

# Estimació del deteriorament d'existències

Roger Baiges Trilla, Pau Hidalgo Pujol, Cai Selvas Sala

25 d'abril de 2024



Col·legi  
de Censors Jurats  
de Comptes  
de Catalunya

=

**EL CØL·L3G1**

Col·legi de Censors Jurats de Comptes de Catalunya

Challenge: Intel·ligència Artificial i Auditoria

Grup GIA UPC — Document Final

---

## Resum

En aquest informe es comenten tots els detalls sobre la implementació dels models i tècniques de IA aplicats en el projecte per la detecció de deteriorament d'existències presentat en el Challenge IA i Auditoria del Col·legi de Censors Jurats de Comptes de Catalunya, el 25 d'abril de 2024.

L'objectiu del projecte ha estat proporcionar eines basades en IA que facilitin la feina als auditors a l'hora de detectar deteriorament d'existències en l'*stock* d'una empresa. A més, els models implementats s'ha buscat que siguin adaptables a noves dades, personalitzables per l'auditor i fàcils d'utilitzar sense necessitat de tenir coneixements sobre IA ni programació.

A més, en aquest projecte no només s'han aplicat models per solucionar els problemes plantejats, sinó que s'han afegit altres implementacions extra i se n'han plantejat més per un futur pensant en aportar encara més valor i facilitats a l'auditor en l'àmbit de la detecció del deteriorament de les existències.

# Índex

<b>1</b>	<b>Introducció</b>	<b>5</b>
1.1	Desafiaments i objectius . . . . .	5
1.2	Aspectes tècnics . . . . .	6
<b>2</b>	<b>Implementació</b>	<b>7</b>
2.1	Explicació bàsica i conceptual del model . . . . .	7
2.1.1	Càlcul basat en fórmula . . . . .	7
2.1.2	Càlcul basat en models avançats de sèries temporals . . . . .	7
2.1.3	Càlcul basat en xarxa neuronal . . . . .	8
2.1.4	Combinació de càlculs . . . . .	9
2.1.5	Traçabilitat i interpretació dels resultats . . . . .	9
2.2	Explicació tècnica i detallada del model . . . . .	9
2.2.1	Preparació de dades . . . . .	10
2.2.2	Implementació i justificació . . . . .	10
2.2.3	Càlcul basat en fórmula . . . . .	11
2.2.4	Càlcul basat en models avançats de sèries temporals . . . . .	11
2.2.5	Càlcul basat en xarxes neuronals . . . . .	13
2.2.6	Combinació de càlculs . . . . .	15
2.2.7	Recomanació de compra / venda de productes . . . . .	15
2.2.8	Traçabilitat i interpretació dels resultats . . . . .	15
2.3	Resultats . . . . .	16
<b>3</b>	<b>Propostes addicionals</b>	<b>17</b>
3.1	Detecció de frau . . . . .	17
3.2	Xat . . . . .	19
3.3	Classificació de productes . . . . .	20



---

3.4	Predicció en la sèrie temporal . . . . .	21
3.5	Anàlisi del producte . . . . .	21
3.6	Transfer learning . . . . .	21
3.7	Sistemes expert . . . . .	21
3.8	Visualització de les dades . . . . .	22
3.9	Aprenentatge supervisat . . . . .	23
<b>4</b>	<b>Conclusions</b>	<b>25</b>
<b>5</b>	<b>Referències</b>	<b>27</b>

## 1 Introducció

L'estimació del deteriorament és un procés que combina extracció d'informació d'un grup de dades, amb criteris d'expert per tal d'obtenir una predicció. Se'ns presentava el cas d'una empresa de comercialització de taps d'ampolla, amb informació sobre les vendes i l'*stock* dels materials amb els quals tracta, i se'ns plantejava com es podrien utilitzar eines d'IA per intentar automatitzar aquesta tasca. En les següents pàgines es planteja el nostre punt de vista del problema, així com la nostra solució a aquest (amb implementació real i a punt per ser usada) i altres propostes extra per complementar-la.

### 1.1 Desafiaments i objectius

En el procés d'auditoria, s'han d'analitzar tot un conjunt de dades i realitzar certes estimacions sobre l'evolució d'un producte per tal d'arribar a una conclusió, en aquest cas sobre el deteriorament d'existències en l'*stock* d'una empresa.

El primer pas a dur a terme, i per tant el primer objectiu, serà extreure informació i conclusions de les dades. La quantitat de dades és molta per una persona, però amb un ordinador i models d'anàlisi de dades i IA és fàcil de gestionar i interpretar. Això no obstant, per poder tractar millor amb aquestes dades, serà necessari disposar de dades estructurades i ajuntades en una sola base de dades.

El deteriorament d'existències es basa, al cap i a la fi, en la predicció de la pèrdua de valor, que sovint requereix una anàlisi minuciosa de les dades de l'empresa. Sabent això, un altre objectiu sobre l'aplicació de la IA en aquest projecte és millorar l'eficiència de tot aquest procés, però també l'eficàcia, ja que les prediccions que pot arribar a dur a terme un model prou ben entrenat poden ser millors que les d'un humà, per la capacitat de la IA per trobar i interpretar patrons altament complexos i ocults en les dades.

Un deteriorament té lloc quan el valor net realitzable, que és el valor pel qual pot vendre l'empresa un producte (menys els costos de venda...), esdevé inferior al cost. En aquest punt, s'haurà de comptabilitzar com a pèrdua. El valor net realitzable disminuirà quan un producte tingui, per exemple, problemes d'obsolescència.

Per tant, la proposta principal d'aquest projecte se centra a crear un model de intel·ligència artificial capaç d'estimar aquest valor de deteriorament (o d'un altre valor que ens permeti derivar-lo directament). Per al model, s'utilitzaran les dades de les quals disposem més alguna altra relacionada amb l'obsolescència o altres factors que creiem oportuns. A més, s'interpretaran aquests valors per tal d'indicar quines són les característiques rellevants de cada material que influeixen en aquest deteriorament.

Adicionalment, també es contempla com a objectiu proposar altres possibles aplicacions de IA, més enllà de les ja requerides per resoldre el problema del Challenge, que permetin complementar aquest model o dur a terme estimacions diferents. També es buscarà que tot sigui el més flexible possible per tal de poder adaptar-se a situacions i problemes diferents i poder ser personalitzat per un auditor segons la naturalesa de cada problema (sense necessitat de coneixements en IA ni en

programació).

## 1.2 Aspectes tècnics

Alguns comentaris del document fan referència a conceptes de programació que, per algú no familiaritzat, poden resultar estranys o no entendre's. Aquest apartat és simplement realitzar una petita definició de què signifiquen alguns d'aquests conceptes per tal de facilitar la comprensió del document.

