

Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Evaluación de Métodos de Aprendizaje Automático aplicados al Trading de Forex

Trabajo Fin de Máster

Presentado por: Apolo Viteri, Paúl Andrés

Director/a: Fernández García, Antonio Jesús

Ciudad: Quito

Fecha: 23/02/2022

Resumen

El mercado Forex es uno de los más grandes a nivel mundial. Este mercado es el más líquido del mundo debido a que comercia aproximadamente 6,6 billones de dólares estadounidenses diarios. Los agentes de trading son atraídos a este tipo de mercados financieros, donde tienen como objetivo aplicar estrategias de compraventa que les generen una rentabilidad sostenida. La hipótesis de mercados eficientes postula que no se pueden pronosticar los valores de los mercados financieros de forma consistente a un largo plazo. En este sentido, se realiza la comparación de diferentes metodologías de aprendizaje automático para generar evidencia empírica en contra de esta hipótesis. Los resultados de los experimentos sugieren que se puede obtener un retorno acumulado significativo mediante la aplicación de algoritmos de aprendizaje automático.

Palabras Clave: mercado Forex – estrategias compraventa – aprendizaje automático

Abstract

The foreign exchange market (Forex) is one of the largest in the world. This market is the most liquid in the world, trading approximately 6.6 trillion US dollars daily. Trading agents are attracted to this type of financial markets, where they aim to apply trading strategies that generate sustained profitability. The efficient markets hypothesis says that financial market values cannot be forecast consistently over the long term. In this sense, the comparison of different machine learning methodologies is carried out to generate empirical evidence against this hypothesis. The results of the experiments suggest that a significant cumulative return can be obtained by applying machine learning algorithms.

Keywords: Forex – trading – machine learning

Índice de contenidos

1. Introducción.....	1
1.1 Motivación	1
1.2 Planteamiento del trabajo	3
1.3 Estructura de la memoria	4
2. Contexto y estado del arte.....	5
3. Objetivos y metodología de trabajo	11
3.1. Objetivo general.....	11
3.2. Objetivos específicos	11
3.3. Metodología del trabajo	11
4. Planteamiento de la comparativa	13
4.1 Configuración de los experimentos	13
4.2 Conjunto de datos y selección de variables	15
4.2.1 Preprocesamiento	22
4.2.2 Características generales.....	23
4.3 Métricas de Rendimiento	29
4.4 Clasificadores de Aprendizaje Automático	31
Regresión Logística	32
Árboles de Decisión	32
Bosques Aleatorios	33
Máquinas de Vectores de Soporte	34
Clasificador Gaussian Process.....	34
Clasificador Ridge	35
Clasificador Extra Trees	35
Clasificador Gradient Boosting	36
AdaBoost	36
CatBoost	37

XGBoost.....	37
Red Neuronal con arquitectura Perceptrón Multicapa	38
4.5 Metodologías de los experimentos.....	39
5. Desarrollo de la comparativa.....	41
5.1 Par de Divisa EUR/USD	41
5.2 Par de Divisa USD/JPY	44
5.3 Par de Divisa AUD/JPY	48
6. Discusión y análisis de resultados.....	53
6.1. Limitaciones.....	55
6.2. Comparación con el estado del arte.....	56
7. Conclusiones y trabajo futuro	58
7.1. Conclusiones	58
7.2. Líneas de trabajo futuro	59
8. Bibliografía	60
Anexos.....	64
Anexo I. Módulo de Extracción	64
Anexo II. Módulo de Transformación y Preprocesamiento	64
Anexo III. Módulo de Modelación y Backtesting.....	68
Anexo. Artículo de investigación	74

Índice de tablas

Tabla 1. Resumen del estado del arte	10
Tabla 2. Descripción del conjunto de datos bruto	16
Tabla 3. Variables seleccionadas para los modelos	17
Tabla 4. Características generales de las variables categóricas.....	24
Tabla 5. Estadística descriptiva de las variables numéricas	28
Tabla 6. Resultados de la aplicación de las distintas metodologías para la predicción de EUR/USD.....	41
Tabla 7. Resultados de la aplicación de las distintas metodologías para la predicción de USD/JPY.....	45
Tabla 8. Resultados de la aplicación de las distintas metodologías para la predicción de AUD/JPY.....	49

Índice de figuras

Figura 1. Sistema multi agente de trading	14
Figura 2. Parámetros de entrenamiento de la regresión logística	32
Figura 3. Parámetros de entrenamiento de los árboles de decisión	33
Figura 4. Parámetros de entrenamiento de los bosques aleatorios	34
Figura 5. Parámetros de entrenamiento de la SVM con kernel lineal	34
Figura 6. Parámetros de entrenamiento de la SVM con kernel radial	34
Figura 7. Parámetros de entrenamiento del clasificador por proceso gaussiano	35
Figura 8. Parámetros de entrenamiento del clasificador ridge	35
Figura 9. Parámetros de entrenamiento del clasificador extra trees	36
Figura 10. Parámetros de entrenamiento del clasificador gradient boosting	36
Figura 11. Parámetros de entrenamiento del adaboost	37
Figura 12. Parámetros de entrenamiento del catboost	37
Figura 13. Parámetros de entrenamiento del xgboost	38
Figura 14. Parámetros de entrenamiento de la red neuronal perceptrón multicapa	39
Figura 15. Retorno acumulado de la metodología SVM por cada experimento para EUR/USD	42
Figura 16. Retorno acumulado de la metodología del mejor modelo por cada experimento para EUR/USD.....	43
Figura 17. Retorno acumulado de la metodología del conjunto de modelos por cada experimento para EUR/USD	44
Figura 18. Retorno acumulado de la metodología SVM por cada experimento para USD/JPY	46
Figura 19. Retorno acumulado de la metodología del mejor modelo por cada experimento para USD/JPY.....	47
Figura 20. Retorno acumulado de la metodología del conjunto de modelos por cada experimento para USD/JPY	48
Figura 21. Retorno acumulado de la metodología SVM por cada experimento para AUD/JPY	50

Figura 22. Retorno acumulado de la metodología del mejor modelo por cada experimento para AUD/JPY51

Figura 23. Retorno acumulado de la metodología del conjunto de modelos por cada experimento para AUD/JPY52

1. Introducción

1.1 Motivación

Las principales bolsas de valores del mundo mueven miles de millones de dólares diariamente. Entre las principales podemos encontrar a la bolsa de valores de Nueva York (NYSE), NASDAQ y la bolsa de valores de Tokio (TSE). Por ejemplo, la NYSE registra un movimiento de unos 3 billones de dólares estadounidenses al día solo en préstamos repos¹, según los datos proporcionados por la BBC en septiembre de 2019. A pesar de todo el dinero que transan las bolsas de valores, existe un mercado en donde se mueven aún más dólares diarios. El mercado de divisas (Forex) es el mercado más líquido del mundo y tiene relación con todas las bolsas de valores. Al Forex se le conoce como “el mercado que nunca duerme” y comercia aproximadamente 6,6 billones de dólares estadounidenses diarios.

El trading (i.e., la compraventa de activos, acciones, divisas, futuros y derivados) tiene lugar en estos grandes mercados financieros como lo son las bolsas de valores y el mercado de divisas, debido a que tienen una muy alta liquidez y cada operación se puede realizar casi instantáneamente. En general, las grandes instituciones financieras (e.g., bancos, fondos de inversión, entre otros) concentran la mayor cantidad de dinero que se mueve diariamente en los mercados financieros; sin embargo, también existen individuos que invierten su dinero realizando trading. Las principales técnicas usadas por la mayoría de los agentes que realizan trading son el análisis fundamental y el análisis técnico. El análisis fundamental se basa en predecir a largo plazo el comportamiento de las series financieras en función de fundamentos macroeconómicos (e.g., tipo de interés, la inflación, el producto interno bruto, tasa de desempleo, entre otros). Por otro lado, el análisis técnico tiene como finalidad predecir el movimiento de las series financieras a un corto o medio plazo a partir de sus líneas de tendencia, indicadores técnicos o análisis estadísticos.

Los cambios y movimientos que existen en los mercados financieros pueden ser muy difíciles de predecir. La mayoría de las veces el comportamiento de las series financieras no sigue un patrón claro y los cambios en sus valores dependen de muchas variables y factores (e.g., expectativas de los agentes, factores macroeconómicos, inestabilidad política, confianza en el mercado, entre otros). Por tanto, el análisis fundamental y técnico no son técnicas infalibles que aseguran obtener una rentabilidad con tu inversión, es más, a pesar de su uso muchos

¹ Operación de recompra en la que una entidad financiera vende a un inversor un activo con el compromiso de comprarlo a una fecha y precio determinado.

inversionistas no logran obtener una rentabilidad constante y pueden perder su capital al cabo de algunos meses o años. Si bien es cierto que las grandes instituciones financieras pueden generar rentabilidad a través del trading, su rendimiento comúnmente es bajo porque se enfocan más en seguir una estrategia conservadora que minimice su riesgo.

Los agentes que invierten dentro de las bolsas de valores y el mercado de divisas han buscado técnicas cada vez más complejas que les generen una mayor rentabilidad. Existe mucho interés gracias a las enormes cantidades de dinero que se mueven en estos mercados. Si se encuentra algún algoritmo o técnica que pueda predecir consistentemente los precios de las series financieras, a pesar de su volatilidad, sería un escenario ideal y casi utópico para cualquier inversionista. La hipótesis de mercados eficientes asegura que, en cualquier momento dado, el precio de un instrumento siempre refleja completamente toda la información disponible, la cual sería imposible de conseguir. La industria financiera aún debate la idea de aceptar que no existe forma posible de “ganarle al mercado”, es decir, que los precios de los instrumentos financieros no son predecibles (J. Li & Tsang, 1999). En otras palabras, la idea de que es imposible obtener un rendimiento constante más alto que el crecimiento de un índice (e.g., Dow Jones, S&P 500, Nasdaq) con una simple estrategia de compraventa aún es controversial (Becket, 2004).

La inteligencia artificial es un campo que está siendo ampliamente utilizado por investigadores públicos y privados para hacer más eficiente el uso de recursos, optimizar procesos, innovar y generar mayores beneficios económicos. Dentro de la inteligencia artificial, se encuentran los campos de minería de datos y aprendizaje automático, los cuales han demostrado ser capaces de resolver con gran precisión problemas de clasificación y regresión de la vida real en muchas áreas diferentes como comunicaciones (Di & Joo, 2007), análisis del tráfico en la internet (Nguyen & Armitage, 2008), imagen médica (Wernick et al., 2010), astronomía (Freed & Lee, 2013), análisis de series temporales (Qi & Zhang, 2008), entre otras. Actualmente, el trading en la industria financiera también está empezando a usar técnicas de aprendizaje automático para el análisis cuantitativo y la predicción del comportamiento de series temporales (Gerlein et al., 2016); sin embargo, a comparación de las demás áreas anteriormente mencionadas, la alta volatilidad e injerencia de muchos factores en este campo hacen más difícil tener resultados precisos.

La aplicación de las técnicas de aprendizaje automático en la predicción financiera presenta una serie de complicaciones estructurales que merman su rendimiento. Por lo general, el entrenamiento de los modelos predictivos en esta área utiliza indicadores técnicos financieros en lugar de un vector bruto de precios. Maggini, Giles & Horne (1997) señalan que existe una inherente dificultad estadística en generar indicadores técnicos confiables, esto se debe a que

las reglas inferidas para producir predicciones precisas cambian continuamente en las series temporales financieras. Además, existen muchos casos contradictorios en los conjuntos de entrenamiento, donde al aplicar una serie de reglas en un momento dado tiene buenos resultados, en otro momento puede producir malos resultados. Esta situación se refleja en varias investigaciones que han reportado rendimientos por debajo del 60% con modelos de aprendizaje automático, los cuales han mostrado impresionantes resultados en otras áreas que la predicción financiera (Gerlein et al., 2016).

Existen muchas investigaciones que han tratado de encontrar el algoritmo adecuado o una opción viable para la predicción financiera. Por lo general, en lugar de guiarse solamente por el rendimiento del modelo, también consideran la rentabilidad y ganancias que se obtendrían con el mismo. Particularmente, el uso de modelos más complejos como redes neuronales, máquinas de vector de soporte o modelos híbridos (Cai et al., 2012) ha sido ampliamente estudiado y obtenido resultados prometedores. A pesar de esto, gran parte de estos estudios no han salido al público debido a que son propiedad intelectual de empresas privadas. En un mundo en constante cambio por la transformación digital, las personas e instituciones financieras que no incorporen inteligencia artificial en sus operaciones de trading pueden llegar a ser obsoletas en un futuro. Además, las grandes corporaciones tienen mucha ventaja en este sentido, debido a que cuentan con los recursos para ser los pioneros de la transformación en la industria y podrían quedarse con la mayor parte de los beneficios que traerá el uso de la nueva tecnología.

1.2 Planteamiento del trabajo

La revisión de la literatura indica que existen algunas técnicas de aprendizaje automático que tienen un rendimiento aceptable para modelizar el volátil comportamiento de las series financieras. Específicamente, los modelos más complejos como las máquinas de vectores de soporte o las redes neuronales parecen tener un mejor desempeño que otros modelos más simples. Sin embargo, ningún modelo es completamente adecuado para todos los tipos de mercados; además, los mismos deben ser reentrenados frecuentemente para mantener su precisión (Gerlein et al., 2016).

Este trabajo tiene como fin generar evidencia empírica contra la hipótesis de mercados eficientes, la cual postula que es imposible predecir de manera consistente los precios de los instrumentos financieros y, por tanto, ser rentables a un largo plazo a partir del trading. Existen varios estudios que han presentado resultados notables mediante el uso de técnicas de aprendizaje automático en distintos mercados financieros. El mercado Forex, a priori, es el

más difícil de modelizar debido a su alta liquidez y volatilidad. Si se puede predecir el comportamiento de las series financieras dentro de este mercado, se podría esperar lo mismo de los demás mercados financieros. Por tanto, el propósito de este trabajo es identificar y evaluar el desempeño de los mejores modelos y métodos de aprendizaje automático sobre el mercado forex.

Las técnicas y métodos de aprendizaje automático que se van a comparar y evaluar serán elegidas mediante una extensiva revisión del estado del arte. Una vez identificadas las principales técnicas, se procederá a entrenar los modelos sobre un mismo conjunto de datos (el histórico de los pares de divisas más representativos de algunos tipos de mercados Forex, más sus indicadores técnicos como variables explicativas). Este conjunto de datos será dividido en entrenamiento, validación y prueba para evaluar el desempeño de las distintas metodologías con el uso de técnicas de validación cruzada. Además, se conservará una partición de los datos para realizar un backtesting que simule su aplicación en la vida real y el retorno financiero que se obtendría de cada modelo.

1.3 Estructura de la memoria

En la sección 2 de este trabajo, se hará una revisión de las distintas técnicas de aprendizaje automático usadas para predecir el comportamiento de los mercados financieros. En la sección 3, se establecerán los principales objetivos que tiene el estudio. Partiendo de la revisión del estado del arte, en la sección 4, se indicará las distintas metodologías planteadas para modelizar las series financieras y se abordará la forma de comparación. Posteriormente, en la sección 5, se expondrá los principales resultados de cada uno de los métodos de aprendizaje automático. La sección 6 está dedicada al análisis de los resultados obtenidos al comparar las distintas técnicas. Finalmente, la sección 7 expone las conclusiones y posibles extensiones al trabajo.

2. Contexto y estado del arte

El área de trading financiero está compuesta por varias ramas, dentro de las cuales particularmente dos han reportado implementaciones de técnicas de aprendizaje automático con relativo éxito: fijación de precios de derivados y predicción financiera (Gerlein et al., 2016). La literatura es amplia con respecto a los modelos de aprendizaje automático que se han probado para la predicción de series financieras, donde se puede encontrar desde modelos simples como el Naive Bayes hasta modelos mucho más complejos como lo son las redes neuronales artificiales convolucionales (CNN) o recurrentes (RNN). No obstante, la dificultad implícita en la predicción de este tipo de series ha provocado que un gran número de investigaciones (Chen et al., 2006; Eng et al., 2008; Kim, 2003; Lee et al., 2007; S.-T. Li & Kuo, 2008; Tenti, 1996) reporten un bajo rendimiento de sus modelos (con métricas de precisión por debajo del 60%), cuando los mismos modelos tienen rendimientos sobresalientes en otras áreas fuera de la predicción financiera.

Las redes neuronales artificiales son probablemente el método más utilizado en la predicción de series financieras. Investigaciones tempranas como la de Tenti (Tenti, 1996) comparó el rendimiento de tres redes neuronales recurrentes con base en sus retornos, al simular su aplicación en el mercado de futuros de divisas. Las variables de entrada de las redes se componían de indicadores técnicos (índice de movimiento direccional promedio, índice de movimiento de tendencia y tasa de cambio). Tenti también considero los costos del trading, lo cual hace más robusto sus resultados positivos del retorno financiero en la simulación que realizó, demostrando que las técnicas de redes neuronales pueden ser usadas como herramientas para la predicción financiera.

Por otro lado, en el estudio de Maggini et al. (1997) se investigó el uso de un método mucho más simple para pronosticar la variación de precios. Los autores propusieron un método heurístico para seleccionar diferentes variables que sirvan de entrada para algoritmos de aprendizaje automático no lineales. Además, el enfoque fue cambiado de predecir series temporales a clasificar el tipo de variación de precios (sube, baja o estable), lo cual limita y simplifica el análisis. La técnica seleccionada para abordar el problema de clasificación fue la de vecinos próximos (K-nearest neighbours) con un conjunto de datos con una ventana móvil usada para reentrenar el modelo en cada paso de tiempo. Esta investigación señala que es imposible predecir la variación de los precios con suficiente precisión, a pesar de las limitaciones y simplificaciones propuestas. Sin embargo, los autores se enfocaron solo en la precisión del modelo y no proveen el retorno financiero que este podría producir por su aplicación en la realidad.

En un estudio realizado por Li y Tsang (J. Li & Tsang, 1999), se realizó la predicción del retorno esperado del Dow Jones Industrial Average² (DJIA) usando programación genética financiera. Los autores compararon sus resultados con decisiones aleatorias y un árbol de decisión C4.5. Las variables que usaron fueron simples indicadores técnicos como lo son los movimientos promedio a corto y largo plazo y los filtros de precio a corto y largo plazo. El trabajo no solo se enfocó en la precisión de las predicciones, sino también en los retornos esperados de un conjunto de inversiones que sigan los pronósticos de los modelos. El modelo de programación genética obtuvo un retorno positivo por encima del 60% y un retorno anual por encima del 40% sobre un periodo de trading de cuatro años. Así mismo, el modelo del árbol de decisión C4.5 obtuvo un retorno anual por encima del 40% para los cuatro años. En ambos casos, los resultados presentan un asombroso retorno financiero aún sin considerar los costos del trading, lo cual sugiere que los indicadores técnicos podrían generar modelos rentables basados en reglas para predecir complejas series temporales financieras.

Existen otras técnicas de aprendizaje automático que han mostrado resultados prometedores dentro del área de predicción financiera. Por ejemplo, las máquinas de vector de soporte (SVM) son una de las técnicas que han obtenido mejores rendimientos después de las redes neuronales. Kim (2003) investigó el uso de las SVM para la predicción de la dirección diaria de precios del índice bursátil KOSPI³. El modelo usó indicadores de análisis técnico (momento, Williams %R y el índice de canales de materia prima) como variables de entrada y su mejor precisión reportada, después de entrenar el modelo bajo diferentes parámetros, fue de 57.83%. El trabajo también presentó la comparación con un modelo de redes neuronales entrenadas con *back propagation* (54.76% de precisión) y otro modelo de vecinos más próximos (51.98% de precisión). Estos niveles medios de precisión son esperados debido a la alta volatilidad de las series temporales financieras. Este estudio concluyó que ningún modelo se ajusta perfectamente a todas las condiciones del mercado y, aún más importante, que los modelos necesitan reentrenarse frecuentemente para mantener sus pronósticos precisos.

Kamruzzaman y Sarker (2003) compararon el rendimiento de modelos estadísticos más clásicos (ARIMA) contra algunos modelos de redes neuronales artificiales para la predicción los tipos de cambio de pares de divisas en el mercado Forex. Las redes neuronales fueron entrenadas con *back propagation* en conjunto de regularización bayesiana. Las métricas de error usadas para evaluar el rendimiento global de los modelos mostraron que, todos los

² Índice que mide el desempeño de las 30 mayores sociedades anónimas que cotizan en el mercado bursátil de Estados Unidos.

³ KOSPI es el índice bursátil de Corea del Sur, el cual está conformado por todas las compañías negociadas en el mercado de valores de Corea "Korea Stock Exchange".

modelos de redes neuronales superaron al modelo ARIMA con una reducción del error en 80%. Cabe recalcar que los autores solo reportaron los mejores resultados que obtuvieron y que, en general, las técnicas de aprendizaje automático no presentan altos niveles de precisión con data que no ha sido proporcionada o difiere significativamente del conjunto de entrenamiento; por tanto, estos impresionantes resultados deben ser tomados con precaución.

La investigación de Barbosa y Belo se dividió en varios trabajos en los cuales utilizaron sistemas de agentes únicos e inteligentes para ejecutar operaciones algorítmicas en el mercado Forex (Barbosa & Belo, 2008b), manejar fondos de inversiones (Barbosa & Belo, 2010) y realizar trading en múltiples mercados (Barbosa & Belo, 2008a), enfocándose en la rentabilidad y la reducción máxima de la inversión como métricas de rendimiento. La arquitectura propuesta se dividió en tres módulos a cargo de (a) predecir inmediatamente la siguiente tendencia usando una combinación de modelos de clasificación binaria, (b) un módulo de manejo del riesgo para decidir cuánto invertir en cada transacción por un motor basado en casos que analiza operaciones pasadas, y (c) un sistema basado en reglas. Este sistema de reglas se construye a partir de la experiencia humana, su función es dirigir las decisiones de trading y añadir límites, órdenes de parar las pérdidas, y políticas de abrir y cerrar una transacción. El sistema realiza el aprendizaje por medio del promedio de la combinación ponderada de los resultados de simples modelos de aprendizaje automático, como lo son el OneR, C4.5, JRip, Logistic Model Tree, KStar, NN, SVM y Naive Bayes. Además, los clasificadores vuelven a ser reentrenados en periodos fijos para adaptarlos a las nuevas condiciones del mercado. El módulo de clasificación tiene como resultado una precisión baja del 52.47%; sin embargo, el sistema completo produce una impresionante tasa de 66.67% de rentabilidad en el periodo de pruebas, realizando menos pero más rentables operaciones y evitando transacciones que no se esperaban rentables gracias a la combinación de los diferentes módulos, con un alto nivel de automatización.

De acuerdo con Sewell y Yan (Sewell & Yan, 2008), para ciertos mercados como los de futuros y Forex, se podría llegar a necesitar predicciones con una precisión marginalmente mayor que las obtenidas por clasificadores aleatorios para obtener rentabilidad debido a dos principales factores: bajos costos y apalancamiento financiero. Una forma de optimizar el rendimiento de los modelos de aprendizaje automático podría ser optar por una combinarlos mediante técnicas de *ensemble models*⁴. De la misma forma que Barbosa y Belo, McDonald et al. (2014) investigan la efectividad de algunos modelos de aprendizaje automático y sus combinaciones

⁴ Métodos de aprendizaje automático que utilizan un conjunto de algoritmos o modelos para lograr una mejor predicción que la que se podría obtener de cualquiera de los algoritmos de aprendizaje constituyentes.

para generar pronósticos de series temporales financieras. Los autores encontraron que los modelos híbridos, que consisten en modelos lineales estadísticos y algoritmos de aprendizaje automático no lineales, son efectivos en la predicción de la dirección de futuros valores de las series financieras.

Otras investigaciones han actualizado los resultados obtenidos por los modelos anteriormente mencionados en esta sección, sin obtener significativamente unas mejores métricas de precisión o retorno financiero. Chen et al. (2006) compararon el modelo de SVM con las NN para predecir el valor de seis índices asiáticos. Los autores evaluaron la dirección del pronóstico obteniendo un nivel de precisión promedio de 57.2% con el SVM y 56.7% con el modelo de NN. De la misma forma, Lu y Wu (2009) evaluaron el rendimiento de un modelo de NN frente a un modelo ARIMA para el pronóstico de la dirección de los valores futuros del índice S&P 500. Este estudio mostró que el sistema basado en NN obtuvo mejores resultados que el modelo ARIMA solo en condiciones estables de mercado, dado que el mismo sistema tuvo una precisión del 23% frente al 42% del ARIMA en escenarios más volátiles.

Gerlein et al. (2016) contribuyó con el estado del arte investigando el uso de técnicas más simples de aprendizaje automático. Los autores se basaron en el estudio realizado por Barbosa y Belo (Barbosa & Belo, 2008a), donde demostraron que la combinación de varios modelos simples de aprendizaje automático puede ser usada para pronosticar series temporales financieras. El trabajo evaluó la mayoría de los modelos usados por Barbosa y Belo, quitando unos y añadiendo otros, de manera individual midiendo su precisión y retorno financiero en el mercado de divisas Forex. Los mejores resultados que obtuvieron los modelos de todos los escenarios que plantearon fueron los siguientes: OneR obtuvo una precisión del 58.9% y un retorno acumulado del 32%, C4.5 tuvo una precisión del 64.08% y un retorno acumulado del 57.17%, Jrip presentó una precisión de 63.75% y un retorno acumulado de 95.94%, LMT presentó una precisión de 64.37% y un retorno acumulado de 39.46%, K-star obtuvo una precisión de 56.88% y un retorno acumulado de 73.16%, y finalmente Naive Bayes tuvo una precisión del 63.69% y un retorno acumulado de 22.62%. El retorno acumulado fue el rendimiento financiero que se obtuvo simulando el trading durante todo el periodo del experimento (seis años).

Los estudios más recientes se han enfocado en el uso de modelos de aprendizaje profundo (Deep Learning) mediante la implementación de redes neuronales con arquitecturas mucho más complejas que las de sus antecesoras. Carapuço et al. (2018) proponen un modelo de aprendizaje por refuerzo que usen agentes de redes neuronales con tres capas ocultas de neuronas ReLU bajo el algoritmo de Q-learning. Los autores se basan en el retorno financiero para evaluar el rendimiento del modelo propuesto sobre el mercado Forex. Los mejores

resultados que reportaron son un retorno anual de 16.3% y un retorno acumulado de 114% por la duración del experimento (10 años). Finalmente, Ni et al. (2019) proponen un método de aprendizaje profundo C-RNN para la predicción de series temporales de Forex con base en las redes neuronales recurrentes profundas (RNN) y las redes neuronales convolucionales profundas (CNN). El método propuesto fue comparado con los dos en los que se basa; es decir, se evalúa los resultados de la C-RNN frente a los de la RNN y CNN. La métrica con la cual realizan la comparación es la RMSE. El estudio demuestra que el modelo C-RNN propuesto tiene un menor error que los RNN y CNN; sin embargo, no reportaron cuál sería el retorno financiero bajo una simulación de la aplicación de los modelos.

En la mayoría de los trabajos revisados existe evidencia a favor del uso de modelos de aprendizaje automático para realizar pronósticos de series temporales financieras. En general, los artículos reportaron resultados positivos que podrían ser vistos como evidencia en contra de la hipótesis de mercados eficientes. Si bien es cierto que las métricas de rendimiento del modelo (precisión, RMSE, entre otras) indican resultados pobres comparando con otras áreas, al evaluar también la rentabilidad y el retorno financiero se observa que la mayoría de los modelos pueden servir como una herramienta útil para realizar trading en los mercados financieros.

