Universidad de Cantabria

Facultad de Ciencias

Procesos Gaussianos con aplicaciones en inversiones

(Gaussian Processes with investment applications)

Trabajo de Fin de Máster
para acceder al

Máster en Data Science UC-UIMP

Autor: Marcos Cobo Carrillo
Director: Steven Van Vaerenbergh

Septiembre 2022

A mis padres, familiares y amigos.
A las personas que me han apoyado y ayudado a lo largo de mi vida.

Agradecimientos

En primer lugar, me gustaría dar las gracias a Steven Van Vaerenbergh por su trabajo y dedicación depositados en estos últimos meses. Gracias por confiar en mí y darme la oportunidad de trabajar juntos. En segundo lugar, me gustaría dar las gracias a mis padres, por ser mi principal apoyo y los formadores de la persona que soy, sin ellos y sin su amor y cariño jamás habría llegado hasta donde estoy. Gracias a mis familiares y amigos, los cuales me han acompañado durante toda mi vida y con los cuales he compartido mis mejores momentos. También estoy muy agradecido a mis compañeros de facultad, quienes siempre me han ayudado con cualquier cosa que he necesitado. Por último, quiero dar las gracias a todos los profesores que he tenido a lo largo de mi vida y a todas aquellas personas que han contribuido en mi formación como estudiante y como persona.

Marcos Cobo Carrillo, El Astillero, 2022.

Resumen

El Trading Algorítmico es una modalidad de operación en mercados financieros que hace uso de algoritmos, reglas y procedimientos automatizados para ejecutar operaciones de compra y venta. Está asociado con los inicios de la computación, ya que con los primeros ordenadores se ejecutaban modelos sencillos para intentar predecir el precio de acciones y bonos. En la década de 1960 se asientan las bases para la revolución del Trading Algorítmico y las finanzas informatizadas mediante la creación de modelos estadísticos para la gestión de carteras de inversión. Desde entonces, el trading algorítmico ha experimentado una gran evolución en cuanto a modelos se refiere, contando actualmente con gran variedad de algoritmos de machine learning para operar en el mercado. En este TFM desarrollaremos una estrategia para operar en el mercado financiero, más en concreto, dentro del mercado de materias primas, fundamentada en el uso de los Procesos Gaussianos como regresor, y estudiaremos el rendimiento de dicha estrategia en este mercado. Finalmente, extraeremos algunas conclusiones del potencial que poseen los Procesos Gaussianos dentro del campo de la economía, y en concreto, dentro del mundo de las inversiones.

Palabras clave: Trading Algorítmico, Mercados financieros, Materias primas, Procesos Gaussianos, GPs, Data Mining, Machine Learning.

Abstract

Algorithmic Trading is a trading methodology in financial markets based on the use of algorithms, rules and automated procedures to execute buying and selling operations. It is associated with the beginnings of computing, as the first computers were used to run simple models to predict the price of stocks. In the 1960s, the base for the revolution of Algorithmic Trading and computerized finance was set by the creation of statistical models for the management of investment portfolios. Since then, algorithmic trading has evolved rapidly in terms of models, with all kinds of machine learning algorithms currently being used to operate in markets. In this TFM we will develop a strategy for trading in financial markets, more specifically, in commodities market, based on the use of Gaussian Processes as regressor, and we will study the performance of this strategy in this market. Finally, we will draw some conclusions about the potential of Gaussian Processes in the field of economics, and more specifically, in the world of investments.

Keywords: Algorithmic Trading, Financial Markets, Commodities, Gaussian Processes, GPs, Data Mining, Machine Learning.

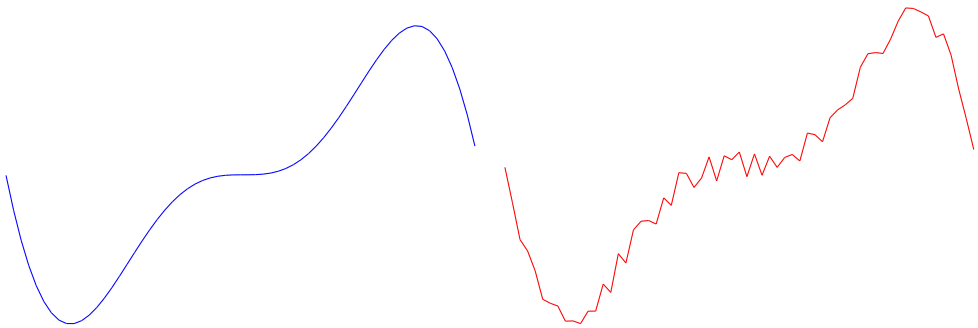
Índice general

Capítulo 1

Introducción

Un sistema es "un objeto complejo cuyas partes o componentes se relacionan con al menos alguna de las demás componentes, ya sea de manera conceptual o material" [6]. Podemos entender así un sistema como "una combinación de elementos que actúan juntos y realizan un objetivo determinado". Los sistemas dinámicos son aquellos cuyo estado evoluciona con el tiempo. Los sistemas físicos en situación no estacionaria son ejemplos de sistemas dinámicos, pero también existen modelos económicos, matemáticos y de otros tipos que son sistemas abstractos y, a su vez, dinámicos.

En los modelos ideales no existen perturbaciones, pero, en la vida real, éstas tienen un gran impacto en los sistemas, sobre todo, en los que son a nivel industrial o de gran complejidad y precisión.



(a) Sistema (b) Sistema perturbado

Figura 1.1: Sistema vs Sistema perturbado

El interés por la estimación del estado de un sistema perturbado es muy antiguo y ha sido objetivo de muchos pensadores a lo largo de la historia. En la Edad Moderna, Newton desarrolló la Teoría de la Mecánica y la Gravedad Universal, lo que dió paso al desarrollo de modelos dinámicos. Un siglo después, fue Gauss quien determinó la órbita de cuerpos celestes utilizando el método de "Mínimos Cuadrados". A partir de entonces, la estimación del estado de un sistema ha sido objeto de estudio de muchos matemáticos alrededor del mundo.

Con la llegada de los mercados financieros tal como los conocemos hoy en día, muchos fueron los que empezaron a especular con el precio de diversos activos financieros y empezaron a ver el mercado financiero como una forma rentable de ganar dinero con la compra y venta de dichos activos (acciones, futuros, ETFs, CFDs...). Dentro de este grupo de inversores, unos pocos empezaron a ver los modelos matemáticos como una herramienta para determinar la posición futura, o precio, de

los distintos activos en sus respectivos mercados. Aparecen así los primeros modelos y estrategias diseñadas específicamente para la predicción del precio de distintos activos financieros. Pero la verdadera revolución en este nicho vino de la mano del desarrollo de la computación. Nace así el Trading Algorítmico.

El Trading Algorítmico, o trading basado en reglas y procesos, llevado a cabo por bots de negociación o bots de trading, es una forma de negociación que utiliza plataformas electrónicas para introducir órdenes bursátiles dejando que un algoritmo decida sobre diversos aspectos de la orden, como la hora de apertura o de cierre (el momento), el precio o el volumen de la orden, en muchos casos sin ninguna intervención humana [8].

El Trading Algorítmico es muy utilizado por grandes instituciones financieras, como bancos, fondos de inversión, fondos de pensiones, etc. Estas grandes instituciones, además de pequeños fondos de inversión y muchos inversores privados, utilizan estas herramientas para operar sistemáticamente en el mercado y dividir las operaciones con el fin de gestionar el impacto y el riesgo de las mismas en el mercado.

Existe una categoría de trading algorítmico denominada trading de alta frecuencia (HFT), en la que los ordenadores toman decisiones complejas para emitir órdenes basadas en la información que se recibe electrónicamente, antes de que los inversores sean capaces de procesar la información que observan. Esta modalidad ha crecido rápidamente desde la transición de las bolsas a plataformas totalmente informatizadas y la entrada en vigor de la norma Reg NMS (Regulation National Market System), la cual tiene como objetivo garantizar que los inversores reciban las mejores ejecuciones de precios (NBBO) para sus órdenes fomentando la competencia en el mercado.

Esto ha dado lugar a un cambio radical en la micro-estructura de los mercados bursátiles, especialmente en la forma de proporcionar liquidez. En la HFT, el ritmo de creación de órdenes es el nanosegundo; el ritmo de colocación de órdenes es el microsegundo. Es tanta la velocidad y competencia entre distintos bots por la ejecución de órdenes, que para garantizar que sus órdenes se tengan en cuenta, las principales instituciones de inversión por HFT alquilan espacios en los centros informáticos próximos a las bolsas en las que operan.

En el Trading Algorítmico puede utilizarse cualquier estrategia de inversión, incluida la negociación por diferencias, el arbitraje o la especulación pura. De hecho, estas últimas modalidades tienden a amplificar los movimientos del mercado, dado que el gran número de decisiones se toman juntas, en la misma dirección y al mismo tiempo. Es más, desde la década de 2010, se estima que la proporción de órdenes cursadas en los mercados de valores a través de robots de negociación algorítmica oscila entre el 70 % y el 90 % . En 2006, esta proporción se situaba entre el 30 % y 40 % [8].

En los algoritmos no solo es importarte garantizar la velocidad y ejecución de las órdenes, si no que lo realmente importante es la calidad de las mismas. Hay casos en los que grandes firmas de inversión han perdido millones de dólares por poner a trabajar un bot en el cual el modelo subyacente para la estimación del precio no se adecuaba al sistema. Es por esta influencia que poseen los modelos matemáticos y estadísticos subyacentes en la toma de decisiones de los algoritmos, que los traders llevan tanto tiempo estudiando los mismos.

El Machine Learning, o aprendizaje automático, es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan. Se dice que un agente aprende cuando su desempeño mejora con la

experiencia y mediante el uso de datos, es decir, cuando la habilidad no estaba presente en su genotipo o rasgos de nacimiento. En machine learning, un computador observa datos, construye un modelo basado en esos datos y utiliza ese modelo como una hipótesis acerca del mundo y una pieza capaz de resolver problemas [4].

Dentro del mundo del trading algorítmico, se usan multitud de técnicas pertenecientes al campo del machine learning, normalmente relativas al aprendizaje supervisado, como GLM (generalized linear model), árboles de decisión, SVM (support-vector machines), redes neuronales (DNN, RNN, LSTM, CNN...) o procesos gaussianos (GPs), pero también se usan técnicas de aprendizaje por refuerzo, o aprendizaje no supervisado, como Hierarchical clustering o PCA, entro muchos otros.

En este trabajo nos centraremos en la técnica de procesos gaussianos. En machine learning, un proceso gaussiano (GP) es un proceso estocástico (una colección de variables aleatorias indexadas por tiempo o espacio), de modo que cada colección finita de esas variables aleatorias tiene una distribución normal multivariante, es decir, cada combinación lineal finita de ellas está normalmente distribuida. La distribución de un proceso gaussiano es la distribución conjunta de todas esas (infinitas) variables aleatorias y, como tal, es una distribución sobre funciones con un dominio continuo, por ejemplo, tiempo o espacio.

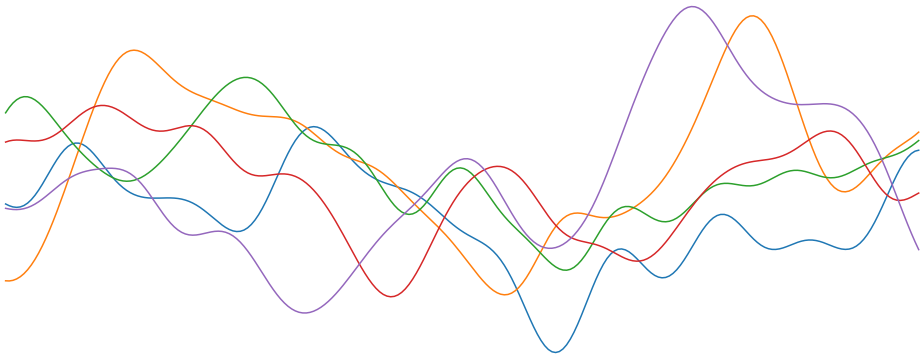


Figura 1.2: Procesos Gaussianos: SE Kernel

El principal objetivo de este trabajo es estudiar el posible potencial que posee el uso de los procesos gaussianos como predictor del precio de numerosas materias primas en el mercado, además de las ventajas que los mismos otorgan al inversor a la hora de hacer trading. Para conseguir estos objetivos, implementaremos un bot de trading fundamentado en los procesos gaussianos y estudiaremos el rendimiento de dicho bot en el mercado de materias primas realizando un backtesting (prueba de rendimiento), estudiando una serie de métricas, que definiremos más adelante y nos permitirán caracterizar la estrategia y fundamentar nuestras conclusiones.

En el siguiente capítulo introduciremos el estado del arte relativo a los procesos gaussianos dentro del mundo de las finanzas, y cómo se usan éstos para modelar y predecir el precio de determinados instrumentos financieros en sus respectivos mercados.

Capítulo 2

Estado del arte

Un proceso gaussiano es una generalización de la distribución gaussiana: representa una distribución de probabilidad sobre funciones que está completamente especificada por una media y una función de covarianza. Si dividimos una serie temporal de precios de un activo financiero en diversos segmentos, y entendemos cada segmento como una variable aleatoria, seremos capaces de caracterizar la distribución conjunta de todas las variables aleatorias, o subseries, y de pronosticar en base a esta distribución obtenida, acompañando a la predicción de una incertidumbre dada de nuevo por la distribución conjunta.

En [3] se pueden ver aplicadas las ideas que acabamos de comentar a diferentes series de precios (serie de precios del S&P 500 y acciones de las compañías Boeing y Starbucks). Estudiando el caso de la serie temporal de Starbucks, comprendida entre 2008 y 2016 (ambos incluidos), podemos ver cómo el autor divide la misma en segmentos anuales y trata las serie anuales normalizadas, con media cero y desviación típica la unidad, como variables de entrada independientes para el modelo. Además, añade al modelo el año, el trimestre y el día para otorgar a las series una característica que permita heredar conocimiento de series anteriores.

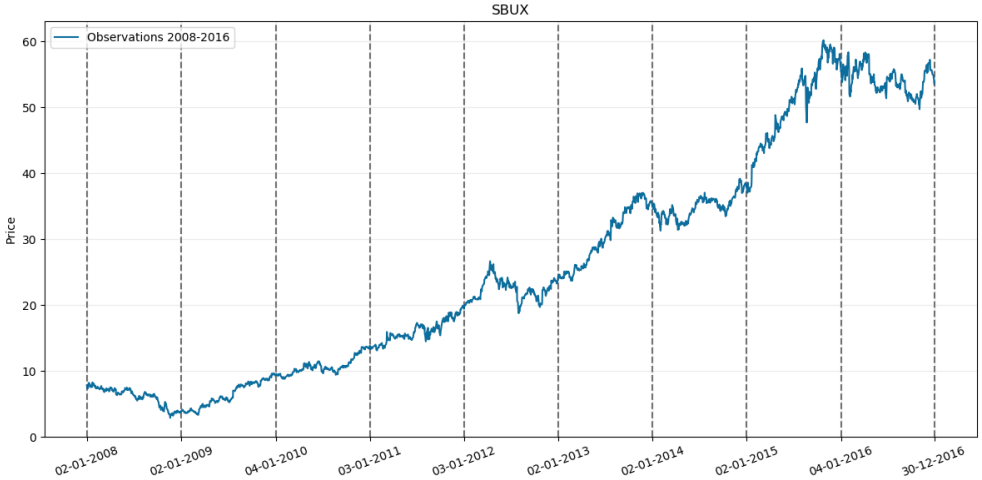


Figura 2.1: Serie de precios de Starbucks [3]

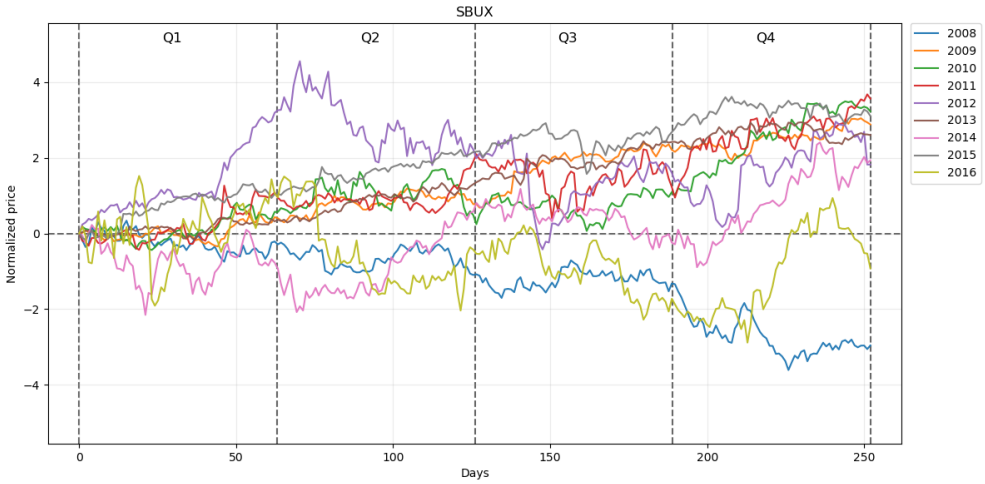


Figura 2.2: Series anuales de precios de Starbucks normalizadas [3]

De esta forma, el autor, con una función de covarianza o kernel diseñada para este problema, consigue dar la siguiente predicción para la acción de Starbucks a lo largo de 2017.

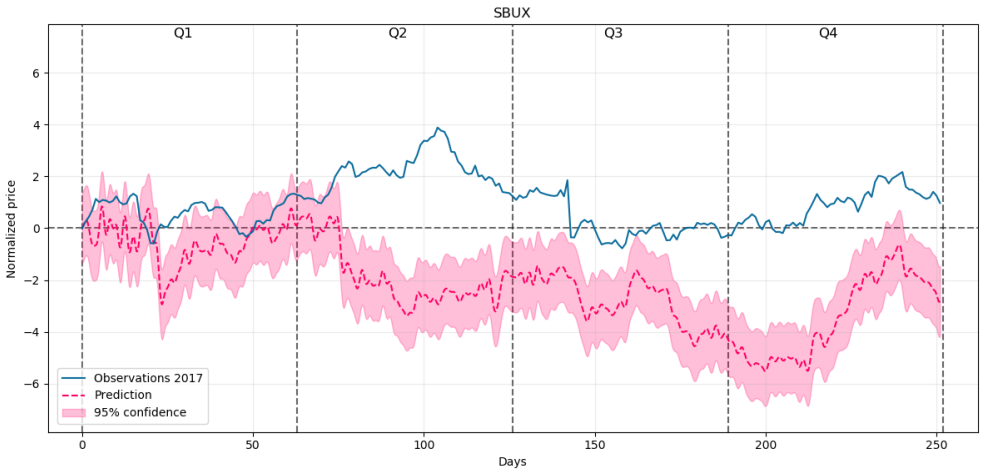


Figura 2.3: Predicción del precio de Starbucks en 2017 [3]

Como podemos contemplar en la imagen, aunque no sea similar a la curva medida, la predicción obtenida va acompañada de una banda de confianza. Esta característica hace de los procesos gaussianos un modelo muy atractivo para ser utilizado en este tipo de problemas de regresión, ya que, es muy difícil dar una predicción casi exacta del sistema, pero sí puedes estimar un intervalo en el que es probable que se encuentre dicho sistema a través de esta confianza.

El autor que acabamos de analizar ha elegido estudiar los procesos gaussianos sobre tres instrumentos financieros pertenecientes a la bolsa de Nueva York (dos acciones y un índice que representa las 500 acciones con mayor capitalización bursátil). En este trabajo, del mismo modo que lo hace el autor del segundo artículo que estudiaremos, aplicaremos los procesos gaussianos para predecir la cotización futura de diversos contratos futuros del mercado de materias primas.

Los mercados de materias primas o productos básicos (en inglés commodities) son los mercados mundiales, de carácter descentralizado, en los que se negocian productos no manufacturados y genéricos caracterizados por su bajo nivel de diferenciación. Existen en el mundo unos 50 mercados

organizados principales en los que se transmiten y cotizan este tipo de bienes. Los productos intercambiados en estos mercados pueden ser productos agrícolas, metales, energía o minerales, distinguiéndose entre bienes de carácter perecederos como son la mayoría de los productos agrícolas y no perecederos que son básicamente los productos minerales extraídos.

El siguiente artículo que vamos a comentar, [1], el cual tomaremos como base para este trabajo, emplea los procesos gaussianos para predecir el precio de las cotizaciones de distintos futuros sobre materias primas, como el algodón, gasolina, soja, trigo... La metodología es, fundamentalmente, igual que la de la publicación anterior: segmentamos la serie temporal completa en series anuales y tratamos cada serie como una función distinta. La diferencia en este caso es que, además de emplear una función de covarianza o kernel más complejo, el autor incluye más características para cada serie. Así, cada serie está caracterizada por el año, el día del año y otras variables exógenas, como las series de precios más parecidas a la estudiada dentro del mercado de materias primas o variables macroeconómicas.

