Ampliació de funcionalitats i generació de datasets sintètics mitjançant l'eina "Deebee"

Carlos Vázquez Membrado

Resum– Moltes companyies i institucions desitgen treure profit de les dades que generen, siguin en gran quantitat o en mesures més petites, poden sorgir problemes que dificultin el seu ús, vendre-les, etc. Un dels problemes més importants pels enginyers o científics de dades, que són els responsables de processar-les i analitzar-les, és que no són capaços d'utilitzar-les amb la seva màxima eficiència, ja que les mateixes dades no segueixen uns estàndards de qualitat, en alguns casos fins i tot la necessitat de més dades genera problemes. Per evitar el primer cas s'ha de passar per un procés anomenat Data Cleansing, el qual pot ser molt costós. A causa d'això apareix la necessitat de crear una manera de gestionar aquest procés de manera ràpida, parametritzada, automatitzada i reproduïble. Deebee[1] dona una solució a aquest problema, sent una eina per no només per la neteja de dades de tota mena, la qual és fàcilment comprensible i senzilla de fer servir, sinó també per la generació de noves dades per la màxima explotació.

Paraules clau– Generació de dades, Neteja de dades, Omplir NA, Webapp, Dataset sintètic.

Abstract– Many companies and institutions wish to take advantage of the data they generate, whether in large numbers or in smaller measures, problems may arise that make it difficult to use, sell them, etc. One of the most important problems for data engineers or scientists, who are responsible for processing and analyzing data, is that they are unable to use it with maximum efficiency, as the data itself does not follow the quality standards, in some cases even the need for more data causes problems. To avoid the first case one must go through a process called Data Cleansing, which can be very expensive. As a result, there is a need to create a way of managing this process quickly, parameterized, automated and reproducible. Deebee[1] provides a solution to this problem, being a tool not only for cleaning data of all kinds, which is easily understandable and easy to use, but also for generating new data for maximum exploitation.

Keywords– Data Generation, Data Cleanup, Fill NA, Webapp, Synthetic Dataset.

1 Introducció - Context del treball

ACTUALMENT les dades són un recurs molt important per a quasi qualsevol tipus de negoci. L'era en la qual vivim, amb l'ús de tantes eines digitals i l'efecte de la globalització ens donen la capacitat de generar un gran nombre de dades en un temps molt reduït, i és evident que tota aquesta informació ens pot aportar un gran benefici si la podem fer servir de manera adequada. Per això últimament hi ha molta demanda de professionals capaços de gestionar aquestes magnituds de dades i es destina cada cop una més gran quantitat de diners per complir aquests objectius.

Molts enginyers de dades o científics de dades, volent assolir de la millor forma possible els objectius de la seva empresa i intenten recopilar una gran quantitat de dades de diferents fonts, un procés anomenat Data Ingestion. El problema és que totes aquestes dades poden no tenir les mateixes especificacions, i no estar generades de la mateixa manera, creant problemes de compatibilitat o de redundància, ja que cada origen pot estar definit per unes polítiques completament diferents unes de les altres. I això no es limita a dades obtingudes de diferents orígens, també pot passar amb dades de la mateixa companyia.

• E-mail de contacte: cvazquezmembrado@gmail.com

• Treball tutoritzat per: Eduardo Cesar Galobardes tutor (departament)

• Curs 2022/23

A tots aquests inconvenients els anomenem problemes de qualitat de dades i han de ser solucionats pels mateixos enginyers/científics de dades, que podrien utilitzar el seu temps en coses més productives que netejar una gran font de dades. A més a més, no és un treball que pugui ser delegat a una persona no qualificada.

Tot aquest procés és molt necessari per poder obtenir uns resultats adequats i profitosos per al negoci, per tant, tenim un problema molt costós en temps i diners el qual només pot ser solucionat pels mateixos professionals que després hauran d'utilitzar aquestes dades.

A causa d'aquest cost tan elevat que pot suposar una correcta gestió de la Data Quality, hi ha la necessitat de crear una metodologia capaç de reproduir totes les tasques de verificació, validació, neteja, adaptació, d'una manera automatitzada i consistent.

En aquest treball donem per finalitzat majoritàriament tots els processos de Deebe que tinguin relació amb el Data Cleansing (s'aplicaran canvis a millores de qualitat de vida amb baix impacte en el funcionament) i ens hem basat en les noves funcionalitats de l'eina que corresponguin a la generació de dades sintètiques, utilitzades per generar completament nous datasets o per completar alguns ja existents. El problema que poden tenir altres usuaris a l'hora d'intentar entrenar un model amb insuficiència de dades/dades no completes és real, i volem fer que l'eina sigui capaç de gestionar la quantitat més gran possible de tasques amb relació a la gestió de les dades.

Aquesta extensió d'utilitats ha sigut portada a terme amb l'ajuda d'un estudi fet sobre la generació de dades sintètiques[2] que ens dona informació empírica sobre els millors mètodes a implementar dins la nostra eina. Un cop supervisats aquests mètodes amb noves dades i confirmar la seva funcionalitat, s'ha aplicat el més convenient a les aplicacions destinades, l'ompliment de camps buits i la generació de datasets sintètics. Un cop fetes totes les proves necessàries s'ha aplicat un canvi a la GUI de l'eina Deebee[1] per poder utilitzar de la forma més coherent possible aquestes opcions.

La resta d'aquest document està organitzat de la següent manera. La següent secció (secció 2) parla sobre la motivació d'aquest projecte i els objectius marcats i definits de manera més concreta. La secció 3 tracta la metodologia aplicada al desenvolupament del projecte i la planificació en el temps sobre els diferents objectius de manera visual. La quarta secció tracta el desenvolupament complet de tots els objectius de manera ordenada. Finalment la cinquena i última secció és una petita conclusió i alguna reflexió i anàlisi de resultats finals. A la part final trobarem un apartat sobre on obtenir l'eina Deebee amb les noves funcionalitats (inclòs un link al github on està penjat el projecte), una secció d'agraïments, les referències del projecte i 2 apèndixs que aporten informació visual sobre alguns resultats.

2 OBJECTIUS I MOTIVACIÓ

2.1 Motivació

Ara mateix Deebee[1] és una eina desenvolupada com a aplicació web especialitzada en la validació de dades. És capaç d'oferir informes estadístics sobre els dataframes i fer petites correccions a les dades, es pot executar a tots els sistemes operatius capaços d'executar Pyhton3 i és operativa, estable i autosuficient. A més a més, es pot utilitzar per usuaris no experts.

Tot i això, no podem donar-la per finalitzada, ja que pot ser millorada de vàries maneres com l'escalabilitat, millora de seguretat de dades confidencials i millora d'errors concrets de la pròpia aplicació. La motivació principal d'aquest projecte és afegir funcionalitats a l'eina Deebee[1] per poder tenir una manera molt més amplia de gestionar la qualitat de les dades, amb intenció de fer-ho tot de manera modular. A part dels objectius que no van poder ser assolits per la primera versió del projecte, també hem tingut pensat en altres afegits que podrien anar molt bé a la pròpia eina.

Per poder dur a terme tot el que necessitem, comptem amb un estudi previ[2] molt ben documentat sobre diferents mètodes de generació de dades sintètiques i les seves aplicacions, els quals són uns objectius molt similars als nostres. Això ens ajudarà a decidir i implementar de manera més efectiva les solucions determinades.

