UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques
i Informàtica

GRAU DE MATEMÀTIQUES

TREBALL FINAL DE GRAU

INTEL·LIGÈNCIA ARTIFICIAL EXPLICABLE APLICADA A LA VALORACIÓ DE CRÈDIT

Autor: Joan Orteu Saiz

Director: **Dr. Josep Vives**
Realitzat a: **Departament de Matemàtica Econòmica, Financera i Actuarial**

Barcelona, **25 de febrer de 2024**

Índex

Abstract

Machine learning holds immense potential to revolutionize our existing models, but its widespread adoption faces a significant hurdle, the elusive nature of explanations provided by computers. This thesis strives to bridge this crucial gap.

My thesis builds upon the master's thesis by Bornvalue Chitambira titled *Credit Scoring using Machine Learning Approaches*, conducted at *Mälarden University*. Chitambira developed Artificial Intelligence (AI) models for credit scoring. My goal is to apply Explainable Artificial Intelligence (*XAI*) techniques to enhance the transparency of these models and make them more understandable. The research will encompass a comprehensive review of *XAI* literature, an introduction to Chitambira's work, and an in-depth study of *SHAP* and *LIME* methods. Finally, I will apply these techniques to Chitambira's models to provide a clearer understanding of their functionality.

Resum

L'aprenentatge automàtic té un immens potencial per revolucionar els nostres models existents, però la seva adopció generalitzada es troba amb un obstacle significatiu: la naturalesa esquiva de les explicacions proporcionades pels ordinadors. Aquest estudi es proposa superar aquesta bretxa crucial.

La meva tesi es basa en el treball de final de màster de Bornvalue Chitambira, titulat *Credit Scoring using Machine Learning Approaches*, realitzat a la *Universitat de Mälarden*. Chitambira va desenvolupar models d'Intel·ligència Artificial (IA) per a la puntuació de crèdit. El meu objectiu és aplicar tècniques d'explicabilitat (*XAI*) per millorar la transparència d'aquests models i fer-los més comprensibles. La recerca inclourà una revisió exhaustiva de la literatura sobre *XAI*, una introducció al treball de Chitambira, i un estudi aprofundit dels mètodes *SHAP* i *LIME*. Finalment, aplicarem aquestes tècniques als models de Chitambira per proporcionar una comprensió més clara del seu funcionament.

2020 Mathematics Subject Classification. 68T01, 68T99, 91A12

Capítol 1

Introducció

1.1 Recapitulació del Treball Previ

Aquest projecte de *Bornvalue Chitambira* explora enfocaments d'aprenentatge automàtic que s'utilitzen en l'avaluació de crèdit, o sigui, en l'evaluació de la probabilitat d'impagament (Probability of Default, PD). En aquest estudi, es considera l'avaluació de crèdit al consumidor en lloc de l'avaluació de crèdit empresarial i l'enfocament se centra en mètodes que s'utilitzen actualment en la pràctica per part de bancs, com ara la regressió logística (LR) i els arbres de decisió (DT), i també es compara el seu rendiment amb enfocaments d'aprenentatge automàtic com les *Suport Vector Mahines* (SVM), les *Artificial Neural Networks* (ANN) i els *Random Forest* (RF). Als models, s'aborda qüestions importants com desequilibri del conjunt de dades, sobreajustament del model i calibració de les probabilitats del model. Els sis mètodes d'aprenentatge automàtic que s'estudien són els SVM, LR, ANN, DT, RF i KNN (K-Nearest Neighbors). S'implementa aquests models en Python i s'analitza el seu rendiment en un conjunt de dades de crèdit amb 30,000 observacions de Taiwan, extret del repositori d'Aprenentatge Automàtic de la Universitat de Califòrnia Irvine (UCI).

1.1.1 Base de dades

El conjunt de dades utilitzat en aquest estudi es va obtenir del repositori d'Aprenentatge Automàtic de la UCI: `https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients`. Les dades es van recopilar inicialment per a la investigació centrada en els impagaments dels clients a Taiwan a l'octubre del 2005. Dins de les 30,000 observacions totals, 6636 observacions (22.12%) corresponien als titulars de targetes amb

impagament. Aquesta investigació va utilitzar una variable binària – *default payment* (Sí = 1, No = 0), com a variable de resposta i va fer servir 23 variables com a variables explicatives. A la taula de l'Annex (3) es proporciona un resum dels atributs, el tipus de variable i les categories corresponents per a les variables categòriques. Les característiques es divideixen en variables numèriques i categòriques que es codifiquen amb etiquetes, on una categoria es substitueix per un valor numèric. El conjunt de dades no conté cap valor que "missing".

1.1.2 Revisió de codi

S'ha fet una revisió exhaustiva del codi, s'han eliminat redundàcies per a fer-lo més eficient i també s'han afegit algunes funcions com la de càrrega i guardar models ja entrenats per fer el procés d'explicació més còmode. La rèplica dels model s'ha fet amb èxit, ja que els resultats de validació de model, proposats pel treball de recerca, han sigut idèntics als previs. Els codis es poden trobar tots al repositori de Github d'aquest treball [19].

Com que el conjunt de dades utilitzat per al problema era desequilibrat, s'hi aplica aprenentatge sensible al cost als algoritmes perquè penalitzessin més les classificacions errònies de la classe minoritària. També s'hi considera tècniques útils com la validació creuada (Cross Validation) per abordar el sobreajustament del model (Overfitting).

S'ajusten els hiperparàmetres del model per abordar les qüestions explicades anteriorment. Per exemple, s'implementa una recerca en graella (Grid Search) per pre-prunar el RF i el DT per evitar el sobreajustament. Pel mètode SVM, es triar un valor moderat pel paràmetre de penalització C per permetre errors moderats en l'algoritme. Com a resultat d'aquests ajustos, els resultats poden semblar parcialment influïts pels objectius. Hi ha, per tant, la possibilitat que s'obtinguin resultats diferents si es consideren hiperparàmetres diferents.

També es calibren els models ja que les probabilitats no calibrades solen ser massa segures o massa poc segures en les seves puntuacions.Les probabilitats calibrades utilitzades, reflecteixen millor la probabilitat d'esdeveniments reals. També s'aplica SMOTE (Synthetic Minority Over-sampling Technique), que és una tècnica de sobre-mostreig utilitzada en l'àmbit de classificació i aprenentatge automàtic per abordar el desequilibri de classes, a l'algorisme dels KNN per abordar el desequilibri del conjunt de dades.

1.1.3 Resultats

En aquesta tesi s'hi examinen les tècniques d'aprenentatge automàtic utilitzades per a la classificació de sol·licitants de crèdit i s'hi estudia els fonaments matemàtics que sustenten aquests mètodes.

Els resultats de les mesures de validació dels models (1 i 2 de l'Annex) donen informació clara. Ja que el nostre objectiu és tenir un model que minimitzi les classificacions errònies dels préstecs dolents com a bons, la "Precision" dóna una indicació més precisa del rendiment del model en aquest sentit. La xarxa neuronal artificial té la precisió més alta, amb un valor de 0,884, per tant estudiarem el model de ANN amb més cura.

1.2 Problema plantejat

Els models complexos d'Aprenentatge Automàtic presenten desafiaments en el sentit que és difícil explicar per què un model produeix determinats resultat, exiteix una desafortunada relació entre precisió i explicabilitat dels mètodes. Explicar els models de forma global, per entendre com està prenent decisions en general (per saber si és un "bon model") i explicacions de prediccions concretes.

1.2.1 Objectius

L'objectiu d'aquest treball és utilitzar els mètodes SHAP i LIME per entendre els models entrenats. En particular, investigarem el model ANN degut a que té una millor Precisió. Un altre objectiu, ha de ser obtenir una visió global del camp de l'Explicabilitat de Models d'Aprenentatge Automàtic (XAI) i aprofundir en con funcionen els mètodes SHAP i LIME escollits.

Capítol 2

Intel·ligència Artificial Explicable

Aquest apartat proporciona una visió introductòria al camp de l'Explicació de Models d'Aprenentatge Automàtic, basant-se en les referències dels llibres "Interpretable Machine Learning" de C. Molnar [2] i "Explainable Artificial Intelligence: An Introduction to Interpretable Machine Learning" de Uday Kamath i John Liu [1]. Donat que aquest treball es limita a l'estudi específic de SHAP i LIME per raons de temps, es pot observar que aquesta és només una fracció dels diversos mètodes disponibles en aquest àmbit.

Es recomana que, en cas de voler explorar una visió més àmplia i detallada sobre els diferents tipus de mètodes d'explicació de models d'aprenentatge automàtic, es busqui una comprensió exhaustiva a través de la lectura dels llibres referenciats [2] i [1]. Aquests llibres ofereixen una perspectiva completa i enriquidora que pot ampliar la comprensió del lector sobre aquest tema complex i fascinant.

2.1 Introducció

La innovació en els algorismes d'aprenentatge automàtic ha portat a grans avanços en la capacitat predictiva i de precisió. No obstant això, aquests algorismes s'han tornat cada vegada més complexos. Aquest és un intercanvi desafortunat entre una millora en la qualitat i la transparència.

A diferència dels models matemàtics que tenen una estructura inherent, els models d'aprenentatge automàtic poden aprendre la relació entre les entrades i les sortides directament de les dades. Per a alguns models, com els arbres de decisió, aquesta relació és fàcilment discernible. Per a

altres, com els boscos d'arbres aleatoris o els models d'aprenentatge profund, esdevé gairebé impossible comprendre com es fan les prediccions.

Molts models d'aprenentatge automàtic i d'aprenentatge profund són fonamentalment "Caixes Negres", que no revelen els mecanismes interns ni les subtilitats de les seves prediccions. Aquesta manca de transparència i comprensió pot tenir conseqüències serioses per a la nostra confiança i adopció d'aquests models.

Per exemple, com sabem si les prediccions del model poden ser incorrectes? Això és especialment important en àmbits de gran importància com la salut. Un metge o pacient confiaria en una predicció de càncer si un model entrenat té una precisió del 99 per cent? I si, sense que nosaltres ho sabem, el model passa per alt els casos més malignes? I si la alta precisió es deu a una filtració de dades en les dades de prova, de manera que el rendiment fora de mostra sigui molt pitjor? Aquesta és la raó per la qual la intel·ligència artificial explicativa (XAI) és vital per a la nostra adopció de l'aprenentatge automàtic. Per a decisions de gran transcendència com préstecs de crèdit, la discriminació en les sol·licituds de llibertat condicional, el diagnòstic mèdic, etc., esdevé imperatiu que els models d'aprenentatge automàtic siguin explicables.

2.2 Taxonomia

Explicacions

Primer de tot, hem de respondre la pregunta, que és una explicació? Bé doncs, segons Miller (2017) una explicació es una resposta a una pregunta del tipus "per què":

- Per què el model ha predit que el pacient tindrà cancer?
- Per què la meva hipoteca ha estat rebutjada?
- Per què no hem conseguit Inteligència Artificial General encara?

Les dues primeres preguntes es poden respondre amb una explicació "quotidiana", mentre que la tercera prové de la categoria de "fenòmens científics més generals i qüestions filosòfiques". Ens centrem en les explicacions del tipus "quotidià", ja que aquestes són rellevants per a l'aprenentatge automàtic interpretable.

XAI

No exeisteix cap algoritme que sigui adequat per tot tipus de dades i problemes, de la mateixa manera, no existeix un mètode d'interpretació que sigui el més òptim per tots els casos. Per tant, podem dividir les explicacions segons 5 grups: Dades, Explicació, Model, Etapa i Abast.

- **Etapa:** Podem classificar les explicacions en tres grups segons la etapa en la que les apliquem:

  "Pre-model": és un conjunt de tècninques que tenen l'objectiu d'adquirir una idea de les dades per ajudar a construir models més efectius i per a poder interpretar els resultats millor, moltes d'aquestes tècniques que resumeixen, visualitzen i exploren les dades han existit durant molt de temps. Exemples de tècniques exploratories inclou visualitzar distribucions dels diferents atributs i analitzar el tipus d'atributs entre altres.

  "Intrínsec": és el conjunt de models interpretables com Regressió Lineal, Regressió Logistica, Arbres de Decisió. Es pot argumentar, que no existeixen els models Intrínsecament Interpretables, ja que per exemple, si tenim 1000 inputs que contribueixen de manera significativa a la predicció, per molt que sigui un model de Regressió Lineal, no podrà ser comprès i per tant s'haurien de crear variables compostes per poder comprendre el funcionament.

  "Post-Hoc": és el conjunt de mètodes creats epecíficament per entendre els models de Caixa Negra, en els quals no tenim accés a les característiques internes del mode.

- **Model:** Podem classificar les explicacions en dos grups segons el model al qual es pot aplicar:

  "Específic": les explicacions de tipus específic només es poden aplicar a un tipus de model i dades concret.

  "Agnòstic": les explicacions de tipus agnòstic no estàn restringides a un tipus de model i dades concret.

- **Abast:** Podem classificar les explicacions en dos grups segons l'abast al qual s'enfoca:

  "Local": és el tipus d'explicacions que es centra en explicar com funciona el model o com ha pres una decisió en una regió local al voltant d'una instància en particular, no és una explicació representativa de com funciona el model al llarg de tota la mostra.

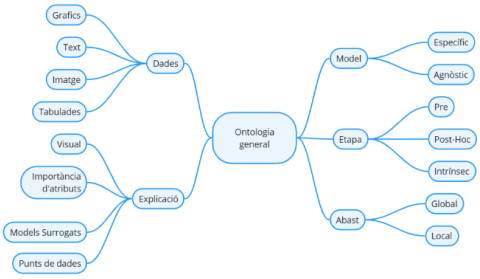


Figura 2.1: Esquema de la ontologia general de XAI.

"Global": és el tipus d'explicació que intenta explicar el comportament del model al llarg de tota la mostra, dona una visió general.

- **Dades:** Hi ha mètodes que poden servir per tot tipus de dades, però hi ha alres que només es poden aplicar o funcionen millor per algún en concret. Els tipus de dades pels quals es pot fer una classificació de les explicacions són: Gràfics, Text, Imatges i Taules.

- **Explicació:** Podem classificar les explicacions en quatre grups segons el tipus d'explicació que ens dona:

  "Visual:" Les tècniques de visualització de dades pot ser utilitzada per entende les prediccions que s'han fet al llarg de les dades d'entrada. Un exemple de tècnica visual és Partial Dependency Plots (PDP), que visualitzen la funció de dependència parcial, la qual mesura l'efecte d'una característica marginalitzant les altres característiques. També són tècniques de visualització les clássiques tècniques de validació de models, com per exemple: les corbes de precisió-sensibilitat (precision-recall), les corbes ROC, regressions i tècniques de "Clustering".

  "Importància d'Atributs:" Els mètodes d'importància busquen quantificar la contribució dels atributs en la predicció del model. Ho fan considerant factors addicionals com el tipus, la robustesa, la comprensibilitat o la qualitat de les explicacions. Les explicacions poden explicar el comportament global d'un model o bé poden ser locals i explicar la predicció d'instàncies individuals. Hi ha molts mètodes, com la interacció entre atributs, la importància de característiques per permutació, SHAP, LIME, entre altres.

