Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática

Creación de Portafolios usando Machine Learning

Informe Práctica Profesional II

Estudiante:	Jorge Novoa Contreras
Empresa:	RialStat SPA
Supervisor:	Ricardo Pérez Sáez
Cargo Supervisor:	Director Ejecutivo
Periodo Práctica:	Inicio: 8 de Enero 2024
	Término: 2 de Febrero 2024

Resumen

En este informe se presenta lo realizado durante la Práctica Profesional II en la empresa RialStat.

El objetivo y tarea principal de esta práctica fue:

- Construcción de portafolios de inversión utilizando modelos matemáticos estadísticos e Inteligencia Artificial.

La metodología para esto consistió en la lectura de un paper que mostraba técnicas de predicción de *Long-Time Series* basadas en modelos de *Transformers*, junto con implementar el modelo señalado en el paper para predecir el valor de un activo en cierta ventana de tiempo.

La metodología para la creación de los portafolios fue, en primera instancia, estudiar la teoría más general de esto y los métodos usuales para resolver el problema de optimización asociado a la elección de los activos dentro de una cartera dada cierta condición (maximizar ganancia, minimizar riesgo).

Como resultados generales, se encontró que en general las predicciones con el modelo del paper parecen prometedoras al predecir activos, pero no se puede concluir nada a ciencia cierta dado que hubo muchas limitantes que no permitieron en esta instancia hacer una predicción "bien hecha" (usar más de una variable, incorporar variables exógenas, entre otras), además de aprender acerca de cómo se hace usualmente la creación de carteras de activos y sus problemas asociados.

Se concluyó que esta experiencia de práctica fue una buena introducción al mundo de la ciencia de datos junto con el mundo de los portafolios, al aprender y profundizar sobre modelos predictivos relacionados a activos.

Resumen Práctica Profesional I

Durante la Práctica Profesional I en RialStat, llevé a cabo actividades como la revisión y validación de bases de datos, así como el desarrollo de modelos estadísticos y de proyección macroeconómica.

En particular, en el desarrollo y uso de modelos estadísticos y predictivos, se hizo hincapié en la aplicación y desarrollo de modelos para agentes económicos a corto plazo, destacando entre ellos la efectividad de modelos como ARIMA y PROPHET.

En conclusión, la Práctica Profesional I proporcionó una introducción valiosa a la ciencia de datos, abordando temas como series de tiempo y modelos predictivos.

Índice de Contenidos

1. Contexto **1**

2. Objetivos y Metodología **2**

3. Desarrollo **4**

4. Conclusiones **6**

Referencias **7**

Anexo A. Imágenes **8**

Índice de Figuras

A.1. Frontera eficiente asociada a una cartera para un conjunto de activos. 8
A.2. Pesos de los activos asociados a la cartera óptima. 8
A.3. *Asignación discreta* para la cartera óptima. 8
A.4. Métricas para predicciones con ventanas a 20 días, 40 días y 192 días. 9
A.5. Predicción usando el Modelo *Linear*. 10
A.6. Predicción usando el Modelo *DLinear*. 10
A.7. Predicción usando el Modelo *NLinear*. 10
A.8. Predicción usando el Modelo *Informer*, usando 3 *epochs* para el entrenamiento y L.R. de 0.0005. 11
A.9. Predicción usando el Modelo *Informer*, usando 5 *epochs* para el entrenamiento y L.R. de 0.05. 11
A.10. Predicción usando el Modelo *Transformer*, usando 3 *epochs* para el entrenamiento y L.R. de 0.0005. 12
A.11. Predicción usando el Modelo *Transformer*, usando 5 *epochs* para el entrenamiento y L.R. de 0.05. 12

1. Contexto

RialStat SPA está integrado por profesionales con más de 30 años de experiencia en el análisis y desarrollo de Modelos Matemáticos Estadísticos.

Su misión empresarial es brindar un servicio altamente especializado, especialmente en el ámbito de riesgo financiero y predictivo.