2.2 Objectius

2.2.1 Opció a carregar un dataframe a través d'una connexió i no de manera manual

Un problema molt comú és el volum de dades que pot contenir un simple dataframe/source de dades. Si multipliquem aquest volum per tots els projectes i subtasques amb diferents dades en les quals poden estar implicats els enginyers/científics de dades, assumim un volum de dades massa gran per tenir còpies de tot en local i utilitzar Deebee[1] d'aquesta manera. Es vol donar solució a aquest problema fent que sigui possible carregar dades mitjançant una crida a un URL específica. D'aquí poden sorgir problemes si l'URL no pertany a un domini públic pel que és una cosa a tenir en compte.

2.2.2 Omplir camps buits (NA)

Una de les tasques principals de la millora de l'eina és poder omplir de manera lògica tots els registres que es presentin buits o en un format incorrecte en un mateix conjunt de dades. En l'estudi[2] comentat en l'apartat anterior del qual disposem, també s'analitzava la manera d'omplir camps no informats per fer més eficient la generació de dades sintètiques, a part de ser un gran afegit per aquest altre objectiu, per l'eina Deebee[1] és un gran afegit de forma individual. En l'estudi s'analitzaven uns mètodes per resoldre aquesta problemàtica, tenint en compte els seus resultats es vol avaluar l'eficiencia de mètodes de machine learning per resoldre aquesta tasca, el enfocament principal és fer-ho de manera predictiva.

2.2.3 Donar visibilitat a classes poc representades dins d'un mateix dataset

En aquest cas l'estudi[2] ens dona vàries opcions, la més valorada és generar dades sintètiques per ampliar les classes necessàries, en aquest cas volem obtenir algun tipus de solució on no sigui necessari un pas tan gran, de fet estaria bé augmentar les dades de les classes menys representades per donar més valor als datasets generats a posterior. Per això s'avaluaran alguns algoritmes senzills i es comprovarà la validesa de les dades.

2.2.4 Generació de datasets sintètics

El problema principal i més tractat en l'estudi[2], aquesta part està parcialment solucionada, tot i que el millor seria supervisar que el mètode menys fructífer (Variational Autoencoder) és realment un mètode poc funcional per aquests tipus de problemes. Per això provarem altres dataframes i contrastarem els nostres resultats amb els resultats de l'estudi amb el qual comptem.

2.2.5 Crear totes les funcionalitats de manera modular

Finalment, totes les anteriors solucions s'han de crear de manera modular, que la implementació funcioni dins de Deebee[1] és important, però no volem limitar aquest funcionament a un únic framework i volem donar llibertat a pròxims projectes per utilitzar aquest desenvolupament.

3 METODOLOGIA I PLANIFICACIÓ

Pel que fa al plantejament del projecte, el criteri general que han d'intentar seguir totes les implementacions és que siguin implementades com a mòduls. Això farà més assequible el manteniment de l'eina i més apte per altres possibles col·laboradors si mantenim el core del projecte de la mateixa manera. El projecte pot tenir una gran quantitat de millores i no només es redueixen a les plantejades en l'apartat anterior; tot i això, decidir l'ordre d'implementació és important per poder tenir unes solucions consistents als objectius plantejats.

3.1 Contacte amb l'aplicació, anàlisi d'inconsistencies i solució de possibles errors

El primer de tot és la familiartització amb el projecte, fer proves per veure quines són les funcions que ja ofereix i les possibles millores més senzilles, d'aquesta manera obtindrem una clara visió sobre el procediment més eficient.

3.2 Implentació d'objectius específics

Un cop es té clar on se situa cada objectiu ens podem posar amb la seva implementació de manera procedural. Cal mantenir un ordre lògic i un historial cronològic i ben informat sobre tots els canvis afegits. Cada objectiu té diverses maneres d'arribar a una solució, més o menys costosa amb millors o pitjors resultats, respectivament. L'estudi[2] comentat prèviament ens dona moltes possibles solucions però simplement fiar-se podria conduir a un procediment erroni, per això caldrà fer proves amb datasets diferents dels que comenta l'estudi i contrastar resultats a les diferents metodologies. El plantejament inicial per les diferents tasques és fer-les independents a l'eina Deebee[1] i integrar-les a posterior de forma modular.

3.3 Testing

El testing haurà de ser per dues parts, la primera que la nova funcionalitat sigui efectiva i funcional, la segona que les posteriors implementacions siguin compatibles amb les primeres, a causa de la implementació modular, és un problema a tenir en compte, ja que cada objectiu haurà de ser independent dels altres. La intenció és fer testing a l'acabar cada implementació, per no concentrar totes els possibles dificultats al final del projecte.

3.4 Diagrama de Gantt

A continuació (Figura 1) una planificació més detallada, amb els objectius concrets marcats amb temps, sent un ideal format a l'inici del projecte.

4 DESENVOLUPAMENT

4.1 Inicialització

Un cop descarregats el projecte de Deebee[1] i el projecte correlacionat a l'estudi de mètodes de generació de dades sintètiques[2] hi ha hagut problemes d'execució degut a les versions necessàries de les llibreries d'ambdós projectes. Per això ha calgut integrar dos "environments"separats on mantenir les versions necessàries per a cada desplegament. Per fer això hem fet servir l'eina Pyenv[3] que ens permet

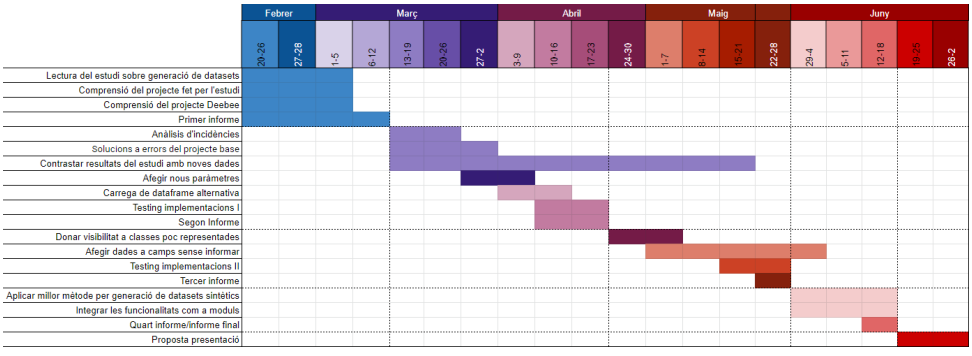

Fig. 1: Diagrama de Gantt per l'organització de les tasques

separar diferents versions de python per a cada environment sempre que siguin versions superiors a la 3.3.

A continuació han aparegut problemes amb les llibreries necessàries per executar el projecte Deebee, tot i que la solució ha sigut ràpida, cal informar d'aquestes integracions per al futur ús de l'eina.

4.2 Implementacions sobre Deebee

S'ha millorat un sistema existent per comprovar que el projecte tingui totes les carpetes necessàries, ja que no comptava amb totes i hi ha hagut algun problema relacionat amb la visualització d'Insights dels dataframes importats.

A més a més, hem afegit una càrrega de fitxers de manera externa, la qual es pot fer servir introduint un URL que apunti cap a un csv.