  "Models Surrogats:" Podem explicar els nostres models complexos utilitzant models simplificats que aproximin les prediccions de la nostra Caixa Negra.

	SHAP	LIME
Etapa	Post-Hoc	Post-Hoc
Model	Agnòstic	Agnòstic
Abast	Local i Global	Local i Global
Dades	Totes	Totes
Explicació	Imp d'Atributs	Imp d'Atributs i Model Surrogat

Taula 2.1: Classificació de LIME i SHAP

"Punts de Dades:" Aquesta categoria inclou tots els mètodes que retornen intàncies d'entrada amb la seva predicció per entendre millor el model.

La taula 2.1 conté la classificació dels mètodes SHAP i LIME dintre de la taxonomia proposada de XAI. En els capítols següents explicarem amb profunditat aquests dos mètodes.

Capítol 3

SHAP

SHAP és un Model-Agnòstic basat en Valor de Shapley, de la teoria de Jocs Cooperatius, creat per explicar predicions fetes per models d'IA. SHAP genera un valor per cada atribut (valor de SHAP) i aquest valor indica com l'atribut ha contribuit a la predicció del punt de dades especificat. Alguns atributs contribuiran positivament i altres negativament. Permetent una millor comprensió de com cada atribut contribueix a la predicció d'un model de manera conjunta.

SHAP (SHapley Additive exPlanations) reformula el clàssic problema dels valors de Shapley per distribuir de manera *justa* la importància de cada atribut en un Model d'Aprenentatge Automàtic.

En aquest capítol, emprendrem una visió detallada de SHAP, explorant cada aspecte des de la teoria del Valor de Shapley fins a la seva adaptació específica en SHAP. Analitzarem la seva implementació, incloent aproximacions i mètodes concrets. A més, explorarem els avantatges i inconvenients d'utilitzar SHAP per a la interpretació de models. Finalment, proporcionarem una guia pràctica sobre com implementar SHAP utilitzant el llenguatge de programació Python. Aquesta exploració exhaustiva permetrà als lectors obtenir una comprensió completa dels fonaments teòrics i pràctics de SHAP en el context de la interpretació de models amb dades tabulades.

Aquest capítol ha estat inspirat principalment per C. Molnar [2] i [3], l'article de Lundberg & S.-I. Lee [6], complementat amb "Machine Learning TV"[10], el TFG de Ignasi Verneda [11] i altres que ja aniré citant.

3.1 Teoria

3.1.1 Història de Valor de Shapley i SHAP

El Valor de Shapley pren el seu nom del seu creador, Lloyd Shapley, el qual el va introduir per primera vegada a l'article *"A Value for n-Person Games"*[4] l'any **1953**. Durant la dècada de 1950, la teoria dels jocs va experimentar un període altament actiu, durant el qual es van formular molts conceptes fonamentals, incloent-hi jocs repetits, el dilema del presoner, el joc fictici i, evidentment, el Valor de Shapley. (Al 2012 Lloyd Shapley i Alvin Roth van ser guardonats amb el Premi Nobel d'Economia pel seu treball en *Disseny de Mercat* i en *Teoria d'Aparellament*)

Aquests valors serveixen com a solució en la teoria de jocs cooperatius, que tracta els jocs en els quals els jugadors col·laboren per aconseguir una recompensa conjunta. Aborden la qüestió d'un grup de jugadors que participen en un joc col·laboratiu, quan treballen junts per assolir una determinada recompensa. La recompensa del joc ha de ser distribuïda entre els jugadors, els quals poden haver contribuït de manera diferent. El Valor de Shapley proporciona un mètode matemàtic per a repartir justament la recompensa entre els jugadors.

El Valor de Shapley s'ha consolidat com a peça clau en la teoria de Jocs Cooperatius, amb aplicacions esteses en diversos camps com la ciència política, l'economia i la informàtica, però no s'aplicava a l'aprenentatge automàtic (ML) degut a que era un camp embrionari. Finalment, a la dècada de 2010, el camp de XAI va començar a rebre atenció. L'any **2010**, es va donar el primer pas cap a l'aplicació del Valor de Shapley en ML per Erik Štrumbelj i Igor Kononenko amb l'article *"An efficient explanation of individual classifications using game theory"* [5]. Però no va tenir èxit degut a que l'article no contenia codi, el camp de XAI no era gaire reconegut encara i els mètodes d'estimació eren relativament lents i no podien ser utilitzats amb imatges ni classificadors de text.

Finalment a l'any 2016, Ribeiro et al. van publicar un article introduint *Local Interpretable Model-Agnostic Explications* (LIME)[14], un mètode que utilitza models de regresió lineal, per explicar prediccions de forma local. Aquest article va ser el catalitzador que necessitava el camp de XAI. A l'any **2017** Scott Lundberg i Su-In Lee van publicar l'article *"A unified Approach to Interpreting Model Predictions"* [6] publicat al NeurIPS, on s'introduia *Shapley Additive exPlanations* (SHAP), un altre mètode per explicar predictors de Models d'Aprenentatge Automàtic. En aquest, presentaven una nova forma d'estimar valors SHAP utilitzant regresió lineal ponde-

rada amb una funció kernel per pesar els punts de dades (Actualment, el paquet SHAP ja no utilitza Kernel SHAP per defecte, per tant dona a aquest article un caire en certa manera històric). Aquest cop, si va tenir èxit i es creu que els factors principals d'adopció va ser el paquet open-source SHAP a Python amb una àmplia gamma de característiques i capacitats de representació gràfica; la recerca continuada pels autors originals i altres col·laboradors ha contribuït al seu desenvolupament i finalment que va ser una obra pionera en un camp de ràpid creixement.

Des de la seva creació, SHAP ha guanyat molta popularitat i s'han anat proposant noves i més eficients aproximacions per calcular els valors SHAP segons el tipus d'algorisme. Per exemple, l'any 2020 Lundberg et al. van proposar un mètode de càlcul SHAP específic per models basats en arbres [8]. Un altre millora significativa ha estat poder utilitzar molts valors SHAP per fer generar interpretacions globals del model.

Vist el context històric, continuarem amb la teoria de Valor de Shapley.

3.1.2 Jocs Cooperatius

En un marc general, la teoria de jocs es divideix en teoria de jocs cooperatius i teoria de jocs no cooperatius. La diferència més notable entre les dues branques és la possibilitat d'establir acords vinculants o no poder-los establir respectivament. Però encara podem fer una classicació més acurada de la teoria de jocs cooperatius en jocs amb utilitat transferible (TU) i jocs sense utilitat transferible (NTU).

En els jocs cooperatius, els jugadors poden formar coalicions o col·laborar d'alguna manera per aconseguir un objectiu comú, en lloc de competir entre si de manera pura. La base dels jocs cooperatius és suposar que els diferents agents es comporten d'un mode en el que busquen també el benefici col·lectiu.

En aquesta branca, el que s'estudia és el mode en com es reparteixen els beneficis de la cooperació. Per tant, en aquesta classe de jocs l'objecte d'estudi passa a ser la coalició. L'èmfasi recau en la negociació, la comunicació i la presa de decisions conjuntes per part dels jugadors. Diferent dels jocs no cooperatius, on cada jugador busca maximitzar el seu benefici personal, en els jocs cooperatius els jugadors busquen solucions que generin beneficis per a tot el grup.

Definició 3.1. *El conjunt total de jugadors, denotat per* $N := \{1, ..., n\}$*, és un conjunt on els seus elements són els agents prenedors de decisions d'un joc en el que tots estan inclosos. Utilitzarem indistingidament conjunt total de jugadors i gran coalició.*

La coalició és un concepte clau en els jocs cooperatius, que és un conjunt de jugadors que s'uneixen per cooperar en el joc. Les coalicions poden ser de diverses formes i mides, i poden canviar al llarg del joc. Una coalició és un subconjunt de jugadors que decideixen col·laborar entre ells. La teoria dels jocs cooperatius se centra en estudiar quines coalicions són estables i com es poden repartir els guanys generats per aquestes coalicions entre els seus membres.

Definició 3.2. *Per a cada subconjnt $S \subset N$, ens referim a S com a coalició. En aquest cas, $|S|$ representa el cardinal de S. És a dir, la quantitat de jugadors involucrats en la coalició S. La coalició N, el grup total, s'anomena la gran coalició.*

La teoria de jocs cooperatius és la branca de la teoria de jocs, que tracta els jocs en els que els jugadors poden formar coalicions, cooperar i arribar a acords. Hi ha diferents característiques que comparteixen els jocs cooperatius i les quatre més fonamentals són les següents:

1. **Interés comú**: Els jugadors comparteixen un interès comú a l'hora d'assolir un objectiu o resultat específic. Han de cooperar i trobar punts comuns per establir una cooperació per assolir aquest interés.

2. **Interacció necessària entre jugadors**: Els jugadors són independents i col·laboren entre ells. Aquest concepte remarca que les decisions i les accions d'uns, afecten significativament els altres jugadors.

3. **Acord obligat**: Un cop s'arriba a un acord, els jugadors tenen la obligació de complir-lo.

4. **Benefici mutu**: Els jugadors voluntariament formen coalicions, on tots els participants de la coalició obtenen part del benefici comú.

Els jocs cooperatius es poden dividir en Jocs d'Utilitat no Transferible (Jocs-NTU) i els d'Utilitat Transferible (Jocs-TU). Intuitivament, l'objectiu dels Jocs-NTU és analitzar situacions on cada jugador té preferències úniques i no es pot transferir la seva utilitat a altres jugadors i els Jocs-TU són jocs que tenen com a objectiu estudiar situacions on la utilitat d'un jugador pot ser transferida a altres mitjançant acords o negociacions.

Un altre forma de veure-ho és que mentre que els Jocs-NTU es centren en les preferències individuals i les relacions directes, els Jocs-TU es centren en la capacitat de transferir utilitat entre jugadors mitjançant acords, negociacions i coalicions. El Valor de Shapley s'estudia en Jocs-TU i per tant aquest caítol es centra en ells. Anem a formalitzar-los.

Jocs d'Utilitat no Transferible (Jocs-NTU)

Definició 3.3. *Donada una coalició $S \subset N$ i un conjunt $A \subset \mathbb{R}^{|S|}$, diem que A és raonable si, per a cada parell $x, y \in \mathbb{R}^{|S|}$ tals que $x \in A$ i $y \leq x$, tenim que $y \in A$. A més, sigui $B \subset \mathbb{R}^{|S|}$ el conjunt raonable més petit que conté A, B és el conjunt completament raonable d'A.*

Definició 3.4. *Un joc d'utilitat no-transferible $(joc - NTU)$ és un parell (N, V), en el que N és el conjunt de jugadors i V és la funció que assigna, a cada coalició $S \subset N$, un conjunt $V(S) \subset \mathbb{R}^{|S|}$. Per convenció, la imatge del buit és el conjunt $\{0\}$. Per a cada coalició $S \subset N$, tenim que si $S \neq \emptyset$:*

1. *$V(S)$ és un conjunt diferent al buit i un subconjunt tancat de $\mathbb{R}^{|S|}$*
2. *$V(S)$ és raonable. Encara més, per a cada $i \in N$, tenim que $V(\{i\}) \neq \mathbb{R}$. Dit d'altra manera, hi ha un $x_i \in \mathbb{R}$ tal que $V(\{i\}) = (-\infty, x_i]$.*
3. *El conjunt $V(S) \cap \{y \in \mathbb{R}^{|S|} : \text{per a cada } i \in S, y_i \geq x_i\}$ és fitat.*

Per tant, un joc-NTU és un cas particular en el que tenim que per a cada $S \subset N$ i cada $x \in V(S)$, hi ha un resultat $r \in \mathbb{R}^{|S|}$ tal que, per a cada $i \in S$, tenim que $x_i = U_i^{|S|}(r)$. En termes més entenedors, $\mathbb{R}^{|S|}$ representa el conjunt de tots els possibles pagaments de cada jugador en la coalició S.

Definició 3.5. *Sigui (N, V) un joc-NTU. Aleshores als vectors en $\mathbb{R}^n$ se'ls diu assignacions o pagaments. Una assignació $x \in \mathbb{R}^n$ és factible si hi ha una partició $\{S_1, ..., S_k\}$ de N que satisfà que per a cada $l \in \{1, ..., k\}$ hi ha un $y \in V(S_l)$ tal que, per a cada $i \in S_l$ tenim que $y_i = x_i$.*

Example 3.6. Un exemple clàssic de joc d'utilitat no transferible és el "Joc del Matrimoni", també conegut com el "Problema del Matrimoni Estable". Aquest joc és utilitzat per modelar i analitzar el procés de fer parelles en situacions on les preferències són úniques per a cada participant.

Suposem que hi ha dos conjunts d'individus, homes i dones, i cada home i cada dona té una llista ordenada de preferències sobre els membres de l'altre conjunt. L'objectiu és crear parelles d'acord amb aquestes preferències de manera que no hi hagi cap parella que prefereixi mutuament una a l'altra. Es a dir, l'objectiu seria trobar una assignació de parelles que maximitzi la satisfacció global, tenint en compte aquestes preferències individuals.

En aquest joc, la utilitat de cada jugador (home o dona) està directament relacionada amb la seva parella assignada, i aquesta utilitat no es pot transferir a altres jugadors.

Jocs d'Utilitat Transferible (Jocs-TU)

Definició 3.7. *Un joc d'utilitat transferible és un parell (N, v), on N és el conjunt de jugadors i la funció característica del joc és $v : 2^N \to \mathbb{R}$ (on 2^N és el conjunt de parts de N). Per convenció, notem que la imatge del buit és zero ($v(\emptyset) := 0$).*

Definició 3.8. *Se li diu G^N a la classe dels jocs cooperatius amb n jugadors.*

Definició 3.9. *Sigui $(N, v) \in G^N$.*

1. *Un jugador $i \in N$ es diu **jugador nul** si, per a qualsevol $S \subset N$, tenim que $v(S \cup \{i\}) - v(S) = 0$.*
2. ***Dos jugadors** i, j **són simètrics** si, per a cada coalició $S \subset N \setminus \{i, j\}$, tenim que $v(S \cup \{i\}) = v(S \cup \{j\})$*

A partir d'ara, farem un abús de notació, quan volguen referir-nos a un Joc-TU escriurem $v \in G^N$, en lloc de $(N, v) \in G^N$ per simplificar notació.

Example 3.10. Cap a l'any 1140 a.C., el rabí Ibn Ezra proposava el problema següenten al Talmud: Jacob mor i cadascun dels seus quatre fills, Ruben, Simeó, Leví i Judà, presenta un escrit en el qual Jacob el reconeix com a hereu i li deixa, respectivament, un quart, un terç, la meitat i la totalitat dels seus béns. Tots els escrits porten la mateixa data i, per tant, cap no té prioritat sobre els altres. Com repartir l'herencia entre els quatre fills? (Exemple extret de [13])

3.1.3 Valor de Shapley

Aquest concepte proporciona una manera de quantificar la contribució de cada jugador dins d'una col·laboració, amb aplicacions que transcendeixen els jocs. En el context de la teoria de la imputació econòmica, el Valor de Shapley ofereix una distribució equitativa del benefici generat per la cooperació entre agents econòmics.

