UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Predicción del precio de venta de los cultivos en invernaderos pasivos utilizando técnicas de aprendizaje automático

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Artes Hernandez, Jose Fernando

Tutor/a: Onaindia de la Rivaherrera, Eva

Cotutor/a externo: CHAMORRO GONZALEZ, JUAN

CURSO ACADÉMICO: 2021/2022

Resum

Hui en dia, quan es subasta la producció a Almería o Alacant, les cooperatives envien a un representant que, coneixedor del context actual, és capaç de predir a quin preu aproximadament licitar cada cultiu. A la zona d'Almería, quasi la totalitat del cultiu es realitza a hivernacles pasius, que no disponen de calefacció ni refrigeració activa, per el que la producció depen directament del clima exterior. Tenint açò en compte, es busca un model que, fent servir el clima a la zona, és capaç de predir el preu futur a la subasta. Per a això, s'obtenen dades climàtiques des de l'aplicació *AgroClimate* que agroturn està desenvolupant, es gasten técniques de *web scraping* per obtindre les pissarres de preus històriques, es realitzen estudis estadístics per estableïr les correlacions entre els paràmetres climàtics y el preu y, per últim, s'estudia el rendiment de diversos models, com els derivats dels ARIMA, reds neuronals recurrents o reds neuronals convolucionals.

Paraules clau: Inteligencia Artificial, Aprendizaje Automático, Redes Neuronales, ARIMA, Invernaderos, Cultivos

Resumen

Hoy en día, cuando se subasta la producción en Almería o Alicante, las cooperativas mandan a un representante que, conocedor del contexto actual, es capaz de decidir a qué precio aproximado pujar cada cultivo. En la zona de Almería, casi la totalidad del cultivo se realiza en invernaderos pasivos, que no disponen de calefacción ni refrigeración activa, con lo que la producción depende directamente del clima exterior. Teniendo esto en cuenta, se busca un modelo que, utilizando el clima en la zona, es capaz de predecir el precio futuro de subasta. Para ello, se obtienen datos climáticos desde la aplicación *AgroClimate* que Agroturn está desarrollando, se emplean técnicas de *web scraping* para obtener las pizarras de precios históricas, se realizan estudios estadísticos para establecer las correlaciones entre los distintos parámetros climáticos y el precio y, por último, se estudia el rendimiento de distintos modelos, como los derivados de los ARIMA, redes neuronales recurrentes o redes neuronales convolucionales.

Palabras clave: Inteligencia Artificial, Aprendizaje Automático, Redes Neuronales, ARIMA, Invernaderos, Cultivos

Abstract

Nowadays, when the production is auctioned at Almeria or Alicante, cooperatives send a manager who, aware of the current context, can decide which price to bid for each crop. At Almeria, almost the whole production is done in passive greenhouses, which don't have active heating or cooling, so production is directly related to external climate. Taking this into account, a model is sought that, using the climate conditions, can predict the auction price. For this purpose, climate data is obtained from *AgroClimate* application that Agroturn is developing, web scraping techniques are used to get the historical auction prices, statistical studies are carried out to establish the correlations between the different climate parameters and the price and, finally, the performance of distinct models, like ARIMA derivates, recurrent neural networks or convolutional neural networks is studied.

Key words: Artificial Intelligence, Machine Learning, Neural Networks, ARIMA, Greenhouse, Crops

Índice general

Índice general v
Índice de figuras vii
Índice de tablas viii

1 Introducción 1
1.1 Motivación . . . 1
1.2 Objetivos . . . 2
1.3 Impacto Esperado . . . 3
1.4 Metodología . . . 3
1.5 Estructura . . . 4
2 Estado del Arte 7
2.1 Predicción de series temporales . . . 7
2.2 Estimación del precio futuro en las Alhóndigas andaluzas . . . 7
2.3 Agroturn . . . 8
2.3.1 Cropture . . . 8
3 Datos 11
3.1 Obtención de datos . . . 11
3.1.1 *Scrapping* web para precios . . . 11
3.1.2 Uso de AgroClimate para obtención del clima . . . 12
3.2 Preprocesado y limpieza . . . 13
3.3 Análisis exploratorio de los datos . . . 14
3.4 Reducción de dimensionalidad . . . 17
3.4.1 Métodos de selección de características . . . 18
3.4.2 Métodos de proyección de características . . . 19
3.4.3 Método escogido . . . 23
4 Modelos 25
4.1 Métricas utilizadas y validación cruzada . . . 25
4.2 Modelo de referencia . . . 26
4.3 Modelos estadísticos básicos . . . 27
4.3.1 SARIMA . . . 27
4.3.2 SARIMAX . . . 29
4.4 Prophet . . . 29
4.5 Redes neuronales (*Deep Learning*) . . . 31
4.5.1 Funcionamiento de una red neuronal . . . 31
4.5.2 Ajuste de hiperparámetros . . . 37
4.5.3 Recurrentes (LSTM) . . . 39
4.5.4 Convolucionales temporales (TCN) . . . 43
4.6 Comparación de resultados . . . 51
5 Conclusiones 53
Bibliografía 55

Apéndices

A Pruebas estadísticas y filtros utilizados **57**
A.1 Pruebas de hipótesis 57
A.1.1 Pruebas de raíz unitaria 57
A.1.2 Prueba Q de Dixon 57
A.2 Filtro de Hampel 57
A.3 Coeficiente de correlación de Pearson 58
B Objetivos de Desarrollo Sostenible (ODS) **59**

Índice de figuras

2.1 Diseño del *Dashboard* de croppture 8
2.2 Selección de la ubicación sobre la que generar el informe climático en AgroClimate 9
2.3 Visualización de las gráficas del reporte y de sus parámetros en AgroClimate 9

3.1 Formato de tabla en la web de *Agroejido* 12
3.2 Ejemplo de valor anómalo en los precios de un mismo día 13
3.3 Serie de precios por día, se puede apreciar la presencia de valores anómalos 13
3.4 Comparación del precio con la temperatura media de Almería tras estandarizar, se aprecia como siguen una misma escala 14
3.5 Precio tras el preprocesamiento y la reducción a frecuencia semanal 14
3.6 Precio de todos los años superpuesto 15
3.7 Descomposición de la serie en tendencia, estacionalidad y residuo 15
3.8 Función de autocorrelación 16
3.9 Función de autocorrelación parcial 16
3.10 Mapa de calor de las correlaciones 17
3.11 Correlaciones de las variables climáticas seleccionadas por el algoritmo personalizado 19
3.12 Matriz de covarianza entre 5 variables 20
3.13 Varianza explicada por cada uno de los componentes principales 21
3.14 Correlaciones de las nuevas características proyectadas por las componentes principales 22
3.15 Correlaciones de las nuevas características obtenidas por el autoencoder . 23

4.1 Validación cruzada con k=5 particiones 25
4.2 Validación cruzada para series temporales con k=5 26
4.3 Resultados del modelo de referencia prediciendo a corto plazo para las 5 particiones 26
4.4 Resultados del modelo de referencia prediciendo a largo plazo para las 5 particiones 27
4.5 Predicción a corto plazo obtenida con Prophet 30
4.6 Predicción a largo plazo obtenida con Prophet 30
4.7 Perceptrón con 3 entradas y una salida, donde f es la función de activación 31
4.8 Sigmoide 32
4.9 TanH 32
4.10 ReLU 32
4.11 Movimiento en la dirección opuesta al gradiente para una función con dos entradas 33
4.12 Función con un mínimo local a la derecha del mínimo global 34
4.13 Error en cada iteración para distintos algoritmos de optimización, se aprecia el rendimiento de Adam 35
4.14 Evolución de los errores sobre el conjunto de test y de entrenamiento con overfitting 36

4.15 Separación de dos clases. En negro, la frontera deseada y en verde, la que presenta overfitting . . . 36
4.16 Ejemplo de distintas tasas de dropout (dr) en capas de una red neuronal . 37
4.17 Búsqueda en rejilla de dos parámetros . . . 38
4.18 Búsqueda aleatoria de dos parámetros . . . 39
4.19 Desplegado de una red neuronal recurrente . . . 39
4.20 Celda LSTM . . . 40
4.21 Parte de la celda que olvida . . . 40
4.22 Parte de la celda que recuerda . . . 41
4.23 Parte de la celda que genera la salida . . . 41
4.24 Arquitectura de la red LSTM . . . 42
4.25 Representación de una imagen con un solo canal (escala de grises) . . . 44
4.26 Capa de convolución . . . 44
4.27 Capa de submuestreo utilizando el máximo . . . 45
4.28 Convolución unidimensional simple con kernel $K^{4,1}$ . . . 46
4.29 Desplazamiento de la convolución unidimensional con kernel $K^{3,1}$ . . . 46
4.30 Convolución con kernel $K^{3,1}$ en la que se utilizan valores futuros . . . 47
4.31 Convolución causal con kernel $K^{3,1}$ . . . 47
4.32 Convoluciónes apiladas todas con kernel $K^{3,1}$ . . . 48
4.33 Dilatación d=2 con kernel $K^{3,1}$ . . . 48
4.34 Múltiples capas convolucionales con dilatación d=2 y kernel $K^{3,1}$ . . . 49
4.35 Dilataciones dinámicas con kernel $K^{3,1}$ y dilatación base d_{base}=2 . . . 49

Índice de tablas

3.1 Correlaciones tras el método de selección personalizado . . . 24
3.2 Correlaciones tras aplicar PCA . . . 24
3.3 Correlaciones tras emplear el autoencoder . . . 24

4.1 Posibles valores de los hiperparámetros de la red LSTM . . . 43
4.2 Valores óptimos de los hiperparámetros de la red LSTM para predicción a corto plazo . . . 43
4.3 Valores óptimos de los hiperparámetros de la red LSTM para predicción a largo plazo . . . 43
4.4 Posibles valores de los hiperparámetros de la red TCN . . . 50
4.5 Valores óptimos de los hiperparámetros de la red TCN para predicción a corto plazo . . . 50
4.6 Valores óptimos de los hiperparámetros de la red TCN para predicción a largo plazo . . . 50
4.7 Comparación de los resultados (MAE) para cada uno de los modelos probados . . . 51

CAPÍTULO 1
Introducción

La agricultura ha sido y es uno de los principales motores económicos en nuestro país y en gran parte del mundo, así como una de las principales actividades que garantizan la seguridad alimentaria. Tras la pandemia por el COVID-19, fue uno de los únicos sectores que se expandió, llegando a representar el 3.4 % del PIB de España [1] y generando 58000 nuevos puestos de trabajo durante 2021, siendo uno de los sectores con mayor ganancia [2].

Pese a ello, la agricultura no está exenta de problemas, durante los años 90 disminuyó la producción agrícola per cápita, debido a una disminución de la inversión y el interés en este sector, lo que frenó su evolución; además, los países menos desarrollados no participan en los mercados agrícolas internacionales y su producto agrícola interior tiene que competir con las importaciones exteriores [3]. La mayoría de países del mundo todavía utilizan las metodologías clásicas de la agricultura y no se atreven a implementar las nuevas tecnologías, ya sea por los altos costes de implantación, la falta de conocimiento o porque sencillamente no conocen las ventajas que estas tecnologías les pueden aportar.

En nuestro país disponemos de una extensa explotación agrícola, alcanzando su máximo exponente en la provincia de Almería, que dispone de más de 30.000 hectáreas de invernaderos (agricultura intensiva), lo que es la mayor superficie invernada del planeta [4].

En Málaga, Granada y, principalmente, en Almería, existen las llamadas Alhóndigas, que son los centros de contratación privados en origen, es decir, los centros donde se produce la oferta de los productos agrícolas por parte de los agricultores para su compra por corredores o comisionistas en un mercado oligopólico. También, las propias Alhóndigas se encargan de exportar parte del producto al extranjero. El método usado es el de subasta a la baja, siendo los vendedores los que pujan con precios bajos para vender su cosecha.

1.1 Motivación

Pese a la evolución tecnológica que ha sufrido nuestra sociedad en los últimos años, la agricultura se ha quedado atrás en muchos aspectos, manteniendo procedimientos obsoletos y desperdiciando la gran cantidad de datos que se generan. Esto es debido, en parte, al poco o nulo interés de los más jóvenes en el mundo agrícola, ya que, en 2016, más de un 40 % de titulares de explotaciones agrarias en España eran mayores de 65 años, y solo un 3 % de ellos eran menores de 35 según la encuesta publicada en 2016 por el INE [5]. La empresa Agroturn Research busca aplicar las últimas tecnologías, como la inteli-

gencia artificial y la realidad virtual a las explotaciones agrícolas y plantear soluciones innovadoras a problemas históricos del sector.

En España, la amplia mayoría de invernaderos de la provincia de Almería son invernaderos pasivos, es decir, no disponen de demasiadas herramientas para controlar las condiciones climáticas dentro del invernadero, por lo que estas dependen en gran medida del clima exterior. El éxito de este tipo de explotaciones es posible debido a las bondades del clima mediterráneo, aunque este no está exento de algún periodo con condiciones convulsas.

Para que el cultivo crezca correctamente, necesita unas condiciones climáticas determinadas, fuera de las cuales el cultivo puede dañarse y sufrir alteraciones. En el manual publicado por Jaramillo Noreña et al. (2016) [6] se exponen algunos de los desórdenes fisiológicos que puede sufrir el tomate durante su cultivo, la mayoría de los cuales son producto de condiciones climáticas adversas.

Es fácil, por tanto, establecer una relación entre las condiciones climáticas exteriores y la calidad del cultivo en los invernaderos de la zona de Almería. Un cultivo de baja calidad es o bien descartado o bien vendido a un precio menor como lote dañado, por lo que la cantidad de cultivo subastada tras un periodo climático adverso es menor y, por tanto, las cooperativas están dispuestas a pagar más por ellos. Esta afirmación está respaldada por De Pablo et al. (2002) [7], cuyo estudio sobre las Alhóndigas y los factores que influyen en los precios determinó que el clima es uno de los factores principales, junto a la demanda exterior, las importaciones, la calidad del cultivo y información privilegiada que poseen los corredores y comisionistas.

Asimismo, la demanda exterior y las importaciones dependen en gran medida de la producción realizada en aquellos países desde los que más se importa y a los que más se exporta, con lo que el precio también depende de la cantidad y calidad del cultivo en esos lugares y, por tanto, de sus condiciones climáticas; un país que importe mucho producto desde España probablemente disminuya su demanda si la producción interna ha sido superior a la media debido a unas buenas condiciones climáticas.

De igual manera, cabe destacar la investigación realizada por Meshram et al. (2021) [8], en la que recorren el ciclo de vida de una planta en la agricultura intensiva, identificando aquellos problemas que las técnicas de Machine Learning son capaces de resolver; entre ellos no se menciona como problema a solucionar la predicción del precio de venta del cultivo, siendo uno de los parámetros más importantes pues los beneficios de una explotación agrícola dependen directamente del precio al que se pueda vender el cultivo.

Por último, la capacidad de predecir el precio del cultivo a futuro es fundamental para los agricultores, ya que pueden establecer las estrategias de cultivo y estimar los beneficios que van a obtener, y también lo es para los corredores y comisionistas, pues pueden tomar decisiones sobre sus pujas de forma más precisa.

1.2 Objetivos

Los objetivos principales son:

- Predecir el precio de venta del cultivo a corto plazo (siguiente semana).
- Predecir el precio de venta del cultivo a largo plazo (próximo año).

Los objetivos secundarios son:

- Estimar si la inflación realmente afecta al precio.

- Sanear correctamente los datos, eliminando valores anómalos e interpolando los datos que faltan.
- Clasificar los parámetros climáticos en base al nivel de correlación con el precio o reducir su dimensionalidad a un número más accesible.

1.3 Impacto Esperado

Para definir el impacto esperado primero tenemos que identificar a los actores que participan en la subasta del cultivo:

- Agricultor principal (gestor invernadero): Controla el invernadero y su papel en la subasta es vender todo su cultivo.
- Cooperativas, corredores y comisionistas: Su papel en la subasta es obtener cultivo para venderlo y obtener beneficios.

Estos actores tienen deseos enfrentados, pues a los agricultores les interesa vender el producto al precio más alto posible, y a las cooperativas les interesa justo lo contrario.

Agricultor principal (gestor invernadero)

Para un agricultor es fundamental conocer el precio al que va a poder vender su cultivo con antelación, pues así puede plantear su estrategia en base a ello. Por ejemplo, durante una temporada con clima adverso, el agricultor puede plantearse instalar nuevas ventanas en su invernadero que mejoren las que ya dispone (permitan entrar más luz), esto conlleva una inversión que es difícil valorar en términos económicos sin saber a ciencia cierta el beneficio que se obtendrá tras la mejora de la producción. También podría decidir que días de la semana vender su cultivo para obtener el máximo beneficio posible.

Cooperativas, corredores y comisionistas

Una cooperativa puede decidir su estrategia de compra de cultivo en base a los precios actuales y futuros, es decir, que cantidad comprar y que día hacerlo. También podría decidir su propio precio minorista en cada momento para maximizar el beneficio.

1.4 Metodología

Los primeros pasos en todo estudio de modelos basados en inteligencia artificial o en estadística son generalmente los mismos:

1. Buscar una fuente de datos fiable y suficiente.
2. Obtener los datos de dicha fuente mediante distintas técnicas (en este caso, scrapping).
3. Preprocesar y limpiar los datos obtenidos, eliminando valores anómalos y obteniendo una serie más representativa.
4. Realizar un análisis exploratorio de los datos, descubriendo patrones y características de la serie.

Tras estos pasos, se dispone de una serie de datos limpia y lista para empezar a entrenar modelos con ella y se han identificado patrones y características de los datos que facilitan la elección y la configuración de los modelos.

Posteriormente, es fundamental estudiar el estado del arte de la disciplina en cuestión, en este caso, la predicción de series temporales, con el fin de acotar la búsqueda de modelos y de parámetros a un conjunto más reducido. En esta memoria, por coherencia narrativa, este estudio se realiza antes que el de los datos.

Una vez identificados los modelos a evaluar, se realiza lo propio dividiendo, en general, el estudio en dos apartados:

- Modelo (construcción y parametrización).
- Resultados (comparando con los modelos previos).

Por último, se exponen las conclusiones obtenidas.

1.5 Estructura

- Estado del arte
  - Predicción de series temporales: Recorrido por los distintos modelos de predicción de series temporales a lo largo de la história y el efecto de la irrupción del *Deep Learning* sobre ellos.
  - Estimación del precio futuro en las Alhóndigas andaluzas: Se repasan los métodos actuales que se emplean para predecir los precios de la subasta agrícola.
  - Agroturn: En esta sección, se dan a conocer tanto a la empresa Agroturn como a sus productos, entre ellos AgroClimate, siendo esencial para los datos climáticos.
- Datos
  - Obtención de datos: En este apartado, se documenta la búsqueda de las fuentes de datos adecuadas para este proyecto y la implementación de algoritmos para obtener dicha información.
  - Preprocesado y limpieza: Análisis de los datos para filtrar valores no deseados y escalar la serie.
  - Análisis exploratorio de los datos: En esta sección, se hace un estudio sobre los datos ya preprocesados para extraer conclusiones que ayuden a elegir los modelos a evaluar.
  - Reducción de dimensionalidad: Se plantean y comparan distintas técnicas para reducir la dimensionalidad de los datos.
- Modelos
  - Métricas utilizadas y validación cruzada: Se introducen las métricas generales a utilizar y el método de validación cruzada.
  - Modelos de referencia: Se plantea un modelo básico sobre el que evaluar el rendimiento de los siguientes.
  - Modelos estadísticos básicos: En este apartado, se estudia el rendimiento de los modelos clásicos de predicción de series temporales.

    - Prophet: En esta sección, se estudia el rendimiento de la librería de predicción de series temporales Prophet.
    - Redes neuronales: Se prueban modelos basados en redes neuronales.
    - Comparación de resultados: Se escogen los modelos que han rendido mejor y se analizan los resultados.

- Conclusiones

- Apéndices

    - Pruebas estadísticas y filtros utilizados: Extensión y desarrollo sobre las pruebas y filtros que se han empleado en los modelos.
        - Pruebas de hipótesis
        - Filtro de Hampel
    - Objetivos de Desarrollo Sostenible (ODS)

CAPÍTULO 2
Estado del Arte

2.1 Predicción de series temporales

La predicción de series temporales es una técnica de análisis exploratorio que consiste en la extracción de patrones y características de una serie temporal, para luego utilizarlas para construir un modelo que pueda predecir el futuro. Durante los últimos años, con la irrupción del *Deep Learning*, se han desarrollado modelos nuevos que se suman a los ya clásicos y que obtienen resultados sorprendentes.