- **Llibreria:** Una llibreria és un conjunt d'eines, funcions o classes, escrites per altra gent i públiques, que es poden utilitzar en els programes per acomplir tasques específiques sense haver de partir des de 0.
- **Funció:** Codi que es crida (s'invoca, s'executa) i permet acomplir una tasca específica. Sol rebre dades d'entrada (paràmetres) i retornar dades de sortida, o dur a terme tasques de modificació.
- **Script:** Fitxer que conté el codi
- **Model de IA:** Algorisme que, entrenat amb dades, permet resoldre un problema d'intel·ligència artificial
- **Variable:** Valor que es guarda i que pot ser canviat. Pot ser per exemple, un número, una llista, text...
- **Paràmetre:** Variable que és passada com a entrada d'una funció, que permet modificar el seu resultat o de la qual depèn.
- **DataFrame:** Taula de dades en la llibreria Pandas
- **Classe:** Una espècie de plantilla de codi, que conté una estructura de variables i funcions, i que permet crear objectes (que són parts del codi que poden fer les tasques definides a la classe i tenen propietats específiques)
- **Mètode:** Funció d'una classe (o objecte), que pots cridar i pot modificar l'objecte o retornar alguna cosa
- **Entrenament d'un model:** Proporcionar dades a un model de IA per tal que aprengui certs "pesos" o paràmetres. En aquest entrenament, el model busca minimitzar els errors
- **Python Notebook:** Eina, o document, que permet escriure text i codi Python conjuntament, i executar el codi per parts i de forma senzilla
- **Clústering:** Model utilitzat per agrupar les dades en conjunts similars
- **Outlier:** En un conjunt de dades, valor que s'allunya de la resta, que és diferent
- **Embeddings:** Representacions vectorials de paraules (o imatges...) que busquen mantenir el màxim d'informació
- **Etiqueta (IA):** Dada associada a una observació que indica la seva resposta correcta

## 2 Implementació

En aquesta secció s'hi pot trobar l'explicació tant de la idea conceptual del model creat per al Challenge com l'explicació en profunditat del model final. En la primera subsecció (2.1) no s'hi troben gaires tecnicismes ni explicacions pròpies de *Machine Learning* o Intel·ligència Artificial, però en la segona (2.2) s'expliquen tots els detalls tècnics i matisos sobre la implementació del model.

Finalment, també es comentaran els resultats obtinguts i com un auditor pot acabar de perfeccionar els paràmetres ajustables del model per tal que els càlculs siguin encara més precisos.

### 2.1 Explicació bàsica i conceptual del model

La idea és arribar a poder realitzar una estimació de la devaluació del producte. Per fer-ho, realitzarem tres càlculs de deteriorament de cada producte i després els combinarem per aconseguir aquesta predicció.

#### 2.1.1 Càlcul basat en fórmula

El primer mètode és un càlcul a partir d'una fórmula:

$$\text{Deteriorament} = -0.3 \times \frac{\text{unitats\_2023} - \text{unitats\_2022}}{\text{unitats\_2022} + 1} - 0.2 \times \frac{\text{variació\_preu\_venda\_unitari\_2022\_2023}}{\text{preu\_venda\_unitari\_2022} + 0.01} - 0.2 \times \log(1 + \text{inactivitat}) - 0.3 \times \frac{\text{unitats\_2023}}{\text{estoc\_final\_2023} + 1}$$

Aquest càlcul no involucra encara models d'aprenentatge automàtic, és més aviat una espècie de “regla”. Tot i això, incorpora uns coeficients que poden ser configurables i adaptats per un expert per tal d'assolir millors resultats.

#### 2.1.2 Càlcul basat en models avançats de sèries temporals

El segon mètode és més complex. En aquest cas, l'objectiu és crear un càlcul del deteriorament basant-se en la disminució de les vendes el pròxim any. Per obtenir aquest valor, haurem de fer una predicció, que realitzarem usant IA.

El primer pas per dur a terme aquesta estimació, però, és partir d'unes dades. Com que no disposem de dades, vam decidir generar-ne (en el model implementat concretarem més com) de mensuals. Amb aquestes, entrenarem un model per predir els 12 mesos següents.

Existeixen molts models per crear prediccions de sèries temporals. Un d'aquests models, bastant conegut, però una mica antic, és l'ARIMA (AutoRegressive Integrated Moving Average). Aquest model pot servir per realitzar prediccions, però cal tenir en compte que tan sols es basa en els valors de la sèrie temporal. Tot i això, rendeix bé és i capaç de tractar, per exemple, amb l'estacionalitat,

un aspecte bastant present quan es tracta de sèries temporals relacionades amb vendes. També volíem plantejar l'ús de models més avançats de IA. Aquests models més avançats són, ordenats de més antic a més recent, els següents:

- **LSTM:** Model basat en les xarxes neuronals recurrents (RNN), que a més implementa certa memòria. Durant molt temps va ser dels models més usats
- **NBEATS:** Model basat en un perceptró multicapa que utilitza aprenentatge profund, amb una arquitectura especialment pensada per tasques de sèries temporals
- **TimesNet:** Model de finals del 2023, basat en una xarxa convolucional
- **Transformers:** en general, els models basats en transformers poden realitzar certes tasques de predicció en sèries temporals, com per exemple el model Non-stationary Transformers.

En definitiva, hi ha moltes propostes diferents de models basats per executar aquesta tasca. Un punt a comentar dels models triats és que, a part de la sèrie temporal, són capaços de tractar també amb variables exògenes, que són factors externs que podrien afectar a la sèrie temporal, però que no es veuen afectats per ella. En les dades proporcionades, no existia cap variable d'aquest tipus, ja que totes estaven relacionades d'alguna forma amb les vendes. Tot i això, amb vista a una possible implementació en el món real, es podria intentar buscar variables de fora l'empresa que puguin afectar a aquesta sèrie temporal, per tal de millorar-ne els resultats (per exemple, la temperatura mitjana mensual si ho relacionem amb les vendes d'ampolles).

### 2.1.3 Càlcul basat en xarxa neuronal

Finalment, el tercer índex es basa en una xarxa neuronal amb una arquitectura (estructura) anomenada *auto-encoder*. Aquesta arquitectura permet obtenir representacions de cada material amb menys dimensions (característiques), però mantenint la informació significant i que caracteritza cada dada (fent més visibles els patrons ocults que contenen). Aquestes representacions s'anomenen *embeddings*.

Una vegada el model està entrenat, és capaç de transformar qualsevol dada (tant les de la base de dades com altres que s'hi afegixin) a un *embedding* (representació de menor dimensionalitat). Sabent això, s'ha creat una dada (material) fictícia de referència amb valors que indiquen un nivell molt alt de deteriorament (disminueixen molt les vendes d'un any a l'altre, hi ha massa *stock*, el cost està per sobre el preu de venda, fa molt que no es ven, etc.) i s'ha enviat a la xarxa neuronal per obtenir la seva representació (*embedding*).

Posteriorment, es calculen les similituds (mitjançant distàncies relatives que es poden calcular gràcies a la representació amb *embeddings*) de cada punt respecte al valor de referència. Amb aquestes similituds es pot determinar quins són els valors més semblants al valor de referència i, per tant, que tenen un major deteriorament.



#### 2.1.4 Combinació de càlculs

Aquests tres càlculs realitzats (fórmula, sèrie temporal i xarxa neuronal) es combinaran mitjançant una suma ponderada que per defecte dona més importància al tercer càlcul, però que l'auditor pot modificar al seu gust.

Per realitzar la predicció usant aquest valor, es resta un llinar (per defecte la mitjana) i es divideix per una "tolerància" (tots dos són paràmetres que poden ser personalitzats per l'auditor), tal com es pot veure a continuació:

$$\text{fair\_price} = \text{preu\_venda\_unitari\_2023} - \text{preu\_venda\_unitari\_2023} \times \left( \frac{\text{sum\_indexes} - \text{threshold}}{\text{tolerance}} \right)$$

El llinar (*threshold*) és el que defineix si el canvi de valor serà positiu o negatiu, mentre que la tolerància (*tolerance*) modifica en quina quantitat es modifica el valor.

La predicció acaba sent sumar al preu unitari del 2023 el percentatge corresponent d'aquest mateix valor.

Amb aquest preu just estimat, simplement es tracta de mirar si està per sota el cost unitari de l'*stock*, i en cas que ho sigui devaluar el producte. Com es pot observar, hi ha certs paràmetres amb els quals l'auditor pot treballar per poder obtenir un valor de depreciació acceptable per l'empresa. Basant-se en aquestes prediccions pot, per exemple, considerar els que el model prediu que es deteriora més i analitzar el perquè i extreure les seves conclusions.

