

ANEXO I: PLANTILLA PARA PORTADA



Universidad de Valladolid

**FACULTAD DE CIENCIAS SOCIALES, JURÍDICAS Y DE LA
COMUNICACIÓN**

Grado en Administración y Dirección de Empresas

TRABAJO DE FIN DE GRADO

Modelo de predicción/alerta en la gestión de riesgos de mercado

Presentado por Alberto Chico Manzano

Tutelado por Susana Alonso Bonis

Segovia, 01 de julio de 2023

1. Introducción

El propósito de este trabajo es elaborar un estudio del uso del *machine learning* para desarrollar modelos para la gestión de riesgos de mercado. Como ejemplo para ello tomaremos como medida el *Value at Risk* (VaR de ahora en adelante), cuya utilización, principalmente, se dedica a la gestión de carteras de inversión para la identificación del capital suficiente para absorber las posibles pérdidas.

En la actualidad, se está llevando a cabo una revolución tecnológica que parte de la ciencia de los datos y cuya adaptación a todos los sectores es primordial. De hecho, esta revolución se ha convertido en uno de los principales desarrollos a llevar a cabo en el entorno financiero, con la adaptación del uso de algoritmos, la Inteligencia Artificial y dentro de esta el *machine learning*, de los cuales ya están empezando a tomar parte de la operativa de, sector.

En definitiva, ante un cambio disruptivo de carácter predominantemente tecnológico, social y económico; este trabajo tiene un fin eminentemente experimental, para tratar de aportar valor a un aspecto fundamental de los modelos para la gestión de riesgos de mercado.

El objetivo concreto es: analizar y elaborar modelos de *machine learning* aplicables al VaR.

2. Inteligencia Artificial: técnicas de *Machine Learning*

El *machine learning*, o aprendizaje automático, es una rama de la inteligencia artificial que permite a las máquinas aprender de los datos, sin necesidad de programación explícita. Esto se logra a través de algoritmos que buscan patrones y relaciones en los datos, y los utilizan para hacer predicciones o tomar decisiones.

a. Concepto¹

La utilización de la Inteligencia Artificial favorece la adaptación y toman decisiones en base a un aprendizaje especializado monitorizando la información estadística. Esta capacidad de aprendizaje se consigue con técnicas de:

- *Machine Learning* (aprendizaje automático).
- *Deep Learning* (aprendizaje profundo).

El *machine learning* o aprendizaje automático es el nexo entre el Big Data y la Inteligencia Artificial.

Las técnicas de *machine learning* permiten a los algoritmos que decidamos crear, identificar patrones complejos entre gran cantidad de datos para, posteriormente, inferir en sus propias reglas para detectar patrones similares en nuevos conjuntos de datos.

b. Funcionalidad

Las principales funcionalidades² del *machine learning* incluyen:

Clasificación: el *machine learning* puede clasificar datos en diferentes categorías o clases basándose en patrones encontrados en los datos de entrenamiento. Por ejemplo, se puede utilizar el *machine learning* para clasificar imágenes en diferentes categorías como animales, objetos, paisajes, etc.

Regresión: el *machine learning* puede predecir valores numéricos basándose en patrones

¹ Fuente: Universidad Autónoma de Madrid (UAM) - Instituto de Ingeniería del Conocimiento

² <https://cleverdata.io/conceptos-basicos-machine-learning/>

encontrados en los datos de entrenamiento. Por ejemplo, se puede utilizar el *machine learning* para predecir el precio de una casa basándose en características como el tamaño, la ubicación, etc.

Agrupamiento: el *machine learning* puede agrupar datos en diferentes grupos o clústeres basándose en su similitud. Por ejemplo, se puede utilizar el *machine learning* para agrupar clientes en diferentes grupos basándose en su comportamiento de compra.

Análisis de texto: el *machine learning* puede analizar texto para extraer información útil como el sentimiento, el tema, las entidades nombradas, etc. Por ejemplo, se puede utilizar el *machine learning* para analizar comentarios de los clientes en línea y detectar problemas comunes.

Detección de anomalías: el *machine learning* puede detectar patrones inusuales o anomalías en los datos que pueden ser indicativos de problemas o errores. Por ejemplo, se puede utilizar el *machine learning* para detectar transacciones fraudulentas en una base de datos de transacciones financieras.

Optimización: el *machine learning* puede encontrar la mejor solución para un problema dado en función de los datos disponibles y los objetivos del problema. Por ejemplo, se puede utilizar el *machine learning* para optimizar una cadena de suministro para minimizar los costos y mejorar la eficiencia.

c. Tipología

Existen varias tipologías³ de *machine learning*, cada una con sus propias características y aplicaciones. A continuación, se describen brevemente algunas de las técnicas más comunes:

Aprendizaje supervisado: en este tipo de aprendizaje, el modelo se entrena con datos etiquetados, es decir, datos que ya están clasificados o etiquetados con una respuesta correcta. El modelo aprende a predecir la respuesta correcta para nuevos datos basándose en el patrón que encuentra en los datos de entrenamiento.

Aprendizaje no supervisado: en este tipo de aprendizaje, el modelo se entrena con datos no etiquetados, es decir, datos que no tienen una respuesta correcta conocida. El modelo busca patrones y relaciones en los datos para agruparlos o clasificarlos según su similitud.

Aprendizaje por refuerzo: en este tipo de aprendizaje, el modelo aprende a tomar decisiones en un entorno interactivo, recibiendo retroalimentación positiva o negativa según el resultado de sus acciones. El modelo ajusta su comportamiento para maximizar la recompensa o minimizar el castigo.

Redes neuronales: las redes neuronales son modelos inspirados en el cerebro humano, que consisten en capas de nodos interconectados. Cada nodo recibe una entrada y produce una salida, y las conexiones entre los nodos tienen pesos que se ajustan durante el entrenamiento para mejorar la precisión del modelo.

Árboles de decisión: los árboles de decisión son modelos que representan decisiones y sus posibles consecuencias en forma de un árbol. Cada nodo representa una decisión o una característica del dato, y las ramas representan las posibles respuestas o resultados.

³ Shelley Pursell (2023, 31 de enero). <https://blog.hubspot.es/marketing/machine-learning> y <https://www.apd.es/algoritmos-del-machine-learning/>

d. Metodología

La metodología⁴ de *machine learning* es un conjunto de pasos y prácticas que se utilizan para desarrollar modelos de aprendizaje automático de manera efectiva y eficiente. A continuación, se describen los principales pasos de la metodología de *machine learning*:

- **Definir el problema:** en esta etapa se define claramente el problema que se quiere resolver, se establecen los objetivos del modelo de *machine learning* y se recopilan los datos necesarios para el entrenamiento.
- **Pre-procesamiento de datos:** antes de entrenar el modelo, es necesario preparar los datos, esto puede incluir la limpieza de los datos, la selección de características relevantes, la normalización y la codificación de las variables categóricas.
- **Selección del modelo:** en esta etapa se selecciona el modelo adecuado para el problema, teniendo en cuenta las características de los datos y los objetivos del proyecto. Esta elección puede involucrar la selección de uno o varios modelos, la definición de los parámetros y la arquitectura del modelo.
- **Entrenamiento del modelo:** en esta etapa se utiliza un conjunto de datos de entrenamiento para ajustar los parámetros del modelo. El objetivo es minimizar el error del modelo y mejorar su capacidad de generalización.
- **Evaluación del modelo:** una vez entrenado el modelo, se evalúa su rendimiento utilizando un conjunto de datos de prueba. Se pueden utilizar diferentes métricas de evaluación, dependiendo del tipo de problema y del objetivo del modelo.
- **Ajuste del modelo:** si el rendimiento del modelo no es satisfactorio, se pueden realizar ajustes en el modelo o en los datos para mejorar su rendimiento.
- **Despliegue del modelo:** una vez que el modelo ha sido entrenado y evaluado satisfactoriamente, se despliega en producción para su uso en la vida real.
- **Monitoreo y mantenimiento del modelo:** una vez que el modelo está en producción, es importante monitorear su rendimiento y hacer ajustes periódicos para mantener su precisión y evitar la degradación del modelo.

