UNIVERSIDAD
NACIONAL
DE COLOMBIA

FACULTAD DE CIENCIAS

DEPARTAMENTO DE ESTADÍSTICA

PREDICCIÓN SERIE TES VS SPREADS Y RENDIMIENTOS

Trabajo de grado: Segundo informe

Autor:

Juan Gabriel Carvajal Negrete

Enero 2025

Contextualización del problema.

En el mercado financiero, los **spreads y los títulos de Tesorería (TES)** son conceptos fundamentales que ayudan a los inversores a evaluar el riesgo, la liquidez y la rentabilidad de sus inversiones. Este informe proporciona una visión general de estos conceptos y su importancia en la toma de decisiones financieras.

Títulos de Tesorería (TES)

Los TES son instrumentos de deuda emitidos por el Gobierno Nacional de Colombia para financiar su déficit fiscal. Estos títulos son considerados una inversión segura debido a su respaldo gubernamental y se negocian tanto en el mercado primario como en el segundario.

Características de los TES

- **Emisión:** Emitidos por el Ministerio de Hacienda y administrados por el Banco de la Republica.
- **Tipos:** Incluyen TES en pesos Y TES en unidades de valor real (UVR), con tasas de interés fijas o variables.
- **Plazos:** varían desde uno hasta diez años
- **Mercado:** Se emiten en subastas públicas y se negocia en el mercado secundario.

Spreads en el Mercado Financiero

El termino Spread se refiere a la diferencia entre dos precios, tasas o rendimiento. En el contexto de los Tes, los spreads pueden contener información valiosa sobre la percepción del riesgo y la liquidez del mercado

Tipos de Spread

- **Spread de rendimiento:** Diferencia entre el rendimiento de los TES y los bonos.
- **Spread de crédito:** Diferencia entre los TES y los bonos corporativos con el mismo vencimiento.
- **Spread de liquidez:** Refleja la diferencia en la liquidez entre los TES y otros instrumentos financieros.

Conocer y analizar los spreads en relación con los TES es crucial por varias razones

- **Evaluación de los riesgos:** Los Spread de rendimiento ayudan a los inversores a evaluar el riesgo relativo de diferentes inversiones.
- **Decisiones de inversión:** Los inversores pueden usar estos Spreads para tomar decisiones informadas, buscando un equilibrio entre riesgo y rendimiento.
- **Indicador económico:** Los Spreads sirven como indicadores de la salud económica y la percepción del riesgo en el mercado financiero.

Descripción de las variables.

lN: logaritmo de años.

OIS: Variable que muestra hacia dos años como está la política monetaria.

Años: AUM de la industria en fondos.

Volatilidad TES: Volatilidad de la deuda pública.

RF_COL_CDS01y_RR_US_DT: Tasa de recovery Rate a un año para Colombia.

RF_COL_CDS02y_RR_US_DT: Tasa de recovery Rate a dos años para Colombia.

RF_COL_CDS05y_RR_US_DT: Tasa de recovery Rate a cincio años para Colombia.

RF_COL_CDS10y_RR_US_DT: Tasa de recovery Rate a diez años para Colombia.

RF_COL_CDS01y_DP_US_D_T: Probabilidad de incumplimiento a un año para Colombia.

RF_COL_CDS02y_DP_US_D_T: Probabilidad de incumplimiento a dos años para Colombia.

RF_COL_CDS05y_DP_US_D_T: Probabilidad de incumplimiento a cinco años para Colombia.

RF_COL_CDS10y_DP_US_D_T: Probabilidad de incumplimiento a diez años para Colombia.

RF_COL_CDS01y_USD_T: Tasa de CDS a un año para Colombia.

RF_COL_CDS02y_USD_T: Tasa de CDS a dos años para Colombia.

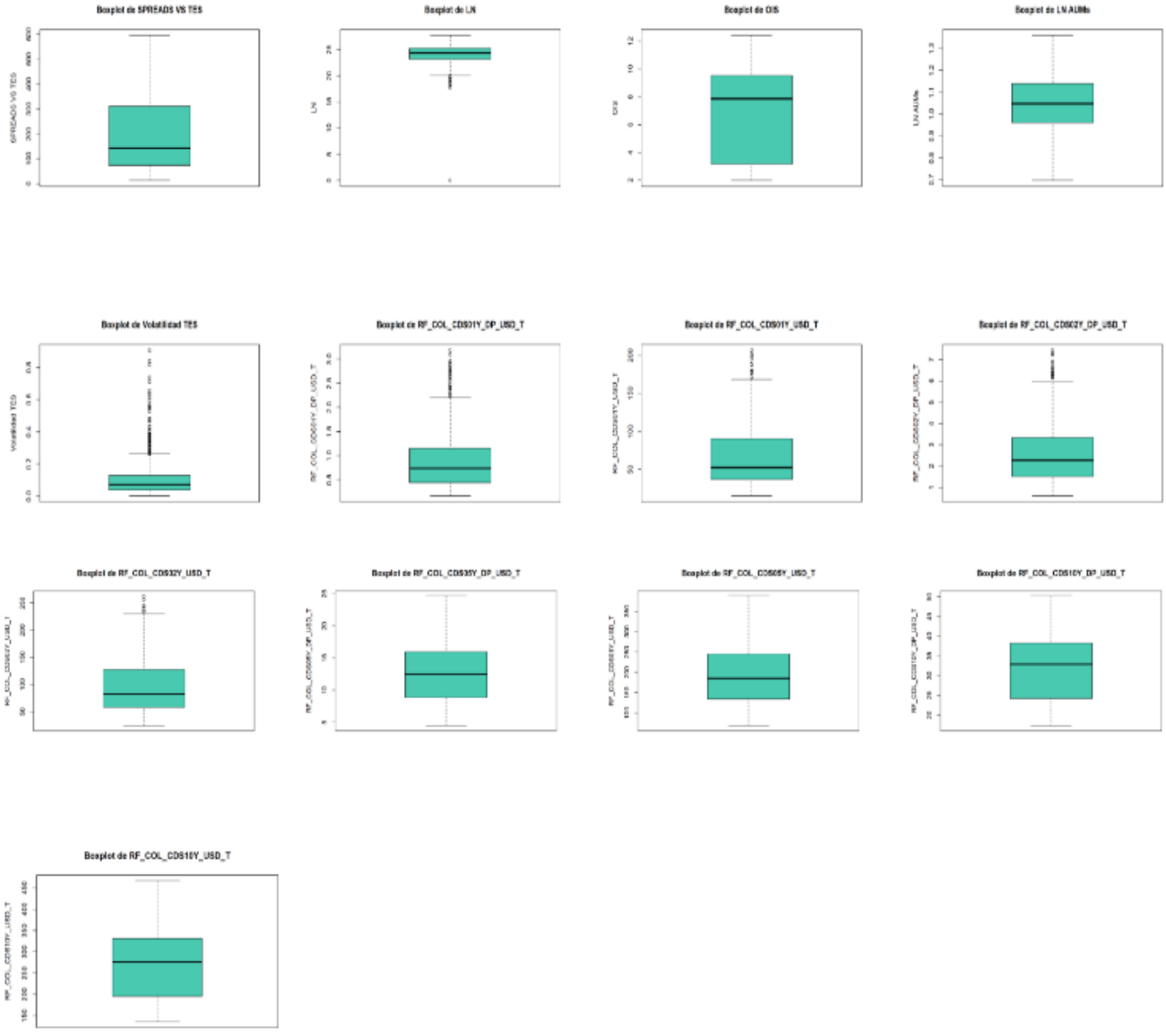
RF_COL_CDS05y_USD_T: Tasa de CDS a cinco años para Colombia.

RF_COL_CDS10y_USD_T: Tasa de CDS a diez años para Colombia.

