PROYECTO INTEGRADOR SEMESTRE II:

MODELO DE RECOMENDACIONES SOBRE EL PRECIO FUTURO DE UNA DIVISA

PRESENTADO POR:

Álvaro Villa Vélez

Luis Rodrigo Vesga Vesga

Jorge Luis Rentería Roa

Edgar Leandro Jiménez Jaimes

Santiago Echeverri Calderón

UNIVERSIDAD EAFIT

MEDELLÍN

MAESTRÍA EN CIENCIAS DE LOS DATOS Y ANALÍTICA

MAYO DE 2020

Tabla de contenido

1. Introducción

En la última década, los sistemas transaccionales del mercado de capitales han permitido que se generen una gran cantidad de datos financieros que han crecido de una forma sin precedentes. Este rápido crecimiento del volumen de datos representa un reto para analizarlos e interpretarlos. Con el presente proyecto se busca desarrollar un enfoque automatizado para un análisis eficiente de estos datos financieros, que involucran información del análisis técnico y el análisis fundamental como insumo para un modelo de Machine Learning. La predicción de decisiones se articulará como un problema de clasificación que proporcione recomendaciones sobre el comportamiento positivo o negativo de una divisa en el futuro.

2. Marco teórico

Malkiel y Fama (1970) afirman en su Journal [Efficient Capital Markets: A Review of Theory and Empirical Work] que en general, los precios actuales reflejan toda la información relevante disponible al fijar el precio de un activo financiero. La hipótesis de ellos surge de observaciones empíricas de cambios en series temporales de precios que son muy similares a un proceso de recorrido aleatorio. Según estos autores, incluso un sistema en el que se generan varias ordenes de compra y venta a corto plazo no es rentable, debido a los costos de transacción y las comisiones. Desde entonces, los trabajos académicos han tratado de mostrar que los precios del mercado de valores son, hasta cierto punto, predecibles. Malkiel (2003, p. 80) [The Efficient Market Hypothesis and Its Critics] concluye que no todos los participantes del mercado son racionales y que hay formaciones de precios irregulares, lo que lleva a patrones de rendimiento explotables en períodos cortos de tiempo.

En este trabajo se busca considerar la posibilidad de que pueda existir un sistema predictivo, el desarrollo de sistemas consistentemente rentables puede constituir evidencia contra la hipótesis de los mercados eficientes (HME). Dichos sistemas pueden beneficiarse de técnicas computacionalmente intensivas, como las que explotan los algoritmos de aprendizaje automático.

3. Contexto y objetivo

Este proyecto se enfocará en predecir el posible comportamiento futuro del peso colombiano, de ahora en adelante se le llamará USDCOD, mediante técnicas de machine learning e intentaremos generar un modelo que recomendará si el precio del USDCOP subirá o bajará en un horizonte de tiempo dado.

Como insumo para el modelo trabajaremos con información histórica de datos para análisis fundamental y también con datos de análisis técnico.

El análisis fundamental consiste en que el precio de un activo financiero es explicado por variables financieras y macroeconómicas como: PIB, Inflación, Desempleo, etc. Como hay una gran cantidad de variables buscaremos cuales son las que mejor predicen USDCOP y miraremos si estos modelos tienen algún poder explicativo del precio futuro o sirve como variable de insumo para el modelo de clasificación de machine learning.

El análisis técnico es un sistema que permite examinar y predecir los movimientos de precios en los mercados financieros a partir de datos históricos (Precio de apertura, Precio de Cierre, Precio mínimo y Precio máximo) y estadísticas de mercado. Se basa en la idea de que, si un inversor puede identificar patrones previos, entonces podrá predecir los movimientos futuros de los precios de manera bastante exacta.

Existen muchos tipos de indicadores calculados en función de las diferentes variables características del comportamiento de los valores analizados, en este trabajo nos enfocaremos en los siguientes indicadores de análisis técnico: Overlap Studies, Momentum Indicators, Volatility Indicators, Price Transform, Cycle Indicators y Pattern Recognition. Cada indicador utiliza un enfoque ligeramente diferente y tiene su propia fórmula. Por ejemplo, los indicadores técnicos de momentum (Momentum Indicators) miden el monto que ha cambiado el precio de un activo durante un período de tiempo determinado y tratan de identificar la tendencia o la falta de una tendencia. Saber cuándo comienza una tendencia y cuándo termina es una información extremadamente útil para el trader. Las fórmulas de los indicadores comparan el precio de cierre, con un precio de cierre anterior o con un precio máximo o mínimo según el plazo e indicador utilizado.

4. Fuente de datos

Para el desarrollo del modelo se usará información de precios de las divisas e información macroeconómica disponible en Bloomberg.

Como datos fundamentales utilizamos **77** variables, la cuales se dividen en variables macroeconómicas, variables de mercado y variables de indicadores lideres. Todas estas variables tienen alguna relación macroeconómica y/o financiera con el comportamiento del Peso Colombiano. Por ejemplo, precio del petróleo, Exportaciones e Importaciones de Colombia, etc.

A continuación, se presentan algunas de las variables usadas:

Ticket de Bloomberg	Nombre Variable
COTREXPM INDEX	Colombia Trade Balance Exports
COTRIMPM INDEX	Colombia Trade Balance Imports
2331Q001 Index	IMF Colombia Total Reserves
COEXTOTL INDEX	Colombia External Debt
EHCACO Index	Colombia Current Account Balan
COBPFD INDEX	Colombia Financial Account Dir
EHBBCO Index	Colombia Budget Balance (% GDP
COCIPIBQ Index	Colombia GDP Constant Prices S
CL1 Comdty	CL1 COMB
USTBEXP INDEX	US Trade Balance of Exports SA
GDP CUR$ Index	GDP US Nominal Dollars SAAR
USURTOT Index	U-3 US Unemployment Rate Total
EHBBUS Index	US Budget Balance (% GDP)
EMPRGBCI Index	US Empire State Manufacturing
OUTFGAF Index	Philadelphia Fed Business Outl
CESIUSD Index	Citi Economic Surprise - Unite
JFRIUS Index	JPMorgan Forecast Revision Ind
CPI CHNG Index	US CPI Urban Consumers MoM SA
CPUPXCHG Index	US CPI Urban Consumers Less Fo
CPI YOY Index	US CPI Urban Consumers YoY NSA
CPI XYOY Index	US CPI Urban Consumers Less Fo
USGG2Y Index	US Generic Govt 2 Yr
USGG10Y Index	US Generic Govt 10 Yr
MWTVWT Index	CPB Merchandise: World Trade V
KOEXTOT Index	South Korea Exports
NAPMEXPT Index	ISM Manufacturing Report on Bu
CNRSACMY Index	China Retail Sales Cumulative
CHVAIOY Index	China Value Added of Industry
CHVAICY Index	China Value Added of Industry
CHLR12MC INDEX	China 1 Year Benchmark Lending

Para el cálculo de las variables de análisis técnico utilizaremos solo información intradía de USDCOP, las cuales son Precio de Apertura, Precio de Cierre, Precio mínimo y Precio máximo.