Dicha metodología será utilizada para llevar a cabo el proceso

3. Métricas de riesgo de mercado: metodología VaR (*Value at Risk*)

En el siguiente apartado, abordaremos los principios, métricas y metodología para el cálculo del VaR

a. Principios

Como ocurre con otros riesgos, no es en absoluto fácil definir el riesgo de mercado, digamos por simplificar se refiere a la posibilidad de que los cambios en el mercado financiero afecten el valor de los activos de una empresa⁵.

Según Holton (2009), para que haya riesgo de mercado deben de existir dos componentes:

- Una exposición, en este caso, la tenencia de una cartera de instrumentos financieros. En una definición menos pura de riesgo de mercado, esto se puede ampliar a pérdidas sufridas por empresas que no tienen exposición a productos financieros, pero que son sensibles a movimientos de los mercados financieros. Piénsese en una empresa con deuda a tasa variable, que tiene un riesgo innegable en el caso de que los tipos de interés aumenten.
- Una incertidumbre. En el caso del riesgo de mercado, la incertidumbre radica en la evolución futura del valor de mercado de los instrumentos mantenidos. En caso de desarrollos adversos del mercado, el inversor está expuesto a una pérdida en lugar de la

⁴ <https://datascientest.com/es/machine-learning-definicion-funcionamiento-usos>

⁵ <https://kghm.com/es/inversionistas/gestion-de-riesgos/riesgo-de-mercado>

ganancia esperada. La posibilidad de sufrir pérdidas forma parte intrínseca de la inversión en cualquier instrumento financiero, de ahí la importancia de que el inversor sepa gestionar este riesgo.

El riesgo de mercado y el riesgo económico son dos conceptos diferentes y no deben confundirse. Algunas diferencias entre ambos son:

El riesgo de mercado se refiere a la posibilidad de que los cambios en el mercado financiero afecten el valor de los activos de una empresa. Por ejemplo, la fluctuación de los precios de las materias primas o de las divisas. En cambio, el riesgo económico se refiere a los factores relacionados con la propia actividad productiva y el mercado en el que está presente la empresa, como los cambios en los hábitos de consumo o las crisis económicas⁶

Se aprecia de inmediato que el riesgo y el resultado están estrechamente relacionados. El riesgo cero no existe cuando hay una inversión y, como regla general, la ganancia potencial aumenta con la asunción de riesgos. Sin embargo, existen opciones de inversión más o menos eficientes en términos de relación beneficio/riesgo.

El conjunto de carteras que maximizan el resultado potencial para un nivel de riesgo dado (o minimizan el nivel de riesgo para un resultado potencial dado) se denomina "frontera eficiente"⁷.

Merece la pena pararse en el concepto de activo libre de riesgo base de la teoría de Markowitz (1952) o de Sharpe (1964), hoy en día parece cada vez más evidente que incluso los activos considerados sin riesgo sí lo presentan. A modo de ejemplo, la deuda pública de EE. UU. está a unos niveles tales que hace difícil sostener que invertir en los *Treasury Bills* de EE. UU. esté completamente exento de riesgo.

En cuanto al resultado, está íntimamente ligado a la valoración de los instrumentos financieros. De esta valoración resultarán en última instancia los flujos financieros generados por la posición ocupada, flujos financieros que impactan directamente en la cuenta de resultados. Esta es la razón por la que la gestión de riesgos se trata, en última instancia, de hacer preguntas como: ¿Cuál es el valor de mi posición hoy, en comparación con lo que valía en el momento de la adquisición? (ganancia o pérdida no realizada); ¿Cuánto valdrá mi posición mañana, dentro de una semana, dentro de un mes, en igualdad de condiciones?; ¿Cuánto valdría mi posición si cambia este o aquel indicador de mercado?

b. Métricas de Riesgo de Mercado

La medición del riesgo de mercado implica modelos de indicadores de mercado, así como funciones de valoración de instrumentos financieros. Entonces es necesario definir una métrica de riesgo y calcular este indicador para la cartera en cuestión. Gracias a este cálculo se obtiene una medida de riesgo. Inmediatamente queda claro que las opciones metodológicas, tanto en la definición del indicador elegido como en la forma de calcularlo, influyen decisivamente en la medición del riesgo.

Un buen indicador es aquel que da medidas coherentes de riesgos (Hull, 2012), es decir:

-
- Si se agrega una cantidad de efectivo K (activo libre de riesgo) a la cartera,

⁶ <https://www.becas-santander.com/es/blog/riesgos-financieros.html>

⁷ <https://economipedia.com/definiciones/frontera-carteras-eficiente.html>

entonces su medida de riesgo debe disminuir en la misma cantidad (K).

- Si el valor de una cartera se multiplica por una cantidad M, entonces su medida de riesgo debe multiplicarse por la misma cantidad (M).
- Si se fusionan dos carteras A y B, la medida de riesgo de la cartera resultante no debe ser mayor que la suma de las medidas de riesgo iniciales de A y B.
- Algunos indicadores de riesgo pueden ser más o menos adecuados y consistentes, dependiendo de la naturaleza de los instrumentos financieros que se posean.

Como se mencionó anteriormente, el riesgo tiene dos dimensiones, una dimensión cuantitativa (cuánto se arriesga a perder) y una dimensión probabilística (con qué probabilidad). Los indicadores de riesgo pueden cuantificar uno u otro aspecto, o una combinación de ambos. Además, una cartera rara vez consta de un solo instrumento, pero como se acaba de ver, el indicador de riesgo se adapta más o menos a la clase de activos (tasa de interés, acciones, divisas, materias primas, crédito) y la linealidad (o no) del instrumento.

De ahí la búsqueda de indicadores sintéticos que dio lugar al concepto de valor en riesgo (VaR). Estos indicadores sintéticos también deben tener en cuenta las correlaciones entre las diferentes variables de mercado involucradas en el cálculo. La correlación entre dos variables indica su tendencia a moverse simultáneamente, en la misma dirección o en direcciones opuestas (correlación no significa necesariamente causalidad).

c. Aplicaciones estadísticas a la metodología VaR

Dado que el VaR es la pérdida esperada de una cartera durante un período de tiempo específico para un nivel de probabilidad establecido, se puede establecer la exposición a riesgo. A modo de ejemplo, si un VaR diario se sitúa en 400.000 €, con una confianza del 95 %, esto indica que en un día se puede estar seguro de que la máxima pérdida serán esos 400.000 en un 95% de las ocasiones o, dicho de otro modo, solo habrá un 5 % de probabilidades de que las pérdidas superen los 400.000€.

El VaR calcula esa pérdida máxima usando el concepto de volatilidad, para lo que, a su vez, necesita calcular las correlaciones entre los activos. El nivel de confianza habitual es el 95 %, no siendo infrecuente calcularla también al 99 %.