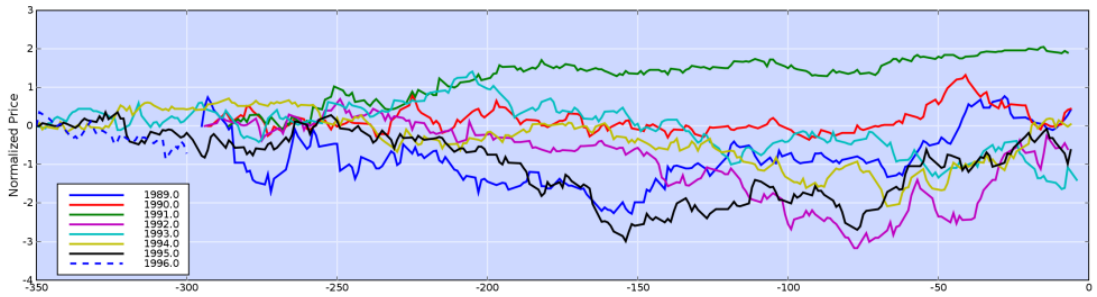


Figura 2.4: Series anuales de precios del Trigo normalizadas [1]

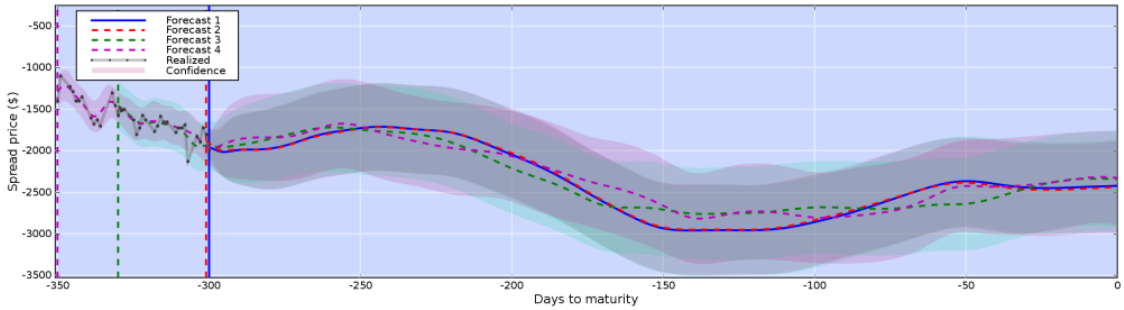


Figura 2.5: Predicción del precio del Trigo en 1996 [1]

También destacamos una pequeña diferencia a la hora de predecir el año deseado. En esta segunda publicación, el autor, conocidos los primeros dos meses del año, predice los diez restantes, dando así un "punto de partida" a la regresión en ese año, a diferencia del primero, que predecía el año en su totalidad.

Además, el autor diseña una estrategia para otorgar al algoritmo la capacidad de decisión de órdenes de compra y venta. El bot, una vez conocida la predicción, genera una señal de compra y una posterior señal de venta maximizando el ratio beneficio entre riesgo del periodo entrada-salida (el riesgo viene determinado por la confianza de la predicción y la duración de la operación).

Este último artículo concluye mostrando los rendimientos que se hubiesen obtenido de haber operado con dicho bot entre los años 1994 y 2007.

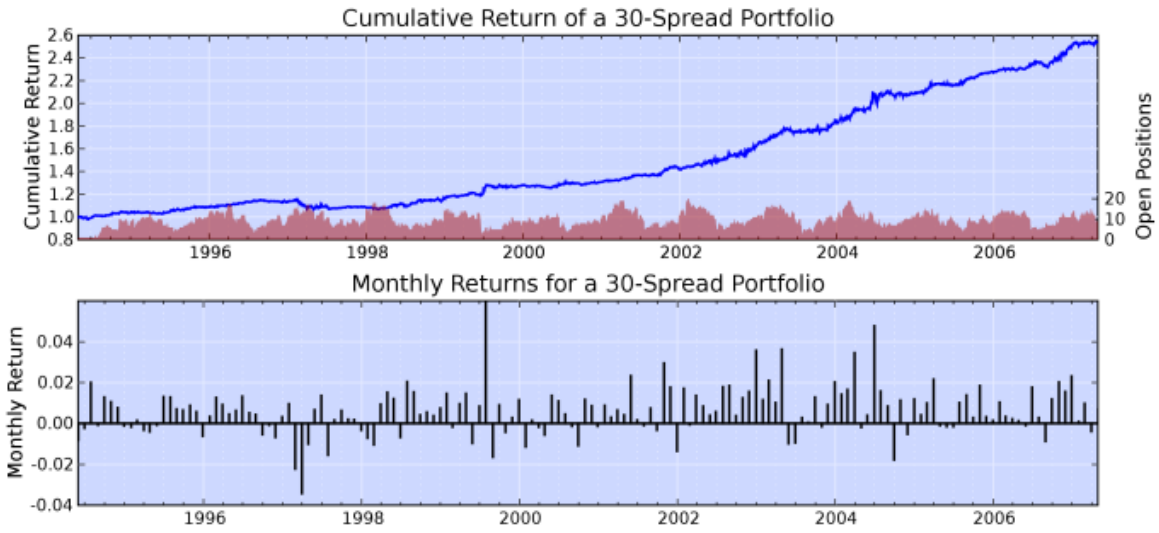


Figura 2.6: Rendimiento acumulado de la estrategia [1]

Como podemos observar la estrategia consigue multiplicar por casi 2,6 veces el capital invertido inicialmente en un período de 14 años, dando un retorno anualizado de entorno al 7 %, lo cual es un resultado muy bueno, ya que, no solo otorga un gran rendimiento del capital invertido, si no que consigue superar ampliamente al rendimiento que hubiese dado el propio mercado.

En este trabajo estudiaremos a fondo el modelo que hay detrás de los resultados anteriores así como la estrategia de compra y venta para, tomando el mismo como referencia y aportando nuestro propio enfoque tanto en las bases del modelo como en la estrategia de compra y venta, estudiar los resultados que puede llegar a obtener hoy en día un modelo de características similares al del autor y al nuestro. En el siguiente capítulo empezaremos introduciendo las matemáticas detrás de los procesos gaussianos como punto de partida de nuestro trabajo.

Capítulo 3

Procesos Gaussianos para regresión

Los procesos gaussianos, o GPs, ofrecen un enfoque probabilístico en el aprendizaje supervisado y pueden ser usados en problemas de clasificación y regresión. En los problemas de clasificación las variables predichas son las etiquetas de clase discretas que conforman el problema, mientras que para lo problemas de regresión los resultados son variables continuas. Como el problema que vamos a estudiar en este trabajo requiere de la predicción de una variable continua, el precio de cotización de determinadas materias primas en el mercado de materias primas, nos centraremos en el enfoque de los procesos gaussianos para regresión o GPR. Los conocimientos explicados en este capítulo tomarán como base el trabajo [2]. No profundizaremos en demostraciones de los conceptos explicados, para obtener un mayor detalle véase el mismo.

3.1. Definición

Hay varias formas de derivar modelos GPR a partir de otros. En este capítulo estudiaremos dos enfoques distintos, Weight-space y Function-space. La primera interpretación es desde el punto de vista del marco bayesiano para regresión, mientras que la segunda obtiene el GP definiendo una distribución sobre funciones, y realizando la inferencia directamente en el espacio de funciones.

3.1.1. Weight-space

Los modelos de regresión lineal simple han sido ampliamente estudiados y utilizados. Nuestra primera introducción para el GPR parte de la regresión lineal bayesiana y sigue con una mejora de este modelo proyectando las entradas en un espacio de características de dimensionalidad alta, utilizando el modelo lineal en dicho espacio.

3.1.1.1. Regresión lineal Bayesiana

El análisis bayesiano del modelo de regresión lineal estándar con ruido gaussiano está definido de la siguiente forma:

$$f(x) = x^T w, \qquad y = f(x) + \epsilon \tag{3.1}$$

donde x es el vector de entrada multidimensional, w es un vector de pesos de este modelo (el tamaño es igual a la dimensión de la entrada x), f es la función subyacente e y es el valor objetivo.

Suponemos que el ruido sigue una distribución normal independiente idénticamente distribuida con media cero y varianza σ_n^2:

$$\epsilon \sim \mathcal{N}(0, \sigma_n^2)$$

Dados los supuestos del modelo, la función de verosimilitud se obtiene mediante:

$$p(y|X,w) = \prod_{i=1}^{n} p(y_i|x_i,w) = \frac{1}{(2\pi\sigma_n^2)^{\frac{n}{2}}} e^{-\frac{1}{2\sigma_n^2}(y-X^Tw)^2} = \mathcal{N}(X^Tw, \sigma_n^2 I_n) \tag{3.2}$$

La inferencia en este modelo de regresión lineal se basa en la distribución posterior sobre los pesos, calculados por la regla de Bayes:

$$p(y|X,w) = \frac{p(y|X,w)p(w)}{p(y|X)} = \frac{p(y|X,w)p(w)}{\int p(y|X,w)p(w)dw} \propto p(y|X,w)p(w) \tag{3.3}$$

donde $\propto$ significa "proporcional a". Suponiendo que w sigue una distribución normal con media cero y matriz de covarianza Σ_p, $w \sim \mathcal{N}(0, \Sigma_p)$, la distribución de los pesos, dado el conjunto de entrenamiento $D = (X, y) = \{(x_i, y_i)|x_i \in \mathbb{R}^p, y_i \in \mathbb{R}, i = 1, \cdots, n\}$, se calcula de la siguiente forma:

$$p(w, D) \propto e^{-\frac{1}{2\sigma_n^2}(y-X^Tw)^T(y-X^Tw)} e^{-\frac{1}{2}w^T\Sigma_p^{-1}w} \propto e^{-\frac{1}{2}(w-\mu_w)^T\Sigma_w(w-\mu_w)} \tag{3.4}$$

donde $\mu_w = \frac{1}{\sigma_n^2}(\frac{1}{\sigma_n^2}XX^T + \Sigma_p^{-1})^{-1}Xy$, $\Sigma_w = (\frac{1}{\sigma_n^2}XX^T + \Sigma_p^{-1})^{-1}$. De hecho, la distribución de pesos es una distribución normal con media μ_w y matriz de covarianza Σ_w^{-1}.

$$p(w|D) \sim \mathcal{N}(\mu_w, \Sigma_w^{-1})$$

Una vez definido todo lo necesario, para hacer predicciones sobre los puntos de prueba, integramos sobre todos los parámetros posibles ponderados por su probabilidad posterior. La distribución predictiva para $f_* = f(z)$ viene dada por el promedio de la salida de todos los modelos lineales posibles con respecto a la distribución posterior:

$$p(f_*|z,D) = \int p(f_*|z,w)p(w|D)dw = \mathcal{N}(z^T\mu_w, z^T\Sigma_w z) \tag{3.5}$$

Según la ecuación (3.5), la distribución predictiva es de nuevo una normal, con media la media de los pesos de (3.5) multiplicada por la entrada de prueba. La varianza predictiva es una forma cuadrática de los puntos de prueba con la matriz de covarianza posterior, lo que significa que las incertidumbres predictivas aumentan con la magnitud de la entrada de prueba. Finalmente, considerando el ruido, σ_n^2 e I como la matriz identidad, y_* está definido por:

$$p(y_*|z,D) = \mathcal{N}(z^T\mu_w, z^T\Sigma_w z + \sigma_n^2 I) \tag{3.6}$$

3.1.1.2. Proyección de las dimensiones de entrada

El principal inconveniente del modelo de regresión lineal estándar es que la salida se limita a ser una combinación lineal de las entradas. Es decir, si la relación entre la entrada y la salida no puede ser aproximada por una función lineal, el rendimiento del modelo cae. Un enfoque sencillo para superar este problema es proyectar el espacio de entrada en un espacio de características de

mayor dimensión utilizando un conjunto de funciones y luego aplicar la regresión lineal bayesiana en este espacio. Ahora consideramos una función base de la forma $\phi(\|x - x_i\|)$, donde ϕ es una función no lineal y $\|x - x_i\|$ es la distancia del vector x al vector prototipo x_i. Para el caso de n puntos de entrenamiento la función de mapeo puede estar definida por:

$$f(x) = \sum_{i=1}^{n} w_i \phi(\|x - x_i\|) = \phi(x)^T w \tag{3.7}$$

donde $\phi(x) = [\phi(\|x - x_1\|), \cdots, \phi(\|x - x_n\|)]^T$. El modelo considerando el ruido queda definido por:

$$y = \phi(x)^T w + \epsilon \tag{3.8}$$

Definimos $\Phi(X)$ como la matriz de agregación de columnas $\phi(x)$ para todas las entradas del conjunto de entrenamiento:

$$\Phi(X) = \begin{bmatrix} \phi(\|x_1 - x_1\|) & \cdots & \phi(\|x_1 - x_n\|) \\ \vdots & \ddots & \vdots \\ \phi(\|x_n - x_1\|) & \cdots & \phi(\|x_n - x_n\|) \end{bmatrix} = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_n)^T \end{bmatrix}$$

El análisis de este modelo es similar al modelo lineal bayesiano estándar excepto que en todas partes X se sustituye por $\Phi = \Phi(X)$ y z por $\phi(z)$. Por lo tanto, la distribución predictiva de la ecuación (3.7) queda de la siguiente manera:

$$p(f_*|z, D) = \mathcal{N}(\phi_*^T \mu'_w, \phi_*^T \Sigma'_w \phi_*) \tag{3.9}$$

donde $\phi_* = \phi(z)$, $\mu'_w = \frac{1}{\sigma_n^2}(\frac{1}{\sigma_n^2}\Phi\Phi^T + \Sigma_p^{-1})^{-1}\Phi y$, $\Sigma'_w = (\frac{1}{\sigma_n^2}\Phi\Phi^T + \Sigma_p^{-1})^{-1}$. Además, la ecuación (3.9) puede expresarse como:

$$p(f_*|z, D) = \mathcal{N}(\phi_*^T \Sigma_p \Phi (K + \sigma_n^2 I)^{-1} y, \phi_*^T \Sigma_p \phi_* - \phi_*^T \Sigma_p \Phi (K + \sigma_n^2 I)^{-1} \Phi_*^T \Sigma_p \phi_*) \tag{3.10}$$

donde $K = \Phi_*^T \Sigma_p \Phi_*$. La equivalencia entre las ecuaciones (3.9) y (3.10) se muestra a continuación. Desde el punto de vista de la media:

$$\begin{aligned} \frac{1}{\sigma_n^2}\Phi(K + \sigma_n^2 I) &= \frac{1}{\sigma_n^2}\Phi(\Phi_*^T \Sigma_p \Phi_* + \sigma_n^2 I) \\ &= \frac{1}{\sigma_n^2}\Phi\Phi_*^T \Sigma_p \Phi_* + \Phi \\ &= (\frac{1}{\sigma_n^2}\Phi\Phi_*^T + \Sigma_p^{-1})\Sigma_p \Phi \end{aligned} \tag{3.11}$$

entonces $\frac{1}{\sigma_n^2}(\frac{1}{\sigma_n^2}\Phi\Phi_*^T + \Sigma_p^{-1})^{-1}\Phi = \Sigma_p\Phi(K + \sigma_n^2 I)^{-1}$ (asumiendo que todas las matrices son invertibles) y la ecuación queda:

$$\phi_*^T \mu'_w = \phi_*^T \frac{1}{\sigma_n^2}(\frac{1}{\sigma_n^2}\Phi\Phi_*^T + \Sigma_p^{-1})^{-1}\Phi y = \phi_*^T \Sigma_p \phi (K + \sigma_n^2 I)^{-1} y \tag{3.12}$$

Las entradas de estas matrices son de la forma:

$$k(x, x') = \phi(x)^T \Sigma_p \phi(x')$$

donde x y x' son los conjuntos de entrenamiento y prueba respectivamente. La función $k(\cdot, \cdot)$ se denomina kernel o función de covarianza. De hecho, sea $\varphi(x)$ una función de mapeo, el núcleo, o kernel, se puede reescribir en forma de espacio prehilbert.

$$\begin{aligned} k(x, x') &= \phi(x)^T \Sigma_p \phi(x') \\ &= \langle \varphi(x), \varphi(x') \rangle \end{aligned} \tag{3.13}$$

En particular, si consideramos el producto escalar como producto interno del espacio prehilbert, existe una sencilla expresión para $k(\cdot, \cdot)$ utilizando $\varphi(x)$. Como Σ_p es semi-definida positiva, podemos definir $\Sigma_p^{\frac{1}{2}}$ y por tanto $\Sigma_p = (\Sigma_p^{\frac{1}{2}})^2$. Según la descomposición en valores singulares, $\Sigma_p = UDU^T$ donde D es diagonal, entonces $\Sigma_p^{\frac{1}{2}} = UD^{\frac{1}{2}}U^T$. Por lo tanto, definiendo $\varphi(x) = \Sigma_p^{\frac{1}{2}}\phi(x)$, $k(\cdot, \cdot)$ se trata de un producto escalar $k(x, x') = \phi(x) \cdot \phi(x')$. Basándonos en las ecuaciones anteriores, podemos observar que el núcleo, o kernel, contiene toda la información de los vectores de características, por lo que es una alternativa muy conveniente para sustituir a los mismos.

3.1.2. Function-space

Un enfoque alternativo al visto anteriormente es obtener los mismos resultados obteniendo la función del modelo directamente. Como sabemos, un GP es una colección de variables aleatorias, y, por el Teorema central del límite, cualquier número finito de ellas tiene una distribución normal conjunta. Un proceso gaussiano está definido por la función media y función de covarianza, o kernel, es decir:

$$\begin{aligned} \mu(x) &= \mathbb{E}\left[f(x)\right] \\ k(x, x') &= cov(f(x), f(x')) \end{aligned}$$

Se puede denotar $f(x) \sim \mathcal{GP}(\mu, k)$. Por ejemplo, en el caso de la regresión lineal bayesiana, $f(x) = \phi(x)^T w$ con distribución a priori $w \sim \mathcal{N}(0, \Sigma_p)$, por lo que la media y función de covarianza resultan:

$$\begin{aligned} \mu(x) &= \mathbb{E}\left[f(x)\right] = \phi(x)^T \mathbb{E}\left[w\right] = 0 \\ k(x, x') &= \mathbb{E}\left[f(x)f(x')\right] = \phi(x)^T \mathbb{E}\left[ww^T\right] \phi(x') = \phi(x)^T \Sigma_p \phi(x') \end{aligned}$$

es decir, $f(x) \sim \mathcal{GP}(0, \phi(x)^T \Sigma_p \phi(x'))$.

Ahora consideremos un modelo de regresión general $y = f(x) + \epsilon$, donde $f(x) \sim \mathcal{GP}(\mu, k)$ y $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Dados n pares de observaciones, $\{(x_i, y_i)\}_{i=1}^n, x_i \in \mathbb{R}^p, y_i \in \mathbb{R}$ resulta que $[f(x_1), \cdots, f(x_n)]$ sigue una distribución normal multivariada:

$$[f(x_1), f(x_2), \cdots, f(x_n)]^T \sim \mathcal{N}(\mu, K)$$

donde $\mu = [\mu(x_1), \mu(x_2), \cdots, \mu(x_n)]^T$ es el vector de la media y K es la matriz de covarianza $n \times n$ donde $K_{ij} = k(x_i, x_j)$. Para predecir $f_* = f(Z)$ en los puntos de prueba $Z = [z_1, \cdots, z_m]^T$, la

distribución conjunta de las observaciones de entrenamiento y los objetivos de predicción f_* vienen dados por

$$\begin{bmatrix} y \\ f_* \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \mu(X) \\ \mu(Z) \end{bmatrix}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(Z,X)^T \\ K(Z,X) & K(Z,Z) \end{bmatrix}\right) \tag{3.14}$$

donde $\mu(X) = \mu$, $\mu(Z) = [\mu(z_1), \cdots, \mu(z_m)]^T$, $K(X,X) = K$, $K(Z,X)$ es una matriz $m \times n$ con $K(Z,X)_{ij} = k(z_i, x_j)$, y $K(Z,Z)$ es una matriz $m \times m$ con $K(Z,Z)_{ij} = k(z_i, z_j)$. Por lo tanto, la distribución predictiva resulta:

$$p(f_*|X, y, Z) = \mathcal{N}(\hat{\mu}, \hat{\Sigma}) \tag{3.15}$$

siendo

$$\begin{aligned} \hat{\mu} &= K(Z,X)^T(K(X,X) + \sigma_n^2 I)^{-1}(y - \mu(X)) + \mu(Z) \\ \hat{\Sigma} &= K(Z,Z) - K(Z,X)^T(K(X,X) + \sigma_n^2 I)^{-1}K(Z,X) \end{aligned} \tag{3.16}$$

Teniendo en cuenta la parte del ruido, la distribución predictiva de los objetivos y_* dado el conjunto de entrenamiento y las ubicaciones de prueba resultan:

$$p(f_*|X, y, Z) = \mathcal{N}(\hat{\mu}, \hat{\Sigma} + \sigma_n^2 I) \tag{3.17}$$

3.2. Kernels

Hasta ahora no hemos hablado de la forma que debe de tener la función de covarianza $k(\cdot, \cdot)$. Hacer una buena elección de esta función es importante para codificar adecuadamente el conocimiento previo sobre el problema a estudiar. Para obtener matrices de covarianza válidas, la función de covarianza debe ser, como mínimo, simétrica y semidefinida positiva, lo que implica que todos sus valores propios sean no negativos

$$\int k(x, x') f(x) f(x') d\mu(x) d\mu(x') \geq 0$$

para todas las funciones f definidas en el espacio apropiado y la medida μ.