La Tabla 1 presenta un resumen de los trabajos mencionados en esta sección. Los aspectos que se resaltan de los diferentes estudios y modelos son los siguientes:

- a) Tipo de modelo de aprendizaje automático
- b) Problema de regresión o clasificación
- c) Mercado en el que se aplica
- d) Métrica de evaluación del rendimiento del modelo
- e) Rendimiento/Error del modelo
- f) Retorno financiero
- g) Periodo del retorno financiero

El principal enfoque del presente trabajo es comparar el rendimiento y retorno de diferentes algoritmos y métodos de aprendizaje automático para pronosticar el comportamiento de los pares de divisas del mercado Forex. Esto para generar evidencia contra la hipótesis de mercados eficientes. Por tanto, con base en esta información, se procederá a escoger a los mejores modelos para plantear distintos métodos de predicción, los cuales serán evaluados y comparados.

Trabajo de Investigación	Modelo	Problema	Mercado	Métrica	Rendimiento/ Error ⁵	Periodo	Retorno
Tenti (1996)	RNN	Regresión	Futuros	NMSE	0.9795	100 días	27.7%
Maggini et al. (1997)	KNN	Clasificación	N/A	N/A	N/A	N/A	N/A
J. Li y Tsang (1999)	C4.5	Clasificación	DJIA	Precisión	55.11%	1 año	47.46%
Kim (2003)	SVM	Clasificación	KOSPI	Precisión	57.83%	N/A	N/A
Kim (2003)	NN	Clasificación	KOSPI	Precisión	54.73%	N/A	N/A
Kamruzzaman y Sarker (2003)	NN	Regresión	Forex	NMSE	0.27	N/A	N/A
Kamruzzaman y Sarker (2003)	ARIMA	Regresión	Forex	NMSE	1.03	N/A	N/A
Chen et al. (2006)	SVM	Regresión	Valores	DS ⁶	57.23%	N/A	N/A
Chen et al. (2006)	NN	Regresión	Valores	DS	56.73%	N/A	N/A
Barbosa y Belo (2008a)	Combinación	Clasificación	Forex	Precisión	52.47%	2 años	66.67%
Gerlein et al. (2016)	OneR	Clasificación	Forex	Precisión	58.90%	6 años	32.00%
Gerlein et al. (2016)	C4.5	Clasificación	Forex	Precisión	64.08%	6 años	57.17%
Gerlein et al. (2016)	Jrip	Clasificación	Forex	Precisión	63.75%	6 años	95.94%
Gerlein et al. (2016)	LMT	Clasificación	Forex	Precisión	64.37%	6 años	39.46%
Gerlein et al. (2016)	Kstar	Clasificación	Forex	Precisión	56.88%	6 años	73.16%
Gerlein et al. (2016)	Naive Bayes	Clasificación	Forex	Precisión	63.69%	6 años	22.63%
Carapuço et al. (2018)	Q-learning	Regresión	Forex	N/A	N/A	1 año	16.3%
Ni et al. (2019)	C-RNN	Regresión	Forex	RMSE	512	N/A	N/A
Ni et al. (2019)	LSTM	Regresión	Forex	RMSE	543	N/A	N/A
Ni et al. (2019)	CNN	Regresión	Forex	RMSE	573	N/A	N/A

Tabla 1. Resumen del estado del arte

⁵ Las métricas de rendimiento o error de los distintos modelos no son directamente comparables dado que cada uno de ellos se realiza sobre diferentes mercados en diferentes lapsos de tiempo.

⁶ Chen et al. (2006) implementaron esta métrica para evaluar la precisión con la que los modelos pronosticaban correctamente la dirección (sube/baja) de la serie temporal.

3. Objetivos y metodología de trabajo

3.1. Objetivo general

El objetivo de este estudio es comparar distintos métodos de aprendizaje automático, con base en los que se detallan en el estado del arte, aplicados al pronóstico de series temporales financieras. De tal forma, el trabajo pretende generar evidencia empírica en contra de la hipótesis de mercados eficientes prediciendo el comportamiento de pares de divisas en el mercado Forex. La predicción debe tener una precisión al menos tan buena como se detalla en la revisión de la literatura. Así mismo, la simulación de la aplicación de los modelos debe dar como resultado una rentabilidad mayor a la que se podría obtener por invertir ese dinero en el sistema financiero convencional.

3.2. Objetivos específicos

- Implementar las distintas técnicas y métodos de aprendizaje automático para predecir el comportamiento de pares de divisas en el mercado Forex.
- Comparar la precisión y rentabilidad de los distintos métodos planteados.
- Determinar la capacidad de las diferentes técnicas de generar predicciones que permitan obtener un retorno financiero mayor al 18% anual.
- Generar evidencia empírica en contra de la hipótesis de mercados eficientes.

3.3. Metodología del trabajo

Para alcanzar los objetivos específicos se deben de realizar una serie de pasos, los cuales se detallan a continuación:

1. Determinar distintos métodos de aprendizaje automático que tengan el potencial de realizar predicciones precisas de los mercados financieros, según el estado del arte.
2. Seleccionar tres pares de divisas representativos de algún tipo de mercado Forex (e.g., mercados estables, exóticos o volátiles) y el periodo de tiempo durante el cual se realizará el experimento.
3. Entrenar y validar los distintos métodos seleccionados de aprendizaje automático.
 - a. Dividir al conjunto de datos en entrenamiento, validación y prueba.

- b. Realizar el entrenamiento de los diferentes modelos de aprendizaje automático usando técnicas de validación cruzada.
 - c. Implementar la metodología seleccionada para generar el modelo final que realice la predicción.
4. Simular la aplicación de los modelos durante el periodo de tiempo de prueba.
 - a. Reentrenar los modelos durante ventanas de tiempo simulando como se lo aplicaría en la realidad.
5. Comparar las métricas de rendimiento de los distintos métodos y su retorno financiero sobre el conjunto de datos de prueba.
6. Demostrar la capacidad de los modelos para generar pronósticos rentables.

4. Planteamiento de la comparativa

El estado del arte señala que se han evaluado varias técnicas de aprendizaje automático para la predicción de las series temporales financieras. Así mismo, los autores han probado distintos enfoques para abordar el problema de aprendizaje automático supervisado (regresión o clasificación) en este campo. Si bien existen distintos modelos o enfoques empleados, las métricas de evaluación muestran que, en general, la aplicación de las técnicas de aprendizaje automático en los mercados financieros tiene un rendimiento más bajo a comparación de su aplicación en otras áreas. A pesar de esto, muchos autores también se han enfocado en comprobar la capacidad de estas técnicas para generar una rentabilidad financiera sostenida y significativa en el tiempo, a pesar de su bajo rendimiento.

El bajo poder predictivo de los modelos es el principal argumento a favor de que no se puede predecir el comportamiento de las series temporales financieras, haciendo elusión a la hipótesis de mercados eficientes. Por tanto, para el desarrollo del trabajo en cuestión, se va a comparar tanto las métricas de rendimiento como el impacto financiero de las distintas técnicas de aprendizaje automático. Esto con el motivo de establecer que no se necesitan métricas de evaluación tan precisas, como en otras áreas, para obtener resultados financieros positivos. Esta sería nuestra principal prueba en contra de la hipótesis de mercados eficientes.

4.1 Configuración de los experimentos

El enfoque que adoptan la mayoría de los investigadores (Barbosa & Belo, 2008a; Gerlein et al., 2016; Kim, 2003; J. Li & Tsang, 1999; Maggini et al., 1997) es simplificar el problema de predicción de series temporales a uno de clasificación. Existe esta facilidad porque solo hay dos tipos de acciones que se puede realizar en el trading: tener una posición en corto (venta) o en largo (compra). El problema puede enfocarse a solamente predecir si en el siguiente periodo el precio sube o baja, de tal forma que se podría adoptar la estrategia de compraventa adecuada en función de esta predicción. En el presente trabajo se utiliza el mismo criterio; por tanto, se trabaja con modelos de aprendizaje automático supervisado para clasificación.

El experimento se realiza sobre un marco de tiempo de seis horas de los precios al cierre de los pares de divisas. Barbosa y Belo (2008a) sugieren realizar el experimento bajo este periodo temporal gracias a que te da la oportunidad de abrir o cerrar las transacciones cuatro veces por día, esperando que los movimientos durante este intervalo sean mayores a los que puedan ocurrir en ventanas de tiempo menores como de tres o una hora. Si se tiene un mayor movimiento de precios, de igual forma las ganancias podrán ser mayores.

Los distintos modelos de aprendizaje automático son evaluados y comparados sobre diferentes escenarios, donde se cambian parámetros como la ventana de tiempo en la que son entrenados los modelos, cada cuánto se reentrenan los modelos y el número de 'folds' para evaluar los modelos. Además, el experimento se realizará sobre tres diferentes pares de divisas, cada par representarán un tipo de mercado diferente (e.g., el par de divisa AUD/JPY representa a los mercados "exóticos" donde existe una mayor volatilidad). Las principales métricas de evaluación serán la precisión del modelo y la rentabilidad financiera obtenida. Para obtener el rendimiento financiero de los distintos modelos es necesario simular su aplicación como si se lo hiciera en el mundo real; por tanto, se construirá un flujo que realice un backtesting de la aplicación de las técnicas de aprendizaje automático sobre el trading en el mercado Forex.

Autores como Barbosa y Belo (2008a) y Gerlein et al. (2016) proponen un sistema multiagente para realizar la simulación del trading a partir de los modelos entrenados. Este sistema de trading se compone de dos agentes: el primero tiene la función de realizar el entrenamiento, validación y predicción (agente de trading); y el segundo realiza la acción en el mercado, en función del pronóstico del primer agente, y extrae más información para seguir alimentando a los modelos (agente de mercado). En la Figura 1 se puede apreciar como estaría conformado este sistema multi agente y como se establecerían sus diferentes relaciones.

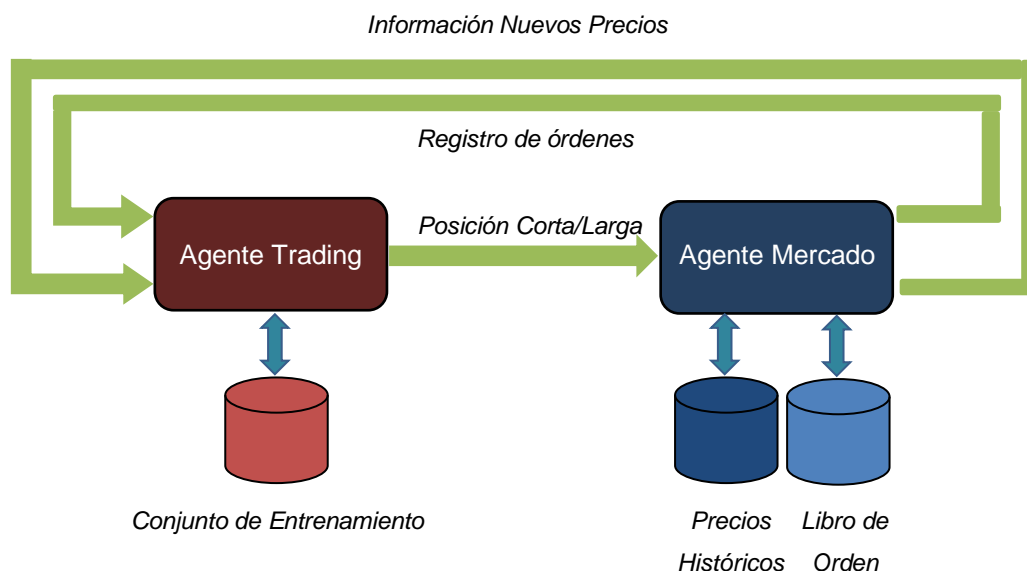


Figura 1. Sistema multi agente de trading

Este sistema multiagente puede ser replicado mediante el entrenamiento y despliegue (simulado) de un modelo de aprendizaje automático durante un número n de iteraciones

sobre una serie temporal de los precios de un par de divisas. Es así como, el flujo que se construye para simular el backtesting será el siguiente:

1. Conformar un conjunto de datos para predecir el precio de un par de divisas.
2. Establecer un periodo sobre el cual se desee realizar el backtesting.
3. Entrenar el modelo de aprendizaje automático sobre un subconjunto de datos del dataset principal que este antes del periodo de backtesting.
4. Predecir los siguientes k valores que quedaron fuera del conjunto de entrenamiento, donde se establece que k es un número natural menor o igual a 15.
5. Mover el rango del subconjunto de datos k posiciones.
6. Repetir el entrenamiento y predicción de los k valores posteriores.
7. Realizar esto un número de n veces hasta completar el periodo de backtesting.

La simulación está programada e implementada en Python, de igual forma que la totalidad del trabajo. Una vez que se haya completado el backtesting será posible calcular cuánta sería la rentabilidad de seguir las predicciones de los diferentes métodos de aprendizaje automático sobre los diferentes pares de divisas. Los detalles de cada parte que conforma el experimento, el conjunto de datos, los diferentes escenarios, las diferentes metodologías usadas, entre otros, serán detallados en las subsecciones posteriores.

4.2 Conjunto de datos y selección de variables

En el presente estudio se emplean las series temporales de los precios de los pares de divisas del mercado Forex extraídas desde el bróker Forex Capital Markets (FXCM). Los pares de divisas seleccionados fueron los siguientes: euro – dólar estadounidense (EUR/USD), dólar estadounidense – yen (USD/JPY) y dólar australiano - yen (AUD/JPY). El periodo de tiempo considerado va desde el primero de enero de 2017 hasta el 31 de diciembre de 2020; sin embargo, el periodo donde se van a medir y evaluar las métricas, y realizar la simulación será desde el primero de enero de 2019 hasta el 31 de diciembre de 2020. Por cada día, los precios extraídos tienen una periodicidad de seis horas; es decir, existen 4 registros de precios por día.

FXCM es uno de los más importantes brokers online de Forex y CFD. Su sede se encuentra en Londres. Esta institución permite a las personas especular y realizar acciones de compraventa en los mercados de divisas o de capitales. Además, FXCM también provee la capacidad para negociar contratos por diferencia (CFDs) sobre los principales índices y

materias prima como el oro y el petróleo crudo. El bróker cuenta en sus bases de datos de información histórica de precios de pares de divisas, índices, derivados e instrumentos financieros. Los datos pueden ser accedidos mediante el uso de la API REST que ofrece gratuitamente⁷ FXCM. La interacción con la API mediante Python es bastante simple debido a que ponen a disposición el paquete “fxcmpy”, el cual permite extraer la información que se necesita mediante programación de alto nivel. El código completo para la extracción de los vectores brutos de precios de los pares de divisas se puede encontrar en el Anexo I. Módulo de Extracción.

Cada uno de los pares de divisas representan a mercados diferentes. El par de divisa euro y dólar estadounidense es el más transado del mundo y tiene como principal característica la liquidez. Por otro lado, el par de divisa dólar estadounidense y yen representa a los mercados menos volátiles, debido a la fortaleza de los bancos centrales estadounidenses y japoneses. Finalmente, el par de divisa dólar australiano y yen es una medida de los mercados más extravagantes y volátiles.

La extracción de los datos de cada par de divisas a partir de la API REST de FXCM retorna un dataframe con los precios de demanda (bid), precios de oferta (ask) y el volumen de transacciones por cada periodo. La información disponible de los precios bid y ask son sus precios de apertura, cierre, máximo y mínimo. Por lo general, la mayoría de los indicadores técnicos utilizan la información de los precios bid o ask para su construcción. La alta liquidez del mercado Forex hace que la diferencia entre los precios bid o los precios ask sea mínima; por tanto, se puede trabajar con cualquiera de los dos para analizar el comportamiento de los precios. En el presente trabajo, los precios ask son los que se tendrán en cuenta para el análisis y se encontrarán en el conjunto de datos final. La estructura de los datos extraídos directamente desde la API por cada par de divisas se detalla en la Tabla 2. Descripción del conjunto de datos bruto

Par de Divisa	Número Instancias	Frecuencia	Periodo	Variables
EUR/USD	4342	H6	02 enero 2017 04h00 - 30 diciembre 22h00	fecha, bid apertura, bid cierre, bid máximo, bid mínimo, ask apertura, ask cierre, ask máximo, ask mínimo
USD/JPY	4337			
AUD/JPY	4341			

Tabla 2. Descripción del conjunto de datos bruto

⁷ A pesar de que el acceso a la API REST sea gratuito, se necesita crear una cuenta demo o real con anterioridad para obtener el token de acceso a los sistemas.

A partir del precio ask de cierre del par de divisas se construye la variable objetivo, la cual indica si en el siguiente periodo el precio sube o baja. Así mismo, los indicadores técnicos usados para la predicción son formados a partir del vector bruto de precios ask de apertura, cierre, máximo, mínimo y volumen. Entonces, el conjunto de datos para cada par de divisa se conforma por la variable objetivo y una serie de indicadores técnicos que, en general, se usan para el análisis técnico en el mercado Forex. Además, también se incluyen variables que describen características generales de las series temporales (e.g., hora o día de la transacción, precio de apertura, precio de cierre, entre otros). En la Tabla 3 se encuentran cuáles fueron las variables seleccionadas para conformar la base de datos por cada par de divisa.

Variable	Naturaleza	Descripción
<i>clase</i>	Categórica	“SUBE” si el precio sube en el siguiente periodo, “BAJA” caso contrario.
<i>bb_bbhi(p)</i>	Binaria	1 si excede la cota superior de las bandas de bollinger, 0 caso contrario.
<i>bb_bbli(p)</i>	Binaria	1 si excede la cota inferior de las bandas de bollinger, 0 caso contrario.
<i>hora</i>	Categórica	Hora del día (04h00, 10h00, 16h00, 22h00).
<i>día</i>	Categórica	Día (lunes, martes, miércoles, jueves o viernes).
<i>apertura</i>	Numérica	Precio ask de apertura del par de divisa.
<i>cierre</i>	Numérica	Precio ask de cierre del par de divisa.
<i>alto</i>	Numérica	Precio ask máximo del par de divisa.
<i>bajo</i>	Numérica	Precio ask mínimo del par de divisa.
<i>volumen</i>	Numérica	Número de transacciones.
<i>roc</i>	Numérica	Cambio porcentual del precio (i) con respecto al precio (i-1).
<i>roc(t-1)</i>	Numérica	El rezago del cambio porcentual del precio.
<i>psar</i>	Numérica	Indicador de sistema parabólico SAR.
<i>bb_bbw(p)</i>	Numérica	Ancho de las bandas de Bollinger.
<i>atr(p)</i>	Numérica	Indicador de rango verdadero promedio.
<i>willr(p)</i>	Numérica	Indicador de Williams %R.
<i>rsi(p)</i>	Numérica	Índice de fuerza relativa.
<i>cierre ma(p)</i>	Numérica	Media móvil de precios al cierre.
<i>roc(t-1) ma(p)</i>	Numérica	Media móvil del rezago del cambio porcentual de precios.
<i>macd(p)</i>	Numérica	Media móvil de convergencia/divergencia.
<i>macd señal(p)</i>	Numérica	Línea de señal de la media móvil de convergencia/divergencia.
<i>stoch(p)</i>	Numérica	Indicador de oscilación estocástica.

Tabla 3. Variables seleccionadas para los modelos

De estas 22 variables, existen 11 variables que están en función del precio de p periodos anteriores. La mayoría de los indicadores técnicos se analizan según una ventana de tiempo, en la cual se consideran un número de valores pasados para realizar el cálculo del indicador. Esto está de acuerdo con la experticia de cada analista y al enfoque que se le quiera dar al problema. Por tanto, las 11 variables en función de los p periodos se incluyen p veces, una vez por cada uno de los siguientes periodos: 5, 10, 15, 20 y 30 periodos. Finalmente, el

conjunto de datos final está compuesto por las 11 variables que no están función de p , más las 5 combinaciones de las 11 variables que están en función de p (55 variables), dando un total de 66 variables. A continuación, se describen los detalles particulares de cada una de las variables seleccionadas para la modelización:

- *clase*: La clase está definida como una variable categórica binaria, la misma es el objetivo de clasificación de los algoritmos de aprendizaje automático. Esta muestra la dirección que el precio va a tomar en el siguiente periodo de compraventa. Las etiquetas para la clase son “SUBE” y “BAJA”, describiendo así si se espera que el precio ask de cierre aumente o disminuya durante la siguiente instancia.
- *bb_bbhi(p)*: Las bandas de Bollinger son bandas de volatilidad colocadas por encima y por debajo de la media móvil. La volatilidad es calculada con base en la desviación estándar, la cuál cambia a medida que aumenta o disminuye la volatilidad. Las bandas pueden ser usadas para identificar puntos en los cuales existe un cambio de tendencia de los precios. Por lo general, esto ocurre cuando el precio se sale del rango de las bandas de Bollinger. En específico, la variable *bb_bbhi* sirve para identificar cuando el precio al cierre ha excedido la cota superior de las bandas de Bollinger. El cálculo de este indicador está definido como:

$$bb_bbhi(p)_i = \begin{cases} 1 & \text{si } cierre_i > \text{banda superior}(p)_i \\ 0 & \text{caso contrario} \end{cases} \quad (1)$$

Donde, la banda superior de Bollinger viene dada por:

$$\text{banda superior}(p)_i = sma(p)_i + (sd(cierre)(p)_i \times 2) \quad (2)$$

$sma(p)_i$ indica la media móvil simple hasta la instancia i y $sd(cierre)(p)_i$ hace referencia a la desviación estándar del precio al cierre desde p periodos antes de i hasta i , dado por las ecuaciones (3) y (4), respectivamente:

$$sma(p)_i = \frac{cierre_{i-p}, cierre_{i-p+1}, \dots, cierre_{i-p+p}}{p} \quad (3)$$

$$sd(cierre)(p)_i = \sqrt{\frac{\sum_j^p (cierre_j - \overline{cierre})^2}{p}}; \quad j \in [i-p, i] \quad (4)$$

- *bb_bbli(p)*: Esta variable indica cuando el precio al cierre ha descendido por debajo de la cota inferior de la banda de bollinger. Su cálculo se lo define como:

$$bb_bbli(p)_i = \begin{cases} 1 & \text{si } cierre_i < \text{banda inferior}(p)_i \\ 0 & \text{caso contrario} \end{cases} \quad (5)$$

Donde, la banda inferior de Bollinger se calcula mediante la ecuación (6):

$$\text{banda superior}(p)_i = \text{sma}(p)_i - (\text{sd}(\text{cierre})(p)_i \times 2) \quad (6)$$

- *hora*: Tiempo del día sobre el cual la instancia i es capturada. La frecuencia con la que los datos fueron extraídos fue de 6 horas; por tanto, las horas posibles son: 4, 10, 16 y 22.
- *día*: Variable categórica que indica el día al que corresponde la instancia i . El mercado de divisas se encuentra abierto las 24 horas solo durante los días laborales. Entonces, esta variable puede tener las siguientes etiquetas: lunes, martes, miércoles, jueves y viernes.
- *apertura*: Atributo numérico que representa el precio ask de apertura registrado durante la instancia i .
- *cierre*: Atributo numérico que representa el precio ask de cierre registrado durante la instancia i .
- *alto*: Atributo numérico que representa el máximo precio ask registrado durante la instancia i .
- *bajo*: Atributo numérico que representa el mínimo precio ask registrado durante la instancia i .
- *volumen*: Número de transacciones de compraventa registradas durante el periodo de trading de la instancia i .
- *roc*: Esta variable numérica describe el cambio relativo del precio ask de cierre durante el periodo actual de trading de la instancia i , dado por la ecuación (7).

$$\text{roc}_{i+1} = ((\text{cierre}_{i+1} - \text{cierre}_i) / \text{cierre}_i) \times 100 \quad (7)$$

- *roc(t-1)*: Atributo numérico que indica el rezago del cambio relativo del precio ask de cierre. En general, los rezagos son incluidos en análisis de series temporales donde se desea ver el efecto de lo sucedido en el pasado sobre el presente. Específicamente, en este caso, este indicador describe el cambio porcentual del precio un periodo atrás de la siguiente forma:

$$\text{roc}(t - 1)_{i+1} = \text{roc}_i \quad (8)$$

- *psar*: La “Parabolic Stop and Reverse” (Parabolic SAR) es un indicador que sigue el precio a medida que la tendencia se extiende con el tiempo. Este indicador se encuentra por debajo del precio cuando ellos están subiendo y viceversa; por lo que, el Parabolic SAR es muy útil para detectar cambios de tendencia o comportamiento. Su cálculo puede ser encontrado con mayor detalle en el estudio realizado por Yazdi & LASHKARI, (2012).
- *bb_bbw(p)*: Este atributo numérico se deriva del cálculo de las bandas de Bollinger especificado anteriormente. El mismo mide el porcentaje de diferencia entre la banda superior y la banda inferior. El indicador tiene como objetivo determinar los momentos de mayor volatilidad y menor volatilidad, que son cuando se aumenta y disminuye la diferencia porcentual, respectivamente. El ancho de las bandas de Bollinger se lo calcula como:

$$bb_bbw(p)_i = (banda\ superior(p)_i - banda\ inferior(p)_i) / sma(p)_i \quad (9)$$

- *atr(p)*: El rango verdadero promedio (ATR) es un indicador que mide la volatilidad. Esta variable sirve para capturar la volatilidad de los movimientos del precio entre alguna brecha o límite. Primero, se debe de calcular el rango verdadero (TR) dado por la ecuación (10). Posteriormente, el cálculo del rango verdadero promedio se realiza mediante la ecuación (11).

$$TR_i = Max[(alto_i - bajo_i), |alto_i - cierre_i|, |bajo_i - cierre_i|] \quad (10)$$

$$atr(p)_i = \left(\frac{1}{p}\right) \sum_j^p TR_j; \quad j \in [i - p, i] \quad (11)$$

- *willr(p)*: El oscilador Williams %R es un indicador técnico cuyos valores oscilan entre 0 y -100. Cuando este indicador está por debajo de -80 o por encima de -20 significa que el instrumento esta sobrevendido o sobrecomprado, respectivamente. Esta variable numérica compara el precio al cierre de la instancia actual i con los precios más bajos y altos en los últimos p periodos, se lo calcula de la siguiente forma:

$$willr(p)_i = ((cierre_{i-1} - alto_p) / (alto_p - bajo_p)) \times 100 \quad (12)$$

- *rsi(p)*: El índice de fuerza relativa también es un indicador técnico que compara la magnitud de las recientes ganancias contra las recientes pérdidas para determinar las condiciones de sobrecompra o sobreventa de un instrumento. El índice de fuerza relativa oscila desde 0 a 100 e indica si un activo se encuentra sobrevaluado o devaluado en alguna instancia i . Si el valor del índice es mayor a 70, el instrumento

podría estar sobrevaluado; mientras que, si el índice es menor a 30, el instrumento posiblemente está infravalorado. El índice de fuerza relativa es calculado como:

$$rsi(p)_i = 100 - 100 \times (1/(1 - RS(p)_i)) \quad (13)$$

$RS(p)_i$ indica la ratio entre el promedio de las p instancias pasadas donde el precio incremento y el promedio de las p instancias pasadas donde el precio decreció, dado por las ecuaciones (14), (15) y (16):

$$RS(p)_i = UP_avg(p)_i / DOWN_avg(p)_i \quad (14)$$

$$UP_avg(p)_i = \left(\sum_k (cierre_k - apertura_k) \right) / p \quad (15)$$

Donde k es un periodo caracterizado por el precio de cierre más alto que el previo precio de cierre, y

$$DOWN_avg(p)_i = \left(\sum_j (apertura_j - cierre_j) \right) / p \quad (16)$$

Donde j es un periodo caracterizado por el precio de apertura más alto que el previo precio de apertura.

