en dins de cada interval en l'espai de paràmetres, en canvi, el LHS separa l'espai de paràmetres en intervals més petits per garantir que totes les parts de l'espai de paràmetres són explorades.

El LHS consisteix en dividir el nostre interval en N trossos on N és el nombre de mostres que volem obtenir, en el nostre cas $N = 1000$. Aquest procés es repeteix per als 11 paràmetres d'entrada. A continuació, associarem aleatòriament, les divisions de les 11 variables entre elles. En total tindrem N conjunts que consistiran en un fragment d'interval per cada una de les variables d'entrada, en total 11 intervals. Aleshores realitzarem un MC dins d'aquests intervals.
El LHS mostreja tot l'espai de solucions del sistema assegurant-nos una millor ocupació de l'espai que no pas el MC.

Per comprovar-ho s'ha fet la mateixa exploració en l'espai de paràmetres usant els dos mètodes. L'objectiu d'un mostreig de paràmetres és aconseguir unes dades no correlacionades que ocupin tot l'espai.
Per comprovar que les dades no estaven correlacionades s'ha calculat la matriu de correlacions entre els diferents paràmetres d'entrada. En ambdós models s'ha vist que les correlacions són mínimes, l'ordre de 10^{-3} en ambdós models. Així que s'ha pogut concloure que cap dels dos models presenta dades correlacionades.
Per comprovar l'ocupació de l'espai s'han realitzat histogrames pels diferents paràmetres d'entrada, en el cas del LHS s'ha pogut observar que els histogrames són completament plans, en canvi, en el cas del MC els histogrames presenten petites desviacions.
S'ha vist, doncs, que al usar LHS s'aconsegueixen dades no correlacionades i a més que ocupen millor l'espai de paràmetres, tal i com s'esperava.

4.3 Anàlisi del model epidemiològic

El resultats obtinguts amb el *C simulator* tenen una component estocàstica important, per tant, les dades amb les que ajustarem el nostre model tindran un interval de tolerància.
D'una banda haguéssim pogut realitzar moltes simulacions i fer-ne la mitjana per eliminar component estocàstic, però computacionalment hagués tingut un cost molt elevat i el resultat obtingut no hagués estat representatiu del resultat que obtindríem al fer només una simulació. És per això que es va decidir treballar amb realitzacions úniques.
L'objectiu del nostre estimador no serà reduir l'error relatiu a 0, sinó que l'error relatiu sigui el més proper possible als resultats de les simulacions.

S'han realitzat 1000 simulacions per avaluar la variància del model usant només 10 conjunts de paràmetres diferents (100 simulacions per cada conjunt) amb una predicció de 10 anys per a cada simulació.

Definim l'error com la mitjana d'errors relatius:

$$Error_{relatiu}Mitjà = \frac{1}{N}\sum_{i=1}^{N}\frac{|S_i - m|}{m}$$

On S_i és el resultat d'una simulació, N el nombre de simulacions i m la mitjana d'aquestes simulacions.

Tot i que en l'apartat 2.2.3 hem definit un total de 7 variables de sortida, nosaltres només treballarem amb una, el nombre de malalts, *n_sick*, ja que és la variable de sortida més important del model i es pot comparar amb dades experimentals.

S'ha avaluat l'error relatiu per cada simulació i se n'ha calculat la mitjana, valor màxim i desviació estàndard. Els errors obtinguts pel primer any de simulació es poden veure en la taula 4 separats per cadascun dels 10 conjunts de paràmetres usats. Es pot veure que els errors són molt semblants entre ells i que no depenen del conjunt de paràmetres usat, per tant la mitjana és un bon indicador per determinar la variància del model.

Taula 4, Error mitjà (EM), màxim dels errors relatius observats (MAX) i desviació estàndard dels errors relatius respecte la mitjana (STD) pels 10 conjunts de paràmetres d'entrada al realitzar 100 simulacions i prenent la mitjana com a valor real.

SET	1	2	3	4	5	6	7	8	9	10	Mitjana
EM	0.091	0.093	0.081	0.076	0.086	0.077	0.075	0.098	0.098	0.079	0.085
MAX	0.316	0.318	0.334	0.229	0.262	0.239	0.261	0.435	0.301	0.261	0.295
STD	0.068	0.067	0.062	0.051	0.062	0.062	0.058	0.081	0.072	0.068	0.065

L'error mitjà esperat per una simulació en el primer any és del 8,5%, el nostre objectiu aleshores serà obtenir predictors amb un error semblant a aquest. No seria bo que un predictor tingués un error més petit perquè aleshores estaríem sobreestimant el model i els resultats del predictor serien més deterministes que els del simulador.

Els error mitjans que s'han calculat pel primer any de simulació també s'han calculat pels tots els anys que duren les simulacions, en la taula 5 se'n poden veure els resultats.

Taula 5, Mitjana de l'error mitjà (EM), màxim dels errors relatius observats (MAX) i desviació estàndard dels errors relatius respecte la mitjana (STD) per a cada any de simulació. Hi ha un total de 10 conjunts de paràmetres diferents, cada simulació s'ha repetit 100 vegades.

ANY	1	2	3	4	5	6	7	8	9	10	Mitjana
EM	0.085	0.085	0.077	0.077	0.074	0.075	0.071	0.072	0.071	0.071	0.076
MAX	0.435	0.335	0.502	0.401	0.425	0.434	0.536	0.511	0.429	0.481	0.443
STD	0.065	0.061	0.059	0.058	0.057	0.054	0.057	0.056	0.054	0.054	0.057

Es pot observar una lleugera tendència a que l'error relatiu decaigui al llarg dels anys, com més avançada està la simulació les simulacions tendeixen al mateix valor pel cada conjunt de paràmetres. La desviació estàndard també decau al llarg dels anys.

Després de veure els resultats de la taula 5 l'objectiu serà trobar predictors amb un error al voltant del 8%. També s'ha de tenir en compte que no volem error per sobre del 45% i que la desviació estàndard de l'error no superi el 6%.

Per comparar el model obtingut usant ML sempre s'usarà l'error relatiu mitjà i s'intentarà que el seu valor sigui el més pròxim possible a les dades observades en les taules 4 i 5.

4.4 Resultats

Les dades es van aconseguir seguint el procés explicat en primer apartat d'aquest capítol. S'han realitzat un total de 1000 simulacions amb conjunts de paràmetres diferents. El cost computacional per aconseguir aquestes dades ha estat de 17 hores.