Verma et al. (2021) [9] realizaron un exhaustivo estudio comparando los metodos clásicos de predicción de series temporales con los más modernos, con el fin de predecir el índice de calidad del aire en Delhi, la capital de India; concluyendo que los modelos basados en *Deep Learning* (LSTM) obtienen mejores resultados que los métodos clásicos estadísticos. Sin embargo, la mayoría de estudios que señalan la superioridad de los métodos de *Deep Learning* disponen de una gran cantidad de datos, y en aquellos en los que estos son limitados, la diferencia es mucho más reducida hasta el punto de que en ocasiones, un modelo básico puede ganar a una red neuronal, como se puede ver en el estudio de Zhang et al. (2022) [10].

2.2 Estimación del precio futuro en las Alhóndigas andaluzas

En la actualidad, la estimación del precio de subasta de los cultivos de la zona de Almería, se realiza principalmente tomando como base los precios previos, donde se estima el precio de la siguiente semana como el precio de la semana actual ($p_s = p_{s-1}$), y el precio del año próximo como el precio del año anterior ($p_{a,s} = p_{a-1,s}$). Como se puede apreciar, no se utiliza ninguna técnica de predicción muy avanzada, y tampoco se emplean estas predicciones para tomar decisiones en base a ellas.

En otros países con más alta tecnología en invernaderos, como Holanda, Suecia o Canadá, los precios se estiman con plantillas ofrecidas por los bancos inversores, que generalmente están desactualizadas y utilizan algoritmos simples como la media de años anteriores o incluso correcciones manuales a los precios del año anterior. Estas estimaciones solo se realizan a largo plazo y solo al inicio de un proyecto para estimar los retornos económicos, aplicando al año base predicho la inflación para ajustar el precio en los siguientes años, pese a que como se podrá observar más adelante, el precio no tiene una tendencia clara y por tanto la inflación no afecta a este de la forma que se cree.

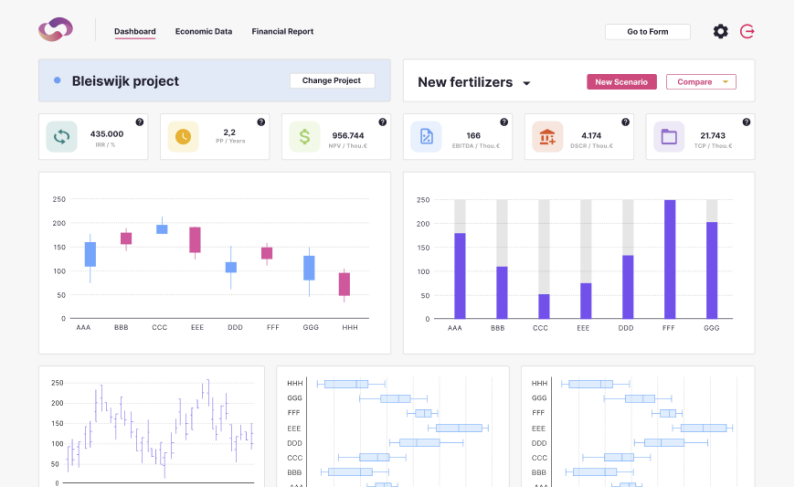

Figura 2.1: Diseño del *Dashboard* de cropture

2.3 Agroturn

La empresa Agroturn B.V. nace para hacer frente a los problemas de gestión de invernaderos en el sector agrícola. Ofrece soluciones en el diseño e implementación de distintos sistemas del invernadero, como la aclimatación, el riego, la iluminación, etc; además de ofrecer inteligencia y intermediación en muchos otros aspectos del sector.

Desde 2020, Agroturn ha decidido orientarse hacia la innovación y buscar la forma de aplicar las últimas tecnologías en el sector; es por ello que está desarrollando algoritmos de inteligencia artificial para la detección de el estado de salud de los frutos, para la optimización de los procesos de producción, para la detección de los movimientos de los trabajadores, entre otros. También ha diseñado sistemas de realidad virtual y realidad aumentada que permiten previsualizar un invernadero antes de estar construido. Por último, ha desarrollado una plataforma de gestión de invernaderos, **Cropture**, que permite administrar los proyectos desde una aplicación web intuitiva, y que ofrece distintos servicios dentro de ella, como la herramienta **AgroClimate**.

2.3.1. Cropture

Cropture

Logo de Cropture

Cropture es una plataforma que permite gestionar los proyectos de invernadero desde una interfaz web sencilla e intuitiva, esto rompe con las anteriores soluciones que existían, que eran más complejas o que incluso necesitaban usar hojas de cálculo en distintas ocasiones para almacenar los datos. En Cropture se diseña el invernadero, eligiendo sus dimensiones y materiales, sus sistemas de aclimatación, riego y iluminación y los cultivos que se van a utilizar; también se introducen los parámetros económicos y, con todo esto, mediante algoritmos se calcula la producción del cultivo, los costos, y por tanto, los beneficios finales. Estas métricas se muestran en un panel de control 2.1 personalizado, donde el usuario ve los parámetros más relevantes y es capaz incluso de comparar distintos proyectos para estudiar su viabilidad.

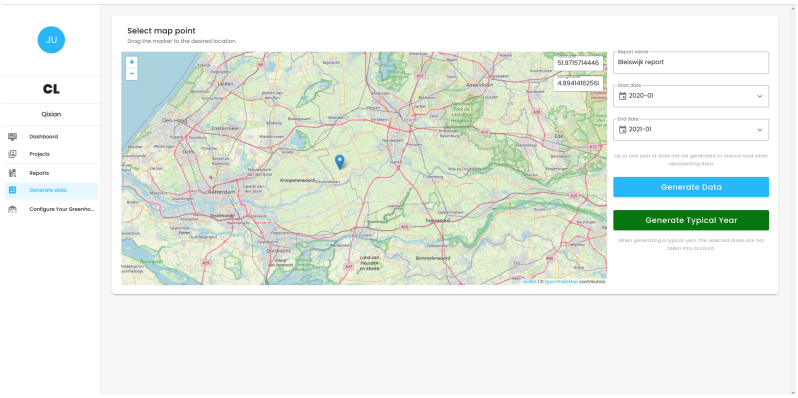

Figura 2.2: Selección de la ubicación sobre la que generar el informe climático en AgroClimate

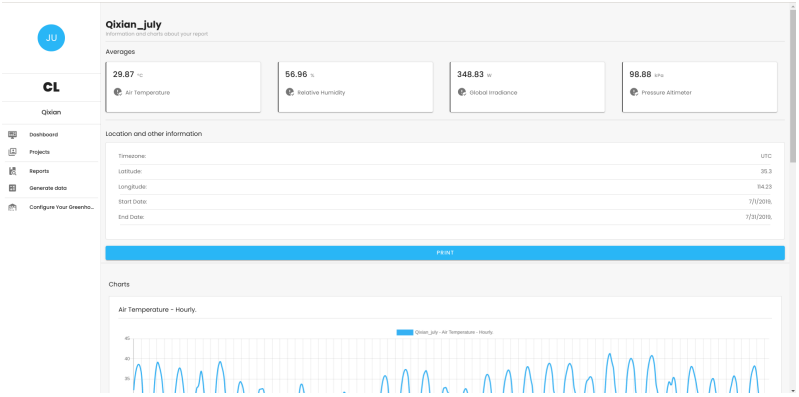

Figura 2.3: Visualización de las gráficas del reporte y de sus parámetros en AgroClimate

AgroClimate

AgroClimate es una herramienta que permite obtener datos climáticos para un periodo de tiempo determinado y generar informes en base a los mismos. Esta aplicación obtiene los datos de la *API* abierta *POWER* ofrecida por la NASA (https://power.larc.nasa.gov/), y calcula nuevos parámetros en base a la configuración del inverndero del usuario, ofreciendo en detalle información de los consumos, energía, temperatura, humedad, etc.

El motor de AgroClimate está integrado en Cropture, de forma que el cliente no genera explícitamente informes sino que estos se generan automáticamente. AgroClimate también se ofrece como una aplicación independiente, en el que el usuario puede configurar los parámetros básicos del invernadero, seleccionar un punto geográfico sobre el que generar el informe 2.2 y visualizar un reporte personalizado con las figuras necesarias y los resultados más importantes 2.3.

CAPÍTULO 3
Datos

3.1 Obtención de datos

El objetivo es predecir el precio del cultivo en base al clima, con lo que vamos a necesitar dos tipos de datos:

- Precio de los cultivos
- Variables climáticas para todas las ubicaciones relevantes

Las Alhóndigas han sufrido un proceso de modernización, recopilando el histórico de precios en sus sitios webs; pese a ello, la mayoría de centros han empezado a hacerlo muy recientemente y no tienen datos con la suficiente antigüedad. La Alhóndiga *AgroEjido* (https://www.agroejido.com/), dispone de una de las mejores bases de datos de precios del sector, recopilando pizarras de precios desde el año 2006 hasta la actualidad. No obstante, hay pocos cultivos que tienen datos de forma consistente desde 2006, y también es importante elegir un cultivo que tenga una temporada muy larga para maximizar la información disponible. Con estos requisitos, se observó que el **calabacín** era el mejor por tener una temporada de unos 10 meses al año (septiembre-junio) y tener suficientes datos desde el inicio, con lo que todos los estudios posteriores se van a realizar utilizando los precios de este cultivo. Por otro lado, España es uno de los principales exportadores de calabacín, siendo sus principales clientes Francia y Holanda. España también importa mucho calabacín desde Portugal, Marruecos y Francia.

3.1.1. *Scrapping* web para precios

La Alhóndiga *AgroEjido* no dispone de una fuente de datos en formato fácilmente legible por un algoritmo, si no que ofrecen la información en tablas HTML 3.1. Para cada fecha, se muestran las distintas ubicaciones donde la Alhóndiga opera, que se pueden considerar como una sola, es decir, los precios para un mismo cultivo en las distintas ubicaciones sigue la misma distribución matemática. Con el fin de extraer todos estos datos de forma sencilla, se usa la herramienta *Selenium* (https://www.selenium.dev/) junto con el lenguaje de programación *Python*; con una sencilla rutina, se itera sobre todas las fechas desde el 1 de enero del 2007 hasta el 31 de diciembre del 2021, se unen todos los precios del calabacín de un mismo día para todas las ubicaciones, y se extrae en un archivo *csv* con dos columnas: fecha y precio, en la que la columna precio es una lista de todos los precios que se han registrado para el cultivo en el mismo día.

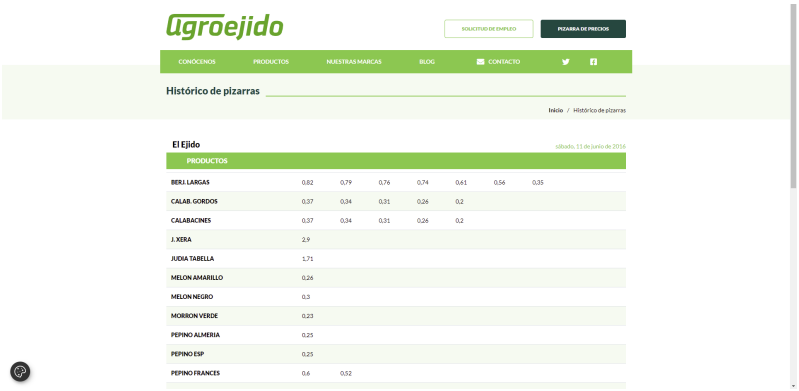

Figura 3.1: Formato de tabla en la web de *Agroejido*

3.1.2. Uso de AgroClimate para obtención del clima

Gracias a la herramienta AgroClimate, es muy sencillo obtener los datos climáticos con una frecuencia diaria para las ubicaciones deseadas en el mismo intervalo que los precios, es decir, desde el año 2007 hasta el 2021. AgroClimate distingue entre dos tipos de reportes: básico y completo; siendo el completo una extensión del primero añadiendo parámetros específicos para el invernadero que el usuario ha configurado; en este caso, se emplea el básico, pues se desea obtener los parámetros climáticos generales de Almería, Portugal, Marruecos, Holanda y Francia. Para estos países se escogen aquellas zonas con mayor presencia de invernaderos de Calabacín. El reporte básico contiene los siguientes parámetros:

- Temperatura del aire y del cielo (°C)
- Temperatura de punto de rocío (°C)
- Temperatura de bulbo seco (°C)
- Humedad absoluta (g/m^3)
- Humedad relativa (%)
- Cobertura de nubes (%)
- Radiaciones solar, difusa, directa y global (W/m^2)
- Presión ajustada a la elevación (kPa)
- Precipitación corregida (l/m^2)
- Velocidad (m/s) y dirección (°) del viento.

Para cada una de las ubicaciones. Para cada parámetro y cada día, se dispone de la siguiente información:

- Valor máximo
- Valor mínimo
- Valor medio
- Valor acumulado

CALABACINES	0,57	0,53	0,51	0,48	0,35

Figura 3.2: Ejemplo de valor anómalo en los precios de un mismo día

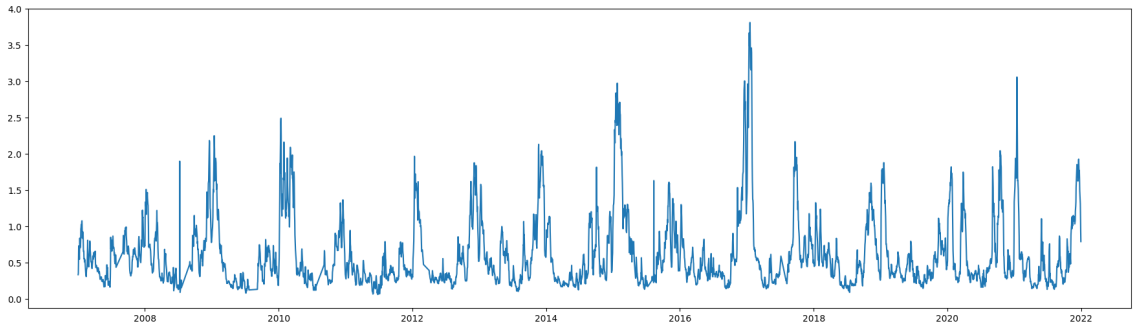

Figura 3.3: Serie de precios por día, se puede apreciar la presencia de valores anómalos

3.2 Preprocesado y limpieza

El primer paso es obtener la media del precio para cada día, pues se dispone de una lista con el precio al que se han vendido los distintos lotes en un mismo día. Analizando dichas listas, se identifica la presencia de valores anómalos que generalmente son bastante más pequeños que los demás 3.2, debiéndose a que puede ser un lote de mala calidad; es necesario, por tanto, buscar una forma de detectar valores anómalos en series con pocos valores, ya que para cada día se disponen de entre 3 y 10 precios en total, que son todos los precios de los lotes en las distintas ubicaciones donde opera la Alhóndiga. El algoritmo elegido es la **Prueba Q de Dixon**, que es un algoritmo que permite el filtrado de valores anómalos en series con muy pocos datos, aunque solo se puede aplicar para series con más de 5 valores. Una vez realizado este primer filtro, se obtiene la media diaria. Posteriormente, al volver a observar los datos, se puede apreciar que siguen habiendo días con un precio claramente anómalo 3.3, que puede ser debido o bien porque ese día había menos de 5 valores y algunos de ellos eran anómalos, impidiendo la aplicación del test de Dixon y afectando al precio final; o bien porque simplemente es un error o un día especial. Esto también puede ocurrir en la serie de datos climáticos, donde puede haber días con valores anómalos. Para solucionar esto, se aplica el **filtro de Hampel**, que es un método para filtrar valores anómalos que funciona especialmente bien en series temporales y luego se aplica interpolación lineal para rellenar los huecos. Es importante excluir de este filtro aquellas características climáticas que, por su naturaleza, pueden ser altamente variables y tener valores que, pese a que filtros como el de Hampel consideren como anómalos, no lo son; estas variables son la precipitación, el viento y la nubosidad. Por último, se reduce la serie (clima y precio) de frecuencia diaria a frecuencia semanal, utilizando para ello la media.

Estandarización Para cada una de las variables de la serie obtenidas (precio y parámetros climáticos) se aplica un escalado para estandarizar los datos y convertirlos en una distribución con media cero y desviación estándar 1. Este escalado se aplica con la siguiente fórmula para cada muestra x:

$$z = \frac{(x - u)}{s}$$

Donde u es la media y s la desviación estándar de la serie.

Esto permite tener todas las características con una escala similar 3.4, que facilita a los modelos la generalización y evita que le den más importancia a algunas variables que por su unidad eran mayores.

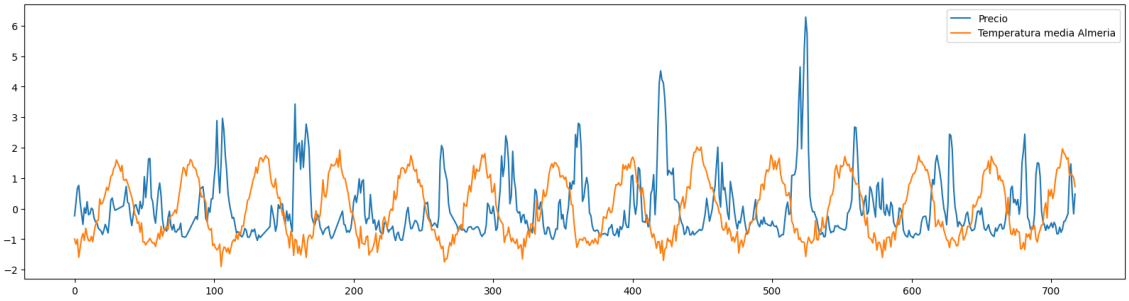

Figura 3.4: Comparación del precio con la temperatura media de Almería tras estandarizar, se aprecia como siguen una misma escala

3.3 Análisis exploratorio de los datos

Analizando los datos del precio tras el preprocesamiento 3.5, podemos observar que no existen valores anómalos en la serie, y que la reducción por semanas ha suavizado la misma. Se observa una clara estacionalidad, ya que se repite un patrón cada año en el que el precio en invierno es mucho más alto que el precio en verano. Para comprobar esto, se muestran los gráficos para cada año superpuestos 3.6, donde claramente se aprecia la estacionalidad mencionada. Por otro lado, no parece apreciarse una tendencia definida del precio a lo largo de los años, lo que choca con el hecho de que en algunos países se aplique una tasa de inflación fija para estimar los precios a largo plazo.

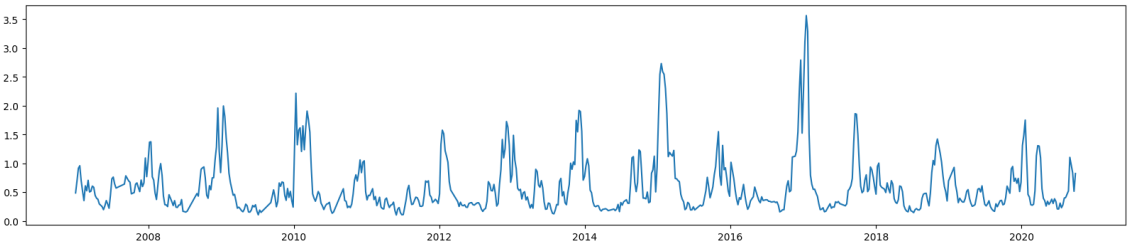

Figura 3.5: Precio tras el preprocesamiento y la reducción a frecuencia semanal

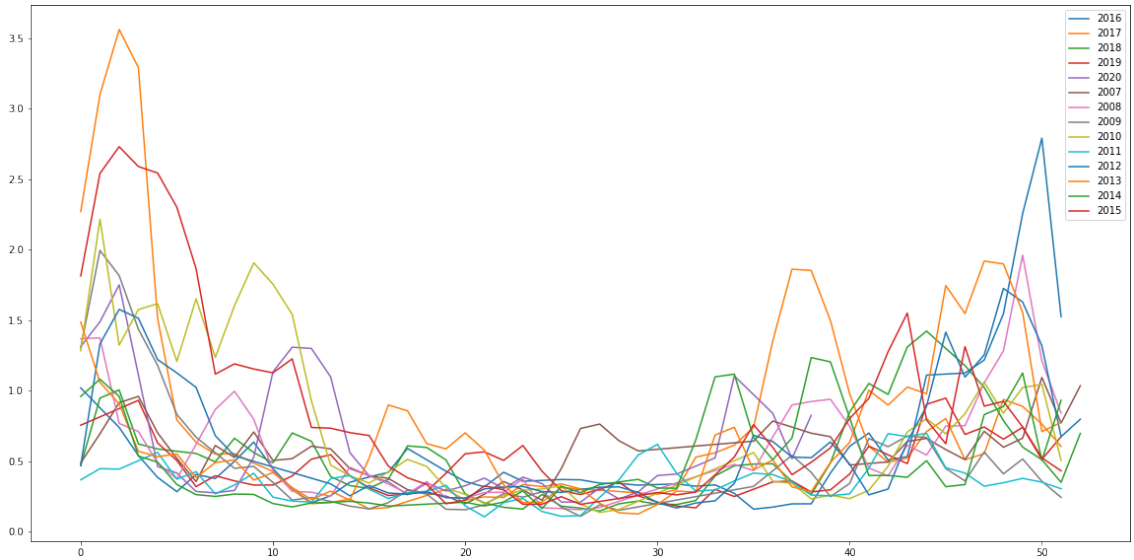

Figura 3.6: Precio de todos los años superpuesto

Posteriormente, se descompone la serie en tendencia, estacionalidad y residuo; visualizando estas tres componentes se pueden confirmar las conclusiones anteriores sobre estacionalidad y tendencia 3.7.