#### 2.1.5 Traçabilitat i interpretació dels resultats

La nostra proposta, però, no acaba aquí, i inclou l'ús d'un altre model de IA. Després de realitzar totes aquestes combinacions d'índexs i prediccions, és molt complicat saber realment quines dades de les originals són les que estan afectant aquesta predicció, fet que pot portar confusió a l'auditor i que aquest sistema no li sigui útil. Per això, i com que entenem que la interpretabilitat i traçabilitat dels resultats és molt important en aquest àmbit, proposem usar un Explainable Boosting Machine. Aquest és un model de la branca anomenada Explainable AI, que busca ser el màxim interpretable. Aplicant aquest model amb les nostres prediccions com a output i les dades originals com a sortida, podem analitzar com afecta cada variable al model en general, i a més, a cada material en particular.

Aquest fet és molt positiu, ja que permet que l'auditor vegi quins valors ha de mirar per aquell material.

## 2.2 Explicació tècnica i detallada del model

En aquesta secció de l'informe es tractarà com s'ha creat tècnicament el model d'IA alhora de com utilitzar-lo mitjançant l'enllaç al *GitHub* [1].

### 2.2.1 Preparació de dades

El primer pas en la implementació del model, que és un pas no plantejat en el conceptual, és la codificació i unificació de les diferents pàgines de la base de dades en una única amb la qual puguem treballar. Partim del fitxer `inventory_data_raw`, eliminem les columnes que en aquest cas no ens interessin i realitzem una primera neteja creant `inventory_data_clean`. Després, ajuntem aquestes diferents pàgines en una, guardat a `inventory_data_merged`. Per dur a terme aquests processos, fem servir un scrip the Python i la llibreria pandas ([2], [3]).

A partir d'aquest punt, ja creem algunes variables noves: dies des de l'última entrada, dies de l'última sortida, diferència entre entrada i sortida, preu de venda unitari 2022 (divisió entre vendes i unitats), variació (en proporció) del preu de venda i la variació (també en proporció) de les vendes i les unitats. Aquestes variables, tot i ser relativament simples, ens permeten tenir informació que s'adequa més al problema, ja que permeten estimar més o menys si un producte tarda molt a vendre's (o si fa molt que no es ven), i quantificar com s'ha modificat el seu preu unitari (i no només les vendes). Aquests càlculs, tot i ser simples, permeten veure directament com canvien les dades d'un any per l'altre. Especialment, les proporcions són interessants perquè una proporció no depèn de l'escala del producte, i per tant permet contextualitzar en un mateix rang de valors els productes amb un nombre elevat d'unitats vengudes amb els d'un nombre més baix.

Aquest procés està implementat en el fitxer `create_new_dataset`, i utilitza un DataFrame de pandas per tal de poder accedir als valors de forma més senzilla.

Aquesta serà la base de dades de la qual partirem per començar a aplicar els diferents models d'IA.

### 2.2.2 Implementació i justificació

La nostra implementació està unificada en una sola classe per facilitar l'execució. L'objectiu és que es pugui crear el model, entrenar usant el mètode `fit`, i realitzar les prediccions amb el predict que és una estructura bastant comuna en les implementacions de models de IA. Aquesta classe està implementada al fitxer `inventory_impairment_class`, com a `InventoryImpairment`.

Per tal d'executar el model que defineix aquesta classe, i que explicarem a continuació, existeix el notebook `run_model`. Un notebook és un tipus de fitxer que permet separar el codi per cel·les, així com intercalar-lo amb text, per tal de crear un espai d'execució més senzill i interactiu. El primer pas serà llegir les dades que s'han creat en l'apartat anterior. Després, es tracta tan sols de crear l'objecte de la classe (una instància del model) i executar el mètode `.fit`.

Aquest mètode (funció que realitza uns càlculs), necessita que li passis les dades (que hem llegit abans). Addicionalment, i per tal que el codi fos el més flexible possible, se li poden passar de forma opcional diversos paràmetres, com la variabilitat amb la qual vols que creï les dades mensuals (en parlarem més endavant) o els pesos de cada mes, així com els coeficients del càlcul basat en la fórmula o els noms de les teves variables en cas d'estar utilitzant una base de dades diferents. El codi està degudament documentat per poder entendre fàcilment què significa cada un d'aquests paràmetres.

El primer que fa és quedar-se tan sols amb els materials que apareixen a *stock*. El problema que

volem solucionar té a veure amb el deteriorament d'aquests materials, i per tant té sentit quedar-se tan sols amb aquests. A continuació, genera les dades mensuals. Per generar-les, itera per cada un dels materials. D'entrada comprova quin és el mes de l'última sortida (l'últim mes en el qual es ven) per tal d'evitar crear dades errònies.

### 2.2.3 Càlcul basat en fórmula

El primer càlcul que realitza és el que està basat en una fórmula, que hem creat basant-nos en el que creiem que era important i intentant imitar el que faria un auditor sense eines d'intel·ligència artificial. Parteix de 4 variables: delta units (resta del nombre d'unitats vengudes els dos anys entre el nombre d'unitats vengudes el primer, més 1 per evitar les divisions entre 0), delta unitary sell price (preu de venda el 2023 entre el del 2022, inventory rotation (nombre d'unitats vengudes el 2023 entre la quantitat de *stock*) i inactivity (logaritme de la suma dels dies d'última entrada més els d'última sortida, per mesurar quants dies fa que el producte no pateix cap canvi, i el logaritme permet regular valors molt alts). Cada un d'aquests elements té una ponderació, que pot ser ajustada per un auditori (en el cas implementat és -0.3, -0.2, 0.2 i -0.3) per tal de donar més valor o menys a cada un dels elements. Aquest càlcul, s'escala entre 0 i 1 i s'obté el primer dels tres valors que usarà en la predicció.

Finalment obtenim que la fórmula és la següent:

$$\text{Deteriorament} = -0.3 \times \frac{\text{unitats\_2023} - \text{unitats\_2022}}{\text{unitats\_2022} + 1} - 0.2 \times \frac{\text{variació\_preu\_venda\_unitari\_2022\_2023}}{\text{preu\_venda\_unitari\_2022} + 0.01} - 0.2 \times \log(1 + \text{inactivitat}) - 0.3 \times \frac{\text{unitats\_2023}}{\text{estoc\_final\_2023} + 1}$$

### 2.2.4 Càlcul basat en models avançats de sèries temporals

A continuació i, per tal de poder entrenar aquests models avançats de sèries temporals, cal generar les dades mensuals pels mesos que corresponen. Per dur a terme aquesta generació, hi ha dues funcions: `generate_monthly_sales` i `calculate_monthly_sales`.

La primera crea les dades, d'entrada, basant-se en la mitjana de l'any (o en l'evolució entre l'any anterior i l'actual, en el cas del segon any) i soroll normal. Aquest soroll és el que es controla amb la variable variabilitat mencionada abans (com més alta, més variació hi haurà entre els mesos). Seguidament, aplica els pesos mensuals també definits abans. En el problema d'exemple particular, els valors escollits han estat 0.5 de variabilitat i uns pesos de [0.8, 1, 1, 1, 1, 1.2, 1.3, 1.1, 1, 1, 1, 1] (un lleuger increment de vendes durant l'estiu, ja que considerem que la venda de taps va molt lligada a la d'ampolles de plàstic, però es pot adaptar a la informació que es conegui). En la figura 1 es pot veure un exemple d'aquestes dades mensuals generades. Un cop es creen aquestes dades amb soroll, es crida la segona funció per tal d'assegurar-se que la suma de vendes dels mesos sigui igual a l'anual real.

Realitzat aquest procés, el model disposa d'unes dades mensuals, que pot utilitzar per realitzar prediccions sobre les vendes del següent any. Per fer-ho, fa servir el mètode `create_auto_arima_and_forecast`, que simplement rep aquestes dades. Si anteriorment, a l'hora de crear-lo o usant un mètode es-

pecífic, li hem definit al model un fitxer amb prediccions, l'utilitza. En cas contrari, entrena un *autoARIMA* per cada un dels materials, i realitza les prediccions dels 12 mesos següents. La implementació d'aquest *autoARIMA* emprada és la de la llibreria *pmdarima*.