Análisis descriptivo de las variables

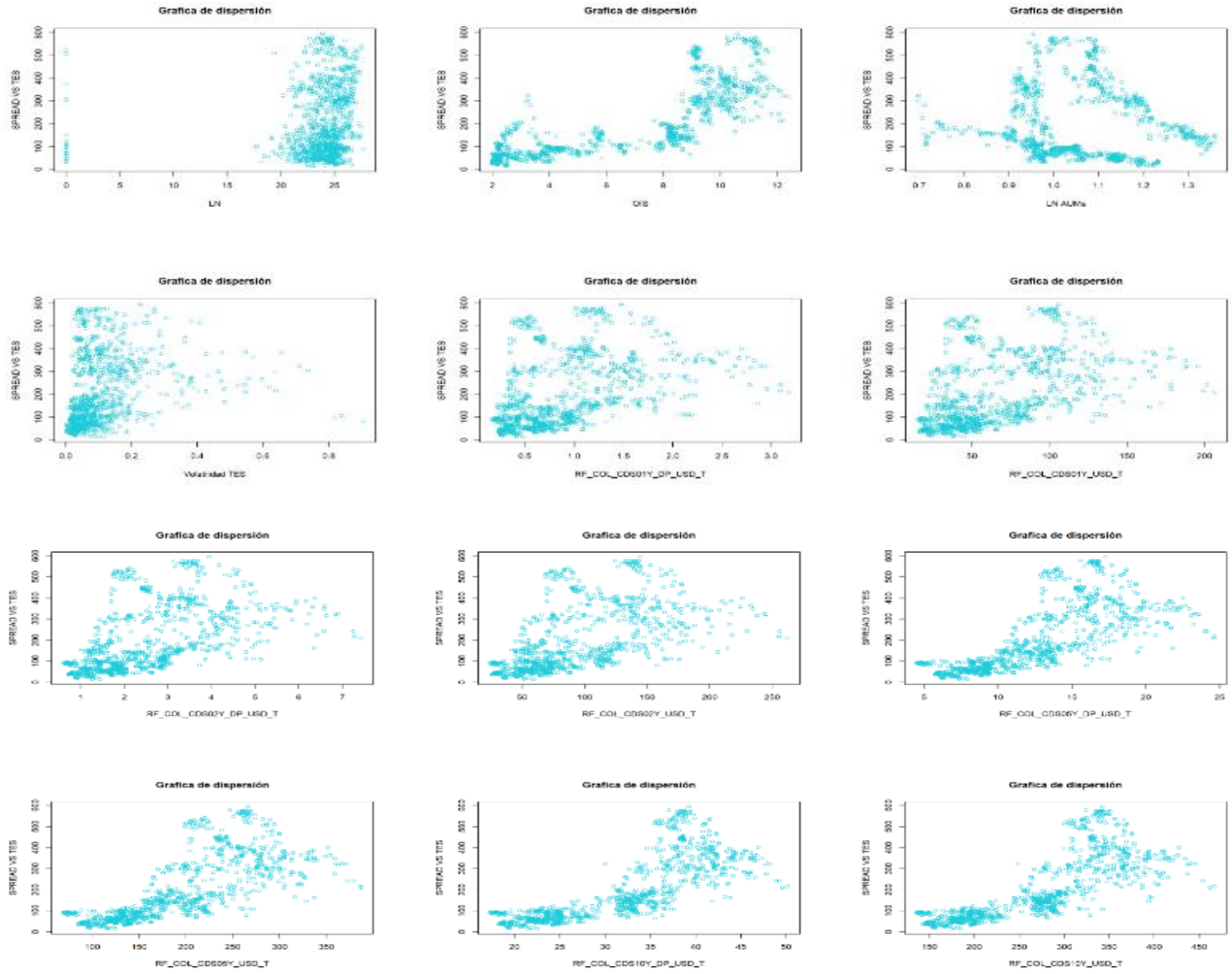
Con estos gráficos lo que buscamos ver es el comportamiento distribucional de cada variable dentro de nuestro conjunto de datos, importante notar que existen variables con un solo valor (Variables constantes), estas variables a partir de aquí nos van a dejar de importar, pues son variables que no van a aportar variabilidad a nuestros modelos, en general notamos que existe pequeños grupos de nuestras variables que presentan comportamientos muy similares hablando distribucionalmente.

Continuemos con gráficos descriptivos individuales que nos ayuden a entender un poco más de nuestras variables y seguir con nuestras observaciones.

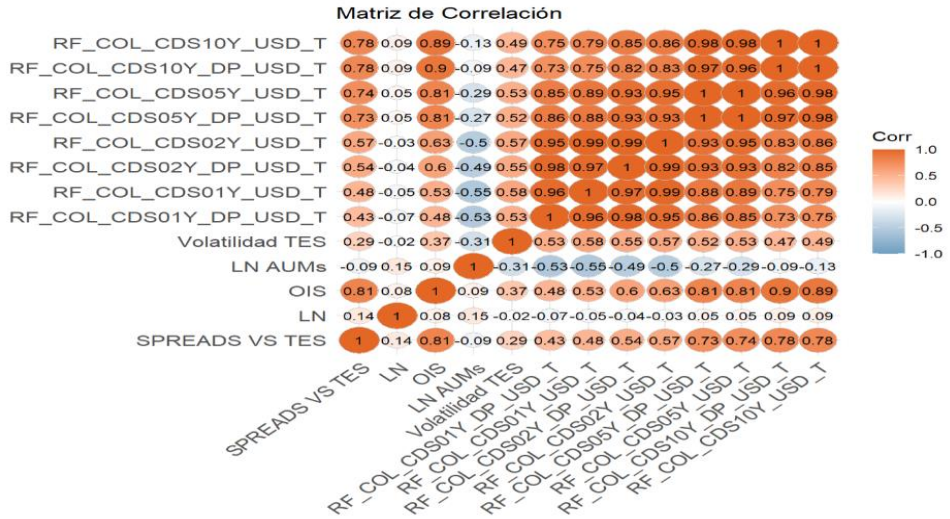


En los gráficos de boxplot notamos que gran parte de nuestras variables presentan valores distantes (outlier) pero cabe resaltar que las variables de nuestro trabajo son variables obtenidas de la aplicación de información **Refinitiv.** Los datos de refinitv suelen contener información financiera detallada y especifica del mercado, en resumen, Eliminar o amputar variables con outlier puede tener consecuencias significativas en el análisis de datos.

Continuemos ahora viendo como nuestras variables se relacionan con nuestra variable de interés.



Notamos que muchas de las variables están fuertemente relacionadas con nuestra variable respuesta, sin embargo, en dichas variables su nube de puntos es muy parecida, lo que nos puede indicar que dentro de las variables de nuestro trabajo existe una correlación entre ella, analicemos esto con la matriz de correlación.



Una vez conocidas las variables y de haber realizado un análisis descriptivos individual y en conjunto de todas las variables, proceso en el que se fueron descartando variables que simplemente no aportaban nada de variabilidad para los modelos aplicar ya que eran variables constantes, además se conocieron las variables que mejor están relacionadas con nuestra variable respuesta, adicionalmente notamos que en nuestros conjunto de datos existe un gran número de variables correlacionadas es decir tenemoss multicolinealidad en nuestros datos. Para intentar reducir el número de variables se realizó el método **Low variance filter.**

Reducción de dimensionalidad (Low Variance Filter)

El Low Variance Filter es una técnica de selección de características que se utiliza en el aprendizaje automático para identificar y eliminar características de baja varianza del conjunto de datos. Esta técnica se utiliza para mejorar el rendimiento del modelo al reducir la cantidad de características utilizadas para entrenarlo y para eliminar las características que tienen poco o ningún poder discriminatorio.

El filtro de baja varianza funciona calculando la varianza de cada característica del conjunto de datos y eliminando las características que tienen una varianza por debajo de un umbral determinado. Esto se hace porque las características con baja varianza tienen poco o ningún poder discriminatorio y es poco probable que sean útiles para predecir la variable objetivo.