S'han construït un total de 3 models usant els diferents algoritmes de ML. Els algoritmes de ML usats són algoritmes ja existents que es poden trobar en la llibreria sklearn de python [3].
Usant aquest algoritmes ja existents es construeixen predictors introduint un conjunt de dades separades en paràmetres d'entrada i variables de sortida. El conjunt de dades que usem per a construir aquests predictors són el que anomenem dades d'entrenament. Cada predictor ens permet entrar-li noves dades que són les que usarem per calcular l'error. Les dades que s'han usat per calcular l'error del model són les que anomenem dades de validació i només consisteixen en el conjunt de paràmetres d'entrada.
Les dades d'entrenament i validació normalment formaran part d'un mateix conjunt de dades. A l'atzar es decidiran quines dades formen part de cada conjunt (entrenament i validació), si aquest procés no és aleatori i es repeteix uns quants cops es corre el risc que el model s'ajusti a un conjunt de dades determinat sense representar la totalitat de l'espai de paràmetres que volem mostrejar. El procés de repetició a l'atzar la divisió entre conjunt d'entrenament i conjunt de validació s'anomena cross validation i s'ha realitzat 10 cops en tots els models per a calcular els errors.

Els 3 models que s'han construït a partir dels algoritmes de ML són:

Model 1: *primer any de infecció*, amb aquest model s'ha ajustat el nombre de persones malaltes al finalitzar el primer any de simulació usant els 11 paràmetres d'entrada del model com a predictors.

Model 2: *anys posteriors*, amb aquest model s'ha ajustat el nombre de persones malaltes al finalitzar un any de simulació usant com a predictors els 11 paràmetres d'entrada del model i el nombre de malalts en l'any anterior, en total 12 variables predictores.

Model 3: *predicció del futur*, amb aquest model s'ha ajustat el nombre de persones malaltes al finalitzar un any de simulació usant com a predictors els 11 paràmetres d'entrada del model i el nombre de malalts en l'any anterior, en total 12 variables predictores. La diferencia entre el model 2 i el 3 és les dades d'entrenament que s'han fet servir, mentres que en el model 2 s'han usat dades de qualsevol any tant predir com per validar, en el model 3 les dades d'entrenament usades sempre són del passat i les dades de validació del futur.

4.4.1 Primer any infecció

S'han usat com a variables predictores els paràmetres d'entrada del model i com a variable de sortida el nombre de malalts al finalitzar el primer any d'infecció. No s'ha introduït cap dada referent a les condicions inicials ja que totes les simulacions comencen amb les mateixes condicions inicials.

Les 1000 dades s'han separat en conjunt d'entrenament (850) i conjunts de validació (150), aquest procés s'ha realitzat 10 cops aleatòriament (cross validation). Cada vegada s'ha entrenat el model amb el conjunt de dades d'entrenament i s'ha calculat l'error amb el conjunt de dades de validació.

En la taula 6 es poden veure els resultats que s'han obtingut usant els diferents algoritmes de ML.

Taula 6, Mitjana de l'error mitjà (EM), màxim dels errors relatius observats (MAX) i desviació estàndard dels errors relatius respecte la mitjana (STD) usant diferents mètodes de Machine Learning.

	SVM	**LR**	**LDA**	**KNC**	**DTC**	**GNB**	**RF**
EM	0.365	0.370	0.127	0.365	0.290	0.254	0.350
STD	0.318	0.370	0.105	0.216	0.275	0.210	0.368
MAX	2.448	3.695	0.788	1.822	2.986	1.792	3.695

Els millors resultats s'han obtingut usant el Linear Discriminant Analysis. En la figura 4 es poden veure els errors per cada conjunt de validació usat (un total de 1500 errors de 150 conjunts de validació fent servir 10-fold cross validation) ordenats de menor a major. Es pot veure que clarament el model que té els errors més petits és el LDA que queda lluny de la resta, el segon amb el que s'obtenen millors resultats (GNB) té una estimació d'error relatiu del doble que l'obtinguda usant LDA. Tot i això, l'error màxim que s'ha observat queda una mica per sobre dels errors desitjats, passa el mateix per la desviació estàndard.

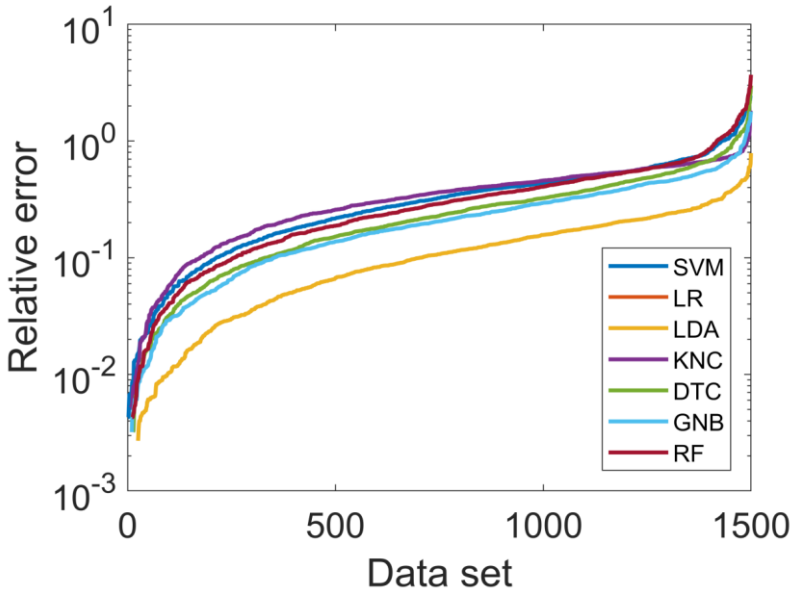

Figura 4 Error relatiu dels diferents conjunts de validació ordenats de menor a major. Podem observar com la corba del LDA és la que va per sota i obté uns millors resultats.

Ajustant els paràmetres de cada algoritme

Cada mètode té una sèrie de paràmetres que es poden ajustar. Després de fer una llarga exploració dels paràmetres que es podien tocar per cada algoritme s'ha observat que els paràmetres per defecte han resultat ser els òptims, en la majoria dels casos.

Per exemple en el cas del LDA, el paràmetre que es pot ajustar és l'algoritme de resolució de sistemes d'equacions, hi ha 3 algoritmes possibles: single value decomposition (svd), least squares solution (lsqr) i eigen value decomposition (eigen). Si calculem l'error per cada un dels mètodes obtenim: $Error_{svd} = 0.127$, $Error_{lsqr} = 0.131$ i $Error_{eigen} = 0.140$. Com es pot observar l'error més baix és el del svd que al mateix temps és el que ens donava per defecte.

