

Projecte fi de màster

Estudi: Màster en Enginyeria Industrial

Títol: La servitització com a nou model de negoci a Cafès Cornellà.

Document: Memòria i annexos

Alumna: Júlia Soler Boada

Tutor: Rudi de Castro Vila  
Departament: Organització, gestió empresarial i  
disseny del producte  
Àrea: Organització d'empreses

Convocatòria (mes/any): Juny 2022



PROJECTE FI DE MÀSTER

---

# La servitització com a nou model de negoci a Cafès Cornellà

---

*Autora:*

Júlia SOLER BOADA

Juny 2022

Màster en Enginyeria Industrial

*Tutor:*

Rudi DE CASTRO VILA



# Agraïments

Vull començar aquest projecte donant les gràcies a en Pere Cornellà per haver-me donat l'oportunitat de formar part del projecte de servitització que està iniciant a l'empresa Cafès Cornellà i per haver-me donat suport en tot moment.



# Índex

<b>1</b>	<b>Glossari</b>	<b>1</b>
<b>2</b>	<b>Proposta</b>	<b>3</b>
2.1	Origen del projecte . . . . .	3
2.2	Motivació . . . . .	4
2.3	Requeriments previs . . . . .	4
<b>3</b>	<b>Introducció</b>	<b>5</b>
3.1	Objectius del projecte . . . . .	6
3.2	Abast del projecte . . . . .	6
<b>4</b>	<b>Servitització</b>	<b>7</b>
4.1	Economia circular i servitització . . . . .	7
4.2	Servitització a nivell català . . . . .	9
<b>5</b>	<b>Marc de treball i conceptes previ</b>	<b>11</b>
5.1	Model de negoci actual . . . . .	11
5.2	Nova proposta de model de negoci . . . . .	14
<b>6</b>	<b>Sèries temporals: estat de l'art i coneixements previs</b>	<b>19</b>
6.1	Sèries temporals . . . . .	20
6.2	Components de les sèries temporals . . . . .	21
6.3	Predicció de les sèries temporals . . . . .	22
6.4	Metodologia . . . . .	23
<b>7</b>	<b>Obtenció de dades i anàlisi exploratòria</b>	<b>25</b>
7.1	EDA realitzat per Eurecat . . . . .	25
7.2	Subconjunt de dades . . . . .	26
<b>8</b>	<b>Models de predicció</b>	<b>29</b>
8.1	ARIMA . . . . .	29
8.1.1	SARIMA . . . . .	31
8.2	Prophet . . . . .	32
8.2.1	Tendència . . . . .	32
8.2.2	Estacionalitat . . . . .	33
8.2.3	Vacances . . . . .	33
8.2.4	Error . . . . .	33
8.2.5	Avantatges respecte el mètode ARIMA . . . . .	34

<b>La servitització com a nou model de negoci a Cafès Cornellà</b>	<b>Memòria</b>
8.3 Xarxa neuronal LSTM . . . . .	34
8.3.1 Xarxa neuronal RNN . . . . .	34
8.3.2 Xarxa neuronal LSTM . . . . .	35
<b>9 Mètriques d'avaluació i càlcul d'error</b>	<b>39</b>
<b>10 Implementació i resultats obtinguts</b>	<b>41</b>
10.1 ARIMA estacional . . . . .	43
10.2 Prophet . . . . .	50
10.3 Xarxa neuronal LSTM . . . . .	56
10.4 Resum resultats obtinguts . . . . .	63
<b>11 Avaluació econòmica</b>	<b>65</b>
<b>12 Conclusions</b>	<b>67</b>
<b>13 Relació de documents</b>	<b>71</b>
<b>14 Bibliografia</b>	<b>73</b>
<b>Annex A. Codi</b>	<b>75</b>
<b>Annex B. Resum resultats</b>	<b>89</b>



# Índex de figures

4.1	Diagrama dels models productius i de consum de l'economia lineal. . . . .	7
4.2	Diagrama dels models productius i de consum de l'economia circular. . . . .	8
5.1	Esquema sistema venda de producte del model de negoci actual.	12
5.2	Plataforma on es recullen les dades obtingudes de les cafeteres.	13
5.3	Primera versió de la plataforma Cafeteria 4.0 destinada als clients.	15
5.4	Esquema model de negoci servititzat. . . . .	16
5.5	Diagrama Debilitats, Amenaces, Fortaleses i Oportunitats. . . . .	17
6.1	Exemples de sèries temporals. . . . .	20
6.2	Exemples de sèries temporals amb diferents característiques. . . . .	21
7.1	Resultats gràfics de l'EDA realitzat per Eurecat. . . . .	26
8.1	Metodologia de previsió Box-Jenkins. . . . .	31
8.2	Arquitectura RNN bàsica. . . . .	35
8.3	Arquitectura RNN extesa. . . . .	35
8.4	Arquitectura model RNN. . . . .	36
8.5	Arquitectura model LSTM. . . . .	36
8.6	Portes i estat intern d'una xarxa LSTM. . . . .	37
10.1	Cafès servits des de gener de 2018 fins a maig de 2022. . . . .	42
10.2	Carregar les dades. . . . .	42
10.3	Importar les llibreries. . . . .	43
10.4	Generació de diferents combinacions dels paràmetres. . . . .	44
10.5	Obtenció de l'AIC per cada combinació del model SARIMA. . . . .	45
10.6	Model SARIMA amb un AIC més baix. . . . .	45
10.7	Ajustament del model SARIMA amb els paràmetres òptims. . . . .	46
10.8	Resultats obtinguts a partir de la comanda <i>summary</i> . . . . .	46
10.9	Resultats obtinguts a partir de la comanda <i>plot_diagnostics</i> . . . . .	47
10.10	Predicció pel període de dates del subconjunt <i>test_data</i> . . . . .	48
10.11	Representació gràfica del model de predicció ARIMA. . . . .	48
10.12	Resultats de la predicció amb el model SARIMA. . . . .	49
10.13	Resultat mètriques d'avaluació. . . . .	49
10.14	Instal·lar i importar llibreries. . . . .	50
10.15	Carregar les dades i posar-les amb el format adequat. . . . .	51

---

10.16 Creació del model Prophet. . . . .	51
10.17 Creació del <i>DataFrame</i> amb les dates que es volen predir. . . . .	52
10.18 Resultats obtinguts de la comanda <i>predict</i> . . . . .	52
10.19 Representació gràfica del model de predicció Prophet. . . . .	53
10.20 Representació del model de predicció Prophet. . . . .	53
10.21 Valors de la predicció amb el model Prophet. . . . .	54
10.22 Components anuals i setmanals de la predicció. . . . .	55
10.23 Resultat de les mètriques d'avaluació. . . . .	55
10.24 Importar les llibreries. . . . .	56
10.25 Escalar les dades amb l'escalador <i>MinMaxScaler</i> . . . . .	57
10.26 Definió de <i>TimeseriesGenerator</i> . . . . .	57
10.27 Definió del model. . . . .	58
10.28 Arquitectura de la xarxa neuronal. . . . .	59
10.29 Ajustament del model. . . . .	59
10.30 Funció de pèrdua. . . . .	60
10.31 Predicció de valors futurs. . . . .	60
10.32 Escalar les dades. . . . .	61
10.33 Representació del model de predicció LSTM. . . . .	61
10.34 Valors de la predicció amb la xarxa LSTM. . . . .	62
10.35 Validació dels resultats obtinguts. . . . .	62
10.36 Resultats obtinguts dels diferents models. . . . .	63

# Índex de taules

10.1	Valors de les dades reals i de les prediccions obtingudes. . . . .	64
10.2	Resultats obtinguts de les mètriques d'avaluació. . . . .	64
11.1	Avaluació econòmica. . . . .	65



## CAPÍTOL 1

# Glossari

---

- PIME: Acrònim de Petita I Mitjana Empresa.
- PIMEC: Acrònim de Petita I Mitjana Empresa de Catalunya.
- HORECA: Acrònim d'HOtels, REstaurants i CAFeteries.
- M2M: Comunicació màquina a màquina. Acrònim de l'anglès Machine to Machine.
- DAFO: Acrònim de Debilitats, Amenaces, Fortaleses i Oportunitats
- ARIMA: Model de mitjana mòbil integrada autorregressiva. Acrònim de l'anglès d'Autoregressive Integrated Moving Average Model.
- SARIMA: Model de mitjana mòbil integrada autorregressiva estacional. Acrònim de l'anglès Seasonal Autoregressive Integrated Moving Average Model.
- RNN: Xarxa neuronal recurrent. Acrònim de l'anglès de Recurrent Neural Network.
- LSTM: Unitats de memòria a curt i llarg termini. Acrònim de l'anglès de Long Short-Term Memory.
- MAE: Error absolut mitjà. Acrònim de l'anglès de Mean Absolute Error.
- MSE: Error quadràtic mitjà. Acrònim en anglès de l'Mean Squared Error.
- RMSE: Arrel de l'error quadràtic mitjà. Acrònim de l'anglès de Root-Mean-Square Error.



## 2.1 Origen del projecte

Tradicionalment les empreses productores han centrat el seu model de negoci en la venda del producte que fabriquen. Tot i que aquest model ja funciona, en aquests darrers anys s'ha iniciat una transició cap a un nou model en el que les empreses volen oferir, a més del producte, un servei. Aquesta nova estratègia és la que es coneix com a servitització. Implantar un model enfocat cap a la servitització implica, entre d'altres, modificar els procediments de treball i també la forma d'apropar-se als clients. És un procés complex en el que és necessari analitzar les diferents decisions i estratègies abans d'aplicar-les per veure l'impacte que tindran.

El projecte que es presenta té l'origen en l'empresa Cafès Cornellà i en el seu interès en implantar un model de negoci enfocat cap a la servitització. Actualment Cafès Cornellà instal·la cafeteres amb tecnologia M2M (Machine to Machine) a alguns dels seus clients a partir de les quals obté dades referents a l'ús d'aquestes, com ara cafès servits o neteges fetes a la cafetera. Aquestes dades i altres són enviades a un servidor de Cafès Cornellà.

Per gestionar aquestes dades, Cafès Cornellà està implementant una plataforma tecnològica, anomenada Cafeteria 4.0, que complementarà el pas al nou model de negoci centrat en la servitització. Aquesta plataforma va destinada als seus clients per tal de que puguin millorar la rendibilitat del seus establiments a partir de les dades que s'obtenen de les cafeteres.

L'objectiu d'aquesta nova proposta de valor basada en la servitització és millorar els resultats del negoci de Cafès Cornellà tot oferint als seus clients un seguit de funcionalitats i informació que fins ara no disposaven. Per exemple, es pretén que amb les dades que les cafeteres envien al servidor es pugui controlar i millorar el rendiment (cafès servits per cada quilogram de cafè), gestionar els manteniments preventius per evitar averies, predir consums, i fins i tot mesurar la qualitat de la tassa de cafè en temps real.

En aquest context, el professor Yago de Zabala del Departament d'Organització i Gestió Empresarial i Disseny del Producte, en el marc de la seva tesi doctoral sobre la servitització, i supervisada per en Rudi de Castro i l'Andrea Bikfalvi, va contactar amb Cafès Cornellà per fer un Case Study del seu nou mo-

del de negoci. El seu interès comú en la temàtica els va portar a oferir una estada en entorn laboral no curricular de la qual sorgeix el projecte que es presenta.

## **2.2 Motivació**

La principal motivació darrera d'aquest projecte ha estat la possibilitat de poder posar en pràctica i ampliar conceptes d'organització d'empresa que s'han vist al llarg del màster en un cas real i alhora aprendre noves tècniques centrades en la gestió de dades.

Per la part d'organització d'empresa, fer un canvi de model de negoci implica, entre d'altres, analitzar els diferents segments de mercat, formular hipòtesis i portar a terme les proves de validació necessàries per al seu desplegament i llançament.

Per la part de gestió de dades, i per tal de poder tractar les dades obtingudes de les cafeteres, suposa fer un estudi a fons dels diferents models de predicció que permetin explotar-les al màxim per tal d'obtenir la informació necessària per posteriorment prendre les millors decisions, tant per part de Cafès Cornellà com dels seus clients.

## **2.3 Requeriments previs**

Inicialment caldrà fer una immersió intensiva en els conceptes i teoria de la servitització, així com entrar en el detall del procés de servitització a Cafès Cornellà. També serà necessari veure en quin punt està la plataforma Cafeteria 4.0 i quines són les dades que ens facilita i que seran objecte de tractament.

També hem de saber quins són els requeriments i les especificacions per portar a terme el projecte amb les mínimes proves viables.



## CAPÍTOL 3

# Introducció

---

La servitització és una estratègia d'economia circular que ofereix una proposta de valor enfocada a passar de vendre productes a vendre serveis o productes amb un servei complementari. Un exemple clàssic de servitització és el de l'empresa Rolls Royce. L'any 1962, va passar de vendre turbines d'avió a oferir un servei avançat anomenat *TotalCare*. Així doncs, enlloc de vendre-les com a producte, conservaven la propietat i les van començar a arrendar. D'aquesta manera, eren capaços de monitoritzar-les constantment, realitzar un millor manteniment, augmentar els seus marges i oferir als seus clients un producte més avançat.

Cafès Cornellà, empresa gironina del sector agroalimentari productora de cafè torrat, actualment basa el seu model de negoci en la venda de quilograms de gra de cafè. Des del 2019 ha iniciat un projecte de canvi de model de negoci per passar de facturar quilograms de cafè a facturar les tasses de cafè venudes amb l'objectiu de fer una transició cap a una proposta de valor més eficient i circular. Aquest canvi va acompanyat d'una plataforma, Cafeteria 4.0, que té per objectiu impulsar aquesta nova proposta de valor. Pràcticament no existeixen casos de servitització en la indústria alimentària en general, i en l'àmbit del cafè en particular, i és sens dubte un cas pioner a la indústria, no només a Catalunya, sinó també a nivell europeu i mundial. La innovació del projecte rau en la transició cap a un model de negoci que aconsegueix desacoblar la facturació de l'empresa respecte el consum de matèries primeres, aportant més valor als clients (hotels, restaurants i cafeteries) i qualitat als clients finals (consumidors).

Aquest procés de transició cap a la servitització es complementa amb la monitorització de les cafeteres iniciada l'any 2016. Aquesta permet fer el seguiment de la venda de tasses de cafès i recollir variables relacionades amb la bona preparació d'un cafè, com per exemple, les neteges que es fan a les cafeteres. Per tant, permet saber què està passant en els establiments connectats. Aquestes dades són essencials per poder validar i definir el nou model de negoci i alhora han de servir per alimentar la plataforma Cafeteria 4.0 podent optimitzar així els resultats.

El nou model de negoci és clar, a partir d'una solució tecnològica s'oferirà una gestió eficient del cafè que permetrà parametritzar i optimitzar el rendiment, la qualitat i la satisfacció dels consumidors, i així augmentar la rendi-

bilitat dels establiments. El repte a nivell d'empresa està en determinar quina informació oferim al client, és a dir, de totes les dades de les que es disposa, quines proporcionen informació útil per poder millorar el servei que ofereix.

### **3.1 Objectius del projecte**

En el moment d'iniciar aquest projecte, Cafès Cornellà disposa d'un conjunt de clients per iniciar una prova pilot. Partint com a cas d'ús les dades d'aquests clients, l'objectiu d'aquest projecte és definir i validar l'estratègia sobre el model de negoci, ja definit, associat a la proposta de servitització.

### **3.2 Abast del projecte**

L'abast del projecte es pot dividir en dos blocs segons els focus principals d'interès, un primer centrat en l'organització de l'empresa i el segon en la gestió de les dades.

Pel que fa a l'organització de l'empresa, abans de definir i validar el nou model de negoci ens cal:

- Estudiar els elements clau de la servitització com a model de negoci. Ens interessa entendre el concepte d'economia circular i el seu potencial a nivell català identificant els sectors claus implicats.
- Analitzar el model de negoci actual de Cafès Cornellà i la proposta del nou model de servitització. Ens interessa entendre el context actual de l'empresa per poder identificar les característiques del model actual i alhora conèixer les estratègies de les que disposa per fer el pas cap a la servitització.

Pel que fa a la gestió de dades ens cal:

- Proposar estratègies i fer les proves mínimes viables per tal de poder validar la nova proposta de valor. Aquest punt es considera el nucli del projecte. Partint de les dades de les quals ja disposa Cafès Cornellà es farà un anàlisi per poder predir la informació que cal proporcionar als clients en el marc del model de servitització. Concretament, es treballarà en una de les funcionalitat de la plataforma, per la qual caldrà estudiar i aplicar models de predicció, definir mètriques d'avaluació dels models predictius i finalment extreure'n les conclusions per retornar la informació adequada al client.

## CAPÍTOL 4

# Servitització

---

La servitització es pot definir com un nou model de negoci en el que l'empresa passa de proveir productes a proveir productes amb un servei complementari com a valor afegit. Per poder oferir aquest servei complementari és imprescindible conèixer les necessitats del client i saber l'ús que fa dels productes. Aquest nou model de negoci es defineix com un model d'economia circular que millora el clàssic model lineal. A continuació, presentem aquests conceptes amb detall.

### 4.1 Economia circular i servitització

L'economia es pot definir com un model d'administració dels recursos disponibles en sistemes de producció, distribució, comerç i consum per satisfer les necessitats humanes. Les diferents etapes de l'economia es poden modelitzar de forma lineal o de forma circular.

El model econòmic actual és lineal perquè, tal i com es mostra en la Figura 4.1, està basat en: (i) extracció de matèria i energia; (ii) producció; (iii) consum i (iv) generació de residus i emissions. Aquest model és insostenible a llarg termini. Per una banda els recursos inicials són finits i per altra banda, la generació de residus i emissions té un impacte negatiu amb conseqüències pèssimes i per tant a evitar.



Figura 4.1: Diagrama dels models productius i de consum de l'economia lineal.

Font: [PIMEC 2020]

Com alternativa a aquest model lineal es proposa el model circular, és a dir, una economia orientada a l'assoliment de sistemes de producció i consum més eficients i resilents, que preservin els recursos dins d'un cicle continu, optimitzant-ne el seu valor. Aquest nou paradigma afecta a tota la cadena de valor, incloent al ciutadà consumidor, i adopta una visió de cicle de vida que integra la perspectiva social, econòmica i ambiental. Totes les polítiques actuals estan enfocades a aquest nou model i centrades en la sostenibilitat.

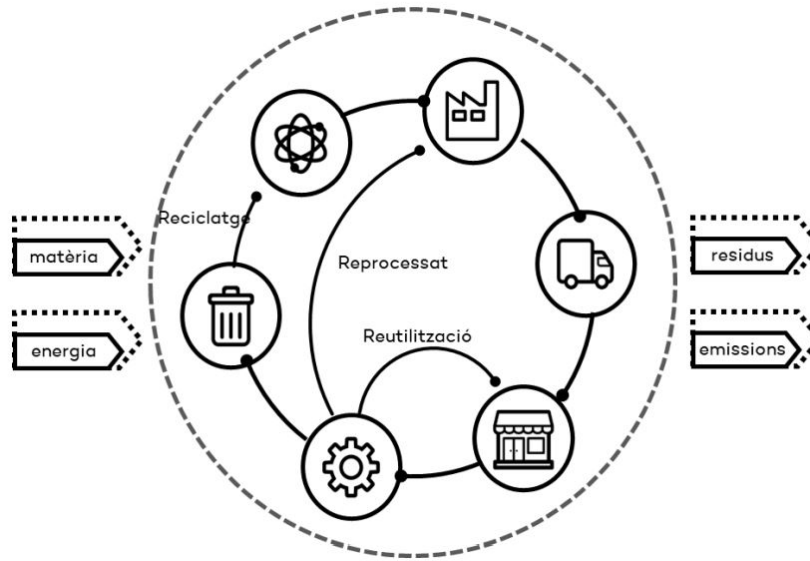


Figura 4.2: Diagrama dels models productius i de consum de l'economia circular.  
Font: [PIMEC 2020]

Actualment, complir amb les normes i objectius de sostenibilitat en el context d'economia circular suposa un desafiament tecnològic a l'hora de desenvolupar productes i processos més respectuosos amb el medi ambient. En aquest context, l'economia circular es posiciona com un marc de referència per aquesta reinvençió i la servitització representa una oportunitat per portar a terme nous models de negoci circular. Podem entendre per tant la servitització com una estratègia d'economia circular que ofereix una proposta de valor enfocada a passar de vendre productes a vendre serveis o productes amb un servei complementari.

Focalitzant-nos en la servitització, poden considerar-se tres nivells en funció de l'orientació del servei. Aquests són:

1. Serveis orientats al producte, per exemple el servei d'assessorament que s'ofereix per fer servir una rentadora de la forma més eficient possible.
2. Serveis orientats a l'ús/accés, per exemple les bugaderies col·lectives on l'usuari pot fer servir rentadores de les que no n'és propietari.
3. Serveis orientats al resultat, per exemple el pagament pel servei de tenir sempre roba neta.

Com es pot veure, en la servitització, el punt focal del negoci no és la creació del producte, sinó el consumidor o client final i el servei que se li vol oferir. Per tant, per poder-lo definir cal conèixer bé les necessitats del client i l'ús que

fa dels productes. D'aquesta manera s'alineen els interessos de tots els agents implicats en els models de producció i consum, fent de la servitització una de les estratègies d'economia circular més atractives pels diferents avantatges que aporta. Entre aquests avantatges podem remarcar:

- A nivell d'empresa, oferir producte i servei permet fidelitzar als clients alhora que genera més beneficis. Això és pel fet que els clients estan disposats a pagar més per un producte quan aquest inclou serveis avançats adaptats a les necessitats de l'usuari final. Si el servei és òptim, la relació empresa-client s'enforteix, l'experiència del consumidor és positiva i la fidelització molt més efectiva.
- A nivell dels consumidors, aquests gaudeixen de productes de major qualitat i es despreocupen de les responsabilitats associades a la propietat.
- A nivell de societat i medi ambient, es minimitza l'ús de recursos i s'adopten hàbits de consum més sostenibles.

L'impacte positiu de la servitització ha fet que la majoria de països defineixin plans i programes d'actuació per afavorir-ne la seva implantació i Catalunya no ha estat una excepció.

