Predicción de series financieras con redes neuronales recurrentes

Autor:
Edwin Jahir Rueda Rojas
ejrueda95g@gmail.com

Director:
Fabio Martinez Carrillo
famacar@saber.uis.edu.co

Co-director:
Raúl Ramos Pollán
rramosp@unal.edu.co

Universidad Industrial de Santander

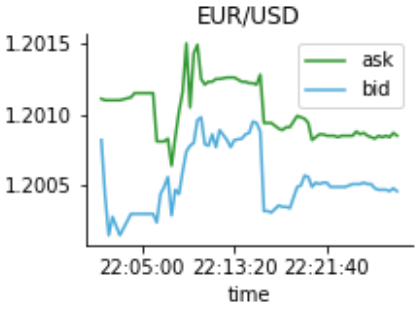
Super Computacion y Calculo Cientifico UIS

Agenda

- **Series financieras**
  - Estado del arte
  - Problema
  - Objetivos
- **Metodología**
  - Pre-procesado
  - Machine learning
  - Deep learning
  - Estrategia de *trading*
  - Validación de los modelos
- **Resultados**
- **Conclusiones y Perspectivas**

Series Financieras

Funcionamiento del mercado de Divisas



par de divisas

EUR/USD

t

moneda base

moneda cotizada

precio de compra

Bid

1.20093

precio de venta

Ask

1.20115

Ask - Bid

Spread

0.00022

Principales pares de divisas

- EUR/USD
- USD/JPY
- GBP/USD
- USD/CAD

tanto el valor del *bid* como el del *ask,* están en términos de la divisa cotizada, en este caso el USD.

Generación de ganancia

EUR/USD	Bid	Ask
t = 30	1.20079	1.20113
t = 35	1.20134	1.20154

$$Compro\,(t = 30) = 1.20113$$

$$Vendo\,(t = 35) = 1.20134$$

$$Ganancia = 0.00021$$

Estado del arte

Figura 1. Tipos de interacción con el mercado

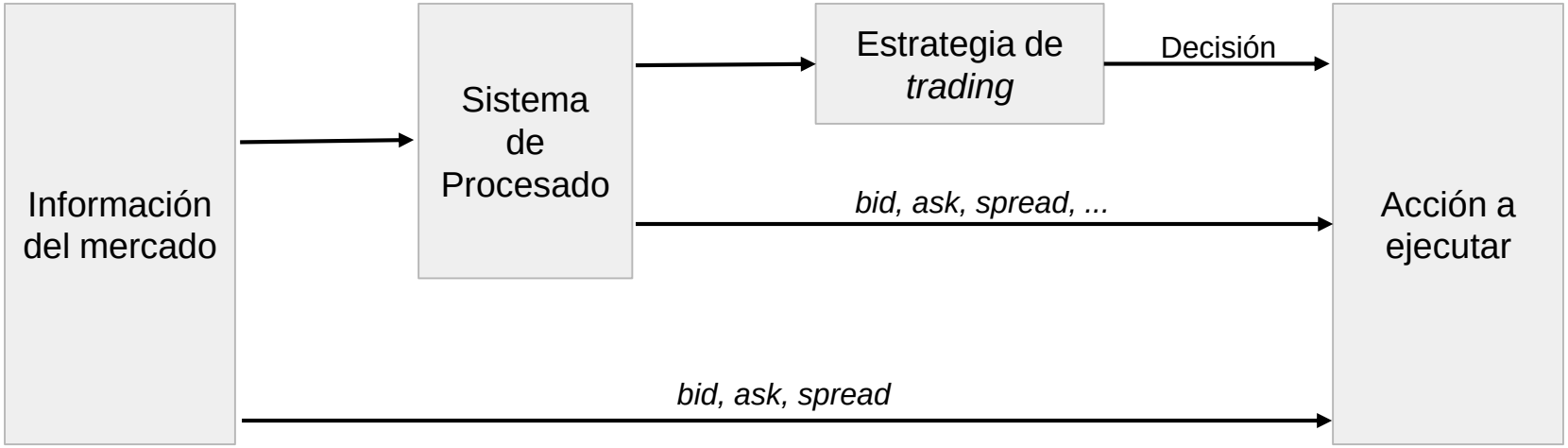
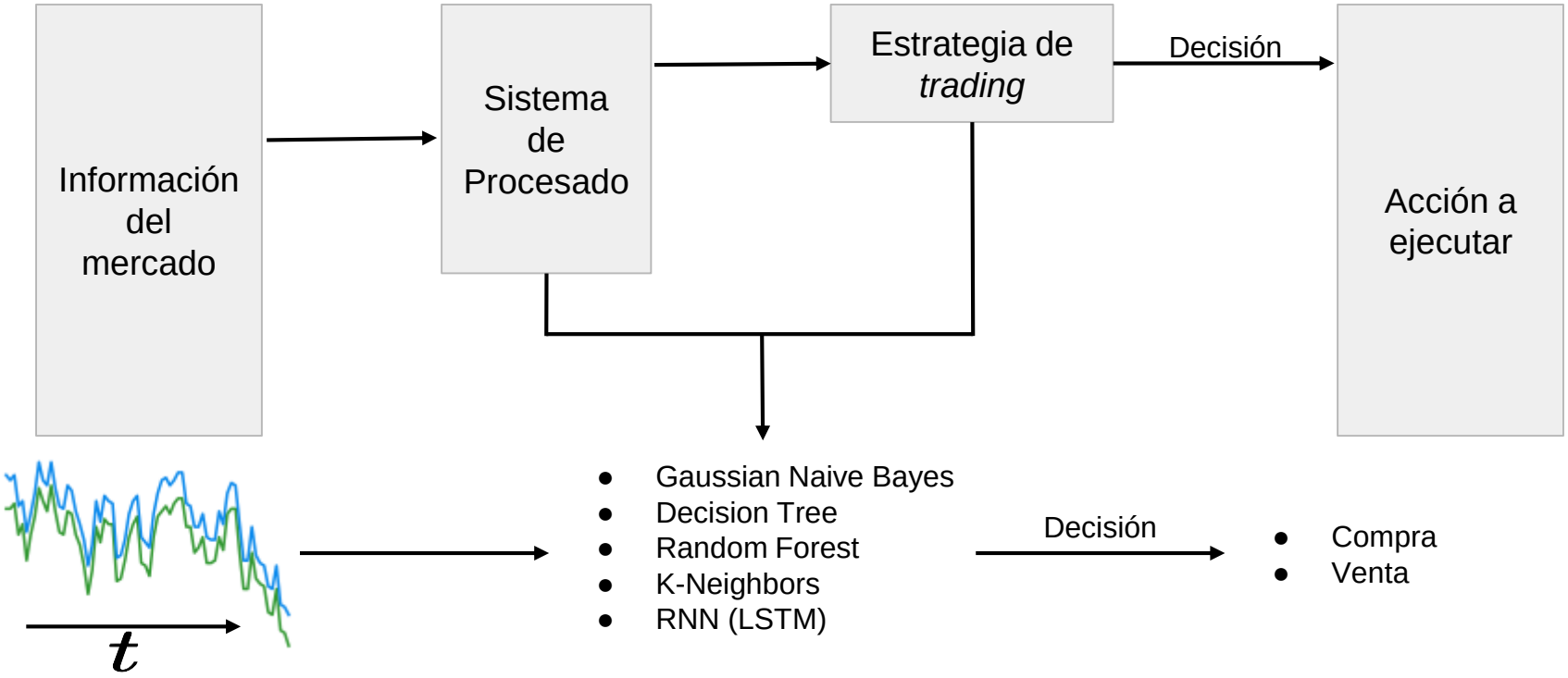


Figura 1. Lai, Ping-fu (Brian)-Wong Chung Hang, Performance of Stock Market Prediction. trading strategic decision.

Estado del arte

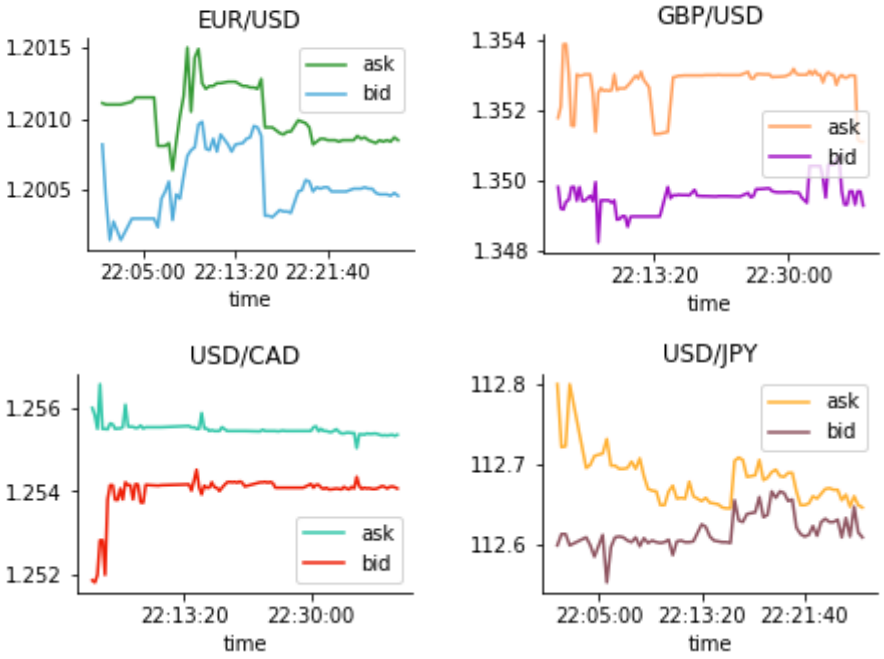
Figura 1. Tipos de interacción con el mercado



Problema de la predicción de series financieras

Los diferentes tipos de series financieras tienen un problema en común y es la volatilidad del precio a través del tiempo, siendo así el mercado de las divisas uno de los mercados más volátiles y significantes en el mundo, representando el volumen más alto.

Además, proponer una estrategia de *trading* acorde a los resultados de un análisis exhaustivo de los datos, es algo que se considera de suma importancia.



Objetivo General

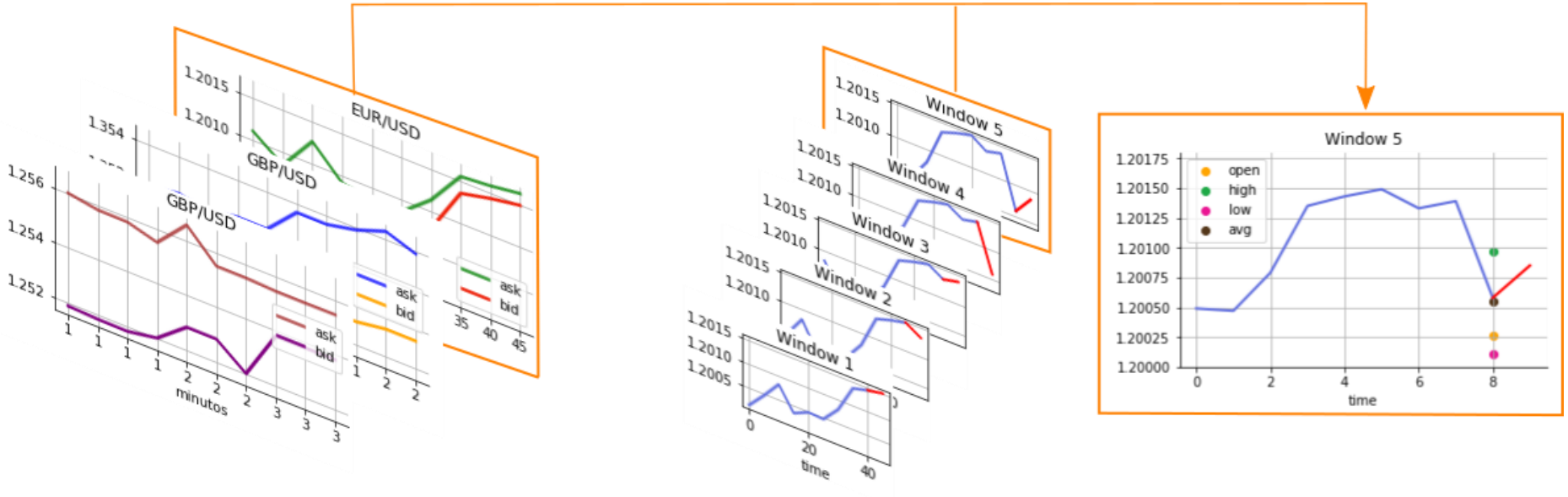
“Diseñar y evaluar redes neuronales recurrentes para la predicción de señales financieras basadas en múltiples señales”