El que s'ha observat en gairebé tots els algoritmes és que l'error que s'obté al configurar el millor possible el model és el mateix que se'ns dona per defecte. Els models ja venen optimitzats per obtenir la millor configuració per defecte.

L'únic mètode on ajustant dos dels seus paràmetres s'ha aconseguit reduir l'error és el SVM on s'ha aconseguit reduir l'error mitjà fins a 0.299 canviant el valor dels paràmetres $C = 0.8$ i $\gamma = 0.001$ [10][15]. Tot i això l'error encara queda molt lluny de l'objectiu i segueix sent més del doble que l'error que s'ha obtingut usant LDA. És per aquest motiu que finalment no s'ha decidit incloure la millora dels diferents mètodes en aquest treball de forma automatitzada.

Anàlisi de la importància del paràmetres d'entrada

Com s'ha vist no s'ha aconseguit millorar els models canviant els conjunts de paràmetres de cada algoritme. Però es vol intentar millorar els resultats. S'ha decidit explorar quina és la dependència de l'error de cada model amb el nombre de variables predictores. Es vol explorar la possibilitat de no usar la totalitat de les variables predictores, però es necessita un criteri per decidir quins són les variables que han de quedar excloses.

Tots aquests mètodes usats per a aproximar el nostre model (regressió) també poden ser usats com a classificadors (classificació). Serveixen per crear classes (grups) que classifiquen els conjunts de paràmetres. En alguns casos aquestes classificacions ens permeten obtenir una importància per cada paràmetre d'entrada i quin és el seu pes per calcular la sortida. Dels 7 mètodes de ML usats n'ha ha dos que ens permeten obtenir una classificació de la importància de les variables d'entrada. Aquests dos algoritmes són els mètodes basats en arbres (DTC i RF).

Executant el model usant tots els conjunts de paràmetres com a conjunt de dades d'entrenament (1000) s'ha pogut obtenir la importància de cada variable d'entrada (predictora). Aquest procés s'ha repetit 100 cops perquè a diferència dels altres 5 algoritmes el resultat que s'obté usant els algoritmes basats en arbres és sempre diferent degut a que la creació dels arbre té una component estocàstica. En la taula 7 es poden observar el valor de la importància per les diferents variables d'entrada.

Els dos mètodes donen una classificació molt semblant. El paràmetre d'entrada del model més important és *p_siceken_all* i el menys important (amb molta diferencia) és *f_sicken*. Els números que no fan coincidir les dues classificacions són molt petits així que donarem les dues classificacions per bones i es faran servir a ambdues.

Taula 7, Importància dels diferents paràmetres d'entrada pel model del primer any usant els mètodes de Machine Learning basats en arbres (Decision tree classifier i Random Forest).

	DTC		RF	
Posició	**variable**	**importància**	**variable**	**importància**
1	*p_siceken_all*	0.0991	*p_sicken_all*	0.0983
2	*f_diabetes*	0.0987	*f_sicken_child*	0.0978
3	*f_sicekn_child*	0.0982	*f_diabetes*	0.0975
4	*p_abandon*	0.0935	*p_infect*	0.0958
5	*p_infect*	0.0935	*f_smocking*	0.0950
6	*f_smocking*	0.0929	*p_abandon*	0.0941
7	*authocton*	0.0927	*authocton*	0.0939
8	*foregin*	0.0902	*foregin*	0.0935
9	*f_HIV*	0.0893	*f_HIV*	0.0932
10	*f_sicken_young*	0.0890	*f_siceken_young*	0.0931
11	*f_sicken*	0.0630	*f_siceken*	0.0476

S'ha decidit usar cada model amb només un nombre determinat de variables predictores, per escollir quines variables entren s'ha fet servir la classificació calculada usant DTC i RF (taula 7). Per exemple, si només s'usen 4 variables predictores amb la classificació de DTC aquestes variables seran *p_sicken_all, f_diabetes, f_sicken_child* i *p_abandon*, en canvi, amb la classificació RF s'usaria *p_infect* en comptes de *p_abandon* ja que té una importància més gran.

Usant aquest criteri s'han realitzat simulacions amb els 7 algoritmes de ML ja explicats i les dues classificacions, en cada simulació s'ha servir un número diferent de variables. En totes les realitzacions s'han usat 850 conjunts de variables com a per entrenament i 150 per la validació del model obtingut, cada simulació s'ha repetit 10 vegades.

En la figura 5 es pot veure els errors relatius en funció del nombre de variables usades. Tots els mètodes al reduir el nombre de variable predictores serveix per construir un millor model.

Hi ha 1 mètode on el millor resultat s'obté només usant 3 variables predictores, és el cas del KNC amb el que s'obté un error de: $E_{KNC} = 0.297$. Hi ha 3 mètodes on el millor resultat s'obté usant 4 variables predictores, és el cas del RF, DTC i GNB amb uns errors de: $E_{RF} = 0.241$, $E_{DTC} = 0.238$ i $E_{GNB} = 0.232$. Hi ha 2 models on els millor resultat s'obtenen usant 6 variables predictores, és el cas del LR i SVM amb uns errors de: $E_{LR} = 0.321$ i $E_{SVM} = 0.299$. Finalment el mètode amb el que s'obtenen uns millors resultats usant un total de 8 de les 11 variables predictores és el LDA amb un error del $E_{LDA} = 0.122$.

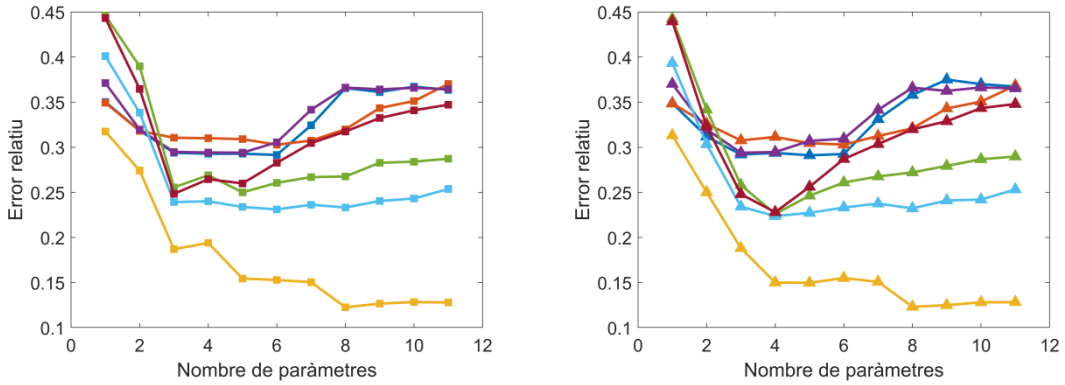

Figura 5 Error relatiu usant 7 mètodes diferents de Machine Learning en funció del nombre de paràmetres usats com a variable predictora. **A** usant classificació DTC **B** usant classificació RF.