USDCOP Curncy

Dates	PX_LAST	PX_OPEN	PX_HIGH	PX_LOW
30/12/2005	$ 2,287	$ 2,287	$ 2,287	$ 2,287
2/01/2006	$ 2,282	$ 2,290	$ 2,290	$ 2,282
3/01/2006	$ 2,282	$ 2,287	$ 2,287	$ 2,281
4/01/2006	$ 2,282	$ 2,282	$ 2,283	$ 2,281
5/01/2006	$ 2,278	$ 2,282	$ 2,282	$ 2,277
6/01/2006	$ 2,278	$ 2,278	$ 2,284	$ 2,277
9/01/2006	$ 2,278	$ 2,279	$ 2,279	$ 2,278
10/01/2006	$ 2,276	$ 2,278	$ 2,278	$ 2,276
11/01/2006	$ 2,274	$ 2,276	$ 2,278	$ 2,273
12/01/2006	$ 2,276	$ 2,274	$ 2,276	$ 2,273
13/01/2006	$ 2,271	$ 2,276	$ 2,276	$ 2,271
16/01/2006	$ 2,274	$ 2,271	$ 2,273	$ 2,271
17/01/2006	$ 2,271	$ 2,275	$ 2,275	$ 2,269

Con los precios anteriores calcularemos todos los indicadores de análisis técnicos que posee la librería TA-Lib de Python (Overlap Studies, Momentum Indicators, Volatility Indicators, Price Transform, Cycle Indicators y Pattern Recognition).

Un ejemplo del dataframe con algunas de las variables de indicadores de análisis técnicos utilizados, es el siguiente:

ADX	ADXR	APO	AROONOSC	BOP	MOM	WILLR	CMO	DX	MINUS_DI	MINUS_DM	PLUS_DI
33.145890	42.561962	0.111526	85.714286	0.913793	0.4950	-0.977199	39.058495	24.714098	16.889881	0.255515	27.978761
32.967977	42.091217	0.131423	92.857143	0.337500	0.5770	-5.089820	44.048113	30.655109	15.979847	0.237264	30.108183
33.223610	41.620839	0.156865	100.000000	0.822581	0.4670	-2.572899	48.964741	36.546839	14.661232	0.220316	31.549964
33.793182	41.350160	0.182647	71.428571	0.567010	0.5010	-1.735016	52.823776	41.197623	13.708297	0.204580	32.916678
34.335032	41.184889	0.206122	78.571429	-0.532258	0.5370	-6.918239	52.973949	41.379078	13.121250	0.189967	31.645190
...	...	...	...	...	...	...	...	...	...	...	...
40.508930	41.707763	-0.048810	28.571429	-0.043367	-0.0195	-64.281753	-28.359084	39.845532	32.565329	0.253095	14.007956
40.924628	42.885245	-0.045502	28.571429	-0.778796	0.0084	-94.290560	-29.117152	46.328705	34.535071	0.275617	12.666975
41.943781	43.646211	-0.047533	-78.571429	-0.470383	-0.0559	-85.615603	-40.902425	55.192763	39.306179	0.325130	11.348476
43.466195	44.077540	-0.052829	-85.714286	-0.667497	-0.2286	-93.966817	-53.628906	63.257584	43.571969	0.387306	9.806218
45.056022	44.298575	-0.055172	-92.857143	-0.613445	-0.3237	-97.714758	-55.528747	65.723769	42.509559	0.391342	8.792145

5. Modelo de análisis fundamental

5.1. Descripción

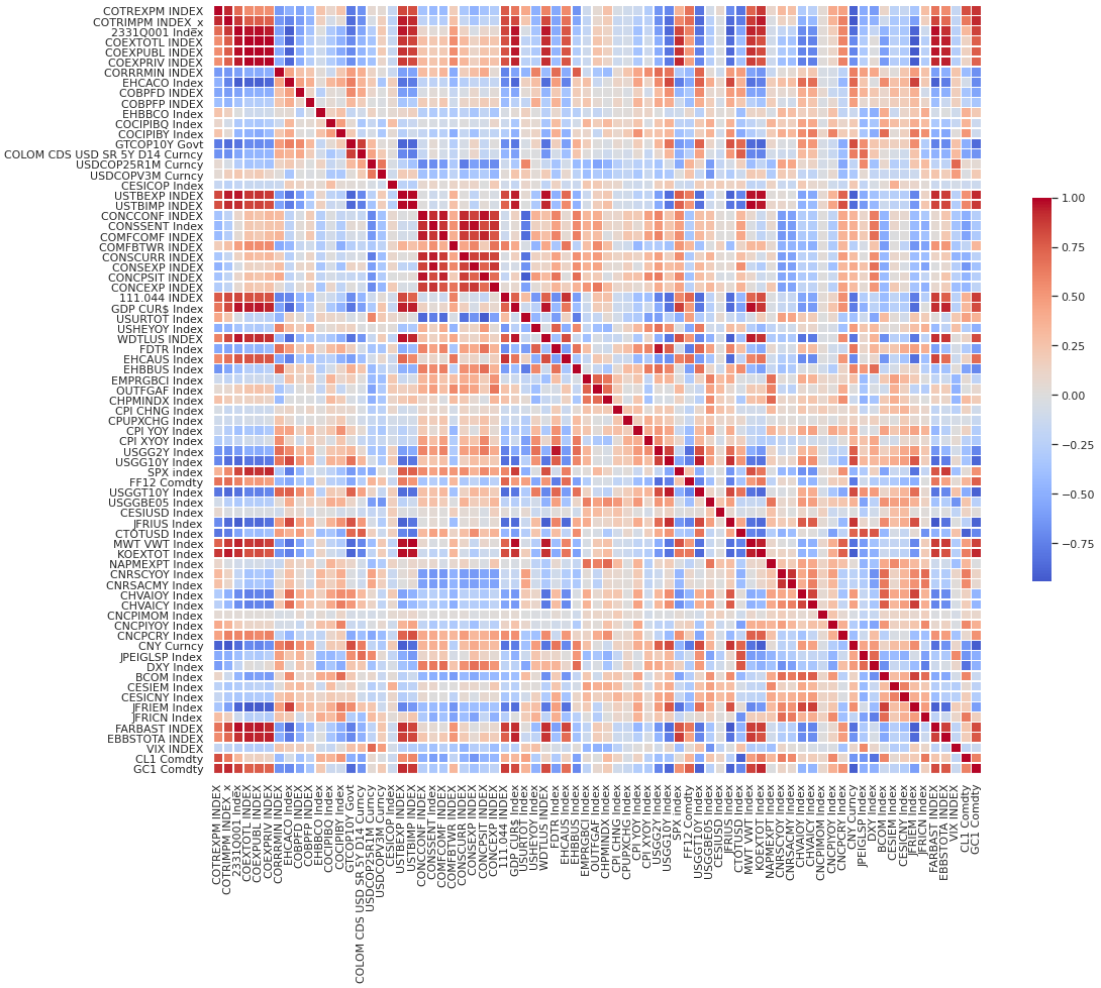
El análisis fundamental estudia factores macroeconómicos con el fin de conocer su influencia en los precios de un activo financiero. La premisa principal de este análisis es que el precio de un activo puede diferir de su valor de mercado, en otras palabras, que el precio de mercado puede estar en ocasiones sobrevalorado o subvalorado. Bajo esta premisa también se espera que el precio de mercado tenderá a acercarse a su valor autentico o fundamental (Fair Value).

