Universidad del Norte
Departamento de Matemáticas y Estadística
Profesor: Dr. Lihki Rubio
Parcial 1 de Machine Learning
Alumno:_____

Duración: 1 semana Fecha: / /

Indicaciones: La presente evaluación debe ser entregada usando solamente un link de pagina web de GitHub creado mediante JBook. Cada figura, tabla, resultado, debe ser interpretado. Resultados y visualizaciones que no cuenten con su respectivo análisis serán evaluados con la nota mas baja.

Ejercicio 1

Análisis Exploratorio de Datos 20 %

(I) Considere la serie de tiempo asociada con las acciones de Bitcoin (ver Bitcoin data importe directamente con el link y pd.read_csv). Realice un Análisis Exploratorio de Datos (EDA) considerando cada una de las metodología abordadas durante el curso. *Es crucial que en todos los ejercicios de la presente evaluación, cada figura descriptiva, tabla o resumen de resultados, cuente con su respectiva interpretación*

- Utilice el conjunto de datos asociados con el precio de Bitcoin, ver (Bitcoin data). Considere para las predicciones con modelos de series de tiempo, la columna Price
- Identifique si existen datos faltantes y realice su imputación usando alguna de las técnicas básicas estudiadas durante el curso.
- Realice un grafico de velas utilizando la librería Plotly para la serie de tiempo (columna Price). Realice también un histograma con Plotly para representar el volumen tradeado diariamente (Volume)
- Realice graficos de subseries considerando agrupación por *semana, mes y año*. Utilice graficos de *series de tiempo y boxplots* para analizar estas subseries.
- Estudie estacionariedad de la serie de tiempo usando la ACF estudiada durante el curso, así como también los respectivos test estadísticos tales como Ljung-Box y Dickey Fuller. Aplique además las transformaciones necesarias para convertir la serie de tiempo en estacionaria.
- Mediante agregación por grupos analice el comportamiento de la media y desviación estándar, agrupando por *semanas, días, meses*. Además, para estas agrupaciones, calcule estadísticos móviles. Dibuje la distribución de frecuencias para diferentes lags como se realizó en las notas del curso.
- Aplique las medias moviles $\widehat{M}_t^{(2)}$, $\overline{\widehat{M}_t^{(2)}}^{(t,t+1)}$, $\widehat{M}_t^{(4)}$, $\overline{\widehat{M}_t^{(4)}}^{(t,t+1)}$, $\widehat{M}_t^{(3)}$ y $\overline{\widehat{M}_t^{(3)}}^{(t-1,t+1)}$ a las serie del precio de cierre de BTC-USD. Posteriormente, uselas para remover tendencia y aplique la prueba de Dickey Fuller y LJung Box para verificar si se logró estacionariedad
- Realice descomposición de la serie temporal del precio de cierre de BTC-USD usando medias moviles y además usando la librería statsmodels.tsa. Verifique al final independencia de los residuos mediante la prueba de LJung Box.

(II) Repita el análisis anterior para las series de tiempo de: retorno acumulado diario y volatilidad, para diferentes ventanas. Para esto tenga en cuenta las siguientes definiciones e implemente manualmente las funciones que definen cada serie de tiempo

- Retorno aculumado diario $\{A_t\}_{t=1}^T$ donde

$$A_t = \sum_{j=1}^{t} R_j, \text{ con } R_t = \frac{P_t - P_{t-1}}{P_{t-1}}, \tag{1}$$

donde $P_{t,t=1,2,\ldots,T}$ es la columna Price en el dataset de Bitcoin y T es el tiempo final.

- La volatilidad $\{\sigma_t\}_{t=1}^{T}$ es la desviación estándar de los retornos diarios

$$\sigma = \text{std}(R_1, R_2, \ldots, R_\omega) \text{ donde } \omega = 7, 14, 21, 28, \tag{2}$$

siendo ω el número de retornos diarios considerados para calcular la desviación estándar

$$\text{std}(R_1, R_2, \ldots, R_\omega) = \sqrt{\frac{1}{\omega - 1}\sum_{t=1}^{\omega}(R_t - \mu)}, \ \mu = \frac{1}{\omega}\sum_{t=1}^{\omega} R_t, \tag{3}$$

obtenida por ventanas móviles de longitud ω, donde ω está dado en días, y esta desviación es calculada sobre la serie de retornos diarios. Para cada ventata ω repita el ítem (I).