Los pasos necesarios para implementar el filtro de baja varianza son los siguientes:

- Normalizar y calcular la varianza de cada una de las características en el conjunto de datos.
- Establezca un umbral para la variación de las características.
- Eliminar las características que tengan una variación por debajo del umbral.
- Utilice las funciones restantes para entrenar el modelo de aprendizaje automático.

Para nuestro caso el umbral que se fijo fue de 0.0002 y las variables que presentaron mayor varianza fueron: **OIS, Volatilidad TES, RF_COL_CDS01Y_DP_USD_T ,RF_COL_CDS01Y_USD_T, RF_COL,CDS02Y_DP_USD_T.**

Importante: En este trabajo no es importante solamente predecir el comportamiento de la serie del spread vs tes, también es importante predecir el comportamiento de los rendimientos de la serie, es decir la serie diferenciada de un valor con su pasado de esta forma se pierde el primer registro de cada variable en nuestro conjunto de datos.

De esta forma los modelos que se apliquen a la serie normal también se aplicaran a la serie de rendimientos.

Empecemos ajustando nuestros primeros modelos de regresión con las variables resultantes luego de aplicar el método de Low variance filter.

Regulación

La regularización o shrinkage, consiste en ajustar el modelo incluyendo todos los predictores, pero aplicando una penalización que fuerce a que las estimaciones de los coeficientes de regresión tiendan a cero. Con esto se intenta evitar overfitting, reducir varianza, atenuar el efecto de la correlación entre predictores y minimizar la influencia en el modelo de los predictores menos relevantes. Por lo general, aplicando regularización se consigue modelos con mayor poder predictivo (generalización). Tres de los métodos de regularización más empleados son Ridge, Lasso y Elastic net.

Modelo de Regresión Ridge.

La regularización *Ridge* penaliza la suma de los coeficientes elevados al cuadrado ($\|\beta\|^2 = \sum_{d=1}^{p} \beta_j^2$). A esta penalización se le conoce como L2 y tiene el efecto de reducir de forma proporcional el valor de todos los coeficientes del modelo, pero sin que estos lleguen a cero. El grado de penalización esta controlado por el hiperparámetro λ.cuando $\lambda = 0$, la penalización es nula y el resultado es equivalente al de un modelo lineal por mínimos cuadrados ordinarios (OLS). A medida que λ aumenta, mayor es la penalización y menor el valor de los predictores.

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \text{suma residuos cuadrados} + \lambda\sum_{j=1}^{p}\beta_j^2$$

La principal ventaja de aplicar ridge frente al ajuste por mínimos cuadrados ordinarios (OLS) es la reducción de varianza. Por lo general, en situaciones en las que la relación entre la variable respuesta y los predictores es aproximadamente lineal, las estimaciones por mínimos cuadrados tienen poco bias, pero aún pueden sufrir alta varianza (pequeños cambios en los datos de entrenamiento tienen mucho impacto en el modelo resultante). Este problema se acentúa conforme el número de predictores introducido en el modelo se aproxima al número de observaciones de entrenamiento, llegando al punto en que, si $p > n$ no es posible ajustar el modelo por mínimos cuadrados ordinarios. Empleando un valor adecuado de λ , el método es capaz de reducir la varianza sin apenas aumentar el bias, consiguiendo así un menor error total.

La desventaja del método ridge es que, el modelo final, incluye todos los predictores. Esto es así porque, si bien la penalización fuerza a que los coeficientes tiendan a cero, nunca llegan a ser exactamente cero (solo si $\lambda = \infty$). Este método consigue minimizar la influencia sobre el modelo de los predictores menos relacionados con variables respuesta, pero en el modelo final, van a seguir apareciendo. Aunque esto no supone un problema para la precisión del modelo, si lo es para su interpretación.

Nota: Estos modelos se aplicaron en el software estadístico R, al momento de realizar el backtesting los datos que se dejaron de prueba fueron todos los registros del 2024

Ajuste y predicciones del modelo Redge

Lastimosamente las predicciones de nuestro primer modelo no parecen ser tan precisas es cierto que en el periodo de inicio de nuestros datos de test vienen precedidos de una caída en la serie por lo que al modelo le cuesta seguir este comportamiento este modelo presenta una MSE de 10.010,04.

Modelo de Regresión Lasso

La regulación Lasso penaliza la suma del valor absoluto de los coeficientes 1de regresión ($\|\beta\| = \sum_{j=1}^{p} |\beta_j|$). A esta penalización se le conoce como L1 y tiene el efecto de forzar a que los coeficientes de los predictores tiendan a cero. Dado que un predictor con coeficientes de regresión cero no influye en el modelo, lasso consigue excluir los predictores menos relevantes. Al igual que en el ridge, el grado de penalización está controlado por el hiperparámetro λ.Cuando $\lambda = 0$, el resultado es equivalente al de un modelo lineal por mínimos cuadrados ordinarios. A medida que λ aumenta, mayor será la penalización y más predictores quedaran excluidos.

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}|\beta_j| = \textbf{suma residuos cuadrados} + \lambda \sum_{j=1}^{p}|\beta_j|$$

Comparación Ridge y Lasso

La principal diferencia práctica entre lasso y ridge es que el primero consigue que algunos coeficientes sean exactamente cero, por lo que realiza selección de predictores, mientras que el segundo no llega a excluir ninguno. Esto supone una ventaja notable de lasso en escenarios donde no todos los predictores son importantes para el modelo y se desea que los menos influyentes queden excluidos.

Por otro lado, cuando existen predictores altamente correlacionados (linealmente), ridge reduce la influencia de todos ellos a la vez y de forma proporcional, mientras que lasso tiende a seleccionar uno de ellos, dándole todo el peso y excluyendo al resto. En presencia de correlaciones, esta selección varía mucho con pequeñas perturbaciones (cambios en los datos de entrenamiento), por lo que, las soluciones de lasso, son muy inestables si los predictores están altamente correlacionados.

Para conseguir un equilibrio óptimo entre estas dos propiedades, se puede emplear lo que se conoce como penalización Elastic net, que combina ambas estrategias.

Ajuste y predicciones del modelo Lasso

Notamos el mismo comportamiento en las predicciones del modelo Lasso, están muy alejados de los datos reales, este modelo presenta un MSE de 12.988,50 es un poco más alto que el modelo anterior.

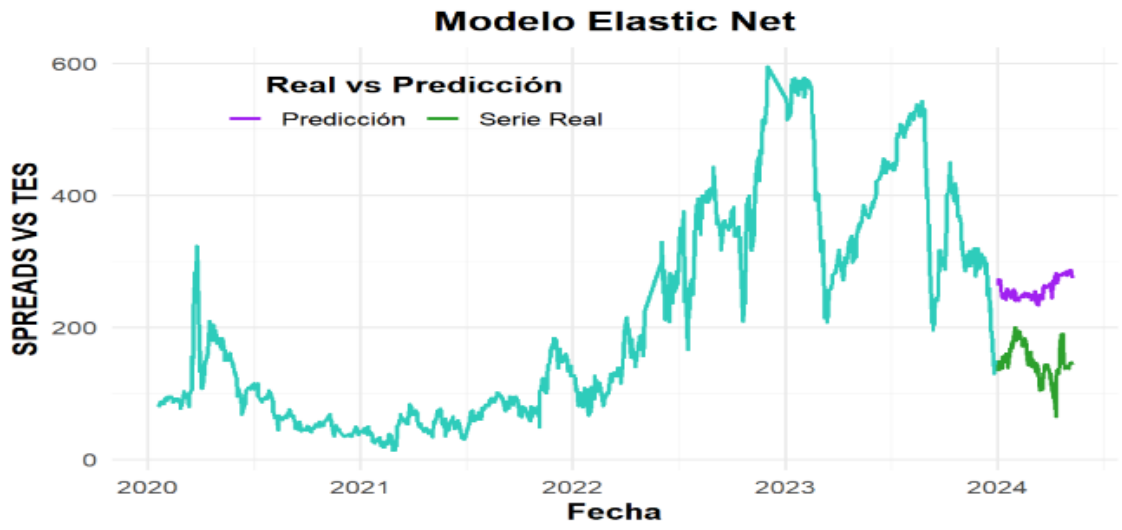
Modelo de regresión Elastic net

Elastic net incluye una regularización que combina la penalización l1 y l2

$(\alpha\lambda \left\|\beta_1\right\| + \frac{1}{2}(1-\alpha)\left\|\beta_2\right\|^2)$ El grado en que influye cada una de las penalizaciones está controlada por el hiperparámetro $\propto$. Su valor está comprendido en el intervalo [0,1]. Cuando $\propto = 0$ se aplica ridge y cuando $\propto = 1$ se aplica lasso. La combinación de ambas penalizaciones suele dar lugar a buenos resultados. Una estrategia frecuentemente utilizada es asignarle casi todo el peso a la penalización L1 ($\propto$ muy próximo a 1) para conseguir seleccionar predictores y un poco a la L2 para dar cierta estabilidad en el caso de que algunos predictores estén correlacionados.