En particular, se desarrollan y construyen Modelos Matemáticos Estadísticos de los siguientes tipos:

- Modelos para el apoyo a la Gestión de Negocios
- Modelos Predictivos
- Modelos de Provisiones
- Construcción de Score (Behavior, Pago, Admisión, entre otros)

RialStat busca incorporarse en la creación de portafolios para sus clientes, en particular, en el uso de herramientas como Machine Learning para la creación automatizada de estos, siendo este último el trabajo que me fue asignado durante la práctica.

A grandes rasgos, un portafolio corresponde a la cantidad de inversiones como acciones, bonos, bienes raíces y otros instrumentos financieros, que posee alguna persona/empresa, por lo que el "problema del portafolio" corresponde a encontrar cuál es la asignación de las inversiones (cuáles acciones comprar, por ejemplo) que generen la mayor ganancia bajo ciertas restricciones. En el sentido matemático, denotamos como "portafolios" a la asignación de pesos respecto a cada activo, por lo que corresponde a un vector $w \in \mathbb{N}^n$ (en el caso que haya n activos, por ejemplo) y que cumple con: $\sum w_i = 1$.

Bajo esto, diremos que un portafolio es "óptimo" si es el que maximiza cierta función de utilidad bajo algunas restricciones. En general, la función de utilidad intenta considerar tanto la ganancia monetaria como el riesgo al invertir en tal portafolio (*Modelo de Markowitz*), y las restricciones están relacionadas con el riesgo, como cuál es el valor máximo permitido para ciertos w_i, la positividad de los w_i, entre otros.

También se puede considerar el problema de optimización del portafolio como el problema de minimización de la Varianza asociada al portafolio, sujeto a restricciones que incluyen una cota inferior de retorno (en resumen, se busca asegurar cierto retorno monetario). Esta "solución" surge naturalmente de la búsqueda de algunos inversores de minimizar el riesgo asociado a un portafolio (que se mide usualmente con la varianza respectiva al portafolio).

Además, debemos considerar que solo utilizar la información en un tiempo presente sobre los activos puede limitar la opción más óptima a largo plazo, ya que puede haber algún activo en el cual se pronostique que estará en caída durante un largo plazo, por lo que la mejor estrategia en tal caso siempre será vender todos esos activos, cosa que no podemos considerar como posibilidad si no generamos esta predicción sobre los precios de los activos.

Así, con todo lo explicado anteriormente (bajo qué contexto surge y cómo se evaluará), tenemos nuestro problema a tratar durante la práctica.

2. Objetivos y Metodología

Antes de empezar la práctica, se acordaron los siguientes objetivos a nivel general:

- Construcción de portafolios de inversión utilizando modelos matemáticos estadísticos e Inteligencia Artificial.

En particular, se solicitó implementar el modelo para la predicción de *Long-Time Series* basado en Transformers mostrado en el siguiente Paper: He, J., Sporea, C., Dima, C. (2022, May 13). *Are Transformers Effective for Time Series Forecasting?* [1] para la predicción de múltiples variables relacionadas a los activos de algún portafolio. Para cumplir con esto, se tuvieron que cumplir los siguientes objetivos (enumerados en orden):

- Lectura de la teoría moderna de creación de portafolios e implementaciones usuales en la industria.
- Implementación de código en Python para la importación de datos relacionados con los activos en el mercado.
- Revisión del Paper [1], e implementación del modelo para Python.
- Comprobación de la eficacia de los modelos mostrados en el paper para el objetivo de la predicción de activos.

La metodología para investigar sobre la teoría moderna de creación de portafolios fue revisar diversas fuentes, en las que destacan Montero, F. C. (2022). *Modern portfolio optimization*. Universitat de Barcelona [2] y Markowitz, H. (1952). Portfolio selection. The Journal of Finance, 7(1), 77-91. [3], buscando entender las distintas estrategias para la solución del problema según las preferencias del cliente.

Con respecto a la metodología usada para la creación del código para importar los datos de los activos en el mercado, se ocupó el paquete *Yfinance* en Python para la importación de los datos (en crudo) y el paquete *Pandas* para la creación y organización de los datasets para su posterior uso.