Uno de los kernel, o función de covarianza, más empleado es el squared exponential kernel (SE), también conocido como kernel gaussiano o radial basis function kernel (RBF). Dicho kernel queda definido de la siguiente forma:

$$k_{SE}(x, x'; \sigma_l) = exp\left(-\frac{\|x - x'\|^2}{2\sigma_l^2}\right) \tag{3.18}$$

donde $\|x - x'\|^2$ es el cuadrado de la distancia euclídea entre dos vectores. El hiperparámetro σ_l rige la longitud de escala de características de la función de covarianza, indicando la anchura de la distribución subyacente.

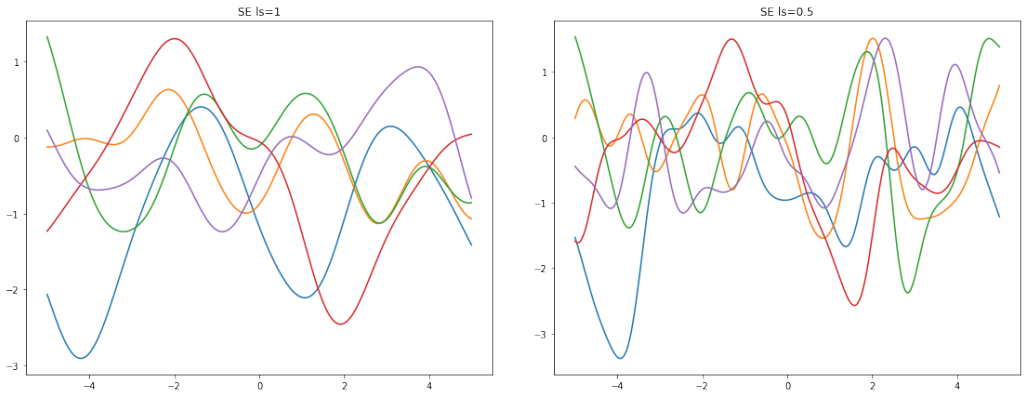


Figura 3.1: Squared Exponential Kernel

En la figura 3.1 podemos ver representadas diferentes funciones extraídas aleatoriamente del squared exponential kernel con lengthscales de 1 y 0,5 respectivamente. Para extraer dichas funciones se ha fijado siempre la misma semilla, por lo que son "similares" entre ellas. Podemos observar como cuanto mayor es el parámetro de longitud de escala, σ_l, mayor es la anchura de la normal que subyace en el kernel por lo que aumenta la dispersión global de la covarianza, es decir, más suaves se mueven las curvas resultantes.

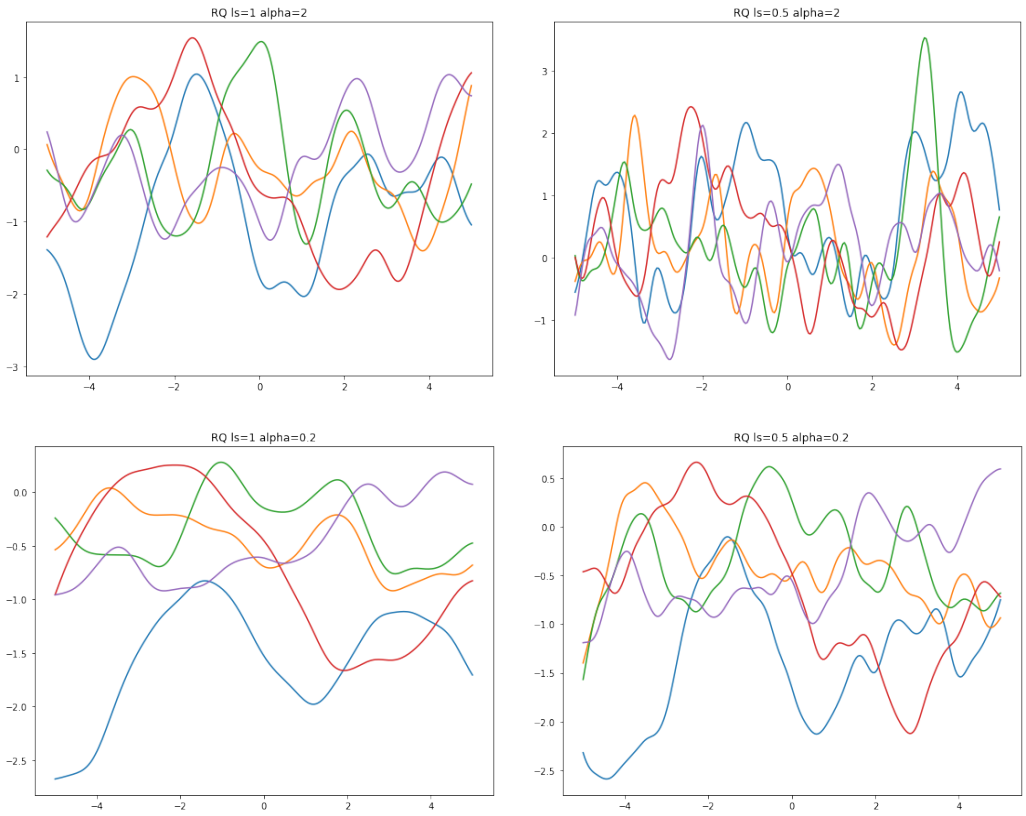


Figura 3.2: Rational Quadratic Kernel

Otro kernel muy usado en los procesos gaussianos es el Rational Quadratic Kernel, el cual está definido de la siguiente forma:

$$k_{RQ}(x, x'; \sigma_l, \alpha) = \left(1 + \frac{\|x - x'\|^2}{2\alpha\sigma_l^2}\right)^{-\alpha} \tag{3.19}$$

donde, de nuevo, $\|x - x'\|^2$ es el cuadrado de la distancia euclídea entre dos vectores, σ_l la longitud de escala de características, o lengthscale, y α el scale-mixture.

El kernel racional cuadrático, o Rational Quadratic Kernel, puede interpretarse como una suma infinita de diferentes kernels cuadráticos, o Squared Exponential Kernels, con diferentes escalas de longitud, con su respectiva ponderación entre las diferentes lengthscales. Cuando $\alpha \to 0$ el Rational Quadratic Kernel converge al Squared Exponential Kernels. En la figura 3.2 podemos observar que σ_l determina la anchura de la normal, es decir, la "amplitud" de las curvas, mientras que α determina el número de variaciones locales en las mismas, por lo que a mayor scale-mixture menos variaciones locales en las curvas, siempre respetando las tendencias marcadas por el lengthscale.

Una función de covarianza o kernel que también se deriva del kernel racional cuadrático es el periódico, o Periodic Kernel, el cual se utiliza para modelar funciones que presentan un patrón periódico. Un método efectivo es la introducción de warping, donde la variable de entrada de 1 dimensión x del kernel se mapea a otra de dimensión 2 para hacer una función periódica de la misma. Antes de definir el kernel periódico debemos destacar la siguiente propiedad:

$$\left(\sin\left(\frac{\pi}{p}x\right) - \sin\left(\frac{\pi}{p}x'\right)\right)^2 + \left(\cos\left(\frac{\pi}{p}x\right) - \cos\left(\frac{\pi}{p}x'\right)\right)^2 = 4\sin\left(\frac{\pi}{2p}(x - x')\right)^2$$

Teniendo en cuenta la misma, este kernel queda definido de la siguiente manera:

$$k_{PER}(x, x'; \sigma_l, p) = k_{SE}(w(x), w(x'); \sigma_l) = exp\left(-\frac{2\sin(\frac{\pi}{p}\|(x - x'\|))^2}{\sigma_l^2}\right) \tag{3.20}$$

donde $w(x) = [\sin(\frac{\pi}{p}x), \cos(\frac{\pi}{p}x)]^T$ y p el periodo.

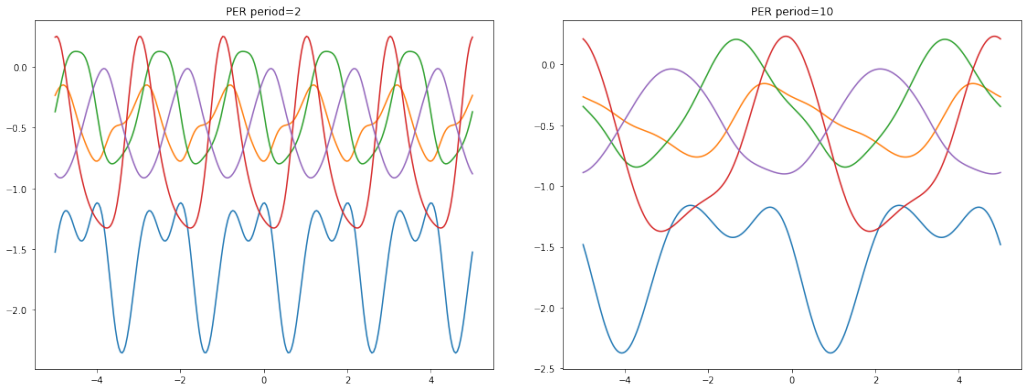


Figura 3.3: Periodic Kernel

En este kernel vemos como, además del lengthscale ya comentado en los anteriores, un hiperparámetro fundamental es el periodo. Dicho parámetro determina la frecuencia con la que la función que forma cada periodo se repite. Cuanto menor sea el parámetro, con mayor frecuencia se repetirá el periodo.

Además del Squared Exponential Kernel, Rational Quadratic Kernel y Periodic Kernel, existen

muchas otras funciones de covarianza o kernels muy empleados. Es el ejemplo del kernel lineal, exponencial, Matern 5/2, blanco, etc. No profundizaremos en estas funciones de covarianza o kernels ya sea por la trivialidad en su deducción o por el no interés de su estudio para este trabajo.

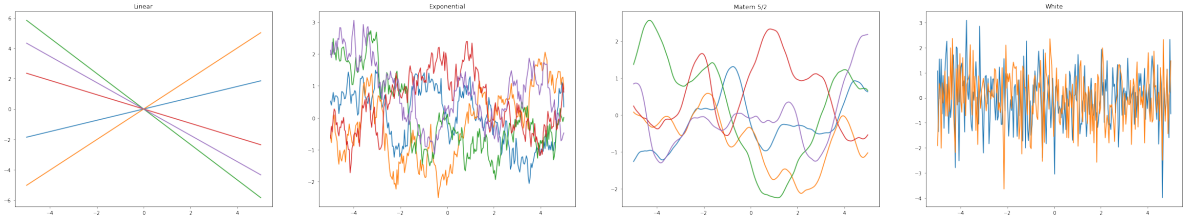


Figura 3.4: Kernels variados

3.2.1. Automatic Relevance Determination

Las funciones de covarianza, o kernels, introducidas en las ecuaciones (3.18), (3.19) y (3.20) se basan en la norma euclídea como medida de similitud entre dos vectores en el espacio de entrada. Estas funciones suponen que una escala de longitud σ_l determina la evaluación de dicha similitud en todas las dimensiones de entrada. Incluso después de normalizar las variables de entrada a la misma escala, la diferencia en la fuerza de predicción y el nivel de ruido entre las variables de entrada puede hacer que sea recomendable utilizar una escala de longitud para cada característica de entrada. Para ello, vamos a otorgar a cada dimensión del kernel un peso o lengthscale distinto. Esto equivale a hacer una simple reescritura de la norma euclídea en las funciones de covarianza introducidas anteriormente como normas ponderadas, con un peso específico asociado a cada dimensión de entrada.

En el caso del Squared Exponential Automatic Relevance Determination Kernel, éste quedaría definido de la siguiente forma:

$$k_{SEard}(x, x') = exp\left(-\frac{(x - x')^T \Theta^{-1}(x - x')}{2}\right) \tag{3.21}$$

donde Θ es una matriz diagonal formada por $\{\sigma_{li}^2\}_{i=1}^n$, que son las lenghtscales para cada dimensión de entrada correspondiente. Así mismo, el Rational Quadratic Automatic Relevance Determination Kernel queda definido por:

$$k_{RQard}(x, x'; \alpha) = \left(1 + \frac{(x - x')^T \Theta^{-1}(x - x')}{2\alpha}\right)^{-\alpha} \tag{3.22}$$

donde Θ es la matriz definida anteriormente y α el scale-mixture. Por último, el Periodic Automatic Relevance Determination Kernel queda definido por:

$$k_{PERard}(x, x') = exp\left(-2\sin(\frac{\pi}{p}(x - x'))^T \Theta^{-1} \sin(\frac{\pi}{p}(x - x'))\right) \tag{3.23}$$

donde $\sin(\frac{x}{p}) = [\sin(\frac{x_1}{p_1}), \cdots, \sin(\frac{x_n}{p_n})]$, $p = \{p_i\}_{i=i}^n$ son los periodos para cada una de las dimensiones del kernel y Θ la matriz anteriormente definida.

Gracias a estas nuevas definiciones de los kernels propuestos en las secciones anteriores somos capaces de dar una importancia distinta a cada dimensión de los mismos, solventando el problema de

sobreajuste que puede ocasionar el tener una o varias dimensiones en distintos ordenes de magnitud o distintas importancias en la predicción.

Los hiperparámetros óptimos de la función de covarianza, o kernel, se encuentran maximizando la probabilidad marginal, o marginal likelihood, de la distribución a priori del modelo en un proceso de entrenamiento (en la siguiente sección profundizaremos en ello). Después de la optimización de los hiperparámetros, las entradas que tienen poca importancia reciben una σ_{l_i} alta, disminuyendo así su importancia en la ponderación de la norma euclídea, y las entradas más importantes o relevantes reciben una σ_{l_i} baja, aumentando así su importancia en la ponderación de la norma euclídea.

3.3. Optimización de los hiperparámetros

En las secciones anteriores, hemos visto cómo construir un modelo de regresión de procesos gaussianos utilizando un núcleo dado y una función de media cero. La media y las varianzas pueden obtenerse siempre que todos los hiperparámetros indeterminados se aprendan a partir de los datos. Por el método bayesiano, necesitamos definir una distribución a priori sobre los hiperparámetros e integrar sobre ellos para hacer predicciones, es decir, tenemos que encontrar:

$$p(y_*|Z, D) = \int p(y_*|Z, \theta)p(\theta|D)d\theta \tag{3.24}$$

donde y_* es la suma de f_* y el ruido, y $\theta = \{\theta_1, \theta_2, \cdots\}$ contiene todos los hiperparámetros. Sin embargo, esta integral suele ser muy costosa de resolver analíticamente. Existen dos métodos para superar esta dificultad:

1. Aproximar la integral utilizando los valores más razonables para los hiperparámetros θ_R, es decir, $p(y_*|Z, D) = p(y_*|Z, D, \theta_R)$.

2. Realizar la integración sobre θ utilizando el método de Monte Carlo.

A pesar de que el métodos de Monte Carlo puede realizar la GPR sin necesidad de estimar hiperparámetros, el enfoque más común es aproximar la integral utilizando los valores más razonables para los hiperparámetros, debido al alto coste computacional de el método de Monte Carlo. El valor más razonable se interpreta como el valor más probable, por lo que el problema se resuelve por un método de estimación por máxima verosimilitud marginal. Para una regresión ruidosa, la función de probabilidad marginal $p(y|X, \theta)$ se representa como:

$$p(y|X, \theta) = \int p(y|f, X, \theta)p(f|X, \theta)df \tag{3.25}$$

En los modelos GPR, la distribución a priori y el likelihood son ambas distribuciones normales:

$$\begin{aligned} p(f|X, \theta) &= \mathcal{N}(0, K) \\ p(y|f, X, \theta) &= \mathcal{N}(f, \sigma_n^2 I) \end{aligned} \tag{3.26}$$

Dadas las dos ecuaciones descritas anteriormente, la probabilidad marginal, o marginal likelihood, también es una normal:

$$p(y|X,\theta) = \int \mathcal{N}(f, \sigma_n^2 I)\mathcal{N}(0, K)df = \mathcal{N}(0, K + \sigma_n^2 I) = \mathcal{N}(0, \Sigma_\theta) \tag{3.27}$$

donde $\Sigma_\theta = K_\theta + \sigma_n^2 I = K + \sigma_n^2 I$ con θ interviniendo en la matriz de covarianza K. En lugar de usar el marginal likelihood, es muy común emplear el negative log marginal likelihood (NLML) en su lugar:

$$\mathcal{L} = -\log(p(y|X,\theta)) = \frac{1}{2} y^T \Sigma_\theta^{-1} y + \frac{1}{2}\log(\det(\Sigma_\theta)) + \frac{n}{2}\log(2\pi) \tag{3.28}$$

Las derivadas parciales de NLML con respecto a los hiperparámetros vienen dadas por:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{2} tr(\Sigma_\theta^{-1} \frac{\partial \Sigma_\theta}{\partial \theta_i}) - \frac{1}{2} y^T \Sigma_\theta^{-1} \frac{\partial \Sigma_\theta}{\partial \theta_i} \Sigma_\theta^{-1} y \tag{3.29}$$

De hecho, los modelos GPR ruidosos también pueden considerarse como la regresión sin ruido con un núcleo ruidoso:

$$y = f, \qquad f \sim \mathcal{GP}(0, k')$$

donde $k' = k'(x_i, x_j) = k(x_i, x_j) + \delta_{ij}\sigma_n^2$ con $\delta_{ij} = 1$ si $i = j$, si no, $\delta_{ij} = 0$.

La probabilidad marginal se considera la distribución normal a priori que multiplica la identidad, la cual, por supuesto, sigue siendo gaussiana. Es la misma expresión que el modelo de regresión con ruido, ya que:

$$p(y|X,\theta) = p(f|X,\theta) = \mathcal{N}(0, K') = \mathcal{N}(0, K + \sigma_n^2 I) = \mathcal{N}(0, \Sigma_\theta) \tag{3.30}$$

donde $K' = K'(X, X) = K(X, X) + \sigma_n^2 I$. La tercera igualdad se debe a la definición del núcleo y de la matriz de covarianza. En este caso, la estimación de los hiperparámetros debe contener el nivel de ruido y entonces la NLML se reescribe como $\mathcal{L}(\theta, \sigma_n^2)$. Las derivadas parciales de la NLML con respecto a σ_n^2 vienen dadas por:

$$\begin{aligned} \frac{\partial}{\partial \theta_i}\mathcal{L}(\theta, \sigma_n^2) &= \frac{1}{2} tr(\Sigma_\theta^{-1} \frac{\partial \Sigma_\theta}{\partial \theta_i}) - \frac{1}{2} y^T \Sigma_\theta^{-1} \frac{\partial \Sigma_\theta}{\partial \theta_i} \Sigma_\theta^{-1} y \\ \frac{\partial}{\partial \sigma_n^2}\mathcal{L}(\theta, \sigma_n^2) &= \frac{1}{2} tr(\Sigma_\theta^{-1}) - \frac{1}{2} y^T \Sigma_\theta^{-1} \Sigma_\theta^{-1} y \end{aligned} \tag{3.31}$$

Aunque sea un método computacionalmente mucho menos costoso, no podemos ignorar las desventajas de la máxima verosimilitud marginal. En primer lugar, para muchos núcleos la función de probabilidad marginal no es convexa con respecto a los hiperparámetros, por lo que el algoritmo de optimización puede estancarse en un óptimo local y no en el global. En consecuencia, los hiperparámetros resultantes de la optimización por máxima verosimilitud y el rendimiento de la GPR pueden depender de los valores iniciales del algoritmo de optimización. Para enfrentarse a la sensibilidad del valor inicial, una estrategia común adoptada en la mayoría de casos en los que se usa GPR es repetir la optimización de los hiperparámetros utilizando varios valores iniciales generados aleatoriamente a partir de una simple distribución a priori. Las estimaciones finales de los hiperparámetros son las que tienen los mayores valores de verosimilitud después de la convergencia.

En segundo lugar, el coste computacional también es un problema. La evaluación del gradiente

del logaritmo de la probabilidad requiere el cálculo de la matriz inversa, que tiene un coste computacional asociado de orden $O(n^3)$ y, por lo tanto, el cálculo de los gradientes es una tarea que requiere mucho tiempo para grandes conjuntos de datos.

Además de Monte Carlo y estimación por máxima verosimilitud marginal, la validación cruzada puede ser considerado un método eficaz para seleccionar un núcleo y estimar los hiperparámetros del mismo, comparando varios modelos y eligiendo el que tiene el menor error. De hecho, se ha demostrado que la validación cruzada puede superar la estimación por máxima verosimilitud cuando el kernel está mal especificado. Pero hoy en día la estimación por máxima verosimilitud marginal sigue siendo el enfoque principal en la implementación de los modelos GPR y adoptaremos este método en lo que resta de trabajo.

Capítulo 4

Diseño de una estrategia

Una vez establecidas las bases sobre las que construir nuestro modelo para predecir y operar en el mercado de materias primas, teniendo en cuenta las características del mismo, diseñaremos una función de covarianza, o kernel, para nuestro modelo. Además dotaremos al mismo de una estrategia de compra y venta, la cual determinará el momento óptimo de entrada y salida del mercado en base a la predicción del GP. A continuación, estudiaremos brevemente las características del mercado de materias primas.

4.1. El mercado de materias primas

Los mercados de materias primas o productos básicos (en inglés commodities) son los mercados mundiales, de carácter descentralizado, en los que se negocian productos no manufacturados y genéricos caracterizados por su bajo nivel de diferenciación. En el mundo existen unos 50 mercados organizados principales en los que se transmiten y cotizan este tipo de bienes. Los más importantes son la Bolsa de Metales de Londres (LME), la Chicago Board of Trade (CBOT) y la New York Mercantile Exchange (NYMEX). Los productos intercambiados en estos mercados pueden ser productos agrícolas, metales, energía o minerales, distinguiéndose entre bienes de carácter perecederos como son la mayoría de los productos agrícolas y no perecederos que son básicamente los productos minerales extraídos.