Mitjançant la idea de la permutació de jugadors, el Valor de Shapley captura la contribució única de cada individu a través de totes les possibles coalicions. Aquest enfocament proporciona una base matemàtica sòlida per distribuir els beneficis de manera "justa"en situacions cooperatives, contribuint a la equitat i la transparència en les decisions conjuntes.

Lloyd Shapley va proposar uns axiomes que definien el com ha de ser un repartiment "just", i d'aquests es deriva la fórmula per calcular els Valors de Shapley. Aquests axiomes són els següents: Eficiència, Jugador nul, Simetria i Additivitat.

Anem a veure la construcció del cos matemàtic per a arribar al Valors de Shapley.

Terme	Terme matemàtic
Jugador	1, ... ,n
Gran Coalició	N
Coalició	S
Mida de la Coalició	$\|S\|$
Utilitat del joc	$v()$
Valor de Shapley	ϕ_i

Taula 3.1: Conté termes importants per a seguir aquesta secció amb més facilitat. Inspirada en capítol "4.4. Calculating Shapley values"de [3].

Definició 3.11. *Se li diu **valor del joc** $v \in G^N$ a una funció ϕ de G^N en $\mathbb{R}^n$ en que $\phi(v)$ és un vector que representa en cada coordenada $\phi_i(v)$ el pagament o assignació que percep el jugador $i \in N$. En altres paraules, cada $\phi_i(v)$ és el Valor de Shapley que correspon al jugador i del joc v.*

Tot seguit presentem el Valor de Shapley tal com el va introduir Ll. Shapley l'any 1953.

Definició 3.12. *El Valor de Shapley, Φ, es defineix per a cada $v \in G^N$ i cada $i \in N$ com:*

$$\Phi_i(v) := \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n-|S|-1)!}{n!}(v(S \cup \{i\}) - v(S)). \tag{3.1}$$

Tot seguit definirem un conjunt de propietats que han de complir els valors d'un joc.

1. **Eficiència:** El valor del joc ϕ satisfà la propietat d'eficiència si, per a qualsevol $v \in G^N$, tenim que $\sum_{i \in N} \phi_i(v) = v(N)$.

2. **Jugador nul:** El valor del joc ϕ satisfà la propietat del jugador nul si, per a cada $v \in G^N$ i cada jugador nul $i \in N$, tenim que el seu valor és zero, $\phi_i(v) = 0$.

3. **Simetria:** El valor del joc ϕ satisfà la propietat de la simetria si per a cada $v \in G^N$ i cada parell $i, j \in N$ de jugadors simètrics, tenim que els seus valors són iguals, $\phi_i(v) = \phi_j(v)$.

4. **Additivitat:** El valor del joc ϕ satisfà la propietat d'additivitat si, per a cada parell de jocs $v, w \in G^N$, tenim que els valors $\phi(v+w) = \phi(v) + \phi(w)$.

Aleshores, en esperit, el Valor de Shapley assigna a cada jugador les contribucions que fa a les diferents coalicions. Cada jugador obté la mitjana ponderada de les seves contribucions, és a dir la mitjana de les contribucions marginals.

El Valor de Shapley es calcula pressuposant que s'acabarà formant la gran coalició, però els jugadors s'hi van afegint seqüencialment en tots els ordres possibles (calculant un per un).

Quan el jugador i entra, obté la seva contribució a la coalició dels jugadors que ja són a l'interior (és a dir, si aquesta coalició és S, obté $v(S \cup \{i\})$ - $v(S)$). L'ordre dels jugadors es decideix de manera aleatòria, amb totes les $n!$ possibles ordenacions sent igualment probables.

Amb aquest raonament, es suggereix una definició alternativa del Valor de Shapley, tot basant-se en els vectors de contribucions marginals. Sigui $\Pi(N)$ el conjunt de totes les possibles permutacions dels elements en el conjunt N i per a cada permutació $\pi \in \Pi(N)$, denotarem amb $P^\pi(i)$ el conjunt dels predecessors de i seguint l'ordre donat per la permutació π. És a dir, $j \in P^\pi(i)$ si i només si $\pi(j) < \pi(i)$.

Definició 3.13. *Sigui $v \in G^N$ un joc-TU. Sigui $\pi \in \Pi(N)$. Aleshores el vector de contribucions marginals associat amb π, que denotem per $m^\pi(v) \in \mathbb{R}^N$, ve definida per a cada $i \in N$ per $m_i^\pi(v) := v(P^\pi(i) \cup \{i\})$ - $v(P^\pi(i))$.*

Amb aquesta definició, doncs, podem simplificar l'escriptura del Valor de Shapley i es pot posar com:

$$\Phi_i(v) := \frac{1}{n!} \sum_{\pi \in \Pi(N)} m_i^\pi(v). \tag{3.2}$$

Tot seguit, presentem un tipus de joc d'utilitat transferible que ens pot ser útil per als nostres estudis. Es tracta dels jocs d'unanimitat de la coalició.

Definició 3.14. *Dins la classe G^N, donada una coalició $S \subset N$, tenim que el joc d'unanimitat de la coalició S, que denotarem per w^S, es defineix de la següent manera. Per a cada $T \subset N$, tenim que $v(T) := 1$ si $S \subset T$ i en canvi, en altre cas ens queda $v(T) := 0$.*

Similarment al teorema vist anteriorment, ara estudiarem un que s'adequa a la nova definició del Valor de Shapley.

Teorema 3.15. *El Valor de Shapley és l'únic valor de joc en G^N que satisfà les propietats d'eficiència, jugador nul, simetria i additivitat simultaniament.*

Demostració. Clarament la propietat de jugador nul es compleix, ja que si un jugador és nul, tindrem que la diferència en la utilitat abans i després de la seva aparició en la coalició és nul·la. Per tant el seu vector de contribucions marginals és nul i aleshores el seu sumatori equival a zero, per tant, el seu Valor de Shapley és també zero. La propietat de l'additivitat de dos jocs independents es compleix per definició del valor. Les altres dues propietats surten de la forma reescrita que li hem donat al Valor de Shapley. La d'eficiència surt de que si sumem totes les utilitats marginals, per la definició dels jocs-TU en G^N, tindrem la utilitat de la coalició de tots els jugadors $v(N)$.

Respecte a la simetria, per cada permutació (amb i, j simètrics) podem trobar una altra permutació que intercanviï les posicions dels jugadors i, j. És a dir, si i, j són simètrics, aleshores dins del conjunt de permutacions, per cada permutació on i estigui en la posició $k-$èssima i j en posició l-èssima, podem trobar una exactament igual que només itercanviï els articles d'aquestes dues. Si ho repetim per totes les permutacions, obtenim exactament el mateix conjunt. Aleshores, tenim que el seu valor és el mateix, per tant satisfà aquesta propietat, i per consequència, compleix totes les quatre condicions.

Ara provem la seva unicitat. Suposem que existeix una altra funció ϕ que satisfà totes les quatre propietats. Cada $v \in G^N$ es pot veure com un vector $\{v(S)\}_{S\in 2^N\setminus\{\emptyset\}} \in \mathbb{R}^{2^n-1}$. Aleshores, G^N pot ser considerat com un espai vectorial $2^n - 1$-dimensional. Ara veiem que els jocs d'unanimitat $U(N)$:=$\{w^S : S \in 2^N \setminus \{\emptyset\}\}$ són una base d'aquest espai vectorial.

Veurem ara amb aquest propòsit que $U(N)$ és un conjunt de vectors linealment independents. Siguin $\{\alpha_S\}_{S\in 2^N\setminus\{\emptyset\}} \subset \mathbb{R}$ tals que $\sum_{S\in 2^N\setminus\{\emptyset\}} \alpha_s w^S$ = 0 i suposem que hi ha un $T \in 2^N \setminus \{\emptyset\}$ amb $\alpha_T \neq 0$.Podem assumir aleshores que no hi ha $R \subset T$ tal que $\alpha_R \neq 0$. Aleshores tenim que 0 = $\sum_{S\in 2^N\setminus\{\emptyset\}} \alpha_s w^S(T) = \alpha_T \neq 0$ i arribem a contradicció. Com ϕ satisfà les propietats d'eficiència, jugador nul i simetria, tenim que per a cada jugador $i \in N$, cada $\emptyset \neq S \subset N$ i cada $\alpha_S \in \mathbb{R}$,

$$\phi_i(\alpha_S w^S) = \begin{cases} \frac{\alpha_S}{s} & \text{si } i \in S \\ 0 & \text{si } i \notin S \end{cases}$$

Aleshores si ϕ també satisfà la propietat de l'additivitat, tenim que ϕ està unívocament determinada, ja que $U(N)$ és una base de G^N i per tant, $\phi = \Phi$. □

3.1.4 Creació de SHAP des de Shapley

Terme	Concepte en ML	Terme matemàtic
Jugador	Índex d'atribut	j
Número de Jugadors	Número d'atributs	N
Coalició	Conjunt d'atributs	$S \subseteq \{1, ..., N\}$
No en la Coalició	Atributs no en S	$C : C = \{1, ..., N\} \backslash S$
Mida de la Coalició	Número d'atributs	$\|S\|$
Utilitat de la gran coalició	Predicció per $x^{(i)}$ menys l'esperança de la predicció	$f(x^{(i)}) - \mathbb{E}(f(X))$
Utilitat de la coalició S	Predicció de la coalició S menys l'esperança del joc	$v_{f,x^{(i)}}(S)$
Valor de Shapley	Contribució de l'atribut j al pagament	$\phi_j^{(i)}$

Taula 3.2: Conté els termes de Shapley aplicats relacionats amb conceptes de ML i la seva representació. Inspirada en capítol "5.2. From Shapley Values to SHAP"de [3].

Ja hem vist d'on surt, que és el Valor de Shapley i com aquest és l'únic valor de joc en G^N que satisfà les propietats d'eficiència, jugador nul, simetria i additivitat simultaniament, i per tant l'únic que dona una distribució equitativa. Ara volem interpretar les prediccions d'un model d'IA com un joc de coalicions.

Be doncs, podem pensar en els atributs d'un model com si fóssin els jugadors d'un joc cooperatiu. La utilitat del joc és la predicció, el valor de la gran coalició N per un joc és la diferència entre la predicció i la esperança del model. En la Taula 3.2 hi ha les transformacions dels conceptes claus, de Shapley a SHAP.

Donat un model f i instància $x^{(i)}$, la funció de Valors SHAP és la següent:

$$v_{f,x^{(i)}}(S) = \int f(x_S^{(i)} \cup X_C)\, d\mathbb{P}_{X_C} - \mathbb{E}(f(X)) \tag{3.3}$$

En aquesta equació, hem fet un abús de notació a l'hora de fer la unió del vector d'atributs $x_S^{(i)} \cup X_C \in \mathbb{R}^N$, on els valors a la posició S tenen forma $x_S^{(i)}$ i la resta, són variables aleatories de X_C. A partir d'ara, $v_{f,x^{(i)}}(S)$ serà escrit $v(S)$ per simplificar notació.

Ara anem a construir les contribucions marginals de j a S i després ja

podrem posar-ho tot junt per construir la fórmula de SHAP:

$$\begin{aligned} v(S \cup \{j\}) - v(S) &= \int f(x^{(i)}_{S\cup\{j\}} \cup X_{C\setminus\{j\}})\, d\mathbb{P}_{X_{C\setminus\{j\}}} - \mathbb{E}(f(X)) \\ &\quad - \left(\int f(x^{(i)}_S \cup X_C)\, d\mathbb{P}_{X_C} - \mathbb{E}(f(X)) \right) \\ &= \int f(x^{(i)}_{S\cup\{j\}} \cup X_{C\setminus\{j\}})\, d\mathbb{P}_{X_{C\setminus\{j\}}} \\ &\quad - \int f(x^{(i)}_S \cup X_C)\, d\mathbb{P}_{X_C}. \end{aligned} \tag{3.4}$$

Combinant tots els termes per construir la equació de Valors de Shapley, obtenim la equació de SHAP:

$$\phi_j^{(i)} = \sum_{S\subseteq\{1,\ldots,N\}\setminus\{j\}} \frac{|S|!(N-|S|-1)!}{N!} \cdot \left(\int f(x^{(i)}_{S\cup\{j\}} \cup X_{C\setminus\{j\}}) d\mathbb{P}_{X_{C\setminus\{j\}}} - \int f(x^{(i)}_S \cup X_C) d\mathbb{P}_{X_C} \right), \tag{3.5}$$

on el valor SHAP $\phi_j^{(i)}$ d'un atribut d'un valor, és la contribució marginal mitjana de l'atribut j del valor $x^{(i)}$ a totes les possibles coalicions. És una fòrmula prou similar a la de Valors de Shapley (3.1). Els axiomes formen els fonaments per definir els Valors de Shapley i com que els valors SHAP són Valors de Shapley amb una funció de valor específica i un joc definit, també segueixen aquests axiomes. Això va ser demostrat per Erik Štrumbelj & Igor Kononenko [5] i per Lundberg & S.-I. Lee [6].

Anem a explicar cada un dels quatre axiomes i com afecta a la interpretació dels Valors SHAP.

1. Eficiència: Els valors SHAP han de sumar la diferència entre la predicció de $x^{(i)}$ i l'esperança de la predicció:

   $$\sum_{j=1}^{N} \phi_j^{(i)} = f(x^{(i)}) - \mathbb{E}(f(X))$$

   Implicacions: Aquest axioma garanteix que els atributs es trobin a l'escala de la sortida, permetent-nos interpretar els resultats com a contribucions a la predicció.

2. Simetria: Si dos atributs d'una predicció particular contribueixen de manera igual a totes les possibles coalicions, la seva contribució haurà de ser la mateixa.

   $$v_{f,x^{(i)}}(S \cup \{j\}) = v_{f,x^{(i)}}(S \cup \{k\}), \quad \forall S \subseteq \{1,\ldots,N\}\setminus\{j,k\}$$

Llavors

$$\phi_j^{(i)} = \phi_k^{(i)}$$

Implicacions: Aquest axioma garanteix que els atributs no hauri-en de dependre de la seva ordenació. Si dos atributs contribueixen el mateix a una predicció, tindran el mateix valor SHAP. Això ens permetrà fer rànquings d'atributs utilitzant *importància SHAP*, que és una aglomeració de la rellevància de totes les prediccións a cada variable.

3. Nul: Un atribut que no contribueix a una predicció, tindrà valor SHAP 0.

$$v_{f,x^{(i)}}(S \cup \{j\}) = v_{f,x^{(i)}}(S), \quad \forall S \subseteq \{1,\dots,N\}\backslash\{j\}$$

Llavors

$$\phi_j^{(i)} = 0$$

Implicacions: Aquest axioma garanteix que atributs no utilitzats en una predicció, tinguin valor SHAP 0.