Posterior a esto, a manera de un primer acercamiento a la optimización en el portafolio, se creará un portafolio "dummy" con los siguientes activos: "PTR" - PetroChina Company Limited ADR; "BUD" - Anheuser Busch Inbev SA (AB InBev); "XOM" - Exxon Mobil Corporation; "BA" - Boeing Co; "CHTR" - Charter Communications Inc; "SHOP" - Shopify Inc; "NVDA" - NVIDIA Corporation; "NKE" - Nike Inc., posterior a esto, se hace un test de correlación para verificar, valga la redundancia, la correlación entre los activos antes mencionados para finalmente el paquete *Pypfopt* para la creación de la *Frontera de Eficiencia* asociada, eligiendo como punto de eficiencia la configuración de los activos donde se alcanza el mayor *Coeficiente de Sharpe*. Por último, para definir la configuración óptima se realiza una *Asignación Discreta* para dar con el portafolio óptimo (dado que las soluciones encontradas por la frontera de eficiencia suelen ser a valores no enteros).

La revisión del paper *Are Transformers Effective for Time Series Forecasting?* [1] consistió en, primero, la lectura de este, buscando consultar con el supervisor los términos y conceptos que no manejase. Posteriormente, se procedió a usar la implementación en código entregada en el mismo paper para testear con las otras variables (en virtud del objetivo de la práctica, estas variables eran justamente relacionadas a los activos).

En virtud de que la implementación de los modelos señalado en el paper fue hecha en *Bash*, se tuvo que investigar acerca de cómo ejecutar el código que estaba en *Bash* y cómo pasarlo a un código más simple (o al menos más manipulable) puesto que el código entregado era muy rígido respecto a los aspectos que se requerían (intentar predecir con otras variables).

Seguido a esto, se ocupó el código para la importación de datos sobre el activo de *Nvidia* para determinar cuán buena es la predicción, usando como variables: "apertura", "cierre", "volumen", "volatilidad", registrando el error en dos métricas: MSE (*Mean Squared Error*) y MAE (*Mean Absolute Error*) con respecto a los valores reales, además, se realizaron predicciones en distintos lapsos de tiempo, siendo estos ventanas de 20 días, 40 días y 192 días.

A raíz de las complicaciones en la manipulación de los datos para el código implementado por el equipo tras el paper, se tuvo que dar con una manera alternativa para usar el modelo, y esta fue ocupando un paquete llamado *NeuralForecast*, el cual implementó el modelo del paper durante el transcurso de la práctica dentro de sus modelos.

Para el uso de *NeuralForecast*, se usó el mismo activo *Nvidia* con la diferencia de que se ocupó una única variable, siendo esta el precio de cierre del activo, además, no solo se ocupó el modelo *DLinear*, sino que se ocuparon *Linear*, *DLineal*, *NLinear*, *Informer* y *Transformer*, donde las ventanas de testeo fueron a 20 días en todas, y para los modelos *Informer*, *Transformer* se ocuparon los siguientes hiperparámetros: L.R. 0.05 con 5 épocas, 0.0005 con 3 épocas, los cuales fueron elegidos mediante prueba y error para intentar reducir el error en la predicción.

Al igual que con el código del paper, se registraron los errores usando dos métricas para ver la calidad de la predicción respecto a los valores reales, siendo estas, nuevamente, MSE (*Mean Squared Error*) y MAE (*Mean Absolute Error*).

3. Desarrollo

Inicialmente, debido a que se debía contextualizar sobre los portafolios, se buscó en distintas fuentes acerca de la teoría moderna para la creación de portafolios, las que más información útil entregaron fueron [4], [3], [2].

Debido a que, si bien el modelo del paper entregaría predicciones acerca de un activo, la elección de los posibles activos debía ser realizada usando técnicas usuales para la elección de los activos dentro de una cartera. Por lo cual se buscó un paquete el cual resolviese este problema de optimización, siendo el paquete usado *Pypfopt*, ya que contenía funciones que entregaban la *frontera de eficiencia* asociada a una cartera, junto con el punto donde se alcanza el máximo para el *coeficiente de Sharpe* (el cual es el óptimo para carteras cuyo cliente busca maximizar ganancia y minimizar riesgo).