S'ha observat doncs que per una banda s'han detectat els paràmetres d'entrada que tenen un efecte molt petit en la variable de sortida i, per tant, es poden considerar eliminar del model epidemiològic. Per altra banda, s'ha vist que al treure aquests paràmetres s'obtenien millors resultats ja que aquests paràmetres distorsionaven i feien augmentar l'error.

Millor model

Al final el millor model que s'ha construït després de provar les diferents combinacions és el Linear Discriminant Analysis amb només 8 dels 11 paràmetres d'entrada (queden fora *f_sicken*, *f_sicken_young* i *f_HIV*) obtenint uns errors de:

$$Error\ mitjà = 0.122$$
$$Error\ màxim = 0.467$$
$$Desviació\ estàndard = 0.095$$

Que comparats amb els valors desitjats:

$$Error\ mitjà = 0.085$$
$$Error\ màxim = 0.295$$
$$Desviació\ estàndard = 0.065$$

Entren dins dels marges acceptables ja que el cost computacional es redueix a pràcticament zero.

4.4.2 Anys posteriors

En aquest model s'han usat com a variables predictores els 11 paràmetres d'entrada i el nombre de malalts inicials, un total de 12 variables d'entrada, i com a variable de sortida (la que s'intenta aproximar) el nombre de malalts finals al cap d'un any.

En total es tenen dades de 9 transicions (entre el primer any i el desé), per tant, es disposa d'un total de 9000 dades. D'aquestes 9000 dades un 85% (7650) s'han fet servir per entrenar el model i un 15% (1350) per validar-lo, aquest procés s'ha repetit 10 cops per a no sobreentrenar el predictor sobre un subconjunt de dades, sinó a tot el conjunt de dades. Al entrenar el model es poden calcular els errors relatius igual que s'ha fet amb anterioritat i s'obté:

$$Error_{SVM} = 0.595$$
$$Error_{LR} = 0.387$$
$$Error_{LDA} = 0.127$$
$$Error_{KNC} = 0.159$$
$$Error_{DTC} = 0.146$$
$$Error_{GNB} = 0.162$$
$$Error_{RF} = 0.156$$

Aquest cop els errors són més baixos per la majoria de mètodes. Tret de SVM i LR amb uns errors molt elevats, la resta de models obtenen errors per sota del 20% que ja són errors raonables.
El millor model obtingut és altre cop el LDA de fet amb el mateix error que ja s'ha obtingut en el cas del primer model.

Millorant el model

En el cas del primer model ja s'ha vist que és complicat millorar el model canviant els paràmetres d'aquest. El que si que s'ha vist en el model anterior és que els models milloraven si se'ls eliminaven les variables d'entrada menys importants que només aportaven soroll. Per decidir les variables d'entrada a eliminar s'ha usat la classificació dels mètodes DTC i RF amb els que obtenim la importància de les diferents variables d'entrada.

A la taula 8 es pot veure que la variable més important és el nombre de malalts inicial, de fet, té molta lògica ja que el nombre de malalts final dependrà dels malalts que hi havia en el passat i en funció dels paràmetres d'entrada augmentarà o disminuirà, però la base són els malalts inicials. La variable que té una menor importància és la *f_sicken*. Aquest resultat ja s'ha obtingut abans així que en un futur es pot considerar eliminar aquesta variable del model epidemiològic.

La importància de les diferents variables d'entrada és diferent en les dues classificacions considerades. Tot i això, les dues classificacions tenen algunes coincidències i no hi ha una diferencia radical entre les dues. La

importància de les variables és molt semblant entre elles en les dues classificacions. La variable *n_sick* acapara molta importància i les diferencies entre les altres variables es veuen afectades.

Taula 8, importància dels diferents paràmetres d'entrada pel model dels anys posteriors usant els mètodes de Machine Learning basats en arbres (Decision tree classifier i Random Forest).

Posició	DTC		RFC	
	variable	importància	variable	importància
1	*n_sick*	0.3240	*n_sick*	0.4495
2	*p_infect*	0.0664	*p_infect*	0.0536
3	*f_sicken_all*	0.0642	*authocton*	0.0530
4	*authocton*	0.0641	*p_sicken_all*	0.0523
5	*f_diabetes*	0.0636	*f_diabetes*	0.0522
6	*f_smoking*	0.0635	*p_abandon*	0.0522
7	*f_sicken_young*	0.0629	*foregin*	0.0520
8	*foregin*	0.0624	*f_smoking*	0.0520
9	*f_HIV*	0.0622	*f_sicken_young*	0.0519
10	*p_abandon*	0.0620	*f_HIV*	0.0519
11	*f_sicken_child*	0.0613	*f_sicken_child*	0.0519
12	*f_sicken*	0.0432	*f_siceken*	0.0279

S'han realitzat simulacions usant cada vegada un nombre diferent de variables d'entrada usant els 7 algoritmes de ML i les dues classificacions ja explicades. En totes les simulacions s'han usat 7650 conjunts de variables per entrenament i 1350 conjunts per validació. Les simulacions s'han realitzat 10 vegades (10-fold cross validation).