Resumiendo, el cálculo se realiza en cuatro grandes etapas:

- Elección del horizonte temporal en el que se quiere estimar la pérdida máxima. Esto lo debe realizar el usuario de la metodología. Dependiendo de la finalidad del cálculo, se han llegado a fijar en la realidad períodos que van desde 1 día a 1 año. Es posible que los reguladores y los participantes en mercados ilíquidos deseen estimar las exposiciones al riesgo de mercado durante un período más largo. En cualquier caso, el responsable de la toma de decisiones debe especificar un horizonte temporal.
- Se debe fijar el nivel de confianza. Cuando se usa el 95 %, quiere decir que 1 día de cada 20 se superará esa pérdida máxima, por lo que puede no ser apropiada en la gestión a largo plazo. Casi siempre los directivos, pero también los accionistas, quieren conocer la pérdida en situaciones catastróficas (99 %).
- Cálculo de la distribución de probabilidad de los rendimientos probables para el instrumento o la cartera en consideración; se pueden utilizar varios métodos.
- Se realiza el cálculo matemático del VaR. Estas cuatro etapas permiten, en teoría, calcular una estimación de VaR "a mano", aunque en la vida real hay ya modelos estadísticos preestablecidos que realizarán estos cálculos.

4. Aplicación del machine learning

En el siguiente apartado trataremos de explicar la aplicación del machine learning en el cálculo del VaR

a. ¿Qué modelo utilizar?

En el siguiente apartado trataremos diferentes modelos en los cuales compararemos los cálculos, utilizando diferentes métodos. Dadas las posibilidades que nos plantea el *machine learning* y cálculos que se utilizan en la actualidad veremos las principales diferencias que nos arrojan.

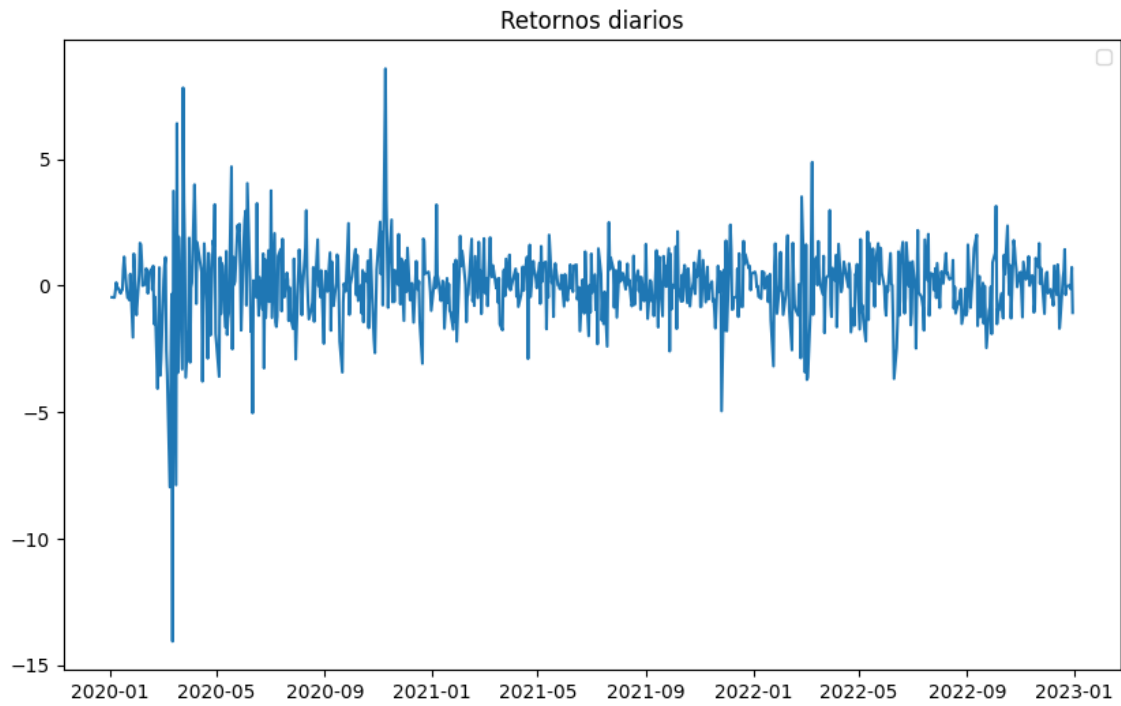
b. Muestra

Para la elaboración de la muestra, utilizaremos diferentes librerías de Python, así como la extracción de los datos de un periodo determinado, (01/01/2020 – 01/01/2023), con los que elaboraremos el estudio y la aplicación de los diferentes modelos.

Una vez, obtenemos el resultado, llevamos a cabo los siguientes cálculos:

Retornos diarios

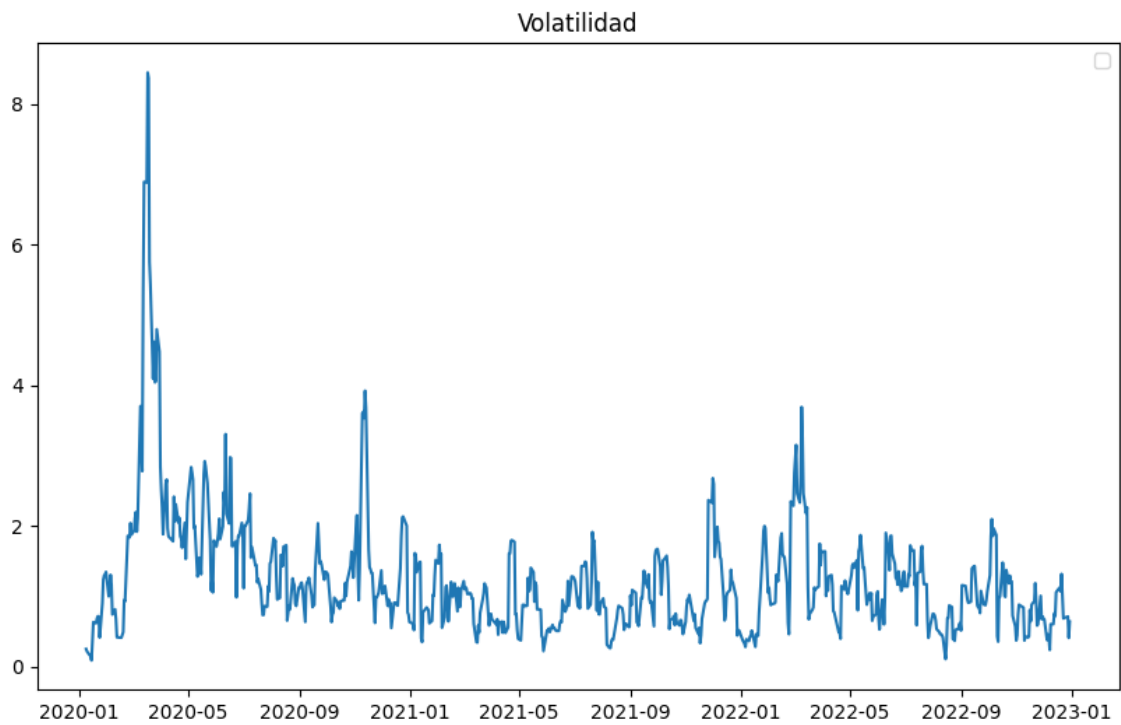
Gráfico 1 Ejemplo de retornos diarios del IBEX 35



(Fuente: elaboración propia en Python)

Volatilidad en una ventana de 5 días.