- *cierre ma(p)*: La media móvil de los precios al cierre es un promedio móvil aritmético calculado mediante la adición de los precios más recientes y luego dividiendo el resultado por el número de instancias o periodos que se incluyeron. Esta variable numérica sirve como referencia para ver la tendencia al alza o baja de los precios durante la ventana de tiempo analizada. La ecuación (3) se usa para el cálculo del indicador.
- *roc(t-1) ma(p)*: Este atributo numérico es calculado construyendo el promedio de los cambios porcentuales de los precios durante los últimos p periodos, dado por:

$$roc(t-1) ma(p)_i = \frac{roc(t-1)_{i-p}, roc(t-1)_{i-p+1}, \dots, roc(t-1)_{i-p+p}}{p} \quad (17)$$

- *macd(p)*: La media móvil de convergencia/divergencia es un indicador de trading usado en el análisis técnico. El indicador está diseñado para revelar cambios en la fuerza, dirección, momento y duración de la tendencia del precio. Appel (2003) detalla el cálculo y la forma de aplicación de este indicador.

- *macd señal(p)*: Atributo numérico en función de la media móvil de convergencia/divergencia. Este indicador sirve para identificar los puntos en los cambios son más fuertes. De igual forma que el anterior indicador, su cálculo y explicación pueden ser encontrados en el trabajo realizado por Appel (2003).
- *stoch(p)*: El oscilador estocástico es un indicador de momento que muestra la ubicación relativa del precio de cierre con respecto al rango de precios alto-bajo sobre un número p de instancias. El objetivo de esta variable es identificar la velocidad o impulso del precio. Como una regla general, el impulso cambia de dirección antes del precio. El cálculo se lo define a continuación:

$$stoch(p)_i = \left(\frac{cierre_i - Min(bajo_p)}{Max(alto_p) - Min(bajo_p)} \right) \times 100 \quad (18)$$

El código desarrollado para la creación de cada una de las variables y conformación de los conjuntos de datos finales por cada par de divisa se puede ver en el Anexo II. Módulo de Transformación y Preprocesamiento. El preprocesamiento realizado a la base de datos se detalla brevemente en la siguiente subsección, de igual forma su implementación puede ser vista en el mismo anexo.

4.2.1 Preprocesamiento

Una vez extraídos los datos desde la fuente y creadas todas las variables necesarias para el estudio, se procedió a validar la consistencia de la información, depurar valores perdidos o agrupar categorías con muy poca frecuencia. En general, la información extraída contó con buena calidad. Los ajustes y procesamientos que necesitó el conjunto de datos, antes de pasar a la implementación de las distintas metodologías de aprendizaje automático, fueron los siguientes:

1. Depuración de valores perdidos

Inicialmente, el conjunto de datos extraído no contó con valores perdidos; sin embargo, la creación de variables en función de p instancias o periodos pasados crea valores perdidos para esa primeros p registros. Los valores perdidos deben de ser eliminados debido a que no aportan información al modelo, de igual forma las ventanas de tiempo y la longitud de los conjuntos de datos de entrenamiento elegidos hacen factible la depuración de estos valores.

2. Consolidar horas de trading de cada instancia

Los datos fueron extraídos con una frecuencia de 6 horas cada uno. A pesar de esto, en el conjunto de datos aparecen horas de 3, 4, 9, 10, 15, 16, 21 y 22. Existen varios motivos por los que puede suceder esto como: el ajuste horario que se da en países como Estados Unidos y Reino Unido, las horas en las que se abren los mercados en cada país o la precisión de la versión gratuita de la API REST de FXCM. A priori, se estima que el impacto del desajuste de esta variable es mínimo, los cambios horarios solo suceden en pocas ocasiones y al final se mantiene consistentemente la diferencia de 6 horas entre cada instancia. Por otro lado, la variable de hora solo debería de contener 4 categorías, para arreglar esto simplemente se crearon los siguientes intervalos para agrupar las horas: 22h00 – 04h00, 04h00 – 10h00, 10h00 – 16h00, 16h00 – 22h00.

3. Agrupar días de trading

Los días de trading solo deberían estar comprendidos entre lunes y viernes, dado que el mercado de divisas se abre solo en días laborables. Sin embargo, al extraer la información se encontró instancias registradas en domingo. De igual forma que las imprecisiones de horas, esta situación se podría explicar debido al cambio horario que existe entre los países. En este caso, los domingos pasaron a formar parte de los lunes para que solo existan 5 categorías. Además, la frecuencia y cantidad de información de instancias en domingo era muy baja como para mantenerla como una categoría aparte.

4.2.2 Características generales

El conjunto de datos final, en este punto, está listo para servir de insumo de las distintas técnicas de aprendizaje automático que se van a comparar en el trabajo. Como se mencionaba anteriormente, el estudio será realizado para distintos pares de divisas, por lo que, se tendrán varios conjuntos de datos por separado. Una breve descripción estadística de cada una de las variables de los conjuntos de datos puede servir para conocer de mejor forma la información que se utilizará para la predicción del movimiento del precio. Por lo general, el análisis es mejor separarlo por tipo de variable. Entonces, en la Tabla 4 se puede encontrar la información relevante de cada una de las variables categóricas que conforman los conjuntos de datos; mientras que, en la Tabla 5 se puede observar puntos estadísticos importantes de las variables numéricas.

Par de Divisa	Variable	# Clases	Más Frecuente	Frecuencia
EUR/USD	hora	4	16	1230
	dia	6	Lunes	823

	clase	2	UP	2159
USD/JPY	hora	4	16	1225
	dia	6	Lunes	823
	clase	2	UP	2192
AUD/JPY	hora	4	16	1229
	dia	6	Lunes	823
	clase	2	UP	2230

Tabla 4. Características generales de las variables categóricas

La descripción de las variables categóricas muestra que la hora de trading más frecuente fue 16h00. El día con mayor número de instancias es el lunes, esto se debe a que en la fase de preprocesamiento se agrupó el domingo con el lunes. Cabe resaltar que el movimiento de precio que más se repite en todos los pares de divisas es el de subida. Por otro lado, la estadística descriptiva de las variables numéricas muestra que los precios de apertura, de cierre, alto y bajo tienen muy poca diferencia. La alta liquidez del mercado Forex contribuye a que se de esta particularidad. Esto también se refleja en el alto volumen promedio de transacciones por instancia o periodo.

Par de Divisa	Variable	Media	Desviación	Mínimo	25%	50%	75%	Máximo
EUR/USD	apertura	1,14	0,04	1,05	1,11	1,14	1,18	1,25
	cierre	1,14	0,04	1,05	1,11	1,14	1,18	1,25
	alto	1,15	0,05	1,05	1,11	1,14	1,18	1,26
	bajo	1,14	0,04	1,05	1,11	1,14	1,18	1,25
	volumen	65942,40	49970,96	1,00	31972,00	55036,00	87577,75	569011,00
	roc	0,00	0,21	-1,47	-0,09	0,00	0,11	1,61
	psar	1,14	0,05	1,05	1,11	1,14	1,18	1,26
	roc(t-1)	0,00	0,21	-1,47	-0,09	0,00	0,11	1,61
	direccion	0,00	0,00	-0,02	0,00	0,00	0,00	0,02
	bb_bbhi (5)	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	bb_bbli (5)	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	bb_bbw (5)	0,62	0,39	0,04	0,34	0,54	0,80	3,78
	atr (5)	0,00	0,00	0,00	0,00	0,00	0,00	0,01
	willr (5)	-49,35	29,68	-100,98	-76,36	-49,13	-21,73	26,62
	rsi (5)	50,54	21,07	2,26	33,91	50,33	66,64	98,00
	close ma (5)	1,14	0,04	1,05	1,11	1,14	1,18	1,25
	roc(t-1) ma (5)	0,00	0,09	-0,46	-0,06	0,00	0,06	0,41
	macd (5)	0,00	0,00	-0,01	0,00	0,00	0,00	0,01
	macd signal (5)	0,00	0,00	-0,01	0,00	0,00	0,00	0,01
	stoch (5)	50,65	29,68	-0,98	23,64	50,87	78,27	126,62
bb_bbhi (10)	0,05	0,22	0,00	0,00	0,00	0,00	1,00	
bb_bbli (10)	0,04	0,20	0,00	0,00	0,00	0,00	1,00	
bb_bbw (10)	0,93	0,50	0,12	0,59	0,83	1,16	3,95	
atr (10)	0,00	0,00	0,00	0,00	0,00	0,00	0,01	

	willr (10)	-49,36	30,20	-100,00	-77,51	-49,04	-21,77	7,24
	rsi (10)	50,65	15,11	9,59	39,20	50,91	62,02	91,51
	close ma (10)	1,14	0,04	1,05	1,11	1,14	1,18	1,25
	roc(t-1) ma (10)	0,00	0,07	-0,38	-0,04	0,00	0,05	0,29
	macd (10)	0,00	0,00	-0,01	0,00	0,00	0,00	0,01
	macd signal (10)	0,00	0,00	-0,01	0,00	0,00	0,00	0,01
	stoch (10)	50,64	30,20	0,00	22,49	50,96	78,23	107,24
	bb_bbhi (15)	0,07	0,25	0,00	0,00	0,00	0,00	1,00
	bb_bbli (15)	0,05	0,21	0,00	0,00	0,00	0,00	1,00
	bb_bbw (15)	1,16	0,59	0,20	0,77	1,03	1,41	5,45
	atr (15)	0,00	0,00	0,00	0,00	0,00	0,00	0,01
	willr (15)	-49,23	30,39	-100,00	-77,73	-48,47	-21,36	4,09
	rsi (15)	50,70	12,26	15,84	41,59	50,85	59,24	84,73
	close ma (15)	1,14	0,04	1,06	1,11	1,14	1,18	1,25
	roc(t-1) ma (15)	0,00	0,05	-0,30	-0,03	0,00	0,04	0,22
	macd (15)	0,00	0,00	-0,01	0,00	0,00	0,00	0,01
	macd signal (15)	0,00	0,00	-0,01	0,00	0,00	0,00	0,01
	stoch (15)	50,77	30,39	0,00	22,27	51,53	78,64	104,09
	bb_bbhi (20)	0,07	0,26	0,00	0,00	0,00	0,00	1,00
	bb_bbli (20)	0,05	0,23	0,00	0,00	0,00	0,00	1,00
	bb_bbw (20)	1,36	0,68	0,23	0,92	1,21	1,62	6,25
	atr (20)	0,00	0,00	0,00	0,00	0,00	0,00	0,01
	willr (20)	-48,95	30,66	-100,00	-78,25	-46,91	-20,83	4,09
	rsi (20)	50,74	10,54	20,84	43,22	50,85	57,99	81,99
	close ma (20)	1,14	0,04	1,06	1,11	1,14	1,18	1,24
	roc(t-1) ma (20)	0,00	0,05	-0,23	-0,03	0,00	0,03	0,20
	macd (20)	0,00	0,00	-0,01	0,00	0,00	0,00	0,01
	macd signal (20)	0,00	0,00	-0,01	0,00	0,00	0,00	0,01
	stoch (20)	51,05	30,66	0,00	21,75	53,09	79,17	104,09
	bb_bbhi (30)	0,08	0,27	0,00	0,00	0,00	0,00	1,00
	bb_bbli (30)	0,05	0,21	0,00	0,00	0,00	0,00	1,00
	bb_bbw (30)	1,67	0,82	0,44	1,12	1,47	1,97	6,95
	atr (30)	0,00	0,00	0,00	0,00	0,00	0,00	0,01
	willr (30)	-48,58	30,57	-100,00	-77,84	-45,89	-20,82	0,42
	rsi (30)	50,79	8,54	26,59	44,92	50,77	56,49	77,84
	close ma (30)	1,14	0,04	1,06	1,11	1,14	1,18	1,24
	roc(t-1) ma (30)	0,00	0,04	-0,20	-0,02	0,00	0,03	0,15
	macd (30)	0,00	0,00	-0,01	0,00	0,00	0,00	0,01
	macd signal (30)	0,00	0,00	-0,01	0,00	0,00	0,00	0,01
	stoch (30)	51,42	30,57	0,00	22,16	54,11	79,18	100,42
USD/JPY	apertura	109,54	2,70	102,30	107,53	109,63	111,61	115,35
	cierre	109,54	2,70	102,30	107,53	109,63	111,61	115,35
	alto	109,70	2,70	102,61	107,70	109,77	111,76	115,63
	bajo	109,36	2,70	101,19	107,34	109,47	111,44	115,19
	volumen	67921,67	48903,81	1,00	43417,00	58862,00	80380,00	648264,00

roc	0,00	0,22	-1,71	-0,11	0,00	0,11	1,70
psar	109,52	2,77	101,19	107,44	109,63	111,71	115,63
roc(t-1)	0,00	0,22	-1,71	-0,11	0,00	0,11	1,70
direccion	0,00	0,24	-1,79	-0,12	0,00	0,11	1,76
bb_bbhi (5)	0,00	0,00	0,00	0,00	0,00	0,00	0,00
bb_bbli (5)	0,00	0,00	0,00	0,00	0,00	0,00	0,00
bb_bbw (5)	0,64	0,47	0,05	0,34	0,53	0,80	4,91
atr (5)	0,35	0,16	0,08	0,25	0,32	0,40	1,64
willr (5)	-47,95	29,12	-139,40	-73,16	-47,65	-21,65	87,07
rsi (5)	50,25	20,14	4,43	35,62	50,38	65,82	98,20
close ma (5)	109,54	2,69	102,61	107,54	109,59	111,60	115,07
roc(t-1) ma (5)	0,00	0,10	-0,60	-0,05	0,00	0,05	0,65
macd (5)	0,00	0,14	-1,03	-0,07	0,00	0,07	1,04
macd signal (5)	0,00	0,13	-0,94	-0,07	0,00	0,07	0,91
stoch (5)	52,05	29,12	-39,40	26,84	52,35	78,35	187,07
bb_bbhi (10)	0,04	0,20	0,00	0,00	0,00	0,00	1,00
bb_bbli (10)	0,05	0,22	0,00	0,00	0,00	0,00	1,00
bb_bbw (10)	0,95	0,63	0,07	0,54	0,79	1,17	5,84
atr (10)	0,35	0,15	0,10	0,26	0,32	0,40	1,39
willr (10)	-48,12	29,58	-122,49	-73,74	-48,44	-21,14	5,78
rsi (10)	50,04	14,41	10,01	39,66	50,39	60,64	92,53
close ma (10)	109,55	2,68	103,31	107,55	109,62	111,58	114,94
roc(t-1) ma (10)	0,00	0,07	-0,46	-0,04	0,00	0,04	0,36
macd (10)	-0,01	0,16	-1,05	-0,09	0,00	0,09	0,82
macd signal (10)	-0,01	0,15	-0,95	-0,09	0,00	0,08	0,71
stoch (10)	51,88	29,58	-22,49	26,26	51,56	78,86	105,78
bb_bbhi (15)	0,06	0,23	0,00	0,00	0,00	0,00	1,00
bb_bbli (15)	0,07	0,25	0,00	0,00	0,00	0,00	1,00
bb_bbw (15)	1,19	0,76	0,12	0,69	1,01	1,46	7,07
atr (15)	0,35	0,14	0,12	0,26	0,32	0,40	1,29
willr (15)	-48,25	29,99	-118,64	-75,29	-47,18	-20,47	5,68
rsi (15)	49,91	11,67	14,00	41,59	50,13	58,15	87,98
close ma (15)	109,55	2,67	103,37	107,57	109,61	111,57	114,90
roc(t-1) ma (15)	0,00	0,06	-0,32	-0,03	0,00	0,03	0,31
macd (15)	-0,01	0,21	-1,32	-0,13	0,00	0,12	1,01
macd signal (15)	-0,01	0,19	-1,13	-0,12	0,00	0,11	0,94
stoch (15)	51,75	29,99	-18,64	24,71	52,82	79,53	105,68
bb_bbhi (20)	0,06	0,23	0,00	0,00	0,00	0,00	1,00
bb_bbli (20)	0,07	0,26	0,00	0,00	0,00	0,00	1,00
bb_bbw (20)	1,39	0,85	0,16	0,84	1,18	1,71	7,22
atr (20)	0,35	0,14	0,13	0,26	0,32	0,40	1,21
willr (20)	-48,31	30,21	-118,64	-76,21	-47,77	-20,14	5,61
rsi (20)	49,82	9,97	16,95	42,84	49,73	56,85	83,98
close ma (20)	109,56	2,66	103,41	107,59	109,62	111,56	115,01
roc(t-1) ma (20)	0,00	0,05	-0,29	-0,03	0,00	0,03	0,27

	macd (20)	-0,01	0,22	-1,34	-0,13	0,00	0,12	1,07
	macd signal (20)	-0,01	0,20	-1,11	-0,12	0,00	0,11	0,96
	stoch (20)	51,69	30,21	-18,64	23,79	52,23	79,86	105,61
	bb_bbhi (30)	0,05	0,23	0,00	0,00	0,00	0,00	1,00
	bb_bbli (30)	0,08	0,27	0,00	0,00	0,00	0,00	1,00
	bb_bbw (30)	1,72	1,01	0,20	1,08	1,50	2,10	7,75
	atr (30)	0,35	0,13	0,15	0,27	0,32	0,40	1,08
	willr (30)	-48,14	30,17	-110,92	-75,37	-46,72	-20,56	5,61
	rsi (30)	49,68	7,91	21,18	44,34	49,49	55,28	77,56
	close ma (30)	109,57	2,64	103,47	107,62	109,60	111,62	115,50
	roc(t-1) ma (30)	0,00	0,04	-0,24	-0,02	0,00	0,02	0,21
	macd (30)	-0,02	0,26	-1,48	-0,17	-0,01	0,13	1,14
	macd signal (30)	-0,02	0,24	-1,21	-0,16	-0,02	0,12	1,01
	stoch (30)	51,86	30,17	-10,92	24,63	53,28	79,44	105,61
AUD/JPY	apertura	79,45	5,62	60,59	75,15	79,38	83,73	90,14
	cierre	79,45	5,62	60,59	75,15	79,38	83,71	90,14
	alto	79,63	5,61	62,95	75,31	79,52	83,90	90,32
	bajo	79,26	5,65	59,91	74,98	79,22	83,53	89,95
	volumen	62952,30	43312,92	2,00	41329,00	55082,00	74692,00	580676,00
	roc	0,00	0,33	-2,92	-0,16	0,01	0,16	3,46
	psar	79,41	5,69	59,91	75,05	79,42	83,87	90,32
	roc(t-1)	0,00	0,33	-2,92	-0,16	0,01	0,16	3,46
	direccion	0,00	0,25	-1,82	-0,13	0,01	0,13	2,15
	bb_bbhi (5)	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	bb_bbli (5)	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	bb_bbw (5)	0,94	0,70	0,07	0,51	0,77	1,17	9,53
	atr (5)	0,37	0,16	0,13	0,28	0,35	0,41	2,04
	willr (5)	-47,59	29,32	-138,81	-73,31	-46,87	-21,18	61,14
	rsi (5)	50,66	20,38	2,89	35,70	50,68	65,97	97,83
	close ma (5)	79,46	5,62	62,47	75,16	79,36	83,73	89,80
	roc(t-1) ma (5)	0,00	0,14	-1,28	-0,07	0,00	0,08	1,54
	macd (5)	0,00	0,15	-1,20	-0,08	0,00	0,08	0,74
	macd signal (5)	0,00	0,14	-1,00	-0,07	0,00	0,08	0,65
	stoch (5)	52,41	29,32	-38,81	26,69	53,13	78,82	161,14
	bb_bbhi (10)	0,05	0,21	0,00	0,00	0,00	0,00	1,00
	bb_bbli (10)	0,05	0,21	0,00	0,00	0,00	0,00	1,00
	bb_bbw (10)	1,39	0,92	0,19	0,81	1,17	1,68	9,05
	atr (10)	0,37	0,15	0,16	0,29	0,35	0,41	1,72
	willr (10)	-47,83	29,77	-136,10	-74,24	-47,16	-20,98	36,63
	rsi (10)	50,26	14,73	6,87	39,89	50,20	60,71	89,20
	close ma (10)	79,46	5,62	63,38	75,16	79,36	83,67	89,55
	roc(t-1) ma (10)	0,00	0,10	-0,71	-0,06	0,00	0,06	0,49
	macd (10)	0,00	0,16	-1,02	-0,10	0,00	0,10	0,77
	macd signal (10)	0,00	0,15	-0,80	-0,09	0,00	0,09	0,68
stoch (10)	52,17	29,77	-36,10	25,76	52,84	79,02	136,63	

bb_bbhi (15)	0,06	0,24	0,00	0,00	0,00	0,00	1,00
bb_bbli (15)	0,06	0,23	0,00	0,00	0,00	0,00	1,00
bb_bbw (15)	1,73	1,07	0,29	1,03	1,48	2,06	9,64
atr (15)	0,37	0,14	0,18	0,30	0,35	0,41	1,56
willr (15)	-47,93	30,16	-122,07	-75,11	-47,63	-20,12	19,35
rsi (15)	50,09	12,09	11,88	41,81	50,39	58,59	84,94
close ma (15)	79,46	5,62	63,71	75,15	79,36	83,65	89,40
roc(t-1) ma (15)	0,00	0,08	-0,69	-0,05	0,00	0,05	0,60
macd (15)	-0,01	0,21	-1,21	-0,13	0,01	0,13	0,94
macd signal (15)	-0,01	0,20	-0,98	-0,12	0,01	0,12	0,84
stoch (15)	52,07	30,16	-22,07	24,89	52,37	79,88	119,35
bb_bbhi (20)	0,06	0,24	0,00	0,00	0,00	0,00	1,00
bb_bbli (20)	0,06	0,24	0,00	0,00	0,00	0,00	1,00
bb_bbw (20)	2,02	1,22	0,45	1,26	1,73	2,40	10,23
atr (20)	0,37	0,13	0,19	0,30	0,36	0,41	1,46
willr (20)	-47,86	30,11	-122,07	-75,14	-47,05	-20,01	18,00
rsi (20)	50,00	10,47	16,53	43,12	50,29	57,15	81,16
close ma (20)	79,47	5,61	63,93	75,16	79,40	83,63	89,32
roc(t-1) ma (20)	0,00	0,07	-0,48	-0,04	0,00	0,04	0,46
macd (20)	-0,01	0,23	-1,22	-0,14	0,01	0,14	0,96
macd signal (20)	-0,01	0,21	-1,00	-0,13	0,01	0,13	0,89
stoch (20)	52,14	30,11	-22,07	24,86	52,95	79,99	118,00
bb_bbhi (30)	0,06	0,23	0,00	0,00	0,00	0,00	1,00
bb_bbli (30)	0,07	0,26	0,00	0,00	0,00	0,00	1,00
bb_bbw (30)	2,49	1,47	0,56	1,56	2,11	2,98	12,32
atr (30)	0,37	0,12	0,20	0,30	0,36	0,41	1,29
willr (30)	-47,84	30,08	-110,21	-75,19	-46,83	-20,21	11,29
rsi (30)	49,88	8,48	21,05	44,27	50,23	55,58	74,50
close ma (30)	79,48	5,61	64,51	75,16	79,60	83,61	89,18
roc(t-1) ma (30)	0,00	0,06	-0,35	-0,04	0,00	0,04	0,29
macd (30)	-0,01	0,28	-1,47	-0,17	0,01	0,17	1,16
macd signal (30)	-0,01	0,26	-1,23	-0,16	0,01	0,16	1,07
stoch (30)	52,16	30,08	-10,21	24,81	53,17	79,79	111,29

Tabla 5. Estadística descriptiva de las variables numéricas

El análisis de las variables numéricas del conjunto de datos final sugiere varios problemas como multicolinealidad o poca variabilidad de ciertas variables. Estos problemas serán resueltos mediante el uso de técnicas de aprendizaje automático automatizado, donde se construirá un flujo de trabajo que remueva variables con alta multicolinealidad o poca variabilidad. Además, el flujo se encargará de determinar las variables explicativas más importantes de todo el conjunto de datos para tener una mayor eficiencia y reducir el posible sesgo. Todos estos aspectos son importantes de considerar, ya que la inclusión de variables

poco relevantes, con problemas de multicolinealidad o poca variabilidad saben tener un efecto negativo en el rendimiento de los algoritmos de aprendizaje automático.

4.3 Métricas de Rendimiento

Existen bastantes técnicas diferentes para medir y evaluar el rendimiento de los modelos de aprendizaje automático. Una de las opciones más populares cuando se tiene datos limitados es aplicar el procedimiento de validación cruzada. Este procedimiento da como resultado métricas de evaluación más confiables para un modelo, particularmente, sobre datos no observados (fuera del conjunto de entrenamiento). Así mismo, la validación cruzada ofrece un procedimiento consistente para comparar la capacidad de clasificación de diferentes modelos sobre un mismo problema (Gerlein et al., 2016). Para los experimentos, el número de k subconjuntos (o folds) en el que se divide el conjunto de entrenamiento depende del número de datos que contenga. En el escenario de entrenamiento con 150 datos, se dividirá a la base en 3 subconjuntos utilizando la estrategia para series temporales (i.e., dividir al conjunto de datos en segmentos ordenados de igual dimensión). De igual forma, en el entrenamiento con 750 y 1500 datos, se tendrán 6 y 10 subconjuntos de datos, respectivamente, para realizar la validación cruzada.

Los modelos serán reentrenados continuamente utilizando una ventana móvil de dimensión k . Esta ventana móvil también será un parámetro con el que se establecerán distintos escenarios. Es así como, si se establece una ventana móvil de 5 registros o instancias, para realizar un backtesting de dos años se tendría que realizar 400 veces el entrenamiento de un modelo (suponiendo que cada año consta de 250 días hábiles de trading y recordando que por cada día se tiene el registro de 4 precios al cierre). El procedimiento de validación cruzada se usa durante cada entrenamiento para medir la precisión promedio del modelo evaluado sobre v subconjuntos de datos; por tanto, para cada periodo se reporta la precisión promedio que se obtiene a partir de este procedimiento. Finalmente, en los resultados se reportará como una métrica de evaluación, la precisión media de todos los periodos de entrenamiento por cada experimento.

El problema de predicción financiera debe ser abordado bajo una perspectiva diferente debido a las complicaciones que pueden invalidar las predicciones (volatilidad alta de los mercados financieros, sesgo en los datos, problemas de generalización, entre otros). Una de las principales problemáticas en el aprendizaje automático es encontrar una compensación

adecuada entre el sesgo y la varianza (Witten et al., 2005). El sesgo es el error de la predicción versus lo observado (i.e., la proporción de predicciones erróneas). La varianza es el error que resulta de ajustar perfectamente un modelo a un pequeño conjunto de datos, el cual no representa a la población en general; por tanto, no tiene la capacidad de generalizar y cae en un sobreajuste. El principal inconveniente en las series temporales financieras es que los patrones que se pueden encontrar en todo el conjunto de datos de un periodo determinado pueden ser obsoletos para un periodo posterior. Los experimentos planteados, bajo los diferentes escenarios y parámetros, tienen como objetivo minimizar el sesgo y la varianza tomando en cuenta estas características especiales que tienen las series temporales de los mercados financieros.