## **4.2 Servitització a nivell català**

ServitiCAT, un estudi sectorial per identificar les oportunitats i els reptes d'incloure la servitització com una estratègia d'economia circular a les pimes catalanes, va publicar el desembre de 2020 l'informe "Aplicació de les servitització a les pimes catalanes" [PIMEC 2020], desenvolupat per PIMEC (Petita i Mitjana Empresa de Catalunya) i amb el suport de l'Agència de Residus de Catalunya. La conclusió principal de l'estudi de ServitiCAT és que l'estratègia de la servitització té un gran potencial per les pimes catalanes i representa una oportunitat per empreses, territori i consumidors, el que justifica que el seu impuls tant des de l'esfera pública com des de la privada sigui una prioritat i encara més dintre del context de crisi generat per la Covid-19, on la sostenibilitat i l'adaptació a nous models de negoci i de propietat pot representar una oportunitat de diferenciar-se i aportar valor afegit sobre la competència. En aquest informe també s'indica que l'any 2015, la Generalitat de Catalunya va publicar una estratègia d'impuls a l'economia verda i circular amb l'objectiu de contribuir a la construcció d'un territori més resiliència a les crisis econòmiques i ambientals, menys dependent dels recursos externs i més inclusiva. Aquesta estratègia identifica els sectors considerats claus per la transformació del teixit productiu català, basant-se en

el sectors econòmics líders identificats per l'Estratègia per a l'Especialització Intel·ligent de Catalunya (RIS3CAT), que apunta als següents àmbits sectorials líders:

- Alimentació
- Química, energia i recursos
- Sistemes industrials
- Indústries relacionades amb el disseny
- Indústries de la mobilitat sostenible
- Indústries de la salut

El sector de l'alimentació inclou la indústria agroalimentària, però també altres graons de la cadena de valor, com els sectors primaris, la distribució, les indústries de l'envasat i l'embalatge, la maquinària per l'alimentació, els additius i matèries primeres, la gastronomia i la restauració. En aquest sector s'hi troba Cafès Cornellà.

# Marc de treball i conceptes previ

---

Cafès Cornellà és una empresa gironina que té el seus orígens el 1920 quan els germans Narcís i Pere Cornellà van posar en marxa la primera planta de torrefacció a Girona. Es tractava d'una activitat completament nova i en poc temps es van convertir en el principal subministrador de cafè de les comarques gironines. El 1954 es va incorporar a l'activitat la segona generació coincidint amb la nova normativa de la venda del cafè envasat. Cafès Cornellà va adquirir en aquest moment la segona màquina automàtica de tot l'Estat espanyol per envasar el cafè. El 1980 es va incorporar a l'empresa la tercera generació i 4 anys més tard es van traslladar l'empresa del centre de Girona a Fornells de la Selva. Des de llavors l'empresa opera en una nau industrial on els processos de producció estan totalment automatitzats, tant pel que fa a la torrefacció, mescles i envasament al buit de tots es seus productes i on es disposa d'un laboratori d'anàlisi per desenvolupar noves fórmules de torrefacció. Actualment, Cafès Cornellà serveix 25 milions de tasses anuals (per més informació, consulteu <https://cafescornella.coffee/>).

## 5.1 Model de negoci actual

Cafès Cornellà es posiciona com un torrefactor i distribuïdor de cafè de gamma alta especialitzada en el sector HORECA, (acrònim de Hosteleria, Restaurants i Cafeteries). El model actual de negoci de l'empresa es basa en la producció de cafè torrat, que subministra juntament amb altres productes, ingredients i els equipaments necessaris per a la producció de les tasses de cafè en els establiments dels clients. Això ha comportat l'adquisició i posada en servei de màquines, molins de cafè i sistemes de tractament d'aigua, juntament amb la implementació d'un Sistema de Qualitat Integral (SIQ). El SIQ permet una millora contínua que comença per una qualitat constant, una formació barista progressiva, un manteniment preventiu de la maquinària i culmina quan el client és conscient que amb aquest sistema millora la seva rendibilitat. [Cafès Cornellà ]

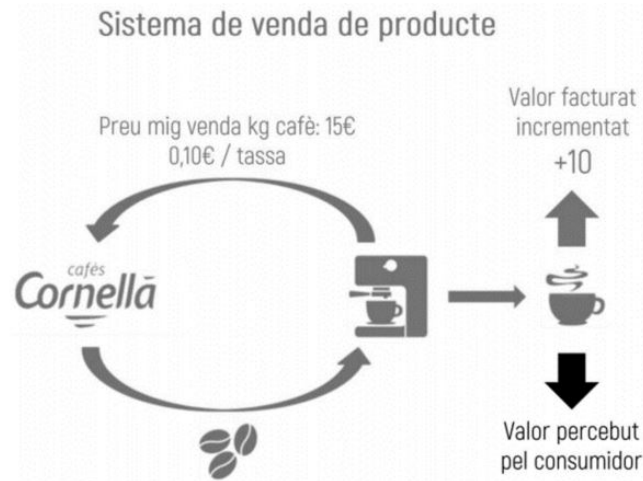


Figura 5.1: Esquema sistema venda de producte del model de negoci actual.  
Font: Cafès Cornellà

Les limitacions d'aquest model de negoci es poden resumir en:

- Es traspassa la responsabilitat de l'eficiència de l'ús de les màquines de cafè i el producte als clients.
- El control que la marca té sobre la qualitat i l'eficiència de l'elaboració de les tasses de cafè és mínim, fent que el marge de negoci sigui limitat i vinculat als kg de producte.
- La capacitat d'innovació i diferenciació del seu negoci és molt limitada ja que existeixen més d'una desena de competidors que ofereixen el producte "gra de cafè" als establiments.

Per tant, amb l'objectiu de resoldre les limitacions del model actual, Cafès Cornellà va iniciar una nova estratègia amb l'objectiu d'obtenir informació dels seus clients. El 2016 van posar en funcionament les primeres cafeteres connectades amb la tecnologia M2M (Machine to Machine). Aquestes màquines, dissenyades per Quality Espresso, envien informació periòdicament de:

- La quantitat de cafès que ha fet la cafetera
- La quantitat de neteges de la màquina que s'han dut a terme: que permetrà saber el manteniment que s'està fent de la màquina.
- La qualitat de cada cafè: si s'ha fet subextret (en massa poc temps) o si ha estat sobreextret (massa fet).
- Alarmes de nivell i de volumètric



Aquestes dades es gestionen des d'una plataforma informàtica de l'empresa que els permet veure les dades dels locals que disposen d'aquest tipus de cafeteres. En la Figura 5.2 es mostra una captura de pantalla de la plataforma informàtica.

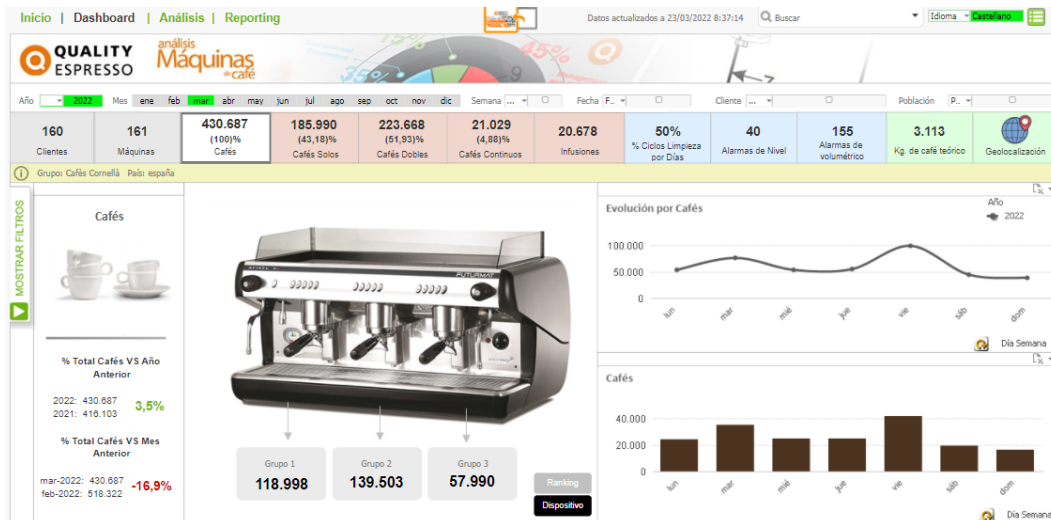


Figura 5.2: Plataforma on es recullen les dades obtingudes de les cafeteres.

Font: Cafès Cornellà

El potencial de les dades que s'obtenien dels clients era molt elevat i podia aportar informació important a l'empresa per poder oferir un millor servei als seus clients. Per poder identificar i extreure aquesta informació Cafès Cornellà contacta amb Eurecat. Eurecat és un proveïdor de tecnologia industrial de Catalunya que ofereix entre d'altres coses, serveis d'R+D aplicat, consultoria tecnològica i desenvolupament de productes i serveis [Eurecat]. Aquest els desenvolupa un sistema de propensió de clients potencials en el que aplicant eines d'intel·ligència artificial es volen identificar els clients potencials més propers a la seva proposta de valor. El sistema desenvolupat, a més dels diferents paràmetres monitoritzats, considera les dades que es recullen en les visites comercials (per exemple, nombre de treballadors, àpats que ofereixen als clients...) i la que obtenen del Google Places (per exemple, horaris, valoracions per part dels clients...). A partir d'aquí la idea de model de servitització ja comença a agafar forma i es planteja el nou model de negoci.

## 5.2 Nova proposta de model de negoci

La servitització planteja passar de vendre quilos de producte a facturar per tasses de cafè, amb l'objectiu de transformar el model de negoci cap a una proposta de valor més eficient i circular.

Es plantegen tres grans eixos de treball:

1. Formació del personal dels bars: Utilitzant els paràmetres adequats per a servir l'espresso perfecte s'obté una tassa de cafè de major qualitat i un ús eficient dels recursos, que pot arribar a suposar un estalvi de fins a un 20% de gra, el que es tradueix en una major rendibilitat i al mateix temps suposa una reducció de la generació de residus orgànics. Per altra banda, l'ús adequat de les màquines permet maximitzar la seva vida útil, que pot allargar-se de 3 a 10 anys.
2. Control del rendiment: Controlant el rendiment de l'etapa d'ús dels serveis oferts per assegurar la màxima eficiència. Per fer-ho, el projecte fa ús de l'Internet de les Coses (IoT per les seves sigles en anglès) implementat a les màquines de cafè espresso instal·lades als bars, que les interconnecta digitalment amb l'objectiu de recollir dades sobre el seu funcionament (número de cicles d'ús, durada, temperatura de l'aigua, pressió, cicles de rentat, etc.), que permetrà generar un informe setmanal del funcionament i emetre recomanacions per tal que els baristes millorin la qualitat i l'eficiència del cafè servit.
3. Manteniment predictiu de les màquines: Detectant possibles errades i defectes en etapes incipients a través de l'IoT contribuint a la maximització de la vida útil de les màquines, aconseguint una alineació amb les estratègies de prevenció de residus (RAEEs).

Tot aquest nou model de negoci es planteja a través de la plataforma Cafeteria 4.0 destinada als clients, de la que ja es disposa d'una primera versió (veure Figura 5.3). Aquesta nova plataforma introdueix l'IoT de manera que el negoci agafi les dades que ell mateix genera i les utilitzi per obtenir la informació més adequada a l'hora de prendre decisions.



Figura 5.3: Primera versió de la plataforma Cafeteria 4.0 destinada als clients.  
Font: Cafès Cornellà

En aquest nou model de servitització, representat a la Figura 5.4, el client només pagarà el cafè de les tasses que hagi servit. El cost de la tassa passarà de 10 cèntims/tassa (que correspon al cost del model actual representat a la Figura 5.1) a 12 cèntims/tassa de manera que aquest increment de facturació pugui mantenir tot el sistema i, fins i tot, augmentar els beneficis de l'empresa. Aquest lleuger increment dels preus no hauria de significar un increment dels costos del client ja que el sistema li permetrà fer una gestió més eficient del cafè, on no es malgastarà cafè ja que el gramatge de cafè per cada tassa serà el correcte. Com a conseqüència la qualitat serà millor.

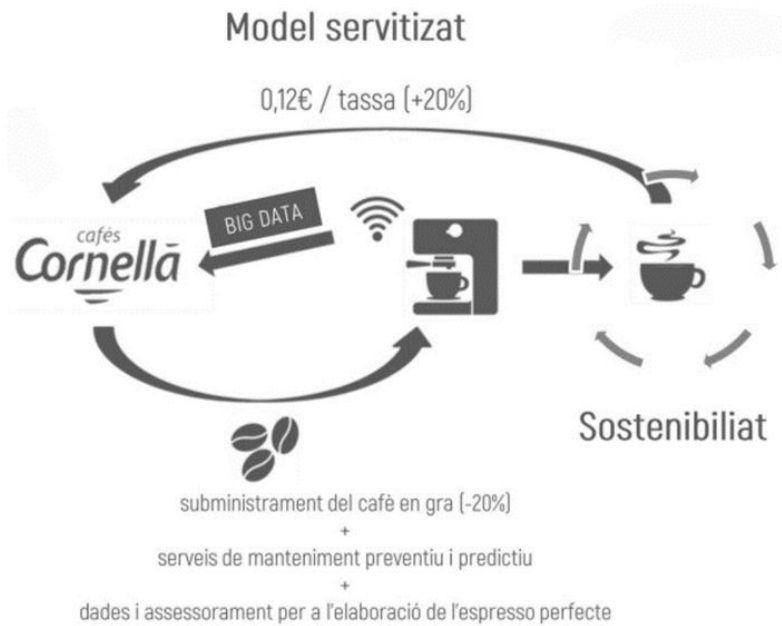


Figura 5.4: Esquema model de negoci servititzat.

Font: Cafès Cornellà

Des del punt de vista ambiental, es preveu un estalvi d'almenys un 10% de matèria prima per tassa de cafè servida. Si aquest estalvi s'apliqués a 50 bars (250.000 tasses de cafè/mes) significaria un estalvi anual de 21 tones de gra de cafè, oferint un millor producte final. A més, incrementaria la vida útil de les cafeteres espresso dels bars de 3 a 10 anys. Es preveu també la reducció de la petjada de carboni associada a l'activitat de l'empresa

A la Figura 5.5 es mostra el diagrama de Debilitats, Amenaces, Fortaleses i Oportunitats (DAFO) del nou model de servitització.

PUNTS FORTS	PUNTS FEBLES
<ul style="list-style-type: none"> <li>• Potent argumentari comercial</li> <li>• Orientació de l'empresa cap a la qualitat i la innovació</li> <li>• Bona imatge en el nostre mercat</li> <li>• El nivell tecnològic dels mitjans i el coneixement tècnic del producte elevat</li> <li>• Vinculació amb la figura barista</li> <li>• Coherència del discurs, som fabricants i distribuïdors del producte</li> <li>• Pla de formació interna per professionalitzar els equips</li> <li>• Estratègia de digitalització i nivell d'implantació per damunt de la nostra competència</li> <li>• Millora de qualitat d'imatge corporativa i excel·lent catàleg de productes</li> </ul>	<ul style="list-style-type: none"> <li>• La comunicació interna és insuficient</li> <li>• Separació entre departaments</li> <li>• Manca de planificació, improvisació, organització reactiva</li> <li>• Absència d'uns objectius més clars. Manca de control i valoració</li> <li>• Manca d'implicació d'una part de la plantilla</li> <li>• Imatge de marca exterior. Poc coneguts fora de Girona</li> <li>• Manca de major consens alhora d'aplicar els criteris i la política comercial</li> </ul>
OPORTUNITATS	AMENACES
<ul style="list-style-type: none"> <li>• Arribar al consumidor final ens permet fidelitzar els nostres clients. Utilitzar TIC, XXSS, etc</li> <li>• Augment de l'exigència del consumidor sobre la qualitat del cafè fora de la llar</li> <li>• Ser els primers en nínxols de mercat i aportar continuament novetats</li> <li>• Canvi en el perfil emprenedor d'hostaleria més preparats i més professionals</li> <li>• Desenvolupar nous productes i serveis enfocats en la sostenibilitat i l'ecologia</li> <li>• Entrada de nous competidors</li> </ul>	<ul style="list-style-type: none"> <li>• Situació de debilitat del mercat amb una oferta superior a la demanda</li> <li>• Competidors molt agressius, competeixen per preu i realitzen fortes ofertes</li> <li>• El sector domina molt ràpidament l'argumentari i les noves propostes que altres introdueixen al mercat</li> <li>• Desconeixement del consumidor per identificar un producte de qualitat</li> <li>• Procés de concentració de la competència</li> <li>• Entrada de nous competidors</li> <li>• Coherència del discurs, som fabricants i distribuïdors del producte</li> </ul>

Figura 5.5: Diagrama Debilitats, Amenaces, Fortaleses i Oportunitats.

Font: Cafès Cornellà



# Sèries temporals: estat de l'art i coneixements previs

---

La base del model de servitització que proposa Cafès Cornellà està en el pas de venda de tasses de cafè enlloc de quilograms de cafè. Aquesta nova aposta implica fer un estudi previ per poder veure quins serveis es poden oferir al voltant de la tassa de cafè. Aquestes nous serveis es presentaran com a funcionalitats de la plataforma Cafeteria 4.0 que s'oferirà als clients. En aquest projecte ens centrarem només en el nombre òptim de cafès, entenent-lo com el nombre de cafès que ha de fer el client per poder obtenir el màxim rendiment dels quilograms de cafè que ha comprat.

Cafès Cornellà defineix com a rendiment el nombre de tasses fetes per cada quilogram de cafè. Tenint en compte que el gramatge ideal és d'entre 7 i 7,5 grams per tassa de cafè, el rendiment òptim és aquell que està al voltant de 130 tasses per quilogram de cafè. En el cas de les cafeteries connectades, el fet de poder tenir informació en temps real de les tasses que s'estan fent permet a Cafès Cornellà aproximar quin és el rendiment de cada cafeteria a temps real.

Tot i que la idea és simple la seva implementació no ho és tant. Cada client és diferent i cal fer ajustaments per adaptar-se a les seves característiques. Per fer aquest ajustament i poder oferir un servei a mida ens cal recórrer als mètodes de predicció. Aquests utilitzen dades històriques com a aportacions per fer estimacions informades que siguin predictives per determinar la direcció de les tendències futures.

Introduïts els conceptes clau del model de servitització a Cafès Cornellà: el rendiment òptim i la predicció, en els següents apartats es presentaran els diferents mètodes per fer la predicció del nombre de tasses de cafè a partir de les dades que s'han recollit del client. Abans, s'explicaran alguns dels conceptes teòrics necessaris per abordar aquest treball, es veurà en què consisteix una sèrie temporal, quines són les seves components i la predicció d'aquestes sèries.

## 6.1 Sèries temporals

Una sèrie temporal és un conjunt d'observacions registrades en intervals regulars de temps. A cada instant, l'observació prové d'una variable que pot tenir una mateixa distribució o, a vegades, diferent. Possibles exemples de sèries temporals són els guanys trimestrals per acció, l'evolució de la temperatura mitjana mensual, el preu de referència del petroli de Brent o l'evolució dels mails d'un determinat servidor (vegeu Figura 6.1).

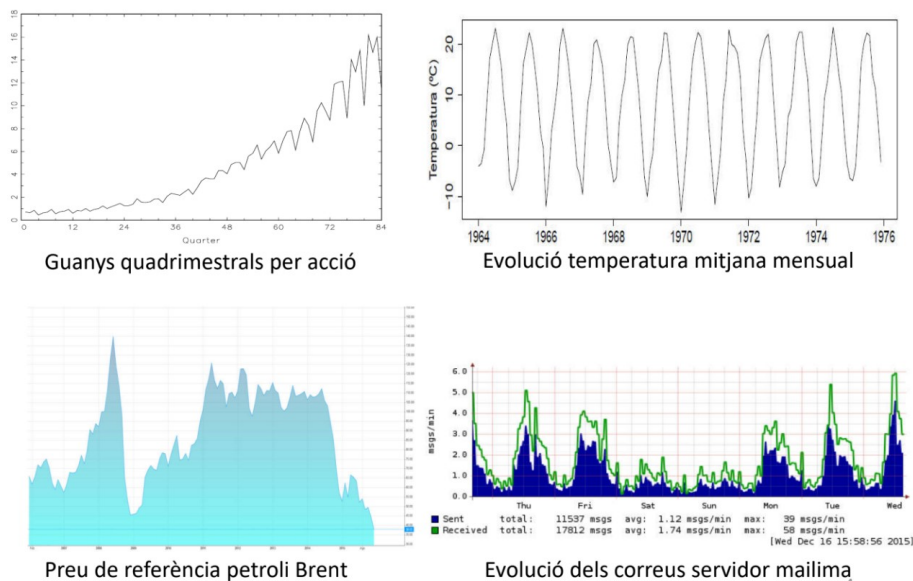


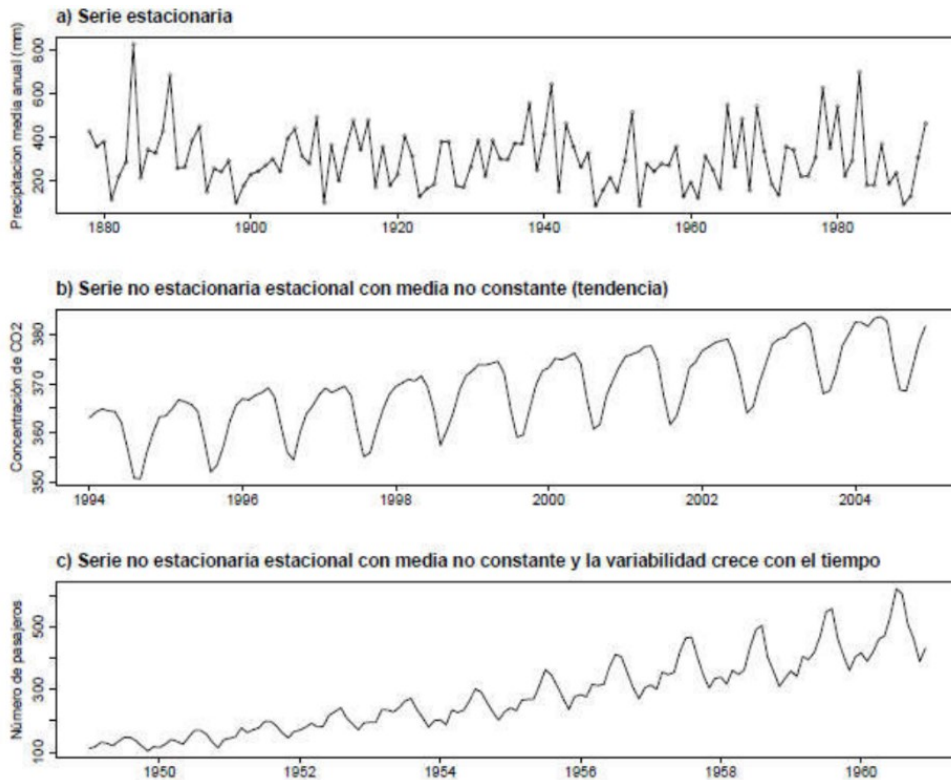
Figura 6.1: Exemples de sèries temporals.  
Font: Apunts estadística UdG (Santi Thió)

Les sèries temporals poden classificar-se en:

- Univariants o multivariants, segons el nombre de variables per cada observació.
- Discretes o contínues, segons si les observacions s'han obtingut cada cert període de temps o de forma contínua.
- Estacionàries o no estacionàries. En una sèrie temporal estacionària els valors fluctuen al voltant d'un valor mitjà, amb variabilitat constant (homoscedàstica), la correlació només depèn del temps transcorregut i no és estacional (és a dir que no oscil·la amb una periodicitat). D'altra banda, les sèries temporals no estacionàries són aquelles en les que no es compleix alguns dels propòsits anteriors.