4. Additivitat: Per un joc amb pagaments combinats, $v_i + v_j$, els seus respectius valors SHAP són:

$$\phi_j^{(i)}(v_1) + \phi_j^{(i)}(v_2)$$

Implicacions: Aquest axioma garanteix que per un model d'ensam-blatge, com pot ser Random Forest, els valors SHAP finals siguin equivalents a la suma dels valors SHAP individuals. En l'exemple de model RF, els valors SHAP final, serien la suma dels valors SHAP de cada arbre.

En aquesta secció hem explorat els valors SHAP teòrics. No obstant això, ens enfrontem a un problema significatiu: en la pràctica, ens manca una expressió en forma tancada per a f i no tenim coneixement de les distribucions de X_C. Això significa que no podem calcular els valors SHAP, però, afortunadament, podem estimar-los.

3.1.5 Aproximacions SHAP

En aquesta secció, estudiarem les aproximacions als valors SHAP. Tot i que els valors SHAP poden ser calculats de forma exacta per jocs simples, s'han d'estimar per dues raons:

- Les funcions predictives que utilitza SHAP, requereixen ser integrades sobre la distribució d'atributs. No obstant això, ja que només disposem de dades i no tenim coneixement de les distribucions, hem de fer servir tècniques d'estimació com la integració de Monte Carlo.
- Els models de ML normalment utilitzen molts atributs. Ja que el nombre de coalicions augmenta exponencialment amb el nombre de característiques (2^N), podria resultar massa consumidor de temps calcular les contribucions marginals d'una característica per a totes les coalicions. En comptes d'això, hem de prendre mostres de les coalicions.

Integració de Monte Carlo ens permet substituir la integral per un sumatori i la distribució $\mathbb{P}$ per un mostreig de dades. Anem a veure com es calcula el valor SHAP:

$$\hat{v}(S) = \frac{1}{n}\sum_{k=1}^{n}(f(x_S^{(i)} \cup x_C^{(k)}) - f(x^{(k)})), \tag{3.6}$$

on n és el nombre de mostres extretes de les dades i $\hat{v}$ indica que això és una estimació de la funció de valor v.

Les contribucions marginals de l'atribut j afegits a la coalició S es donada per:

$$\hat{\triangle}_{S,j} = \hat{v}(S \cup \{j\}) - \hat{v}(S) = \frac{1}{n}\sum_{k=1}^{n}(f(x_{S\cup\{j\}}^{(i)} \cup x_{C\setminus\{j\}}^{(k)}) - f(x_S^{(i)} \cup x_C^{(k)})$$

Ara ja sabem calcular la contribució marginal utilitzant integració de Monte Carlo. Per calcular els valors SHAP de cada atribut, necessitem estimar la contribució marginal per a totes les possibles coalicions.

La fórmula per aproximar els valors SHAP és:

$$\hat{\phi}_j^{(i)} = \sum_{S\subseteq\{1,\ldots,N\}\setminus\{j\}} \frac{|S|!(N-|S|-1)!}{N!}\hat{\triangle}_{S,j} \tag{3.7}$$

El temps de càlcul augmenta exponencialment amb el nombre de característiques a causa del potencial de 2^N coalicions. Quan N és gran, hem de confiar en tècniques d'estimació que no requereixin passar per totes les coalicions.

Actualment, existeixen dues solucions:

- En alguns casos, podem fer servir l'estructura del model. Per a models purament additius, com ara models de regressió lineal sense

termes d'interacció, és suficient calcular una contribució marginal per característica. Inclús per a altres models com ara xarxes neuronals, hi ha mètodes d'estimació específics del model que eviten iterar a través de totes les coalicions.

- És possible fer mostreig de coalicions i després veurem que hi ha moltes formes de fer-ho

Els mètodes d'estimació varien en velocitat, precisió (normalment s'ha de veure compromesa la precisió o la velocitat) i aplicabilitat (alguns estimadors són específics per un tipus de model).

L'estimació a través de permutacions funciona creant una permutació aleatòria dels valors de les característiques d'una instància i després realitzant una generació de coalicions cap endavant i cap enrere. Aquest mètode té bon rendiment comparat amb altres estimadors [12].

Per veure com es calcula, hem de tornar a agafar la definició de SHAP a traves de permutacions. Sigui m el número de permutacions de l'atribut, amb $o(k)$ la k-èssima permutació, llavors podem estimar els valors SHAP de la següent forma:

$$\hat{\phi}_j^{(i)} = \frac{1}{m} \sum_{k=1}^{m} \hat{\triangle}_{o(k),j} \tag{3.8}$$

Però clar, si $m = N!$, haurem d'estimar totes les permutacions, anant en contra de l'objectiu, però m pot ser més petit i es poden mostrejar amb aquesta fòrmula, però com estem fent coalició endavant i enrere, la fórmula és la següent:

$$\hat{\phi}_j^{(i)} = \frac{1}{2m} \sum_{k=1}^{m} (\hat{\triangle}_{o(k),j} + \hat{\triangle}_{-o(k),j}), \tag{3.9}$$

on $-o(k)$ és la inversa versió de la permutació.

Hi ha molts altres mètodes, però no els estudiarem en aquest treball. Ara passarem a veure la implementació de SHAP a Python (llibreria SHAP). Estudiarem les diferents visualitacions i alguna aproximació de valors en funció del model utilitzat.

3.2 Aplicació a Python

El paquet de SHAP de Python [7] ha permès la utilització de Valors de Shapley al camp de la la Interpretabilitat de models d'aprenentatge

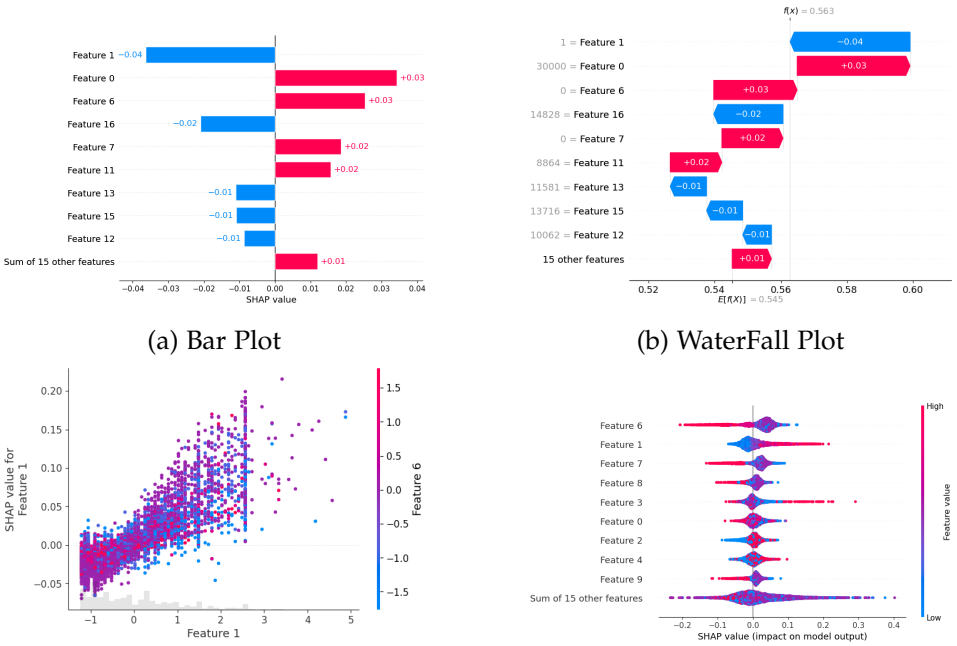
automàtic. A part d'estar basat en matemàtiques sòlides, s'ha popularitzat per diversos factors:

- Interpretació de Models Complexos: SHAP és especialment útil per interpretar models predictius complexos, com ara models d'aprenentatge profund i altres models d'aprenentatge automàtic complexos. Proporciona una explicació global i local dels resultats del model.
- Suport per a Diversos Tipus de Models: SHAP és versàtil i pot ser utilitzat amb diversos tipus de models, inclosos models de regressió, classificació, arbres de decisió, xarxes neuronals, etc. Això fa que sigui una eina àmpliament utilitzada en diferents àmbits d'aprenentatge automàtic.
- Visualitzacions Intuitives: SHAP proporciona eines per a la visualització d'atribucions, cosa que facilita la comprensió i la interpretació de les contribucions de cada característica a la sortida del model.
- Compatibilitat amb Diverses Llibreries d'Aprenentatge Automàtic: SHAP és compatible amb diverses llibreries populars d'aprenentatge automàtic com ara scikit-learn, XGBoost, LightGBM, TensorFlow, i altres.
- Desenvolupament Actiu i Comunitat Activa: SHAP és mantingut activament i té una comunitat activa de desenvolupadors. Això significa que es realitzen millores constants, s'afegeixen funcionalitats noves i es corregeixen possibles errors.

3.2.1 Tipus de visualitzacions

Sense les visualitzacions, no podriem entendre els valors SHAP, per tant entendre bé què es mostra en cada gràfic és crucial. N'hi ha molts i cada un té la seva utilitat, aquí mostrarem algun d'ells i només de les visualitzacions per a dades tabulades, però per obtenir una visió més complerta, recomanem anar a la documentació del paquet de Python SHAP, ja que és molt completa [7].

- Beeswarm: El gràfic beeswarm està dissenyat per mostrar un resum dens d'informació sobre com les característiques principals en un conjunt de dades afecten la sortida del model. Cada instància de l'explicació donada està representada per un únic punt a cada fila de característiques. La posició x del punt està determinada pel valor



(a) Bar Plot

(b) WaterFall Plot

(c) Scatter Plot

(d) Beeswarm Plot

Figura 3.1: Descripció general de les cuatre imatges.

SHAP d'aquella característica, i els punts s'amunteguen al llarg de cada fila de característiques per mostrar la densitat. El color s'utilitza per mostrar el valor original d'una característica.

- Waterfall/Bar: Els gràfics de cascada (waterfall plots) estan dissenyats per mostrar explicacions per a prediccions individuals, pel que esperen una sola fila d'un objecte d'explicació com a entrada. La part inferior d'un gràfic de cascada comença com el valor esperat de la sortida del model, i després cada fila mostra com la contribució positiva (roja) o negativa (blava) de cada característica mou el valor des de la sortida esperada del model sobre el conjunt de dades de fons fins a la sortida del model per a aquesta predicció.

  Els gràfics de barres, fan el mateix, però en lloc d'anar de l'esperança fins a la predicció, estan centrats a 0.

- Scatter: Un gràfic de dispersió de dependència mostra l'efecte que una única característica té sobre les prediccions fetes pel model.

- Force - Clustering: Un Force Plot de SHAP és una visualització que il·lustra com cada característica individual contribueix a la predic-

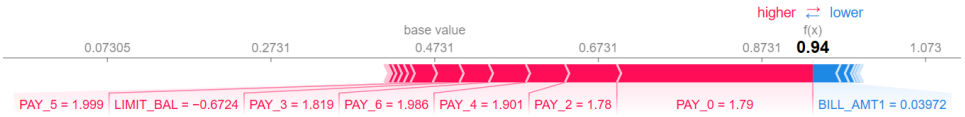


Figura 3.2: Force Plot de SHAP, amb els valors shap d'un fals positiu classificat per l'algorisme ANN.

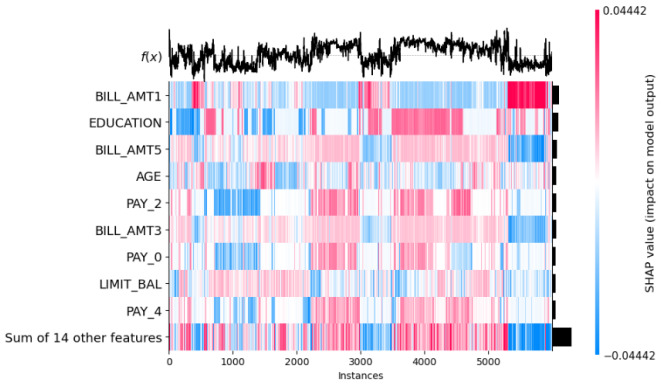


Figura 3.3: Heatmap Plot de SHAP, del model ANN.

ció específica d'un model de machine learning per a una instància concreta de dades. Cada barra horitzontal en el gràfic representa la magnitud i direcció de la contribució d'una característica, amb la barra vertical central indicant la mitjana global de les prediccions. Aquesta representació facilita una interpretació intuïtiva sobre com les característiques influeixen en la predicció, proporcionant enteniments profunds detallats sobre el raonament del model per a una observació particular. A través d'aquests gràfics, els analistes poden comprendre millor com les variables afecten les prediccions i guiar-se amb confiança en la interpretació dels models de machine learning.

El Clustering és simplement posar moltes visualitzacions Force una damunt de l'altre per obtenir una visió global.

- Heatmap: Passar una matriu de valors SHAP a la funció de gràfic de heatmap crea un gràfic amb les instàncies a l'eix de les x, les entrades del model a l'eix de les y, i els valors SHAP codificats en una escala de colors. Per defecte, les mostres s'ordenen utilitzant shap.order.hclust, que ordena les mostres segons una agrupació jeràrquica basada en la similitud de les explicacions.

  La sortida del model es mostra sobre la matriu del heatmap (centra-

da al voltant de la *.base_value* de l'explicació), i la importància global de cada entrada del model es mostra com un gràfic de barres a la dreta del gràfic (per defecte, això és la mesura shap.order.abs.mean de la importància global).

3.3 Limitacions

En aquesta secció examinarem críticament els mètodes. Tot i que SHAP s'està tornant molt popular, no està exempte de les seves limitacions (Tot i no haver estudiat en aquest treball les diferents aproximacions que es duen a terme al paquet SHAP, en parlarem en aquest apartat).

El temps de càlcul pot ser excessiu

Versions com estimació per Kernel o Mostreig són particularment lentes. Tot i això hi ha versions específiques de model que si són relativament ràpides, fent-les viables, com per exemple els Estimadors d'Arbre. Depenent de l'ús que se'n vulgui donar, utilitzar SHAP pot ser innecessàri. Per exemple si es vol veure l'efecte global d'un atribut, és molt més cost-eficient computar *PDPs* (Partial Dependency Plots).

El problema de la correlació

Per sort, tenim varies formes de minimitzar aquest problema, anem a veure dues tècniques:

- Reduir la correlació en el model: Això ho podem fer de diverses formes, principalment, eliminar atributs amb variància mínima, directament eliminant les variables correlacionades, implementar tècniques de reducció de dimensió com un PCA (Principal Component Analysis) que és un Anàlisi de Components Principals (Aquest té la contrapartida de perdre interpretabilitat, per tant, no és gaire desitjable), Enginyeria i combinació d'atributs.

- Jocs d'Owen: Explicació combinada d'atributs correlacionades amb l'explicador de particions. Es tracta d'agrupar les variables en un arbre binari utilitzant una mètrica basada en la correlació que agrupi jeràrquicament els atributs, on a cada nivell, es separa els atributs en dos grups segons la correlació entre ells. Una part bona és que si l'arbre està ben balancejat, passem de complexitat $O(2^p)$ a $O(p^2)$.