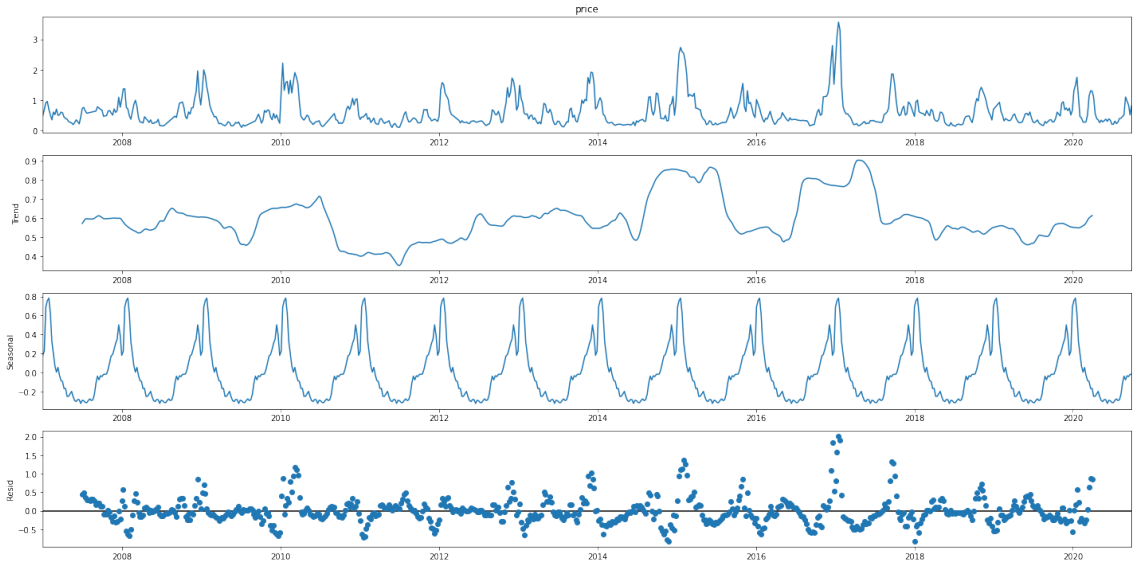

Figura 3.7: Descomposición de la serie en tendencia, estacionalidad y residuo

La **estacionariedad** es una propiedad de aquellas series temporales cuyas características no varían con el tiempo, eso es, que no contienen estacionalidad ni tendencia. Es importante conocer si la serie cumple esta propiedad pues de lo contrario hay que tomar ciertas medidas para que los modelos tengan cuenta de ello. En este caso, se ha concluido anteriormente que la serie tiene estacionalidad, pero para confirmar este supuesto se utiliza el test de Dickey-Fuller aumentado [11] y el test KPSS, ambos son **pruebas de raíz unitaria**, donde la hipótesis nula es que la serie no es estacionaria.

El *p-value* resultado del test de Dickey-Fuller aumentado es 0.0, con lo que al ser menor que 0.05 que es el intervalo de confianza por debajo del cual se rechaza la hipótesis nula, se puede afirmar que la serie es estacionaria. El *p-value* resultado de KPSS, sin embargo, es de 0.1, con lo que no podemos asumir que la serie es estacionaria. Esta dife-

rencia es debida a que el test de Dickey-Fuller se basa especialmente en la presencia de tendencia para determinar su resultado, mientras que el test KPSS tiene más en cuenta la estacionalidad.

A continuación, se analizan las funciones de autocorrelación, tanto la total (ACF) como la parcial (PACF). Estas funciones son útiles para identificar los parámetros de los modelos posteriores, sobretodo los de los modelos SARIMA. En ellas, se muestran las correlaciones entre la serie de datos y la misma serie desplazada. La diferencia entre la ACF y la PACF es que en esta última, para un desplazamiento determinado, se elimina la correlación que pueden explicar los desplazamientos más cortos, manteniendo solo la independiente; por ejemplo, la PACF para el desplazamiento 3 es la correlación que explica este desplazamiento pero que no lo hacen los desplazamientos 1 y 2. En este caso, vemos que la función de autocorrelación total 3.8 tiene una forma geométrica, con un periodo de aproximadamente 52 (la estacionalidad) y la función de autocorrelacion parcial 3.9 tiene 3 desplazamientos significativos y cada 52 valores vuelven a haber valores significativos.

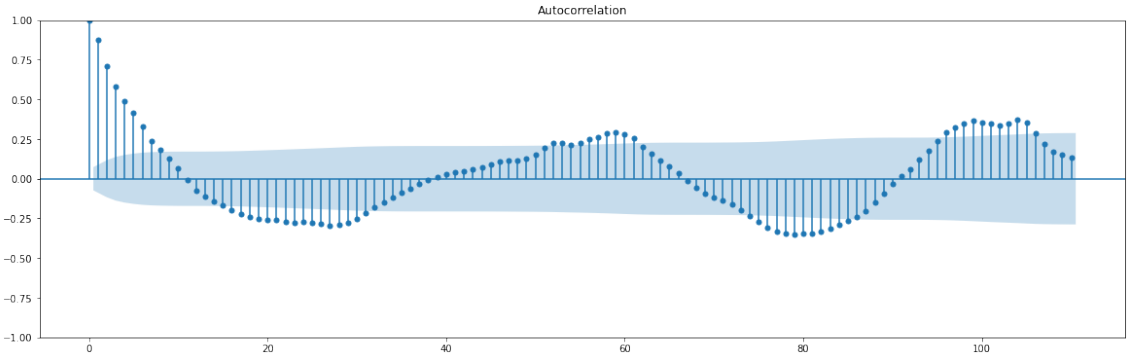

Figura 3.8: Función de autocorrelación

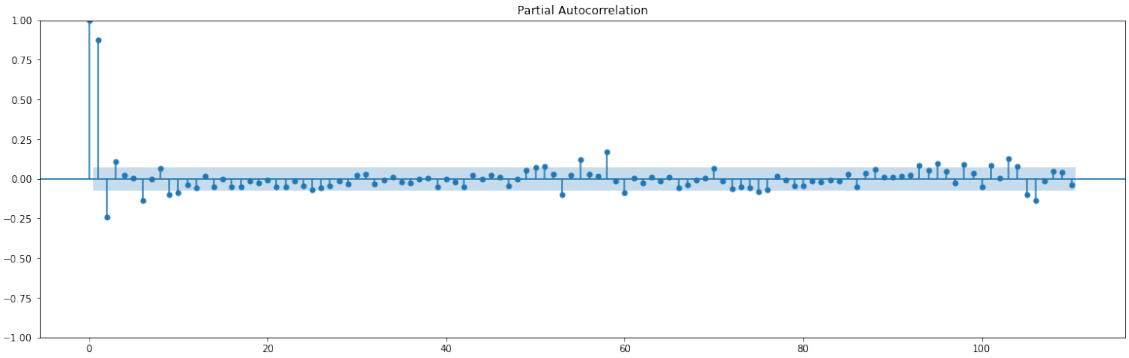

Figura 3.9: Función de autocorrelación parcial

Posteriormente, es interesante determinar que parámetros climáticos están más correlacionados con el precio. Para ello, se utiliza el **coeficiente de correlación de Pearson**.

Esta correlación puede estar desincronizada, es decir, puede ser que dos series temporales estén muy correlacionadas pero sólo si desplazamos una de ellas. En este caso, por lógica y conocimiento previo, se puede asumir que la mayor correlación entre precio y clima será cuando el clima esté desplazado una semana hacia detrás ($C_t = C_{t+1}$), es decir, el precio en una semana t depende del clima en la semana $t-1$. Con todo esto, se calculan las correlaciones entre todas las parejas de variables, se ordenan por la correlación con respecto al precio y se muestran en un mapa de calor las 18 más relevantes 3.10.

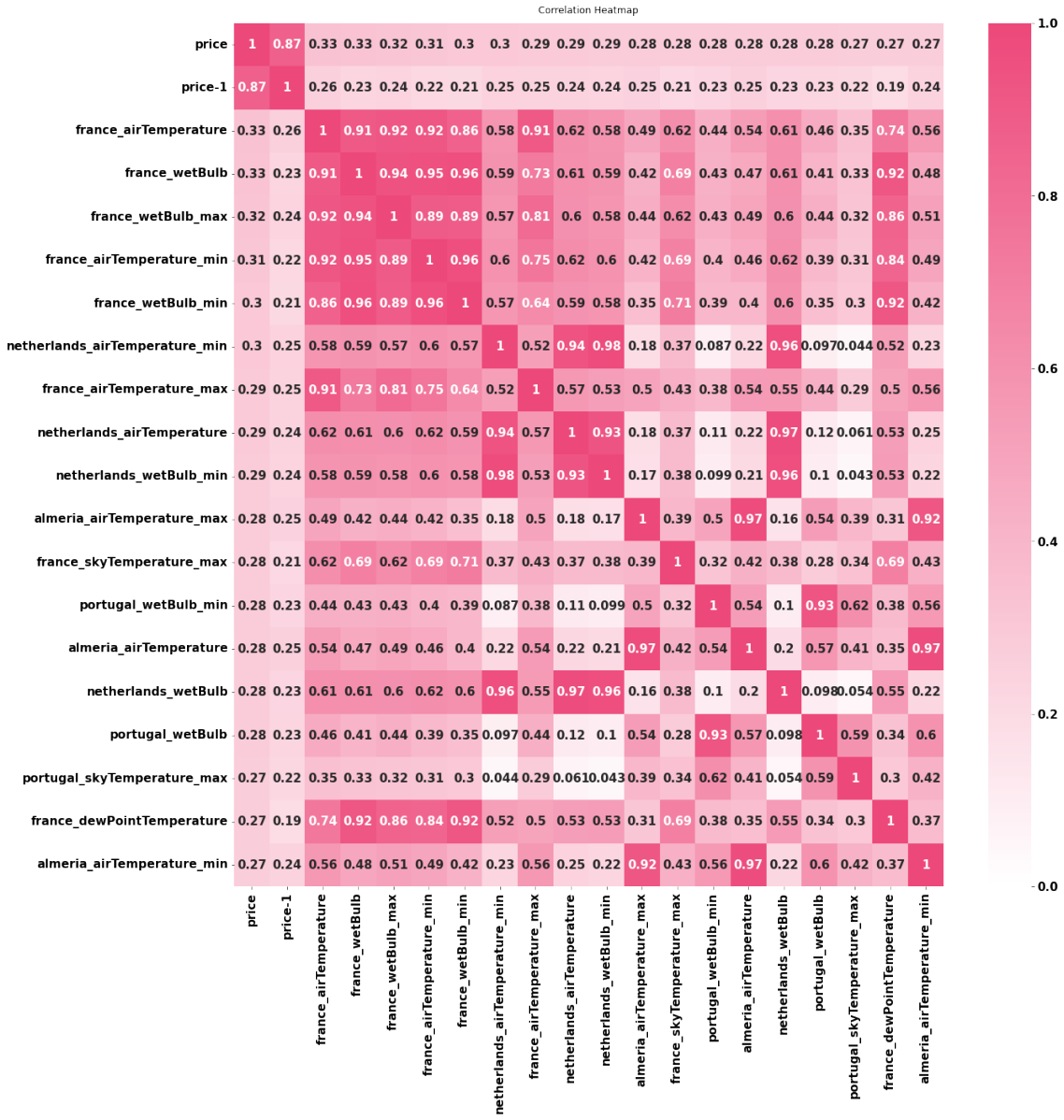

Figura 3.10: Mapa de calor de las correlaciones

Con el mapa de calor, se confirma la afirmación previa sobre que el precio esta correlacionado con el clima de la semana anterior, ya que las correlaciones con el precio son mayores que con precio-1, que es el precio en la misma semana que el clima. Pese a ello, la correlación más fuerte es 0.33, lo cual es una correlación muy débil, además, la mayoría de variables climáticas están altamente correlacionadas entre ellas, por lo que incluir muchas variables al modelo puede no ser interesante y evidencia la necesidad de utilizar técnicas de reducción de dimensionalidad.

3.4 Reducción de dimensionalidad

Tras haber estudiado todas las variables climáticas disponibles (aproximadamente 240), se puede observar como las correlaciones con el precio son bajas, además las correlaciones entre las variables climáticas son generalmente altas, y dado al alto número de parámetros disponibles, utilizarlos todos en un modelo conllevaría problemas de *overfitting* y un aprendizaje muy lento.

Con el fin de solventar este problema, se estudia el uso de distintas técnicas de reducción de dimensionalidad, para intentar obtener menos características que sean más significativas.

Las técnicas de reducción de dimensionalidad se dividen en dos clases:

- **Selección de características**: Se filtran las características existentes para obtener un subconjunto más reducido. Las características no se modifican.
- **Proyección de características**: Se generan nuevas características transformando y combinando las ya existentes.

3.4.1. Métodos de selección de características

Los métodos de selección de características son, generalmente, muy simples; y suelen ser específicos para cada serie de datos, aunque hay algunos genéricos:

- **Selección de las k características más correlacionadas con la variable objetivo**.
- **Selección iterativa**: Se implementa un modelo; en cada iteración, se escoge la variable que mejor resultado aporta en combinación con todas las anteriores. Es complejo en series temporales ya que los modelos usualmente utilizados (bosques aleatorios...) no funcionan bien con estas.
- **Eliminación iterativa**: Contrario al anterior, se empieza con todas las características y se van eliminando las que menos empeoran el modelo.

En este caso, la selección de las k características más correlacionadas escogería variables con correlaciones muy altas entre ellas, con lo que se perdería mucha información. Las otras dos técnicas son muy complejas de aplicar dado a la complejidad de los modelos de series temporales.

En este caso, se plantea una técnica personalizada, derivada de la selección iterativa vista anteriormente, pero que funciona sin modelo, en la que se escogen las características con una correlación con el precio mayor a 0.25 que no estén correlacionadas más de 0.4 con alguna de las ya escogidas:

```
resultado ← []
caracteristicas ← caracteristicas_ordenadas_por_correlacion
for all caracteristica ∈ caracteristicas do
  if corr(caracteristica, precio) > 0,25 then
    independiente ← true
    for all caracteristica_resultado ∈ resultado do
      if corr(caracteristica, caracteristica_resultado) > 0,4 then
        independiente ← false
      end if
    end for
    if independiente then
      resultado ← caracteristica
    end if
  end if
end for=0
```

Este algoritmo escoge 4 variables climáticas 3.11:

- Temperatura del aire media en Francia

- Temperatura del cielo máxima en Portugal
- Precipitación máxima en Almería
- Temperatura media del cielo en Holanda

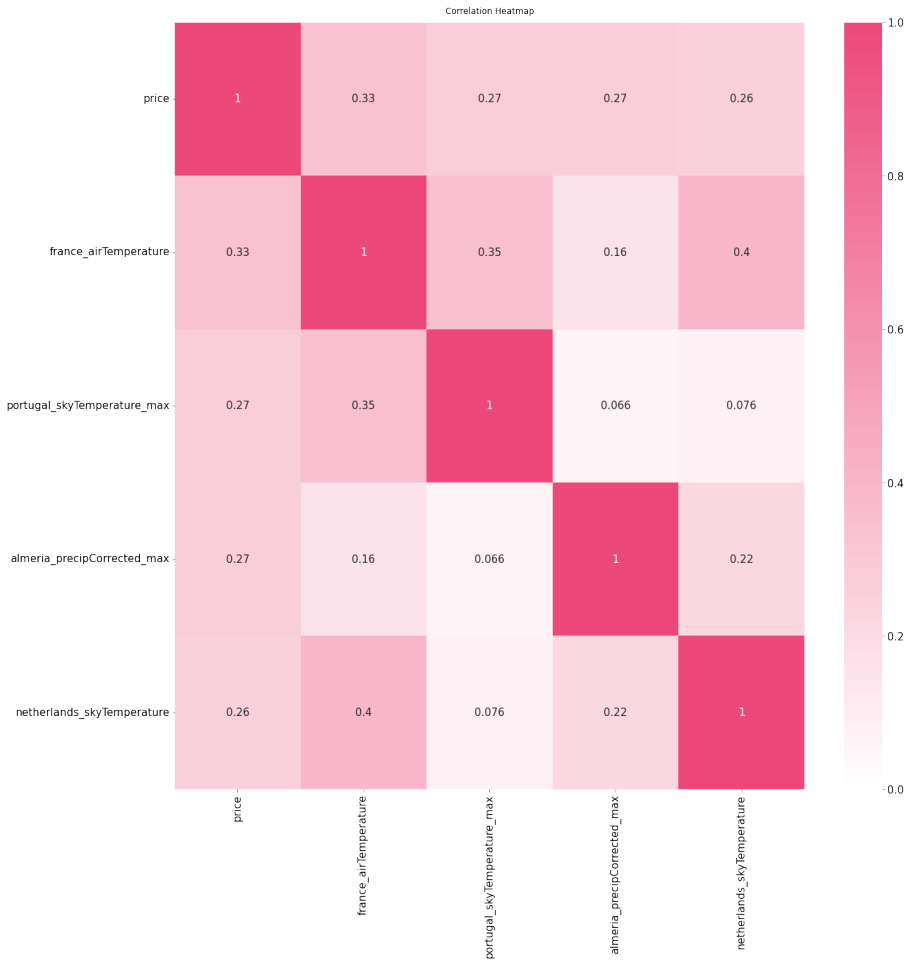

Figura 3.11: Correlaciones de las variables climáticas seleccionadas por el algoritmo personalizado

3.4.2. Métodos de proyección de características

Entre los métodos mas populares de proyección de características se encuentran:

- **PCA**: Intenta maximizar la varianza en el subconjunto de características.
- **LDA**: Intenta maximizar la separabilidad entre clases, principalmente utilizada en problemas de clasificación.
- **Autoencoders**: Encuentran un conjunto de características reducido que pueda ser reconstruido al conjunto original.

En este caso, dado que es un problema de regresión, no se estudia el uso de LDA.

PCA

El **Análisis de Componentes Principales**, o por sus siglas en inglés, PCA, es una técnica estadística cuyo objetivo es representar un conjunto de características en un conjunto de menor tamaño construido como combinación lineal de las características iniciales. Este conjunto esta formado por la proyección de los datos sobre nuevos ejes llamados **Componentes Principales**, que son independientes y **perpendiculares** entre sí y que (idealmente) explican la mayor cantidad posible de varianza de los datos iniciales.