Un derivado financiero es un instrumento financiero cuyo valor se deriva de una materia prima denominada subyacente. Los derivados se negocian en bolsa o en el mercado extrabursátil (OTC). Los derivados, como los contratos de futuros, los swaps (desde 1970), las materias primas negociadas en bolsa (desde 2003) y los contratos a plazo, se han convertido en los principales instrumentos de negociación en los mercados de materias primas. Los futuros se negocian en bolsas de materias primas reguladas.

Los contratos de futuros son la forma más antigua de invertir en materias primas. Los mercados de materias primas pueden incluir el comercio físico y el comercio de derivados utilizando los precios al contado, los contratos a plazo, los futuros y las opciones sobre futuros. Los agricultores han utilizado una forma sencilla de comercio de derivados en el mercado de materias primas durante siglos para la gestión del riesgo de los precios.

Trabajaremos con nuestro modelo sobre una lista de 16 contratos futuros, enumerados a continuación con sus respectivos Tickers (código único que hace referencia a los productos financieros

dentro de sus mercados):

- Oro (GC=F)
- Petróleo (CL=F)
- Gas natural (NG=F)
- Plata (SI=F)
- Gasolina (RB=F)
- Aceite industrial (HO=F)
- Platino (PL=F)
- Cobre (HG=F)
- Paladio (PA=F)
- Maíz (ZC=F)
- Arroz (ZR=F)
- Soja (ZS=F)
- Cacao (CC=F)
- Café (KC=F)
- Algodón (CT=F)
- Azucar (SB=F)

Para estudiar cada una de las series temporales y realizar predicciones sobre éstas necesitamos un dataset que contenga las mismas. Para ello, en primer lugar descargaremos cada una de las series temporales utilizando la API de Yahoo Finance a través del lenguaje de programación python3, lenguaje que usaremos a lo largo del trabajo para cada una de las implementaciones que hagamos.

Una API es un conjunto de procesos, funciones y métodos que brinda una determinada biblioteca de programación a modo de capa de abstracción para ser empleada por otro programa informático externo. Operar mediante API es una manera sencilla de conseguir datos de mercado en tiempo real, precios históricos y ejecutar operaciones, sin tener que rastrear de forma manual los mercados en busca de datos y precios. En lugar de esto, se recibe directamente la información del servidor, lo que asegura una gran eficiencia y rapidez.

Mediante la API antes mencionada conseguimos concentrar, en un único dataset, las 16 series de precios que estudiaremos comprendidas entre Enero de 2001 y Diciembre de 2021:

Date	Gold	Petroleum WTI	Natural Gas	Silver	RBOB Gasoline	Heating Oil	Platinum	Copper	Palladium	Corn Futures	Rough Rice Futures	Soybean Futures	Cocoa	Coffee	Cotton	Sugar
2001-01-02	268.399994	27.200001	8364	4545.0	7950	0.8658	609.900024	0.8180	968.500000	222.50	588.0	500.00	778.0	63.400002	60.349998	10.17
2001-01-03	268.000000	27.950001	8220	4490.0	8100	0.8598	619.400024	0.8235	988.400024	218.75	600.5	493.75	753.0	63.450001	60.959999	10.23
2001-01-04	267.299988	28.200001	9000	4532.0	8186	0.8612	626.900024	0.8150	982.000000	218.50	600.0	499.50	768.0	63.349998	59.610001	10.35
2001-01-05	268.000000	28.000000	9250	4555.0	8205	0.8625	620.700012	0.8320	986.000000	214.50	594.0	493.75	813.0	64.349998	59.419998	10.27
2001-01-08	268.000000	27.350000	9700	4545.0	8350	0.8260	634.500000	0.8330	1025.599976	216.50	594.0	488.00	813.0	63.849998	60.230000	10.47
2001-01-09	267.500000	27.719999	9800	4550.0	8290	0.8062	636.000000	0.8300	1034.400024	214.25	610.0	491.00	798.0	63.799999	59.869999	10.46
2001-01-10	264.700012	29.500000	9180	4535.0	8850	0.8499	641.099976	0.8310	1072.800049	214.50	608.0	490.50	809.0	63.349998	60.340000	10.29
2001-01-11	264.000000	29.500000	8710	4557.0	8885	0.8400	628.000000	0.8325	1045.250000	214.00	605.0	482.00	849.0	63.599998	60.869999	10.26
2001-01-12	263.899994	30.049999	8472	4622.0	9008	0.8421	635.099976	0.8445	1050.800049	216.25	598.0	479.25	860.0	65.000000	61.119999	10.00
2001-01-16	263.299988	30.350000	8103	4677.0	8790	0.8408	631.599976	0.8250	1028.349976	218.50	589.0	481.00	911.0	67.400002	60.959999	10.23
2001-01-17	263.200012	29.650000	6909	4710.0	8579	0.8150	625.500000	0.8315	999.299988	216.25	588.0	477.00	990.0	70.199997	60.520000	10.06
2001-01-18	264.200012	30.450001	7136	4794.0	8580	0.8415	609.599976	0.8670	1030.000000	218.75	589.0	473.50	996.0	66.750000	60.500000	9.87
2001-01-19	264.299988	32.099998	7480	4749.0	8850	0.8797	617.799988	0.8530	1062.500000	218.50	589.0	476.00	957.0	68.300003	60.630001	10.09
2001-01-22	266.399994	32.000000	7457	4774.0	8760	0.8790	621.799988	0.8495	1075.199951	214.50	592.0	469.50	1005.0	71.250000	59.360001	10.05
2001-01-23	266.100006	29.570000	6946	4776.0	8760	0.8622	620.799988	0.8510	1060.000000	216.50	595.5	477.50	1000.0	69.849998	59.279999	10.07
2001-01-24	264.299988	29.049999	7120	4768.0	8740	0.8222	605.599976	0.8400	1047.599976	214.25	580.5	468.00	989.0	70.050003	61.110001	10.05
2001-01-25	264.500000	29.360001	7270	4785.0	8790	0.8301	616.000000	0.8390	1075.000000	214.50	591.0	469.00	1004.0	67.849998	60.599998	10.03
2001-01-26	262.799988	29.780001	7280	4805.0	8850	0.8456	612.099976	0.8390	1079.000000	214.00	589.0	467.00	1056.0	67.099998	61.250000	10.00
2001-01-29	262.799988	29.059999	6620	4803.0	8770	0.8120	598.400024	0.8335	1054.000000	208.75	585.5	459.75	1045.0	65.400002	60.950001	9.74
2001-01-30	265.500000	29.059999	6097	4796.0	8800	0.8003	601.599976	0.8405	1058.750000	208.75	581.0	461.25	1056.0	64.199997	62.009998	9.58
2001-01-31	265.600006	28.700001	5650	4791.0	8600	0.7950	601.099976	0.8485	1054.800049	208.50	575.0	459.75	1020.0	63.750000	61.360001	9.95
2001-02-01	268.500000	29.799999	6400	4790.0	8501	0.7784	597.400024	0.8500	1053.599976	208.50	565.5	458.25	1051.0	64.699997	59.430000	10.25
2001-02-02	267.100006	31.250000	6743	4768.0	8910	0.8220	597.000000	0.8440	1055.150024	211.50	566.0	463.00	1048.0	63.200001	59.759998	10.14
2001-02-05	265.200012	30.500000	5706	4678.0	8858	0.8111	603.700012	0.8295	1082.800049	213.00	581.0	468.00	1065.0	62.700001	60.150002	10.00
2001-02-06	263.299988	30.350000	5764	4653.0	8870	0.8092	606.000000	0.8310	1076.500000	211.50	578.0	461.00	1086.0	62.200001	60.230000	10.01
2001-02-07	262.799988	31.200001	6250	4611.0	9225	0.8280	603.000000	0.8295	1072.000000	214.00	569.0	465.25	1044.0	61.750000	59.200001	9.89
2001-02-08	260.100006	31.559999	6170	4560.0	9290	0.8418	593.299988	0.8295	1041.900024	211.25	558.0	463.00	1005.0	62.000000	59.049999	9.88
2001-02-09	259.899994	31.049999	6210	4553.0	9012	0.8224	593.500000	0.8200	1010.000000	210.25	563.0	453.00	1012.0	62.650002	58.369999	9.80
2001-02-12	260.700012	30.549999	5810	4553.0	8920	0.8014	590.099976	0.8175	965.099976	209.75	564.0	456.50	1037.0	62.299999	58.900002	9.86

Figura 4.1: Dataset de precios de las 16 materias primas

Una vez tenemos las series de precios, las estudiaremos individualmente para tratar de modelar sus características en un kernel apropiado. Tomaremos como ejemplo la serie del Oro. Como los autores de [1] y [3], dividimos las series temporales completas anualmente y tratamos cada subserie como una función independiente.

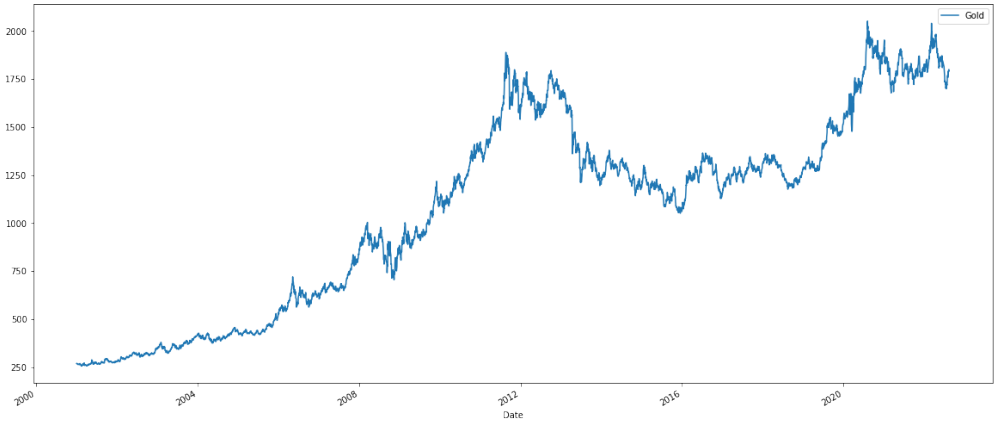


Figura 4.2: Series de precios del Oro entre 2001 y 2021

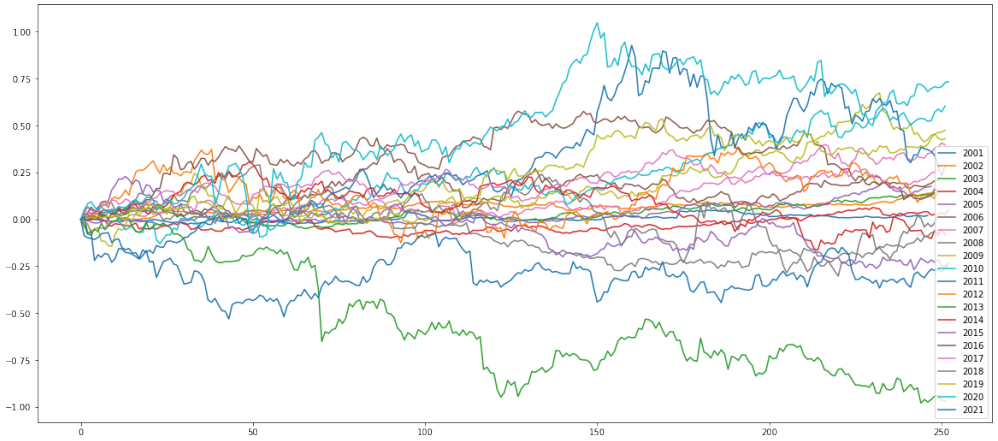


Figura 4.3: Series anuales de precios del Oro normalizadas

4.2. Diseño de un kernel

En esta sección desarrollaremos la función de covarianza, o kernel, que emplearemos en nuestro GP. Antes de comenzar este desarrollo, debemos de destacar una propiedad fundamental de los kernel, y es que como los kernel son espacios prehilbert, por propiedades de los mismos, la suma y producto de dos o más kernels sigue siendo un kernel. Para más detalle véase [2]. En la práctica, debido a los resultados de combinar las covarianzas de kernels, la suma de dos kernels puede interpretarse como una operación OR, y el producto como una operación AND.

En nuestras curvas de precios destacan dos propiedades: la no linealidad subyacente en las curvas y el ruido de las mismas. Nuestro kernel será un kernel compuesto. El kernel principal del mismo será el anteriormente estudiado Rational Quadratic Automatic Relevance Determination Kernel:

$$k_{RQard}(x, x'; \sigma, \alpha) = \left(1 + \frac{(x - x')^T \Theta^{-1} (x - x')}{2\alpha}\right)^{-\alpha} \tag{4.1}$$

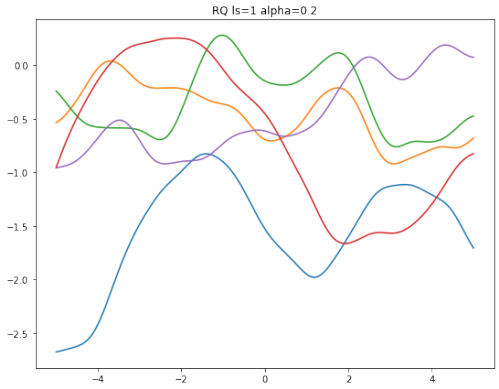


Figura 4.4: Rational Quadratic Kernel

Gracias a este kernel modelaremos la no linealidad que poseen las series de precios a predecir. Utilizaremos el Automatic Relevance Determination kernel para poder dar distinta importancia a cada una de las características de la matriz de entrada. Más adelante detallaremos la misma.

El segundo kernel que emplearemos en la construcción del kernel compuesto es el White Noise Kernel, o kernel de ruido blanco. Siendo σ^2 la varianza del ruido e I la matriz identidad, dicho kernel queda definido como:

$$k_{white}(x, x'; \sigma) = \sigma^2 I \tag{4.2}$$

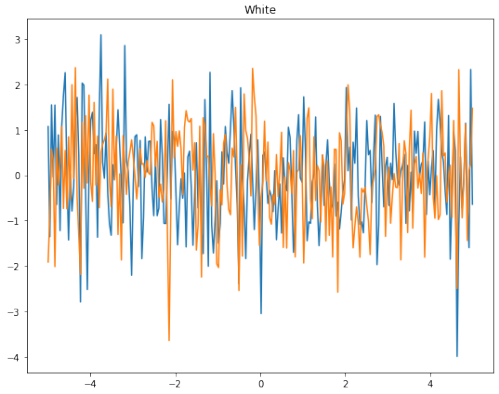


Figura 4.5: White Noise Kernel

Este segundo kernel le emplearemos en la modelización del ruido de nuestras series. Además, por características propias, este kernel solo se activa si se encuentra en dos puntos del espacio de entrada idénticos. Es por esta característica que emplearemos este kernel para modelizar también la relación de características interanuales, es decir, establecer relaciones entre instantes del mismo año (más adelante profundizaremos, pero el año será nuestra primera dimensión del espacio de entrada, por lo que, a diferencia del resto de kernels, éste solo actuará sobre la primera dimensión de dicho espacio).

Por último, el tercer kernel que conformará el kernel compuesto será el Constant Kernel, o kernel constante. Este es el kernel más sencillo que existe y, dada una constante α, está definido como:

$$k_{bias}(x, x'; \alpha) = \alpha \tag{4.3}$$

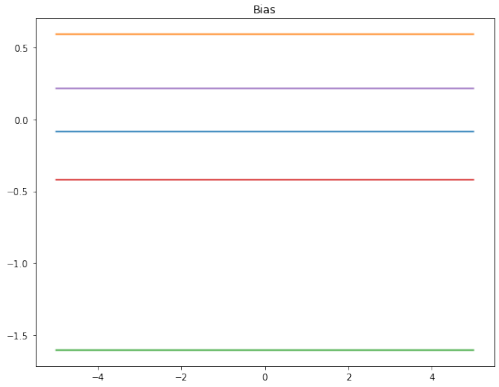


Figura 4.6: Constant Kernel

Este último kernel le dará una ponderación a cada uno de los dos kernels que moderan la no linealidad y el ruido.

El kernel que emplearemos en nuestro modelo será la suma del producto de un kernel constante por el Rational Quadratic Automatic Relevance Determination Kernel, modelando así la no linealidad de la curva, y un kernel constante por el kernel de ruido blanco, modelando las relaciones de puntos con la primera dimensión de entrada idénticos y el ruido de la misma. De esta forma, la función de covarianza que emplearemos a partir de ahora en este trabajo queda definida de la siguiente manera:

$$k(x, x'; \alpha, \sigma_r, \alpha_1, \alpha_2) = \alpha_1 \left(1 + \frac{(x - x')^T \Theta^{-1} (x - x')}{2\alpha}\right)^{-\alpha} + \alpha_2 \sigma_r^2 I \tag{4.4}$$

Destacar que el diseño anterior está fundamentado en las ideas expuestas en [1]. Una vez definido el kernel que usaremos para predecir nuestras curvas de precios, en la siguiente sección estudiaremos cómo obtendremos dichas predicciones.

4.3. Predicción del proceso gaussiano

A continuación detallaremos la metodología que hemos seguido para obtener las predicciones del precio de las materias primas a estudiar. Las ideas que fundamentan los siguientes conceptos han sido extraídas de [1], a las cuales les hemos añadido nuestras aportaciones.

Lo primero que debemos definir para realizar dichas predicciones es un conjunto de puntos de entrenamiento sobre los cuales calcular los hiperparámetros óptimos del kernel anteriormente definido. Este conjunto de entrenamiento estará formado por dos matrices, la matriz de características, X, y la matriz de valores de la serie, y.

Partiremos del caso más sencillo y a la vez más empleado en este tipo de problemas de regresión en procesos gaussianos. Como ya hemos comentado anteriormente, para cada materia prima a estudiar, dividiremos la serie de precios anualmente y caracterizaremos cada subserie normalizada

de forma distinta, por lo que, la matriz X estará formada por el año de la serie que conforma la matriz de precios y. De este modo, las matrices X e y para el entrenamiento del modelo quedarían definidas de la siguiente forma:

$$X = \begin{bmatrix} a_0 \\ \vdots \\ a_0 \\ a_1 \\ \vdots \\ a_1 \\ \vdots \\ a_n \\ \vdots \\ a_n \end{bmatrix} \quad y = \begin{bmatrix} p_{0_0} \\ \vdots \\ p_{0_m} \\ p_{1_0} \\ \vdots \\ p_{1_m} \\ \vdots \\ p_{n_0} \\ \vdots \\ p_{n_m} \end{bmatrix} \tag{4.5}$$

siendo $a_i,\ i = 1, \cdots, n$ los años que abarca la serie de precios en el conjunto de entrenamiento y $p_{i_j},\ i = 1 \cdots, n,\ j = 0 \cdots, m$ los precios de cotización de la materia prima objeto de estudio durante los n años que comprenden el conjunto de entrenamiento, $m + 1$ días por año.

Una vez entrenado el GP, para obtener la predicción para el año $n + 1$ bastaría con tomar como matriz de características $X = [a_{n+1}, \cdots, a_{n+1}]^T$ y tomar como predicción la salida y.

Aunque la matriz de características anteriormente definida sea suficiente para dar estimaciones del precio de las materias primas que queremos predecir, los resultados podrían mejorar caracterizando más nuestras series. El siguiente paso será dar una mejor caracterización temporal a dichas series. Para ello, introduciremos, además del año de la serie, el día que toma la serie dentro del año en el que se encuentra. De esta forma logramos establecer cierta relación entre mismos momentos temporales de distintos años (conseguimos ver comportamientos que pueden seguir las series, por ejemplo, con la entrada del verano, época en la que se reducen las lluvias y aumenta la temperatura, o al final del año, cierre del año comercial). Nuestra nueva matriz de características quedaría de la siguiente manera:

$$X = \begin{bmatrix} a_0 & d_0 \\ \vdots & \vdots \\ a_0 & d_m \\ a_1 & d_0 \\ \vdots & \vdots \\ a_1 & d_m \\ \vdots & \vdots \\ a_n & d_0 \\ \vdots & \vdots \\ a_n & d_m \end{bmatrix} \quad y = \begin{bmatrix} p_{0_0} \\ \vdots \\ p_{0_m} \\ p_{1_0} \\ \vdots \\ p_{1_m} \\ \vdots \\ p_{n_0} \\ \vdots \\ p_{n_m} \end{bmatrix} \tag{4.6}$$

donde a_i y p_{i_j} son los anteriormente definidos y $d_j,\ j = 0, \cdots, m$ son los días naturales del año en el que se encuentra la serie p_{i_j} (d_j tomará valores de 0 a, normalmente, 252, ya que son los días del

año que las bolsas donde se negocian las materias primas a estudiar están abiertas).

La nueva matriz de características es más completa que la que definimos inicialmente, pero en este trabajo vamos a ir un paso más allá, caracterizando la serie de precios con variables exógenas.