$$\frac{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2}{2n} + \lambda(\alpha\sum_{j=1}^{p}|\beta_j| + \frac{1-\alpha}{2} + \sum_{j=1}^{p}\beta_j^2$$

Ajuste y predicciones del modelo Elastic Net



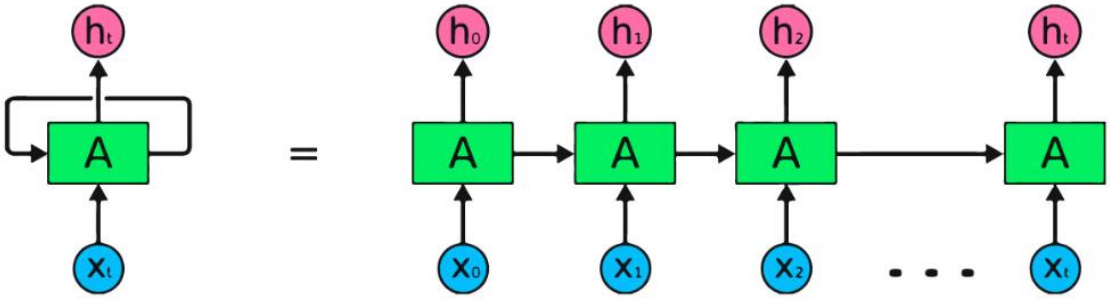
Los tres modelos presentaron predicciones bastante alejadas de las estimaciones reales, este modelo presenta un MSE de 13.003,92. Dentro de los tres modelos ajustados inicialmente el modelo que mejor ajuste presenta es el modelo Redge.

Los modelos de regulación fueron los primeros modelos considerados para intentar dar solución a nuestro problema sin embargo las predicciones de los tres modelos son bastante alejados de la realidad, intentemos tratar este problema con algo mucho más preciso, para este caso vamos a ajustar modelos de redes neuronales recurrentes, a continuación una introducción de esto.

Redes Neuronales Recurrentes

Una red neuronal recurrente (RNN) es el tipo de red neuronal artificial (ANN) que se utiliza en Siri de Apple y en la búsqueda de voz de Google. Las RNN recuerdan entradas pasadas gracias a una memoria interna que es útil para predecir los precios de las acciones y generar texto, transcripciones y traducción automática.

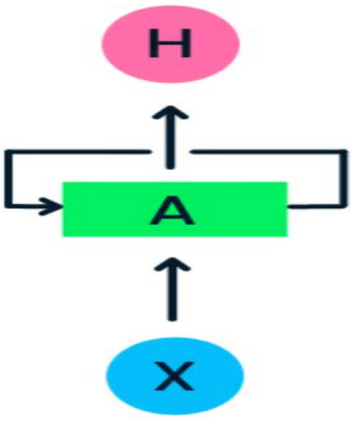
En la red neuronal tradicional, las entradas y las salidas son independientes, mientras que la salida en la RNN depende de elementos anteriores dentro de la secuencia. Las redes neuronales recurrentes también comparten parámetros entre todas las capas de la red. En las redes prealimentadas, hay distintos pesos en cada nodo. Mientras que la RNN comparten los mismos pesos en todas las capas de la red y durante el descenso del gradiente, los pesos y la base se ajustan individualmente para reducir la perdida.



La imagen anterior es una representación sencilla de las redes neuronales recurrentes. Si estamos pronosticando los precios de las acciones utilizando datos sencillos [45,40,38,50,...], cada entrada x_0 a x_t contendrá un valor pasado. Por ejemplo, x_0 tendrá 45, $x_1$tendrá 40, y estos valores se utilizan para predecir el siguiente numero de una secuencia.

Como funcionan las redes neuronales recurrentes

En la RNN, la información circula por el bucle, de modo que salida viene determinada por la entrada actual y las entradas recibidas anteriormente.



La capa de entrada X procesa la entrada inicial y la pasa a la capa intermedia A. La capa intermedia consta de varias capaz ocultas, cada una con sus funciones de activación, sus pesos y sus sesgos. Estos parámetros están estandarizados en toda la capa oculta, de modo que, en lugar de crear varias capas ocultas, crearán una y la repetirán en el bucle.

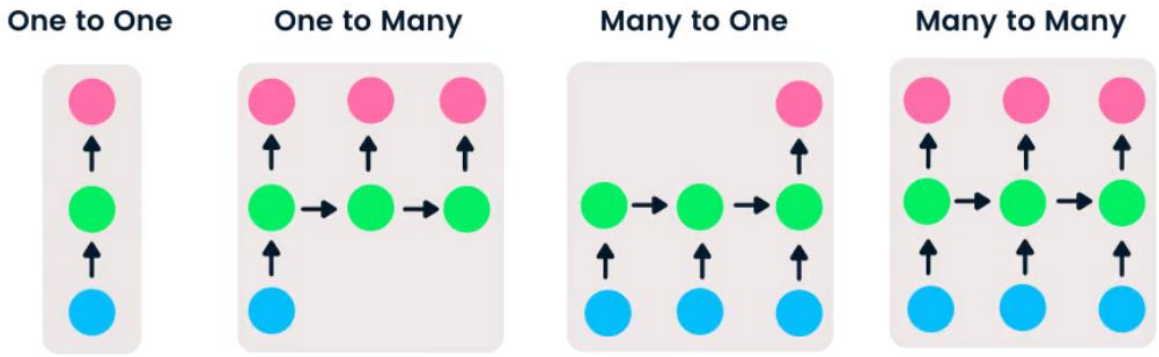
En lugar de utilizar la propagación atrás de sus errores tradicional, las redes neuronales recurrentes utilizan algoritmos de propagación hacia atrás de los errores a través del tiempo (BPTT) para determinar el gradiente. En la propagación hacia atrás de los errores, el modelo ajusta el parámetro calculando los errores de la capa de salida a la de la entrada. La BPTT suma el error en cada paso temporal, ya que la RNN comparte los parámetros en todas las capas.

Tipos de redes neuronales recurrentes

Las redes prealimentadas tienen salida y entradas únicas, mientras que las redes neuronales recurrentes son flexibles ya que se pueden cambiar la longitud de la entrada y las salidas.

Hay cuatro tipos de RNN basados en diferentes longitudes de entradas y salidas.