Ejercicio 2

Modelos estadísticos 20 %

- En cada uno de los siguientes ejercicios considere los siguientes pasos para construir sus modelos predictivos. Debe realizar este procedimiento para las series: precio, retorno acumulado y volatilidad para las diferentes ventanas ($\omega = 7, 14, 21, 28$):
  - Divida el conjunto de datos en entrenamiento, validación y test
  - Para cada conjunto de parámetros, ajuste cada modelo predictivo en el conjunto de entrenamiento, y dentro de sus iteraciones de busqueda, evalue su modelo en el conjunto de validación usando las siguientes métricas: MAPE, MAE, RMSE, MSE y el R cuadrado
  - Realice una tabla donde resuma para el residual (*residuo en el conjunto de entrenamiento*) los scores obtenidos y agregue además las respectivas pruebas de hipóteis para independencia y normalidad. Para esto, usando Pandas construya una tabla con las siguientes columnas: MAPE, MAE, RMSE, MSE, R^2, LJung-Box test (p-value), Jarque-Bera (p-value) y agregue además las figuras para: Serie de residuos, QQPlot, ACF de residuos.
  - Realice una tabla donde resuma los scores obtenidos por cada modelo en el conjunto de test (*error de predicción*) y agregue además prueba de hipóteis para independencia. Para esto, usando Pandas construya una tabla con las siguientes columnas: MAPE, MAE, RMSE, MSE, R^2, LJung-Box test (p-value)
  - Considere conjuntos de validación y test de longitud $\tau = 7_{val,test}, 14_{val,test}, 21_{val,test}, 28_{val,test}$ días, el conjunto de entrenamiento corresponde a las primeras $N_{y_t} - (28_{val} + 28_{test})$ observaciones (ver Figura 1) utilicelas como conjunto de entrenamiento

  Los modelos que debe considerar para este ejercicio son los siguientes:

  - Simple Exponential Smoothing (Sin Librería)
  - Simple Exponential Smoothing (Usando statsmodels.tsa.holtwinters)
  - Double Exponential Smoothing (Sin Librería)
  - Double Exponential Smoothing (Usando statsmodels.tsa.holtwinters)
  - ARIMA (Usando Rolling)
  - ARIMA (Sin Rolling)
  - GARCH (Usando Rolling) *Must investigate*
  - GARCH (Sin Rolling) *Must investigate*

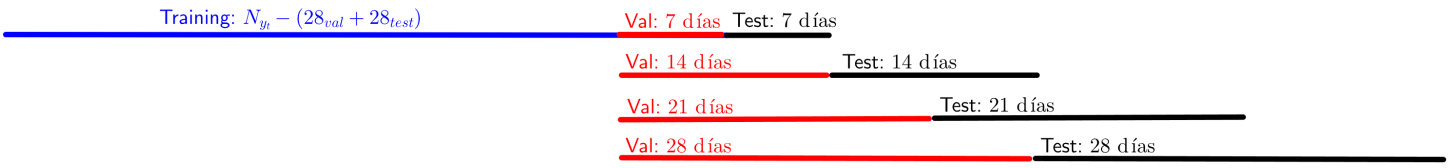

Figura 1: Conjuntos de entrenamiento y de prueba para las series de tiempo: *precio, retorno acumulado, volatilidad.*

Ejercicio 3

Modelos de Deep Learning 60 %

- Contexto del problema: *Para realizar predicción de series de tiempo, en la validación cruzada es crucial considerar el orden natural del tiempo. Por ejemplo, la idea es predecir 7 días futuros, con 31 datos historicos de un mes, no lo contrario. No requerimos predecir el pasado, porque ya lo conocemos. Con base en esto, es necesario construir manualmente una validación cruzada, que no realice ninguna desorganización no natural, de los pliegues de entrenamiento, validación y prueba, la cual esté en contra del orden temporal.*