El reentrenamiento de los modelos permite la disminución del sesgo de los modelos a un corto plazo, mientras al mismo tiempo soluciona la varianza a un largo plazo, lo que evita el uso de muestras grandes para construir los modelos (Maggini et al., 1997). Como se comentaba, los patrones que determinan la variación de precios en los mercados financieros son sumamente cambiantes. En general, los patrones que se encuentran inmediatamente antes del periodo que se desea predecir son mucho más importantes que aquellos que están más alejados. Este comportamiento es mucho más notorio en las series temporales financieras. Por estas razones, en el presente trabajo se establecen experimentos con tamaños relativamente pequeños de datos y ventanas temporales de reentrenamiento cortas. Los modelos entrenados van a aprender solo los patrones más cercanos al periodo que se quiere predecir, además, se realizará la predicción de los valores más próximos. Este proceso se repetirá M veces en función de la ventana de reentrenamiento y el periodo de backtesting establecido.

El backtesting fue realizado como una simulación de la aplicación de los modelos en la vida real. El modelo entrenado sirve para la predicción de los k valores siguientes, recordando que k está definido por el rango de los valores fuera del conjunto de entrenamiento o dimensión de la ventana móvil. Entonces, los k valores son los valores futuros que se desea predecir, estos son totalmente independientes al modelo. La predicción de todos los valores futuros mediante esta simulación también será evaluada y reportada como otra métrica de precisión de backtesting.

Finalmente, la métrica de retorno financiero acumulado también se evalúa para comparar las distintas metodologías. Para las aplicaciones de trading, en general, una mayor precisión del modelo no implica directamente una mayor rentabilidad y viceversa. Si el retorno obtenido a partir de una serie de acciones de compraventa no son lo suficientemente altas para cubrir los costos asociados al trading (comisiones, impuestos, imprecisiones de operaciones, entre otros), cualquier estrategia de trading eventualmente generará pérdidas. En este sentido, el

retorno acumulado sobre un periodo de tiempo representa la métrica más precisa para este tipo de estudios. Adicionalmente, el retorno acumulado representa un promedio ponderado de los beneficios obtenidos a partir de la ratio de predicciones correctas; por tanto, la medida es de especial importancia para medir el rendimiento de la inversión. En resumen, los resultados de las simulaciones de trading, a partir de las diferentes metodologías, serán medidos y comparados con las siguientes métricas:

- Precisión (%): porcentaje de las predicciones correctas para el periodo de trading a partir de la técnica de validación cruzada de cada entrenamiento de los modelos.

$$Precisión = \sum_j^M \frac{Precision\ validación\ cruzada_j}{M} \quad (19)$$

Donde M es el número de veces en las que se reentrenará los modelos de aprendizaje automático.

- Precisión backtesting (%): porcentaje de las predicciones correctas para el periodo de trading simulando su aplicación en la realidad.

$$Precisión\ backtesting = \frac{\# predicciones\ correctas}{\# total\ de\ predicciones} \quad (20)$$

- Retorno acumulado (%): porcentaje del retorno acumulado al final del ciclo de trading. El retorno acumulado para el i -ésimo periodo es calculado como:

$$Retorno\ acumulado_i = (1 + Retorno\ Acumulado_{i-1}) \times (1 + Retorno_i) - 1 \quad (21)$$

4.4 Clasificadores de Aprendizaje Automático

Las distintas metodologías que se van a comparar en el presente trabajo utilizan varios modelos de aprendizaje automático. Es así como, una metodología puede llevar a ajustar y probar distintos modelos para cada uno de los k periodos de ventana móvil sobre los cuales se realizará el entrenamiento. Entonces, el ajuste de los modelos será realizado con técnicas de aprendizaje automático automatizado (o mejor conocido como AutoML). Estas técnicas permiten automatizar la construcción del flujo de trabajo de los modelos de aprendizaje automático (He et al., 2021). En general, un sistema completo de aprendizaje automático automatizado consiste en varios procesos: preparación de data, ingeniería de variables, generación de modelos, y su evaluación. La aplicación y automatización de estos procesos

permiten conformar las metodologías que van a ser comparadas. El módulo PyCaret será usado como marco de trabajo para aplicar el AutoML.

Los clasificadores que se utilizarán para ajustar los modelos, y que PyCaret tiene disponibles, son los siguientes: Regresión Logística, Árboles de Decisión, Bosques Aleatorios, Máquinas de Vectores de Soporte (kernel lineal y radial), Clasificador Gaussian Process, Clasificador Ridge, Clasificador Extra-Trees, Clasificador Gradient Boosting, AdaBoost, CatBoost, XGBoost y Red Neuronal con arquitectura Perceptrón Multicapa (MLP). Como se puede observar, estos clasificadores contemplan la mayoría de los tipos de modelos de aprendizaje automático, que van desde regresiones hasta redes neuronales. El ajuste de parámetros para cada modelo será el que por defecto tiene configurado PyCaret. Antes de detallar las diferentes metodologías que van a ser comparadas en este trabajo, se procederá a describir brevemente las características más relevantes de los modelos de aprendizaje automático usados. Además, se mostrará la configuración de parámetros que el flujo de AutoML les da por defecto a cada uno de los modelos.

Regresión Logística

La regresión logística, a pesar de su nombre, es un modelo de clasificación más que de regresión. La regresión logística es un método más simple y efectivo para problemas de clasificación binarios y lineales. Este es un modelo de clasificación relativamente fácil de realizar y que, en general, tiene un buen rendimiento con clases que son linealmente separables. Además, este es uno de los algoritmos de clasificación con mayor uso en toda la industria (Subasi, 2020). El modelo de regresión logística es uno de los métodos estadísticos clásicos para realizar clasificación.

Los parámetros con los que por defecto viene configurado este clasificador, y con los que se realizó el entrenamiento, se pueden observar en la Figura 2.

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                  intercept_scaling=1, l1_ratio=None, max_iter=1000,
                  multi_class='auto', n_jobs=None, penalty='l2',
                  random_state=777, solver='lbfgs', tol=0.0001, verbose=0,
                  warm_start=False)
```

Figura 2. Parámetros de entrenamiento de la regresión logística

Árboles de Decisión

Esta metodología intuitivamente puede ser vista como un enfoque de divide y conquista aplicada a la clasificación. Los árboles de decisión pueden ser usados para descubrir variables

relevantes y extraer patrones en grandes bases, lo cual permite discriminar y predecir correctamente las clases. Estas características que tiene el modelo, junto con la fácil interpretación que se le puede dar (i.e., no es un algoritmo de caja negra), son algunas de las razones por las que los árboles de decisión han sido tan utilizados para el análisis exploratorio de los datos y la modelización predictiva (Myles et al., 2004). Además, los árboles de decisión son la metodología base para muchos modelos más complejos que se utilizan hoy en día (bosques aleatorios, XGBoost, entre otros).

La librería PyCaret otorga por defecto los parámetros mostrados en la Figura 3 para el entrenamiento de este algoritmo.

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                      max_depth=None, max_features=None, max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, presort='deprecated',
                      random_state=777, splitter='best')
```

Figura 3. Parámetros de entrenamiento de los árboles de decisión

Bosques Aleatorios

El algoritmo de bosques aleatorios es una extensión del método de “bagging”, debido a que utiliza tanto el “bagging” como la selección aleatoria de variables para construir un conjunto de árboles de decisión no correlacionados entre sí. La selección aleatoria de variables genera un subconjunto aleatorio de características, lo cual asegura que exista una baja correlación entre los árboles de decisión (Breiman, 2001). Esta es una diferencia clave entre los árboles de decisión y los bosques aleatorios. Los principales beneficios de los bosques aleatorios a comparación de los árboles de decisión son la reducción del riesgo de caer en un sobreajuste del modelo, una mayor flexibilidad y precisión, y la facilidad de determinar la importancia de cada variable. Por otro lado, las desventajas de este algoritmo son el tiempo de cálculo del proceso, la cantidad de recursos que se necesitan y la complejidad en su interpretación.

La configuración predeterminada de los parámetros del algoritmo de bosques aleatorios se puede observar en la Figura 4.

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                      criterion='gini', max_depth=None, max_features='auto',
                      max_leaf_nodes=None, max_samples=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, n_estimators=100,
                      n_jobs=-1, oob_score=False, random_state=777, verbose=0,
                      warm_start=False)
```


Figura 4. Parámetros de entrenamiento de los bosques aleatorios

Máquinas de Vectores de Soporte

El algoritmo fue desarrollado por Cortes & Vapnik (1995) para la clasificación binaria. Este algoritmo ha sido aplicado para un vasto número de áreas incluyendo categorización de texto, reconocimiento de letra escrita, detección de caras, análisis de datos genéticos, entre muchas otras. El método combina ideas de la teoría del aprendizaje estadístico y la optimización convexa con el fin de encontrar límites posibles en el espacio de datos que separen dos clases de puntos (Mammone et al., 2009). En general, las máquinas de vectores de soporte son bastante sensibles a la elección de parámetros que se le ingresan. Los resultados de estos modelos saben ser bastantes precisos cuando se trata con datos que no pueden ser linealmente separables; sin embargo, el kernel que se elija influye bastante en el rendimiento. Los dos tipos de kernels que serán usados en este trabajo son lineal y radial.

PyCaret predefine los parámetros de entrenamiento de las SVM con kernel lineal y radial como se muestra en la Figura 5 y Figura 6, respectivamente.

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=True, random_state=777, shrinking=True, tol=0.001,
    verbose=False)
```

Figura 5. Parámetros de entrenamiento de la SVM con kernel lineal

```
SGDClassifier(alpha=0.0001, average=False, class_weight=None,
    early_stopping=False, epsilon=0.1, eta0=0.001, fit_intercept=True,
    l1_ratio=0.15, learning_rate='optimal', loss='hinge',
    max_iter=1000, n_iter_no_change=5, n_jobs=-1, penalty='l2',
    power_t=0.5, random_state=777, shuffle=True, tol=0.001,
    validation_fraction=0.1, verbose=0, warm_start=False)
```

Figura 6. Parámetros de entrenamiento de la SVM con kernel radial

Clasificador Gaussian Process

Esta metodología se basa en la estadística bayesiana; por tanto, es un método de clasificación no paramétrico. Este asume alguna distribución previa sobre las densidades de probabilidad subyacentes. La clasificación final se determina como aquella que mejor se ajusta a los datos observados, mientras al mismo tiempo cumple con algunas propiedades de suavidad. En general, este clasificador es bastante efectivo. Williams & Rasmussen (2006) lo han popularizado en la comunidad de aprendizaje automático.

Los parámetros con lo que normalmente se realiza el entrenamiento del clasificador por proceso gaussiano se muestran en la Figura 7.

```
GaussianProcessClassifier(copy_X_train=False, kernel=None, max_iter_predict=100,
                          multi_class='one_vs_rest', n_jobs=-1,
                          n_restarts_optimizer=0, optimizer='fmin_l_bfgs_b',
                          random_state=777, warm_start=False)
```

Figura 7. Parámetros de entrenamiento del clasificador por proceso gaussiano

Clasificador Ridge

La regresión Ridge es un método para estimar los coeficientes de regresiones múltiples en escenarios donde las variables independientes tienen una alta correlación (Hilt & Seegrist, 1977). Esta ha sido usada en muchos campos incluyendo la econometría, química e ingeniería. El problema de regresión puede ser transformado a uno de clasificación simplemente poniendo como valores de 1 y -1 a cada clase, respectivamente. Este método principalmente fue diseñado como una posible solución a la imprecisión de los estimadores de mínimos cuadrados cuando los modelos de regresión lineal presentan problemas de multicolinealidad. En teoría, el clasificador Ridge provee de estimadores más precisos en presencia de multicolinealidad (Gruber, 2017).

El flujo de AutoML realiza el entrenamiento de este algoritmo de manera estándar con los parámetros observados en la Figura 8.

```
RidgeClassifier(alpha=1.0, class_weight=None, copy_X=True, fit_intercept=True,
                max_iter=None, normalize=False, random_state=777, solver='auto',
                tol=0.001)
```

Figura 8. Parámetros de entrenamiento del clasificador ridge

Clasificador Extra Trees

El clasificador de árboles extremadamente aleatorios (clasificador Extra Trees) es un tipo de técnica de aprendizaje en conjunto, la cual agrega los resultados de múltiples árboles de decisión independientes recolectados en un “bosque” para generar el resultado de su clasificación. En concepto, este algoritmo es muy similar al de Bosques Aleatorios. La única diferencia es la manera de construir los árboles de decisión dentro de los bosques. Además de la precisión, la principal fortaleza de este algoritmo es la eficiencia computacional (Geurts et al., 2006).

En la Figura 9 se puede observar los parámetros que por defecto utiliza PyCaret para el entrenamiento del clasificador Extra Trees.

```
ExtraTreesClassifier(bootstrap=False, ccp_alpha=0.0, class_weight=None,
                    criterion='gini', max_depth=None, max_features='auto',
                    max_leaf_nodes=None, max_samples=None,
                    min_impurity_decrease=0.0, min_impurity_split=None,
                    min_samples_leaf=1, min_samples_split=2,
                    min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=-1,
                    oob_score=False, random_state=777, verbose=0,
                    warm_start=False)
```

Figura 9. Parámetros de entrenamiento del clasificador extra trees

Clasificador Gradient Boosting

Los algoritmos de potenciación de gradiente son una familia de poderosas técnicas de aprendizaje automático que han mostrado considerable éxito en muchas aplicaciones prácticas (Natekin & Knoll, 2013). Estos pueden ser altamente personalizados para el problema en cuestión; por ejemplo, los algoritmos pueden aprender de diferentes funciones de pérdidas determinadas por el analista. La principal idea de esta metodología es mediante el entrenamiento de modelos “débiles” (e.g., árboles de decisión) se puede entrenar secuencialmente un conjunto de modelos que actúen como uno solo. En cada iteración o nuevo entrenamiento, el modelo débil se entrenará sobre el error que tiene el conjunto de modelos hasta ese momento. Lo que permitirá ir reduciendo el error de manera paulatina. El clasificador Gradient Boosting es la versión básica de esta metodología de potenciación de gradiente.

Los parámetros con los que de manera estándar PyCaret entrena el algoritmo de Gradient Boosting se los muestra a continuación, en la Figura 10.

```
GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                          learning_rate=0.1, loss='deviance', max_depth=3,
                          max_features=None, max_leaf_nodes=None,
                          min_impurity_decrease=0.0, min_impurity_split=None,
                          min_samples_leaf=1, min_samples_split=2,
                          min_weight_fraction_leaf=0.0, n_estimators=100,
                          n_iter_no_change=None, presort='deprecated',
                          random_state=777, subsample=1.0, tol=0.0001,
                          validation_fraction=0.1, verbose=0,
                          warm_start=False)
```

Figura 10. Parámetros de entrenamiento del clasificador gradient boosting

AdaBoost

Uno de los métodos más populares de “boosting” es el algoritmo Boosting Adaptativo (AdaBoost). Esta técnica realiza un modelamiento iterativo en función de clasificadores débiles o base. Una vez realizado el primer entrenamiento, el algoritmo les asigna mayor

importancia a los datos mal clasificados para la siguiente iteración, y de esta forma logra un nuevo clasificador (Barbona & Beltrán, 2018). El AdaBoost, con árboles de decisión como clasificadores débiles, a menudo se lo conoce como el mejor clasificador “listo para usar” (Wang et al., 2008).

El modelo de AdaBoost se entrena por defecto en el flujo establecido de AutoML con los parámetros mostrados en la Figura 11.

```
AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None, learning_rate=1.0,
                  n_estimators=50, random_state=777)
```

Figura 11. Parámetros de entrenamiento del adaboost

CatBoost

CatBoost es un miembro de la familia de técnicas de aprendizaje automático basadas en potenciación de gradiente. Desde su publicación en 2018, los investigadores han usado exitosamente esta metodología para varias áreas del aprendizaje automático. El algoritmo también se basa en árboles de decisión; por tanto, el mismo es eficaz en tareas de aprendizaje automático que involucran data categórica y heterogénea (Hancock & Khoshgoftaar, 2020). Uno de los principales problemas que expone la literatura acerca de este algoritmo es su sensibilidad al ajuste de hiper parámetros y la importancia de la optimización de estos hiper parámetros.

La configuración predeterminada de los parámetros del algoritmo de CatBoost se puede observar en la Figura 12.

```
{'nan_mode': 'Min', 'eval_metric': 'Logloss', 'iterations': 1000, 'sampling_frequency': 'PerTree',
 'leaf_estimation_method': 'Newton', 'grow_policy': 'SymmetricTree', 'penalties_coefficient': 1, 'boosting_type': 'Plain', 'model_shrink_mode': 'Constant', 'feature_border_type': 'GreedyLogSum', 'bayesian_matrix_reg': 0.10000000149011612, 'force_unit_auto_pair_weights': False, 'l2_leaf_reg': 3, 'random_strength': 1, 'rsm': 1, 'boost_from_average': False, 'model_size_reg': 0.5, 'pool_metadata_options': {'tags': {}}, 'subsample': 0.800000011920929, 'use_best_model': False, 'class_names': [0, 1], 'random_seed': 777, 'depth': 6, 'posterior_sampling': False, 'border_count': 254, 'classes_count': 0, 'auto_class_weights': 'None', 'sparse_features_conflict_fraction': 0, 'leaf_estimation_backtracking': 'AnyImprovement', 'best_model_min_trees': 1, 'model_shrink_rate': 0, 'min_data_in_leaf': 1, 'loss_function': 'Logloss', 'learning_rate': 0.019084999337792397, 'score_function': 'Cosine', 'task_type': 'CPU', 'leaf_estimation_iterations': 10, 'bootstrap_type': 'MVS', 'max_leaves': 64}
```

Figura 12. Parámetros de entrenamiento del catboost

XGBoost

El algoritmo de XGBoost ha ganado popularidad rápidamente los últimos años. Este tiene un impresionante récord histórico en las competiciones de ciencia de datos como Kaggle, donde es uno de los algoritmos ganadores más usados. Aunque conceptualmente es similar a los

métodos de potenciación de gradiente, el algoritmo difiere de estos en muchas maneras (Nielsen, 2016). Las diferencias, que pueden ser consideradas como ventajas con relación a los demás algoritmos de la familia de potenciación de gradiente, se las puede resumir en los siguientes puntos:

- Penalización inteligente de los árboles
- Uso del algoritmo Newton de Boosting
- Reducción proporcional de los nodos de las hojas
- Añade una mayor aleatorización de parámetros
- Selecciona automáticamente las mejores variables

En la Figura 13 se puede observar los parámetros que por defecto utiliza PyCaret para el entrenamiento del algoritmo XGBoost.

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
              gamma=0, gpu_id=-1, importance_type=None,
              interaction_constraints='', learning_rate=0.300000012,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=-1,
              num_parallel_tree=1, objective='binary:logistic',
              predictor='auto', random_state=777, reg_alpha=0, reg_lambda=1,
              scale_pos_weight=1, subsample=1, tree_method='auto',
              use_label_encoder=True, validate_parameters=1, verbosity=0)
```

Figura 13. Parámetros de entrenamiento del xgboost

Red Neuronal con arquitectura Perceptrón Multicapa

Las redes neuronales artificiales son un tipo de aprendizaje automático denominado “aprendizaje profundo” (deep learning). Existen varias clases de redes neuronales, en función de como están estructuradas o las características especiales que se añaden para enfocarse en la resolución de un problema en específico. Un perceptrón multicapa (MLP) es una clase de red neuronal artificial “feedforward” (ANN) (Hastie et al., 2009). Este tipo de redes neuronales contienen al menos tres capas de nodos: una capa de entrada, otra capa oculta y una capa de salida. En general, este tipo de redes neuronales añaden una cantidad n de capas ocultas para darle mayor poder a la red neuronal. Los algoritmos de redes neuronales están a la vanguardia de los métodos de aprendizaje automático, siendo los que obtienen la mayor parte de las veces los mejores resultados; sin embargo, su alto coste computacional supone una desventaja en algunas situaciones.

PyCaret predefine los parámetros de entrenamiento de las redes neuronales con arquitectura perceptrón multicapa como se muestra en la Figura 14.

```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(100,), learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=777, shuffle=True, solver='adam',
              tol=0.0001, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

Figura 14. Parámetros de entrenamiento de la red neuronal perceptrón multicapa

4.5 Metodologías de los experimentos

Las metodologías que se van a experimentar harán uso de los modelos de aprendizaje automático descritos en la sección anterior. El uso de la librería PyCaret ayudará a realizar todo el flujo de trabajo necesario para la modelización y evaluación de los distintos modelos. Además, el uso de la técnica de AutoML servirá para establecer las distintas metodologías que se detallarán a continuación:

1. Máquinas de Vectores de Soporte

La primera metodología sirve como punto base para comparar el rendimiento de las otras dos metodologías experimentales contrastando con uno de los modelos más usados en la literatura como lo son las Máquinas de Vectores de Soporte.

2. Mejor Modelo

En esta metodología se obtiene el modelo con el mejor rendimiento por cada ventana k de entrenamiento; es decir, cada vez que se vuelva a reentrenar los modelos (aplicando el flujo de trabajo automatizado) se escogerá al que mejor métrica de precisión presente a través de la evaluación cruzada. Es así como, en teoría, se espera aplicar el modelo que mejor se ajuste en función de las condiciones del mercado que se tengan en ese punto de tiempo. Como se mencionó en la revisión de la literatura, las series temporales financieras son muy volátiles y cambiantes, su estructura puede cambiar si se la analiza en distintas ventanas de tiempo; por tanto, un modelo con un alto rendimiento en una ventana de tiempo puede tener un bajo rendimiento en otra ventana de tiempo alejada de la anterior.

3. Conjunto de modelos

Finalmente, la última metodología consiste en aplicar técnicas de evaluación conjunta de modelos (ensemble models) para realizar las predicciones. Para lo cual, se escoge a los mejores 3 modelos en función de su precisión y se los consolida mediante una votación por mayoría simple. La idea es el pronóstico de cada punto depende de los resultados obtenidos por cada modelo, si dos de ellos pronosticaron que el siguiente periodo subirá el precio y el otro pronóstico que bajará, entonces por mayoría simple se dejará el pronóstico de la subida del precio. Este procedimiento puede realizar predicciones más consistentes debido a que se usan varios modelos para realizar cada una de las estimaciones. De la misma forma que la anterior metodología, en cada ventana k de entrenamiento se realizará este proceso. La precisión que va a ser evaluada es la obtenida a partir de la técnica de evaluación cruzada.

Los experimentos que se llevan a cabo tienen como objetivo probar el más amplio espectro de parámetros distintos que pueden llegar a afectar el rendimiento de cada una de las metodologías. Los parámetros mediante los cuales se establecerán los distintos experimentos serán la longitud de la ventana de entrenamiento de los modelos y el número de datos de entrenamiento. En función de los resultados obtenidos por Gerlein et al. (2016) de sus experimentos empíricos, se determina los siguientes valores para cada uno de los parámetros:

- Longitud de la ventana de entrenamiento: 5 y 15 periodos
- Número de datos de entrenamiento: 150, 750 y 1500 registros

Si se considera las combinaciones de ambos parámetros, se obtiene que existen seis posibles escenarios para probar las tres metodologías planteadas por cada par de divisa. Las tres metodologías son evaluadas por medio de las métricas de evaluación establecidas en la sección respectiva. Los resultados de cada métrica son presentados por cada uno de los pares de divisa y experimentos, y al final se realizará un resumen de los mejores resultados conseguidos. Así mismo, se presenta estadística descriptiva de los resultados para observar de manera más intuitiva cuál fue el rendimiento obtenido de las distintas metodologías, los modelos más empleados, entre otros.

5. Desarrollo de la comparativa

Los resultados de las tres metodologías planteadas serán divididos por cada mercado (o par de divisa) y experimento. En cada uno de ellos se detallará las métricas de evaluación y sus características más relevantes. Se puede ver el código completo implementado para el desarrollo de esta sección en el Anexo III. Módulo de Modelación y Backtesting.

5.1 Par de Divisa EUR/USD

A continuación, en la Tabla 6 se presentan los resultados de la aplicación de las diferentes metodologías para predecir un periodo posterior el movimiento del precio del par de divisa euro y dólar estadounidense. El color verde identifica donde se producen los mejores resultados de cada métrica de evaluación, mientras que, el color rojo indica los peores resultados.

Par de Divisa:			EUR/USD			
Longitud Ventana Entrenamiento	Número Datos Entrenamiento	Número Experimento	Métrica	SVM	Mejor Modelo	Conjunto de Modelos
5	150	1	Precisión	49,83%	57,45%	56,20%
			Precisión backtesting	48,60%	47,49%	47,77%
			Retorno acumulado	1,51%	-17,92%	-14,04%
	750	2	Precisión	49,96%	53,67%	53,29%
			Precisión backtesting	51,82%	50,07%	48,83%
			Retorno acumulado	14,43%	-7,12%	-11,27%
	1500	3	Precisión	50,02%	52,99%	52,72%
			Precisión backtesting	48,51%	49,52%	52,51%
			Retorno acumulado	-11,80%	4,66%	24,03%
15	150	4	Precisión	50,50%	57,38%	56,34%
			Precisión backtesting	49,06%	49,06%	48,41%
			Retorno acumulado	5,71%	-11,98%	-12,72%
	750	5	Precisión	50,11%	53,69%	53,43%
			Precisión backtesting	50,16%	50,21%	48,97%
			Retorno acumulado	3,18%	7,03%	-0,37%
	1500	6	Precisión	50,02%	53,01%	52,73%
			Precisión backtesting	48,64%	50,85%	54,30%
			Retorno acumulado	-14,80%	17,05%	39,23%

Tabla 6. Resultados de la aplicación de las distintas metodologías para la predicción de EUR/USD

En general, se observa que, los peores resultados de cada métrica de evaluación ocurren en el experimento 1, el cual corresponde a los valores mínimos de longitud de ventana de

entrenamiento y del tamaño del conjunto de datos. La precisión del modelo SVM es 49,83%, la cual es la más baja de todas las metodologías y experimentos realizados; sin embargo, como se puede observar su retorno acumulado es de 1,51%. A pesar de que la metodología de escoger el mejor modelo por cada periodo de entrenamiento presenta la mejor precisión con un 57,47% en el experimento 1, la misma tiene el peor retorno acumulado (-17,92%).

La metodología del conjunto de modelos presenta los mejores resultados de precisión de backtesting y retorno acumulado con un 54,30% y 39,23%, respectivamente, en el experimento 6. En contraste del experimento 1, el experimento 6 tiene los valores máximos de longitud de ventana de entrenamiento y tamaño del conjunto de datos. No se observa una relación clara entre la precisión y el retorno acumulado, los resultados de estas dos métricas son contradictorios para las metodologías en la mayoría de los experimentos. Por otro lado, si se observa una clara relación entre la precisión de backtesting y el retorno acumulado.