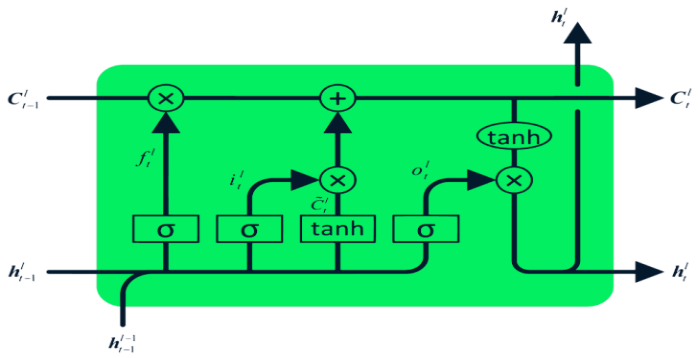
- **Uno a uno** es una red neuronal sencilla. Se suele utilizar para problemas de machine learning que tienen entrada y salida únicas.
- **Uno a muchos** tiene entrada única y varias salidas. Se utiliza para generar pies de foto.
- **Muchos a uno:** toma una secuencia de varias entradas y predice una salida única. Es popular en la clasificación de sentimiento, donde la entrada es texto y la salida es una categoría.
- **Muchos a muchos:** toma varias entradas y salidas. La aplicación más común es la traducción automática.



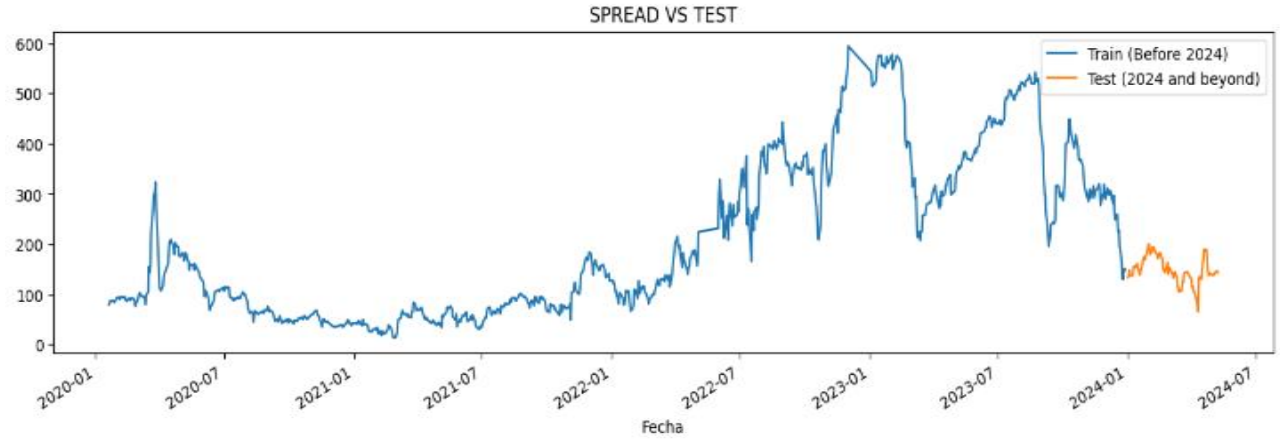
Dentro del proyecto que se está llevando acabo para intentar predecir los Spreads vs TES se implementaron dos redes neuronales recurrentes **Long short-term memory (LSTM) y Unidad recurrente cerrada (GRU)** estos modelos se implementaron en **Python** con la ayuda de la librería **tensorflow y Keras**.

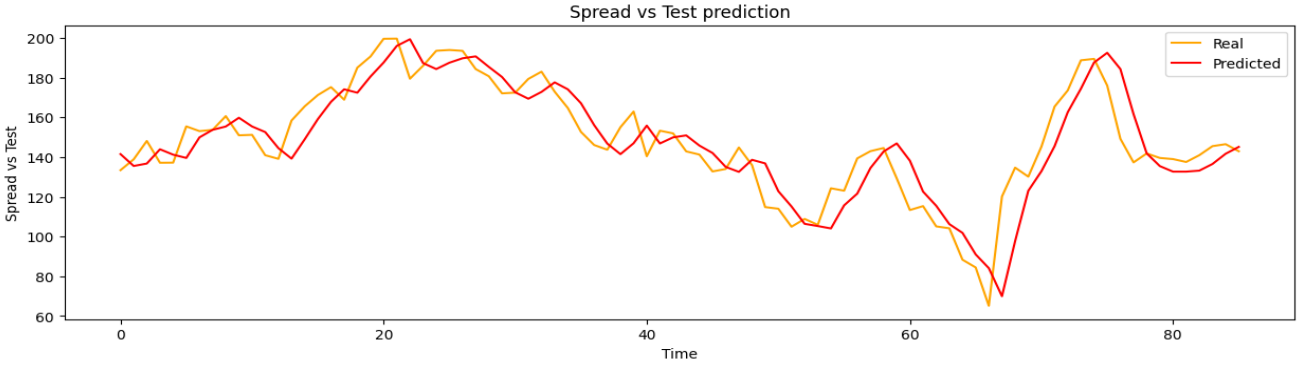
Long short-term memory (LSTM)

La Long short-term memory (LSTM) es el tipo avanzado de RNN, que se diseñó para evitar tanto los problemas de desvanecimiento como los de explosión de gradiente. Al igual que la RNN, la LSTM tiene módulos que se repiten, pero la estructura es diferente. En lugar de tener una capa única de tanh, la LSTM tiene cuatro capas que interactúan y se comunican. Esta estructura de cuatro capas ayuda a la LSTM a conservar la memoria a largo plazo y puede utilizarse en varios problemas secuenciales, como la traducción automática, la síntesis del habla, el reconocimiento del habla y el reconocimiento de la escritura a mano.



Ajuste y predicciones del modelo LSTM



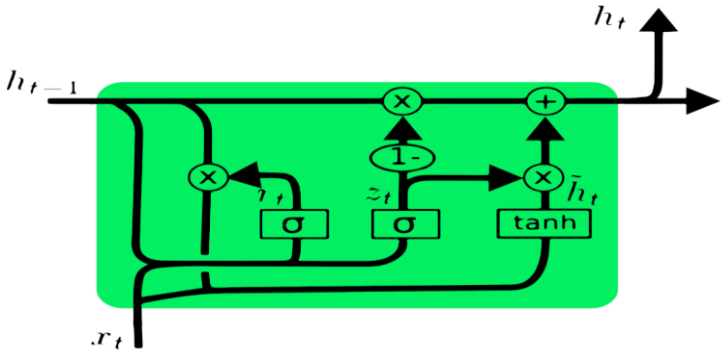


Vemos un gran cambio con respecto a los modelos de regresión, la red neuronal logra capturar el comportamiento de la serie, es importante acá que estos modelos son de uso únicamente predictivo pues no on interpretativos en sus parámetros. Este modelo tiene un MSE de 13.12

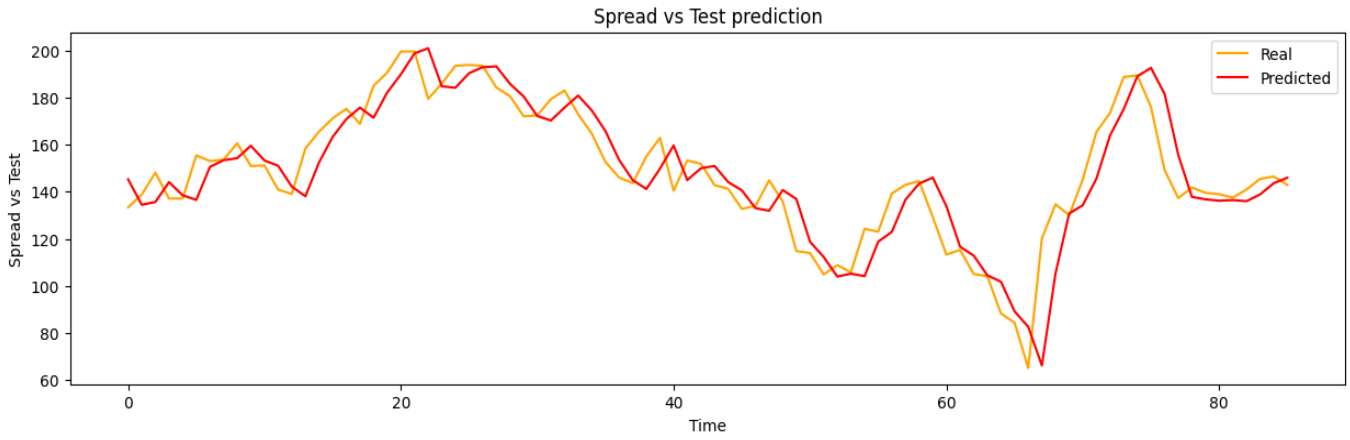
Unidad recurrente cerrada (GRU)

La unidad recurrente cerrada (GRU) es una variación de la LSTM, ya que ambas tienen similitudes de diseño y, en algunos casos, producen resultados parecidos. La GRU utiliza una compuerta de actualización y una compuerta de reinicio para resolver el problema de desvanecimiento de gradiente. Estas compuertas deciden qué información es importante y la pasan a la salida. Las compuertas pueden entrenarse para almacenar información de hace mucho tiempo, sin que se desvanezca con el tiempo y sin eliminar la información irrelevante.