En el modelo de recomendación, el análisis fundamental se incluirá con el fin de proporcionarle al clasificador información externa sobre la economía y el mercado. Para esto se desarrollará un modelo previo que permita estimar el precio fundamental de la divisa. La desviación del precio fundamental vs el precio de mercado será un nuevo atributo que se usará como variable predictora en el modelo de clasificación final.

5.2. Datos y preprocesamiento

Para entrenar el modelo de precio fundamental se recolectaron de Bloomberg 77 variables de los últimos 14 años, estas variables incluyen información macroeconómica de Colombia y Estados Unidos, información del mercado de ambas divisas (COP y USD) e índices cambiarios de diferentes mercados a nivel global. La variable de predicción será el precio de la divisa USDCOP.

Inicialmente se realizó un análisis de la correlación entre pares de variables predictoras, en el cual se identificaron variables altamente correlacionadas como puede observarse en la siguiente matriz en donde las intersecciones con rojo y azul más intenso son las correlaciones más altas:



Con el fin de evitar problemas de multicolinealidad se realizó una selección de variables, eliminando las que tuvieran un coeficiente de correlación con otra variable mayor que 0.9. En esta selección se procuró conservar las variables que tuvieran una periodicidad diaria y que tuvieran mayor correlación con la variable de predicción. Después de este proceso el conjunto de datos se redujo a 56 variables.

Para obtener el precio fundamental se exploraron y compararon 2 modelos de regresión, un modelo básico y de poca varianza (una regresión lineal), y un modelo de machine learning de mayor complejidad y varianza (un regresor Random Forest).

El conjunto de datos se dividió en dos, una porción para entrenamiento para lo cual se tomaron los datos entre abril de 2006 y junio del 2019; y otra porción como conjunto de pruebas con la información comprendida entre octubre de 2019 y febrero de 2020.

Para comparar los modelos se usaron el coeficiente de determinación (R^2) y la raíz del error cuadrático medio (RMSE en inglés), buscando así medir la proporción de variación de los resultados que puede explicar el modelo y el error medio en COP de las predicciones con cada una de las métricas respectivamente.

Es importante mencionar que antes de entrenar los modelos las variables se escalaron centrando los datos con respecto a la media y dividiéndolos entre la desviación estándar. También que el modelo Random Forest se optimizó buscando el número de estimadores y la profundidad de los árboles con los que se obtuviera el mejor RMSE sobre una porción de validación del conjunto de entrenamiento.

5.3. Modelos y resultados

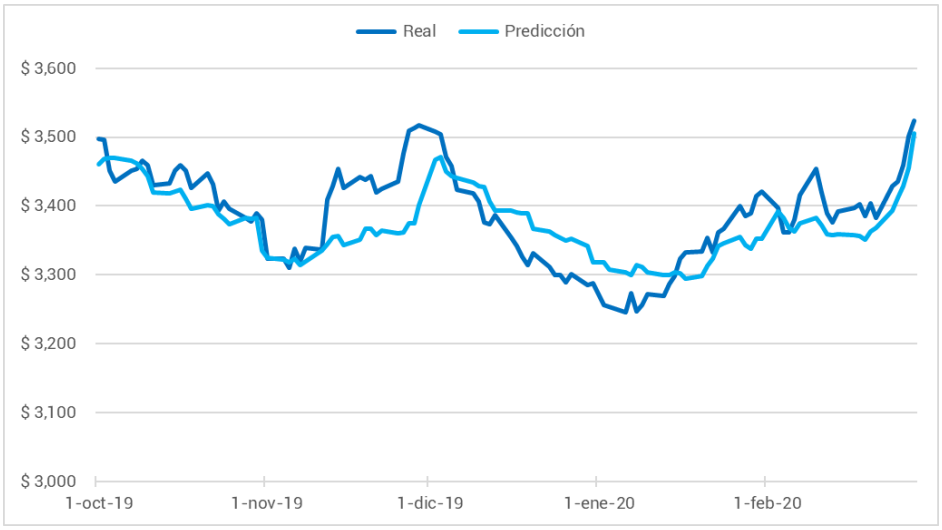
Los resultados fueron los siguientes:

Regresión lineal		**Regresión Random Forest**	
Entrenamiento	**Pruebas**	**Entrenamiento**	**Pruebas**
$R^2 = 0.91$ $RMSE = 49$	$R^2 = 0.52$ $RMSE = 47.8$	$R^2 = 0.99$ $RMSE = 7.3$	$R^2 = 0$ $RMSE = 206.4$