Interaccions poden ser poc interpretables

Aquesta noció prové de Kumar et al. (2020) [9], que va dir que l'axioma d'additivitat pot ser antiintuitiu. Una explicació comprensible hauria de ser conscisa, contrastiva i centrar-se en les causes 'anormals' [2]. En resum, ve deguda per les interaccións entre atributs, que poden ser complexes i donar peu a la construcció de visualitzacions extranyes. Un exemple molt clar és si dues variables que tenen la mateixa importància en una predicció però estan en escales diferents, es veuran amb valors diferents tot i en essència tenir el mateix valor predictiu.

Dos atributs interactuen quan la predicció no pot ser explicada per la suma dels dos atributs, sinó que el valor d'una modifica la dependència de l'altre. Les interaccions són la clau dels models d'aprenentatge automàtic i poder-les interpretar bé. La visualització global de les interaccions pot ser confusa, la bona notícia és que fent diversos tipus de visualitzacions (scatter, heatmap, beeswarm) podem obtenir una idea molt més clara.

Aquesta complexitat afegida pot comportar que els valors SHAP siguin malinterpretats o fins i tot puguin ser mostrats de forma enganyosa per a finalitats no legítimes maliciosament. Es possible crear intencionadament interpretacions enganyoses de SHAP; això no és un problema per l'analista que utilitza el model SHAP, però si ho és per al consumidor que no pot estar segur de la veracitat de la explicació.

Els Valors de SHAP no habiliten acció

A diferència de LIME i altres models surrogats, SHAP no pot ser utilitzat per fer afirmacions sobre canvis en prediccions degut a canvis de valors d'entrada. Només ens mostra com ha afectat cada atribut a la predicció concreta. Tot i això, si pot indicar quins atributs han de ser preservats per evitar regressió a la mitjana.

Example 3.16. Un usuari va al banc a demanar una hipoteca i li deneguen. Li agradaria saber per què ha estat denegada, però també què podria fer per conseguir-la. El per què, pot ser respost amb SHAP, però no podem saber quina acció hauria de fer l'usuari per a que li acceptessin la hipoteca. Aquesta segona pregunta pot ser resposta amb LIME 4 o amb Explicacions Contrafactuals.

Example 3.17. Considerem un model que prediu el rendiment del blat de moro basat en múltiples factors, com el temps i el fertilitzant. En un cas en el que es prediu una producció baixa, potser el mètode SHAP indica que el fertilitzant ha tingut un impacte positiu a la predicció. Llavors, hauriem

d'augmentar la utilització de fertilitzant? Només utlitzant SHAP, no ho podem saber.

Necessita accés a totes les dades

Aquest és un punt delicat, ja que la privacitat de les dades és violada completament, ja que es necessita accés a totes les dades. Només pot ser superat creant una nova base de dades que s'assembli a les dades reals però que no siguin instàncies de les dades d'entrenament, i això porta els seus propis problemes.

Capítol 4

LIME

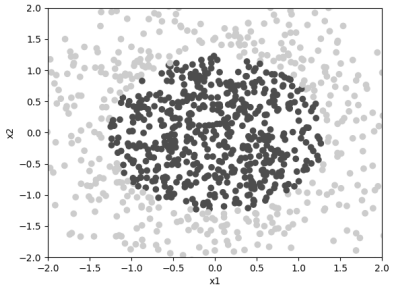
El 2016, Ribeiro et al. van introduir LIME (Local Interpretable Model-agnostic Explanations) [14], un mètode local d'explicació amb model surrogat. LIME és un mètode d'explicació que té com a objectiu "explicar les prediccions de qualsevol classificador d'una manera interpretable i fidel, aprenent un model interpretable localment al voltant de la predicció".

Els autors proposen LIME com una solució al "problema de confiança en l'aprenentatge automàtic", que comprèn els components "confiar en una predicció" i "confiar en el model". De fet, d'una banda, LIME proporciona explicacions per a resultats individuals de l'aprenentatge automàtic, oferint així una solució al problema "confiar en una predicció"; d'altra banda, seleccionant múltiples explicacions generades per LIME, aquest últim pot proporcionar una solució al problema "confiar en el model". LIME es caracteritza per tres propietats principals:

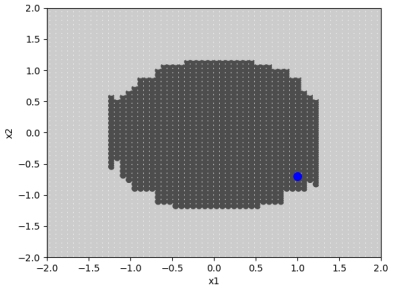
- Interpretabilitat: les explicacions generades són interpretables per humans, mitjançant l'ús de representacions interpretables del resultat del model seleccionat, utilitzant un espai de característiques més simple.

- Agnosticisme del model: LIME tracta el model original d'aprenentatge automàtic com una caixa negra i calcula aproximacions locals, independentment del tipus de model.

- Fidelitat local: LIME crea aproximacions locals de les instàncies individuals (és a dir, resultats del model), amb l'objectiu de romandre fidel al comportament original del model a la proximitat de la instància a explicar, però no a nivell global.



(a) Visualització de la base de dades sintètica de l'exemple LIME. Els punts gris clar representen un 0 i els punts gris fosc representen un 1.



(b) Visualització de la malla d'observacions i prediccions pel model RF. El punt blau és el punt a estudiar amb LIME.

Figura 4.1: Passos previs a aplicar LIME

4.1 Teoria

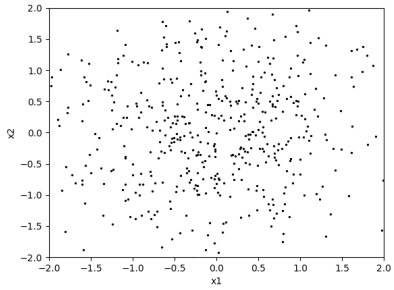
4.1.1 Visió general

LIME treballa generant mostres pertorbades del conjunt de dades original i observa com el model respon a aquests exemples. A continuació, ajusta un model interpretatiu local, com ara una regressió lineal, a aquestes mostres pertorbades i els resultats del model original. Aquest model local interpretable serveix com una representació simplificada del comportament del model de màquina en una àrea específica de l'espai de atributs.

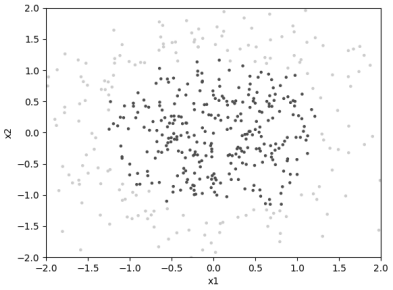
Si es desitja més detall sobre els càlculs, tots els codis utilitzats en aquest TFG es troben disponibles a [19]. A aquest repositori, es poden consultar tots els codis que han estat emprats per realitzar les visualitzacions i els models en aquest estudi. En particular, ara seguirem el programa "$LIME\_simplified.ipynb$", per veure pas a pas, com funciona un Model Surrogat Local en un exemple molt simple.

Mitjançant aquesta implementació pràctica de LIME, es busca demostrar com aquesta eina pot ser una peça clau per comprendre i interpretar les decisions dels models de màquines, augmentant la confiança i la utilitat en una àmplia gamma d'aplicacions.

En aquest exemple, es crea un conjunt de dades artificial amb dues variables explicatives com a entrada i una variable binària categòrica com a sortida. Aquest conjunt de dades, es generarà amb un conjunt agrupat clar per fer visualitzacions més interessants (un cluster de valors positius), amb dades amb fronteres de decisió no lineals (4.1a).



(a) Pertorbacions aleatòries al voltant de la instància a explicar sense evaluar.



(b) pertorbacions aleatòries al voltant de la instància a explicar evaluades al model.

Figura 4.2

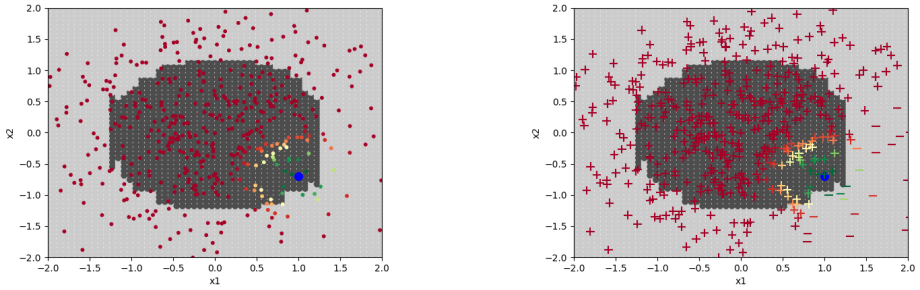
S'entrena el model, en aquest cas, entrenem un Random Forest (RF) i es crea una malla per observar les fronteres de decisió del classificador entrenat. Si no estàs familiaritzat amb el concepte de malla o la representació de les fronteres de decisió, pots pensar-ho com a mostreig de molts valors de x_1 i x_2 en un rang específic i utilitzar el classificador d'aprenentatge automàtic per predir la sortida i observar visualment les fronteres de decisió del model entrenat. El gràfic que es mostra a continuació il·lustra les fronteres de decisió del classificador (4.1b).

Ara ja podem procedir a amb els 4 passos per generar l'explicació utilitzant LIME:

a) **Generar Pertorbacions aleatòries al voltant de la instància a explicar:** Pel cas de dades tabulars, es recomana fer mostreig al voltant de la mitjana i la desviació estàndard de les variables explicatives.

b) **Utilitzar el classificador d'aprenentatge automàtic per predir les classes del nou conjunt de dades generat:** El classificador RF entrenat en els passos anteriors s'utilitza aquí per predir la classe de cada parella (x_1, x_2) en el nou conjunt de dades generat.

c) **Calcular les distàncies entre la instància a explicar i cada pertorbació i calcular els pesos (importància) de les instàncies generades:** Es calcula la distància entre cada instància generada aleatòriament i la instància que s'està explicant utilitzant la distància euclidiana (4.2a). En aquesta visualització, es mostra el valor dels pesos amb colors, el verd representa pesos més grans o instàncies amb major importància.



(a) Mostres amb pesos (b) Explicació Final

Figura 4.3: Visualització final de l'exemple

Per a explicacions d'altres tipus de dades, com ara imatges o dades de text, es pot utilitzar la distància del cosinus. Aquestes distàncies es mapegen després a un valor entre zero i un (pes) mitjançant una funció de nucli. Per a dades tabulars, la definició de l'amplada del nucli requereix atenció especial, especialment si les dades no s'han estandarditzat. Una discussió més detallada sobre aquest tema de l'amplada del nucli serà exposada posteriorment.

d) **Utilitzar el nou conjunt de dades generat, les seves prediccions de classe i la seva importància (pesos) per ajustar un model més simple i interpretable (lineal):** S'ajusta un model lineal (o un altre model intrínsecament interpretable). Aquest model interpretable, genera noves fronteres de decisió que són localment fidels al voltant de la instància explicada. Aquesta frontera de decisió lineal es pot veure amb els marcadors amb el símbol + i - (la visualització és (4.2b)). És important destacar que aquesta nova frontera de decisió lineal no és globalment fidel perquè es suposa que és un discriminador adequat només a la localitat de la instància que s'està explicant (punt blau). Això és degut a que LIME és una interpretació del model a través d'un model surrogat (en aquest cas regressió lineal), intrínsecament interpretable, d'un model RF que és NO interpretable.

Ara ja hem vist un exemple explicatiu de com funciona LIME, anem ara a sentar les bases matemàtiques de forma rigorosa. Per a fer-ho, ens recolzarem amb els articles originals de M. T. Ribeiro et. al. [14] [15], la recerca de Michaela Benk et. al. [17] i el TFG d'Aleix Nieto Juscafesa [18].

4.1.2 Fonaments

Definició 4.1. *Sigui D el conjunt de dades on $h(x, Y) \in D$, amb $x \in X$ i $Y \in \mathcal{Y}$. Sigui $H = \{h : X \to \mathcal{Y}\}$ un conjunt de models d'aprenentatge automàtic: Tot element $h \in H$ entrenat amb D és anomenat Caixa Negra. (Suposarem que $X \subset \mathbb{R}^d$, per un $d > 0$).*

Definició 4.2. *Sigui $x \in X$ una instància. Definim $\pi_x : \mathbb{R}^{d^1} \to \mathbb{R}$ amb $d^1 \leq d$, la mesura de proximitat a x.*

Definició 4.3. *$G = \{g : \mathbb{R}^{d^1} \to \mathcal{Y}\}$ com la classe de models d'aprenentatge automàtic intrínsecament interpretables.*

Tot i que LIME es pugui aplicar a tot tipus de dades (Taules, Text i Imatge), només ens centrarem en les dades tabulades. Per tant les instàncies (dades d'entrada) ja són interpretables, del contrari, hauriem de *Toquenitzar* o algun altre forma d'interpretació.

Si denotem $x \in \mathbb{R}^d$ com una instància i $h \in H$ una Caixa Negra, LIME utilitza entrades simplificades $x' \in \mathbb{R}^{d^1}$ per entrenar models $g \in G$ intrínsecament interpretable que sigui fidel localment a h, utilitzant com a mesura de proximitat π_x.
La explicació LIME $\xi(x)$ en el punt x es troba minimitzant $\mathcal{L}$, mantenint prou petit $\Omega(g)$ (penalització per la complexitat del model g):

$$\xi(x) = \arg\min_{g \in G} \left(\mathcal{L}(h, g, \pi_x) + \Omega(g)\right), \tag{4.1}$$

on:

- La pèrdua $\mathcal{L}(h, g, \pi_x)$ mesura la precisió amb què el model interpretable g s'aproxima al model "caixa negra" h per a la predicció donada $h(x)$, utilitzant una mesura de proximitat π_x.

- La mesura de la complexitat del model $\Omega(g)$ s'afegeix per penalitzar la complexitat dels models interpretables $g \in G$ (per exemple, en el cas dels arbres de decisió, $\Omega(g)$ pot denotar la seva profunditat).

Per exemple, si g és una regressió lineal en d_1 variables, la qual està entrenada en les representacions interpretables de dimensió d_1 d'una submostra de dades "propera" a x, llavors $\xi(x) \in \mathbb{R}^{d_1}$ és el vector amb d_1 components, que conté els coeficients estimats de g, o bé la seva projecció en un subespai $\mathbb{R}^{d_1'}$, amb $d_1' < d_1$. En el cas de la regressió lineal π_x és el número de components: d_1.