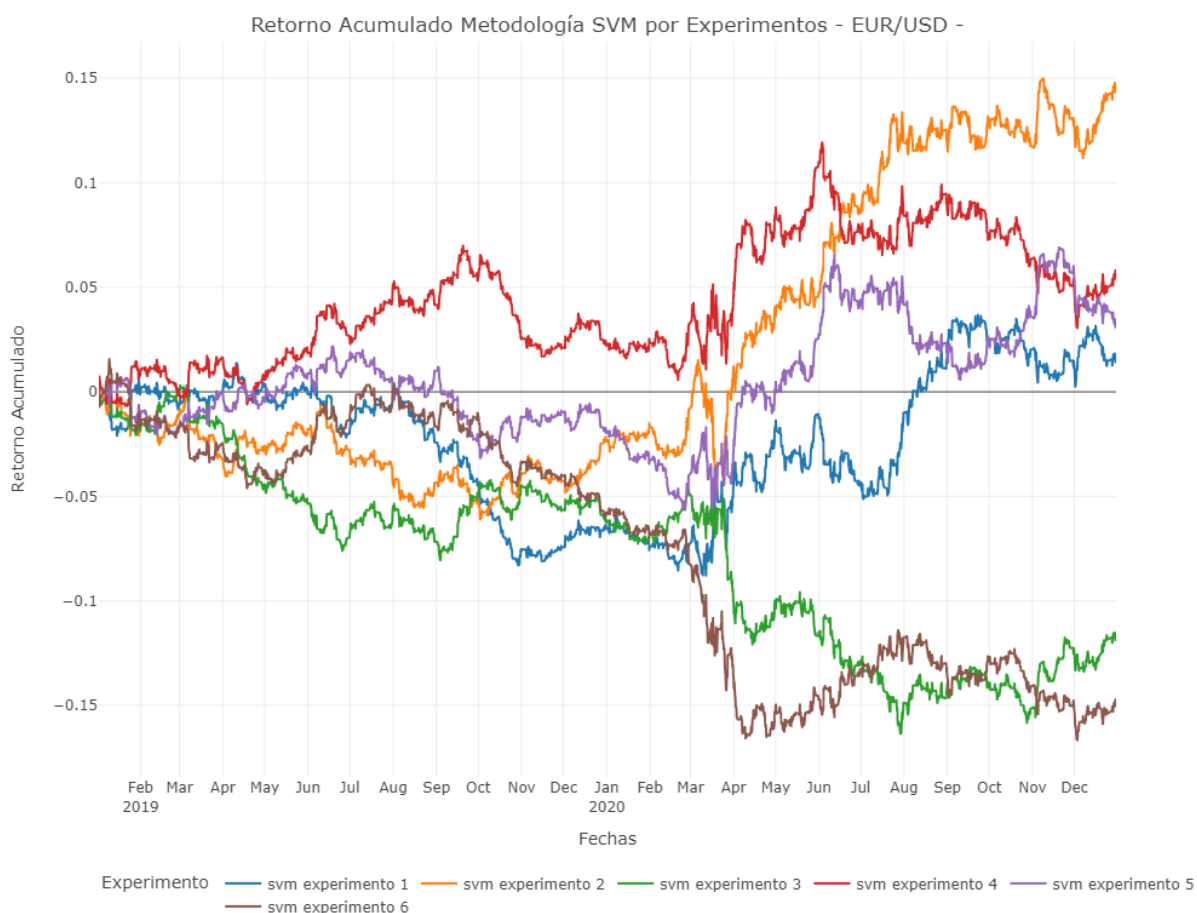


Figura 15. Retorno acumulado de la metodología SVM por cada experimento para EUR/USD

En la Figura 15 se puede muestra el comportamiento del retorno acumulado para la metodología de SVM por cada experimento. La tendencia de las series temporales presenta un cambio estructural durante el inicio de la pandemia del covid-19 (entre los meses de marzo y abril de 2020). Antes del inicio de la pandemia, el modelo SVM, entrenado cada 15 periodos y con un tamaño de datos de entrenamiento de 150 registros, mostraba el mejor y más consistente retorno acumulado. Después del cambio estructural, el modelo SVM que tuvo mejor desempeño fue el entrenado bajo una ventana de 5 periodos y con un tamaño de datos de 750 registros.

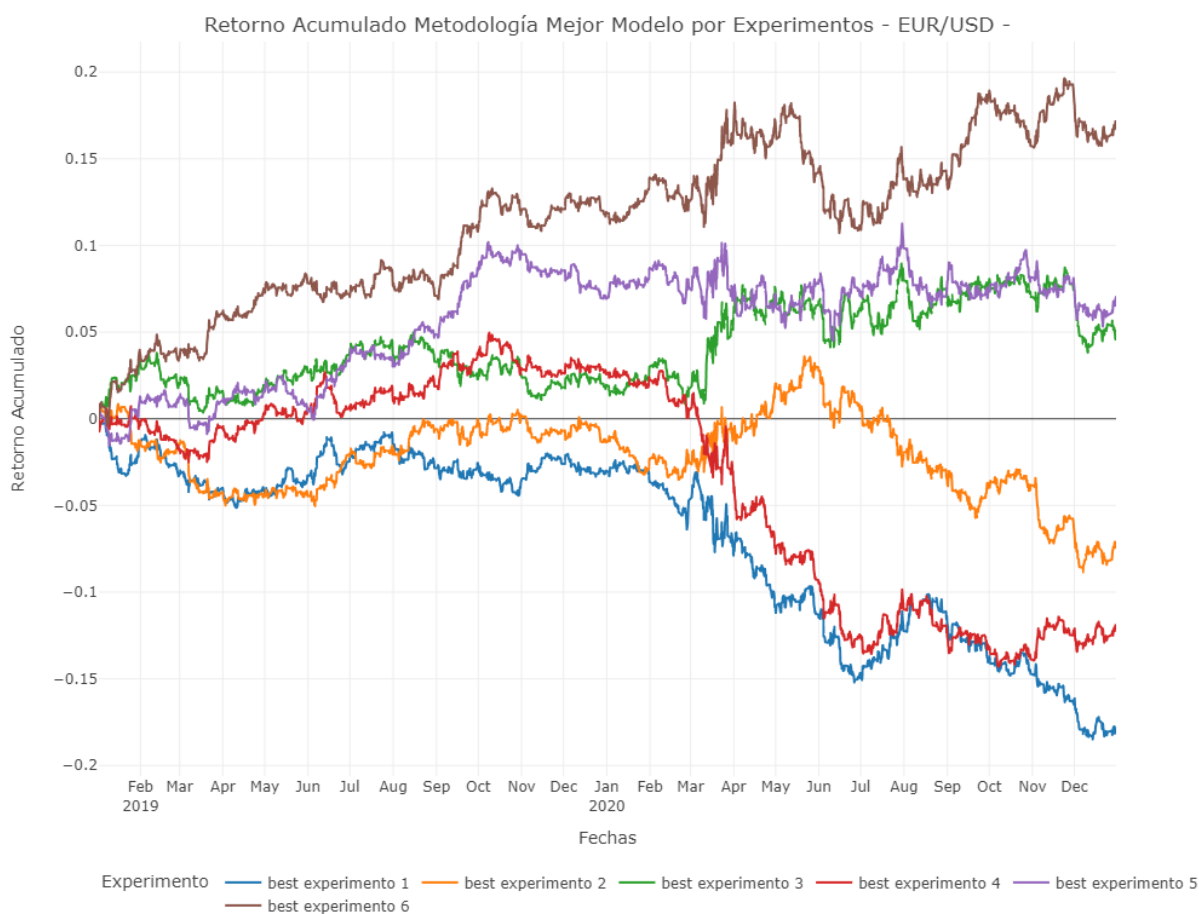


Figura 16. Retorno acumulado de la metodología del mejor modelo por cada experimento para EUR/USD

El retorno acumulado de la metodología de escoger el mejor modelo muestra un comportamiento más estable en la Figura 16. Los experimentos que tuvieron el mejor retorno fueron los que contaron con un mayor tamaño del conjunto de entrenamiento, a excepción del experimento 2. Además, el cambio estructural debido a la pandemia no tuvo un impacto tan fuerte en los retornos acumulados, a diferencia de la metodología de SVM.

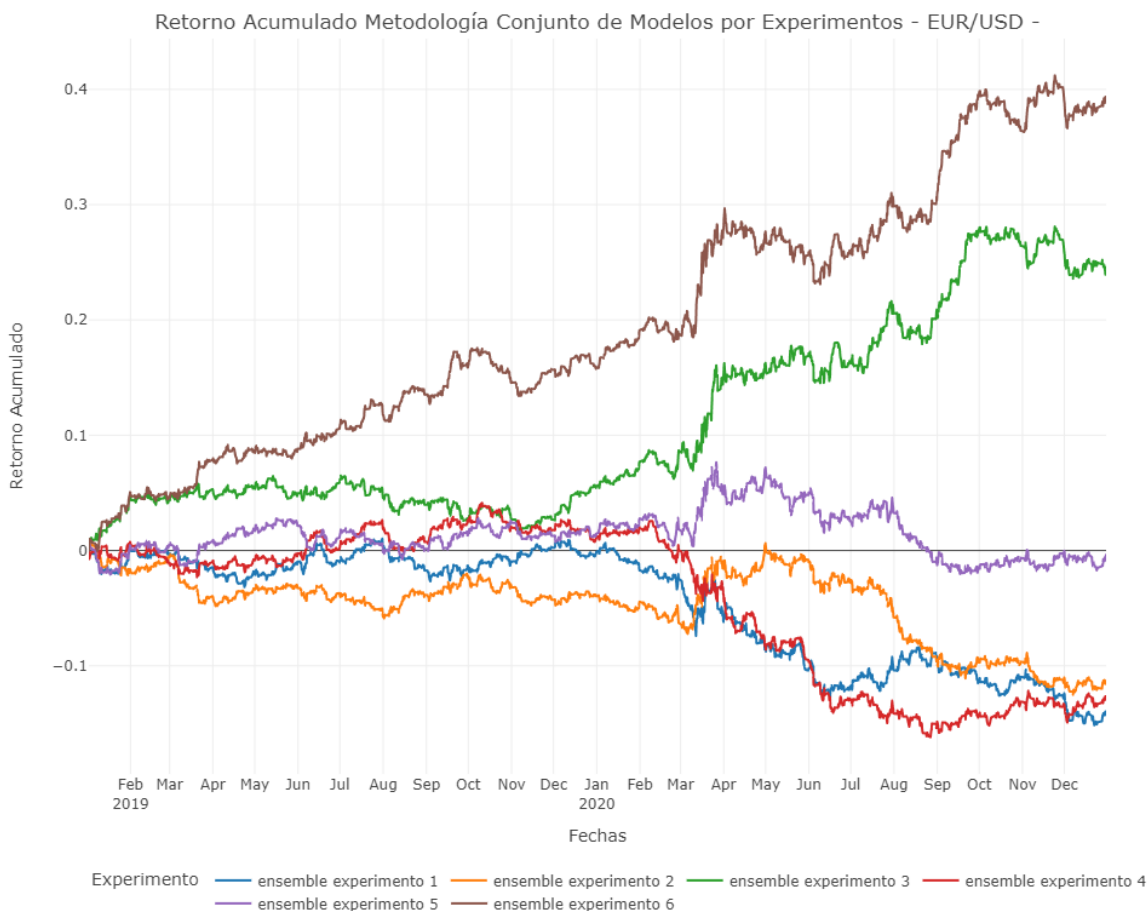


Figura 17. Retorno acumulado de la metodología del conjunto de modelos por cada experimento para EUR/USD

La metodología del conjunto de modelos muestra los mejores retornos acumulados llegando hasta aproximadamente un 40% en el experimento 6. Como se muestra en la Figura 17, esta metodología presenta retornos acumulados positivos al finalizar el periodo de backtesting para los experimentos 3 y 6, los cuales son los que tienen el mayor conjunto de datos de entrenamiento. Así mismo, en la mayoría de ellos se logra ajustar correctamente el cambio estructural causado por la pandemia, debido a que no se observa un gran impacto en su tendencia después del inicio de la pandemia.

5.2 Par de Divisa USD/JPY

Los resultados de la aplicación de las metodologías para la predicción de la subida o bajada del precio del tipo de cambio dólar estadounidense y yen se detallan en la

Tabla 7.

Par de Divisa:			USD/JPY			
Longitud Ventana Entrenamiento	Número Datos Entrenamiento	Número Experimento	Métrica	SVM	Mejor Modelo	Conjunto de Modelos
5	150	1	Precisión	49,87%	58,23%	56,93%
			Precisión backtesting	50,14%	50,37%	50,23%
			Retorno acumulado	1,33%	-11,03%	4,16%
	750	2	Precisión	49,98%	54,23%	53,90%
			Precisión backtesting	47,74%	50,60%	51,43%
			Retorno acumulado	-6,49%	0,42%	5,14%
	1500	3	Precisión	50,07%	53,97%	53,95%
			Precisión backtesting	50,88%	52,95%	53,09%
			Retorno acumulado	6,64%	28,10%	38,70%
15	150	4	Precisión	49,72%	58,12%	56,91%
			Precisión backtesting	49,52%	50,94%	51,17%
			Retorno acumulado	2,80%	6,41%	11,19%
	750	5	Precisión	50,06%	54,23%	53,96%
			Precisión backtesting	48,46%	52,05%	53,84%
			Retorno acumulado	-11,14%	9,51%	28,93%
	1500	6	Precisión	50,10%	53,95%	53,92%
			Precisión backtesting	50,76%	54,25%	53,29%
			Retorno acumulado	12,82%	39,92%	24,64%

Tabla 7. Resultados de la aplicación de las distintas metodologías para la predicción de USD/JPY

El modelo SVM muestra las peores métricas de evaluación en los experimentos 2, 4 y 5. Mientras que, la metodología de elegir el mejor modelo presenta las mejores métricas de evaluación. A pesar de que esta metodología logra una precisión de 58,23% en el experimento 1, su retorno acumulado para el mismo es de -11,03%. La precisión de backtesting y el retorno acumulado llegan al máximo en el experimento 6, con un conjunto grande de datos de entrenamiento y la mayor ventana de entrenamiento.

En este mercado, las métricas con las que se evalúan cada una de las metodologías presentan mejores resultados a comparación del mercado euro y dólar estadounidense. Solo se tienen tres situaciones en las que existe un retorno acumulado negativo (Mejor Modelo – experimento 1, SVM – experimento 2 y SVM – experimento 5). Además, tanto la métrica de precisión como la de precisión de backtesting, en general, muestran rendimientos superiores al 50%.

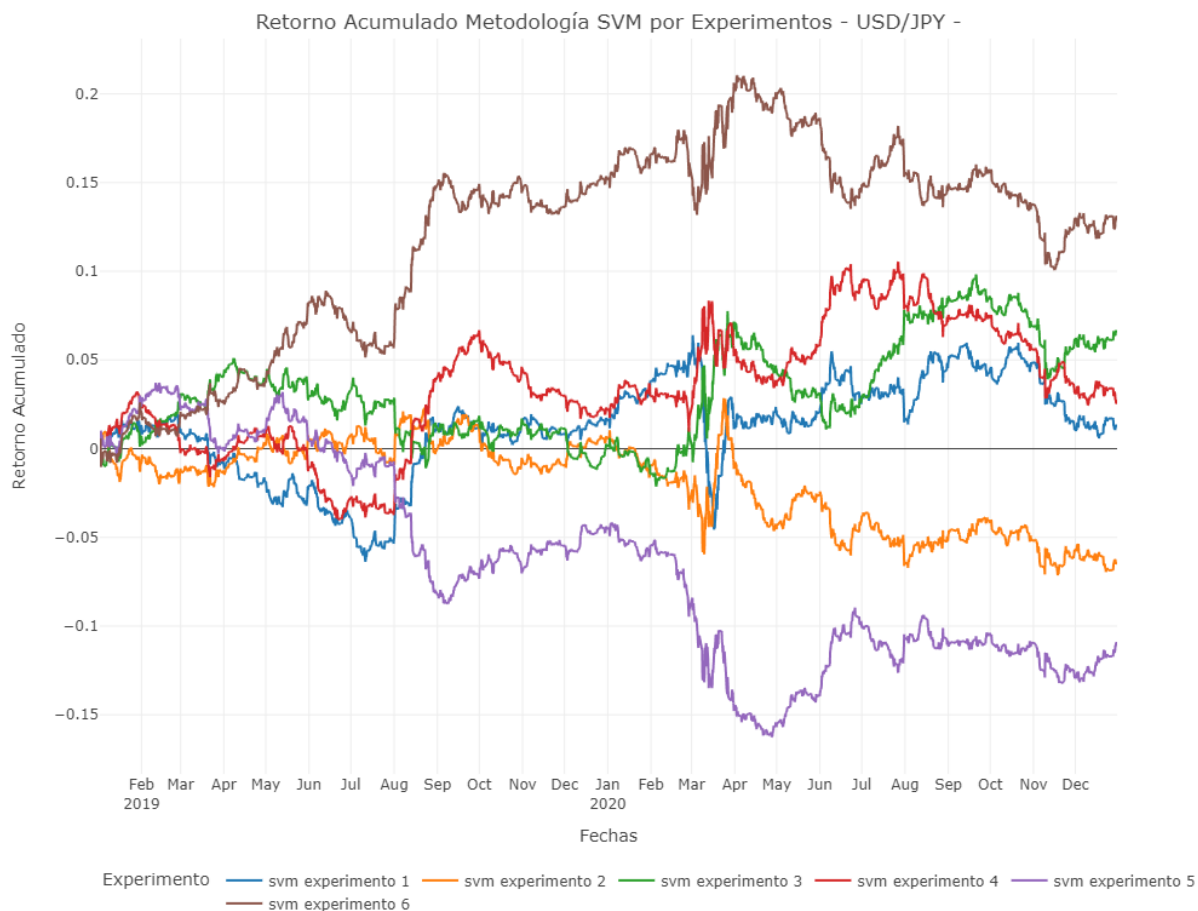


Figura 18. Retorno acumulado de la metodología SVM por cada experimento para USD/JPY

En la Figura 18 se observa el comportamiento del retorno acumulado para la metodología SVM por cada experimento realizado. Los experimentos 2 y 5, con un conjunto de entrenamiento de 750 datos, tienen el peor rendimiento financiero. Por otro lado, el experimento 6 presenta los retornos acumulados más altos para el modelo SVM. Nuevamente, se puede observar como el cambio estructural y volatilidad durante el inicio de la pandemia afecta a las series temporales.

La aplicación del mejor modelo solo presenta un experimento con retorno acumulado negativo, el resto de los experimentos muestran rendimientos financieros positivos al finalizar el periodo de backtesting. Los experimentos con los conjuntos de datos más grandes de entrenamiento son los que tienen el mayor y más estable retorno acumulado, como se muestra en la Figura 19. Durante los meses de marzo y abril de 2020 se observa mayor volatilidad

debido al inicio de la pandemia; sin embargo, en la mayoría de los experimentos se logra ajustar el efecto y se tiene un mayor retorno acumulado después de este periodo.

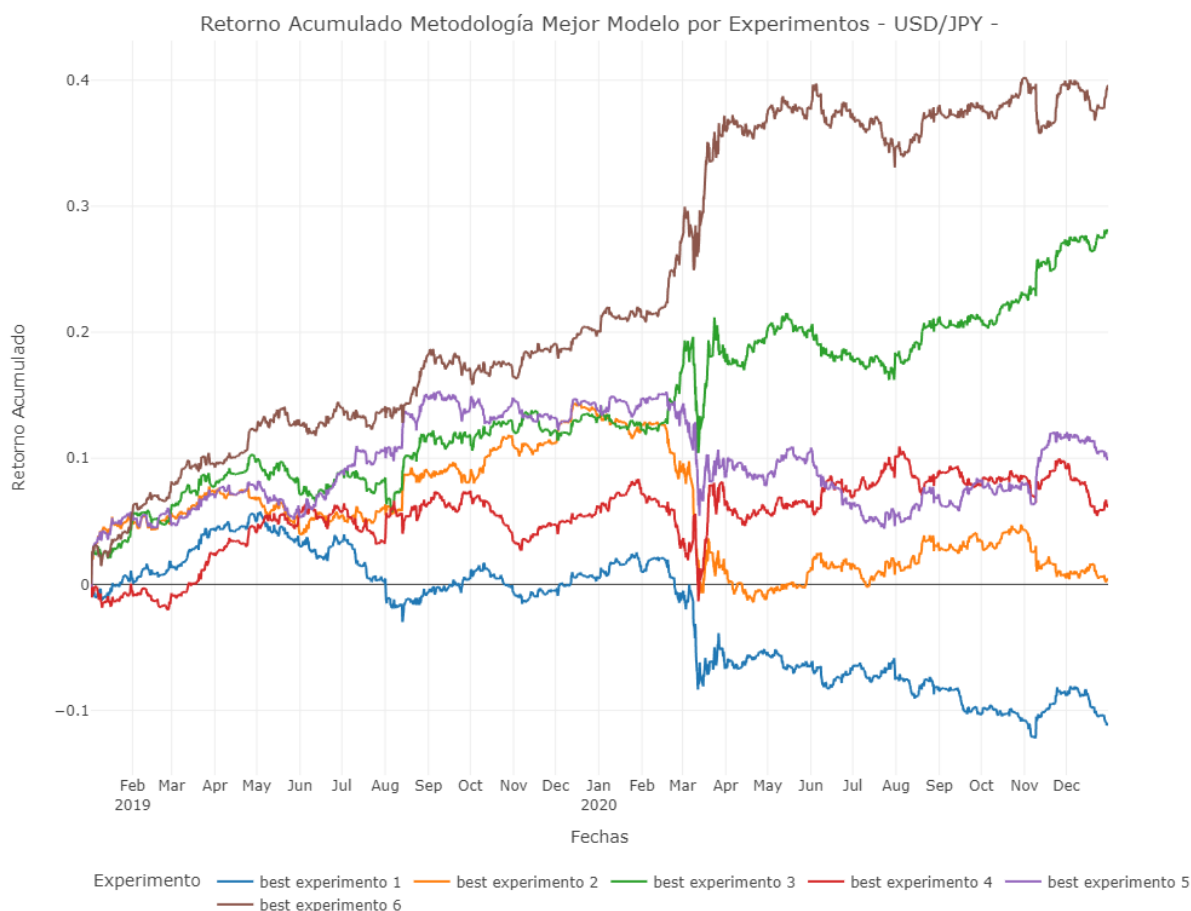


Figura 19. Retorno acumulado de la metodología del mejor modelo por cada experimento para USD/JPY

La Figura 20 señala que la metodología del conjunto de modelos presenta los mejores resultados financieros, a comparación de las otras dos metodologías. Aunque esta metodología no cuente con el valor máximo de retorno por a penas una diferencia de un 1,22% con la metodología de mejor modelo, presenta retornos acumulados positivos para cada uno de los experimentos. El ajuste de la metodología al inicio de la pandemia es bueno en la mayoría de los escenarios. El rendimiento financiero de esta metodología esta entre un rango del 4,16% al 38,70%. De la misma forma que las anteriores metodologías, los mejores resultados se obtienen con conjuntos de datos de entrenamiento grandes.

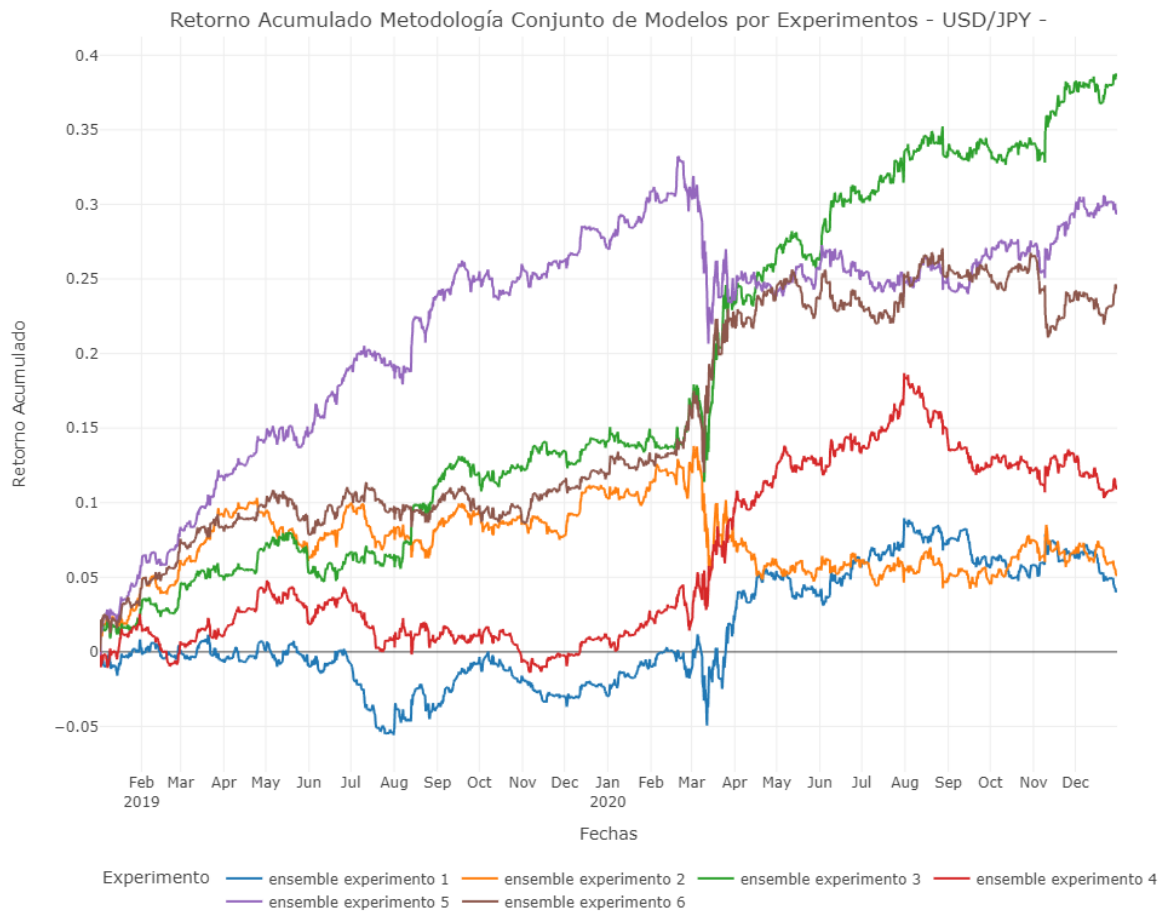


Figura 20. Retorno acumulado de la metodología del conjunto de modelos por cada experimento para USD/JPY

5.3 Par de Divisa AUD/JPY

Los resultados de la aplicación de las metodologías para la predicción del movimiento del precio en el mercado de divisas del dólar australiano y yen se observan en la Tabla 8.

Par de Divisa:			AUD/JPY			
Longitud Ventana Entrenamiento	Número Datos Entrenamiento	Número Experimento	Métrica	SVM	Mejor Modelo	Conjunto de Modelos
5	150	1	Precisión	50,00%	58,00%	56,47%
			Precisión backtesting	51,95%	50,71%	50,71%
			Retorno acumulado	49,31%	-9,08%	11,35%
	750	2	Precisión	49,98%	53,54%	53,16%
			Precisión backtesting	48,97%	52,28%	51,49%
			Retorno acumulado	-2,61%	13,53%	8,92%
	1500	3	Precisión	49,79%	52,76%	52,62%
			Precisión backtesting	49,24%	49,47%	50,67%
			Retorno acumulado	11,00%	11,76%	42,00%
15	150	4	Precisión	50,47%	58,27%	56,78%
			Precisión backtesting	52,18%	50,02%	50,62%
			Retorno acumulado	26,33%	2,01%	7,89%
	750	5	Precisión	49,75%	53,60%	53,20%
			Precisión backtesting	47,63%	52,05%	52,78%
			Retorno acumulado	-26,04%	13,14%	46,86%
	1500	6	Precisión	49,83%	52,79%	52,72%
			Precisión backtesting	49,98%	50,39%	49,93%
			Retorno acumulado	-2,10%	12,99%	11,87%

Tabla 8. Resultados de la aplicación de las distintas metodologías para la predicción de AUD/JPY

La aplicación del modelo SVM presenta los peores resultados para todas las métricas en el experimento 5. Sin embargo, en esta ocasión el modelo SVM tiene el máximo valor de retorno acumulado de todos los experimentos analizados en todos los mercados con un 49,31% en el experimento 1. Por otro lado, las metodologías de escoger el mejor modelo y utilizar un conjunto de modelos se reparten la mejor precisión y precisión backtesting, respectivamente. Cabe destacar que la metodología del conjunto de modelos tiene retornos acumulados por encima del 40% en dos experimentos.