A la par de esto, se estableció un código para importar los datos de los activos posibles usando el paquete *Yfinance* y *Pandas* en Python, exportando así la información sobre los activos "PTR" - PetroChina Company Limited ADR; "BUD" - Anheuser Busch Inbev SA (AB InBev); "XOM" - Exxon Mobil Corporation; "BA" - Boeing Co; "CHTR" - Charter Communications Inc; "SHOP" - Shopify Inc; "NVDA" - NVIDIA Corporation; "NKE" - Nike Inc, para poder probar cómo se resolvía el problema de optimización con el paquete *Pypfopt*, usando 1000 USD de capital (lo cual es la restricción de costo sobre la cartera), obteniendo así la *frontera de eficiencia* y el *coeficiente de Sharpe* mostrados en la Figura A.1.

Posterior a esto, debido a que la cartera óptima quedó con cantidades indiscretizadas, como se muestra en la Figura A.2, se tuvo que usar la herramienta de *Asignación Discreta* que también está en el paquete *Pypfopt*, obteniendo así la configuración óptima para la cartera, la cual se puede apreciar en la Figura A.3.

Durante la lectura del paper [1] hubo muchos conceptos los cuales no se tenían un conocimiento previo, en particular respecto a las *soluciones para Long Time Series Forecast usando Transformers*, puesto que no se tenía un conocimiento previo de esto, por lo que mediante las preguntas surgidas al supervisor y la búsqueda en distintas fuentes, se logró comprender a modo superficial en lo que consistía.

Siguiendo con el paper, pero esta vez estudiando el código en el cual implementaron los distintos modelos *Linear*, surgieron complicaciones a la hora de ejecutar los ejemplos mostrados en el paper, ya que el código estaba diseñado para ser ejecutado en *Bash*, el cual es un lenguaje de comandos que por mi parte nunca había manejado, por lo que tuve que investigar cómo funciona este lenguaje de comandos.

Después de entender un poco de *Bash*, procedí a cambiar el dataset usado en uno de los ejemplos con el dataset de un activo (en este caso ocupé el activo de *Nvidia*) usando como variables del activo: "apertura", "cierre", "volumen", "volatilidad". Así, se procedió a ocupar el modelo DLinear sobre esta prueba, dando así las siguientes métricas mostradas en la Figura A.4. Si bien estos resultados no estaban tan mal considerando las restricciones para los datos, se esperaban mejores métricas, por lo que se tuvo que recurrir a buscar algún método que nos permitiese trabajar con DLinear y se cumpliese alguna de las dos siguientes: disminuyese el error

- Disminuyera el error de la predicción.
- Se tuviese más libertad con la manipulación de las variables.

Lamentablemente, la “solución” que se encontró no cumplía ninguna de estas, pero, pese a esto, nos dio un *insight* de cómo estaba funcionando el modelo respecto a los datos que le entregábamos del activo.

La solución antes mencionada corresponde al paquete *NeuralForecast*, ya que durante la realización de la práctica, el equipo encargado del paquete incorporó una automatización de los modelos *Linear* del paper [1].

La ventaja de este paquete era que, como bien se mencionó antes, estaba automatizado el método de predicción, era mucho más *user-friendly* que la implementación que entregó el equipo del paper, y permite agregar modelos propios (ya sean modificaciones de modelos previamente vistos o alguno completamente nuevo). Dada la dificultad para manejar los datos con la implementación debido a que estaba escrita en *Bash*, se limitó a estudiar los modelos *Linear* usando el paquete *NeuralForecast*.

Los únicos grandes problemas de esta opción fueron los siguientes:

- **Aún no estaba agregado de manera oficial al paquete:** solo se encontraba en el git del paquete, pero no en la última versión que permitía solo incorporar *pip install ...*), por lo que se tuvo que instalar de forma manual, lo cual fue una tarea relativamente compleja dado mi poco conocimiento para programar.
- **Los modelos *Linear* solo permitían una variable:** la razón de esto era la misma que tuve al yo implementar el modelo, el código que dio el equipo del paper [1] era difícil de manipular sin generar problemas en paralelo.