Si, en lloc de l'*autoARIMA*, es vol utilitzar un model més avançat de predicció de sèries temporals, cal dirigir-se al fitxer *neural\_forecast*. Aquest conté un altre notebook que llegeix les dades, en fa una petita transformació necessària per als models que fa servir, i seguidament realitza l'entrenament i les prediccions utilitzant tres models diferents: *NBEATS* ([4]), *TimesNet* i *FEDformer* ([5]). Aquests, estan implementats a la llibreria de python *neuralforecast* ([6]), i són models molt més complexos. Han estat escollits perquè tots tres fan ús de tipus de xarxes neuronals diferents, i perquè en principi obtenen molt bons resultats. De fet, *TimesNet* és el model més complex en l'apartat de Short-term Forecasting ([7]), i *FEDformer* el tercer. De forma superficial, *NBEATS* emprà l'arquitectura de perceptró multicapa, *TimesNet* una xarxa convolucional i *FEDformer* un transformer.

Aquestes prediccions es fan servir en el segon càlcul (índex en el codi), el de sèries temporals. També s'usen en un mètode extra, *stock management*, per tal de realitzar algunes recomanacions de com hauria l'empresa de tractar la quantitat de *stock* que té. Si les previsions de vendes són altes, recomana que en compri més, per evitar quedar-se sense *stock*. En cas contrari, recomana desfer-se d'aquest *stock* si n'hi ha massa. Això tot i no ser un objectiu principal del repte considerem que pot aportar un gran valor a una empresa.

El segon càlcul efectua una operació molt simple: suma les prediccions dels 12 mesos de l'any següent, i calcula el percentatge en el qual es preveu que disminueixin les vendes. Si es preveu que disminueixin molt, s'interpreta que és un indicador que s'hauria de devaluar aquell producte. En certa manera, és una certa predicció de la demanada que tindrà aquell determinat material, i evidentment sense demanda no hi haurà valor. Aquests valors acaben sent normalitzats entre 0 i 1 per tal d'evitar valors extrems, i poder combinar els càlculs després més fàcilment. Podem concloure que el càlcul del model es realitza mitjançant aquesta operació matemàtica:

$$\text{Deteriorament} = \begin{cases} 100 & \text{si predicció} \leq 0 \\ - \left( \frac{\text{venda actual} - \text{venda predita}}{\text{venda actual}} \right) \times 100 & \text{si venda actual} \neq 0 \end{cases}$$

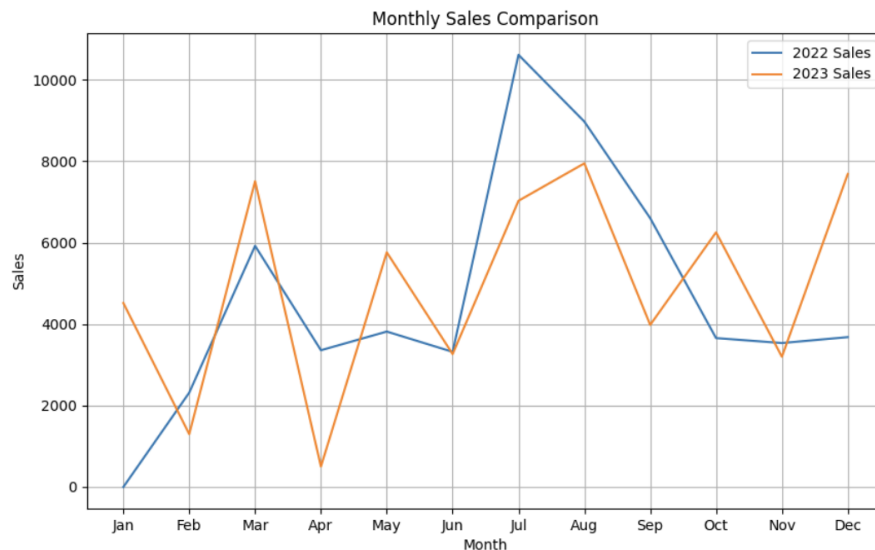


Figura 1: Exemple de dades mensuals generades artificialment

### 2.2.5 Càlcul basat en xarxes neuronals

Finalment, el tercer índex es basa en una xarxa neuronal amb una arquitectura (estructura) anomenada *auto-encoder*. Aquesta arquitectura el que fa és entrenar una xarxa neuronal per tal de predir el mateix que rep, però col·locant enmig de la xarxa una capa amb una dimensionalitat baixa (com un "obstacle" pel model), tal com es pot veure en la figura 2. Això provoca que no es puguin reconstruir les dades amb 100% d'exactitud, i a més permet obtenir una representació amb menys dimensions (mantenint només la informació significant i que caracteritza cada dada) de les dades originals. Aquesta representació capta certs patrons ocults a les dades i és la que utilitzarà aquest càlcul. El model serà entrenat amb totes les dades, i una vegada hagi minimitzat el seu error reconstruint (decodificant) les dades, s'eliminarà de la xarxa neuronal la secció dedicada a la decodificació i es mantindrà només la part encarregada de la codificació per aconseguir el punt on tota la informació d'entrada es troba molt reduïda.

El model rep variables de proporcions, per tal que l'escala dels productes no influeixi tant. Aquestes proporcions són: la variació en el preu unitari, la variació en les unitats venudes, la variació en les vendes, la diferència entre l'última entrada i sortida, els dies des de l'última sortida, la proporció de vendes respecte l'*stock*, la ràtio cost-valor i l'índex de les sèries temporals. Aquestes són les variables escollides, en part perquè tampoc disposàvem de més dades, però si tinguéssim més informació sobre els productes (fins i tot les seves característiques, ja que poden influir en el seu comportament de vendes), es podrien també incorporar. A continuació, s'entrena el model ajustant els pesos de la xarxa neuronal per tal d'obtenir la màxima informació possible en molt poques dimensions.

Per definir aquesta xarxa neuronal, s'ha utilitzat la llibreria Keras de Tensorflow. S'ha creat una xarxa consistent en dues capes d'encoder (de codificació), amb 128 i 64 neurones respectivament, la capa intermèdia de mida 4, i dues de decoder també amb 64 i 128 (amb dimensió de sortida igual que

la d'entrada). Les funcions d'activació usades han estat la ReLu, comunment utilitzada, trobant així patrons no lineals. L'optimitzador escollit ha estat ADAM (també un dels més utilitzats), i l'error el mean squared error (ja que totes les variables eren numèriques i, per tant, era un problema de regressió).

Aquest model s'entrena intentant reconstruir les dades, fent servir també una partició de validació per certificar els resultats. El *MSE* de validació assolit ha estat de 0.3; aquest valor significa que en promig per a cada variable a predir el model s'ha equivocat de  $\sqrt{0.3}$  indicant així que la major part de la informació ha estat correctament captada en les baixes dimensions de l'encoder.

A continuació, i per tal de poder crear l'última mesura de deteriorament, es crea un material fictici on totes les proporcions de variació són del -100% (perd totes les vendes i valor), hi ha 730 dies entre l'entrada i la sortida, fa 365 que no se'n ven cap, la proporció respecte a l'*stock* i al cost és de 0.01 i el càlcul usant el segon model és d'1. Aquest, s'usarà de referència. Es passen tots els valors, inclòs aquest, per la meitat de l'autoencoder (tan sols l'encoder) per obtenir uns valors en la dimensió latent (de mida 4 en aquest cas). A continuació, es realitzen les similituds de tots els materials amb la referència negativa: com més a prop, més probable és que els calgui devaluar. Aquestes similituds poden ser calculades fent ús de tant la distància euclidiana com la similitud de cosinus (recomanem utilitzar la distància euclidiana, ja que la dimensionalitat en aquesta dimensionalitat no gaire alta aconsegueix millors resultats). Amb aquestes distàncies escalades altre cop entre 0 i 1, es crea el tercer índex.