- En cada uno de los siguientes ejercicios considere los siguientes pasos para construir sus modelos predictivos. Debe realizar este procedimiento para las series: precio, retorno acumulado y volatilidad para las diferentes ventanas ($\omega = 7, 14, 21, 28$):

  1. Defina funciones en Python para dividir el conjunto de datos en *entrenamiento, validación y test* con base en la Figuras 2-4. Observe que en las tres figuras, aparecen ejemplos para $\tau = 1, 2, 3$, debe extenderlo hasta los diferentes horizontes: $\tau = 7, 14, 21, 28$. Al final debe unir las predicciones de cada punto para obtener el horizonte predicho completo de τ días, luego debe compararlo con el resultado usando como output el y multivariado de τ días, tal como se muestra en la Figura 5.
  2. Nótese que TimeSeriesSplit no es una librería que funcione aquí, debido a que los pliegues que entrega no coinciden con los de las Figuras 2-4 que son los solicitados en este examen, por favor, no intente usarla, implemente sus propios pliegues. Al final de la construcción de la función que entrega dichas divisiones, debe visualizarlas dentro de su JBook usando las funciones de Python adecuadas para esto. El objetivo es verificar y confirmar que ha contruido adecuadamente los pliegues de entrenamiento, validación y prueba y coinciden sus figuras con las Figuras 2-5.
  3. Para cada conjunto de parámetros, ajuste cada modelo predictivo en el conjunto de entrenamiento, y dentro de sus iteraciones de busqueda, evalue su modelo en el conjunto de validación usando las siguientes métricas: MAPE, MAE, RMSE. MSE y el R cuadrado.
  4. Realice una tabla donde resuma para el residual (*residuo en el conjunto de entrenamiento*) scores obtenidos y agregue además las respectivas pruebas de hipóteis para independencia y normalidad. Para esto, usando Pandas construya una tabla con las siguientes columnas: MAPE, MAE, RMSE, MSE, R^2, LJung-Box test (p-value), Jarque-Bera (p-value) y agregue las figuras para Serie de residuos, QQPlot, ACF de residuos.
  5. Realice una tabla donde resuma los scores obtenidos por cada modelo en el conjunto de test (*error de predicción*) y agregue además prueba de hipóteis para independencia. Para esto, usando Pandas construya una tabla con las siguientes columnas: MAPE, MAE, RMSE, MSE, R^2, LJung-Box test (p-value)
  6. Considere conjuntos de validación y test de longitud $\tau = 7, 14, 21, 28$ días. Para el conjunto de entrenamiento considere todos los casos asociados con las combinaciones

$$\dim(X^{(j)}_{\text{train}}) = (M^{(j)}_{\text{rows}}, N^{(j)}_{\text{cols}}) \text{ con } M^{(j)}_{\text{rows}}, N^{(j)}_{\text{cols}} = 7, 14, 21, 28$$

     .
  7. Realice las siguientes variaciones durante la construcción de su modelo y seleccione la que entrega el menor error posible: Dropout: 0.2, 0.4, 0.6, 0.8; One layer and: (10,), (100,) (1000,) (10000,) Neurons; Batchsize: 16, 32, 64, 128. Advertencia; Debe utilizar la librería Tensorflow-Keras utilizada durante el curso de series de tiempo.