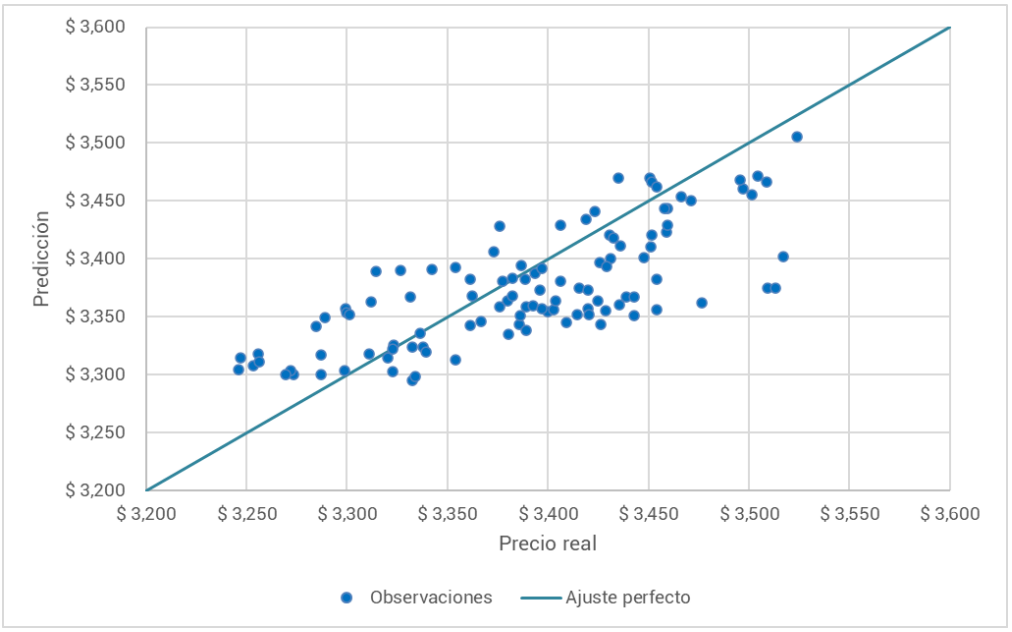
Se puede observar que, aunque ambos modelos son capaces de ajustarse muy bien a los datos de entrenamiento, la regresión lineal logra hacer una mejor generalización y tiene un error medio de COP$47 frente a COP$206 del random forest en los datos de prueba. El random forest al ser un modelo de mayor varianza se sobreajusta a los datos de entrenamiento y no tiene un buen desempeño con datos no vistos, mientras que la regresión lineal hace una mejor caracterización general de las variables y logra tener una mejor capacidad predictiva con datos nuevos. Por este motivo se seleccionó el modelo de regresión lineal para el cálculo del precio fundamental.

A continuación, se presentan 2 visualizaciones de las predicciones del modelo de regresión lineal en los datos de prueba:

Precio USDCOP (Real vs Predicción del modelo)



Precio USDCOP (Real vs Predicción del modelo)



En ambas gráficas se puede observar que las predicciones siguen la tendencia de los datos reales. En la primera gráfica es importante aclarar que la fecha no es una variable del modelo, las fechas se añadieron a los datos para efectos de la gráfica, pues en la modelación no se buscó ningún tipo de relación temporal entre los datos.

Después de obtener este modelo se calculó la predicción para cada uno de los días desde 2006 hasta febrero de 2020 (es decir el precio fundamental de cada día). Este precio se comparó con el precio de cierre del mercado de cada día y se calculó la diferencia entre ambos precios expresada en desviaciones

estándar. La desviación estándar que se usó para el cálculo fue la del precio de cierre del mercado en el periodo que se tomó para el conjunto de entrenamiento del modelo.

Estos valores resultantes se usaron entonces como un nuevo atributo en el modelo de análisis técnico.

6. Modelo de análisis técnico

6.1. Descripción

La mayoría de los agentes (traders) suelen tomar decisiones basadas en un gran cúmulo de variables que pueden comprender análisis técnico (indicadores y chartismos), y fundamental (información macroeconómica). En el presente modelo se pretende hacer uso de métodos avanzados de aprendizaje automático para estimar el comportamiento del USDCOP en un periodo futuro y poder recomendar una de tres decisiones (Comprar, Vender, No operar). Encontrar este periodo futuro o de predicción será uno de los objetivos de esta etapa, es decir, a cuantos días a futuro se hará la predicción.

Como se mencionó anteriormente la predicción se articulará como un problema de clasificación, en el que se usarán 3 clases para determinar el comportamiento de la divisa: *sube*, *se mantiene estable* o *baja*.

Para etiquetar los datos con estas clases se halló la variación del precio durante el día, mediante la comparación del precio de cierre día actual (t_0) y el precio de cierre del día futuro (t_p, donde p es el periodo de predicción) con la siguiente función:

$$\text{variación} = \frac{\text{precio } t_0}{\text{precio } t_p} - 1$$

Después de hallada esa variación del precio, se colocaron las etiquetas así:

Clase	Acción	Descripción	
0	Estable	Si el precio se mantiene estable	Si $-0.05\% \leq$ variación $\leq 0.05\%$
1	Sube	Si el precio va a subir	Si variación $> 0.05\%$
-1	Baja	Si el precio va a caer	Si variación $< -0.05\%$

Este umbral de decisión (0.05%) se seleccionó después de realizar un proceso iterativo modificando el umbral y evaluando las métricas de desempeño en un modelo base de clasificación.

Posteriormente desplazamos hacia arriba la columna de las etiquetas. La idea de subir las etiquetas es con el sentido de que el dataset final quede los indicadores técnicos (X) del día y la etiqueta (Y) corresponda a lo que pasó con el precio en el futuro.

Ejemplo: si el periodo de predicción son 15 días la variable Y se desplazará 15 días.