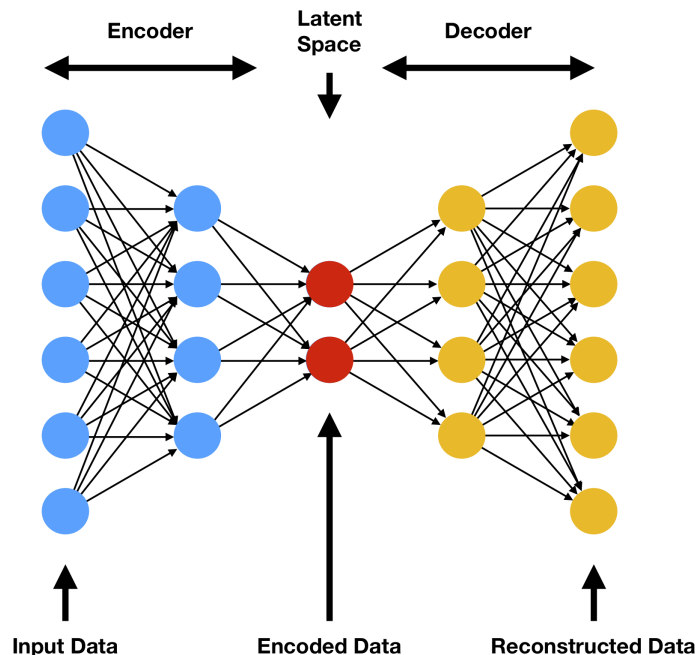


Figura 2: Arquitectura de xarxa neuronal de tipus *auto-encoder*

### 2.2.6 Combinació de càlculs

Arribats a aquest punt, es considera que el model està entrenat ja ha realitzat els tres càlculs, i el pas que falta realitzar és fer la predicció. Aquesta es realitza utilitzant el mètode *predict* de la classe.

Rep com a paràmetres uns pesos per cada índex (per tal que l'expert pugui decidir a quin donar-li més valor), un *threshold*, que és a partir de quin valor de la suma dels índexs es considera positiu o negatiu, i una tolerància, que afecta al percentatge de devaluació. La predicció és molt senzilla: simplement es realitza la suma ponderada, es resta el *threshold* (llindar) i es divideix per la tolerància, veure en la fórmula 2.2.6. Llavors, s'interpreta que aquesta és la proporció de deteriorament del valor respecte al del 2023, i es modifica. Amb això, obtenim les prediccions del model. Aquest valor llindar, es pot optar per calcular-lo automàticament usant, per exemple, la mitjana dels valors.

$$\text{fair\_price} = \text{preu\_venda\_unitari\_2023} - \text{preu\_venda\_unitari\_2023} \times \left( \frac{\text{sum\_indexes} - \text{threshold}}{1.5} \right)$$

### 2.2.7 Recomanació de compra / venda de productes

El mètode que utilitza la predicció per tal de realitzar una petita recomanació sobre com es pot tractar l'*stock* actual funciona de la següent manera: si la predicció diu que en el pròxim any no s'aconseguirà vendre l'*stock*, et recomana que el redueixis; si diu que en els pròxims 4 mesos et quedaràs sense, et diu que en compris més. Com es pot observar, pot semblar senzill, però permet solucionar un problema molt important per a l'empresa que pugui estar utilitzant el nostre model.

### 2.2.8 Traçabilitat i interpretació dels resultats

Com es àmpliament sabut en el món de l'economia i empresa, qualsevol model que utilitzi algorismes de *machine learning* o IA per tal de predir certes característiques sobre un producte cal poder justificar en tot moment com el model o algorisme ha arribat a la predicció.

Com que al fer servir models de *deep learning* en els nostres càlculs fa que això sigui impossible degut a la gran dimensió d'aquesta xarxa i degut a la immensa quantitat de connexions entre les neurones. Per tal de poder solucionar aquest problema vam idear entrenar un altre model que tingui una capacitat molt alta de predicció aproximant-se als models més avançats de *deep learning* però mantenint l'explicabilitat de models més senzills aquest és l'*EBM*.

Finalment, el mètode *explain*, entrena un Explainable Boosting Machine (EBM de la llibreria *interpret*, [8] [9]) amb les dades inicials, intentant predir el percentatge de deteriorament que ha calculat el model. Aquest, és capaç d'adaptar-se gairebé perfectament a les prediccions, un MSE molt baix d'aproximadament 0.05. L'avantatge que té aquest mètode és que és explicable, és a dir: és capaç de justificar per què realitza les prediccions que fa, i quines són les variables que

influeixen. D'aquesta manera, podem mostrar la seva explicació de variables global i retorna aquest model entrenat per si es vol obtenir també l'explicació de variables local per cada un dels elements (o usar-lo amb materials nous, tot i que caldrà que continguin les mateixes dades). Aquest mateix model, l'EBM, en cas de disposar de prediccions del deteriorament realitzades per un comptable o auditor, també podria realitzar el mateix procediment: intentar aprendre quins són els patrons que aplica. D'aquesta manera, podem argumentar quines variables són les que tenen més importància per aquella persona a la hora d'estimar els valors. Aquest fet, tot i semblar una mica trivial, és molt útil, ja que aporta traçabilitat i interpretabilitat dels resultats, que són crucials en l'àmbit de l'auditoria on totes les decisions han de ser justificables.

## 2.3 Resultats

Els resultats del valor just varien bastant en funció de la tolerància usada, i de les prediccions usades, però en general els índexs tenen bastant sentit. Analitzant les dades a simple vista, les dades com l'element 115030252, que tenen una disminució de les vendes (tant en unitats com en preu de venda unitari), bastanta diferència entre l'entrada i la sortida i un valor unitari el 2023 a prop del cost, són els que s'acaben devaluant (ja que recordem que per que hi hagi devaluació aquest valor "just" ha de caure per sota el cost).

És important mencionar que, en el nostre cas, no disposem de un rerefons econòmic, i que per tant no ha sigut possible acabar d'ajustar el model per tal que les prediccions puguin ser usades directament. Tot i això, serveix bastant bé com a indicador dels productes que en principi es devaluaràn, i creiem que pot ser molt útil en el context d'auditoria. A més, molts dels paràmetres són configurables i ajustables per tal que un auditor doni més importància a les característiques que considera oportunes.

Analitzant l'EBM, la variable que influeix més en el preu just és, com era d'esperar, el preu de venda unitari el 2023, ja que al cap i a la fi el càlcul que fem és una modificació d'aquest. El segon que influeix més són les vendes el 2023, i el tercer els dies des de l'última entrada (últim cop que l'empresa en va comprar). Cal mencionar, però, que tot això depèn de quins paràmetres s'hagin utilitzat, o de quins models per fer la predicció.

Podem analitzar també com evolucionen els "scores" del model en funció dels nivells d'aquestes variables. Com més dies faci de l'última entrada, més es devalua. Si el 2023 té 0 vendes, aquesta puntuació també és molt negativa. També realitza les puntuacions de les interaccions: per exemple, entre el preu de venda unitari el 2023 i el cost unitari, si el cost és petit i el preu alt, l'score és positiu, mentre que si passa al revés, és negatiu.



### 3 Propostes addicionals

La implementació realitzada, com s'ha comentat, és capaç de realitzar una predicció d'aquest deteriorament, i a més interpretar quins són els elements que més afecten a aquest. Tot i això, hi ha altres propostes d'usos de models de IA, que considerem prou interessants com per mencionar però que no hem implementat. Aquestes permetran aprofundir més en altres models de intel·ligència artificial i en els diferents apartats del procés en el que es poden aplicar. L'objectiu del Challenge era permetre, a través d'un anàlisi de les dades de l'empresa, proporcionar informació útil per tal de calcular aquest deteriorament. Si bé es pot intentar calcular aquest valor (el que realitza la nostra implementació), també es poden usar tècniques per obtenir informació sobre altres aspectes de les existències. En definitiva, aquest apartat és un recopilatori d'altres estructures de models que també podrien aportar valor en el problema.