8. Para cada modelo obtenido dibuje la curva Runs vs Error/Score de entrenamiento historico del modelo. En el eje y represente el Error/Score para los conjuntos de entrenamiento, validación y test y en el eje x runs representa las iteraciones realizadas sobre cada j-ésimo pliegue de entrenamiento, validación y prueba.
9. Dibuje tres boxplots (uno para cada set: train/validation/test) donde represente los errores obtenidos en el ítem anterior en el eje y. Verifique que las medianas de los tres boxplots son cercanas.
10. Dibuje para cada combinación la serie de tiempo original, el conjunto de validación, el conjunto de test y la predicción del conjunto de test. Tenga en cuenta cada uno de los horizontes $\tau = 7, 14, 21, 28$

Los modelos que debe considerar para este ejercicio son los siguientes:

- MLP (*Multilayer Perceptrón*)
- RNN (*Must investigate*)
- LSTM (*Long short-term memory*)

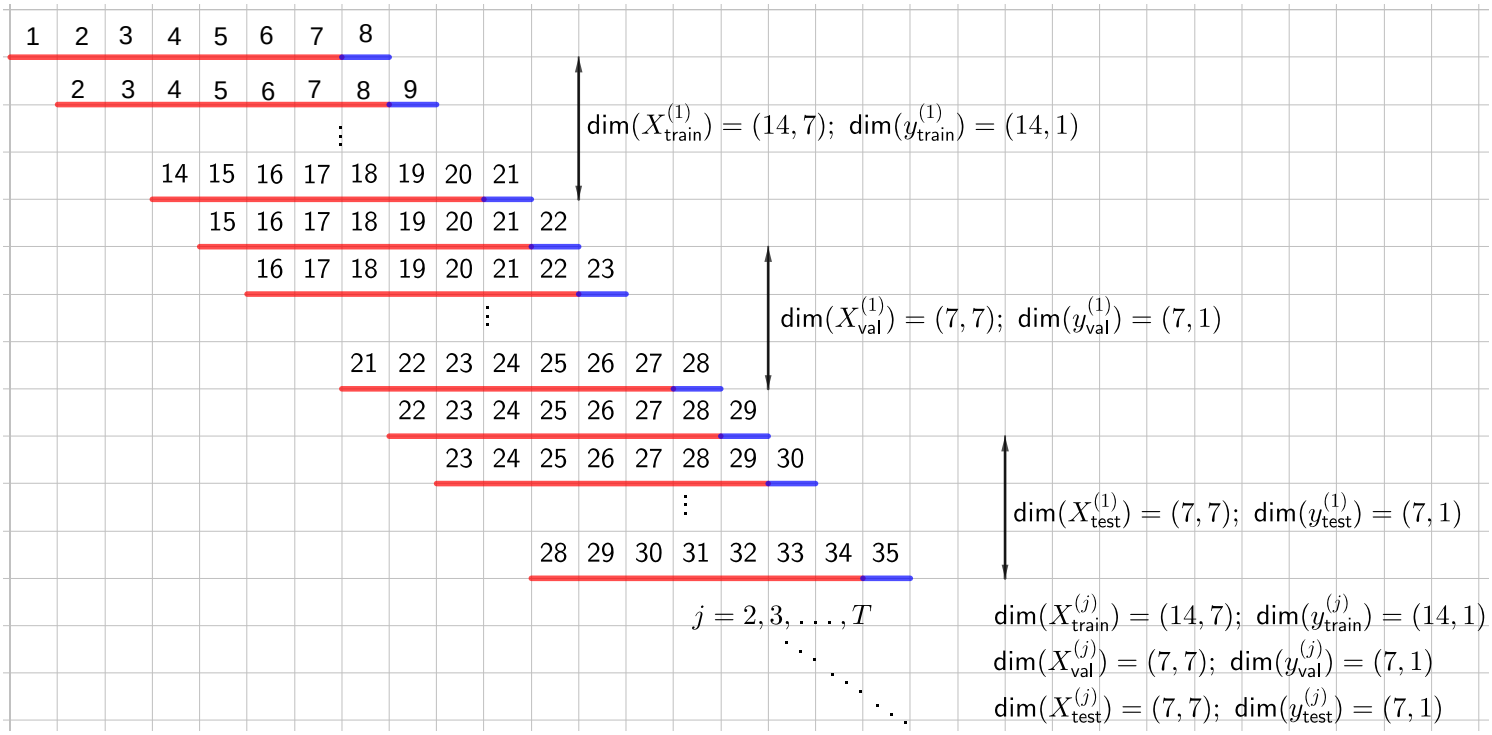

Figura 2: Conjuntos de entrenamiento, validación y test para modelos de Deep Learning. Valor predicho es el primer τ.