4.3 Contrastar resultats de l'estudi amb altres fonts de dades

4.3.1 Elecció de noves fonts de dades

Degut a la sorpresa dels resultats empírics de l'estudi[2] on queda demostrat que l'ús d'un Autoencoder funciona relativament malament per a la generació de datasets sintètics, s'estan elaborant proves amb diferents datasets dels ja testejats amb tots els mètodes utilitzats. Per poder confirmar que els resultats són similars davant d'una gran varietat de dades s'ha intentat escollir 3 diferents dataframes amb diferents tipologies de dades.

Aquests dataframes són els següents:

- Energia global del 1990 a 2020.[4] Conté dades senzilles, sobretot numèriques, sobre la quantitat d'energia feta servir per cada país i els percentatges de cada tipus d'energia feta servir per produir electricitat. El "problema"que conté aquest dataset és que treballa amb diferents magnituds de valors en els seus registres i s'ha de tenir en compte al tractar amb les més petites.

- Rendiment d'estudiants en matemàtiques.[5] En aquest cas compta amb poques dades numèriques però no molta quantitat de registres per el que ens serà més fàcil trobar correlació entre les dades.

- Dades de pacients per la predicció d'atacs de cor.[6] Aquest dataset compta amb dades binàries, numèriques i alfanumèriques. Tracta l'edat, sexe, pressió sanguínia, etc, de diferents pacients de 5 hospitals diferents.

S'ha tingut en compte que tots els dataframes tinguin un índex d'usabilitat alt, ja que volem evitar tractar amb dades inventades o que no tinguin cap tipus de sentit/relació entre elles. Per això la pàgina web "kaggle"[7] conté dades sobre cada dataframe en quant a l'origen, descripcions, freqüència d'updates, etc. A més a més 2 dels dataframes es poden fer servir per prediccions, ja que la creació de dades sintètiques és completament compatible amb aquesta temàtica es dona preferència a datasets que tinguin aquestes característiques.

4.3.2 Mètodes de generació de dades

Comptem amb 3 diferents mètodes per la generació de dades sintètiques, utilitzats en el estudi[2] anomenat anteriorment. Aquests mètodes són els següents:

- **Mètode d'estimació amb densitats de kernels (KDE).** Aquest és un model molt bàsic. És un mètode no paramètric per estimar la funció de densitat de probabilitat d'una variable aleatòria basada en nuclis com a pesos. Consisteix a fer una aproximació de la funció de distribució original donant major probabilitat als punts propers al conjunt de mostres.
  Siguin
  $$(x_1, x_2, ..., x_n) \tag{1}$$
  mostres independents i distribuïdes de manera idèntica extretes d'alguna distribució univariada amb una densitat desconeguda f en qualsevol punt x donat. Ens interessa estimar la forma d'aquesta funció f. El seu estimador de densitat de nucli és
  $$\widehat{f_h}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{2}$$
  on K és el nucli, i h és un paràmetre de suavització anomenat amplada de banda. Volem triar una h tan petita com ho permetin les dades tenint en compte de no esbiaixar l'estimador. Aquesta funció és molt ràpida i aquesta h és l'únic paràmetre que necessita com input.

- **Mètode amb Model de mixtura de gaussianes (GMM).** Aquest és un model similar a l'anterior, ja que també es calcula la suma de diversos kernels per crear la funció de distribució de dades. Un model de mixtures correspon a una "Mixture distribution"que representa la probabilitat de distribució de les observacions en una població general. Com que els problemes associats a aquestes "Mixture distributions"venen de derivar propietats a les poblacions generals quan realment, són propietats de les subpoblacions internes, els models de mixtures es fan servir per fer inferències estadístiques sobre les propietats de les subpoblacions donades observacions únicament de la població general sense informació sobre les subpoblacions. En aquest cas fem servir el kernel gaussià únicament. Les k gaussianes poden tenir una ponderació desigual, variàncies diferents entre elles i una matriu de covariàncies substituint a l'anterior valor "h". Tots aquests canvis donen la següent fórmula:
  $$\widehat{P}(X = x) = \sum_{i=1}^{k}(\omega_i \Phi_{\mu_i,\Sigma_i}(x)) \tag{3}$$
  on ω_i,μ_i i Σ_i són els pesos, les mitjanes i les matrius de covariància/vectors de variàncies respectivament de cada gaussiana. Φ és la funció de distribució de gaussiana de paràmetres μ i Σ. Aquesta funció té l'inconvenient de què té bastants més paràmetres a calcular.

- **Mètode d'autocodificador amb soroll (VAE).** Aquest model és l'únic que utilitza xarxes neuronals per aprendre les dades i generar-ne de noves. El procediment consta de les etapes de Codificació de

les dades, afegir soroll a les dades i finalment la descodificació de les dades. Aquest model, aprèn les correlacions entre les dades forçant-les a passar per un coll d'ampolla mentre intenta mantenir el valor original d'aquestes el millor possible. Al codificar les dades a un espai latent de dimensions inferiors obliguem a mantenir aquelles correlacions més importants per tal poder descodificar el valor original intentant invertir el procés de codificació.

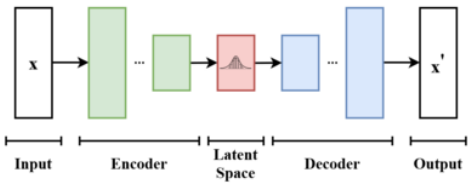

Fig. 2: Variational autoencoder

Al fer servir una xarxa neuronal també hem de tenir molt en compte la quantitat d'hiperparàmetres a proporcionar així com el nombre d'èpoques, ritme d'aprenentatge de la xarxa, la mida del coll d'ampolla, el nombre de capes entre el codificador i descodificador, etc. A més a més, utilitzar aquest model és lent i necessita més dades que els altres.

4.3.3 Resultats i conclusió

Durant el procés de testeig de mètodes amb els nous datasets, també s'han fet diferents proves amb el dataset iris[8], el qual és un dataset molt utilitzat i confiable, a més que tenim proves fetes amb aquest mateix dataset i ens serveix per veure si mantenim els mateixos resultats.

Abans de començar amb les proves s'ha estudiat cada dataset amb l'eïna Deebee, el qual ens ha donat informació molt valuosa com les correlacions que hi ha entre els diferents registres de cada dataset. Per un altra banda, esperàvem resultats difícils d'analitzar pel dataset d'energia global[4], ja que és un dataset difícil de classificar, no té cap registre que esdevingui de les altres dades, simplement actua com una base de dades d'informació sobre l'energia utilitzada de manera global. En canvi, els altres datasets, sí que tenen aquests tipus de registres que els fa més fàcils de classificar. Per una banda, tenim el registre de Species pel dataset iris[8] que ens diu en format text quina Espècie de flor es, tenim el registre HeartDisease pel dataset de predicció d'atacs de cor[6], que ens dona en format Booleà si hi ha alguna malaltia i tenim el registre math score[5] en el dataset de rendiment d'estudiants en matemàtiques, que ens dona en format numèric la puntuació de cada alumne en el test.