Periodo	Precio (Y)	Precio (Y lagged)	Variables
0	y_0	y_{0+15}	$x_1, x_2, \cdots x_n$
1	y_1	y_{1+15}	$x_1, x_2, \cdots x_n$
⋮	⋮	⋮	⋮
n	y_n		$x_1, x_2, \cdots x_n$

Los análisis técnicos son interpretaciones realizadas sobre indicadores calculados con base en el precio de cierre, precio de apertura, precio mínimo y precio máximo de la moneda en un periodo específico de tiempo. Esta categoría de análisis puede clasificarse en 8 subcategorías con enfoque diferente las cuáles representaran un total de 95 columnas:

- **Momentum**: Buscan identificar la tendencia del mercado basado en los valores máximos y mínimos del mercado.
- **Transformaciones de precio:** En esta subcategoría se analizan indicadores calculados sobre los precios. Aquí se estudia el precio promedio, precio mediano y precio típico de la moneda.
- **Ciclo:** Son indicadores que estudian si el comportamiento del activo responde a un ciclo repetitivo o es una nueva tendencia.
- **Volatilidad:** Calculan el grado de la volatilidad de los precios. No proporcionan información relevante acerca de la tendencia.
- **Overlap:** Estos indicadores analizan cuan normales son los movimientos actuales del precio de cierre. En otras palabras, si el comportamiento actual está dentro de las bandas típicas de comportamiento.

Además del análisis técnico, se calcularon 63 indicadores que buscan representar las decisiones en los mercados financieros tomadas utilizando como referencia el comportamiento "gráfico" de los precios (Chartismo). En este tipo de análisis se busca reconocer patrones para anticipar cuál será el comportamiento futuro de la moneda.

6.2. Librerías y funciones

Librerías usadas en el código

Las siguientes librerías fueran usadas en el modelo:

- Pandas → Manipular dataframes en python
- Numpy → Cálculos matemáticos y algebra lineal
- Random → Generar aleatorios
- Drive (Google Colab) → Cargar archivos de Drive
- Plt (Matplotlib) → Graficar entre líneas del notebook
- Datetime → manipular *TimeStamps*
- Math → Realizar redondeos
- LinearRegresion → Ajustar regresión lineal a los datos

- RandomForestClassifier → Construir arboles de decisiones
- SVC → Aplicar máquinas de soporte vectorial para clasificar las clases
- LinearSVC → Aplicar máquinas de soporte vectorial para clasificar las clases
- GradientBoostingClassifier -> Construir un clasificador empleando métodos del gradiente
- VotingClassifier → Ponderar modelos previos y construir uno robusto
- F1_Score → Métrica objetivo
- Classification_report → Resumen del desempeño del modelo
- Confusion_mtatrix → Resumen de la precisión del modelo
- Train_test_split → Participación de daos en conjuntos de prueba y entrenamiento.
- Talib → Librería de indicadores técnicos

Funciones propias

A continuación, se presenta una breve documentación y descripción de las funciones que se crearon para en la implementación del modelo:

Nombre	Descripción	Descripción argumento	Tipo retorno
Leer	Carga el archivo de Excel proveniente de bloomberg donde está el histórico de la moneda, estandariza la fecha y omite las filas vacías en el encabezado.	PathDF → Ruta madre donde se encuentran los archivos. DFname → Nombre de archivo V_name → Nombre de la moneda a analizar (USDCOP) Skip → Número de filas vacías en encabezado. Default 5	DataFrame
Llenarvacios	Asigna los datos del día anterior a los registros faltantes (días no hábiles), además, agrupa la información con base a la periodicidad deseada.	df → DataFrame con histórico de moneda n → Días en los que se quiere agrupar la información Date_name→Nombre columna *Fecha* n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame
Momentum	Utiliza las funciones de *momemtum* de la librería **TALIB** para construir un nuevo DataFrame con 29 nuevas columnas.	df → DataFrame con histórico de moneda n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame
PriceTransformF	Utiliza las funciones de *PriceTransform* de la librería **TALIB** para construir un nuevo DataFrame con 4 nuevas columnas.	df → DataFrame con histórico de moneda n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame
CycleIndicator	Utiliza las funciones de *CycleIndicator* de la librería **TALIB** para construir un nuevo DataFrame con 5 nuevas columnas	df → DataFrame con histórico de moneda n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame
PatternRecognition	Utiliza las funciones de *PatternRecognition* de la librería **TALIB** para construir un nuevo DataFrame con 63 nuevas columnas	df → DataFrame con histórico de moneda n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame
StatisticsFuncitions	Utiliza las funciones de *StatisticsFunctions* de la librería **TALIB** para construir un nuevo DataFrame con 9 nuevas columnas	df → DataFrame con histórico de moneda n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame

Nombre	Descripción	Descripción argumento	Tipo retorno
MathTransform	Utiliza las funciones de *MathTransform* de la librería **TALIB** para construir un nuevo DataFrame con 15 nuevas columnas	df → DataFrame con histórico de moneda n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame
MathOperator	Utiliza las funciones de *MathOperator* de la librería **TALIB** para construir un nuevo DataFrame con 13 nuevas columnas	df → DataFrame con histórico de moneda n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame
Volatility	Utiliza las funciones de *Volatility* de la librería **TALIB** para construir un nuevo DataFrame con 3 nuevas columnas	df → DataFrame con histórico de moneda n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame
Overlap	Utiliza las funciones de *Overlap* de la librería **TALIB** para construir un nuevo DataFrame con 16 nuevas columnas	df → DataFrame con histórico de moneda n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame
bigdata	Consolida en un DataFrame todas las nuevas columnas creadas, además, crea las clases *-1,0,1* dependiendo de la variación que se tuvo con base al día siguiente.	df → DataFrame con histórico de moneda n_close → Nombre columna *PX_LAST* n_high → Nombre columna *PX_HIGH* n_low → Nombre columna *PX_LOW* n_open → Nombre columna *PX_OPEN*	DataFrame

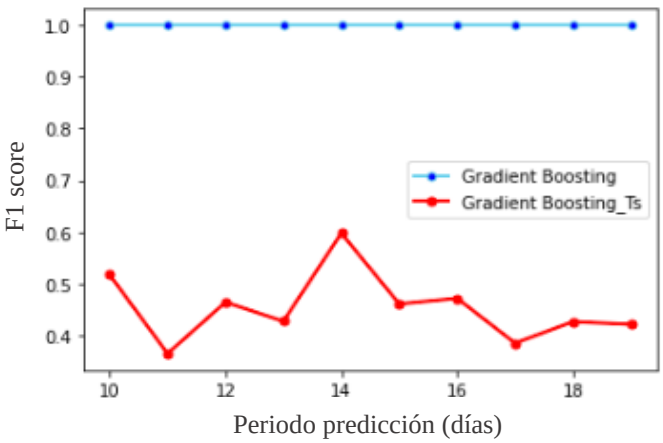
6.3. Modelos y resultados

Buscaremos el clasificador con mejor F1 score iterando el periodo de predicción entre 1 y 20 días.