Este algoritmo se compone de los siguientes pasos:

1. Cálculo de la matriz de covarianzas La covarianza es la medida de la relación entre dos variables, una covarianza positiva indica que ambas variables se mueven en la misma dirección (si una aumenta, la otra también), mientras que una covarianza negativa indica que lo hacen de forma inversa. La fórmula de la covarianza entre dos variables se define como sigue:

$cov_{x,y} = \frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{N-1}$

La matriz de covarianza es una matriz simétrica en la que el valor en $C_{i,j}$ es igual a la covarianza entre la variable i y la j 3.12.

$$\begin{bmatrix} V_a & C_{a,b} & C_{a,c} & C_{a,d} & C_{a,e} \\ C_{a,b} & V_b & C_{b,c} & C_{b,d} & C_{b,e} \\ C_{a,c} & C_{b,c} & V_c & C_{c,d} & C_{c,e} \\ C_{a,d} & C_{b,d} & C_{c,d} & V_d & C_{d,e} \\ C_{a,e} & C_{b,e} & C_{c,e} & C_{d,e} & V_e \end{bmatrix}$$

Figura 3.12: Matriz de covarianza entre 5 variables

2. Cálculo de los eigenvectores y los eigenvalores Las matrices diagonalizables (como la de covarianza, ya que es simétrica y estas siempre son diagonalizables) pueden ser descompuestas en el producto de los llamados **eigenvectores** y una matriz diagonal formada por sus respectivos **eigenvalores**. En el contexto de la matriz de covarianza, los eigenvectores se pueden interpretar como las direcciones de propagación de los datos, y los eigenvalores la magnitud de esa propagación (varianza). Por tanto se pueden obtener las direcciones que mayor varianza explican del conjunto inicial.

Para encontrar los diferentes eigenvalores (λ), se encuentran las posibles soluciones para la siguiente ecuación:

$|C - \lambda * I| = 0$

Cada eigenvalor tiene un eigenvector v asociado, que cumple la siguiente propiedad:

$(C - \lambda_n) * v_n = 0$

La cantidad de varianza (%) explicada por cada eigenvector se define de la siguiente forma:

$V(v_i) = \frac{\lambda_i}{\sum(\lambda)}$

Se pueden ordenar los eigenvectores por importancia según el valor descendente de sus eigenvalores.

3. Obtención del vector de características El vector de características es la combinación de eigenvectores como columnas que forman la matriz F que proyectará los datos. Esta matriz esta formada por tantos eigenvectores como dimensiones a las que se quiere proyectar, aunque hay estrategias para elegir el número de estas de forma algorítmica.

Por último, para obtener los datos proyectados, se multiplican los datos por el vector de proyección:

$X_{PCA} = X * F$

4. Obtención de proyecciones Tras aplicar PCA a el conjunto de datos climáticos, reduciendo a 5 caracerísticas, se obtienen 5 direcciones, explicando la primera de ellas el 47.5 % de la varianza y las demás menos del 10 % cada una 3.13. La proyección sobre esta primera componente principal tiene una correlacion de 0.37 con el precio, siendo la mayor 3.14.

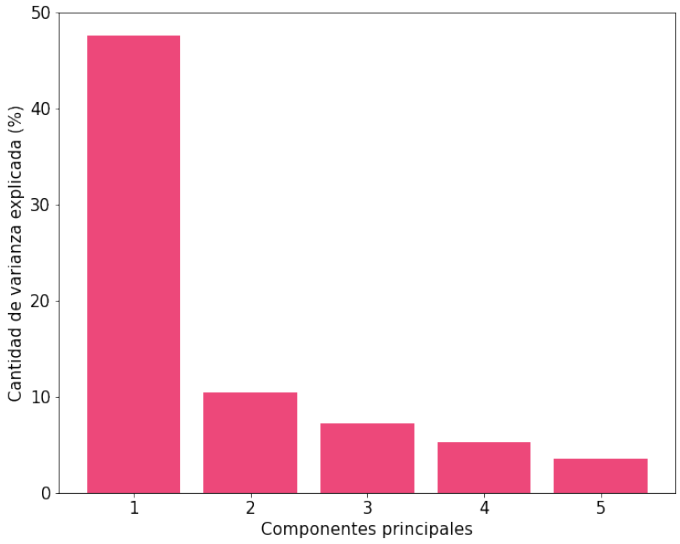

Figura 3.13: Varianza explicada por cada uno de los componentes principales

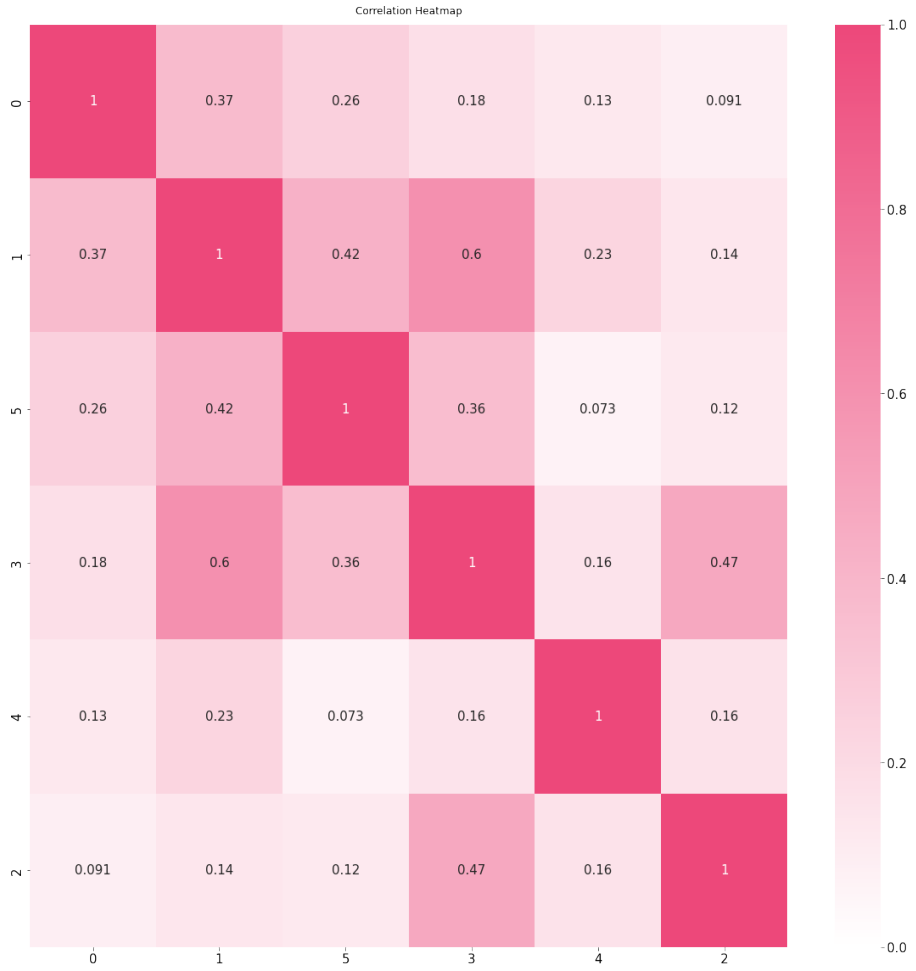

Figura 3.14: Correlaciones de las nuevas características proyectadas por las componentes principales

LSTM Autoencoder

Las redes LSTM, son un tipo de red neuronal que utiliza unas neuronas especiales, diseñadas para ser capaces de trabajar con estructuras temporales complejas; estas serán analizadas y estudiadas más adelante, pero también se pueden utilizar para autocodificar datos temporales y comprimirlos.

Los **autoencoders** o auto-codificadores son un tipo de algoritmos que aprenden a codificar o comprimir unos datos, para luego intentar reconstruirlos con el menor error posible a partir de la compresión; son algoritmos de **aprendizaje propiamente supervisado**, pues aunque requieren de una variable objetivo, esta es la misma que la variable de entrada.

En este caso, se dispone de 240 dimensiones de entrada, y se planea reducirlas a 5. La estructura del autoencoder escogida es (entre paréntesis las dimensiones tras cada capa):

- Entrada (240)
- LSTM (128)
- LSTM (64)
- LSTM (32)

- LSTM (16)
- LSTM (5) Hasta aquí el codificador.
- LSTM (16)
- LSTM (32)
- LSTM (64)
- LSTM (128)
- LSTM y Salida (240) Hasta aquí el decodificador y el autocodificador.

Para que la red sea capaz de tener suficientes muestras, se dividen los datos de entrada en secuencias de 52, es decir, de un año; cada secuencia es un desplazamiento a la derecha de la anterior, con lo que los valores están repetidos en múltiples secuencias. Tras entrenar con estos datos la red anterior, con 35000 épocas, una tasa de aprendizaje de 0.001, y usando el **MAE** como función de error del autoencoder, el resultado son cinco características, la primera de ellas con una correlación de 0.24 3.15.

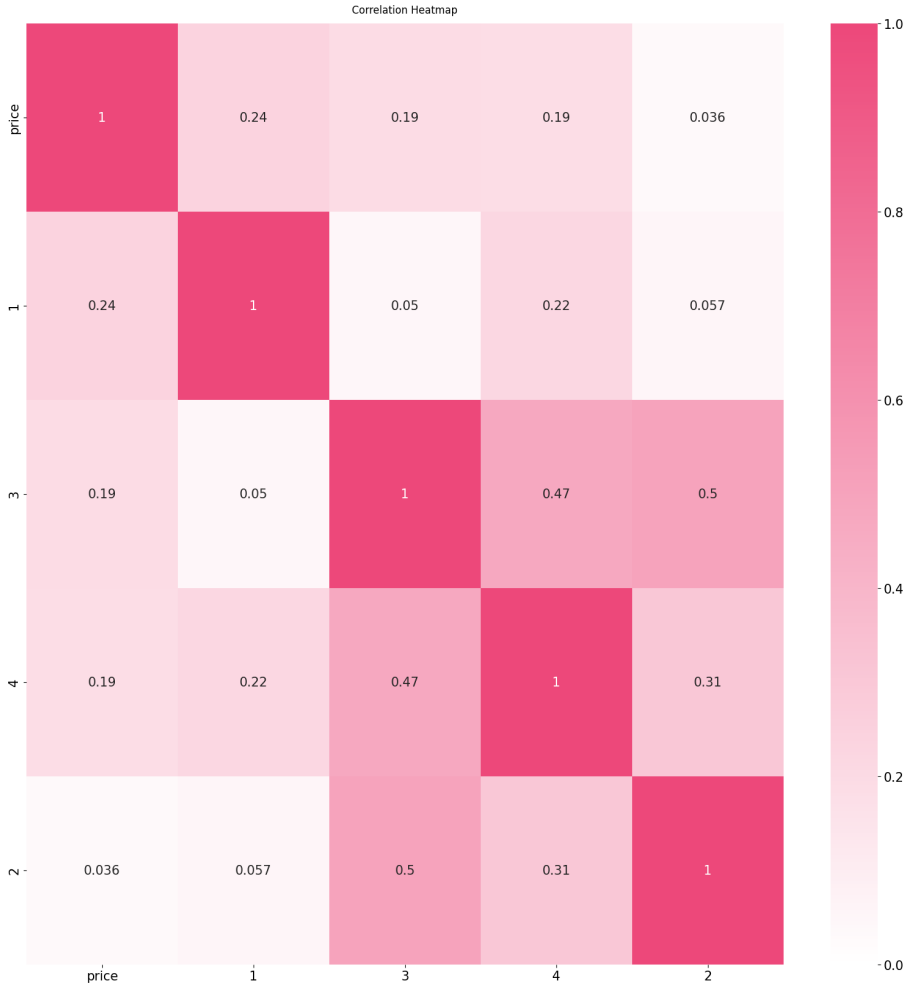

Figura 3.15: Correlaciones de las nuevas características obtenidas por el autoencoder

3.4.3. Método escogido

Se comparan las correlaciones de las características obtenidas tras aplicar el método de selección de características 3.1, el método PCA ?? y el autoencoder ??. Se puede

apreciar que las correlaciones siguen siendo no muy elevadas, con lo que para reducir la complejidad de los modelos, es preferible utilizar pocas características que tengan la máxima correlación; en este caso podemos observar que PCA obtiene la característica con más correlación (C1), y tiene otra característica (C5) con una correlación relativamente alta, con lo que se elige este método por delante de los demás, ya que el autoencoder no ha logrado buenas correlaciones, y el de selección tiene todas las características con algo de correlación, pero ninguna llega a la primera del PCA.

	Precio	C1	C2	C3	C4
Precio	1	0.33	0.27	0.27	0.26
C1	0.33	1	0.35	0.16	0.4
C2	0.27	0.35	1	0.066	0.076
C3	0.27	0.16	0.066	1	0.22
C4	0.26	0.4	0.076	0.22	1

Tabla 3.1: Correlaciones tras el método de selección personalizado

	Precio	C1	C5	C3	C4	C2
Precio	1	0.37	0.26	0.18	0.13	0.091
C1	0.37	1	0.42	0.6	0.23	0.14
C5	0.26	0.42	1	0.36	0.073	0.12
C3	0.18	0.6	0.36	1	0.16	0.47
C4	0.13	0.23	0.073	0.16	1	0.16
C2	0.091	0.14	0.12	0.47	0.16	1

Tabla 3.2: Correlaciones tras aplicar PCA

	Precio	C1	C3	C4	C2
Precio	1	0.24	0.036	0.19	0.19
C1	0.24	1	0.06	0.05	0.22
C3	0.19	0.05	0.50	1	0.47
C4	0.19	0.22	0.31	0.47	1
C2	0.04	0.06	1	0.50	0.31

Tabla 3.3: Correlaciones tras emplear el autoencoder

CAPÍTULO 4
Modelos

Existen multitud de modelos destinados a la predicción de series temporales, estos van desde los modelos estadísticos utilizados durante mucho tiempo como los derivados de los ARIMA hasta modelos basados en *Deep Learning*, como las redes neuronales recurrentes y las convolucionales temporales.

4.1 Métricas utilizadas y validación cruzada

Como métrica general de error se va a usar el **Error Absoluto Medio**, o *MAE* por sus siglas en inglés. El *MAE*, penaliza de forma homogénea los errores y es muy representativo de la realidad, ya que se puede interpretar como la diferencia en céntimos del precio real y predicho.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

Para evaluar los modelos, se dividen los datos disponibles en entrenamiento y test, y con el fin de garantizar unos resultados adecuados y que estos sean independientes de la partición en datos de entrenamiento y test realizada, se usa la técnica de **validación cruzada**, o *cross-validation*. El método más común es el *k-fold cross validation*, en el que se divide el conjunto de datos en **k** particiones, y se entrena el modelo **k** veces, usando en cada una de ellas una de las particiones como test y las demás como entrenamiento 4.1.

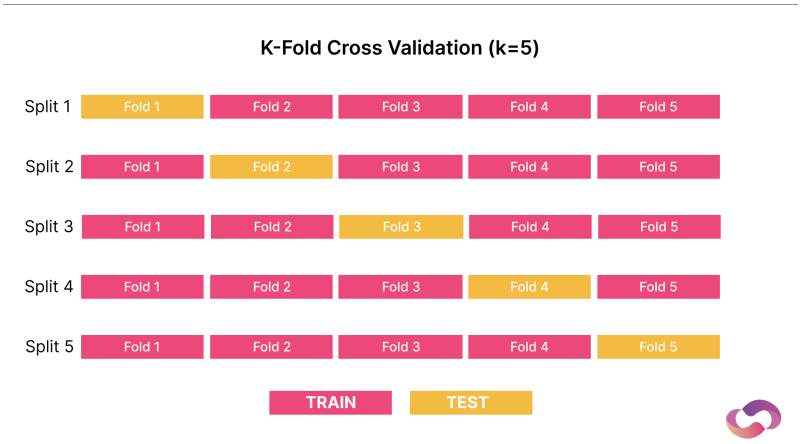

Figura 4.1: Validación cruzada con k=5 particiones

Este método no es aplicable a series temporales, ya que en algunas particiones se utilizarían datos del futuro para entrenar el modelo, usando, por ejemplo, el año 2015 como test pero entrenando con los años 2008 . . . 2014 y 2016 . . . 2020.

Una solución a este problema es eliminar las particiones detrás de los datos de test, nunca usando para entrenar muestras posteriores a las que se usan para testeo 4.2.

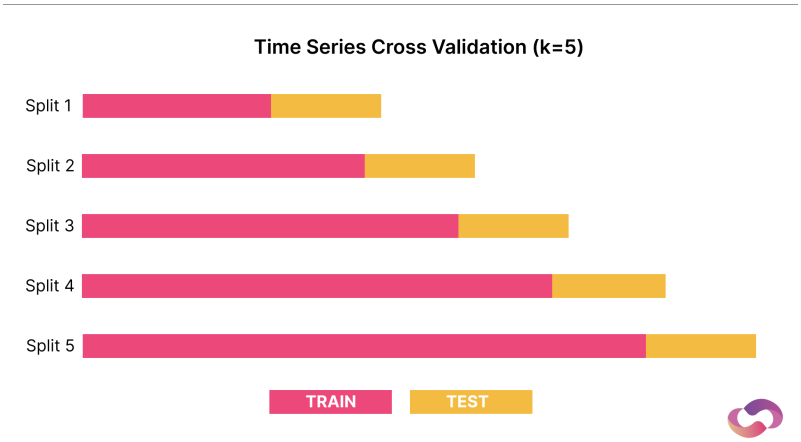

Figura 4.2: Validación cruzada para series temporales con k=5

4.2 Modelo de referencia

El modelo de referencia es de gran ayuda para comparar los modelos generados con este y ver cuál es la mejora. Este modelo debe ser lo más sencillo posible para que su implementación no tome demasiado tiempo pero suficientemente potente como para poder llegar a ser usado como solución al problema.

Modelo

En este caso, el objetivo de este estudio es ser capaces de predecir el precio con más precisión que los métodos actuales usados hoy en día; es interesante, por tanto, utilizar estos métodos como modelos de referencia. Para estimar el precio en una semana s (p_s), la forma más extendida es la de utilizar directamente el precio de la semana anterior ($p_s = p_{s-1}$). De igual forma, para estimar el precio del año próximo, se utiliza el precio del año anterior ($p_{a,s} = p_{a-1,s}$).

Resultados

Para la predicción a corto plazo, el error obtenido es un MAE de 0.168 4.3 y para la predicción a largo plazo, un MAE de 0.439 4.4.

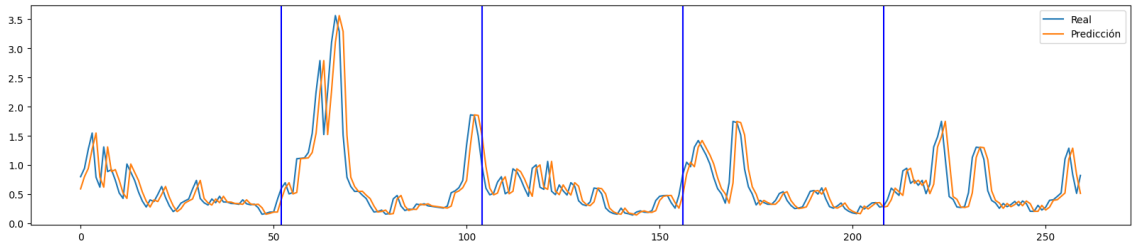

Figura 4.3: Resultados del modelo de referencia prediciendo a corto plazo para las 5 particiones

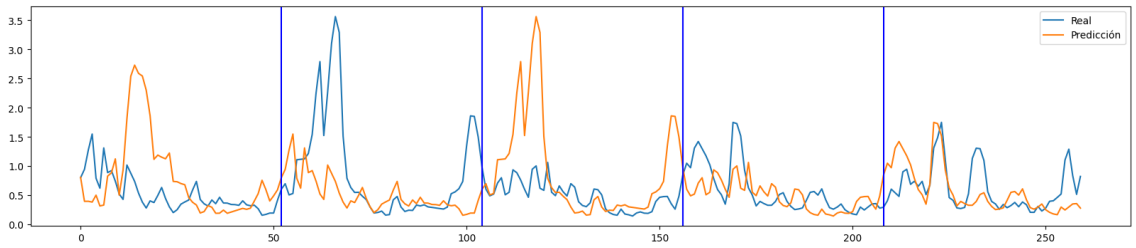

Figura 4.4: Resultados del modelo de referencia prediciendo a largo plazo para las 5 particiones

4.3 Modelos estadísticos básicos

4.3.1. SARIMA

Los modelos SARIMA, son procesos estadísticos que utilizan series temporales para realizar predicciones a futuro, SARIMA es un acrónimo para ***S**easonal **A**uto**R**egressive **I**ntegrated **M**oving **A**verage* (Modelo Autorregresivo Integrado de Promedio Móvil con Estacionalidad). El funcionamiento de un modelo SARIMA es muy simple, se entrena con el histórico de datos y es capaz de predecir tantos valores futuros como se desee.