Objetivos Específicos

- Adquirir datos financieros de diferentes fuentes.
- Desarrollar un pre-procesado de los datos multiseñal obtenidos anteriormente.
- Establecer métricas de predicción y tiempo de ejecución para evaluar el rendimiento del objetivo.
- Establecer una línea base de desempeño predictivo con métodos clásicos de *machine learning.*
- Realizar una exploración de arquitecturas RNNs y configuraciones multiseñal.
- Evaluar el rendimiento de los modelos predictivos y proponer estrategias de *trading.*

Metodología

Metodología propuesta para la predicción de series financieras



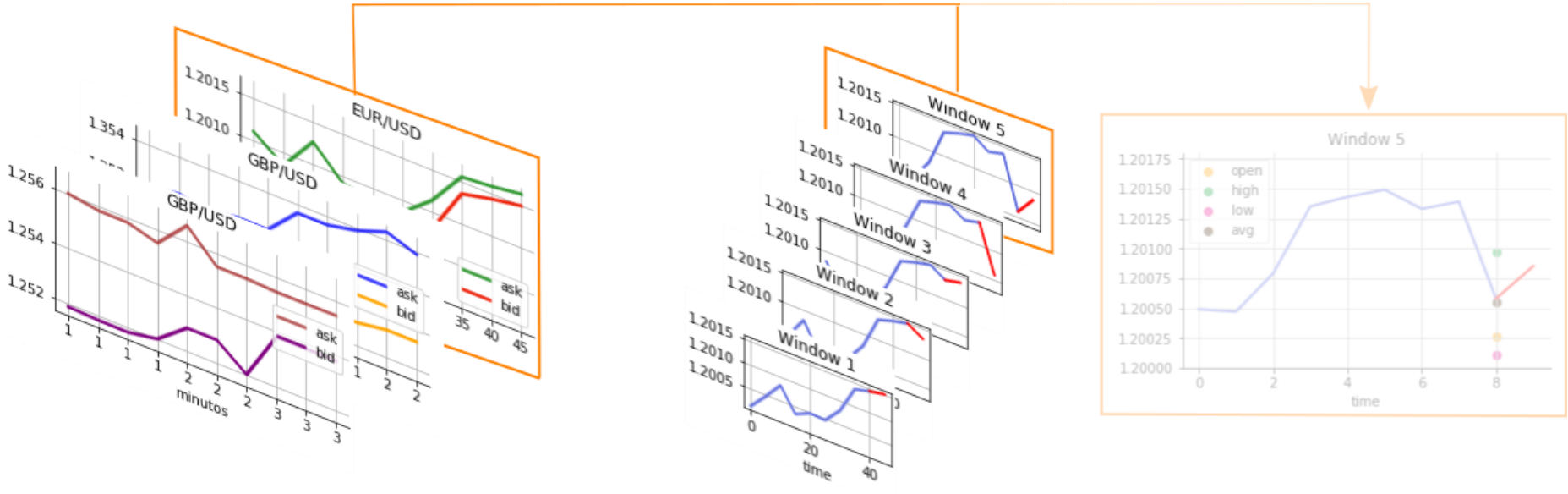
Time	Window + features				Target
0	1.20015	1.20047	...	1.20079	1
5	1.20047	1.20082	...	1.20135	0
10	1.20080	1.20075	...	1.20130	-

Machine Learning
Deep Learning

Estrategia de *trading*

Acción a ejecutar

Remuestreo de la señal y generación de ventana



Time	Window + features				Target
0	1.20015	1.20047	...	1.20079	1
5	1.20047	1.20082	...	1.20135	0
10	1.20080	1.20075	...	1.20130	-

Machine Learning
Deep Learning

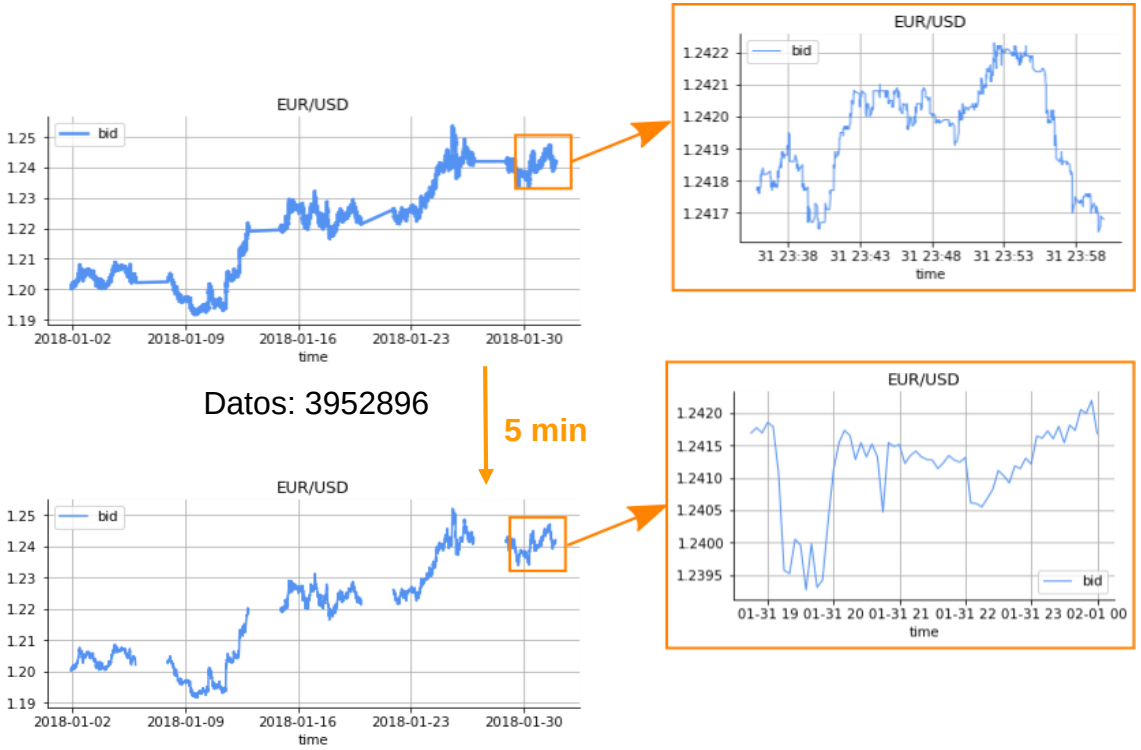
Estrategia de *trading*

Acción a ejecutar

Remuestreo de la señal y generación de ventana

Las señales iniciales presentan un delta de tiempo no homogéneo y demasiado corto como para generar ingresos debido a la estrategias de *trading* propuestas más adelante.

Se recurre a hacer un remuestreo con un delta de tiempo igual a cinco minutos.

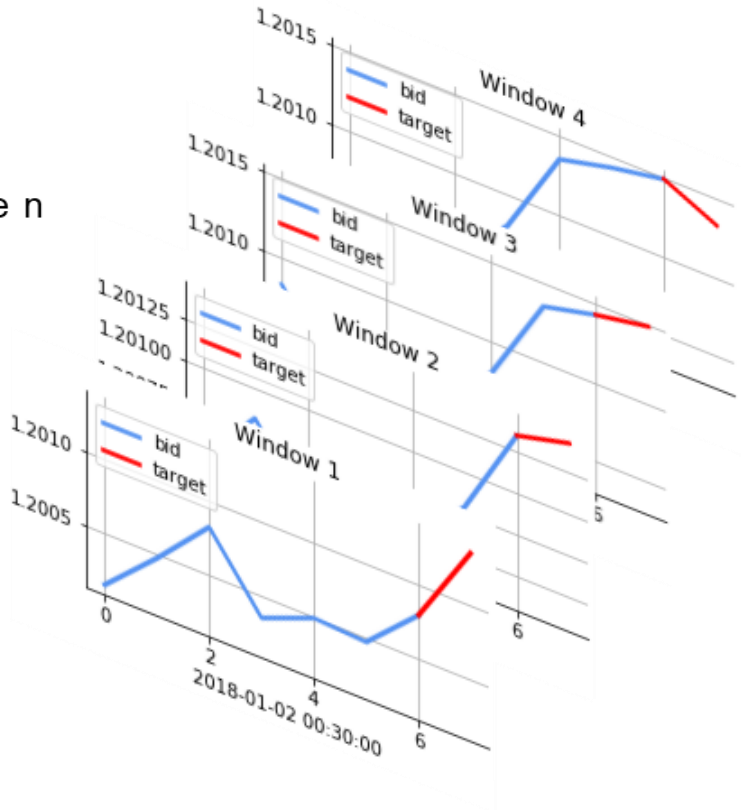


Remuestreo de la señal y generación de ventana

Siendo X la señal a la cual se le aplicó el remuestreo, un tamaño de ventana 7 vendría dado así:

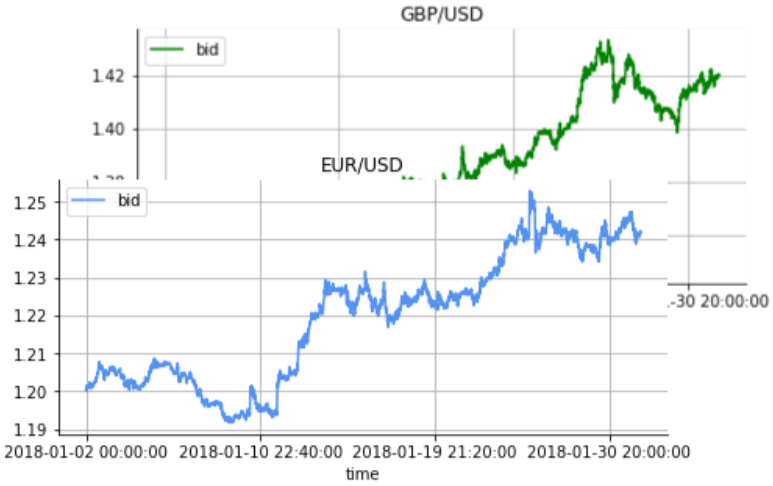
$$[X(t=1), X(t=2), X(t=3), \ldots, X(t=7)] \rightarrow X(t=8)$$

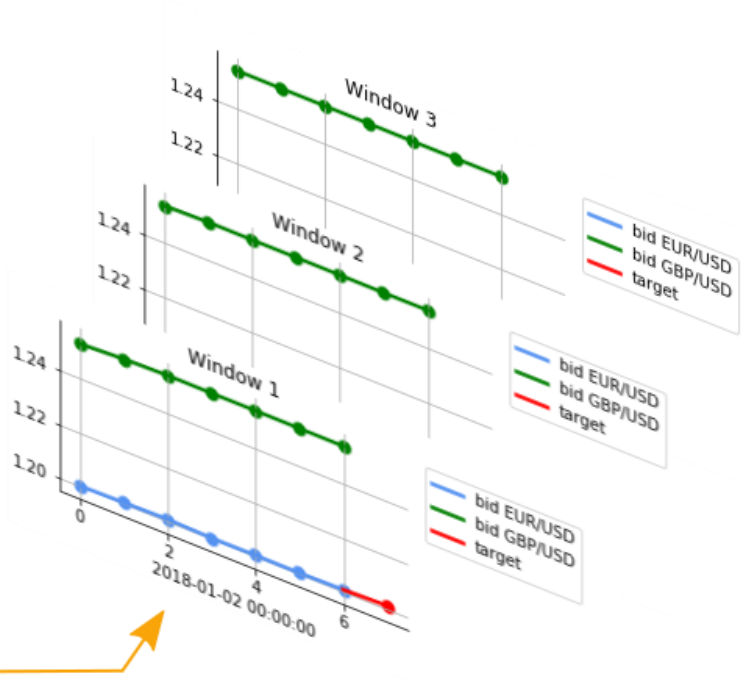
Donde el dataframe resultante tendrá un tamaño nx8, donde n es el número de filas de la señal X.