En el mercado hay una gran diversidad de materias primas, las cuales pueden interactuar entre ellas. No es de extrañar que existiese una fuerte correlación entre el precio del oro y el de la plata, o entre el precio del petróleo y la gasolina o el aceite usado en la industria, y que el precio del uno tenga influencia en el otro y viceversa. Por este motivo, introduciremos tres nuevas variables exógenas a nuestra matriz de características teniendo en cuenta estas posibles relaciones que existen en el mercado. Estas tres nuevas variables serán las tres series de precios más fuertemente correlacionadas con la serie objeto de estudio. De esta forma, la nueva matriz de características será:

$$X = \begin{bmatrix} a_0 & d_0 & p^1_{0_0} & p^2_{0_0} & p^3_{0_0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0 & d_m & p^1_{0_m} & p^2_{0_m} & p^3_{0_m} \\ a_1 & d_0 & p^1_{1_0} & p^2_{1_0} & p^3_{1_0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1 & d_m & p^1_{1_m} & p^2_{1_m} & p^3_{1_m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_n & d_0 & p^1_{n_0} & p^2_{n_0} & p^3_{n_0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_n & d_m & p^1_{n_m} & p^2_{n_m} & p^3_{n_m} \end{bmatrix} \quad y = \begin{bmatrix} p_{0_0} \\ \vdots \\ p_{0_m} \\ p_{1_0} \\ \vdots \\ p_{1_m} \\ \vdots \\ p_{n_0} \\ \vdots \\ p_{n_m} \end{bmatrix} \tag{4.7}$$

siendo a_i, d_j y p_{i_j} las anteriormente definidas y $p^k_{i_j},\ i = 1 \cdots, n,\ j = 0 \cdots, m,\ k = 1, 2, 3$ las series de precios de la k-ésima materia prima más correlacionada con la estudiada.

Dada una materia prima a predecir, para obtener la matriz de características anteriormente definida debemos establecer un método sistemático que seleccione las series de materias primas más relacionadas con ella misma. Para esto, tomaremos como las tres series más relacionadas las tres series con mayor coeficiente de correlación con la serie dada.

Para estudiar el coeficiente de correlación entre las series, primero debemos de estudiar la normalidad de las mismas. Para ello, utilizaremos el test de normalidad de Kolmogorov-Smirnov, incluido en el módulo de estadística de la librería de python3 Scipy. Una vez realizado el test para las 16 series de precios que estudiaremos en este trabajo, obtenemos un p-valor de cero para cada una de ellas, lo cual nos indica que las serias no se distribuyen bajo una normal. Por este motivo, nos decantaremos por estudiar la correlación de Spearman en lugar de la de Pearson, puesto que esta última solo se puede usar bajo la hipotesis de normalidad de las variables.

Una vez seleccionada la correlación que emplearemos, ya estamos en condiciones de calcular los coeficientes de correlación de todos los posibles pares de series. Para ello, mediante un bucle, de las 16 que estudiaremos, vamos recorriendo serie a serie calculando su coeficiente de correlación con el resto de series disponibles. Pero cabe destacar que podría ocurrir el siguiente suceso: dos series muestran un coeficiente de correlación alto pero no están relacionadas entre sí, es decir, hay dos series "falsamente correlacionadas". Para evitar este suceso, antes de calcular el coeficiente de correlación entre 2 series, realizaremos el test de contraste de hipótesis de Spearman, el cual toma como hipótesis nula y alternativa las siguientes:

$$H_0 : Corr = 0$$
$$H_1 : Corr \neq 0 \tag{4.8}$$

El objetivo es, dadas dos series de precios y un nivel de significación previamente fijado, $\alpha = 0{,}05$, realizar el test de correlación de Spearman y, cuando el p-valor arrojado por el mismo sea menor que α, calcular el valor absoluto del coeficiente de correlación de Spearman (tomaremos el valor absoluto para poder tomar las series con mayor correlación, independientemente de si la correlación es positiva o negativa), en caso contrario, tomaremos este último como cero. Tanto el p-valor resultante del test como el valor del coeficiente de correlación de Spearman los obtendremos mediante la función spearmanr incluida en el módulo de estadística de la librería de python3 Scipy. Por ende, representando en un mapa de calor los coeficientes de correlación obtenidos en valor absoluto resulta:

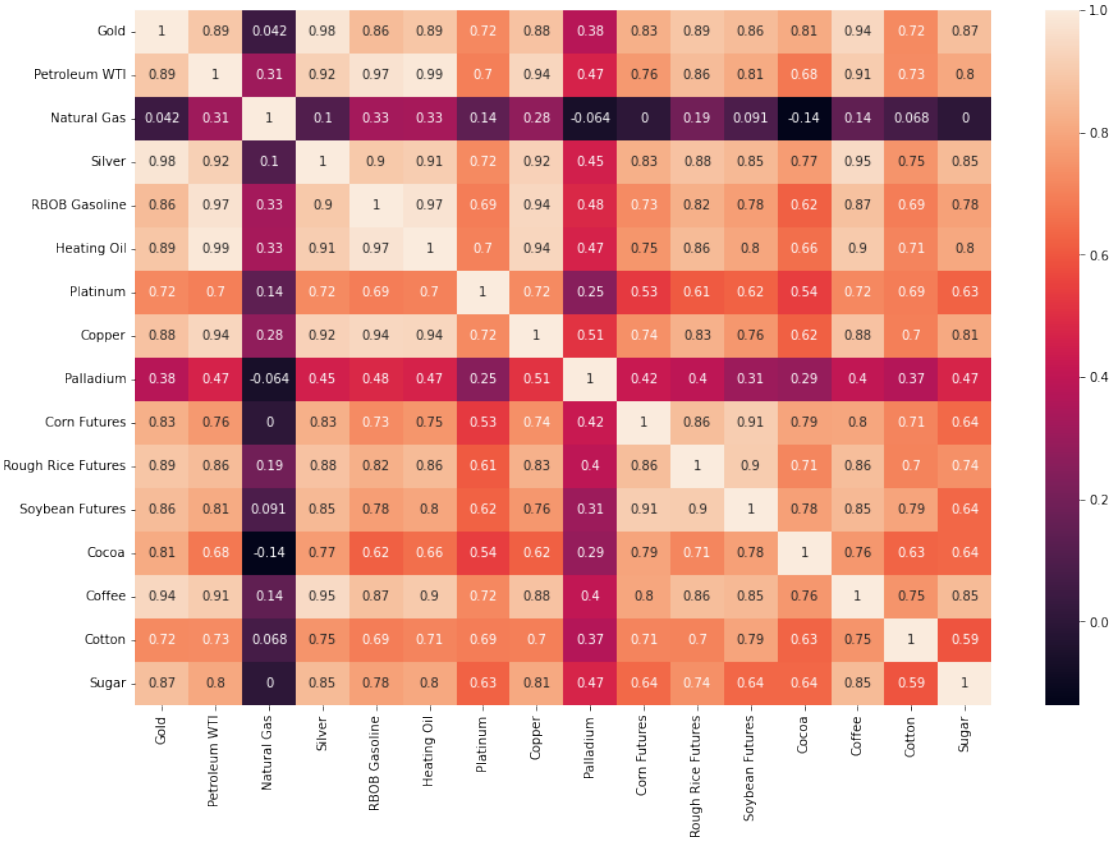


Figura 4.7: Coeficientes de correlación entre pares de series

De esta forma es sencillo seleccionar automáticamente las tres series más relacionadas con la que estemos estudiando para formar su matriz de características X. Así, por ejemplo, para el oro, la matriz X estará formada por el año, el día del año y los precios de las series de la plata, café y petróleo.

Esta nueva caracterización es realmente útil para predecir el precio de las materias primas, ya que el kernel aprenderá las relaciones no lineales que existen entre distintas materias primas a lo largo del tiempo. Pero con la matriz de características actualmente definida en (4.7) existe un

problema, y es que para predecir la serie de precios de la materia prima que estemos estudiando, podemos tomar el año como $n+1$ y los días desde 0 hasta m, pero no conocemos los valores futuros de las 3 series que estamos introduciendo en la matriz.

Para solventar este problema introducimos una nueva característica, la cual representará el momento en el cual realizamos la mediación respecto al momento que nos marca las características año y día. De esta forma, nuestra nueva matriz de características quedará redefinida, y para cada fila que teníamos anteriormente, tendremos m filas:

$$X = \begin{bmatrix} a_0 & d_0 & \Delta_0 & p^1_{0_0} & p^2_{0_0} & p^3_{0_0} \\ a_0 & d_0 & \Delta_1 & p^1_{0_1} & p^2_{0_1} & p^3_{0_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0 & d_0 & \Delta_m & p^1_{0_m} & p^2_{0_m} & p^3_{0_m} \\ a_0 & d_1 & \Delta_0 & p^1_{0_1} & p^2_{0_1} & p^3_{0_1} \\ a_0 & d_1 & \Delta_1 & p^1_{0_2} & p^2_{0_2} & p^3_{0_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0 & d_1 & \Delta_{m-1} & p^1_{0_{m-1}} & p^2_{0_{m-1}} & p^3_{0_{m-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_n & d_m & \Delta_0 & p^1_{n_m} & p^2_{n_m} & p^3_{n_m} \end{bmatrix} \quad y = \begin{bmatrix} p_{0_0} \\ p_{0_1} \\ \vdots \\ p_{0_m} \\ p_{0_1} \\ p_{0_2} \\ \vdots \\ p_{0_m} \\ \vdots \\ p_{n_m} \end{bmatrix} \tag{4.9}$$

Esta nueva matriz de características soluciona el problema de la predicción, puesto que bastaría con tomar la siguiente matriz para obtener los valores de y predichos:

$$X = \begin{bmatrix} a_n & d_m & \Delta_1 & p^1_{n_m} & p^2_{n_m} & p^3_{n_m} \\ a_n & d_m & \Delta_2 & p^1_{n_m} & p^2_{n_m} & p^3_{n_m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_n & d_m & \Delta_m & p^1_{n_m} & p^2_{n_m} & p^3_{n_m} \end{bmatrix} \tag{4.10}$$

Pero tomar la matriz X definida en (4.9) como matriz de entrenamiento para el GP es una tarea demasiado costosa de resolver debido al gran tamaño de la misma. Para lidiar con este problema realizamos un muestreo aleatorio simple sin reemplazamiento de la matriz total. El tamaño de la matriz resultante dependerá de lo preciso que deseemos hacer el entrenamiento. Después de realizar determinadas pruebas sobre el tamaño adecuado de la misma, hemos decidido obtener mil muestras aleatorias simples de nuestra matriz total, es decir, para cada una de las 6 características de entrada tendremos mil puntos sobre los que aprenderá nuestro proceso gaussiano. De esta forma entrenaremos con una matriz de dimensión 1000×6, la cual nos aporta un detalle bastante adecuado en las predicciones reduciendo drásticamente el coste computacional de obtener la misma.

Gracias a esta metodología de predicción conseguimos realizar predicciones como la siguiente:

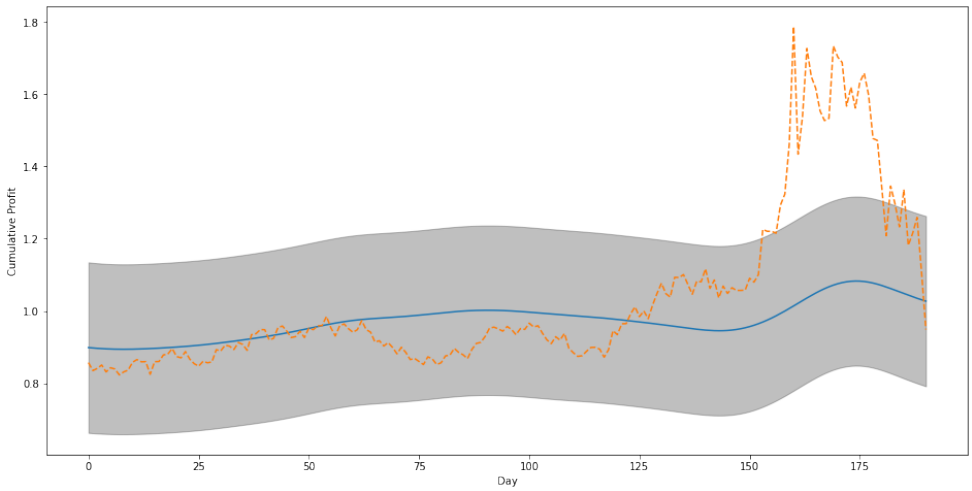


Figura 4.8: Predicción sobre el precio del Gas Natural en 2018

En la imagen anterior podemos ver la curva de precios del Gas Natural para el año 2018 representada por la línea discontinua naranja, y sobre ella, la curva predicha para estos mismos precios utilizando el GP que hemos definido, representada en azul. Esta curva es la función media del proceso gaussiano, y representado en gris tenemos las bandas de confianza que nos otorga el propio GP. Estas bandas nos dan un intervalo en el que, según el GP, es probable que se encuentre el punto predicho.

El hecho de que la curva media del proceso gaussiano, o línea azul, no se ajuste demasiado a la curva de precios reales, o naranja, y que simplemente se limite a describir la tendencia suavizada que sigue dicha curva es una de las características que hace atractivos a los procesos gaussianos para este trabajo.

En la siguiente sección propondremos un método para conceder al algoritmo poder de decisión sobre cuando comprar y vender en base a la media y bandas de confianza que nos proporciona el proceso gaussiano. Para no sobreajustar estas decisiones y devolver señales de compra y venta erróneas, será fundamental la "suavidad" que acabamos de describir en las curvas de predicciones.

4.4. Diseño de estrategia de compra y venta

En esta sección diseñaremos una estrategia para dotar al algoritmo que implementaremos, con base el proceso gaussiano descrito en anteriores secciones, de capacidad decisiva para operar de forma autónoma en el mercado. En primer lugar, el algoritmo, o bot, deberá, pasándole como entrada la curva histórica de precios y ciertos parámetros que más adelante definiremos, obtener la predicción (curva media y banda de confianza) mediante el proceso gaussiano previamente definido. En función de la forma que tenga esta curva, es decir, los precios que tomará la materia prima subyacente en el futuro, el bot decidirá la operación óptima, así como los correspondientes momentos de entrada y salida del mercado, maximizando el beneficio de la misma. Para minimizar el riesgo no permitiremos que el bot realice más de una operación al año, puesto que una mala predicción podría resultar en una pérdida de capital mucho mayor a la resultante de operar una única vez.

Supongamos que la predicción del precio de la materia prima en la que se encuentra operando el bot es la siguiente:

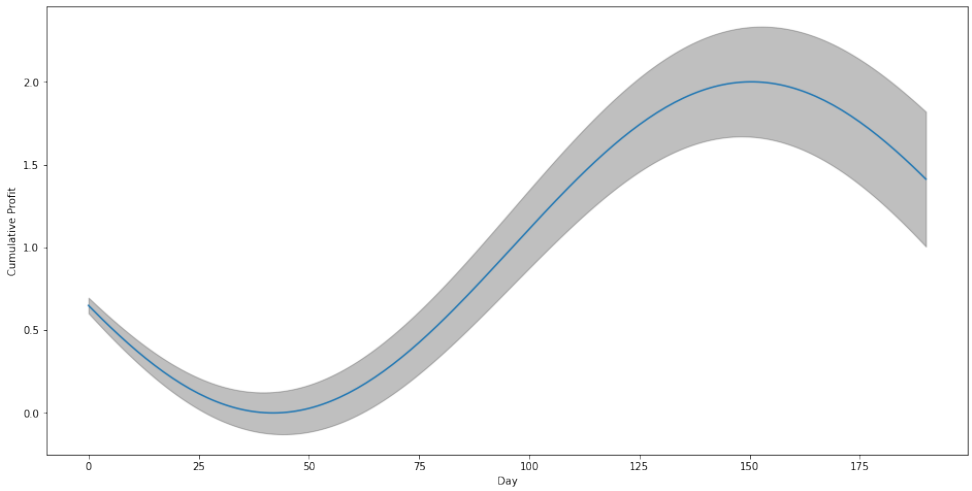


Figura 4.9: Predicción del GP

Nosotros como humanos observando esa curva podemos ver de forma clara que la operación óptima consistiría en entrar al mercado con una operación de compra, u operación en largo, un poco antes del instante 50, y salirnos del mismo con una operación de venta, u operación en corto, entorno al instante 150. Ahora, necesitamos un método programático mediante el cual nuestro bot sea capaz de, "observando" la predicción, determinar la operación óptima. Para esto, introduciremos el Market Score (MS), el cual definiremos como el rendimiento obtenido de la operación evaluada.

$$MS = \frac{1 + p_1}{1 + p_0} \tag{4.11}$$

donde p_0 es el precio de entrada de la operación y p_1 el de salida. Quedaría resolver el problema de maximización resultante de evaluar el Market Score de todas las posibles operaciones que el bot puede realizar. Tomando en el eje x el momento de compra y en el eje y el momento de venta, el problema de optimización se traduce en encontrar el máximo de la siguiente superficie, representada mediante curvas de nivel:

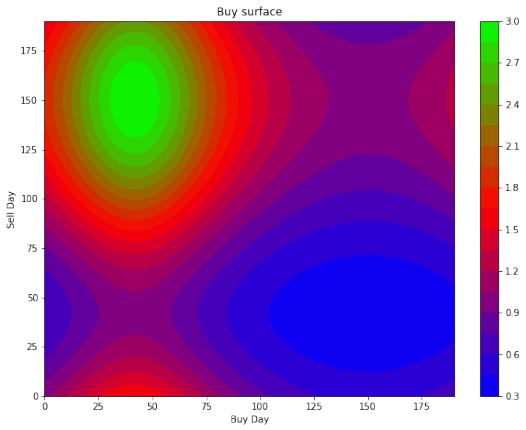


Figura 4.10: Market Score para operaciones de compra

En el caso anterior hemos buscado maximizar el beneficio de una operación de compra, es decir, entrar al mercado en largo y salir en corto, pero también existe la opción opuesta. Si la máxima variación de la curva predicha es negativa en lugar de positiva, es decir, los precios caen, como

inversores también podemos tomar ventaja entrando en el mercado con una operación en corto, es decir, vendiendo, y salir con una operación en largo, o comprando. Para que nuestro bot sea capaz de evaluar las dos posibles opciones, ir a favor del mercado o en contra, debemos definir, junto con el anterior, un nuevo Market Score para el caso de operaciones de venta:

$$\begin{aligned} MS_b &= \frac{1+p_1}{1+p_0} \\ MS_s &= \frac{1+p_0}{1+p_1} \end{aligned} \tag{4.12}$$

Este nuevo Market Score nos arrojará dos superficies, la resultante de evaluar todas las posibles operaciones de compra, 4.10, y la resultante de evaluar todas las posibles operaciones de venta:

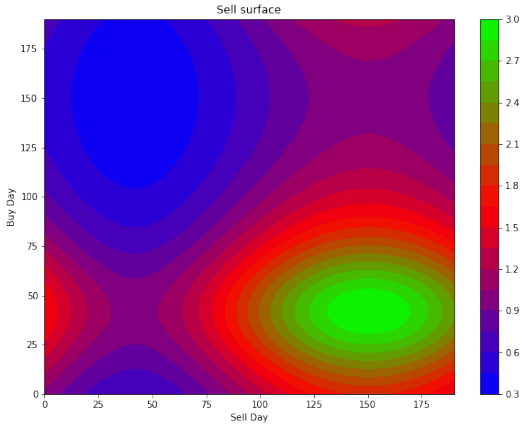


Figura 4.11: Market Score para operaciones de venta

En este punto el bot simplemente tendría que quedarse con las coordenadas, o instantes de entrada y salida del mercado, de la superficie para los cuales se obtenga el mayor beneficio. Pero debemos de introducir una pequeña restricción a la hora de buscar este máximo, y es que el máximo ha de buscarse en la parte superior de la superficie, es decir, donde $y > x$, o lo que es lo mismo, donde el momento de salida del mercado sea estrictamente mayor al momento de entrada. Ahora, el bot buscará el máximo entre las dos siguientes superficies:

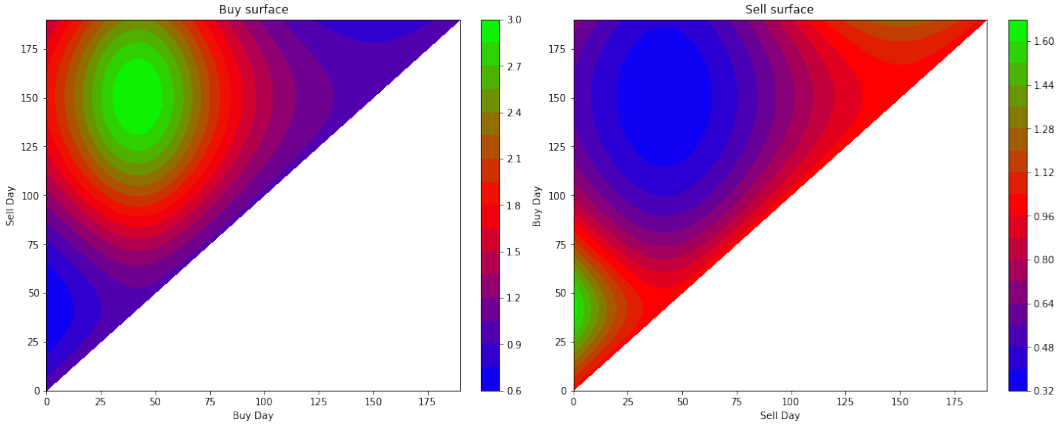


Figura 4.12: Market Score para operaciones de compra y venta

Gracias al Market Score definido, para la curva de predicción de 4.9, nuestro bot consigue determinar como óptima la siguiente operación de compra, definida en el intervalo verde:

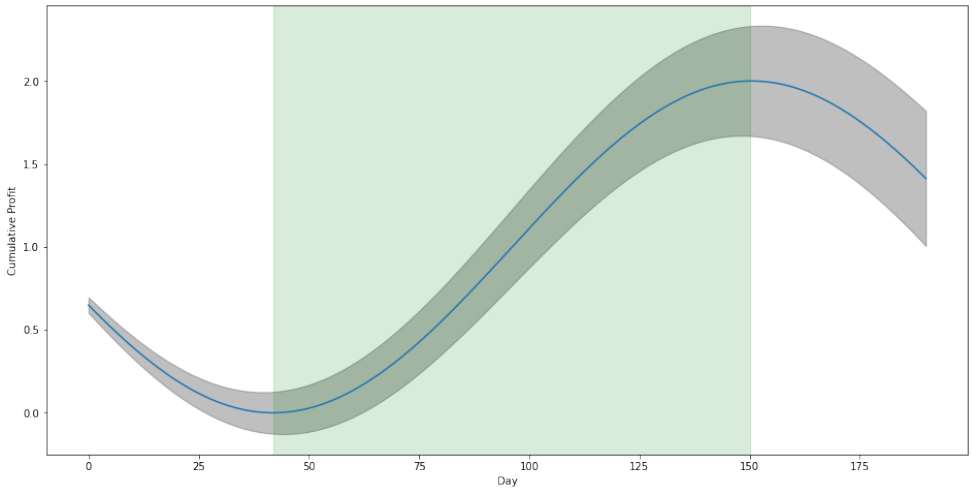


Figura 4.13: Operación con máximo Market Score

Como ya hemos mencionado anteriormente, en este trabajo no solo emplearemos la función media que nos proporciona el proceso gaussiano para determinar las operaciones a realizar en el mercado, si no que también basaremos las mismas en las bandas de confianza del GP. Estas bandas de confianza se pueden interpretar como el riesgo que asumimos al tomar esa predicción como valor futuro. Cuanto más ancha sea la banda de confianza en una zona de la curva media, más variabilidad debemos esperar de la predicción en dicha zona, dicho de otra forma, debemos confiar más en las predicciones donde la banda de confianza sea más estrecha. Además, otro factor de riesgo que existe en el mercado es el tiempo. Como inversores, entre dos operaciones con rendimientos parecidos, siempre escogeremos la operación con menos tiempo de duración, procurando que sea ésta la más cercana al momento en el que hemos obtenido la predicción, puesto que el mercado tendrá menos tiempo para reaccionar de forma opuesta.