Aquest testeig amb nous datasets, s'ha basat en comprovar empíricament els resultats del mètode del variable autoencoder, ja que per una eïna que cada cop es fa servir més, esperàvem millors resultats. Es va començar reproduint el testeig amb el dataset Iris[8], que va donar molts mals resultats de classificació, informació que ja teníem. Així que es van començar a fer proves amb els nous datasets una mica més grans que el dataset iris, ja que es presumia que la necessitat de dades del VAE, feia inviable la seva aplicació. Amb aquests datasets es van donar millors resultats que amb els anteriors, així i tot, no tan bons resultats com amb els mètodes GMM i KDE, que donaven resultats molt similars. Les observacions han mostrat que el GMM determina millor els llindars que el model de KDE, això ens funciona de manera correcta quan hem d'omplir buits en els datasets però es veu esbiaixat quan s'han de generar dades noves, cosa que fa de manera més natural el mètode KDE.

Donat que l'aplicació Deebee té com a premissa ser una eina que es pugui fer servir per a qualsevol mena de dataset i que puguin fer servir diferents departaments amb diferents tipus de dades, l'opció d'implementar el KDE tant per omplir buits com per generar dades noves semblava l'encertada, és un mètode molt ràpid (dels 3 el més ràpid amb diferencia) que necessita molt pocs inputs i funciona bé dins dels resultats observats. Tot i això l'objectiu final serà utilitzar el GMM per omplir buits de dades i el KDE per generar nous datasets.

4.4 Implementacións dins de l'eina Deebee

En aquest apartat, com s'ha comentat anteriorment, hi ha algun problema i és que hi ha conflictes entre les dependències de les llibreries d'ambdós projectes, així i tot, sense fer molt canvi al codi s'ha pogut trobat un punt en comú de versions i s'han especificat dins del fitxer requirements.txt del nou projecte, per la correcta instal·lació d'aquest en qualsevol màquina.
Un cop solucionada aquesta part, s'ha començat a canviar la part d'interfície de la webapp. El resultat, de moment, és el de la Figura 2.

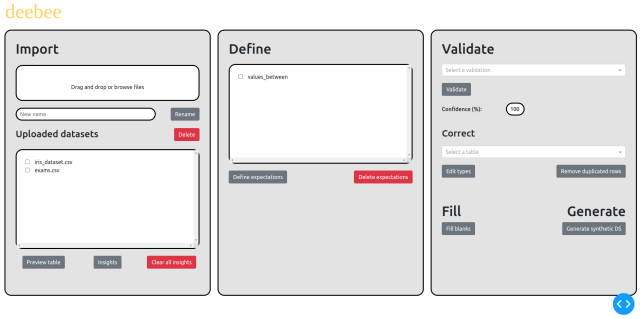

Fig. 3: Nova GUI Deebee

Els afegits són els botons de Fill i Generate, continuant amb l'estètica original, d'aquí sobren unes altres finestres on es configurarà cada opció, com ha sigut la dinàmica de l'eina fins ara.

A partir d'aquí s'ha treballat en la implementació de les dues funcionalitats esmentades.

4.4.1 Implementació de la funcionalitat de omplir NA

Aquesta part està implementada a través del botó de *"Fill blanks"* de l'apartat Fill de l'eina Deebee. En aquest cas, com s'ha comentat anteriorment, s'ha implementat el mètode amb Model de mixtura de gaussianes, aquest model necessita preestablir pesos, mitjanes, variàncies/covariàncies i el nombre de gaussianes a fer servir

(k). Peró donat que volem una implementació el més general possible, s'han implementat dos mètodes per fer la cerca de millors paràmetres de forma automàtica.

En el cas dels pesos, mitjanes i variàncies, utilitzem el mètode d'Esperança-Maximització[2] que ens permet trobar un màxim local per aquests paràmetres. El mètode d'Esperança-Maximització és un mètode iteratiu que permet fer una cerca d'un màxim local de la versemblança d'un conjunt de paràmetres d'un model estadístic, on el model depén d'unes variables desconegudes, les quals alhora depenen d'aquest mateix model. Cada iteració es divideix en dos passos, esperança i maximització. En el pas de l'esperança es calcula l'esperança de la versemblança dels paràmetres respecte de les variables desconegudes, estimades amb els paràmetres de la iteració actual. En el pas de la maximització s'agafen com a nous paràmetres els que maximitzin el valor de l'esperança calculat. Com que acaba al trobar un màxim local, l'ajust de la distribució pot variar considerablement, per remeiar aquest problema una mica, recalculem els valors un parell més de vegades i ens quedem amb els millors d'aquests. Com que el mètode iteratiu va refinant els valors, per la primera iteració necessitem uns valors inicials. Per les mitjanes agafem punts a l'atzar normalment distribuïts, a les variàncies o covariàncies els assignem la matriu identitat i els mateixos pesos per totes les gaussianes.

Pel que fa al mètode per trobar k, ens basem simplement a provar diversos valors de k fins a trobar el que doni millors resultats, aquesta cerca pot portar a temps molt elevats i en alguns casos, no arribar a convergir.

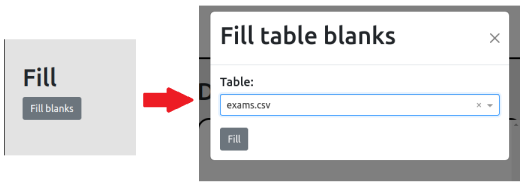

Fig. 4: Flow de la funcionalitat d'omplir buïts

Tot i així aquesta funcionalitat no és perfecte, i compta amb algunes limitacions. Una d'elles és que no podem discriminar columnes específiques on no vulguem omplir buits, donat que és possible que algunes variables necessitin el valor NA com a possibilitat, l'única manera de no omplir aquests buits és importar el dataset sense la columna en qüestió. Una funcionalitat que es podria aplicar, és la de seleccionar quines columnes vols discriminar a l'utilitzar aquesta funció.

Un altra cosa a tenir en compte, és que aquesta funcionalitat crea un nou dataset un cop aplicats els canvis, això significa que s'haurà de canviar de dataset objectiu en el cas de voler fer servir qualsevol altra funcionalitat de Deebee sobre el nou dataset, no és una gran problemàtica però sí que s'haurà de tenir en compte. Veiem el dataset omplert amb nous valors a la secció de *Üploaded datasets"* a la figura 5.

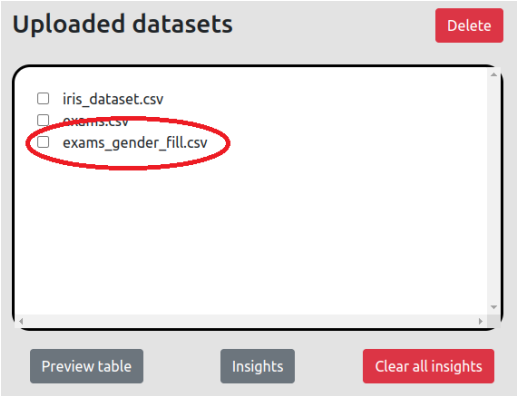

Fig. 5: Nou dataset generat a partir de l'opció Fill

4.4.2 Implementació de la funcionalitat de generació de datasets sintétics

Aquesta part està implementada a través del botó de *"Generate synthetic DS"* de l'apartat Generate de l'eina Deebee. Com hem comentat, aquesta funcionalitat s'ha implementat amb el mètode d'estimació amb densitats de kernels. Aquest model simplement necessita seleccionar 2 paràmetres, la funció kernel a utilitzar i quin valor assignar a h (reescalador del kernel) i, igual que a l'anterior, aquests valors no s'han d'escollir per part de l'usuari.

