Uso del modelo de Black-Scholes para estimación de rendimiento de productos financieros

Para nuestro proyecto, decidimos desarrollar un programa en C++ para estimar los valores futuros de productos financieros utilizando la ecuación de Black-Scholes, la cual estimamos usando polinomios de Adomian. Implementamos este algoritmo en una web app utilizando el paquete de Dash en Python. Esta app nos permite crear un portafolio virtual de productos financieros, obtener los valores de estos productos en la bolsa en tiempo real y estimar el rendimiento futuro del portafolio.

El método de Adomian

El método de descomposición de Adomian (ADM) permite encontrar una solución analítica para una ecuación diferencial, tanto ordinaria como en derivadas parciales. Consiste en identificar en la ecuación diferencial las partes lineal y no lineal, para luego invertir el operador diferencial de mayor orden que se encuentre en la parte lineal y posteriormente considerar la función por conocer como una serie cuyos sumandos por ADM quedarán bien determinados; en seguida se descompone la parte no lineal en términos de los polinomios de Adomian. Definimos las condiciones iniciales o de frontera y los términos que involucran a la variable independiente como aproximación inicial. Así, se encuentran de manera sucesiva los términos de la serie que nos da la solución del problema.

El método está basado en la búsqueda de una solución en forma de serie y en la descomposición de un operador lineal o no lineal en términos de otra serie en la cual los sumandos se calculan de forma recurrente utilizando unos polinomios, los llamados polinomios de Adomian. Bajo ciertas condiciones de convergencia, la suma de la serie dará la solución exacta, pero en general la serie se truncará para dar una buena aproximación.

De forma general el método es el siguiente:

Dada una ecuación diferencial

$$\begin{aligned} Fu(x,t) &= g(x,t) \\ u(x,0) &= f(x) \end{aligned}$$

Donde F representa una combinación lineal de operadores diferenciales que involucra tanto términos lineales como no lineales, por lo que la ecuación puede escribirse como:

$$L_t u(x,t) + Ru(x,t) + Nu(x,t) = g(x,t) \;..... \;(1)$$

Con $L_t = \partial/\partial t$, R operador lineal (con derivadas parciales u ordinarias, respecto a x)), N un operador no lineal y g un término no homogéneo independiente de u.

De la ecuación (1) podemos despejar a L_t

$$L_t\, u(x,t) = g(x,t) - Ru(x,t) - Nu(x,t)$$

Como tenemos que L_t es un operador invertible, tendríamos que $L^{-1}{}_t(.) = \int_0^t (.)ds$. Aplicando esto a la ecuación, obtenemos

$$L^{-1}{}_t L_t\, u(x,t) = L^{-1}{}_t g(x,t) - L^{-1}{}_t R\, u(x,t) - L^{-1}{}_t Nu(x,t)$$

De donde obtenemos:

$$u(x,t) = f(x) + L^{-1}{}_t g(x,t) - L^{-1}{}_t R\, u(x,t) - L^{-1}{}_t Nu(x,t)$$

Con $f(x)$ como la constante de integración (respecto a t) que satisface $L_t f = 0$. El método de Adomian consiste en descomponer una serie por medio de la función desconocida $u(x,t)$ dada por:

$$u(x,t) = \sum_{n=0}^{\infty} u_n(x,t)$$

Dónde el término no lineal $Nu(x,t)$ se descompone de la forma

$$Nu(x,t) = \sum_{n=0}^{\infty} A_n(u_0, u_1, \dots, u_n)$$

Donde $\{A_n\}$ es la sucesión conocida como los polinomios de Adomian. Con esto, el problema que teníamos se convierte en:

$$\sum_{n=0}^{\infty} u_n(x,t) = f(x) + L^{-1}{}_t g(x,t) - L^{-1}{}_t R \sum_{n=0}^{\infty} u_n(x,t) - L^{-1} \sum_{n=0}^{\infty} A_n(u_0, u_1, \dots, u_n)$$

Con esto, obtenemos el algoritmo recursivo

$$u_0(x,t) = f(x) + L^{-1}{}_t g(x,t)$$
$$u_{n+1}(x,t) = L^{-1}{}_t R\, u_n(x,t) - L^{-1}{}_t A_n(u_0,\ u_1, \dots,\ u_n)$$

Por lo tanto la aproximación a la solución quedaría como

$$u(x,t) \approx \sum_{n=0}^{k} u_n(x,t)$$

Y cuando k tiende a infinito tendríamos la igualdad.