A la següent figura, es mostren diferents casos de sèries amb diferents característiques.





Series temporales en R (version 1.2)  
Publicado por: Luis Cayuela y Ana Justel

Figura 6.2: Exemples de sèries temporals amb diferents característiques.  
Font: Apunts estadística UdG (Santi Thió)

Els principals objectius de l'anàlisi d'una sèrie temporal són:

- Explicar l'evolució d'un fenomen aleatori al llarg del temps
- Estudi de les característiques de les autocorrelacions
- Preveure els seus valors futurs
- Control òptim de processos

## 6.2 Components de les sèries temporals

L'anàlisi clàssic de sèries temporals considera que una sèrie temporal queda formada per quatre components: [Parra 022]

1.  $T_t$ : tendència que representa l'evolució de la sèrie a llarg termini.

2.  $C_t$ : component cíclica que indica les fluctuacions de caràcter periòdic, però no necessàriament regular, a mig termini en torn a la tendència.
3.  $S_t$ : component estacional que recull aquells comportaments de tipus regular i repetitiu que tenen lloc al llarg d'un període de temps, generalment igual o inferior a un any, i que són produïts per factors com vacances, dies festius...
4.  $E_t$ : variacions irregulars o soroll, són fluctuacions produïdes per factors eventuals, esporàdics i imprevisibles, que no mostren una periodicitat reconeixible.

No totes les sèries tenen tots els elements, tot i això l'associació d'aquests components en una sèrie temporal  $Y_t$ , pot respondre a diferents formes. La sèrie és de tipus additiu si:

$$Y_t = T_t + C_t + S_t + E_t \quad (6.1)$$

La sèrie és de tipus multiplicatiu si:

$$Y_t = T_t C_t S_t E_t \quad (6.2)$$

O bé podria ser una combinació de les dues, per exemple:

$$Y_t = T_t C_t S_t + E_t \quad (6.3)$$

### 6.3 Predicció de les sèries temporals

La predicció de sèries temporals es basa en el següent principi: coneixent el comportament passat d'una sèrie, és possible realitzar prediccions sobre el seu comportament a curt o llarg termini.

Siguin  $x = x_1, x_2, \dots, x_T$  els valors passats d'una sèrie temporal i  $H$  l'horitzó de predicció, la tasca de predicció consisteix en obtenir els valors  $x = x_{T+1}, \dots, x_{T+H}$ , de mode que si  $\hat{x} = \hat{x}_{T+1}, \hat{x}_{T+2}, \dots, \hat{x}_{T+H}$  és el vector de valors reals de la sèrie, llavors els valors obtinguts han de minimitzar la següent equació: [Puerta 022]

$$E = \sum_{i=1}^H |x_{T+i} - \hat{x}_{T+i}| \quad (6.4)$$

Els mètodes tradicionals estadístics utilitzats per predir sèries temporals es basen en la construcció de funcions lineals a partir de les observacions passades recents per obtenir prediccions futures. Tot i això, l'aplicació dels mètodes tradicionals s'ha de fer tenint en compte algunes consideracions exigides pels

algoritmes. Per exemple, en el cas de l'ARIMA (*AutoRegressive Integrated Moving Average*) les sèries temporals han de ser estacionàries. Un altre problema que tenen aquests mètodes és que la seva aplicabilitat es limita a situacions en les què es disposa de suficients dades històriques i una estructura explicable. A més, aquests mètodes creen un model per cada sèrie temporal individual, la qual cosa no fa possible compartir l'aprenentatge entre casos similars. Per aquestes raons, durant els darrers anys han començat a guanyar popularitat les tècniques d'intel·ligència artificial, les quals permeten construir models globals que comparteixen el coneixement en múltiples sèries relacionades.

En aquest projecte es treballarà amb:

- Tècniques de predicció tradicionals
  - Mètode ARIMA estacional
  - Facebook Prophet
- Tècniques d'intel·ligència artificial
  - Xarxa neuronal LSTM (Long Short-Term Memory)

## **6.4 Metodologia**

La metodologia que es seguirà per tal de dur a terme aquestes prediccions és la següent:

1. Obtenció de les dades i anàlisi exploratòria
2. Predicció amb tres models diferents (ARIMA, Prophet Facebook i xarxa neuronal LSTM)
3. Avaluació dels resultats

En els següents apartats, s'expliquen aquests passos teòricament i es mostren els resultats que s'han obtingut.



# Obtenció de dades i anàlisi exploratòria

---

El sistema d'adquisició de dades a partir de les cafeteres que ha establert Cafès Cornellà li ha permès obtenir informació detallada dels seus clients. Cal, però, saber com tractar aquestes dades per poder extreure la informació més rellevant i alhora determinar quines són les dades que realment li interessin. Per poder resoldre aquest problema cal realitzar un anàlisi exploratòria de dades.

L'anàlisi exploratòria de dades (EDA) s'utilitza per analitzar conjunts de dades i resumir-ne les característiques principals per tal de respondre les preguntes que s'estiguin plantejant en funció del problema a resoldre. L'EDA generalment utilitza mètodes de visualització de dades i permet descobrir patrons, detectar anomalies, o comprovar una o més hipòtesis. D'aquesta manera proporciona una millor comprensió de les variables del conjunt de dades i les relacions entre elles.

L'objectiu principal de l'EDA és estudiar les dades abans de fer cap hipòtesi. Un cop completada l'EDA i extreure les idees, aquestes es poden utilitzar per a una anàlisi o modelització de dades més sofisticada, o fins i tot en tècniques d'aprenentatge automàtic. [IBM 022b]

## 7.1 EDA realitzat per Eurecat

En aquest cas s'ha aprofitat l'EDA realitzat per Eurecat el desembre de 2021. Les característiques més importants d'aquest estudi es resumeixen a continuació:

- Hipòtesi plantejada: saber si hi havia una alguna correlació entre el nombre de cafès setmanals amb la resta de variables.
- Dades d'entrada: les dades considerades van ser: cafès venuts les tres setmanes anteriors, cicles de neteja, alarmes de volumètric, reinicis de la cafetera, grup de preus i facturació del client, ubicació, horaris d'obertura, gamma i volum del cafè i dies festius.
- Resultats gràfics obtinguts de la correlació de les dades amb el nombre de cafès venuts: a la Figura 7.1 es mostra a l'eix x les variables estudiades

(dades d'entrada), i a l'eix y, la seva correlació amb els cafès setmanals venuts. Quan més propera és la correlació a 1, més relació hi ha entre les variables. Es pot veure que els *lags* (en aquest cas, els *lags* són els cafès venuts en setmanes anteriors) són els que estan més correlacionats.

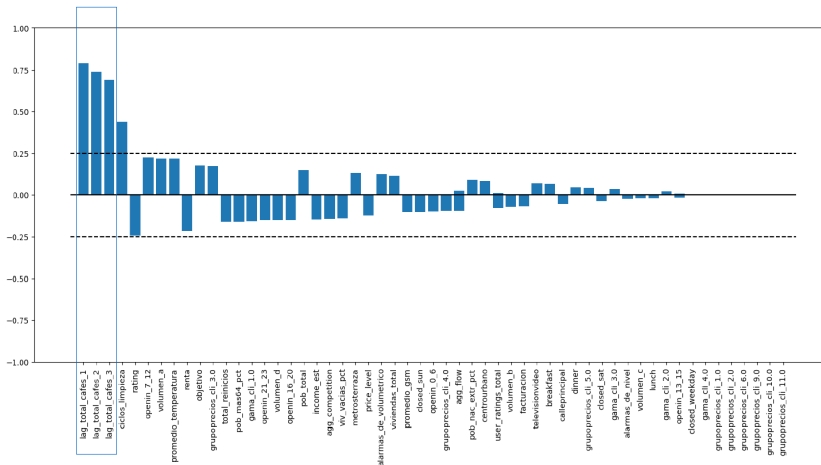


Figura 7.1: Resultats gràfics de l'EDA realitzat per Eurecat.  
Font: Cafès Cornellà

- **Conclusió:** La conclusió a la que va arribar Eurecat és que: “No hi havia variables molt correlacionades amb els cafès setmanals venuts. Si es consideraven els *lags* (cafès venuts en setmanes anteriors) eren de fet les variables més correlacionades”. És a dir, només el nombre de cafès venuts les setmanes anteriors estan relacionats amb el nombre de cafès setmanals venuts.

Donat que la cartera de clients actius actuals de l'empresa és molt similar a la del desembre 2021, es partirà d'aquest EDA per formular el nostre model de predicció. Per tant, les variables d'entrada que es tindran en compte a l'hora de fer el pronòstic seran la data i els cafès diaris venuts del client en qüestió, és a dir l'històric de cafès diaris. Aquestes dades es poden obtenir de la plataforma de Quality Espresso (veure Figura 5.2) en format csv.

## 7.2 Subconjunt de dades

Dins els diferents models de predicció que es definiran més endavant, les dades del fitxer d'entrada es separaran en dos subconjunts: *train\_data* i *test\_data*. Això es fa amb l'objectiu de poder comparar els models entre ells, per fer-ho s'executaran els algorismes de la següent forma:

1. S'ajustarà el model utilitzant el subconjunt *train\_data*
2. Es realitzarà la projecció de les dades amb un horitzó donat pel nombre d'observacions del subconjunt *test\_data*
3. Es compararan els valors reals del subconjunt *test\_data* amb els valors predits pel model





# Models de predicció

---

En aquest apartat es definiran teòricament els tres models que s'han utilitzat per fer l'estudi: ARIMA estacional, Prophet i LSTM.

## 8.1 ARIMA

La metodologia ARIMA (*AutoRegressive Integrated Moving Average*) va ser desenvolupada per Box i Jenkins l'any 1976 i des de llavors s'ha utilitzat en infinitat de models financers que requereixen l'anàlisi i estimació de sèries temporals. Els models es coneixen amb el nom genèric  $ARIMA(p, d, q)$ , mètode “autoregressiu integrat de mitjanes mòbils”, el qual deriva de les seves tres components: [IBM 022a]

- AR ( $p$ ): és la part del model autoregressiu, el paràmetre  $p$  és el nombre d'ordres autoregressives del model. Les ordres autoregressives especifiquen els valors previs de la sèrie utilitzats per predir els valors actuals. Per exemple, un ordre autoregressiu igual a 2 especifica que s'utilitzaran els valors de la sèrie corresponents a dos períodes de temps del passat per predir el valor actual.
- I ( $d$ ): és la part de diferenciació, el paràmetre  $d$  és l'ordre de diferenciació aplicat a la sèrie abans d'estimar els models. La diferenciació és necessària si hi ha tendències (les sèries amb tendències solen ser no estacionàries i el modelatge d'ARIMA assumeix l'estacionarietat) i s'utilitza per eliminar-ne l'efecte. El paràmetre  $d$  correspon al grau de la tendència de la sèrie (per exemple  $d=1$  correspon a les tendències lineals,  $d=2$  a les tendències quadràtiques...).
- MA ( $q$ ): és la part del model que correspon a la mitjana mòbil, la  $q$  és el nombre d'ordres de mitjana mòbil presents al model. Els ordres de mitjana mòbil especifiquen la manera com s'utilitzen les desviacions de la mitjana de la sèrie per als valors previs per tal de predir els valors actuals. Per exemple, els ordres de mitjana mòbil d'1 i 2 especifiquen que les desviacions respecte al valor mig de la sèrie de cadascun dels dos últims períodes de temps es tenen en compte al predir els valors actuals de la sèrie.

La metodologia de previsió (representada a la Figura 8.1), sovint anomenada Box-Jenkins, és la següent: [de Madrid 022]

1. Identificació. Identificar el possible model ARIMA que segueix la sèrie, cosa que requereix:
  - Decidir quines transformacions aplicar per convertir la sèrie observada en una sèrie estacionària. D'aquí obtindrem el paràmetre  $d$ , que ens indicarà les vegades que haurem diferenciat la sèrie per tal de que sigui estacionària.
  - Determinar un model ARMA per a la sèrie estacionària, és a dir, els ordres  $p$  i  $q$  de la seva estructura autoregressiva i de mitjana mòbil.
2. Estimació. Selecció provisionalment un model per a la sèrie estacionària, es passa a la segona etapa d'estimació on els paràmetres AR i MA del model s'estimen per màxima versemblança i se n'obtenen els errors estàndard i els residus del model.
3. Comprovació. Es comprova que els residus no tenen estructura de dependència i segueixen un procés de soroll blanc (el soroll blanc és un senyal aleatori que es caracteritza pel fet que els seus valors de senyal en dos instants de temps diferents no guarden correlació estadística). Si els residus mostren una estructura, es modifica el model per incorporar-la i es repeteixen les etapes anteriors fins a obtenir un model adequat.
4. Predicció. Un cop s'ha obtingut un model adequat es fan prediccions amb el mateix.

Sovint, les sèries temporals tenen un component estacional que es repeteix a totes les observacions, com el nostre cas, que hi ha un fort patró setmanal. Per fer front a l'estacionalitat, els processos ARIMA han estat generalitzats, establint els models SARIMA (Seasonal Autoregressive Integrated Moving Average Model). A l'apartat següent hem aprofundit en l'ús de models ARIMA amb estacionalitat.

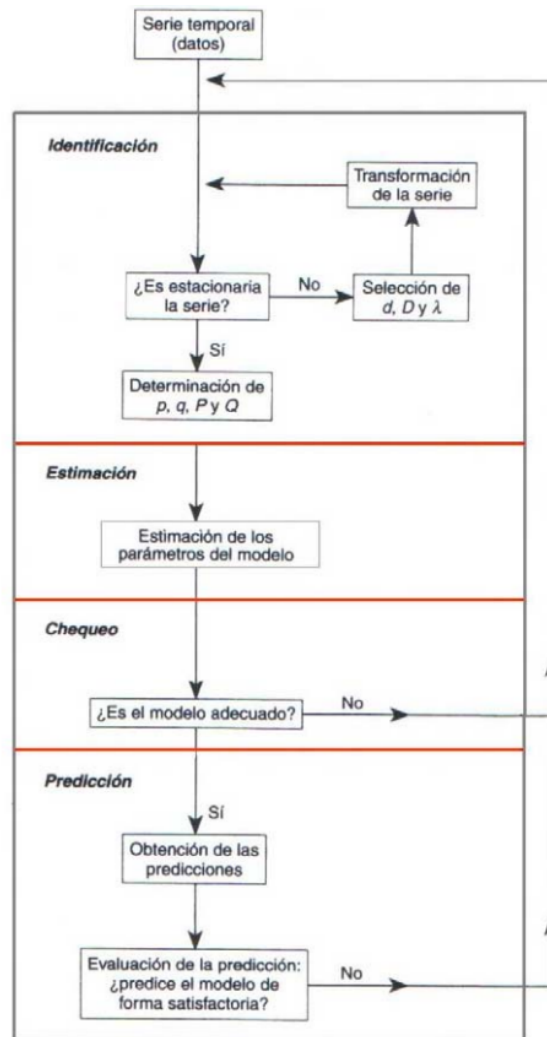


Figura 8.1: Metodología de previsió Box-Jenkins.  
Font: [de Madrid 022]

### 8.1.1 SARIMA

El model SARIMA inclou la contribució de l'estacionalitat a la previsió, la qual té una importància evident a l'hora de fer la previsió, i que el model ARIMA no és capaç de tenir en compte. L'addició de la estacionalitat afegeix robustesa al model ARIMA i es representa com  $SARIMA(p, d, q)(P, D, Q)_s$ . Les parts autoregressives (AR(p)), integrades (I(d)) i mitjanes mòbils (MA(q)) del model es mantenen com les de l'ARIMA. Els components d'autocorrelació estacional (P, D, Q) tenen els mateixos rols que els components no estacionals corresponents. No obstant, en el cas dels ordres estacionals, els valors de la sèrie actual es veuen afectats pels valors de la sèrie anterior separats per un o més períodes estacio-

nals. Aquest període estacional és el que correspon al valor de  $s$ . Per exemple, per observacions mensuals amb un patró anual  $s=12$ , per observacions diàries amb un patró setmanal  $s=7$ .

Quan s'hagi d'implementar aquest model de predicció, es buscaran de forma iterativa i automàtica els diferents paràmetres  $(p,d,q)$  i  $(P,D,Q)$ ; això es veurà en el capítol 10.

## 8.2 Prophet

L'any 2017 Facebook va alliberar una eina de codi obert, disponible per Python i R, anomenada Prophet. Aquesta llibreria permet construir models d'ajustament i pronòstics de sèries, utilitzant uns mètodes que ells denominen *curve fitting*. Aquest model pot ser formulat de la següent forma: [Sean J. Taylor 022]

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (8.1)$$

on:

- $y(t)$ : pronòstic.
- $g(t)$ : funció de la tendència que modela els canvis no periòdics a la sèrie temporal (canvis durant un llarg període de temps).
- $s(t)$ : representa els canvis periòdics (per exemple: estacionalitat setmanal i anual).
- $h(t)$ : efectes que es produeixen en dates irregulars, com ara les vacances o dies festius.
- $\varepsilon_t$ : terme d'error que representa els canvis que no s'assumeixen en el model.

### 8.2.1 Tendència

Prophet implementa dos possibles models per la tendència: el model de creixement saturat no lineal i el model lineal a trossos.

- Model de creixement saturat no lineal: aquest es representa com un model de creixement logístic. Aquesta configuració és útil quan la sèrie temporal té un límit en què els valors que s'estan modelant es saturen i no poden superar un valor màxim o mínim.

- Model lineal a trossos: per la previsió de sèries que no presenten un creixement saturat, un model lineal a trossos proporciona un model més útil. Aquesta és la configuració predeterminada de Prophet, la qual utilitza un conjunt d'equacions lineals a trossos amb pendents diferents entre els punts de canvi. Els "punts de canvi" són una idea incorporada a Facebook Prophet en què es suposa que la tendència de creixement pot estar present en tots els punts de les dades o es pot alterar. Prophet pot detectar automàticament els punts de canvi o bé es poden configurar. Per exemple, en el cas de recompte de casos per COVID-19, els punts de canvi podrien ser que els casos comencessin a disminuir després de la vacuna, o que els casos augmentessin quan aparegués una nova variant.

### 8.2.2 Estacionalitat

La funció d'estacionalitat es modelitza a través d'una sèrie de Fourier en funció del temps. La sèrie de Fourier és una eina matemàtica que permet obtenir informació d'una funció determinada mitjançant una transformació (entenen "transformació" al procés que redueix la complexitat d'una equació). La idea de la sèrie de Fourier és que qualsevol funció, generalment periòdica, es pot aproximar mitjançant funcions simples sinusoidals. De forma que quan més coincideix una ona simple amb les dades observades, més pes té en la determinació de la funció original. Prophet detecta automàticament un nombre òptim de termes de la sèrie. Tot i això, també hi ha l'opció de triar l'ordre de Fourier en funció de les necessitats del conjunt de dades en qüestió. També es pot escollir entre estacionalitat additiva i multiplicativa (tipus d'estacionalitats explicada a l'apartat 6).

### 8.2.3 Vacances

La funció de vacances permet a Facebook Prophet ajustar la previsió quan un dia és festiu o quan hi ha un esdeveniment important que pot canviar la previsió. Per això, cal una llista de dates i indicar si s'ha d'afegir o restar valor al pronòstic quan aquestes dates estan presents a la previsió. També hi ha la possibilitat d'identificar una sèrie de dies al voltant d'unes dates concretes (per exemple: el període entre Nadal i any nou, caps de setmana festius, accions com *Black Friday* o *CyberMonday*...).

### 8.2.4 Error

Finalment, el terme d'error  $\varepsilon_t$  representa qualsevol informació no contemplada pel model. Generalment, s'assumeix que  $\varepsilon_t$  segueix una distribució normal  $N$

$(0, \sigma^2)$  amb mitjana zero i variància desconeguda  $\sigma$  que varia en funció de les dades.

### 8.2.5 Avantatges respecte el mètode ARIMA

Respecte l'ARIMA el model Prophet proporciona els següents avantatges:

- Flexibilitat: podem adaptar-nos fàcilment a l'estacionalitat amb múltiples períodes i així poder fer diferents suposicions sobre les tendències. A diferència dels models ARIMA, les mesures no s'han d'espaiar regularment, i no necessitem interpolar els valors que falten per eliminar els valors atípics.
- El model de previsió té paràmetres fàcilment interpretables que es poden modificar per tal d'imposar les hipòtesis sobre la previsió, ja que és fàcil ampliar el model per incloure-hi nous components.

## 8.3 Xarxa neuronal LSTM

Les xarxes neuronals artificials són un model computacional inspirat en la forma que funciona el sistema nerviós dels animals. Consisteix en un conjunt d'unitats, anomenades neurones artificials, connectades entre si per transmetre's senyals. La informació d'entrada travessa la xarxa neuronal (on se sotmet a diverses operacions) produint uns valors de sortida. [Viquipèdia 022b]

Un tipus de xarxes neuronals artificials són les recurrents (RNN), i dins d'aquestes trobem les xarxes neuronals LSTM (Long Short Term Memory). Un dels models de predicció que s'utilitzarà en aquest projecte és la LSTM.

### 8.3.1 Xarxa neuronal RNN

Les RNNs són un tipus de xarxes capaces de reconèixer i predir seqüències de dades al llarg del temps, com ara textos, genomes, o sèries numèriques. Aquest tipus de xarxes es fonamenten en bucles que permeten que la sortida de la xarxa o d'una part d'ella en un moment donat serveixi com a entrada de la pròpia xarxa en el següent moment. El bucle que connecta la xarxa amb si mateixa és el mecanisme que permet que la xarxa tingui memòria.

A la Figura 8.2 es mostra l'arquitectura d'una RNN bàsica, on  $A$  es la xarxa neuronal,  $X_t$  l'entrada de la xarxa i  $h_t$  la sortida de la xarxa.

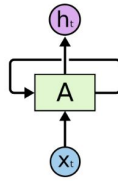


Figura 8.2: Arquitectura RNN bàsica.  
Font: [Mañas 022]

Les RNNs també es poden veure com a múltiples còpies de la mateixa xarxa, cadascuna passant informació a la seva successora. Això es pot veure a la Figura 8.3, on en cada moment del temps  $t$ , la xarxa rep com a entrada tant  $X_t$  com la seva pròpia sortida  $h_{t-1}$  a l'instant  $t - 1$ .