En el cas de la funció kernel a fer servir, hi havia dues opcions (les dues més usades), el kernel gaussià i l'epanechnikov. Com no suposaven una diferència gaire significativa, es va escollir fer servir el kernel gaussià.

Pel que fa al valor d'h es vol fer servir un valor constant en funció del nombre de mostres i les dimensions d'aquestes o que maximitzi la logversemblança.[2] Per arribar a aquest resultat, les dades han d'estar normalitzades, la variància del kernel serà inversament proporcional al nombre de mostres i s'ha de tenir en compte que el valor final varia exponencialment segons el nombre de dimensions. Fent servir aquestes suposicions trobem el valor h que volem i s'observen bons resultats quan hi ha dades no correlacionades i empitjora amb l'aparició de correlacions entre diferents registres.

Per fer la cerca del millor valor d'h, agafem el valor constant prèviament esmentat com a guia per on començar i, a partir d'aquest, busquem uns límits superiors i inferiors sobre on es pot trobar el valor d'h que maximitzi la logversemblança. Per tal de no trigar massa cercant aquest rang, pels casos on el valor òptim es trobi lluny d'aquesta guia, fem les proves cada vegada més separades d'on podrien estar els límits. Un cop tenim límits, podem utilitzar la cerca d'ajust quadràtic per anar reduint aquest interval fins tenir un valor prou precís. Per decidir quan un valor d'h és millor que un altre fem servir el valor de la logversemblança del model amb validació creuada, és a dir, separem les dades en dos grups, utilitzem un del grups pel model i l'altre per avaluar la logversemblança. A l'utilitzar la logversemblança amb dades que el model no coneix, assegurem que aquesta funció tingui un màxim, i si utilitzem els mateixos grups durant tota la cerca, garantim que la funció serà suau, complint així les dues condicions necessàries per utilitzar la cerca d'ajust quadràtic. Si analitzem l'efectivitat d'aquest nou valor d'h, veiem que

dona bons resultats sense importar si les dades estan correlacionades o no (Taules 1 i 2) a costa d'un increment en el temps d'execució.

Pel que fa al "flow"de la funcionalitat dins de l'eina, és molt similar al de la funcionalitat d'omplir buits, però en aquest cas, s'ha d'escollir la columna objectiu a partir de la qual volem generar les dades i s'haurà d'escollir el nombre d'exemplars de cada classe a generar. Això és important, ja que ve lligat amb les limitacions de la funcionalitat.

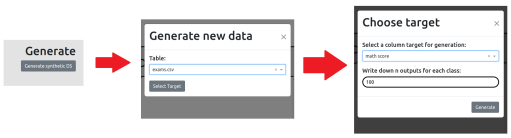

Fig. 6: Flow de la funcionalitat de generar dades

En aquest cas la generació és molt còmode, el dataset generat es crea en una carpeta del projecte, en cas de voler tractar-lo dins de la pròpia eina, s'haurà d'importar com qualsevol altre dataset. Veiem un exemple de generació d'un dataset sintètic dins d'una nova carpeta a la ruta *"deebee-main/data/generateds"* a la Figura 7.

Fig. 7: Nou dataset generat a partir de la funció Generate

Per un altra banda, la generació té una limitació clara que no té la funcionalitat d'omplir buits i és que els valors que pertanyen al camp objectiu, hauran de formar part d'un domini discret, és a dir els possibles valors han de ser limitats, de no ser així la cerca dels millors paràmetres es farà insostenible i no serà possible cap generació.

Això fa que un dels nostres dataset de prova (Energia global del 1990 a 2020.[4]) no sigui un bon dataset on fer servir aquestes funcionalitats, això es perquè és una base de dades i no un dataset de classificació.

4.5 Mostra i anàlisi de resultats

En aquest apartat es mostraran els diferents resultats que podem extreure de les noves funcionalitats, cal dir que els datasets escollits són bastant diferents entre ells i fan possible ensenyar com podem obtenir una solució ideal o com podem obtenir una solució menys confiable.

4.5.1 Resultats de la funcionalitat Fill

Per analitzar els resultats passarem per cadascun dels diferents datasets mostrant els resultats que podem obtenir de millor a pitjor. La manera en la qual s'han creat resultats és introduint valors NA en posicions aleatòries dels datasets i després, passant aquests per la funció fill. En alguns casos es parlarà de les correlacions entre variables de cada dataset, es poden trobar les taules de correlacions a l'apèndix A.1.

- **Dataset Iris [8], molt bons resultats.** En aquest cas, podíem assumir que omplir funcionarien correctament, i és que el dataset iris té molt poques variables i hi ha molta correlació entre totes elles.

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Length	Petal.Width	Species	Species
0	5.1	3.5	1.4	1.413931	0.2	Iris-setosa	Iris-setosa
1	4.9	3.0	1.4	1.197175	0.2	Iris-setosa	Iris-setosa
2	4.7	3.2	1.3	1.316379	0.2	Iris-setosa	Iris-setosa
71	6.1	2.8	4.0	4.268779	1.3	Iris-versicolor	Iris-versicolor
72	6.3	2.5	4.9	4.751291	1.5	Iris-versicolor	Iris-versicolor
73	6.1	2.8	4.7	4.147057	1.2	Iris-versicolor	Iris-versicolor
147	6.5	3.0	5.2	5.847909	2.0	Iris-virginica	Iris-virginica
148	6.2	3.4	5.4	5.479136	2.3	Iris-virginica	Iris-virginica
149	5.9	3.0	5.1	5.164784	1.8	Iris-virginica	Iris-virginica

Fig. 8: Fill de dades al dataset Iris, els camps omplerts estan enquadrats en vermell, amb el camp original al costat per poder fer la comparativa.

Observem a la figura 8, com ha omplert el camp "Species"de manera perfecta, ja que es tracta de valors discrets i el camp de "Petal.Length"també apropant-se molt als valors reals. Com hem comentat aquests resultats són els ideals que busquem en qualsevol dataset, depèn de la capacitat de cada dataset per poder ser reduït a les variables més importants per obtenir aquests resultats.

- **Dataset rendiment estudiantil [5], bons resultats.** Aquest dataset és similar bastant similar a l'anterior quant a quantitat de variables, així i tot, té unes correlacions molt més dèbils pel que podem esperar resultats menys precisos, on té les relacions més fortes és entre les notes dels tres tests.

	gender	raceethnicity	parental level of education	lunch	test preparation course	math score	math score	reading score	reading score	writing score	writing score
0	female	group D	some college	standard	completed	59	65.0	70		78	
105	female	group D	some high school	standard	none	74		75	83.0	74	
245	male	group C	high school	standard	completed	88		75		72	79.0
310	male	group D	high school	standard	completed	88	77.0	77		78	
420	male	group D	bachelor's degree	standard	none	76		65	68.0	70	
555	female	group E	high school	free/reduced	none	66		66		75	67.0
684	female	group E	high school	standard	none	85	84.0	86		88	
747	female	group E	some high school	standard	completed	90		96	96.0	98	
899	female	group B	some college	standard	completed	92		100		100	99.0

Fig. 9: Fill de dades al dataset de rendiment acadèmic, els camps omplerts estan enquadrats en vermell, amb el camp original al costat per poder fer la comparativa.