Gráfico 2 Ejemplo de volatilidad del IBEX 35



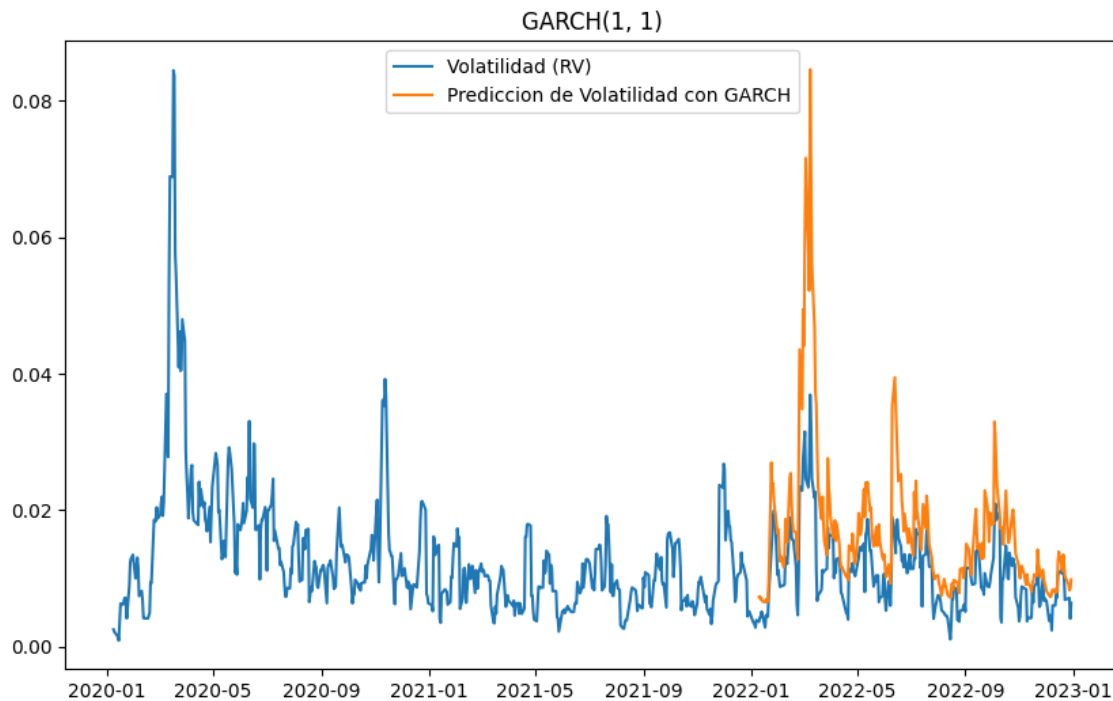
(Fuente: elaboración propia en Python)

c. Modelo A

Para este primer modelo utilizaremos el modelo GARCH (*Generalized Autoregressive Conditional Heteroskedasticity*). El modelo GARCH es un modelo de volatilidad que se utiliza comúnmente en finanzas para modelar la heteroscedasticidad condicional en los datos de series de tiempo. En este caso, se utiliza el modelo GARCH (1,1), que especifica un proceso autorregresivo de orden 1 para la varianza condicional y un proceso de media móvil de orden 1 para los errores condicionales. La utilización de este modelo nos proporcionará un modelo ajustado GARCH (1,1), a los retornos diarios del IBEX35 y muestra un resumen del modelo ajustado.

Una vez, ejecutado el modelo, nos disponemos a llevar a cabo, la predicción de la volatilidad del IBEX35 utilizando el modelo GARCH ajustado previamente. Con ello podremos obtener la predicción de la volatilidad del IBEX35 para el próximo año de negociación.

Gráfico 3 Predicción de la volatilidad mediante la utilización del modelo GARCH (1,1)

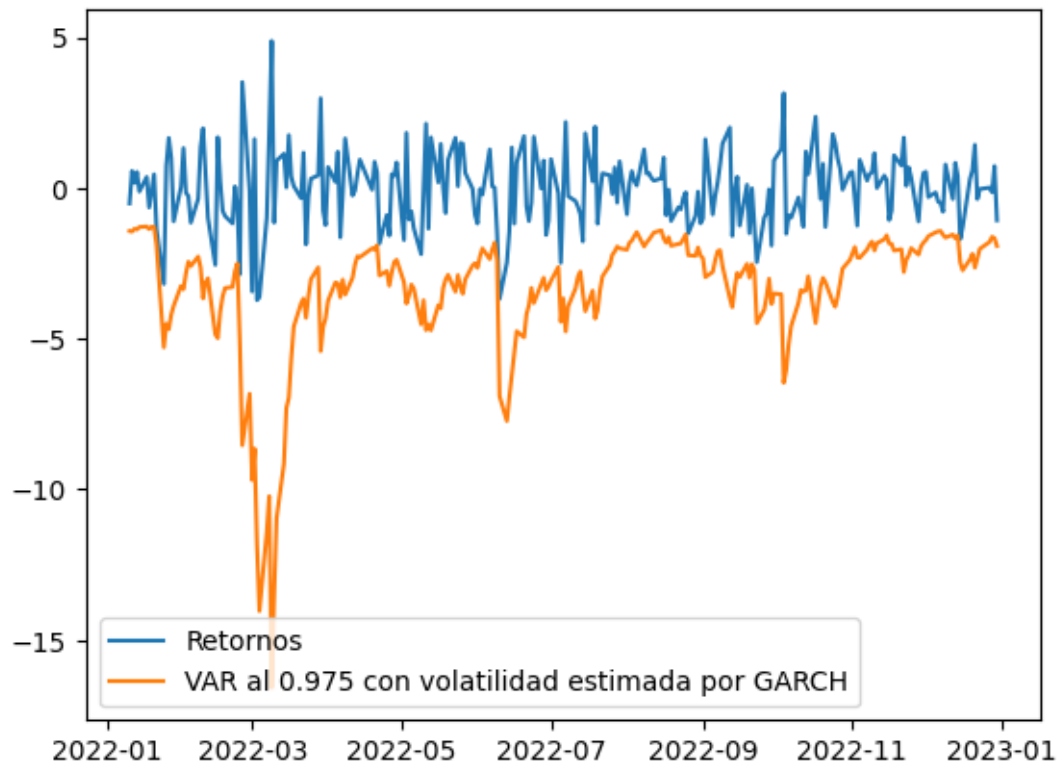


(Fuente: elaboración propia en Python)

El siguiente paso, llevaremos a cabo el cálculo del error cuadrático medio (RMSE) entre la volatilidad realizada y la predicción de la volatilidad del modelo GARCH ajustado, para comparar con el modelo que utilizaremos una técnica de *machine learning*. En este caso, el dato obtenido es un error cuadrático medio de 0.1194, cuanto más cercano a 0, mejor será la predicción del modelo.

Por último, del modelo y del tratamiento de datos anterior, nos disponemos a calcular VaR con un nivel de confianza del 97,5%.

Gráfico 4 cálculo del VaR con un nivel de confianza del 97,5% con el modelo GARCH (1,1)



(Fuente: elaboración propia en Python)