Ecuación de Black-Scholes

La ecuación de Black-Scholes nace en 1973 resultado del trabajo de Black, Scholes y Merton, quienes derivaron una ecuación diferencial cuyo resultado es el precio de una opción de compra; en este punto, la ecuación es de carácter lineal. Derivado del trabajo realizado, se encontraron diversas variantes. Presentaremos una versión no lineal, donde la no linealidad es derivada de la volatilidad; es decir, dependerá del tiempo y del activo subyacente.

$$\hat{\sigma}^2 = \hat{\sigma}^2(t, S, V_s, V_{ss})$$

A lo largo del tiempo se han estudiado diversos enfoques para mejorar el modelo, planteando en ellos diversas expresiones para la volatilidad. La expresión que aquí utilizaremos fue deducida por Frey y Patie.

$$\hat{\sigma}^2(t, S, V_s, V_{ss}) = \frac{\sigma}{1-\rho\, S\, \lambda(S)\, u_{ss}}$$

Con σ la volatilidad tradicional, ρ una constante y $\lambda(S)$ una función continua positiva. Para el desarrollo de la aplicación, consideraremos $\lambda = 1$, con lo que la expresión para la ecuación quedaría.

$$V_\tau(S,\tau) + \frac{1}{2}\sigma^2 S^2 \frac{V_{ss}(S,\tau)}{(1-\rho S V_{ss}(S,\tau))^2} + rSV_s(S,\tau) - rV(S,\tau) = 0$$
$$V(S,\tau) = f(S),\ 0 < S < \infty$$

Tomaremos la siguiente aproximación de la función

$$\frac{1}{(1-F)^2} \approx 1 + 2F + O(F)^3 .$$

Y así llegamos a la ecuación que utilizaremos en el desarrollo de la aplicación.

$$V_\tau(S,\tau) + \frac{1}{2}\sigma^2 S^2 V_{ss}(S,\tau)(1 + 2\rho S V_{ss}(S, \tau)) + rSV_s(S,\tau) - rV(S,\tau) = 0$$
$$V(S,\tau) = f(S),\ 0 < S < \infty$$

Hagamos una traslación respecto al tiempo; es decir, $t = T - \tau$ y cambiando $V(S, \tau) = u(S, \tau)$ obtenemos

$$u_t(S,t) + \frac{1}{2}\sigma^2 S^2 u_{SS}(S,t)(1 + 2\rho S u_{SS}(S, t)) + rSu_S(S,t) - r\, u(S,t) = 0$$
$$u(S,0) = f(S),\ 0 < S < \infty$$

Aplicación del método de Adomian a la ecuación de Black-Scholes y desarrollo del código

Veamos cómo quedan las expresiones del método de Adomian para el caso de la ecuación previamente planteada.

$$L_t = \frac{\partial}{\partial t}, \; R = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} + rS\frac{\partial}{\partial S} - r \;, \; N = \rho\sigma^2 S^3 (\frac{\partial^2}{\partial S^2})^2 \text{ y } g(S,t) = 0\,.$$

Llegamos a las expresión recursiva.

$$u_0(S,t) = f(S)$$
$$u_{n+1}(S,t) = L^{-1}{}_t R\, u_n(S,t) - \rho\sigma^2 S^3 L^{-1}{}_t A_n(u_0, u_1, \ldots, u_n)$$

Como el término no lineal es $(\frac{\partial^2}{\partial S^2})^2$, tenemos las siguientes expresiones para los polinomios de Adomian.