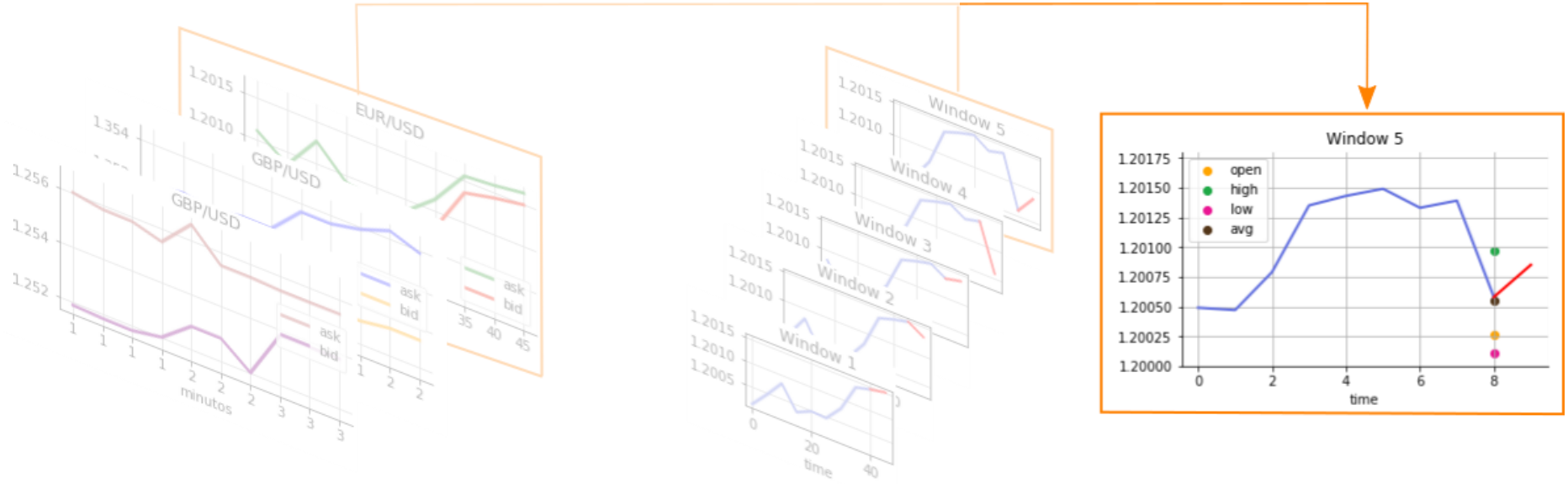
Para el caso de las multiseñales

Para las multiseñales, a la ventana de características de la señal a predecir se le agrega la ventana de las características de las diferentes señales que se contemplen útiles.





Extracción de características



Time	Window + features				Target
0	1.20015	1.20047	...	1.20079	1
5	1.20047	1.20082	...	1.20135	0
10	1.20080	1.20075	...	1.20130	-

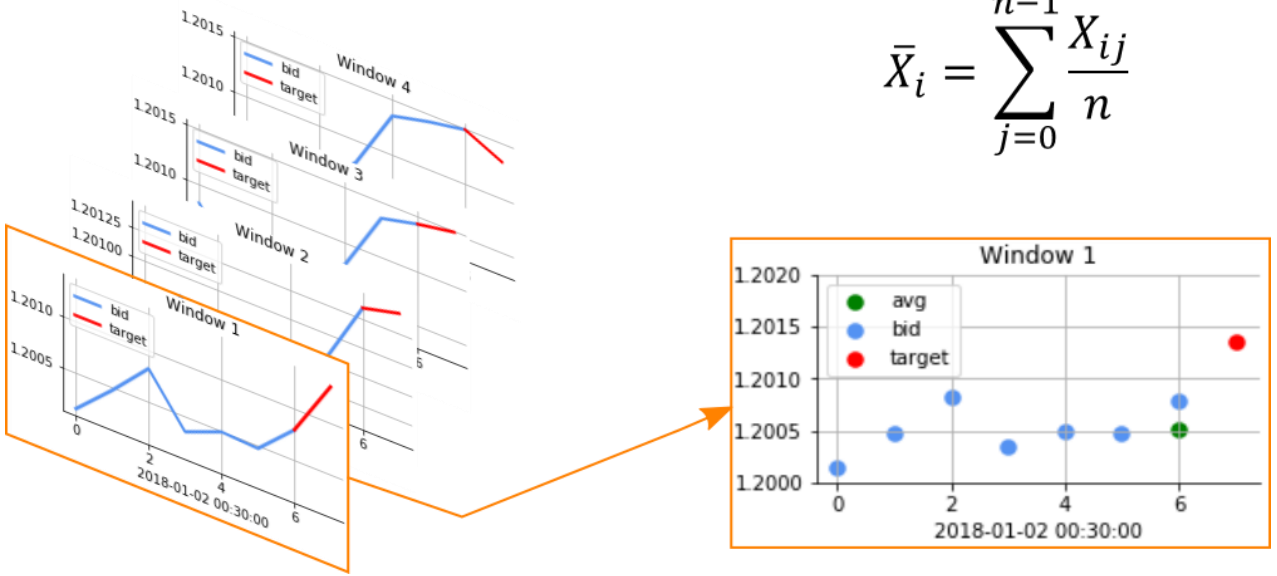
Machine Learning
Deep Learning

Estrategia de *trading*

Acción a ejecutar

Extracción de características

La extracción de características es fundamental para enriquecer cada ventana debido a que solo se cuenta con los datos del precio de las divisas, para eso se recurre a calcular el promedio de cada ventana, siendo Xij[t] la señal procesada en un muestreo de frecuencia t, de fila i, donde cada fila representa una ventaja y cada j una muestra de dicha ventana, la media de cada ventana i está dada por:

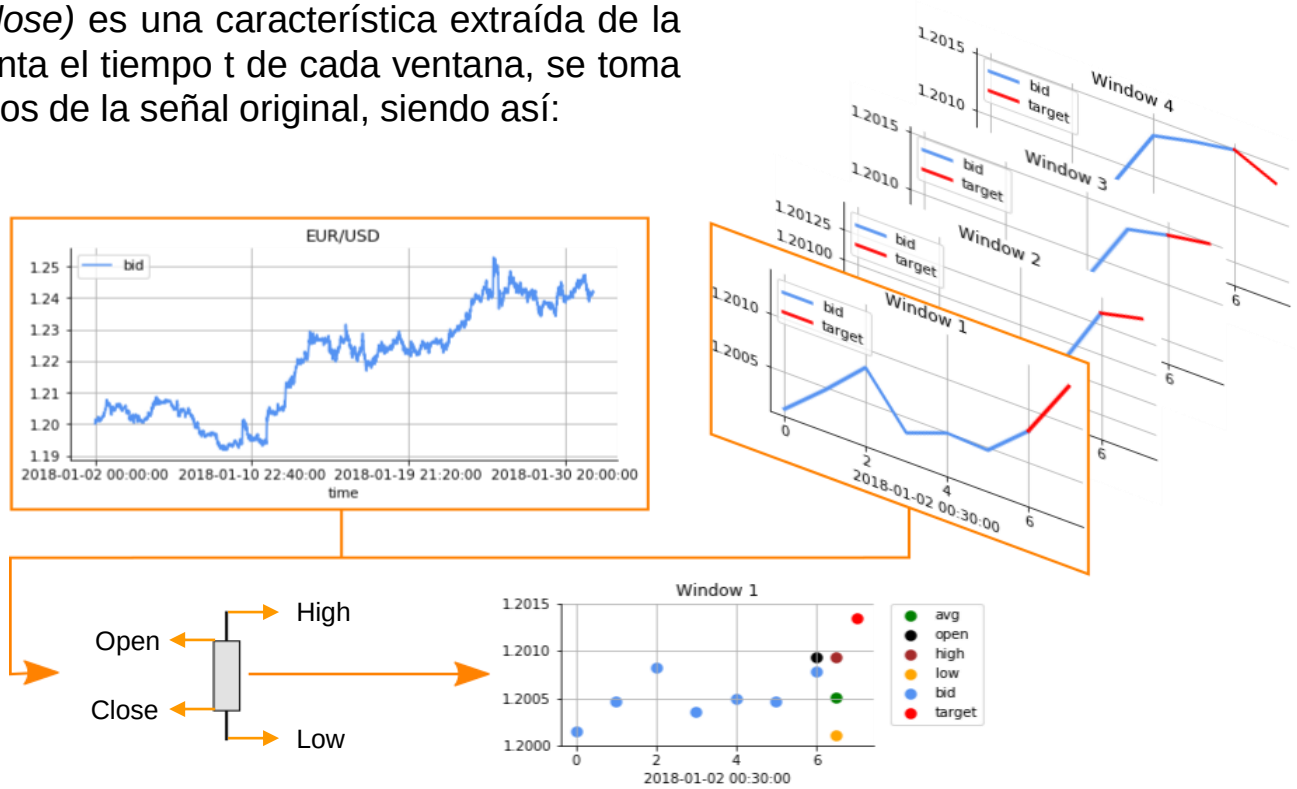


$$\bar{X}_i = \sum_{j=0}^{n-1} \frac{X_{ij}}{n}$$

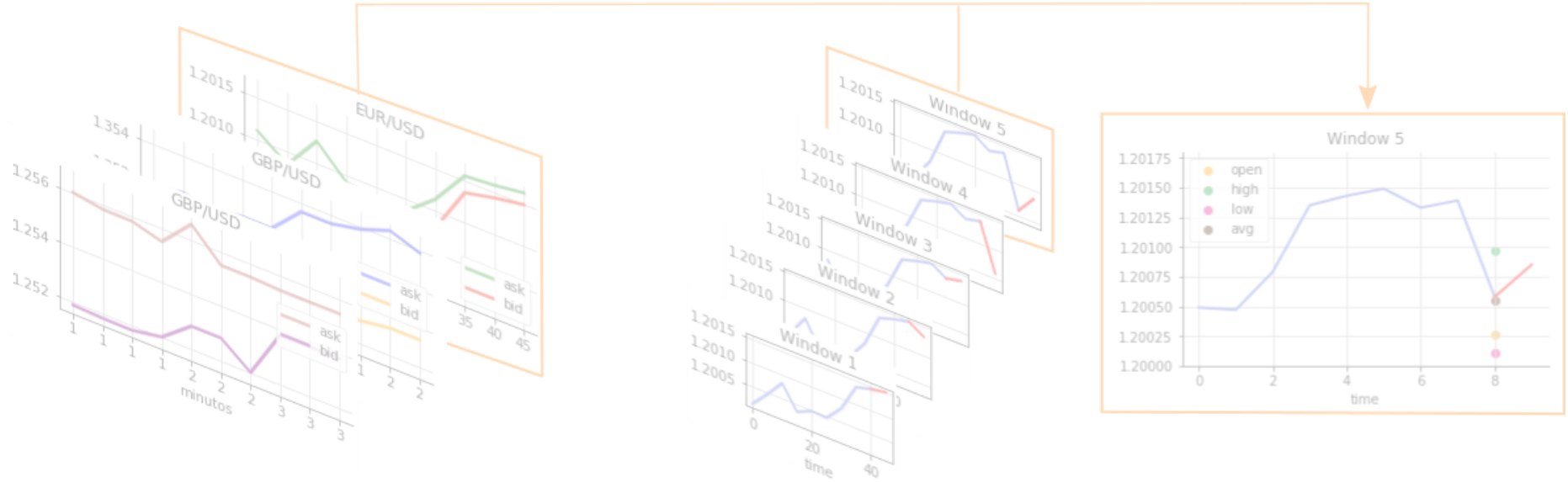
Extracción de características

El *OHLC* (*Open, High, Low, Close)* es una característica extraída de la señal original, teniendo en cuenta el tiempo t de cada ventana, se toma el *OHLC* de los últimos 5 minutos de la señal original, siendo así:

Cabe resaltar que el valor *'Close'* no se muestra en la gráfica resultante, debido a que este es igual al último valor del *bid* para cada ventana.