Pese a esto, se realizó el ejercicio de hacer una predicción (con la consideración de ahora ocupar una única variable: la variable a predecir) usando los modelos *Linear*, obteniendo así los resultados mostrados en las Figuras A.5, A.6 y A.7. Dado que los resultados son muy parecidos a los logrados previo al uso de *NeuralForecast*, se pudo concluir que el "peso"que se le estaba dando a las otras variables dentro de la predicción no era significativo.

También se tuvo la oportunidad de probar los modelos *Informer* y *Transformer* para la predicción del activo, al menos para ver si estos modelos pueden ser prometedores bajo la nueva restricción de trabajar con una única variable (ya que ocupamos la forma automatizada entregada por *NeuralForecast* para los dos modelos antes mencionados), dando así los resultados mostrados en las Figuras A.8, A.9, A.10 y A.11.

Como se pudo observar en las figuras antes referenciadas, el error era muy elevado y claramente el modelo no lograba aprender de los datos, lo cual es una característica que comparten *Informer* y *Transformer*, ya que requieren de una cantidad significativa de variables para aprender bien.

4. Conclusiones

Durante la Práctica Profesional en la empresa RialStat, se desarrollaron varias habilidades, destacando las de trabajo en equipo y las de carácter analítico. Las habilidades de trabajo en equipo se utilizaron para coordinar y distribuir las labores, mientras que las habilidades analíticas estuvieron relacionadas con el análisis del paper y la solución de las distintas problemáticas que surgieron durante cada implementación.

Uno de los logros principales de la práctica fue adentrarse, aunque fuera un poco, en el mundo de los modelos de portafolios, tanto en su teoría como en el proceso de creación de los portafolios. De la mano con esto, se comprendió la importancia que tienen los modelos de Series de Tiempo y los distintos métodos para hacer predicciones sobre estos al ser aplicados en sectores como la economía, y por qué son de uso casi obligatorio dentro de cualquier empresa.

Otro logro que debo señalar fue continuar aprendiendo sobre los métodos de análisis y predicción para series de tiempo (en este caso, la serie de tiempo correspondía al precio del activo de *Nvidia*), ya que se profundizó en los modelos basados en *Transformers* para la predicción de series de tiempo a través de la predicción de este activo, complementando lo visto en mi primera Práctica Profesional.

Por último, se destaca haber aprendido a solucionar algunos problemas cuando algún código se encuentra en *Bash*, ya que, como se observó en esta experiencia, no es algo poco común implementar códigos en ese lenguaje.

Como limitaciones principales, estuvieron el poco conocimiento de todo lo relacionado con modelos de *Machine Learning* (como el modelo de *Transformer*) junto con el poco conocimiento que tenía acerca de programar en *Bash*, lo que hizo que la resolución de algún problema simple fuera un poco más larga de lo esperado, ya que primero había que entender lo realizado.

Por último, me gustaría destacar la excelente disposición de Ricardo Perez y Wilfredo Palma (respectivamente, el supervisor y la persona que me guió en RialStat) para enseñarme y motivarme a aprender acerca de los temas vistos en esta práctica.

Referencias

[1] Zeng, A., Chen, M., Zhang, L., y Xu, Q., "Are transformers effective for time series forecasting?," 2022.

[2] Montero, F. C., MODERN PORTFOLIO OPTIMIZATION. 2022.

[3] Markowitz, H., "Portfolio selection," The Journal of Finance, vol. 7, no. 1, pp. 77–91, 1952.

[4] Setayesh, A., "Modern portfolio theory.", https://www.stat.berkeley.edu/users/aldous/24/Posted/Ali_Setayesh.pdf.