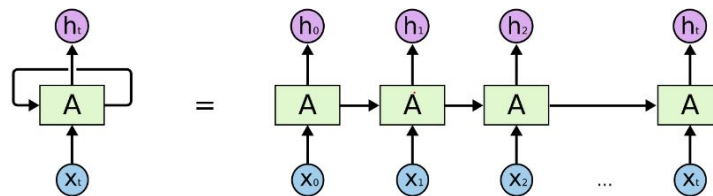


Figura 8.3: Arquitectura RNN estesa.  
Font: [Mañas 022]

Les RNNs tenen un inconvenient conegut com a problema de l'esvaïment del gradient, és a dir, tenen dificultats per aprendre dependències de llarg abast. Quan ens movem cap enrere a la xarxa i calculem els gradients de pèrdua (error) respecte als pesos, els gradients tendeixen a ser cada vegada més petits. Això significa que les neurones a les capes anteriors aprenen molt lentament en comparació amb les neurones a les capes posteriors a la jerarquia. Això és un gran inconvenient a l'hora de fer pronòstics de sèries temporals.

Afortunadament, aquest problema es resol mitjançant la xarxes neuronals LSTM (Long Short Term Memory). Aquestes són capaces d'aprendre dependències a llarg termini.

### 8.3.2 Xarxa neuronal LSTM

A diferència de les RNN estàndard, que tenen un mòdul de repetició amb una sola capa de xarxa neuronal (representada amb groc a la Figura 8.4), les LSTM tenen un mòdul de repetició amb una estructura més complexa on hi ha quatre

capes que interactuen (veure a la Figura 8.5). Aquestes permeten que els models LSTM tinguin la capacitat per eliminar o afegir informació a la memòria.

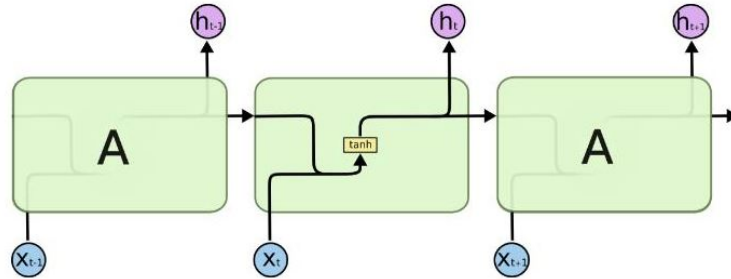


Figura 8.4: Arquitectura model RNN.

Font: [Clic 022]

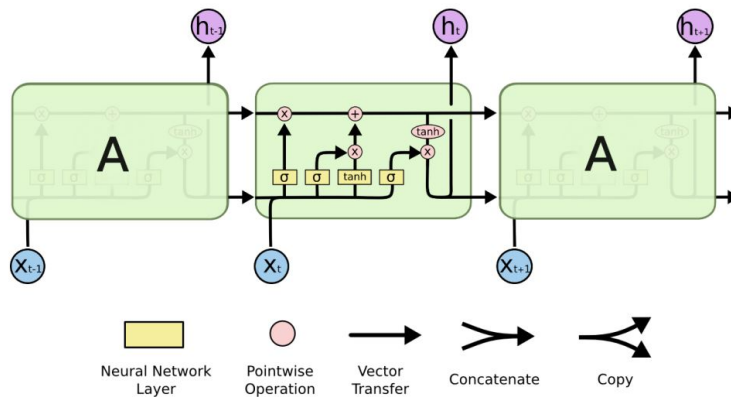


Figura 8.5: Arquitectura model LSTM.

Font: [Mañas 022]

### 8.3.2.1 Estructura del model LSTM

Per tal de dur a terme les operacions d'eliminar o afegir informació a la memòria en un moment donat hi ha unes estructures internes, anomenades portes, que ho regulen. Les portes tenen diverses activacions anomenades sigmoïdes, que contenen valors que van de 0 a 1. Això permet decidir la quantitat d'informació que el component ha de deixar passar. Si les dades es multipliquen per 1, el valor d'aquestes dades continua sent el mateix. Tot i això, si les dades es multipliquen per 0, el valor es converteix en 0 i no es deixa passar res.

La LSTM té tres portes: la d'oblit, la d'entrada i la de sortida (veure Figura 8.6). A continuació s'explicarà pas a pas com és el funcionament intern, el recorregut de la informació en una LSTM i quina funció tenen aquestes portes: [Santana 022]



1. El primer que fa la xarxa LSTM és decidir quina informació rebutjarà de la memòria. Aquesta decisió és presa per la porta anomenada **porta d'oblit** (*Forget gate*). Com es pot apreciar a la Figura 8.6,  $h_{t-1}$  i  $x_t$  es concatenen i el resultat és l'entrada a la petita xarxa que conforma la porta d'oblit. El resultat de la xarxa servirà per decidir si l'estat de la memòria es deixarà com està o s'alterarà eliminant algun element.
2. El següent pas és decidir quina informació nova s'emmagatzemarà a l'estat de memòria. Això es fa en dues etapes. Primer, una porta anomenada **porta d'entrada** (*Input Gate*) decideix quins valors actualitzarem. Llavors una petita xarxa amb  $\tanh$  crea un vector de valors nous candidats, que podrien afegir-se a l'estat. Al següent pas, es combinen els dos resultats per crear una actualització d'estat.
3. Ara ja es pot actualitzar l'estat de la memòria canviant  $C_{t-1}$  pel nou estat  $C_t$ . Per fer-ho, només cal aplicar les operacions de cada porta.
4. Per acabar, s'ha de decidir quina informació sortirà per la **porta de sortida** de la cèl·lula LSTM. Aquesta s'obté mitjançant el producte de dos elements. El primer és la sortida de la xarxa amb sigmoide, que servirà per decidir quins elements de la memòria es combinaran. El segon element serà el filtratge de dades des de la memòria per una  $\tanh$  (els valors han d'estar entre -1 i 1). Aquests dos elements es multiplicaran donant com a resultat la nova sortida de la cèl·lula.

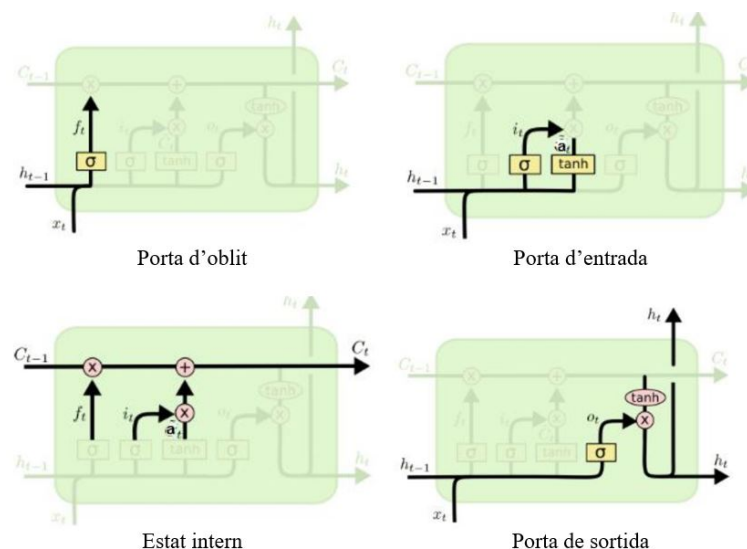


Figura 8.6: Portes i estat intern d'una xarxa LSTM.

Font: [Clic 022]

Afortunadament, per utilitzar aquest tipus de xarxa no cal implementar-les des de zero, sinó que es poden utilitzar diferents frameworks. Això es veurà en el capítol 10 on s'utilitza la llibreria *Keras*.

# Mètriques d'avaluació i càlcul d'error

---

En termes generals, un model de sèrie temporal es pot considerar com un model de regressió específic, per tant, es pot avaluar amb les mètriques d'aquest tipus de model. A continuació es defineixen les mètriques estadístiques per l'avaluació dels models d'aquest treball. En totes elles, la notació considerada és  $y_j$  com les dades reals (que corresponen al subconjunt de *test\_data* tal i com s'ha explicat a l'apartat 7),  $\hat{y}_j$  les dades pronosticades i  $n$  el total d'observacions.

- Error quadràtic mitjà (MSE): calcula l'error quadràtic mitjà entre el valor real i el pronosticat.

$$MSE = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (9.1)$$

- Error absolut mitjà (MAE): és el valor absolut de la diferència entre el valor real i el valor pronosticat.

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (9.2)$$

Per aquestes dues mètriques (MSE i MAE) el rang està entre 0 i infinit. Si els valors predits són iguals amb els valors reals, és igual a 0, el que significa que és un model perfecte. Per tant, com més gran és el valor, pitjor s'ajusta el model.

- Arrel de l'error quadràtic mitjà (RMSE): és una de les mètriques més habituals per avaluar un model de regressió, ja que mesura la quantitat d'error que hi ha entre dos subconjunts de dades, en el cas del model de predicció proporciona la diferència entre el valor pronosticat pel model i el valor real.

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (9.3)$$

- $R^2$ : el rang del valor  $R^2$  és d'entre 0 i 1. Com més proper és el valor de  $R^2$  a 1, més ajustat és el model. Si els valors predits són iguals amb els valors

reals, és igual a 1.

$$R^2 = 1 - \frac{\sum_j (y_j - \hat{y}_j)^2}{\sum_j (y_j - \bar{y}_j)^2} \quad (9.4)$$

on  $\bar{y}_j$  és la mitjana de le dades observades:

$$\bar{y}_j = \frac{1}{n} \sum_{t=1}^j y_j \quad (9.5)$$

# Implementació i resultats obtinguts

---

En aquest apartat es presenten com s'han implementat i els resultats que s'han obtinguts per cadascun dels models considerats. Per presentar-ho, s'ha fet a partir del cas concret d'un sol client. El mateix codi que es presenta als següents apartats ha sigut executat per dades de quatre clients diferents.

En aquest cas, per tots els models s'aplicarà la mateixa metodologia:

- Es treballarà amb el fitxer 'cafes.csv', les dades del qual s'extreuen de la plataforma actual on es recull la informació que prové de les cafeteres (veure Figura 5.2).
- Es considerarà el període que va des del 01/03/2021 fins al 31/05/2022. D'aquest, es faran dos subconjunts de dades, el *test\_data* on hi haurà les últimes 21 observacions (de l'11/05/2022 al 31/05/2022) i el *train\_data* on hi haurà la resta.

Cal remarcar que la pandèmia ha limitat el nombre de dades que es podien considerar. Per una banda, durant la pandèmia el sector de la restauració estava tancat i per tant no es disposa d'informació d'aquest període. Per altra banda, la informació d'abans de la pandèmia segueix una tendència diferent a la del darrer any i això afectaria al nostre model. En el gràfic de la Figura 10.1, es representa l'històric de cafès diaris d'un client per tal de corroborar el que s'acaba de comentar.

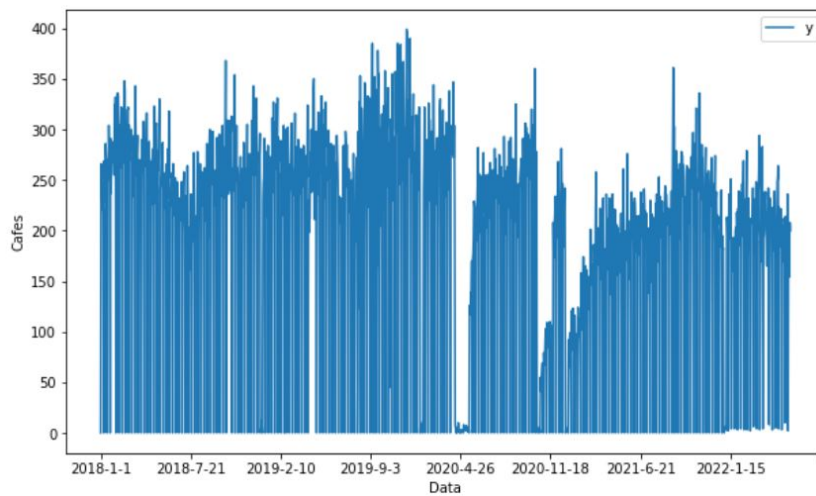


Figura 10.1: Cafès servits des de gener de 2018 fins a maig de 2022.

Font: elaboració pròpia

Així doncs, només es tindrà en compte el període comentat anteriorment. Si grafiquem aquest període, podem veure que apareixen alguns patrons distingibles, com ara un patró d'estacionalitat setmanal evident.

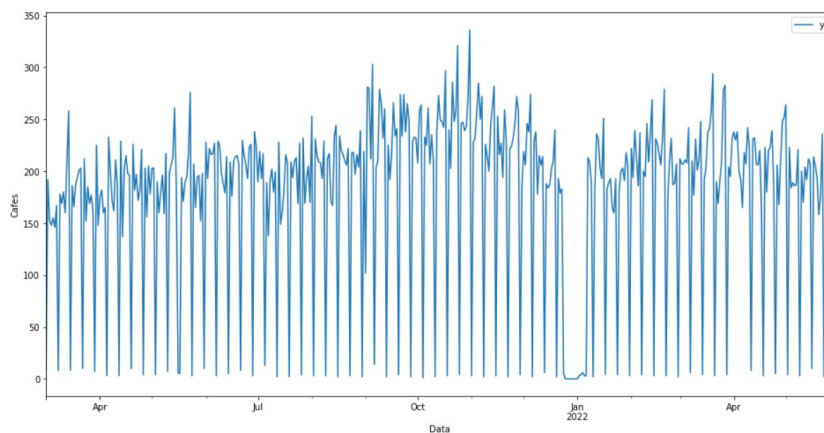


Figura 10.2: Carregar les dades.

Font: elaboració pròpia

A continuació, es presentarà, per cada model, els passos que s'han realitzat acompanyats del codi que s'ha executat.

## 10.1 ARIMA estacional

A continuació s'explica tot el procés que s'ha seguit per implementar el model ARIMA estacional (SARIMA).

### 1. Importar llibreries i carregar les dades

En primer lloc s'importen les llibreries necessàries, com ara *itertools*, *pandas*, *numpy*, *matplotlib* i *statsmodels* (veure Figura 10.3).

```
# Importar llibreries
import warnings
import itertools
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
import statsmodels.tsa as sm
from datetime import datetime
import plotly.graph_objects as go
import os
import seaborn as sns
from statsmodels.tsa.seasonal import DecomposeResult
import itertools
import statsmodels.api as sm
from sklearn.metrics import mean_squared_error, r2_score
```

Figura 10.3: Importar les llibreries.  
Font: elaboració pròpia

A continuació, es carreguen les dades i es divideixen en dos subconjunts tal i com s'havia comentat al capítol 7 (*train\_data* i *test\_data*). El subconjunt *test\_data* tindrà les dades corresponents a les darreres tres setmanes de dades.

### 2. Selecció de paràmetres pel model ARIMA estacional

Quan busquem ajustar les dades de sèries temporals amb un model ARIMA estacional, el nostre primer objectiu és trobar els valors dels paràmetres  $(p, d, q)(P, D, Q)_s$  que optimitzen la mètrica d'interès. Hi ha diferents programes i bones pràctiques per aconseguir aquest objectiu.

En aquest cas es resoldrà amb Python i es farà un codi per automatitzar la cerca dels paràmetres òptims  $(p, d, q)$  i  $(P, D, Q)$ . Per fer-ho, s'utilitzarà una 'grid search' per explorar de manera iterativa diferents combinacions de paràmetres. En el següent fragment de codi s'iteren combinacions de paràmetres i s'utilitza la funció SARIMAX de *statsmodels* per adaptar-se al model ARIMA estacional corresponent. L'argument *order* especifica els paràmetres  $(p, d, q)$ , mentre que l'argument *seasonal\_order* especifica el component estacional  $(P, D, Q, s)$  del model ARIMA estacional, on *s* serà 7 ja que les dades són diàries amb patrons setmanals (veure Figura 10.4).

```
# Definició dels rangs dels paràmetres q i p
q = d = range(0, 2)
# Definició del rang del paràmetre d
p = range(0, 2)

# Generació de les diferents combinacions de p, d i q
pdq = list(itertools.product(p, d, q))

# Generació de les diferents combinacions estacionals P, Q i D
seasonal_pdq = [(x[0], x[1], x[2], 7) for x in list(itertools.product(p, d, q))]

print('Exemples de combinacions del model Seasonal ARIMA:')
print('SARIMA: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMA: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMA: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMA: {} x {}'.format(pdq[2], seasonal_pdq[4]))

Exemples de combinacions del model Seasonal ARIMA:
SARIMA: (0, 0, 1) x (0, 0, 1, 7)
SARIMA: (0, 0, 1) x (0, 1, 0, 7)
SARIMA: (0, 1, 0) x (0, 1, 1, 7)
SARIMA: (0, 1, 0) x (1, 0, 0, 7)
```

Figura 10.4: Generació de diferents combinacions dels paràmetres.

Font: elaboració pròpia

Per a cada combinació de paràmetres, es busca el coeficient AIC (*Akaike Information Criterion*). Aquest coeficient és un estimador de l'error de predicció i, per tant, la qualitat relativa dels models estadístics per a un conjunt de dades determinat. [Viquipèdia 022a]. Així doncs, ens interessa trobar el model que tingui el valor AIC més baix. A la Figura 10.5 es mostra la sortida en què s'indica la combinació del model SARIMA i el coeficient AIC.



```
# Obtenció de l'AIC per cada combinació del model SARIMA
warnings.filterwarnings("ignore") # specify to ignore warning messages

AIC = []
SARIMAX_model = []
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(train_data,
                                           order=param,
                                           seasonal_order=param_seasonal,
                                           enforce_stationarity=False,
                                           enforce_invertibility=False)

            results = mod.fit()
            AIC.append(results.aic)
            SARIMAX_model.append([param, param_seasonal])

            print('SARIMA{}x{} - AIC:{}'.format(param, param_seasonal, results.aic))
        except:
            continue

SARIMA(0, 0, 0)x(1, 1, 0, 7) - AIC:4441.051183557463
SARIMA(0, 0, 0)x(1, 1, 1, 7) - AIC:4357.329863290296
SARIMA(0, 0, 1)x(0, 0, 0, 7) - AIC:5565.829244031389
SARIMA(0, 0, 1)x(0, 0, 1, 7) - AIC:5101.546441925124
SARIMA(0, 0, 1)x(0, 1, 0, 7) - AIC:4474.54862362797
SARIMA(0, 0, 1)x(0, 1, 1, 7) - AIC:4303.602024506261
SARIMA(0, 0, 1)x(1, 0, 0, 7) - AIC:4488.032916897682
SARIMA(0, 0, 1)x(1, 0, 1, 7) - AIC:4377.559815450107
SARIMA(0, 0, 1)x(1, 1, 0, 7) - AIC:4377.408963903472
SARIMA(0, 0, 1)x(1, 1, 1, 7) - AIC:4277.7573053195565
```

Figura 10.5: Obtenció de l'AIC per cada combinació del model SARIMA.  
Font: elaboració pròpia

En aquest cas i tal i com es mostra a la Figura 10.6, el valor AIC més baix es dona amb la següent combinació SARIMA(1,0,1)x(1,1,1,7). Per tant, es considera aquesta com l'opció òptima entre tots els models que s'han considerat.

```
# Mostrar el model SARIMA amb un AIC més petit
min(AIC)
print('SARIMA{}x{} - AIC:{}'.format(SARIMAX_model[AIC.index(min(AIC))][0],
                                   SARIMAX_model[AIC.index(min(AIC))][1], min(AIC)))

SARIMA(1, 0, 1)x(1, 1, 1, 7) - AIC:4212.474555041986
```

Figura 10.6: Model SARIMA amb un AIC més baix.  
Font: elaboració pròpia

### 3. Ajustament del model ARIMA estacional

A l'apartat anterior, s'han identificat el conjunt de paràmetres  $(p, d, q)$  i  $(P, D, Q)$  que donaven el model que s'ajustava millor a la sèrie temporal. Ara es passa a analitzar el model amb aquests paràmetres òptims (veure Figura 10.7).

```
# Let's fit this model
mod = sm.tsa.statespace.SARIMAX(train_data,
                                order=SARIMAX_model[AIC.index(min(AIC))][0],
                                seasonal_order=SARIMAX_model[AIC.index(min(AIC))][1],
                                enforce_stationarity=False,
                                enforce_invertibility=False)

results = mod.fit()
```

Figura 10.7: Ajustament del model SARIMA amb els paràmetres òptims.  
Font: elaboració pròpia

En primer lloc, amb la comanda *summary* podem obtenir un resum dels resultats del model en forma de taula. Aquesta torna gran quantitat d'informació. Cal posar atenció a la columna de coeficients (coef), la qual mostra el pes (és a dir, la importància) de cada component i com influeix cadascuna en la sèrie temporal. La columna  $P > |z|$  ens informa de la importància de cada pes de característica (veure Figura 10.8). En aquest cas, cada pes té un valor  $p$  inferior o proper a 0,05, per la qual cosa és raonable tenir-los tots en el nostre model. [Holicka 022].

```

=====
Statespace Model Results
=====
Dep. Variable:          y          No. Observations:      433
Model:                SARIMAX(1, 0, 1)x(1, 1, 1, 7)  Log Likelihood        -2101.237
Date:                  Tue, 07 Jun 2022             AIC                   4212.475
Time:                  19:43:24                    BIC                   4232.640
Sample:                03-01-2021                   HQIC                  4220.447
                    - 05-07-2022
Covariance Type:      opg
=====
              coef  std err          z      P>|z|      [0.025   0.975]
-----
ar.L1          0.9176    0.016     57.905    0.000     0.887     0.949
ma.L1         -0.6173    0.032    -19.151    0.000    -0.680    -0.554
ar.S.L7         0.0963    0.034     2.851    0.004     0.030     0.163
ma.S.L7        -1.0503    0.028   -37.051    0.000    -1.106    -0.995
sigma2         1225.3244    55.510    22.074    0.000   1116.526   1334.123
=====
Ljung-Box (Q):                25.47   Jarque-Bera (JB):                1122.06
Prob(Q):                       0.96   Prob(JB):                          0.00
Heteroskedasticity (H):        2.67   Skew:                               -0.30
Prob(H) (two-sided):           0.00   Kurtosis:                          11.01
=====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

```

Figura 10.8: Resultats obtinguts a partir de la comanda *summary*.  
Font: elaboració pròpia

En segon lloc, utilitzant la comanda *plot\_diagnostics* es generen diagnòstics del model que permeten analitzar qualsevol comportament inusual. Cal assegurar-nos que els residus del nostre model no estiguin correlacionats, i que es distribueixin normalment amb una mitjana zero. Si el model SARIMA no compleix aquestes propietats, és un indicatiu de que el model es pot millorar.