Construcción del conjunto de datos



Time	Window + features				Target
0	1.20015	1.20047	...	1.20079	1
5	1.20047	1.20082	...	1.20135	0
10	1.20080	1.20075	...	1.20130	-

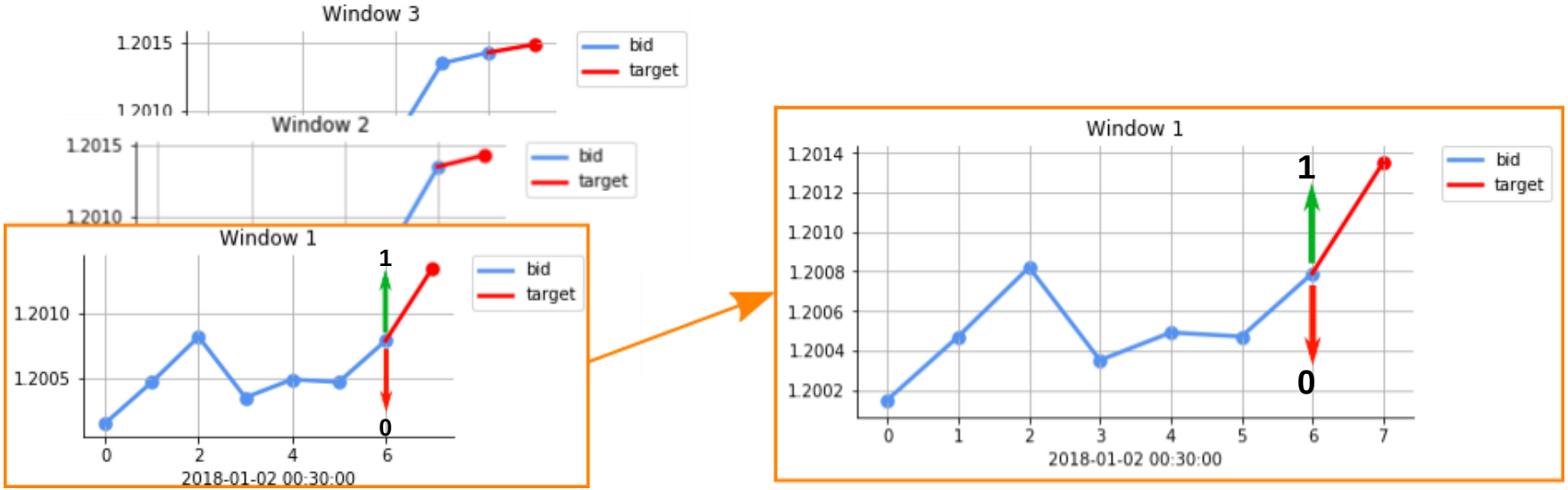
Machine Learning
Deep Learning

Estrategia de *trading*

Acción a ejecutar

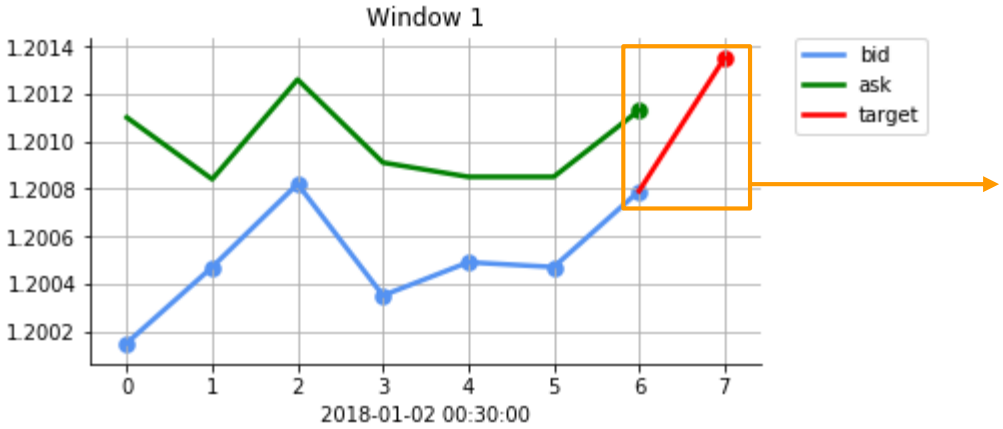
Construcción del conjunto de datos

Se plantean dos métodos para la construcción del conjunto de datos, el primero basándose solo en la señal del *bid*, y el segundo teniendo en cuenta tanto la señal del *bid* como la del *ask.* A continuación se muestra el método uno, el cual categoriza el *target* como 1 si el *bid* tiende a la alta ó 0 si tiende a la baja.



Construcción del conjunto de datos

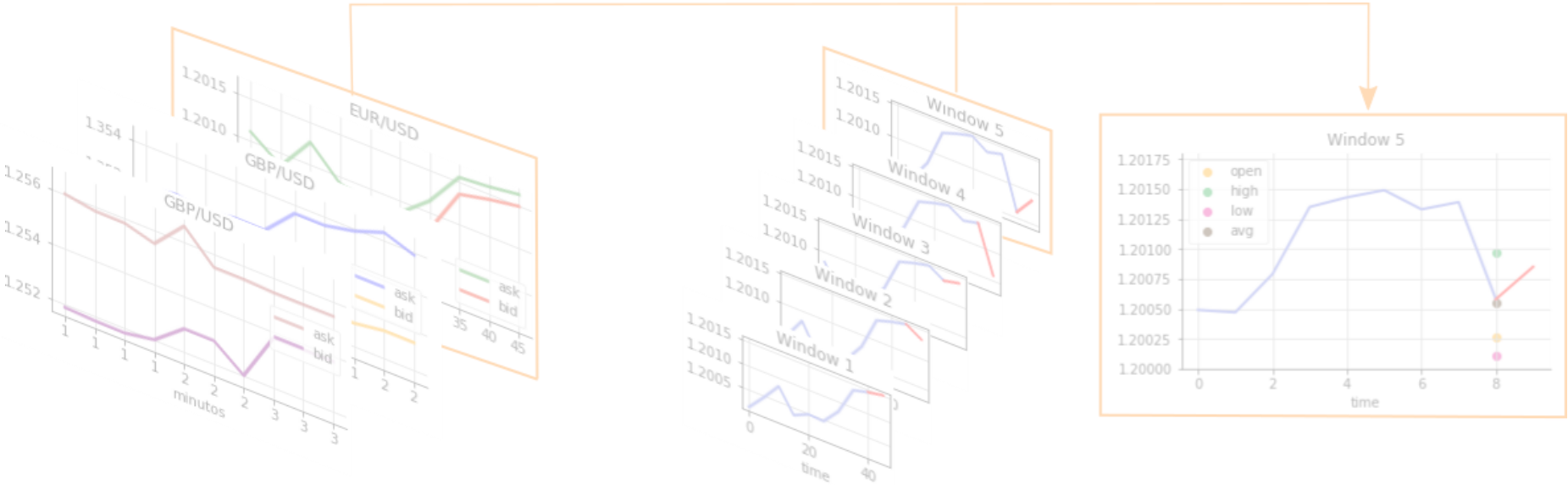
El segundo método propuesto consiste en categorizar el *target* en cuatro distintas clases dependiendo del comportamiento tanto del *bid*, como del *ask.*



$$\boldsymbol{bid(t+1) > bid(t)}$$
$$bid(t+1) > ask(t) \rightarrow 1, sube\ y\ gano$$
$$bid(t+1) < ask(t) \rightarrow 3, sube\ y\ pierdo$$

$$\boldsymbol{bid(t+1) < bid(t)}$$
$$ask(t+1) < bid(t) \rightarrow 0, baja\ y\ gano$$
$$ask(t+1) > bid(t) \rightarrow 2, baja\ y\ pierdo$$

Aprendizaje supervisado: *Machine Learning & Deep Learning*



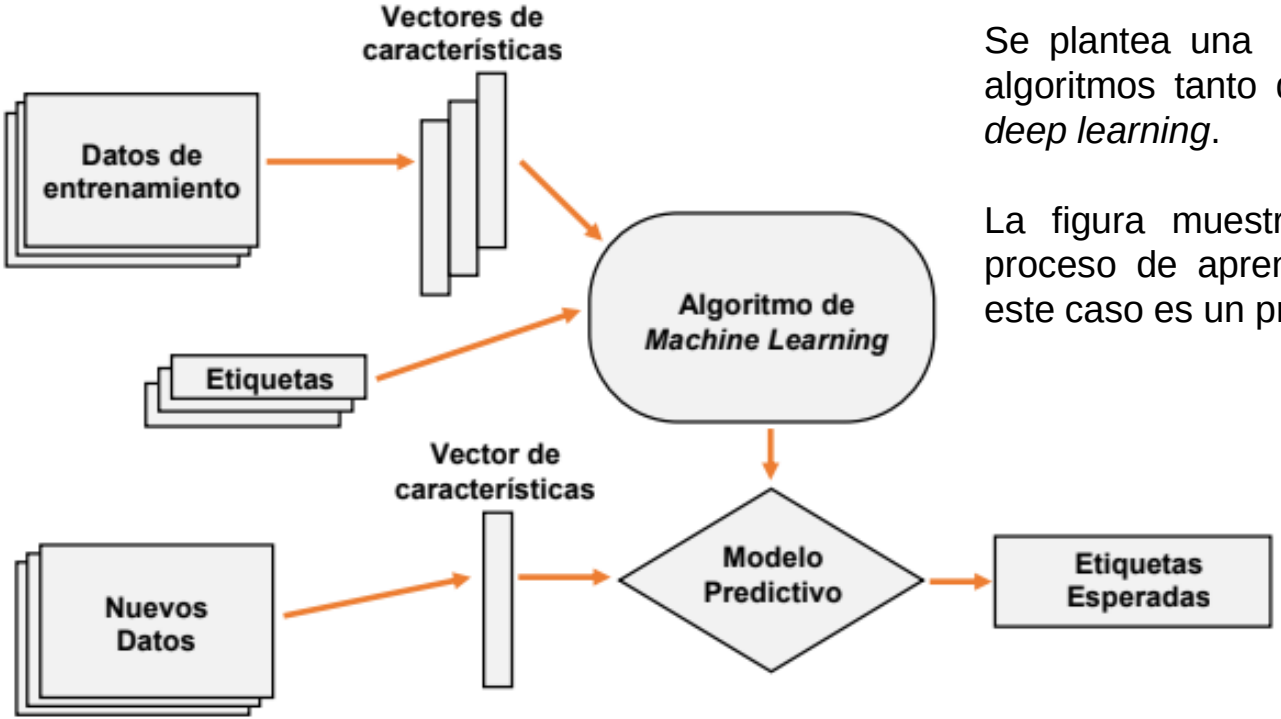
Time	Window + features				Target
0	1.20015	1.20047	...	1.20079	1
5	1.20047	1.20082	...	1.20135	0
10	1.20080	1.20075	...	1.20130	-

Machine Learning
Deep Learning

Estrategia de *trading*

Acción a ejecutar

Aprendizaje supervisado: *Machine Learning & Deep Learning*



Se plantea una tarea de clasificación usando algoritmos tanto de *machine learning* como de *deep learning*.