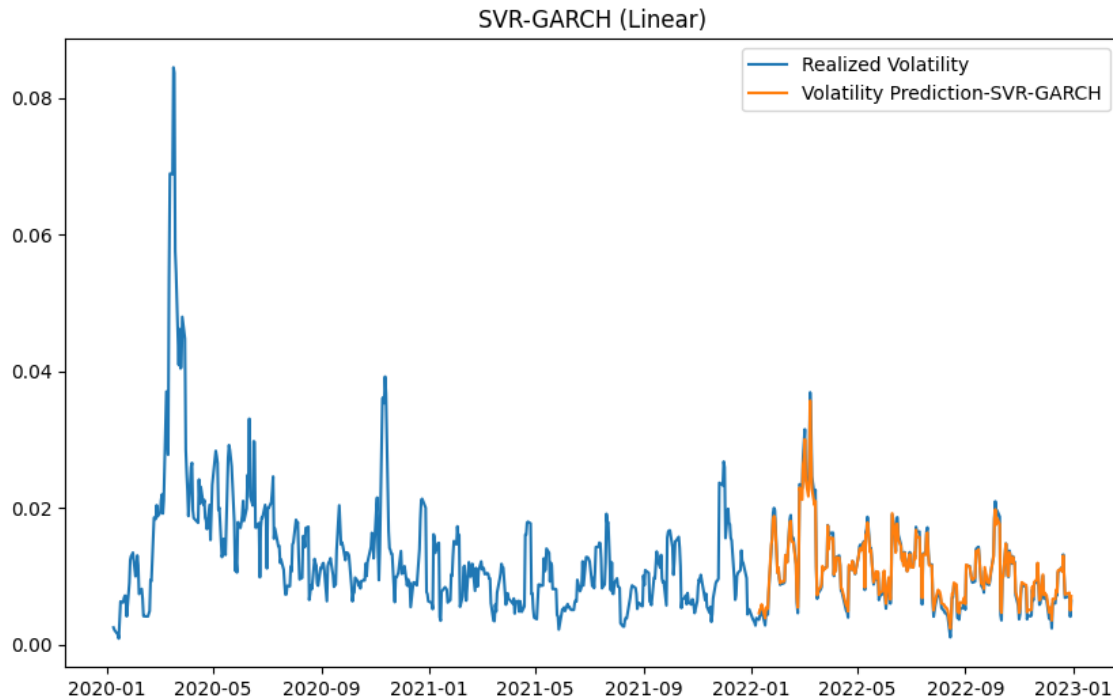
d. Modelo B

Para este segundo modelo utilizaremos el modelo Regresión de Vectores de Soporte (SVR, por sus siglas en inglés) es un algoritmo de aprendizaje automático utilizado para realizar tareas de regresión. A diferencia de otros métodos de regresión, SVR utiliza un enfoque basado en vectores de soporte para encontrar una función que se ajuste a los datos de entrenamiento y sea capaz de hacer predicciones precisas en nuevos datos.

Una vez, llevada a cabo la transformación de los datos para ejecutar el modelo, nos disponemos a utilizar tres modelos de regresión de vectores de soporte (SVR) con diferentes tipos de Kernel (es una función matemática que se utiliza para transformar los datos de entrada en un espacio de características de mayor dimensión para que se puedan separar mejor las clases): polinómico, lineal y de función de base radial (RBF).

Este modelo lo compararemos con el modelo GARCH, en el cual comprobaremos la capacidad del modelo de predecir los datos con las diferentes funciones:

Gráfico 5 Predicción de la volatilidad mediante la utilización del modelo SVR

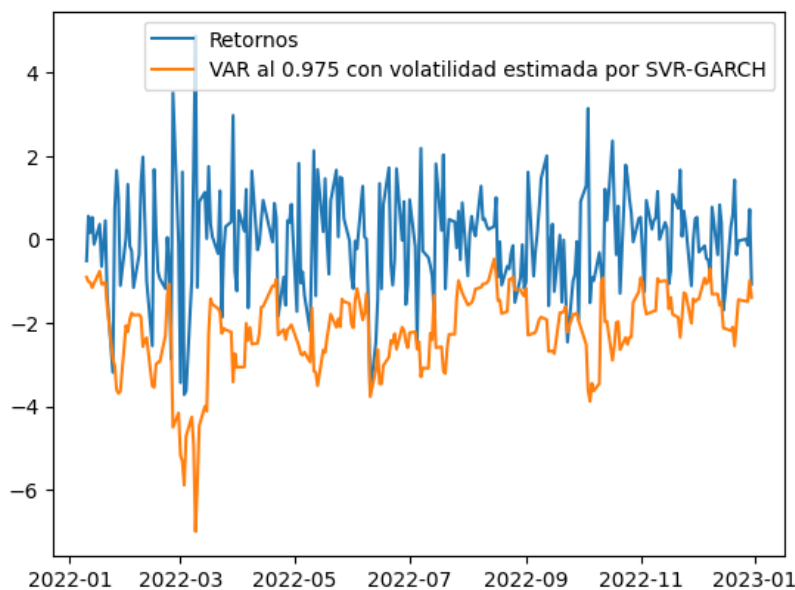


(Fuente: elaboración propia en Python)

El siguiente paso, llevaremos a cabo el cálculo del error cuadrático medio (RMSE) entre la volatilidad realizada y la predicción de la volatilidad del modelo GARCH ajustado, para comparar con el modelo que utilizaremos una técnica de *machine learning*. En este caso, el dato obtenido es un error cuadrático medio de 0.000662, cuanto más cercano a 0, mejor será la predicción del modelo.

Por último, del modelo y del tratamiento de datos anterior, nos disponemos a calcular VaR con un nivel de confianza del 97,5%.

Gráfico 5 cálculo del VaR con un nivel de confianza del 97,5% con el modelo SVR



(Fuente: elaboración propia en Python)

5. Conclusiones

A partir de la elaboración de los diferentes modelos para el cálculo del VaR, podemos definir las siguientes conclusiones:

Capacidad para capturar relaciones no lineales: SVR es un modelo de aprendizaje automático que puede capturar relaciones no lineales entre las características de entrada y la variable objetivo. En el contexto del cálculo del VaR, los rendimientos financieros a menudo exhiben patrones no lineales y dependencias no lineales entre las variables, como podemos observar al tener dentro de nuestra muestra los años de gran volatilidad debido a la pandemia del COVID. SVR tiene la capacidad de modelar estas relaciones no lineales y, por lo tanto, proporcionar estimaciones más precisas del VaR en comparación con GARCH, que se enfoca principalmente en la volatilidad y tiene dificultades para capturar patrones no lineales en los datos.

Flexibilidad en la modelización: SVR es un método no paramétrico que no asume ninguna distribución particular para los datos. Esto lo hace más flexible en la captura de patrones complejos y no lineales en los datos financieros. Por otro lado, GARCH asume una estructura específica para la volatilidad condicional y lo hace menos flexible en la adaptación a diferentes condiciones de mercado.

Mejor rendimiento en datos no estacionarios: Al tratarse de datos financieros que exhiben propiedades de no estacionalidad, como cambios en la volatilidad a lo largo del tiempo o tendencias no lineales, el modelo GARCH tiene dificultades para adaptarse a estas dinámicas. SVR, por otro lado, es más preciso para abordar datos no estacionarios, ya que no se basa en supuestos específicos sobre la estructura de los datos.

6. Bibliografía

Fuente: Universidad Autónoma de Madrid (UAM) - Instituto de Ingeniería del Conocimiento

<https://cleverdata.io/conceptos-basicos-machine-learning/>

Shelley Pursell (2023, 31 de enero). <https://blog.hubspot.es/marketing/machine-learning>

<https://www.apd.es/algoritmos-del-machine-learning/>

<https://datascientest.com/es/machine-learning-definicion-funcionamiento-usos>

<https://kghm.com/es/inversionistas/gestion-de-riesgos/riesgo-de-mercado>

1 <https://www.becas-santander.com/es/blog/riesgos-financieros.html>

1 <https://economipedia.com/definiciones/frontera-carteras-eficiente.html>

Holton, L. (2009). “Is Markowitz wrong? Market turmoil fuels nontraditional approaches to managing investment risk”. *Journal of Financial Planning*, 22(1), pp. 20-26.

Hull, J. (2012). *Risk management and financial institutions*,+ Web Site. (Vol. 733). Hoboken: John Wiley & Sons.

Hull, J. (2013). *The Credit Crisis of 2007 and Its Implications for Risk Management*. In M. Pinedo Y I. Walter (eds.). *Global Asset Management*. (Pp. 93-106). London: Palgrave Macmillan.