$A_0 = u^2{}_{0,ss}$
$A_1 = 2\, u_{\,0,ss} u_{\,1,ss}$
$A_1 = 2\, u_{\,0,ss} u_{\,2,ss} + u^2{}_{1,ss}$
...

Ahora que tenemos las expresiones, procedemos a programar el método. Para esta parte del desarrollo de la aplicación, utilizaremos C++ como lenguaje. Dentro del código definiremos distintas funciones, considerando que la condición inicial se puede modelar mediante una regresión de polinomios; tenemos funciones para la condición inicial, para las expresiones $A_0, A_1,\ u_1,\ u_2$ y para el cálculo de las derivadas.

Para las derivadas de distintos orden, utilizamos la aproximación de Taylor en diferencias centradas, las cuales tienen las siguientes expresiones.

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0-h)}{2h}$$

$$f''(x_0) \approx \frac{f(x_0+h) - 2f(x_0) + f(x_0-h)}{h^2}$$

$$f^{(3)}(x_0) \approx \frac{f(x_0+2h) - 2f(x_0+h) + 2f(x_0-h) - f(x_0-2h)}{2h^3}$$

$$f^{(4)}(x_0) \approx \frac{f(x_0+2h) - 4f(x_0+h) + 6f(x_0) - 4f(x_0-h) - f(x_0-2h)}{h^4}$$

Después del análisis realizado para distintas combinaciones de betas en las funciones, se determinó que $h = 0.01$. Una vez que obtuvimos estas aproximaciones, las incluimos en los cálculos. Es importante aclarar, que para la implementación del método usamos $k = 2$ en la aproximación de la solución.

Finalmente, el programa recibe el valor de los coeficientes de la regresión, la desviación estándar de los valores de producto financiero y el tiempo que dura la inversión (en meses), realiza las operaciones y, finalmente, regresa la estimación del valor de la acción.

Aplicación web

Para ejemplificar el alcance de la implementación del método de Adomian a la ecuación de Black Scholes, creamos un app que permite generar análisis financieros reales. La app fue desarrollada en Python 3.7.3, empleando el framework de Dash 0.43.0, y permite al usuario seleccionar un portafolio compuesto de distintos productos financieros -básicamente, *commodities*- listados en diversos mercados financieros globales.