En la figura 6 es pot veure els errors relatius en funció del nombre de variables usades. En tots els mètodes reduir el nombre de variable predictores serveix per construir un millor model. La millor configuració que s'ha trobat en cada cas és:

$$E_{SVM} = 0.227 \quad nombre\ variables = 2$$
$$E_{LR} = 0.259 \quad nombre\ variables = 3$$
$$E_{LDA} = 0.121 \quad nombre\ variables = 11$$
$$E_{KNC} = 0.159 \quad nombre\ variables = 12$$
$$E_{DTC} = 0.144 \quad nombre\ variables = 7$$
$$E_{GNB} = 0.132 \quad nombre\ variables = 3$$
$$E_{RF} = 0.137 \quad nombre\ variables = 3$$

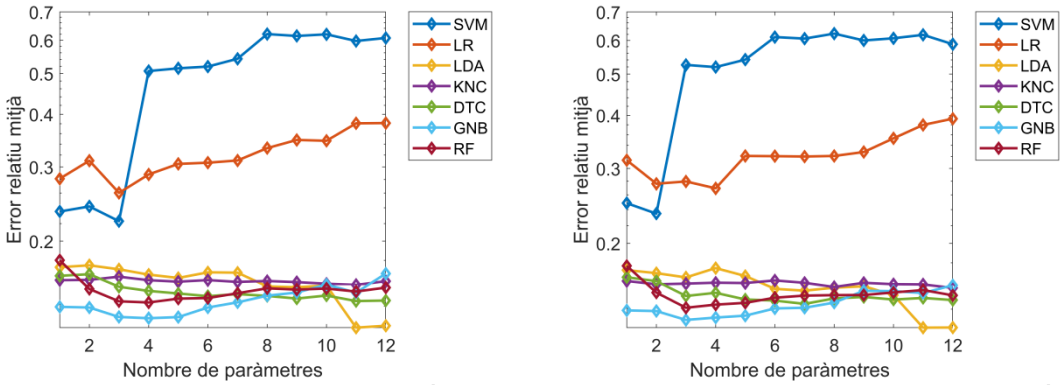

Figura 6, Error relatiu usant 7 mètodes diferents de Machine Learning en funció del nombre de paràmetres usats com a variable predictora. **A** usant classificació DTC **B** usant classificació RF.

Millor model

El millor model construït ha estat usant 11 de les 12 variables predictores amb l'algoritme Linear Discriminant Analysis. La variable predictora que queda fora és *f_sicken*. Els errors obtinguts usant aquest model són:

$$Error\ mitjà = 0.121$$
$$Error\ màxim = 0.453$$
$$Desviació\ estàndard = 0.097$$

Que comparats amb els valors desitjats:

$$Error\ mitjà = 0.076$$
$$Error\ màxim = 0.447$$
$$Desviació\ estàndard = 0.057$$

Entren dins dels marges acceptables ja que reduïm el cost computacional a pràcticament zero.

4.4.3 Predicció del futur

En aquest model s'han usat com a variables predictores els 11 paràmetres d'entrada i el nombre de malalts inicials, un total de 12 variables d'entrada, i com a variable de sortida (la que s'intenta aproximar) el nombre de malalts finals al cap d'un any.
En aquest model només s'han usat dades del passat per calcular el futur. Les dades d'entrenament sempre seran dades de temps passats i les dades del futur seran per temps futurs.

Pel primer model construït usant aquest sistema s'han usat com a dades d'entrenament la transició entre el primer i el segon any de simulació i la resta de dades s'han usat com a conjunt de validació.
El conjunt de validació són la resta de transicions entre el segon i el desè any. S'ha calculat l'error relatiu per totes les dades del conjunt de validació separant per cada any. En la taula 9 es poden observar els errors relatius per al conjunt de dades de validació separats en funció de l'any.

Taula 9, Error relatiu mitjà que obtenim al entrenar un model amb les 1000 dades de la transició entre el primer i el segon any de infecció (1→2). Calculem els errors pels 7 algoritmes de Machine Learning usats i per cada any de transició.

	2→3	3→4	4→5	5→6	6→7	7→8	8→9	9→10	Mitjana
SVM	0.551	0.618	0.667	0.711	0.733	0.759	0.777	0.841	0.707
LR	0.411	0.453	0.484	0.532	0.561	0.582	0.563	0.592	0.522
LDA	0.238	0.403	0.516	0.600	0.662	0.707	0.704	0.731	0.570
KNC	0.204	0.293	0.364	0.405	0.437	0.463	0.467	0.479	0.389
DTC	0.279	0.357	0.403	0.431	0.458	0.475	0.478	0.513	0.425
GNB	0.291	0.356	0.401	0.436	0.461	0.484	0.506	0.561	0.437
RF	0.252	0.356	0.418	0.463	0.488	0.508	0.517	0.548	0.444
Mean	0.318	0.405	0.465	0.511	0.543	0.568	0.573	0.610	-

Hi ha una clara tendència a augmentar l'error al augmentar la distància (temporal) entre els conjunts d'entrenament i els de validació. El millor model s'ha obtingut a partit de l'algoritme de clustering (KNC) i no amb el LDA com en la resta de models. L'error obtingut és massa gran, gairebé 3 cops l'error desitjat.

Per millorar el model s'usaran les 3 primeres transicions (1→2, 2→3 i 3→4) com a dades d'entrenament i la resta com a dades de validació. Hi ha 3 cops més dades que en el model anterior, en total 3000 conjunts de variables d'entrada i de sortida. Es coneix que l'error varia amb el nombre de dades d'entrenament que es fa servir. Per només avaluar la diferencia entre els dos models sense que hi hagi una interferència causada per l'augment del nombre de dades usades s'ha decidit usar nomes 1000 de les 3000 dades. Els 1000 conjunts de dades que s'usaran com a entrenament s'escullen a l'atzar entre els 3000 disponibles. Cada simulació es realitza 100 vegades per evitar efectes degut a el subconjunt de dades utilitzades (100-fold cross validation). Els errors obtinguts per aquest segon model es poden veure en la taula 10.

Hi ha dos factors importants a parar atenció en aquests resultats, per una banda, l'error augmenta amb la llunyania de les dades, cosa ja observada amb anterioritat, i per altra banda veiem que els errors han disminuït i estan dins del rang acceptable desitjat.

S'ha repetit aquets mateix procés usant les dades dels 5 primers anys com a entrenament, els resultats es poden veure en la taula 11. Per últim s'han realitzat les mateixes simulacions usant les dades dels 8 primers anys com a conjunt d'entrenament, només validant les dades en l'última transició. Els resultats d'aquest últim model es poden veure en la taula 12.

Taula 10, Error relatiu mitjà que obtenim al entrenar un model amb 1000 dades de la transició entre el primer i el quart any de infecció (1→2, 2→3 i 3→4). Calculem els errors pels 7 algoritmes de Machine Learning usats i per cada any de transició.