En este mercado se tienen 4 situaciones con rendimientos financieros negativos, donde 3 de ellas corresponden a la aplicación del modelo SVM. El retorno acumulado de la metodología de SVM en el experimento 5 de -26,04% es el peor de todos los escenarios incluyendo todos los mercados. También, se observa que una alta precisión no implica un alto retorno acumulado, mientras que, una alta precisión de backtesting si se relaciona con un mejor rendimiento financiero.

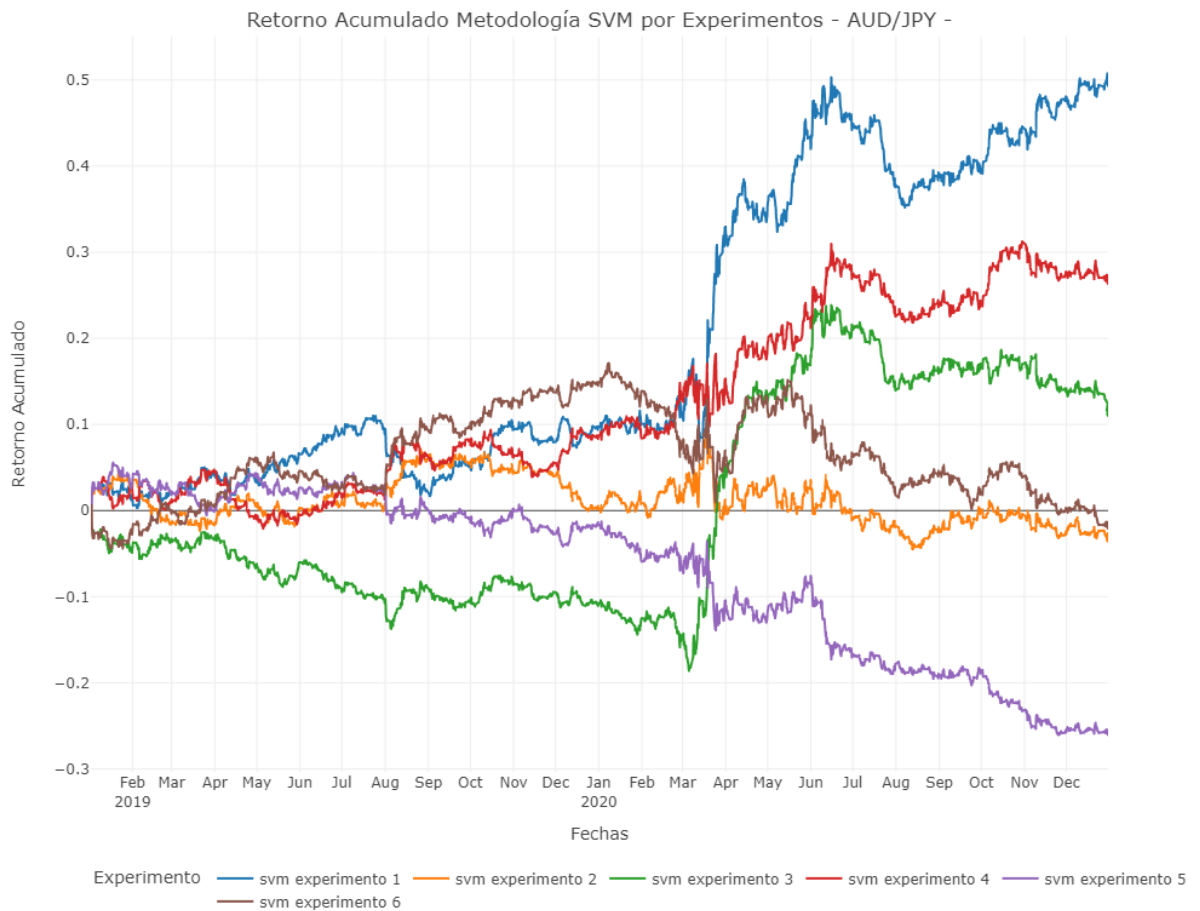


Figura 21. Retorno acumulado de la metodología SVM por cada experimento para AUD/JPY

En la Figura 21, se observa como los mejores retornos acumulados se obtienen en los experimentos con el menor número de datos para el conjunto de entrenamiento. Los experimentos 1 y 3 muestran una particularidad durante el cambio estructural producido por la pandemia. En ambos experimentos, el modelo SVM aprovecha el inicio de la pandemia para incrementar casi de manera exponencial su retorno acumulado. La tendencia de estos dos experimentos cambia bastante, de tal forma que en el experimento 5 en unos pocos meses se pasa de tener un retorno acumulado de aproximadamente -20% a estar por encima del 20% en su punto máximo.

La metodología de escoger el mejor modelo tiene un comportamiento diferente al SVM. La Figura 22 señala que esta metodología tiene el peor rendimiento financiero en el experimento 1, en contraste de la metodología del SVM donde el mejor retorno financiero se obtenía a partir de este mismo experimento. Los demás experimentos muestran retornos positivos entre

un rango de 2,01% a 13,53%. El periodo de inicio de pandemia causa bastante volatilidad en el retorno acumulado para esta metodología, se puede observar los picos que existen entre marzo y mayo de 2020.

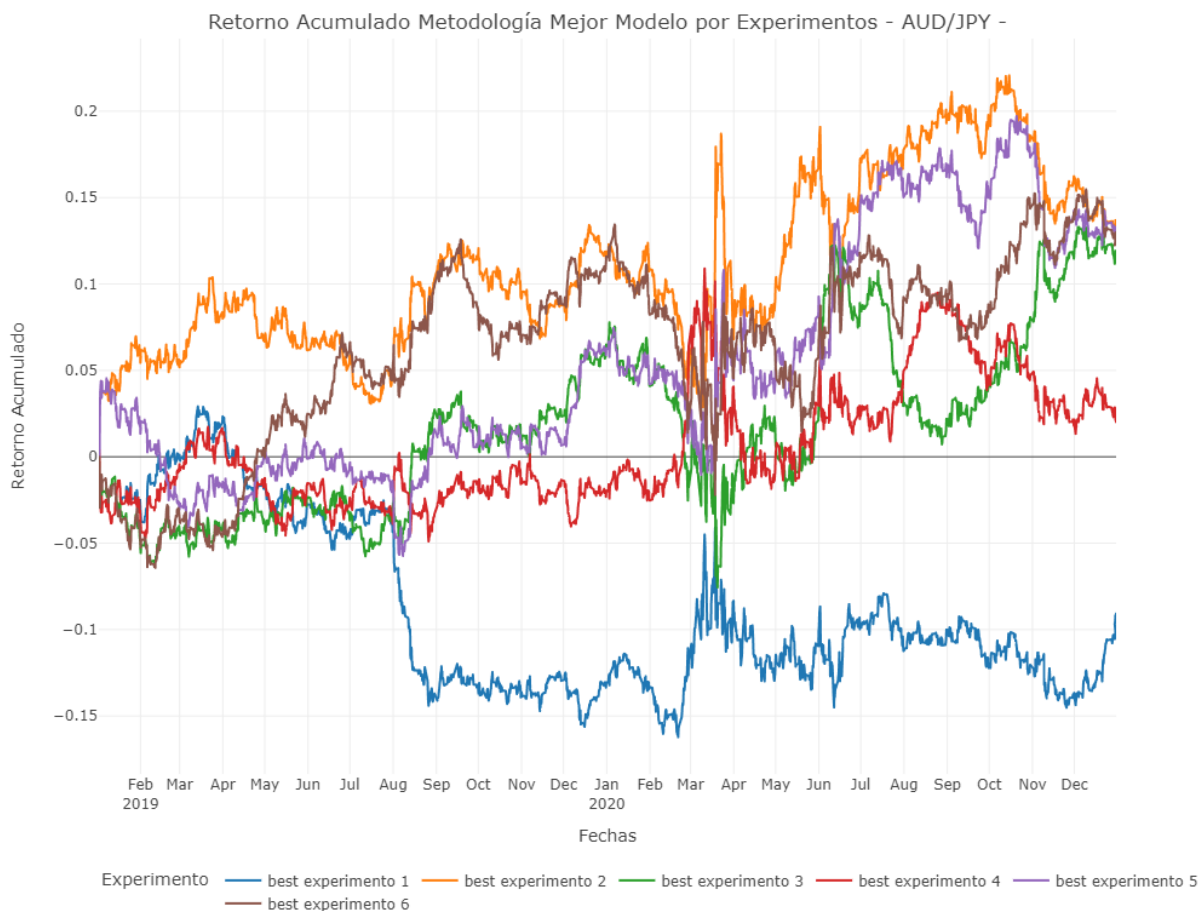


Figura 22. Retorno acumulado de la metodología del mejor modelo por cada experimento para AUD/JPY

Finalmente, la metodología del utilizar un conjunto de modelos para la predicción vuelve a dar los mejores resultados de rendimiento financiero. La metodología obtuvo un retorno acumulado positivo en todos los experimentos. Aunque no obtuvo el retorno más alto en este mercado, en los experimentos 3 y 6 se estuvo cerca de llegar a un retorno acumulado de un 50%. En la Figura 23, se puede observar como el retorno de los modelos con un mayor conjunto de datos de entrenamiento evoluciona positivamente después del periodo de pandemia. Aunque en ambos casos ya se mostraba una tendencia creciente del retorno.

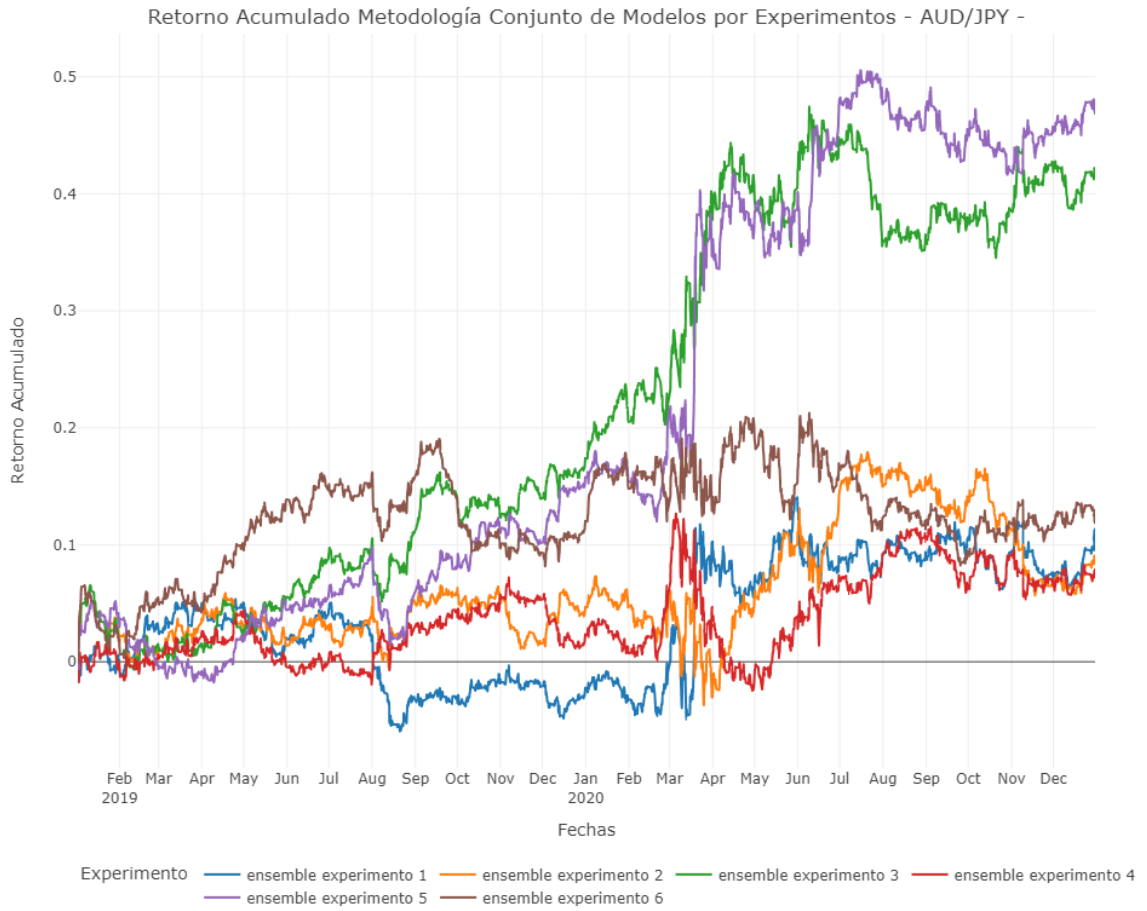


Figura 23. Retorno acumulado de la metodología del conjunto de modelos por cada experimento para AUD/JPY

6. Discusión y análisis de resultados

Los resultados sugieren que las metodologías que trabajan con varios modelos para realizar la predicción tienen una capacidad para generar mejores predicciones que el modelo SVM. En general, se observó que el modelo SVM presentaba las peores métricas de precisión, precisión backtesting y retorno acumulado. Además, el uso de solo este modelo hacía que se magnifique el efecto de la pandemia sobre el retorno acumulado. Las otras dos metodologías por su diseño podían mitigar el cambio estructural y volatilidad durante el inicio de la pandemia.

A pesar de que las métricas de precisión mostraron resultados bajos, estos concuerdan con lo reportado y visto en el estado del arte. Las series temporales financieras son difíciles de predecir, y aún más las correspondientes a los mercados Forex donde existe tanta volatilidad e inestabilidad. Así mismo, el retorno acumulado para un periodo de dos años concuerda con lo reportado por los diversos autores que han contribuido a la literatura (Barbosa & Belo, 2008b; Gerlein et al., 2016; J. Li & Tsang, 1999; Tenti, 1996).

Los experimentos mostraron que la capacidad predictiva de los modelos es mejor con un conjunto grande de datos de entrenamiento. El parámetro de la longitud de la ventana en la que se volvían a reentrenar los modelos no tuvo un mayor impacto sobre los resultados. Esto podría sugerir que el comportamiento del mercado de divisas no es tan cambiante como para establecer periodos de reentrenamiento tan pequeños; por tanto, se podría mantener válido un mismo modelo durante un mayor tiempo y ahorrar coste computacional. Los parámetros de longitud de ventana de entrenamiento y el número del tamaño de entrenamiento sirvieron para validar la consistencia de las distintas metodologías sobre distintos escenarios, donde se pueden resaltar varios puntos.

Primero, la aplicación de las metodologías y su rendimiento depende bastante de las condiciones específicas del mercado en que se encuentre. Por ejemplo, los retornos acumulados máximos y mínimos ocurrieron en el mercado de divisas del dólar australiano y el yen. Este es un mercado bastante volátil y exótico para los inversores debido a los grandes cambios de precios que tiene. La volatilidad en los precios ayuda a que los resultados se magnifiquen, sea para bien o para mal. Por otro lado, el mercado de divisas dólar estadounidense y yen es mucho más estable gracias a la fortaleza de los bancos centrales que respaldan estas monedas (Estados Unidos y Japón). En este mercado se obtuvo el rendimiento financiero más consistente, dado que solo hubo tres escenarios donde se obtuvo un retorno acumulado negativo.

La relación entre las métricas de precisión, precisión backtesting y retorno acumulado también es importante. En general, se observa que la precisión no tiene una relación clara con la precisión backtesting ni con el retorno acumulado. Esto sugiere que la precisión de validación cruzada no es un buen indicador para medir el rendimiento de la metodología aplicada al mundo real. Esta es una clara señal de la dificultad que tiene predecir series temporales financieras, debido a que no te puedes guiar por métricas de evaluación dadas por el conjunto de entrenamiento. En cambio, la precisión de backtesting sí se relaciona directamente con el retorno acumulado. Mientras mayor sea este indicador, más alto será el retorno acumulado. En los escenarios que se presentaban las tres métricas de evaluación altas, los resultados eran mucho más consistentes y el retorno acumulado presentaba una tendencia creciente sostenida sin mayor alteración.

Los cambios estructurales suelen ser un serio problema para los modelos de aprendizaje automático. En este caso, no hubo excepción dado que se podía observar como durante el inicio de la pandemia el retorno acumulado era mucho más volátil. Sin embargo, las metodologías donde se trabajaban con varios modelos, escogiendo el mejor de ellos o realizando la predicción en conjunto, mitigaron más el efecto de la pandemia a comparación del modelo SVM. Esto podría sugerir que los modelos seleccionados durante este periodo por las dos distintas metodologías tenían una mayor capacidad de adaptación al cambio estructural que el SVM. Sin embargo, cabe destacar que el mayor retorno acumulado se lo llevo el modelo SVM con una cantidad pequeña de datos de entrenamiento. El rendimiento financiero que tuvo durante el periodo de pandemia en el par de divisas AUD/JPY fue excelente. Lo que indica que existe un modelo específico correcto para el momento adecuado.

Finalmente, los resultados muestran patrones que pueden ayudar para escoger la mejor metodología con la cual trabajar. Por ejemplo, como se comentaba, el uso del conjunto de modelos para la predicción tuvo como resultado la menor cantidad de retornos acumulados negativos y métricas de evaluación bastante consistentes entre los distintos mercados. Además, esta metodología muestra mejores resultados cuando se utilizan conjuntos de entrenamiento grandes. Si filtramos los resultados de la metodología del conjunto de modelos con un número de datos de entrenamiento de 1500, se obtiene un retorno acumulado promedio de 30,08% para los dos años del periodo de backtesting, lo cual está por encima de las demás metodologías.

6.1. Limitaciones

Los resultados que arrojaron los distintos experimentos sugieren que existe la manera de obtener un retorno acumulado estable durante cierto periodo de tiempo. La ventana de tiempo analizada incluye periodos de relativa estabilidad e inestabilidad en los mercados financieros. Un punto claro para identificar estos dos periodos mencionados es el inicio de las restricciones de movilidad que se dieron a partir de la pandemia de covid-19, alrededor de marzo de 2020. Antes de esta fecha, el mundo y los mercados financieros tuvieron un periodo sin mucha volatilidad; mientras que, los meses posteriores a marzo de 2020 estuvieron caracterizados por una mayor agitación de los mercados financieros. La inclusión de ambos periodos les da consistencia y validez a los resultados experimentales; sin embargo, la volatilidad de los mercados financieros debido a la pandemia no se acerca a la volatilidad vivida en la crisis financiera de 2008. Autores como Gerlein et al. (2016) reportan un impacto negativo fuerte debido a la crisis financiera que mermaron el rendimiento financiero obtenido. Por tanto, los resultados experimentales podrían obtener un rendimiento diferente en otra ventana temporal con distintas y más adversas condiciones.

La forma en la que está construida la experimentación puede estar asociada al sesgo por dragado de datos, búsqueda de significado o p-hacking. Este problema se relaciona al mal uso del análisis de datos para encontrar patrones en los datos que puedan ser presentados como estadísticamente significativos. Por ejemplo, si se quiere encontrar una relación entre dos variables, se puede realizar un número x de experimentos con distintos parámetros y combinaciones hasta que uno de ellos te de los resultados que esperabas encontrar. Estos resultados, en general, se encuentran sesgados debido a que están atados a circunstancias muy particulares para que sean válidos. Dicho esto, la experimentación realizada tiene ciertas características particulares para minimizar los efectos de este sesgo, las cuales son:

- a) Los resultados de todos los experimentos son reportados, sin importar si son relativamente buenos o malos.
- b) Las pruebas de combinaciones de diferentes parámetros tienen como objetivo de darle consistencia y validez a los resultados. No se analizan los resultados de manera individual, más bien se trata de analizar todos los experimentos en conjunto para determinar patrones o escenarios donde existan mejores condiciones y rendimiento de las metodologías.
- c) Se utilizan varias métricas de rendimiento para evaluar a las distintas metodologías, cada una de ellas con un enfoque diferente con lo cual se desea obtener una visión más integral.

- d) Los experimentos se desarrollan en distintos mercados Forex para medir la sensibilidad de las metodologías bajo distintas condiciones.

6.2. Comparación con el estado del arte

El principal objetivo de este estudio es evaluar y comparar las capacidades de las distintas metodologías de aprendizaje automático para realizar trading, de tal forma para generar evidencia empírica en contra de la hipótesis de mercado eficientes. En este punto, el análisis y comparación de los resultados obtenidos contra los reportados en el estado del arte también es relevante y aporta valor al trabajo. El siguiente análisis debe ser tomado con precaución debido a que los distintos estudios para la predicción de series temporales se han desarrollado bajo diferentes mercados financieros, ventanas de tiempo, tipos de instrumentos financieros y estrategias de trading; por tanto, la mayoría de las métricas de rendimiento o error no son directamente comparables entre sí. Esta subsección tiene como fin comparar e ilustrar la capacidad de predicción de las metodologías de aprendizaje automático en aplicaciones de trading.

La comparación se enfoca en las metodologías evaluadas en el estudio y que aparecen de igual forma en la revisión de la literatura. Kim (2003) aplica un modelo SVM para la predicción del movimiento del índice bursátil KOSPI. El experimento obtiene una precisión del 57,83%, pero no reporta el retorno financiero del modelo. Así mismo, Chen et al. (2006) realiza la predicción de los precios de acciones en el mercado de valores con el algoritmo SVM. Los resultados reportan un error de 57,23%. La mejor precisión obtenida a partir de la aplicación del modelo SVM en el presente estudio es de 50,50%. Si bien está un poco por debajo del rendimiento obtenido en los otros dos trabajos, en general, los resultados se encuentran acordes a las situaciones particulares del experimento (e.g., implementación en el mercado Forex, periodo de tiempo distinto, entre otros)

Barbosa & Belo (2008a) son los únicos autores que prueban técnicas de “ensemble” de modelos. Su estudio tiene un enfoque muy similar al presente trabajo, dado que trata de predecir el movimiento de precios del mercado Forex mediante un problema de clasificación. Los resultados muestran una precisión de 52,47% y un retorno acumulado de 66,67% para un periodo de 2 años. En contraste, los mejores resultados reportados en el trabajo muestran una precisión de 56,93% y un retorno acumulado de 46,86%. Aquí es interesante ver que, a pesar de contar con una mayor precisión, no se obtiene un mayor retorno acumulado. El principal motivo es la estrategia de trading con la cual se simulan las operaciones de compraventa, lo cual tiene bastante injerencia sobre los resultados.

Por otro lado, si se analiza el rendimiento general de los distintos modelos de aprendizaje automático que se han revisado en el estado del arte, se encuentra que los resultados experimentales obtenidos son bastante consistentes con estos. En la mayoría de los estudios, la aplicación de uno u otro algoritmo de aprendizaje automático, por más simple o complejo que sea, logra obtener relativamente el mismo rendimiento y retorno. Esto podría sugerir que no existe determinado algoritmo que sea mejor para la predicción en los mercados financieros. Así mismo, por este motivo las metodologías que toman en consideración un conjunto de modelos salen beneficiadas, ya que el uso de distintos modelos, a priori, aporta consistencia y estabilidad a la predicción.

7. Conclusiones y trabajo futuro

7.1. Conclusiones

En la presente investigación se realiza una comparación de distintas metodologías para predecir el comportamiento del alza o baja del precio de varios pares de divisas del mercado Forex. Esto con el motivo de generar evidencia empírica en contra de la hipótesis de mercados eficientes, la cual señala que no es posible realizar predicciones consistentes a un medio o largo plazo de los mercados financieros y, por ende, generar rentabilidad de ellos. El estudio se enfoca en analizar distintas formas de aplicar modelos de aprendizaje automático para resolver el problema de pronosticar las variaciones de los precios. Las variables que tienen a su disposición los modelos son un conjunto de indicadores técnicos que sirven como referencia para analizar el mercado.

En general, se observa que, las tres metodologías comparadas tuvieron métricas de evaluación acorde a lo que está en el estado del arte. Sin embargo, cada una de ellas, bajo determinadas condiciones, tuvieron la capacidad de generar un rendimiento financiero positivo significativo. Si se observa un top 10 de los retornos acumulados de las distintas metodologías por los diferentes escenarios, este supera ampliamente al retorno que se podría obtener por el uso del sistema financiero tradicional (i.e, alrededor del 10% de interés como máximo). Por tanto, los resultados de este trabajo contribuyen a reforzar al aprendizaje automático como una herramienta para predicción de series temporales financieras, generando evidencia empírica en contra de la hipótesis de mercados eficientes.

La experimentación se realizó sobre distintos escenarios y tipos de mercado, donde se pudieron encontrar combinaciones consistentes y con baja sensibilidad al cambio de las condiciones del mercado a las que se las expusieron. Esto valida los resultados obtenidos, debido a que sistemáticamente se encontraron parámetros con los cuales se puede tener un mayor retorno financiero. Cabe recalcar que las metodologías no fueron probadas sobre todas las condiciones posibles, por lo cual, los resultados de su aplicación en otra ventana de tiempo con condiciones adversas pueden ser variables.

El uso de herramientas de aprendizaje automático automatizado sirve para darle un mayor poder a las técnicas convencionales de aprendizaje automático. Hasta lo que se conoce, no se han realizado estudios de la aplicación de estos métodos para la predicción de series temporales financieras. En mercados tan volátiles, la automatización del flujo de trabajo de ajuste, comparación, elección y optimización de modelos de aprendizaje automático puede

ser clave para generar mejores y más consistentes resultados, como se evidencia en el trabajo.

7.2. Líneas de trabajo futuro

Para futuras investigaciones se podría extender el estudio considerando la inclusión de variables fundamentales, que indiquen la situación económica, social y política relacionada con el mercado analizado. La ventana de tiempo sobre la cual se realiza la simulación se la puede extender para que cubra muchos más años y se pueda validar los resultados sobre distintas condiciones de mercado. Por otro lado, se podría realizar una optimización de hiperparámetros de los modelos ajustados con las técnicas de AutoML. En teoría, esto les debería de dar una mayor capacidad predictiva a los modelos, lo que resultaría en un incremento del retorno financiero. Finalmente, el parámetro clave con el cuál plantear un mayor número de experimentos o escenarios sería el tamaño del conjunto de entrenamiento. Se recomienda probar escenarios de manera iterativa hasta encontrar el número de datos óptimo para el conjunto de entrenamiento.