En el nostre cas, tal i com es pot veure a la Figura 10.9, el model suggereix que els residus del model es distribueixen segons la normal, basant-nos en:

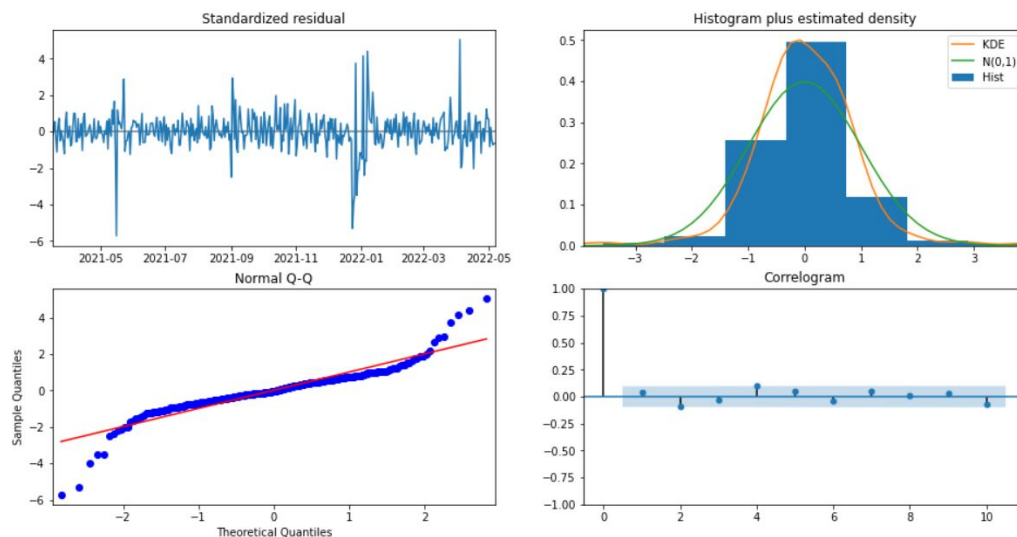


Figura 10.9: Resultats obtinguts a partir de la comanda `plot_diagnostics`.  
Font: elaboració pròpia

- Els residus al llarg del temps (gràfic superior esquerre) no mostren cap estacionalitat evident i semblen ser soroll blanc. Això es confirma amb el gràfic d'autocorrelació (gràfic inferior dreta), que mostra que els residus de la sèrie temporal tenen una baixa correlació amb les versions retardades de si mateix.
- En el gràfic de la part superior dreta, veiem que la línia taronja KDE (*Kernel Density Estimation*) té una forma similar a la línia  $N(0,1)$  (on  $N(0,1)$  és la notació estàndard per a una distribució normal amb mitjana 0 i desviació estàndard d'1). Aquesta és una indicació de que els residus es distribueixen normalment.
- El diagrama q-q (gràfic inferior esquerra) mostra que la distribució ordenada de residus (punts blaus) segueix la tendència lineal de les mostres preses d'una distribució normal estàndard amb  $N(0, 1)$ . De nou, això és un fort indicatiu que els residus es distribueixen normalment.

Aquestes observacions ens porten a concloure que el nostre model produeix un ajust satisfactori que ens podria ajudar a entendre les dades de les nostres sèries temporals i preveure els valors futurs.

Tot i que l'ajust és satisfactori, alguns paràmetres del nostre model ARIMA estacional es podrien canviar per millorar l'ajust del model. Per exemple, es podria ampliar el rang de combinacions de paràmetres  $(d, p, q)$  i  $(D, P, Q)$ , per tal que a l'analitzar més models potser se'n podrien trobar de millors.

#### 4. Validació de les previsions

Arribats a aquest punt, ja podem utilitzar el model obtingut per fer les previsions. En primer lloc, s'entrena el model i es fa la predicció per les dates del subconjunt del *test\_data*. D'aquesta manera, un cop obtinguda la previsió, es podran comparar les dades obtingudes amb les reals i veure com s'ajusta el model a la realitat.

Les comandes *get\_prediction()* i *conf\_int()* ens permeten obtenir els valors i els intervals de confiança associats a les previsions de la sèrie temporal. L'argument *dynamic = False* indica que les previsions en cada punt es generen utilitzant tot l'historial fins aquell moment (veure Figura 10.10).

```
# Predicció pel període de dates del subconjunt test_data
pred = results.get_prediction(start=inici, end=end, dynamic=False, full_results=True)
pred_ci = pred.conf_int()
```

Figura 10.10: Predicció pel període de dates del subconjunt *test\_data*.  
Font: elaboració pròpia

A la Figura 10.11 es mostra la representació gràfica de la predicció, el nombre de tasses reals es representa de color blau i les pronosticades de color taronja. A la Figura 10.12 podem veure els valors numèrics d'aquesta predicció.

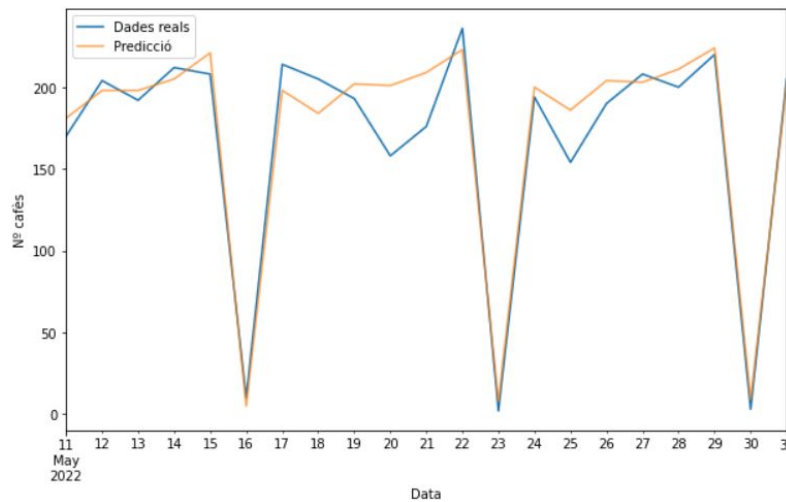


Figura 10.11: Representació gràfica del model de predicció ARIMA.  
Font: elaboració pròpia

```
# Valors de la predicció
round(pred.predicted_mean)

2022-05-11    181.0
2022-05-12    198.0
2022-05-13    198.0
2022-05-14    205.0
2022-05-15    221.0
2022-05-16     5.0
2022-05-17    198.0
2022-05-18    184.0
2022-05-19    202.0
2022-05-20    201.0
2022-05-21    209.0
2022-05-22    223.0
2022-05-23     8.0
2022-05-24    200.0
2022-05-25    186.0
2022-05-26    204.0
2022-05-27    203.0
2022-05-28    211.0
2022-05-29    224.0
2022-05-30     9.0
2022-05-31    201.0
Freq: D, dtype: float64
```

Figura 10.12: Resultats de la predicció amb el model SARIMA.  
Font: elaboració pròpia

Per tal de quantificar l'exactitud de les nostres previsions, s'utilitzaran les mètriques d'avaluació definides al capítol 9. A la Figura 10.13 podem veure els resultats que s'han obtingut (MAE: 12.9; MSE: 227.5; RMSE: 16.6;  $R^2$ :0.94).

```
# Mètriques d'avaluació
import sklearn.metrics as metrics
import numpy as np

mae = metrics.mean_absolute_error(test_data, round(pred.predicted_mean))
mse = metrics.mean_squared_error(test_data, round(pred.predicted_mean))
rmse = np.sqrt(mse)
r2 = metrics.r2_score(test_data, round(pred.predicted_mean))

print("Resultats:")
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)

Resultats:
MAE: 12.904761904761905
MSE: 277.4761904761905
RMSE: 16.65761659050269
R-Squared: 0.9429149763574411
```

Figura 10.13: Resultat mètriques d'avaluació.  
Font: elaboració pròpia

## 5. Predicció de valors futurs

Finalment, es presenta com es poden preveure valors futurs. Per fer-ho, s'han d'utilitzar totes les dades disponibles (i no fer subconjunts de dades) i es segueix el mateix procediment que s'ha explicat en els punts anteriors canviant la comanda `get_prediction()` per `get_forecast()`, on cal indicar el nombre d'observacions que es volen predir.

Tant les previsions com l'interval de confiança associat es poden utilitzar per entendre millor la sèrie temporal i preveure què podem esperar. A mesura que es preveu més endavant en el futur, és natural que tinguem menys confiança en els valors que s'obtenen.

## 10.2 Prophet

En aquest apartat, s'explicarà el procediment seguit per implementar el model Prophet.

### 1. Importar llibreries i carregar les dades

El primer pas, igual que en el model anterior, és importar les llibreries necessàries. En aquest cas serà necessària la llibreria `fbprophet`. Aquesta es basa en el llenguatge de programació STAN, i per això abans de carregar `fbprophet`, cal instal·lar l'entorn de `pystan` "wrapper" a STAN.

També es carregaran altres llibreries com ara: `matplotlib`, `pandas`, `matplotlib` o `numpy` (veure Figura 10.14).

```
!pip install pystan==2.19.1.1
!pip install prophet
!pip install pandas

import pandas as pd
from fbprophet import Prophet
import matplotlib.pyplot as plt
import matplotlib as mpl
import matplotlib.patches as mpatches
import numpy as np
from datetime import datetime, timedelta
import seaborn as sns
import warnings
import sklearn.metrics as metrics
warnings.filterwarnings('ignore')
```

Figura 10.14: Instal·lar i importar llibreries.  
Font: elaboració pròpia

A continuació, es llegeixen les dades de la nostra sèrie temporal, per fer-ho carreguem el mateix fitxer csv que s'ha carregat pel model ARIMA estacional.

La biblioteca Prophet espera com a entrada un `DataFrame` amb una columna que contingui la informació del temps i una altra columna que contingui la

mètrica que volem preveure. És important destacar que s'espera que la columna de temps sigui del tipus *datetime*, així que ens assegurem de convertir-la en aquest format (veure Figura 10.15). Prophet també imposa la condició estricta de que les columnes d'entrada s'anomenin *ds* (la columna del temps) i *y* (la columna mètrica, en el nostre cas correspon al nombre de tasses de cafè), així que canviem el nom de les columnes del nostre *DataFrame*. Com s'ha comentat anteriorment, es dividiran les dades amb dos subconjunts (train i test).

```
# Llegir dades del csv
df = pd.read_csv('cafes.csv', sep=',')
df.head()
# Convertim la columna temps a datetime
df['ds'] = pd.to_datetime(df['ds'], format='%Y-%m-%d')
```

Figura 10.15: Carregar les dades i posar-les amb el format adequat.  
Font: elaboració pròpia

## 2. Predicció amb el model Prophet

Un cop carregades les llibreries i les dades, es comença creant l'objecte *my\_model*, amb el qual s'incialitza el model Prophet. A continuació, es crida el fitxer que volem que serveixi com a entrada per ajustar el model. A més, Prophet ens permet especificar una sèrie d'arguments, com ara l'interval d'incertesa establint el paràmetre *interval\_width*. (veure Figura 10.16)

```
# Crear el model amb un interval d'incertesa del 95%
my_model = Prophet(interval_width=0.95)

my_model.fit(train)
```

Figura 10.16: Creació del model Prophet.  
Font: elaboració pròpia

Per obtenir previsions de la sèrie temporal, hem de proporcionar a Prophet un nou *DataFrame* que contingui una columna *ds* amb les dates per a les quals volem fer prediccions. Per fer-ho, s'utilitza la funció *future\_dataframe*, en aquesta cal indicar la freqüència de la sèrie i el nombre de valors que es vol predir. Remarcar que quan es treballa amb Prophet és important tenir en compte la freqüència de les sèries temporals, en aquest cas estem treballant amb dades diàries, i és per això que s'ha especificat la freqüència 'D'. Per tant, en aquest cas a la comanda *make\_future\_dataframe* s'especifica que es vol fer una predicció de 21 dies.



```
# Predicció de 21 dies
future_dates = my_model.make_future_dataframe(periods=21, freq='D')
future_dates.tail()
```

	ds
452	2022-05-27
453	2022-05-28
454	2022-05-29
455	2022-05-30
456	2022-05-31

Figura 10.17: Creació del *DataFrame* amb les dates que es volen predir.  
Font: elaboració pròpia

El *DataFrame* amb les dates futures serà el que s'utilitzarà com a entrada per la comanda *predict*. El resultat que obtindrem serà un nou *DataFrame* amb diferents columnes (veure Figura 10.18):

- *ds*: data del valor previst
- *yhat*: el valor previst de la nostra mètrica (*yhat* és una notació que s'utilitza tradicionalment per representar els valors predits d'un valor *y*), en aquest cas nombre de cafès
- *yhat\_lower*: el límit inferior de les nostres previsions
- *yhat\_upper*: el límit superior de les nostres previsions

```
# Resultats obtinguts
forecast = my_model.predict(future_dates)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

	ds	yhat	yhat_lower	yhat_upper
452	2022-05-27	206.079032	124.300329	285.818510
453	2022-05-28	209.133921	112.719194	294.106869
454	2022-05-29	224.547425	136.306096	313.279524
455	2022-05-30	9.746278	-77.749292	93.389799
456	2022-05-31	209.098089	125.592118	295.740119

Figura 10.18: Resultats obtinguts de la comanda *predict*.  
Font: elaboració pròpia



Prophet també ofereix una per graficar els resultats. El resultat que s'obté és un gràfic on es representa amb negre els valors observats de la sèrie temporal (és a dir, els valors reals) i amb una línia blava els valors de la predicció (veure Figura 10.19).

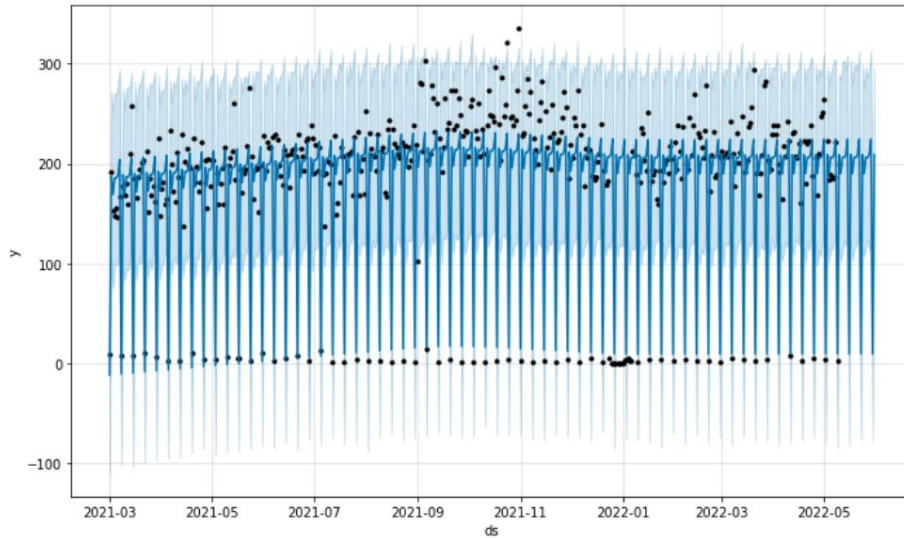


Figura 10.19: Representació gràfica del model de predicció Prophet.  
Font: elaboració pròpia

Si només ens centrem en el període de dates del *test\_data* els resultats obtinguts són els que es mostren a la Figura 10.20, on es representa amb color blau les dades reals i amb color taronja les dades que s'han predit. Els valors numèrics de la predicció són els que es mostren a la Figura 10.21.

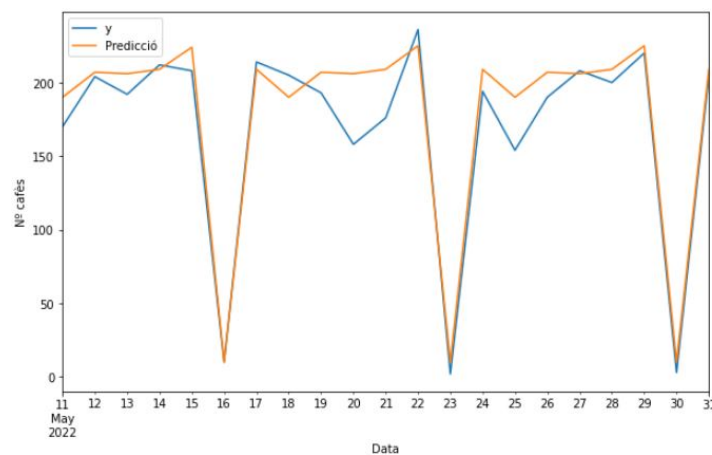


Figura 10.20: Representació del model de predicció Prophet.  
Font: elaboració pròpia

```
# Valors de la predicció  
res[['ds', 'Predicció']]
```

	ds	Predicció
436	2022-05-11	190.0
437	2022-05-12	207.0
438	2022-05-13	206.0
439	2022-05-14	209.0
440	2022-05-15	224.0
441	2022-05-16	10.0
442	2022-05-17	209.0
443	2022-05-18	190.0
444	2022-05-19	207.0
445	2022-05-20	206.0
446	2022-05-21	209.0
447	2022-05-22	225.0
448	2022-05-23	10.0
449	2022-05-24	209.0
450	2022-05-25	190.0
451	2022-05-26	207.0
452	2022-05-27	206.0
453	2022-05-28	209.0
454	2022-05-29	225.0
455	2022-05-30	10.0
456	2022-05-31	209.0

Figura 10.21: Valors de la predicció amb el model Prophet.  
Font: elaboració pròpia

Una altra funcionalitat de Prophet és que pot retornar els components de les prediccions. Això pot ajudar a revelar patrons setmanals i anuals de la sèrie temporal que contribueixen als valors globals previstos.

El primer gràfic mostra que el volum mensual de cafès venuts s'ha mantingut amb una tendència lineal durant els darrers 5 mesos. En el segon gràfic podem veure el patró setmanal, on es veu clarament quin és el dia festiu de la setmana (veure Figura 10.22).

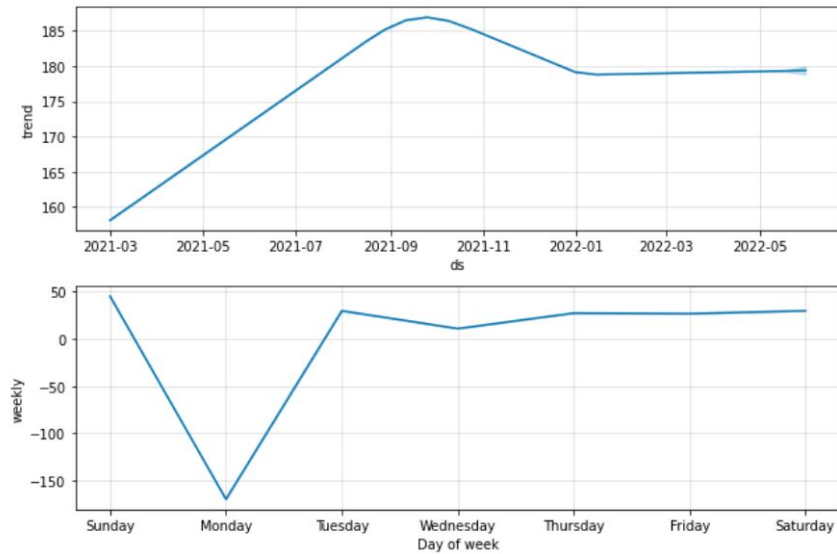


Figura 10.22: Components anuals i setmanals de la predicció.  
Font: elaboració pròpia

### 3. Validació del model

Per tal de quantificar l'exactitud de les prediccions, s'utilitzaran les mètriques d'avaluació definides en el capítol 9. A la Figura 10.23 podem veure els resultats que s'han obtingut (MAE: 13.6; MSE: 327.6; RMSE: 18.1;  $R^2$ :0.93).

```
# Mètriques d'avaluació
mae = metrics.mean_absolute_error(test.y, round(forecast.yhat[len(forecast.yhat)-len(test):]))
mse = metrics.mean_squared_error(test.y, round(forecast.yhat[len(forecast.yhat)-len(test):]))
rmse = np.sqrt(mse)
r2 = metrics.r2_score(test.y, round(forecast.yhat[len(forecast.yhat)-len(test):]))

print("Resultats:")
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)

Resultats:
MAE: 13.571428571428571
MSE: 327.57142857142856
RMSE: 18.09893445955945
R-Squared: 0.9326089106509075
```

Figura 10.23: Resultat de les mètriques d'avaluació.  
Font: elaboració pròpia

#### 4. Predicció de valors futurs

Per tal de predir valors futurs, cal seguir el mateix procediment que s'ha utilitzat per validar el model, amb la diferència que s'utilitzaran totes les dades disponibles i es farà la predicció pels dies que ens interessin.

### 10.3 Xarxa neuronal LSTM

Finalment, es descriuen els passos que s'han seguit per crear la xarxa neuronal LSTM.

#### 1. Importar paquets i carregar les dades

En primer lloc, i tal i com s'ha fet en els altres models, es carregaran les llibreries necessàries i les dades. A la Figura 10.24 es mostren quines llibreries s'han carregat, en aquest cas es destaquen els mòduls de *Keras* (*Keras* és una biblioteca de xarxes neuronals artificials de codi obert).

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
from statsmodels.tsa.seasonal import seasonal_decompose
import math
from sklearn.preprocessing import MinMaxScaler
from keras.preprocessing.sequence import TimeseriesGenerator
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
import sklearn.metrics as metrics
from sklearn.metrics import mean_squared_error
from math import sqrt
```

Figura 10.24: Importar les llibreries.  
Font: elaboració pròpia

Pel que fa les dades, tal i com s'ha fet en els dos models anteriors, es definiran els subconjunts de test i train.

#### 2. Escalar les dades

Per obtenir els millors resultats optimitzats amb el model, s'escalen les dades. Per fer-ho, s'utilitza l'escalador *MinMaxScaler* de la biblioteca *scikit-learn*, el qual retorna valors entre 0 i 1 (veure Figura 10.25). El funcionament del *MinMaxScaler* és: resta el valor mínim de la sèrie de dades i després divideix per l'interval. L'interval és la diferència entre el màxim original i el mínim original. Aquest escalador conserva la forma de la distribució original i no canvia significativament la informació de les dades originals ni redueix la importància dels valors atípics.

```
# Escalar les dades amb el MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(train)
scaled_train = scaler.transform(train)
scaled_test = scaler.transform(test)
```

Figura 10.25: Escalar les dades amb l'escalador *MinMaxScaler*.  
Font: elaboració pròpia

### 3. Preparació de les dades

*Keras* proporciona el *TimeseriesGenerator* que es pot utilitzar per transformar automàticament un conjunt de dades de sèrie temporal en un problema d'aprenentatge supervisat. El *TimeseriesGenerator* pren una seqüència de punts de dades recopilats a intervals iguals per fer lots d'entrenament/validació. Per fer aquesta transformació cal especificar les següents dades:

- **Entrada i sortida:** en la majoria dels problemes de predicció de sèries de temps, les sèries d'entrada i sortida seran la mateixa sèrie, en el nostre cas *scaled\_train* (les dades de *train\_data* escalades).
- ***length*:** nombre d'observacions utilitzat per entrenar la xarxa.
- ***batch\_size*:** és el nombre de dades que té cada iteració d'un *epoch*. L'*epoch* és el nombre de vegades que s'executen els algorismes. En cada *epoch* totes les dades d'entrenament passen per la xarxa neuronal perquè aquesta aprengui. Si el nombre de *batch\_size* és petit, significa que la xarxa té memòria de poca quantitat de dades, i que entrena més ràpid.