Gradient Boosting

Gradient Boosting es un método de aprendizaje de máquinas fundamentado en el supuesto que un modelo futuro al ser combinado con un modelo anterior minimiza los errores totales del conjunto de datos. La idea principal es calcular los impactos sobre el error del modelo anterior y los utiliza como valor objetivo. Los resultados objetivo para cada caso se establecen en función del error del gradiente con respecto a la predicción. Cada nuevo modelo da un paso en la dirección que minimiza el error de predicción, en el espacio de las posibles predicciones para cada caso de entrenamiento.

Se implementaron cerca de 300 Gradient Boosting optimizando cada uno de ellos por el número de estimadores, la profundidad máxima y el número de días a agrupar en el conjunto de datos. Para comparar el desempeño de cada uno de los modelos se empleó la métrica F1 Score teniendo en cuenta que es importante la participación de todas las clases en la recomendación del modelo.

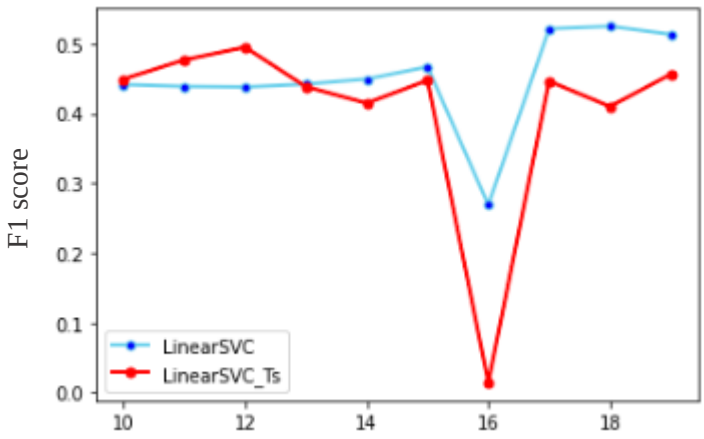


Se logra una aproximación bastante buena en el modelo de mejor desempeño *Gradient Bosting Classifier* tras realizar mejoras en las recomendaciones. En términos económicos durante el periodo de testeo se tuvo un desempeño cercano al 60%.

	precision	recall	f1-score	support
-1	0.52	0.88	0.66	25
0	0.00	0.00	0.00	1
1	0.75	0.32	0.45	28
accuracy			0.57	54
macro avg	0.42	0.40	0.37	54
weighted avg	0.63	0.57	0.54	54

Support Vector Machines

Las máquinas de soporte vectorial buscan una línea (o hiperplano) que separe dos clases a través de los vectores de soporte (Son los puntos más cercanos de diferente clase). Lo que busca este algoritmo es maximizar la distancia entre esos puntos, es decir, el margen.

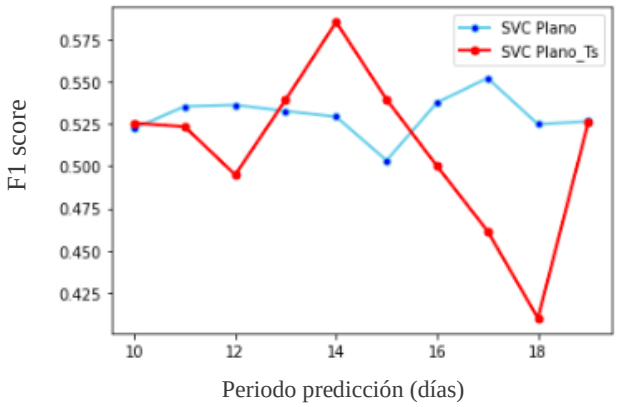


Dado el gran volumen de reglas que se deben considerar para dar recomendaciones acertadas, el problema actual presenta un escenario no clasificable con separaciones lineales. Por ello, se pretenden hacer transformaciones sobre los kernels buscando calcular los datos en otras dimensiones y encontrar un hiperplano que se ajuste a las clases.

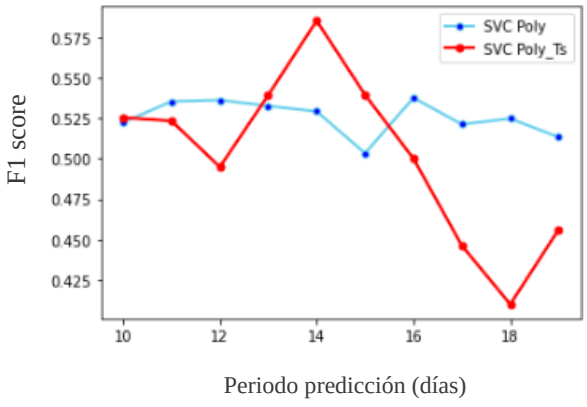
Transformaciones del kernels

El análisis de los datos en otros espacios dimensionales permite encontrar subconjuntos capaces de encontrar la decisión que divida de manera clara las diferentes clases. Este análisis suele ser muy costo en materia computacional pues aumenta drásticamente la dimensionalidad de los datos a estudiar, por ello, se aplica el método “truco del kernel” el cual permite realizar esos cálculos multidimensionales sobre el espacio inicial.

Gaussian Kernel



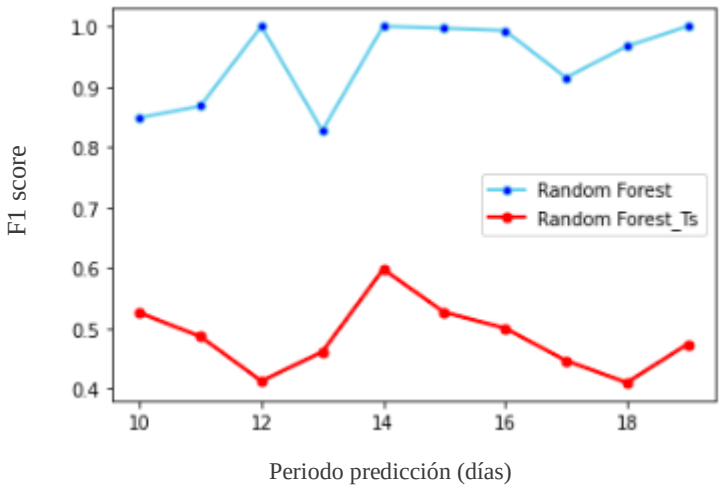
Polinomial Kernel



Aún después de evaluar nuevos espacios, utilizando los puntos de soporte no se encuentra un plano capaz de distinguir correctamente entre clases, esta es la razón por la cual ninguna de las máquinas de soporte vectorial logra generalizar de manera correcta el conjunto de datos de USDCOP. En otras palabras, las máquinas en el mejor de los casos (sin importar la transformación de kernel que se emplee) presentan el mejor comportamiento cuándo en todas las situaciones toman la decisión de *1*, la cuál es la clase con mayor frecuencia.