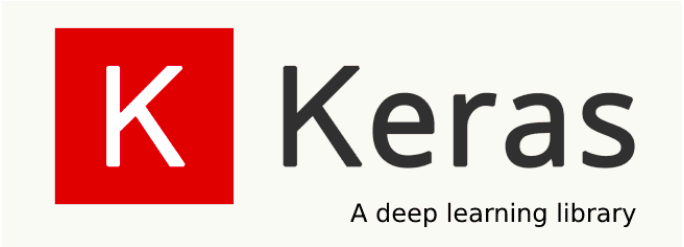
La figura muestra la dinámica que sigue el proceso de aprendizaje supervisado, que para este caso es un problema de clasificación.

Algoritmos de clasificación

Con el fin de plantear una línea base de precisión, se utilizaron los siguientes algoritmos de *machine learning*:

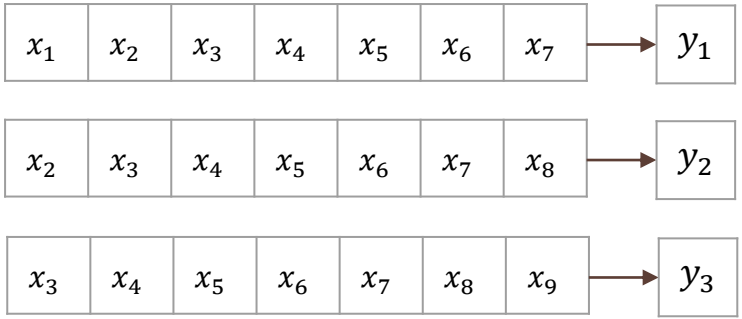
- Naive bayes
- K-Neighbors
- Árboles de decisión
- Bosques aleatorios

Por último se recurre a la utilización del *deep learning* usando las redes neuronales recurrentes con unidades LSTM (*Long Short Term Memory*).



Naive bayes

Es basado en el Teorema de Bayes, con una suposición ingenua (*naive*) sobre la independencia de los datos. El algoritmo asume que la probabilidad de las características es Gaussiana, la cual se obtiene de la siguiente manera:

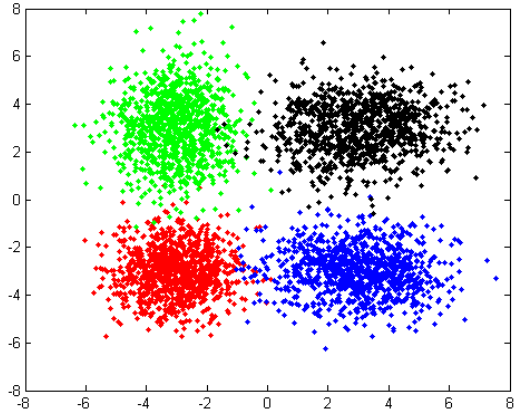


$$P(X_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(X_i - \mu_y)^2}{2\sigma_y^2}\right)$$

K-Neighbors

La clasificación basada en vecinos, es un tipo de aprendizaje basado en instancias, o no generalizado, el cual no intenta construir un modelo interno general, sino que almacena instancias de los datos de entrenamiento. La clasificación se calcula a partir de un voto de mayoría simple de los K vecinos más cercanos del punto, siendo la distancia Euclidiana la más comúnmente utilizada:

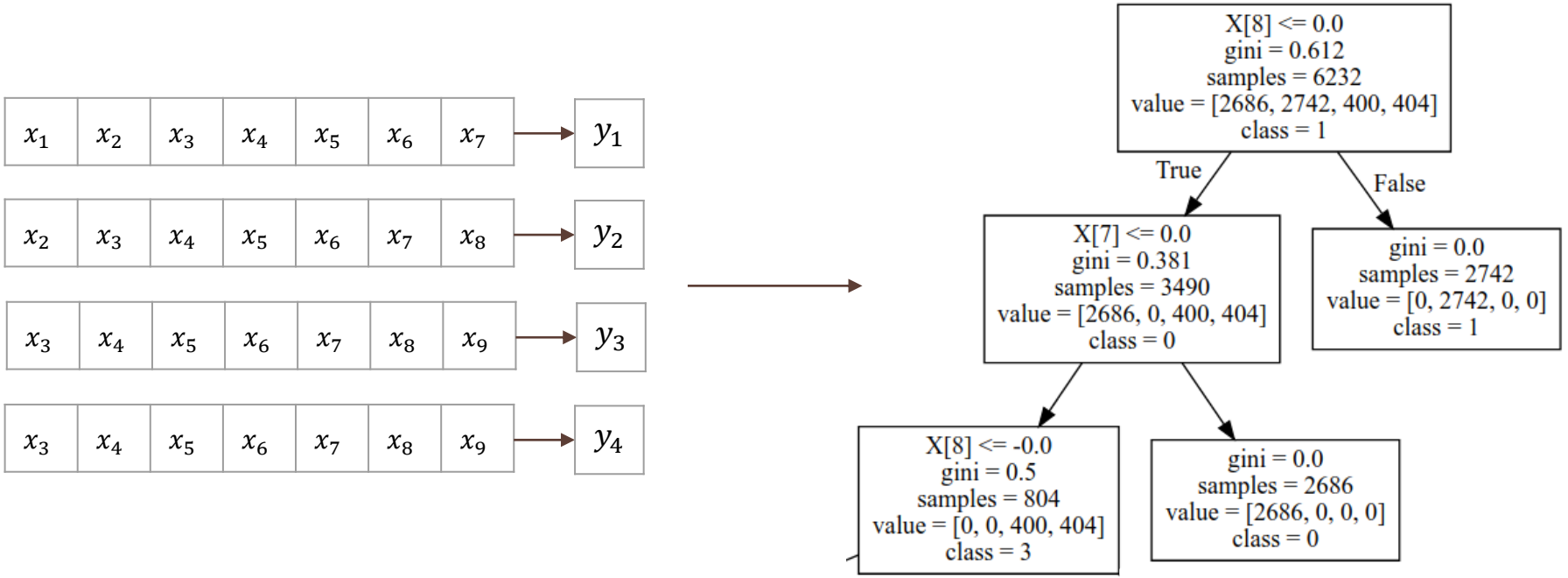
x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_1
x_2	x_3	x_4	x_5	x_6	x_7	x_8	y_2
x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_3
x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_4



$$d(x, x') = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + \cdots + (x_n - x_n')^2}$$

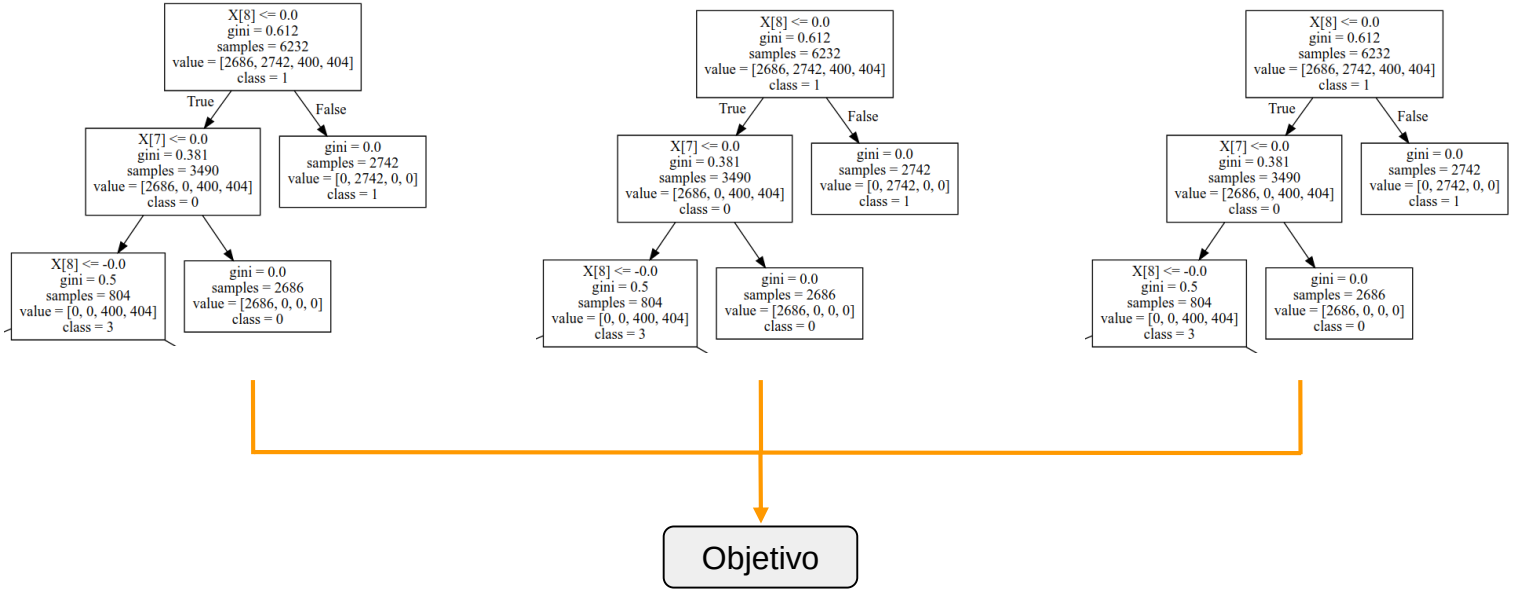
Árboles de decisión

Un árbol de decisión es un grafo dirigido y sin bucles, cuyo objetivo es predecir el valor esperado por una serie de reglas de decisión simples.



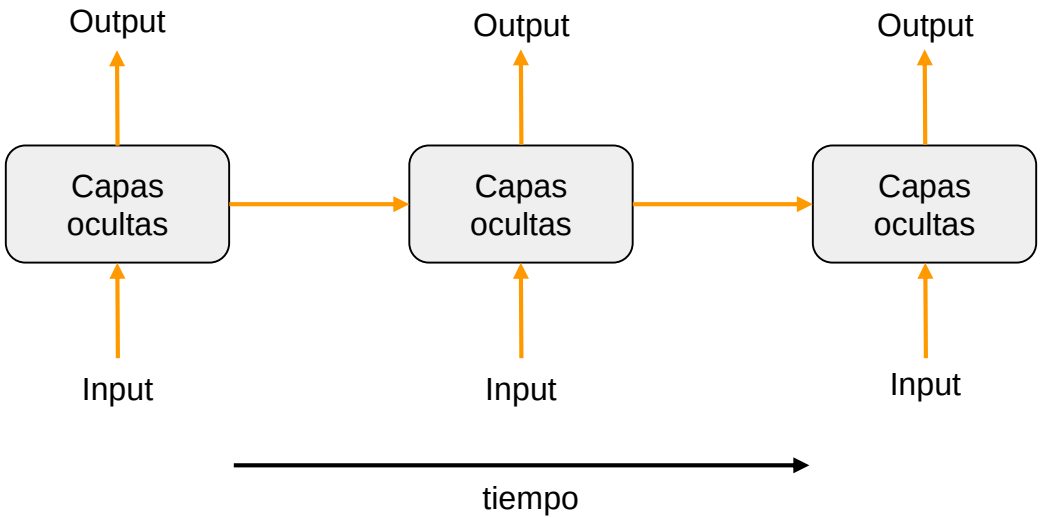
Bosques aleatorios

Un bosque aleatorio entrena un número de clasificadores de árboles de decisión en varias submuestras del conjunto de datos de entrada y utiliza el promedio de estas predicciones para así dar una predicción con un ajuste menos excesivo.