Tal i com es pot veure en la Figura 10.26, s'ha definit una *length* de 7, per tant, utilitza una finestra de 7 valors per predir els següent valor.

```
# Definió del TimeseriesGenerator
n_features = 1
n_input = 7
generator = TimeseriesGenerator(scaled_train, scaled_train, length=n_input, batch_size=1)

X,y = generator[0]
print(f'Donada la matriu: \n{X.flatten()}')
print(f'Prediu: \n {y}')

Donada la matriu:
[0.02678571 0.57142857 0.45535714 0.44047619 0.46130952 0.43452381
 0.49702381]
Prediu:
[[0.02380952]]
```

Figura 10.26: Definió de *TimeseriesGenerator*.  
Font: elaboració pròpia

#### 4. Definició del model

El següent pas és dissenyar el model i posar en marxa la xarxa LSTM. Primer s'especifica la classe *Sequential*. Aquesta serà la classe del model i llavors s'afegiran les capes *LSTM* i *Dense* amb el mètode d'addició.

Primer es defineix la capa LSTM, especificant: el nombre de neurones que volem a la capa i l'*input\_shape*, aquest argument pren una tupla de dos valors que defineixen el nombre de passos i característiques de temps.

Per fer que el nostre model sigui més robust, s'afegeix la capa *Dense* al final del model. El nombre de neurones d'aquesta capa s'establirà en 1, ja que només es vol predir un sol valor a la sortida.

Les màquines aprenen mitjançant una funció de pèrdua, un mètode per avaluar com s'ajusta el model a les dades reals. Si les prediccions es desvien molt de les observacions reals, aquesta funció té un valor molt alt. Poc a poc, amb l'ajuda de la funció d'optimització, la funció de pèrdua aprèn a reduir l'error de la predicció. Sabent això, es compila la xarxa LSTM i s'especifica l'error quadrat mitjà com a funció de pèrdua, i per reduir la pèrdua o per optimitzar l'algorisme s'utilitza l'optimitzador *adam* (veure Figura 10.27).

```
# Definició del model
model = Sequential()
model.add(LSTM(100, activation='relu', input_shape=(n_input, n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

# Mostrar resum de l'arquitectura de la xarxa LSTM
model.summary()
```

Figura 10.27: Definició del model.

Font: elaboració pròpia

Quan s'executa la comanda *model\_summary*, s'obté la taula de la Figura 10.28, on es mostra l'arquitectura de la xarxa neuronal definida.

```
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 100)	40800
dense_3 (Dense)	(None, 1)	101

```
=====  
Total params: 40,901  
Trainable params: 40,901  
Non-trainable params: 0  
=====
```

Figura 10.28: Arquitectura de la xarxa neuronal.  
Font: elaboració pròpia

### 5. Ajustament del model

Ara que ja s'ha definit el model es comença a entrenar la xarxa, es fa amb la comanda *fit* i especificant el nombre d'*epochs* (veure Figura 10.29).

```
# Ajustament del model  
model.fit(generator, epochs=50)  
429/429 [=====] - 1s 3ms/step - loss: 0.0145  
Epoch 23/50  
429/429 [=====] - 1s 3ms/step - loss: 0.0154  
Epoch 24/50  
429/429 [=====] - 1s 3ms/step - loss: 0.0145  
Epoch 25/50  
429/429 [=====] - 1s 3ms/step - loss: 0.0147  
Epoch 26/50  
429/429 [=====] - 1s 3ms/step - loss: 0.0143  
Epoch 27/50
```

Figura 10.29: Ajustament del model.  
Font: elaboració pròpia



En la Figura 10.30 es pot observar que la xarxa neuronal s'entrena de manera satisfactòria, ja que els valors d'error van disminuint a cada *epoch*.

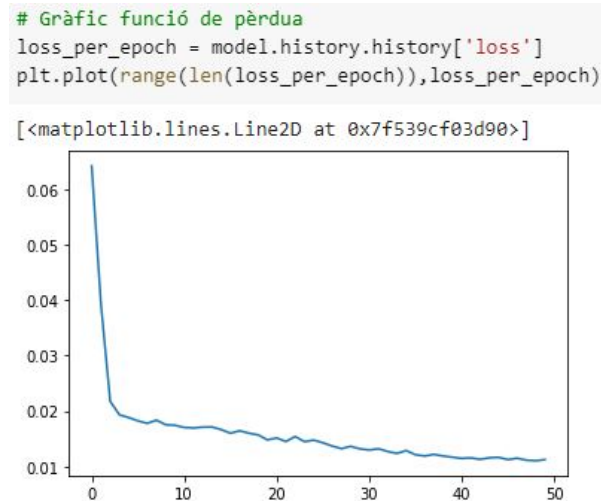


Figura 10.30: Funció de pèrdua.  
Font: elaboració pròpia

## 6. Predicció de valors futurs

Un cop entrenat el model, es fa la predicció de valors futurs (veure Figura 10.31). Cal tenir en compte que les dades s'hauran de reescalar perquè tornin a l'escala original (veure Figura 10.32).

```
# Predicció de valors futurs
test_predictions = []

first_eval_batch = scaled_train[-n_input:]
current_batch = first_eval_batch.reshape((1, n_input, n_features))

for i in range(len(test)):

    # Obtenir el valor de predicció del primer batch
    current_pred = model.predict(current_batch)[0]

    # Afegir la predicció a la matriu
    test_predictions.append(current_pred)

    # Utilitzar la predicció per actualitzar el batch i eliminar el primer valor
    current_batch = np.append(current_batch[:,1:,:],[[current_pred]],axis=1)
```

Figura 10.31: Predicció de valors futurs.  
Font: elaboració pròpia



```
# Escalar les dades de predicció a l'escala original i graficar els resultats  
# obtinguts per comparar amb les dades reals  
true_predictions = scaler.inverse_transform(test_predictions)  
test['Predictions'] = true_predictions  
test.plot(figsize=(14,5))
```

Figura 10.32: Escalar les dades.  
Font: elaboració pròpia

A la Figura 10.33 es mostren els resultats obtinguts, amb color blau les dades reals i amb color taronja les dades pronosticades, i a la Figura 10.34 el valor numèric d'aquestes prediccions.

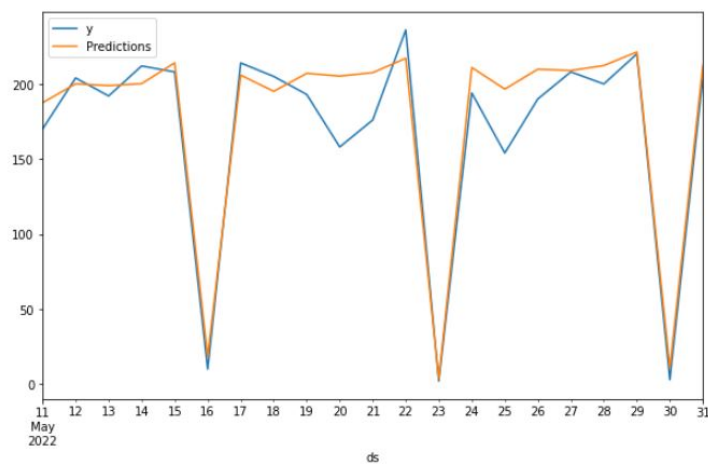


Figura 10.33: Representació del model de predicció LSTM.  
Font: elaboració pròpia

```
# Valors de la predicció
round(test['Predictions'])

ds
2022-05-11    184.0
2022-05-12    196.0
2022-05-13    194.0
2022-05-14    198.0
2022-05-15    208.0
2022-05-16     5.0
2022-05-17    196.0
2022-05-18    184.0
2022-05-19    197.0
2022-05-20    195.0
2022-05-21    199.0
2022-05-22    206.0
2022-05-23     3.0
2022-05-24    196.0
2022-05-25    184.0
2022-05-26    197.0
2022-05-27    196.0
2022-05-28    200.0
2022-05-29    206.0
2022-05-30     5.0
2022-05-31    196.0
Freq: D, Name: Predictions, dtype: float64
```

Figura 10.34: Valors de la predicció amb la xarxa LSTM.  
Font: elaboració pròpia

## 7. Mètriques d'avaluació

Finalment, tal i com s'ha fet en els altres models, s'apliquen les mètriques d'avaluació per posteriorment comparar els models entre ells. A la Figura 10.35 podem veure els resultats que s'han obtingut (MAE: 12.1; MSE: 259.7; RMSE: 16.1;  $R^2$ :0.95).

```
# Mètriques d'avaluació
mae = metrics.mean_absolute_error(test['y'],round(test['Predictions']))
mse = metrics.mean_squared_error(test['y'],round(test['Predictions']))
rmse = np.sqrt(mse)
r2 = metrics.r2_score(test['y'],round(test['Predictions']))

print("Resultats:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)

Resultats:
MAE: 12.113788786388579
MSE: 259.6969661226804
RMSE: 16.115116075370988
R-Squared: 0.9465726863787036
```

Figura 10.35: Validació dels resultats obtinguts.  
Font: elaboració pròpia

## 8. Predicció de valors futurs

Per tal de predir valors futurs, cal seguir el mateix procediment que s'ha utilitzat per validar el model, amb la diferència que s'utilitzaran totes les dades disponibles i no es faran dos subconjunts.

## 10.4 Resum resultats obtinguts

En aquest apartat, es resumeixen els resultats obtinguts dels diferents models pel client en qüestió.

A la Figura 10.36 es mostren gràficament les dades reals i les prediccions obtingudes per cada model, tal i com mostra la llegenda es representen amb blau les dades reals, amb groc les prediccions del model SARIMA, amb verd les del Prophet i amb vermell les de la xarxa LSTM. Amb aquest gràfic s'observa que el pronòstic dels model SARIMA i Prophet és força similar entre ells, mentre que la xarxa LSTM difereix més.

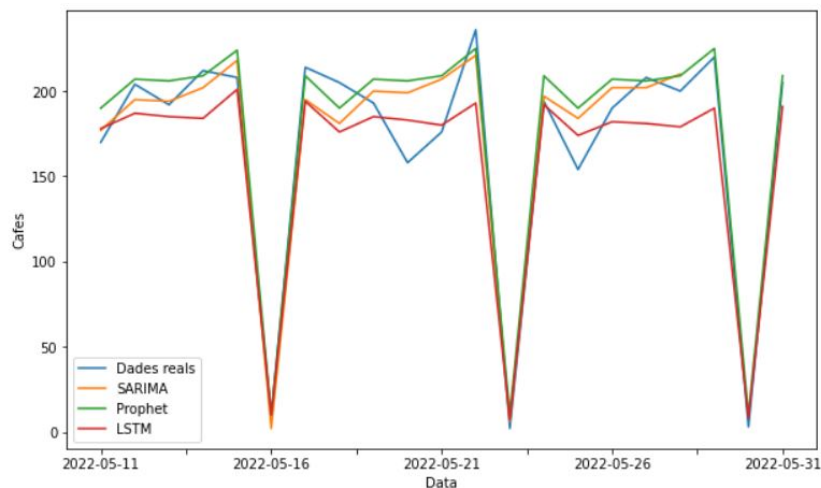


Figura 10.36: Resultats obtinguts dels diferents models.

Font: elaboració pròpia

Finalment, a la Taula 10.1 es mostren els valors numèrics de les dades reals i de les prediccions dels diferents models, i a la Taula 10.2 es presenten els resultats de les diferents mètriques d'avaluació per cadascun dels models.

Data	Dades reals	SARIMA	Prophet	LSTM
2022-05-11	170	181	190	184
2022-05-12	204	198	207	196
2022-05-13	192	198	206	194
2022-05-14	212	205	209	198
2022-05-15	208	221	224	208
2022-05-16	10	5	10	5
2022-05-17	214	198	209	196
2022-05-18	205	184	190	184
2022-05-19	193	202	207	197
2022-05-20	158	201	206	195
2022-05-21	176	209	209	199
2022-05-22	236	223	225	206
2022-05-23	2	8	10	3
2022-05-24	194	200	209	196
2022-05-25	154	186	190	184
2022-05-26	190	204	207	197
2022-05-27	208	203	206	196
2022-05-28	200	211	209	200
2022-05-29	220	224	225	206
2022-05-30	3	9	10	5
2022-05-31	205	201	209	196

Taula 10.1: Valors de les dades reals i de les prediccions obtingudes.

Mètrica	SARIMA	Prophet	LSTM
MAE	12.9	13.6	12.1
MSE	277.5	328.8	259.7
RMSE	16.6	18.1	16.1
$R^2$	0.94	0.93	0.95

Taula 10.2: Resultats obtinguts de les mètriques d'avaluació.

Tal i com es pot observar, els tres models s'han ajustat bé, ja que el valor d' $R^2$  en els tres casos és superior a 0.9. Tot i això, per aquest client, el model que s'ha ajustat millor és la xarxa LSTM i el que pitjor el model Prophet.

A l'Annex A es troba el codi complet dels tres models, i a l'Annex B es presenta, per cada client, els resultats obtinguts per cadascun dels models.

# Avaluació econòmica

---

L'avaluació econòmica del projecte s'ha fet considerant les diferents tasques realitzades. En particular s'ha considerat una primera fase d'adquisició i preparació de les dades que ha consistit en veure quina informació i quines dades disposava l'empresa. Una segona fase d'anàlisi exploratòria per tal de trobar quines eren les variables que ens podien afectar. Finalment, s'han implementat els tres models de predicció per posteriorment poder-los validar amb les dades dels clients. A la Taula 11.1 es recullen aquests conceptes amb el seu cost corresponent (no s'han tingut en compte costos indirectes).

Concepte	Hores	€/h	Preu(€)
Adquisició i preparació del conjunt de dades	40	15	600
Anàlisi exploratòria i selecció de característiques	30	40	1.200
Model de predicció del nombre de tasses	150	40	6.000
Validació del model pels casos d'ús	50	40	2.000
TOTAL			9.800

Taula 11.1: Avaluació econòmica.

El cost d'aquest estudi és de 9.800€ (nou mil vuit-cents euros).



## CAPÍTOL 12

# Conclusions

---

En un model de negoci tant important com adaptar-se a les necessitats dels clients és la capacitat de l'empresa d'avançar-se a les noves necessitats que aquests puguin tenir. En aquest context, la servitització es pot considerar com una nova estratègia de negoci en la que enlloc de productes s'ofereixen serveis. És a dir, que es passa d'un model d'economia lineal a un model d'economia circular, on es busquen sistemes de producció i consum més eficients i resilients, que preservin els recursos dins d'un cicle continu, optimitzant-ne el seu valor.

Un exemple d'aplicació de servitització és la nova proposta de model de negoci de Cafès Cornellà, on es planteja passar de vendre quilograms de cafè a facturar tasses de cafè, amb l'objectiu d'anar cap a una proposta de valor més eficient i circular. Aquesta proposta va acompanyada d'una plataforma (Cafeteria 4.0) destinada als clients i que té per objectiu augmentar la rendibilitat dels seus establiments. La plataforma tindrà diferents funcionalitats, com ara informar del nombre òptim de tasses que s'haurien de servir o avisar de quan cal fer manteniment de les cafeteres i així evitar avaries.

Aquest projecte s'ha centrat en la funcionalitat de predicció de tasses de cafè diàries. L'objectiu ha estat validar si es podrien predir el nombre de cafès que ha de servir una cafeteria. Per fer-ho cal disposar de dades que ens ajudin a validar la proposta i també models de predicció que s'ajustin a les característiques d'aquestes dades.

Aplicant els models s'espera treure conclusions per acabar decidint quina és la millor opció per informar del nombre de cafès que s'haurien de servir. Tot i que el problema és simple de descriure trobar la solució no ho és tant, ja que hi ha diferents models de predicció i no sempre està clar quin cal aplicar. La variabilitat de les dades o la manca d'informació pot complicar l'estudi.

En primer lloc, s'ha fet un estudi de l'empresa per veure quina era exactament la nova proposta de valor i quines eren les dades de les quals es disposa i podien ser útils per treballar amb la plataforma Cafeteria 4.0. Ha calgut analitzar les diferents fonts d'informació de l'empresa veient que la plataforma desenvolupada per QualityEspresso era la més rellevant. Aquesta, des de fa uns anys està recollint el nombre de cafès que es serveixen a diferents establiments que tenen la cafetera connectada amb tecnologia M2M (Machine to Machine). Partint d'aquestes dades i tenint en compte les restriccions imposades al sector

de l'hostaleria per la COVID-19 (que han limitat la possibilitat d'explotar tota la informació) s'ha definit com a període d'estudi de l'01/03/2021 al 31/05/2022.

En segon lloc, s'han analitzat tres models diferents de predicció de sèries temporals: dos mètodes aplicant tècniques de predicció tradicionals (ARIMA estacional, Prophet) i un amb tècniques d'intel·ligència artificial (xarxa neuronal LSTM).

Finalment, amb els tres models definits, s'han fet prediccions amb les dades seleccionades i s'han utilitzat les mètriques d'avaluació MAE, MSE, RMSE i  $R^2$  per poder-los comparar entre ells. Aquest estudi s'ha fet per quatre clients analitzant la informació de forma independent i també de forma conjunta. D'aquest estudi s'ha pogut concloure que:

- Els tres models (ARIMA estacional, Prophet i xarxa LSTM) s'ajusten bé a les dades.
- En el cas de les xarxes LSTM, tot i que s'obtenen bons resultats, el temps i cost computacional necessari per generar el model és major a les altres dues opcions.
- En el model ARIMA estacional es fa una iteració de diferents models per cada client i així triar els paràmetres que s'adapten millor a les dades, la qual cosa comporta que tardi més a executar-se respecte el Prophet.
- Es considera que el model més adequat és el Prophet desenvolupat per Facebook. A més, cal tenir en compte que aquest model dóna la possibilitat d'afegir dates quan hi ha esdeveniments que poden canviar la previsió (per exemple dies festius o accions de màrqueting).

Els resultats obtinguts ens garanteixen poder predir el nombre de cafès i per tant aquesta funcionalitat es podria integrar en la plataforma Cafeteria 4.0 que es vol oferir als clients. Tot i aquesta validació, l'acceptació per part dels clients no està clara ja que una gestió eficient de l'establiment no afecta només als cafès sinó a tota la gestió dels productes que ofereixen. Dit això, la iniciativa i la forma de pensar de Cafès Cornellà crec que és disruptiva i innovadora, i li permet diferenciar-se de la competència.

No vull acabar aquest document sense abans fer una valoració personal del que ha suposat per mi tant el treball realitzat en aquest projecte com el meu pas per la Universitat de Girona. Per una banda, dir que el projecte m'ha donat una visió real del món de l'empresa i des de diferents punts de vista. El fet de ser una petita empresa m'ha estat fàcil accedir a tota la informació i a les diferents metodologies de treball que s'apliquen. Crec que ha estat un procés molt enriquidor que ha complementat molts dels aspectes treballats en diferents assignatures. Per altra banda, m'ha obert la porta a la ciència de dades,



on he adquirit coneixement nous i he vist la importància que té en el món de l'enginyeria.



# Relació de documents

---

El treball consta d'un sol document: "Memòria i annexos".



## CAPÍTOL 14

# Bibliografia

---

- [Cafès Cornellà ] Cafès Cornellà. <https://cafescornella.coffee/> (Consultada: Abril 2022). (Cited on page 11.)
- [Clic 022] Programador Clic. *Estructura detallada de LSTM*, (Consultada: Maig 2022). Disponible a: <https://programmerclick.com/article/26441537544/>. (Cited on pages 36 and 37.)
- [de Madrid 022] Unviuersidad Autónoma de Madrid. *Series temporales, Modelo Arima Metodología de Box-Jenkins*, (Consultada: Maig 2022). Disponible a: <https://www.estadistica.net/ECONOMETRIA/SERIES-TEMPORALES/modelo-arima.pdf>. (Cited on pages 30 and 31.)
- [Eurecat ] Eurecat. <https://eurecat.org/> (Consultada: Abril 2020). (Cited on page 13.)
- [Holicka 022] Nikol Holicka. *Interpreting ARMA model results in Statsmodels for absolute beginners*, (Consultada: Maig 2022). Disponible a: <https://medium.com/analytics-vidhya/interpreting-arma-model-results-in-statsmodels-for-absolute-beginners-a4d22253ad1c>. (Cited on page 46.)
- [IBM 022a] IBM. *Estructura ARIMA*, (Consultada: Maig 2022). Disponible a: <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=options-arima-structure>. (Cited on page 29.)
- [IBM 022b] Cloud Education IBM. *Exploratory Data Analysis*, (Consultada: Maig 2022). Disponible a: <https://www.ibm.com/cloud/learn/exploratory-data-analysis>. (Cited on page 25.)
- [Mañas 022] Andrés Mañas Mañas. *Pronóstico del flujo de tráfico en la ciudad de Madrid*, (Consultada: Maig 2022). Disponible a: <https://bookdown.org/amanas/traficomadrid/m%C3%A9todos-basados-en-deep-learning.html#lstm-univariado>. (Cited on pages 35 and 36.)

- [Parra 022] Francisco Parra. *Estadística y Machine Learning con R*, (Consultada: Maig 2022). Disponible a: <https://bookdown.org/content/2274/series-temporales.html#descomposicion-temporal>. (Cited on page 21.)
- [PIMEC 2020] PIMEC and Inèdit amb el suport de l'Agència de Residus de Catalunya. *Aplicació de la servitització a les pimes catalanes*. Desembre 2020. (Cited on pages 7, 8 and 9.)
- [Puerta 022] Maite Puerta. *Análisis y Predicción Individual del Comportamiento de los Usuarios con Abono Tercera Edad en el Transporte Público de la Comunidad de Madrid*, (Consultada: Maig 2022). Disponible a: [https://oa.upm.es/67935/1/TFM\\_MAITE\\_PUERTA\\_BELDARRAIN.pdf](https://oa.upm.es/67935/1/TFM_MAITE_PUERTA_BELDARRAIN.pdf). (Cited on page 22.)
- [Santana 022] Besay Montesdeoca Santana. *Estudios de predicción en series temporales de datos meteorológicos utilizando redes neuronales recurrentes*, (Consultada: Maig 2022). Disponible a: [https://accedacris.ulpgc.es/bitstream/10553/18829/1/0728348\\_00000\\_0000.pdf](https://accedacris.ulpgc.es/bitstream/10553/18829/1/0728348_00000_0000.pdf). (Cited on page 36.)
- [Sean J. Taylor 022] Benjamin Letham Sean J. Taylor. *Forecasting at Scale*, (Consultada: Abril 2022). Disponible a: <https://peerj.com/preprints/3190/>. (Cited on page 32.)
- [Viquipèdia 022a] Viquipèdia. *Akaike information criterion*, (Consultada: Maig 2022). Disponible a: [https://en.wikipedia.org/wiki/Akaike\\_information\\_criterion](https://en.wikipedia.org/wiki/Akaike_information_criterion). (Cited on page 44.)
- [Viquipèdia 022b] Viquipèdia. *Xarxa neuronal artificial*, (Consultada: Maig 2022). Disponible a: [https://ca.wikipedia.org/wiki/Xarxa\\_neuronal\\_artificial](https://ca.wikipedia.org/wiki/Xarxa_neuronal_artificial). (Cited on page 34.)

# Annex A. Codi

---

En aquest annex s'adjunta el codi implementat pels tres models de predicció en el següent ordre: ARIMA estacional, Prophet Facebook i xarxa LSTM.

## ▼ ARIMA estacional

### 1. IMPORTAR LLIBRERIES I CARREGAR LES DADES