	4→5	5→6	6→7	7→8	8→9	9→10	Mitjana
SVM	0.661	0.707	0.731	0.762	0.875	0.850	0.750
LR	0.401	0.432	0.450	0.469	0.471	0.507	0.455
LDA	0.133	0.152	0.180	0.215	0.224	0.260	0.194
KNC	0.120	0.128	0.148	0.172	0.180	0.198	0.158
DTC	0.250	0.282	0.311	0.333	0.339	0.368	0.314
GNB	0.259	0.298	0.321	0.351	0.370	0.425	0.337
RF	0.289	0.337	0.361	0.380	0.392	0.420	0.363
Mean	0.302	0.334	0.357	0.383	0.394	0.433	----

Taula 11, Error relatiu mitjà que obtenim al entrenar un model amb 1000 dades de la transició entre el primer i el sisé any de infecció (1→2, 2→3, 3→4, 4→5, 5→6). Calculem els errors pels 7 algoritmes de Machine Learning usats i per cada any de transició.

	6→7	7→8	8→9	9→10	Mitjana
SVM	0.742	0.775	0.804	0.885	0.802
LR	0.394	0.414	0.415	0.447	0.418
LDA	0.131	0.154	0.164	0.189	0.160
KNC	0.096	0.106	0.114	0.127	0.111
DTC	0.238	0.267	0.276	0.302	0.271
GNB	0.286	0.315	0.339	0.403	0.336
RF	0.292	0.314	0.333	0.363	0.325
Mean	0.311	0.335	0.349	0.388	----

Taula 12, Error relatiu mitjà que obtenim al entrenar un model amb 1000 dades de la transició entre el primer i el novè any de infecció (1→2, 2→3, 3→4, 4→5, 5→6). Calculem els errors pels 7 algoritmes de Machine Learning usats i per cada any de transició.

	9→10
SVM	0.966
LR	0.395
LDA	0.144
KNC	0.098
DTC	0.215
GNB	0.319
RF	0.303

A partir d'observar els resultats s'ha arribat a tres conclusions importants: En primer lloc observem que l'error augment amb la llunyania de les dades, cosa que indica que per una banda no totes les transicions entre els diferents anys són iguals i per altra banda que com més propers siguin els anys d'entrenament i validació millors resultats s'obtindran.
En segon lloc l'error disminueix al usar dades d'entrenament amb més varietat de transicions.
Per acabar els millors models són els obtinguts amb l'algoritme de clustering (KNC) ja que és el que millor funciona quan la variació de les dades de validació són petites.

El millor mètode aconseguit usant només 1000 dades d'entrenament és el que s'usa per predir la transició del 9é al 10é any usant dades d'entrenament de tots els 9 anys anteriors amb el mètode de KNC. Els errors obtinguts per aquest model són:

$$Error\ mitjà = 0.098$$
$$Error\ màxim = 0.412$$
$$Desviació\ estàndard = 0.071$$

Que comparats amb els valors desitjats:

$$Error\ mitjà = 0.076$$
$$Error\ màxim = 0.447$$
$$Desviació\ estàndard = 0.057$$

Són uns molt bons resultats.

Millorant el model

S'ha intentat millorar aquest model eliminant algunes variables. El procediment seguit és el mateix que en els altres models. En la taula 13 es poden veure les importàncies obtingudes mitjançant els algoritmes DTC i RF. En la figura 7 es pot veure el resultat d'eliminar variables. Només en dos casos (SVM i LR) s'ha aconseguit reduir l'error relatiu mitjà eliminant variables. Crida especial atenció el fet que encara que l'error no

millora al eliminar variables tampoc augmenta significativament en la majoria de mètodes.

Taula 13, Importància dels diferents paràmetres d'entrada pel model per predir el futur usant els mètodes de Machine Learning basats en arbres (Decision tree classifier i Random Forest).

Posició	DTC classifier		RFC	
	Variable	importancia	variable	importancia
1	*N_sick*	0.3080	N_sick	0.4396
2	*p_abandon*	0.0729	P_infect	0.0563
3	*f_sicken_young*	0.0684	autocthon	0.0558
4	*F_HIV*	0.0679	F_smocking	0.0533
5	*F_diabetes*	0.0664	F_sicken_child	0.0533
6	*Foreing*	0.0656	foreign	0.0532
7	*F_sicken_all*	0.0634	F_diabetes	0.0528
8	*authocton*	0.0630	P_abandon	0.0527
9	*P_infect*	0.0618	F_sicken_all	0.0523
10	*F_smocking*	0.0613	F_sicekn_young	0.0521
11	*F_sicken_child*	0.0593	F_HIV	0.0508
12	*F_sicken*	0.0420	F_*siceken*	0.0279

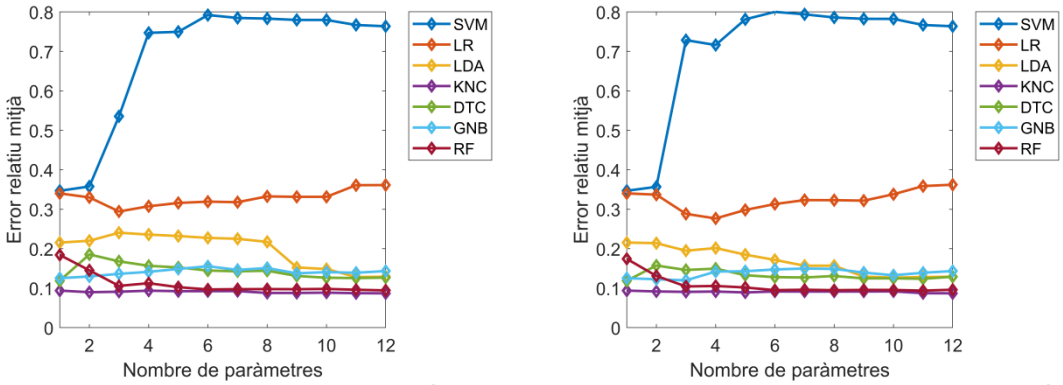

Figura 7, Error relatiu usant 7 mètodes diferents de Machine Learning en funció del nombre de paràmetres usats com a variable predictora pel model que serveix per predir els anys futurs. **A** usant classificació DTC **B** usant classificació RF.

En aquest cas no hem estat capaços de millorar el model tot i eliminar algunes variables d'entrada. Aquest model té una forta dependència amb el nombre de malalts inicial. Tot i això aquest model té marge de millora perquè disposem de més dades per entrenar al model. En el següent apartat on es veu la dependència entre la bondat del model i la quantitat de dades d'entrenament ho podrem veure.

4.4.4 Efecte del nombre de dades

La bondat d'un model varia en funció de la quantitat de dades d'entrenament. En general, com més dades d'entrenament s'usin per entrenar un model s'obtindran millors resultats.