8. Bibliografía

- Appel, G. (2003). Become your own technical analyst: How to identify significant market turning points using the moving average convergence-divergence indicator or macd. *The Journal of Wealth Management*, 6(1), 27–36.
- Barbona, I., & Beltrán, C. (2018). *Aplicación del algoritmo Boosting Adaptativo (ADABOOST) a un problema de clasificación automática de textos*.
- Barbosa, R. P., & Belo, O. (2008a). Algorithmic Trading Using Intelligent Agents. *IC-AI*, 136–142.
- Barbosa, R. P., & Belo, O. (2008b). Autonomous forex trading agents. *Industrial Conference on Data Mining*, 389–403.
- Barbosa, R. P., & Belo, O. (2010). The agent-based hedge fund. *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2, 449–452.
- Becket, M. (2004). *How the Stock Market Works: A Beginner's Guide to Investment*. Kogan Page Publishers.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Cai, X., Hu, S., & Lin, X. (2012). Feature extraction using restricted Boltzmann machine for stock price prediction. *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, 3, 80–83.
- Carapuço, J., Neves, R., & Horta, N. (2018). Reinforcement learning applied to Forex trading. *Applied Soft Computing*, 73, 783–794.
- Chen, W.-H., Shih, J.-Y., & Wu, S. (2006). Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets. *International Journal of Electronic Finance*, 1(1), 49–67.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- Di, M., & Joo, E. M. (2007). A survey of machine learning in wireless sensor networks from networking and application perspectives. *2007 6th International Conference on Information, Communications & Signal Processing*, 1–5.

- Eng, M. H., Li, Y., Wang, Q.-G., & Lee, T. H. (2008). Forecast forex with ANN using fundamental data. *2008 International Conference on Information Management, Innovation Management and Industrial Engineering*, 1, 279–282.
- Freed, M., & Lee, J. (2013). Application of support vector machines to the classification of galaxy morphologies. *2013 International Conference on Computational and Information Sciences*, 322–325.
- Gerlein, E. A., McGinnity, M., Belatreche, A., & Coleman, S. (2016). Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, 54, 193–207.
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42.
- Gruber, M. H. J. (2017). *Improving efficiency by shrinkage: The James-Stein and ridge regression estimators*. Routledge.
- Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: an interdisciplinary review. *Journal of Big Data*, 7(1), 1–45.
- Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.
- He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212, 106622.
- Hilt, D. E., & Seegrift, D. W. (1977). *Ridge, a computer program for calculating ridge regression estimates* (Vol. 236). Department of Agriculture, Forest Service, Northeastern Forest Experiment~....
- Kamruzzaman, J., & Sarker, R. (2003). Comparing ANN based models with ARIMA for prediction of forex rates. *Asor Bulletin*, 22(2), 2–11.
- Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1–2), 307–319.
- Lee, J. W., Park, J., Jangmin, O., Lee, J., & Hong, E. (2007). A multiagent approach to q - learning for daily stock trading. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6), 864–877.

- Li, J., & Tsang, E. P. K. (1999). Investment decision making using FGP: a case study. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 2, 1253–1259.
- Li, S.-T., & Kuo, S.-C. (2008). Knowledge discovery in financial investment for forecasting and trading strategy through wavelet-based SOM networks. *Expert Systems with Applications*, 34(2), 935–951.
- Lu, C.-C., & Wu, C.-H. (2009). Support vector machine combined with GARCH models for call option price prediction. *2009 International Conference on Artificial Intelligence and Computational Intelligence*, 1, 35–40.
- Maggini, M., Giles, C. L., & Home, B. (1997). Financial time series forecasting using k-nearest neighbors classification. *Nonlinear Financial Forecasting: Proceedings of the First INFFC*, 169–181.
- Mammone, A., Turchi, M., & Cristianini, N. (2009). Support vector machines. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(3), 283–289.
- McDonald, S., Coleman, S., McGinnity, T. M., Li, Y., & Belatreche, A. (2014). A comparison of forecasting approaches for capital markets. *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFER)*, 32–39.
- Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(6), 275–285.
- Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neuroinformatics*, 7, 21.
- Nguyen, T. T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, 10(4), 56–76.
- Ni, L., Li, Y., Wang, X., Zhang, J., Yu, J., & Qi, C. (2019). Forecasting of forex time series data based on deep learning. *Procedia Computer Science*, 147, 647–652.
- Nielsen, D. (2016). *Tree boosting with xgboost-why does xgboost win" every" machine learning competition?*
- Qi, M., & Zhang, G. P. (2008). Trend time-series modeling and forecasting with neural networks. *IEEE Transactions on Neural Networks*, 19(5), 808–816.

- Sewell, M. V., & Yan, W. (2008). Ultra high frequency financial data. *Proceedings of the 10th Annual Conference Companion on Genetic and Evolutionary Computation*, 1847–1850.
- Subasi, A. (2020). *Practical Machine Learning for Data Analysis Using Python*. Academic Press.
- Tenti, P. (1996). Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence*, 10(6), 567–582.
- Wang, L., Sugiyama, M., Yang, C., Zhou, Z.-H., & Feng, J. (2008). On the margin explanation of boosting algorithms. *COLT*, 479–490.
- Wernick, M. N., Yang, Y., Brankov, J. G., Yourganov, G., & Strother, S. C. (2010). Machine learning in medical imaging. *IEEE Signal Processing Magazine*, 27(4), 25–38.
- Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning* (Vol. 2, Issue 3). MIT press Cambridge, MA.
- Witten, I. H., Frank, E., Hall, M. A., Pal, C. J., & DATA, M. (2005). Practical machine learning tools and techniques. *DATA MINING*, 2, 4.
- Yazdi, S. H. M., & LASHKARI, Z. H. (2012). Technical analysis of Forex by Parabolic SAR Indicator. *International Islamic Accounting and Finance Conference*.

Anexos

Anexo I. Módulo de Extracción

Extracción

A continuación, se puede observar el código implementado para extraer la información de los pares de divisas a través de la API REST de FXCM.

```
##---- Librerías ----
import fxcmpy
import socketio
import pandas as pd
import datetime as dt

##---- Conexion API REST ----

# Conexion
TOKEN = "XXXXX"
con = fxcmpy.fxcmpy(access_token = TOKEN, log_level = 'debug', log_file = 'log.txt
')

##--- Extraccion y Guardado ----

# Periodo de Tiempo
start = dt.datetime(2017, 1, 1)
stop = dt.datetime(2020, 12, 31)

# EUR/USD
ts_eur = con.get_candles('EUR/USD', period = 'H6', start = start, stop = stop)
ts_eur.to_pickle("../data/raw/ts_eur_usd.pkl")

# USD/JPY
ts_usd = con.get_candles('USD/JPY', period = 'H6', start = start, stop = stop)
ts_usd.to_pickle("../data/raw/ts_usd_jpy.pkl")

# AUD/JPY
ts_aud = con.get_candles('AUD/JPY', period = 'H6', start = start, stop = stop)
ts_aud.to_pickle("../data/raw/ts_aud_jpy.pkl")

# Desconectar
con.close()
```

Anexo II. Módulo de Transformación y Preprocesamiento

```
##---- Librerías ----

import pandas as pd
```

```

import numpy as np
from ta.volatility import BollingerBands, AverageTrueRange
from ta.momentum import ROCIndicator, WilliamsRIndicator, RSIIndicator, Stochastic
Oscillator
from ta.trend import SMAIndicator, MACD, PSARIndicator
import plotly.express as px
import plotly.io as pio
from ta.utils import dropna

##---- Cargar Datos ----

# Cargar las series temporales
eur_usd = pd.read_pickle("../data/raw/ts_eur_usd.pkl")
usd_jpy = pd.read_pickle("../data/raw/ts_usd_jpy.pkl")
aud_jpy = pd.read_pickle("../data/raw/ts_aud_jpy.pkl")

##---- Funciones ----

# Funcion para tener el formato limpio que se necesita
def format_ts(df):

    # Columnas necesarias
    df_format = df.reset_index()
    df_format = df_format.loc[:, df_format.columns.str.contains('date|ask|tick')]

    # Renombrar
    df_format.columns = ['date', 'open', 'close', 'high', 'low', 'volume']

    return df_format

# Funcion para formar las variables adicionales para la modelizacion
def create_by_window(df, size_window = 20, desv_window = 2):

    ##---- Bandas de Bollinger ----

    # Inicializar el indicador de las bandas de Bollinger
    indicator_bb = BollingerBands(close = df["close"], window = size_window, windo
w_dev = desv_window)

    # Añadir indicadores
    df[f'bb_bbbhi ({size_window})'] = indicator_bb.bollinger_hband_indicator()
    df[f'bb_bbbli ({size_window})'] = indicator_bb.bollinger_lband_indicator()
    df[f'bb_bbbw ({size_window})'] = indicator_bb.bollinger_wband()

    ##---- AverageTrueRange ----

    # Inicializar ATR
    indicator_atr = AverageTrueRange(high = df["high"], low = df["low"], close = d
f["close"], window = size_window)

    # Añadir variable
    df[f'atr ({size_window})'] = indicator_atr.average_true_range()

    ##---- William's %R ----

    # Inicializar William's %R
    indicator_w = WilliamsRIndicator(high = df["high"], low = df["low"], close = d
f["close"], lbp = size_window)

```



```

# Añadir variable
df[f'willr ({size_window})'] = indicator_w.williams_r()

##---- Índice de Fuerza Relativa ----

# Inicializar RSI
indicator_rsi = RSIIndicator(close = df["close"], window = size_window)

# Añadir variable
df[f'rsi ({size_window})'] = indicator_rsi.rsi()

##---- Media Movil ----

# Inicializar MA
indicator_ma = SMAIndicator(close = df["close"], window = size_window)

# Añadir variable
df[f'close ma ({size_window})'] = indicator_ma.sma_indicator()

# Promedio móvil de la variable de rezagos
indicator_ma = SMAIndicator(close = df["roc(t-1)"], window = size_window)
df[f'roc(t-1) ma ({size_window})'] = indicator_ma.sma_indicator()

##---- MACD ----

# Inicializar MACD
indicator_macd = MACD(close = df["close"],
                      window_slow = size_window,
                      window_fast = size_window // 2,
                      window_sign = (size_window * 2) // 5)

# Añadir variable
df[f'macd ({size_window})'] = indicator_macd.macd()
df[f'macd signal ({size_window})'] = indicator_macd.macd_signal()

##---- STOCH ----

# Inicializar STOCH
indicator_stoch = StochasticOscillator(close = df["close"], high = df["high"],
low = df["low"], window = size_window)

# Añadir variable
df[f'stoch ({size_window})'] = indicator_stoch.stoch()

return df

# Función para crear el dataframe principal
def create_principal(df):

##---- Cambio Porcentual del Precio ----

# Inicializar ROC
indicator_roc = ROCIndicator(close = df["close"], window = 1)

# Añadir variable
df['roc'] = indicator_roc.roc()

##---- Parabolic SAR ----

```

```

    # Inicializar PSAR
    indicador_psar = PSARIndicator(high = df["high"], low = df["low"], close = df[
"close"])

    # Añadir variable
    df['psar'] = indicador_psar.psar()

    ##---- Variables adicionales ----

    # Variable rezagos
    df['roc(t-1)'] = df["roc"].shift(1)

    # Temporalidad
    df['hour'] = df['date'].dt.hour
    df['day'] = df["date"].dt.day_name()

    ##---- Variable Objetivo ----

    # Direccion del precio en el siguiente periodo
    df['direccion'] = df["close"].diff().shift(-1)

    return df

# Consolidar las fases del procesamiento
def process_df(df, size_window = [10, 15, 20], desv_window = 2):

    # Formato
    df = format_ts(df)

    ##---- Creación de variables ----

    # Variables principales
    df = create_principal(df)

    for sz in size_window:
        df = create_by_window(df, sz, desv_window)

    # Limpiar valores Nan
    df = df.dropna().copy()

    # Creacion variable objetivo
    df['class'] = np.where(df["direccion"] > 0, "UP", "DOWN")

    return df

# Aplicación creación de variables
eur_usd_td = process_df(eur_usd, size_window = [5, 10, 15, 20, 30])
usd_jpy_td = process_df(usd_jpy, size_window = [5, 10, 15, 20, 30])
aud_jpy_td = process_df(aud_jpy, size_window = [5, 10, 15, 20, 30])

##---- Preprocesamiento adicional ----

##--- Consolidar horas del día ----

# Definir función para agrupar los valores
def conditions(hr):
    if hr <= 4 or hr <= 4:
        return 4
    elif hr <= 10:

```

```

        return 10
    elif hr <= 16:
        return 16
    else:
        return 22

# Colapsar horas trading
eur_usd_td['hour'] = eur_usd_td['hour'].apply(conditions)
usd_jpy_td['hour'] = usd_jpy_td['hour'].apply(conditions)
aud_jpy_td['hour'] = aud_jpy_td['hour'].apply(conditions)

##--- Consolidar días ----

# Definir función para agrupar Los valores
def conditions_days(day):
    if day == "Sunday":
        return "Monday"
    else:
        return day

# Colapsar horas trading
eur_usd_td['day'] = eur_usd_td['day'].apply(conditions_days)
usd_jpy_td['day'] = usd_jpy_td['day'].apply(conditions_days)
aud_jpy_td['day'] = aud_jpy_td['day'].apply(conditions_days)

```

Anexo III. Módulo de Modelación y Backtesting

```

##---- Librerías ----

import pandas as pd
import numpy as np
import re
from datetime import timedelta, datetime
from pycaret.classification import *
from sklearn.metrics import *

##---- Funciones ----

# Dividir lista en bloques
def chunks(lst, n):
    """Yield successive n-sized chunks from lst."""
    for i in range(0, len(lst), n):
        yield lst[i:i + n]

# Realizar backtesting
def backtesting(df_train, df_backtesting, columns_include, numeric_cols, categoric_
_cols,
                pca_features = False,
                n_folds = 6,
                lst_models = ['ada', 'lr', 'ridge', 'rf', 'gbc', 'dt', 'svm', 'cat
boost', 'xgboost',
                             'et', 'rbfsvm', 'gpc', 'mlp']):

##---- Flujo de AutoML ----

```

```

# Configuración Inicial
s = setup(data = df_train[columns_include],
          train_size = 0.8,
          target = 'class',
          fold_strategy = 'timeseries',
          data_split_shuffle = False,
          numeric_features = numeric_cols,
          categorical_features = categoric_cols,
          fold = n_folds,
          feature_selection = True, feature_selection_threshold = 0.9,
          remove_multicollinearity = True,
          pca = pca_features, # Sirve para La NN
          ignore_low_variance = True,
          silent = True, session_id = 777)

# Entrenar Los distintos modelos
top3 = compare_models(n_select = 3, sort = 'Accuracy', include = lst_models)
results_models = pull()
names_top3 = ";".join(list(results_models.index)[:3])
model_smv = create_model('svm')
results_svm = pull()

# Colapsar el top 3 de modelos mediante su combinación
blender_models = blend_models(top3)
results_blender = pull()

# Consolidar modelos
lst_models = [top3[0], model_smv, blender_models]
name_models = ['best', 'svm', 'ensemble']

# Finalizar Modelo (Entrenar sobre todo el conjunto)
final_models = [finalize_model(i) for i in lst_models]

# Predicción sobre el conjunto no visto
lst_predictions = [predict_model(i, data = df_backtesting) for i in final_models]

ls]

##---- Obtener metricas de precision ----
accuracy_models = dict()

# Formar diccionario donde guardar Las metricas
accuracy_models["best"] = results_models.iloc[0,1]
accuracy_models["svm"] = results_svm.loc["Mean", "Accuracy"]
accuracy_models["ensemble"] = results_blender.loc["Mean", "Accuracy"]

##---- Consolidar Las predicciones ----

# Creacion DataFrame para guardar valores
res_consolidados = df_backtesting[['date', 'close', 'class']]

# Predicciones
for i in range(len(lst_predictions)):

    res_consolidados[name_models[i]] = lst_predictions[i]['Label']

# Reporte Accuracy
for k,v in accuracy_models.items():

```

```

        res_consolidados[f"{k} acc"] = v

    # Mejores Modelos
    res_consolidados["models top3"] = names_top3

    return res_consolidados

# Realizar backtesting en una ventana movil de tiempo
def backtesting_by_window(df, columns_include, numeric_cols, categoric_cols,
                          begin_backtesting, end_backtesting,
                          pca_features = False,
                          n_folds = 6,
                          lst_models = ['ada', 'lr', 'ridge', 'rf', 'gbc', 'dt', '
svm', 'catboost', 'xgboost',
                                      'et', 'rbfsvm', 'gpc', 'mlp'],
                          size_window = 7, size_train = 250):

    # Recortar df
    df = df[df['date'] <= end_backtesting]

    # Parametros Auxiliares
    punto_corte = np.argmax(df['date'] >= begin_backtesting)

    # Conformar Los distintos grupos de simulacion y backtesting
    idx_bck = list(range(punto_corte, len(df['date'])))
    grupos_bck = list(chunks(idx_bck, size_window))

    # Formar Los grupos de indices para entrenar y realizar el backtesting
    lst_idx = [list(range(grupo_idx[0] - size_train, grupo_idx[0])) + grupo_idx fo
r grupo_idx in grupos_bck]

    # Realizar el Backtesting
    lst_backtesting = list()

    for idx in lst_idx:

        print(f"entrenamiento bloque {idx[0]}---{idx[-1]}")

        # Dividir al dataframe en entrenamiento y simulacion
        idx_train = idx[:-size_window]
        idx_simulacion = idx[-size_window:]

        df_train = df.iloc[idx_train]
        df_simulacion = df.iloc[idx_simulacion]

        # Realizar Backtesting
        result_bck = backtesting(df_train, df_simulacion, columns_include, numeric
_cols,
                                categoric_cols, pca_features, n_folds, lst_models
)

        # Guardar Los resultados en una Lista
        lst_backtesting.append(result_bck)

    # Colapsar Predicciones
    res_consolidados_bck = pd.DataFrame()

```

```

    for result_bck in lst_backtesting:
        res_consolidados_bck = res_consolidados_bck.append(result_bck)

    return res_consolidados_bck

##---- Cargar datos ----

# Cargar las series temporales
eur_usd_td = pd.read_pickle("../data/tidy/ts_eur_usd_tidy.pkl")
usd_jpy_td = pd.read_pickle("../data/tidy/ts_usd_jpy_tidy.pkl")
aud_jpy_td = pd.read_pickle("../data/tidy/ts_aud_jpy_tidy.pkl")

##---- Desarrollos modelos y backtesting ----

# Columnas que se incluyen en el modelo
columns_include = ['open', 'close', 'high', 'low', 'volume', 'roc', 'psar',
                   'roc(t-1)', 'hour', 'day', 'bb_bbhi (5)', 'bb_bbli (5)',
                   'bb_bbw (5)', 'atr (5)', 'willr (5)', 'rsi (5)', 'close ma (5)'
                   ,
                   'roc(t-1) ma (5)', 'macd (5)', 'macd signal (5)', 'stoch (5)',
                   'bb_bbhi (10)', 'bb_bbli (10)', 'bb_bbw (10)', 'atr (10)', 'wil
lr (10)',
                   'rsi (10)', 'close ma (10)', 'roc(t-1) ma (10)', 'macd (10)',
                   'macd signal (10)', 'stoch (10)',
                   'bb_bbhi (15)', 'bb_bbli (15)', 'bb_bbw (15)', 'atr (15)', 'wil
lr (15)',
                   'rsi (15)', 'close ma (15)', 'roc(t-1) ma (15)', 'macd (15)',
                   'macd signal (15)', 'stoch (15)', 'bb_bbhi (20)', 'bb_bbli (20)'
                   ,
                   'bb_bbw (20)', 'atr (20)', 'willr (20)', 'rsi (20)', 'close ma
(20)',
                   'roc(t-1) ma (20)', 'macd (20)', 'macd signal (20)',
                   'stoch (20)',
                   'bb_bbhi (30)', 'bb_bbli (30)',
                   'bb_bbw (30)', 'atr (30)', 'willr (30)', 'rsi (30)', 'close ma
(30)',
                   'roc(t-1) ma (30)', 'macd (30)', 'macd signal (30)',
                   'stoch (30)', 'class']

numeric_cols = ['open', 'close', 'high', 'low', 'volume', 'roc', 'psar',
                'roc(t-1)', 'bb_bbw (5)', 'atr (5)', 'willr (5)', 'rsi (5)', 'clos
e ma (5)',
                'roc(t-1) ma (5)', 'macd (5)', 'macd signal (5)', 'stoch (5)',
                'bb_bbw (10)', 'atr (10)', 'willr (10)', 'rsi (10)', 'close ma (10
)',
                'roc(t-1) ma (10)', 'macd (10)', 'macd signal (10)', 'stoch (10)',
                'bb_bbw (15)', 'atr (15)', 'willr (15)', 'rsi (15)', 'close ma (15
)',
                'roc(t-1) ma (15)', 'macd (15)', 'macd signal (15)', 'stoch (15)',
                'bb_bbw (20)', 'atr (20)', 'willr (20)', 'rsi (20)', 'close ma (20
)',
                'roc(t-1) ma (20)', 'macd (20)', 'macd signal (20)', 'stoch (20)',
                'bb_bbw (30)', 'atr (30)', 'willr (30)', 'rsi (30)', 'close ma (30
)',
                'roc(t-1) ma (30)', 'macd (30)', 'macd signal (30)', 'stoch (30)']

categoric_cols = ['hour', 'day', 'bb_bbhi (5)', 'bb_bbli (5)', 'bb_bbhi (10)',
                  'bb_bbli (10)', 'bb_bbhi (15)', 'bb_bbli (15)', 'bb_bbhi (20)',
                  'bb_bbli (20)',

```

```

        'bb_bbhi (30)', 'bb_bbli (30)']

##---- Escenarios ----

# EUR/USD
esc1_150_eur_usd = backtesting_by_window(eur_usd_td, columns_include, numeric_cols
, categoric_cols,
begin_backtesting = datetime.strptime('20
19-01-01', '%Y-%m-%d'),
end_backtesting = datetime.strptime('2020
-12-31', '%Y-%m-%d'),
pca_features = False,
n_folds = 3,
size_window = 5,
size_train = 150)

esc1_750_eur_usd = backtesting_by_window(eur_usd_td, columns_include, numeric_cols
, categoric_cols,
begin_backtesting = datetime.strptime('20
19-01-01', '%Y-%m-%d'),
end_backtesting = datetime.strptime('2020
-12-31', '%Y-%m-%d'),
pca_features = False,
n_folds = 6,
size_window = 5,
size_train = 750)

esc1_1500_eur_usd = backtesting_by_window(eur_usd_td, columns_include, numeric_col
s, categoric_cols,
begin_backtesting = datetime.strptime('2
019-01-01', '%Y-%m-%d'),
end_backtesting = datetime.strptime('202
0-12-31', '%Y-%m-%d'),
pca_features = False,
n_folds = 10,
size_window = 5,
size_train = 1500)

# Guardar resultados
esc1_150_eur_usd.to_pickle("../output/escenario 1/eur_usd/esc1_150_eur_usd.pkl")
esc1_750_eur_usd.to_pickle("../output/escenario 1/eur_usd/esc1_750_eur_usd.pkl")
esc1_1500_eur_usd.to_pickle("../output/escenario 1/eur_usd/esc1_1500_eur_usd.pkl")

# USD/JPY
esc1_150_usd_jpy = backtesting_by_window(usd_jpy_td, columns_include, numeric_cols
, categoric_cols,
begin_backtesting = datetime.strptime('20
19-01-01', '%Y-%m-%d'),
end_backtesting = datetime.strptime('2020
-12-31', '%Y-%m-%d'),
pca_features = False,
n_folds = 3,
size_window = 5,
size_train = 150)

esc1_750_usd_jpy = backtesting_by_window(usd_jpy_td, columns_include, numeric_cols
, categoric_cols,
begin_backtesting = datetime.strptime('20
19-01-01', '%Y-%m-%d'),
end_backtesting = datetime.strptime('2020

```

```

-12-31', '%Y-%m-%d'),
                                pca_features = False,
                                n_folds = 6,
                                size_window = 5,
                                size_train = 750)

esc1_1500_usd_jpy = backtesting_by_window(usd_jpy_td, columns_include, numeric_cols,
categoric_cols,
                                begin_backtesting = datetime.strptime('2019-01-01', '%Y-%m-%d'),
                                end_backtesting = datetime.strptime('2020-12-31', '%Y-%m-%d'),
                                pca_features = False,
                                n_folds = 10,
                                size_window = 5,
                                size_train = 1500)

# Guardar resultados
esc1_150_usd_jpy.to_pickle("../output/escenario 1/usd_jpy/esc1_150_usd_jpy.pkl")
esc1_750_usd_jpy.to_pickle("../output/escenario 1/usd_jpy/esc1_750_usd_jpy.pkl")
esc1_1500_usd_jpy.to_pickle("../output/escenario 1/usd_jpy/esc1_1500_usd_jpy.pkl")

# AUD/JPY
esc1_150_aud_jpy = backtesting_by_window(aud_jpy_td, columns_include, numeric_cols,
categoric_cols,
                                begin_backtesting = datetime.strptime('2019-01-01', '%Y-%m-%d'),
                                end_backtesting = datetime.strptime('2020-12-31', '%Y-%m-%d'),
                                pca_features = False,
                                n_folds = 3,
                                size_window = 5,
                                size_train = 150)

esc1_750_aud_jpy = backtesting_by_window(aud_jpy_td, columns_include, numeric_cols,
categoric_cols,
                                begin_backtesting = datetime.strptime('2019-01-01', '%Y-%m-%d'),
                                end_backtesting = datetime.strptime('2020-12-31', '%Y-%m-%d'),
                                pca_features = False,
                                n_folds = 6,
                                size_window = 5,
                                size_train = 750)

esc1_1500_aud_jpy = backtesting_by_window(aud_jpy_td, columns_include, numeric_cols,
categoric_cols,
                                begin_backtesting = datetime.strptime('2019-01-01', '%Y-%m-%d'),
                                end_backtesting = datetime.strptime('2020-12-31', '%Y-%m-%d'),
                                pca_features = False,
                                n_folds = 10,
                                size_window = 5,
                                size_train = 1500)

# Guardar resultados
esc1_150_aud_jpy.to_pickle("../output/escenario 1/aud_jpy/esc1_150_aud_jpy.pkl")

```



```
esc1_750_aud_jpy.to_pickle("../output/escenario 1/aud_jpy/esc1_750_aud_jpy.pkl")  
esc1_1500_aud_jpy.to_pickle("../output/escenario 1/aud_jpy/esc1_1500_aud_jpy.pkl")
```

Anexo. Artículo de investigación

Evaluación de Métodos de Aprendizaje Automático aplicados al Trading de Forex

Paúl Andrés Apolo Viteri

Universidad Internacional de la Rioja, Logroño (España)

Febrero 23, 2022

RESUMEN

El mercado Forex es uno de los más grandes a nivel mundial. Este mercado es el más líquido del mundo debido a que comercia aproximadamente 6,6 billones de dólares estadounidenses diarios. Los agentes de trading son atraídos a este tipo de mercados financieros, donde tienen como objetivo aplicar estrategias de compraventa que les generen una rentabilidad sostenida. La hipótesis de mercados eficientes postula que no se pueden pronosticar los valores de los mercados financieros de forma consistente a un largo plazo. En este sentido, se realiza la comparación de diferentes metodologías de aprendizaje automático para generar evidencia empírica en contra de esta hipótesis. Los resultados de los experimentos sugieren que se puede obtener un retorno acumulado significativo mediante la aplicación de algoritmos de aprendizaje automático.

I. INTRODUCCIÓN

Las principales bolsas de valores del mundo mueven miles de millones de dólares diariamente. Entre las principales podemos encontrar a la bolsa de valores de Nueva York (NYSE), NASDAQ y la bolsa de valores de Tokio (TSE). Por ejemplo, la NYSE registra un movimiento de unos 3 billones de dólares estadounidenses al día solo en préstamos repos, según los datos proporcionados por la BBC en septiembre de 2019. A pesar de todo el dinero que transan las bolsas de valores, existe un mercado en donde se mueven aún más dólares diarios. El mercado de divisas (Forex) es el mercado más líquido del mundo y tiene relación con todas las bolsas de valores.