Conociendo los dos factores de riesgo mencionados, la confianza en la predicción y el paso del tiempo, debemos de redefinir el Market Score, introduciendo estos dos nuevos conceptos. Para tener en cuenta la confianza del GP en el Market Score, dividiremos el rendimiento obtenido de las operaciones de compra y venta entre la varianza de la diferencia de tiempos. Así mismo, para proporcionar valores más altos a operaciones más cortas y cercanas en el tiempo, añadiremos un coeficiente que, cuanto más alejados sean los momentos de entrada y salida del mercado del inicio, menor será. De esta forma, el Market Score queda definido de la siguiente manera:

$$\begin{aligned} MS_b &= \frac{(1+p_1)/(1+p_0)}{Var(p_1-p_0)}\alpha^{t_1-t_0} \\ MS_s &= \frac{(1+p_0)/(1+p_1)}{Var(p_1-p_0)}\alpha^{t_1-t_0} \end{aligned} \tag{4.13}$$

donde p_0 y p_1 son los definidos anteriormente, t_0 y t_1 los instantes de tiempo de entrada y salida del mercado respectivamente y $Var(p_1 - p_0)$ puede redefinirse como $Var(p_0) + Var(p_1) - 2Cov(p_0, p_1)$, elementos situados en la matriz de covarianza del proceso gaussiano.

Además de estos últimos cambios sobre el Market Score, hemos decidido pasarle al bot, como parámetros de entrada, 2 nuevos valores, el Market Score mínimo para el cual toma como válida la operación resultante y la longitud temporal mínima que ha de tener la misma. Mediante el primero de los dos nuevos parámetros conseguimos evitar tomar posiciones que, en base a la curva predicha, tenga un beneficio potencial bajo y nos haga asumir el riesgo asociado a esa operación. El segundo parámetro puede ser un poco confuso, ya que en el párrafo anterior introducíamos un parámetro que daba preferencia a operaciones de poca duración, pero este nuevo parámetro evita que se escojan operaciones demasiado cortas, ya que puede darse el caso que una operación, ya sea por la confianza o por un parámetro de temporalidad inadecuado, resulte óptima con una duración de uno o dos días. En realidad, siguiendo el planteamiento de nuestra estrategia, la cual opera a medio plazo (operaciones entre aproximadamente 2 semanas y 6 meses), como inversores no tomaríamos posiciones un día para salirnos del mercado al siguiente o a los días. Para solventar esto, extrapolamos el concepto de buscar el Market Score óptimo por encima de la diagonal, y buscamos estos máximos cuando $y > x + d$ siendo d el número de días mínimo para operar. Tomando el coeficiente temporal $\alpha = 1 \cdot 10^{-3}$, el Market Score mínimo para operar como $min_{MS} = 1$ y el número de días mínimo para operar $d = 20$ (tomamos un valor bastante grande para ver el ejemplo, en la práctica este valor será entorno a 10, es decir, 2 semanas), resultan las siguientes superficies:

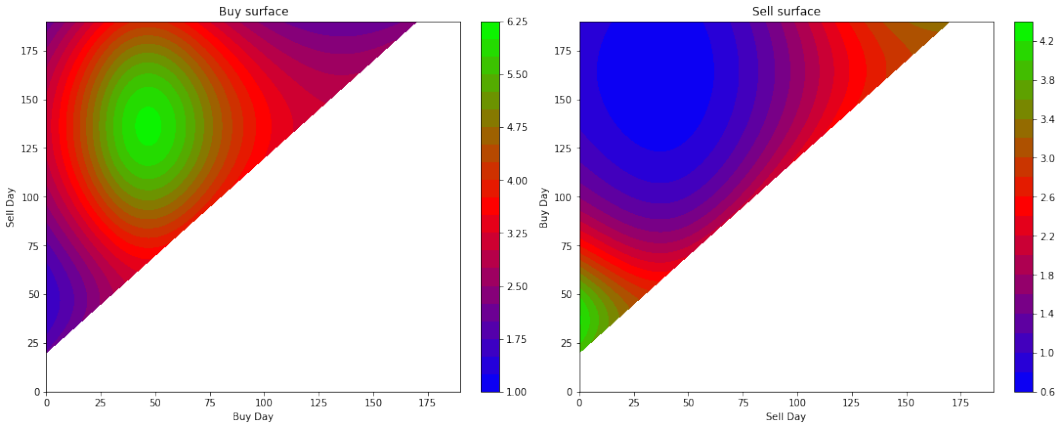


Figura 4.14: Operación con máximo Market Score y parámetros

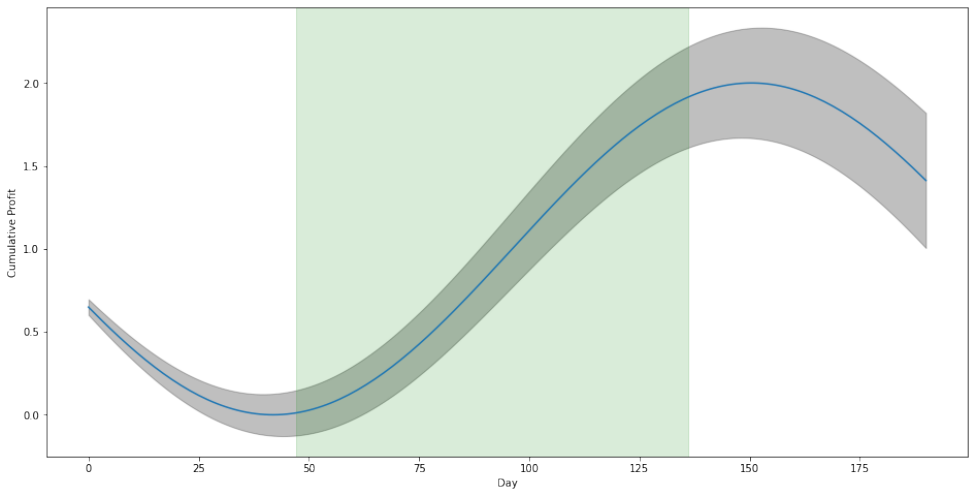


Figura 4.15: Operación con máximo Market Score y parámetros

Gracias a estos nuevos cambios, vemos como el bot es capaz de redefinir el intervalo de operación en el mercado. El punto de entrada en el mercado es el mismo, pero gracias a la combinación del factor riesgo, introducido por las bandas de confianza, con el factor temporal, el punto de salida se ha adelantado. Debido a que, cuanto más tiempo pasa, más ancha se va haciendo la banda de confianza, el bot busca adelantar el punto de salida, ya que, en las proximidades al máximo de la curva, aunque el precio suba, éste no crece al mismo ritmo que la banda de confianza se va ensanchando. A su vez, el factor α que hemos introducido penaliza este avance de tiempo, y ambos combinados encuentran unos diez días antes el momento de salida óptimo.

Para el caso de la figura 4.8, mostrada en la sección anterior, nuestro algoritmo determina el siguiente intervalo de compra como operación óptima, la cual ha resultado ser bastante rentable conociendo el precio real al que cotizaba el Gas Natural en 2018.

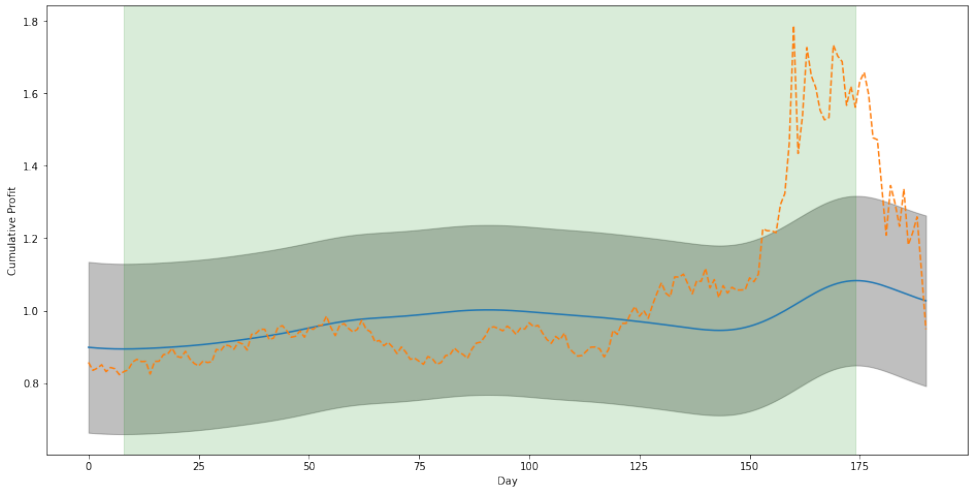


Figura 4.16: Predicción sobre el precio del Gas Natural en 2018

En este punto, ya hemos definido todo lo necesario para que el bot implementado empiece a operar en el mercado. En el siguiente y último capitulo estudiaremos los resultados que éste hubiese obtenido de operar en el mercado durante los últimos 11 años, es decir, desde 2011 hasta 2021, ambos incluidos.

Capítulo 5

Resultados

5.1. Backtesting

El backtesting es un tipo especial de validación cruzada que consiste en probar la relevancia de un modelo o una estrategia a partir de un gran conjunto de datos históricos reales [5]. Se puede aplicar a cualquier conjunto de datos, pero se usa con mayor frecuencia en ciencias sociales y ciencias naturales que producen datos medibles y requieren un enfoque estadístico. En finanzas, permite comprobar la validez y rentabilidad de una estrategia de inversión. La mayoría de las veces, la cantidad de datos necesarios hace que deban centralizarse en una base de datos, para que el proceso pueda automatizarse.

Al aplicar estas técnicas a los mercados de capitales, el backtesting determina el rendimiento de una estrategia financiera, si realmente se ha utilizado en períodos anteriores y en las mismas condiciones de mercado. Las pruebas retrospectivas con datos reales superan las pruebas realizadas en conjuntos de datos simulados artificialmente. Si bien el backtesting no puede predecir cómo se comportará una estrategia en condiciones futuras, su principal beneficio radica en comprender las vulnerabilidades de una estrategia a través de su aplicación a las condiciones reales encontradas del pasado. Como inversores, esto nos permite "aprender de los errores" sin tener que hacerlo con dinero real.

Un elemento clave del backtesting que lo diferencia de otras formas de pruebas históricas es que calcula el rendimiento si la estrategia se hubiera aplicado realmente en el pasado. Esto requiere que la prueba reproduzca las condiciones de mercado del momento en cuestión para obtener un resultado preciso. Los ejemplos de tales condiciones de mercado incluyen el seguimiento de precios, la compra y venta de acciones que ya no existen, o el uso de índices de mercado basados en su composición original, en lugar de su composición actual. Históricamente, estas pruebas han sido utilizadas por instituciones y administradores de fondos profesionales, debido al costo de adquirir estos conjuntos de datos o conjuntos de prueba. Sin embargo, con la llegada de las bolsas de valores a internet se pueden realizar pruebas de backtesting a diferentes tipos de estrategias del mercado de capitales sin coste alguno.

Para estudiar el comportamiento histórico de nuestra estrategia sobre la materia prima que deseemos estudiar (las siguientes explicaciones las realizaremos sobre el precio del oro) realizaremos una prueba de backtesting a la misma. Para ello, en primer lugar, descargamos los datos históricos del oro, materia prima a estudiar, y las 3 materias primas más fuertemente relacionadas con ella,

en este caso, plata, café y petróleo.

Date	High	Low	Open	Close	Volume	Adj Close
2001-01-02	268.399994	268.399994	268.399994	268.399994	0.0	268.399994
2001-01-03	268.000000	268.000000	268.000000	268.000000	1.0	268.000000
2001-01-04	267.299988	267.299988	267.299988	267.299988	1.0	267.299988
2001-01-05	268.000000	268.000000	268.000000	268.000000	0.0	268.000000
2001-01-08	268.000000	268.000000	268.000000	268.000000	0.0	268.000000
2001-01-09	267.500000	267.500000	267.500000	267.500000	0.0	267.500000
2001-01-10	264.700012	264.700012	264.700012	264.700012	0.0	264.700012
2001-01-11	264.000000	264.000000	264.000000	264.000000	0.0	264.000000
2001-01-12	263.899994	263.899994	263.899994	263.899994	0.0	263.899994
2001-01-16	263.299988	263.299988	263.299988	263.299988	0.0	263.299988
2001-01-17	263.200012	263.200012	263.200012	263.200012	0.0	263.200012
2001-01-18	264.200012	264.200012	264.200012	264.200012	0.0	264.200012
2001-01-19	264.299988	264.299988	264.299988	264.299988	0.0	264.299988
2001-01-22	266.399994	266.399994	266.399994	266.399994	0.0	266.399994
2001-01-23	266.100006	266.100006	266.100006	266.100006	0.0	266.100006
2001-01-24	264.299988	264.299988	264.299988	264.299988	0.0	264.299988
2001-01-25	264.500000	264.500000	264.500000	264.500000	0.0	264.500000
2001-01-26	262.799988	262.799988	262.799988	262.799988	0.0	262.799988
2001-01-29	262.799988	262.799988	262.799988	262.799988	15489.0	262.799988
2001-01-30	266.600006	262.899994	262.899994	265.500000	1754.0	265.500000
2001-01-31	266.000000	263.700012	265.200012	265.600006	566.0	265.600006

Figura 5.1: Dataset histórico del oro

Una vez tenemos los datasets históricos del oro y de las otras 3 materias primas, recorremos año a año obteniendo los instantes de entrada y salida del mercado óptimos según nuestra estrategia. Explicado más en detalle, nos situamos en un año (simulando que es el actual), pasamos como entrada los n años anteriores a nuestro bot (normalmente todos los que tengamos disponibles) y sobre estos calculamos la predicción del año actual. Una vez tenemos la curva de predicción con su respectiva banda de confianza, evaluamos los Market Score de las posibles operaciones de compra y venta, y nos quedamos con los instantes temporales en los que se da lugar la operación óptima. Una vez conocemos los momentos de entrada y salida del mercado, añadimos una nueva columna al dataset histórico, llamada Action. Esta columna representa, para cada día del año, la situación en la que nos encontramos: valores de 0 representan que estamos fuera del mercado, y por lo tanto, nuestro beneficio se mantiene estable, valores de 1 representan que nos encontramos en una operación de compra, por lo que nuestro beneficio crecerá junto con el mercado, y por último, valores de -1 representan que nos encontramos en una operación de venta, es decir, nuestro beneficio crece de forma inversa al mercado.

Date	High	Low	Open	Close	Volume	Adj Close	Action
2001-01-02	268.399994	268.399994	268.399994	268.399994	0.0	268.399994	0.0
2001-01-03	268.000000	268.000000	268.000000	268.000000	1.0	268.000000	0.0
2001-01-04	267.299988	267.299988	267.299988	267.299988	1.0	267.299988	0.0
2001-01-05	268.000000	268.000000	268.000000	268.000000	0.0	268.000000	1.0
2001-01-08	268.000000	268.000000	268.000000	268.000000	0.0	268.000000	1.0
2001-01-09	267.500000	267.500000	267.500000	267.500000	0.0	267.500000	1.0
2001-01-10	264.700012	264.700012	264.700012	264.700012	0.0	264.700012	1.0
2001-01-11	264.000000	264.000000	264.000000	264.000000	0.0	264.000000	1.0
2001-01-12	263.899994	263.899994	263.899994	263.899994	0.0	263.899994	1.0
2001-01-16	263.299988	263.299988	263.299988	263.299988	0.0	263.299988	1.0
2001-01-17	263.200012	263.200012	263.200012	263.200012	0.0	263.200012	1.0
2001-01-18	264.200012	264.200012	264.200012	264.200012	0.0	264.200012	1.0
2001-01-19	264.299988	264.299988	264.299988	264.299988	0.0	264.299988	1.0
2001-01-22	266.399994	266.399994	266.399994	266.399994	0.0	266.399994	1.0
2001-01-23	266.100006	266.100006	266.100006	266.100006	0.0	266.100006	1.0
2001-01-24	264.299988	264.299988	264.299988	264.299988	0.0	264.299988	1.0
2001-01-25	264.500000	264.500000	264.500000	264.500000	0.0	264.500000	1.0
2001-01-26	262.799988	262.799988	262.799988	262.799988	0.0	262.799988	1.0
2001-01-29	262.799988	262.799988	262.799988	262.799988	15489.0	262.799988	1.0
2001-01-30	266.600006	262.899994	262.899994	265.500000	1754.0	265.500000	1.0
2001-01-31	266.000000	263.700012	265.200012	265.600006	566.0	265.600006	0.0

Figura 5.2: Dataset histórico del oro con columna Action

El siguiente paso es calcular los incrementos diarios que sufre el mercado, en la columna Growth, y multiplicar estos últimos por el valor de la columna action de la fila anterior (columna Profit_Growth), ya que, si al cierre del mercado de un día decidimos hacer una operación de compra, ésta no se ejecuta hasta la apertura del mercado del día siguiente. De esta forma conseguimos, en los momentos que nos encontramos fuera del mercado tener un crecimiento nulo, en los momentos de compra crecer lo mismo que el mercado, y en los momentos de venta crecer

de forma inversa al mismo, es decir, si un día de compra el mercado crece positivamente, nosotros también, pero si en un día de venta el mercado decrece, nosotros crecemos positivamente. El último paso sería calcular los rendimientos acumulados del mercado y de nuestra estrategia para poder compararlos conjuntamente, los cuales son 1 más el producto acumulado de las columnas Growth y Profit_Growth respectivamente.