Anexo A. Imágenes

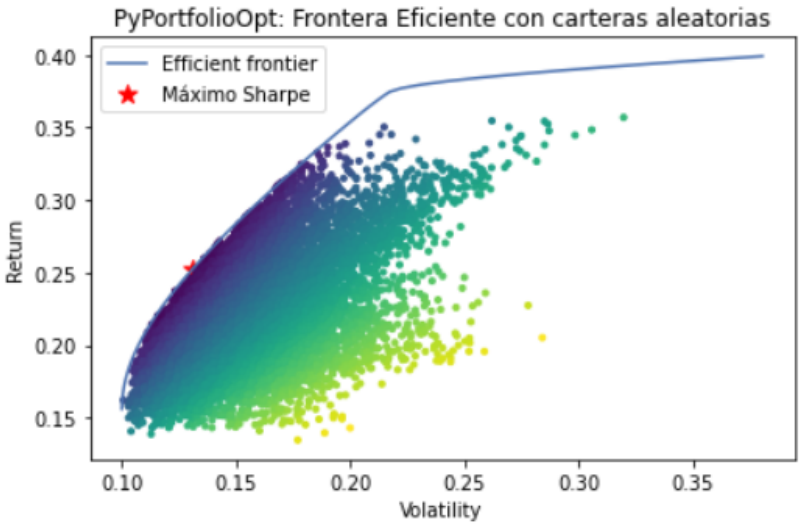


Figura A.1: Frontera eficiente asociada a una cartera para un conjunto de activos.


```
"BA": 0.01825,
"BUD": 0.25757,
"CHTR": 0.176,
"NKE": 0.04644,
"NVDA": 0.10607,
"SHOP": 0.374,
"XOM": 0.02167
```


Figura A.2: Pesos de los activos asociados a la cartera óptima.


```
{'BUD': 8, 'CHTR': 1, 'NKE': 1, 'SHOP': 9, 'XOM': 1}
```


Figura A.3: *Asignación discreta* para la cartera óptima.

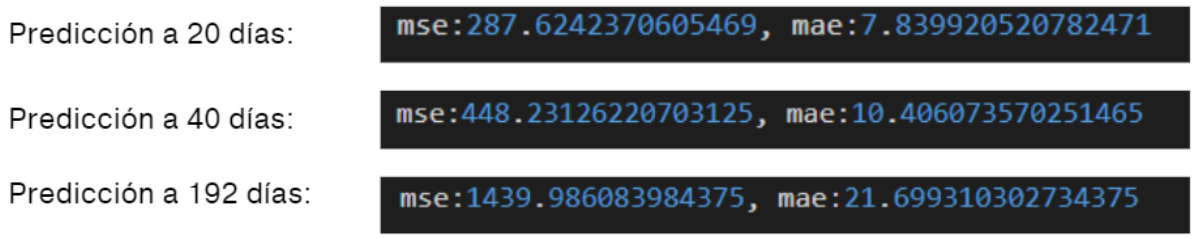
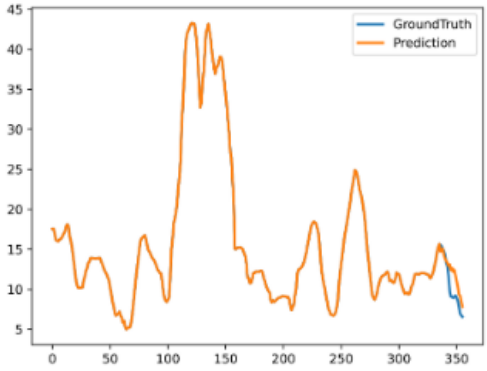
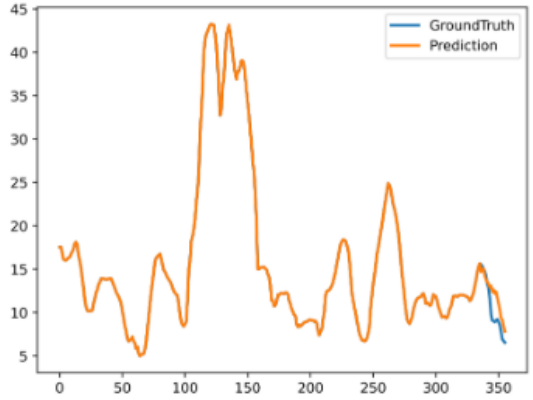


Figura A.4: Métricas para predicciones con ventanas a 20 días, 40 días y 192 días.