El trading (i.e., la compraventa de activos, acciones, divisas, futuros y derivados) tiene lugar en estos grandes mercados financieros como lo son las bolsas de valores y el mercado de divi-

sas, debido a que tienen una muy alta liquidez y cada operación se puede realizar casi instantáneamente. Las principales técnicas usadas por la mayoría de los agentes que realizan trading son el análisis fundamental y el análisis técnico. El análisis fundamental se basa en predecir a largo plazo el comportamiento de las series financieras en función de fundamentos macroeconómicos (e.g., tipo de interés, la inflación, el producto interno bruto, tasa de desempleo, entre otros). Por otro lado, el análisis técnico tiene como finalidad predecir el movimiento de las series financieras a un corto o medio plazo a partir de sus líneas de tendencia, indicadores técnicos o análisis estadísticos.

Los agentes que invierten dentro de las bolsas de valores y el mercado de divisas han buscado técnicas cada vez más complejas que les generen una mayor rentabilidad. Existe mucho interés gracias a las enormes cantidades de dinero que se mueven en estos mercados. La hi-

unir
LA UNIVERSIDAD
EN INTERNET

PALABRAS CLAVE

mercado Forex, estrategias de compraventa, aprendizaje automático

pótesis de mercados eficientes asegura que, en cualquier momento dado, el precio de un instrumento siempre refleja completamente toda la información disponible, la cual sería imposible de conseguir. La industria financiera aún debate la idea de aceptar que no existe forma posible de “ganarle al mercado”, es decir, que los precios de los instrumentos financieros no son predecibles [1].

La inteligencia artificial es un campo que está siendo ampliamente utilizado por investigadores públicos y privados para hacer más eficiente el uso de recursos, optimizar procesos, innovar y generar mayores beneficios económicos. Dentro de la inteligencia artificial, se encuentran los campos de minería de datos y aprendizaje automático, los cuales han demostrado ser capaces de resolver con gran precisión problemas de clasificación y regresión de la vida real en muchas áreas diferentes como comunicaciones [2], análisis del tráfico en la internet [3], imagen médica [4], astronomía [5], análisis de series temporales [6], entre otras.

Actualmente, el trading en la industria financiera también está empezando a usar técnicas de aprendizaje automático para el análisis cuantitativo y la predicción del comportamiento de series temporales [7]; sin embargo, a comparación de las demás áreas anteriormente mencionadas, la alta volatilidad e injerencia de muchos factores en este campo hacen más difícil tener resultados precisos.

La aplicación de las técnicas de aprendizaje automático en la predicción financiera presenta una serie de complicaciones estructurales que merman su rendimiento. Maggini, Giles & Horne en 1997 señalan que existe una inherente dificultad estadística en generar indicadores técnicos confiables, esto se debe a que las reglas inferidas para producir predicciones precisas cambian continuamente en las series temporales financieras [8]. Esta situación se refleja en varias investigaciones que han reportado rendimientos por debajo del 60% con modelos de aprendizaje automático, los cuales han mostrado impresionantes resultados en otras áreas que la predicción financiera [7].

II. ESTADO DEL ARTE

El área de trading financiero está compuesta por varias ramas, dentro de las cuales particularmente dos han reportado implementaciones de técnicas de aprendizaje automático con relativo éxito: fijación de precios de derivados y predicción financiera [7].

La literatura es amplia con respecto a los modelos de aprendizaje automático que se han probado para la predicción de series financieras, donde se puede encontrar desde modelos simples como el Naive Bayes hasta modelos mucho más complejos como lo son las redes neuronales artificiales convolucionales (CNN) o recurrentes (RNN). No obstante, la dificultad implícita en la predicción de este tipo de series ha provocado que un gran número de investigaciones [9, 10, 11, 12, 13, 14] reporten un bajo rendimiento de sus modelos (con métricas de precisión por debajo del 60%), cuando los mismos modelos tienen rendimientos sobresalientes en otras áreas fuera de la predicción financiera.

Las redes neuronales artificiales son probablemente el método más utilizado en la predicción de series financieras. Investigaciones tempranas como la de Tenti [14] comparó el rendimiento de tres redes neuronales recurrentes con base en sus retornos, al simular su aplicación en el mercado de futuros de divisas.

En un estudio realizado por Li y Tsang [1], se realizó la predicción del retorno esperado del Dow Jones Industrial Average (DJIA) usando programación genética financiera. El trabajo no solo se enfocó en la precisión de las predicciones, sino también en los retornos esperados de un conjunto de inversiones que sigan los pronósticos de los modelos. El modelo de programación genética obtuvo un retorno positivo por encima del 60% y un retorno anual por encima del 40% sobre un periodo de trading de cuatro años. Los resultados presentan un asombroso retorno financiero aún sin considerar los costos del trading, lo cual sugiere que los indicadores técnicos podrían generar modelos rentables basados en reglas para predecir complejas series temporales financieras.

Existen otras técnicas de aprendizaje auto-

mático que han mostrado resultados prometedores dentro del área de predicción financiera. Por ejemplo, las máquinas de vector de soporte (SVM) son una de las técnicas que han obtenido mejores rendimientos después de las redes neuronales. Kim en 2003 investigó el uso de las SVM para la predicción de la dirección diaria de precios del índice bursátil KOSPI [11]. El modelo usó indicadores de análisis técnico (momento, Williams %R y el índice de canales de materia prima) como variables de entrada y su mejor precisión reportada, después de entrenar el modelo bajo diferentes parámetros, fue de 57.83 %.

La investigación de Barbosa y Belo se dividió en varios trabajos en los cuales utilizaron sistemas de agentes únicos e inteligentes para ejecutar operaciones algorítmicas en el mercado Forex [15], manejar fondos de inversiones [16] y realizar trading en múltiples mercados [17], enfocándose en la rentabilidad y la reducción máxima de la inversión como métricas de rendimiento. El sistema realiza el aprendizaje por medio del promedio de la combinación ponderada de los resultados de simples modelos de aprendizaje automático, como lo son el OneR, C4.5, JRip, Logistic Model Tree, KStar, NN, SVM y Naive Bayes. El módulo de clasificación tiene como resultado una precisión baja del 52.47 %; sin embargo, el sistema completo produce una impresionante tasa de 66.67 % de rentabilidad en el periodo de pruebas.

Gerlein et al. (2016) contribuyó con el estado del arte investigando el uso de técnicas más simples de aprendizaje automático. Los mejores resultados que obtuvieron los modelos de todos los escenarios que plantearon fueron los siguientes: OneR obtuvo una precisión del 58.9 % y un retorno acumulado del 32 %, C4.5 tuvo una precisión del 64.08 % y un retorno acumulado del 57.17 %, Jrip presentó una precisión de 63.75 % y un retorno acumulado de 95.94 %, LMT presentó una precisión de 64.37 % y un retorno acumulado de 39.46 %, K-star obtuvo una precisión de 56.88 % y un retorno acumulado de 73.16 %, y finalmente Naive Bayes tuvo una precisión del 63.69 % y un retorno acumulado de 22.62 %.

III. OBJETIVOS Y METODOLOGÍA

A. OBJETIVO

El objetivo de este estudio es comparar distintos métodos de aprendizaje automático, con base en los que se detallan en el estado del arte, aplicados al pronóstico de series temporales financieras. De tal forma, el trabajo pretende generar evidencia empírica en contra de la hipótesis de mercados eficientes prediciendo el comportamiento de pares de divisas en el mercado Forex.

B. METODOLOGÍA

Para alcanzar los objetivos específicos se deben de realizar una serie de pasos, los cuales se detallan a continuación:

1. Determinar distintos métodos de aprendizaje automático que tengan el potencial de realizar predicciones precisas de los mercados financieros, según el estado del arte.
2. Seleccionar tres pares de divisas representativos de algún tipo de mercado Forex (e.g., mercados estables, exóticos o volátiles) y el periodo de tiempo durante el cual se realizará el experimento.
3. Entrenar y validar los distintos métodos seleccionados de aprendizaje automático.
 - a) Dividir al conjunto de datos en entrenamiento, validación y prueba.
 - b) Realizar el entrenamiento de los diferentes modelos de aprendizaje automático usando técnicas de validación cruzada.
 - c) Implementar la metodología seleccionada para generar el modelo final que realice la predicción.
4. Simular la aplicación de los modelos durante el periodo de tiempo de prueba.
 - a) Reentrenar los modelos durante ventanas de tiempo simulando como se lo aplicaría en la realidad.

5. Comparar las métricas de rendimiento de los distintos métodos y su retorno financiero sobre el conjunto de datos de prueba.
6. Demostrar la capacidad de los modelos para generar pronósticos rentables.

IV. CONTRIBUCIÓN

El estado del arte señala que se han evaluado varias técnicas de aprendizaje automático para la predicción de las series temporales financieras. Así mismo, los autores han probado distintos enfoques para abordar el problema de aprendizaje automático supervisado (regresión o clasificación) en este campo. Si bien existen distintos modelos o enfoques empleados, las métricas de evaluación muestran que, en general, la aplicación de las técnicas de aprendizaje automático en los mercados financieros tiene un rendimiento más bajo a comparación de su aplicación en otras áreas.

El bajo poder predictivo de los modelos es el principal argumento a favor de que no se puede predecir el comportamiento de las series temporales financieras, haciendo elusión a la hipótesis de mercados eficientes. Por tanto, para el desarrollo del trabajo en cuestión, se va a comparar tanto las métricas de rendimiento como el impacto financiero de las distintas técnicas de aprendizaje automático. Esto con el motivo de establecer que no se necesitan métricas de evaluación tan precisas, como en otras áreas, para obtener resultados financieros positivos. Esta sería nuestra principal prueba en contra de la hipótesis de mercados eficientes.

Autores como Barbosa y Belo en 2008 y Gerlein et al. en 2016 proponen un sistema multiagente para realizar la simulación del trading a partir de los modelos entrenados [17, 7]. Este sistema de trading se compone de dos agentes: el primero tiene la función de realizar el entrenamiento, validación y predicción (agente de trading); y el segundo realiza la acción en el mercado, en función del pronóstico del primer agente, y extrae más información para seguir alimentando a los modelos (agente de mercado).

Este sistema multiagente puede ser replicado mediante el entrenamiento y despliegue (simulado) de un modelo de aprendizaje automático durante un número n de iteraciones sobre una serie temporal de los precios de un par de divisas. Es así como, el flujo que se construye para simular el backtesting será el siguiente:

1. Conformar un conjunto de datos para predecir el precio de un par de divisas.
2. Establecer un periodo sobre el cual se desee realizar el backtesting.
3. Entrenar el modelo de aprendizaje automático sobre un subconjunto de datos del dataset principal que este antes del periodo de backtesting.
4. Predecir los siguientes k valores que quedaron fuera del conjunto de entrenamiento, donde se establece que k es un número natural menor o igual a 15.
5. Mover el rango del subconjunto de datos k posiciones.
6. Repetir el entrenamiento y predicción de los k valores posteriores.
7. Realizar esto un número de n veces hasta completar el periodo de backtesting.

En el presente estudio se emplean las series temporales de los precios de los pares de divisas del mercado Forex extraídas desde el bróker Forex Capital Markets (FXCM). Los pares de divisas seleccionados fueron los siguientes: euro – dólar estadounidense (EUR/USD), dólar estadounidense – yen (USD/JPY) y dólar australiano - yen (AUD/JPY). El periodo de tiempo considerado va desde el primero de enero de 2017 hasta el 31 de diciembre de 2020; sin embargo, el periodo donde se van a medir y evaluar las métricas, y realizar la simulación será desde el primero de enero de 2019 hasta el 31 de diciembre de 2020.

A partir del precio ask de cierre del par de divisas se construye la variable objetivo, la cual indica si en el siguiente periodo el precio sube o baja. Así mismo, los indicadores técnicos usados para la predicción son formados a partir

del vector bruto de precios ask de apertura, cierre, máximo, mínimo y volumen. Entonces, el conjunto de datos para cada par de divisas se conforma por la variable objetivo y una serie de indicadores técnicos que, en general, se usan para el análisis técnico en el mercado Forex. Además, también se incluyen variables que describen características generales de las series temporales (e.g., hora o día de la transacción, precio de apertura, precio de cierre, entre otros).

Las métricas de precisión con las cuales se va a evaluar el desempeño de las distintas metodologías serán las siguientes:

- Precisión (%): porcentaje de las predicciones correctas para el periodo de trading a partir de la técnica de validación cruzada de cada entrenamiento de los modelos.
- Precisión backtesting (%): porcentaje de las predicciones correctas para el periodo de trading simulando su aplicación en la realidad.
- Retorno acumulado (%): porcentaje del retorno acumulado al final del ciclo de trading.

Las distintas metodologías que se van a comparar en el presente trabajo utilizan varios modelos de aprendizaje automático. Es así como, una metodología puede llevar a ajustar y probar distintos modelos para cada uno de los k periodos de ventana móvil sobre los cuales se realizará el entrenamiento. Entonces, el ajuste de los modelos será realizado con técnicas de aprendizaje automático automatizado (o mejor conocido como AutoML). El módulo PyCaret será usado como marco de trabajo para aplicar el AutoML.

Los clasificadores que se utilizarán para ajustar los modelos, y que PyCaret tiene disponibles, son los siguientes: Regresión Logística, Árboles de Decisión, Bosques Aleatorios, Máquinas de Vectores de Soporte (kernel lineal y radial), Clasificador Gaussian Process, Clasificador Ridge, Clasificador Extra-Trees, Clasificador Gradient Boosting, AdaBoost, CatBoost, XGBoost y Red Neuronal con arquitectura Perceptrón Multicapa (MLP). Como

se puede observar, estos clasificadores contemplan la mayoría de los tipos de modelos de aprendizaje automático, que van desde regresiones hasta redes neuronales.

A. Metodología de los experimentos

Las metodologías que se van a experimentar harán uso de los modelos de aprendizaje automático descritos en la sección anterior. El uso de la librería PyCaret ayudará a realizar todo el flujo de trabajo necesario para la modelización y evaluación de los distintos modelos. Además, el uso de la técnica de AutoML servirá para establecer las distintas metodologías que se detallarán a continuación:

1. *Máquinas de Vectores de Soporte*

La primera metodología sirve como punto base para comparar el rendimiento de las otras dos metodologías experimentales contrastando con uno de los modelos más usados en la literatura como lo son las Máquinas de Vectores de Soporte.

2. *Mejor Modelo*

En esta metodología se obtiene el modelo con el mejor rendimiento por cada ventana k de entrenamiento; es decir, cada vez que se vuelva a reentrenar los modelos (aplicando el flujo de trabajo automatizado) se escogerá al que mejor métrica de precisión presente a través de la evaluación cruzada.

3. *Conjunto de Modelos*

Finalmente, la última metodología consiste en aplicar técnicas de evaluación conjunta de modelos (ensemble models) para realizar las predicciones. Para lo cual, se escoge a los mejores 3 modelos en función de su precisión y se los consolida mediante una votación por mayoría simple.

Los experimentos que se llevan a cabo tienen como objetivo probar el más amplio espectro de parámetros distintos que pueden llegar a afectar el rendimiento de cada una de las metodologías. Los parámetros mediante los cuales se establecerán los distintos experimentos serán la longitud de la ventana de entrenamiento de los

modelos y el número de datos de entrenamiento. En función de los resultados obtenidos por Gerlein et al. en 2016 de sus experimentos empíricos [1], se determina los siguientes valores para cada uno de los parámetros:

- Longitud de la ventana de entrenamiento: 5 y 15 periodos
- Número de datos de entrenamiento: 150, 750 y 1500 registros

V. RESULTADOS

A continuación, en la figura 1 se presentan los resultados de la aplicación de las diferentes metodologías para predecir un periodo posterior el movimiento del precio del par de divisa euro y dólar estadounidense. El color verde identifica donde se producen los mejores resultados de cada métrica de evaluación, mientras que, el color rojo indica los peores resultados.

Par de Divisa:			EUR/USD			
Longitud Ventana Entrenamiento	Número Datos Entrenamiento	Número Experimento	Métrica	SVM	Mejor Modelo	Conjunto de Modelos
5	150	1	Precisión	49,83%	57,45%	56,20%
			Precisión backtesting	48,60%	47,49%	47,77%
			Retorno acumulado	1,51%	-17,92%	-14,04%
	750	2	Precisión	49,96%	53,67%	53,29%
			Precisión backtesting	51,82%	50,07%	48,83%
			Retorno acumulado	14,43%	-7,12%	-11,27%
	1500	3	Precisión	50,02%	52,99%	52,72%
			Precisión backtesting	48,51%	49,52%	52,51%
			Retorno acumulado	-11,80%	4,66%	24,03%
15	150	4	Precisión	50,50%	57,38%	56,34%
			Precisión backtesting	49,06%	49,06%	48,41%
			Retorno acumulado	5,71%	-11,98%	-12,72%
	750	5	Precisión	50,11%	53,69%	53,43%
			Precisión backtesting	50,16%	50,21%	48,97%
			Retorno acumulado	3,18%	7,03%	-0,37%
	1500	6	Precisión	50,02%	53,01%	52,73%
			Precisión backtesting	48,64%	50,85%	54,30%
			Retorno acumulado	-14,80%	17,05%	39,23%

Figura 1: Resultados aplicación metodologías sobre EUR/USD

En general, se observa que, los peores resultados de cada métrica de evaluación ocurren en el experimento 1, el cual corresponde a los valores mínimos de longitud de ventana de entrenamiento y del tamaño del conjunto de datos. La precisión del modelo SVM es 49,83%, la cual es la más baja de todas las metodologías y experimentos realizados; sin embargo, como se puede observar su retorno acumulado es de 1,51%. A pesar de que la metodología de escoger el mejor modelo por cada periodo de entrenamiento presenta la mejor precisión con

un 57,47% en el experimento 1, la misma tiene el peor retorno acumulado (-17,92%).

Par de Divisa:			USD/JPY			
Longitud Ventana Entrenamiento	Número Datos Entrenamiento	Número Experimento	Métrica	SVM	Mejor Modelo	Conjunto de Modelos
5	150	1	Precisión	49,87%	58,23%	56,93%
			Precisión backtesting	50,14%	50,37%	50,23%
			Retorno acumulado	1,37%	-11,03%	4,16%
	750	2	Precisión	49,98%	54,23%	53,90%
			Precisión backtesting	47,74%	50,60%	51,43%
			Retorno acumulado	-6,49%	0,42%	5,14%
	1500	3	Precisión	50,07%	53,97%	53,95%
			Precisión backtesting	50,88%	52,95%	53,09%
			Retorno acumulado	6,64%	28,10%	38,70%
15	150	4	Precisión	49,72%	58,12%	56,91%
			Precisión backtesting	49,52%	50,94%	51,17%
			Retorno acumulado	2,80%	6,41%	11,19%
	750	5	Precisión	50,06%	54,23%	53,96%
			Precisión backtesting	48,46%	52,05%	53,84%
			Retorno acumulado	-11,14%	9,51%	28,93%
	1500	6	Precisión	50,10%	53,95%	53,92%
			Precisión backtesting	50,76%	54,25%	53,29%
			Retorno acumulado	12,82%	39,92%	24,64%

Figura 2: Resultados aplicación metodologías sobre USD/JPY

En la figura 2 El modelo SVM muestra las peores métricas de evaluación en los experimentos 2, 4 y 5. Mientras que, la metodología de elegir el mejor modelo presenta las mejores métricas de evaluación. A pesar de que esta metodología logra una precisión de 58,23% en el experimento 1, su retorno acumulado para el mismo es de -11,03%. La precisión de backtesting y el retorno acumulado llegan al máximo en el experimento 6, con un conjunto grande de datos de entrenamiento y la mayor ventana de entrenamiento.

Par de Divisa:			AUD/JPY			
Longitud Ventana Entrenamiento	Número Datos Entrenamiento	Número Experimento	Métrica	SVM	Mejor Modelo	Conjunto de Modelos
5	150	1	Precisión	50,00%	58,00%	56,47%
			Precisión backtesting	51,95%	50,71%	50,71%
			Retorno acumulado	49,31%	-9,08%	11,35%
	750	2	Precisión	49,98%	53,54%	53,16%
			Precisión backtesting	48,97%	52,28%	51,49%
			Retorno acumulado	-2,61%	13,53%	8,92%
	1500	3	Precisión	49,79%	52,76%	52,62%
			Precisión backtesting	49,24%	49,47%	50,67%
			Retorno acumulado	11,00%	11,76%	42,00%
15	150	4	Precisión	50,47%	58,27%	56,78%
			Precisión backtesting	52,18%	50,02%	50,62%
			Retorno acumulado	26,33%	2,01%	7,89%
	750	5	Precisión	49,75%	53,60%	53,20%
			Precisión backtesting	47,63%	52,05%	52,78%
			Retorno acumulado	-26,04%	13,14%	46,86%
	1500	6	Precisión	49,83%	52,79%	52,72%
			Precisión backtesting	49,98%	50,39%	49,93%
			Retorno acumulado	-2,10%	12,99%	11,87%

Figura 3: Resultados aplicación metodologías sobre AUD/JPY

La aplicación del modelo SVM presenta los peores resultados para todas las métricas en el experimento 5, como se puede observar en la figura 3. Sin embargo, en esta ocasión el modelo SVM tiene el máximo valor de retorno acumu-

lado de todos los experimentos analizados en todos los mercados con un 49,31 % en el experimento 1. Por otro lado, las metodologías de escoger el mejor modelo y utilizar un conjunto de modelos se reparten la mejor precisión y precisión backtesting, respectivamente. Cabe destacar que la metodología del conjunto de modelos tiene retornos acumulados por encima del 40 % en dos experimentos.

VI. DISCUSIÓN

Los resultados sugieren que las metodologías que trabajan con varios modelos para realizar la predicción tienen una capacidad para generar mejores predicciones que el modelo SVM. En general, se observó que el modelo SVM presentaba las peores métricas de precisión, precisión backtesting y retorno acumulado. Además, el uso de solo este modelo hacía que se magnifique el efecto de la pandemia sobre el retorno acumulado. Las otras dos metodologías por su diseño podían mitigar el cambio estructural y volatilidad durante el inicio de la pandemia.

A pesar de que las métricas de precisión mostraron resultados bajos, estos concuerdan con lo reportado y visto en el estado del arte. Las series temporales financieras son difíciles de predecir, y aún más las correspondientes a los mercados Forex donde existe tanta volatilidad e inestabilidad. Así mismo, el retorno acumulado para un periodo de dos años concuerda con lo reportado por los diversos autores que han contribuido a la literatura.

VII. CONCLUSIONES

En la presente investigación se realiza una comparación de distintas metodologías para predecir el comportamiento del alza o baja del precio de varios pares de divisas del mercado Forex. Esto con el motivo de generar evidencia empírica en contra de la hipótesis de mercados eficientes, la cual señala que no es posible realizar predicciones consistentes a un medio o largo plazo de los mercados financieros y, por ende, generar rentabilidad de ellos. El estudio

se enfoca en analizar distintas formas de aplicar modelos de aprendizaje automático para resolver el problema de pronosticar las variaciones de los precios. Las variables que tienen a su disposición los modelos son un conjunto de indicadores técnicos que sirven como referencia para analizar el mercado.

En general, se observa que, las tres metodologías comparadas tuvieron métricas de evaluación acorde a lo que está en el estado del arte. Sin embargo, cada una de ellas, bajo determinadas condiciones, tuvieron la capacidad de generar un rendimiento financiero positivo significativo. Si se observa un top 10 de los retornos acumulados de las distintas metodologías por los diferentes escenarios, este supera ampliamente al retorno que se podría obtener por el uso del sistema financiero tradicional (i.e, alrededor del 10 % de interés como máximo). Por tanto, los resultados de este trabajo contribuyen a reforzar al aprendizaje automático como una herramienta para predicción de series temporales financieras, generando evidencia empírica en contra de la hipótesis de mercados eficientes.

El uso de herramientas de aprendizaje automático automatizado sirve para darle un mayor poder a las técnicas convencionales de aprendizaje automático. Hasta lo que se conoce, no se han realizado estudios de la aplicación de estos métodos para la predicción de series temporales financieras. En mercados tan volátiles, la automatización del flujo de trabajo de ajuste, comparación, elección y optimización de modelos de aprendizaje automático puede ser clave para generar mejores y más consistentes resultados, como se evidencia en el trabajo.

Referencias

- [1] Jin Li and Edward PK Tsang. Investment decision making using fgp: a case study. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 2, pages 1253–1259. IEEE, 1999.
- [2] Ma Di and Er Meng Joo. A survey of machine learning in wireless sensor networks

- from networking and application perspectives. In *2007 6th international conference on information, communications & signal processing*, pages 1–5. IEEE, 2007.
- [3] Thuy TT Nguyen and Grenville Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE communications surveys & tutorials*, 10(4):56–76, 2008.
- [4] Miles N Wernick, Yongyi Yang, Jovan G Brankov, Grigori Yourganov, and Stephen C Strother. Machine learning in medical imaging. *IEEE signal processing magazine*, 27(4):25–38, 2010.
- [5] Matthew Freed and Jeonghwa Lee. Application of support vector machines to the classification of galaxy morphologies. In *2013 International Conference on Computational and Information Sciences*, pages 322–325. IEEE, 2013.
- [6] Min Qi and G Peter Zhang. Trend time-series modeling and forecasting with neural networks. *IEEE Transactions on neural networks*, 19(5):808–816, 2008.
- [7] Eduardo A Gerlein, Martin McGinnity, Ammar Belatreche, and Sonya Coleman. Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, 54:193–207, 2016.
- [8] M Maggini, CL Giles, and B Horne. Financial time series forecasting using k-nearest neighbors classification. *Nonlinear Financial Forecasting: Proceedings of the First INFFC*, pages 169–181, 1997.
- [9] Wun-Hua Chen, Jen-Ying Shih, and Soushan Wu. Comparison of support-vector machines and back propagation neural networks in forecasting the six major asian stock markets. *International Journal of Electronic Finance*, 1(1):49–67, 2006.
- [10] Ming Hao Eng, Yang Li, Qing-Guo Wang, and Tong Heng Lee. Forecast forex with ann using fundamental data. In *2008 International Conference on Information Management, Innovation Management and Industrial Engineering*, volume 1, pages 279–282. IEEE, 2008.
- [11] Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319, 2003.
- [12] Jae Won Lee, Jonghun Park, O Jangmin, Jongwoo Lee, and Euyseok Hong. A multiagent approach to q -learning for daily stock trading. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6):864–877, 2007.
- [13] Sheng-Tun Li and Shu-Ching Kuo. Knowledge discovery in financial investment for forecasting and trading strategy through wavelet-based som networks. *Expert Systems with applications*, 34(2):935–951, 2008.
- [14] Paolo Tenti. Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence*, 10(6):567–582, 1996.
- [15] Rui Pedro Barbosa and Orlando Belo. Autonomous forex trading agents. In *Industrial Conference on Data Mining*, pages 389–403. Springer, 2008.
- [16] Rui Pedro Barbosa and Orlando Belo. The agent-based hedge fund. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 449–452. IEEE, 2010.
- [17] Rui Pedro Barbosa and Orlando Belo. Algorithmic trading using intelligent agents. In *IC-AI*, pages 136–142, 2008.