Date	High	Low	Open	Close	Volume	Adj Close	Action	Growth	Cumulative_Growth	Profit_Growth	Cumulative_Profit_Growth
2001-01-02	268.399994	268.399994	268.399994	268.399994	0.0	268.399994	0.0	0.000000	1.000000	0.000000	1.000000
2001-01-03	268.000000	268.000000	268.000000	268.000000	1.0	268.000000	0.0	-0.001490	0.998510	-0.000000	1.000000
2001-01-04	267.299988	267.299988	267.299988	267.299988	1.0	267.299988	0.0	-0.002612	0.995902	-0.000000	1.000000
2001-01-05	268.000000	268.000000	268.000000	268.000000	0.0	268.000000	1.0	0.002619	0.998510	0.000000	1.000000
2001-01-08	268.000000	268.000000	268.000000	268.000000	0.0	268.000000	1.0	0.000000	0.998510	0.000000	1.000000
2001-01-09	267.500000	267.500000	267.500000	267.500000	0.0	267.500000	1.0	-0.001866	0.996647	-0.001866	0.998134
2001-01-10	264.700012	264.700012	264.700012	264.700012	0.0	264.700012	1.0	-0.010467	0.986215	-0.010467	0.987687
2001-01-11	264.000000	264.000000	264.000000	264.000000	0.0	264.000000	1.0	-0.002645	0.983607	-0.002645	0.985075
2001-01-12	263.899994	263.899994	263.899994	263.899994	0.0	263.899994	1.0	-0.000379	0.983234	-0.000379	0.984701
2001-01-16	263.299988	263.299988	263.299988	263.299988	0.0	263.299988	1.0	-0.002274	0.980998	-0.002274	0.982463
2001-01-17	263.200012	263.200012	263.200012	263.200012	0.0	263.200012	1.0	-0.000380	0.980626	-0.000380	0.982090
2001-01-18	264.200012	264.200012	264.200012	264.200012	0.0	264.200012	1.0	0.003799	0.984352	0.003799	0.985821
2001-01-19	264.299988	264.299988	264.299988	264.299988	0.0	264.299988	1.0	0.000378	0.984724	0.000378	0.986194
2001-01-22	266.399994	266.399994	266.399994	266.399994	0.0	266.399994	1.0	0.007946	0.992548	0.007946	0.994030
2001-01-23	266.100006	266.100006	266.100006	266.100006	0.0	266.100006	1.0	-0.001126	0.991431	-0.001126	0.992910
2001-01-24	264.299988	264.299988	264.299988	264.299988	0.0	264.299988	1.0	-0.006764	0.984724	-0.006764	0.986194
2001-01-25	264.500000	264.500000	264.500000	264.500000	0.0	264.500000	1.0	0.000757	0.985469	0.000757	0.986940
2001-01-26	262.799988	262.799988	262.799988	262.799988	0.0	262.799988	1.0	-0.006427	0.979136	-0.006427	0.980597
2001-01-29	262.799988	262.799988	262.799988	262.799988	15489.0	262.799988	1.0	0.000000	0.979136	0.000000	0.980597
2001-01-30	266.600006	262.899994	262.899994	265.500000	1754.0	265.500000	1.0	0.010274	0.989195	0.010274	0.990672
2001-01-31	266.000000	263.700012	265.200012	265.600006	566.0	265.600006	0.0	0.000377	0.989568	0.000377	0.991045

Figura 5.3: Dataset histórico del oro con backtesting

En este trabajo no vamos a simular los cierres de operaciones de compra y venta mediante el posicionamiento de ordenes de Stop Loss (corte de pérdidas) y Take Profit (recogida de ganancias) dado que nuestra estrategia no hace uso de las mismas. El uso de estas órdenes depende del riesgo que esté dispuesto a asumir cada inversor, pero por el comportamiento histórico de las curvas de precios de las materias primas a estudiar y de las predicciones de la estrategia sobre éstas, asi como la limitación a una única operación por año, no hemos considerado necesario el uso de éste tipo de órdenes. Dada la gran competencia entre brokers y la aparición de brokers low cost el coste de operación en el mercado se ha reducido considerablemente con el tiempo, dando lugar a una gran variedad de precios y comisiones de operación. Por este motivo, y para facilitar el cálculo del rendimiento, tampoco simularemos estas comisiones, ya que cada inversor deberá de tener en cuenta los coste asociados al broker que use.

Una vez realizado el backtesting anterior, obtenemos una curva de rendimiento acumulado resultante de operar con nuestro bot sobre la cotización del oro. Pero para saber si una estrategia es apropiada para operar en tiempo real con ella hay que tener en cuenta muchos factores, no solo el rendimiento final que ésta obtiene. En este trabajo estudiaremos conjuntamente la curva de rendimiento acumulado del mercado y de nuestra estrategia, así como la curva de máximo Drawdown acumulado (es una forma de evaluar el riesgo del sistema de trading que mide el retroceso actual de la curva de rendimiento acumulado respecto a su máximo anterior). También estudiaremos una serie de métricas que enumeramos y definimos brevemente a continuación:

- Inicio: Fecha de inicio del backtesting.
- Fin: Fecha de finalización del backtesting.
- Duración: Número de días hábiles (sin incluir fines de semana y festivos) de duración del backtesting.
- Tiempo de exposición: Porcentaje del tiempo que se encuentra el sistema operando respecto del tiempo total de duración del backtesting.
- Operaciones totales: Número de operaciones totales realizadas.

- Operaciones de compra: Número de operaciones totales de compra realizadas.
- Ratio victorias operaciones de compra: Porcentaje de operaciones de compra con rendimiento positivo respecto del número total de operaciones de compra.
- Operaciones de venta: Número de operaciones totales de venta realizadas.
- Ratio victorias operaciones de venta: Porcentaje de operaciones de venta con rendimiento positivo respecto del número total de operaciones de venta.
- Beneficio: Rendimiento total obtenido en tanto por cien.
- Beneficio anual: Rendimiento total anualizado obtenido en tanto por cien.
- Factor beneficio: Ratio número de operaciones con rendimiento positivo entre el número de operaciones totales.
- Drawdown máximo: Máxima caída acumulada del beneficio obtenido respecto a su máximo anterior en tanto por cien.
- Drawdown medio: Caída acumulada media del beneficio obtenido respecto a su máximo anterior en tanto por cien.
- Factor de recuperación: Ratio beneficio total obtenido entre Drawdown máximo.
- Coeficiente de determinación R2: Proporción de la varianza total del rendimiento acumulado explicada por la regresión lineal del mismo.

Una vez explicada la metodología que seguiremos para realizar las pruebas de backtesting a nuestro sistema, y detalladas las curvas y métricas que estudiaremos, en la siguiente sección agruparemos los resultados de las pruebas para cada una de las materias primas en las que hemos operado, así como el backtesting resultante de operar en todas ellas simultáneamente.

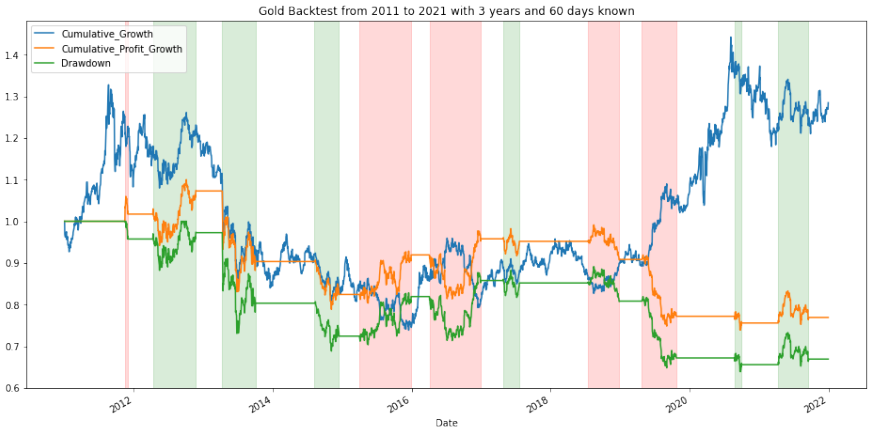
Comentar que no todas las materias primas se comportan igual, y que hay ciertas materias primas que tienen mucha más influencia en el mercado que otras. Dado el comportamiento histórico de cada una de ellas, hemos decidido optimizar la cartera de inversión y, manteniendo todas las materias primas estudiadas, ponderar cada una de éstas de forma distinta dentro de nuestra cartera. De esta forma conseguimos reducir ligeramente el riesgo resultante de operar en materias primas con cotizaciones muy volátiles o con comportamientos históricos bastante negativos.

5.2. Resultados obtenidos

A continuación se muestran los resultados que ha obtenido la estrategia diseñada en este trabajo desde enero de 2011 hasta diciembre de 2021. En cada una de las figuras vemos, a su izquierda, una tabla que recoge todos los valores que toman las métricas definidas en la sección anterior, y a su derecha, un gráfico que muestra las curvas de rendimientos del mercado y la estrategia aplicada sobre éste, así como la curva de Drawdown de la misma. En dicho gráfico están señalados en color verde los periodos en los que el bot ha decidido comprar y en rojo los periodos en los que ha decidido vender. Además, posteriormente presentaremos los resultados globales obtenidos de haber operado simultáneamente en el mercado con las 16 materias primas, ponderadas con sus respectivos pesos dentro de nuestra cartera. Sobre esta última prueba de backtesting estudiaremos los resultados obtenidos en detalle, analizando cada una de las métricas calculadas y observando el comprotamiento a lo largo del tiempo de las curvas de rendimientos acumulados y Drawdown.

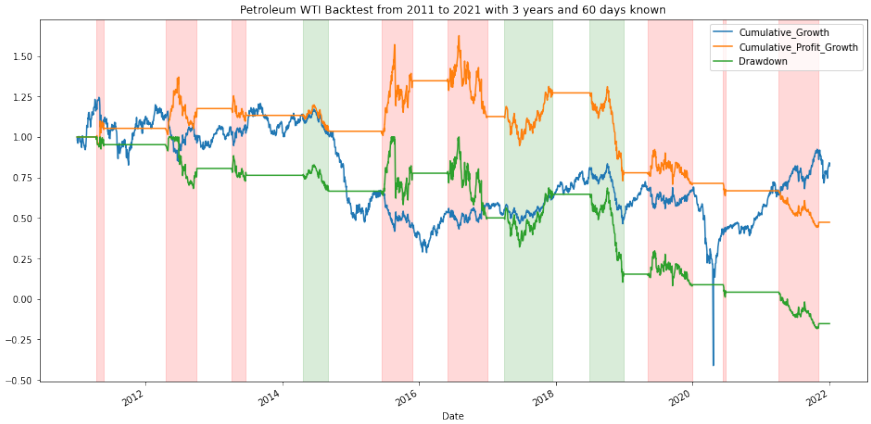
Oro

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	43.26 %
Op. totales	11
Op. compra	6
Ratio op. compra	33.33 %
Op. venta	5
Ratio op. venta	60.00 %
Beneficio	-23.09 %
Beneficio anual	-2.36 %
Factor beneficio	0.95
Drawdown max.	36.12 %
Drawdown med.	18.96 %
Factor recuperación	0.57
Coef. R2	0.53



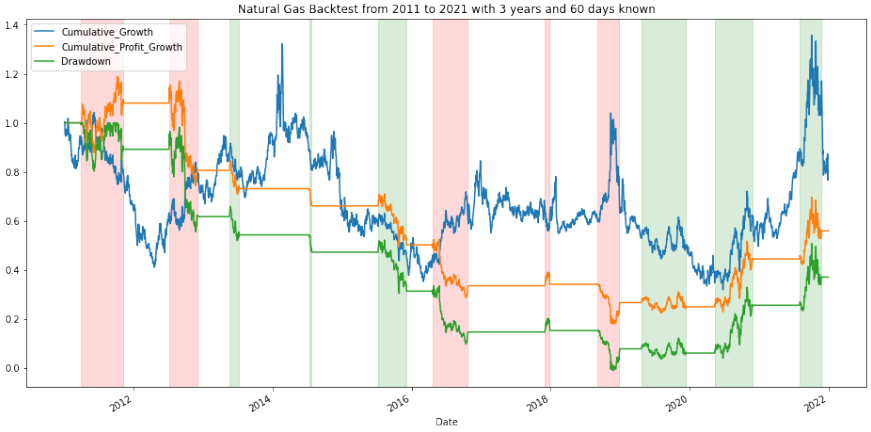
Petróleo

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	43.03 %
Op. totales	11
Op. compra	3
Ratio op. compra	33.33 %
Op. venta	8
Ratio op. venta	37.50 %
Beneficio	-52.64 %
Beneficio anual	-6.57 %
Factor beneficio	0.95
Drawdown max.	118.35 %
Drawdown med.	45.05 %
Factor recuperación	0.22
Coef. R2	0.36



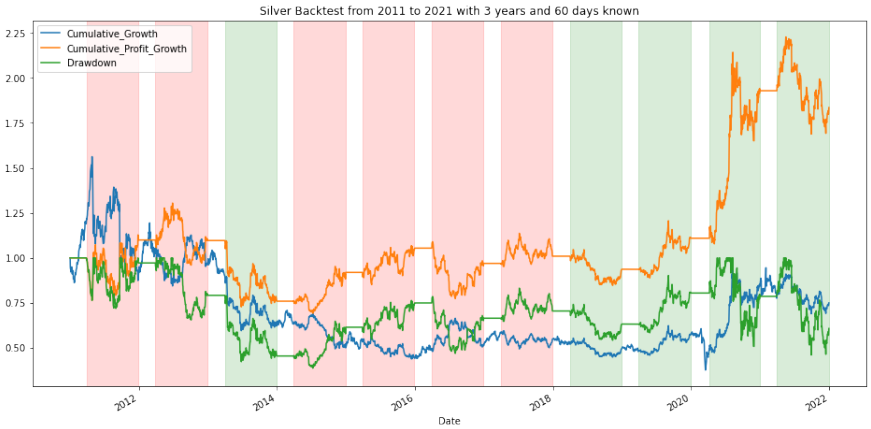
Gas natural

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	36.92 %
Op. totales	11
Op. compra	6
Ratio op. compra	33.33 %
Op. venta	5
Ratio op. venta	40.00 %
Beneficio	-44.08 %
Beneficio anual	-5.15 %
Factor beneficio	1
Drawdown max.	101.09 %
Drawdown med.	60.94 %
Factor recuperación	0.28
Coef. R2	0.72



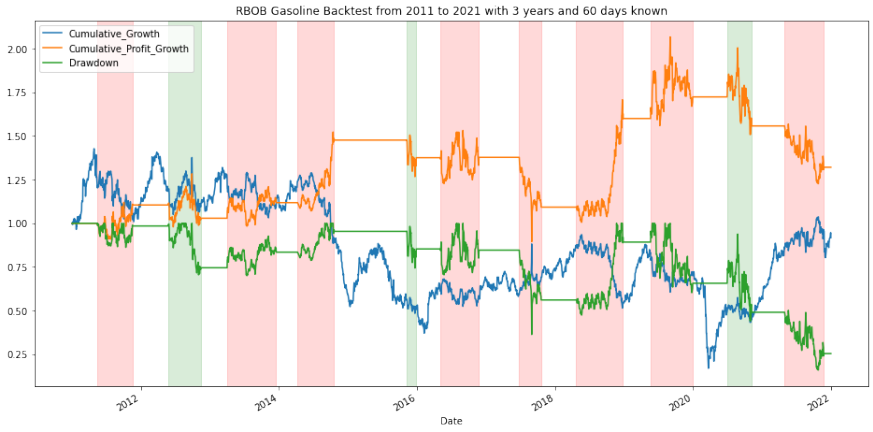
Plata

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	75.49 %
Op. totales	11
Op. compra	5
Ratio op. compra	40.00 %
Op. venta	6
Ratio op. venta	83.33 %
Beneficio	83.48 %
Beneficio anual	5.67 %
Factor beneficio	1.08
Drawdown max.	61.21 %
Drawdown med.	29.36 %
Factor recuperación	1.14
Coef. R2	0.35



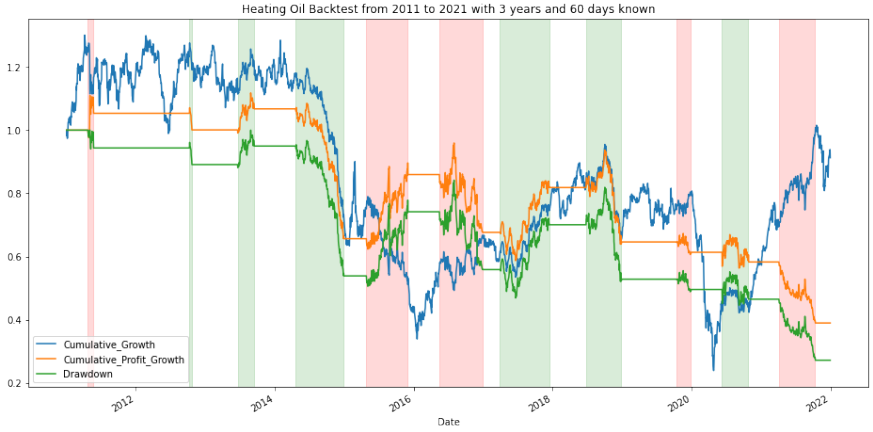
Gasolina

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	50.16 %
Op. totales	11
Op. compra	3
Ratio op. compra	0.00 %
Op. venta	8
Ratio op. venta	75.00 %
Beneficio	32.11 %
Beneficio anual	2.56 %
Factor beneficio	1.06
Drawdown max.	84.03 %
Drawdown med.	22.34 %
Factor recuperación	0.72
Coef. R2	0.50



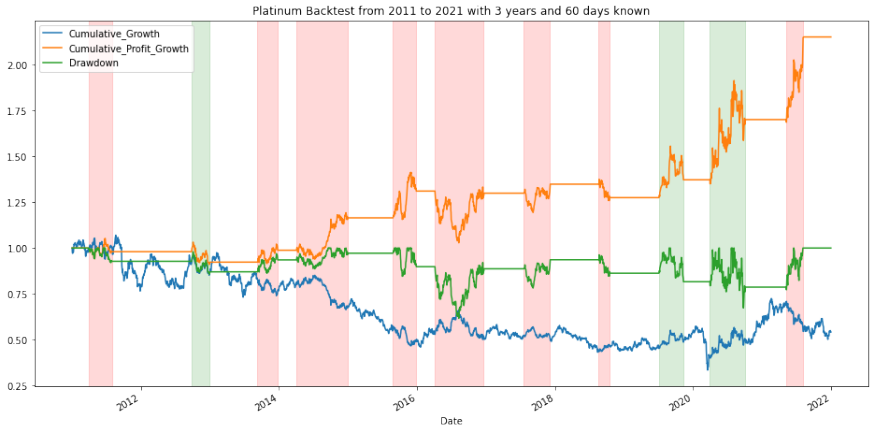
Aceite industrial

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	42.36 %
Op. totales	11
Op. compra	6
Ratio op. compra	33.33 %
Op. venta	5
Ratio op. venta	40.00 %
Beneficio	-61.04 %
Beneficio anual	-8.21 %
Factor beneficio	0.90
Drawdown max.	72.82 %
Drawdown med.	30.13 %
Factor recuperación	0.23
Coef. R2	0.79



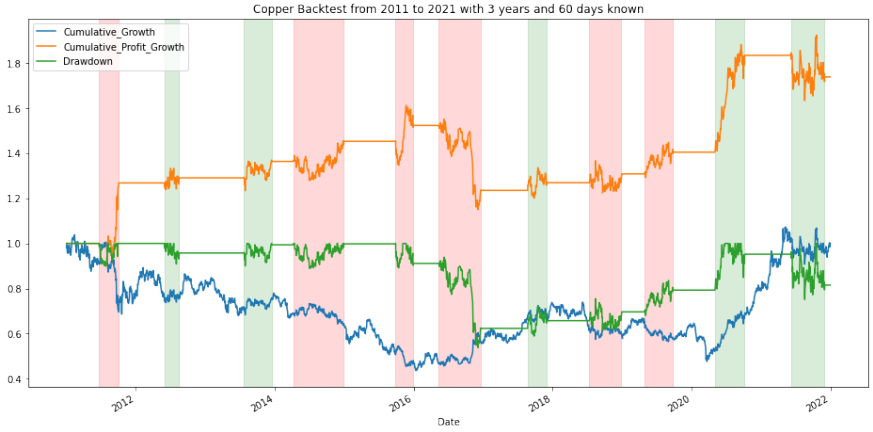
Platino

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	39.85 %
Op. totales	11
Op. compra	3
Ratio op. compra	66.66 %
Op. venta	8
Ratio op. venta	75.00 %
Beneficio	115.18 %
Beneficio anual	7.21 %
Factor beneficio	1.16
Drawdown max.	38.39 %
Drawdown med.	9.91 %
Factor recuperación	1.55
Coef. R2	0.77



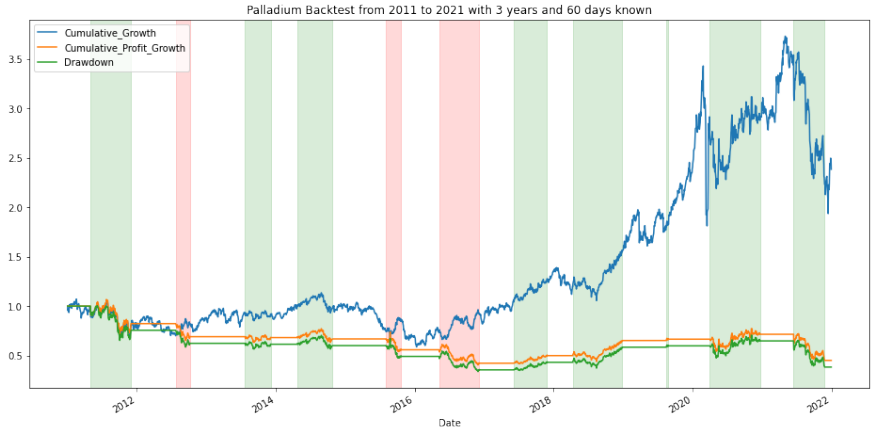
Cobre

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	41.92 %
Op. totales	11
Op. compra	5
Ratio op. compra	80.00 %
Op. venta	6
Ratio op. venta	83.33 %
Beneficio	73.94 %
Beneficio anual	5.16 %
Factor beneficio	1.12
Drawdown max.	46.25 %
Drawdown med.	13.17 %
Factor recuperación	1.19
Coef. R2	0.47