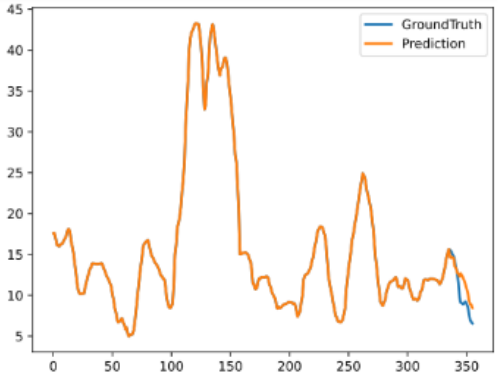

```
mse:292.598876953125, mae:8.118897438049316
```

Figura A.5: Predicción usando el Modelo *Linear*.



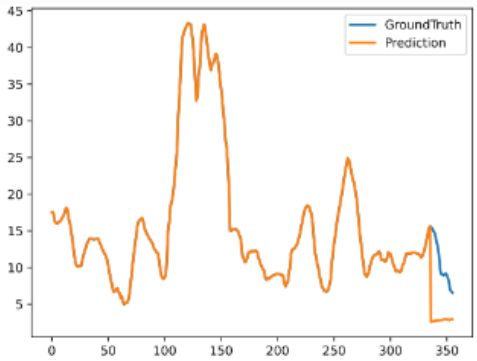

```
mse:295.02978515625, mae:8.18729305267334
```

Figura A.6: Predicción usando el Modelo *DLinear*.



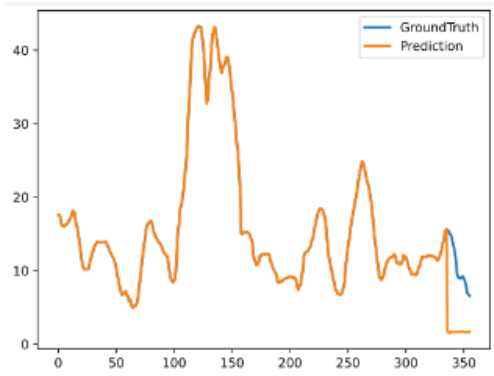

```
mse:287.1707763671875, mae:7.913345813751221
```

Figura A.7: Predicción usando el Modelo *NLinear*.



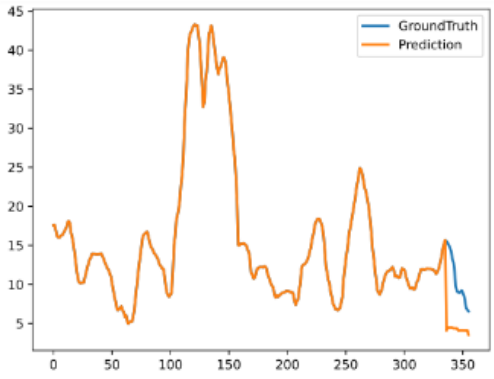

```
mse:7335.2861328125, mae:58.02734375
```

Figura A.8: Predicción usando el Modelo *Informer*, usando 3 *epochs* para el entrenamiento y L.R. de 0.0005.



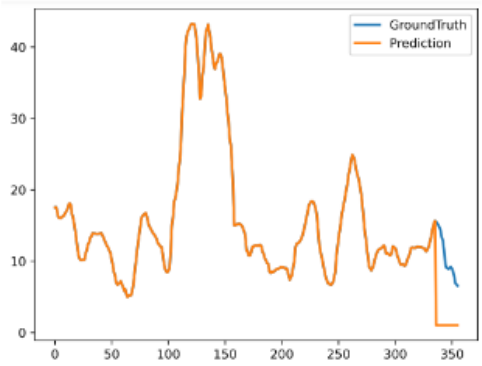

```
mse:7400.1103515625, mae:58.358829498291016
```

Figura A.9: Predicción usando el Modelo *Informer*, usando 5 *epochs* para el entrenamiento y L.R. de 0.05.




```
mse:7251.357421875, mae:57.66730499267578
```

Figura A.10: Predicción usando el Modelo *Transformer*, usando 3 *epochs* para el entrenamiento y L.R. de 0.0005.




```
mse:7444.83544921875, mae:58.6176643371582
```

Figura A.11: Predicción usando el Modelo *Transformer*, usando 5 *epochs* para el entrenamiento y L.R. de 0.05.