Mostreig per a l'Exploració Local

Volem minimitzar la pèrdua local $\mathcal{L}(h, g, \pi_x)$ sense fer cap suposició sobre h, ja que volem que l'explicador sigui independent del model. Així doncs, amb la finalitat d'aprendre el comportament local de h mentre les entrades varien, aproximem $\mathcal{L}(h, g, \pi_x)$ generant "instàncies pertorbades" aleatòries $z_1, \ldots, z_q \in \mathbb{R}^d$ i calcula els resultats $h(z_1), \ldots, h(z_q)$, que són les etiquetes asociades. Com generem aquestes pertorbacions? En dades tabulars, LIME crea noves instàncies pertorbant individualment cada característica de x a partir d'una distribució normal inferida del conjunt d'entrenament. Donat aquest conjunt de dades Z amb mostres pertorbades amb les etiquetes associades, optimitzem l'Equació (4.1) per obtenir una explicació $\xi(x)$.

(S'ha de ser molt meticulós en aquest aspecte, ja que definir un veïnat significatiu al voltant d'un punt pot ser difícil. Actualment, LIME utilitza un nucli de suavitzat exponencial per definir el veïnatge.)

Explicacions lineals disperses (Sparse Linear Explanations)

A LIME, es selecciona la família de models intèrprets G com la classe de models lineals $g(z) = w_g \cdot z$. Aquest tipus de model es fa servir per minimitzar una regressió lineal ponderada:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in Z} \pi_x(z)(h(z) - g(z))^2 \tag{4.2}$$

Aquesta funció de pèrdua minimitza les distàncies al quadrat entre les prediccions dede la caixa negra h i les del nostre model més senzill g. Això es fa per totes les instàncies pertorbades assignant a cadascuna d'elles un pes $\pi_x(z)$.

En altres paraules, LIME busca ajustar el model g de manera que minimitzi la diferència quadràtica ponderada entre les prediccions de la Caixa Negra h i les del model interpretable. Aquesta minimització es fa amb una consideració especial del pes de cada instància pertorbada, donat per $\pi_x(z)$, que reflecteix la proximitat relativa d'aquesta instància z a la instància d'interès x.

La funció $\pi_x(z) = \exp\left(-D(x, z)^2/\sigma^2\right)$ és un *nucli exponencial* definit sobre alguna mesura de distància $D(x, z)$. Aquest nucli exponencial s'utilitza com a funció de pes en el context de LIME per assignar pesos a les instàncies pertorbades z, basant-se en la seva distància a la instància d'interès x i σ com a amplada.

4.1.3 Algorisme SP-LIME

L'algorisme SP-LIME (Submodular Pick For Explaning Models) és una variació de LIME (segueix sent explicació de model agnóstic) que ens ajuda a tenir una visió global de com funciona el model, no només de com funciona al voltant d'una instància concreta. La idea darrera de l'algorisme és evaluar el model en diversos punts, que siguin representatius de la mostra global i juntar totes aquestes importàncies en una importància global de cada atribut. Aquesta importància augmenta en la mesura de que aquest atribut expliqui més instàncies.

Hi ha diversos punts d'interès en aquest algorisme, anem a veure'ls un per un:

- **Selecció d'Instàncies de Manera Submodulars:** S'ha de seleccionar un subgrup d'instàncies de la base de dades D, ha de tenir el mínim número possible d'elements ja que l'hem d'interpretar les persones i tenim temps/paciència/capacitats limitades (diguem que B és el nombre màxim d'instàncies que un individu pot interpretar), però al mateix temps ha de ser una mostra representativa global. A més s'ha de triar elements que tinguin explicacions diferents per evitar redundàncies. Per tant, la primera idea, seria triar aquest subgrup amb criteri expert. A partir d'ara, aquest subgrup li direm X i tindrà lingitud $|X| = n$. Per tant volem que $n \leq B$.

- **Càlcul d'importància:** Per a calcular la importància global de cada atribut, evaluarem el subgrup X d'instàncies i posarem les explicacions en una matriu, on cada fila serà un element de X i cada columna, la importància d'un atribut a cada instància 4.4. Es defineix una funció d'importància global I_j per a cada component j, que indica la importància d'aquest component en l'espai d'explicacions. Aquesta funció pot ser definida de diverses formes, en variables tipus text, es sol utilitzar: $I_j = \sqrt{\sum_{i=1}^{n} W_{ij}}$ i per imatges I ha de ser comparable al llarg de diferents super-pixels en diferents imatges.

- **Problema de selecció:** Trobar el subgrup que tingui cobertura màxima de la base de dades V tq $|V| \leq B$, és un problema NP-Hard ("NP-Hard" és una classe de problemes en teoria de la complexitat computacional. Un problema és considerat NP-hard si és, com a mínim, tan difícil de resoldre com els problemes NP o "No Determinista Polinòmic". Això significa que no hi ha un algorisme

$$W = \begin{array}{|c|cccc|} \hline & h_1 & h_2 & \cdots & h_{d'} \\ \hline X_1 & w_{11} & w_{12} & \cdots & w_{1d'} \\ X_2 & w_{21} & w_{22} & \cdots & w_{2d'} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X_n & w_{n1} & w_{n2} & \cdots & w_{nd'} \\ \hline \end{array}$$

Figura 4.4: Matriu W que representa la importància dels d' atributs per a totes les instàncies del subgrup X.

eficient polinòmic conegut per resoldre el problema en temps raonable). L'algorisme SP-LIME ofereix una aproximació *greedy* (golafre) que, en cada iteració, selecciona la instància amb el major guany marginal en cobertura.

Es vol una cobertura màxima i minimitzar instàncies. Formalitzem aquesta intuïció de cobertura no redundant en l'Equació (4.3), on definim la cobertura com la funció de conjunt c, que, donats W i I, calcula la importància total de les característiques que apareixen en almenys una instància en un conjunt V.

$$c(V, W, I) = \sum_{j=1}^{d'} \mathbf{1}_{[\exists i \in V : W_{ij} > 0]} I_j \tag{4.3}$$

El problema de la selecció, definit a l'Equació (4.4), consisteix en trobar el conjunt V, amb $|V| \leq B$, que aconsegueixi la major cobertura.

$$\text{Pick}(W, I) = \underset{V, |V| \leq B}{\text{argmax}}\, c(V, W, I) \tag{4.4}$$

El problema descrit a l'Equació (4.4) consisteix a maximitzar una funció de cobertura ponderada i és NP-hard. Sigui $c(V \cup \{i\}, W, I) - c(V, W, I)$ el guany de cobertura marginal d'afegir una instància i a un conjunt V. Degut a la submodularitat, un algorisme greedy que afegix iterativament la instància amb la major guanyança de cobertura marginal a la solució ofereix una garantia d'aproximació constant de $1 - \frac{1}{e}$ respecte a l'òptim.

4.1.4 Avantatges

L'avanatge més destacat de LIME és la seva **agnosticitat**. Inclús si el model d'aprenentatge automàtic subjacent és substituit, el mateix model local interpretable pot ser utilitzat per a l'explicació. I LIME és un dels pocs mètodes que funciona per a **taules, text i imatges**.

Un altre avantatge de LIME és que les explicacions generades són **concises i comprensibles** pels humans, els models locals de substitució es beneficien de la literatura i l'experiència en la formació i interpretació de models interpretables.

Com a resultat, l'aplicació de LIME pot ser més apropiada en situacions en què el destinatari de l'explicació és una persona sense coneixements especialitzats o algú amb temps limitat. En situacions en què puguem estar legalment obligats a explicar detalladament una predicció, això podria ser insuficient.

4.1.5 Limitacions

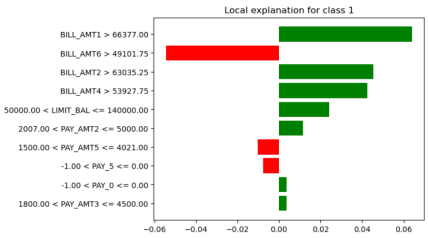
La **definició correcta del veïnat és un problema** molt gran i no resolt quan s'utilitza LIME amb dades tabulars. Aquest és el principal problema de LIME i la raó per la qual s'ha d'utilitzar amb molta cura. Per a cada aplicació, cal provar diferents configuracions de nucli i veure si les explicacions tenen sentit. Desafortunadament, aquest és l'única solució per trobar amplades de nucli adequades fins a dia d'avui.

El mostratge podria millorar en la implementació actual de LIME. Es mostregen punts de dades des d'una distribució gaussiana, ignorant la correlació entre característiques. Això pot conduir a punts de dades improbables que després es poden utilitzar per aprendre models d'explicació locals.

La **complexitat del model d'explicació** ha de ser definida amb antelació. L'usuari sempre ha de definir el compromís entre fidelitat i especificitat.

Un altre problema realment gran és la **inestabilitat de les explicacions**. Les explicacions de dos punts molt propers poden variar considerablement. També, si es repeteix el procés de mostratge, les explicacions resultants poden ser diferents. La inestabilitat significa que és difícil confiar en les explicacions, i cal ser molt crític.

Conclusió: Els models de substitució locals, amb LIME com a implementació concreta, són molt prometedors. Però el mètode encara està en fase de desenvolupament i molts problemes han de resoldre's abans que



(a) Bar Plot

	Rule	Value	Importance
0	BILL_AMT1 > 66377.00	70122	0.063998
1	BILL_AMT6 > 49101.75	70212	-0.054412
2	BILL_AMT2 > 63035.25	69080	0.045362
3	BILL_AMT4 > 53927.75	69753	0.042438
4	50000.00 < LIMIT_BAL <= 140000.00	70000	0.024244
5	2007.00 < PAY_AMT2 <= 5000.00	3112	0.011432
6	1500.00 < PAY_AMT5 <= 4021.00	2500	-0.010324
7	-1.00 < PAY_5 <= 0.00	0	-0.007640
8	-1.00 < PAY_0 <= 0.00	0	0.003741
9	1800.00 < PAY_AMT3 <= 4500.00	3000	0.003681

(b)

Figura 4.5: Descripción general de las cuatro imágenes.

es pugui aplicar amb seguretat.

4.2 Aplicació a Python

En la implementació de LIME a Python, podem utilitzar la biblioteca 'lime' [16] que proporciona eines específiques per a entendre com el model funciona i com pren decisions particulars. Hi ha diversos tipus de visualitzacions i resultats que ens permet produir el paquet 'lime'. Els exemples de codi, els podrem trobar al repositori de Github del treball de recerca [19].

Hi ha diverses visualitzacions, aquí en veurem dues molt útils i en l'apartat de Resultats (5), quan estudiem els models utilitzarem la resta. La primera visualització que podem observar (4.5a) és un gràfic que conté les normes del model surrogat i la importància de cada atribut en forma de gràfic de barres horitzontal.

La segona visualització (4.5b) és una taula creada en aquest treball (no en el paquet) que serveix per tenir una visió clara i directa de tot el que LIME et pot aportar: la interpretació del model surrogat, la importància de cada atribut (en verd si és positiu i en vermell si és negatiu per fer-ho més visual) i el valor de cada atribut per tenir el context.

Capítol 5

Resultats i evaluació

En aquest treball, ens hem immers en el fascinant camp de la Intel·ligència Artificial Explicable (XAI), que busca proporcionar transparència i interpretació als models d'aprenentatge automàtic. Hem iniciat la nostra exploració introduint-nos en el món de la XAI, explorant les diverses tècniques i metodologies que permeten entendre millor com els models de machine learning prenen decisions. Aquesta introducció ha estat fonamental per a la comprensió dels conceptes clau que governen la interpretació dels models, i ha establert el marc necessari per aprofundir en dues de les metodologies més prominents en aquest àmbit: SHAP i LIME.

En el següent apartat, hem realitzat una visió profunda de LIME (Local Interpretable Model-Agnostic Explanations) i SHAP (SHapley Additive exPlanations). Hem examinat les seves particularitats, com la manera en què generen explicacions locals per a les prediccions dels models de machine learning, així com les diferències fonamentals en els seus enfocaments i aplicacions. Aquesta exploració detallada ha posat de manifest les forces i les limitacions de cadascun d'aquests mètodes.

Un aspecte clau del nostre treball ha estat l'aplicació pràctica d'aquests mètodes a l'anàlisi de la Probabilitat de Default (PD). Mitjançant l'ús de SHAP i LIME, hem examinat de manera crítica la interpretabilitat i la confiança dels models en el context específic de la PD. La comparació entre aquests dos mètodes ha estat particularment reveladora, destacant els seus avantatges i desavantatges en la identificació i comprensió dels factors determinants que contribueixen a les prediccions de la PD. Els resultats obtinguts han ofert insights valuosos que contribueixen a una millor comprensió dels models de credit scoring i, al mateix temps, han posat de manifest els reptes associats amb la interpretació en aquesta àrea

específica.

5.1 SHAP vs LIME

Comparativa entre SHAP i LIME:

Eficiència en la Distribució de l'Explicació:

- **SHAP:** La diferència entre la predicció i la mitjana està distribuïda de manera justa entre les característiques, amb propietats eficients dels valors de Shapley.
- **LIME:** La interpretació de les explicacions contrafactuals és clara i directa, permetent canvis als valors de les característiques per obtenir una nova predicció específica.

Garantia d'Explicacions i Propietats Teòriques:

- **SHAP:** Té una base teòrica sòlida amb axiomes com eficiència, simetria i additivitat, garantint propietats com consistència i exactitud local.
- **LIME:** Ofereix explicacions clares de contrafactuals, sense suposicions addicionals, però manca de propietats teòriques sòlides com les de SHAP.

Velocitat de Càlcul:

- **SHAP:** Requereix considerable temps de càlcul per obtenir una solució exacta, la qual cosa pot ser impracticable en molts casos reals.
- **LIME:** És més ràpid, ja que les explicacions contrafactuals poden ser generades de manera més eficient.

Aplicació a Dades Reals:

- **SHAP:** Necessita accés a les dades per calcular el valor de Shapley per a noves instàncies, amb dificultats en situacions de correlació entre característiques.
- **LIME:** No requereix accés a les dades, només a la funció de predicció del model, oferint una opció atractiva per protegir la privadesa de les dades.

Complexitat de les Explicacions:

- **SHAP:** Té una complexitat potencialment alta, amb explicacions que inclouen totes les característiques.
- **LIME:** Permet explicacions més simples i selectives, ja que es poden destacar les característiques canviades en els contrafactuals.

En resum, SHAP destaca per la seva solidesa teòrica i distribució eficient de l'explicació, però amb una velocitat de càlcul més lenta. LIME, en canvi, és més ràpid i ofereix explicacions clares, tot i que amb menys garanties teòriques i una complexitat potencialment menor. La elecció entre ambdós dependrà de les prioritats específiques de l'aplicació i les restriccions computacionals.

5.2 PD

Anem a veure els resultats aplicats. En general, podem fer referència a la incapacitat d'un client per pagar, el seu impagament d'una quota o la seva fallida personal, tot com a possibles problemes de no pagament. No obstant això, cadascun d'aquests escenaris és el resultat de circumstàncies diferents. A vegades, és a causa d'un canvi sobtat en la font d'ingressos d'una persona a causa de la pèrdua de feina, problemes de salut o la incapacitat per treballar. Altres vegades, és deliberat, per exemple, quan el client sap que no té prou solvència per utilitzar una targeta de crèdit, però encara la utilitza fins que la targeta és bloquejada pel banc. En aquest últim cas, es tracta d'un tipus de frau, que és molt difícil de preveure i constitueix un gran problema per als creditors.