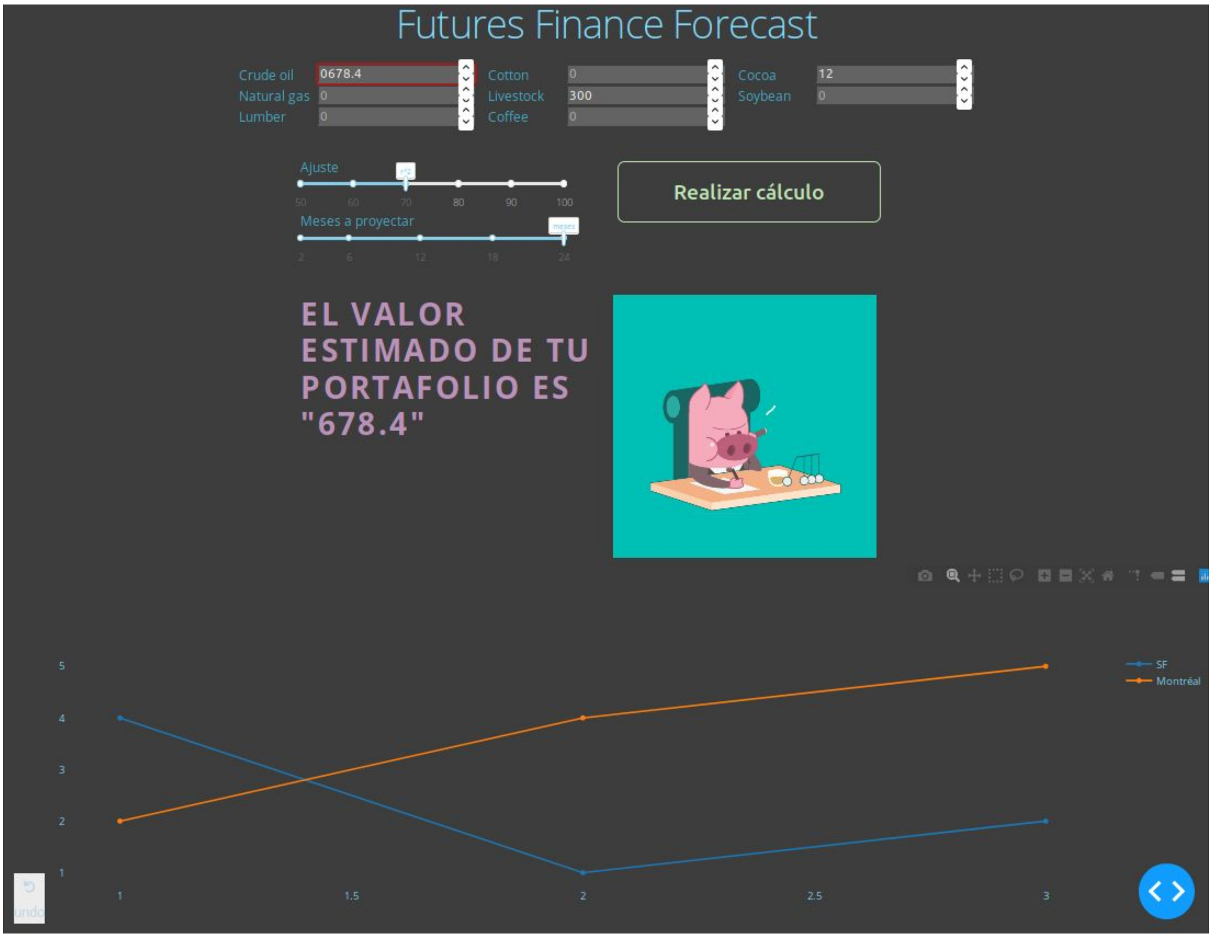

Figura 1. Diseño preliminar de la aplicación.

Una vez seleccionados los productos financieros, el usuario elige un periodo de tiempo determinado y el programa calcula el rendimiento de su portafolio para el término de ese periodo. El portafolio consiste en la selección de cualquier combinación posible de los siguientes productos.

Figura 1. Diseño preliminar de la aplicación. Producto	Fuente	Código
Gold, London A.M. Fixing	LBMA	LBMA/GOLD
Oats, CBOT Oats Futures	CME	CHRIS/CME_O1
Silver, London fixing	LBMA	LBMA/SILVER
Dairy, CME Milk Futures	CME	CHRIS/CME_DA1
Pork, CME Lean Hog Futures	CME	CHRIS/CME_LN1
Corn, CBOT Corn Futures	CME	CHRIS/CME_C1
Rice, CBOT Rice Futures	CME	CHRIS/CME_RR1
Lumber Futures	CME	CHRIS/CME_LB1
NYMEX RBOB Gasoline Futures	NYMEX	CHRIS/CME_RB1
NYMEX Natural Gas Futures	NYMEX	CHRIS/CME_NG1
Platinum, NYMEX Platinum Futures	NYMEX	CHRIS/CME_PL1
Soybean, CBOT Soybean Futures	CME	CHRIS/CME_S1

Elegimos estos productos específicos porque representan una canasta diversificada, que incluye categorías como: metales, granos, productos de la granja, petróleo y sus derivados y recursos forestales. Además de esto, representan a dos grandes mercados: el europeo por medio del London Bullion Markets Association (LBMA) y el americano a través del Chicago Mercantile Exchange (CME).

El CME es uno de los mercados más grandes de opciones y futuros en el mundo. Nació en el 2007 tras la unión del Chicago Mercantile Exchange (CME) y el Chicago Board of Trade (CBOT) y posteriormente adquirió el New York Mercantile Exchange (NYMEX), del cual también incluimos productos. Por otro lado, el LBMA es uno de los mayores mercados de oro y plata en el mundo.