En la figura 8 es poden veure els errors relatius dels nostres models en funció del nombre de dades usades per fer l'entrenament. Les dades usades com a validació sempre són les mateixes, 150 en el cas del primer model i 1000 en el cas dels altres 2 models.

Com era d'esperar l'error disminueix al augmentar el nombre de dades d'entrenament. El model KNC normalment és el que acostuma a funcionar millor quan hi ha molt poques dades. Alguns models saturen a un error determinat (LDA, LR, KNC i SVM), en canvi, n'hi ha que no paren de disminuir (DTC, GNB i RF). De fet, sembla que si tinguéssim més dades usant un algoritme de Random Forest potser podríem arribar a uns millors resultats.

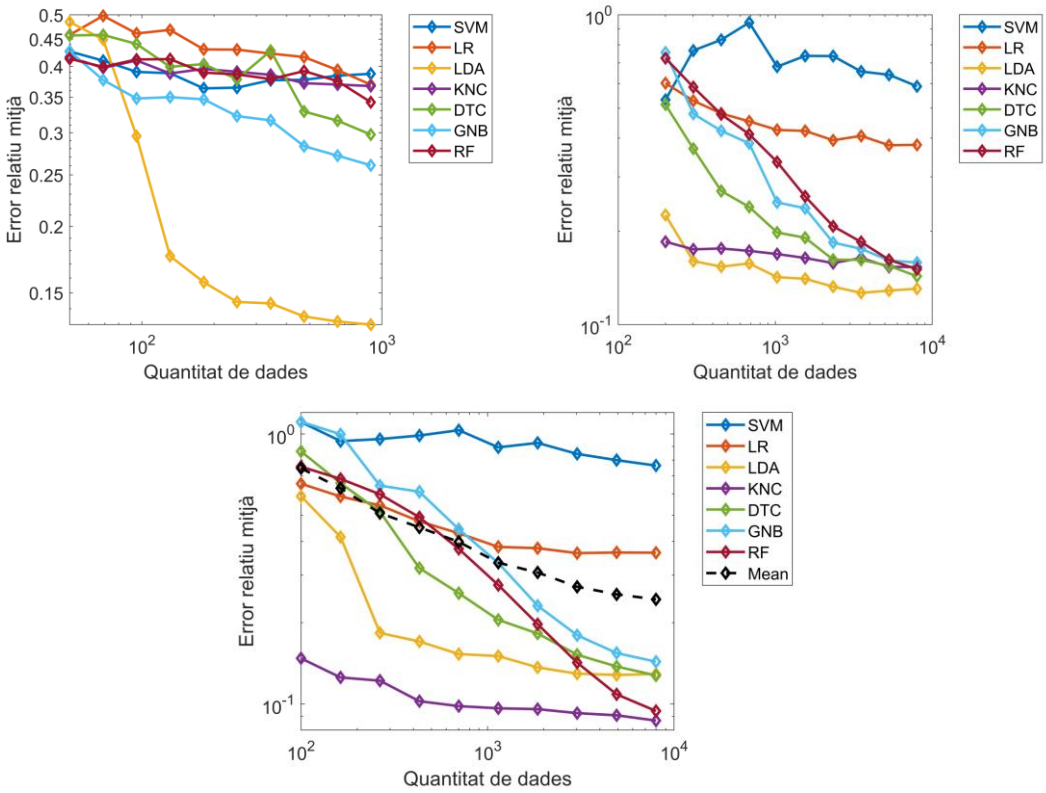

Figura 8, Error relatiu pels 3 models, **A** Primer any d'ifecció **B** Anys posteriors **C** Predicció del futur, en funció del nombre de dades d'entrenament que usem. Podem veure que hi ha algoritmes comés el cas del LDA que satura i té el mateix error encara que augmentem el nombre de dades d'entrenament, però hi ha altres algoritmes que no paren de millorar sempre que introduïm més dades.

Millors models

En la Taula 14 es resumeixen els 3 models construïts usant el màxim de dades d'entrenament possibles al costat s'hi pot veure la comparació amb els errors marcats com a objectiu. L'error del model de predicció del futur és millor que el que havíem obtingut abans, això és degut a que usem com a entrenament les 8000 dades disponibles i no només 1000 tal i com s'ha fet inicialment. L'error d'aquest model és pràcticament el desitjat.

Taula 14, Mitjana de l'error mitjà (EM), màxim dels errors relatius observats (MAX) i desviació estàndard dels errors relatius respecte la mitjana (STD) pel millor predictor que hem aconseguit per cada un dels models que hem dissenyat.

Model	Primer any (LDA)	Objectiu	Anys posteriors (LDA)	Objectiu	Predicció futur (KNC)	Objectiu
EM	**0.122**	0.085	**0.121**	0.076	**0.087**	0.076
MAX	0.467	0.295	0.453	0.447	0.406	0.447
STD	0.095	0.065	0.097	0.071	0.070	0.057

5. Conclusions

La feina feta al llarg d'aquest treball ens ha permès completar els objectius tal i com els havíem reformulat després de la primera modificació. A part, s'ha arribat a algunes conclusions interessants per seguir treballant en aquest marc:

- S'ha aconseguit modificar el *C simulator* perquè els paràmetres d'entrada es puguin introduir des d'un fitxer extern. Els valors de les variables de sortida també s'escriuen en un fitxer extern, de manera que el simulador no ha de ser compilat cada cop que es vulgui usar.

- S'han determinat els paràmetres d'entrada del model, considerant paràmetre d'entrada tot aquell valor que no tingui una justificació mèdica o demogràfica al darrere, també hem considerat paràmetres d'entrada aquells als quals la salut pública pot incidir. Els paràmetres s'han explorat entre la meitat i el doble del seu valor per defecte.

- La variable de sortida més important en la que ens hem centrat en aquest estudi ha estat el nombre de malalts.

- Hem estat capaços d'implementar un mostreig basat en la tècnica Latin Hypercube Sample que a més, ens ha permès veure que té unes millors propietats espacials que no pas el mostreig de Monte Carlo.

- S'han aconseguit implementar els algoritmes de Machine Learning desitjats, usant la llibreria sklearn de python.

- Les configuracions per defecte dels algoritmes de sklearn venen correctament preparats i escullen el millor conjunt de paràmetres possible per minimitzar l'error del model. No ha estat possible millorar els resultats, tret del cas del Support Vector Machine.