A continuació, s'expliquen dues d'aquestes propostes addicionals que hem implementat, per tal de poder-se fer a la idea de com podrien servir. Posteriorment, s'expliquen altres propostes que no s'han afegit a aquest projecte però també podrien aportar-hi valor.

#### 3.1 Detecció de frau

Un dels reptes més comuns en la auditoria és la detecció del frau. Una idea d'implementació, en aquest cas, dependria de dades etiquetades com a fraudulent. En aquestes, es podria aplicar un algorisme de classificació, com podria ser un SVM (support vector machine), i entrenar un model. Amb aquest model entrenat, es podrien a continuació classificar les noves entrades com a possibles o no frauds, i aleshores un auditor s'encarregaria d'analitzar-les. Mencionar que, en desenvolupar un model d'aquest estil, convindria intentar minimitzar el nombre de falsos negatius, encara que els existeixin més falsos positius. Per exemple, en [10] s'utilitza un SVM per detectar frau en targetes de crèdit.

Una altra proposta alternativa per aquest apartat seria usar algorismes de detecció d'outliers, que és el que hem implementat. Un model que es pot usar per aquest aspecte és un Isolation Forest, que utilitza arbres binaris per detectar outliers ([11]). Aquests es poden observar a la figura 3.

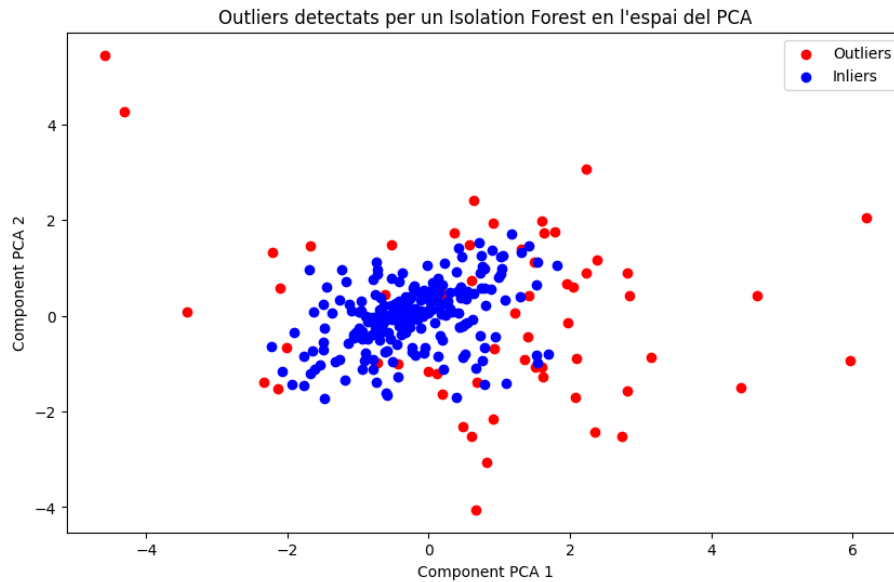


Figura 3: Detecció d'outliers en els taps d'ampolla usant Isolation Forest, visualitzat en el PCA

Aquest valors estranys, però, no sempre seran negatius: a vegades pot ser que siguin positius. Per aquest motiu, en el codi després d'executar el mode, utilitzem una tècnica de clústering (KNN) per tal de separar aquests outliers en positius i negatius (figura 4). D'aquesta manera podriem obtenir una etiqueta per cada producte, i probablement seria útil per a un auditor. Aquestes dades de frau poden anar bastant relacionades amb la deterioració, ja que és probable que una empresa vulgui amagar que té pèrdues pel deteriorament del valor d'existències.

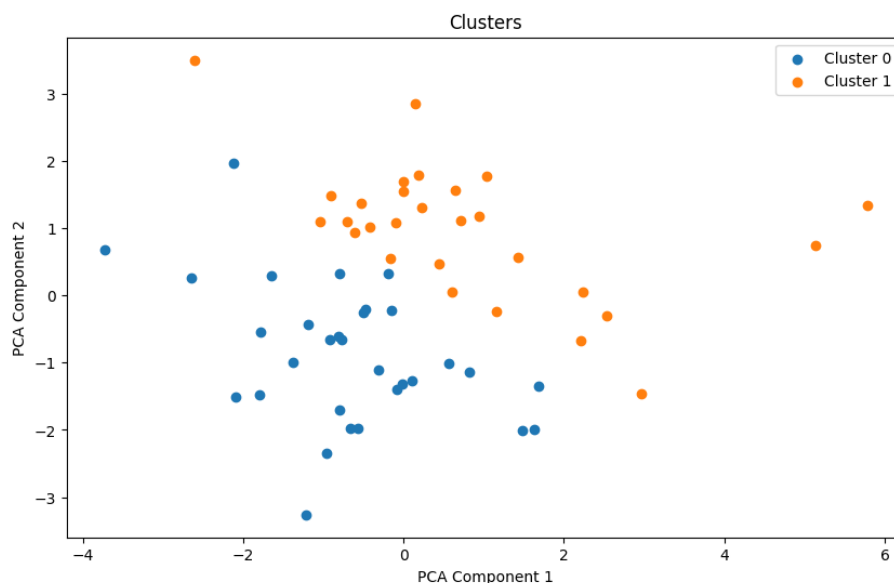


Figura 4: Classificació dels outliers en dos grups, visualitzat en el PCA

## 3.2 Xat

En els últims temps, els models de llenguatge han guanyat molta popularitat i s'han convertit en una gran eina d'intel·ligència artificial. Aquests, també poden ser aplicats per tal d'obtenir més informació sobre les dades, o per facilitar-ne el tractament i la interacció.

La proposta implementada és un xat que permet, un cop s'ha executat el model anterior, aconseguir informació determinada sobre els productes de la base dades (l'Excel). Implementa el que s'anomena retrieval-augmented generation (RAG). Aquesta tècnica permet expandir els usos dels LLMs (models de llenguatge, Large Language Models en anglès) permetent que tinguin accés a certa informació (en aquest cas, la base de dades).

La forma en la qual ha estat implementat és molt bàsica, en part perquè els models grans de llenguatge necessiten molta capacitat de càlcul (i, per tant, s'ha usat una alternativa que permet realitzar una cosa similar però de manera més eficient). Com funciona és que nosaltres hem predefinit un conjunt d'operacions possibles sobre les dades: aconseguir tots els valors d'un material, obtenir només l'índex de l'auto encoder, obtenir les prediccions del model, obtenir la devaluació determinada pel model o obtenir la devaluació total de la base de dades. Aquestes, en una implementació real, es podrien expandir per tal de permetre moltes més consultes diferents, a diversos elements alhora...

Llavors, associa una instrucció de llenguatge natural (un exemple de frase que diries si vols assolir allò) a cada una de les funcions. En aquest punt és on intervenen els models de llenguatge. Utilitza un model molt simple que serveix tan sols per crear embeddings (és a dir, no genera text, només el codifica de forma similar al que feia l'auto-encoder amb les nostres dades) i crea aquestes repre-

sentacions per totes les frases de referència. Llavors, cada cop que l'usuari introdueix el que vol, es comparen aquests índexs amb els de les preguntes i s'executa la funció relacionada.