Random Forest

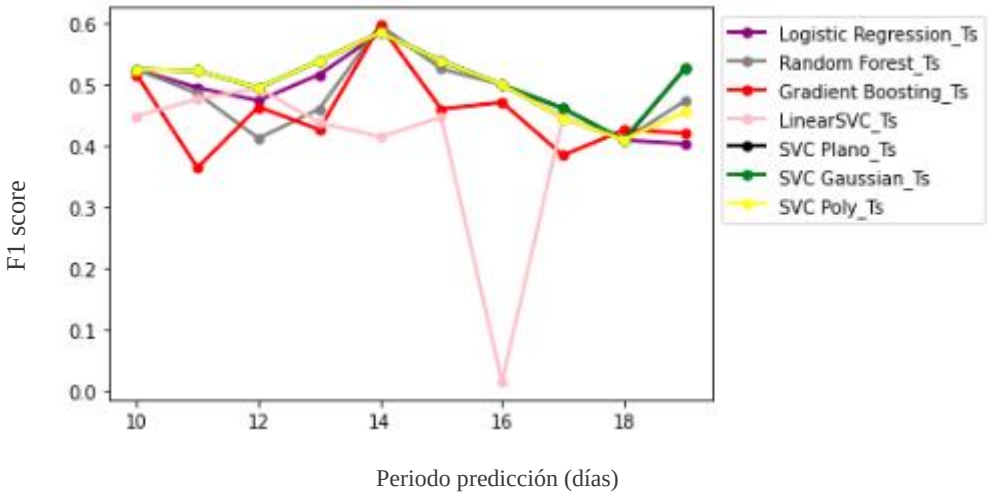
El Random Forest es un algoritmo de aprendizaje de máquinas basado en el concepto de encontrar una pregunta sobre una característica que genere la mejor partición de los datos. Su objetivo es realizar este paso repetitivamente creando ramas de conjuntos de datos a través de múltiples decisiones.



A diferencia de las máquinas de soporte vectorial, este método logra identificar más de una clase en el conjunto de datos USDCOP hasta tener un valor F1 del 59%, pese a ello, tras las iteraciones en las

agrupaciones de decisión presenta oscilaciones muy marcadas en la métrica sobre el conjunto de entrenamiento.

Comparación integral de modelos (test)



Periodo predicción (días)

Luego de evaluar todos los modelos explicados, se encontró como mejor modelo el *Gradient Boosting* con decisiones tomadas cada 14 días para los datos de USDCOP. Aunque a que todos los modelos a excepción de la máquina de soporte vectorial lineal tuvieron un buen desempeño en esta agrupación, el Gradient Boosting obtuvo cerca de un punto porcentual por encima del resto sin sacrificar precisión sobre los datos de entrenamiento.

6.4. Optimización de modelos y reducción de dimensión

Se redujo la dimensionalidad utilizando el método de Análisis Discriminante Líneal (LDA por sus siglas en inglés) con el fin de reducir el número de variables que se tenían en el conjunto de datos, el cual era de 148. Para llevar a cabo esta reducción se utilizaron los datos escalados y posteriormente se aplicó el LDA que es un método utilizado para encontrar una combinación lineal de patrones que caracterizan o separan dos o más clases de objetos o eventos. Para aplicar este método se utilizó la librería LDA de ScikitLearn la cual según su documentación dice que es un modelo que ajusta una densidad gaussiana a cada clase, suponiendo que todas las clases comparten la misma matriz de covarianza. También es usado como un clasificador con un límite de decisión lineal, generado al ajustar densidades condicionales de clase a los datos y usar la regla de Bayes, pero en este caso es empleado para reducir la dimensionalidad de los datos de entrada al proyéctalos en las direcciones más discriminatorias.

El resultado en entrenamiento del modelo Gradient Boosting después de hacer la reducción de dimensionalidad mediante LDA es el siguiente:

	precision	recall	f1-score	support
-1	0.56	0.40	0.47	25
0	0.00	0.00	0.00	1
1	0.59	0.71	0.65	28
accuracy			0.56	54
macro avg	0.38	0.37	0.37	54
weighted avg	0.56	0.56	0.55	54

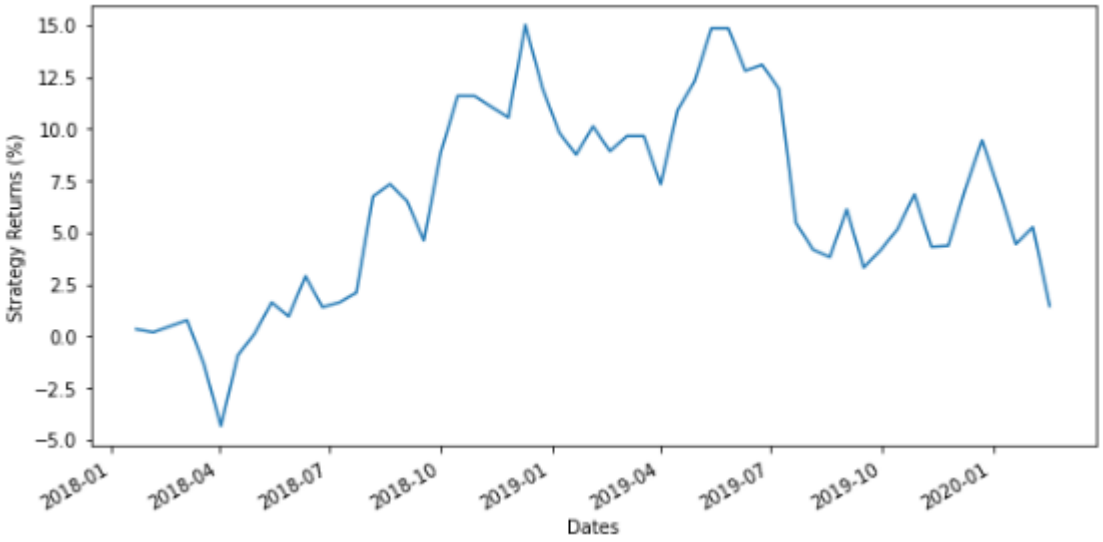
7. Backtesting

Además de medir el desempeño del modelo por medio de métricas analíticas, es importante conocer cuál sería el desempeño de sus recomendaciones en situaciones reales de operación. Para ello, se realizó un backtesting sobre el periodo que enmarca los datos de prueba, esta validación tiene por objeto estimar cuánto sería el porcentaje de retorno de la estrategia sin considerar un monto inicial invertido.