Per abordar aquest problema, les empreses de targetes de crèdit intenten predir el potencial impagament o avaluar la probabilitat de risc d'una quota amb antelació. Des del punt de vista del creditor, com més aviat es detectin els comptes amb potencialment alt risc d'impagament, menor serà la pèrdua. Per a això, és crucial una estratègia eficaç per predir amb antelació un compte amb potencial d'impagament si els creditors volen prendre accions preventives. A més, també podrien investigar i ajudar al client proporcionant suggeriments necessaris per evitar la fallida i minimitzar les pèrdues.

Com ja hem dit abans, aplicarem SHAP i LIME al model ANN, degut a que té un menor número de falsos positius. Amb aquest estudi, pretenem obtenir una millor idea de com funciona, de forma general i de forma específica per a cada instància. Ara procedirem a veure resultats d'aplicar SHAP i LIME.

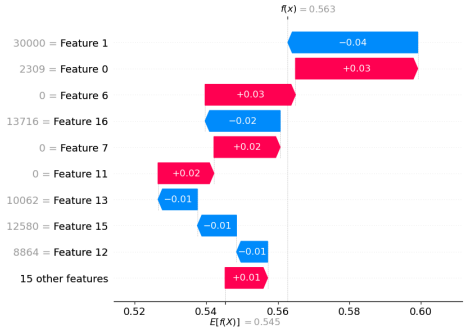


Figura 5.1: En aquest gràfic es pot observar com la "Feature 0", amb valor 2309, és l'ID i està augmentant un 3% la probabilitat d'impagament.

5.2.1 Experiment 1

S'ha fet una visualització general amb SHAP de Beeswarm i Waterfall per anar guanyant familiaritat amb les visualitzacions. S'ha trobat que el model s'havia entrenat utilitzant l'ID com a atribut del model, el que no hauria de ser, ja que és una variable aleatoria donada per l'entitat bancària. En (5.1), es pot observar l'impacte que ha tingut l'ID d'un usuari a la predicció final del model ANN. A l'Annex, hi ha dos visualitzacions més al respecte, la primera 2, és un gràfic de "Beeswarm" que mostra l'efecte general que tenen els atributs a les prediccions. Al (3) es pot veure un gràfic de Dispersió en el que es pot observar la interacció entre l'ID i el Sexe amb la predicció.

A partir de la següent secció, totes les següents visualitzacions seran, havent tornat a entrenar tots els models i recalculat tots els valors SHAP. Cal remarcar, la complexitat de càlcul dels valors SHAP, per a les aproximacions de SVM, no es va poder calcular tots, sinó que només 100 valors degut a que per a calcular els valors per a les 6000 instàncies es va calcular un temps estimat de 300h!!

5.2.2 Experiment 2

Un cop reentrenats els models i recalculats els valors SHAP, ja podem procedir a l'estudi del model. Començarem estudiant diverses visualitzacions dels valors SHAP i després les visualitzacions LIME buscant unificar resultats.

En quan a SHAP, començarem estudiant la importància relativa dels diferents atributs, que és un llistat dels atributs ordenats per la mitjana en valor absolut dels seus valor SHAP al llarg de tota la mostra. Aquest

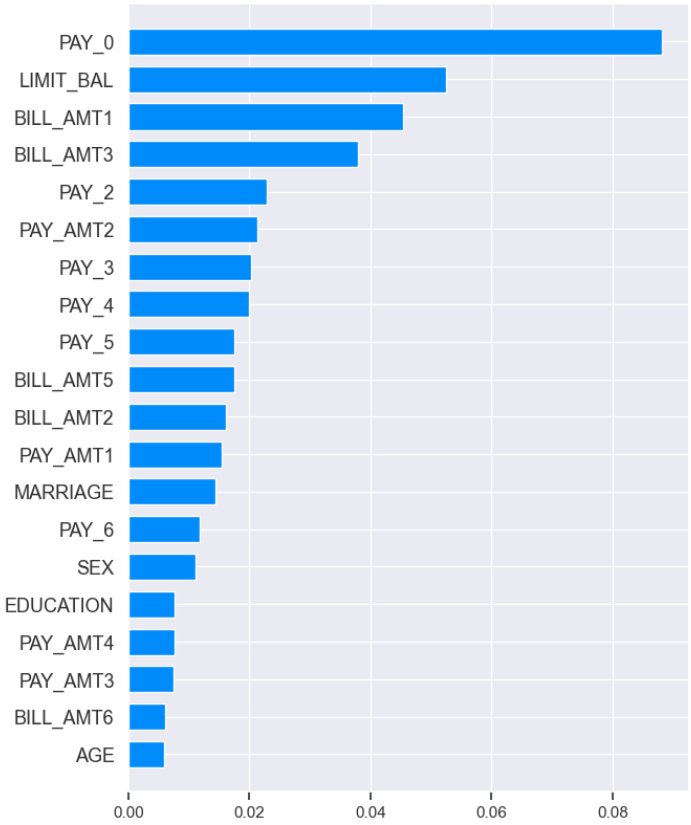


Figura 5.2: Importància de cada atribut a la predicció del model ANN, al llarg de tota la mostra.

gràfic (5.2) ens aporta molta informació, ja que podem veure quins atributs són més rellevants pel model. Per exemple, si observéssim que una de les variables més importants són l'AGE o el SEXE, ja sabriam que cal modificar-lo. Per complementar aquesta informació, podem visualitzar el gràfic "Beeswarm" amb valors absolut (1) de SHAP, per veure l'impacte absolut de totes les instàncies al llarg del model.

Els gràfics Beeswarm, es poden organitzar de dues formes, la primera és en l'ordre de la importància relativa que acabem d'estudiar, i l'altre és en funció dels valors SHAP més extrems en valor absolut, a l'Annex (4a) hi ha un gràfic comparatiu entre dos gràfics Beewsarm, ordenats de les dues formes esmentades. En ells, podem observar com cada atribut contribueix a la predicció de cada instància de la mostra.
Ara estudiarem alguns dels resultats obtingut en el gràfic (4b):

- **PAY_0:** En aquest gràfic es mostra que els valors més alts (vermells)

tenen una contribució més altament positiva (un valor de SHAP) a la PD. Per valors baixos, la contribució és menor i negativa. És a dir, que per valors alts de PAY_0, la probabilitat d'impagament augmenta i per valors baixos, la probabilitat d'impagament disminueix. I té sentit, ja que aquesta variable indica els mesos de retard que es té en el mes 09/2005.

- **LIMIT_BAL:** En aquest gràfic, valors vermells a l'esquerra i valors blaus a la dreta, és a dir, que hi ha una correlació inversa entre l'import de crèdit concedit i la probabilitat d'impagament. I té sentit, ja que el banc vol donar molts diners als clients que son "bons pagadors".

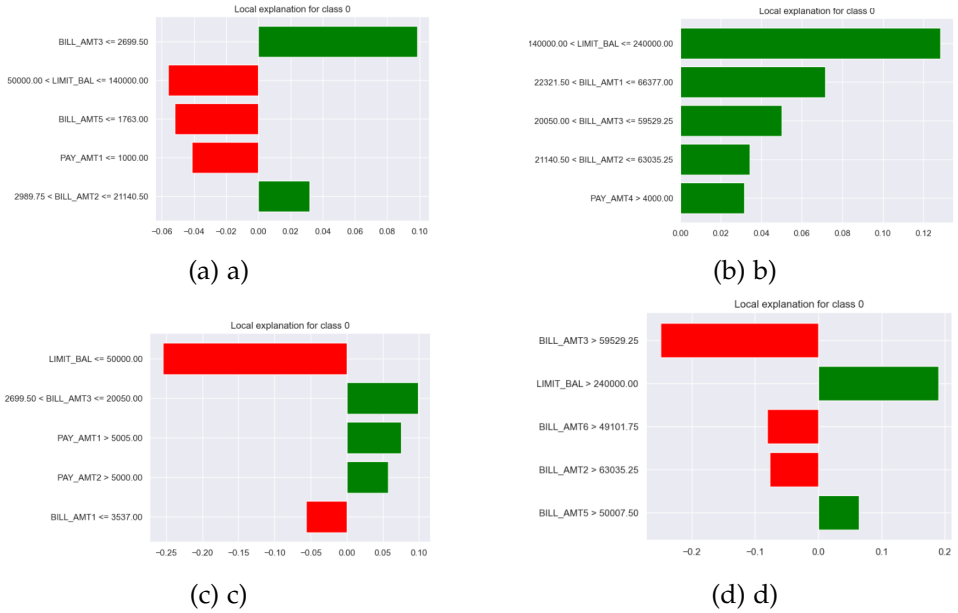
- **SEX:** En aquest gràfic, podem observar que els valors vermells estan a l'esquerra i els blaus a la dreta, per tant, que (segons el nostre model d'ANN) la probabilitat d'impagar és més alta per als homes que les dones.

- **AGE:** En aquest gràfic, podem observar que els valrs blaus estan a la dreta i els vermells a l'esquerra, el que ens indica que hi ha una correlació directa entre l'edat i la probabilitat d'impagament.

- **EDUCATION:** En aquest gràfic, podem observar que els valors vermells estan a l'esquerra i els valors blaus a la dreta, per tant, com més formada està la persona, menys probabilitat té d'impagar.

- **BILL_AMT1:** En el gràfic es pot observar com hi ha punts vermells als extrems i el centre té tant punts liles com blaus, com vermells. Per tant, es pot interpretar que hi ha un rang en el que a mesura que aubmenta l'atribut, la PD disminueix i fora d'aquest rang, la PD augmenta.

Podem comparar aquest resultats amb les visualitzacions de "SB-LIME". Calcular l'explicació LIME per a totes les instàncies és costós en termes de temps. En aquest cas, el mètode utilitza explicacions candidates preses de les dades de manera uniforme i aleatòria. La grandària de la mostra es dóna per '$sample\backslash size$'. En cas contrari, les explicacions es generaran per a tot el conjunt de dades. Fem servir 500 mostres aleatòries i en seleccionem les 4 amb la major cobertura d'informació. Per veure alguna cosa nova i explorar les capacitats de LIME, establim a 5 el nombre de característiques que volem per a cada explicació (les 5 més importants en cada cas). Des d'aquí podem comparar-ho amb la importància global de les característiques de la SHAP.



(a) a) (b) b)

(c) c) (d) d)

Figura 5.3: Visualització de l'algorisme SB-LIME sobre el model ANN.

En aquesta figura (5.3), podem observar les 4 explicacions més representatives de la mostra, amb els 5 atributs més signficatius de l'algorisme SB-LIME sobre el model ANN. Les regles que es generen per a cada observació són diferents, ja que són generades per un model local i surrogat. Per exemple, es pot observar que la variable LIMIT\BAL apareix en tres de les quatre explicacions, però en dues d'elles, per a disminuir la PD, cal que el valor sigui superior a 140000, però en la tercera observació, és suficient amb que el límit sigui superior a 50000 per a disminuir la PD.
També es pot estudiar l'efecte de cada atribut a la predicció final amb l'ajuda dels gràfics "Scatter". De fet, aquests gràfics ens permeten estudiar l'impacte de les interaccions entre qualsevol parell atributs. A l'Annex (5a 5b), es poden trobar dues visualitzacions d'aquests tipus de gràfic, que mostren el mateix gràfic per al model ANN i KNN, on podem observar que l'ANN es comporta de forma més elegant. També es pot observar que els resultats obtinguts en aquests dos gràfics concorden amb el gràfic "Beeswarm".

Els resultats obtinguts en aquesta prova tenen sentit, tant en les visualitzacions de SHAP com en les de LIME. Hem generat visualitzacions per a obtenir un millor coneixement de com funciona el nostre model a nivell general, ara només resta veure casos particulars.

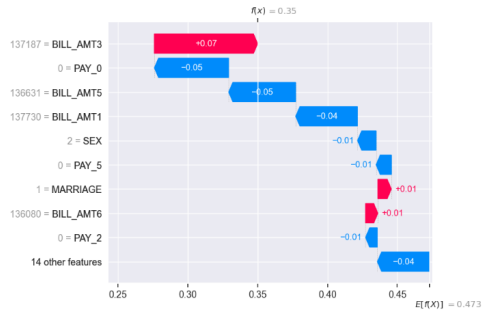


Figura 5.4: Aquest gràfic mostra les contribucions marginals de cada atribut a la predicció

5.2.3 Experiment 3

Les llibreries SHAP i LIME ens aporten diverses eines per visualitzar el comportament del model en observacions particulars. no cal estudiar totes les observacions, ja que ja hem obtingut una visió general del model, ara volem estudiar observacions en les que el model hagi fet una predició errònia; estudiarem un fals positiu, en particular, el fals positiu amb menys predicció d'impagament.

En aquest gràfic (5.4, podem veure les contribucions marginals de cada atribut a la predicció de la probabilitat d'impagament en aquesta observació, utilitzant els valors SHAP. En aquestes dues imatges (7 6), podem veure les normes i els pesos de cada atribut segons LIME.

Capítol 6

Treball Futur

Un aspecte a considerar per a futures investigacions és l'exploració en profunditat de les diferents aproximacions de SHAP en funció del model. Aquest és un camp molt interessant degut a la impossibilitat de calcular els valors SHAP exactes en la majoria dels casos. Un altre forma d'extendre aquesta investigació, és estudiant les explicacions detallades de les interaccions entre els diferents atributs. També es podrien realitzar comparacions detallades dels resultats obtinguts amb tècniques com les Gràfiques de Dependència Parcial (PDP), Explicacions Individuals Continues (ICE), i Efectes Locals Acumulats (ALE).

Bibliografia

[1] U. Kamath & J. Liu, Explainable Artificial Intelligence: An Introduction to Interpretable Machine Learning, 2021.

[2] C. Molnar, Interpretable machine learning, 2022.
https://christophm.github.io/interpretable-ml-book/

[3] C. Molnar, Interpreting Machine Learning Models With SHAP: A Guide with Python Examples And Theory On Shpaley Values, 2023.

[4] L. Shapley, A Value for n-Person Games, 1953.

[5] E. Štrumbelj & I. Kononenko, An Efficient Explanation of Individual Classifications using Game Theory, 2010.

[6] S. Lundberg & S.-I. Lee, A unified approach to interpreting model predictions: Advances in Neural Information Processing Systems, 2017.

[7] S. Lundberg & S.-I. Lee, Python Implementation of SHAP, 2017.
https://github.com/shap/shap
https://shap.readthedocs.io/en/latest/

[8] S. Lundberg et al, From local explanations to global understanding with explainable AI for trees, 2020

[9] Kumar et al, Problems with Shapley-value-based explanations as feature importance measures, 2020

[10] Machine Learning TV, Understanding The Shapley Value. YouTube, 2021.
https://www.youtube.com/watch?v=9OFMRiAVH-w

[11] I. Verneda, Teoria de Jocs: Ínexs de Poder i Aplicació en Aliances Post-Electorals, 2023.

https://diposit.ub.edu/dspace/bitstream/2445/203620/2/tfg_verneda_i_esteve_ignasi.pdf

[12] R. Mitchell & J. Cooper et al, Sampling permutations for Shapley value estimation, 2022.