A diferencia de la LSTM, la GRU no tiene estado de celda Ct. Solo tiene un estado oculto ht, y debido a la sencillez de su arquitectura, la GRU tiene un tiempo de entrenamiento menor que los modelos LSTM. La arquitectura GRU es fácil de entender, ya que toma la entrada xt y el estado oculto de la marca temporal anterior ht-1 y da como salida el nuevo estado oculto ht.



Ajuste y predicción del modelo GRU



El modelo presenta un MSE de 12.45, este modelo a pesar de ser mas simple que la red LSTM ajusta mejor las predicciones. Sin duda alguna estos dos últimos modelos son los que deberíamos elegir para intentar predecir el comportamiento de la serie de Spread vs TES.

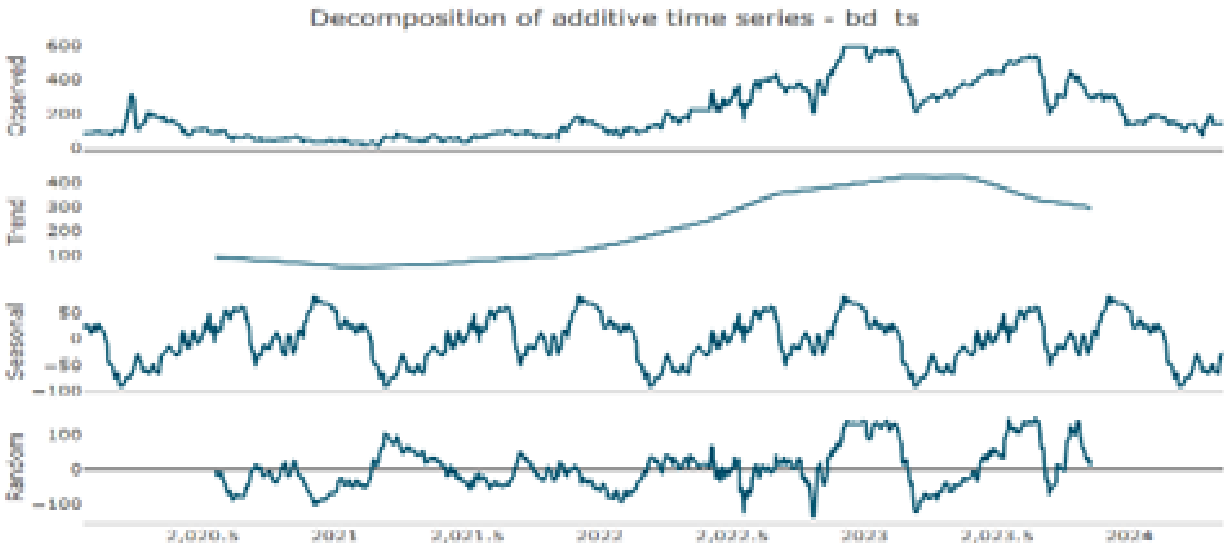
Viendo el problema como una serie de tiempo

Considerando la serie de Spreads vs TES y la de sus rendimientos aprovechando que tenemos una variable Fecha, se decide ver el problema desde la parte de series de tiempo y para esto comenzamos analizando las características de la serie, como su tendencia, estacionariedad, su componente estacional y su ruido aleatorio.

Luego empezamos implementando modelos tradicionales de serie de tiempo como lo son AR(p), MA(q) y ARIMA(p,d,q) esto dependiendo de sus funciones ACF y PACF.

Importante: En este parte nos vamos a enfocar más en la serie de los rendimientos y compararemos el mejor modelo de serie de tiempo obtenido para comparar su error con el mejor modelo de regularización.

Análisis descriptivo de la serie de tiempo



A partir del análisis de la gráfica, podemos inferir que la variable de interés presenta una tendencia a la baja a lo largo del tiempo, lo que sugiere una disminución constante en el Spreas vs TES. Las otras variables muestran patrones más fluctuantes y menos definidos. No se observa una estacionalidad clara en ninguna de las series, lo que indica que no hay patrones repetitivos a intervalos regulares. Sin embargo, es importante considerar que la ausencia de estacionalidad podría deberse a la naturaleza de los datos o a la escala de tiempo utilizada. La componente de residuos, que representa la parte de la serie que no se explica por la tendencia ni la estacionalidad, muestra una alta variabilidad en todas las variables, lo que sugiere la presencia de un componente aleatorio significativo.

Como se mencionó antes también es de interés modelar la serie de rendimientos por los que nos enfocaremos ahora a obtener un modelo para predecir su comportamiento.

Modelos AR(p)

son modelos que se regresan en sí mismos. Es decir, la variable dependiente y la variable explicativa son la misma con la diferencia que la variable dependiente estará en un momento del tiempo posterior (t) al de la variable independiente (t-1). Decimos ordenados cronológicamente porque actualmente nos encontramos en el momento (t) del tiempo. Si avanzamos un período nos trasladamos a (t+1) y si retrocedemos un período nos vamos a (t-1).

El valor de p se llama orden. Por ejemplo AR(1) sería un proceso regresivo de primer orden. La variable de resultado en un proceso AR de primer orden en algún tiempo t está relacionada solo con periodos de tiempo separados por un periodo (es decir el valor de la variable en t-1). Un proceso AR de segundo orden o tercer orden estaría relacionado con datos separados por dos o tres periodos.

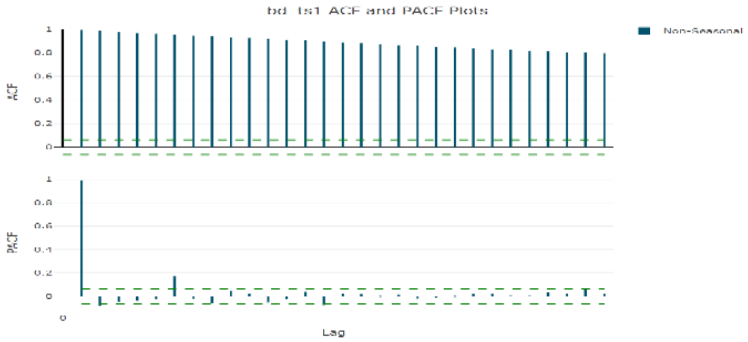
El Modelo AR(p) está definido por la ecuación:

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + a_t, \quad t \in \mathbb{Z}.$$

En esta ecuación:

- y_t: Es el valor actual de la serie.
- C: es una constante.
- $\emptyset_1, \emptyset_2, \ldots, \emptyset_p$: Son los coeficientes autorregresivos.
- $y_{t-1}, y_{t-2}, \ldots, y_{t-p}$: Son los valores pasados de la serie de tiempo.
- ϵ_t: Es el termino de error (ruido blanco) en el tiempo t.

Para estimar los coeficientes del modelo AR(p) se pueden usar varios métodos siendo los más comunes la mínima varianza y el método de máxima verosimilitud en **R** la función **arima**() del paquete **Stats** usa por defecto el método de máxima verosimilitud.