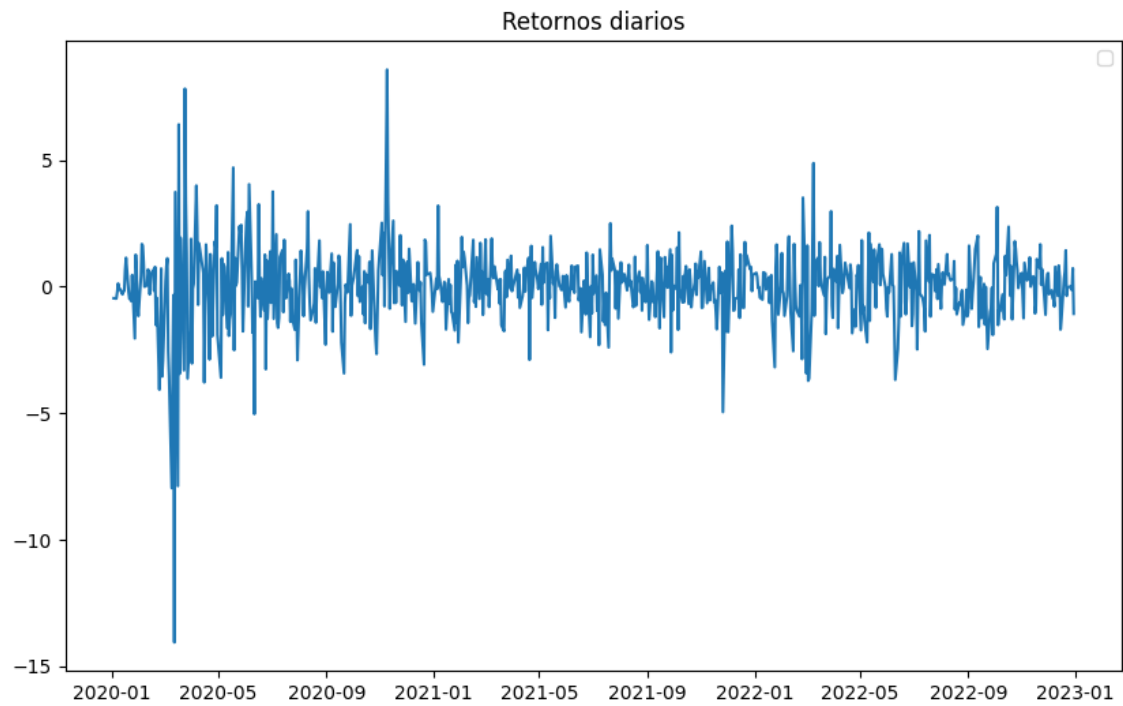
7. Anexos

```
8. import numpy as np
9. from scipy.stats import norm
10. import scipy.optimize as opt
11. import yfinance as yf
12. import pandas as pd
13. import datetime
14. import time
15. from arch import arch_model
16. import matplotlib.pyplot as plt
17. from numba import jit
18. from sklearn.metrics import mean_squared_error as mse
```

```
stocks = '^IBEX'
start = datetime.datetime(2020, 1, 1)
end = datetime.datetime(2023, 1, 1)
ibex35 = yf.download(stocks, start=start, end=end, interval='1d') #Datos diarios del IBEX35
```

```
ret = 100* ibex35.pct_change()[1:]['Adj Close']
plt.figure(figsize=(10,6))
plt.plot(ret)
plt.title('Retornos diarios', fontsize=12)
plt.legend()
```

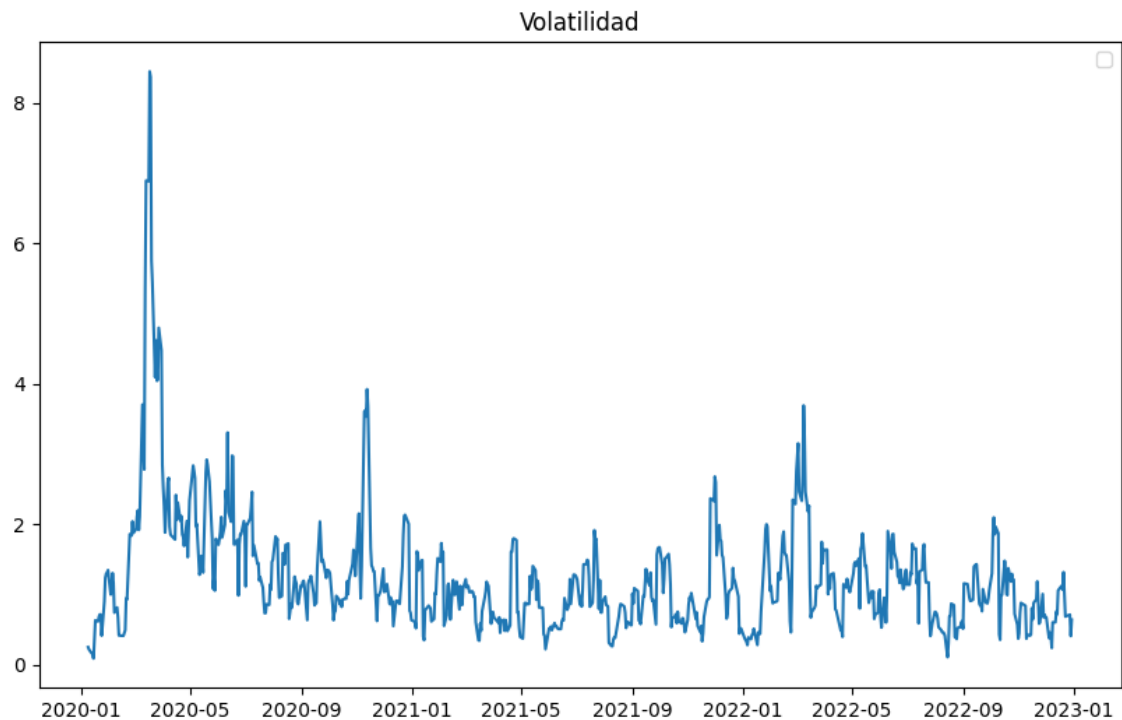
```
plt.show()
```



```
realized_vol = ret.rolling(5).std()
plt.figure(figsize=(10,6))
plt.plot(realized_vol) #Volatilidad (Realized Volatility) con ventana de 5 días
plt.title('Volatilidad', fontsize=12)
plt.legend()

plt.show()

[<matplotlib.lines.Line2D at 0x2039b7eab30>]
```



```
garch = arch_model(ret, mean='zero', vol='GARCH', p=1, o=0, q=1).fit(displ='off')
print(garch.summary())
```

```

Zero Mean - GARCH Model Results
=====
Dep. Variable:          Adj Close    R-squared:              0.000
Mean Model:            Zero Mean    Adj. R-squared:         0.001
Vol Model:             GARCH        Log-Likelihood:        -1289.98
Distribution:          Normal       AIC:                   2585.97
Method:                Maximum Likelihood    BIC:                   2599.90
                                     No. Observations:     769
Date:                  Wed, Jun 07 2023    Df Residuals:          769
Time:                  16:17:17    Df Model:               0
Volatility Model
=====

```

	coef	std err	t	P> t	95.0% Conf. Int.
omega	0.1168	4.191e-02	2.786	5.334e-03	[3.463e-02, 0.199]
alpha[1]	0.1812	4.920e-02	3.683	2.304e-04	[8.478e-02, 0.278]
beta[1]	0.7727	3.743e-02	20.644	1.114e-94	[0.699, 0.846]