[13] A. Magaña, Els jocs cooperatius amb utilitat transferible, 1998.

[14] M. T. Ribeiro & S. Singh & C. Guestrin, Local Interpretable Model-Agnostic Explications, 2016.

[15] M. T. Ribeiro & S. Singh & C. Guestrin, Why should I trust you?, 2016.

[16] M. T. Ribeiro & S. Singh & C. Guestrin et al, LIME Documentation, 2016.
https://lime-ml.readthedocs.io/en/latest/

[17] M. Benk & A. Ferrario, Explaining Interpretable Machine Learning: Theory, Methods and Applications, 2020.

[18] A. Nieto, An introduction to explainable artificial intelligence with LIME and SHAP, 2022
https://github.com/aleixnieto/TFG

[19] J. Orteu, Github amb tots els codis utilitzats per generar les visualitzacions i els models d'aquest estudi, 2023
https://github.com/Joanorteu99/Interpreting-Credit-Score-Models

Annex A: Taules

	Accuracy	**Precision**	**Recall**	**F1**	**AUROC**	**BSS**
DT	0.798	0.802	0.983	0.883	0.561	0.855
ANN	0.765	0.883	0.794	0.835	0.713	0.565
LR	0.815	0.839	0.943	0.888	0.654	0.858
SVM	0.808	0.846	0.920	0.881	0.665	0.860
RF	0.813	0.845	0.931	0.886	0.664	0.859
KNN	0.754	0.851	0.831	0.840	0.658	0.818

Taula 1: Resum del Rendiment Predictiu de sis Enfocaments de Puntuació de Crèdit (Amb Validació Creuada) a partir dels experiments previs.

	Positiu Real	**Fals Negatiu**	**Fals Positiu**	**Negatiu Real**
DT	2239	40	570	93
ANN	1855	481	246	418
LR	2204	133	422	242
SVM	2150	186	391	272
RF	2173	164	398	266
KNN	1942	395	342	322

Taula 2: Resultats de les *Confusion Matrix* dels sis Enfocaments de Puntuació de Crèdit (Amb Validació Creuada) a partir dels experiments previs.

Atribut	**Atribut ID**	**Descripció**	**Tipus**	**Categories**
LIMIT BAL	X1	Import crèdit concedit	Numèrica	
SEX	X2	Gènere	Categòrica	1-Masculí 2-Femení
EDUCATION	X3	Educació	Categòrica	1-Postgrau 2-Universitat 3-Secundària 4-Altres
MARRIAGE	X4	Estat Civil	Categòrica	1-Casat 2-Solter 3-Altres
AGE	X5	Edat	Numèrica	
PAY_0	X6	Estat de pagament 09-2005	Categòrica	-1-Al dia 1-retràs d'un mes ⋮ 9-retràs de 9 mesos
PAY_2	X7	Estat de pagament 08-2005	Categòrica	(igual que l'anterior)
PAY_3	X8	Estat de pagament 07-2005	Categòrica	(igual que l'anterior)
PAY_4	X9	Estat de pagament 06-2005	Categòrica	(igual que l'anterior)
PAY_5	X10	Estat de pagament 05-2005	Categòrica	(igual que l'anterior)
PAY_6	X11	Estat de pagament 04-2005	Categòrica	(igual que l'anterior)
BILL AMT_1	X12	Extracte de factura 09-2005	Numèrica	
BILL AMT_2	X13	Extracte de factura 08-2005	Numèrica	
BILL AMT_3	X14	Extracte de factura 07-2005	Numèrica	
BILL AMT_4	X15	Extracte de factura 06-2005	Numèrica	
BILL AMT_5	X16	Extracte de factura 05-2005	Numèrica	
BILL AMT_6	X17	Extracte de factura 04-2005	Numèrica	
PAY AMT_1	X18	Total pagat 09-2005	Numèrica	
PAY AMT_2	X19	Total pagat 08-2005	Numèrica	
PAY AMT_3	X20	Total pagat 07-2005	Numèrica	
PAY AMT_4	X21	Total pagat 06-2005	Numèrica	
PAY AMT_5	X22	Total pagat 05-2005	Numèrica	
PAY AMT_6	X23	Total pagat 04-2005	Numèrica	

Taula 3: Descripció de la base de dades utilitzada

Annex B: Programari

Codi 1: Codi Python genèric per calcular Valors SHAP

```python
import shap
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test =
   train_test_split(X, y, test_size=0.2,
   random_state=42)
scaled_X_train=scaler.fit_transform(X_train)
scaled_X_test=scaler.transform(X_test)

x_sub = shap.sample(scaled_X_train,100)
explainer = shap.Explainer(model.predict_proba,
   x_sub)
shap_values = explainer(scaled_X_test)
```

Codi 2: Codi Python per visualitzar diversos gràfics SHAP

```python
#Beeswarm
shap.plots.beeswarm(shap_values[:,:,0],max_display=10)

#Waterfall & Bar
shap_val_0 = shap_values[index, :, 0]
shap_val_0.data = X_test.iloc[index]
shap.plots.waterfall(shap_val_0, max_display =
   max_display)
shap.plots.bar(shap_val_0, max_display = max_display)

#Heatmap
shap.plots.heatmap(shap_values[:, :, 0])

```

```
#Scatter
shap.plots.scatter(shap_values[:,1,0],color =
    shap_values[:,0,0])

#Summary Plot
shap.summary_plot(shap_values[:,:,0], X_2,
    plot_type="bar")
```

Codi 3: Codi Python per visualitzar diversos gràfics LIME

```
explainer = lime.lime_tabular.LimeTabularExplainer(
            X_train.values, mode="classification",
            feature_names=X_train.columns)

instance_to_explain = X_test.iloc[2]
explanation = explainer.explain_instance(
                instance_to_explain,
                model.predict_proba)

explanation.show_in_notebook(show_table=True)
f = explanation.as_pyplot_figure()
```

Annex C: Visualitzacions

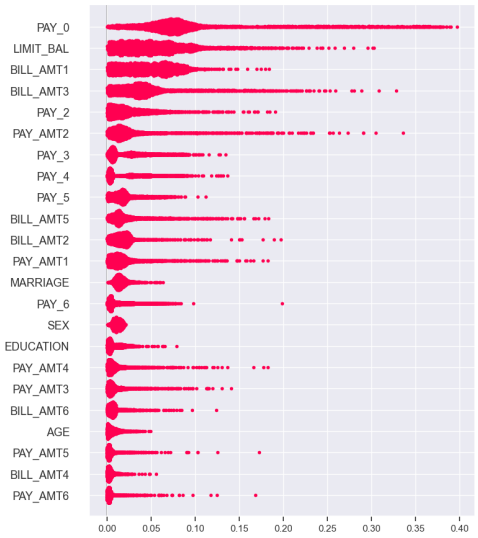


Figura 1: Gràfic "Beeswarm" amb contribució SHAP en valor absolut, aquest gràfic pot contribuir a una millor visualització de la importància de cada atribut al llarg de la mostra sobre el model ANN.

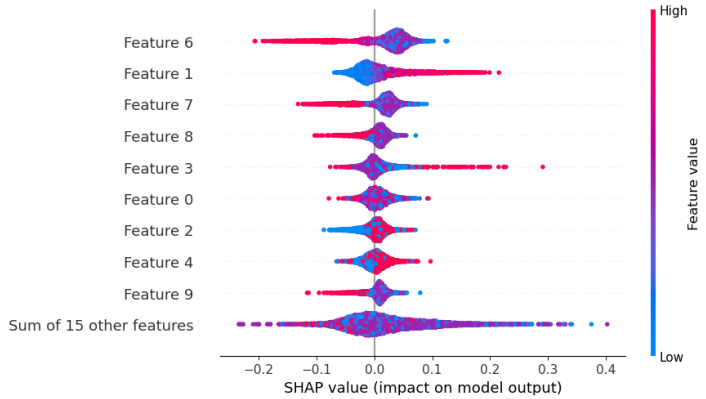


Figura 2: Gràfic "Beeswarm" en la base de dades que inclou l'ID, per això hi ha 24 atributs

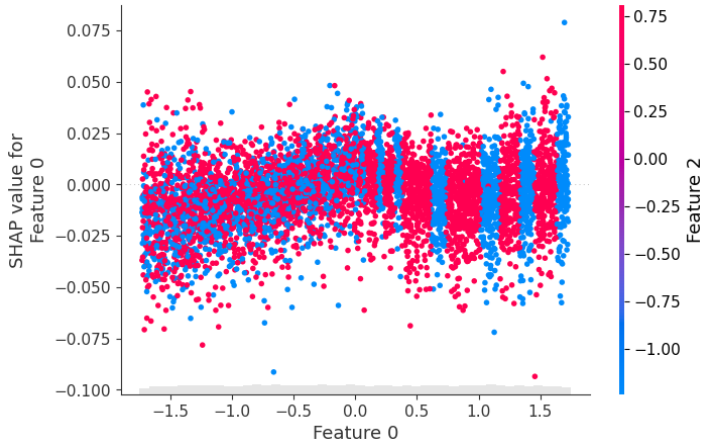
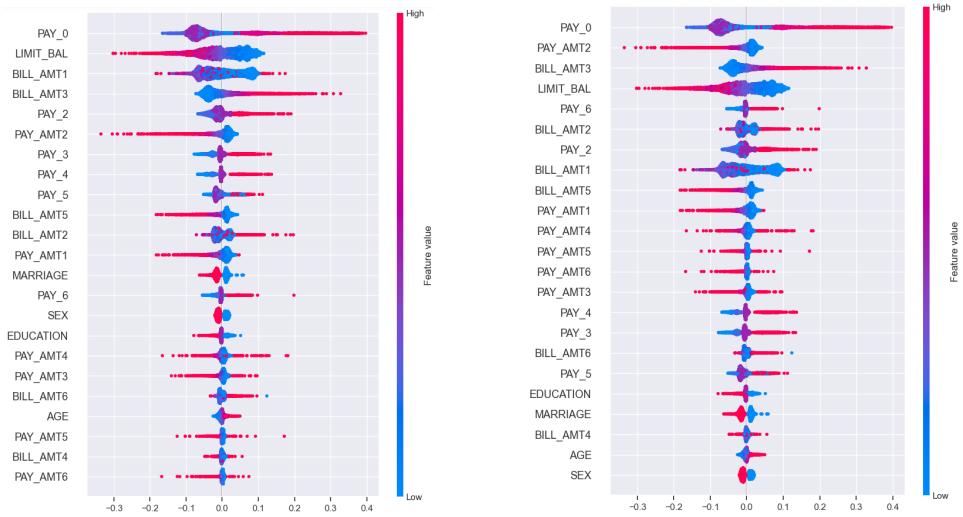


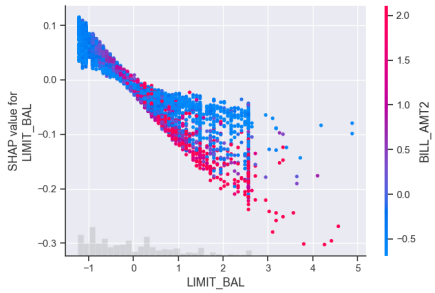
Figura 3: Gràfic "Scatter" en el que es veu com l'ID i el sexe afecten a les prediccions.



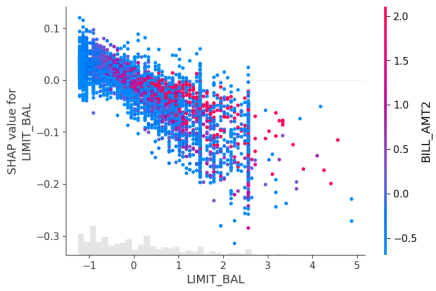
(a) Ordenat en funció de la importància mitjana dels atributs

(b) Ordenat en funció de la importància màxima absoluta dels atributs

Figura 4: Gràfics Beeswarm del model ANN, que mostren una visió global de com funciona el model al llarg de la mostra



(a) Gràfic de ”Difusió” dels dos atributs més importants del model ANN



(b) Gràfic de ”Difusió” dels dos atributs més importants del model KNN

Figura 5: Gràfics ”Scatter” o de ”Difusió”, on es pot veure la interacció clara entre aquests dos atributs i com afecten a la predició, també es pot veure clarament que l'ANN és més regular.

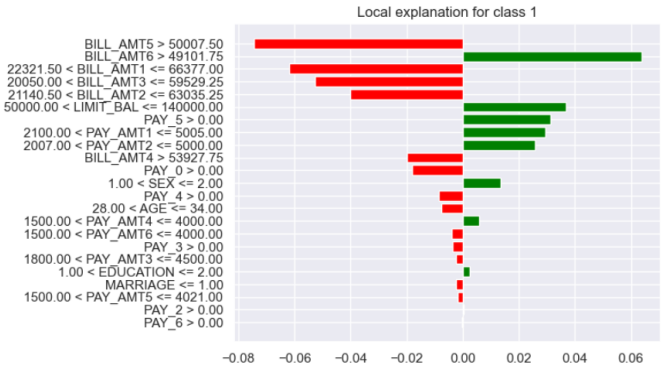


Figura 6: Gràfic de les normes i pesos de cada atribut en la observació de la pitjor predicció dels falsos positius.

	Rule	Value	Importance
0	BILL_AMT3 > 59529.25	137187	0.275008
1	BILL_AMT5 > 50007.50	136631	-0.077817
2	BILL_AMT6 > 49101.75	136080	0.075723
3	BILL_AMT2 > 63035.25	137498	0.062150
4	PAY_AMT4 > 4000.00	5029	-0.049385
5	50000.00 < LIMIT_BAL <= 140000.00	140000	0.030113
6	AGE > 42.00	60	0.021551
7	BILL_AMT4 > 53927.75	136807	-0.020767
8	BILL_AMT1 > 66377.00	137730	0.016217
9	-1.00 < PAY_0 <= 0.00	0	-0.015009
10	PAY_AMT5 > 4021.00	4792	-0.011345
11	-1.00 < PAY_5 <= 0.00	0	0.011144
12	PAY_AMT3 > 4500.00	4897	-0.010486
13	2100.00 < PAY_AMT1 <= 5005.00	4990	0.009335
14	2007.00 < PAY_AMT2 <= 5000.00	4975	0.009002
15	EDUCATION > 2.00	3	0.007080
16	-1.00 < PAY_2 <= 0.00	0	0.005072
17	1.00 < SEX <= 2.00	2	-0.004245
18	PAY_AMT6 > 4000.00	4987	0.004081
19	-1.00 < PAY_4 <= 0.00	0	0.002829
20	-1.00 < PAY_6 <= 0.00	0	-0.001442
21	MARRIAGE <= 1.00	1	0.001421
22	-1.00 < PAY_3 <= 0.00	0	0.001324

Figura 7: Taula de les normes i pesos de cada atribut en la observació de la pitjor predicció dels falsos positius.