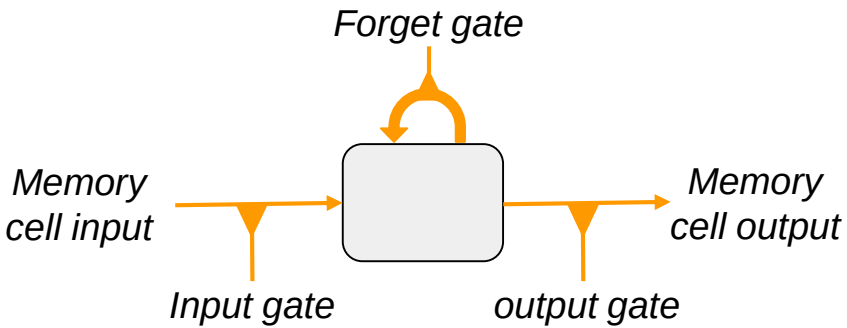
Redes neuronales recurrentes

Las RNN's están clasificadas en el ámbito del *deep learning* debido a su estructura robusta aplicada en capas la cual permite construir un modelo más robusto de lo que permiten los modelos de *machine learning*. A continuación se ilustra el comportamiento de una RNN's:

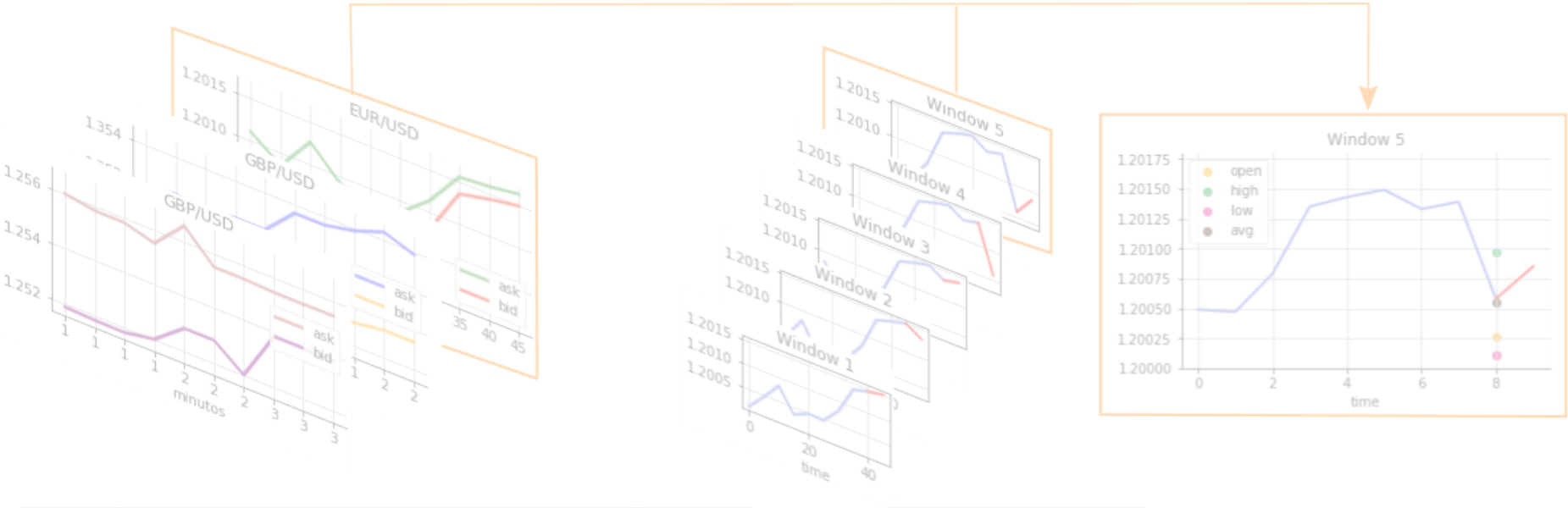


Unidades LSTM

Las LSTM desarrolladas por Hochreiter y Schmindhuber (1997) introducen una nueva estructura llamada *celda de memoria* la cual se ve ilustrada en la siguiente figura:



Planteamiento de la estrategia de *trading*



Time	Window + features				Target
0	1.20015	1.20047	...	1.20079	1
5	1.20047	1.20082	...	1.20135	0
10	1.20080	1.20075	...	1.20130	-

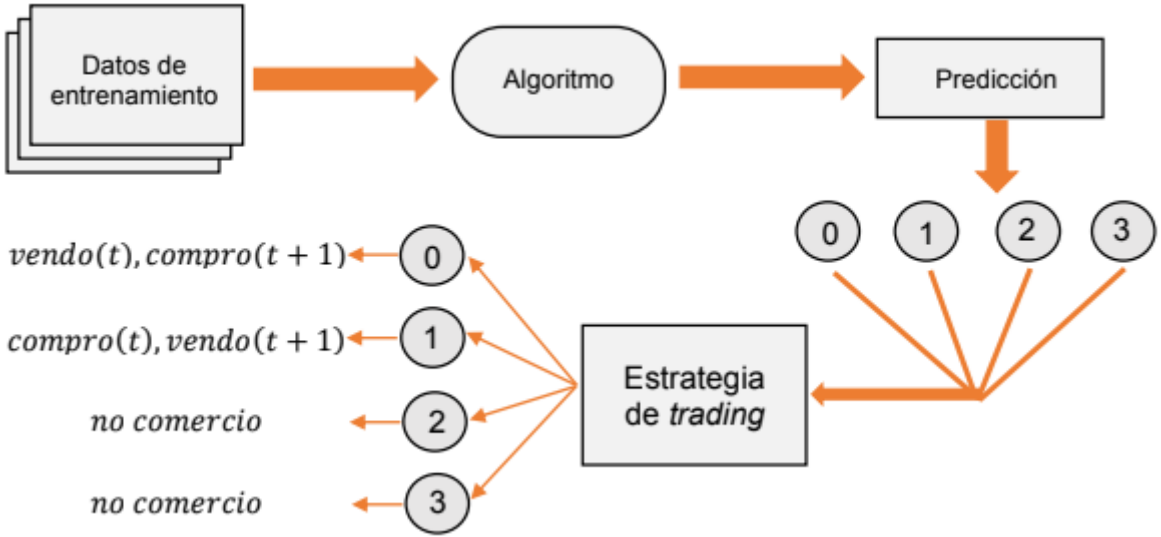
Machine Learning
Deep Learning

Estrategia de *trading*

Acción a ejecutar

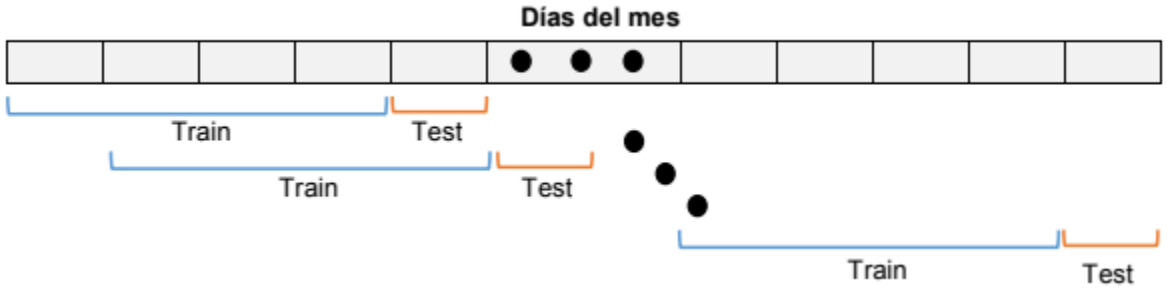
Planteamiento de la estrategia de *trading*

Con base a la estrategia de *trading* es que se calcula el rendimiento de los modelos planteados. La siguiente figura ilustra la estrategia de *trading* propuesta:



Validación de los modelos

Para validar los modelos planteados se tomó un criterio de validación por días, siendo así que cada modelo se entrenaba con cuatro días de datos y posteriormente se ponía a prueba con el siguiente día, así mismo se corría diariamente el modelo para ir prediciendo diariamente.



Métricas de rendimiento

Las métricas de rendimiento nos dicen qué tan robustos son los modelos planteados, para este trabajo se plantearon dos métricas principales, las cuales se mencionan a continuación:

- **Profit and Loss (PNL)**
  Esta métrica nos dice cuánto ganan o pierden nuestros modelos planteados al aplicarseles la estrategia de *trading* planteada.

Predicción	Operación	PNL 0	PNL 1
0	venta/compra	$venta(t) - compra(t+1)$	-
1	compra/venta	-	$venta(t+1) - compra(t)$
2	venta/compra	-	-
3	compra/venta	-	-

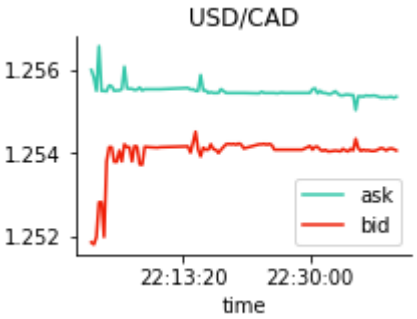
- **Precisión**
  La precisión del modelo se obtiene calculando los aciertos obtenidos de la siguiente manera:

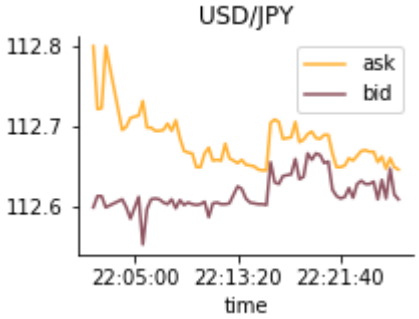
$$precisión = \sum_{i=0}^{n-1} \frac{y_{predict} = y_{target}}{n}$$

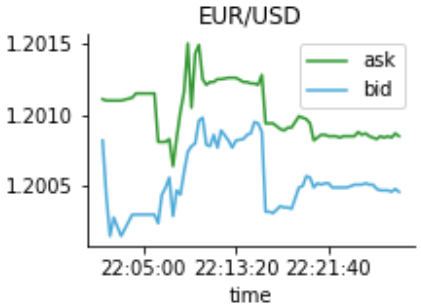
Resultados

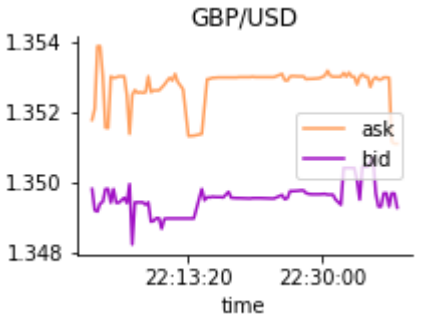
Resultados

La metodología propuesta fue evaluada con datos libres adquiridos de TrueFX.