Estos modelos son el resultado de incluir un componente estacional a los modelos ARIMA, que a su vez están constituidos por los siguientes componentes:

- **AR: Autorregresión.** Modelo en el que el valor de una variable depende de los valores previos de la misma. Por ejemplo, para predecir el precio de mañana, se utilizan los precios de los **p** días previos.
- **I: Integración.** Representa la diferenciación de los datos para lograr una serie de datos estacionaria. Esto implica que, en vez de predecir el precio de mañana, se predice la diferencia entre el precio de hoy y el de mañana. Esta diferenciación se realiza **d** veces.
- **MA: Promedio Móvil.** Modelo en el que el valor de una variable depende de los errores residuales de los valores previos. Por ejemplo, para predecir el precio de mañana, se utilizan los errores residuales de la predicción realizada por el propio modelo en los **q** días previos.

Los 3 parámetros introducidos corresponden a:

- **p**: Número de valores previos utilizados en el modelo de autorregresión.
- **d**: Orden de diferenciación (número de veces que los datos son diferenciados)
- **q**: Tamaño de la ventana del modelo de promedio móvil.

Un modelo ARIMA, se representa con la notación $ARIMA(p, d, q)$, mientras que un modelo SARIMA, se representa con la notación $SARIMA(p, d, q)x(P, D, Q, s)$. Donde s es el periodo de estacionalidad y P, D, Q son los mismos parámetros definidos anteriormente pero aplicados al componente estacional.

Modelo

Para elegir los parámetros adecuados del modelo SARIMA, se utiliza un método que prueba todas las combinaciones de parámetros del modelo para elegir la mejor de ellas. Este método recibe los rangos para cada variable de las mencionadas anteriormente, y crea un modelo para cada combinación, obteniendo para cada unos de ellos el **AIC**, eligiendo el menor valor de ellos.

Los rangos elegidos son entre 0 y 4 para todos los parámetros excepto para la estacionalidad, que está fijo a 52; y para la integración, que está fijo a 0 ya que la serie no tiene tendencia. Esto se extrae de las funciones de autocorrelación, donde la PACF tiene un máximo de desplazamientos relevantes de 3 (se añade uno de margen) y la ACF es geométrica, lo que indica que probablemente el parámetro q sea 0, es decir, no haya modelo de media móvil, pero debido al poco coste de entrenar los modelos ARIMA, se decide probar también de 0 a 4.

El **AIC**, o **Criterio de Información de Akaike**, es una medida de la calidad relativa del ajuste para modelos estadísticos, esto nos permite comparar como distintos modelos se adecuan a una misma distribución de datos. Esta medida encuentra un equilibrio entre la complejidad del modelo (overfitting) y la precisión (underfitting). La fórmula del **AIC** es:

$$AIC = 2k - 2ln(\hat{L})$$

Donde k es el número de parámetros del modelo y L es la log-similitud (*Log Likelihood*). Esta medida representa la cantidad de información de los datos originales perdida al usar el modelo candidato, debido a esto, cuanto menor sea el **AIC**, más preciso es el modelo. Según Stone (1977) [12], bajo ciertas condiciones, AIC es asintóticamente equivalente a la **validación cruzada dejando uno fuera (LOOCV)**, que a su vez es equivalente en el caso de series temporales a la validación cruzada explicada anteriormente con el mismo número de particiones que de muestras.

En este caso, se prefiere el uso del AIC para escoger el modelo por delante de la validación cruzada con el MAE debido a que este último es sustancialmente más lento; además, el modelo ARIMA se debe actualizar cada vez que se realiza una predicción para realizar la siguiente, con lo que el uso del método de validación cruzada con MAE resulta muy lento.

Con todo esto, el modelo SARIMA que mejores resultados aporta es el $SARIMA(3,0,0)x(2,0,1,52)$, obteniendo un AIC de -179.323. El parametro q ha sido 0, con lo que se confirman las conclusiones obtenidas de la ACF.

Resultados

El AIC es una medida relativa que nos permite comparar distintos modelos SARIMA para elegir el mejor, pero para comparar este modelo con los demás que se proponen a continuación se necesita utilizar la métrica común, el MAE. Para ello, usando validación cruzada, para cada iteración de la misma se entrena el modelo sobre los datos de entrenamiento, y posteriormente, se predice uno a uno cada valor de test, actualizando el modelo tras cada predicción.

El MAE obtenido es de 0.147 para la predicción a corto plazo, mejorando al del modelo de referencia en un 12.5 %. Se puede interpretar como que para cada valor predicho, el modelo se equivoca 14.7 céntimos de media; 2.1 menos que el modelo de referencia.

El modelo escogido, para predicción a largo plzo logró un MAE sobre la validación cruzada de 0.282, superando al modelo de referencia en un 35.8 %.

4.3.2. SARIMAX

Los modelos SARIMAX son una extensión de los previamente introducidos modelos SARIMA, que incluyen el uso de **variables exógenas**, es decir, variables externas a la que se desea predecir, pero que influyen en esta. Debido al uso de estas variables exógenas, estos modelos no pueden funcionar correctamente prediciendo a largo plazo pues se necesitarían los valores de las variables exógenas para el futuro, teniendo que predecirlos y acumulando un error mayor al del modelo SARIMA normal.

Modelo

De forma similar a como ocurría en el modelo SARIMA, se emplea el uso de AIC junto con un método que comprueba que parámetros son los más óptimos con los mismos rangos para escoger el mejor modelo SARIMAX, siendo el escogido el $SARIMAX(1,0,1)x(3,0,0,52)$, con un AIC de -216.492.

Los modelos SARIMA, una vez entrenados, ofrecen una serie de estadísticas que permiten determinar, entre otros, la significancia de los términos que participan en el modelo. Entre estos términos, se encuentran las dos características exógenas provenientes de PCA (C1 y C5). Para ello, SARIMA calcula los valores t de la prueba de significancia; esta prueba es una **prueba de hipótesis**, igual que la prueba Q de Dixon o que otras pruebas de raíz unitaria vistas con anterioridad; en este caso, la hipótesis nula es que los términos no son significantes y el valor t es la log-verosimilitud de los valores en caso de que la hipótesis nula se cumpla, con lo que si este valor es menor a 0.05 (intervalo de confianza), se puede rechazar la hipótesis nula y asumir que el término es significativo.

En este caso, el valor t para C1 es de 0.002, y para C5 de 0.003; con lo que ambas características son significativas para el modelo, es decir, aportan información.

Resultados

El MAE obtenido para predicción a corto plazo es de 0.145, mejorando al del modelo de referencia en un 13.7 %. La mejora respecto al modelo SARIMA sin variables exógenas es de tan solo un 1.4 %, lo que es debido a las correlaciones tan bajas.

4.4 Prophet

Prophet es una librería desarrollada por Facebook para los lenguajes de programación R y Python; cuya función es predecir series de datos temporales utilizando un modelo aditivo (lineal) con tendencia, estacionalidades y días festivos:

$$y_t = tendencia_t + estacionalidades_t + festivos_t + error_t$$

Prophet funciona especialmente bien en series de datos con múltiples estacionalidades y con tendencias dinámicas. Prophet también puede manejar variables exógenas en forma de **regresores adicionales**. Además, está centrado en predicción de ventas en tiendas, con lo que incluye el concepto de festivos que en este caso no se puede utilizar.

Modelo

Prophet dispone de muy pocos parámetros, ya que es un modelo más simple que los ARIMA y sus derivados. Los más importantes son:

- **growth**: La forma que sigue la tendencia, puede ser *None*, *Linear* o *Logistic*. En este caso, el precio no tiene tendencia y se usa None.
- **seasonality_yearly, seasonality_weekly y seasonality_daily**: Valores booleanos indicando la presencia de estas estacionalidades. En este caso, solo **seasonality_yearly** es True.

Por tanto, el modelo escogido es un modelo Prophet sin tendencia y con estacionalidad anual.

Al ser un modelo aditivo y lineal, y funcionar modelando la tendencia y la estacionalidad, Prophet está especialmente pensado para predicciones a largo plazo y no tanto para corto plazo, pese a que también puede ser utilizado.

Resultados

Para predicción a corto plazo, se ha obtenido un MAE de 0.306. Es el error más elevado de todos los modelos, superando incluso al de referencia, pero Prophet no está pensado para predicción a corto plazo, y como se puede observar en los resultados 4.5, la predicción es similar a la de largo plazo 4.6; ya que Prophet realiza las predicciones modelando la tendencia y la estacionalidad, con lo que la salida son patrones estacionales.

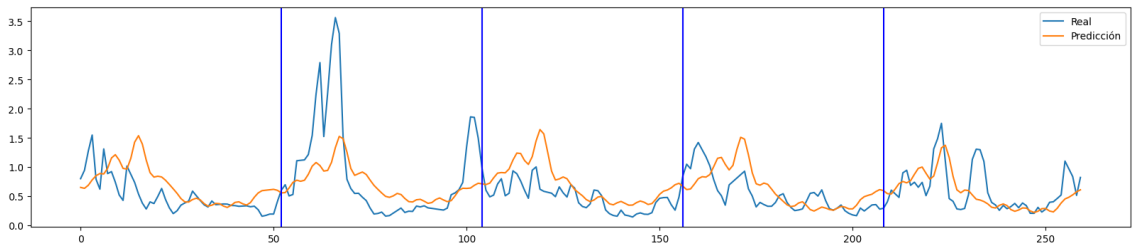

Figura 4.5: Predicción a corto plazo obtenida con Prophet

La predicción a largo plazo, sin embargo, ha obtenido un MAE de 0.310; mejorando al modelo de referencia en un 29.4 % pero empeorando respecto al modelo SARIMA un 9 %. Pese a que quizás la serie no es la más adecuada para el uso de Prophet, ya que no presenta varias estacionalidades ni tendencia, Prophet se ha acercado al modelo SARIMA, siendo un modelo mucho más sencillo de entender, implementar y entrenar.

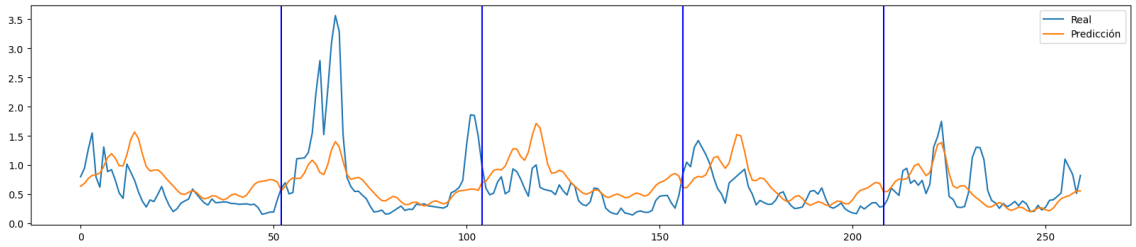

Figura 4.6: Predicción a largo plazo obtenida con Prophet

4.5 Redes neuronales (*Deep Learning*)

Las redes neuronales son algoritmos computacionales que recrean el funcionamiento del cerebro humano y que permiten encontrar relaciones y patrones en los datos para realizar toda clase de funciones. La primera red neuronal fue implementada en 1943 por el neurofisólogo McCulloch y el matemático Pitts [13] utilizando circuitos eléctricos simples. En 1958, Rosenblatt [14] creó el perceptron, que no es más que una red neuronal con una sola neurona 4.7. Un año más tarde, en 1959, Widrow y Hoff crearon *ADALINE* y *MADALINE*, que fueron las primeras redes neuronales comerciales, esta última fue utilizada para eliminar el eco en señales telefónicas. Pero desde los 60 hasta los 80, el avance en el hardware no fue capaz de seguir al avance en el campo de la inteligencia artificial, a esto se sumaron dilemas filosóficos que apelaban al peligro de que estos algoritmos cobrasen consciencia, y también algunos artículos científicos que determinaron que no era posible implementar redes neuronales multicapa; todo esto llevó a una época conocida como el **invierno de la inteligencia artificial**, donde los fondos destinados a su investigación y el interés en la misma disminuyeron drásticamente. En 1982, Hopfield introdujo la idea de crear redes con conexiones bidireccionales entre neuronas, y en 1986 Rumelhart et al [15] introdujeron la idea de **propagación hacia atrás** o **backpropagation**, que es un método para ajustar los pesos de las neuronas basados en el error obtenido, donde dicho error se propaga en sentido inverso al funcionamiento normal de la red; esto permitió entrenar a las redes de forma mucho más rápida y obtener resultados más precisos; a raíz de estos estudios, el interés sobre la inteligencia artificial y el *Deep Learning* se reactivó, lo que trajo el desarrollo de redes multicapa más avanzadas.

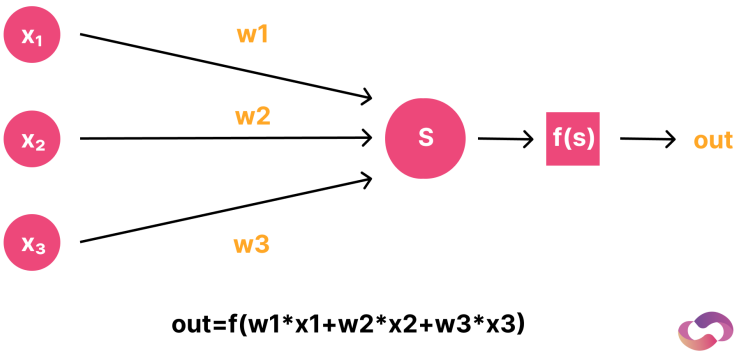

Figura 4.7: Perceptrón con 3 entradas y una salida, donde f es la función de activación

Actualmente, las redes neuronales se utilizan para distintos propósitos y existen muchas variaciones del modelo de neurona original.

4.5.1. Funcionamiento de una red neuronal

Una red neuronal, generalmente está formada por su entrada, algunas capas intermedias y una capa de salida. Estas capas están formadas por **neuronas**, que como se ha podido ver antes en el perceptron, son unidades que contienen un peso w para cada una de las entradas, y su salida es la suma de los productos de los pesos por las entradas más un *bias* y la aplicación de una función de activación f.

$$s = f(\sum(w_i * x_i + b_i)$$

Funciones de activación Las funciones de activación son necesarias en las redes neuronales para aprovechar la potencia que aporta el uso de múltiples capas intermedias. En una red neuronal sin funciones de activación, no importa cuantas capas intermedias se disponga, siempre equivale a una red neuronal de una sola capa (solo la de salida). Por ejemplo, una red neuronal de una capa intermedia, lo que es una red de dos capas, ya que hay dos conjuntos de pesos (W_1 y W_2), tiene una salida dada por la siguiente ecuación:

$$y = W_2 * (W_1 * X)$$

Lo que aplicando propiedades de la multiplicación y las matrices, equivale a:

$$y = (W_2 * W_1) * X$$

Que a su vez:

$$y = (W) * X$$

Y, por tanto, equivale a tener solo una capa de pesos, es decir, una red de una sola capa y ninguna intermedia. Para solventar este problema, se debe añadir algún tipo de no-linealidad a la salida de cada capa, lo que se consigue empleando las funciones de activación.

Hay muchos tipos de funciones de activación, las más comunes son:

- Sigmoide 4.8
- TanH 4.9
- ReLU 4.10

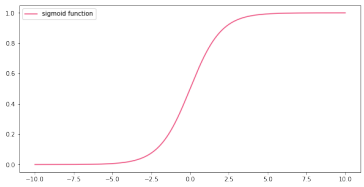

Figura 4.8: Sigmoide

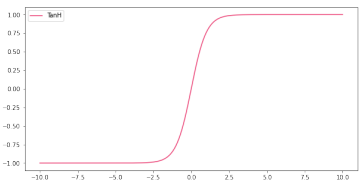

Figura 4.9: TanH

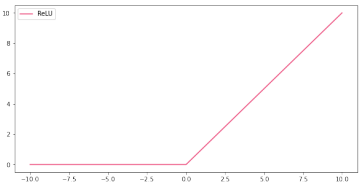

Figura 4.10: ReLU

Aprendizaje

La red neuronal es capaz de ajustar sus pesos de forma automática utilizando un algoritmo llamado **backpropagation**. Este algoritmo calcula cuanta culpa tiene cada peso de cada neurona del error obtenido. Este error viene dado por la llamada **función de perdida**. La actualización de los pesos de cada neurona se realiza siguiendo la estrategia marcada por el **algoritmo de optimización**. Este proceso se realiza de forma iterativa sobre los datos, repitiéndolo un número de veces determinado (**épocas**).

Funciones de perdida Las llamadas funciones de perdida sirven para evaluar los resultados. Estas dependen del tipo de tarea que se esta realizando (clasificación, regresión...) y del formato de la salida deseado. Entre las más comunes para regresión están:

- **Error cuadrático medio o MSE** $\sum_{i=1}^{D}(x_i - y_i)^2$
- **Error absoluto medio o MAE** $\sum_{i=1}^{D}|x_i - y_i|$

Descenso del Gradiente El objetivo de la red neuronal es minimizar la función de pérdidas para que esta logre su valor mínimo; para una función matemática simple, como por ejemplo, $f(x) = x^2 - 4$, podríamos igualar su derivada a 0 para obtener que el valor mínimo se obtiene cuando $x = 0$, pero cuando la función de error depende de todos los pesos, biases y entradas, la función se vuelve demasiado compleja para resolverla de esta forma. El descenso del gradiente se aprovecha del significado de la derivada (gradiente), que representa la dirección sobre la que una función aumenta; por tanto, la dirección negativa del gradiente permite desplazarse en la dirección en la que la función disminuye para lograr el mínimo local de dicha función 4.11. Es importante destacar el hecho de que este mecanismo solo permite encontrar **mínimos locales**, y que sí el mínimo global se encuentra alejado, muy probablemente no sea capaz de desplazarse hasta él.

$$x_{n+1} = x_n - \nabla f(x_n)$$

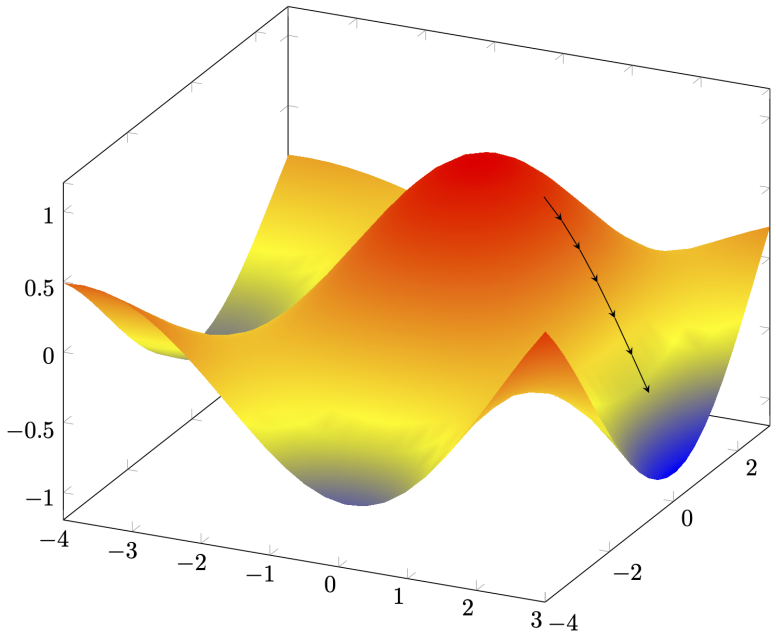

Figura 4.11: Movimiento en la dirección opuesta al gradiente para una función con dos entradas

Algoritmos de optimización El descenso del gradiente es uno de los algoritmos de optimización disponibles para entrenar una red neuronal, pero hay muchos más, y han evolucionado hasta el punto donde el uso del descenso del gradiente básico es marginal en redes neuronales. Uno de los problemas que el descenso del gradiente presenta es que se puede bloquear muy fácil en **mínimos locales**, encontrando una solución que posiblemente esté lejos de ser la mejor 4.12. Además, pese a ser un algoritmo simple y muy rápido computacionalmente, requiere de mucha memoria ya que cada iteración del mismo se realiza sobre todo el conjunto de los datos. Por todos estos motivos, aparecen variantes de este algoritmo, entre ellas:

- **Descenso del gradiente estocástico**: Similar al descenso del gradiente, pero actualiza los pesos cada pocas muestras en vez de necesitarlas todas cada iteración.
- **Momentum**: Aplica un momentum al algoritmo del descenso del gradiente, ponderando el error con los errores previos para evitar quedarse en mínimos locales.