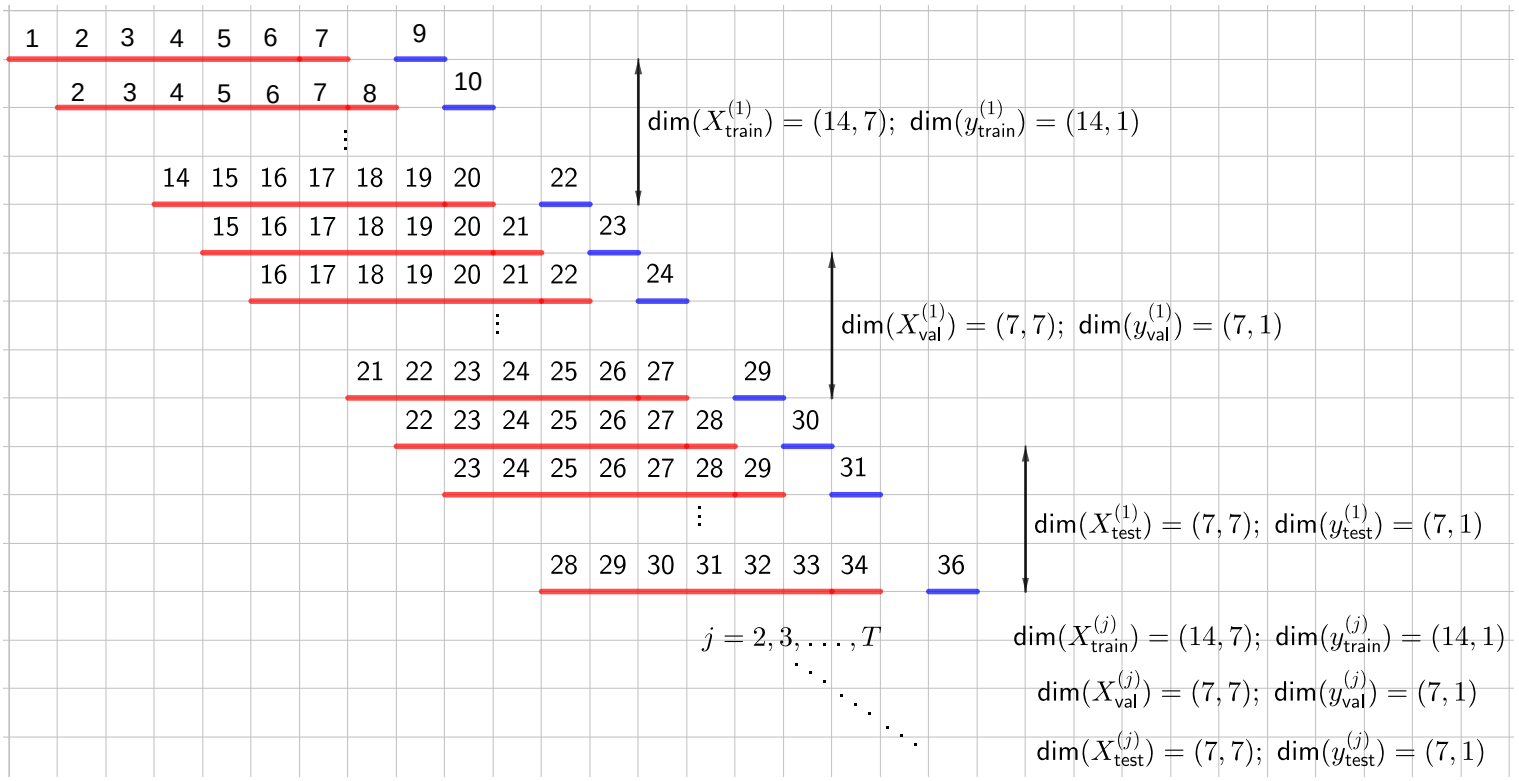

Figura 3: Conjuntos de entrenamiento, validación y test para modelos de Deep Learning. Valor predicho es el segundo valor de τ.