En nuestra investigación se ajustaron dos modelos AR de orden 5 y 6 todo dependiendo de las funciones ACF y PACF, las predicciones no fueron muy satisfactorias, así que este enfoque lo usaremos para predecir la serie de rendimientos.

Modelo ARIMA

El modelo arima (Autoregresive integral moving average) combina tres componentes: autorregresivo (AR) de media móvil (MA) y de integración (I). La fórmula general del modelo arima(p,d,q) es :

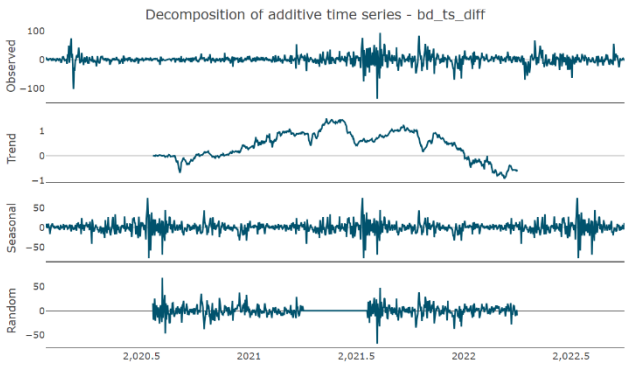
$$\emptyset(B)(1-B)^d y_t = \theta(B)\epsilon_t$$

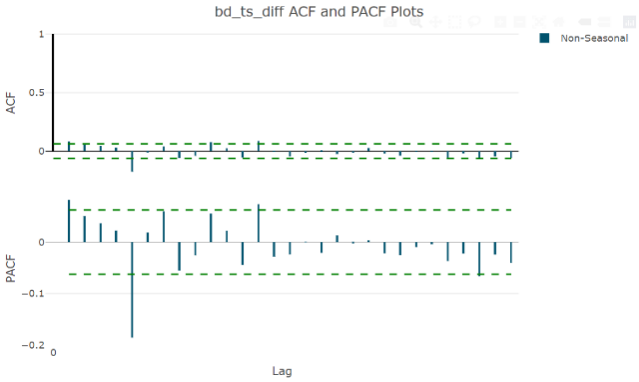
Donde:

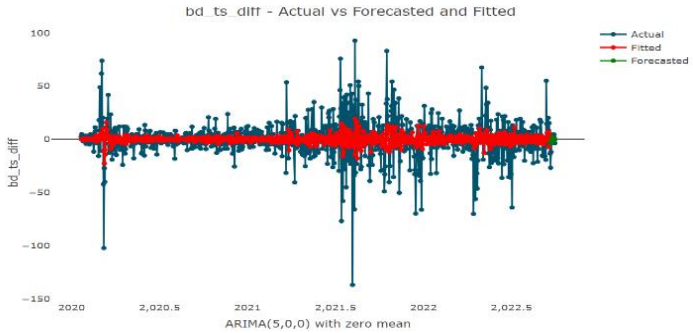
- y_t: Es el valor actual de la serie temporal.
- $\emptyset(B) = 1 - \emptyset_1 B - \emptyset_2 B^2 - \cdots - \emptyset_p B^2$: Es el polinomio autorregresivo de orden p.
- $(1-B)^d$: Es el operador de diferenciación de orden (d).
- $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots .. + \theta_q B^q$: Es el polinomio de media móvil de orden (q)
- ϵ_t: Es el termino de error (ruido blanco).
- (B): Es el operador de retardo, donde ($By_t = y_{t-1}$).

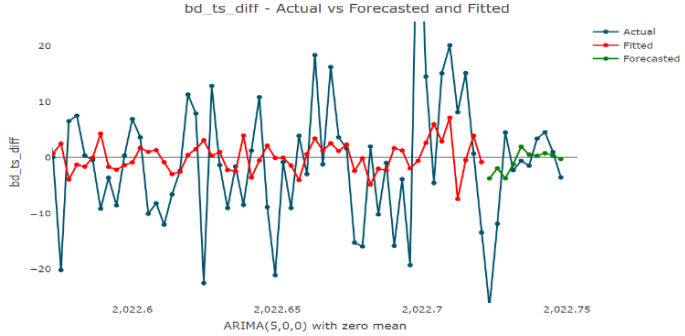
Este Modelo es muy flexible se utiliza para analizar y predecir series temporales que puedan no ser estacionarias.

Descomposición de la serie sin tendencia y predicciones del modelo arima









Con ayuda de la función **auto.arima** encontramos el mejor modelo para la serie de rendimientos esta función sugiere un modelo arima(5,0,0) y anteriormente vemos cuales fueron sus predicciones, este modelo presentó un MSE de 74.40528

Implementación de modelos ARCH y GARCH

Los modelos ARCH (Autoregresive Conditional Heteroskedasticity) y GARCH (Generalized Autoregresive Heteroskedasticity) son herramientas fundamentales en el análisis de series temporales, especialmente en el campo de las finanzas, donde la volatilidad de los datos s una característica importante.

Modelo ARCH

El Modelo ARCH fue introducido por Robert Engle en 1982. Este modelo se utiliza para describir una serie temporal cuya varianza cambia con el tiempo y dependen de los valores pasados de la serie. La fórmula general del modelo ARCH(q) es.

$$y_t = \sigma_t \epsilon_t$$

$$\sigma_t^2 = \propto_0 + \propto_1 y_{t-1}^2 + \propto_2 y_{t-2}^2 + \cdots + \propto_q y_{t-q}^2$$

Donde:

- y_t: Es el valor actual de la serie.
- σ_t^2: Es la varianza condicional en el tiempo t.
- ϵ_t: Es un término de error con media cero y varianza constante.
- $\propto_0, \propto_1, \propto_2, \propto_3, \ldots\ldots, \propto_q$: Son los parámetros del modelo.

Modelo GARCH

Es una extensión del modelo ARCH, fue desarrollado por Tim Bollerslev en 1986.Este modelo incluye términos autorregresivos en la varianza condicional lo que permite una descripción mas simple y precia de volatilidad. La formula general del modelo Garch(p,q) es:

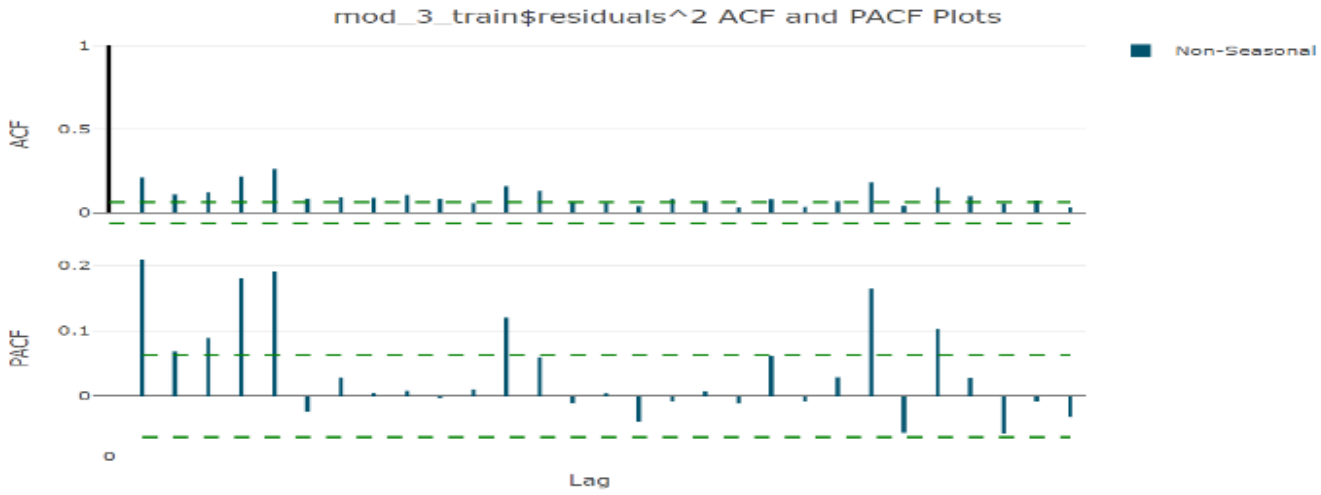
$$y_t = \sigma_t \epsilon_t$$

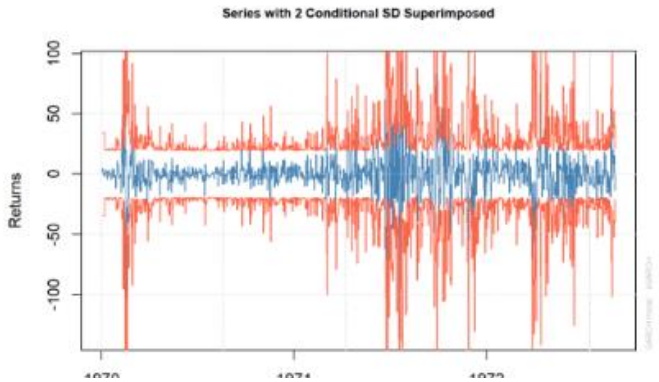
$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i y_{t-1}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

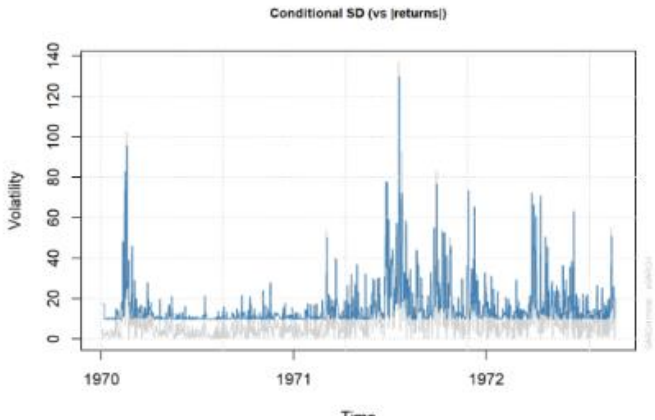
Donde:

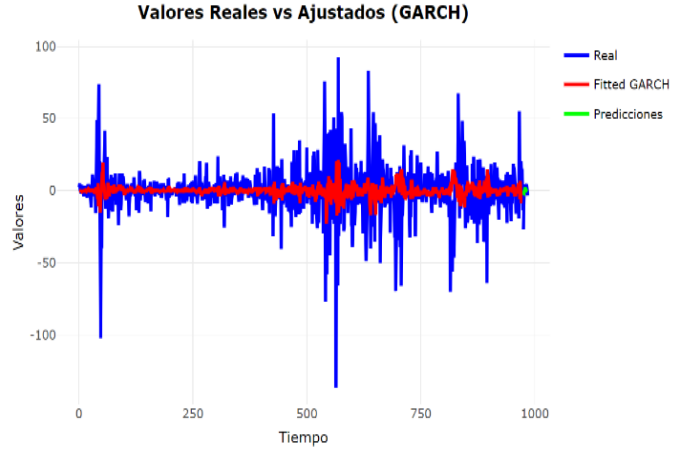
- y_t: Es el valor actual de la serie temporal.
- σ_t^2: Es la varianza condicional en el tiempo t.
- ϵ_t: Es un termino de error con media cero y varianza constante.
- $\propto_0, \propto_i, \beta_j$: Son los parámetros del modelo.

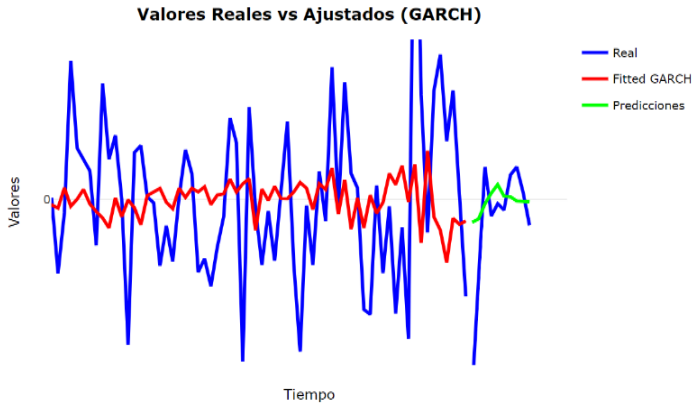
Grafica del modelo y comentarios sobre cuál de los dos modelos de volatilidad es mejor





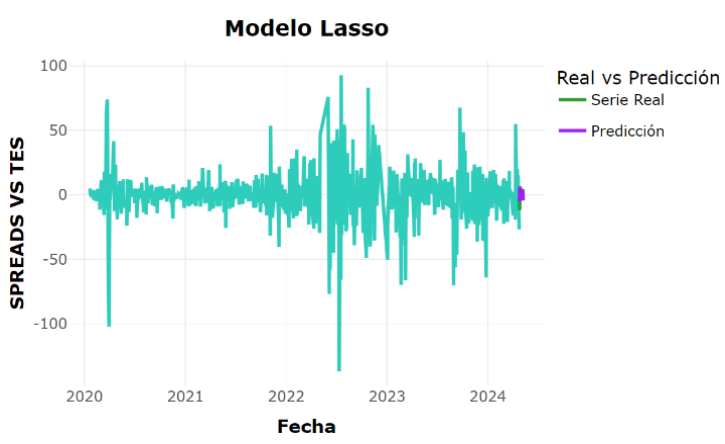


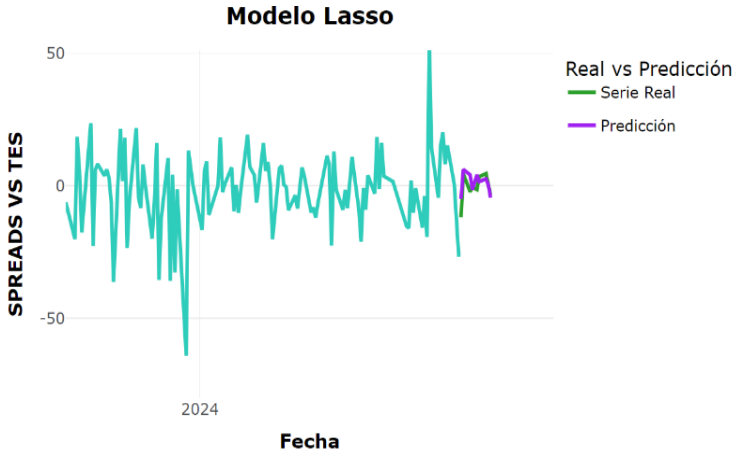




En esta parte es importante resaltar que se ajustaron dos modelos de volatilidad un modelo Garch y un modelo Arch, al evaluar el AIC y BIC de cada modelo notamos que era muy parecido además sus predicciones eran de igual forma muy parecidas sin embargo nos quedarnos con el modelo Arch dado que tiene menos parámetros y además resultan ser todos significativos, el modelo presentó un MSE 73.22722

Como última opción vamos a tener en cuenta en modelo de regresión Lasso, ya que este modelo fue el que presentó menor error de predicción entre los tres modelos de regresión que mencionamos al principio pero, para modelar la serie de rendimientos, los resultados se muestran a continuación.





El modelo Lasso tuvo el mejor error de predicción que los modelos de series de tiempo con un MSE de 14.6989

Esto nos permite concluir que el modelo que debemos elegir para intentar modelar los rendimientos de la serie debe ser el modelo Lasso.

Para resumir lo hecho en el trabajo, buscamos modelos que nos permitieran predecir de manera satisfactoria, la red neuronal GRU fue el mejor modelo para predecir la serie original y el modelo Lasso con las variables sugeridas por el método low variance flter para predecir la serie de rendimientos.