TrueFX, historical-tick-by tick data. Free. truefx.com

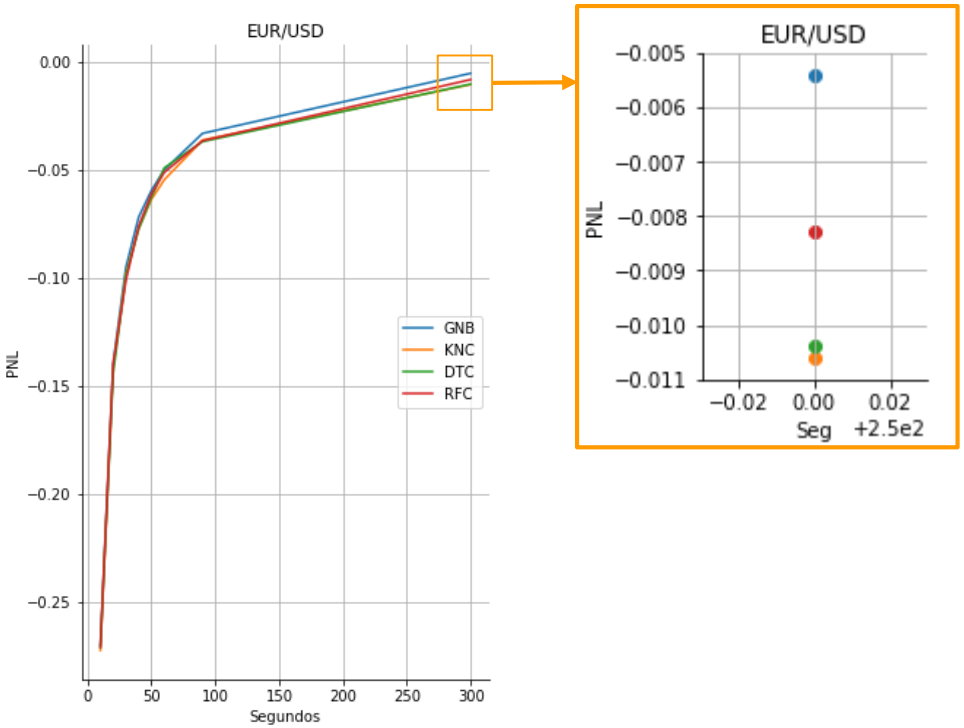
Resultados

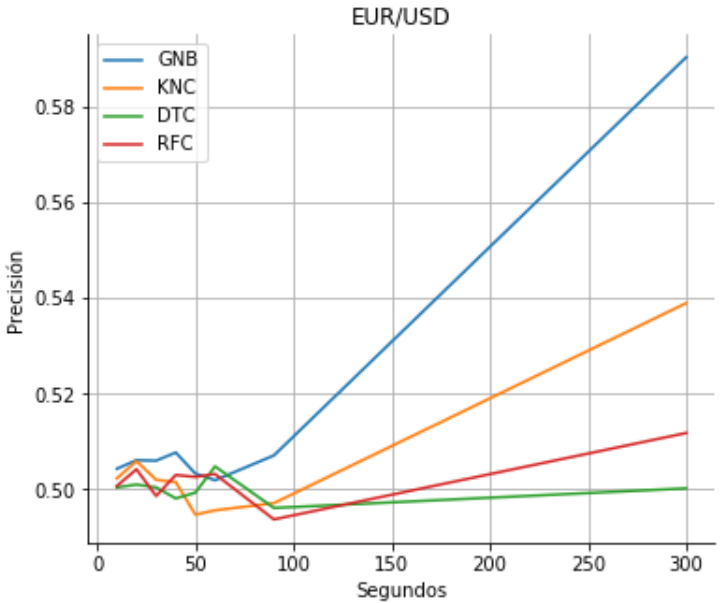
La siguiente tabla muestra la reducción en los datos dependiendo del remuestreo establecido:

Dimensión original	Muestreo	Nueva dimensión
(3952896,1)	10 [s]	(184135,1)
(3952896,1)	20 [s]	(94850,1)
(3952896,1)	30 [s]	(63473,1)
(3952896,1)	40 [s]	(47651,1)
(3952896,1)	50 [s]	(38128,1)
(3952896,1)	1 [min]	(31777,1)
(3952896,1)	1 [min] y 30 [s]	(21186,1)
(3952896,1)	5 [min]	(6360,1)

Resultados

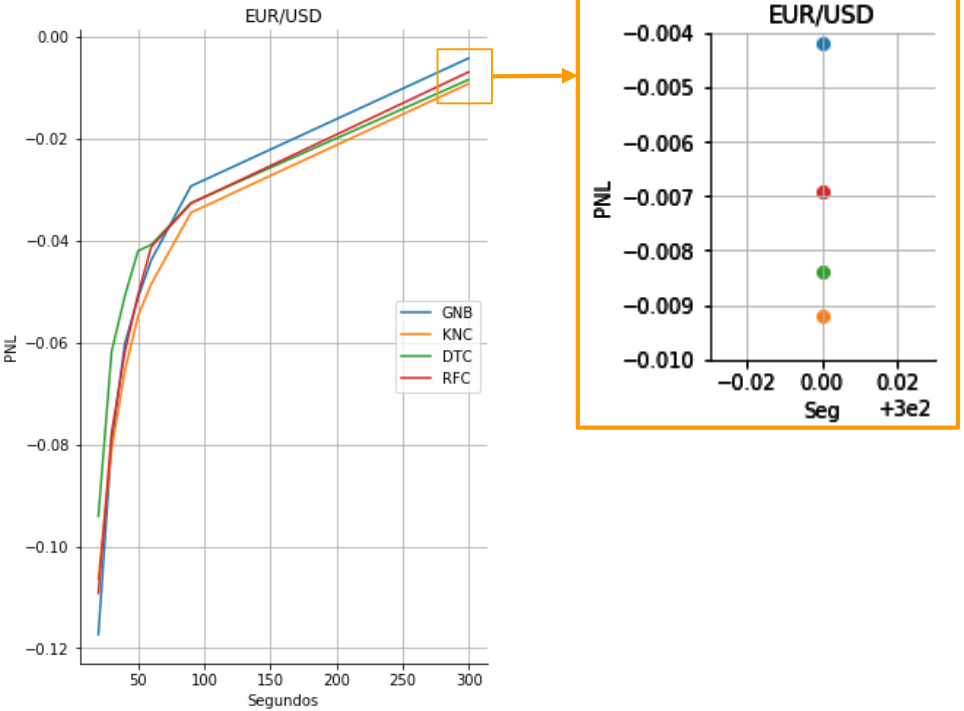
Los resultados mostrados a continuación corresponden al diseño de construcción de los datos en el cual se presentó un problema de clasificación binaria, uno o cero y el cual cuenta solo con las características de la ventana n=3 y su media.

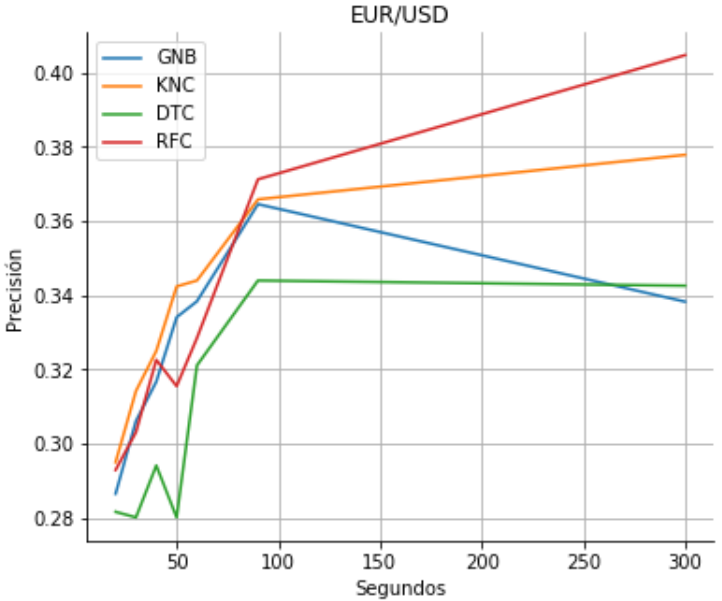




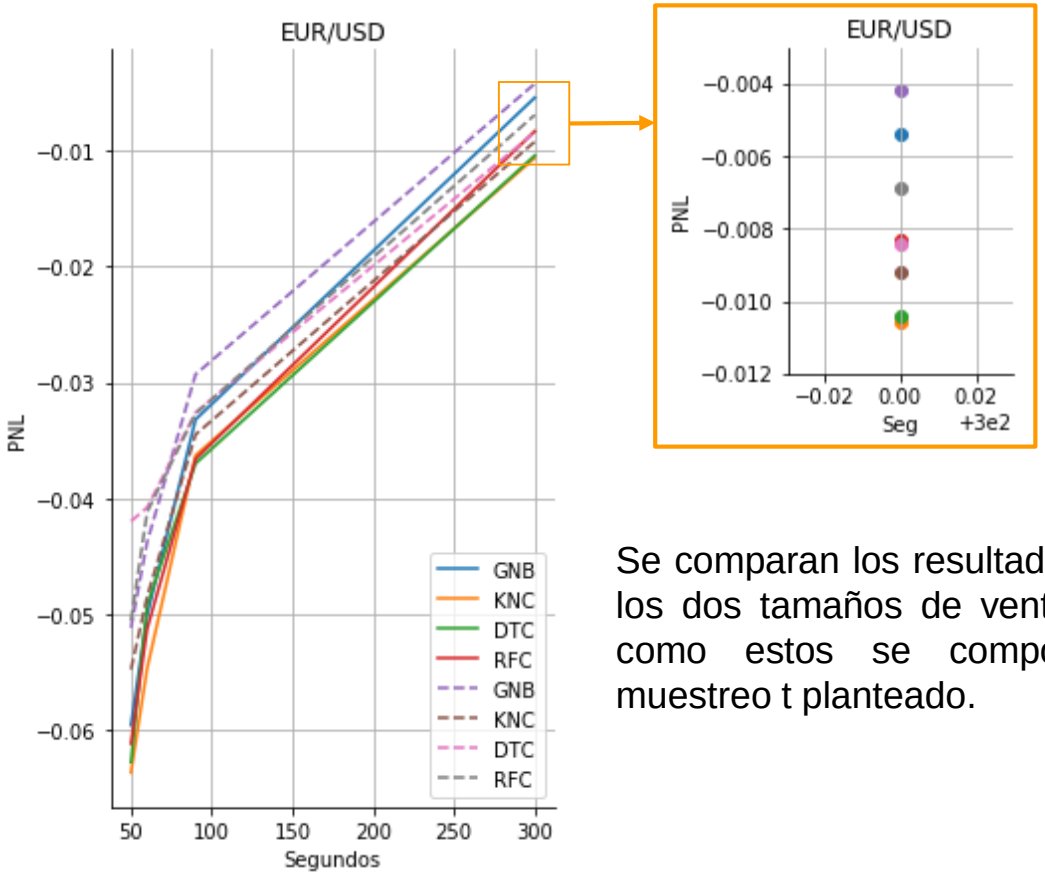
Resultados

Los resultados mostrados a continuación corresponden al diseño de construcción de los datos en el cual se presentó un problema de clasificación multiclase a cuatro clases con una ventana n=7 y con la media de esta como característica adicional.





Resultados



Se comparan los resultados obtenidos con los dos tamaños de ventana diferentes y como estos se comportan según el muestreo t planteado.

Resultados

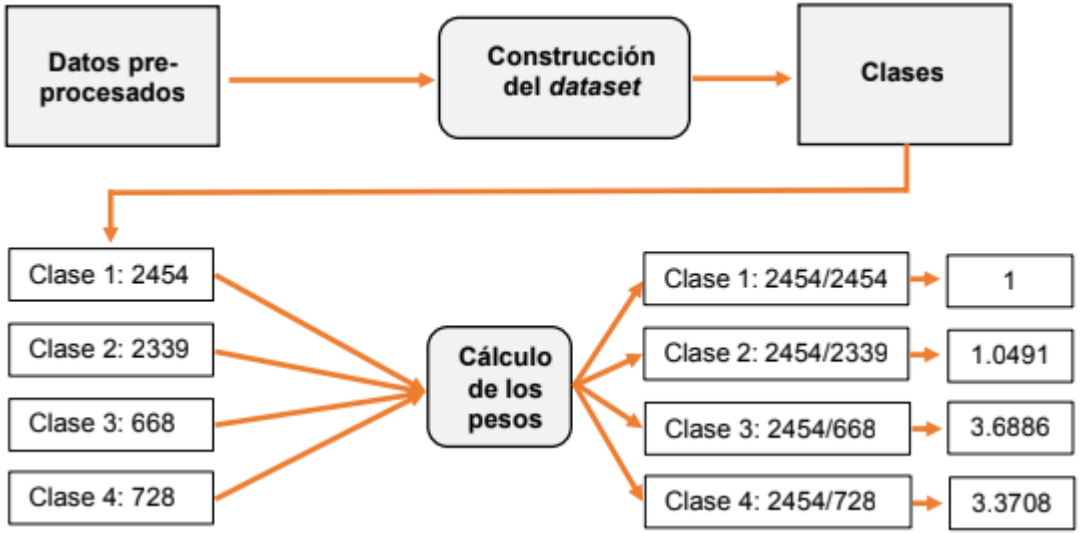
Se probó con una multiseñal, la cual tenía de entrada la señal del *bid* de los pares EUR/USD, GBP/USD y CAD/USD, para así predecir el comportamiento del par EUR/USD. La siguiente tabla muestra el resultado obtenido:

Clasificador	Muestreo	Tamaño de ventana	PNL	Precisión
GaussianNB	5 [min]	7	-0.0034	28.31%
Kneighbors	5 [min]	7	-0.0085	39.46%
DecisionTree	5 [min]	7	-0.0055	31.89%
RandomForest	5 [min]	7	-0.0075	38.69%

Rendimiento de los modelos con el planteamiento de clasificación multiclase a cuatro clases y una multiseñal como entrada. PNL dado en dólares diarios.