- El paràmetre d'entrada que determina la corba de la probabilitat d'infecció té molt poc pes en els models calculats, es pot considerar eliminar-la, com a paràmetre d'entrada del model.

- En els models entre diferents anys d'infecció, la importància del nombre de malalts en l'estat inicial és molt elevada i es poden construir models amb un error relatiu per sota del 20% només considerant aquesta variable. De fet, el que passa és que el nombre de malalts no acostuma a canviar més d'un 20% en un any de simulació.

- S’ha vist que el nombre de dades usades per entrenar el model fa reduir clarament l’error relatiu. Per una banda, hem vist que hi ha models que saturen i encara que augmentem el nombre de dades l’error relatiu no millora. Per altra banda, hem vist que hi ha algoritmes que milloren sempre que els introduïm noves dades, és el cas del Random Forest, que si tinguéssim un escombrat de dades més ampli, segurament seria el que ajustaria les dades experimentals millor.

- Tenint en compte els millors models que hem estat capaços de construir, podem estimar el nombre de malalts al finalitzar el primer any amb un error del 12.2% de mitjana, el nombre de malalts entre els anys dos i deu de simulació amb un error del 12.1% i creiem que podem estimar el nombre de malalts en l’onzè any de simulació amb un error del 8.7%.

Després de tota la feina feta encara es pot anar més lluny. Per una banda, es poden millorar els models obtinguts usant un escombrat de dades més ampli o centrat en intervals més petits. Però l’ús més important que té aquest treball en el futur, serà utilitzar aquests models computacionals per substituir el C Simulator a l’hora de trobar els paràmetres d’entrada que determinen un conjunt de variables de sortida desitjades. Això ens permetrà en un temps raonable, obtenir el conjunt de paràmetres desitjat. Per fer el mateix amb el C Simulator s’estima que es trigaria unes 100 vegades més.

6. Glossari

DTC

Decision **T**ree **C**lassifier, és un mètode de Machine Learning basat en la creació d'arbres classificadors, per a més informació veure pàgina 10.

GNB

Gaussian **N**aive **B**ayes, és un mètode de Machine Learning basat en la hipòtesis de Bayes, per a més detalls veure pàgina 11.

KNC

KNeighbors **C**lassifier, és un mètode de Machine Learning basat en els clusters (agrupacions), per a més detalls veure pàgina 10.

LDA

Linear **D**iscriminant **A**nalysis, és un mètode de Machine Learning basat en la regressió lineal, per a més detalls veure pàgina 11.

LHS

Latin **H**ypercube **S**ample, és un mètode de mostreig a l'atzar que té molt bones propietats d'ocupació de l'espai, per a més detalls veure pàgina 15.

LR

Logistic **R**egression, és un mètode de Machine Learning basat en regressió usant una funció logística, per a més detalls veure pàgina 10.

LTBI

Latent **T**u**b**erculosis **I**nfection, és un dels resultats d'una possible infecció tuberculosa, la infecció es diu que està latent i no desenvolupa la malaltia, per tant, qui la pateix no se n'adona, per a més detalls veure pàgina 1.

Mtb

M*ycobacterium* ***tu****b****e****rculosis*, és un bacil que causa la tuberculosi, és el més comú de tots, per a més detalls veure pàgina 1.

MC

Monte **C**arlo sampling, és un mètode mostreig a l'atzar molt popular i senzill que serveix per mostrejar un conjunt de paràmetres en un espai finit, per a més detalls veure pàgina 15.

ML

Machine Learning, branca de la intel·ligència artificial, per a més detalls veure pàgina 7.

SVM

Support **V**ector **M**achine, és un mètode de Machine Learning, per a més detalls veure pàgina 11.

RF

Random **F**orest, és un mètode de Machine Learning basat en la creació d'arbres classificadors, per a més informació veure pàgina 10.

TB

Tu**b**erculosi, és una malaltia infecciona generalment causada pel bacil Mycobacterium tuberculosis que afecta a un terç de la població mundial, per a més informació veure pàgina 1.

7. Bibliografia

[1] World Health Organization et al. Global tuberculosis report 2016. 2016.

[2] Cristina Montañola-Sales, Joan Francesc Gilabert-Navarro, Josep Casanovas-Garcia, Clara Prats, Daniel López, Joaquim Valls, Pere Joan Cardona, i Cristina Vilaplana. Modeling tuberculosis in barcelona. a solution to speed-up agent-based simulations. In Winter Simulation Conference (WSC), 2015.

[3] http://scikit-learn.org/stable/, 1/12/17.

[4] Joan Francesc Gilabert. Development of a simulator to evaluate public health control strategies of tuberculosis in big cities. B.S. Thesis, Universitat Politècnica de Catalunya, 2015.

[5] Julia Vila. Analysis and individual-based modelling of the tuberculosis epidemiology in barcelona. the role of age, gender and origin. B.S. Thesis, Universitat Politècnica de Catalunya, 2017.

[6] Bernat Puig, Analysis and enhancement of an Individual Based Model strategy to study tuberculosis at a city level. Treball final de grau. Universitat Politècnica de Catalunya. 2017.

[7] Volker Grimm, Uta Berger, Donald L DeAngelis, J Gary Polhill, Jarl Giske, and Steven F Railsback. The odd protocol: a review and first update. Ecological modelling, 221, 2010.

[8] SH Ferebee. Controlled chemoprophylaxis trials in tuberculosis. a general review. Bibliotheca tuberculosea, 26:28, 1970.

[9] Pere Joan Cardona i Joan Ruiz Manzano. On the nature of mycobacterium tuberculosis latent bacilli. European Respiratory Journal, 24, 2004.

[10] Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.

[11] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html, 1/12/17.

[12] http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html, 1/12/17.

[13] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html, 1/12/17.

[14] http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html, 1/12/17.

[15] http://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html, 1/12/17.

[16] http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html, 1/12/17.

[17] http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC, 1/12/17

[18] McKay, M.D.; Beckman, R.J.; Conover, W.J "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". American Statistical Association. 2, 1979.

[19] Robert, Christian, Casella, George, Monte Carlo Statistical Methods, Springer, 2004.

[20] Ajuntament de Barcelona. Anuari estadístic de la ciutat de Barcelona. 2016.

[21] Agència de Salut Pública de Barcelona. La tuberculosi a Barcelona. 2015.

[22] http://www.tbcoalition.eu/what-is-tuberculosis/tb-in-europe/ 10/01/18

[23] https://github.com/marticatalasabate/CodisTFM 10/01/18