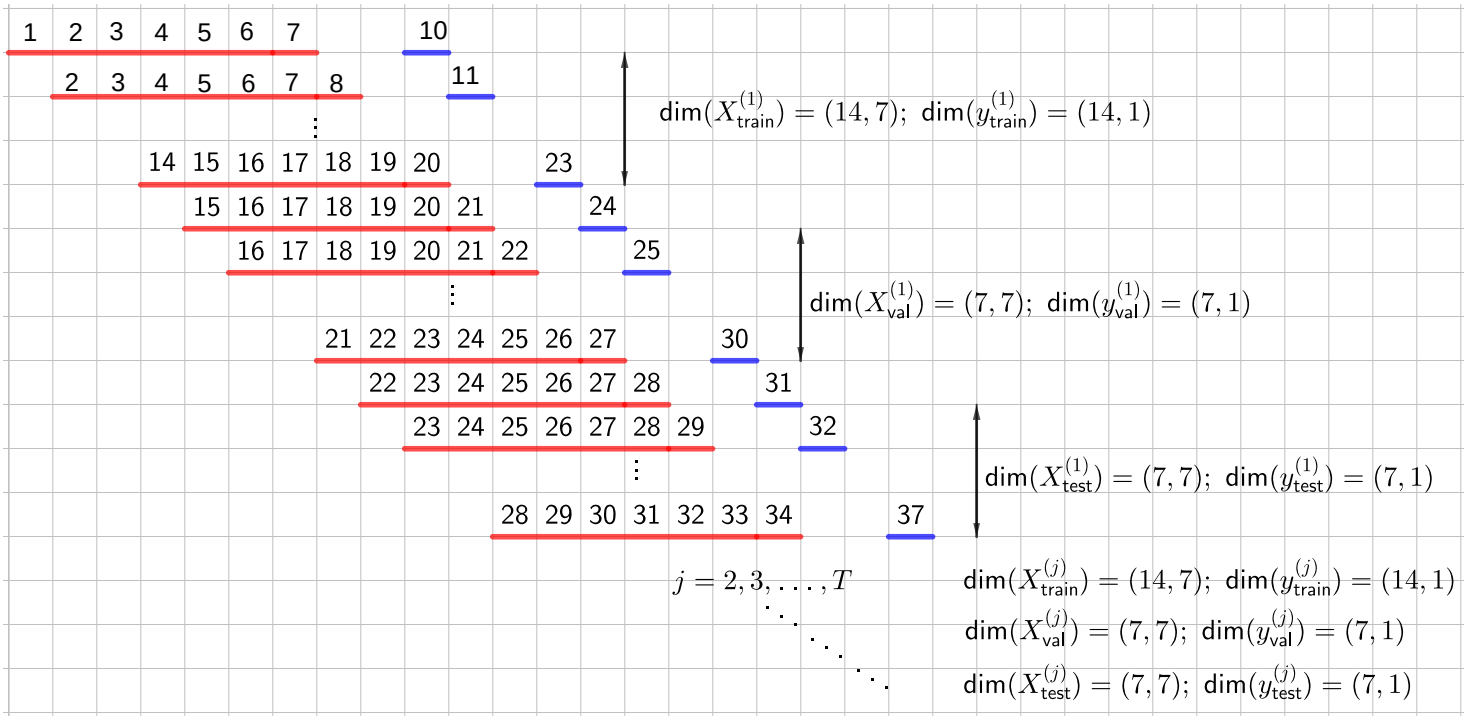

Figura 4: Conjuntos de entrenamiento, validación y test para modelos de Deep Learning. Valor predicho es el tercer τ. Nótese que este es un ejemplo. Debe extenderse considerando $\tau = 7, 14, 21, 28$.