Com podem observar a la figura 9, en aquest cas hem tret dades de tres camps diferents, i quant a valors numèrics podem afirmar que funciona de manera òptima, a part d'algun valor que se'n va una mica de l'original, els marges estan ben marcats.

- **Dataset dades mèdiques[6], resultats casi aleatoris.** Ens trobem amb un dataset molt més complicat que els anteriors, i és que a part de tenir moltes variables amb informació molt diferent, no tenen quasi cap correlació entre elles. En aquest cas mostrem 3 variables diferents que han sigut omplertes amb valors NA (per separat, és a dir, no s'han tret els valors a la vegada, sinó que s'ha executat la funció de fill 3 vegades).
Veient la figura 10, s'observa una gran diferència entre resultats, els categòrics no tenen una gran estabilitat i els numèrics obtenen valors, en la majoria dels casos, molt diferents dels reals. Aquest és un cas clar de què els datasets que vulguem omplir, s'han de tractar abans d'alguna manera si no tenen correlacions clares entre les seves dades i tenen valors molt dispars, si no

	Age	Sex	ChestPainType	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	ST_Slope	HeartDisease
1	49	F	NAP	TA	160	180	0	Normal	156	154.0	N	1.0	Flat	Flat	1
105	57	M	ATA	ATA	140	260	1	Normal	140	176.0	N	0.0	Up	Up	0
215	30	F	TA	ATA	170	237	0	ST	170	150.0	N	0.0	Up	Up	0
310	65	M	ASY	ASY	155	0	0	Normal	154	171.0	N	1.0	Up	Flat	0
407	62	M	ASY	ASY	115	0	1	Normal	72	95.0	Y	-0.5	Flat	Flat	1
555	58	M	NAP	ASY	150	219	0	ST	118	110.0	Y	0.0	Flat	Up	1
656	62	F	ASY	ATA	124	209	0	Normal	163	140.0	N	0.0	Up	Flat	0
750	46	F	NAP	NAP	142	177	0	LVH	160	159.0	Y	1.4	Down	Up	0
899	61	M	ASY	NAP	148	203	0	Normal	161	138.0	N	0.0	Up	Flat	1

Fig. 10: Fill de dades al dataset de dades mèdiques, els camps omplerts estan enquadrats en vermell, amb el camp original al costat per poder fer la comparativa.

aconseguim un dataset sobre el qual l'algoritme pugui treballar correctament, no aconseguirem cap resultat eficaç.

- **Dataset energia gloabl[4], no es pot omplir.** Finalment, tenim un dataset del qual no esperem gaire cosa, a causa de la seva naturalesa, i és que no tractem aquest dataset com un que puguem fer servir per ingestar en un model, sinó com una base de dades. Després de vàries execucions per intentar omplir el camp çountry", i diversos intents per fer d'aquest dataset, un més òptim, hem vist que era impossible generar dades que omplissin els buits. El problema principal és que teníem massa poques dades de cada classe per fer funcionar, de manera que el mateix algoritme detecta que NA és una possible opció a retornar, cosa que no és acceptable. D'aquesta manera tenim un nou problema, necessitem suficients dades sobre una mateixa classe per poder omplir buits.

4.5.2 Resultats de la funcionalitat Generate

En aquest cas és més complicat subdividir entre qualitat de resultats, ja que els resultats són prometedors fins i tot pels casos menys esperats.

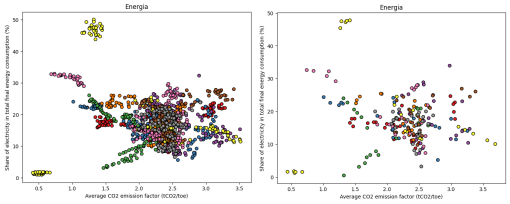

Fig. 11: Comparativa entre dataset d'energia global original (esquerra) i dataset sintètic (dreta), la columna objectiu ha sigut çountry"

Com veiem a la figura 11, tot i que ha sigut impossible generar un fill de buits pel dataset d'energia global, la generació de dades sintètiques ha anat prou bé, en aquest cas, s'han generat 5 registres per cada país diferent (variable çountryçom a objectiu) i es pot observar com segueixen els patrons de les dades originals.

Les columnes que es mostren en el gràfic tampoc són aleatòries, s'ha escollit aquelles columnes que més correlació tenien amb la columna objectiu. Per veure les comparatives dels altres datasets cal dirigir-se a l'Apèndix A.2.

És a dir, quant a la generació tenim un sistema que de primeres sembla més sòlid, ara bé, també han sorgit problemes. En el cas del dataset de rendiment acadèmic[5], en intentar generar dades seleccionant com a objectiu de la generació una de les columnes numèriques (math score), no ha sigut possible executar la generació, i és que aquesta únicament funciona en camps on les classes són finites, si seleccionem una variable sense aquestes característiques la generació de dades serà impossible.

Així que finalment, el que podem treure com a conclusió, és que per generar noves dades, no és tan important modificar el dataset per obtenir les columnes amb millor correlació cap a l'objectiu, però sí que s'ha de triar un objectiu vàlid. Pel que fa a l'ompliment de buits, per aconseguir bons resultats caldran tractar el nostre dataset, tenir les correlacions clares i ajustar els dominis de cada variable, a més a més de tenir suficients registres de cada classe per poder executar l'algorisme.

5 Conclusió

Després de confirmar quins són els millors mètodes per utilitzar en cadascuna de les funcionalitats i fer proves amb elles, podem assegurar que s'ha complert l'objectiu d'implementar funcionalitats de generació de datasets sintètics i ompliment de buits en l'eina Deebee.

També hem complert amb un dels objectius que teníem des d'un principi i és que les noves implementacions són completament modulars js que no interfereixen de cap manera en les altres funcionalitat de l'eina Deebee i es poden fer servir de manera independent, així i tot, ambdues parts, tant les noves implementacions com les ja existents, són un gran complement per l'altra part, fent d'aquest projecte una eina molt completa.

Tot i així, s'ha de veure també la part millorable i és que les noves implementacions no són mètodes completament funcionals per tots els casos, tenen molt marge de millora i per obtenir un dataset sintètic que segueixi els valors i correlacions de les dades originals, l'usuari haurà d'aportar coneixements sobre les dades i sobre la gestió d'aquestes. El millor que es pot fer per aconseguir els millors resultats possibles és fer vàries generacions de datasets. Intentar trobar les correlacions i treure les columnes amb menys informació perquè els algoritmes i models puguin treballar el millor possible amb les dades aportades.

De moment no ens serà possible tenir un dataset sintètic perfecte a partir d'un dataset sense manipular (en la majoria de casos) però és un pas cap a la bona direcció.