Es pot observar un exemple de sessió amb aquest xat a la figura 5.

```
You: hola que tal
ChatBot: Digues quina informació de la base de dades vols. Per exemple, dona'm el percentatge de devaluació del material 100022432
You: vull el valor just estimat pel material 135060910
ChatBot: El valor just estimat pel model és 0.1301862182852903
You: i el valor total de devaluació de la base?
ChatBot: Valor total devaluat: 261980.13071124017, en un total de 49 elements
You: quin index prediu l'autoencoder en 135062142?
ChatBot: El càlcul usant l'auto encoder (xarxa neuronal) és de 0.1088326592287745
```

Figura 5: Exemple de sessió amb el xat implementat

Cal mencionar que aquest mini xat, evidentment, és molt senzill: no té memòria, les respostes que realitza són predeterminades (no genera text, ja que és una tasca molt complicada i complexa) i necessita que quan facis la petició li diguis també el número del material que vols. Tot i això, una versió millorada d'aquest xat no és gaire complicada de fer seguint la mateixa estructura i utilitzant un model de llenguatge gran (LLM esmentats abans). En lloc de retornar la resposta directament, per exemple, se li pot enviar al LLM dient-li que la reescrigui, i d'aquesta forma no serien respostes predeterminades. A més, un LLM té capacitat de memòria i detectaria els identificadors millor (i el que vols, ja que l'actual és fàcil que falli si la frase és un pèl ambigua).

És important tenir en compte que, si bé aquest tipus de models obtenen resultats molt vistosos, cal sempre tenir en compte que poden patir al·lucinacions i generar informació no fiable (en aquest cas, respondre a una pregunta que no toca, que és fàcil de veure, però en LLMs et poden tornar una resposta que sembli perfectament bona, però no ho sigui). A més, a causa de la seva mida, els LLM són costosos d'implementar i de mantenir. Tot i això, gràcies a la seva naturalesa es poden aplicar també a dades no estructurades, i permeten analitzar per exemple contractes escrits, correus electrònics... (tot i que evidentment cal tenir en compte la protecció de dades).

Les altres propostes ja no han estat implementades, tot i que gràcies als nostres coneixements en l'àmbit sabem que són possibles i viables per aplicar-les en aquest problema.

### 3.3 Classificació de productes

Classificar els productes en funció de les seves característiques pot ser molt útil per, posteriorment, establir un deteriorament acceptable. Agrupar els productes en diferents grups i observar-ne els comportaments pot permetre obtenir grups on hi hauria d'haver deteriorament i altres on no, i aleshores el procés és tan sols observar a quin grup cau cada material.

Una altra proposta de classificació, que a més es relaciona també amb sèries temporals, seria realitzar un clústering de time series. No tots els productes tindran la mateixa vida ni es comportaran de la mateixa forma, i d'aquesta manera els podrem classificar en diferents grups. Per tal de crear aquests clústers, dependrem de dades de materials (o altres actius, en cas que vulguem aplicar-ho a una empresa productora) al llarg de tota la seva vida útil, que classifiquem en grups. Un cop aconseguits els clústers, podem intentar, per cada un dels actius actuals, predir a quin clúster aniria.

Alguns d'aquests clústers tindran tendència a devaluar-se ràpid, i per tant si al producte li toca anar en aquell, tocarà devaluar-lo. En canvi, d'altres igual es mantenen més anys.

### 3.4 Predicció en la sèrie temporal

Es poden realitzar millors prediccions de vendes, per exemple, obtenint informació de més factors externs que poden afectar el preu com a variables exògenes. Per exemple, un model pot tenir informació sobre l'evolució de vendes de la competència, del mercat en general, anàlisi textual de notícies o twitts a internet...

Aconseguir aquesta informació, depèn de com, pot requerir també utilitzar al seu torn altres models, com per exemple models de reconeixement d'imatge o de processament del llenguatge natural.

### 3.5 Anàlisi del producte

Un producte no només es pot devaluar en funció de la demanda: també pot fer-ho per les condicions en les quals es troba, per exemple. Es poden usar algorismes de IA, combinats amb sistemes de IoT (Internet of Things) per tal d'obtenir dades sobre l'estat de conservació d'aquests. Un exemple, contextualitzat en les dades de les quals disposem seria monitorar la humitat/temperatura per taps biodegradables. A més, tècniques de visió per computador poden servir per detectar desperfectes en els taps.

### 3.6 Transfer learning

Una de les propostes, fruit d'una idea mentre estàvem realitzant la implementació, seria utilitzar alguna base de dades completa i amb moltes dades, per tal d'entrenar un model, per després acabar de realitzar fine-tuning d'aquest amb les nostres dades. Aquest fet permet que, amb poques dades, puguem ser capaç d'assolir resultats molt bons. Aquest concepte es coneix en el camp de la IA com a transfer learning, i té l'avantatge de permetre implementar models de manera molt fàcil en diverses amb dades similars, però diferents, i adaptar-les millor a cada cas.

### 3.7 Sistemes expert

Els sistemes experts són una branca de la IA més antiga, que no té a veure amb l'aprenentatge automàtic sinó que es basa a intentar emular el comportament d'un expert en un domini mitjançant l'ús de regles. Evidentment, això limita les seves capacitats, i mai seran més bons que l'expert que ha proporcionat aquestes regles, però en alguns casos poden ser prou útils.

En aquest projecte, es podria implementar un sistema de regles en Prolog, per exemple, basat en valors de certesa o probabilitats, que defineixin en funció de característiques que puguem tenir un element si seria possible devaluar-lo o no. Per exemple, si fa més d'un any que no es ven, és molt més probable que si fa un dia.

Sistemes d'aquest tipus s'utilitzen en l'àmbit de l'empresa, amb sistemes que implementen de forma automàtica alguna forma de *Business Logic* ([12]). Tot i això, els sistemes experts tenen moltes dificultats en situacions ambigües, necessiten molt de coneixement sobre el domini (que s'ha de programar), no són adaptables a noves situacions... Una versió més moderna d'aquest tipus de sistemes són els sistemes intel·ligents, que incorporen als sistemes experts certes habilitats d'aprenentatge automàtic per tal que puguin inferir certes normes automàticament, i analitzar grans quantitats de dades. Tot i això, considerem que altres mètodes d'aprenentatge automàtic pur són millors.

### 3.8 Visualització de les dades

Tot i no estar relacionat directament amb la intel·ligència artificial, es pot dur a terme (no és ben bé una proposta de model, sinó una espècie de menció del que es pot fer) utilitzant anàlisi de dades i programació, una visualització més avançada de les dades. L'Excel mateix disposa de moltes eines útils per realitzar diferents plots i gràfics, i aquest permeten observar relacions en les dades de forma molt més visual i en molts casos entenedora. Algunes tècniques més avançades, com el PCA, poden servir per visualitzar les dades en dues dimensions que representin el màxim d'informació. A més, amb aquest tipus d'anàlisi, es poden veure les variables més relacionades (direccions similars) o possibles outliers d'una forma bastant senzilla.