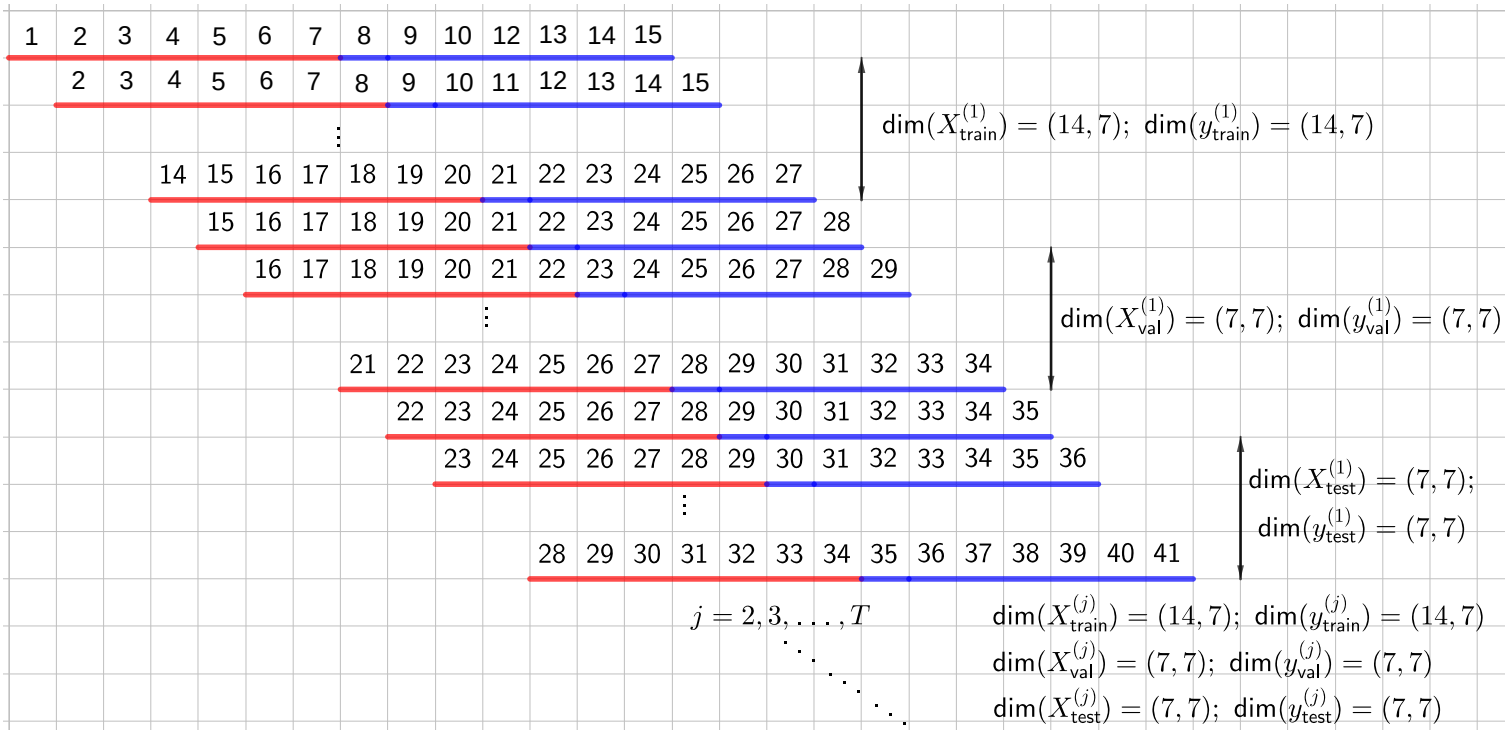

Figura 5: Conjuntos de entrenamiento, validación y test para modelos de Deep Learning. Valor predicho es el tercer τ. Nótese que este caso corresponde al horizonte τ completo de, en este caso 7 días.