```
# Importar llibreries
import warnings
import itertools
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
import statsmodels.tsa as sm
from datetime import datetime
import plotly.graph_objects as go
import os
import seaborn as sns
from statsmodels.tsa.seasonal import DecomposeResult
import itertools
import statsmodels.api as sm
from sklearn.metrics import mean_squared_error, r2_score
```

```
# Carregar les dades
path_csv = 'cafes.csv'
data = pd.read_csv(path_csv, engine='python', skipfooter=3)
data['ds']=pd.to_datetime(data['ds'], format='%Y-%m-%d')
data.set_index(['ds'], inplace=True)
y=data.y
y = y.resample('D').mean()

# Representació gràfica de les dades
plt.rcParams['figure.figsize'] = (10.0, 6.0)
data.plot()
plt.ylabel('Cafes')
plt.xlabel('Data')
plt.show()
```

```
# Crear els dos subconjunts de dades: train i test
len_df=len(data)
train_data = pd.DataFrame(y[:(len_df-21)])
test_data = pd.DataFrame(y[-21:])
```

### 2. SELECCIÓ DE PARÀMETRES PEL MODEL ARIMA ESTACIONAL

```
# Definició dels rangs dels paràmetres q i p
q = d = range(0, 2)
# Definció del rang del paràmetre d
p = range(0, 2)

# Generació de les diferents combinacions de p, d i q
pdq = list(itertools.product(p, d, q))

# Generació de les diferents combinacions estacionals P, Q i D
```



```

seasonal_pdq = [(x[0], x[1], x[2], 7) for x in list(itertools.product(p, d, q))]

print('Exemples de combinacions del model Seasonal ARIMA:')
print('SARIMA: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMA: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMA: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMA: {} x {}'.format(pdq[2], seasonal_pdq[4]))

```

```

# Obtenció de l'AIC per cada combinació del model SARIMA
warnings.filterwarnings("ignore") # specify to ignore warning messages

AIC = []
SARIMAX_model = []
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(train_data,
                                             order=param,
                                             seasonal_order=param_seasonal,
                                             enforce_stationarity=False,
                                             enforce_invertibility=False)

            results = mod.fit()
            AIC.append(results.aic)
            SARIMAX_model.append([param, param_seasonal])

            print('SARIMA{}x{} - AIC:{}'.format(param, param_seasonal, results.aic))
        except:
            continue

```

```

# Mostrar el model SARIMA amb un AIC més petit
min(AIC)
print('SARIMA{}x{} - AIC:{}'.format(SARIMAX_model[AIC.index(min(AIC))][0],
                                    SARIMAX_model[AIC.index(min(AIC))][1], min(AIC)))

```

### 3. AJUSTAMENT DEL MODEL

```

# Ajustament del model
mod = sm.tsa.statespace.SARIMAX(train_data,
                                order=SARIMAX_model[AIC.index(min(AIC))][0],
                                seasonal_order=SARIMAX_model[AIC.index(min(AIC))][1],
                                enforce_stationarity=False,
                                enforce_invertibility=False)

```

```

results = mod.fit()

```

```

# Resum
print(results.summary())

```

```

                    Statespace Model Results
=====
Dep. Variable:                y      No. Observations:                436
Model:                SARIMAX(1, 0, 1)x(1, 1, 1, 7)      Log Likelihood                -2114.826
Date:                Wed, 08 Jun 2022      AIC                4239.651
Time:                16:21:18      BIC                4259.853
Sample:                03-01-2021      HQIC                4247.636

```

- 05-10-2022

Covariance Type:

opg

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	0.9171	0.016	57.993	0.000	0.886	0.948
ma.L1	-0.6173	0.032	-19.233	0.000	-0.680	-0.554
ar.S.L7	0.1008	0.034	3.005	0.003	0.035	0.167
ma.S.L7	-0.9541	0.026	-37.415	0.000	-1.004	-0.904
sigma2	1340.8865	51.448	26.063	0.000	1240.050	1441.723

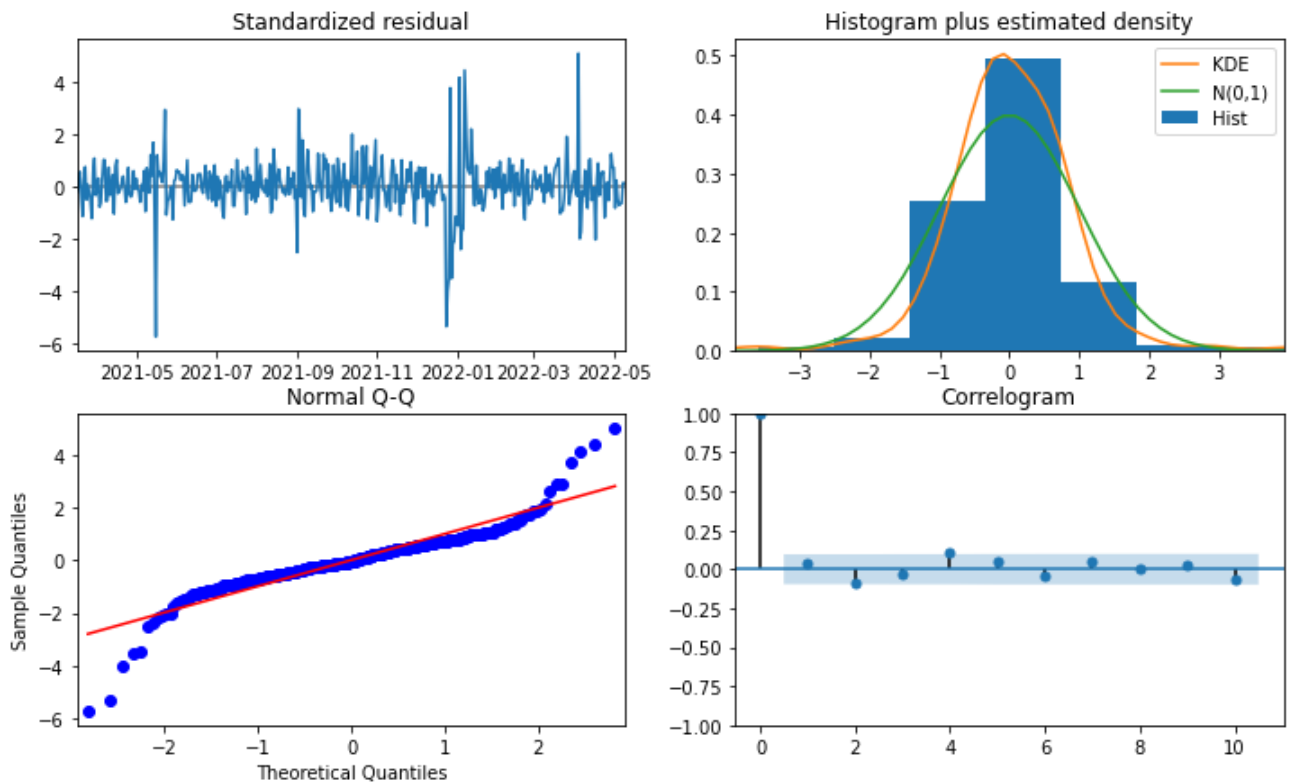
  

Ljung-Box (Q):	25.61	Jarque-Bera (JB):	1146.93
Prob(Q):	0.96	Prob(JB):	0.00
Heteroskedasticity (H):	2.65	Skew:	-0.30
Prob(H) (two-sided):	0.00	Kurtosis:	11.07

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).

```
# Diagnòstics del model
results.plot_diagnostics(figsize=(12, 7))
plt.show()
```



#### 4. VALIDACIÓ DE LES PREVISIONS

```
inici=test_data.index.min()
end=test_data.index.max()
print(inici, end)
train_size = train_data.shape[0]
test_size = test_data.shape[0]
```

```
# Predicció pel període de dates del subconjunt test_data
pred = results.get_prediction(start=inici, end=end, dynamic=False, full_results=True)
pred_ci = pred.conf_int()
```

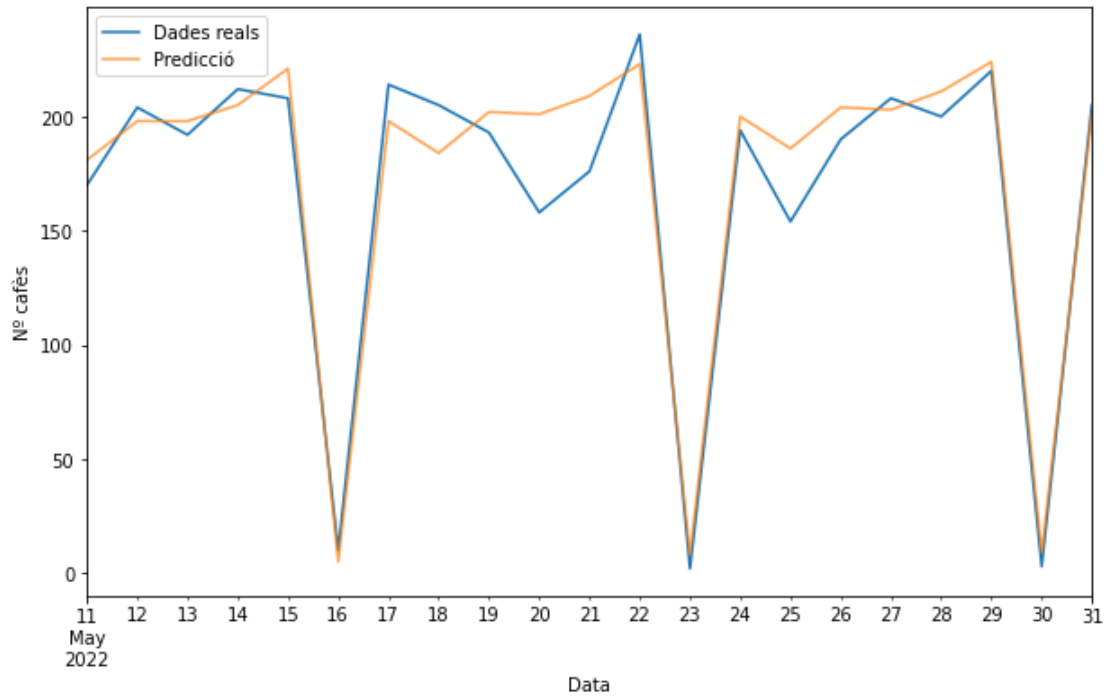
```

# Representació gràfica dels resultats
plt.rcParams['figure.figsize'] = (10.0, 6.0)
ax = y[-21:].plot(label='Dades reals')
round(pred.predicted_mean).plot(ax=ax, label='Predicció', alpha=.7)

ax.set_xlabel('Data')
ax.set_ylabel('Nº cafès')
plt.legend()

plt.show()

```



```

# Valors de la predicció
round(pred.predicted_mean)

```

```

2022-05-11    181.0
2022-05-12    198.0
2022-05-13    198.0
2022-05-14    205.0
2022-05-15    221.0
2022-05-16     5.0
2022-05-17    198.0
2022-05-18    184.0
2022-05-19    202.0
2022-05-20    201.0
2022-05-21    209.0
2022-05-22    223.0
2022-05-23     8.0
2022-05-24    200.0
2022-05-25    186.0
2022-05-26    204.0
2022-05-27    203.0
2022-05-28    211.0
2022-05-29    224.0
2022-05-30     9.0
2022-05-31    201.0
Freq: D, dtype: float64

```

```

# Mètriques d'avaluació

```

```
import sklearn.metrics as metrics
import numpy as np

mae = metrics.mean_absolute_error(test_data, round(pred.predicted_mean))
mse = metrics.mean_squared_error(test_data, round(pred.predicted_mean))
rmse = np.sqrt(mse)
r2 = metrics.r2_score(test_data, round(pred.predicted_mean))

print("Resultats:")
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
```

```
Resultats:
MAE: 12.904761904761905
MSE: 277.4761904761905
RMSE: 16.65761659050269
R-Squared: 0.9429149763574411
```

## 5. PREDICCIÓ DE VALORS FUTURS

Per fer la predicció amb totes les dades, és el mateix procediment, però utilitzant `get_forecast` enlloc de `get_prediction`. Per exemple: `pred_uc = results.get_forecast(7)`

## ▼ PROPHET

### 1. IMPORTAR LLIBRERIES I CARREGAR LES DADES

```
!pip install pystan==2.19.1.1
!pip install prophet
!pip install pandas
```

```
import pandas as pd
from fbprophet import Prophet
import matplotlib.pyplot as plt
import matplotlib as mpl
import matplotlib.patches as mpatches
import numpy as np
from datetime import datetime, timedelta
import seaborn as sns
import warnings
import sklearn.metrics as metrics
warnings.filterwarnings('ignore')
```

```
# Llegir dades del csv
df = pd.read_csv('cafes.csv', sep=',')
df.head()
# Convertim la columna temps a dateime
df['ds'] = pd.to_datetime(df['ds'], format='%Y-%m-%d')
```

```
# Representació gràfica de les dades
ax = df.set_index(['ds']).plot(figsize=(10, 4))
ax.set_ylabel('Nº cafès')
ax.set_xlabel('Data')
plt.show()
```

```
# Crear els dos subconjunts de dades: train i test
len_df=len(df)
train = df.iloc[:(len_df-21)]
test = df.iloc[-21:]
```

### 2. CREACIÓ DEL MODEL PROPHET I PREDICCIÓ

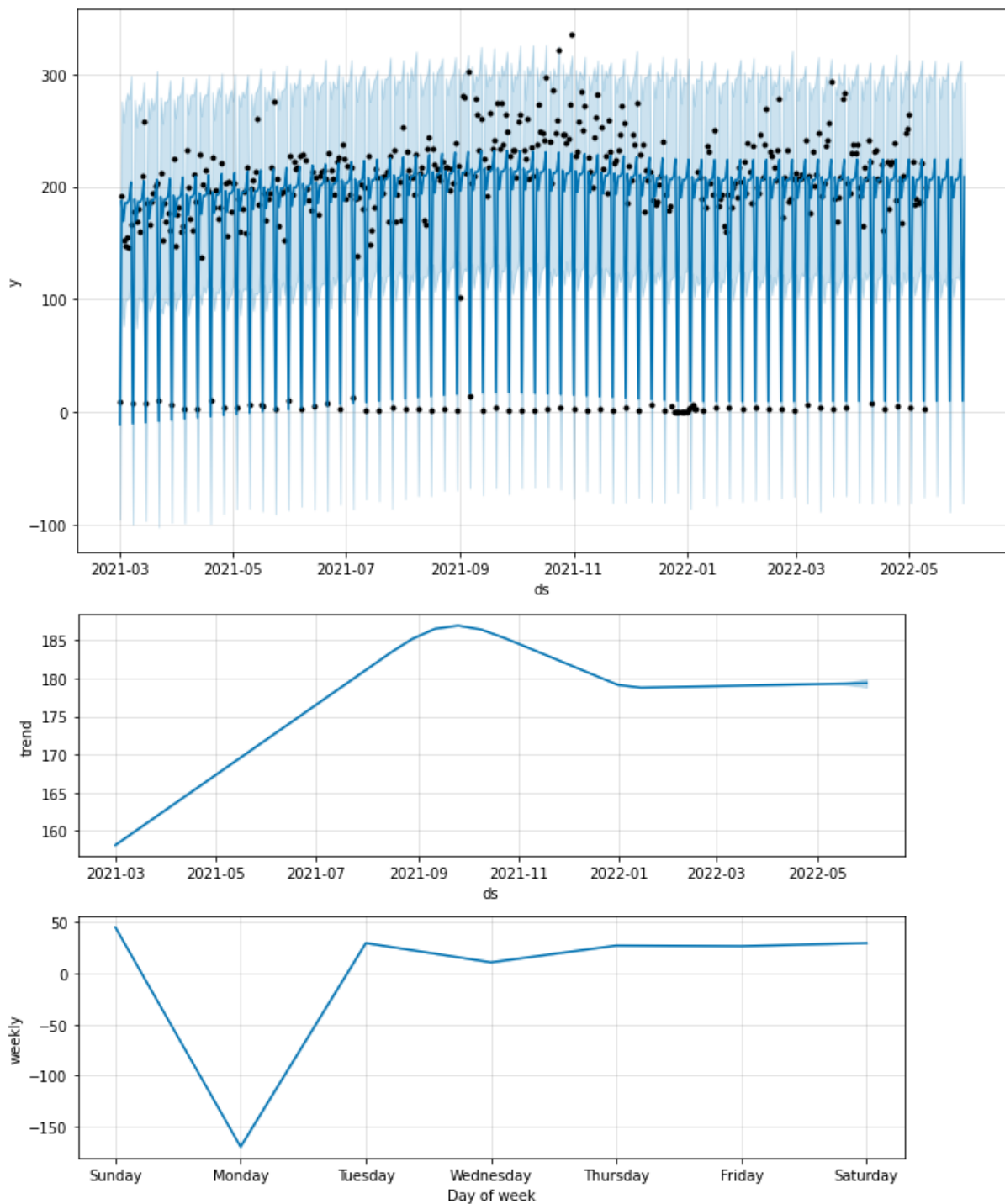
```
# Crear el model amb un interval d'incertesa del 95%
my_model = Prophet(interval_width=0.95)
```

```
my_model.fit(train)
```

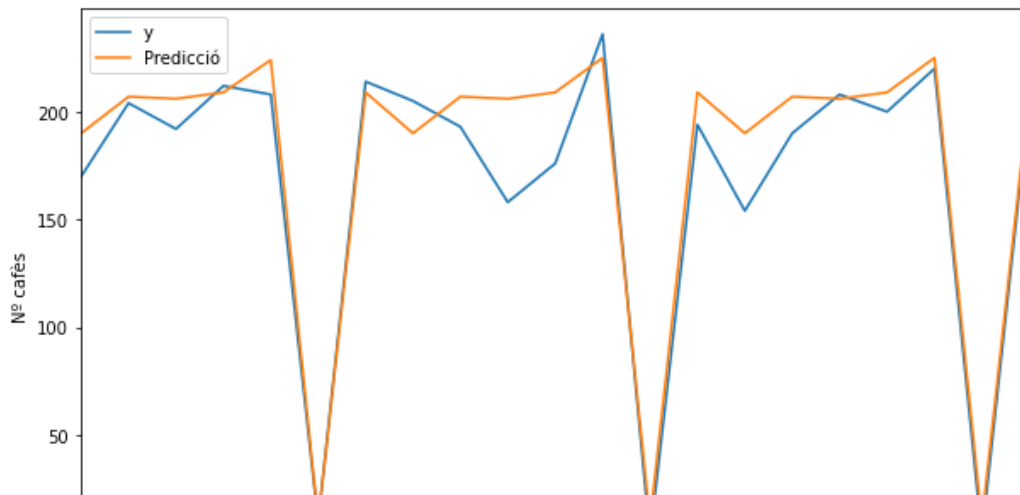
```
# Predicció de 21 dies
future_dates = my_model.make_future_dataframe(periods=21, freq='D')
future_dates.tail()
```

```
# Resultats obtinguts
forecast = my_model.predict(future_dates)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']]
```


```
# Representació gràfica de la predicció, la tendència anual i el patró setmanal
fig1=my_model.plot(forecast)
fig2=my_model.plot_components(forecast)
```



```
res=pd.DataFrame(test)
res.insert(2,"Predicció",round(forecast.yhat[len(forecast.yhat)-len(test):]),True)
ax1 = res.set_index(['ds']).plot(figsize=(10, 6))
ax1.set_ylabel('Nº cafès')
ax1.set_xlabel('Data')
plt.show()
```



```
# Valors de la predicció
res[['ds', 'Predicció']]
```

	ds	Predicció	
436	2022-05-11	190.0	
437	2022-05-12	207.0	
438	2022-05-13	206.0	
439	2022-05-14	209.0	
440	2022-05-15	224.0	
441	2022-05-16	10.0	
442	2022-05-17	209.0	
443	2022-05-18	190.0	
444	2022-05-19	207.0	
445	2022-05-20	206.0	
446	2022-05-21	209.0	
447	2022-05-22	225.0	
448	2022-05-23	10.0	
449	2022-05-24	209.0	
450	2022-05-25	190.0	
451	2022-05-26	207.0	
452	2022-05-27	206.0	
453	2022-05-28	209.0	
454	2022-05-29	225.0	
455	2022-05-30	10.0	
456	2022-05-31	209.0	

### 3. MÈTRIQUES D'AVVALUACIÓ

```
# Mètriques d'avaluació
mae = metrics.mean_absolute_error(test.y, round(forecast.yhat[len(forecast.yhat)-len(test):]))
mse = metrics.mean_squared_error(test.y, round(forecast.yhat[len(forecast.yhat)-len(test):]))
```

```
rmse = np.sqrt(mse)
r2 = metrics.r2_score(test.y, round(forecast.yhat[len(forecast.yhat)-len(test):]))

print("Resultats:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
```

```
Resultats:
MAE: 13.571428571428571
MSE: 327.57142857142856
RMSE: 18.09893445955945
R-Squared: 0.9326089106509075
```

---

✓ 0 s completado a las 19:44





## ▼ XARXA NEURONAL LSTM

### 1. IMPORTAR PAQUETS I CARREGAR DADES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
from statsmodels.tsa.seasonal import seasonal_decompose
import math
from sklearn.preprocessing import MinMaxScaler
from keras.preprocessing.sequence import TimeseriesGenerator
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
import sklearn.metrics as metrics
from sklearn.metrics import mean_squared_error
from math import sqrt

# Carregar les dades i definir la freqüència diària
df = pd.read_csv('/content/cafes.csv', index_col='ds', parse_dates=True)
df.index.freq='D'
```

```
# Mostrar els primers valors del .csv carregat
df.head()
```

```
# Representació gràfica de les dades
df.plot(figsize=(10,4))
```

```
# Crear els dos subconjunts de dades: train i test
len_df=len(df)
train = df.iloc[:(len_df-21)]
test = df.iloc[-21:]
```

### 2. ESCALAR LES DADES

```
# Escalar les dades amb el MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(train)
scaled_train = scaler.transform(train)
scaled_test = scaler.transform(test)
```

### 3. PREPARACIÓ DE LES DADES

```
# Definió del TimeseriesGenerator
n_features = 1
n_input = 7
generator = TimeseriesGenerator(scaled_train, scaled_train, length=n_input, batch_size=1)
```

```
X,y = generator[0]
print(f'Donada la matriu: \n{X.flatten()}')
print(f'Prediu: \n {y}')
```

```
Donada la matriu:  
[0.02678571 0.57142857 0.45535714 0.44047619 0.46130952 0.43452381  
0.49702381]  
Prediu:  
[[0.02380952]]
```

```
X.shape
```