- **Gradiente acelerado de Nesterov**: Evolución del algoritmo Momentum, al utilizar un momentum, se puede hacer un estimado de como se van a actualizar los parámetros antes de derivar (ya que tenemos las actualizaciones previas), con lo que se calcula el gradiente del error con respecto a los parámetros futuros estimados.

- **AdaGrad**: Ajusta la tasa de aprendizaje para cada parámetro según la distribución y frecuencia del mismo, disminuyéndola cada vez que se actualiza. Esto produce que la tasa de aprendizaje vaya reduciéndose hasta ser excesivamente pequeña.

- **AdaDelta**: Evolución del AdaGrad que disminuye la tasa de aprendizaje sólo con las actualizaciones que han ocurrido en una ventana w, en vez de todas. Reduce el problema del desvanecimiento de la tasa de aprendizaje de AdaGrad.

- **Adam**: Combinación de AdaDelta + Momentum. Es el algoritmo que generalmente da mejores resultados y el estándar.

En este estudio, se elije el uso del algoritmo Adam, por ser el que mayor rendimiento ofrece en la actualidad 4.13 y la opción *de facto* en la mayoría de redes.

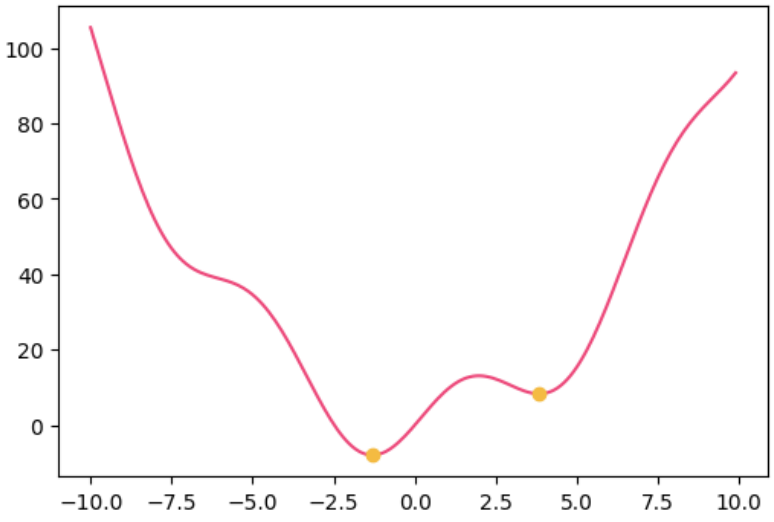

Figura 4.12: Función con un mínimo local a la derecha del mínimo global

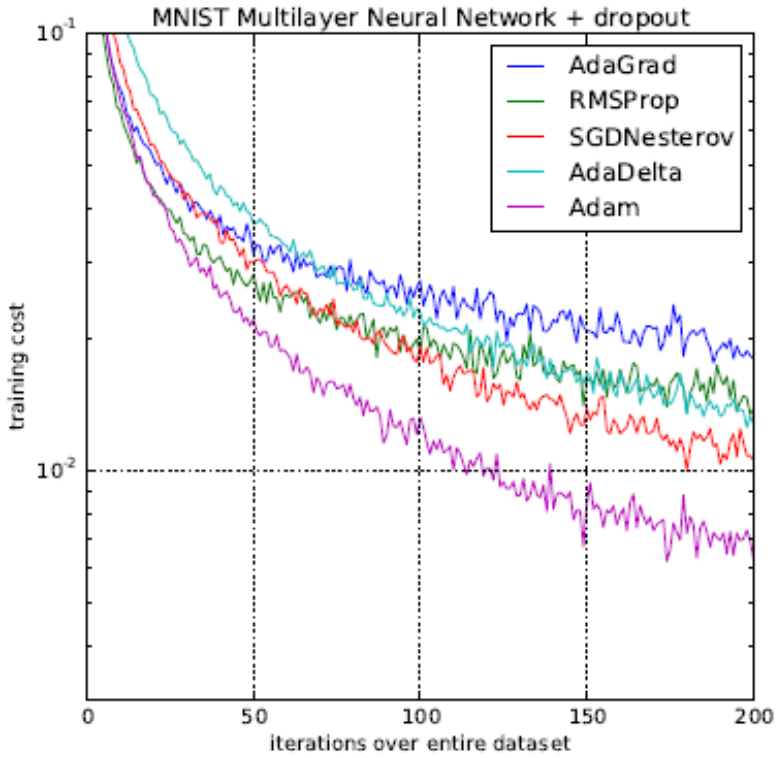

Figura 4.13: Error en cada iteración para distintos algoritmos de optimización, se aprecia el rendimiento de Adam

Backpropagation Tras cada predicción realizada por la red neuronal (paso adelante) se realiza un paso atrás ajustando los parámetros del modelo (pesos y biases). El descenso del gradiente permite descubrir en que dirección se debe de mover la solución para minimizar el error, pero, no todos los parámetros de la red neuronal tienen el mismo error, y no todos se deben mover en la misma dirección y proporción; cada neurona, o incluso grupos de neuronas, pueden realizar internamente funciones distintas, y cada una tiene su parte de error; para discriminar que parte del error total le corresponde a cada neurona, se emplean las **derivadas parciales**, por tanto, cada neurona actualiza sus pesos según el error parcial de los mismos:

$$W_{i,j}^{k} = W_{i,j}^{k-1} - \alpha * \frac{\partial L}{\partial W_{i,j}^{k-1}}$$

Donde α es la tasa de aprendizaje. Gracias a la **regla de la cadena**, se puede obtener la derivada parcial sobre el error de los parámetros de una neurona en una capa en función de las derivadas parciales de la capa siguiente, por lo que el cálculo iterativo de estos gradientes se debe hacer hacia atrás.

Overfitting en redes neuronales La red neuronal aprende sobre el conjunto de datos de entrenamiento y se evalúa sobre el conjunto de test; uno de los problemas más comunes en estas es cuando el error sobre el conjunto de entrenamiento se reduce y llega a un punto muy bajo pero sin embargo el error sobre el conjunto de test no disminuye o incluso, aumenta 4.14. Esto es debido a que el modelo es excesivamente complejo para los datos sobre los que se quiere aprender 4.15.

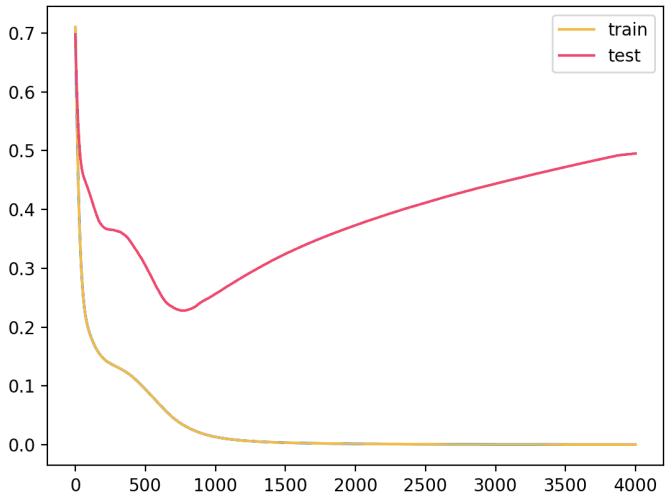

Figura 4.14: Evolución de los errores sobre el conjunto de test y de entrenamiento con overfitting

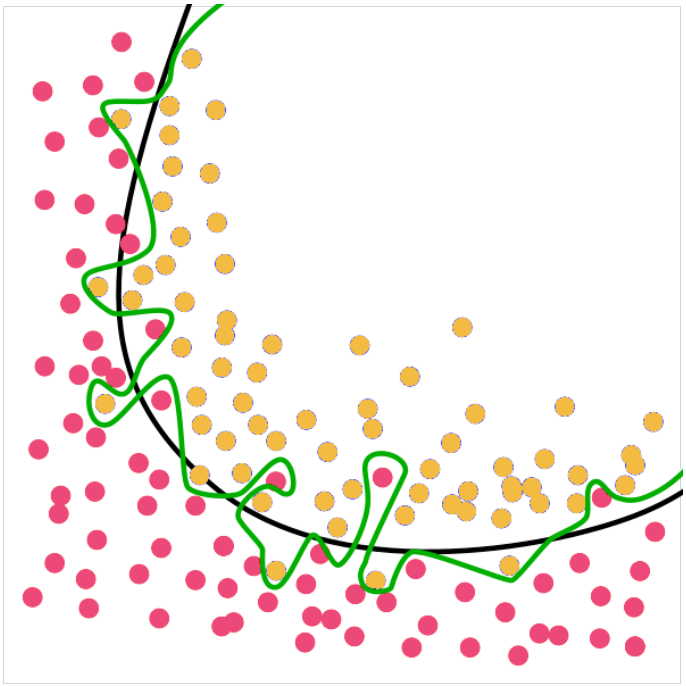

Figura 4.15: Separación de dos clases. En negro, la frontera deseada y en verde, la que presenta overfitting

Para tratar de solventar este problema, que es más frecuente en redes neuronales que en otros algoritmos más simples, existen multitud de estrategias. Dos de las más utilizadas y las que se van a emplear en este estudio son la **parada temprana** (early stopping) y el **dropout**.

Parada temprana La red neuronal entrena durante un número de épocas determinado como un hiperparámetro. El overfitting puede presentarse a partir de una época determinada, y llegar a una solución buena para luego empeorar. Si ocurre esto, una posible

estrategia es calcular el error sobre el conjunto de test en cada época y detener el entrenamiento tras n épocas de aumento de dicho error.

Los dos parámetros de esta técnica son:

- **Paciencia**: Número de épocas que tiene que no mejorar el error de validación para detener el entrenamiento.
- **Delta mínima**: Diferencia positiva mínima entre el error en t y el error en $t-1$ para que sea considerado una mejora.

La paciencia debe ser suficiente como para que la red pueda tolerar empeoramientos ocasionales sin detener por completo el entrenamiento. Si el empeoramiento durante estas épocas es significativo y el modelo no vuelve a mejorar, la solución final puede ser mucho peor a la óptima obtenida épocas atrás; para solventar esto, se pueden utilizar **puntos de control**, guardando siempre el mejor modelo hasta la época para restaurarlo al final del entrenamiento.

Dropout El Dropout fue introducido en 2014 por Srivastava et al [16] como un método para prevenir el overfitting de las redes neuronales, esta técnica consiste en aleatoriamente eliminar las salidas de ciertas neuronas de la red (ponerlas a 0), con el objetivo de añadir una especie de ruido que permita a la red neuronal generalizar mejor. La probabilidad de que una neurona se descarte viene dada por la **tasa de dropout**, que se introduce como un hiperparámetro para cada capa de la red 4.16.

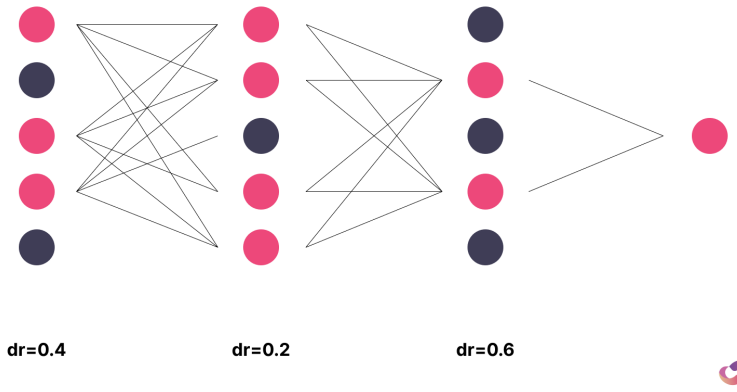

Figura 4.16: Ejemplo de distintas tasas de dropout (dr) en capas de una red neuronal

4.5.2. Ajuste de hiperparámetros

Un hiperparámetro es un valor que se usa en los modelos para controlar el aprendizaje y el funcionamiento de los mismos. Estos parámetros no se aprenden y se establecen al definir el modelo de forma fija. En los modelos previos, la selección de hiperparámetros era sencilla o derivada del análisis exploratorio de los datos, incluso en los modelos derivados del ARIMA, se emplaba el uso de métricas y algoritmos que permiten escoger los más óptimos. En el caso de las redes neuronales, los hiperparámetros son muy importantes y sus valores definen el resultado y el tiempo que un modelo toma para entrenar. Algunos de los hiperparámetros más importantes comunes a todas las redes neuronales son:

- **Tamaño del lote (batch size)**: Cada iteración del algoritmo de optimización se realiza sobre un subconjunto de los datos, el tamaño de este está definido como un hiperparámetro. Según el algoritmo de optimización, el modelo y los datos, los rangos de este pueden variar.

- **Tase de aprendizaje (learning rate)**: Este valor controla la cantidad de cambio en las neuronas como respuesta al error producido.

- **Número de épocas**

No existe, en general, una ciencia o metodología estándar para realizar esta optimización de hiperparámetros, y existen distintos algoritmos que intentan solventar este problema, entre ellos:

- **Búsqueda en rejilla**: Se establece una cuadrícula con una serie de valores predefinidos para cada hiperparámetro y se evalua toda combinación posible a fuerza bruta, escogiendo la de mejor resultado 4.17.

- **Búsqueda aleatoria**: Se establece una cuadrícula con una serie de distribuciones probabilísticas para cada hiperparámetro y se evaluan combinaciones aleatorias de los mismos 4.18.

- **Optimización Bayesiana**: La métrica de error escogida se establece como una distribución probabilística desconocida, y tras evaluar distintas combinaciones de hiperparámetros, se asumen las características de dicha distribución y se busca el máximo.

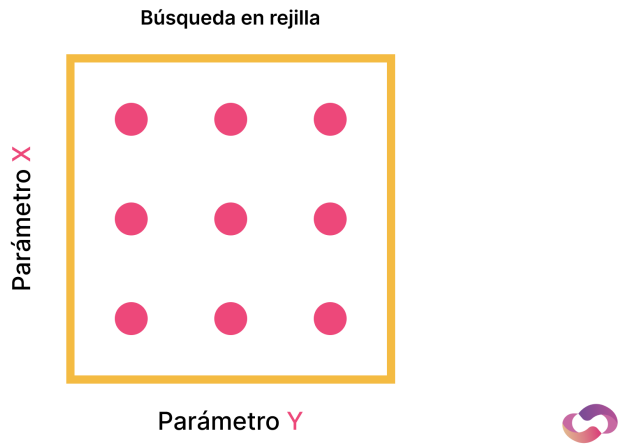

Figura 4.17: Búsqueda en rejilla de dos parámetros

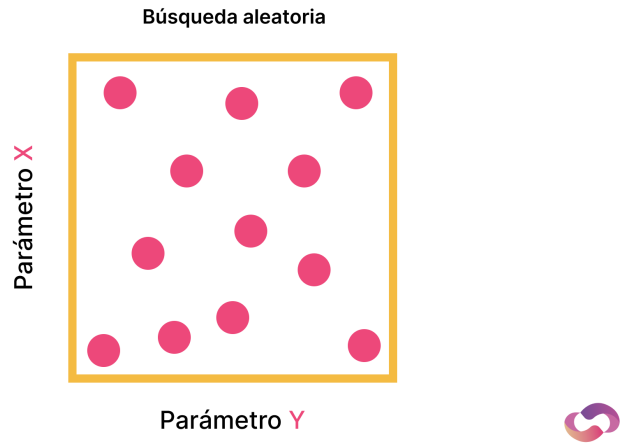

Figura 4.18: Búsqueda aleatoria de dos parámetros

En este caso, se escoge el uso de búsqueda en rejilla, por ser el más simple y por tener un conjunto de hiperparámetros suficientemente acotado como para que sea viable el uso de la misma.

4.5.3. Recurrentes (LSTM)

Las redes neuronales recurrentes fueron introducidas en el propio trabajo sobre backpropagation de Rumelhart et al [15], donde se definieron como redes circulares que disponen de "memoria", permitiendo almacenar información en el tiempo, donde para cada celda recibe una entrada x_t, produce una salida h_t y envía un mensaje a la siguiente celda 4.19. Posteriormente, Hochreiter et al (1997) [17] introdujeron las redes LSTM para lidiar con el problema del **desvanecimiento del gradiente**, que implicaba que al entrenar una red neuronal recurrente con el algoritmo de descenso del gradiente, el gradiente de cada neurona se iba reduciendo hasta valores muy pequeños, debido a que las derivadas parciales de algunas funciones de activación tenían rango $[0, 1]$, haciendo que algunas neuronas no se actualizasen de forma eficaz; también resolvía el problema opuesto, **gradiente explosivo**.

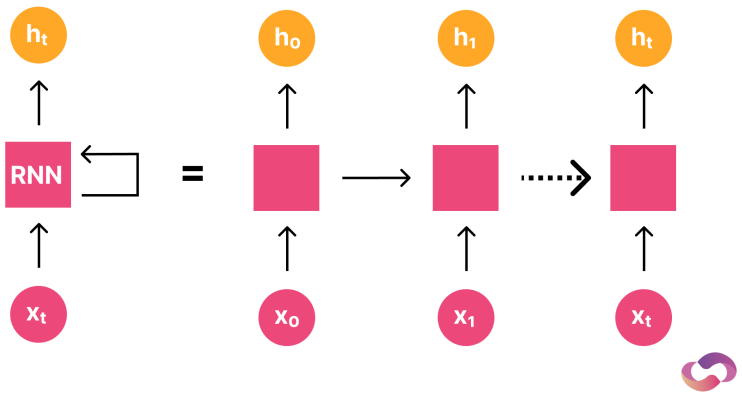

Figura 4.19: Desplegado de una red neuronal recurrente

Las celdas LSTM están compuestas por dos estados, el estado de celda C, que es la memoria a largo plazo de la red; y el estado escondido h, que es la salida de cada neurona en cada paso 4.20.

Una celda LSTM recibe como entrada :

- Estado de celda previo (C_{t-1})
- Estado escondido previo (h_{t-1})
- Dato de entrada (x_t)

Y genera como salida :

- Estado de celda nuevo (C_t)
- Estado escondido nuevo y dato de salida de la neurona (h_t)

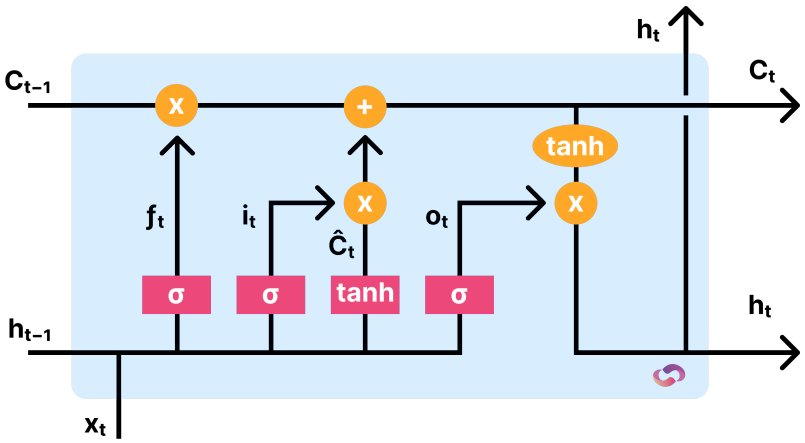

Figura 4.20: Celda LSTM

Una celda LSTM realiza el siguiente procedimiento:

1. Olvidar En este paso, la celda decide cuál de la información a largo plazo de la red (estado de celda) es importante mantener dada la entrada y el estado escondido. Para ello utiliza una red neuronal con una activación sigmoide y luego realiza una multiplicación punto a punto. Al usar una activación sigmoide, la salida de dicha operación esta entre el intervalo $[0, 1]$, con lo que al multiplicar por el estado de celda previo, se realiza una especie de filtro sobre la información a largo plazo 4.21.