Al igual que en la mayoría de campos, en materia económica existen situaciones atípicas que ocurren en momentos específicos del tiempo las cuáles tienen impactos imprevistos sobre la variable objetivo, con base en esto, en análisis económico de las recomendaciones del modelo se divide en dos partes, la primera para estimar su comportamiento en situaciones normales del día a día y la segunda para dimensionar el impacto de hechos atípicos sobre las variables económicas como lo ha sido la pandemia del año 2020 referente a la transmisión del coronavirus.

Durante los primeros meses del 2018 las recomendaciones del modelo habrían significado una pérdida de aproximadamente el 5% con respecto al valor inicial, desde ello y hasta final de ese mismo año, el retorno presentó una tendencia creciente hasta obtener una rentabilidad de 15%. De allí se sostuvo con valores positivos de rendimiento hasta los dos primeros meses del año 2020, los cuales empezaban a mostrar señales económicas de la pandemia.

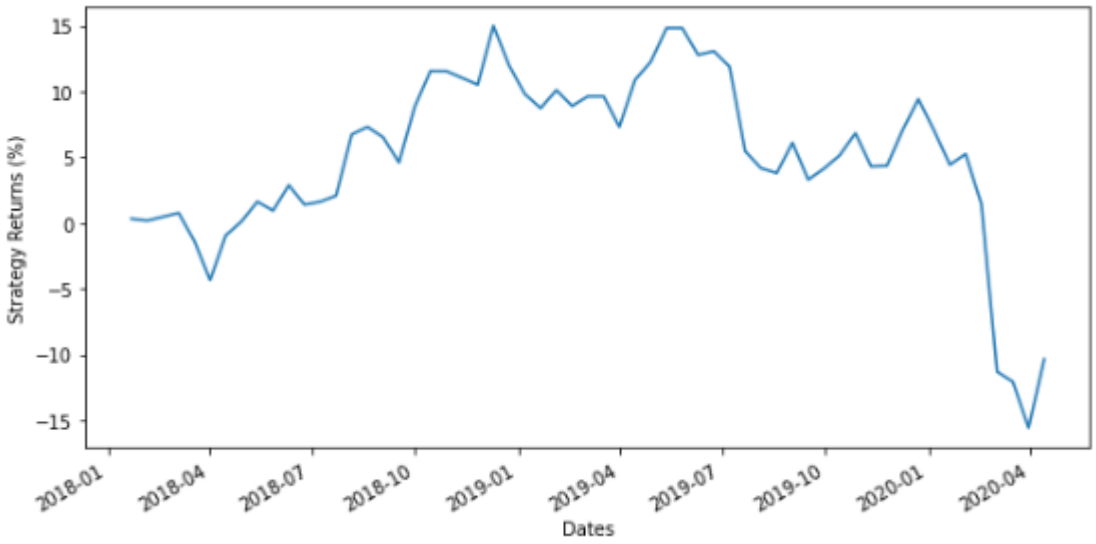
Retorno de la estrategia USDCOP (Enero 2018 – Febrero 2020)



Impacto del coronavirus 2020 sobre la rentabilidad de la estrategia de inversión.

En los primeros meses del año 2020, los mercados financieros se encontraban a la expectativa de cómo evolucionaría la epidemia del coronavirus, en el mes de marzo fue concebida como pandemia y presentó su primer caso en Colombia. Los mercados de capitales en menos de una semana reaccionaron a la situación sanitaria y todos los indicadores presentaron grandes fluctuaciones, como nuestras recomendaciones son a 14 días la estrategia de inversión no pudo reaccionar ante este evento atípico y absorbió la alta volatilidad que ocurrió en esas 2 semanas.

Retorno de la estrategia USDCOP (Enero 2018 – Mayo 8 2020)



Es importante aclarar que este es un modelo de recomendación y no trading, que se corre cada 14 días, si fuese un modelo de trading algorítmico tendríamos otros inputs en la estrategia como *stop-loss* (parar pérdidas superiores a un porcentaje definido), en este caso el modelo no hubiera incurrido en una rentabilidad tan negativa en esos 14 días.

8. Conclusiones

Con base en el comportamiento histórico del activo financiero USDCOP fue posible desarrollar un modelo de clasificación, el cual recomienda operaciones de compra o venta para periodos de 14 días basado en indicadores financieros técnicos y fundamentales.

Los métodos de clasificación empleando técnicas de aprendizaje de máquinas convencionales (como lo es Gradient Boosting) optimizados en algunos de sus parámetros, permiten obtener recomendaciones acertadas sobre cómo se comportará el mercado financiero en periodos futuros logrando niveles de accuracy del 60% aproximadamente.

Luego de evaluar todos los modelos explicados, se encontró como mejor modelo el Gradient Boosting con decisiones tomadas cada 14 días para los datos de USDCOP.

Dado el accuracy cercano al 60% si se utiliza este modelo en un horizonte grande de tiempo se espera tener una buena tasa de aciertos que son el insumo para una estrategia rentable de trading. Es importante aclarar que este no es modelo de trading algorítmico el cual debe tener otro tipo de variables que se deben considerar al momento de la estrategia (como stop-loss, bid-offer, costos transaccionales, etc).

9. Referencias

[1]Huang, J. Z., Huang, W., & Ni, J. (2019). Predicting bitcoin returns using high-dimensional technical indicators. *The Journal of Finance and Data Science*, *5*(3), 140-155.

[2]Dash, R., & Dash, P. K. (2016). A hybrid stock trading framework integrating technical analysis with machine learning techniques. *The Journal of Finance and Data Science*, *2*(1), 42-57.