#### 4. DEFINICIÓ DEL MODEL

```
# Definió del model  
model = Sequential()  
model.add(LSTM(100, activation='relu', input_shape=(n_input, n_features)))  
model.add(Dense(1))  
model.compile(optimizer='adam', loss='mse')
```

```
# Mostrar resum de l'arquitectura de la xarxa LSTM  
model.summary()
```

```
Model: "sequential_1"
```

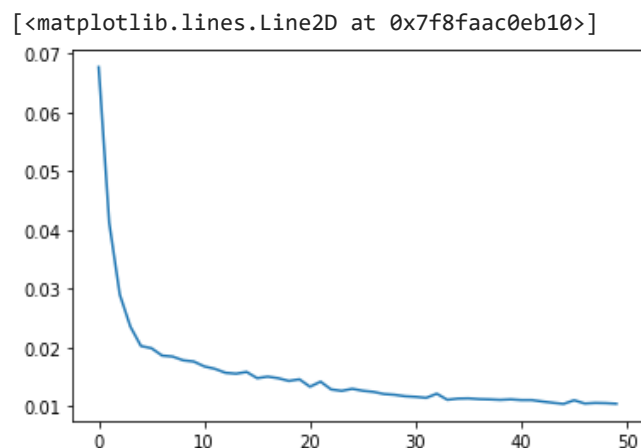
Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 100)	40800
dense_1 (Dense)	(None, 1)	101

```
=====  
Total params: 40,901  
Trainable params: 40,901  
Non-trainable params: 0  
=====
```

#### 5. AJUSTAMENT DEL MODEL

```
# Ajustament del model  
model.fit(generator, epochs=50)
```

```
# Gràfic funció de pèrdua  
loss_per_epoch = model.history.history['loss']  
plt.plot(range(len(loss_per_epoch)), loss_per_epoch)
```



#### 6. PREDICCIÓ DE VALORS FUTURS

```
last_train_batch = scaled_train[-7:]
last_train_batch = last_train_batch.reshape((1, n_input, n_features))
```

```
model.predict(last_train_batch)
```

```
scaled_test[0]
```

```
# Predicció de valors futurs
test_predictions = []

first_eval_batch = scaled_train[-n_input:]
current_batch = first_eval_batch.reshape((1, n_input, n_features))

for i in range(len(test)):

    # Obtenir el valor de predicció del primer batch
    current_pred = model.predict(current_batch)[0]

    # Afegir la predicció a la matriu
    test_predictions.append(current_pred)

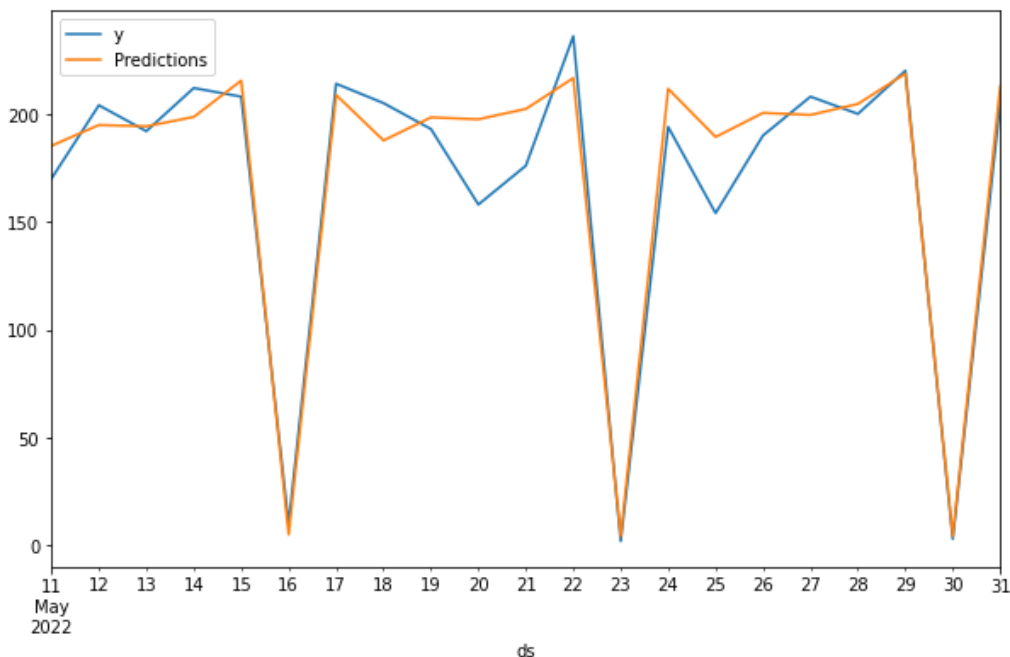
    # Utilitzar la predicció per actualitzar el batch i eliminar el primer valor
    current_batch = np.append(current_batch[:,1:,:], [[current_pred]],axis=1)
```

```
test_predictions
```

```
# Escalar les dades de predicció a l'escala original i graficar els resultats
# obtinguts per comparar amb les dades reals
true_predictions = scaler.inverse_transform(test_predictions)
test['Predictions'] = true_predictions
```

```
test.plot(figsize=(10,6))
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f8faa9f0e10>



```
# Valors de la predicció
```

```
round(test['Predictions'])
```

```
ds
2022-05-11    185.0
2022-05-12    195.0
2022-05-13    194.0
2022-05-14    199.0
2022-05-15    215.0
2022-05-16     5.0
2022-05-17    209.0
2022-05-18    188.0
2022-05-19    198.0
2022-05-20    198.0
2022-05-21    202.0
2022-05-22    217.0
2022-05-23     4.0
2022-05-24    212.0
2022-05-25    189.0
2022-05-26    200.0
2022-05-27    200.0
2022-05-28    205.0
2022-05-29    219.0
2022-05-30     4.0
2022-05-31    213.0
Freq: D, Name: Predictions, dtype: float64
```

## 7. MÈTRIQUES D'AVVALUACIÓ

```
# Mètriques d'avaluació
mae = metrics.mean_absolute_error(test['y'],round(test['Predictions']))
mse = metrics.mean_squared_error(test['y'],round(test['Predictions']))
rmse = np.sqrt(mse)
r2 = metrics.r2_score(test['y'],round(test['Predictions']))

print("Resultats:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
```

```
Resultats:
MAE: 11.952380952380953
MSE: 254.14285714285714
RMSE: 15.941858647687765
R-Squared: 0.9477153301561118
```

# Annex B. Resum resultats

---

En aquest annex es recullen els resultats obtinguts pels diferents clients. Per cada un d'ells es presenten dues taules. A la primera taula es mostren les dades reals i les obtingudes pels tres models pel període de dates del 11 al 31 de maig. A la segona taula es presenten els valors obtinguts de les mètriques d'avaluació. Finalment, es mostra la representació gràfica d'aquests resultats per cadascun dels models (ARIMA estacional, Prophet i la xarxa LSTM).

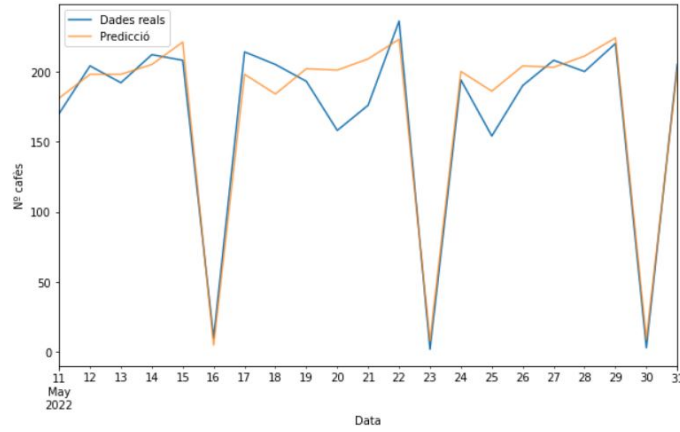
## Client 1

Taula resum dels resultats obtinguts pel client 1				
Data	Dades reals	SARIMA	Prophet	LSTM
2022-05-11	170	181	190	184
2022-05-12	204	198	207	196
2022-05-13	192	198	206	194
2022-05-14	212	205	209	198
2022-05-15	208	221	224	208
2022-05-16	10	5	10	5
2022-05-17	214	198	209	196
2022-05-18	205	184	190	184
2022-05-19	193	202	207	197
2022-05-20	158	201	206	195
2022-05-21	176	209	209	199
2022-05-22	236	223	225	206
2022-05-23	2	8	10	3
2022-05-24	194	200	209	196
2022-05-25	154	186	190	184
2022-05-26	190	204	207	197
2022-05-27	208	203	206	196
2022-05-28	200	211	209	200
2022-05-29	220	224	225	206
2022-05-30	3	9	10	5
2022-05-31	205	201	209	196

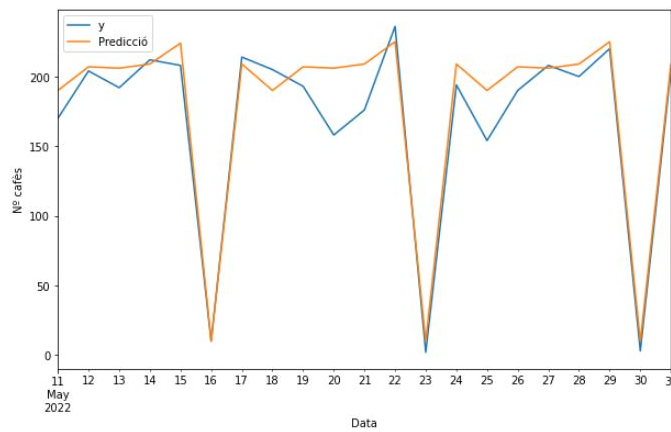
---

Taula resum de les mètriques d'avaluació del client 1			
Mètrica	SARIMA	Prophet	LSTM
MAE	12.9	13.6	12.1
MSE	277.5	328.8	259.7
RMSE	16.6	18.1	16.1
$R^2$	0.94	0.93	0.95

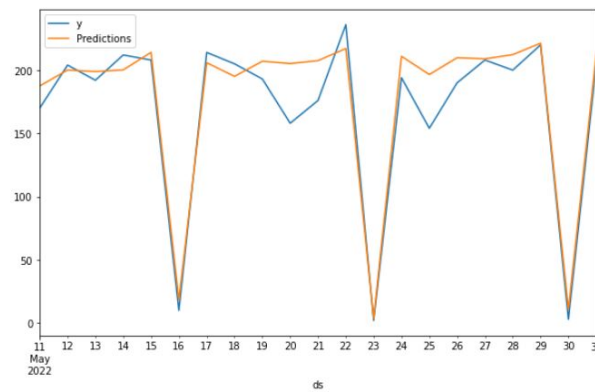
Prediccions obtingudes pel client 1 amb els models: (a) SARIMA (b) Prophet  
(c) Xarxa LSTM.



(a)



(b)



(c)

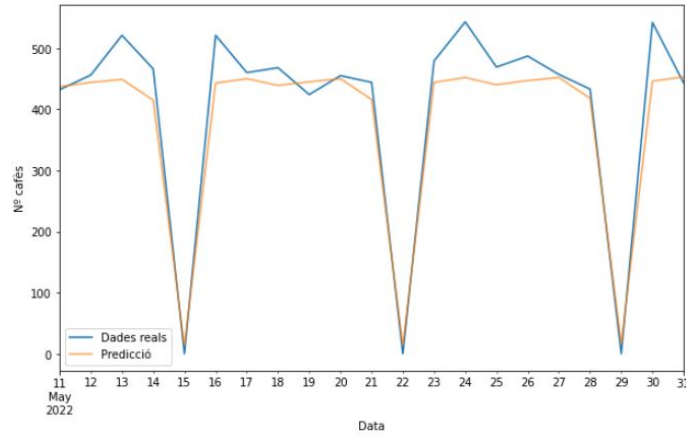
Client 2

Taula resum dels resultats obtinguts pel client 2				
	Dades reals	SARIMA	Prophet	LSTM
2022-05-11	432	437	458	421
2022-05-12	456	444	465	428
2022-05-13	521	449	470	424
2022-05-14	466	415	435	405
2022-05-15	0	13	33	4
2022-05-16	521	443	462	405
2022-05-17	460	450	470	412
2022-05-18	468	439	459	396
2022-05-19	424	445	466	395
2022-05-20	455	450	470	392
2022-05-21	444	416	436	386
2022-05-22	0	14	34	12
2022-05-23	479	444	463	383
2022-05-24	543	452	471	385
2022-05-25	469	440	460	378
2022-05-26	487	447	466	376
2022-05-27	457	452	471	375
2022-05-28	433	418	437	372
2022-05-29	0	16	34	18
2022-05-30	542	446	463	370
2022-05-31	443	453	471	371

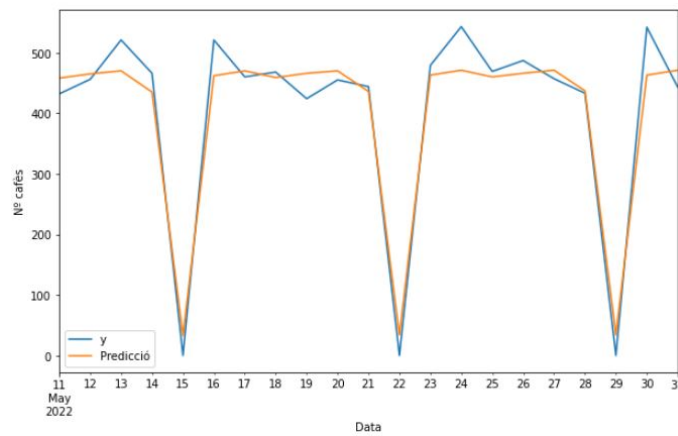
Taula resum de les mètriques d'avaluació del client 2			
Mètrica	SARIMA	Prophet	LSTM
MAE	32.1	28.76	69.5
MSE	1826.8	1268.3	6839.6
RMSE	42.74	35.6	82.7
$R^2$	0.94	0.96	0.76



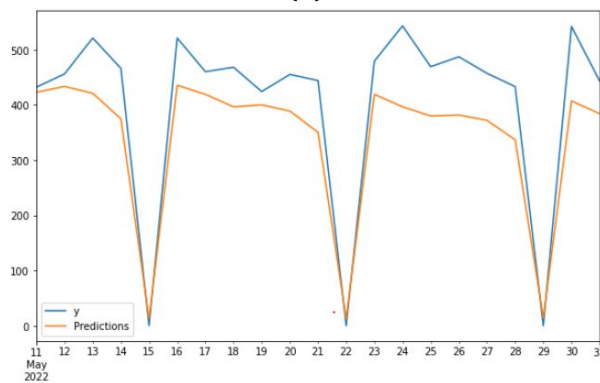
Prediccions obtingudes pel client 2 amb els models: (a) SARIMA (b) Prophet  
(c) Xarxa LSTM.



(a)



(b)



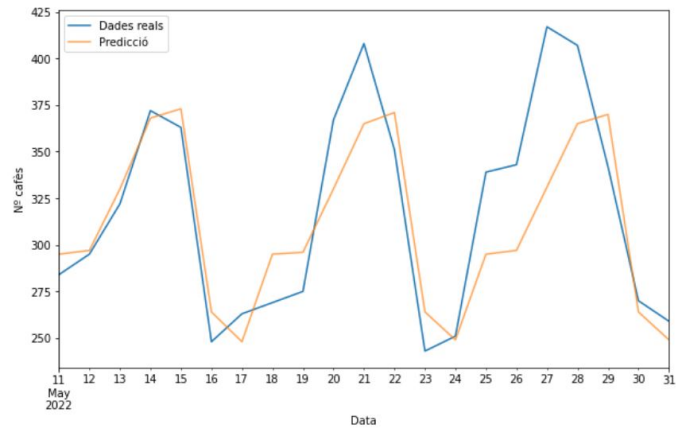
(c)

**Client 3**

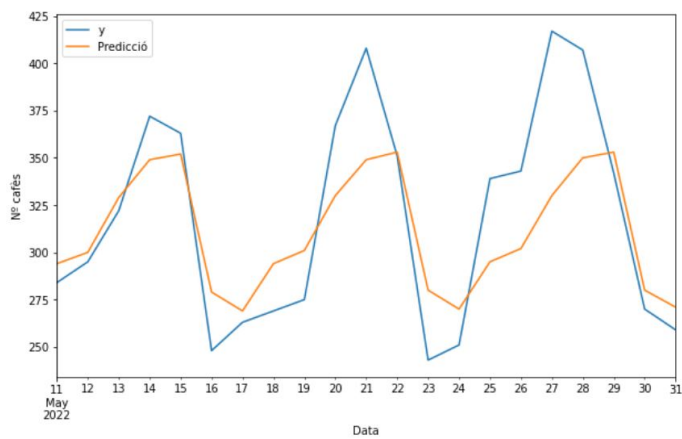
<b>Taula resum dels resultats obtinguts pel client 3</b>				
Data	Dades reals	SARIMA	Prophet	LSTM
2022-05-11	284	295	294	290
2022-05-12	295	297	300	321
2022-05-13	322	330	329	354
2022-05-14	372	368	349	385
2022-05-15	363	373	352	401
2022-05-16	248	264	279	310
2022-05-17	263	248	269	272
2022-05-18	269	295	294	297
2022-05-19	275	296	301	333
2022-05-20	367	330	330	365
2022-05-21	408	365	349	392
2022-05-22	351	371	353	406
2022-05-23	243	264	280	344
2022-05-24	251	249	270	304
2022-05-25	339	295	295	316
2022-05-26	343	297	302	349
2022-05-27	417	331	330	378
2022-05-28	407	365	350	401
2022-05-29	342	370	353	413
2022-05-30	270	264	280	374
2022-05-31	259	249	271	339

<b>Taula resum de les mètriques d'avaluació del client 3</b>			
Mètrica	SARIMA	Prophet	LSTM
MAE	23.7	26.7	39.4
MSE	957.05	1166.9	2480.8
RMSE	30.93	34.2	49.8
$R^2$	0.69	0.63	0.2

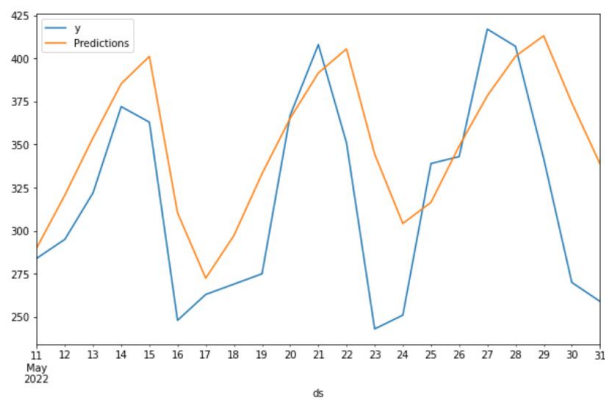
Prediccions obtingudes pel client 3 amb els models: (a) SARIMA (b) Prophet  
(c) Xarxa LSTM.



(a)



(b)



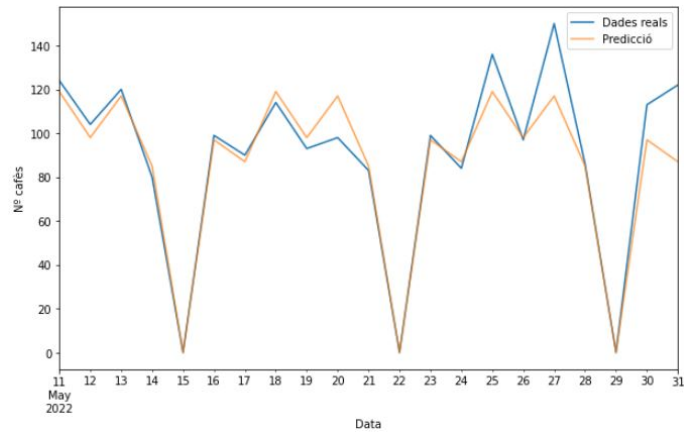
(c)

Client 4

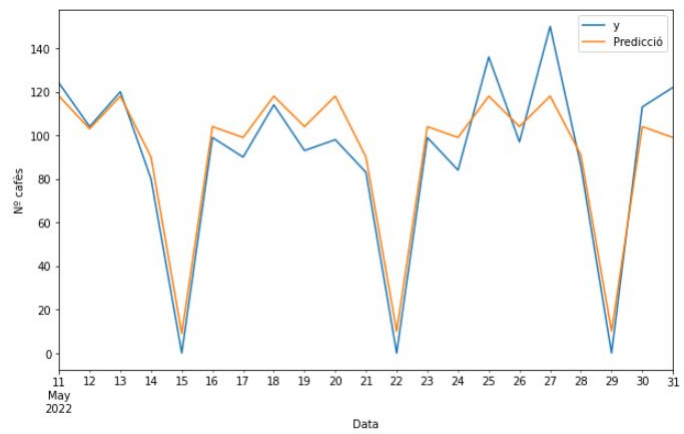
Taula resum dels resultats obtinguts pel client 4				
Data	Dades reals	SARIMA	Prophet	LSTM
2022-05-11	124	119	118	109
2022-05-12	104	98	103	101
2022-05-13	120	117	118	113
2022-05-14	80	85	90	80
2022-05-15	0	0	9	-8
2022-05-16	99	97	104	99
2022-05-17	90	87	99	96
2022-05-18	114	119	118	114
2022-05-19	93	98	104	108
2022-05-20	98	117	118	115
2022-05-21	83	85	90	89
2022-05-22	0	0	10	-7
2022-05-23	99	97	104	99
2022-05-24	84	87	99	97
2022-05-25	136	119	118	117
2022-05-26	97	98	104	109
2022-05-27	150	117	118	117
2022-05-28	86	85	91	92
2022-05-29	0	0	10	-8
2022-05-30	113	97	104	101
2022-05-31	112	87	99	98

Taula resum de les mètriques d'avaluació del client 4			
Mètrica	SARIMA	Prophet	LSTM
MAE	7.8	10.4	10
MSE	161.7	161.7	169.8
RMSE	12.7	12.7	13
$R^2$	0.9	0.9	0.9

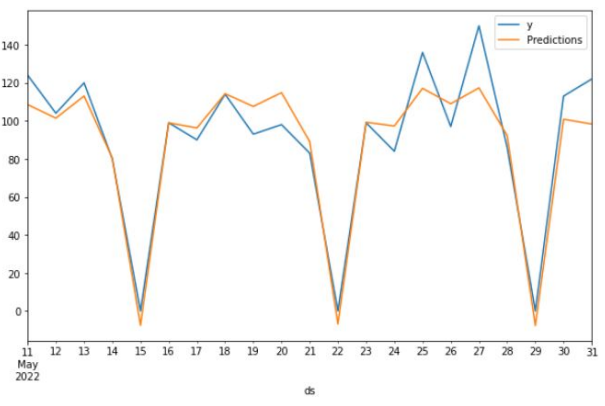
Prediccions obtingudes pel client 4 amb els models: (a) SARIMA (b) Prophet  
(c) Xarxa LSTM.



(a)



(b)



(c)