Funcionamiento y estructura

La aplicación realiza dos principales procesos: el primero consiste en extraer la información en tiempo real de todos los productos y almacenarla localmente en el dispositivo del usuario. Todos los datos se cargan desde el inicio debido a que Quandl, la API proveedora de esta información, cuenta con un límite de llamadas diarias y resultaría ineficiente realizar una llamada cada vez que el usuario analiza un portafolio diferente.

Para volver este proceso eficiente se implementó una arquitectura de cómputo en paralelo para este proceso por medio de Dask 1.2.2: esto permite que la descarga de los datos se ejecute en un tiempo relativamente menor al tiempo que se tardaría en descargar los datos secuencialmente. Este proceso crea una tabla en formato CSV que es la base para los futuros análisis de portafolios que el usuario realice.

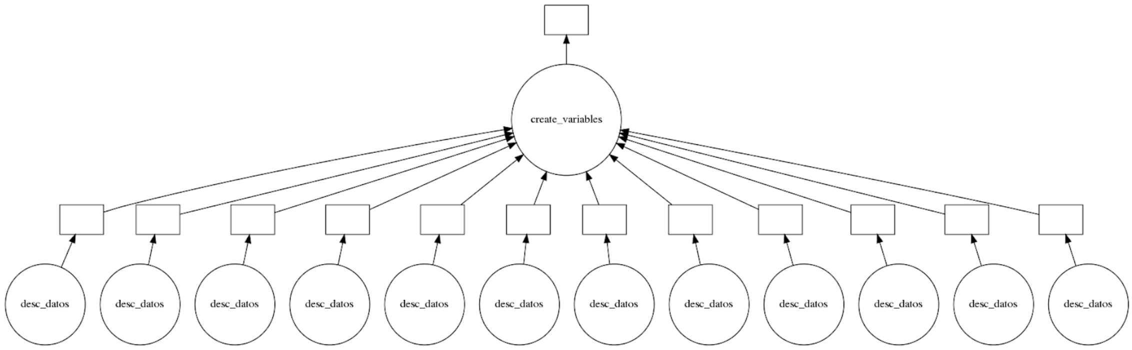

Figura 2. Arquitectura en paralelo del proceso de carga de datos.

La figura anterior muestra la arquitectura que empleamos y que permite descargar la totalidad de los datos simultáneamente y juntarlos dentro de un mismo CSV.

El segundo proceso principal que ejecuta nuestra app es la predicción del valor de cada uno de los productos que el usuario seleccionó y el cálculo del rendimiento después de los meses indicados. Para esto se creó una función que recibe la información correspondiente al número de meses dentro de los que se desea predecir y una lista de productos sobre los que se desea hacer esta predicción.

Para esto se lee la columna del CSV correspondiente al nombre del producto, se extrae información de la serie de tiempo - que incluye la desviación estándar de los precios y el valor último del stock- y se realiza una regresión por medio de la función polyfit del paquete Numpy 1.16.2. Esta información provee los datos de entrada del modelo de Black-Scholes por medio del método de Adomian, implementado en C++ .

La función ejecuta el código implementado en C++ y obtiene el valor predicho del producto en el tiempo previamente establecido. Finalmente, con esto se calcula el rendimiento del producto escogido en el tiempo y se reporta al usuario el desempeño total de su portafolio.

Referencias

I. Paz Cendejas, Francisco (2015). *El método de Adomian para la solución de un modelo de Black-Scholes no-lineal*. Universidad Autónoma Metropolitana.

II. John H. Mathews and Kurtis K. Fink (2004). *Numerical methods using Matlab*. New Jersey, USA: Prentice-Hall Inc.

III. Palacios, Erick (2019). *Notas de la clase Métodos Numéricos y Optimización 2019.* ITAM. Sitio web: https://github.com/ITAM-DS/analisis-numerico-computo-cientifico