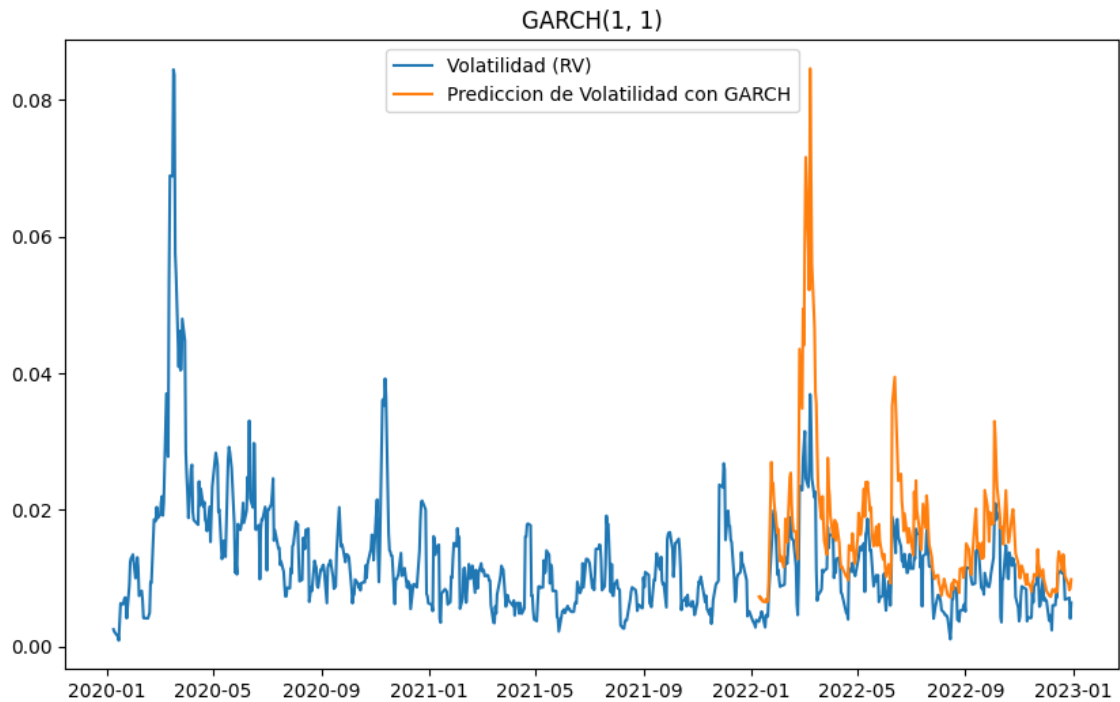
Covariance estimator: robust

```
n = 252 #Predicción de un año
split_date = ret.iloc[-n:].index
forecast = garch.forecast(start=split_date[0])
forecast_garch = forecast
```

```

plt.figure(figsize=(10,6))
plt.plot(realized_vol /100, label='Volatilidad (RV)')
plt.plot(forecast_garch.variance / 100, label='Prediccion de Volatilidad con GARCH')
plt.title('GARCH(1, 1)', fontsize=12)
plt.legend()
plt.show()

```



```

rmse_garch = np.sqrt(mse(realized_vol[-n:] / 100, np.sqrt(forecast_garch.variance.iloc[-
len(split_date):] / 100)))
print(f'RMSE de GARCH(1, 1) = {rmse_garch}') #Raiz del error cuadrático medio

```

```

from sklearn.svm import SVR
from scipy.stats import uniform as sp_rand
from sklearn.model_selection import RandomizedSearchCV

```

```

realized_vol = ret.rolling(5).std()
realized_vol = pd.DataFrame(realized_vol)
realized_vol.reset_index(drop=True, inplace=True)

```

```

returns_svm = ret ** 2
returns_svm = returns_svm.reset_index()
del returns_svm['Date']

```



```
X = pd.concat([realized_vol, returns_svm], axis=1, ignore_index=True) #Realised
volatility y Returns SVM son los parámetros q y p de GARCH, respectivamente
X = X[4:].copy()
X = X.reset_index()
X.drop('index', axis=1, inplace=True)
```

```
realized_vol = realized_vol.dropna().reset_index()
realized_vol.drop('index', axis=1, inplace=True)
```

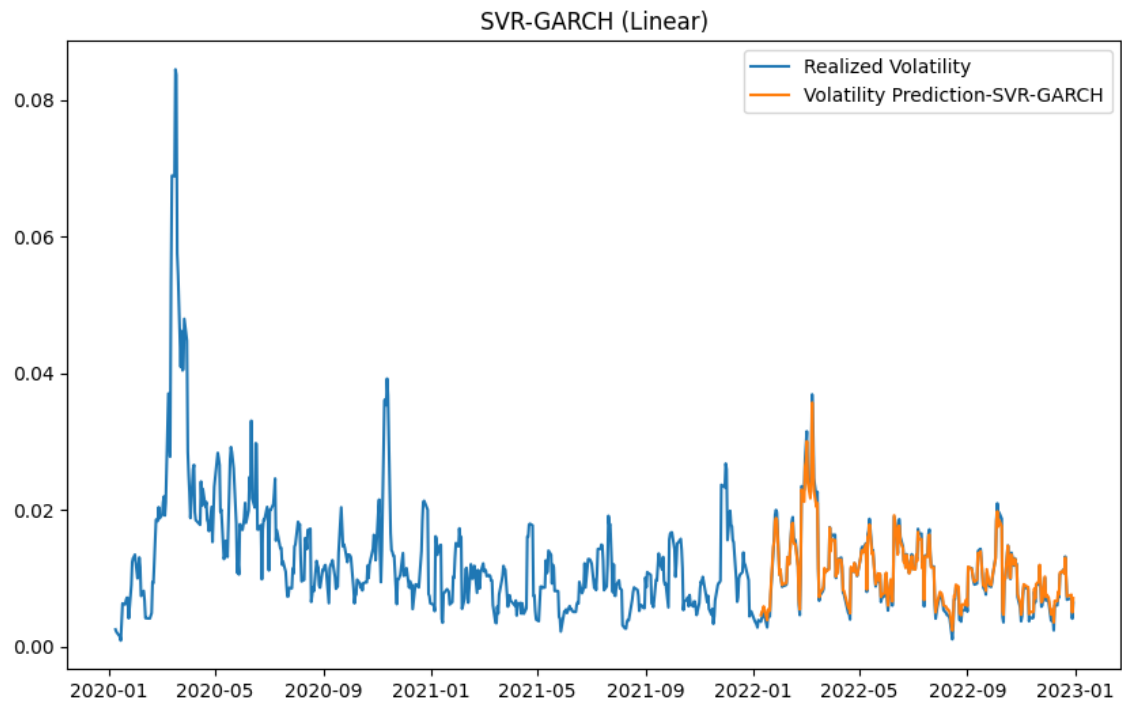
```
svr_poly = SVR(kernel='poly', degree=2) #Polynomial Kernel
svr_lin = SVR(kernel='linear') #Linear Kernel
svr_rbf = SVR(kernel='rbf') #Radial Basis Function Kernel
```

```
para_grid = {'gamma': sp_rand(), 'C': sp_rand(), 'epsilon':
sp_rand()} #Hiperparámetros del Linear Kernel
clf = RandomizedSearchCV(svr_lin, para_grid) #Optimización de
Hiperparámetros
clf.fit(X.iloc[:-n].values, realized_vol.iloc[1:-(n-1)].values.reshape(-
1,)) #Entrenamiento del modelo ML
predict_svr_lin = clf.predict(X.iloc[-n:]) #Predicción de la
volatilidad
```

```
predict_svr_lin = pd.DataFrame(predict_svr_lin)
predict_svr_lin.index = ret.iloc[-n:].index
```

```
realized_vol.index = ret.iloc[4:].index
```

```
plt.figure(figsize=(10, 6))
plt.plot(realized_vol / 100, label='Realized Volatility')
plt.plot(predict_svr_lin / 100, label='Volatility Prediction-SVR-GARCH')
plt.title('SVR-GARCH (Linear)', fontsize=12)
plt.legend()
plt.show()
```



```
mse_svr = np.sqrt(mse(realized_vol.iloc[-n:] / 100, predict_svr_lin / 100))
print('RMSE de SVR con Linear Kernel = {:.6f}'.format(rmse_svr))
```