Resultados

Los resultados mostrados a continuación corresponden al diseño de construcción de los datos en el cual se presentó un problema de clasificación multiclase con cuatro etiquetas. Debido que al generar las cuatro clases se presentaba un problema de desbalance de clases lo que aumentaba el sobreajuste de los modelos, se propuso un pesado de las clases como lo muestra la siguiente figura:



Resultados

Clasificador	Muestreo	Tamaño de ventana	PNL	Precisión
GaussianNB	5 [min]	7	-0.0023	26.15%
Kneighbors	5 [min]	7	-0.0087	37.51%
DecisionTree	5 [min]	7	-0.0079	39.25%
RandomForest	5 [min]	7	-0.0068	32.66%

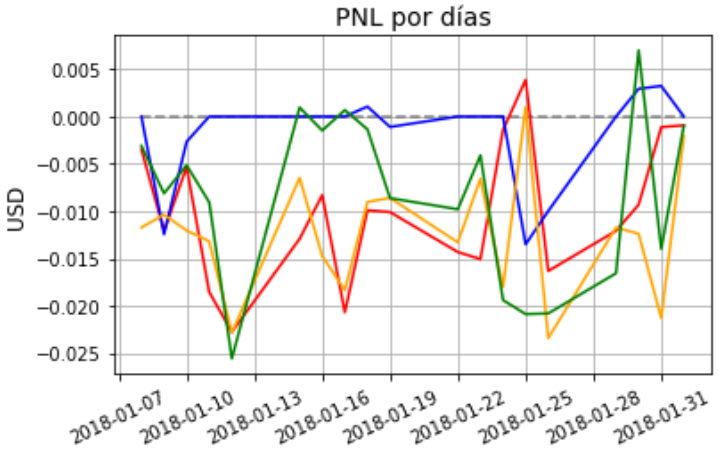
Rendimiento de los modelos con el planteamiento de clasificación multiclase a cuatro clases y peso para las clases. PNL dado en dólares diarios.

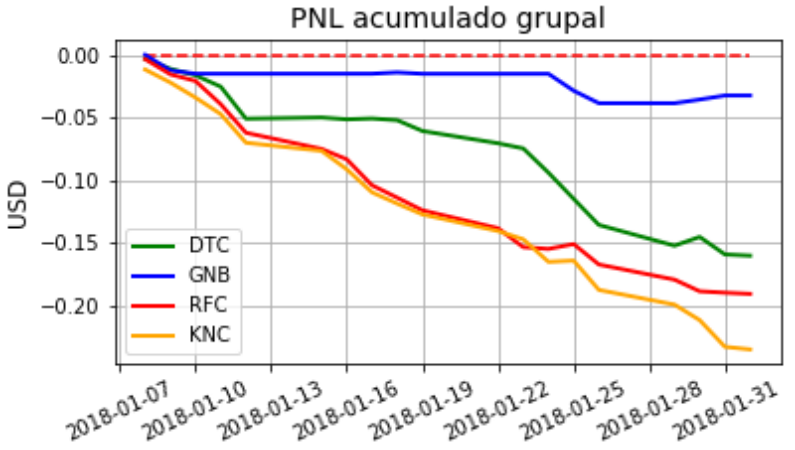
Clasificador	Muestreo	Tamaño de ventana	PNL	Precisión
GaussianNB	5 [min]	7	-0.0029	24.52%
Kneighbors	5 [min]	7	-0.0079	39.13%
DecisionTree	5 [min]	7	-0.0072	39.88%
RandomForest	5 [min]	7	-0.0072	32.56%

Rendimiento de los modelos con el planteamiento de clasificación multiclase a cuatro clases con multiseñal como entrada y peso para las clases. PNL dado en dólares diarios.

Resultados

Se prueba añadiendo a la señal la característica del *OHLC,* siendo este el mejor resultado obtenido por los modelos de *machine learning.*

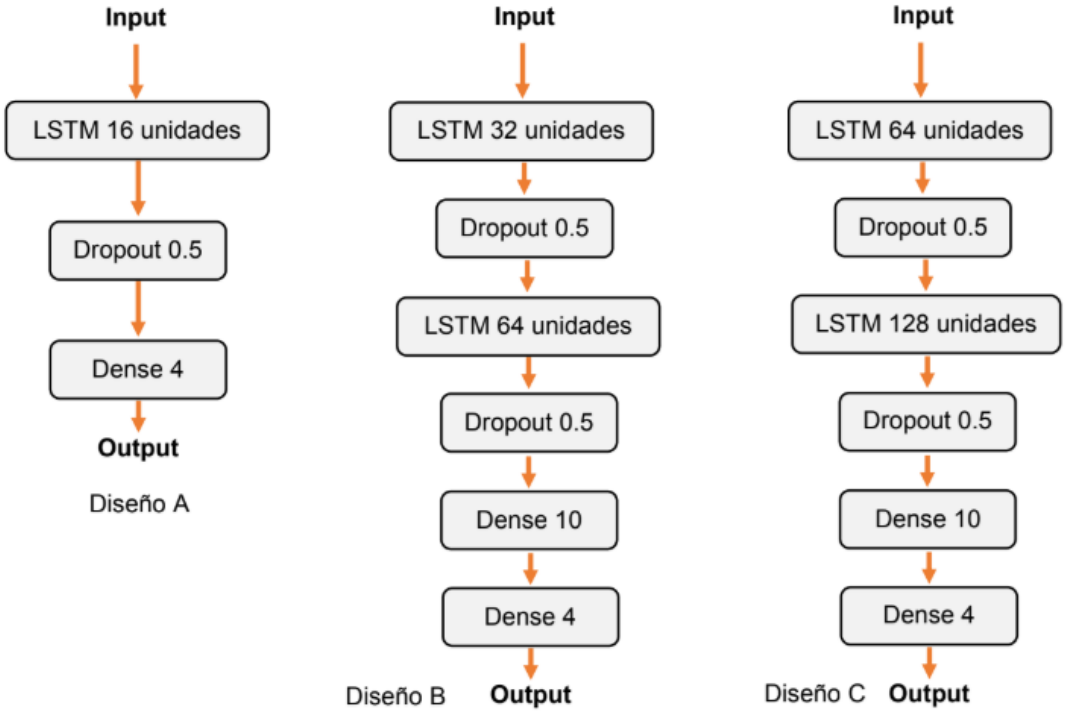




Clasificador	Muestreo	Tamaño de ventana	PNL	Precisión
GaussianNB	5 [min]	7	-0.0013	26.12%
Kneighbors	5 [min]	7	-0.0105	34.47%
DecisionTree	5 [min]	7	-0.0072	34.60%
RandomForest	5 [min]	7	-0.0073	33.65%

Resultados

Se plantean 3 tipos de arquitecturas de RNN’s con unidades LSTM para medir sus desempeños.



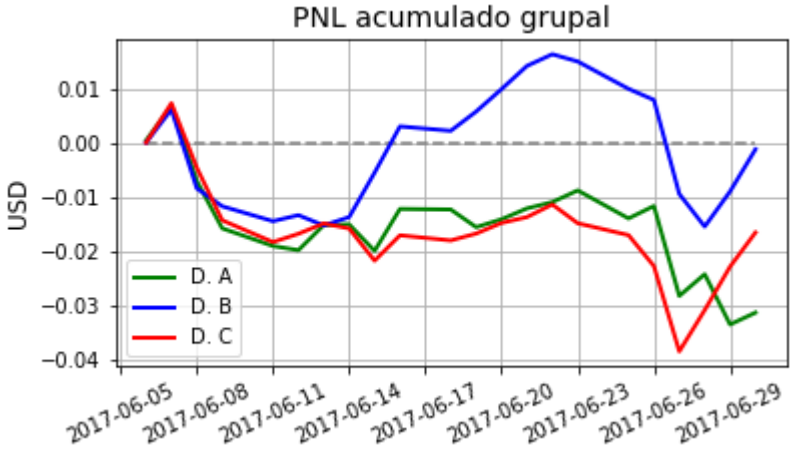
Resultados

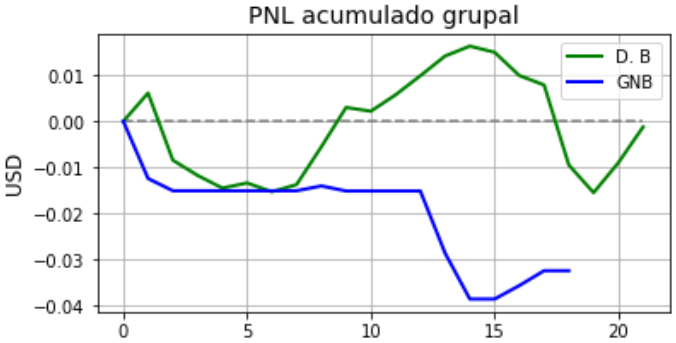
Como era de esperarse, los resultados obtenidos por los 3 tipos de arquitecturas planteadas se comportan mejor que los obtenidos por los métodos de *machine learning,* esto debido a la robustez de dichos modelos.

Diseño	Muestreo	Tamaño de ventana	PNL	Precisión
A	5 [min]	7	-0.0011	29.41%
B	5 [min]	7	-0.000055	28.75%
C	5 [min]	7	-0.00075	27.63%

Rendimiento de los diseños de RNN's con el planteamiento de clasificación multiclase a cuatro clases con peso para las clases y el *OHLC* para la ventana. PNL dado en dólares diarios.

Resultados





Se comparan los dos mejores resultados de los modelos planteados con *machine learning* y *deep learning*

Resultados

Se comparan los dos modelos en base a su PNL y su precisión tanto clase por clase como por la clasificación general, dejando así a la luz que la arquitectura de red propuesta se comporta mejor.

Modelo	GaussianNB	RNN B
Tamaño de ventana	7	7
Muestreo	5 [min]	5 [min]
PNL	**-0.0013**	**-0.00005566**
% acierto clase 1	19.29%	38.34%
% acierto clase 0	4.87%	13.27%
% acierto clase 3	27.91%	55.39%
% acierto clase 2	53.21%	18.57%
% acierto total	26.12%	28.57%

Conclusiones y Perspectivas

Conclusiones

- Añadir características a la ventana nos permite aumentar la precisión de los modelos haciendo que la pérdida de dinero disminuya.

- Las unidades LSTM se comportan mejor que los modelos de *machine learning*, lo que nos da a entender que estas unidades tratan de aprender un mayor número de características.

- Al aplicar múltiples señales se permite aumentar en promedio el rendimiento de los modelos, corroborando que las señales financieras son interdependientes.

Perspectivas

- Para futuras investigaciones, se deben tener en cuenta las características propias de la estadística financiera para añadirlas a cada ventana generada.

- Se podría construir una RNN la cual cuente con más capas y unidades en ella para ver cómo se comporta. La investigación permite estimar que la precisión aumenta.

- Plantear una estrategia de *trading* que no solo dependa de la clase arrojada por el predictor, sino también del porcentaje de precisión con el cual el modelo optó por esa clase.

¡Gracias por la atención prestada!

¿Preguntas?