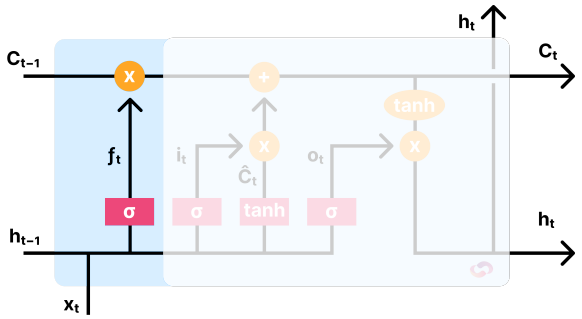

Figura 4.21: Parte de la celda que olvida

2. Recordar Posteriormente, la celda decide que nueva información recordar. Para ello utiliza dos redes neuronales que trabajan juntas; la primera de ellas utiliza una función tangente hiperbólica como función de activación, esta produce resultados en el intervalo

$[-1, 1]$, y su resultado se puede entender como un **vector de actualización de la memoria**; la otra red, muy similar a la del paso 1, actúa como un filtro con una función de activación sigmoide, y su resultado es multiplicado punto a punto con el de la otra red y este es el vector que se suma al estado de celda 4.22.

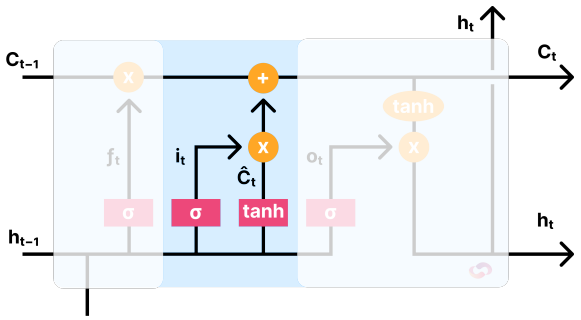

Figura 4.22: Parte de la celda que recuerda

3. Generar salida Por último, la celda genera su salida, que es el estado escondido de la misma. De nuevo, para ello se necesita una red neuronal que actúe de filtro con una función de activación sigmoide, y como novedad, un filtro (no es red neuronal) que utilice una función de activación tangente hiperbólica para forzar al estado de celda a estar en el intervalo $[-1, 1]$; los resultados se multiplican punto a punto y el vector resultante es el estado escondido 4.23.

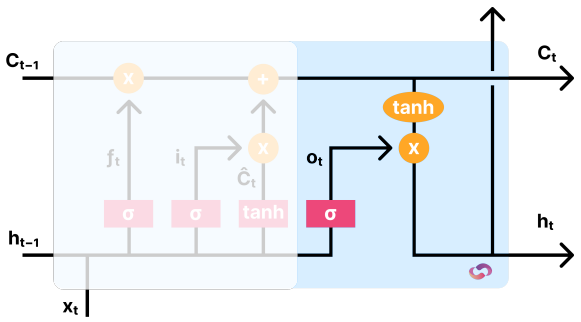

Figura 4.23: Parte de la celda que genera la salida

Modelo

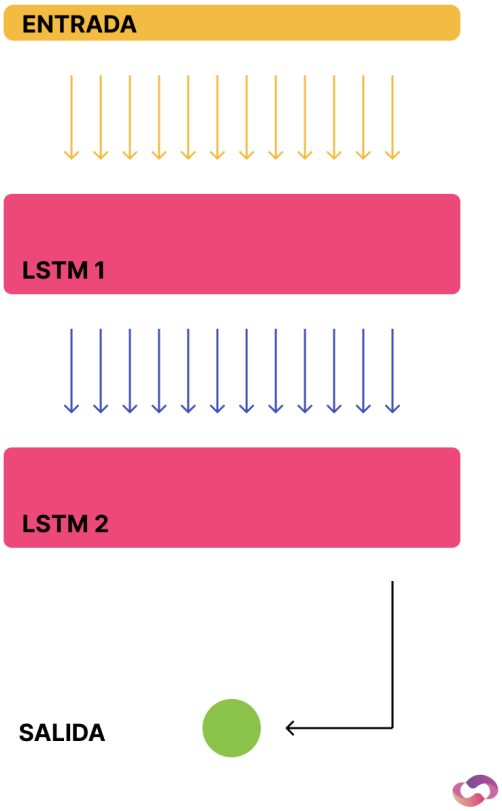

Figura 4.24: Arquitectura de la red LSTM

Para elegir el modelo, se utiliza la **búsqueda en rejilla** juntamente con el método de **validación cruzada** con el **MAE** como métrica. La validación cruzada se usa con una **paciencia** de 20 épocas y una **delta mínima** de 0. Para utilizar la búsqueda en rejilla, se necesita seleccionar una serie de valores para cada uno de los hiperparámetros del modelo 4.1. También se utiliza la parada temprana. En este caso, se plantea una red compuesta por dos capas LSTM apiladas, ya que en 2013, Graves et al [18] demostraron cómo el uso de varias capas apiladas mejoraba la predicción frente a el uso de una sola; donde la primera de ellas devuelve datos con la misma longitud que su entrada (número de semanas previas utilizadas para la predicción), pero con un número de dimensiones determinado llamado **unidades**, y la segunda toma las salidas de la primera y devuelve un resultado único, la predicción final 4.24. Cada capa LSTM está formada por tantas celdas como longitud tenga el vector de entrada, y todas estas celdas utilizan una misma función de activación, la **ReLU**. Las distintas puertas de cada celda LSTM pueden tener **dropout**, siendo el mismo para todas las celdas de toda la red neuronal. Con todo esto, los hiperparámetros a elegir y sus posibles valores son:

Hiperparámetro	Posibles Valores
Tasa de dropout	0/0.25/0.5/0.7
Unidades de la primera capa LSTM	1/2/6/10/16/20
Longitud del vector entrada	1/8/12/24/60/80
Utilización de variables exógenas (clima)	si/no

Tabla 4.1: Posibles valores de los hiperparámetros de la red LSTM

La única diferencia entre el modelo de predicción a corto plazo y el modelo a largo plazo es la longitud del vector de salida producido por la segunda capa LSTM, que en el caso de predicción a corto plazo es de longitud 1, y en el caso de predicción a largo plazo, de 52. Además, el modelo de predicción a largo plazo no puede emplear variables exógenas.

Resultados

El mejor resultado obtenido en predicción a corto plazo es un MAE de 0.138, mejorando en un 17.9 % al modelo baseline y en un 4.8 % al modelo SARIMAX. Este modelo corresponde a los siguientes valores de los hiperparámetros:

Hiperparámetro	Valor escogido
Tasa de dropout	0.25
Unidades de la primera capa LSTM	6
Longitud del vector entrada	12
Utilización de variables exógenas (clima)	si

Tabla 4.2: Valores óptimos de los hiperparámetros de la red LSTM para predicción a corto plazo

Por otro lado, en predicción a largo plazo, se ha obtenido un MAE de 0.280, mejorando al modelo de referencia en un 36.2 % y al modelo SARIMA en un 0.7 %; correspondiendo a los siguientes valores para cada hiperparámetro:

Hiperparámetro	Valor escogido
Tasa de dropout	0.5
Unidades de la primera capa LSTM	10
Longitud del vector entrada	60
Utilización de variables exógenas (clima)	no

Tabla 4.3: Valores óptimos de los hiperparámetros de la red LSTM para predicción a largo plazo

El modelo a corto plazo tiene internamente 489 parámetros a entrenar y el a largo plazo, 3288.

4.5.4. Convolucionales temporales (TCN)

Las redes neuronales convolucionales (CNN) son una clase de algoritmos de *Deep Learning* utilizados comunmente para análisis de imágenes. Las imágenes se representan de forma interna como una matriz de valores entre 0 y 1 para cada canal (rojo, verde, azul) 4.25. Estas redes, nacen para solucionar el problema que las redes neuronales normales tienen con las imágenes: necesitan aplanarlas en un vector de $((ancho * alto), 1)$, perdiendo las dependencias entre píxeles que están en distintas filas. Las CNN están formadas por dos tipos de capas: las convolucionales y las de submuestreo.

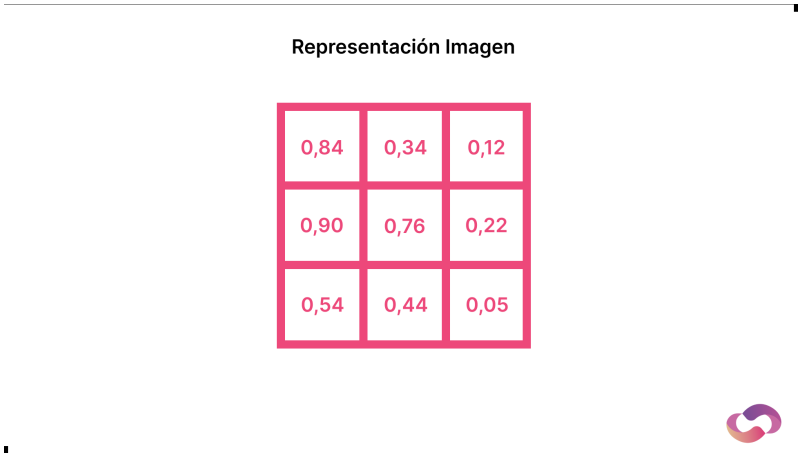

Figura 4.25: Representación de una imagen con un solo canal (escala de grises)

Convolución Las capas de convolución aplican un filtro (**kernel**) a la imagen para obtener el llamado mapa de características. Este filtro es aplicado de forma recurrente desplazándose por todas las ventanas de la imagen, como una multiplicación punto a punto 4.26. Puede aplicarse *padding* a los lados, en forma de ceros, para no perder la información de los laterales.

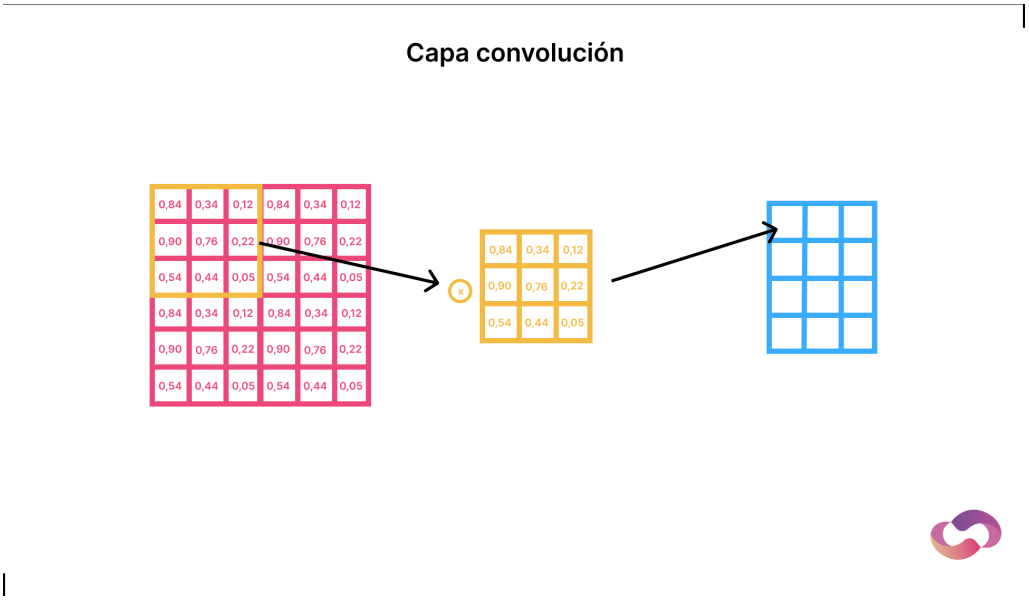

Figura 4.26: Capa de convolución

Submuestreo Las capas de submuestreo son otro filtro, que se aplica después de las capas de convolución, cuyo tamaño es menor al del mapa de características obtenido (generalmente tamaño 2x2) y que sirve para reducir las dimensiones del mismo 4.27. Este filtro no se aprende, si no que se especifica al diseñar el modelo, los más comunes son la media de los elementos, el máximo y el mínimo .

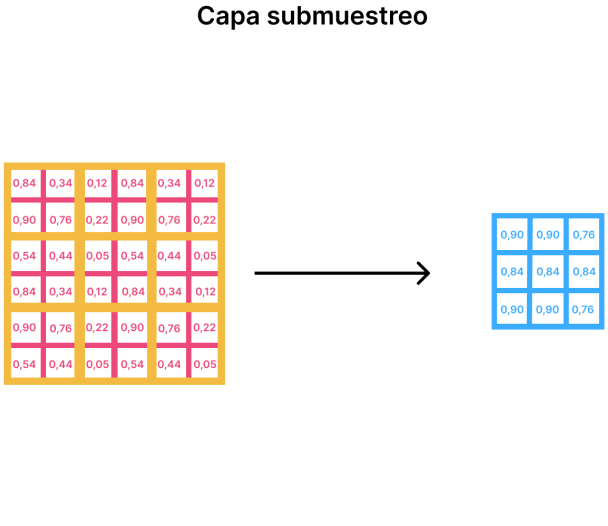

Figura 4.27: Capa de submuestreo utilizando el máximo

En 1980, Fukushima, K. [19] introdujo la idea del **neocognitron**, que es considerado el precursor de las redes convolucionales, en este se mencionaban la extracción de características, las capas de convolución y de submuestreo; posteriormente, en 1998, LeCun et al. [20] definieron la primera CNN, **LeNet-5**, que batió a todos los algoritmos existentes en detección de dígitos manuscritos. A partir de entonces, surgió el dataset **ImageNet**, que contiene más de 14 millones de imágenes para clasificar, y sobre el que, hasta 2017, se organizó una competición en la que surgieron los modelos convolucionales mas conocidos como el GoogLeNet o ResNet.

Posteriormente, en 2018, Bai et al [21], compararon las redes recurrentes clásicas con el uso de redes convolucionales para modelar series temporales, estas últimas fueron bautizadas como Redes Convolucionales Temporales (TCN).

Convolución unidimensional Al contrario que las imágenes, comúnmente formadas por dos dimensiones, las series temporales tienen solo una (pueden tener múltiples características, pero no dejan de ser series independientes unidimensionales). Por ello, se utilizan capas convolucionales unidimensionales. Estas funcionan empleando un Kernel $K^{N,D}$ donde N es el tamaño del kernel y D el número de dimensiones o características de la serie temporal. Este kernel es multiplicado de forma escalar 4.28 con todos los desplazamientos del input para formar una salida con la misma longitud que la entrada 4.29. Aunque la entrada sea multidimensional, el kernel siempre se desplaza en una sola dimensión.

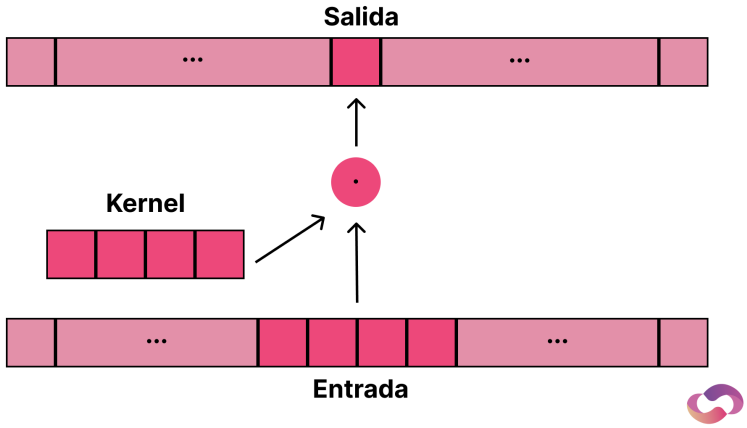

Figura 4.28: Convolución unidimensional simple con kernel $K^{4,1}$

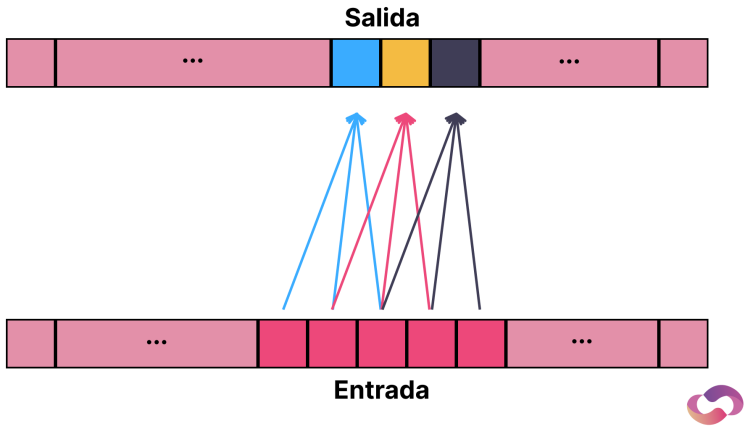

Figura 4.29: Desplazamiento de la convolución unidimensional con kernel $K^{3,1}$

Convolución causal En el ejemplo anterior, se puede observar que para todo kernel $K^{N,D}$ con $N > 1$ la salida en t_i será producto de la convolución de, entre otros, valores futuros de la entrada 4.30. Es deseable que esto no ocurra, y que para un kernel $K^{N,D}$, el valor de salida St_i sea producto de la convolución de los valores $Et_{i-D}...Et_i$. Para ello se introduce el concepto de convolución causal, que es aquella convolución que sigue esta propiedad gracias a aplicar más padding en la izquierda, con lo que cada valor de la salida tiene como última dependencia el valor de la entrada en el mismo índice 4.31.

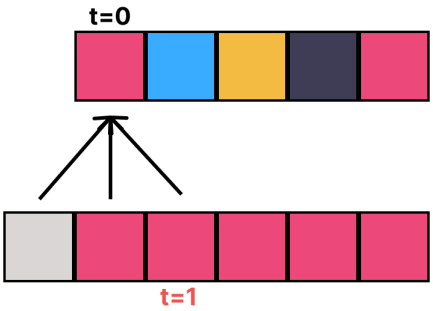

Figura 4.30: Convolución con kernel $K^{3,1}$ en la que se utilizan valores futuros

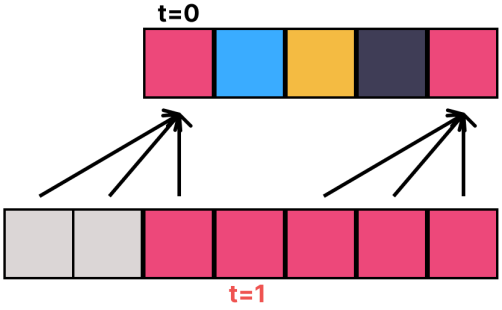

Figura 4.31: Convolución causal con kernel $K^{3,1}$

Múltiples capas y dilatación El valor de salida St_i dada la convolución con un kernel $K^{N,D}$ depende de N valores de la entrada, siendo el más lejano el valor Et_{i-N}. Idealmente, cada valor de la salida debería depender de todos los valores anteriores de la entrada para lograr una cobertura histórica completa. Para tratar de conseguir esta propiedad, se utilizan distintas técnicas y modificaciones. En primer lugar, se pueden apilar capas convolucionales, cada una de ellas con un kernel distinto, para ampliar las dependencias que cada valor de la salida tiene. Sin embargo, para ser capaces de que cada valor de la salida dependa de todos los anteriores de la entrada, necesitamos un número de capas C dado por la siguiente ecuación:

$$C = \frac{L-1}{N-1}$$

donde L es la longitud de la entrada y N el tamaño del kernel. El número de capas por tanto, crece de forma lineal con la longitud de la entrada, y considerando que cada capa tiene un kernel distinto, la cantidad de parámetros a entrenar para lograr una cobertura completa sería excesiva.

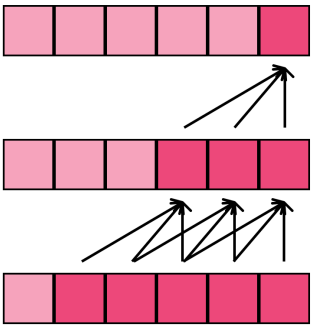

Figura 4.32: Convoluciónes apiladas todas con kernel $K^{3,1}$

Por otro lado, se emplea el uso del concepto de **dilatación**, utilizado también en las CNN comunes. La dilatación es la aplicación de un margen d entre cada valor de la entrada que se utiliza en una convolución 4.33. Todos los ejemplos vistos anteriormente utilizan una dilatación d=1.

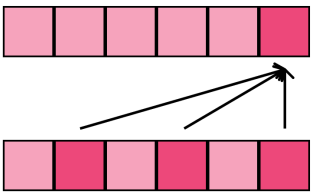

Figura 4.33: Dilatación d=2 con kernel $K^{3,1}$

La dilatación se puede combinar con el uso de múltiples capas convolucionales apiladas, ampliando mucho la cobertura aunque manteniendo la necesidad de un número de capas muy grande pues sigue creciendo de forma linear con la longitud del vector de entrada.