conclusion: RMSE de SVR con Linear Kernel = 0.000662

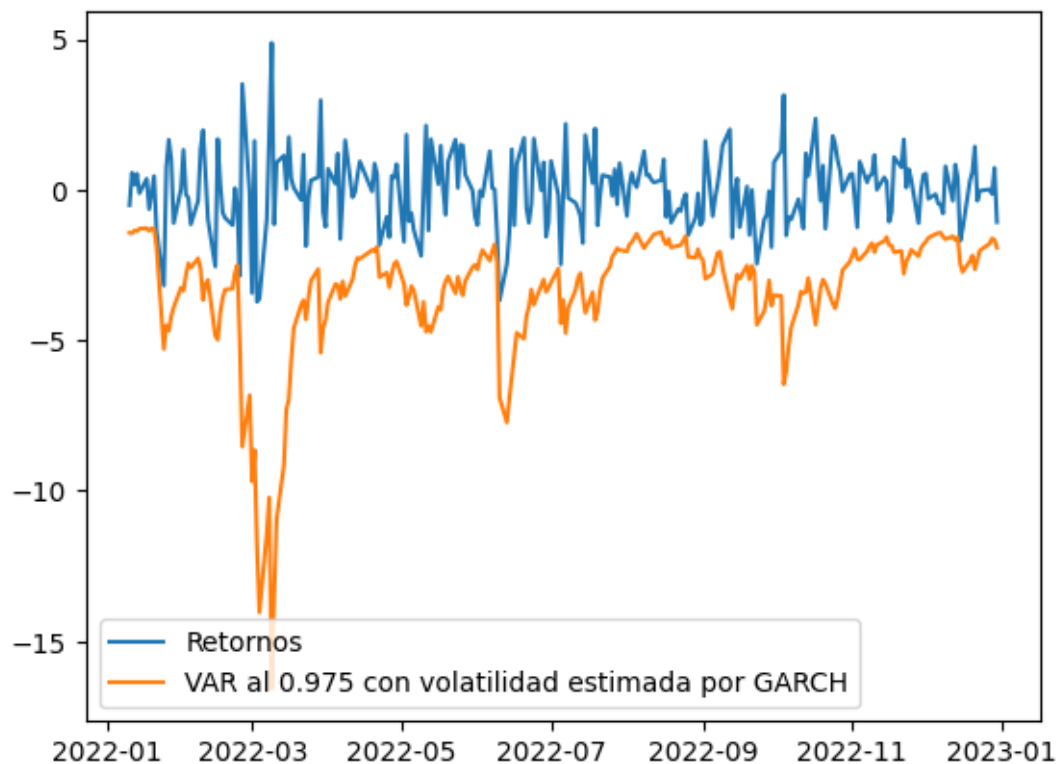
VALUE AT RISK 97.5% con GARCH

```
from scipy.stats import norm
```

```
vol_GARCH = (forecast_garch.variance).dropna()
var_GARCH = pd.DataFrame(norm.ppf(0.975, 0, vol_GARCH)*-1) #Value at Risk
Paramétrico con Distribución Normal
var_GARCH.index = vol_GARCH.index
```

```
plt.plot(ret.iloc[-n:], label = 'Retornos')
plt.plot(var_GARCH, label = 'VAR al 0.975 con volatilidad estimada por GARCH')
plt.legend()
```

<matplotlib.legend.Legend at 0x2039bd007c0>



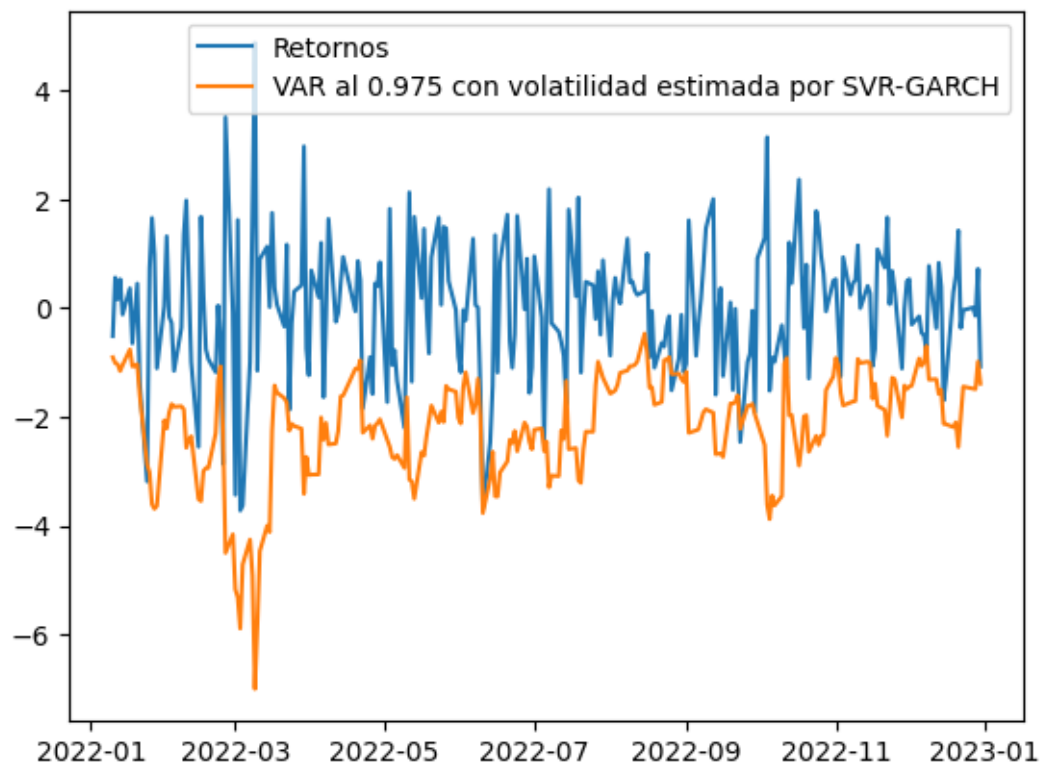
```
exc_GARCH = []
for i in range(0, len(var_GARCH)):
    exc_GARCH.append(ret.iloc[-n:].values[i] > var_GARCH.values[i])
exc_GARCH.count(False) / len(var_GARCH) #Porcentaje de excepciones (Pérdidas
por debajo del VAR)
```

Resultado= 0.0

```
var_SVR = pd.DataFrame(norm.ppf(0.975, 0, predict_svr_lin)*-1)
var_SVR.index = predict_svr_lin.index
```

```
plt.plot(ret.iloc[-n:], label = 'Retornos')
plt.plot(var_SVR, label = 'VAR al 0.975 con volatilidad estimada por SVR-GARCH')
plt.legend()
```

<matplotlib.legend.Legend at 0x2039bd5e230>



```
exc_SVR = []
for i in range(0, len(var_SVR)):
    exc_SVR.append(ret.iloc[-n:].values[i] > var_SVR.values[i])
exc_SVR.count(False) / len(var_SVR) #Porcentaje de excepciones (Pérdidas por debajo del VAR)
```

Resultado = 0.027777777777777776