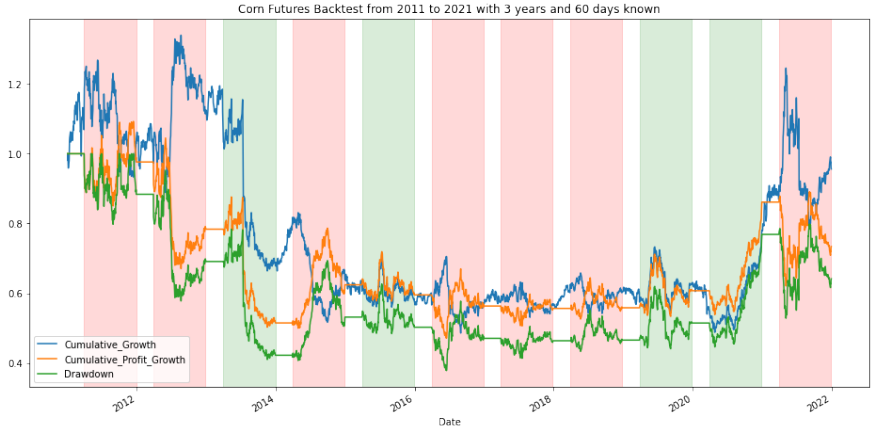
Paladio

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	44.87 %
Op. totales	11
Op. compra	8
Ratio op. compra	50.00 %
Op. venta	3
Ratio op. venta	0.00 %
Beneficio	-54.93 %
Beneficio anual	-6.99 %
Factor beneficio	0.94
Drawdown max.	66.21 %
Drawdown med.	41.29 %
Factor recuperación	0.27
Coef. R2	0.30



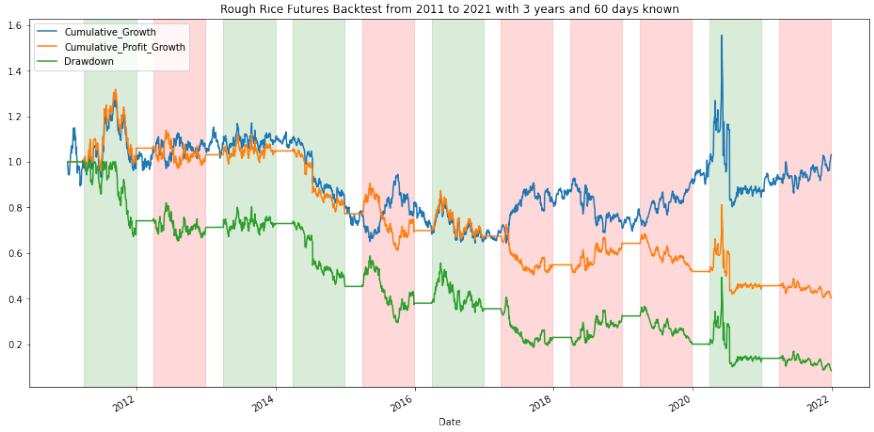
Maíz

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	75.70 %
Op. totales	11
Op. compra	4
Ratio op. compra	50.00 %
Op. venta	7
Ratio op. venta	71.43 %
Beneficio	-26.59 %
Beneficio anual	-2.77 %
Factor beneficio	1.00
Drawdown max.	62.16 %
Drawdown med.	40.99 %
Factor recuperación	0.45
Coef. R2	0.18



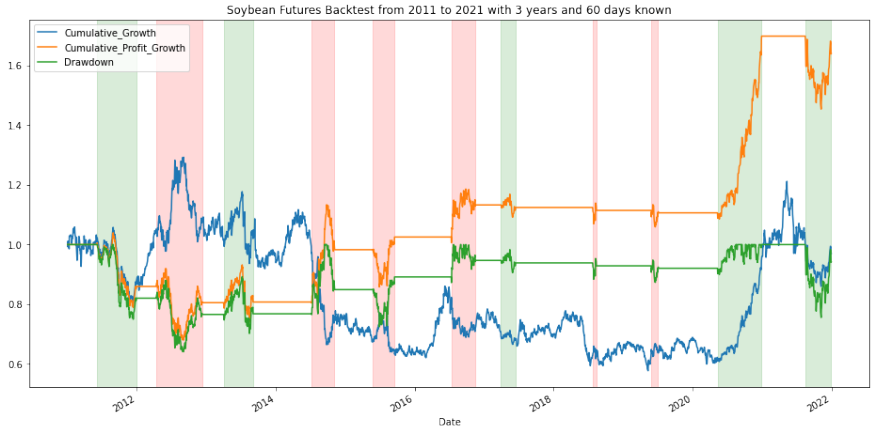
Arroz

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	75.64 %
Op. totales	11
Op. compra	5
Ratio op. compra	40.00 %
Op. venta	6
Ratio op. venta	16.67 %
Beneficio	-59.76 %
Beneficio anual	-7.94 %
Factor beneficio	0.95
Drawdown max.	91.62 %
Drawdown med.	54.77 %
Factor recuperación	0.21
Coef. R2	0.89



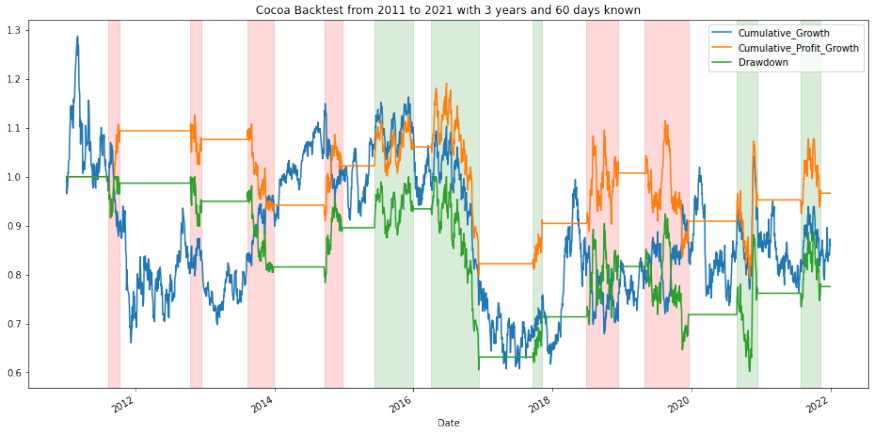
Soja

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	37.12 %
Op. totales	11
Op. compra	5
Ratio op. compra	40.00 %
Op. venta	6
Ratio op. venta	66.67 %
Beneficio	64.01 %
Beneficio anual	4.60 %
Factor beneficio	1.12
Drawdown max.	35.91 %
Drawdown med.	11.01 %
Factor recuperación	1.21
Coef. R2	0.66



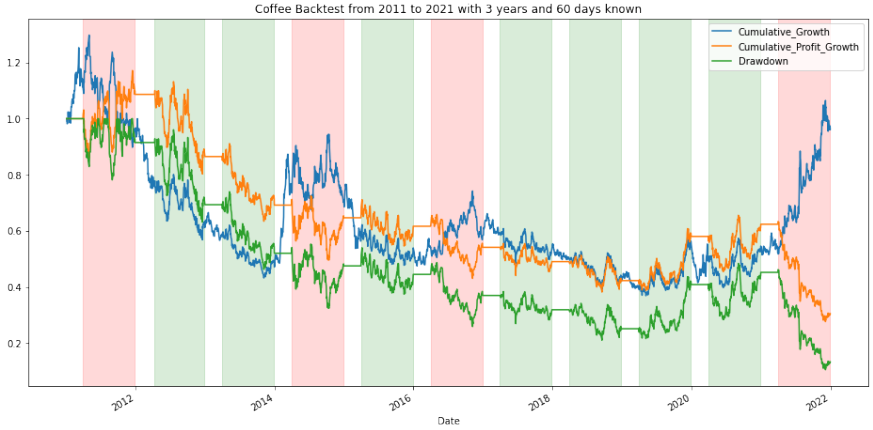
Cacao

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	36.83 %
Op. totales	11
Op. compra	5
Ratio op. compra	80.00 %
Op. venta	6
Ratio op. venta	50.00 %
Beneficio	-3.37 %
Beneficio anual	-0.31 %
Factor beneficio	1.02
Drawdown max.	39.85 %
Drawdown med.	16.22 %
Factor recuperación	0.69
Coef. R2	0.22



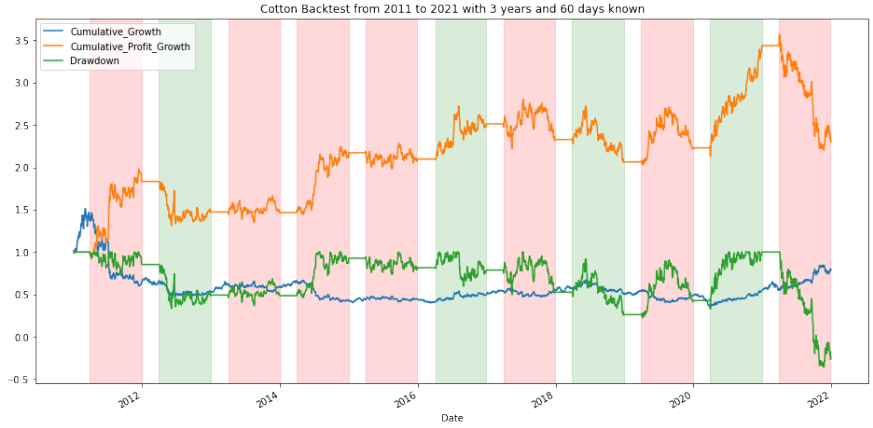
Café

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	75.45 %
Op. totales	11
Op. compra	7
Ratio op. compra	28.57 %
Op. venta	4
Ratio op. venta	50.00 %
Beneficio	-69.58 %
Beneficio anual	-4.92 %
Factor beneficio	0.95
Drawdown max.	89.49 %
Drawdown med.	52.02 %
Factor recuperación	0.16
Coef. R2	0.72



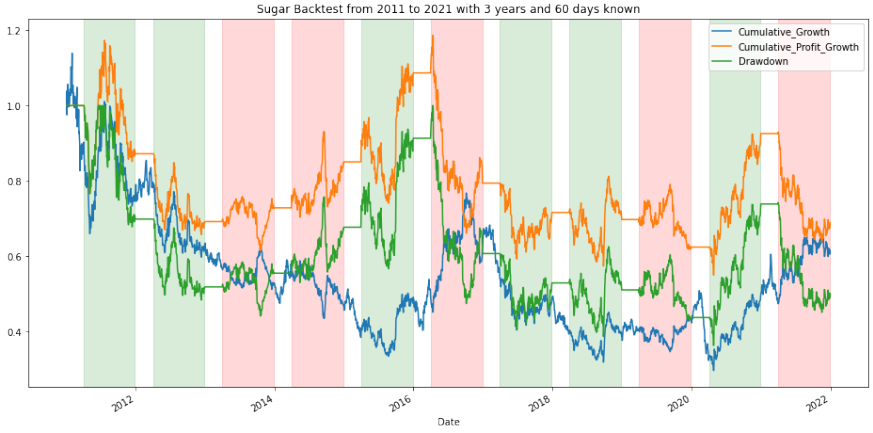
Algodón

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	75.68 %
Op. totales	11
Op. compra	4
Ratio op. compra	50.00 %
Op. venta	7
Ratio op. venta	57.14 %
Beneficio	113.18 %
Beneficio anual	8.00 %
Factor beneficio	1.10
Drawdown max.	135.88 %
Drawdown med.	31.32 %
Factor recuperación	0.99
Coef. R2	0.74



Azucar

Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	75.24 %
Op. totales	11
Op. compra	6
Ratio op. compra	50.00 %
Op. venta	5
Ratio op. venta	40.00 %
Beneficio	-31.42 %
Beneficio anual	-2.51 %
Factor beneficio	1.0
Drawdown max.	63.63 %
Drawdown med.	38.93 %
Factor recuperación	0.42
Coef. R2	0.13



Total	
Inicio	2011-01-03
Fin	2021-12-31
Duración	4015 d
Tiempo exp.	54.51 %
Op. totales	176
Op. compra	81
Ratio op. compra	44.44 %
Op. venta	95
Ratio op. venta	55.79 %
Beneficio	36.43 %
Beneficio anual	2.86 %
Factor beneficio	1.05
Drawdown max.	41.29 %
Drawdown med.	6.91 %
Factor recuperación	0.97
Coef. R2	0.60

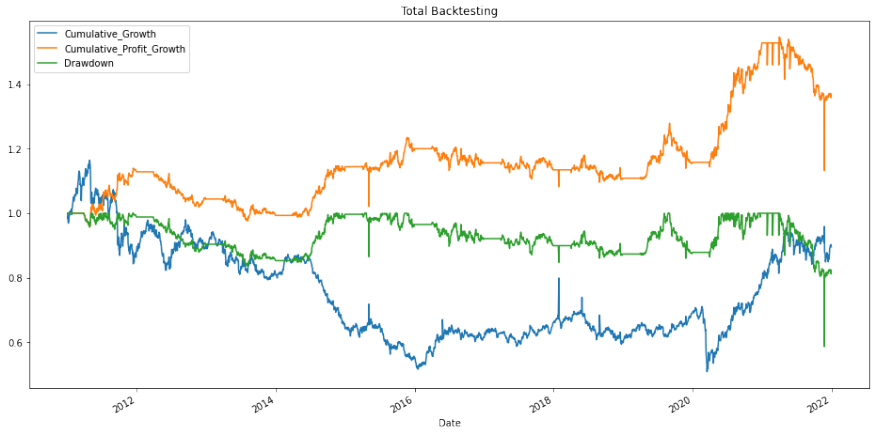


Figura 5.4: Backtesting total de la cartera de inversión

En la figura anterior, 5.4, podemos observar que, después de estar 4015 días operando en el mercado de materias primas con el bot diseñado hemos obtenido un rendimiento final del $36,43\,\%$, es decir, por cada 1000$ invertidos inicialmente, hubiésemos obtenido $1364,30$$. De los 4015 días de operación, el $54,51\,\%$ de las veces (2185 días aproximadamente) el bot de trading se ha posicionado en una orden de compra o venta. Esto muestra que la estrategia opta por operar a un medio y largo plazo, sacando ventaja de movimientos grandes del mercado. También observamos que, de un total de 176 operaciones, 81 son de compra y 95 de venta, las cuales tienen una tasa de acierto del $44,44\,\%$ y $55,79\,\%$ respectivamente. Podemos observar así como el algoritmo tiene cierta predisposición a tomar posiciones de venta en el mercado, para las cuales, además, tiene mayor tasa de acierto. Pero son en las operaciones de compra en las cuales el algoritmo obtiene mayor beneficio operando. Aunque para la operaciones de compra se equivoque más veces de las que acierta, cada acierto supone más beneficio que pérdida en cada fallo, es decir, el bot predice grandes movimientos alcistas de las cotizaciones de las materias primas a costa de confundirse varias veces. Aún así, el bot consigue obtener un profit factor, o factor beneficio, final de $1,05$. Esto quiere decir que, por cada 100 veces que el bot prediga el mercado erróneamente, existen otras 105 en las que lo hace de manera acertada. Este hecho nos hace pensar que, aunque el bot encadene varias operaciones seguidas de forma negativa, es cuestión de seguir operando para volver al punto de partida y seguir aumentando el capital invertido. El recovery factor, o factor de recuperación, de $0,97$ contrasta esta hipótesis. Vemos como, en todas las grandes caídas del beneficio acumulado que hemos tenido, el algoritmo ha sido capaz de cambiar la dirección de dicha curva con relativa rapidez. Este es un dato muy positivo, puesto que no solo importa el rendimiento final que obtengamos, si no el riesgo que hayamos asumido para obtenerlo, y, en todas las caídas que hemos tenido, en menos de un año hemos recuperado todo el capital perdido e incrementado el mismo. Además, a excepción de dos grandes caídas (excepciones de un único día de duración en las que al día siguiente habíamos recuperado todo el capital perdido en esa caída) todas las caídas en el beneficio son relativamente pequeñas. La caída, o Drawdown, media del $6,91\,\%$ corrobora este hecho. Por último, vemos que el coeficiente de determinación de las curva de rendimiento acumulado toma un valor de $0,6$. La forma lineal de la curva junto con este coeficiente son un indicio de la consistencia y bajo riesgo que poseen las operaciones del bot al largo plazo.

Capítulo 6

Conclusión

Desde su introducción en la década de 1960, el trading algorítmico ha revolucionado el sector de las finanzas e inversiones. Lo que en sus comienzos se valoró como una opción más de trading, hoy en día engloba más del 90 % de las operaciones realizadas en los mercados financieros. Este hecho, como hemos podido comprobar en este trabajo, plantea un inconveniente: cada vez es más difícil obtener rendimientos altos de estrategias de inversión ya conocidas. Dado la multitud de inversores que usan estrategias fundamentalmente similares, explotar las ventajas que estas estrategias predicen es cada vez más costoso, puesto que los algoritmos competidores han de repartirse los beneficios, e incluso pueden ejecutar antes la misma orden disminuyendo el beneficio de nuestro algoritmo. Es por este motivo por el que, como podemos ver en [1], una estrategia basada en procesos gaussianos que explote las predicciones del mismo con conceptos similares a los explicados en este trabajo, en la década de 1990 obtenía un rendimiento anual entorno a un 7 %, mientras que en la década de 2000, algo inferior al 3 %.

Los procesos gaussianos son una herramienta bastante conocida en el mundo del trading algorítmico. En este trabajo hemos obtenido un rendimiento anual superior al del mercado, es decir, si al comienzo del periodo de backtesting hubiésemos posicionado una orden de compra y hubiésemos cerrado la misma al finalizar el backtesting, hubiésemos obtenido un resultado mucho peor, es más, mientras el mercado ha obtenido un rendimiento negativo, nuestra estrategia operando sobre el mismo ha obtenido un resultado significativamente positivo. En este trabajo hemos observado una clara ventaja que poseen los procesos gaussianos sobre el mercado, y es por esto por lo que son tan empleados en los bots de trading.

Aún quedan muchos resultados por estudiar y descubrir. Como posibilidad de ampliación para este trabajo estaría la continuidad de caracterización de nuestra matriz de características aportando más información exógena a la curva de precios, como pueden ser indicadores microeconómicos y macroeconómicos, datos atmosféricos, etc. Otra posible vía de estudio sería la mejora de las predicciones del proceso gaussiano mediante un rediseño del kernel que ajuste mejor el conocimiento heredado a la curva predicha. Por último, se podría seguir experimentando con determinados parámetros mencionados en la estrategia para la colocación de órdenes en el mercado, como acompañar las mismas con un Stop Loss y Take Profit y operar con una mayor frecuencia anual, asumir más riesgo en las operaciones, extrapolar el modelo y aplicarle en otros mercados bursátiles, etc. Todos estos avances son una posible continuación para este trabajo, en el cual hemos asentado las bases para operar con los procesos gaussianos en el mercado de materias primas y hemos visto

cómo es posible batir al mercado y obtener un rendimiento anual positivo usando los mismos.

Los procesos gaussianos son una de las técnicas clásicas del machine learning. Hemos visto que para crear una estrategia de inversión rentable no es necesario apostar por las últimas novedades en el campo del machine learning e inteligencia artificial. Debemos tratar de entender las condiciones en las que se encuentra el mercado, modelarlas, encontrar un modelo que se ajuste a nuestras necesidades, y preparar y procesar los datos para permitir al modelo extraer toda la información que los mismos contienen. Los procesos gaussianos llevan demostrando más de 20 años el potencial que poseen dentro del campo de las finanzas. Son la base de grandes estrategias de inversión que conglomerados económicos emplean es sus sistemas y son una muy buena herramienta para muchos inversores particulares.

En último lugar mencionar que todo el código desarrollado en este trabajo se encuentra disponible en un repositorio de GitHub, véase [10].

Bibliografía

[1] Nicolas Chapados, Yoshua Bengio (2007), *Forecasting and Trading Commodity Contract Spreads with Gaussian Processes. Département d'informatique et de recherche opérationnelle, Université de Montréal.*

[2] Zexun Chen (2017), *Gaussian process regression methods and extensions for stock market prediction. Doctoral dissertation, University of Leicester.*

[3] Grzegorz Roguski (2018), *Gaussian Process Regression and Forecasting Stock Trends. GitHub. Disponible en* `https://github.com/gdroguski/GaussianProcesses`

[4] Kevin P. Murphy (2012), *Machine learning: a probabilistic perspective. MIT press.*

[5] Marcos López (2018), *Advances in financial machine learning. Advances in financial machine learning.*

[6] Marcos Cobo, Luis Alberto Fernández, Juan Carlos Fernández (2021), *El filtro de Kalman con aplicaciones en inversiones. Universidad de Cantabria. Disponible en* `https://repositorio.unican.es/xmlui/handle/10902/22794`

[7] Colaboradores de Wikipedia (2022), *Gaussian process. Wikipedia, la enciclopedia libre. Disponible en* `https://en.wikipedia.org/wiki/Gaussian_process`

[8] Colaboradores de Wikipedia (2022), *Algorithmic trading. Wikipedia, la enciclopedia libre. Disponible en* `https://en.wikipedia.org/wiki/Algorithmic_trading`

[9] Colaboradores de Wikipedia (2022), *Commodity market . Wikipedia, la enciclopedia libre. Disponible en* `https://en.wikipedia.org/wiki/Commodity_market`

[10] Marcos Cobo (2022), *Procesos Gausianos con aplicaciones en inversiones (Commodity market). GitHub. Disponible en* `https://github.com/marcoscobo/GaussianProcesses`