5.1 Punts a millorar

Tot i tenir un sistema de generació funcional, cal dir que és millorable com a eina. Evitant petits detalls d'implementació o planificació que podrien haver sigut més estructurats durant el procés, la millora principal que seria molt positiva és que es podrien afegir diferents mètodes a cadascuna de les funcionalitats, per deixar que l'usuari escollís i tingués més opcions fins a trobar el resultat que desitja. També podríem aportar més personalització a la configuració de les implementacions, tot i que els mètodes emprats per obtenir els millors paràmetres per cada cas són fiables, pot ser que l'usuari tingui una sensació de caixa negra al tenir tan poques opcions de configuració.

Així i tot, continuant amb l'opció d'implementar nous mètodes de generació de dades, ja que cap mètode dels testejats ens dona una fiabilitat insuperable, és fàcilment deduïble que alguns datasets funcionaran millor que altres

amb diferents mètodes, per això crec que és un canvi a futur molt prometedor.

Repositori de Deebee

La nova versió de Deebee s'ha penjat en el meu github personal a l'URL https://github.com/Wasques/DeebeeExtension.
Compta amb un README integrat que explica els passos d'instal·lació i execució de l'eina Deebee.

Agraïments

Volia agrair primer de tot a en Martí Miranda, antic company de carrera, per la creació de l'eina Deebee mantenint un codi molt net i clar, i ajudar-me en els problemes del Deebee original que m'hagin pogut sortir durant el desenvolupament. Per una altra banda, m'agradaria agrair també a en David Candela pel seu estudi sobre la generació de dades sintètiques i la creació de la llibreria synthdata que ha sigut crucial pel desenvolupament del projecte. Finalment, vull agrair al meu tutor Eduardo Cesar Galobardes per la paciència i la comprensió durant tot el desenvolupament del projecte.

Referències

[1] Martí Miranda;*Data Quality tool to make passive validations and massive corrections on data*, Data Engineering UAB, https://github.com/martimm00/deebee

[2] David Candela; *Estudi empíric de mètodes de generació de dades sintètiques i les seves aplicacions*, MatCAD UAB, https://github.com/Littleote/TFG/blob/master/pyproject.toml

[3] Dominic Fraser; *How to manage multiple Python versions and virtual environments*, Setembre 2018, https://www.freecodecamp.org/news/manage-multiple-python-versions-and-virtual-environments-venv-pyenv-pyvenv-a29fb00c296f/

[4] MIKHAIL-MKS; *World energy data 1990 - 2020*, https://www.kaggle.com/datasets/shub218/energy-data-1990-2020?resource=download

[5] KIATTISAK RATTANAPORN; *Student performance prediction*, https://www.kaggle.com/datasets/rkiattisak/student-performance-in-mathematics

[6] FEDESORIANO; *Heart Failure Prediction Dataset*, https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

[7] *Kaggle*, https://www.kaggle.com/

[8] R. A. Fisher; *Iris Dataset*, 1988, https://archive.ics.uci.edu/dataset/53/iris

[9] Gianluca Malato; *Filling blanks in a dataset with R*, Gener 2019, https://medium.datadriveninvestor.com/filling-blanks-in-a-dataset-with-r-90ece2329b2c

[10] Satyam Kumar; *7 Ways to Handle Missing Values in Machine Learning*, Juliol 2020, https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e

[11] Jason Brownlee; *8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset*, Agost 2015, https://t.ly/tX-KH

[12] *Kernel Density Estimation*, https://en.wikipedia.org/wiki/Kernel_density_estimation

[13] *Gaussian Mixture Models Explained*, https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

[14] *Understanding Variational Autoencoders (VAEs)*, https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[15] Ane Berasategi; *Earth mover's distance. A semantic measure for document similarity in semantic search*, Abril 2019 https://towardsdatascience.com/earth-movers-distance-68fff0363ef2

[16] Jason Brownlee; *A Gentle Introduction to Expectation-Maximization (EM Algorithm)*, Novembre 2019 https://machinelearningmastery.com/expectation-maximization-em-algorithm/