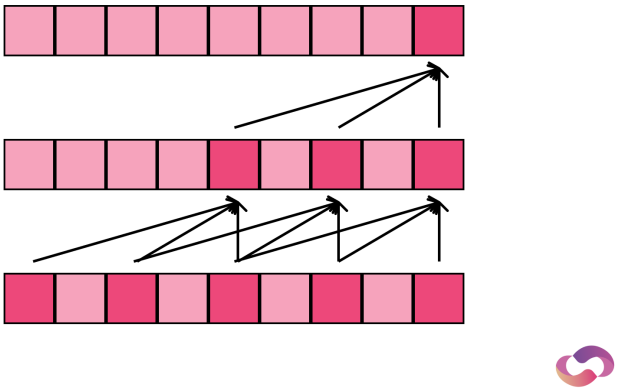

Figura 4.34: Múltiples capas convolucionales con dilatación d=2 y kernel $K^{3,1}$

Con el fin de solventar este problema, se introduce el concepto de **dilatación dinámica**, utilizando una dilatación creciente según el número de la capa. Para una capa i, su dilatación d_i es:

$d_i = (d_{base})^i$

Donde d_{base} es la dilatación base, elegida como un hiperparámetro.

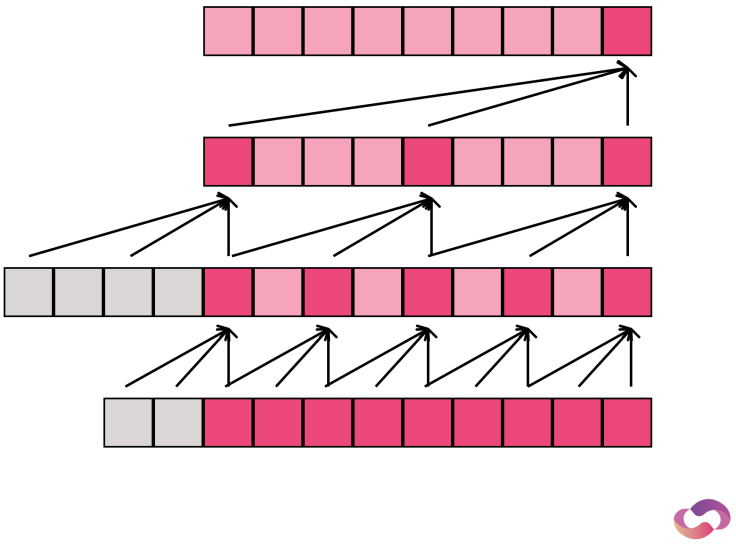

Figura 4.35: Dilataciones dinámicas con kernel $K^{3,1}$ y dilatación base d_{base}=2

Modelo

De nuevo, se emplea el uso de **búsqueda en rejilla** para escoger los valores de los hiperparámetros 4.4; con **validación cruzada** y **MAE**, con **parada temprana** con los mismos parámetros que el modelo LSTM (20 épocas de paciencia y una delta mínima de 0).

Para este modelo, se emplea un número determinado de capas convolucionales temporales apiladas, con una dilatación base de 2 ($d_i = d_{base}$), un Kernel de iguales dimensiones para cada capa ($K^{N,D}$), una tasa de dropout igual para todas las capas convolucio-

nales, y ReLU como función de activación para todas las neuronas de la red. En este caso, no se emplean capas de submuestreo debido a que estas, en el contexto de redes convolucionales temporales, funcionan en caso de tener entradas muy largas como medida de reducción; en este caso, en el peor de los casos son vectores de longitud 80, con lo que no es necesario su uso y podría llevar a perdida de información.

Hiperparámetro	Posibles Valores
Tasa de dropout	0/0.10/0.25/0.5
Número de capas convolucionales apiladas	1/2/3/4
Longitud del vector entrada	1/8/12/24/60/80
Tamaño del kernel de convolución	$K^{2,D}/K^{3,D}/K^{4,D}$
Utilización de variables exógenas (clima)	si/no

Tabla 4.4: Posibles valores de los hiperparámetros de la red TCN

Resultados

El mejor resultado obtenido en predicción a corto plazo es un MAE de 0.147, siendo igual que el obtenido por el modelo SARIMA sin variables exógenas. El modelo elegido corresponde a los siguientes valores de los hiperparámetros:

Hiperparámetro	Valor escogido
Tasa de dropout	0
Número de capas convolucionales apiladas	1
Longitud del vector entrada	8
Tamaño del kernel de convolución	$K^{3,D}$
Utilización de variables exógenas (clima)	si

Tabla 4.5: Valores óptimos de los hiperparámetros de la red TCN para predicción a corto plazo

Para predicción a largo plazo, el mejor resultado obtenido es un MAE de 0.297, mejorando en un 32.3 % al modelo de referencia pero empeorando en un 5.1 % y 5.7 % a los modelos SARIMA y LSTM respectivamente. El modelo elegido corresponde a los siguientes valores de los hiperparámetros:

Hiperparámetro	Valor escogido
Tasa de dropout	0.10
Número de capas convolucionales apiladas	2
Longitud del vector entrada	60
Tamaño del kernel de convolución	$K^{2,D}$
Utilización de variables exógenas (clima)	no

Tabla 4.6: Valores óptimos de los hiperparámetros de la red TCN para predicción a largo plazo

Los resultados son peores de lo esperado, sobretodo teniendo en cuenta los numerosos estudios que indican que las redes TCN generalmente mejoran a las recurrentes. En este caso, los parámetros internos del modelo de corto plazo a entrenar son 1368, casi 3 veces más que los del modelo LSTM; este número es excesivo para la cantidad de datos disponibles, lo que impide que el modelo aprenda de forma correcta.

4.6 Comparación de resultados

Método	Predicción a corto plazo	Predicción a largo plazo
Referencia	0.168	0.439
SARIMA	0.147	0.282
SARIMAX	0.145	-
Prophet	0.306	0.310
LSTM	**0.138**	**0.280**
TCN	0.147	0.297

Tabla 4.7: Comparación de los resultados (MAE) para cada uno de los modelos probados

Para ambos tipos de predicción, el mejor modelo ha sido el LSTM, seguido de cerca por los modelos SARIMAX para corto plazo y SARIMA para largo plazo. El modelo Prophet no ha funcionado bien debido a la naturaleza compleja de los datos y el modelo TCN ha rendido por debajo de las expectativas, en gran parte debido al excesivo número de parámetros a entrenar en el mismo 4.7.

CAPÍTULO 5
Conclusiones

Tras haber analizado todos los modelos, los resultados obtenidos han mejorado a los de los modelos de referencia, sin embargo, las mejoras no son del todo significativas, y los resultados no son los esperados; esto es debido a que las Alhóndigas intervienen en los precios y manipulan la subasta, de forma que el precio final no sólo depende de la oferta y la demanda, sino que hay un fuerte componente humano difícil de modelar. Este componente humano no estaría presente en el caso de estar prediciendo los kilogramos de producción, para ello se podrían llegar a acuerdos con distintos invernaderos de la zona y obtener datos para poder estimar la producción por hectárea.

Por otro lado, el procesamiento y análisis de los datos ha confirmado la existencia de cierta correlación del precio con el clima, lo cual evidencia que catástrofes naturales como la del Mar Menor o la más reciente crisis climática producida por la calima del polvo del Sáhara, pueden influir en la producción de invernaderos con baja tecnología, siendo este un argumento a favor de la modernización de estos invernaderos. También se ha determinado que la inflación no afecta, al menos en Almería, al precio a largo plazo; ya que no se observa una tendencia en el precio a lo largo de los años.

También se ha podido observar como los modelos de *Deep Learning* si que pueden mejorar a los modelos estadísticos clásicos en tareas de regresión, y como la reducción de dimensionalidad en datos muy extensos favorece el resultado de todos los modelos. Sin embargo, los modelos convolucionales temporales tienen muchos más parámetros internos, con lo que no son adecuados para la predicción de series temporales con una cantidad de datos limitada.

Para todo ello, se ha tenido que profundizar en algunos conceptos estudiados durante la carrera; en la asignatura **Estadística** se aprendieron algunos conceptos estadísticos básicos utilizados en algunas de las pruebas y modelos utilizados; en la asignatura **Sistemas Inteligentes** se introdujo el concepto de inteligencia artificial y aprendizaje automático, junto a sus variantes y también se aprendieron algunos algoritmos básicos; en la asignatura **Percepción** se profundizó en algunos conceptos de la asignatura anterior, introduciendo también la reducción de dimensionalidad junto a algoritmos como el PCA y el LDA, el primero de ellos utilizado en este estudio, además de métricas como el error absoluto medio; posteriormente, en un intercambió académico realizado en la universidad *Kungliga Tekniska högskolan (KTH)* de Estocolmo, se realizaron asignaturas más especializadas en la inteligencia artificial y el machine learning, en la asignatura **Machine Learning** se introdujeron otros algoritmos básicos y el concepto de validación cruzada; en la asignatura **Artificial Neural Networks and Deep Learning** se introdujeron las redes

neuronales básicas, las convolucionales y las recurrentes (LSTM) y sus hiperparámetros. Para toda la información y metodología del mundo de la agricultura se ha contactado con distintos expertos del sector y también con el co-tutor de este trabajo. Las técnicas de análisis exploratorio de los datos, los modelos derivados de los ARIMA, el modelo Prophet y las redes TCN han sido investigadas de forma autodidacta en distintas publicaciones y artículos científicos.

Bibliografía

[1] Nieves, V. (2021). España, sector a sector: la agricultura se hace fuerte en la crisis y alcanza su mayor peso en el PIB en 15 años. *elEconomista*https://www.agroclm.com/2022/01/27/espana-creo-58-000-empleos-en-la-agricultura-en-2021/.

[2] España creó 58.000 empleos en la agricultura en 2021. (2022). *agroclm*https://www.agroclm.com/2022/01/27/espana-creo-58-000-empleos-en-la-agricultura-en-2021/.

[3] García, Z. (2016). *Agricultura, expansión del comercio y equidad de género*. Disponible en : https://www.fao.org/publications/card/en/c/c88b7b9c-16ef-52f6-a8f4-aeca52fd3021/

[4] León, M. (2016). Almería corona la cumbre de las 30.000 hectáreas de invernadero. *La Voz de Almería* https://www.lavozdealmeria.com/noticia/20/economia/98698/almeria-corona-la-cumbre-de-las-30-000-hectareas-de-invernadero.

[5] INE. (2016). *Encuesta sobre la estructura de las explotaciones agrícolas año 2016* https://www.ine.es/jaxi/Tabla.htm?path=/t01/p044/a2016/ccaa00/l0/&file=1101.px.

[6] Jaramillo Noreña, J., Patricia Rodríguez, V., Guzmán A., M. y A. Zapata, Miguel. (2006). *El cultivo de tomate bajo invernadero*

[7] De Pablo Valenciano, J. y Perez Mesa, J. C. (2002). Las alhóndigas: pasado, presente y futuro. *Distribución y consumo* 12(66), pp.88-98 https://www.mapa.gob.es/ministerio/pags/biblioteca/revistas/pdf_DYC/DYC_2002_66_88_96.pdf

[8] Meshram, V., Patil, K., Meshram, V., Hanchate, D. y Ramkteke, S.D,. (2021). Machine learning in agriculture domain: A state-of-art survey. *Artificial Intelligence in the Life Sciences* 1 https://doi.org/10.1016/j.ailsci.2021.100010.

[9] Verma, P., Reddy, S. V., Ragha, L. y Datta, D. (2021). Comparison of Time-Series Forecasting Models. *2021 International Conference on Intelligent Technologies (CONIT)*, pp. 1-7 https://doi.org/10.1109/CONIT51480.2021.9498451

[10] Zhang, R., Song, H., Chen, Q., Wang, Y., Wang, S. y Li, Y. (2022). Comparison of ARIMA and LSTM for prediction of hemorrhagic fever at different time scales in China. *PLOS ONE*, 1(17), pp. 1-14 https://doi.org/10.1371/journal.pone.0262009.

[11] Dickey, A. D. y Fuller Wayne, A. (1979). Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association*, 74(336a), pp. 427-431 https://doi.org/10.1080/01621459.1979.10482531

[12] Stone, M. (1977). An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1), pp. 44-47 http://www.jstor.org/stable/2984877

[13] McCulloch, W. S. y Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), pp. 115-133 https://doi.org/10.1007/BF02478259

[14] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), pp. 386-408 https://doi.org/10.1037/h0042519

[15] Rumelhart, D., Hinton, G. y Williams, R. (1986). ThLearning representations by back-propagating errors. *Nature*, 323, pp. 533–536 https://doi.org/10.1038/323533a0

[16] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. y Salakhutdinov R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56), pp. 1929-1958 https://doi.org/10.5555/2627435.2670313

[17] Hochreiter, S. y Schmidhuber, J. (1997). Long Short-term Memory. *Neural computation*, 9(8), pp. 1735–1780 https://doi.org/10.1162/neco.1997.9.8.1735

[18] Graves, A., Jaitly, N. y Mohamed, A. (2013). Hybrid speech recognition with Deep Bidirectional LSTM. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 273-278 https://doi.org/10.1109/ASRU.2013.6707742

[19] Fukushima, K. (1980). Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biol. Cybernetics*, 36, pp. 193-202 https://www.rctn.org/bruno/public/papers/Fukushima1980.pdf

[20] LeCun, Y., Bottou, L., Bengio. Y. y Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), pp. 2278-2324 https://doi.org/10.1109/5.726791

[21] Bai, S., Kolter, J. Z. y Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv: Learning* https://doi.org/10.48550/ARXIV.1803.01271

[22] Dean, R. B. y Dixon, W. J. (1951). Simplified Statistics for Small Numbers of Observations. *Anal. Chem. 1951*, 23(4), pp. 636–638 https://doi.org/10.1109/CONIT51480.2021.9498451

APÉNDICE A
Pruebas estadísticas y filtros utilizados

A.1 Pruebas de hipótesis

Una prueba de hipótesis es un tipo de prueba estadística donde se especifica si se debe aceptar o rechazar una información determinada en base a los datos. En las pruebas de hipótesis, se plantea una hipótesis nula y una hipótesis alternativa, y se determina si se puede o no rechazar la hipótesis nula en base a una estadística obtenida desde los datos. Esta estadística se denomina **p-value**, y es la log-similitud de los datos si se cumple la hipótesis nula, y si este valor es menor que el nivel de significación (generalmente 0.05) se puede rechazar la hipótesis nula y por tanto aceptar la alternativa.

A.1.1. Pruebas de raíz unitaria

Una prueba de raíz unitaria comprueba si una serie temporal es estacionaria; son pruebas de hipótesis, donde la hipótesis nula es que la seria posee una raíz unitaria y por tanto no es estacionaria, y la hipótesis alternativa es que la serie es estacionaria.

A.1.2. Prueba Q de Dixon

La prueba Q de Dixon [22] es una prueba estadística de hipótesis nula que permite determinar si un determinado valor en una serie de datos **ordenada ascendentemente** es un valor anómalo, la fórmula de esta prueba para cada valor de una serie es:

$$Q = \frac{separacin}{rango}$$

Donde *separación* es la diferencia absoluta entre el valor a estudiar y el siguiente en la serie ($|x_n - x_{n+1}|$) y *rango* es la diferencia entre el primer y último valor ($|x_0 - x_N|$). El valor Q resultado es entonces comparado con un valor referencia que depende del número de elementos que contiene la serie y del intervalo de confianza elegido (Q_{crit}). Si $Q \geq Q_{crit}$, el valor es anómalo y se elimina de la serie.

A.2 Filtro de Hampel

El objetivo del filtro de Hampel es encontrar valores anómalos en una serie de datos temporal. Este filtro funciona estableciendo una ventana de un tamaño v, y para cada

valor, toma los v elementos circundantes, calcula la mediana de sus desviaciones típicas ($mediana(x_i - u)$, y imputa al valor como anómalo si su desviación típica difiere más de y veces la de la mediana.

A.3 Coeficiente de correlación de Pearson

El coeficiente de correlación de Pearson es una medida de la relación recíproca entre dos variables, su formula es la siguiente:

$$\rho = \frac{Covarianza(X,Y)}{\sqrt{Var(X) \cdot Var(Y)}} \tag{A.1}$$

Este coeficiente se expresa en el rango [-1,1], donde 0 indica que no hay correlación entre las variables, -1 implica una correlación total en direcciones opuestas y 1 en la misma dirección.

[11pt]article

ods$_e tsinf$

APÉNDICE B
Objetivos de Desarrollo Sostenible (ODS)

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	**Alto**	**Medio**	**Bajo**	**No procede**
ODS 1. **Fin de la pobreza.**		X		
ODS 2. **Hambre cero.**	X			
ODS 3. **Salud y bienestar.**		X		
ODS 4. **Educación de calidad.**				X
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**			X	
ODS 7. **Energía asequible y no contaminante.**			X	
ODS 8. **Trabajo decente y crecimiento económico.**		X		
ODS 9. **Industria, innovación e infraestructuras.**			X	
ODS 10. **Reducción de las desigualdades.**				X
ODS 11. **Ciudades y comunidades sostenibles.**				X
ODS 12. **Producción y consumo responsables.**		X		
ODS 13. **Acción por el clima.**			X	
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**				X
ODS 17. **Alianzas para lograr objetivos.**				X

La agricultura es uno de los principales agentes que garantizan la alimentación y el bienestar de la sociedad. Además, es uno de los principales generadores de empleo, sobretodo en aquellos países menos desarrollados. La evolución tecnológica y las nuevas técnicas de cultivo y explotación intensiva han traído grandes avances para el sector, fomentando explotaciones mas sostenibles.

Este estudio se ha basado en las subastas de la zona de Almería, pero se podría extender a toda zona con características similares (invernaderos pasivos); la agricultura de los países menos desarrollados, que no puede competir en los mercados internacionales, suele ser de invernaderos pasivos, con lo que se podría aplicar este algoritmo y ayudar a los agricultores a tomar ciertas decisiones con información muy barata, ya que solo se necesita un ordenador y una fuente de datos de precio y de clima; esto podría ayudarles en su lucha contra la pobreza, lo que esta relacionado con el ODS 1.

Por otro lado, muchas veces se planta menos cultivo del que se puede por dificultades para gestionarlo todo o porque si el precio ha estado bajo, algunos agricultores deciden no plantar; esto provoca épocas con menos cultivo en subasta y hambrunas en aquellos países más dependientes del cultivo. Además, relacionado con el ODS 1, en algunos lugares los invernaderos no son rentables por el poco beneficio que obtienen tras emplear estrategias incorrectas. La predicción del precio del cultivo podría tratar de solventar los dos problemas y contribuir a la lucha por el hambre cero, que corresponde al ODS 2.

De igual manera, gracias al algoritmo los agricultores pueden obtener más beneficio y plantar más, con lo que el hambre disminuye y la salud general de la población aumenta. Pero además, las mayores ganancias por parte de los agricultores pueden llevarlos a decidir invertir en técnicas más avanzadas de cultivo para mejorar la calidad de los cultivos, mejorando el bienestar de la población en general y fomentando el consumo de productos saludables; todo esto esta relacionado con el ODS 3.

Asimismo, gran parte de cultivo al año es desperdiciado debido a que los beneficios obtenidos tras su venta son menores al coste de distribución y transporte del producto una vez ya cosechado; esto conlleva una alta perdida en materias y alimentos. Con el algoritmo de predicción de precios, se podría controlar cuando no hay que cultivar, emitiendo muchos menos residuos sobre el agua (ODS 6) y consumiendo menos electricidad (ODS 7 y 13) y reduciendo la cantidad de alimento desperdiciado (ODS 12).

Relacionado con el desperdicio de cultivo anterior, al perder una cosecha no solo se pierden alimentos y materia prima, también se pierde el tiempo y el esfuerzo de los trabajadores de cosecha, a menudo trabajando en condiciones muy duras, obteniendo cero beneficio a cambio. Gracias a que, utilizando el algoritmo, solo se planta y cosecha cuando se va a obtener beneficio, este trabajo no es desperdiciado, con lo que el personal de cosecha puede trabajar menos generando el mismo beneficio; esto esta ligado con el OBS 8.

En conclusión, la agricultura es uno de los factores diferenciales que muchos países disponen para garantizar la calidad de vida de sus habitantes. Es importante fomentar la modernización de el sector y desarrollar soluciones a problemas aplicando las últimas tecnologías, como lo es el *Machine Learning*.