En definitiva, aquest apartat és per comentar que, tot i que en el model proposat no hi ha gaire espai d'interpretació de les dades inicials i visualització, ja que ens volíem centrar en un model d'intel·ligència artificial com a tal, és possible crear figures i plots que mostrin alguna informació de les dades. Un exemple, seria la matriu de correlacions que es pot veure a la figura 6:

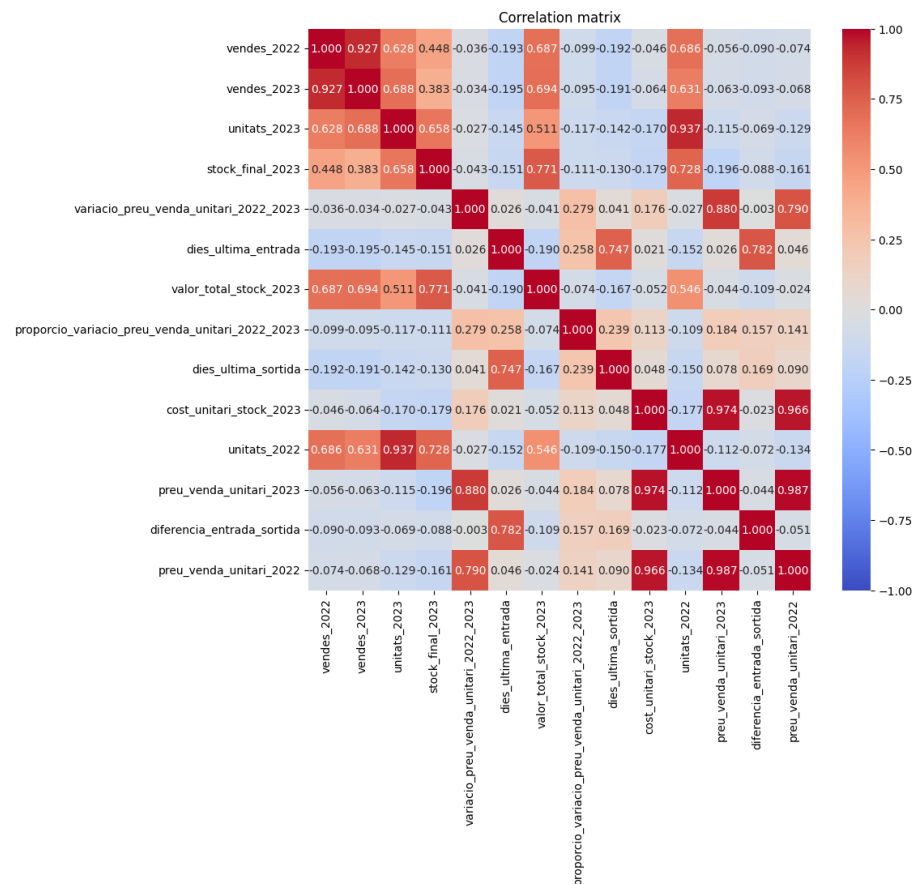


Figura 6: Matriu de correlacions de les variables

### 3.9 Aprenentatge supervisat

En general, la majoria d'opcions plantejades, així com el model, formen part de la branca de l'aprenentatge automàtic anomenada aprenentatge no supervisat. Aquest tipus de IA es realitza en dades no etiquetades (com eren les que se'ns proporcionaven), i busca aprendre patrons de forma automàtica sense tenir referència (com el que feia l'auto-encoder). Un factor positiu d'aquest tipus de sistemes és que no necessiten que prèviament un expert hagi etiquetat les dades, i per tant són més fàcils de generalitzar i d'aplicar en molts casos. Tot i això, cal mencionar que, en cas de tenir un expert amb el temps necessari per etiquetar una gran quantitat de dades, la forma de resoldre el problema probablement canviaria, i passaria a ser un d'aprenentatge supervisat.

Aquests, aprenen, partint d'unes dades etiquetades, a replicar la forma en la qual estan etiquetades. En aquest cas, podríem posar d'exemple el que s'ha mencionat al principi de l'apartat de detecció de frau: si tens dades etiquetades com a fraudulent, pots aplicar directament un classificador que aprengui a etiquetar. Tot i això, partim de les dades proporcionades per al challenge, on no hi



havia etiquetes, i per aquest motiu aplicar aquest tipus de models no era viable. Considerem que amb la proposta presentada, es pot observar molt bé com funciona l'aprenentatge no supervisat, quines són les seves virtuts i capacitats sobretot a l'hora de trobar patrons.



## 4 Conclusions

Amb el model presentat, es pot realitzar una estimació de la deterioració del valor de les existències, indicant també els motius d'aquesta estimació. Un auditor, probablement, no haurà de dur a terme aquesta estimació, sinó analitzar si és correcta la que du a terme l'empresa, però es podria analitzar la diferència entre aquesta estimació i la de l'empresa, per exemple, facilitant i agilitzant la tasca de l'auditor.

De cada un dels índexs podem extreure'n conclusions pròpies:

El primer, basat en una fórmula, ens proporciona una base sòlida, molt transparent i fàcil d'entendre, que destaca alguns dels factors que influeixen en el deteriorament i ens serveix de punt de partida.

El segon, que utilitza sèries temporals, permet obtenir un valor relacionat amb la tendència del producte, i a més pot servir per a portar a cap una millor gestió de cara al futur.

Finalment, el tercer presenta una visió més innovadora i moderna del problema, que mitjançant una arquitectura específica de xarxa neuronal i partint d'un sol punt de referència, permet catalogar tots els altres.

En cas de disposar de dades etiquetades (amb el deteriorament ja indicat), el nostre model és capaç d'adaptar aquest tercer càlcul (el basat en una xarxa neuronal de tipus autoencoder) a les dades etiquetades per l'auditor, aprenent els patrons que hi troba i aplicant-los en les seves futures prediccions sobre el deteriorament d'existències. Tot i això, el model presentat és capaç de detectar patrons complexos i fer estimacions amb sentit, que a més poden acabar de ser refinades per un auditor i interpretades automàticament.

Addicionalment, al model de predicció, hem incorporat una forma d'afegir traçabilitat, ja que la considerem molt important en l'àmbit de l'auditoria, i també altres models addicionals que permeten veure més tecnologies que es podrien aplicar, com és el cas del mini-Xat (exemple de la pàgina anterior) que permet crear una interfície per interactuar amb les dades molt més fàcilment. Amb això, creiem que aportem un valor afegit als auditors que pot facilitar molt la realització de la seva tasca, que, al cap i a la fi, és l'objectiu del challenge.

S'ha observat com pot agilitar la intel·ligència artificial el procés i s'han plantejat també propostes addicionals per acabar de complementar-lo. El problema del deteriorament del valor d'existències és molt complex, ja que depèn de molts aspectes del producte i d'elements que molts cops no són fixos. Per tant, no hi ha una fórmula exacta que et permeti saber amb exactitud quin és el valor més adequat. Amb la proposta presentada, es busca guiar una mica aquests valors que es poden considerar acceptables, a partir de patrons trobats de forma automàtica en les dades. A part d'aquesta estimació, s'han tingut en compte altres problemes relacionats amb l'auditoria, com la justificació de les decisions, i s'ha aportat una traçabilitat que dona encara més valor al model.

Altres tècniques, si bé no afronten el problema directament, s'ha vist que poden facilitar aquesta tasca, com per exemple el fet de poder obtenir informació d'una gran base de dades a partir de llenguatge natural, o de poder classificar automàticament els productes en diferents grups.



---

Aquest informe final del projecte ha contemplat tots els detalls tècnics i conceptuals del model implementat i totes les propostes addicionals fetes. Tot el codi del projecte es pot trobar a [1].

## 5 Referències

- [1] Cai Selvas Sala, Roger Baiges Trilla, and Pau Hidalgo Pujol. <https://github.com/caiselvass/challenge-auditoria>, Apr 2024.
- [2] The pandas development team. pandas-dev/pandas: Pandas, February 2020.
- [3] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.
- [4] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting, 02 2020.
- [5] Zhong-Qun Tian, Ziqing Ma, Quan Wen, Xue Wang, L Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. 01 2022.
- [6] Kin G. Olivares, Cristian Challú, Federico Garza, Max Mergenthaler Canseco, and Artur Dubrawski. NeuralForecast: User friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022, 2022.
- [7] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. 10 2022.
- [8] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. 08 2013.
- [9] Microsoft Corporation et al. interpret: A unified framework for machine learning interpretability, 2021.
- [10] Sheo Kumar, Vinit Kumar Gunjan, Mohd Dilshad Ansari, and Rashmi Pathak. Credit card fraud detection using support vector machine. *Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications*, pages 27–37, 2022.
- [11] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.
- [12] C.W. Holsapple and A.B. Whinston. Business expert systems-gaining a competitive edge. *Twenty-Third Annual Hawaii International Conference on System Sciences*, 02 1990.