APÈNDIX

A.1 Heatmaps

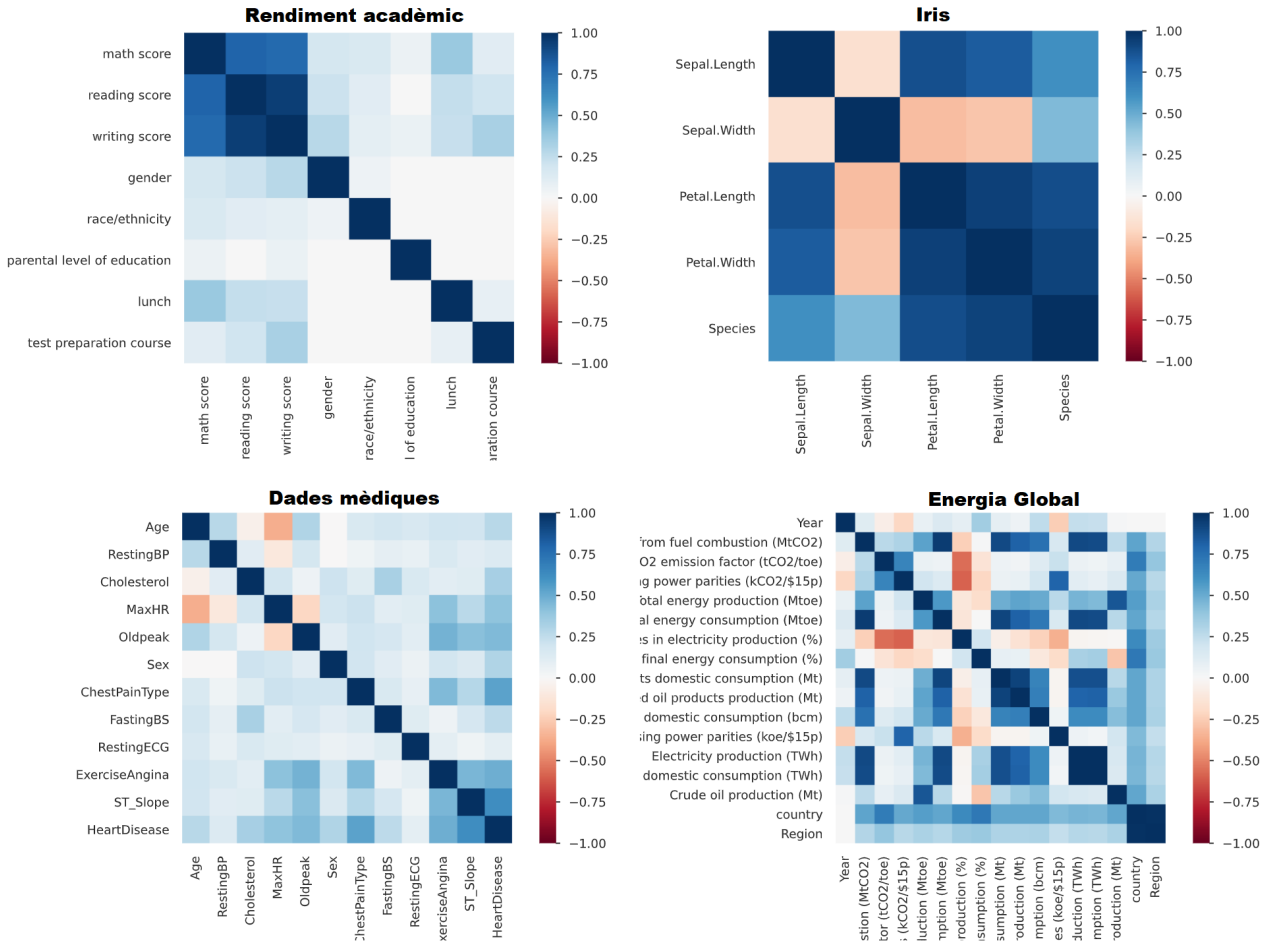

Fig. 12: Heatmap Iris dataset

A.2 Comparatives datasets generats

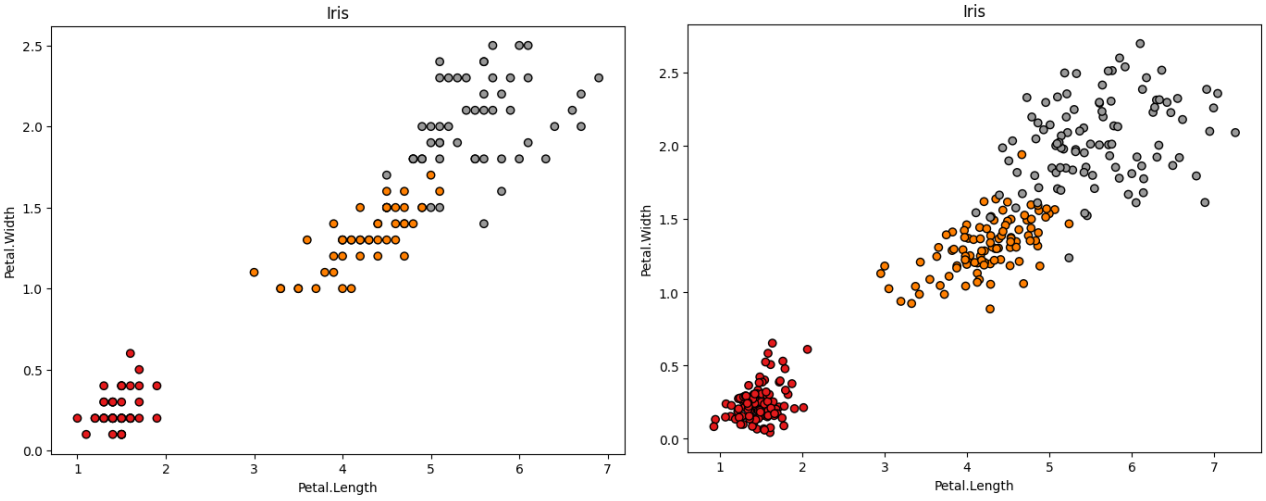

Fig. 13: Comparativa entre dataset Iris original (esquerra) i dataset sintétic (dreta). Columna objectiu: ”Species”

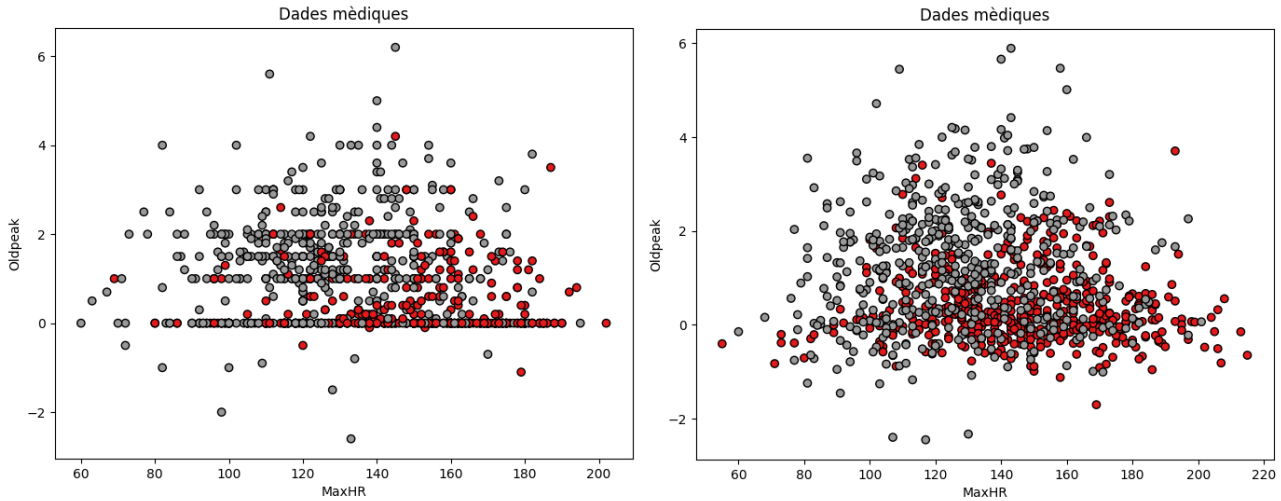

Fig. 14: Comparativa entre dataset dades mèdiques original (esquerra) i dataset sintétic (dreta). Columna objectiu: ”HeartDisease”

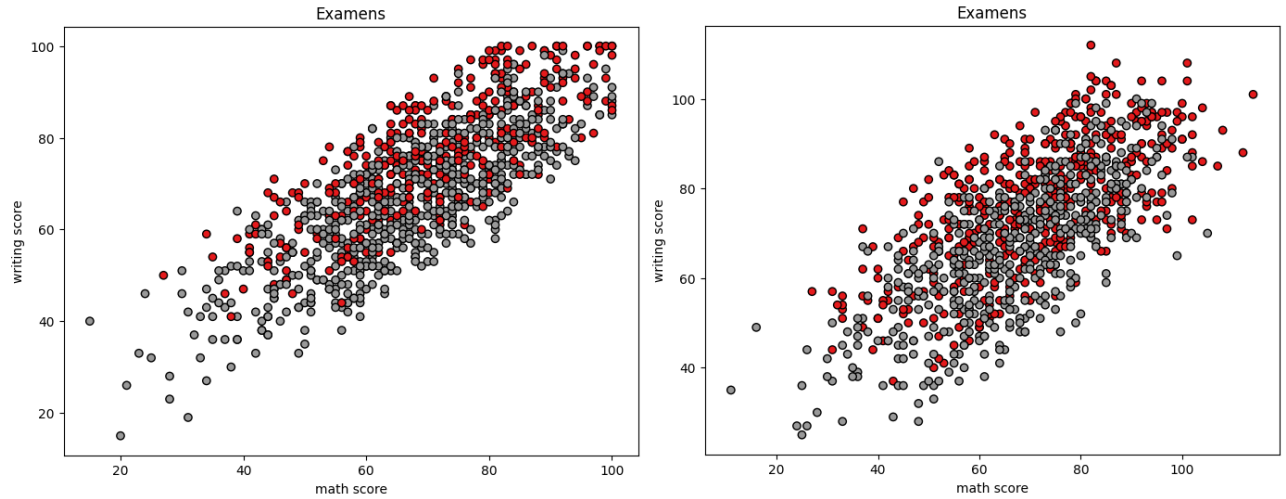

Fig. 15: Comparativa entre dataset rendiment estudiantil original (esquerra) i dataset sintétic (dreta). Columna objectiu: ”test preparation course”